

ITS WP10 RU Python

Generated by Doxygen 1.8.15

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 Namespace Documentation	11
4.1 charm_testbeam Namespace Reference	11
4.1.1 Detailed Description	12
4.2 chip Namespace Reference	12
4.2.1 Detailed Description	12
4.3 config Namespace Reference	13
4.3.1 Detailed Description	13
4.4 counter_monitor Namespace Reference	13
4.4.1 Detailed Description	13
4.5 dctrl Namespace Reference	13
4.5.1 Detailed Description	14
4.6 drp_bridge Namespace Reference	14
4.6.1 Detailed Description	14
4.7 gbt_sca Namespace Reference	14
4.7.1 Detailed Description	14
4.8 gbt_x_controller Namespace Reference	14
4.8.1 Detailed Description	14
4.9 gbt_x_flow_monitor Namespace Reference	15
4.9.1 Detailed Description	15
4.10 i2c Namespace Reference	15
4.10.1 Detailed Description	15
4.11 mmcm_monitor Namespace Reference	15
4.11.1 Detailed Description	15
4.12 module_includes Namespace Reference	15
4.12.1 Detailed Description	15
4.13 power_unit Namespace Reference	16
4.13.1 Detailed Description	16
4.14 proasic3 Namespace Reference	16
4.14.1 Detailed Description	17
4.15 radiation_monitor Namespace Reference	17
4.15.1 Detailed Description	17
4.16 ru_board Namespace Reference	17
4.16.1 Detailed Description	17
4.17 ru_chip_power Namespace Reference	17

4.17.1 Detailed Description	18
4.18 ru_datapath_monitor Namespace Reference	18
4.18.1 Detailed Description	18
4.19 ru_gbt_packer Namespace Reference	18
4.19.1 Detailed Description	18
4.20 ru_transceiver Namespace Reference	18
4.20.1 Detailed Description	18
4.21 RUv1_regression Namespace Reference	19
4.21.1 Detailed Description	20
4.21.2 Variable Documentation	20
4.21.2.1 formatter	20
4.21.2.2 formatter_ch	21
4.21.2.3 GPIO_SUBSET_MAP	21
4.21.2.4 SIMULATION	21
4.22 sca_ru Namespace Reference	21
4.22.1 Detailed Description	21
4.23 simulation_if Namespace Reference	21
4.23.1 Detailed Description	22
4.23.2 Function Documentation	22
4.23.2.1 chunks()	22
4.24 sysmon Namespace Reference	22
4.24.1 Detailed Description	22
4.25 testbench Namespace Reference	22
4.25.1 Detailed Description	23
4.25.2 Variable Documentation	23
4.25.2.1 GPIO_SENSOR_MASK	23
4.25.2.2 GPIO_SUBSET_MAP	24
4.25.2.3 modules_path	24
4.25.2.4 USB_COMM_EXEC	24
4.26 trigger Namespace Reference	24
4.26.1 Detailed Description	24
4.27 trigger_handler Namespace Reference	24
4.27.1 Detailed Description	25
4.28 trigger_handler_monitor Namespace Reference	25
4.28.1 Detailed Description	25
4.29 usb_communication.communication Namespace Reference	25
4.29.1 Detailed Description	25
4.29.2 Function Documentation	26
4.29.2.1 enqueue_output()	26
4.30 userdefinedexceptions Namespace Reference	26
4.30.1 Detailed Description	26
4.31 validate_scrubbing Namespace Reference	26

4.31.1 Detailed Description	27
4.32 wishbone_module Namespace Reference	27
4.32.1 Detailed Description	27
4.33 wishbone_wait Namespace Reference	27
4.33.1 Detailed Description	28
4.34 ws_i2c_gbtX Namespace Reference	28
4.34.1 Detailed Description	28
4.35 ws_master_monitor Namespace Reference	28
4.35.1 Detailed Description	28
4.36 ws_status Namespace Reference	28
4.36.1 Detailed Description	28
4.37 ws_usb_if Namespace Reference	28
4.37.1 Detailed Description	28
5 Class Documentation	29
5.1 pALPIDE.Addr Class Reference	29
5.1.1 Detailed Description	30
5.2 pALPIDE3.Addr Class Reference	31
5.2.1 Detailed Description	32
5.3 usb_communication.communication.AddressMismatchError Class Reference	32
5.3.1 Detailed Description	32
5.4 pALPIDE.Alpide Class Reference	33
5.4.1 Detailed Description	37
5.4.2 Member Function Documentation	37
5.4.2.1 clear_cmueerrors()	37
5.4.2.2 configure_dtu()	38
5.4.2.3 dump_config()	38
5.4.2.4 getreg_adc_avdd_value()	38
5.4.2.5 getreg_adc_avss_value_()	39
5.4.2.6 getreg_adc_bg2v_value()	39
5.4.2.7 getreg_adc_calibration_value()	39
5.4.2.8 getreg_adc_ctrl()	40
5.4.2.9 getreg_adc_dac_input_value_()	40
5.4.2.10 getreg_adc_dvdd_value()	41
5.4.2.11 getreg_adc_dvss_value()	41
5.4.2.12 getreg_adc_iaux2_value()	42
5.4.2.13 getreg_adc_ibias_value()	42
5.4.2.14 getreg_adc_idb_value()	42
5.4.2.15 getreg_adc_iref_value()	43
5.4.2.16 getreg_adc_ireset_value()	43
5.4.2.17 getreg_adc_ithr_value()	43
5.4.2.18 getreg_adc_t2v_value()	44

5.4.2.19	getreg_adc_vcasn2_value()	44
5.4.2.20	getreg_adc_vcasn_value()	44
5.4.2.21	getreg_adc_vcasn_value()	45
5.4.2.22	getreg_adc_vclip_value()	45
5.4.2.23	getreg_adc_vpulseh_value()	45
5.4.2.24	getreg_adc_vpulsel_value()	46
5.4.2.25	getreg_adc_vresetd_value()	46
5.4.2.26	getreg_adc_vresetp_value()	46
5.4.2.27	getreg_adc_vtemp_value()	47
5.4.2.28	getreg_analog_monitor_and_override()	47
5.4.2.29	getreg_buffer_bypass()	48
5.4.2.30	getreg_busy_min_width()	48
5.4.2.31	getreg_cmd()	49
5.4.2.32	getreg_cmu_and_dmu_cfg()	49
5.4.2.33	getreg_cmu_and_dmu_status()	50
5.4.2.34	getreg_dac_settings_cmu_io_buffers()	50
5.4.2.35	getreg_dac_settings_dclk_and_mclk_io_buffers()	50
5.4.2.36	getreg_disable_regions_lsbs()	51
5.4.2.37	getreg_disable_regions_msbs()	51
5.4.2.38	getreg_dmu_data_fifo_lsbs()	52
5.4.2.39	getreg_dmu_data_fifo_msbs()	52
5.4.2.40	getreg_dtu_cfg()	52
5.4.2.41	getreg_dtu_dacs()	53
5.4.2.42	getreg_dtu_pll_lock_1()	53
5.4.2.43	getreg_dtu_pll_lock_2()	54
5.4.2.44	getreg_dtu_test_1()	54
5.4.2.45	getreg_dtu_test_2()	55
5.4.2.46	getreg_dtu_test_3()	55
5.4.2.47	getreg_fromu_cfg_1()	55
5.4.2.48	getreg_fromu_cfg_2()	56
5.4.2.49	getreg_fromu_cfg_3()	56
5.4.2.50	getreg_fromu_pulsing1()	57
5.4.2.51	getreg_fromu_pulsing_2()	57
5.4.2.52	getreg_fromu_status_1()	57
5.4.2.53	getreg_fromu_status_2()	58
5.4.2.54	getreg_fromu_status_3()	58
5.4.2.55	getreg_fromu_status_4()	58
5.4.2.56	getreg_fromu_status_5()	59
5.4.2.57	getreg_IAUX2()	59
5.4.2.58	getreg_IBIAS()	59
5.4.2.59	getreg_IDB()	60
5.4.2.60	getreg_IRESET()	60

5.4.2.61 getreg_ITHR()	60
5.4.2.62 getreg_mode_ctrl()	61
5.4.2.63 getreg_pixel_cfg()	61
5.4.2.64 getreg_VCASN()	62
5.4.2.65 getreg_VCASN2()	62
5.4.2.66 getreg_VCASP()	62
5.4.2.67 getreg_VCLIP()	63
5.4.2.68 getreg_VPULSEH()	63
5.4.2.69 getreg_VPULSEL()	63
5.4.2.70 getreg_VRESETD()	64
5.4.2.71 getreg_VRESETP()	64
5.4.2.72 getreg_VTEMP()	64
5.4.2.73 init_pixel_matrix()	65
5.4.2.74 initialize()	65
5.4.2.75 initialize_readout()	65
5.4.2.76 initialize_readout_slave()	65
5.4.2.77 mask_all_pixels()	66
5.4.2.78 mask_row()	66
5.4.2.79 pattern30()	66
5.4.2.80 propagate_clock()	66
5.4.2.81 propagate_comma()	67
5.4.2.82 propagate_data()	67
5.4.2.83 propagate_prbs()	67
5.4.2.84 pulse()	68
5.4.2.85 pulse_all_pixels_disable()	68
5.4.2.86 pulse_all_pixels_enable()	68
5.4.2.87 pulse_pixel_enable()	68
5.4.2.88 pulse_row_disable()	69
5.4.2.89 pulse_row_enable()	69
5.4.2.90 region_control_register_mask_all_double_columns()	69
5.4.2.91 region_control_register_pulse_gating()	69
5.4.2.92 region_control_register_unmask_all_double_columns()	70
5.4.2.93 reset()	70
5.4.2.94 reset_pll()	70
5.4.2.95 set_default_variables()	70
5.4.2.96 set_xoff()	71
5.4.2.97 setreg_adc_ctrl()	71
5.4.2.98 setreg_adc_dac_input_value_()	72
5.4.2.99 setreg_analog_monitor_and_override()	72
5.4.2.100 setreg_buffer_bypass()	73
5.4.2.101 setreg_busy_min_width()	74
5.4.2.102 setreg_cmd()	74

5.4.2.103 setreg_cmu_and_dmu_cfg()	75
5.4.2.104 setreg_dac_settings_cmu_io_buffers()	76
5.4.2.105 setreg_dac_settings_dclk_and_mclk_io_buffers()	76
5.4.2.106 setreg_disable_regions_lsbs()	77
5.4.2.107 setreg_disable_regions_msbs()	77
5.4.2.108 setreg_dtu_cfg()	77
5.4.2.109 setreg_dtu_dacs()	78
5.4.2.110 setreg_dtu_pll_lock_2()	78
5.4.2.111 setreg_dtu_test_1()	79
5.4.2.112 setreg_dtu_test_2()	80
5.4.2.113 setreg_dtu_test_3()	80
5.4.2.114 setreg_fromu_cfg_1()	81
5.4.2.115 setreg_fromu_cfg_2()	81
5.4.2.116 setreg_fromu_cfg_3()	82
5.4.2.117 setreg_fromu_pulsing1()	82
5.4.2.118 setreg_fromu_pulsing_2()	82
5.4.2.119 setreg_IAUX2()	83
5.4.2.120 setreg_IBIAS()	83
5.4.2.121 setreg_IDB()	84
5.4.2.122 setreg_IRESET()	84
5.4.2.123 setreg_ITHR()	84
5.4.2.124 setreg_mode_ctrl()	85
5.4.2.125 setreg_pixel_cfg()	86
5.4.2.126 setreg_VCASN()	86
5.4.2.127 setreg_VCASN2()	86
5.4.2.128 setreg_VCASP()	87
5.4.2.129 setreg_VCLIP()	87
5.4.2.130 setreg_VPULSEH()	88
5.4.2.131 setreg_VPULSEL()	88
5.4.2.132 setreg_VRESETD()	88
5.4.2.133 setreg_VRESETP()	89
5.4.2.134 setreg_VTEMP()	89
5.4.2.135 trigger()	90
5.4.2.136 unmask_all_pixels()	90
5.4.2.137 unmask_col()	90
5.4.2.138 unmask_pixel()	90
5.4.2.139 unmask_reg()	91
5.4.2.140 unmask_row()	91
5.4.2.141 write_read_test()	91
5.5 config.Analysis Class Reference	91
5.5.1 Detailed Description	92
5.5.2 Member Function Documentation	92

5.5.2.1 set_basepath()	92
5.6 beam_shutter.BeamShutter Class Reference	92
5.6.1 Detailed Description	93
5.6.2 Constructor & Destructor Documentation	93
5.6.2.1 __init__()	93
5.6.3 Member Function Documentation	93
5.6.3.1 beam_off()	93
5.6.3.2 beam_on()	94
5.6.3.3 beam_shutter_2_off()	94
5.6.3.4 beam_shutter_2_on()	94
5.7 charm_testbeam.BeamShutterDataPoint Class Reference	94
5.8 analyse_bitfile.BitFileAnalyser Class Reference	95
5.8.1 Constructor & Destructor Documentation	95
5.8.1.1 __init__()	95
5.8.2 Member Function Documentation	96
5.8.2.1 analyse_bitfile()	96
5.9 trigger.BitMap Class Reference	96
5.9.1 Detailed Description	96
5.10 proasic3.CcActiveState Class Reference	97
5.10.1 Detailed Description	97
5.11 proasic3.CcCmdOpcode Class Reference	97
5.11.1 Detailed Description	98
5.12 proasic3.CcStatus Class Reference	98
5.12.1 Detailed Description	98
5.13 charm_trigger.CharmTrigger Class Reference	98
5.13.1 Member Function Documentation	99
5.13.1.1 get_last_spill()	99
5.14 chip.Chip Class Reference	99
5.14.1 Detailed Description	100
5.14.2 Member Function Documentation	100
5.14.2.1 configure_dtu()	101
5.14.2.2 dump_config()	101
5.14.2.3 initialize()	101
5.14.2.4 initialize_readout()	101
5.14.2.5 is_inner_barrel()	102
5.14.2.6 is_outer_barrel_master()	102
5.14.2.7 is_outer_barrel_slave()	102
5.14.2.8 propagate_clock()	102
5.14.2.9 propagate_comma()	103
5.14.2.10 propagate_data()	103
5.14.2.11 propagate_prbs()	103
5.14.2.12 read_reg()	104

5.14.2.13 read_region_reg()	104
5.14.2.14 reset()	104
5.14.2.15 reset_pll()	104
5.14.2.16 set_xoff()	105
5.14.2.17 trigger()	105
5.14.2.18 write_opcode()	105
5.14.2.19 write_read_test()	106
5.14.2.20 write_reg()	106
5.14.2.21 write_region_reg()	106
5.15 userdefinedexceptions.ChipError Class Reference	107
5.15.1 Detailed Description	107
5.16 userdefinedexceptions.ChipidMismatchError Class Reference	107
5.16.1 Detailed Description	108
5.17 ru_chip_power.ChipPower Class Reference	108
5.17.1 Detailed Description	108
5.17.2 Member Function Documentation	108
5.17.2.1 dump_config()	109
5.17.2.2 enable_voltages()	109
5.17.2.3 init_ADC()	109
5.17.2.4 initialize()	109
5.17.2.5 log_currents()	110
5.17.2.6 read_values()	110
5.17.2.7 read_values_raw()	110
5.17.2.8 readreg_powerbank_switch()	110
5.17.2.9 set_voltages()	110
5.18 power_supplies.ChipPowerSupply Class Reference	111
5.18.1 Detailed Description	111
5.18.2 Member Function Documentation	111
5.18.2.1 power_on()	111
5.19 power_supplies.ChipPowerSupplyHameg Class Reference	111
5.19.1 Member Function Documentation	112
5.19.1.1 power_on()	112
5.20 power_supplies.ChipPowerSupplyHamegStave Class Reference	112
5.20.1 Member Function Documentation	113
5.20.1.1 power_on()	113
5.21 power_supplies.ChipPowerSupplyInternal Class Reference	113
5.21.1 Member Function Documentation	114
5.21.1.1 power_on()	114
5.22 power_supplies.ChipPowerSupplyMixed Class Reference	114
5.22.1 Detailed Description	115
5.22.2 Member Function Documentation	115
5.22.2.1 power_on()	115

5.23 pALPIDE.CommandRegisterOpcode Class Reference	115
5.23.1 Detailed Description	116
5.24 usb_communication.communication.Communication Class Reference	116
5.24.1 Detailed Description	117
5.24.2 Member Function Documentation	117
5.24.2.1 diagnose_read_results()	117
5.24.2.2 disable_log()	117
5.24.2.3 disable_r derr_exception()	117
5.24.2.4 enable_log()	118
5.24.2.5 enable_r derr_exception()	118
5.24.2.6 flush()	118
5.24.2.7 read_dp2()	118
5.24.2.8 read_dp3()	118
5.24.2.9 read_reg()	119
5.24.2.10 read_results()	119
5.24.2.11 register_read()	119
5.24.2.12 register_write()	119
5.24.2.13 single_read()	120
5.24.2.14 single_write()	120
5.24.2.15 stop()	120
5.24.2.16 write_reg()	120
5.25 power_supplies.CompositeChipPowerSupply Class Reference	121
5.25.1 Member Function Documentation	121
5.25.1.1 power_on()	121
5.26 ru_board.ConfigurationMemoryMapping Class Reference	122
5.26.1 Detailed Description	122
5.27 ru_chip_power.ConversionParameter Class Reference	122
5.27.1 Detailed Description	122
5.27.2 Member Function Documentation	122
5.27.2.1 step_to_value()	123
5.27.2.2 value_to_step()	123
5.28 trigger_handler_monitor.CountersBitMapping Class Reference	123
5.28.1 Detailed Description	124
5.29 ru_datapath_monitor.DatapathMonitor Class Reference	124
5.29.1 Member Function Documentation	125
5.29.1.1 latch_counters()	125
5.29.1.2 read_all_counters()	125
5.29.1.3 read_counters()	125
5.29.1.4 reset_counters()	125
5.29.1.5 reset_lane_counters()	126
5.29.2 Member Data Documentation	126
5.29.2.1 wb_register_mapping	126

5.30 ru_datapath_monitor.DatapathMonitorDual Class Reference	126
5.30.1 Detailed Description	127
5.30.2 Member Function Documentation	127
5.30.2.1 latch_counters()	127
5.30.2.2 read_all_counters()	127
5.30.2.3 read_counters()	127
5.30.2.4 reset_counters()	128
5.30.2.5 reset_lane_counters()	128
5.31 charm_testbeam.DataPoint Class Reference	128
5.31.1 Detailed Description	130
5.32 validate_scrubbing.DataPoint Class Reference	130
5.32.1 Detailed Description	131
5.33 userdefinedexceptions.DataReadbackMismatchError Class Reference	131
5.33.1 Detailed Description	132
5.34 dctrl.Dctrl Class Reference	132
5.34.1 Detailed Description	133
5.34.2 Member Function Documentation	133
5.34.2.1 busy_force()	133
5.34.2.2 busy_force_release()	133
5.34.2.3 disable_manchester_tx()	134
5.34.2.4 dump_config()	134
5.34.2.5 enable_manchester_tx()	134
5.34.2.6 force_phase()	134
5.34.2.7 get_auto_phase_offset()	135
5.34.2.8 get_busy_transceiver_mask()	135
5.34.2.9 get_busy_transceiver_status()	135
5.34.2.10 get_counters()	135
5.34.2.11 get_dclk_mask()	135
5.34.2.12 get_dclk_parallel()	136
5.34.2.13 get_dctrl_mask()	136
5.34.2.14 get_idelays()	136
5.34.2.15 get_input()	136
5.34.2.16 get_manchester_rx_detected()	136
5.34.2.17 get_manchester_tx()	137
5.34.2.18 get_mismatch()	137
5.34.2.19 get_phase()	137
5.34.2.20 get_wait_cycles()	137
5.34.2.21 read_chip_reg()	137
5.34.2.22 readreg_write_phase()	138
5.34.2.23 release_phase_force()	138
5.34.2.24 reset_counters()	138
5.34.2.25 set_auto_phase_offset()	138

5.34.2.26 set_busy_transceiver_mask()	138
5.34.2.27 set_dclk_mask()	139
5.34.2.28 set_dclk_parallel()	139
5.34.2.29 set_dctrl_mask()	139
5.34.2.30 set_idelays()	139
5.34.2.31 set_input()	140
5.34.2.32 set_wait_cycles()	140
5.34.2.33 write_chip_opcode()	140
5.34.2.34 write_chip_reg()	140
5.35 RUv1_regression.DctrlBaseTest Class Reference	141
5.36 drp_bridge.DrpBridge Class Reference	141
5.36.1 Member Function Documentation	141
5.36.1.1 read_drp()	141
5.36.1.2 write_drp()	142
5.37 sysmon.DrpSysmonAddress Class Reference	142
5.37.1 Detailed Description	143
5.38 sysmon.DrpSysmonConfigMasks Class Reference	144
5.39 sysmon.DrpSysmonFlagMasks Class Reference	145
5.40 beam_shutter.DummyBeamShutter Class Reference	145
5.40.1 Member Function Documentation	146
5.40.1.1 beam_off()	146
5.40.1.2 beam_on()	146
5.41 power_supplies.DummyPowerSupply Class Reference	146
5.41.1 Detailed Description	147
5.41.2 Member Function Documentation	147
5.41.2.1 power_on()	147
5.42 proasic3.EccCmdFlags Class Reference	147
5.42.1 Detailed Description	147
5.43 proasic3.EccStatus Class Reference	148
5.43.1 Detailed Description	148
5.44 events.Event Class Reference	148
5.45 events.EventDecoder Class Reference	149
5.45.1 Member Function Documentation	149
5.45.1.1 get_eventcount_map()	149
5.45.1.2 process()	150
5.45.1.3 set_max_errors()	150
5.46 events.EventPlotter Class Reference	150
5.47 validate_scrubbing.ExitStatus Class Reference	151
5.48 charm_testbeam.ExitStatus Class Reference	151
5.49 userdefinedexceptions.ExternalShutdownError Class Reference	152
5.49.1 Detailed Description	152
5.50 ru_eyescan.EyeScan Class Reference	153

5.50.1 Detailed Description	153
5.50.2 Member Function Documentation	153
5.50.2.1 eye_scan()	154
5.50.2.2 eye_scan_adaptive()	154
5.50.2.3 initialize()	154
5.50.2.4 resume_eye_scan()	155
5.51 ru_eyescan.EyeScanGth Class Reference	155
5.51.1 Detailed Description	156
5.51.2 Member Function Documentation	156
5.51.2.1 get_vertical_range()	156
5.52 ru_eyescan.EyeScanGtx Class Reference	156
5.52.1 Detailed Description	157
5.53 proasic3.FifoState Class Reference	157
5.53.1 Detailed Description	158
5.54 proasic3.FifoWriterCmdOpcode Class Reference	158
5.54.1 Detailed Description	158
5.55 proasic3.FifoWriterStatus Class Reference	158
5.55.1 Detailed Description	159
5.56 flash_test.FlashChecker Class Reference	159
5.56.1 Detailed Description	160
5.56.2 Constructor & Destructor Documentation	160
5.56.2.1 __init__()	160
5.56.3 Member Function Documentation	160
5.56.3.1 analyse_block()	160
5.56.3.2 set_first_byte_bug()	160
5.56.3.3 set_is_inverted()	161
5.57 proasic3.FlashCmdOpcode Class Reference	161
5.57.1 Detailed Description	161
5.58 proasic3.FlashFsmBlockEraseControllerState Class Reference	161
5.58.1 Detailed Description	162
5.59 proasic3.FlashFsmNandControllerState Class Reference	163
5.59.1 Detailed Description	163
5.60 proasic3.FlashFsmPageReadController Class Reference	163
5.60.1 Detailed Description	164
5.61 proasic3.FlashFsmPageWriteController Class Reference	165
5.61.1 Detailed Description	166
5.62 proasic3.FlashFsmReadIdControllerState Class Reference	166
5.62.1 Detailed Description	166
5.63 proasic3.FlashSelectIcOpcode Class Reference	167
5.63.1 Detailed Description	167
5.64 proasic3.FlashStatus Class Reference	167
5.64.1 Detailed Description	168

5.65 simulation_if.GbtCruGbtxBridge Class Reference	168
5.65.1 Detailed Description	168
5.65.2 Member Function Documentation	169
5.65.2.1 clear_gbt_cru()	169
5.65.2.2 clear_gbt_x()	169
5.65.2.3 flush_gbt_cru()	169
5.65.2.4 flush_gbt_x()	169
5.65.2.5 open()	170
5.66 ru_gbt_packer.GbtPacker Class Reference	170
5.66.1 Detailed Description	171
5.66.2 Member Function Documentation	171
5.66.2.1 dump_config()	171
5.66.2.2 initialize()	171
5.66.2.3 update_masks()	172
5.67 ru_gbt_packer.GbtPackerMonitor Class Reference	172
5.67.1 Member Function Documentation	173
5.67.1.1 latch_counters()	173
5.67.1.2 read_counters()	173
5.67.1.3 reset_counters()	173
5.67.2 Member Data Documentation	173
5.67.2.1 wb_register_mapping	173
5.68 gbt_x_controller.GBTxController Class Reference	174
5.68.1 Detailed Description	174
5.68.2 Member Function Documentation	174
5.68.2.1 disable_controller()	175
5.68.2.2 dump_config()	175
5.68.2.3 enable_controller()	175
5.68.2.4 get_bitslip_rx()	175
5.68.2.5 get_bitslip_tx()	175
5.68.2.6 get_idelay()	176
5.68.2.7 get_loss_lock_flag()	176
5.68.2.8 get_mismatch()	176
5.68.2.9 get_tx1_pattern()	176
5.68.2.10 get_tx_pattern()	176
5.68.2.11 reset_gbt_x_asic()	177
5.68.2.12 rst_loss_lock_flag()	177
5.68.2.13 set_bitslip_rx()	177
5.68.2.14 set_bitslip_tx()	177
5.68.2.15 set_idelay()	178
5.68.2.16 set_tx1_pattern()	178
5.68.2.17 set_tx_pattern()	178
5.69 gbt_x_flow_monitor.GbtXFlowMonitor Class Reference	178

5.69.1 Detailed Description	179
5.69.2 Member Function Documentation	179
5.69.2.1 dump_config()	179
5.69.2.2 latch_counters()	179
5.69.2.3 read_counters()	180
5.69.2.4 reset_counters()	180
5.70 simulation_if.GbtxSim Class Reference	180
5.70.1 Detailed Description	181
5.70.2 Member Function Documentation	181
5.70.2.1 clear()	181
5.70.2.2 flush_gbtx()	181
5.70.2.3 get_data()	182
5.70.2.4 get_swt()	182
5.70.2.5 open()	182
5.70.2.6 readFromGbt()	182
5.70.2.7 writeToGbt()	182
5.71 ru_transceiver.GpioFrontend Class Reference	183
5.71.1 Detailed Description	184
5.71.2 Member Function Documentation	184
5.71.2.1 align_transceivers()	184
5.71.2.2 enable_alignment()	184
5.71.2.3 enable_data()	184
5.71.2.4 enable_prbs()	185
5.71.2.5 enable_realign()	185
5.71.2.6 get_transceivers()	185
5.71.2.7 initialize()	185
5.71.2.8 is_aligned()	185
5.71.2.9 read_prbs_counter()	186
5.71.2.10 reset_prbs_counter()	186
5.71.2.11 scan_idelays()	186
5.71.2.12 set_lane_chip_mask()	186
5.71.2.13 set_transceivers()	187
5.72 ru_transceiver.GthFrontend Class Reference	187
5.72.1 Detailed Description	188
5.72.2 Member Function Documentation	188
5.72.2.1 align_transceivers()	188
5.72.2.2 enable_alignment()	188
5.72.2.3 enable_data()	189
5.72.2.4 enable_prbs()	189
5.72.2.5 get_transceivers()	189
5.72.2.6 initialize()	189
5.72.2.7 is_aligned()	190

5.72.2.8 is_cdr_locked()	190
5.72.2.9 is_reset_done()	190
5.72.2.10 read_drp()	190
5.72.2.11 read_prbs_counter()	190
5.72.2.12 readback_all_drp()	191
5.72.2.13 reset_prbs_counter()	191
5.72.2.14 reset_receivers()	191
5.72.2.15 set_transceivers()	191
5.72.2.16 write_drp()	192
5.73 hameg.Hameg Class Reference	192
5.73.1 Detailed Description	192
5.73.2 Member Function Documentation	193
5.73.2.1 activate_channel()	193
5.73.2.2 activate_channels()	193
5.73.2.3 activate_output()	193
5.73.2.4 configure_channel()	193
5.73.2.5 deactivate_channel()	194
5.73.2.6 deactivate_channels()	194
5.73.2.7 deactivate_output()	194
5.73.2.8 get_channel_config()	194
5.73.2.9 get_channel_config_all()	194
5.73.2.10 get_current()	195
5.73.2.11 get_fuse_triggered()	195
5.73.2.12 get_model()	195
5.73.2.13 get_power()	195
5.73.2.14 get_power_all()	195
5.73.2.15 get_voltage()	196
5.73.2.16 is_channel_on()	196
5.73.2.17 setup_logging()	196
5.74 i2c.I2CModule Class Reference	196
5.74.1 Detailed Description	197
5.74.2 Member Function Documentation	197
5.74.2.1 dump_config()	197
5.75 jcm_control_server.JCM Class Reference	197
5.75.1 Detailed Description	198
5.75.2 Member Function Documentation	198
5.75.2.1 is_stopped()	198
5.75.2.2 read_messages()	198
5.75.2.3 rename_readback_file()	199
5.75.2.4 start_blind_scrubber()	199
5.75.2.5 start_calibrate_device()	199
5.75.2.6 start_full_configure()	199

5.75.2.7 start_random_fault_injection()	200
5.75.2.8 start_readback()	200
5.75.2.9 stop_blind_scrubber()	200
5.75.2.10 stop_random_fault_injection()	200
5.75.2.11 test_function()	201
5.76 jcm_control_server.JcmControlClient Class Reference	201
5.76.1 Detailed Description	202
5.76.2 Member Function Documentation	202
5.76.2.1 is_readback_finished()	202
5.76.2.2 is_stopped()	202
5.76.2.3 read_calibrate_device_results()	202
5.76.2.4 read_fault_injection_results()	202
5.76.2.5 read_full_configure_results()	203
5.76.2.6 read_messages()	203
5.76.2.7 read_start_blind_scrubber_results()	203
5.76.2.8 start_blind_scrubber()	203
5.76.2.9 start_calibrate_device()	203
5.76.2.10 start_full_configure()	204
5.76.2.11 start_random_fault_injection()	204
5.76.2.12 start_readback()	204
5.76.2.13 stop_blind_scrubber()	204
5.76.2.14 stop_random_fault_injection()	204
5.76.2.15 test_function()	205
5.77 jcm_control_server.JcmControlServer Class Reference	205
5.77.1 Detailed Description	205
5.77.2 Constructor & Destructor Documentation	205
5.77.2.1 __init__()	205
5.78 mmcm_monitor.MmcmMonitor Class Reference	206
5.78.1 Detailed Description	206
5.78.2 Constructor & Destructor Documentation	206
5.78.2.1 __init__()	206
5.78.3 Member Function Documentation	207
5.78.3.1 dump_config()	207
5.78.3.2 get_counters()	207
5.78.3.3 get_lock_counter()	207
5.78.3.4 get_mismatch()	207
5.79 mmcm_monitor.MmcmMonitorAddress Class Reference	208
5.79.1 Detailed Description	208
5.80 usb_communication.communication.NetUsbComm Class Reference	208
5.80.1 Detailed Description	209
5.80.2 Member Function Documentation	209
5.80.2.1 close_connections()	209

5.80.2.2 stop()	210
5.81 usb_communication.communication.NetUsbTimeoutException Class Reference	210
5.81.1 Detailed Description	210
5.82 jcm_control_server.NoReadBackFileError Class Reference	211
5.82.1 Detailed Description	211
5.82.2 Member Function Documentation	211
5.82.2.1 log_info()	211
5.83 pALPIDE.Opcodes Class Reference	212
5.83.1 Detailed Description	212
5.84 pALPIDE3.Opcodes Class Reference	212
5.84.1 Detailed Description	213
5.85 proasic3.Pa3Register Class Reference	213
5.86 pALPIDE3.PAlpide3 Class Reference	215
5.86.1 Detailed Description	218
5.86.2 Member Function Documentation	218
5.86.2.1 dump_config()	219
5.86.2.2 getreg_analog_monitor_and_override()	219
5.86.2.3 getreg_buffer_bypass()	220
5.86.2.4 getreg_busy_min_width()	220
5.86.2.5 getreg_cmd()	220
5.86.2.6 getreg_cmu_and_dmu_cfg()	221
5.86.2.7 getreg_cmu_errors_counter()	221
5.86.2.8 getreg_dac_settings_cmu_io_buffers()	222
5.86.2.9 getreg_dac_settings_dclk_and_mclk_io_buffers()	222
5.86.2.10 getreg_disable_regions_lsbs()	223
5.86.2.11 getreg_disableof_regions_msbs()	223
5.86.2.12 getreg_dmu_and_tru_state()	223
5.86.2.13 getreg_dtu_cfg()	224
5.86.2.14 getreg_dtu_dacs()	224
5.86.2.15 getreg_dtu_pll_lock_1()	225
5.86.2.16 getreg_dtu_pll_lock_2()	225
5.86.2.17 getreg_dtu_test_1()	226
5.86.2.18 getreg_dtu_test_2()	226
5.86.2.19 getreg_dtu_test_3()	227
5.86.2.20 getreg_fromu_cfg_1()	227
5.86.2.21 getreg_fromu_cfg_2()	228
5.86.2.22 getreg_fromu_pulsing1()	228
5.86.2.23 getreg_fromu_pulsing_2()	228
5.86.2.24 getreg_fromu_status_1()	229
5.86.2.25 getreg_fromu_status_2()	229
5.86.2.26 getreg_fuses_read_lsbs()	229
5.86.2.27 getreg_fuses_read_msbs()	230

5.86.2.28	getreg_fuses_write_lsbs()	230
5.86.2.29	getreg_fuses_write_msbs()	230
5.86.2.30	getreg_IAUX2()	231
5.86.2.31	getreg_IBIAS()	231
5.86.2.32	getreg_IDB()	231
5.86.2.33	getreg_IRESET()	232
5.86.2.34	getreg_ITHR()	232
5.86.2.35	getreg_periphery_ctrl()	232
5.86.2.36	getreg_pixel_cfg_1()	233
5.86.2.37	getreg_pixel_cfg_2()	233
5.86.2.38	getreg_pixel_cfg_3()	234
5.86.2.39	getreg_temperature_sensor()	234
5.86.2.40	getreg_VCASN()	235
5.86.2.41	getreg_VCASN2()	235
5.86.2.42	getreg_VCASP()	235
5.86.2.43	getreg_VCLIP()	236
5.86.2.44	getreg_VPULSEH()	236
5.86.2.45	getreg_VPULSEL()	236
5.86.2.46	getreg_VRESETD()	237
5.86.2.47	getreg_VRESETP()	237
5.86.2.48	getreg_VTEMP()	237
5.86.2.49	init_pixel_matrix()	238
5.86.2.50	initialize()	238
5.86.2.51	initialize_readout()	238
5.86.2.52	propagate_clock()	239
5.86.2.53	propagate_comma()	239
5.86.2.54	propagate_data()	239
5.86.2.55	propagate_prbs()	240
5.86.2.56	reset()	240
5.86.2.57	reset_pll()	240
5.86.2.58	set_default_variables()	240
5.86.2.59	set_xoff()	241
5.86.2.60	setreg_analog_monitor_and_override()	241
5.86.2.61	setreg_buffer_bypass()	242
5.86.2.62	setreg_busy_min_width()	242
5.86.2.63	setreg_cmd()	243
5.86.2.64	setreg_cmu_and_dmu_cfg()	243
5.86.2.65	setreg_dac_settings_cmu_io_buffers()	244
5.86.2.66	setreg_dac_settings_dclk_and_mclk_io_buffers()	244
5.86.2.67	setreg_disable_regions_lsbs()	245
5.86.2.68	setreg_disableof_regions_msbs()	245
5.86.2.69	setreg_dtu_cfg()	246

5.86.2.70 setreg_dtu_dacs()	246
5.86.2.71 setreg_dtu_pll_lock_2()	247
5.86.2.72 setreg_dtu_test_1()	247
5.86.2.73 setreg_dtu_test_2()	248
5.86.2.74 setreg_dtu_test_3()	248
5.86.2.75 setreg_fromu_cfg_1()	249
5.86.2.76 setreg_fromu_cfg_2()	249
5.86.2.77 setreg_fromu_pulsing1()	250
5.86.2.78 setreg_fromu_pulsing_2()	250
5.86.2.79 setreg_fuses_write_lsbs()	250
5.86.2.80 setreg_fuses_write_msbs()	251
5.86.2.81 setreg_IAUX2()	251
5.86.2.82 setreg_IBIAS()	252
5.86.2.83 setreg_IDB()	252
5.86.2.84 setreg_IRESET()	252
5.86.2.85 setreg_ITHR()	253
5.86.2.86 setreg_periphery_ctrl()	253
5.86.2.87 setreg_pixel_cfg_1()	254
5.86.2.88 setreg_pixel_cfg_2()	254
5.86.2.89 setreg_pixel_cfg_3()	255
5.86.2.90 setreg_VCASN()	255
5.86.2.91 setreg_VCASN2()	256
5.86.2.92 setreg_VCASP()	256
5.86.2.93 setreg_VCLIP()	256
5.86.2.94 setreg_VPULSEH()	257
5.86.2.95 setreg_VPULSEL()	257
5.86.2.96 setreg_VRESETD()	258
5.86.2.97 setreg_VRESETP()	258
5.86.2.98 setreg_VTEMP()	258
5.86.2.99 trigger()	259
5.87 power_unit.PowerUnit Class Reference	259
5.87.1 Detailed Description	261
5.87.2 Member Function Documentation	261
5.87.2.1 ConfigureBiasADC()	261
5.87.2.2 ConfigurePowerADC()	261
5.87.2.3 ConfigureRTD()	261
5.87.2.4 DisableBiasAll()	262
5.87.2.5 DisablePowerAll()	262
5.87.2.6 EnableBiasAll()	262
5.87.2.7 EnableBiasWithMask()	262
5.87.2.8 EnablePower()	263
5.87.2.9 EnablePowerAll()	263

5.87.2.10 EnablePowerWithMask()	263
5.87.2.11 get_bias_enable_status()	263
5.87.2.12 get_power_enable_status()	264
5.87.2.13 LowerThresholdsToMin()	264
5.87.2.14 RaiseThresholdsToMax()	264
5.87.2.15 ReadBiasADC()	264
5.87.2.16 ReadBiasADCCchannel()	264
5.87.2.17 ReadPowerADC()	265
5.87.2.18 ReadPowerADCCchannel()	265
5.87.2.19 ReadRTD()	265
5.87.2.20 SetBiasVoltage()	265
5.87.2.21 SetPowerVoltage()	266
5.87.2.22 SetPowerVoltageAll()	266
5.87.2.23 SetThreshold()	266
5.87.2.24 SetThresholdAll()	266
5.87.2.25 SetThresholdWithMask()	267
5.88 usb_communication.communication.PrefetchCommunication Class Reference	267
5.88.1 Detailed Description	268
5.88.2 Constructor & Destructor Documentation	268
5.88.2.1 __init__()	268
5.88.3 Member Function Documentation	268
5.88.3.1 get_sequence()	268
5.88.3.2 load_sequence()	268
5.88.3.3 prefetch()	269
5.88.3.4 start_recording()	269
5.88.3.5 stop()	269
5.88.3.6 stop_recording()	269
5.89 proasic3.ProAsic3 Class Reference	270
5.89.1 Detailed Description	271
5.89.2 Member Function Documentation	272
5.89.2.1 dump_config()	272
5.89.2.2 flash_dump_page()	272
5.89.2.3 flash_read_page()	272
5.89.2.4 flash_write_firmware()	273
5.89.2.5 flash_write_parameter_page()	273
5.89.2.6 get_cc_command_register()	273
5.89.2.7 get_clock_config()	274
5.89.2.8 get_clock_status()	274
5.89.2.9 get_version()	274
5.89.2.10 read_fifo_reg()	274
5.89.2.11 read_reg()	274
5.89.2.12 set_cc_command_register()	275

5.89.2.13 set_clock_config()	275
5.89.2.14 set_ecc_command_register()	275
5.89.2.15 set_i2c_channel()	275
5.89.2.16 set_reset()	276
5.89.2.17 set_start_program()	276
5.89.2.18 write_fifo_reg()	276
5.89.2.19 write_reg()	276
5.90 trigger_handler.Pulse_nTrigger Class Reference	277
5.90.1 Detailed Description	277
5.91 usb_communication.communication.PyUsbComm Class Reference	277
5.91.1 Detailed Description	278
5.92 proasic3.RcCmdOpcode Class Reference	278
5.92.1 Detailed Description	278
5.93 proasic3.RcState Class Reference	278
5.93.1 Detailed Description	279
5.94 ru_board.ReadoutBoard Class Reference	279
5.94.1 Member Function Documentation	279
5.94.1.1 flush()	280
5.94.1.2 read()	280
5.94.1.3 read_all()	280
5.94.1.4 write()	280
5.95 sca_ru.RuScaReg Class Reference	281
5.95.1 Detailed Description	281
5.96 ru_board.RUv0_CRU Class Reference	281
5.96.1 Detailed Description	282
5.96.2 Member Function Documentation	283
5.96.2.1 check_git_hash_and_date()	283
5.96.2.2 get_dna_value()	283
5.96.2.3 get_microprocessor_counters()	283
5.96.2.4 read_gbt_x_data_from_usb()	283
5.96.2.5 send_end_of_continuous()	284
5.96.2.6 send_end_of_triggered()	284
5.96.2.7 send_heartbeat()	284
5.96.2.8 send_start_of_continuous()	284
5.96.2.9 send_start_of_triggered()	285
5.96.2.10 send_trigger()	285
5.96.2.11 wait()	285
5.96.2.12 write_gbt_x()	285
5.96.3 Member Data Documentation	286
5.96.3.1 SCA_ADC_COUNTERS	286
5.97 ru_board.RUv1 Class Reference	286
5.97.1 Detailed Description	288

5.97.2 Member Function Documentation	288
5.97.2.1 check_git_hash_and_date()	288
5.97.2.2 enable_chip2connector_lut()	288
5.97.2.3 get_chip2connector_lut()	288
5.97.2.4 get_dna_value()	289
5.97.2.5 set_chip2connector_lut()	289
5.97.2.6 wait()	289
5.98 gbt_sca.Sca Class Reference	289
5.98.1 Detailed Description	290
5.98.2 Member Function Documentation	290
5.98.2.1 init_communication()	290
5.98.2.2 initialize_i2c_channel()	291
5.98.2.3 read_adc_channel()	291
5.98.2.4 read_gbt_x_register()	291
5.98.2.5 set_adc_channel()	291
5.98.2.6 set_gpio()	292
5.98.2.7 set_gpio_direction()	292
5.98.2.8 set_i2c_w_ctrl_reg()	292
5.98.2.9 set_xcku_reset()	292
5.98.2.10 write_gbt_x_register()	293
5.98.2.11 write_gpio()	293
5.99 sca_ru.Sca_RU Class Reference	293
5.99.1 Detailed Description	294
5.99.2 Member Function Documentation	294
5.99.2.1 clear_status_latch()	294
5.99.2.2 get_status_latch()	294
5.99.2.3 init_communication()	294
5.100 gbt_sca.ScaBadErrorFlagError Class Reference	295
5.100.1 Detailed Description	295
5.100.2 Member Function Documentation	295
5.100.2.1 log_info()	295
5.101 gbt_sca.ScaChannel Class Reference	296
5.101.1 Detailed Description	296
5.102 gbt_sca.ScaErrorFlag Class Reference	296
5.102.1 Detailed Description	297
5.103 gbt_sca.ScaGpio Class Reference	297
5.104 gbt_sca.Scal2cBadStatusError Class Reference	298
5.104.1 Detailed Description	298
5.104.2 Member Function Documentation	298
5.104.2.1 log_info()	298
5.105 gbt_sca.Scal2cCmd Class Reference	299
5.106 gbt_sca.Scal2cSpeed Class Reference	299

5.107 gbt_sca.Scal2cStatusFlag Class Reference	300
5.107.1 Detailed Description	300
5.108 charm_testbeam.ScrubbingCyclesDataPoint Class Reference	300
5.109 validate_scrubbing.ScrubbingCyclesDataPoint Class Reference	301
5.110 testbench.SensorMatrixPattern Class Reference	301
5.111 charm_testbeam.SensorPoweringScheme Class Reference	302
5.112 simulation_if.SimulationServer Class Reference	302
5.112.1 Detailed Description	303
5.112.2 Member Function Documentation	303
5.112.2.1 start()	303
5.113 proasic3.SmapCmdOpcode Class Reference	303
5.113.1 Detailed Description	304
5.114 proasic3.SmapStatus Class Reference	304
5.114.1 Detailed Description	304
5.115 sysmon.Sysmon Class Reference	304
5.115.1 Detailed Description	305
5.115.2 Member Function Documentation	305
5.115.2.1 dump_config()	305
5.115.2.2 get_drp_data()	306
5.115.2.3 get_mismatch()	306
5.115.2.4 set_drp_address()	306
5.115.2.5 set_drp_data()	306
5.116 validate_scrubbing.TestbeamTest Class Reference	307
5.116.1 Member Function Documentation	308
5.116.1.1 check_status()	308
5.116.1.2 readback_jcm()	308
5.117 charm_testbeam.TestbeamTest Class Reference	308
5.117.1 Member Function Documentation	310
5.117.1.1 check_status()	310
5.117.1.2 manual_single_scrub()	310
5.117.1.3 send_stimuli()	311
5.118 testbench.Testbench Class Reference	311
5.118.1 Detailed Description	312
5.118.2 Member Function Documentation	313
5.118.2.1 chip()	313
5.118.2.2 setup_boards()	313
5.118.2.3 setup_comm_cru()	313
5.118.2.4 setup_comm_rdo()	313
5.118.2.5 setup_comms()	314
5.118.2.6 test_chips()	314
5.118.2.7 test_chips_continuous()	314
5.118.2.8 test_chips_fast()	314

5.119 RUv1_regression.TestcaseBase Class Reference	315
5.119.1 Member Function Documentation	316
5.119.1.1 send_end_of_continuous()	316
5.119.1.2 send_end_of_triggered()	316
5.119.1.3 send_idle()	316
5.119.1.4 send_start_of_continuous()	317
5.119.1.5 send_start_of_triggered()	317
5.119.1.6 sync()	317
5.120 RUv1_regression.TestCRUSWTRaceCondition Class Reference	317
5.121 RUv1_regression.TestDatapathMonitorModules Class Reference	318
5.122 RUv1_regression.DctrlBaseTest.TestDctrl Class Reference	319
5.122.1 Member Function Documentation	319
5.122.1.1 test_readback_wait_cycles()	319
5.123 RUv1_regression.TestDctrl0 Class Reference	320
5.124 RUv1_regression.TestDctrl1 Class Reference	320
5.125 RUv1_regression.TestDctrl2 Class Reference	321
5.126 RUv1_regression.TestDctrl3 Class Reference	321
5.127 RUv1_regression.TestDctrl4 Class Reference	322
5.128 RUv1_regression.TestGbtFlowMonitor Class Reference	322
5.128.1 Detailed Description	323
5.129 lab_check_setup.TestHardwareConnections Class Reference	323
5.129.1 Detailed Description	323
5.130 charm_testbeam_check_setup.TestHardwareConnections Class Reference	324
5.130.1 Detailed Description	324
5.131 RUv1_regression.TestI2C Class Reference	324
5.132 charm_testbeam.TestMode Class Reference	325
5.133 validate_scrubbing.TestMode Class Reference	325
5.134 RUv1_regression.TestPythonScripts Class Reference	326
5.135 RUv1_regression.TestScaReset Class Reference	326
5.135.1 Member Function Documentation	327
5.135.1.1 test_reset()	327
5.136 RUv1_regression.TestSimExitHandling Class Reference	327
5.137 charm_testbeam_check_setup.TestSoftware Class Reference	328
5.138 RUv1_regression.TestSysmon Class Reference	328
5.139 RUv1_regression.TestTransceiverFrontend Class Reference	329
5.140 RUv1_regression.TestTransceiverGpioFrontend Class Reference	329
5.141 RUv1_regression.TestTransceiverGthFrontend Class Reference	330
5.141.1 Member Function Documentation	331
5.141.1.1 test_drp()	331
5.142 RUv1_regression.TriggerHandlerBaseTest.TestTriggerHandler Class Reference	331
5.143 RUv1_regression.TestTriggerHandlerContinuousMode Class Reference	332
5.143.1 Detailed Description	332

5.144 RUv1_regression.TestTriggerHandlerNoMode Class Reference	333
5.144.1 Detailed Description	333
5.144.2 Member Function Documentation	333
5.144.2.1 test_mode_status_continuous()	333
5.144.2.2 test_mode_status_triggered()	333
5.145 RUv1_regression.TestTriggerHandlerTriggeredMode Class Reference	334
5.145.1 Detailed Description	334
5.145.2 Member Function Documentation	334
5.145.2.1 test_send_multiple_triggers()	334
5.145.2.2 test_send_pulse()	335
5.145.2.3 test_trigger_gating()	335
5.145.2.4 test_trigger_too_close_reject()	335
5.146 RUv1_regression.TestWb2Fifo Class Reference	335
5.147 RUv1_regression.TestWishboneDualMaster Class Reference	336
5.148 RUv1_regression.TestWishboneMaster Class Reference	337
5.148.1 Detailed Description	337
5.148.2 Member Function Documentation	337
5.148.2.1 test_access_nonexist()	337
5.149 RUv1_regression.TestWishboneSlaves Class Reference	337
5.149.1 Detailed Description	338
5.149.2 Member Function Documentation	339
5.149.2.1 test_gbt_x_flow_monitor()	339
5.149.2.2 test_sysmon()	339
5.149.2.3 test_wsmstr()	339
5.150 RUv1_regression.TestWsRadiationMonitor Class Reference	339
5.150.1 Detailed Description	340
5.151 RUv1_regression.TestWsStatus Class Reference	340
5.151.1 Detailed Description	340
5.152 trigger_handler.TriggerHandler Class Reference	341
5.152.1 Detailed Description	341
5.152.2 Member Function Documentation	341
5.152.2.1 dump_config()	342
5.152.2.2 get_opcode_gating()	342
5.152.2.3 get_operating_mode()	342
5.152.2.4 get_trigger_minimum_distance()	342
5.152.2.5 get_trigger_period()	343
5.152.2.6 get_trigger_source()	343
5.152.2.7 set_opcode_gating()	343
5.152.2.8 set_trigger_minimum_distance()	343
5.152.2.9 set_trigger_period()	344
5.152.2.10 set_trigger_source()	344
5.153 RUv1_regression.TriggerHandlerBaseTest Class Reference	344

5.154 trigger_handler_monitor.TriggerHandlerMonitor Class Reference	344
5.154.1 Detailed Description	345
5.154.2 Member Function Documentation	345
5.154.2.1 dump_config()	345
5.154.2.2 get_counters()	345
5.154.2.3 latch_counters()	345
5.154.2.4 reset_counters()	346
5.155 tti.TTI Class Reference	346
5.155.1 Detailed Description	346
5.155.2 Member Function Documentation	347
5.155.2.1 activate_output_all_channels()	347
5.155.2.2 activate_output_channel()	347
5.155.2.3 activate_output_channels()	347
5.155.2.4 clear_triggered_fuse()	347
5.155.2.5 configure_channel()	348
5.155.2.6 deactivate_output_all_channels()	348
5.155.2.7 deactivate_output_channel()	348
5.155.2.8 deactivate_output_channels()	348
5.155.2.9 get_channel_config()	349
5.155.2.10 get_channel_config_all()	349
5.155.2.11 get_current()	349
5.155.2.12 get_fuse_triggered()	349
5.155.2.13 get_model()	349
5.155.2.14 get_power()	350
5.155.2.15 get_power_all()	350
5.155.2.16 get_voltage()	350
5.155.2.17 is_channel_on()	350
5.155.2.18 setup_logging()	350
5.156 usb_communication.communication.UsbCommServer Class Reference	351
5.156.1 Detailed Description	351
5.156.2 Member Function Documentation	351
5.156.2.1 is_stopped()	351
5.156.2.2 read_messages()	351
5.156.2.3 start()	352
5.156.2.4 stop()	352
5.157 simulation_if.UsbCommSim Class Reference	352
5.157.1 Detailed Description	353
5.157.2 Member Function Documentation	353
5.157.2.1 clear()	353
5.157.2.2 flush_dp0()	353
5.157.2.3 open()	353
5.158 simulation_if.Wb2GbtComm Class Reference	354

5.158.1 Detailed Description	354
5.158.2 Member Function Documentation	354
5.158.2.1 send_trigger()	354
5.159 power_unit.Wbl2cPuAddress Class Reference	355
5.160 wishbone_module.WishboneModule Class Reference	356
5.160.1 Detailed Description	356
5.160.2 Member Function Documentation	357
5.160.2.1 dump_config()	357
5.160.2.2 firmware_wait()	357
5.160.2.3 flush()	357
5.160.2.4 read()	357
5.160.2.5 read_all()	358
5.160.2.6 write()	358
5.161 usb_communication.communication.WishboneReadError Class Reference	358
5.161.1 Detailed Description	359
5.161.2 Member Function Documentation	359
5.161.2.1 print_info()	359
5.162 ru_wb2fifo.WishboneToFifo Class Reference	359
5.162.1 Detailed Description	360
5.163 wishbone_wait.WishboneWait Class Reference	360
5.163.1 Detailed Description	360
5.163.2 Member Function Documentation	360
5.163.2.1 dump_config()	361
5.163.2.2 get_counters()	361
5.163.2.3 rst_ctrl_cntrs()	361
5.163.2.4 wait()	361
5.164 counter_monitor.WsCounterMonitor Class Reference	362
5.164.1 Detailed Description	362
5.164.2 Member Function Documentation	362
5.164.2.1 latch_all_counters()	362
5.164.2.2 reset_all_counters()	363
5.164.2.3 reset_counter()	363
5.165 counter_monitor.WsCounterMonitorAddress Class Reference	363
5.166 dctrl.WsDctrlAddress Class Reference	364
5.166.1 Detailed Description	365
5.167 gbt_x_controller.WsGbt_xControllerAddress Class Reference	365
5.167.1 Detailed Description	366
5.168 gbt_x_flow_monitor.WsGbt_xFlowMonitorAddress Class Reference	366
5.168.1 Detailed Description	367
5.169 gbt_x_flow_monitor.WsGbt_xFlowMonitorAddress_bit Class Reference	367
5.169.1 Detailed Description	367
5.170 ws_i2c_gbt_x.WsI2cGbt_x Class Reference	367

5.170.1 Detailed Description	368
5.170.2 Constructor & Destructor Documentation	368
5.170.2.1 <code>__init__()</code>	368
5.170.3 Member Function Documentation	368
5.170.3.1 <code>dump_config()</code>	368
5.170.3.2 <code>dump_gbt_x_config()</code>	369
5.170.3.3 <code>gbt_x_config()</code>	369
5.170.3.4 <code>get_address()</code>	369
5.170.3.5 <code>read_data()</code>	369
5.170.3.6 <code>set_address()</code>	370
5.170.3.7 <code>write_data()</code>	370
5.171 <code>ws_i2c_gbt_x.WsI2cGbt_xAddress</code> Class Reference	370
5.171.1 Detailed Description	371
5.172 <code>ws_master_monitor.WsMasterMonitor</code> Class Reference	371
5.172.1 Detailed Description	371
5.172.2 Member Function Documentation	371
5.172.2.1 <code>dp23_monitor_get_register()</code>	372
5.172.2.2 <code>dp23_monitor_latch_counters()</code>	372
5.172.2.3 <code>dp23_monitor_rst_counters()</code>	372
5.172.2.4 <code>dump_config()</code>	372
5.172.2.5 <code>get_dp_data_producer()</code>	372
5.172.2.6 <code>get_mismatch()</code>	373
5.172.2.7 <code>read_counters()</code>	373
5.172.2.8 <code>set_dp_data_producer()</code>	373
5.173 <code>ws_master_monitor.WsMasterMonitorAddress</code> Class Reference	373
5.173.1 Detailed Description	374
5.174 <code>radiation_monitor.WsRadiationMonitor</code> Class Reference	374
5.174.1 Detailed Description	374
5.174.2 Member Function Documentation	375
5.174.2.1 <code>dump_config()</code>	375
5.174.2.2 <code>get_all()</code>	375
5.174.2.3 <code>get_counters()</code>	375
5.175 <code>radiation_monitor.WsRadiationMonitorAddress</code> Class Reference	375
5.175.1 Detailed Description	376
5.176 <code>radiation_monitor.WsRadiationMonitorSizes</code> Class Reference	376
5.176.1 Detailed Description	377
5.177 <code>ws_status.WsStatus</code> Class Reference	377
5.177.1 Detailed Description	377
5.177.2 Constructor & Destructor Documentation	377
5.177.2.1 <code>__init__()</code>	378
5.177.3 Member Function Documentation	378
5.177.3.1 <code>check_git_hash_and_date()</code>	378

5.177.3.2 dump_config()	378
5.177.3.3 get_date()	378
5.177.3.4 get_dipswitch()	379
5.177.3.5 get_dna_value()	379
5.177.3.6 get_git_hash()	379
5.177.3.7 get_os()	379
5.178 ws_status.WsStatusAddress Class Reference	380
5.178.1 Detailed Description	380
5.179 sysmon.WsSysmonAddress Class Reference	380
5.179.1 Detailed Description	381
5.180 trigger_handler.WsTriggerHandlerAddress Class Reference	381
5.180.1 Detailed Description	381
5.181 trigger_handler_monitor.WsTriggerHandlerMonitorAddress Class Reference	382
5.181.1 Detailed Description	382
5.182 ws_usb_if.WsUsbif Class Reference	382
5.182.1 Detailed Description	383
5.182.2 Constructor & Destructor Documentation	383
5.182.2.1 __init__()	383
5.182.3 Member Function Documentation	383
5.182.3.1 dp23_monitor_get_register()	383
5.182.3.2 dp23_monitor_latch_counters()	384
5.182.3.3 dp23_monitor_rst_counters()	384
5.182.3.4 dump_config()	384
5.182.3.5 get_dp_data_producer()	384
5.182.3.6 set_dp_data_producer()	385
5.183 ws_usb_if.WsUsbifAddress Class Reference	385
5.183.1 Detailed Description	385
5.184 wishbone_wait.WsWishboneWaitAddress Class Reference	386
5.184.1 Detailed Description	386
5.185 analyse_bitfile.XsType Class Reference	386
Index	387

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

charm_testbeam	11
chip	12
config	13
counter_monitor	13
dctrl	13
drp_bridge	14
gbt_sca	14
gbtx_controller	14
gbtx_flow_monitor	15
i2c	15
mmcm_monitor	15
module_includes	15
power_unit	16
proasic3	16
radiation_monitor	17
ru_board	17
ru_chip_power	17
ru_datapath_monitor	18
ru_gbt_packer	18
ru_transceiver	18
RUv1_regression	19
sca_ru	21
simulation_if	21
sysmon	22
testbench	22
trigger	24
trigger_handler	24
trigger_handler_monitor	25
usb_communication.communication	25
userdefinedexceptions	26
validate_scrubbing	26
wishbone_module	27
wishbone_wait	27
ws_i2c_gbtx	28
ws_master_monitor	28
ws_status	28
ws_usb_if	28

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ru_chip_power.ConversionParameter	122
ru_datapath_monitor.DatapathMonitorDual	126
RUv1_regression.DctrlBaseTest	141
events.Event	148
events.EventDecoder	149
Exception	
gbt_sca.ScaBadErrorFlagError	295
gbt_sca.Scal2cBadStatusError	298
jcm_control_server.NoReadBackFileError	211
usb_communication.communication.AddressMismatchError	32
usb_communication.communication.NetUsbTimeoutException	210
usb_communication.communication.WishboneReadError	358
userdefinedexceptions.ChipError	107
userdefinedexceptions.ChipidMismatchError	107
userdefinedexceptions.DataReadbackMismatchError	131
userdefinedexceptions.ExternalShutdownError	152
hameg.Hameg	192
IntEnum	
testbench.SensorMatrixPattern	301
object	
analyse_bitfile.BitFileAnalyser	95
beam_shutter.BeamShutter	92
beam_shutter.DummyBeamShutter	145
charm_testbeam.BeamShutterDataPoint	94
charm_testbeam.DataPoint	128
charm_testbeam.ExitStatus	151
charm_testbeam.ScrubbingCyclesDataPoint	300
charm_testbeam.TestbeamTest	308
charm_trigger.CharmTrigger	98
chip.Chip	99
pALPIDE3.PAlpide3	215
pALPIDE.Alpide	33
config.Analysis	91
events.EventPlotter	150
flash_test.FlashChecker	159

gbt_sca.Sca	289
sca_ru.Sca_RU	293
jcm_control_server.JCM	197
jcm_control_server.JcmControlClient	201
jcm_control_server.JcmControlServer	205
pALPIDE3.Addr	31
pALPIDE3.Opcode	212
pALPIDE.Addr	29
pALPIDE.Opcode	212
pALPIDE.CommandRegisterOpcode	115
power_supplies.ChipPowerSupply	111
power_supplies.ChipPowerSupplyHameg	111
power_supplies.ChipPowerSupplyHamegStave	112
power_supplies.ChipPowerSupplyInternal	113
power_supplies.ChipPowerSupplyMixed	114
power_supplies.CompositeChipPowerSupply	121
power_supplies.DummyPowerSupply	146
proasic3.ProAsic3	270
ru_board.ConfigurationMemoryMapping	122
ru_eyescan.EyeScan	153
ru_eyescan.EyeScanGth	155
ru_eyescan.EyeScanGtx	156
simulation_if.GbtCruGbtXBridge	168
simulation_if.GbtXSim	180
testbench.Testbench	311
usb_communication.communication.Communication	116
usb_communication.communication.NetUsbComm	208
usb_communication.communication.PrefetchCommunication	267
usb_communication.communication.PyUsbComm	277
validate_scrubbing.DataPoint	130
validate_scrubbing.ExitStatus	151
validate_scrubbing.ScrubbingCyclesDataPoint	301
validate_scrubbing.TestbeamTest	307
wishbone_module.WishboneModule	356
counter_monitor.WsCounterMonitor	362
ws_master_monitor.WsMasterMonitor	371
dctrl.Dctrl	132
drp_bridge.DrpBridge	141
gbtx_controller.GBTxController	174
gbtx_flow_monitor.GbtXFlowMonitor	178
i2c.I2CModule	196
mmcm_monitor.MmcmMonitor	206
power_unit.PowerUnit	259
radiation_monitor.WsRadiationMonitor	374
ru_chip_power.ChipPower	108
ru_datapath_monitor.DatapathMonitor	124
ru_gbt_packer.GbtPacker	170
ru_gbt_packer.GbtPackerMonitor	172
ru_transceiver.GpioFrontend	183
ru_transceiver.GthFrontend	187
ru_wb2fifo.WishboneToFifo	359
sca_ru.Sca_RU	293
sysmon.Sysmon	304
trigger_handler.TriggerHandler	341
trigger_handler_monitor.TriggerHandlerMonitor	344
wishbone_wait.WishboneWait	360
ws_i2c_gbtX.WsI2cGbtX	367
ws_status.WsStatus	377

ws_usb_if.WsUsbif	382
ru_board.ReadoutBoard	279
ru_board.RUv0_CRU	281
ru_board.RUv1	286
simulation_if.SimulationServer	302
TestCase	
charm_testbeam_check_setup.TestHardwareConnections	324
charm_testbeam_check_setup.TestSoftware	328
lab_check_setup.TestHardwareConnections	323
RUv1_regression.TestcaseBase	315
RUv1_regression.DctrlBaseTest.TestDctrl	319
RUv1_regression.TestDctrl0	320
RUv1_regression.TestDctrl1	320
RUv1_regression.TestDctrl2	321
RUv1_regression.TestDctrl3	321
RUv1_regression.TestDctrl4	322
RUv1_regression.TestCRUSWTRaceCondition	317
RUv1_regression.TestDatapathMonitorModules	318
RUv1_regression.TestGbtFlowMonitor	322
RUv1_regression.TestI2C	324
RUv1_regression.TestPythonScripts	326
RUv1_regression.TestScaReset	326
RUv1_regression.TestSimExitHandling	327
RUv1_regression.TestSysmon	328
RUv1_regression.TestTransceiverFrontend	329
RUv1_regression.TestTransceiverGpioFrontend	329
RUv1_regression.TestTransceiverGthFrontend	330
RUv1_regression.TestWb2Fifo	335
RUv1_regression.TestWishboneDualMaster	336
RUv1_regression.TestWishboneMaster	337
RUv1_regression.TestWishboneSlaves	337
RUv1_regression.TestWsRadiationMonitor	339
RUv1_regression.TestWsStatus	340
RUv1_regression.TriggerHandlerBaseTest.TestTriggerHandler	331
RUv1_regression.TestTriggerHandlerContinuousMode	332
RUv1_regression.TestTriggerHandlerNoMode	333
RUv1_regression.TestTriggerHandlerTriggeredMode	334
RUv1_regression.TriggerHandlerBaseTest	344
tti.TTI	346
usb_communication.communication.UsbCommServer	351
Communication	
simulation_if.UsbCommSim	352
simulation_if.Wb2GbtFlowComm	354
Enum	
charm_testbeam.SensorPoweringScheme	302
charm_testbeam.TestMode	325
validate_scrubbing.TestMode	325
IntEnum	
analyse_bitfile.XsType	386
counter_monitor.WsCounterMonitorAddress	363
dctrl.WsDctrlAddress	364
gbt_sca.ScaChannel	296
gbt_sca.ScaGpio	297
gbt_sca.Scal2cCmd	299
gbt_sca.Scal2cSpeed	299
gibt_flow_monitor.WsGibtFlowMonitorAddress	365
gibt_flow_monitor.WsGibtFlowMonitorAddress	366
gibt_flow_monitor.WsGibtFlowMonitorAddress_bit	367

mmcm_monitor.MmcmMonitorAddress	208
power_unit.Wbl2cPuAddress	355
proasic3.CcActiveState	97
proasic3.CcCmdOpcode	97
proasic3.FifoWriterCmdOpcode	158
proasic3.FlashCmdOpcode	161
proasic3.FlashFsmBlockEraseControllerState	161
proasic3.FlashFsmNandControllerState	163
proasic3.FlashFsmPageReadController	163
proasic3.FlashFsmPageWriteController	165
proasic3.FlashFsmReadIdControllerState	166
proasic3.FlashSelectICOpcode	167
proasic3.Pa3Register	213
proasic3.RcCmdOpcode	278
proasic3.SmapCmdOpcode	303
radiation_monitor.WsRadiationMonitorAddress	375
radiation_monitor.WsRadiationMonitorSizes	376
sca_ru.RuScaReg	281
sysmon.DrpSysmonAddress	142
sysmon.DrpSysmonConfigMasks	144
sysmon.DrpSysmonFlagMasks	145
sysmon.WsSysmonAddress	380
trigger_handler.Pulse_nTrigger	277
trigger_handler.WsTriggerHandlerAddress	381
trigger_handler_monitor.WsTriggerHandlerMonitorAddress	382
wishbone_wait.WsWishboneWaitAddress	386
ws_i2c_gbtx.WsI2cGbtAddress	370
ws_master_monitor.WsMasterMonitorAddress	373
ws_status.WsStatusAddress	380
ws_usb_if.WsUsbifAddress	385
IntFlag	
gbt_sca.ScaErrorFlag	296
gbt_sca.Scal2cStatusFlag	300
proasic3.CcStatus	98
proasic3.EccCmdFlags	147
proasic3.EccStatus	148
proasic3.FifoState	157
proasic3.FifoWriterStatus	158
proasic3.FlashStatus	167
proasic3.RcState	278
proasic3.SmapStatus	304
trigger.BitMap	96
trigger_handler_monitor.CountersBitMapping	123

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

pALPIDE.Addr	29
pALPIDE3.Addr	31
usb_communication.communication.AddressMismatchError	32
pALPIDE.Alpide	33
config.Analysis	91
beam_shutter.BeamShutter	92
charm_testbeam.BeamShutterDataPoint	94
analyse_bitfile.BitFileAnalyser	95
trigger.BitMap	96
proasic3.CcActiveState	97
proasic3.CcCmdOpcode	97
proasic3.CcStatus	98
charm_trigger.CharmTrigger	98
chip.Chip	99
userdefinedexceptions.ChipError	107
userdefinedexceptions.ChipidMismatchError	107
ru_chip_power.ChipPower	108
power_supplies.ChipPowerSupply	111
power_supplies.ChipPowerSupplyHameg	111
power_supplies.ChipPowerSupplyHamegStave	112
power_supplies.ChipPowerSupplyInternal	113
power_supplies.ChipPowerSupplyMixed	114
pALPIDE.CommandRegisterOpcode	115
usb_communication.communication.Communication	116
power_supplies.CompositeChipPowerSupply	121
ru_board.ConfigurationMemoryMapping	122
ru_chip_power.ConversionParameter	122
trigger_handler_monitor.CountersBitMapping	123
ru_datapath_monitor.DatapathMonitor	124
ru_datapath_monitor.DatapathMonitorDual	126
charm_testbeam.DataPoint	128
validate_scrubbing.DataPoint	130
userdefinedexceptions.DataReadbackMismatchError	131
dctrl.Dctrl	132
RUv1_regression.DctrlBaseTest	141

drp_bridge.DrpBridge	141
sysmon.DrpSysmonAddress	142
sysmon.DrpSysmonConfigMasks	144
sysmon.DrpSysmonFlagMasks	145
beam_shutter.DummyBeamShutter	145
power_supplies.DummyPowerSupply	146
proasic3.EccCmdFlags	147
proasic3.EccStatus	148
events.Event	148
events.EventDecoder	149
events.EventPlotter	150
validate_scrubbing.ExitStatus	151
charm_testbeam.ExitStatus	151
userdefinedexceptions.ExternalShutdownError	152
ru_eyescan.EyeScan	153
ru_eyescan.EyeScanGth	155
ru_eyescan.EyeScanGtx	156
proasic3.FifoState	157
proasic3.FifoWriterCmdOpcode	158
proasic3.FifoWriterStatus	158
flash_test.FlashChecker	159
proasic3.FlashCmdOpcode	161
proasic3.FlashFsmBlockEraseControllerState	161
proasic3.FlashFsmNandControllerState	163
proasic3.FlashFsmPageReadController	163
proasic3.FlashFsmPageWriteController	165
proasic3.FlashFsmReadIdControllerState	166
proasic3.FlashSelectIOOpcode	167
proasic3.FlashStatus	167
simulation_if.GbtCruGbtXBridge	168
ru_gbt_packer.GbtPacker	170
ru_gbt_packer.GbtPackerMonitor	172
gbtx_controller.GBTXController	174
gbtx_flow_monitor.GbtXFlowMonitor	178
simulation_if.GbtXSim	180
ru_transceiver.GpioFrontend	183
ru_transceiver.GthFrontend	187
hameg.Hameg	192
i2c.I2CModule	196
jcm_control_server.JCM	197
jcm_control_server.JcmControlClient	201
jcm_control_server.JcmControlServer	205
mmcm_monitor.MmcmMonitor	206
mmcm_monitor.MmcmMonitorAddress	208
usb_communication.communication.NetUsbComm	208
usb_communication.communication.NetUsbTimeoutException	210
jcm_control_server.NoReadBackFileError	211
pALPIDE.Opcode	212
pALPIDE3.Opcode	212
proasic3.Pa3Register	213
pALPIDE3.PAlpide3	215
power_unit.PowerUnit	259
usb_communication.communication.PrefetchCommunication	267
proasic3.ProAsic3	270
trigger_handler.Pulse_nTrigger	277
usb_communication.communication.PyUsbComm	277
proasic3.RcCmdOpcode	278
proasic3.RcState	278

ru_board.ReadoutBoard	279
sca_ru.RuScaReg	281
ru_board.RUv0_CRU	281
ru_board.RUv1	286
gbt_sca.Sca	289
sca_ru.Sca_RU	293
gbt_sca.ScaBadErrorFlagError	295
gbt_sca.ScaChannel	296
gbt_sca.ScaErrorFlag	296
gbt_sca.ScaGpio	297
gbt_sca.Scal2cBadStatusError	298
gbt_sca.Scal2cCmd	299
gbt_sca.Scal2cSpeed	299
gbt_sca.Scal2cStatusFlag	300
charm_testbeam.ScrubbingCyclesDataPoint	300
validate_scrubbing.ScrubbingCyclesDataPoint	301
testbench.SensorMatrixPattern	301
charm_testbeam.SensorPoweringScheme	302
simulation_if.SimulationServer	302
proasic3.SmapCmdOpcode	303
proasic3.SmapStatus	304
sysmon.Sysmon	304
validate_scrubbing.TestbeamTest	307
charm_testbeam.TestbeamTest	308
testbench.Testbench	311
RUv1_regression.TestcaseBase	315
RUv1_regression.TestCRUSWTRaceCondition	317
RUv1_regression.TestDatapathMonitorModules	318
RUv1_regression.DctrlBaseTest.TestDctrl	319
RUv1_regression.TestDctrl0	320
RUv1_regression.TestDctrl1	320
RUv1_regression.TestDctrl2	321
RUv1_regression.TestDctrl3	321
RUv1_regression.TestDctrl4	322
RUv1_regression.TestGbtXFlowMonitor	322
lab_check_setup.TestHardwareConnections	323
charm_testbeam_check_setup.TestHardwareConnections	324
RUv1_regression.TestI2C	324
charm_testbeam.TestMode	325
validate_scrubbing.TestMode	325
RUv1_regression.TestPythonScripts	326
RUv1_regression.TestScaReset	326
RUv1_regression.TestSimExitHandling	327
charm_testbeam_check_setup.TestSoftware	328
RUv1_regression.TestSysmon	328
RUv1_regression.TestTransceiverFrontend	329
RUv1_regression.TestTransceiverGpioFrontend	329
RUv1_regression.TestTransceiverGthFrontend	330
RUv1_regression.TriggerHandlerBaseTest.TestTriggerHandler	331
RUv1_regression.TestTriggerHandlerContinuousMode	332
RUv1_regression.TestTriggerHandlerNoMode	333
RUv1_regression.TestTriggerHandlerTriggeredMode	334
RUv1_regression.TestWb2Fifo	335
RUv1_regression.TestWishboneDualMaster	336
RUv1_regression.TestWishboneMaster	337
RUv1_regression.TestWishboneSlaves	337
RUv1_regression.TestWsRadiationMonitor	339
RUv1_regression.TestWsStatus	340

trigger_handler.TriggerHandler	341
RUv1_regression.TriggerHandlerBaseTest	344
trigger_handler_monitor.TriggerHandlerMonitor	344
tti.TTI	346
usb_communication.communication.UsbCommServer	351
simulation_if.UsbCommSim	352
simulation_if.Wb2GbtXComm	354
power_unit.Wb12cPuAddress	355
wishbone_module.WishboneModule	356
usb_communication.communication.WishboneReadError	358
ru_wb2fifo.WishboneToFifo	359
wishbone_wait.WishboneWait	360
counter_monitor.WsCounterMonitor	362
counter_monitor.WsCounterMonitorAddress	363
dctrl.WsDctrlAddress	364
gbtx_controller.WsGbtXControllerAddress	365
gbtx_flow_monitor.WsGbtXFlowMonitorAddress	366
gbtx_flow_monitor.WsGbtXFlowMonitorAddress_bit	367
ws_i2c_gbtX.WsI2cGbtX	367
ws_i2c_gbtX.WsI2cGbtXAddress	370
ws_master_monitor.WsMasterMonitor	371
ws_master_monitor.WsMasterMonitorAddress	373
radiation_monitor.WsRadiationMonitor	374
radiation_monitor.WsRadiationMonitorAddress	375
radiation_monitor.WsRadiationMonitorSizes	376
ws_status.WsStatus	377
ws_status.WsStatusAddress	380
sysmon.WsSysmonAddress	380
trigger_handler.WsTriggerHandlerAddress	381
trigger_handler_monitor.WsTriggerHandlerMonitorAddress	382
ws_usb_if.WsUsbif	382
ws_usb_if.WsUsbifAddress	385
wishbone_wait.WsWishboneWaitAddress	386
analyse_bitfile.XsType	386

Chapter 4

Namespace Documentation

4.1 charm_testbeam Namespace Reference

Classes

- class [BeamShutterDataPoint](#)
- class [DataPoint](#)
- class [ExitStatus](#)
- class [ScrubbingCyclesDataPoint](#)
- class [SensorPoweringScheme](#)
- class [TestbeamTest](#)
- class [TestMode](#)

Functions

- def **is_beam** (test_mode)
- def **is_pa3_programming** (test_mode)
- def **heardKeypress** ()

Variables

- string **jcm_server_address** = "localhost"
- int **jcm_server_port** = 8001
- **TEST_MODE** = TestMode.JCM_FAULT_INJECTION
- bool **USE_BEAM_SHUTTER** = False
- int **NR_FAULTS_INJECT** = 100000
- int **FAULT_INJECT_DELAY** = 90
- int **FAULT_INJECT_DELAY_AFTER** = 10
- bool **FAULT_INJECT_CORRECTION** = True
- bool **READ_uC** = False
- bool **READ_PA3_VALUES** = False
- bool **READ_PA3_SCRUB_COUNTER** = False
- int **END_TIME** = 60 * 10
- bool **RESUME_RUN** = False
- **MANUAL_SCRUBBING_CYCLE** = None
- bool **EVENT_ANALYSIS_ONLINE** = False

- string **PORT_BEAM_SHUTTER** = '/dev/ttyBEAM_SHTR_CNTRL'
- string **USB_COMM_EXEC** = "../modules/usb_if/software/usb_comm_server/build/usb_comm"
- string **SERIAL_CRU** = "000001"
- string **SERIAL_RDO** = "000000"
- string **HAMEG_PORT** = "/dev/ttyHAMEG"
- int **GITHASH_CRU** = 0x0E74C47F
- int **GITHASH_RDO** = 0x01ab6ae
- int **BS_BITFILE_CRC** = 0x3292A090
- bool **GTH_ACTIVE** = True

Configure Readout mode: # GPIO_ACTIVE, GTH_ACTIVE: Activate Readout of either (or both) modules # READOUT_SOURCE: Main Source for readout (forwarded to GbtX) # TRIGGER_FREQUENCY, SENSOR_PATTERN: Trigger configuration #.

- bool **GPIO_ACTIVE** = True
- string **READOUT_SOURCE** = "GPIO"
- int **TRIGGER_FREQUENCY** = 10000
- **SENSOR_PATTERN** = testbench.SensorMatrixPattern.EMPTY
- int **GTH_CONNECTOR** = 4
- **READOUT_GTH_LIST** = list(range(9))
- int **GPIO_CONNECTOR** = 0
- **READOUT_GPIO_LIST** = list(range(7))
- int **MAIN_CONNECTOR** = GTH_CONNECTOR
- **SENSOR_POWERING_SCHEME** = SensorPoweringScheme.POWERUNIT
- list **POWERUNIT_OUTPUT_CHANNEL_LIST** = []
- **testbeam_test** = [TestbeamTest](#)()
- bool **RUv1_force_powercycle** = False
- bool **RUv0_force_powercycle** = False
- **force_powercycle**
- **ruv1_is_on** = testbeam_test.check_powerstate_RUv1()
- **commitTransaction**
- **connector_nr**
- **e**
- **exc_info**

4.1.1 Detailed Description

Script for running the Charm testbeam

4.2 chip Namespace Reference

Classes

- class [Chip](#)

4.2.1 Detailed Description

file for the abstraction chip abstraction layer

4.3 config Namespace Reference

Classes

- class [Analysis](#)

Variables

- dictionary **analysis**
- **USR** = `getpass.getuser()`
- **MODULE_PATH** = `os.path.dirname(sys.modules[__name__].__file__)`
- string **VIVADOLAB_PATH** = `'/opt/Xilinx/Vivado_Lab/2017.1/bin/vivado_lab'`
- string **SEM_IP_PORT** = `'/dev/ttySEM_IP_UART'`
- string **POWERSUPPLY_PORT** = `'/dev/ttyHAMEG0'`
- string **BEAM_SHUTTER_PORT** = `'/dev/BEAM_SHUTTER'`
- int **TESTBEAM_MODE** = 0
- int **MAX_REPEAT** = 25

4.3.1 Detailed Description

File configuring different analysis for different flash readout

Matteo Lupi <matteo.lupi@cern.ch>
Configuration file to set platform dependent options

4.4 counter_monitor Namespace Reference

Classes

- class [WsCounterMonitor](#)
- class [WsCounterMonitorAddress](#)

4.4.1 Detailed Description

Implements the control for the counter_monitor wishbone slave

4.5 dctrl Namespace Reference

Classes

- class [Dctrl](#)
- class [WsDctrlAddress](#)

4.5.1 Detailed Description

Class to implement the latest capabilities of the dctrl/ctrl block relative to trigger commands with busy management

4.6 drp_bridge Namespace Reference

Classes

- class [DrpBridge](#)

4.6.1 Detailed Description

DRP bridge implementation

4.7 gbt_sca Namespace Reference

Classes

- class [Sca](#)
- class [ScaBadErrorFlagError](#)
- class [ScaChannel](#)
- class [ScaErrorFlag](#)
- class [ScaGpio](#)
- class [ScaI2cBadStatusError](#)
- class [ScaI2cCmd](#)
- class [ScaI2cSpeed](#)
- class [ScaI2cStatusFlag](#)

4.7.1 Detailed Description

Generic SCA module

4.8 gbt_x_controller Namespace Reference

Classes

- class [GBTxController](#)
- class [WsGbtXControllerAddress](#)

4.8.1 Detailed Description

Class to implement the latest capabilities of the GBTxController

4.9 gbtx_flow_monitor Namespace Reference

Classes

- class [GbtxFLOWMonitor](#)
- class [WsGbtxFLOWMonitorAddress](#)
- class [WsGbtxFLOWMonitorAddress_bit](#)

4.9.1 Detailed Description

Class to implement the latest capabilities of the GbtxFLOWMonitor

4.10 i2c Namespace Reference

Classes

- class [I2CModule](#)

4.10.1 Detailed Description

Generic I2C module

4.11 mmcm_monitor Namespace Reference

Classes

- class [MmcmMonitor](#)
- class [MmcmMonitorAddress](#)

4.11.1 Detailed Description

file implementing the control for the mmcm_monitor wishbone slave

4.12 module_includes Namespace Reference

Variables

- **spec** = `importlib.util.spec_from_file_location(module_name, _MODULES_FOLDER_PATH + module_path)`
- **module** = `importlib.util.module_from_spec(spec)`

4.12.1 Detailed Description

Include paths for external module files

4.13 power_unit Namespace Reference

Classes

- class [PowerUnit](#)
- class [Wbl2cPuAddress](#)

4.13.1 Detailed Description

Power unit wishbone module

NOTE: use latest documentation

Twiki: <https://twiki.cern.ch/twiki/bin/view/ALICE/Documentation>

Documentation at: https://twiki.cern.ch/twiki/pub/ALICE/Documentation/2018_09_04_ITS_power_system_V15_32-channel_operation_manual
and
https://twiki.cern.ch/twiki/pub/ALICE/Documentation/2018_09_04_ITS_power_system_V15_32-channel_operation_manual

4.14 proasic3 Namespace Reference

Classes

- class [CcActiveState](#)
- class [CcCmdOpcode](#)
- class [CcStatus](#)
- class [EccCmdFlags](#)
- class [EccStatus](#)
- class [FifoState](#)
- class [FifoWriterCmdOpcode](#)
- class [FifoWriterStatus](#)
- class [FlashCmdOpcode](#)
- class [FlashFsmBlockEraseControllerState](#)
- class [FlashFsmNandControllerState](#)
- class [FlashFsmPageReadController](#)
- class [FlashFsmPageWriteController](#)
- class [FlashFsmReadIdControllerState](#)
- class [FlashSelectICOpcode](#)
- class [FlashStatus](#)
- class [Pa3Register](#)
- class [ProAsic3](#)
- class [RcCmdOpcode](#)
- class [RcState](#)
- class [SmapCmdOpcode](#)
- class [SmapStatus](#)

Variables

- int **PAGE_SIZE** = 4096
- int **PAGE_SIZE_ECC** = 4192
- int **PAGES_PER_BLOCK** = 64
- int **BLOCK_SIZE** = PAGE_SIZE*PAGES_PER_BLOCK
- int **BLOCK_COUNT** = 8192

4.14.1 Detailed Description

Class implementing an abstraction layer for the ProAsic3

4.15 radiation_monitor Namespace Reference

Classes

- class [WsRadiationMonitor](#)
- class [WsRadiationMonitorAddress](#)
- class [WsRadiationMonitorSizes](#)

Functions

- def **format_counter** (value)

4.15.1 Detailed Description

Class to implement the latest capabilities of the SCAController

4.16 ru_board Namespace Reference

Classes

- class [ConfigurationMemoryMapping](#)
- class [ReadoutBoard](#)
- class [RUv0_CRU](#)
- class [RUv1](#)

4.16.1 Detailed Description

Readout unit implementation of the Board interface.

4.17 ru_chip_power Namespace Reference

Classes

- class [ChipPower](#)
- class [ConversionParameter](#)

4.17.1 Detailed Description

Chip power module.

Enable chip voltages, set voltages, readback voltages and currents for AVDD,DVDD

4.18 ru_datapath_monitor Namespace Reference

Classes

- class [DatapathMonitor](#)
- class [DatapathMonitorDual](#)

4.18.1 Detailed Description

Datapath monitor module for Reading out Lane counters

4.19 ru_gbt_packer Namespace Reference

Classes

- class [GbtPacker](#)
- class [GbtPackerMonitor](#)

4.19.1 Detailed Description

Trigger handler class for RU

4.20 ru_transceiver Namespace Reference

Classes

- class [GpioFrontend](#)
- class [GthFrontend](#)

4.20.1 Detailed Description

Transceiver module for Xilinx GTX Transceiver functions

4.21 RUv1_regression Namespace Reference

Classes

- class [DctrlBaseTest](#)
- class [TestcaseBase](#)
- class [TestCRUSWTRaceCondition](#)
- class [TestDatapathMonitorModules](#)
- class [TestDctrl0](#)
- class [TestDctrl1](#)
- class [TestDctrl2](#)
- class [TestDctrl3](#)
- class [TestDctrl4](#)
- class [TestGbtxFLOWMonitor](#)
- class [TestI2C](#)
- class [TestPythonScripts](#)
- class [TestScaReset](#)
- class [TestSimExitHandling](#)
- class [TestSysmon](#)
- class [TestTransceiverFrontend](#)
- class [TestTransceiverGpioFrontend](#)
- class [TestTransceiverGthFrontend](#)
- class [TestTriggerHandlerContinuousMode](#)
- class [TestTriggerHandlerNoMode](#)
- class [TestTriggerHandlerTriggeredMode](#)
- class [TestWb2Fifo](#)
- class [TestWishboneDualMaster](#)
- class [TestWishboneMaster](#)
- class [TestWishboneSlaves](#)
- class [TestWsRadiationMonitor](#)
- class [TestWsStatus](#)
- class [TriggerHandlerBaseTest](#)

Functions

- def [tearDownModule](#) ()

Variables

- string **SERIAL_CRU** = "000001"
- string **SERIAL_RDO** = "000000"
- bool **SIMULATION** = True
- *Diagnostic.*
- bool **SIMULATE_CRU** = False
- bool **USB_MASTER** = True
- int **DCTRL_GTH** = 0
- int **DCTRL_GPIO** = 0
- list **DCTRL_CONNECTORS** = [DCTRL_GTH]
- list **SENSOR_LIST** = [1]
- **GTH_LIST** = list(range(9))
- dictionary **GTH_CONNECTOR_LUT** = {i : DCTRL_GTH for i in SENSOR_LIST}
- dictionary **GPIO_CONNECTOR_LUT** = {i : DCTRL_GPIO for i in SENSOR_LIST}

- **GPIO_LIST** = list(range(7))
- dictionary **GPIO_SENSORS_PER_LANE** = {i:7 for i in GPIO_LIST}
- dictionary **GPIO_SENSOR_MASK** = {i:0x0 for i in GPIO_LIST}
- int **GITHASH** = 0xBADCAFE
- bool **USE_LTU** = True
- int **USE_TB_V1L** = 0
- list **GPIO_SUBSET_MAP**
- **boardGlobal** = None
- **boardGlobal_usb** = None
- **cruGlobal** = None
- **ltuGlobal** = None
- **gbtx_sim_comm** = None
- **gbtx2_sim_comm** = None
- **sim_serv** = None
- **gbt_sim** = None
- **gbt2_sim** = None
- **comm** = None
- **serv** = None
- int **timeout** = 1
- **logger** = logging.getLogger()
- string **log_file** = "Ruv1_regression.log"
- string **log_file_errors** = "Ruv1_regression_errors.log"
- **fh** = logging.FileHandler(log_file)
- **fh2** = logging.FileHandler(log_file_errors)
- **ch** = logging.StreamHandler()
- **formatter**
- **formatter_ch**
- **comm2** = [simulation_if.Wb2GbtxComm](#)(gbtx_sim=gbt2_sim)
- **comm_usb** = [simulation_if.UsbCommSim](#)(server=sim_serv.server)
- **comm_prefetch** = communication.PrefetchCommunication(comm)
- **comm_usb_prefetch** = communication.PrefetchCommunication(comm_usb)
- dictionary **connection_lut** = {sensor: DCTRL_CONNECTORS[0] for sensor in SENSOR_LIST}
- **commitTransaction**
- **verbosity**
- **exit**
- **file**

4.21.1 Detailed Description

Regression test list for RUV1

4.21.2 Variable Documentation

4.21.2.1 formatter

RUV1_regression.formatter

Initial value:

```
1 = logging.Formatter(
2     "%(asctime)s - %(name)s - %(levelname)s - %(message)s")
```

4.21.2.2 formatter_ch

```
RUv1_regression.formatter_ch
```

Initial value:

```
1 = logging.Formatter(
2     "%(name)s - %(levelname)s - %(message)s")
```

4.21.2.3 GPIO_SUBSET_MAP

```
list RUv1_regression.GPIO_SUBSET_MAP
```

Initial value:

```
1 = [[0,1,2,10,11,12,13],
2     [3,14,15,16,17,26,27],
3     [18,19,20,22,23,24,25],
4     [4,5,6,7,8,9,21]]
```

4.21.2.4 SIMULATION

```
bool RUv1_regression.SIMULATION = True
```

Diagnostic.

```
import importlib ht_exist = importlib.util.find_spec("hanging_threads") if ht_exist: from hanging_threads import
start_monitoring monitoring_thread = start_monitoring()
```

4.22 sca_ru Namespace Reference

Classes

- class [RuScaReg](#)
- class [Sca_RU](#)

4.22.1 Detailed Description

Generic SCA module

4.23 simulation_if Namespace Reference

Classes

- class [GbtCruGbtBridge](#)
- class [GbtSim](#)
- class [SimulationServer](#)
- class [UsbCommSim](#)
- class [Wb2GbtComm](#)

Functions

- def [chunks](#) (l, n)

Variables

- int **COMM_SIM_PORT** = 32223
- int **GLOBAL_TIMEOUT** = 900
- string **rundir** = os.path.dirname(os.path.realpath(__file__)) + '/../../../sim/run/'

4.23.1 Detailed Description

Python to Systemverilog Simulation interface

4.23.2 Function Documentation

4.23.2.1 [chunks\(\)](#)

```
def simulation_if.chunks (
    l,
    n )
```

Yield successive n-sized chunks from l.

4.24 sysmon Namespace Reference

Classes

- class [DrpSysmonAddress](#)
- class [DrpSysmonConfigMasks](#)
- class [DrpSysmonFlagMasks](#)
- class [Sysmon](#)
- class [WsSysmonAddress](#)

4.24.1 Detailed Description

Class to communicate with the Xilinx System Monitor (SYSMON)

4.25 testbench Namespace Reference

Classes

- class [SensorMatrixPattern](#)
- class [Testbench](#)

Variables

- **script_path** = os.path.dirname(os.path.realpath(__file__))
- **modules_path**
- bool **USE_USB_COMM** = False
- **USB_COMM_EXEC**
- string **SERIAL_CRU** = "000001"
- string **SERIAL_RDO** = "000000"
- bool **STANDALONE_RUN** = False
- bool **USE_RDO_USB** = False
- bool **USE_CRU** = True
- int **USE_TB_V1L** = 0
- list **GPIO_SUBSET_MAP**
- dictionary **GPIO_SENSOR_MASK**
- **tb** = [Testbench\(\)](#)

4.25.1 Detailed Description

Generic Testbench for testing different routines and interactive access to RU modules

4.25.2 Variable Documentation

4.25.2.1 GPIO_SENSOR_MASK

dictionary testbench.GPIO_SENSOR_MASK

Initial value:

```
1 = { 0:0b1110111,
2      1:0b1011111,
3      2:0b0000000,
4      3:0b1111011,
5      4:0b0000000,
6      5:0b1110111,
7      6:0b1111011,
8      7:0b1101111,
9      8:0b0111111,
10     9:0b1111110,
11    10:0b1111110,
12    11:0b0111111,
13    12:0b1101111,
14    13:0b1111011,
15    14:0b1011111,
16    15:0b1101111,
17    16:0b0111111,
18    17:0b1111110,
19    18:0b1110111,
20    19:0b1011111,
21    20:0b0000000,
22    21:0b1011111,
23    22:0b1111110,
24    23:0b0111111,
25    24:0b1101111,
26    25:0b1111011,
27    26:0b0000000,
28    27:0b1110111}
```

4.25.2.2 GPIO_SUBSET_MAP

```
list testbench.GPIO_SUBSET_MAP
```

Initial value:

```
1 = [[0,1,2,10,11,12,13],
2      [3,14,15,16,17,26,27],
3      [18,19,20,22,23,24,25],
4      [4,5,6,7,8,9,21]]
```

4.25.2.3 modules_path

```
testbench.modules_path
```

Initial value:

```
1 = os.path.join(
2     script_path, '../..//modules/board_support_software/software/py/')
```

4.25.2.4 USB_COMM_EXEC

```
testbench.USB_COMM_EXEC
```

Initial value:

```
1 = os.path.join(
2     script_path, "../..//modules/usb_if/software/usb_comm_server/build/usb_comm")
```

4.26 trigger Namespace Reference

Classes

- class [BitMap](#)

4.26.1 Detailed Description

File describing the trigger types while the trigger handler class is not present

4.27 trigger_handler Namespace Reference

Classes

- class [Pulse_nTrigger](#)
- class [TriggerHandler](#)
- class [WsTriggerHandlerAddress](#)

4.27.1 Detailed Description

Class to implement the trigger handler wishbone slave abstraction layer

4.28 trigger_handler_monitor Namespace Reference

Classes

- class [CountersBitMapping](#)
- class [TriggerHandlerMonitor](#)
- class [WsTriggerHandlerMonitorAddress](#)

4.28.1 Detailed Description

Class to implement the trigger handler wishbone slave abstraction layer

4.29 usb_communication.communication Namespace Reference

Classes

- class [AddressMismatchError](#)
- class [Communication](#)
- class [NetUsbComm](#)
- class [NetUsbTimeoutException](#)
- class [PrefetchCommunication](#)
- class [PyUsbComm](#)
- class [UsbCommServer](#)
- class [WishboneReadError](#)

Functions

- def [sig_alarm_handler](#) (signum, frame)
- def [enqueue_output](#) (out, queue)

Variables

- string **ON_POSIX** = 'posix' in sys.builtin_module_names

4.29.1 Detailed Description

Communication classes, used to communicate with a board, using the FX3 chip and usb_if firmware.

The provided communication classes are either direct via libusb, over the usb_comm server, or via simulation. They all share the common base class `Communication`

4.29.2 Function Documentation

4.29.2.1 enqueue_output()

```
def usb_communication.communication.enqueue_output (
    out,
    queue )
```

Add message to message_queue

4.30 userdefinedexceptions Namespace Reference

Classes

- class [ChipError](#)
- class [ChipidMismatchError](#)
- class [DataReadbackMismatchError](#)
- class [ExternalShutdownError](#)

4.30.1 Detailed Description

File to define the user-defined exceptions, to be imported

4.31 validate_scrubbing Namespace Reference

Classes

- class [DataPoint](#)
- class [ExitStatus](#)
- class [ScrubbingCyclesDataPoint](#)
- class [TestbeamTest](#)
- class [TestMode](#)

Functions

- def [is_pa3_programming](#) (test_mode)
- def [heardKeypress](#) ()

Variables

- string **jcm_server_address** = "localhost"
- int **jcm_server_port** = 8001
- **TEST_MODE** = TestMode.PA3
- int **NR_FAULTS_INJECT** = 100
- int **FAULT_INJECT_DELAY** = 10
- string **USB_COMM_EXEC** = "../modules/usb_if/software/usb_comm_server/build/usb_comm"
- string **SERIAL_CRU** = "000001"
- string **SERIAL_RDO** = "000000"
- string **HAMEG_PORT** = "/dev/ttyHAMEG0"
- int **GITHASH_RDO** = 0x0a045881
- int **GITHASH_CRU** = 0x048ff336
- **testbeam_test** = [TestbeamTest](#)()
- bool **RUv1_force_powercycle** = False
- bool **RUv0_force_powercycle** = False
- **force_powercycle**
- **ruv1_is_on** = testbeam_test.check_powerstate_RUv1()
- **commitTransaction**
- **e**
- **exc_info**

4.31.1 Detailed Description

Script for running the Charm testbeam

4.32 wishbone_module Namespace Reference

Classes

- class [WishboneModule](#)

4.32.1 Detailed Description

Abstraction of a wishbone accessed module.

Provides Read/Write functionality to a module with given module_id on construction time.

4.33 wishbone_wait Namespace Reference

Classes

- class [WishboneWait](#)
- class [WsWishboneWaitAddress](#)

4.33.1 Detailed Description

Class to implement the latest capabilities of the wishbone_wait block relative to wait commands

4.34 ws_i2c_gbtx Namespace Reference

Classes

- class [Wsl2cGbtx](#)
- class [Wsl2cGbtxAddress](#)

4.34.1 Detailed Description

file implementing the control for the ws_i2_gbt wishbone slave

4.35 ws_master_monitor Namespace Reference

Classes

- class [WsMasterMonitor](#)
- class [WsMasterMonitorAddress](#)

4.35.1 Detailed Description

Implements the control for the ws_master_monitor wishbone slave

4.36 ws_status Namespace Reference

Classes

- class [WsStatus](#)
- class [WsStatusAddress](#)

4.36.1 Detailed Description

file implementing the control for the ws_status wishbone slave

4.37 ws_usb_if Namespace Reference

Classes

- class [WsUsbif](#)
- class [WsUsbifAddress](#)

4.37.1 Detailed Description

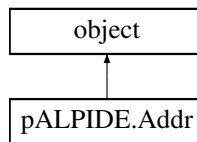
file implementing the control for the ws_usbif wishbone slave

Chapter 5

Class Documentation

5.1 pALPIDE.Addr Class Reference

Inheritance diagram for pALPIDE.Addr:



Static Public Attributes

- tuple **CMD** = (0 << 8) | 0
- tuple **MODE_CTRL** = (0 << 8) | 1
- tuple **DISABLE_REGIONS_LSBS** = (0 << 8) | 2
- tuple **DISABLE_REGIONS_MSBS** = (0 << 8) | 3
- tuple **FROMU_CFG_1** = (0 << 8) | 4
- tuple **FROMU_CFG_2** = (0 << 8) | 5
- tuple **FROMU_CFG_3** = (0 << 8) | 6
- tuple **FROMU_PULSING1** = (0 << 8) | 7
- tuple **FROMU_PULSING_2** = (0 << 8) | 8
- tuple **FROMU_STATUS_1** = (0 << 8) | 9
- tuple **FROMU_STATUS_2** = (0 << 8) | 10
- tuple **FROMU_STATUS_3** = (0 << 8) | 11
- tuple **FROMU_STATUS_4** = (0 << 8) | 12
- tuple **FROMU_STATUS_5** = (0 << 8) | 13
- tuple **DAC_SETTINGS_DCLK_AND_MCLK_IO_BUFFERS** = (0 << 8) | 14
- tuple **DAC_SETTINGS_CMU_IO_BUFFERS** = (0 << 8) | 15
- tuple **CMU_AND_DMU_CFG** = (0 << 8) | 16
- tuple **CMU_AND_DMU_STATUS** = (0 << 8) | 17
- tuple **DMU_DATA_FIFO_LSBS** = (0 << 8) | 18
- tuple **DMU_DATA_FIFO_MSBS** = (0 << 8) | 19
- tuple **DTU_CFG** = (0 << 8) | 20
- tuple **DTU_DACS** = (0 << 8) | 21
- tuple **DTU_PLL_LOCK_1** = (0 << 8) | 22
- tuple **DTU_PLL_LOCK_2** = (0 << 8) | 23

- tuple **DTU_TEST_1** = (0 << 8) | 24
- tuple **DTU_TEST_2** = (0 << 8) | 25
- tuple **DTU_TEST_3** = (0 << 8) | 26
- tuple **BUSY_MIN_WIDTH** = (0 << 8) | 27
- tuple **PIXEL_CFG** = (5 << 8) | 0
- tuple **ANALOG_MONITOR_AND_OVERRIDE** = (6 << 8) | 0
- tuple **VRESETP** = (6 << 8) | 1
- tuple **VRESETD** = (6 << 8) | 2
- tuple **VCASP** = (6 << 8) | 3
- tuple **VCASN** = (6 << 8) | 4
- tuple **VPULSEH** = (6 << 8) | 5
- tuple **VPULSEL** = (6 << 8) | 6
- tuple **VCASN2** = (6 << 8) | 7
- tuple **VCLIP** = (6 << 8) | 8
- tuple **VTEMP** = (6 << 8) | 9
- tuple **IAUX2** = (6 << 8) | 10
- tuple **IRESET** = (6 << 8) | 11
- tuple **IDB** = (6 << 8) | 12
- tuple **IBIAS** = (6 << 8) | 13
- tuple **ITHR** = (6 << 8) | 14
- tuple **BUFFER_BYPASS** = (6 << 8) | 15
- tuple **ADC_CTRL** = (6 << 8) | 16
- tuple **ADC_DAC_INPUT_VALUE** = (6 << 8) | 17
- tuple **ADC_CALIBRATION_VALUE** = (6 << 8) | 18
- tuple **ADC_AVSS_VALUE** = (6 << 8) | 19
- tuple **ADC_DVSS_VALUE** = (6 << 8) | 20
- tuple **ADC_AVDD_VALUE** = (6 << 8) | 21
- tuple **ADC_DVDD_VALUE** = (6 << 8) | 22
- tuple **ADC_VCASN_VALUE** = (6 << 8) | 23
- tuple **ADC_VCASP_VALUE** = (6 << 8) | 24
- tuple **ADC_VPULSEH_VALUE** = (6 << 8) | 25
- tuple **ADC_VPULSEL_VALUE** = (6 << 8) | 26
- tuple **ADC_VRESETP_VALUE** = (6 << 8) | 27
- tuple **ADC_VRESETD_VALUE** = (6 << 8) | 28
- tuple **ADC_VCASN2_VALUE** = (6 << 8) | 20
- tuple **ADC_VCLIP_VALUE** = (6 << 8) | 30
- tuple **ADC_VTEMP_VALUE** = (6 << 8) | 31
- tuple **ADC_ITHR_VALUE** = (6 << 8) | 32
- tuple **ADC_IREF_VALUE** = (6 << 8) | 33
- tuple **ADC_IDB_VALUE** = (6 << 8) | 34
- tuple **ADC_IBIAS_VALUE** = (6 << 8) | 35
- tuple **ADC_IAUX2_VALUE** = (6 << 8) | 36
- tuple **ADC_IRESET_VALUE** = (6 << 8) | 37
- tuple **ADC_BG2V_VALUE** = (6 << 8) | 38
- tuple **ADC_T2V_VALUE** = (6 << 8) | 39
- int **SEU_ERROR_COUNTER** = 0x700
- int **TEST_CONTROL** = 0x0701

5.1.1 Detailed Description

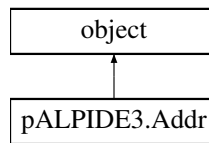
Addresses supported by ALPIDE (NOTE that only region independent addresses are supported here)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/pALPIDE.py

5.2 pALPIDE3.Addr Class Reference

Inheritance diagram for pALPIDE3.Addr:



Static Public Attributes

- tuple **CMD** = (0 << 8) | 0
- tuple **PERIPHERY_CTRL** = (0 << 8) | 1
- tuple **DISABLE_REGIONS_LSBS** = (0 << 8) | 2
- tuple **DISABLEOF_REGIONS_MSBS** = (0 << 8) | 3
- tuple **FROMU_CFG_1** = (0 << 8) | 4
- tuple **FROMU_CFG_2** = (0 << 8) | 5
- tuple **FROMU_PULSING1** = (0 << 8) | 6
- tuple **FROMU_PULSING_2** = (0 << 8) | 7
- tuple **FROMU_STATUS_1** = (0 << 8) | 8
- tuple **FROMU_STATUS_2** = (0 << 8) | 9
- tuple **DAC_SETTINGS_DCLK_AND_MCLK_IO_BUFFERS** = (0 << 8) | 10
- tuple **DAC_SETTINGS_CMU_IO_BUFFERS** = (0 << 8) | 11
- tuple **CMU_AND_DMU_CFG** = (0 << 8) | 12
- tuple **CMU_ERRORS_COUNTER** = (0 << 8) | 13
- tuple **DTU_CFG** = (0 << 8) | 14
- tuple **DTU_DACS** = (0 << 8) | 15
- tuple **DTU_PLL_LOCK_1** = (0 << 8) | 16
- tuple **DTU_PLL_LOCK_2** = (0 << 8) | 17
- tuple **DTU_TEST_1** = (0 << 8) | 18
- tuple **DTU_TEST_2** = (0 << 8) | 19
- tuple **DTU_TEST_3** = (0 << 8) | 20
- tuple **BUSY_MIN_WIDTH** = (0 << 8) | 21
- tuple **FUSES_WRITE_LSBS** = (0 << 8) | 22
- tuple **FUSES_WRITE_MSBS** = (0 << 8) | 23
- tuple **FUSES_READ_LSBS** = (0 << 8) | 24
- tuple **FUSES_READ_MSBS** = (0 << 8) | 25
- tuple **TEMPERATURE_SENSOR** = (0 << 8) | 26
- tuple **DMU_AND_TRU_STATE** = (4 << 8) | 0
- tuple **PIXEL_CFG_1** = (5 << 8) | 0
- tuple **PIXEL_CFG_2** = (5 << 8) | 1
- tuple **PIXEL_CFG_3** = (5 << 8) | 2
- tuple **ANALOG_MONITOR_AND_OVERRIDE** = (6 << 8) | 0
- tuple **VRESETP** = (6 << 8) | 1
- tuple **VRESETD** = (6 << 8) | 2
- tuple **VCASP** = (6 << 8) | 3
- tuple **VCASN** = (6 << 8) | 4
- tuple **VPULSEH** = (6 << 8) | 5
- tuple **VPULSEL** = (6 << 8) | 6
- tuple **VCASN2** = (6 << 8) | 7
- tuple **VCLIP** = (6 << 8) | 8

- tuple **VTEMP** = (6 << 8) | 9
- tuple **IAUX2** = (6 << 8) | 10
- tuple **IRESET** = (6 << 8) | 11
- tuple **IDB** = (6 << 8) | 12
- tuple **IBIAS** = (6 << 8) | 13
- tuple **ITHR** = (6 << 8) | 14
- tuple **BUFFER_BYPASS** = (6 << 8) | 15

5.2.1 Detailed Description

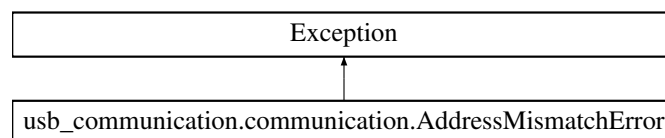
Addresses supported by ALPIDE (NOTE that only region independent addresses are supported here)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/pALPIDE3.py

5.3 usb_communication.communication.AddressMismatchError Class Reference

Inheritance diagram for usb_communication.communication.AddressMismatchError:



Public Member Functions

- def **__init__**(self, value="", requested_address=0, rd_address=0, rderr_flag=0)
- def **__str__**(self)
- def **__requested_address__**(self)
- def **__rd_address__**(self)
- def **__rderr_flag__**(self)

Public Attributes

- **value**
- **requested_address**
- **rd_address**
- **rderr_flag**

5.3.1 Detailed Description

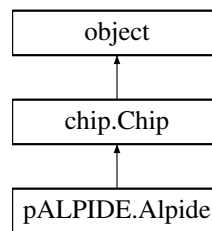
basic class to define an error mismatch between the address requested by the read command and the one received in the package read from the board

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py

5.4 pALPIDE.Alpide Class Reference

Inheritance diagram for pALPIDE.Alpide:



Public Member Functions

- def **__init__** (self, board, chipid, readback=False, log=False, commitTransaction=True, chip_number=None)
- def **reset** (self)
- def **initialize** (self, disable_manchester=1, grst=True, grst_use_opcode=True, cfg_ob_module=False)
- def **configure_dtu** (self, PLLDAC=None, DriverDAC=None, PreDAC=None, PLLDelayStages=None, LockWaitCycles=None, UnlockWaitCycles=None, commitTransaction=None, log=None, readback=None)
- def **initialize_readout** (self, PLLDAC=None, DriverDAC=None, PreDAC=None, PLLDelayStages=None, commitTransaction=None, log=None, readback=None)
- def **initialize_readout_slave** (self, commitTransaction=None, log=None, readback=None)
- def **reset_pll** (self)
- def **trigger** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **pulse** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **clear_cmerrors** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **region_control_register_mask_all_double_columns** (self, broadcast=False, readback=None, log=None, commitTransaction=None)
- def **region_control_register_unmask_all_double_columns** (self, broadcast=True, readback=None, log=None, commitTransaction=None)
- def **region_control_register_pulse_gating** (self, region, enabled_doublecolumns, log=False, commitTransaction=True)
- def **init_pixel_matrix** (self, readback=None, log=None, commitTransaction=None)
- *MATRIX #.*
- def **mask_all_pixels** (self, readback=None, log=None, commitTransaction=None)
- def **unmask_all_pixels** (self, readback=None, log=None, commitTransaction=None)
- def **pulse_all_pixels_enable** (self, readback=None, log=None, commitTransaction=None)
- def **pulse_all_pixels_disable** (self, readback=None, log=None, commitTransaction=None)
- def **unmask_reg** (self, reg, readback=None, log=None, commitTransaction=None)
- def **unmask_row** (self, row, readback=None, log=None, commitTransaction=None)
- def **mask_row** (self, row, readback=None, log=None, commitTransaction=None)
- def **unmask_col** (self, col, readback=None, log=None, commitTransaction=None)
- def **unmask_pixel** (self, pixel_coordinates, readback=None, log=None, commitTransaction=None)
- def **pulse_pixel_enable** (self, pixel_coordinates, readback=None, log=None, commitTransaction=None)
- def **pulse_row_enable** (self, row, readback=None, log=None, commitTransaction=None)
- def **pulse_row_disable** (self, row, readback=None, log=None, commitTransaction=None)
- def **write_read_test** (self, repeat, address=Addr.DTU_TEST_2, quiet=False)
- *TEST #.*
- def **set_default_variables** (self, commitTransaction, log, readback=None)
- def **pattern30** (self, pattern)
- def **propagate_pattern** (self, pattern)
- def **propagate_comma** (self, commitTransaction=None, readback=None, log=None, verbose=False)

- def [propagate_data](#) (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def [propagate_prbs](#) (self, PrbsRate=None, commitTransaction=None, readback=None, log=None, verbose=False)
- def [propagate_clock](#) (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def [set_xoff](#) (self, XOff=1, commitTransaction=None, readback=None, log=None, verbose=False)
- def [dump_config](#) (self)
- def [pll_check_lock](#) (self)
- def [setreg_cmd](#) (self, Command=None, commitTransaction=None, log=None, readback=None, verbose=False)
- AUTOGENERATED FUNCTIONS #.*
- def [getreg_cmd](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_mode_ctrl](#) (self, ChipModeSelector=None, EnClustering=None, MatrixROSpeed=None, IB↔SerialLinkSpeed=None, EnSkewGlobalSignals=None, EnSkewStartReadout=None, EnReadoutClock↔Gating=None, EnReadoutFromCMU=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_mode_ctrl](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_disable_regions_lsbs](#) (self, RegionDisable=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_disable_regions_lsbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_disable_regions_msbs](#) (self, RegionDisable=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_disable_regions_msbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_cfg_1](#) (self, MEBMask=None, EnStrobeGeneration=None, EnBusyMonitoring=None, PulseMode=None, EnPulse2Strobe=None, EnRotatePulseLines=None, TriggerDelay=None, commit↔Transaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_cfg_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_cfg_2](#) (self, FrameDuration=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_cfg_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_cfg_3](#) (self, FrameGap=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_cfg_3](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_pulsing1](#) (self, PulseDelay=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_pulsing1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_pulsing_2](#) (self, PulseDuration=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_pulsing_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fromu_status_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fromu_status_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fromu_status_3](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fromu_status_4](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fromu_status_5](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dac_settings_dclk_and_mclk_io_buffers](#) (self, DCLKReceiver=None, DCLKDriver=None, MCL↔KReceiver=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dac_settings_dclk_and_mclk_io_buffers](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dac_settings_cmu_io_buffers](#) (self, DCTRLReceiver=None, DCTRLDriver=None, commit↔Transaction=None, log=None, readback=None, verbose=False)
- def [getreg_dac_settings_cmu_io_buffers](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_cmu_and_dmu_cfg](#) (self, PreviousChipID=None, InitialToken=None, DisableManchester=None, EnableDDR=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_cmu_and_dmu_cfg](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_cmu_and_dmu_status](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_dmu_data_fifo_lsbs](#) (self, commitTransaction=None, log=None, verbose=False)

- def [getreg_dmu_data_fifo_msbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_cfg](#) (self, VcoDelayStages=None, PllBandwidthControl=None, PllOffSignal=None, Ser↵Phase=None, PLLReset=None, LoadENStatus=None, commitTransaction=None, log=None, read↵back=None, verbose=False)
- def [getreg_dtu_cfg](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_dacs](#) (self, PLLDAC=None, DriverDAC=None, PreDAC=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_dacs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_dtu_pll_lock_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_pll_lock_2](#) (self, LockWaitCycles=None, UnlockWaitCycles=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_pll_lock_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_test_1](#) (self, TestEN=None, IntPatternEN=None, TestSingleMode=None, PrbsRate=None, Bypass8b10b=None, BDIN8b10b0=None, BDIN8b10b1=None, BDIN8b10b2=None, K0=None, K1=None, K2=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_test_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_test_2](#) (self, DIN0=None, DIN1=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_test_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_test_3](#) (self, DIN2=None, ForceSetLoadEN=None, ForceUnsetLoadEN=None, commit↵Transaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_test_3](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_busy_min_width](#) (self, BusyMinWidth=None, commitTransaction=None, log=None, read↵back=None, verbose=False)
- def [getreg_busy_min_width](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_pixel_cfg](#) (self, PIXCNFG_REGSEL=None, PIXCNFG_DATA=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_pixel_cfg](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_analog_monitor_and_override](#) (self, VoltageDACSel=None, CurrentDACSel=None, SWCN↵TL_DACMONI=None, SWCNTL_DACMONV=None, IRefBufferCurrent=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_analog_monitor_and_override](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VRESETP](#) (self, VRESETP=None, commitTransaction=None, log=None, readback=None, ver↵bose=False)
- def [getreg_VRESETP](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VRESETD](#) (self, VRESETD=None, commitTransaction=None, log=None, readback=None, ver↵bose=False)
- def [getreg_VRESETD](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCASP](#) (self, VCASP=None, commitTransaction=None, log=None, readback=None, ver↵bose=False)
- def [getreg_VCASP](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCASN](#) (self, VCASN=None, commitTransaction=None, log=None, readback=None, ver↵bose=False)
- def [getreg_VCASN](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VPULSEH](#) (self, VPULSEH=None, commitTransaction=None, log=None, readback=None, ver↵bose=False)
- def [getreg_VPULSEH](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VPULSEL](#) (self, VPULSEL=None, commitTransaction=None, log=None, readback=None, ver↵bose=False)
- def [getreg_VPULSEL](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCASN2](#) (self, VCASN2=None, commitTransaction=None, log=None, readback=None, ver↵bose=False)
- def [getreg_VCASN2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCLIP](#) (self, VCLIP=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VCLIP](#) (self, commitTransaction=None, log=None, verbose=False)

- def `setreg_VTEMP` (self, VTEMP=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_VTEMP` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_IAUX2` (self, IAUX2=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_IAUX2` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_IRESET` (self, IRESET=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_IRESET` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_IDB` (self, IDB=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_IDB` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_IBIAS` (self, IBIAS=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_IBIAS` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_ITHR` (self, ITHR=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_ITHR` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_buffer_bypass` (self, VCASNBYPASS=None, VCASN2BYPASS=None, VCASPBYPASS=None, VCLIPBYPASS=None, IRESETBYPASS=None, IBIASBYPASS=None, ITHRBYPASS=None, IDBBYPASS=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_buffer_bypass` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_adc_ctrl` (self, Mode=None, SelInput=None, SetIComp=None, DiscriSign=None, RampSpd=None, HalfLSBTrim=None, CompOut=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_adc_ctrl` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_adc_dac_input_value_` (self, DACValue=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_adc_dac_input_value_` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_calibration_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_avss_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_dvss_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_avdd_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_dvdd_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vcasn_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vcasp_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vpulseh_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vpulsel_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vresetp_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vresetd_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vcasn2_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vclip_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_vtemp_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_ithr_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_iref_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_idb_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_ibias_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_iaux2_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_ireset_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_bg2v_value` (self, commitTransaction=None, log=None, verbose=False)
- def `getreg_adc_t2v_value` (self, commitTransaction=None, log=None, verbose=False)

Public Attributes

- `chipid`
- `logger`
- `PLLLockCounter`
- `chip_number`

5.4.1 Detailed Description

ALPIDE chip

agreed basic function structure

```

setreg_<REGNAME>(self, <list of parameters=ManualDefaults>,
                 commitTransaction=None, log=None,
                 readback=None, verbose=False):
    \"\"\"DOCSTRING\"\"\"
    <Input Assertions>
    if commitTransaction == None:
        commitTransaction = self.commitTransaction
    if log == None:
        log = self.log
    if readback == None:
        readback = self.readback

    dataawrite = <parameter assignment>
    self.write_reg(address=address,
                   data=dataawrite,
                   readback=readback,
                   log=log,
                   commitTransaction=commitTransaction,
                   verbose=verbose)

getreg_<REGNAME>(self, commitTransaction=None, log=None, verbose=False):
    \"\"\"DOCSTRING\"\"\"
    if commitTransaction == None:
        commitTransaction = self.commitTransaction
    if log == None:
        log = self.log

    dataread = self.read_reg(address=address,
                              log=log,
                              commitTransaction=commitTransaction,
                              verbose=verbose)

    <dataread unwrap in dictionary>
    ret = {"ret": dataread, ...:...}

```

5.4.2 Member Function Documentation

5.4.2.1 clear_cmerrors()

```

def pALPIDE.Alpide.clear_cmerrors (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )

```

Send a clear_cmerrors to the chip

5.4.2.2 configure_dtu()

```
def pALPIDE.Alpide.configure_dtu (
    self,
    PLLDAC = None,
    DriverDAC = None,
    PreDAC = None,
    PLLDelayStages = None,
    LockWaitCycles = None,
    UnlockWaitCycles = None,
    commitTransaction = None,
    log = None,
    readback = None )
```

Configure DTU

5.4.2.3 dump_config()

```
def pALPIDE.Alpide.dump_config (
    self )
```

Dump chip configuration

Reimplemented from [chip.Chip](#).

5.4.2.4 getreg_adc_avdd_value()

```
def pALPIDE.Alpide.getreg_adc_avdd_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC AVDD Value Register
Register fields
=====

AVDDValue	[10:0] :	0x0
-----------	----------	-----

Orgininal docstring
=====

ADC AVDD Value Register

This is an autogenerated function. Do not modify directly.

5.4.2.5 getreg_adc_avss_value_()

```
def pALPIDE.Alpide.getreg_adc_avss_value_ (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC AVSS Value Register (also used to store measure in manual mode)
Register fields

=====

AVSSValue [10] : 0x0

Original docstring

=====

ADC AVSS Value Register (also used to store measure in manual mode)

This is an autogenerated function. Do not modify directly.

5.4.2.6 getreg_adc_bg2v_value()

```
def pALPIDE.Alpide.getreg_adc_bg2v_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC BG2V Value Register
Register fields

=====

BG2VValue [10:0] : 0x0

Original docstring

=====

ADC BG2V Value Register

This is an autogenerated function. Do not modify directly.

5.4.2.7 getreg_adc_calibration_value()

```
def pALPIDE.Alpide.getreg_adc_calibration_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC Calibration Value Register
Register fields

=====

CalibrationValue [10:0] : 0x0

Original docstring

=====

ADC Calibration Value Register

This is an autogenerated function. Do not modify directly.

5.4.2.8 getreg_adc_ctrl()

```
def pALPIDE.Alpide.getreg_adc_ctrl (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC Control Register
Register fields
=====
Mode                [1:0] : 0x0
SelInput            [5:2] : 0x0
SetIComp            [7:6] : 0x0
DiscriSign          [8] : 0x0
RampSpd             [10:9] : 0x2
HalfLSBTrim         [11] : 0x0
CompOut             [15] : 0x0

Origininal docstring
=====
ADC Control Register
Mode:
0 Manual
1 Calibration
2 Auto
3 Super-Manual
Sel input values for manual modes (values not listed will select AVSS):
0 AVSS
1 DVSS
2 AVDD
3 DVDD
4 V bandgap through voltage scaling
5 DACMONV
6 DACMONI
7 bandgap (direct measurement)
8 temperature (direct measurement)
Comparator current:
0
1
2
3

Ramp speed:
0 500ns
1 1us (default)
2 2us
3 4us

CompOut is read-only

This is an autogenerated function. Do not modify directly.
```

5.4.2.9 getreg_adc_dac_input_value_()

```
def pALPIDE.Alpide.getreg_adc_dac_input_value_ (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC DAC input value (used in super-manual mode)

Register fields

=====

DACValue [10:0] : 0x0

Original docstring

=====

ADC DAC input value (used in super-manual mode)

This is an autogenerated function. Do not modify directly.

5.4.2.10 getreg_adc_dvdd_value()

```
def pALPIDE.Alpide.getreg_adc_dvdd_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC DVDD Value Register

Register fields

=====

DVDDValue [10:0] : 0x0

Original docstring

=====

ADC DVDD Value Register

This is an autogenerated function. Do not modify directly.

5.4.2.11 getreg_adc_dvss_value()

```
def pALPIDE.Alpide.getreg_adc_dvss_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC DVSS Value Register

Register fields

=====

DVSSValue [10:0] : 0x0

Original docstring

=====

ADC DVSS Value Register

This is an autogenerated function. Do not modify directly.

5.4.2.12 getreg_adc_iaux2_value()

```
def pALPIDE.Alpide.getreg_adc_iaux2_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC IAUX2 Value Register
Register fields

```
=====
IAUX2Value          [10:0] :   0x0
```

Original docstring

```
=====
ADC IAUX2 Value Register
```

This is an autogenerated function. Do not modify directly.

5.4.2.13 getreg_adc_ibias_value()

```
def pALPIDE.Alpide.getreg_adc_ibias_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC IBIAS Value Register
Register fields

```
=====
IBIASValue          [10:0] :   0x0
```

Original docstring

```
=====
ADC IBIAS Value Register
```

This is an autogenerated function. Do not modify directly.

5.4.2.14 getreg_adc_idb_value()

```
def pALPIDE.Alpide.getreg_adc_idb_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC IDB Value Register
Register fields

```
=====
IDBValue            [10:0] :   0x0
```

Original docstring

```
=====
ADC IDB Value Register
```

This is an autogenerated function. Do not modify directly.

5.4.2.15 getreg_adc_iref_value()

```
def pALPIDE.Alpide.getreg_adc_iref_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC IREF Value Register
Register fields

```
=====
    IREFValue          [10:0] :    0x0
```

Original docstring

```
=====
ADC IREF Value Register
```

This is an autogenerated function. Do not modify directly.

5.4.2.16 getreg_adc_ireset_value()

```
def pALPIDE.Alpide.getreg_adc_ireset_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC IRESET Value Register
Register fields

```
=====
    IRESETValue        [10:0] :    0x0
```

Original docstring

```
=====
ADC IRESET Value Register
```

This is an autogenerated function. Do not modify directly.

5.4.2.17 getreg_adc_ithr_value()

```
def pALPIDE.Alpide.getreg_adc_ithr_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for ADC ITHR Value Register
Register fields

```
=====
    ITHRValue          [10:0] :    0x0
```

Original docstring

```
=====
ADC ITHR Value Register
```

This is an autogenerated function. Do not modify directly.

5.4.2.18 getreg_adc_t2v_value()

```
def pALPIDE.Alpide.getreg_adc_t2v_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC T2V Value Register
Register fields
=====
    T2VValue           [10:0] :    0x0

Origininal docstring
=====
ADC T2V Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.19 getreg_adc_vcasn2_value()

```
def pALPIDE.Alpide.getreg_adc_vcasn2_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VCASN2 Value Register
Register fields
=====
    VCASN2Value        [10:0] :    0x0

Origininal docstring
=====
ADC VCASN2 Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.20 getreg_adc_vcasn_value()

```
def pALPIDE.Alpide.getreg_adc_vcasn_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VCASN Value Register
Register fields
=====
    VCASNValue         [10:0] :    0x0

Origininal docstring
=====
ADC VCASN Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.21 getreg_adc_vcasp_value()

```
def pALPIDE.Alpide.getreg_adc_vcasp_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VCASP Value Register
Register fields
=====
    VCASPValue          [10:0] :    0x0

Origininal docstring
=====
ADC VCASP Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.22 getreg_adc_vclip_value()

```
def pALPIDE.Alpide.getreg_adc_vclip_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VCLIP Value Register
Register fields
=====
    VCLIPValue          [10:0] :    0x0

Origininal docstring
=====
ADC VCLIP Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.23 getreg_adc_vpulseh_value()

```
def pALPIDE.Alpide.getreg_adc_vpulseh_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VPULSEH Value Register
Register fields
=====
    VPULSEHValue        [10:0] :    0x0

Origininal docstring
=====
ADC VPULSEH Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.24 getreg_adc_vpulsel_value()

```
def pALPIDE.Alpide.getreg_adc_vpulsel_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VPULSE Value Register
Register fields
=====
    VPULSEValue           [10:0] :    0x0

Origininal docstring
=====
ADC VPULSE Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.25 getreg_adc_vresetd_value()

```
def pALPIDE.Alpide.getreg_adc_vresetd_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VRESETD Value Register
Register fields
=====
    VRESETDValue          [10:0] :    0x0

Origininal docstring
=====
ADC VRESETD Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.26 getreg_adc_vresetp_value()

```
def pALPIDE.Alpide.getreg_adc_vresetp_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VRESETP Value Register
Register fields
=====
    VRESETPValue          [10:0] :    0x0

Origininal docstring
=====
ADC VRESETP Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.27 getreg_adc_vtemp_value()

```
def pALPIDE.Alpide.getreg_adc_vtemp_value (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for ADC VTEMP Value Register
Register fields
=====
    VTEMPValue          [10:0] :    0x0

Origininal docstring
=====
ADC VTEMP Value Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.28 getreg_analog_monitor_and_override()

```
def pALPIDE.Alpide.getreg_analog_monitor_and_override (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Analog Monitor and Override Register
Register fields
=====
    VoltageDACSel        [3:0] :    0x0
    CurrentDACSel        [6:4] :    0x0
    SWCNTL_DACMONI       [7] :    0x0
    SWCNTL_DACMONV       [8] :    0x0
    IRefBufferCurrent     [10:9] :   0x2

Origininal docstring
=====
Analog Monitor and Override Register
DACMONI current selection (ignored if ADC in Auto mode):
0  IRESET
1  IAUX2
2  IBIAS
3  IDB
4  IREF
5  ITHR
6  IREFBuffer

DACMONV voltage selection (ignored if ADC in Auto mode):
0  VCASN
1  VCASP
2  VPULSEH
3  VPULSEL
4  VRESETP
5  VRESETD
6  VCASN2
7  VCLIP
8  VTEMP
9  ADCDAC

SWCNTL_DACMONV/I are ignored if ADC in Auto mode
```

```

IRefBuffer current setting:
0    0,25uA
1    0,75uA
2    1,00uA (default)
3    1,25uA

```

This is an autogenerated function. Do not modify directly.

5.4.2.29 getreg_buffer_bypass()

```

def pALPIDE.Alpide.getreg_buffer_bypass (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

```

Autogenerated function for Buffer Bypass Register

Register fields
=====

VCASNBYPASS	[0] :	0x0
VCASN2BYPASS	[1] :	0x0
VCASPBYPASS	[2] :	0x0
VCLIPBYPASS	[3] :	0x0
IRESETBYPASS	[4] :	0x0
IBIASBYPASS	[5] :	0x0
ITHRBYPASS	[6] :	0x0
IDBBYPASS	[7] :	0x0

Original docstring

=====

Buffer Bypass Register

This is an autogenerated function. Do not modify directly.

5.4.2.30 getreg_busy_min_width()

```

def pALPIDE.Alpide.getreg_busy_min_width (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

```

Autogenerated function for BUSY min width

Register fields
=====

BusyMinWidth	[4:0] :	0x8
--------------	---------	-----

Original docstring

=====

BUSY min width

This is an autogenerated function. Do not modify directly.

5.4.2.31 getreg_cmd()

```
def pALPIDE.Alpide.getreg_cmd (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for Command Register.

Register fields

=====

Command	[15:0] :	0x0
---------	----------	-----

Original docstring

=====

Command Register.

Valid opcodes:

16'h00D2	GRST
16'h00E4	PRST
16'h0078	PULSE
16'h0036	BCRST
16'h0063	RORST
16'h00AA	DEBUG
16'h00B1, 0055, 00C9, 002D	TRIGGER
16'h009C	WROP
16'h004E	RDOP
16'hFF00	CMUCLRERR
16'hFF01	FIFOTEST
16'hFF02	LOADOBDEF CFG
16'hFF10	XOFF
16'hFF11	XON
16'hFF20	ADCMEASURE

This is an autogenerated function. Do not modify directly.

5.4.2.32 getreg_cmu_and_dmu_cfg()

```
def pALPIDE.Alpide.getreg_cmu_and_dmu_cfg (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for CMU and DMU Configuration Register

Register fields

=====

PreviousChipID	[3:0] :	0xf
InitialToken	[4] :	0x0
DisableManchester	[5] :	0x0
EnableDDR	[6] :	0x1

Original docstring

=====

CMU and DMU Configuration Register

This is an autogenerated function. Do not modify directly.

5.4.2.33 getreg_cmu_and_dmu_status()

```
def pALPIDE.Alpide.getreg_cmu_and_dmu_status (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for CMU and DMU Status Register
Register fields
=====
    CMUErrorsCounter      [3:0] :    0x0
    CMUTimeOutCounter     [7:4] :    0x0
    CMUOpCounter          [11:8] :    0x0

Original docstring
=====
CMU and DMU Status Register

This is an autogenerated function. Do not modify directly.
```

5.4.2.34 getreg_dac_settings_cmu_io_buffers()

```
def pALPIDE.Alpide.getreg_dac_settings_cmu_io_buffers (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for DAC settings for CMU I/O buffers
Register fields
=====
    DCTRLReceiver         [3:0] :    0xa
    DCTRLDriver           [7:4] :    0xa

Original docstring
=====
DAC settings for CMU I/O buffers

This is an autogenerated function. Do not modify directly.
```

5.4.2.35 getreg_dac_settings_dclk_and_mclk_io_buffers()

```
def pALPIDE.Alpide.getreg_dac_settings_dclk_and_mclk_io_buffers (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DAC settings for DCLK and MCLK I/O buffers
Register fields

=====

```
DCLKReceiver          [3:0] :    0xa
DCLKDriver            [7:4] :    0xa
MCLKReceiver          [11:8] :   0xa
```

Original docstring

=====

DAC settings for DCLK and MCLK I/O buffers

This is an autogenerated function. Do not modify directly.

5.4.2.36 getreg_disable_regions_lsbs()

```
def pALPIDE.Alpide.getreg_disable_regions_lsbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for Disable of regions 0-15

Register fields

=====

```
RegionDisable          [15:0] :   0x0
```

Original docstring

=====

Disable of regions 0-15

This is an autogenerated function. Do not modify directly.

5.4.2.37 getreg_disable_regions_msbs()

```
def pALPIDE.Alpide.getreg_disable_regions_msbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for Disable of regions 16-31

Register fields

=====

```
RegionDisable          [15:0] :   0x0
```

Original docstring

=====

Disable of regions 16-31

This is an autogenerated function. Do not modify directly.

5.4.2.38 getreg_dmu_data_fifo_lsbs()

```
def pALPIDE.Alpide.getreg_dmu_data_fifo_lsbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for DMU Data FIFO [15:0]
Register fields
=====
    DMUDataFifoLSB          [15:0] :    0x0

Original docstring
=====
DMU Data FIFO [15:0]
Accessible when EnReadoutFromCMU=1

This is an autogenerated function. Do not modify directly.
```

5.4.2.39 getreg_dmu_data_fifo_msbs()

```
def pALPIDE.Alpide.getreg_dmu_data_fifo_msbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for DMU Data FIFO [23:16]
Register fields
=====
    DMUDataFifoMSB          [7:0] :    0x0

Original docstring
=====
DMU Data FIFO [23:16]
Read operation will pop word from FIFO
Accessible when EnReadoutFromCMU=1

This is an autogenerated function. Do not modify directly.
```

5.4.2.40 getreg_dtu_cfg()

```
def pALPIDE.Alpide.getreg_dtu_cfg (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU Configuration Register
Register fields

=====

VcoDelayStages	[1:0] :	0x1
PllBandwidthControl	[2] :	0x1
PllOffSignal	[3] :	0x1
SerPhase	[7:4] :	0x8
PLLReset	[8] :	0x0
LoadENStatus	[12] :	0x0

Original docstring

=====

DTU Configuration Register

LoadENStatus is read-only

This is an autogenerated function. Do not modify directly.

5.4.2.41 getreg_dtu_dacs()

```
def pALPIDE.Alpide.getreg_dtu_dacs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU DACs Register

Register fields

=====

PLLDAC	[3:0] :	0x8
DriverDAC	[7:4] :	0x8
PreDAC	[11:8] :	0x0

Original docstring

=====

DTU DACs Register

This is an autogenerated function. Do not modify directly.

5.4.2.42 getreg_dtu_pll_lock_1()

```
def pALPIDE.Alpide.getreg_dtu_pll_lock_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU PLL Lock Register 1

Register fields

=====

LockCounter	[7:0] :	0x0
LockFlag	[8] :	0x0
LockStatus	[9] :	0x0

Original docstring

=====

DTU PLL Lock Register 1

This is an autogenerated function. Do not modify directly.

5.4.2.43 getreg_dtu_pll_lock_2()

```
def pALPIDE.Alpide.getreg_dtu_pll_lock_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU PLL Lock Register 2
Register fields

=====

LockWaitCycles	[7:0] :	0x0
UnlockWaitCycles	[15:8] :	0x0

Original docstring

=====

DTU PLL Lock Register 2

This is an autogenerated function. Do not modify directly.

5.4.2.44 getreg_dtu_test_1()

```
def pALPIDE.Alpide.getreg_dtu_test_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 1
Register fields

=====

TestEN	[0] :	0x0
IntPatternEN	[1] :	0x0
TestSingleMode	[2] :	0x0
PrbsRate	[4:3] :	0x0
Bypass8b10b	[5] :	0x0
BDIN8b10b0	[7:6] :	0x0
BDIN8b10b1	[9:8] :	0x0
BDIN8b10b2	[11:10] :	0x0
K0	[12] :	0x0
K1	[13] :	0x0
K2	[14] :	0x0

Original docstring

=====

DTU Test Register 1

This is an autogenerated function. Do not modify directly.

5.4.2.45 getreg_dtu_test_2()

```
def pALPIDE.Alpide.getreg_dtu_test_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 2
Register fields

=====

DIN0	[7:0] :	0x0
DIN1	[15:8] :	0x0

Original docstring

=====

DTU Test Register 2

This is an autogenerated function. Do not modify directly.

5.4.2.46 getreg_dtu_test_3()

```
def pALPIDE.Alpide.getreg_dtu_test_3 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 3
Register fields

=====

DIN2	[7:0] :	0x0
ForceSetLoadEN	[8] :	0x0
ForceUnsetLoadEN	[9] :	0x0

Original docstring

=====

DTU Test Register 3

This is an autogenerated function. Do not modify directly.

5.4.2.47 getreg_fromu_cfg_1()

```
def pALPIDE.Alpide.getreg_fromu_cfg_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 1
Register fields

=====

MEBMask	[2:0] :	0x0
EnStrobeGeneration	[3] :	0x0
EnBusyMonitoring	[4] :	0x1
PulseMode	[5] :	0x0
EnPulse2Strobe	[6] :	0x0
EnRotatePulseLines	[7] :	0x0
TriggerDelay	[10:8] :	0x0

Original docstring

=====

FROMU Configuration Register 1

EnBusyMonitoring default = 1

Trigger delay, 0-175ns

This is an autogenerated function. Do not modify directly.

5.4.2.48 getreg_fromu_cfg_2()

```
def pALPIDE.Alpide.getreg_fromu_cfg_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 2
Register fields

=====

FrameDuration	[15:0] :	0x14
---------------	----------	------

Original docstring

=====

FROMU Configuration Register 2

This is an autogenerated function. Do not modify directly.

5.4.2.49 getreg_fromu_cfg_3()

```
def pALPIDE.Alpide.getreg_fromu_cfg_3 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 3
Register fields

=====

FrameGap	[15:0] :	0x0
----------	----------	-----

Original docstring

=====

FROMU Configuration Register 3

This is an autogenerated function. Do not modify directly.

5.4.2.50 getreg_fromu_pulsing1()

```
def pALPIDE.Alpide.getreg_fromu_pulsing1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Pulsing Register1
Register fields
=====
    PulseDelay          [15:0] :    0x0

Origininal docstring
=====
FROMU Pulsing Register1

This is an autogenerated function. Do not modify directly.
```

5.4.2.51 getreg_fromu_pulsing_2()

```
def pALPIDE.Alpide.getreg_fromu_pulsing_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Pulsing Register 2
Register fields
=====
    PulseDuration       [15:0] :    0x0

Origininal docstring
=====
FROMU Pulsing Register 2

This is an autogenerated function. Do not modify directly.
```

5.4.2.52 getreg_fromu_status_1()

```
def pALPIDE.Alpide.getreg_fromu_status_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Status Register 1
Register fields
=====
    TriggerCounter      [15:0] :    0x0

Origininal docstring
=====
FROMU Status Register 1

This is an autogenerated function. Do not modify directly.
```

5.4.2.53 getreg_fromu_status_2()

```
def pALPIDE.Alpide.getreg_fromu_status_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Status Register 2
Register fields
=====
    StrobeCounter          [15:0] :    0x0

Origininal docstring
=====
FROMU Status Register 2

This is an autogenerated function. Do not modify directly.
```

5.4.2.54 getreg_fromu_status_3()

```
def pALPIDE.Alpide.getreg_fromu_status_3 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Status Register 3
Register fields
=====
    ReadoutCounter         [15:0] :    0x0

Origininal docstring
=====
FROMU Status Register 3

This is an autogenerated function. Do not modify directly.
```

5.4.2.55 getreg_fromu_status_4()

```
def pALPIDE.Alpide.getreg_fromu_status_4 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Status Register 4
Register fields
=====
    FrameCounter           [15:0] :    0x0

Origininal docstring
=====
FROMU Status Register 4

This is an autogenerated function. Do not modify directly.
```

5.4.2.56 getreg_fromu_status_5()

```
def pALPIDE.Alpide.getreg_fromu_status_5 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Status Register 5
Register fields
=====
    BunchCounter          [11:0] :    0x0
    EventCounter          [13:12] :    0x0
    FrameExtended         [14] :    0x0

Origininal docstring
=====
FROMU Status Register 5

This is an autogenerated function. Do not modify directly.
```

5.4.2.57 getreg_IAUX2()

```
def pALPIDE.Alpide.getreg_IAUX2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    IAUX2                  [7:0] :    0x0

Origininal docstring
=====

This is an autogenerated function. Do not modify directly.
```

5.4.2.58 getreg_IBIAS()

```
def pALPIDE.Alpide.getreg_IBIAS (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    IBIAS                  [7:0] :    0x40

Origininal docstring
=====

This is an autogenerated function. Do not modify directly.
```

5.4.2.59 getreg_IDB()

```
def pALPIDE.Alpide.getreg_IDB (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

```
IDB                                [7:0] : 0x40
```

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.60 getreg_IRESET()

```
def pALPIDE.Alpide.getreg_IRESET (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

```
IRESET                            [7:0] : 0x32
```

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.61 getreg_ITHR()

```
def pALPIDE.Alpide.getreg_ITHR (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

```
ITHR                               [7:0] : 0x33
```

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.62 getreg_mode_ctrl()

```
def pALPIDE.Alpide.getreg_mode_ctrl (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Mode Control register
Register fields
=====
    ChipModeSelector          [1:0] : 0x0
    EnClustering               [2] : 0x1
    MatrixROSpeed              [3] : 0x1
    IBSerialLinkSpeed          [5:4] : 0x3
    EnSkewGlobalSignals        [6] : 0x1
    EnSkewStartReadout         [7] : 0x1
    EnReadoutClockGating       [8] : 0x1
    EnReadoutFromCMU           [9] : 0x0

Original docstring
=====
Mode Control register
Chip Mode Selector values:
0=Cfg Mode, 1=Triggered Mode, 2=Continuous Mode
EnClustering default = 1
MatrixROSpeed 0=10MHz, 1=20MHz (default)
IBSerialLinkSpeed 0=400Mbit, 1=600Mbit, 2-3=1200Mbit (default)
EnSkewGlobalSignals default = 1
EnSkewStartReadout default = 1
EnClockGating default = 1

This is an autogenerated function. Do not modify directly.
```

5.4.2.63 getreg_pixel_cfg()

```
def pALPIDE.Alpide.getreg_pixel_cfg (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Pixel Configuration Register
Register fields
=====
    PIXCNFG_REGSEL            [0] : 0x0
    PIXCNFG_DATA               [1] : 0x0

Original docstring
=====
Pixel Configuration Register
RegSel 0 = MASK, RegSel 1 = PULSE

This is an autogenerated function. Do not modify directly.
```

5.4.2.64 getreg_VCASN()

```
def pALPIDE.Alpide.getreg_VCASN (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCASN                [7:0] : 0x39

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.4.2.65 getreg_VCASN2()

```
def pALPIDE.Alpide.getreg_VCASN2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCASN2               [7:0] : 0x40

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.4.2.66 getreg_VCASP()

```
def pALPIDE.Alpide.getreg_VCASP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCASP                [7:0] : 0x56

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.4.2.67 getreg_VCLIP()

```
def pALPIDE.Alpide.getreg_VCLIP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCLIP                [7:0] :    0x0

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.4.2.68 getreg_VPULSEH()

```
def pALPIDE.Alpide.getreg_VPULSEH (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VPULSEH              [7:0] :   0xff

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.4.2.69 getreg_VPULSEL()

```
def pALPIDE.Alpide.getreg_VPULSEL (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VPULSEL              [7:0] :    0x0

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.4.2.70 getreg_VRESETD()

```
def pALPIDE.Alpide.getreg_VRESETD (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
VRESETD                [7:0] :   0x0
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.71 getreg_VRESETP()

```
def pALPIDE.Alpide.getreg_VRESETP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
VRESETP                [7:0] :  0x75
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.72 getreg_VTEMP()

```
def pALPIDE.Alpide.getreg_VTEMP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
VTEMP                  [7:0] :   0x0
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.73 init_pixel_matrix()

```
def pALPIDE.Alpide.init_pixel_matrix (
    self,
    readback = None,
    log = None,
    commitTransaction = None )
```

MATRIX #.

init the matrix by unmasking and disabling pulse

5.4.2.74 initialize()

```
def pALPIDE.Alpide.initialize (
    self,
    disable_manchester = 1,
    grst = True,
    grst_use_opcode = True,
    cfg_ob_module = False )
```

Perform chip initialization procedure ([grst], manchester, RORST)

5.4.2.75 initialize_readout()

```
def pALPIDE.Alpide.initialize_readout (
    self,
    PLLDAC = None,
    DriverDAC = None,
    PreDAC = None,
    PLLDelayStages = None,
    commitTransaction = None,
    log = None,
    readback = None )
```

Perform initialization sequence for readout

5.4.2.76 initialize_readout_slave()

```
def pALPIDE.Alpide.initialize_readout_slave (
    self,
    commitTransaction = None,
    log = None,
    readback = None )
```

Perform readout sequence for Slave chip

5.4.2.77 mask_all_pixels()

```
def pALPIDE.Alpide.mask_all_pixels (
    self,
    readback = None,
    log = None,
    commitTransaction = None )
```

masks all the pixels

5.4.2.78 mask_row()

```
def pALPIDE.Alpide.mask_row (
    self,
    row,
    readback = None,
    log = None,
    commitTransaction = None )
```

masks the row in the given reg

5.4.2.79 pattern30()

```
def pALPIDE.Alpide.pattern30 (
    self,
    pattern )
```

Propagate 30 bit pattern

5.4.2.80 propagate_clock()

```
def pALPIDE.Alpide.propagate_clock (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate 600 MHz clock from chip

Reimplemented from [chip.Chip](#).

5.4.2.81 propagate_comma()

```
def pALPIDE.Alpide.propagate_comma (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate COMMA pattern from chip (for realignment)

Reimplemented from [chip.Chip](#).

5.4.2.82 propagate_data()

```
def pALPIDE.Alpide.propagate_data (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate Data from chip

Reimplemented from [chip.Chip](#).

5.4.2.83 propagate_prbs()

```
def pALPIDE.Alpide.propagate_prbs (
    self,
    PrbsRate = None,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate PRBS test pattern from chip

5.4.2.84 pulse()

```
def pALPIDE.Alpide.pulse (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Send a trigger to the chip

5.4.2.85 pulse_all_pixels_disable()

```
def pALPIDE.Alpide.pulse_all_pixels_disable (
    self,
    readback = None,
    log = None,
    commitTransaction = None )
```

enable pulse for all pixels

5.4.2.86 pulse_all_pixels_enable()

```
def pALPIDE.Alpide.pulse_all_pixels_enable (
    self,
    readback = None,
    log = None,
    commitTransaction = None )
```

enable pulse for all pixels

5.4.2.87 pulse_pixel_enable()

```
def pALPIDE.Alpide.pulse_pixel_enable (
    self,
    pixel_coordinates,
    readback = None,
    log = None,
    commitTransaction = None )
```

Enable pulse for the given pixels

5.4.2.88 pulse_row_disable()

```
def pALPIDE.Alpide.pulse_row_disable (
    self,
    row,
    readback = None,
    log = None,
    commitTransaction = None )
```

disables the pulse for the row in the given reg

5.4.2.89 pulse_row_enable()

```
def pALPIDE.Alpide.pulse_row_enable (
    self,
    row,
    readback = None,
    log = None,
    commitTransaction = None )
```

disables the pulse for the row in the given reg

5.4.2.90 region_control_register_mask_all_double_columns()

```
def pALPIDE.Alpide.region_control_register_mask_all_double_columns (
    self,
    broadcast = False,
    readback = None,
    log = None,
    commitTransaction = None )
```

Set the region control registers to 1: disable reading all the columns

5.4.2.91 region_control_register_pulse_gating()

```
def pALPIDE.Alpide.region_control_register_pulse_gating (
    self,
    region,
    enabled_doublecolumns,
    log = False,
    commitTransaction = True )
```

Set the pulse gating Control register of a region. Expects a 16 bit mask for enabling a double columns within a region

5.4.2.92 region_control_register_unmask_all_double_columns()

```
def pALPIDE.Alpide.region_control_register_unmask_all_double_columns (
    self,
    broadcast = True,
    readback = None,
    log = None,
    commitTransaction = None )
```

Set the region control registers to 0: disable reading all the columns

5.4.2.93 reset()

```
def pALPIDE.Alpide.reset (
    self )
```

Reset the chip

Reimplemented from [chip.Chip](#).

5.4.2.94 reset_pll()

```
def pALPIDE.Alpide.reset_pll (
    self )
```

Reset the pll

5.4.2.95 set_default_variables()

```
def pALPIDE.Alpide.set_default_variables (
    self,
    commitTransaction,
    log,
    readback = None )
```

Set default variables for write/read operations (resolve None)

5.4.2.96 `set_xoff()`

```
def pALPIDE.Alpide.set_xoff (
    self,
    XOff = 1,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Send XOFF to the chip. 1 -> Xoff, 0 -> Xon

Reimplemented from [chip.Chip](#).

5.4.2.97 `setreg_adc_ctrl()`

```
def pALPIDE.Alpide.setreg_adc_ctrl (
    self,
    Mode = None,
    SelInput = None,
    SetIComp = None,
    DiscriSign = None,
    RampSpd = None,
    HalfLSBTrim = None,
    CompOut = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for ADC Control Register
Register fields

=====

Mode	[1:0] :	0x0
SelInput	[5:2] :	0x0
SetIComp	[7:6] :	0x0
DiscriSign	[8] :	0x0
RampSpd	[10:9] :	0x2
HalfLSBTrim	[11] :	0x0
CompOut	[15] :	0x0

Original docstring

=====

ADC Control Register

Mode:

```
0 Manual
1 Calibration
2 Auto
3 Super-Manual
```

Sel input values for manual modes (values not listed will select AVSS):

```
0 AVSS
1 DVSS
2 AVDD
3 DVDD
4 V bandgap through voltage scaling
5 DACMONV
6 DACMONI
7 bandgap (direct measurement)
```

```

8    temperature (direct measurement)
Comparator current:
0
1
2
3

```

```

Ramp speed:
0    500ns
1    1us (default)
2    2us
3    4us

```

CompOut is read-only

This is an autogenerated function. Do not modify directly.

5.4.2.98 setreg_adc_dac_input_value_()

```

def pALPIDE.Alpide.setreg_adc_dac_input_value_ (
    self,
    DACValue = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

```

Autogenerated function for ADC DAC input value (used in super-manual mode)

Register fields
=====

```

DACValue          [10:0] :   0x0

```

Original docstring

=====

ADC DAC input value (used in super-manual mode)

This is an autogenerated function. Do not modify directly.

5.4.2.99 setreg_analog_monitor_and_override()

```

def pALPIDE.Alpide.setreg_analog_monitor_and_override (
    self,
    VoltageDACSel = None,
    CurrentDACSel = None,
    SWCNTL_DACMONI = None,
    SWCNTL_DACMONV = None,
    IRefBufferCurrent = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

```

Autogenerated function for Analog Monitor and Override Register
Register fields

=====

VoltageDACSel	[3:0]	: 0x0
CurrentDACSel	[6:4]	: 0x0
SWCNTL_DACMONI	[7]	: 0x0
SWCNTL_DACMONV	[8]	: 0x0
IRefBufferCurrent	[10:9]	: 0x2

Orgininal docstring

=====

Analog Monitor and Override Register

DACMONI current selection (ignored if ADC in Auto mode):

0	IRESET
1	IAUX2
2	IBIAS
3	IDB
4	IREF
5	ITHR
6	IREFBuffer

DACMONV voltage selection (ignored if ADC in Auto mode):

0	VCASN
1	VCASP
2	VPULSEH
3	VPULSEL
4	VRESETP
5	VRESETD
6	VCASN2
7	VCLIP
8	VTEMP
9	ADCDAC

SWCNTL_DACMONV/I are ignored if ADC in Auto mode

IRefBuffer current setting:

0	0,25uA
1	0,75uA
2	1,00uA (default)
3	1,25uA

This is an autogenerated function. Do not modify directly.

5.4.2.100 setreg_buffer_bypass()

```
def pALPIDE.Alpide.setreg_buffer_bypass (
    self,
    VCASNBYPASS = None,
    VCASN2BYPASS = None,
    VCASPBYPASS = None,
    VCLIPBYPASS = None,
    IRESETBYPASS = None,
    IBIASBYPASS = None,
    ITHRBYPASS = None,
    IDBBYPASS = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Buffer Bypass Register
Register fields

```

=====
VCASNBYPASS          [0] : 0x0
VCASN2BYPASS         [1] : 0x0
VCASPBYPASS          [2] : 0x0
VCLIPBYPASS          [3] : 0x0
IRESETBYPASS         [4] : 0x0
IBIASBYPASS          [5] : 0x0
ITHRBYPASS           [6] : 0x0
IDBBYPASS            [7] : 0x0

Origininal docstring
=====
Buffer Bypass Register

This is an autogenerated function. Do not modify directly.

```

5.4.2.101 setreg_busy_min_width()

```

def pALPIDE.Alpide.setreg_busy_min_width (
    self,
    BusyMinWidth = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for BUSY min width
Register fields
=====
    BusyMinWidth          [4:0] : 0x8

Origininal docstring
=====
BUSY min width

This is an autogenerated function. Do not modify directly.

```

5.4.2.102 setreg_cmd()

```

def pALPIDE.Alpide.setreg_cmd (
    self,
    Command = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

```

AUTOGENERATED FUNCTIONS #.

Autogenerated function for Command Register.

Register fields

=====

Command [15:0] : 0x0

Original docstring

=====

Command Register.

Valid opcodes:

16'h00D2	GRST
16'h00E4	PRST
16'h0078	PULSE
16'h0036	BCRST
16'h0063	RORST
16'h00AA	DEBUG
16'h00B1,0055,00C9,002D	TRIGGER
16'h009C	WROP
16'h004E	RDOP
16'hFF00	CMUCLRERR
16'hFF01	FIFOTEST
16'hFF02	LOADOBDEF CFG
16'hFF10	XOFF
16'hFF11	XON
16'hFF20	ADCMEASURE

This is an autogenerated function. Do not modify directly.

5.4.2.103 setreg_cmu_and_dmu_cfg()

```
def pALPIDE.Alpide.setreg_cmu_and_dmu_cfg (
    self,
    PreviousChipID = None,
    InitialToken = None,
    DisableManchester = None,
    EnableDDR = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for CMU and DMU Configuration Register

Register fields

=====

PreviousChipID	[3:0] : 0xf
InitialToken	[4] : 0x0
DisableManchester	[5] : 0x0
EnableDDR	[6] : 0x1

Original docstring

=====

CMU and DMU Configuration Register

This is an autogenerated function. Do not modify directly.

5.4.2.104 setreg_dac_settings_cmu_io_buffers()

```
def pALPIDE.Alpide.setreg_dac_settings_cmu_io_buffers (
    self,
    DCTRLReceiver = None,
    DCTRLDriver = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for DAC settings for CMU I/O buffers
Register fields
=====
    DCTRLReceiver          [3:0] :   0xa
    DCTRLDriver            [7:4] :   0xa

Original docstring
=====
DAC settings for CMU I/O buffers

This is an autogenerated function. Do not modify directly.
```

5.4.2.105 setreg_dac_settings_dclk_and_mclk_io_buffers()

```
def pALPIDE.Alpide.setreg_dac_settings_dclk_and_mclk_io_buffers (
    self,
    DCLKReceiver = None,
    DCLKDriver = None,
    MCLKReceiver = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for DAC settings for DCLK and MCLK I/O buffers
Register fields
=====
    DCLKReceiver          [3:0] :   0xa
    DCLKDriver            [7:4] :   0xa
    MCLKReceiver          [11:8] :   0xa

Original docstring
=====
DAC settings for DCLK and MCLK I/O buffers

This is an autogenerated function. Do not modify directly.
```

5.4.2.106 setreg_disable_regions_lsbs()

```
def pALPIDE.Alpide.setreg_disable_regions_lsbs (
    self,
    RegionDisable = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for Disable of regions 0-15
Register fields
=====
    RegionDisable          [15:0] :    0x0

Origininal docstring
=====
Disable of regions 0-15

This is an autogenerated function. Do not modify directly.
```

5.4.2.107 setreg_disable_regions_msbs()

```
def pALPIDE.Alpide.setreg_disable_regions_msbs (
    self,
    RegionDisable = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for Disable of regions 16-31
Register fields
=====
    RegionDisable          [15:0] :    0x0

Origininal docstring
=====
Disable of regions 16-31

This is an autogenerated function. Do not modify directly.
```

5.4.2.108 setreg_dtu_cfg()

```
def pALPIDE.Alpide.setreg_dtu_cfg (
    self,
    VcoDelayStages = None,
    PllBandwidthControl = None,
    PllOffSignal = None,
    SerPhase = None,
    PLLReset = None,
    LoadENStatus = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU Configuration Register
Register fields

=====

VcoDelayStages	[1:0]	: 0x1
PllBandwidthControl	[2]	: 0x1
PllOffSignal	[3]	: 0x1
SerPhase	[7:4]	: 0x8
PLLReset	[8]	: 0x0
LoadENStatus	[12]	: 0x0

Original docstring

=====

DTU Configuration Register

LoadENStatus is read-only

This is an autogenerated function. Do not modify directly.

5.4.2.109 setreg_dtu_dacs()

```
def pALPIDE.Alpide.setreg_dtu_dacs (
    self,
    PLLDAC = None,
    DriverDAC = None,
    PreDAC = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU DACs Register
Register fields

=====

PLLDAC	[3:0]	: 0x8
DriverDAC	[7:4]	: 0x8
PreDAC	[11:8]	: 0x0

Original docstring

=====

DTU DACs Register

This is an autogenerated function. Do not modify directly.

5.4.2.110 setreg_dtu_pll_lock_2()

```
def pALPIDE.Alpide.setreg_dtu_pll_lock_2 (
    self,
    LockWaitCycles = None,
    UnlockWaitCycles = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU PLL Lock Register 2
Register fields

=====

LockWaitCycles	[7:0] :	0x0
UnlockWaitCycles	[15:8] :	0x0

Original docstring

=====

DTU PLL Lock Register 2

This is an autogenerated function. Do not modify directly.

5.4.2.111 setreg_dtu_test_1()

```
def pALPIDE.Alpide.setreg_dtu_test_1 (
    self,
    TestEN = None,
    IntPatternEN = None,
    TestSingleMode = None,
    PrbsRate = None,
    Bypass8b10b = None,
    BDIN8b10b0 = None,
    BDIN8b10b1 = None,
    BDIN8b10b2 = None,
    K0 = None,
    K1 = None,
    K2 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 1
Register fields

=====

TestEN	[0] :	0x0
IntPatternEN	[1] :	0x0
TestSingleMode	[2] :	0x0
PrbsRate	[4:3] :	0x0
Bypass8b10b	[5] :	0x0
BDIN8b10b0	[7:6] :	0x0
BDIN8b10b1	[9:8] :	0x0
BDIN8b10b2	[11:10] :	0x0
K0	[12] :	0x0
K1	[13] :	0x0
K2	[14] :	0x0

Original docstring

=====

DTU Test Register 1

This is an autogenerated function. Do not modify directly.

5.4.2.112 setreg_dtu_test_2()

```
def pALPIDE.Alpide.setreg_dtu_test_2 (
    self,
    DIN0 = None,
    DIN1 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 2
Register fields

=====

DIN0	[7:0] :	0x0
DIN1	[15:8] :	0x0

Original docstring

=====

DTU Test Register 2

This is an autogenerated function. Do not modify directly.

5.4.2.113 setreg_dtu_test_3()

```
def pALPIDE.Alpide.setreg_dtu_test_3 (
    self,
    DIN2 = None,
    ForceSetLoadEN = None,
    ForceUnsetLoadEN = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 3
Register fields

=====

DIN2	[7:0] :	0x0
ForceSetLoadEN	[8] :	0x0
ForceUnsetLoadEN	[9] :	0x0

Original docstring

=====

DTU Test Register 3

This is an autogenerated function. Do not modify directly.

5.4.2.114 setreg_fromu_cfg_1()

```
def pALPIDE.Alpide.setreg_fromu_cfg_1 (
    self,
    MEBMask = None,
    EnStrobeGeneration = None,
    EnBusyMonitoring = None,
    PulseMode = None,
    EnPulse2Strobe = None,
    EnRotatePulseLines = None,
    TriggerDelay = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 1
Register fields

```
=====
MEBMask                [2:0] : 0x0
EnStrobeGeneration     [3] : 0x0
EnBusyMonitoring       [4] : 0x1
PulseMode              [5] : 0x0
EnPulse2Strobe         [6] : 0x0
EnRotatePulseLines     [7] : 0x0
TriggerDelay           [10:8] : 0x0
```

Original docstring

```
=====
```

FROMU Configuration Register 1

EnBusyMonitoring default = 1

Trigger delay, 0-175ns

This is an autogenerated function. Do not modify directly.

5.4.2.115 setreg_fromu_cfg_2()

```
def pALPIDE.Alpide.setreg_fromu_cfg_2 (
    self,
    FrameDuration = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 2
Register fields

```
=====
FrameDuration          [15:0] : 0x14
```

Original docstring

```
=====
```

FROMU Configuration Register 2

This is an autogenerated function. Do not modify directly.

5.4.2.116 setreg_fromu_cfg_3()

```
def pALPIDE.Alpide.setreg_fromu_cfg_3 (
    self,
    FrameGap = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 3
 Register fields
 =====

FrameGap	[15:0] :	0x0
----------	----------	-----

Original docstring
 =====

FROMU Configuration Register 3

This is an autogenerated function. Do not modify directly.

5.4.2.117 setreg_fromu_pulsing1()

```
def pALPIDE.Alpide.setreg_fromu_pulsing1 (
    self,
    PulseDelay = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for FROMU Pulsing Register1
 Register fields
 =====

PulseDelay	[15:0] :	0x0
------------	----------	-----

Original docstring
 =====

FROMU Pulsing Register1

This is an autogenerated function. Do not modify directly.

5.4.2.118 setreg_fromu_pulsing_2()

```
def pALPIDE.Alpide.setreg_fromu_pulsing_2 (
    self,
    PulseDuration = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

```

Autogenerated function for FROMU Pulsing Register 2
Register fields
=====
    PulseDuration          [15:0] :    0x0

Origininal docstring
=====
FROMU Pulsing Register 2

This is an autogenerated function. Do not modify directly.

```

5.4.2.119 setreg_IAUX2()

```

def pALPIDE.Alpide.setreg_IAUX2 (
    self,
    IAUX2 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    IAUX2                  [7:0] :    0x0

Origininal docstring
=====

This is an autogenerated function. Do not modify directly.

```

5.4.2.120 setreg_IBIAS()

```

def pALPIDE.Alpide.setreg_IBIAS (
    self,
    IBIAS = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    IBIAS                  [7:0] :   0x40

Origininal docstring
=====

This is an autogenerated function. Do not modify directly.

```

5.4.2.121 setreg_IDB()

```
def pALPIDE.Alpide.setreg_IDB (
    self,
    IDB = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
IDB                                [7:0] : 0x40
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.122 setreg_IRESET()

```
def pALPIDE.Alpide.setreg_IRESET (
    self,
    IRESET = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
IRESET                            [7:0] : 0x32
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.123 setreg_ITHR()

```
def pALPIDE.Alpide.setreg_ITHR (
    self,
    ITHR = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields
=====

ITHR [7:0] : 0x33

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.124 setreg_mode_ctrl()

```
def pALPIDE.Alpide.setreg_mode_ctrl (
    self,
    ChipModeSelector = None,
    EnClustering = None,
    MatrixROSpeed = None,
    IBSerialLinkSpeed = None,
    EnSkewGlobalSignals = None,
    EnSkewStartReadout = None,
    EnReadoutClockGating = None,
    EnReadoutFromCMU = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Mode Control register
Register fields
=====

ChipModeSelector [1:0] : 0x0
EnClustering [2] : 0x1
MatrixROSpeed [3] : 0x1
IBSerialLinkSpeed [5:4] : 0x3
EnSkewGlobalSignals [6] : 0x1
EnSkewStartReadout [7] : 0x1
EnReadoutClockGating [8] : 0x1
EnReadoutFromCMU [9] : 0x0

Original docstring
=====

Mode Control register
Chip Mode Selector values:
0=Cfg Mode, 1=Triggered Mode, 2=Continuous Mode
EnClustering default = 1
MatrixROSpeed 0=10MHz, 1=20MHz (default)
IBSerialLinkSpeed 0=400Mbit, 1=600Mbit, 2-3=1200Mbit (default)
EnSkewGlobalSignals default = 1
EnSkewStartReadout default = 1
EnClockGating default = 1

This is an autogenerated function. Do not modify directly.

5.4.2.125 setreg_pixel_cfg()

```
def pALPIDE.Alpide.setreg_pixel_cfg (
    self,
    PIXCNFG_REGSEL = None,
    PIXCNFG_DATA = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Pixel Configuration Register
Register fields

```
=====
    PIXCNFG_REGSEL          [0] :   0x0
    PIXCNFG_DATA            [1] :   0x0
```

Original docstring

```
=====
Pixel Configuration Register
RegSel 0 = MASK, RegSel 1 = PULSE
```

This is an autogenerated function. Do not modify directly.

5.4.2.126 setreg_VCASN()

```
def pALPIDE.Alpide.setreg_VCASN (
    self,
    VCASN = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
    VCASN                    [7:0] :  0x39
```

Original docstring

```
=====
```

This is an autogenerated function. Do not modify directly.

5.4.2.127 setreg_VCASN2()

```
def pALPIDE.Alpide.setreg_VCASN2 (
    self,
    VCASN2 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VCASN2 [7:0] : 0x40

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.128 setreg_VCASP()

```
def pALPIDE.Alpide.setreg_VCASP (
    self,
    VCASP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VCASP [7:0] : 0x56

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.129 setreg_VCLIP()

```
def pALPIDE.Alpide.setreg_VCLIP (
    self,
    VCLIP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VCLIP [7:0] : 0x0

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.130 setreg_VPULSEH()

```
def pALPIDE.Alpide.setreg_VPULSEH (
    self,
    VPULSEH = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
    VPULSEH                [7:0] : 0xff
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.131 setreg_VPULSEL()

```
def pALPIDE.Alpide.setreg_VPULSEL (
    self,
    VPULSEL = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
    VPULSEL                [7:0] : 0x0
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.4.2.132 setreg_VRESETD()

```
def pALPIDE.Alpide.setreg_VRESETD (
    self,
    VRESETD = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VRESETD [7:0] : 0x0

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.133 setreg_VRESETP()

```
def pALPIDE.Alpide.setreg_VRESETP (
    self,
    VRESETP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VRESETP [7:0] : 0x75

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.134 setreg_VTEMP()

```
def pALPIDE.Alpide.setreg_VTEMP (
    self,
    VTEMP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VTEMP [7:0] : 0x0

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.4.2.135 trigger()

```
def pALPIDE.Alpide.trigger (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Send a trigger to the chip

Reimplemented from [chip.Chip](#).

5.4.2.136 unmask_all_pixels()

```
def pALPIDE.Alpide.unmask_all_pixels (
    self,
    readback = None,
    log = None,
    commitTransaction = None )
```

unmasks all the pixels

5.4.2.137 unmask_col()

```
def pALPIDE.Alpide.unmask_col (
    self,
    col,
    readback = None,
    log = None,
    commitTransaction = None )
```

unmasks the column in the given reg

5.4.2.138 unmask_pixel()

```
def pALPIDE.Alpide.unmask_pixel (
    self,
    pixel_coordinates,
    readback = None,
    log = None,
    commitTransaction = None )
```

unmasks the given pixels

5.4.2.139 unmask_reg()

```
def pALPIDE.Alpide.unmask_reg (
    self,
    reg,
    readback = None,
    log = None,
    commitTransaction = None )
```

unmasks the region in the given reg

5.4.2.140 unmask_row()

```
def pALPIDE.Alpide.unmask_row (
    self,
    row,
    readback = None,
    log = None,
    commitTransaction = None )
```

unmasks the row in the given reg

5.4.2.141 write_read_test()

```
def pALPIDE.Alpide.write_read_test (
    self,
    repeat,
    address = Addr.DTU_TEST_2,
    quiet = False )
```

TEST #.

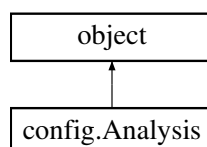
execute multiple times the readback on the DTU_test_register2 [default]

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/pALPIDE.py

5.5 config.Analysis Class Reference

Inheritance diagram for config.Analysis:



Public Member Functions

- `def __init__ (self, basepath=str(pathlib.Path.home()), startblock=0, stopblock=8191, RUv1_version=1, RUv1_progressive=6, is_inverted=False, first_byte_bug=False, exclude_blocks=[], fluence=[])`
- `def set\_basepath (self, basepath)`

Public Attributes

- **basepath**
- **RUv1_version**
- **RUv1_progressive**
- **startblock**
- **stopblock**
- **first_byte_bug**
- **exclude_blocks**
- **is_inverted**
- **fluence**

5.5.1 Detailed Description

Class describing the attributes of the analysis

5.5.2 Member Function Documentation

5.5.2.1 `set_basepath()`

```
def config.Analysis.set_basepath (
    self,
    basepath )
```

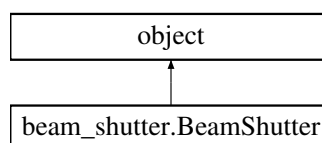
Set path for analysis

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/software/py/flash_analysis/config.py`

5.6 `beam_shutter.BeamShutter` Class Reference

Inheritance diagram for `beam_shutter.BeamShutter`:



Public Member Functions

- def `__init__` (self, port)
- def `beam_off` (self)
- def `beam_on` (self)
- def `beam_shutter_2_off` (self)
- def `beam_shutter_2_on` (self)

Public Attributes

- `logger`
- `port_opened`
- `beam_shutter_serial_port`

5.6.1 Detailed Description

Beam Shutter class

5.6.2 Constructor & Destructor Documentation

5.6.2.1 `__init__()`

```
def beam_shutter.BeamShutter.__init__ (
    self,
    port )
```

Init function

5.6.3 Member Function Documentation

5.6.3.1 `beam_off()`

```
def beam_shutter.BeamShutter.beam_off (
    self )
```

Close the beam shutter

5.6.3.2 beam_on()

```
def beam_shutter.BeamShutter.beam_on (
    self )
```

Open the beam shutter

5.6.3.3 beam_shutter_2_off()

```
def beam_shutter.BeamShutter.beam_shutter_2_off (
    self )
```

Open the beam shutter

5.6.3.4 beam_shutter_2_on()

```
def beam_shutter.BeamShutter.beam_shutter_2_on (
    self )
```

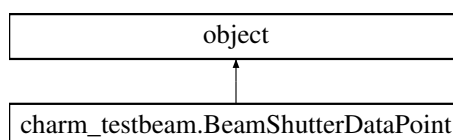
Open the beam shutter

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/beam_shutter.py

5.7 charm_testbeam.BeamShutterDataPoint Class Reference

Inheritance diagram for charm_testbeam.BeamShutterDataPoint:



Public Member Functions

- `def __init__(self)`

Public Attributes

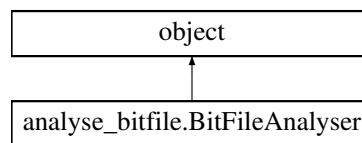
- **timestamp**
- **beam_is_on**

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam.py

5.8 analyse_bitfile.BitFileAnalyser Class Reference

Inheritance diagram for analyse_bitfile.BitFileAnalyser:



Public Member Functions

- def [__init__](#) (self, name=".bit")
- def **set_name** (self, name)
- def **setup_logging** (self)
- def **get_xs_bitfile** (self, xs_type, inverted=False)
- def [analyse_bitfile](#) (self)

Public Attributes

- **path**
- **ones_percent**
- **zeros_percent**
- **logger**
- **name**
- **logdir**

5.8.1 Constructor & Destructor Documentation

5.8.1.1 __init__()

```
def analyse_bitfile.BitFileAnalyser.__init__ (
    self,
    name = ".bit" )
```

Init function

5.8.2 Member Function Documentation

5.8.2.1 analyse_bitfile()

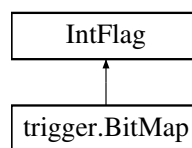
```
def analyse_bitfile.BitFileAnalyser.analyse_bitfile (
    self )
```

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/flash_analysis/analyse_bitfile.py

5.9 trigger.BitMap Class Reference

Inheritance diagram for trigger.BitMap:



Static Public Attributes

- int **ORBIT** = 0
- int **HBa** = 1
- int **HBr** = 2
- int **HBc** = 3
- int **PHYSICS** = 4
- int **PP** = 5
- int **CAL** = 6
- int **SOT** = 7
- int **EOT** = 8
- int **SOC** = 9
- int **EOC** = 10
- int **TF** = 11

5.9.1 Detailed Description

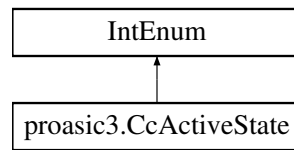
Trigger bit mapping as from gbtX_pkg.vhd

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/trigger.py

5.10 proasic3.CcActiveState Class Reference

Inheritance diagram for proasic3.CcActiveState:



Static Public Attributes

- int **IDLE** = 0x0
- int **INIT_SM_WAIT** = 0x1
- int **INIT_FLASH_READ** = 0x2
- int **FIFO_STATUS** = 0x3
- int **TRANSFER_DATA** = 0x4
- int **TRANSFER_DATA_DONE** = 0x5
- int **STARTUP_WAIT** = 0x6

5.10.1 Detailed Description

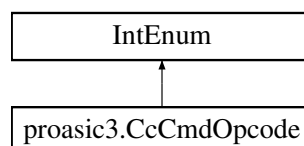
Config controller FSM state

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.11 proasic3.CcCmdOpcode Class Reference

Inheritance diagram for proasic3.CcCmdOpcode:



Static Public Attributes

- int **INIT_CONFIG** = 0x01
- int **CONT_SCRUB** = 0x02
- int **SINGLE_SCRUB** = 0x04
- int **STOP** = 0x08
- int **SCRUB_CNT_CLR** = 0x10

5.11.1 Detailed Description

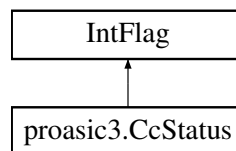
Config controller opcode mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.12 proasic3.CcStatus Class Reference

Inheritance diagram for proasic3.CcStatus:



Static Public Attributes

- int **IDLE** = 0
- int **INIT_CONFIG_DONE** = 1
- int **SCRUBBING_ONGOING** = 2
- int **SCRUBBING_DONE** = 3

5.12.1 Detailed Description

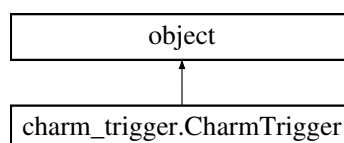
CC status

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.13 charm_trigger.CharmTrigger Class Reference

Inheritance diagram for charm_trigger.CharmTrigger:



Public Member Functions

- def **__init__** (self, hostname=UDP_IP, port=UDP_PORT)
- def **initialize_trigger** (self)
- def **get_last_spill** (self)
- def **check_spill** (self)

Public Attributes

- **hostname**
- **port**
- **sock**
- **last_spill**

5.13.1 Member Function Documentation

5.13.1.1 get_last_spill()

```
def charm_trigger.CharmTrigger.get_last_spill (  
    self )
```

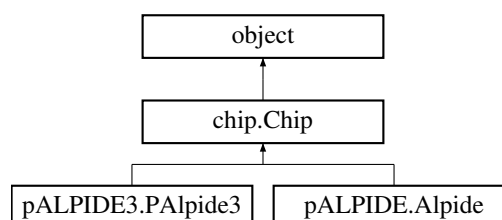
Returns data:
{Y|N}:<last_spill_timestamp>

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/charm_trigger.py

5.14 chip.Chip Class Reference

Inheritance diagram for chip.Chip:



Public Member Functions

- def `__init__` (self, board, chipid, modulename, readback=False, log=False, commitTransaction=True)
- def `write_reg` (self, address, data, commitTransaction=None, readback=None, log=None, verbose=False)
- def `write_region_reg` (self, rgn_add, base_add, sub_add, data, readback=None, log=None, commitTransaction=None, verbose=False)
- def `write_opcode` (self, opcode, commitTransaction=None, log=None, verbose=False)
- def `read_reg` (self, address, commitTransaction=True, log=None, verbose=False)
- def `read_region_reg` (self, rgn_add, base_add, sub_add, log=None, commitTransaction=None, verbose=False)
- def `is_outer_barrel_master` (self)
HLF common to pALPIDE and pALPIDE3 #.
- def `is_outer_barrel_slave` (self)
- def `is_inner_barrel` (self)
- def `write_read_test` (self, repeat, quiet=False)
- def `reset` (self)
- def `initialize` (self)
- def `configure_dtu` (self)
- def `initialize_readout` (self, PLLDAC=None, DriverDAC=None, PreDAC=None, PLLDelayStages=None, PLLBandwidthControl=None, commitTransaction=None, log=None, readback=None)
- def `reset_pll` (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def `trigger` (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def `propagate_comma` (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def `propagate_data` (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def `propagate_prbs` (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def `propagate_clock` (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def `set_xoff` (self, XOff=1, commitTransaction=None, readback=None, log=None, verbose=False)
- def `dump_config` (self)

Public Attributes

- `chipid`
- `readback`
- `log`
- `commitTransaction`
- `board`
- `modulename`
- `logger`

5.14.1 Detailed Description

Basic Chip class describing a generic ALPIDE/pALPDIDE chip.

This abstract class provides the basic W/R blocks used to write and read data between chip and host.

This is an abstract class, and is NOT intended to be used directly. Instead, use the actual implementations ALPIDE/pALPIDE instead

5.14.2 Member Function Documentation

5.14.2.1 `configure_dtu()`

```
def chip.Chip.configure_dtu (
    self )
```

Configure DTU

5.14.2.2 `dump_config()`

```
def chip.Chip.dump_config (
    self )
```

Return a configuration string of all registers on the chip

Reimplemented in [pALPIDE.Alpide](#), and [pALPIDE3.PAlpide3](#).

5.14.2.3 `initialize()`

```
def chip.Chip.initialize (
    self )
```

Perform chip initialization procedure

5.14.2.4 `initialize_readout()`

```
def chip.Chip.initialize_readout (
    self,
    PLLDAC = None,
    DriverDAC = None,
    PreDAC = None,
    PLLDelayStages = None,
    PLLBandwidthControl = None,
    commitTransaction = None,
    log = None,
    readback = None )
```

Perform initialization sequenca for readout

Reimplemented in [pALPIDE3.PAlpide3](#).

5.14.2.5 is_inner_barrel()

```
def chip.Chip.is_inner_barrel (
    self )
```

Check if chip is IB

5.14.2.6 is_outer_barrel_master()

```
def chip.Chip.is_outer_barrel_master (
    self )
```

HLF common to pALPIDE and pALPIDE3 #.

Check if chip is OB master

5.14.2.7 is_outer_barrel_slave()

```
def chip.Chip.is_outer_barrel_slave (
    self )
```

Check if chip is OB slave

5.14.2.8 propagate_clock()

```
def chip.Chip.propagate_clock (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate 600 MHz clock from chip

Reimplemented in [pALPIDE.Alpide](#), and [pALPIDE3.PAlpide3](#).

5.14.2.9 propagate_comma()

```
def chip.Chip.propagate_comma (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate COMMA pattern from chip (for realignment)

Reimplemented in [pALPIDE.Alpide](#), and [pALPIDE3.PAlpide3](#).

5.14.2.10 propagate_data()

```
def chip.Chip.propagate_data (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate Data from chip

Reimplemented in [pALPIDE.Alpide](#), and [pALPIDE3.PAlpide3](#).

5.14.2.11 propagate_prbs()

```
def chip.Chip.propagate_prbs (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate PRBS test pattern from chip

Reimplemented in [pALPIDE3.PAlpide3](#).

5.14.2.12 read_reg()

```
def chip.Chip.read_reg (
    self,
    address,
    commitTransaction = True,
    log = None,
    verbose = False )
```

basic read_reg function invoking the corresponding board function

5.14.2.13 read_region_reg()

```
def chip.Chip.read_region_reg (
    self,
    rgn_add,
    base_add,
    sub_add,
    log = None,
    commitTransaction = None,
    verbose = False )
```

Basic function to write to a chip register using the extended address

5.14.2.14 reset()

```
def chip.Chip.reset (
    self )
```

Reset the chip

Reimplemented in [pALPIDE.Alpide](#), and [pALPIDE3.PAlpide3](#).

5.14.2.15 reset_pll()

```
def chip.Chip.reset_pll (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Reset the pll

Reimplemented in [pALPIDE3.PAlpide3](#).

5.14.2.16 set_xoff()

```
def chip.Chip.set_xoff (
    self,
    XOff = 1,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Send XOFF to the chip. 1 -> Xoff, 0 -> Xon

Reimplemented in [pALPIDE.Alpide](#), and [pALPIDE3.PAlpide3](#).

5.14.2.17 trigger()

```
def chip.Chip.trigger (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Send a trigger to the chip

Reimplemented in [pALPIDE.Alpide](#), and [pALPIDE3.PAlpide3](#).

5.14.2.18 write_opcode()

```
def chip.Chip.write_opcode (
    self,
    opcode,
    commitTransaction = None,
    log = None,
    verbose = False )
```

basic function to send an opcode to the chip by calling the corresponding function on the board

5.14.2.19 write_read_test()

```
def chip.Chip.write_read_test (
    self,
    repeat,
    quiet = False )
```

abstract function (to be implemented on two different addresses in pALPIDE3 and pALPIDE) to execute multiple t

5.14.2.20 write_reg()

```
def chip.Chip.write_reg (
    self,
    address,
    data,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Basic write function which makes use of a single address

5.14.2.21 write_region_reg()

```
def chip.Chip.write_region_reg (
    self,
    rgn_add,
    base_add,
    sub_add,
    data,
    readback = None,
    log = None,
    commitTransaction = None,
    verbose = False )
```

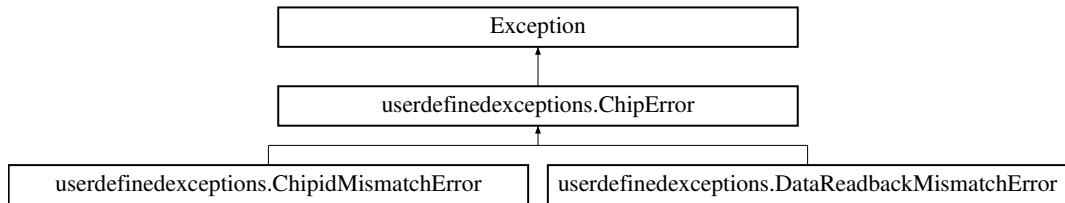
Basic function to write to a chip register using the extended address

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/chip.py

5.15 userdefinedexceptions.ChipError Class Reference

Inheritance diagram for userdefinedexceptions.ChipError:



Public Member Functions

- `def __init__(self)`

5.15.1 Detailed Description

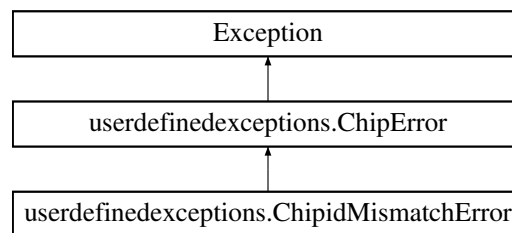
defines a basic class for ALPIDE exceptions

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/userdefinedexceptions.py`

5.16 userdefinedexceptions.ChipidMismatchError Class Reference

Inheritance diagram for userdefinedexceptions.ChipidMismatchError:



Public Member Functions

- `def __init__(self, value="", requested_chipid=-1, rd_chipid=-1)`
- `def __str__(self)`
- `def __requested_chipid__(self)`
- `def __rd_chipid__(self)`

Public Attributes

- `value`
- `requested_chipid`
- `rd_chipid`

5.16.1 Detailed Description

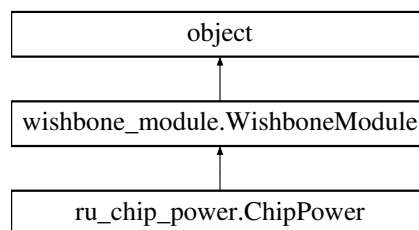
basic class to handle the chipid mismatch exception when reading back from the chip

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/userdefinedexceptions.py

5.17 ru_chip_power.ChipPower Class Reference

Inheritance diagram for ru_chip_power.ChipPower:



Public Member Functions

- def `__init__` (self, comm, moduleId, ru_nr=0)
- def `initialize` (self)
- def `enable_voltages` (self, AVDD=True, DVDD=True, commitTransaction=True)
- def `set_voltages` (self, AVDD=None, DVDD=None, commitTransaction=True)
- def `init_ADC` (self)
- def `log_currents` (self)
- def `read_values` (self)
- def `read_values_raw` (self)
- def `readreg_powerbank_switch` (self)
- def `dump_config` (self)

Public Attributes

- `parameters`

5.17.1 Detailed Description

Class to handle chip power settings

5.17.2 Member Function Documentation

5.17.2.1 dump_config()

```
def ru_chip_power.ChipPower.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.17.2.2 enable_voltages()

```
def ru_chip_power.ChipPower.enable_voltages (
    self,
    AVDD = True,
    DVDD = True,
    commitTransaction = True )
```

Enable AVDD and / or DVDD.
True: Enable,
False: Disalbe,
None: keep

5.17.2.3 init_ADC()

```
def ru_chip_power.ChipPower.init_ADC (
    self )
```

Initialize the ADC for readout

5.17.2.4 initialize()

```
def ru_chip_power.ChipPower.initialize (
    self )
```

Initialize the module

5.17.2.5 log_currents()

```
def ru_chip_power.ChipPower.log_currents (
    self )
```

Logs ADC values (AVDD, DVDD currents and voltages)

5.17.2.6 read_values()

```
def ru_chip_power.ChipPower.read_values (
    self )
```

Read values for AVDD,DVDD (voltages, currents)

5.17.2.7 read_values_raw()

```
def ru_chip_power.ChipPower.read_values_raw (
    self )
```

Read raw ADC values for AVDD,DVDD (voltages, currents)

5.17.2.8 readreg_powerbank_switch()

```
def ru_chip_power.ChipPower.readreg_powerbank_switch (
    self )
```

Read the Powerbank switch register

5.17.2.9 set_voltages()

```
def ru_chip_power.ChipPower.set_voltages (
    self,
    AVDD = None,
    DVDD = None,
    commitTransaction = True )
```

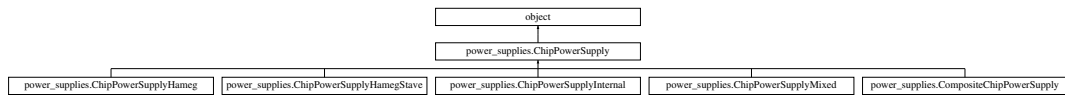
Set AVDD and DVDD voltages (if not none)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_chip_power.py

5.18 power_supplies.ChipPowerSupply Class Reference

Inheritance diagram for power_supplies.ChipPowerSupply:



Public Member Functions

- def **power_on** (self)
- def **power_off** (self)
- def **set_voltages** (self, avdd=1.8, dvdd=1.8, pvdd=None, backbias=None)
- def **get_values** (self)

5.18.1 Detailed Description

Implementation of the chip powering scheme

5.18.2 Member Function Documentation

5.18.2.1 power_on()

```
def power_supplies.ChipPowerSupply.power_on (
    self )
```

power on

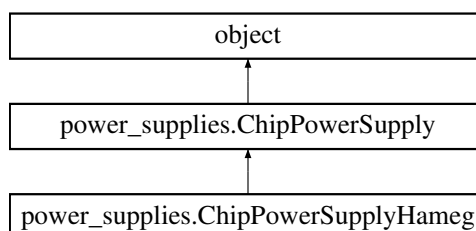
Reimplemented in [power_supplies.ChipPowerSupplyMixed](#), [power_supplies.ChipPowerSupplyHamegStave](#), [power_supplies.CompositeChipPowerSupply](#), [power_supplies.ChipPowerSupplyHameg](#), and [power_supplies.ChipPowerSupplyInternal](#)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_supplies.py

5.19 power_supplies.ChipPowerSupplyHameg Class Reference

Inheritance diagram for power_supplies.ChipPowerSupplyHameg:



Public Member Functions

- `def __init__ (self, hameg_supply)`
- `def power\_on (self)`
- `def power_off (self)`
- `def set_voltages (self, avdd=1.8, dvdd=1.8, pvdd=None, backbias=None)`
- `def get_values (self)`

Public Attributes

- `hameg_supply`
- `pvdd_used`

5.19.1 Member Function Documentation

5.19.1.1 `power_on()`

```
def power_supplies.ChipPowerSupplyHameg.power_on (
    self )
```

power on

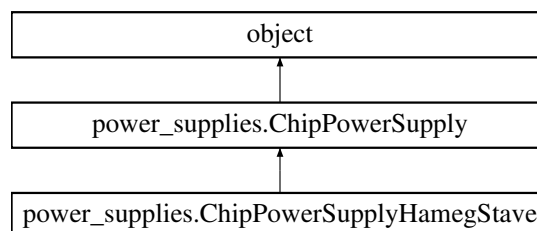
Reimplemented from [power_supplies.ChipPowerSupply](#).

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_supplies.py`

5.20 `power_supplies.ChipPowerSupplyHamegStave` Class Reference

Inheritance diagram for `power_supplies.ChipPowerSupplyHamegStave`:



Public Member Functions

- `def __init__ (self, hameg_supply)`
- `def power\_on (self)`
- `def power_off (self)`
- `def set_voltages (self, avdd=None, dvdd=None, pvdd=None, backbias=None)`
- `def get_values (self)`

Public Attributes

- **hameg_supply**
- **pvdd_used**
- **logger**

5.20.1 Member Function Documentation

5.20.1.1 power_on()

```
def power_supplies.ChipPowerSupplyHamegStave.power_on (
    self )
```

power on

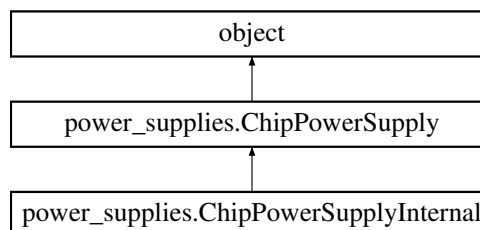
Reimplemented from [power_supplies.ChipPowerSupply](#).

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_supplies.py

5.21 power_supplies.ChipPowerSupplyInternal Class Reference

Inheritance diagram for power_supplies.ChipPowerSupplyInternal:



Public Member Functions

- def **__init__** (self, chippower)
- def [power_on](#) (self)
- def **power_off** (self)
- def **set_voltages** (self, avdd=1.8, dvdd=1.8, pvdd=None, backbias=None)
- def **get_values** (self)

Public Attributes

- **chippower**

5.21.1 Member Function Documentation

5.21.1.1 power_on()

```
def power_supplies.ChipPowerSupplyInternal.power_on (
    self )
```

power on

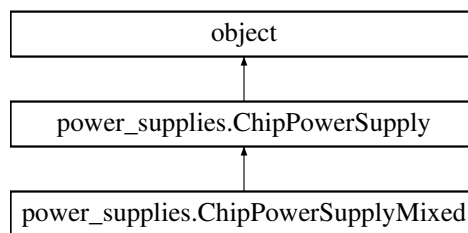
Reimplemented from [power_supplies.ChipPowerSupply](#).

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_supplies.py

5.22 power_supplies.ChipPowerSupplyMixed Class Reference

Inheritance diagram for power_supplies.ChipPowerSupplyMixed:



Public Member Functions

- def **__init__** (self, chippower, hameg_supply, max_bb_current=2)
- def [power_on](#) (self)
- def **power_off** (self)
- def **set_voltages** (self, avdd=1.8, dvdd=1.8, pvdd=None, backbias=None)
- def **get_values** (self)

Public Attributes

- **hameg_supply**
- **chippower**
- **backbias_used**
- **max_bb_current**
- **bb_voltage**
- **bb_current_limit**
- **logger**

5.22.1 Detailed Description

Mixed Powersupply with Ru internal for AVDD and DVDD and HAMEG for Backbias

5.22.2 Member Function Documentation

5.22.2.1 power_on()

```
def power_supplies.ChipPowerSupplyMixed.power_on (
    self )
```

power on

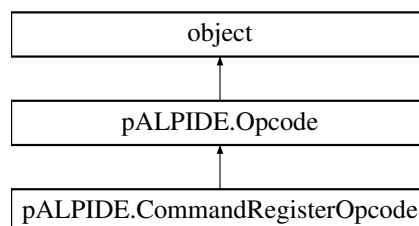
Reimplemented from [power_supplies.ChipPowerSupply](#).

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_supplies.py

5.23 pALPIDE.CommandRegisterOpcode Class Reference

Inheritance diagram for pALPIDE.CommandRegisterOpcode:



Static Public Attributes

- int **CMUCLRERR** = 0xFF00
- int **FIFOTEST** = 0xFF01
- int **LOADOBDEFCFG** = 0xFF02
- int **XOFF** = 0xFF10
- int **XON** = 0xFF11
- int **ADCMEASURE** = 0xFF20

5.23.1 Detailed Description

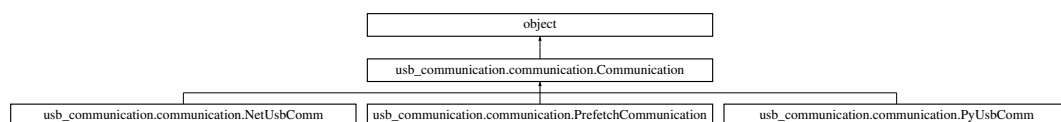
Class extending the opcodes the command register available commands

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/pALPIDE.py

5.24 usb_communication.communication.Communication Class Reference

Inheritance diagram for usb_communication.communication.Communication:



Public Member Functions

- def `__init__` (self, `enable_r derr_exception=False`)
- def `enable_log` (self)
- def `disable_log` (self)
- def `enable_r derr_exception` (self)
- def `disable_r derr_exception` (self)
- def `stop` (self)
- def `single_write` (self, module, address, data)
- def `single_read` (self, module, address, log=None)
- def `write_reg` (self, module, address, data)
- def `read_reg` (self, module, address)
- def `register_write` (self, module, address, data)
- def `register_read` (self, module, address)
- def `flush` (self)
- def `read_results` (self, log=None)
- def `diagnose_read_results` (self, log=None)
- def `discardall_dp1` (self, maxReads=10)
- def `discardall_dp2` (self, maxReads=10)
- def `discardall_dp3` (self, maxReads=10)
- def `read_dp2` (self, size)
- def `read_dp3` (self, size)

Public Attributes

- `logger`
- `log`
- `max_retries`

5.24.1 Detailed Description

Basic Communication class between Host and Device.

This abstract class provides the basic communication blocks used to write and read data between chip and host.

Access to Wishbone is possible via `single_write()` or `single_read()`. To send a batch of commands, the functions `register_write()`, `register_read()`, `flush()` and `read_results()` are used. In both cases, results are written as a 2-tuple in the form `(addr,data)`.

Access to data ports is given via `read_DP2()` and `read_DP3()`. The result is given as array of 4 byte ints.

This is an abstract class, and is NOT intended to be used directly. Instead, use the actual implementations `PyUsbComm`, `NetUsbComm`, `UsbCommSim` or `Wb2GbtXComm`.

5.24.2 Member Function Documentation

5.24.2.1 `diagnose_read_results()`

```
def usb_communication.communication.Communication.diagnose_read_results (
    self,
    log = None )
```

Read results for debugging and diagnose.

This function reads results from a previously flushed communication. In comparison to `read_results`, this function does not check the results for read errors, but returns `rderrorflag` as part of the tuple. The return value is `[(rderr0,addr0,data0),...,(rderrN,addrN,dataN)]`

5.24.2.2 `disable_log()`

```
def usb_communication.communication.Communication.disable_log (
    self )
```

disables the exception throw when receiving a wishbone read error flag

5.24.2.3 `disable_rderr_exception()`

```
def usb_communication.communication.Communication.disable_rderr_exception (
    self )
```

disables the exception throw when receiving a wishbone read error flag

5.24.2.4 enable_log()

```
def usb_communication.communication.Communication.enable_log (  
    self )
```

enables the exception throw when receiving a wishbone read error flag

5.24.2.5 enable_r derr_exception()

```
def usb_communication.communication.Communication.enable_r derr_exception (  
    self )
```

enables the exception throw when receiving a wishbone read error flag

5.24.2.6 flush()

```
def usb_communication.communication.Communication.flush (  
    self )
```

Flush the buffer.

This function sends all previously registered commands to the device.

5.24.2.7 read_dp2()

```
def usb_communication.communication.Communication.read_dp2 (  
    self,  
    size )
```

Reads <size> bytes of data from DP2.
Returns a list of integers.

5.24.2.8 read_dp3()

```
def usb_communication.communication.Communication.read_dp3 (  
    self,  
    size )
```

Reads <size> bytes of data from DP3.
Returns a list of integers.

5.24.2.9 read_reg()

```
def usb_communication.communication.Communication.read_reg (
    self,
    module,
    address )
```

DEPRECATED! used for Probecard while waiting to modify local implementation

5.24.2.10 read_results()

```
def usb_communication.communication.Communication.read_results (
    self,
    log = None )
```

Read results from communication.

This function reads the results from a previously flushed communication. The function expects 1 result per read command registered.

Returns a List of data tuples in the form
[(addr0,data0), (addr1,data1), ..., (addrN,dataN)]

5.24.2.11 register_read()

```
def usb_communication.communication.Communication.register_read (
    self,
    module,
    address )
```

Register a read register command in the buffer

5.24.2.12 register_write()

```
def usb_communication.communication.Communication.register_write (
    self,
    module,
    address,
    data )
```

Register a write register command in the buffer

5.24.2.13 single_read()

```
def usb_communication.communication.Communication.single_read (
    self,
    module,
    address,
    log = None )
```

A Single read operation.

Perform a single read operation to a given module and address. This sends the command directly to the device and reads back the result. Previously registered commands are not sent.

5.24.2.14 single_write()

```
def usb_communication.communication.Communication.single_write (
    self,
    module,
    address,
    data )
```

A Single write operation.

Perform a single write operation to a given module and address. This sends the command directly to the device. Previously registered commands are not sent.

5.24.2.15 stop()

```
def usb_communication.communication.Communication.stop (
    self )
```

Perform any operation on close

Reimplemented in [usb_communication.communication.NetUsbComm](#), and [usb_communication.communication.PrefetchCommunication](#)

5.24.2.16 write_reg()

```
def usb_communication.communication.Communication.write_reg (
    self,
    module,
    address,
    data )
```

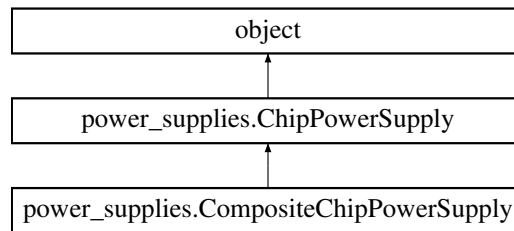
DEPRECATED! used for Probecard while waiting to modify local implementation

The documentation for this class was generated from the following file:

- [/media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py](#)

5.25 power_supplies.CompositeChipPowerSupply Class Reference

Inheritance diagram for power_supplies.CompositeChipPowerSupply:



Public Member Functions

- def **__init__** (self, supplies)
- def **power_on** (self)
- def **power_off** (self)
- def **set_voltages** (self, avdd=1.8, dvdd=1.8, pvdd=None, backbias=None)
- def **get_values** (self)

Public Attributes

- **power_supplies**

5.25.1 Member Function Documentation

5.25.1.1 power_on()

```
def power_supplies.CompositeChipPowerSupply.power_on (  
    self )
```

power on

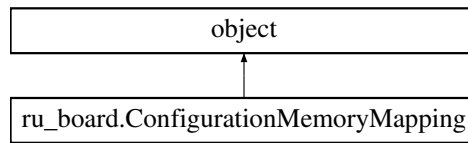
Reimplemented from [power_supplies.ChipPowerSupply](#).

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_supplies.py

5.26 ru_board.ConfigurationMemoryMapping Class Reference

Inheritance diagram for ru_board.ConfigurationMemoryMapping:



Static Public Attributes

- int **alpine_vdd_on** = 0

5.26.1 Detailed Description

configuration register mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_board.py

5.27 ru_chip_power.ConversionParameter Class Reference

Public Member Functions

- def **__init__** (self, k=1, d=0)
- def [step_to_value](#) (self, step)
- def [value_to_step](#) (self, voltage)

Public Attributes

- **k**
- **d**

5.27.1 Detailed Description

Simple class to store conversion parameters and convert values

5.27.2 Member Function Documentation

5.27.2.1 step_to_value()

```
def ru_chip_power.ConversionParameter.step_to_value (
    self,
    step )
```

Convert a DAC/ADC step to a value

5.27.2.2 value_to_step()

```
def ru_chip_power.ConversionParameter.value_to_step (
    self,
    voltage )
```

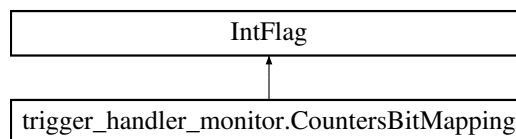
Convert a value to a DAC/ADC step

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_chip_power.py

5.28 trigger_handler_monitor.CountersBitMapping Class Reference

Inheritance diagram for trigger_handler_monitor.CountersBitMapping:



Static Public Attributes

- int **TRIGGER_SENT** = 0
- int **TRIGGER_NOT_SENT** = 1
- int **TRG_FIFO_GTH_FULL** = 2
- int **TRG_FIFO_GTH_OVFL** = 3
- int **TRG_FIFO_GPIO_FULL** = 4
- int **TRG_FIFO_GPIO_OVFL** = 5
- int **RX_FIFO_DATA_AVAILABLE** = 6
- int **TRIGGER_GATED** = 7

5.28.1 Detailed Description

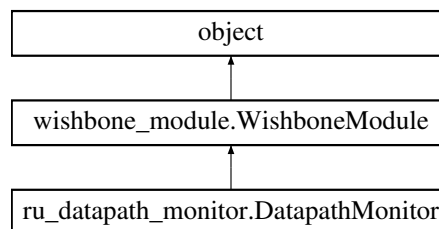
bit mapping of the reset/latch counter registers

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/trigger_handler_monitor.py

5.29 ru_datapath_monitor.DatapathMonitor Class Reference

Inheritance diagram for ru_datapath_monitor.DatapathMonitor:



Public Member Functions

- def **__init__** (self, moduleid, board_obj, lanes, offset=0)
- def **set_lanes** (self, lanes)
- def **get_lanes** (self)
- def **get_default_lanes** (self)
- def **reset_counters** (self, commitTransaction=True)
- def **latch_counters** (self, commitTransaction=False)
- def **get_lane_idx** (self, lane)
- def **reset_lane_counters** (self, lane, commitTransaction=True)
- def **read_counters** (self, lanes=None, counters=None)
- def **read_counter** (self, lanes, counter)
- def **read_all_counters** (self)

Public Attributes

- **REG_LATCH**
- **REG_RESET**
- **COUNTER_OFFSET**
- **lanes**
- **default_lanes**
- **offset**
- **nr_regs**
- **counter_mapping**
- **counter_order**
- **counter_transform**

Static Public Attributes

- list **wb_register_mapping**
- **nr_counter_regs** = len(wb_register_mapping)

5.29.1 Member Function Documentation

5.29.1.1 latch_counters()

```
def ru_datapath_monitor.DatapathMonitor.latch_counters (
    self,
    commitTransaction = False )
```

Latches values into counters

5.29.1.2 read_all_counters()

```
def ru_datapath_monitor.DatapathMonitor.read_all_counters (
    self )
```

Read all counters of monitor

5.29.1.3 read_counters()

```
def ru_datapath_monitor.DatapathMonitor.read_counters (
    self,
    lanes = None,
    counters = None )
```

Read Counters(array) from lanes(array)

5.29.1.4 reset_counters()

```
def ru_datapath_monitor.DatapathMonitor.reset_counters (
    self,
    commitTransaction = True )
```

Reset all counters

5.29.1.5 reset_lane_counters()

```
def ru_datapath_monitor.DatapathMonitor.reset_lane_counters (
    self,
    lane,
    commitTransaction = True )
```

Reset all counters for lane i

5.29.2 Member Data Documentation

5.29.2.1 wb_register_mapping

```
list ru_datapath_monitor.DatapathMonitor.wb_register_mapping [static]
```

Initial value:

```
= [
    "EVENT_COUNT_LOW",
    "EVENT_COUNT_HIGH",
    "DECODE_ERROR_COUNT",
    "EVENT_ERROR_COUNT",
    "BUSY_COUNT",
    "DOUBLE_BUSY_ON_COUNT",
    "IDLE_WORD_COUNT",
    "CPLL_LOCK_LOSS_COUNT",
    "CDR_LOCK_LOSS_COUNT",
    "ALIGNED_LOSS_COUNT",
    "ELASTIC_BUF_OVERFLOW_COUNT",
    "LANE_FIFO_FULL_COUNT",
    "LANE_PACKAGER_LANE_TIMEOUT_COUNT",
    "GBT_PACKER_LANE_STOPS_COUNT",
    "LANE_PACKAGER_STOP_COUNT",
    "LANE_PACKAGER_START_COUNT",
    "GBT_PACKER_LANE_START_VIOLATION_COUNT"
]
```

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_datapath_monitor.py

5.30 ru_datapath_monitor.DatapathMonitorDual Class Reference

Public Member Functions

- def **__init__** (self, datapath_mon1, datapath_mon2)
- def **set_lanes** (self, lanes)
- def **get_lanes** (self)
- def **reset_counters** (self, commitTransaction=True)
- def **latch_counters** (self, commitTransaction=False)
- def **reset_lane_counters** (self, lane, commitTransaction=True)
- def **read_counters** (self, lanes=None, counters=None)
- def **read_counter** (self, lanes=None, counter='EVENT_COUNT')
- def **read_all_counters** (self)

Public Attributes

- **datapath_mon1**
- **datapath_mon2**
- **lanes**
- **counter_mapping**

5.30.1 Detailed Description

Monitor for Datapath status counters

5.30.2 Member Function Documentation

5.30.2.1 latch_counters()

```
def ru_datapath_monitor.DatapathMonitorDual.latch_counters (
    self,
    commitTransaction = False )
```

Latches values into counters

5.30.2.2 read_all_counters()

```
def ru_datapath_monitor.DatapathMonitorDual.read_all_counters (
    self )
```

Read all counters of monitor

5.30.2.3 read_counters()

```
def ru_datapath_monitor.DatapathMonitorDual.read_counters (
    self,
    lanes = None,
    counters = None )
```

Read Counters(array) from lanes(array)

5.30.2.4 reset_counters()

```
def ru_datapath_monitor.DatapathMonitorDual.reset_counters (
    self,
    commitTransaction = True )
```

Reset all counters

5.30.2.5 reset_lane_counters()

```
def ru_datapath_monitor.DatapathMonitorDual.reset_lane_counters (
    self,
    lane,
    commitTransaction = True )
```

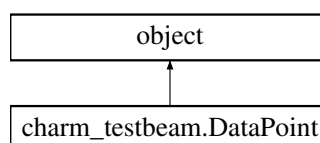
Reset all counters for lane i

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_datapath_monitor.py

5.31 charm_testbeam.DataPoint Class Reference

Inheritance diagram for charm_testbeam.DataPoint:



Public Member Functions

- `def __init__(self)`

Public Attributes

- timestamp
- hameg_values
- cru_counters
- cru_adcs
- cru_gpio
- pa3_values
- powerunit_values
- chipdata
- dctrl_counters
- trigger_handler_mon
- gth_aligned
- gth_status
- lane_counters
- radmon_dctrl
- radmon_gbt01
- radmon_gbt2
- radmon_wishbone_master
- radmon_dp_fifos
- radmon_sysmon
- radmon_trigger_handler
- radmon_datapath
- radmon_mismatch_dctrl
- radmon_mismatch_gbt01
- radmon_mismatch_gbt2
- radmon_mismatch_wishbone_master
- radmon_mismatch_sysmon
- radmon_mismatch_trigger_handler
- wsmstr_rderr
- wsmstr_wrerr
- sysmon_vccint
- sysmon_vccaux
- sysmon_vccbram
- sysmon_vcc_alpide
- sysmon_vcc_sca
- sysmon_temp
- sysmon_tmr_status
- mmcm_monitor_counters
- mmcm_monitor_lock_counters
- gbt_flow_monitor_counters
- spi_micro_test_counters
- lane_counters_gpio
- gbt_packer_monitor_gth
- gbt_packer_monitor_gpio
- gbt_packer_config_gth
- gbt_packer_config_gpio
- gth_config
- gpio_config

5.31.1 Detailed Description

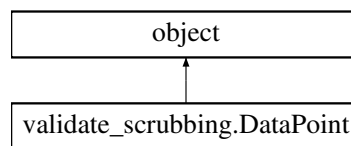
Datapoint for collecting continuous readout data

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam.py

5.32 validate_scrubbing.DataPoint Class Reference

Inheritance diagram for validate_scrubbing.DataPoint:



Public Member Functions

- `def __init__(self)`

Public Attributes

- `timestamp`
- `hameg_values`
- `cru_counters`
- `cru_adcs`
- `cru_gpio`
- `pa3_values`
- `powerboard_values`
- `chipdata`
- `dctrl_counters`
- `gth_aligned`
- `gth_status`
- `lane_counters`
- `radmon_dctrl`
- `radmon_gbt01`
- `radmon_gbt2`
- `radmon_wishbone_master`
- `radmon_dp_fifos`
- `radmon_sysmon`
- `radmon_mismatch_dctrl`
- `radmon_mismatch_gbt01`
- `radmon_mismatch_gbt2`
- `radmon_mismatch_wishbone_master`
- `radmon_mismatch_sysmon`
- `wsmstr_rderr`
- `wsmstr_wrerr`

- `sysmon_vccint`
- `sysmon_vccaux`
- `sysmon_vccbram`
- `sysmon_vcc_alpide`
- `sysmon_vcc_sca`
- `sysmon_temp`
- `sysmon_tmr_status`
- `mmcm_monitor_counters`
- `mmcm_monitor_lock_counters`
- `gbtx_flow_monitor_counters`
- `spi_micro_test_counters`

5.32.1 Detailed Description

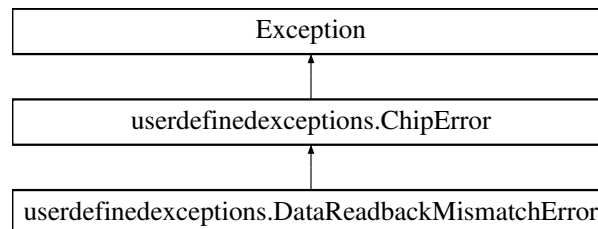
Datapoint for collecting continuous readout data

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/software/py/validate_scrubbing.py`

5.33 userdefinedexceptions.DataReadbackMismatchError Class Reference

Inheritance diagram for `userdefinedexceptions.DataReadbackMismatchError`:



Public Member Functions

- `def __init__(self, value="", written_data="", rd_data="")`
- `def __str__(self)`
- `def __written_data__(self)`
- `def __rd_data__(self)`

Public Attributes

- `value`
- `written_data`
- `rd_data`

5.33.1 Detailed Description

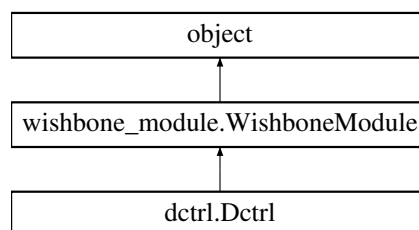
basic class to handle the data readback mismatch exception when writing to the chip

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/userdefinedexceptions.py

5.34 dctrl.Dctrl Class Reference

Inheritance diagram for dctrl.Dctrl:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def **force_phase** (self, phase, commitTransaction=True)
- def **release_phase_force** (self, commitTransaction=True)
- def **get_phase_force** (self, commitTransaction=True)
- def **get_phase** (self, commitTransaction=True)
- def **set_auto_phase_offset** (self, offset, commitTransaction=True)
- def **get_auto_phase_offset** (self, commitTransaction=True)
- def **write_chip_reg** (self, address, data, chipid, commitTransaction=True, readback=False)
- def **read_chip_reg** (self, address, chipid, commitTransaction=True)
- def **write_chip_opcode** (self, opcode, commitTransaction=True)
- def **on_poweron_chip** (self, avdd, dvdd, pvdd, backbias)
- def **on_poweroff_chip** (self)
- def **busy_force** (self, force_value)
- def **busy_force_release** (self)
- def **rst_ctrl_cntrs** (self, commitTransaction=True)
- def **reset_counters** (self, commitTransaction=True)
- def **get_counters** (self, reset_after=False, commitTransaction=True)
- def **get_busy_transceiver_mask** (self)
- def **set_busy_transceiver_mask** (self, mask)
- def **get_busy_transceiver_status** (self)
- def **set_input** (self, dctrl_input=0, force=False, commitTransaction=True)
- def **get_input** (self)
- def **set_dctrl_mask** (self, mask)
- def **get_dctrl_mask** (self)
- def **set_dclk_mask** (self, mask)
- def **get_dclk_mask** (self)
- def **enable_manchester_tx** (self, commitTransaction=True)
- def **disable_manchester_tx** (self, commitTransaction=True)

- def [get_manchester_tx](#) (self)
- def [set_idelays](#) (self, idelays=None, idelay_all=None, commitTransaction=True)
- def [get_idelays](#) (self)
- def [get_mismatch](#) (self)
- def [set_dclk_parallel](#) (self, phase, index, commitTransaction=True)
- def [get_dclk_parallel](#) (self, index, commitTransaction=True)
- def [set_wait_cycles](#) (self, value, commitTransaction=True)
- def [get_wait_cycles](#) (self, commitTransaction=True)
- def [get_manchester_rx_detected](#) (self, commitTransaction=True)
- def [dump_config](#) (self)
- def [readreg_write_phase](#) (self)

Public Attributes

- **verbose**
- **connector_used**
- **phase_dict**
- **phase_dict_inv**

5.34.1 Detailed Description

dctrl/cltr wishbone slave

5.34.2 Member Function Documentation

5.34.2.1 busy_force()

```
def dctrl.Dctrl.busy_force (  
    self,  
    force_value )
```

forces the busy logic to the given value

Note that the force is not applied to the chip and this force is only relative to the logic steering the trigger generation inside the ctrl/dctrl block

5.34.2.2 busy_force_release()

```
def dctrl.Dctrl.busy_force_release (  
    self )
```

releases the force on busy logic

5.34.2.3 disable_manchester_tx()

```
def dctrl.Dctrl.disable_manchester_tx (
    self,
    commitTransaction = True )
```

Disables the manchester encoding in transmission

5.34.2.4 dump_config()

```
def dctrl.Dctrl.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.34.2.5 enable_manchester_tx()

```
def dctrl.Dctrl.enable_manchester_tx (
    self,
    commitTransaction = True )
```

Enables the manchester encoding in transmission

5.34.2.6 force_phase()

```
def dctrl.Dctrl.force_phase (
    self,
    phase,
    commitTransaction = True )
```

Forces offset sampling phase for DCTRL

5.34.2.7 get_auto_phase_offset()

```
def dctrl.Dctrl.get_auto_phase_offset (
    self,
    commitTransaction = True )
```

gets the phase offset for the automatic phase detection

5.34.2.8 get_busy_transceiver_mask()

```
def dctrl.Dctrl.get_busy_transceiver_mask (
    self )
```

gets the actual value of the busy transceiver mask

5.34.2.9 get_busy_transceiver_status()

```
def dctrl.Dctrl.get_busy_transceiver_status (
    self )
```

gets the status of the busy flag for the transceivers

5.34.2.10 get_counters()

```
def dctrl.Dctrl.get_counters (
    self,
    reset_after = False,
    commitTransaction = True )
```

latches and reads all the counters

5.34.2.11 get_dclk_mask()

```
def dctrl.Dctrl.get_dclk_mask (
    self )
```

gets a 1-hot mask for transmission of DCLK

5.34.2.12 `get_dclk_parallel()`

```
def dctrl.Dctrl.get_dclk_parallel (
    self,
    index,
    commitTransaction = True )
```

gets the dclk parallel phase [0:45:360-45]

5.34.2.13 `get_dctrl_mask()`

```
def dctrl.Dctrl.get_dctrl_mask (
    self )
```

gets a 1-hot mask for transmission on DCTRL

5.34.2.14 `get_idelays()`

```
def dctrl.Dctrl.get_idelays (
    self )
```

gets the idelay values and check if values are consistent,
returns a tuple with the idelays 0 to 4

5.34.2.15 `get_input()`

```
def dctrl.Dctrl.get_input (
    self )
```

Read on which line the dctrl receives replies

5.34.2.16 `get_manchester_rx_detected()`

```
def dctrl.Dctrl.get_manchester_rx_detected (
    self,
    commitTransaction = True )
```

Returns 1 if manchester has been detected

5.34.2.17 get_manchester_tx()

```
def dctrl.Dctrl.get_manchester_tx (
    self )
```

check if the manchester encoding in transmission is enabled

5.34.2.18 get_mismatch()

```
def dctrl.Dctrl.get_mismatch (
    self )
```

returns the instantaneous value of the mismatch register

5.34.2.19 get_phase()

```
def dctrl.Dctrl.get_phase (
    self,
    commitTransaction = True )
```

Get offset sampling phase for DCTRL

5.34.2.20 get_wait_cycles()

```
def dctrl.Dctrl.get_wait_cycles (
    self,
    commitTransaction = True )
```

gets the wait cycles for the automatic phase detection

5.34.2.21 read_chip_reg()

```
def dctrl.Dctrl.read_chip_reg (
    self,
    address,
    chipid,
    commitTransaction = True )
```

Read from a specific chip Register. commitTransaction flushes all pending write operations and sends it to the board.

5.34.2.22 readreg_write_phase()

```
def dctrl.Dctrl.readreg_write_phase (
    self )
```

Read the Write phase register

5.34.2.23 release_phase_force()

```
def dctrl.Dctrl.release_phase_force (
    self,
    commitTransaction = True )
```

Release the force on offset sampling phase for DCTRL

5.34.2.24 reset_counters()

```
def dctrl.Dctrl.reset_counters (
    self,
    commitTransaction = True )
```

resets all the counters

5.34.2.25 set_auto_phase_offset()

```
def dctrl.Dctrl.set_auto_phase_offset (
    self,
    offset,
    commitTransaction = True )
```

sets the phase offset for the automatic phase detection

5.34.2.26 set_busy_transceiver_mask()

```
def dctrl.Dctrl.set_busy_transceiver_mask (
    self,
    mask )
```

sets the busy transceiver mask

5.34.2.27 set_dclk_mask()

```
def dctrl.Dctrl.set_dclk_mask (
    self,
    mask )
```

sets a 1-hot mask for transmission on DCTRL

5.34.2.28 set_dclk_parallel()

```
def dctrl.Dctrl.set_dclk_parallel (
    self,
    phase,
    index,
    commitTransaction = True )
```

sets the dclk parallel with the given phase [0:45:360-45]

5.34.2.29 set_dctrl_mask()

```
def dctrl.Dctrl.set_dctrl_mask (
    self,
    mask )
```

sets a 1-hot mask for transmission on DCTRL

5.34.2.30 set_idelays()

```
def dctrl.Dctrl.set_idelays (
    self,
    idelays = None,
    idelay_all = None,
    commitTransaction = True )
```

sets the input delay of all the dctrl ports, in they are not None

5.34.2.31 `set_input()`

```
def dctrl.Dctrl.set_input (
    self,
    dctrl_input = 0,
    force = False,
    commitTransaction = True )
```

Sets which DCTRL line is used to receive chip replies

5.34.2.32 `set_wait_cycles()`

```
def dctrl.Dctrl.set_wait_cycles (
    self,
    value,
    commitTransaction = True )
```

sets the wait cycles for the automatic phase detection

5.34.2.33 `write_chip_opcode()`

```
def dctrl.Dctrl.write_chip_opcode (
    self,
    opcode,
    commitTransaction = True )
```

Write a specific opcode to the chip. `commitTransaction` flushes all pending write operations and sends it to the board.

5.34.2.34 `write_chip_reg()`

```
def dctrl.Dctrl.write_chip_reg (
    self,
    address,
    data,
    chipid,
    commitTransaction = True,
    readback = False )
```

Write to a specific register on the chip. `commitTransaction` flushes all pending write operations and sends it to the board.

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/dctrl.py`

5.35 RUv1_regression.DctrlBaseTest Class Reference

Classes

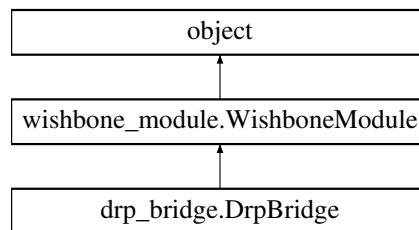
- class [TestDctrl](#)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.36 drp_bridge.DrpBridge Class Reference

Inheritance diagram for drp_bridge.DrpBridge:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def [write_drp](#) (self, lane, address, data, commitTransaction=True)
- def [read_drp](#) (self, lane, address, commitTransaction=True)

Static Public Attributes

- int **DRP_ADDRESS** = 1
- int **DRP_DATA** = 2

Additional Inherited Members

5.36.1 Member Function Documentation

5.36.1.1 read_drp()

```

def drp_bridge.DrpBridge.read_drp (
    self,
    lane,
    address,
    commitTransaction = True )
  
```

Read from the drp port. Set Lane via lane setting; defaults to self.lanes[0]

5.36.1.2 write_drp()

```
def drp_bridge.DrpBridge.write_drp (
    self,
    lane,
    address,
    data,
    commitTransaction = True )
```

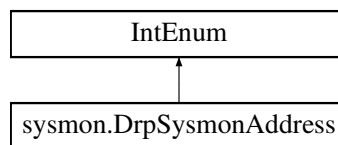
Write to the drp port. Set Lane via lane setting; defaults to self.lanes[0]

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/drp_bridge.py

5.37 sysmon.DrpSysmonAddress Class Reference

Inheritance diagram for sysmon.DrpSysmonAddress:



Static Public Attributes

- int **STATUS_TEMPERATURE** = 0x00
- int **STATUS_VCC_INT** = 0x01
- int **STATUS_VCC_AUX** = 0x02
- int **STATUS_VP_VN** = 0x03
- int **STATUS_VREF_P** = 0x04
- int **STATUS_VREF_N** = 0x05
- int **STATUS_VCC_BRAM** = 0x06
- int **STATUS_SUPPLY_OFF** = 0x08
- int **STATUS_ADC_OFF** = 0x09
- int **STATUS_ADC_GAIN** = 0x0A
- int **STATUS_MAX_TEMP** = 0x20
- int **STATUS_MAX_VCC_INT** = 0x21
- int **STATUS_MAX_VCC_AUX** = 0x22
- int **STATUS_MAX_VCC_BRAM** = 0x23
- int **STATUS_MIN_TEMP** = 0x24
- int **STATUS_MIN_VCC_INT** = 0x25
- int **STATUS_MIN_VCC_AUX** = 0x26
- int **STATUS_MIN_VCC_BRAM** = 0x27
- int **STATUS_FLAG_1** = 0x3E
- int **STATUS_FLAG_0** = 0x3F
- int **STATUS_VUSER_0** = 0x80
- int **STATUS_VUSER_1** = 0x81
- int **STATUS_VUSER_2** = 0x82

- int **STATUS_VUSER_3** = 0x83
- int **STATUS_MAX_VUSER_0** = 0xA0
- int **STATUS_MAX_VUSER_1** = 0xA1
- int **STATUS_MAX_VUSER_2** = 0xA2
- int **STATUS_MAX_VUSER_3** = 0xA3
- int **STATUS_MIN_VUSER_0** = 0xA8
- int **STATUS_MIN_VUSER_1** = 0xA9
- int **STATUS_MIN_VUSER_2** = 0xAA
- int **STATUS_MIN_VUSER_3** = 0xAB
- int **CONFIG_REGISTER_0** = 0x40
- int **CONFIG_REGISTER_1** = 0x41
- int **CONFIG_REGISTER_2** = 0x42
- int **CONFIG_REGISTER_3** = 0x43
- int **CONFIG_REGISTER_4** = 0x44
- int **CONFIG_SEQCHSEL0** = 0x46
- int **CONFIG_SEQCHSEL1** = 0x48
- int **CONFIG_SEQCHSEL2** = 0x49
- int **CONFIG_SLOWCHSEL0** = 0x7A
- int **CONFIG_SLOWCHSEL1** = 0x7B
- int **CONFIG_SLOWCHSEL2** = 0x7C
- int **CONFIG_SEQAVG0** = 0x47
- int **CONFIG_SEQAVG1** = 0x4A
- int **CONFIG_SEQAVG2** = 0x4B
- int **CONFIG_SEQINMODE0** = 0x4C
- int **CONFIG_SEQINMODE1** = 0x4D
- int **CONFIG_SEQACQ0** = 0x4E
- int **CONFIG_SEQACQ1** = 0x4F
- int **CONFIG_TEMP_UPP** = 0x50
- int **CONFIG_VCC_INT_UPP** = 0x51
- int **CONFIG_VCC_AUX_UPP** = 0x52
- int **CONFIG_OT_UPP** = 0x53
- int **CONFIG_TEMP_LOW** = 0x54
- int **CONFIG_VCC_INT_LOW** = 0x55
- int **CONFIG_VCC_AUX_LOW** = 0x56
- int **CONFIG_OT_LOW** = 0x57
- int **CONFIG_VCC_BRAM_UPP** = 0x58
- int **CONFIG_VCC_BRAM_LOW** = 0x5C
- int **CONFIG_VUSER_0_UPP** = 0x60
- int **CONFIG_VUSER_1_UPP** = 0x61
- int **CONFIG_VUSER_2_UPP** = 0x62
- int **CONFIG_VUSER_3_UPP** = 0x63
- int **CONFIG_VUSER_0_LOW** = 0x68
- int **CONFIG_VUSER_1_LOW** = 0x69
- int **CONFIG_VUSER_2_LOW** = 0x6A
- int **CONFIG_VUSER_3_LOW** = 0x6B

5.37.1 Detailed Description

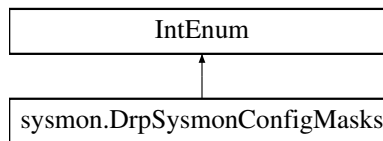
Dynamic Reconfiguration Port Register mapping for SYSMON

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/sysmon.py

5.38 sysmon.DrpSysmonConfigMasks Class Reference

Inheritance diagram for sysmon.DrpSysmonConfigMasks:



Static Public Attributes

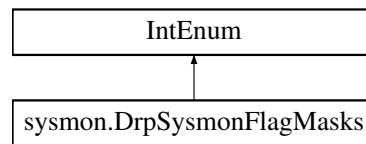
- tuple **CFG0_CH0** = (1 << 0)
- tuple **CFG0_CH1** = (1 << 1)
- tuple **CFG0_CH2** = (1 << 2)
- tuple **CFG0_CH3** = (1 << 3)
- tuple **CFG0_CH4** = (1 << 4)
- tuple **CFG0_CH5** = (1 << 5)
- tuple **CFG0_ACQ** = (1 << 8)
- tuple **CFG0_EC** = (1 << 9)
- tuple **CFG0_BU** = (1 << 10)
- tuple **CFG0_MUX** = (1 << 11)
- tuple **CFG0_AVG0** = (1 << 12)
- tuple **CFG0_AVG1** = (1 << 13)
- tuple **CFG0_CAVG** = (1 << 15)
- tuple **CFG1_OT** = (1 << 0)
- tuple **CFG1_ALM0** = (1 << 1)
- tuple **CFG1_ALM1** = (1 << 2)
- tuple **CFG1_ALM2** = (1 << 3)
- tuple **CFG1_CAL0** = (1 << 4)
- tuple **CFG1_CAL2** = (1 << 6)
- tuple **CFG1_CAL3** = (1 << 7)
- tuple **CFG1_ALM3** = (1 << 8)
- tuple **CFG1_ALM4** = (1 << 9)
- tuple **CFG1_ALM5** = (1 << 10)
- tuple **CFG1_ALM6** = (1 << 11)
- tuple **CFG1_SEQ0** = (1 << 12)
- tuple **CFG1_SEQ1** = (1 << 13)
- tuple **CFG1_SEQ2** = (1 << 14)
- tuple **CFG1_SEQ3** = (1 << 15)
- tuple **CFG2_CD0** = (1 << 8)
- tuple **CFG2_CD1** = (1 << 9)
- tuple **CFG2_CD2** = (1 << 10)
- tuple **CFG2_CD3** = (1 << 11)
- tuple **CFG2_CD4** = (1 << 12)
- tuple **CFG2_CD5** = (1 << 13)
- tuple **CFG2_CD6** = (1 << 14)
- tuple **CFG2_CD7** = (1 << 15)
- tuple **CFG3_ALM8** = (1 << 0)
- tuple **CFG3_ALM9** = (1 << 1)
- tuple **CFG3_ALM10** = (1 << 2)
- tuple **CFG3_ALM11** = (1 << 3)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/sysmon.py

5.39 sysmon.DrpSysmonFlagMasks Class Reference

Inheritance diagram for sysmon.DrpSysmonFlagMasks:



Static Public Attributes

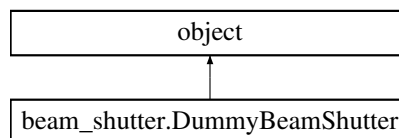
- tuple **FLAG0_ALM0** = (1 << 0)
- tuple **FLAG0_ALM1** = (1 << 1)
- tuple **FLAG0_ALM2** = (1 << 2)
- tuple **FLAG0_OT** = (1 << 3)
- tuple **FLAG0_ALM3** = (1 << 4)
- tuple **FLAG0_ALM4** = (1 << 5)
- tuple **FLAG0_ALM5** = (1 << 6)
- tuple **FLAG0_ALM6** = (1 << 7)
- tuple **FLAG0_JTGR** = (1 << 9)
- tuple **FLAG1_ALM8** = (1 << 0)
- tuple **FLAG1_ALM9** = (1 << 1)
- tuple **FLAG1_ALM10** = (1 << 2)
- tuple **FLAG1_ALM11** = (1 << 3)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/sysmon.py

5.40 beam_shutter.DummyBeamShutter Class Reference

Inheritance diagram for beam_shutter.DummyBeamShutter:



Public Member Functions

- def **__init__** (self)
- def **beam_off** (self)
- def **beam_on** (self)

Public Attributes

- **logger**

5.40.1 Member Function Documentation

5.40.1.1 beam_off()

```
def beam_shutter.DummyBeamShutter.beam_off (
    self )
```

Close the beam shutter

5.40.1.2 beam_on()

```
def beam_shutter.DummyBeamShutter.beam_on (
    self )
```

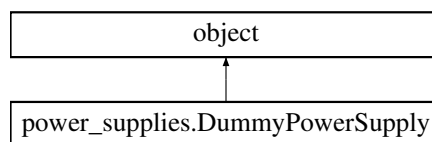
Open the beam shutter

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/beam_shutter.py

5.41 power_supplies.DummyPowerSupply Class Reference

Inheritance diagram for power_supplies.DummyPowerSupply:



Public Member Functions

- `def __init__ (self)`
- `def power_on (self)`
- `def power_off (self)`
- `def set_voltages (self, avdd=1.8, dvdd=1.8, pvdd=None, backbias=None)`
- `def get_values (self)`

Public Attributes

- **logger**

5.41.1 Detailed Description

Implementation of a dummy power supply

5.41.2 Member Function Documentation

5.41.2.1 power_on()

```
def power_supplies.DummyPowerSupply.power_on (
    self )
```

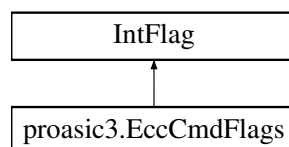
power on

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_supplies.py

5.42 proasic3.EccCmdFlags Class Reference

Inheritance diagram for proasic3.EccCmdFlags:



Static Public Attributes

- int **ECC_ENABLE** = 0
- int **CLEAR_STATUS_AND_COUNTERS** = 1

5.42.1 Detailed Description

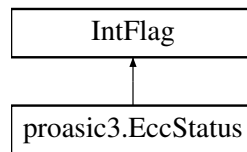
ECC command register bit meaning

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.43 proasic3.EccStatus Class Reference

Inheritance diagram for proasic3.EccStatus:



Static Public Attributes

- int **DB_ERROR** = 0
- int **SB_ERROR** = 1
- int **ECCSB_ERROR** = 2
- int **TMR_ERROR_IN_READ_FIFO** = 3

5.43.1 Detailed Description

ECC status

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.44 events.Event Class Reference

Public Member Functions

- def **__init__**(self)
- def **__str__**(self)

Public Attributes

- **chipid**
- **frame_data_start**
- **readout_flags**
- **errors**
- **hit_cnt**
- **hits**
- **hitlist**
- **currentRegion**
- **currentPosition**

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/events.py

5.45 events.EventDecoder Class Reference

Public Member Functions

- `def __init__ (self)`
- `def set\_max\_errors (self, max_errors)`
- `def print_events (self, number)`
- `def get\_eventcount\_map (self)`
- `def process (self, data)`
- `def hit2xy (self, region, current_pos, hitmap)`

Public Attributes

- **event**
- **error_event**
- **events**
- **state**
- **errors**
- **padding**
- **idleCount**
- **logger**
- **max_errors**
- **max_errors_reached**

Static Public Attributes

- `int CHIP_COMMA = 0B10111100`
- `int CHIP_IDLE = 0B11111111`
- `int CHIP_BUSY_ON = 0B11110001`
- `int CHIP_BUSY_OFF = 0B11110000`
- `int CHIP_HEADER = 0B1010`
- `int CHIP_TRAILER = 0B1011`
- `int CHIP_EMPTY_FRAME = 0B1110`
- `int CHIP_REGION = 0B110`
- `int CHIP_DATA_SHORT = 0B01`
- `int CHIP_DATA_LONG = 0B00`
- `int PADDING_GBTX = 0x00`

5.45.1 Member Function Documentation

5.45.1.1 `get_eventcount_map()`

```
def events.EventDecoder.get_eventcount_map (  
    self )
```

Return a Event count per chipid

5.45.1.2 process()

```
def events.EventDecoder.process (
    self,
    data )
```

Process a single byte of the protocol. Skip if max errors is reached

5.45.1.3 set_max_errors()

```
def events.EventDecoder.set_max_errors (
    self,
    max_errors )
```

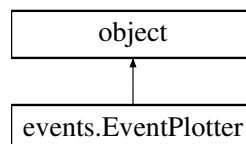
Set a maximum of errors before decoder stops processing

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/events.py

5.46 events.EventPlotter Class Reference

Inheritance diagram for events.EventPlotter:



Public Member Functions

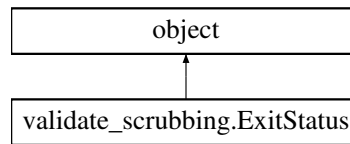
- `def __init__(self)`
- `def plot(self, coordinates_list)`
- `def heatmap_events(self, event_list)`
- `def heatmap_encoder(self, address_list)`
- `def plot_heatmap(self, hit_matrix, max_events=None)`
- `def heatmap(self, coordinates_list, max_events=None)`

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/events.py

5.47 validate_scrubbing.ExitStatus Class Reference

Inheritance diagram for validate_scrubbing.ExitStatus:



Public Member Functions

- `def __init__(self)`

Public Attributes

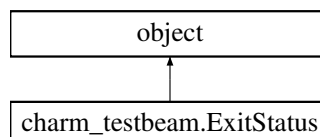
- `timestamp`
- `exit_id`
- `msg`
- `sca_gpio`
- `powerlatch_status`
- `chip0_pll_status`
- `run_error`
- `faults_injected`
- `fuse_triggered`
- `discardall_dp1_success`

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/software/py/validate_scrubbing.py`

5.48 charm_testbeam.ExitStatus Class Reference

Inheritance diagram for charm_testbeam.ExitStatus:



Public Member Functions

- `def __init__(self)`

Public Attributes

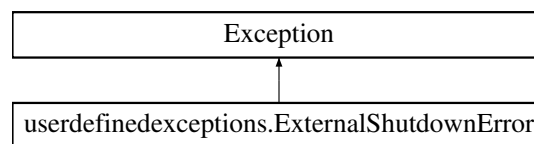
- **timestamp**
- **exit_id**
- **msg**
- **sca_gpio**
- **powerlatch_status**
- **chip0_pll_status**
- **run_error**
- **faults_injected**
- **fuse_triggered**
- **discardall_dp1_success**

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam.py

5.49 userdefinedexceptions.ExternalShutdownError Class Reference

Inheritance diagram for userdefinedexceptions.ExternalShutdownError:



Public Member Functions

- `def __init__(self, signum)`
- `def __signum__()`

Public Attributes

- **signum**

5.49.1 Detailed Description

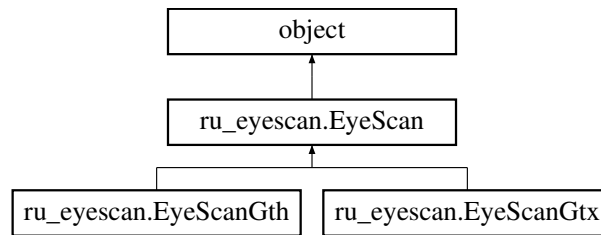
basic class to handle the external shutdown of the daemon

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/userdefinedexceptions.py

5.50 ru_eyescan.EyeScan Class Reference

Inheritance diagram for ru_eyescan.EyeScan:



Public Member Functions

- `def __init__(self, transceiver)`
- `def initialize(self, is_lpm_equalizer=True, horizontal_max=128, vertical_max=127)`
- `def eye_scan(self, v_steps=5, h_steps=5, prescale=4, ber=10, output_file=None, verbose=False, table=None, no_center=False, func=None)`
- `def resume_eye_scan(self, output_file_name, v_steps=5, h_steps=5, prescale=4, ber=10, verbose=False, no_center=False, func=None)`
- `def eye_scan_adaptive(self, v_steps=5, h_steps=5, prescale=4, ber=10, threshold=50, output_file_name=None, resume=False, no_center=False, verbose=False, func=None)`

Public Attributes

- `transceiver`
- `offset_v`
- `offset_v_ut`
- `offset_h`
- `prescale`
- `is_lpm`
- `eye_scan_horizontal_max`
- `eye_scan_vertical_max`
- `logger`

5.50.1 Detailed Description

Encapsulates Eye scan functionality

5.50.2 Member Function Documentation

5.50.2.1 eye_scan()

```
def ru_eyescan.EyeScan.eye_scan (
    self,
    v_steps = 5,
    h_steps = 5,
    prescale = 4,
    ber = 10,
    output_file = None,
    verbose = False,
    table = None,
    no_center = False,
    func = None )
```

Perform an eye scan measurement, defining Vertical steps, horizontal steps, default prescaling, ber(maximum prescaling) and an output file. An additional function can be added to be run during scanning. The output is a table with all scanned points. For resuming a scan, resume_eye_scan can be called.

5.50.2.2 eye_scan_adaptive()

```
def ru_eyescan.EyeScan.eye_scan_adaptive (
    self,
    v_steps = 5,
    h_steps = 5,
    prescale = 4,
    ber = 10,
    threshold = 50,
    output_file_name = None,
    resume = False,
    no_center = False,
    verbose = False,
    func = None )
```

Perform an adaptive eye scan measurement, defining Vertical steps, horizontal steps, default prescaling, ber(maximum prescaling) and an output file. An additional function can be added to be run during scanning. The output is a table with all scanned points. For resuming a scan, resume=True. Runs a Scan with minimal prescale through the whole range. All points below threshold errors will be stored and rerun with a higher prescale in a second pass. Continues until BER prescale is reached or no points are open.

5.50.2.3 initialize()

```
def ru_eyescan.EyeScan.initialize (
    self,
    is_lpm_equalizer = True,
    horizontal_max = 128,
    vertical_max = 127 )
```

Initialize eye scan functionality

5.50.2.4 resume_eye_scan()

```
def ru_eyescan.EyeScan.resume_eye_scan (
    self,
    output_file_name,
    v_steps = 5,
    h_steps = 5,
    prescale = 4,
    ber = 10,
    verbose = False,
    no_center = False,
    func = None )
```

Resume a previously started eye scan.

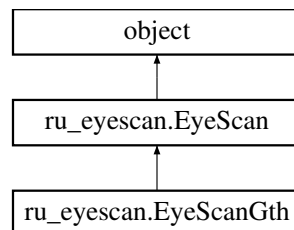
Reads scanned points from an old file and resumes from the last point

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_eyescan.py

5.51 ru_eyescan.EyeScanGth Class Reference

Inheritance diagram for ru_eyescan.EyeScanGth:



Public Member Functions

- def **__init__** (self, transceiver, vertical_range=0)
- def [get_vertical_range](#) (self)
- def **initialize** (self)

Public Attributes

- **eyescan_vs_range**
- **vert_offset_defaults**
- **horz_offset_defaults**
- **offset_h**
- **offset_v**
- **offset_v_ut**
- **prescale**

Static Public Attributes

- int **ES_CONTROL** = 0x3C
- int **ES_EYE_SCAN_EN** = 0x3C
- int **ES_PRESCALE** = 0x3C
- tuple **ES_QUAL_MASK** = (0x48,0x47,0x46,0x45,0x44)
- tuple **ES_SDATA_MASK** = (0x4D, 0x4C, 0x4B, 0x4A, 0x49)
- int **ES_HORZ_OFFSET** = 0x4F
- int **ES_VERT_OFFSET** = 0x97
- int **ES_CONTROL_STATE** = 0x153
- int **ES_SAMPLE_COUNT** = 0x152
- int **ES_ERROR_COUNT** = 0x151

5.51.1 Detailed Description

Eye scan functionality for GTX transceiver

5.51.2 Member Function Documentation

5.51.2.1 get_vertical_range()

```
def ru_eyescan.EyeScanGth.get_vertical_range (
    self )
```

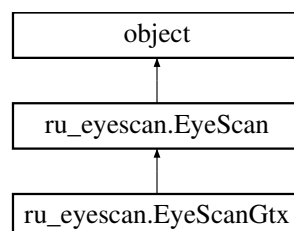
Return the vertical range per code in mV

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_eyescan.py

5.52 ru_eyescan.EyeScanGtx Class Reference

Inheritance diagram for ru_eyescan.EyeScanGtx:



Public Member Functions

- `def __init__(self, transceiver)`

Public Attributes

- `offset_h`

Static Public Attributes

- `int ES_VERT_OFFSET_PRESCALE = 0x3B`
- `int ES_HORZ_OFFSET = 0x3C`
- `tuple ES_SDATA_MASK = (0x3A, 0x39, 0x38, 0x37, 0x36)`
- `ES_QUAL_MASK = range(0x31, 0x36)`
- `int ES_CONTROL = 0x3D`
- `int ES_CONTROL_STATE = 0x151`
- `int ES_SAMPLE_COUNT = 0x150`
- `int ES_ERROR_COUNT = 0x14F`
- `int PMA_RSV2 = 0x82`

5.52.1 Detailed Description

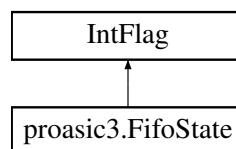
Eye scan functionality for GTX transceiver

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_eyescan.py`

5.53 proasic3.FifoState Class Reference

Inheritance diagram for proasic3.FifoState:



Static Public Attributes

- `int INTERNAL_RD_FIFO_FULL = 0`
- `int INTERNAL_RD_FIFO_EMPTY = 1`
- `int INTERNAL_WR_FIFO_FULL = 2`
- `int INTERNAL_WR_FIFO_EMPTY = 3`
- `int XILINX_FIFO_WR_FULL = 4`

5.53.1 Detailed Description

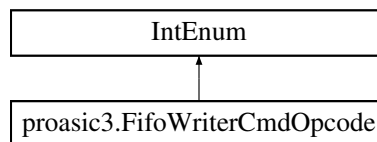
Fifo status flags

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.54 proasic3.FifoWriterCmdOpcode Class Reference

Inheritance diagram for proasic3.FifoWriterCmdOpcode:



Static Public Attributes

- int **WRITE_XILINX** = 0x01
- int **WRITE_BUS** = 0x02
- int **STOP** = 0x04
- int **CLEAR_ERRORS** = 0x08

5.54.1 Detailed Description

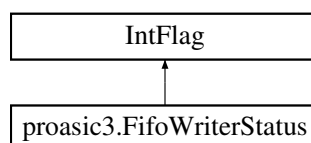
Fifo writer command opcodes

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.55 proasic3.FifoWriterStatus Class Reference

Inheritance diagram for proasic3.FifoWriterStatus:



Static Public Attributes

- int **READY** = 0
- int **ACTIVE_INPUT_IS_XIL_FIFO_NOT_I2C** = 1
- int **STOPPING** = 2
- int **FIFO_WRITE_ERROR** = 3
- int **WRONG_COMMAND_ERROR** = 4

5.55.1 Detailed Description

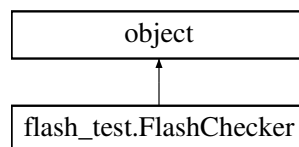
Fifo Writer flags

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.56 flash_test.FlashChecker Class Reference

Inheritance diagram for flash_test.FlashChecker:



Public Member Functions

- def **`__init__`** (self, path, startblock, stopblock, first_byte_bug, is_inverted, name="block_{0}.dat")
- def **`set_path`** (self, path)
- def **`set_name`** (self, name)
- def **`set_blocks`** (self, startblock, stopblock)
- def **`set_first_byte_bug`** (self, first_byte_bug)
- def **`set_is_inverted`** (self, is_inverted)
- def **`setup_logging`** (self)
- def **`get_filepath`** (self, block)
- def **`check_byte`** (self, byte, reference_byte)
- def **`analyse_block`** (self, block)

Public Attributes

- **path**
- **name**
- **startblock**
- **stopblock**
- **first_byte_bug**
- **is_inverted**
- **logger**
- **logdir**

5.56.1 Detailed Description

Class for checking the flash content readback

5.56.2 Constructor & Destructor Documentation

5.56.2.1 `__init__()`

```
def flash_test.FlashChecker.__init__ (
    self,
    path,
    startblock,
    stopblock,
    first_byte_bug,
    is_inverted,
    name = "block_{0}.dat" )
```

Init function

5.56.3 Member Function Documentation

5.56.3.1 `analyse_block()`

```
def flash_test.FlashChecker.analyse_block (
    self,
    block )
```

5.56.3.2 `set_first_byte_bug()`

```
def flash_test.FlashChecker.set_first_byte_bug (
    self,
    first_byte_bug )
```

Set attribute first_byte_bug

5.56.3.3 set_is_inverted()

```
def flash_test.FlashChecker.set_is_inverted (
    self,
    is_inverted )
```

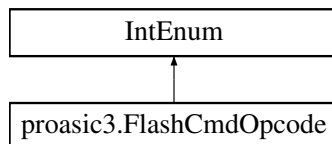
Set attribute is_inverted

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/flash_analysis/flash_test.py

5.57 proasic3.FlashCmdOpcode Class Reference

Inheritance diagram for proasic3.FlashCmdOpcode:



Static Public Attributes

- int **PAGE_WRITE** = 0x01
- int **PAGE_READ** = 0x02
- int **PAGE_READ_ID** = 0x04
- int **READ_STATUS** = 0x08
- int **RESET_FLASH** = 0x10
- int **BLOCK_ERASE** = 0x20

5.57.1 Detailed Description

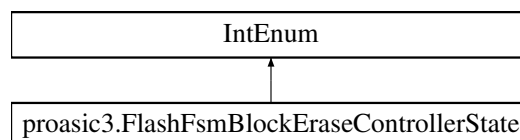
Flash command register opcodes

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.58 proasic3.FlashFsmBlockEraseControllerState Class Reference

Inheritance diagram for proasic3.FlashFsmBlockEraseControllerState:



Static Public Attributes

- int **IDLE** = 0x00
- int **WR_CMDL0** = 0x01
- int **WR0_CMDL0** = 0x02
- int **WR1_CMDL0** = 0x03
- int **END_CMDL0** = 0x04
- int **WR_ADD0** = 0x05
- int **WR0_ADD0** = 0x06
- int **WR1_ADD0** = 0x07
- int **END_ADD0** = 0x08
- int **WR_ADD1** = 0x09
- int **WR0_ADD1** = 0x0A
- int **WR1_ADD1** = 0x0B
- int **END_ADD1** = 0x0C
- int **WR_ADD2** = 0x0D
- int **WR0_ADD2** = 0x0E
- int **WR1_ADD2** = 0x0F
- int **END_ADD2** = 0x10
- int **WR_CMDL1** = 0x11
- int **WR0_CMDL1** = 0x12
- int **WR1_CMDL1** = 0x13
- int **END_CMDL1** = 0x14
- int **CHK_RnB** = 0x15
- int **WAIT1_RNB** = 0x16
- int **WAIT2_RNB** = 0x17
- int **WAIT3_RNB** = 0x18
- int **WAIT4_RNB** = 0x19
- int **WAIT5_RNB** = 0x1A
- int **STATS0_CMD** = 0x1B
- int **STATS1_CMD** = 0x1C
- int **STATS2_CMD** = 0x1D
- int **STATS3_CMD** = 0x1E
- int **RST0_CMD** = 0x1F
- int **RST1_CMD** = 0x21
- int **RST2_CMD** = 0x22
- int **RST3_CMD** = 0x23
- int **ENDRST_CMD** = 0x24
- int **DONE** = 0x25

5.58.1 Detailed Description

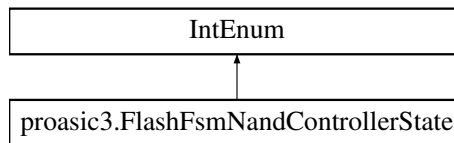
[21:16] FSM block erase controller

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.59 proasic3.FlashFsmNandControllerState Class Reference

Inheritance diagram for proasic3.FlashFsmNandControllerState:



Static Public Attributes

- int **IDLE** = 0x0
- int **INIT_RDID** = 0x1
- int **WAIT_RDID** = 0x2
- int **INIT_ERASE** = 0x3
- int **WAIT_ERASE** = 0x4
- int **INIT_READ** = 0x5
- int **WAIT_READ** = 0x6
- int **INIT_WRITE** = 0x7
- int **WAIT_WRITE** = 0x8
- int **SET_RESET** = 0x9
- int **INIT_RESET** = 0xA
- int **WAIT_RESET** = 0xB

5.59.1 Detailed Description

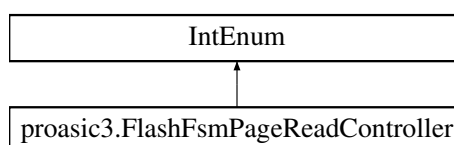
[3:0] FSM NAND Flash Controller

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.60 proasic3.FlashFsmPageReadController Class Reference

Inheritance diagram for proasic3.FlashFsmPageReadController:



Static Public Attributes

- int **IDLE** = 0x00
- int **WR_CMDL0** = 0x01
- int **WR0_CMDL0** = 0x02
- int **WR1_CMDL0** = 0x03
- int **END_CMDL0** = 0x04
- int **WR_ADD0** = 0x05
- int **WR0_ADD0** = 0x06
- int **WR1_ADD0** = 0x07
- int **END_ADD0** = 0x08
- int **WR_ADD1** = 0x09
- int **WR0_ADD1** = 0x0A
- int **WR1_ADD1** = 0x0B
- int **END_ADD1** = 0x0C
- int **WR_ADD2** = 0x0D
- int **WR0_ADD2** = 0x0E
- int **WR1_ADD2** = 0x0F
- int **END_ADD2** = 0x10
- int **WR_ADD3** = 0x11
- int **WR0_ADD3** = 0x12
- int **WR1_ADD3** = 0x13
- int **END_ADD3** = 0x14
- int **WR_ADD4** = 0x15
- int **WR0_ADD4** = 0x16
- int **WR1_ADD4** = 0x17
- int **END_ADD4** = 0x18
- int **RD_INIT** = 0x19
- int **RD0_DATA** = 0x1A
- int **RD1_DATA** = 0x1B
- int **FIFO_WAIT** = 0x1C
- int **END_DATA** = 0x1D
- int **WR_CMDL1** = 0x1E
- int **WR0_CMDL1** = 0x1F
- int **WR1_CMDL1** = 0x20
- int **END_CMDL1** = 0x21
- int **CHK_RnB** = 0x22
- int **WAIT1_RNB** = 0x23
- int **WAIT2_RNB** = 0x24
- int **WAIT3_RNB** = 0x25
- int **WAIT4_RNB** = 0x26
- int **WAIT5_RNB** = 0x27
- int **DONE** = 0x28

5.60.1 Detailed Description

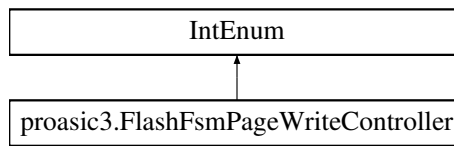
[15:9] FSM page read controller

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.61 proasic3.FlashFsmPageWriteController Class Reference

Inheritance diagram for proasic3.FlashFsmPageWriteController:



Static Public Attributes

- int **IDLE** = 0x00
- int **WR_CMDL0** = 0x01
- int **WR0_CMDL0** = 0x02
- int **WR1_CMDL0** = 0x03
- int **END_CMDL0** = 0x04
- int **WR_ADD0** = 0x05
- int **WR0_ADD0** = 0x06
- int **WR1_ADD0** = 0x07
- int **END_ADD0** = 0x08
- int **WR_ADD1** = 0x09
- int **WR0_ADD1** = 0x0A
- int **WR1_ADD1** = 0x0B
- int **END_ADD1** = 0x0C
- int **WR_ADD2** = 0x0D
- int **WR0_ADD2** = 0x0E
- int **WR1_ADD2** = 0x0F
- int **END_ADD2** = 0x10
- int **WR_ADD3** = 0x11
- int **WR0_ADD3** = 0x12
- int **WR1_ADD3** = 0x13
- int **END_ADD3** = 0x14
- int **WR_ADD4** = 0x15
- int **WR0_ADD4** = 0x16
- int **WR1_ADD4** = 0x17
- int **END_ADD4** = 0x18
- int **WR_INIT** = 0x19
- int **WR0_DATA** = 0x1A
- int **WR1_DATA** = 0x1B
- int **FIFO_WAIT** = 0x1C
- int **END_DATA** = 0x1D
- int **WR_CMDL1** = 0x1E
- int **WR0_CMDL1** = 0x1F
- int **WR1_CMDL1** = 0x20
- int **END_CMDL1** = 0x21
- int **WAIT1_RNB** = 0x22
- int **WAIT2_RNB** = 0x23
- int **WAIT3_RNB** = 0x24
- int **WAIT4_RNB** = 0x25
- int **WAIT5_RNB** = 0x26
- int **CHK_RnB** = 0x27
- int **DONE** = 0x28

5.61.1 Detailed Description

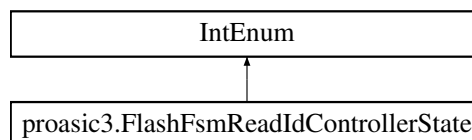
[9:4] FSM page write controller

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.62 proasic3.FlashFsmReadIdControllerState Class Reference

Inheritance diagram for proasic3.FlashFsmReadIdControllerState:



Static Public Attributes

- int **IDLE** = 0x00
- int **WR_CMDL0** = 0x01
- int **WR0_CMDL0** = 0x02
- int **WR1_CMDL0** = 0x03
- int **END_CMDL0** = 0x04
- int **WR_ADD0** = 0x05
- int **WR0_ADD0** = 0x06
- int **WR1_ADD0** = 0x07
- int **END_ADD0** = 0x08
- int **RD_INIT** = 0x09
- int **RD0_DATA** = 0x0A
- int **RD1_DATA** = 0x0B
- int **END_DATA** = 0x0C
- int **DONE** = 0x0D

5.62.1 Detailed Description

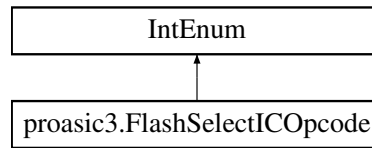
[25:22] FSM Read ID controller

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.63 proasic3.FlashSelectICOpcode Class Reference

Inheritance diagram for proasic3.FlashSelectICOpcode:



Static Public Attributes

- int **NA** = 0x00
- int **FLASH_IC1** = 0x01
- int **FLASH_IC2** = 0x02
- int **FLASH_BOTH_IC** = 0x03

5.63.1 Detailed Description

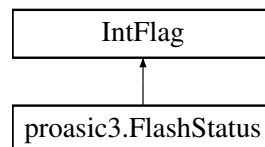
Flash command register opcodes

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.64 proasic3.FlashStatus Class Reference

Inheritance diagram for proasic3.FlashStatus:



Static Public Attributes

- int **TRX_DONE** = 0
- int **FIFO_STATUS** = 1
- int **TRX_ERROR** = 2
- int **WRITE_ACTIVE** = 3
- int **READ_ACTIVE** = 4
- int **COMMAND_ACTIVE** = 5
- int **TRX_DONE_STICKY** = 6

5.64.1 Detailed Description

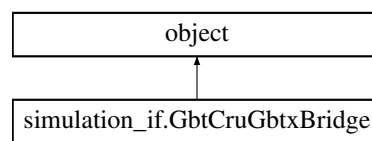
Flash controller status

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.65 simulation_if.GbtCruGbtxBridge Class Reference

Inheritance diagram for simulation_if.GbtCruGbtxBridge:



Public Member Functions

- def **__init__** (self, server)
- def **open** (self, filelist=range(4))
- def **close** (self)
- def **flush_gbt_cru** (self)
- def **flush_gbt_x** (self)
- def **clear_gbt_cru** (self)
- def **clear_gbt_x** (self)

Public Attributes

- **gbtx0_queue**
- **mutex**
- **files_names**
- **files**
- **server**

Static Public Attributes

- int **GBT_CRU_TX** = 3
- int **GBT_CRU_RX** = 2
- int **GBTX_TX** = 1
- int **GBTX_RX** = 0

5.65.1 Detailed Description

GBT communication bridge between GBTX and GBT_CRU simulations

5.65.2 Member Function Documentation

5.65.2.1 clear_gbt_cru()

```
def simulation_if.GbtCruGbtxBridge.clear_gbt_cru (  
    self )
```

Clear buffers: Close and reopen files

5.65.2.2 clear_gbt()

```
def simulation_if.GbtCruGbtxBridge.clear_gbt (  
    self )
```

Clear buffers: Close and reopen files

5.65.2.3 flush_gbt_cru()

```
def simulation_if.GbtCruGbtxBridge.flush_gbt_cru (  
    self )
```

Flush GBT CRU - Copy from GBTX

5.65.2.4 flush_gbt()

```
def simulation_if.GbtCruGbtxBridge.flush_gbt (  
    self )
```

Flush gbt: Copy from CRU

5.65.2.5 open()

```
def simulation_if.GbtCruGbtxBridge.open (
    self,
    filelist = range(4) )
```

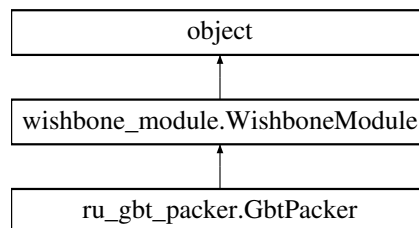
(re)open buffer files for data exchanges

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/simulation_if.py

5.66 ru_gbt_packer.GbtPacker Class Reference

Inheritance diagram for ru_gbt_packer.GbtPacker:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def **read_config** (self)
- def **dump_config** (self)
- def **initialize** (self, enable_data_forward=0, commitTransaction=True)
- def **set_settings** (self, enable_data_forward=0, commitTransaction=True)
- def **get_settings** (self, commitTransaction=True)
- def **update_masks** (self, lanes, commitTransaction=True)
- def **set_datalane_mask** (self, mask=0x0, commitTransaction=True)
- def **get_datalane_mask** (self, commitTransaction=True)
- def **set_fee_id** (self, fee_id, commitTransaction=True)
- def **get_fee_id** (self, commitTransaction=True)
- def **set_gbt_priority** (self, gbt_priority, commitTransaction=True)
- def **get_gbt_priority** (self, commitTransaction=True)
- def **set_max_word_count** (self, max_word_count, commitTransaction=True)
- def **get_max_word_count** (self, commitTransaction=True)
- def **set_max_packet_count** (self, max_packet_count, commitTransaction=True)
- def **get_max_packet_count** (self, commitTransaction=True)
- def **set_wait_data_timeout** (self, wait_data_timeout, commitTransaction=True)
- def **get_wait_data_timeout** (self, commitTransaction=True)
- def **set_send_data_timeout** (self, send_data_timeout, commitTransaction=True)
- def **get_send_data_timeout** (self, commitTransaction=True)
- def **set_lane_timeout** (self, lane_timeout, commitTransaction=True)
- def **get_lane_timeout** (self, commitTransaction=True)

Static Public Attributes

- int **ADD_FEE_ID** = 0
- int **ADD_GBT_PRIORITY** = 1
- int **ADD_MAX_WORD_COUNT** = 2
- int **ADD_MAX_PACKET_COUNT** = 3
- int **ADD_WAIT_DATA_TIMEOUT** = 4
- int **ADD_SEND_DATA_TIMEOUT** = 5
- int **ADD_LANE_TIMEOUT** = 6
- int **ADD_MASK_DATA_LANE_L** = 7
- int **ADD_MASK_DATA_LANE_H** = 8
- int **ADD_GBT_PACKER_SETTINGS** = 9

Additional Inherited Members

5.66.1 Detailed Description

Trigger Handler Interface

5.66.2 Member Function Documentation

5.66.2.1 dump_config()

```
def ru_gbt_packer.GbtPacker.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.66.2.2 initialize()

```
def ru_gbt_packer.GbtPacker.initialize (
    self,
    enable_data_forward = 0,
    commitTransaction = True )
```

Initialize trigger handler

5.66.2.3 update_masks()

```
def ru_gbt_packer.GbtPacker.update_masks (
    self,
    lanes,
    commitTransaction = True )
```

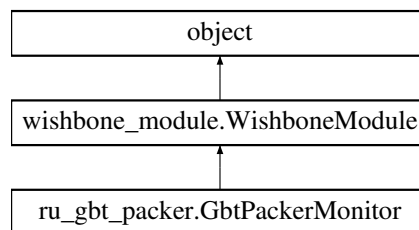
Mask out all transceivers not in [lanes]

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_gbt_packer.py

5.67 ru_gbt_packer.GbtPackerMonitor Class Reference

Inheritance diagram for ru_gbt_packer.GbtPackerMonitor:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def **reset_counters** (self, commitTransaction=True)
- def **latch_counters** (self, commitTransaction=False)
- def **read_counters** (self, counters=None)

Public Attributes

- **REG_LATCH**
- **REG_RESET**
- **nr_regs**
- **counter_mapping**
- **multi_counter_order**

Static Public Attributes

- list **wb_register_mapping**
- **nr_counter_regs** = len(wb_register_mapping)

5.67.1 Member Function Documentation

5.67.1.1 latch_counters()

```
def ru_gbt_packer.GbtPackerMonitor.latch_counters (
    self,
    commitTransaction = False )
```

Latches values into counters

5.67.1.2 read_counters()

```
def ru_gbt_packer.GbtPackerMonitor.read_counters (
    self,
    counters = None )
```

Read Counters(array) from lanes(array)

5.67.1.3 reset_counters()

```
def ru_gbt_packer.GbtPackerMonitor.reset_counters (
    self,
    commitTransaction = True )
```

Reset all counters

5.67.2 Member Data Documentation

5.67.2.1 wb_register_mapping

```
list ru_gbt_packer.GbtPackerMonitor.wb_register_mapping [static]
```

Initial value:

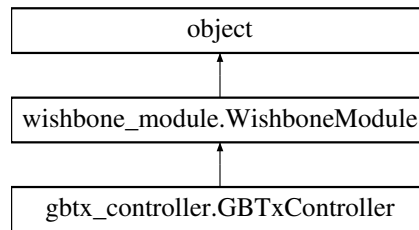
```
= [
    "TRIGGER_RD_LOW",
    "TRIGGER_RD_HIGH",
    "SEND_SOP_LOW",
    "SEND_SOP_HIGH",
    "SEND_EOP_LOW",
    "SEND_EOP_HIGH",
    "PACKET_DONE_LOW",
    "PACKET_DONE_HIGH",
    "PACKET_TIMEOUT",
    "STATE_MISMATCH",
    "FIFO_FULL",
    "FIFO_OVERFLOW",
    "EMPTY_PACKET"
]
```

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_gbt_packer.py

5.68 gbtx_controller.GBTxController Class Reference

Inheritance diagram for gbtx_controller.GBTxController:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def **enable_controller** (self, commitTransaction=True)
- def **disable_controller** (self, commitTransaction=True)
- def **reset_gbtx_asic** (self, commitTransaction=True)
- def **set_idelay** (self, idelays=10 * [None], idelay_all=None, commitTransaction=True)
- def **get_idelay** (self)
- def **set_bitslip_rx** (self, value, commitTransaction=True)
- def **get_bitslip_rx** (self)
- def **set_bitslip_tx** (self, value, commitTransaction=True)
- def **get_bitslip_tx** (self)
- def **set_tx_pattern** (self, value, commitTransaction=True)
- def **get_tx_pattern** (self)
- def **set_tx1_pattern** (self, value, commitTransaction=True)
- def **get_tx1_pattern** (self)
- def **get_mismatch** (self)
- def **get_loss_lock_flag** (self)
- def **rst_loss_lock_flag** (self)
- def **dump_config** (self)

Public Attributes

- **verbose**

5.68.1 Detailed Description

GBTx controller wishbone slave

5.68.2 Member Function Documentation

5.68.2.1 disable_controller()

```
def gbtx_controller.GBTxController.disable_controller (
    self,
    commitTransaction = True )
```

Enables the gbtx controller

5.68.2.2 dump_config()

```
def gbtx_controller.GBTxController.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.68.2.3 enable_controller()

```
def gbtx_controller.GBTxController.enable_controller (
    self,
    commitTransaction = True )
```

Enables the gbtx controller

5.68.2.4 get_bitslip_rx()

```
def gbtx_controller.GBTxController.get_bitslip_rx (
    self )
```

gets the bitslip value in rx to be used on the GBTx module

5.68.2.5 get_bitslip_tx()

```
def gbtx_controller.GBTxController.get_bitslip_tx (
    self )
```

gets the bitslip value in tx to be used on the GBTx module

5.68.2.6 get_idelay()

```
def gbt_x_controller.GBTxController.get_idelay (  
    self )
```

gets the idelay values and check if values are consistent,
returns a tuple with the idelays 0 to 9

5.68.2.7 get_loss_lock_flag()

```
def gbt_x_controller.GBTxController.get_loss_lock_flag (  
    self )
```

returns the status of the RXRDY lost flag

5.68.2.8 get_mismatch()

```
def gbt_x_controller.GBTxController.get_mismatch (  
    self )
```

returns the mismatch flags for the gbt_x module.
Returns a tuple containing a dictionary, and the bare value of the register read

5.68.2.9 get_tx1_pattern()

```
def gbt_x_controller.GBTxController.get_tx1_pattern (  
    self )
```

gets the transmission pattern to be used on the GBTx module

5.68.2.10 get_tx_pattern()

```
def gbt_x_controller.GBTxController.get_tx_pattern (  
    self )
```

gets the transmission pattern to be used on the GBTx module

5.68.2.11 reset_gbtx_asic()

```
def gbtx_controller.GBTxController.reset_gbtx_asic (
    self,
    commitTransaction = True )
```

Resets the GBTx ASIC

5.68.2.12 rst_loss_lock_flag()

```
def gbtx_controller.GBTxController.rst_loss_lock_flag (
    self )
```

resets the RXRDY lost flag to 0

5.68.2.13 set_bitslip_rx()

```
def gbtx_controller.GBTxController.set_bitslip_rx (
    self,
    value,
    commitTransaction = True )
```

sets the bitslip value in rx to be used on the GBTx module

5.68.2.14 set_bitslip_tx()

```
def gbtx_controller.GBTxController.set_bitslip_tx (
    self,
    value,
    commitTransaction = True )
```

sets the bitslip value in tx to be used on the GBTx module

5.68.2.15 set_idelay()

```
def gbt_x_controller.GBTxController.set_idelay (
    self,
    idelays = 10*[None],
    idelay_all = None,
    commitTransaction = True )
```

sets the input delay of all the elinks groups ports, in they are not None

5.68.2.16 set_tx1_pattern()

```
def gbt_x_controller.GBTxController.set_tx1_pattern (
    self,
    value,
    commitTransaction = True )
```

sets the transmission pattern to be used on the GBTx module

5.68.2.17 set_tx_pattern()

```
def gbt_x_controller.GBTxController.set_tx_pattern (
    self,
    value,
    commitTransaction = True )
```

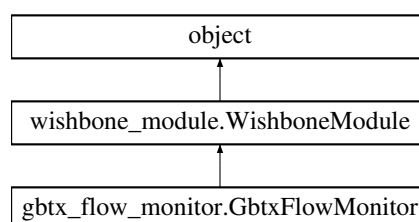
sets the transmission pattern to be used on the GBTx module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_x_controller.py

5.69 gbt_x_flow_monitor.GbtxFLOWMonitor Class Reference

Inheritance diagram for gbt_x_flow_monitor.GbtxFLOWMonitor:



Public Member Functions

- `def __init__ (self, moduleid, board_obj)`
- `def reset_counters (self, mask=2 **WsGbtxFLOWMonitorAddress_bit.NUM_COUNTERS-1, commit↵ Transaction=True)`
- `def latch_counters (self, mask=(2 **WsGbtxFLOWMonitorAddress_bit.NUM_COUNTERS) -1, commit↵ Transaction=True)`
- `def read_counters (self)`
- `def dump_config (self)`

Public Attributes

- `verbose`

5.69.1 Detailed Description

GBTx flow monitor wishbone slave

5.69.2 Member Function Documentation

5.69.2.1 dump_config()

```
def gbtx_flow_monitor.GbtxFLOWMonitor.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.69.2.2 latch_counters()

```
def gbtx_flow_monitor.GbtxFLOWMonitor.latch_counters (
    self,
    mask = (2**WsGbtxFLOWMonitorAddress_bit.NUM_COUNTERS) -1,
    commitTransaction = True )
```

Latches the counters

A bit mask can be eventually provided: if the n-th bit is one the corresponding counter is latched.
Default it all reset

5.69.2.3 read_counters()

```
def gbt_x_flow_monitor.GbtXFlowMonitor.read_counters (
    self )
```

Reads all counters: returns a list.
The list is ordered as in WsGbtXflowmonitoraddress_bit

5.69.2.4 reset_counters()

```
def gbt_x_flow_monitor.GbtXFlowMonitor.reset_counters (
    self,
    mask = 2**WsGbtXFlowMonitorAddress_bit.NUM_COUNTERS-1,
    commitTransaction = True )
```

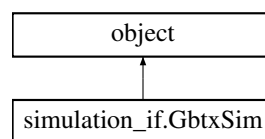
Resets the counters
A bit mask can be eventually provided: if the n-th bit is one the corresponding counter is reset.
Default it all reset

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_x_flow_monitor.py

5.70 simulation_if.GbtXSim Class Reference

Inheritance diagram for simulation_if.GbtXSim:



Public Member Functions

- def **__init__** (self, server, gbtXnum=0)
- def **open** (self)
- def **close** (self)
- def **flush_gbtX** (self)
- def **clear** (self)
- def **writeToGbt** (self, data)
- def **readFromGbt** (self)
- def **read_sort_data** (self)
- def **get_data** (self)
- def **get_swt** (self)

Public Attributes

- **gbtx_queue**
- **gbtxnum**
- **mutex**
- **files**
- **server**
- **logger**
- **swt_queue**
- **data_queue**

Static Public Attributes

- int **SWT_CODE** = 0x3
- int **SOP_CODE** = 0x1
- int **EOP_CODE** = 0x2

5.70.1 Detailed Description

Gbt Communication wrapper connecting to simulation.

This wrapper writes read/write actions to a set of fifos, which will be checked by a simulation testbench and then forwarded to the simulation environment

5.70.2 Member Function Documentation

5.70.2.1 clear()

```
def simulation_if.GbtxSim.clear (  
    self )
```

Clear buffers: Close and reopen files

5.70.2.2 flush_gbtx()

```
def simulation_if.GbtxSim.flush_gbtx (  
    self )
```

Flush gbtx: send data to fifo

5.70.2.3 get_data()

```
def simulation_if.GbtSim.get_data (
    self )

return a GBT data word (datavalid=1)
```

5.70.2.4 get_swt()

```
def simulation_if.GbtSim.get_swt (
    self )

return a single word transaction
```

5.70.2.5 open()

```
def simulation_if.GbtSim.open (
    self )

(re)open buffer files for data exchanges
```

5.70.2.6 readFromGbt()

```
def simulation_if.GbtSim.readFromGbt (
    self )

Read all data from gbt port
```

5.70.2.7 writeToGbt()

```
def simulation_if.GbtSim.writeToGbt (
    self,
    data )

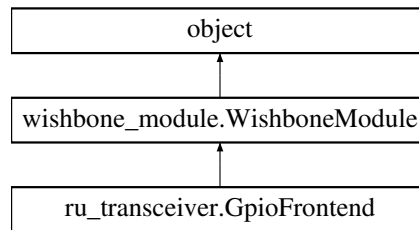
Write to GBT port. Expects tuple (valid,data). Data is 10 byte bytearray
```

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/simulation_if.py

5.71 ru_transceiver.GpioFrontend Class Reference

Inheritance diagram for ru_transceiver.GpioFrontend:



Public Member Functions

- def **__init__** (self, moduleid, board_obj, transceivers=None)
- def **read_config** (self)
- def **subset** (self, transceivers)
- def **set_transceivers** (self, transceivers)
- def **get_transceivers** (self)
- def **initialize** (self, commitTransaction=True)
- def **align_transceivers** (self, check_aligned=True, max_retries=10)
- def **scan_idelays** (self, stepsize=10, waittime=0.1, set_optimum=True, verbose=True)
- def **is_aligned** (self)
- def **enable_alignment** (self, enable=True, commitTransaction=True)
- def **enable_realign** (self, enable=True, commitTransaction=True)
- def **enable_data** (self, enable=True, commitTransaction=True)
- def **load_idelay** (self, value, commitTransaction=True)
- def **enable_prbs** (self, enable=True, commitTransaction=True)
- def **reset_prbs_counter** (self, commitTransaction=True)
- def **read_prbs_counter** (self, reset=False)
- def **set_lane_chip_mask** (self, lane, mask, commitTransaction=True)
- def **get_lane_chip_mask** (self, lane, commitTransaction=True)

Public Attributes

- **transceivers**

Static Public Attributes

- int **NR_TRANSCEIVERS** = 28
- int **ENABLE_ALIGNMENT_L** = 0
- int **ENABLE_ALIGNMENT_H** = 1
- int **ALIGNMENT_STATUS_L** = 2
- int **ALIGNMENT_STATUS_H** = 3
- int **ENABLE_REALIGN_L** = 4
- int **ENABLE_REALIGN_H** = 5
- int **ENABLE_DATA_L** = 6
- int **ENABLE_DATA_H** = 7
- int **IDELAY_VALUE** = 8
- int **IDELAY_LOAD_L** = 9
- int **IDELAY_LOAD_H** = 10
- int **ENABLE_PRBS_CHECK_L** = 11
- int **ENABLE_PRBS_CHECK_H** = 12
- int **PRBS_RESET_COUNTER_L** = 13
- int **PRBS_RESET_COUNTER_H** = 14
- int **PRBS_COUNTER_LANE_0** = 15
- int **LANE_CHIP_MASK_0** = 47

5.71.1 Detailed Description

Software module for communicating with the `alpine_frontend_ob_wishbone` module.

5.71.2 Member Function Documentation

5.71.2.1 `align_transceivers()`

```
def ru_transceiver.GpioFrontend.align_transceivers (
    self,
    check_aligned = True,
    max_retries = 10 )
```

Perform transceiver alignment procedure:

- * enable alignment
- * check that all transceivers are aligned
- * disable alignment

5.71.2.2 `enable_alignment()`

```
def ru_transceiver.GpioFrontend.enable_alignment (
    self,
    enable = True,
    commitTransaction = True )
```

Enable alignment flag for Transceivers. Does not touch transceivers not in list

5.71.2.3 `enable_data()`

```
def ru_transceiver.GpioFrontend.enable_data (
    self,
    enable = True,
    commitTransaction = True )
```

Enable data flag for Transceivers. Does not touch transceivers not in list

5.71.2.4 enable_prbs()

```
def ru_transceiver.GpioFrontend.enable_prbs (
    self,
    enable = True,
    commitTransaction = True )
```

Enable PRBS checker for Transceivers. Does not touch transceivers not in list

5.71.2.5 enable_realign()

```
def ru_transceiver.GpioFrontend.enable_realign (
    self,
    enable = True,
    commitTransaction = True )
```

Enable realign flag for Transceivers. Does not touch transceivers not in list

5.71.2.6 get_transceivers()

```
def ru_transceiver.GpioFrontend.get_transceivers (
    self )
```

Return board transceivers

5.71.2.7 initialize()

```
def ru_transceiver.GpioFrontend.initialize (
    self,
    commitTransaction = True )
```

Initialise Transceiver: enable alignment

5.71.2.8 is_aligned()

```
def ru_transceiver.GpioFrontend.is_aligned (
    self )
```

Return lock alignment status of each transceivers as array

5.71.2.9 read_prbs_counter()

```
def ru_transceiver.GpioFrontend.read_prbs_counter (
    self,
    reset = False )
```

Read / log the total amount of Prbs errors since the last check

5.71.2.10 reset_prbs_counter()

```
def ru_transceiver.GpioFrontend.reset_prbs_counter (
    self,
    commitTransaction = True )
```

Reset internal Transceiver PRBS counter

5.71.2.11 scan_idelays()

```
def ru_transceiver.GpioFrontend.scan_idelays (
    self,
    stepsize = 10,
    waittime = 0.1,
    set_optimum = True,
    verbose = True )
```

Scan all Idelay values to find the best phase. Requires chips to be set up for PRBS400

5.71.2.12 set_lane_chip_mask()

```
def ru_transceiver.GpioFrontend.set_lane_chip_mask (
    self,
    lane,
    mask,
    commitTransaction = True )
```

Set the chip mask for lane <lane>.
A masked lane will not be considered for generating end of event for packaging

5.71.2.13 set_transceivers()

```
def ru_transceiver.GpioFrontend.set_transceivers (
    self,
    transceivers )
```

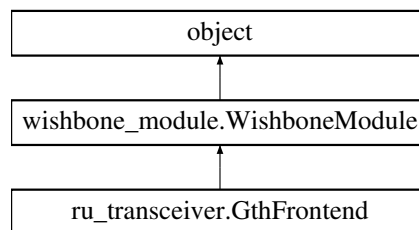
Set array of transceivers to be addressed

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_transceiver.py

5.72 ru_transceiver.GthFrontend Class Reference

Inheritance diagram for ru_transceiver.GthFrontend:



Public Member Functions

- def **__init__** (self, moduleid, board_obj, drp_bridge, transceivers=None)
- def **read_config** (self)
- def **set_transceivers** (self, transceivers)
- def **get_transceivers** (self)
- def **initialize** (self, commitTransaction=True, check_reset_done=True, max_retries=10)
- def **align_transceivers** (self, check_aligned=True, max_retries=10)
- def **is_reset_done** (self)
- def **is_cdr_locked** (self)
- def **is_aligned** (self)
- def **get_gth_status** (self)
- def **enable_alignment** (self, enable=True, commitTransaction=True)
- def **enable_data** (self, enable=True, commitTransaction=True)
- def **reset_receivers** (self, commitTransaction=True)
- def **enable_prbs** (self, enable=True, commitTransaction=True)
- def **reset_prbs_counter** (self, commitTransaction=True)
- def **read_prbs_counter** (self, reset=False)
- def **write_drp** (self, address, data, commitTransaction=True, transceiver=None)
- def **read_drp** (self, address, commitTransaction=True, transceiver=None)
- def **readback_all_drp** (self)

Public Attributes

- **drp_bridge**
- **transceivers**

Static Public Attributes

- int **NR_TRANSCEIVERS** = 9
- int **ENABLE_ALIGNMENT** = 0
- int **ALIGNMENT_STATUS** = 1
- int **ENABLE_DATA** = 2
- list **DRP_READBACK_ADDRESSES** = [2, 3, 4, 5, 6, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 202, 203, 204, 205, 206, 207, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 355, 361]
- int **GTH_RESET** = 5
- int **GTH_STATUS** = 6
- int **ENABLE_PRBS_CHECK** = 7
- int **PRBS_COUNTER_RESET** = 8

5.72.1 Detailed Description

Software module for communicating with the `alpine_frontend_ib_wishbone` module.

5.72.2 Member Function Documentation

5.72.2.1 `align_transceivers()`

```
def ru_transceiver.GthFrontend.align_transceivers (
    self,
    check_aligned = True,
    max_retries = 10 )
```

Perform transceiver alignment procedure:

- * enable alignment
- * check that all transceivers are aligned
- * disable alignment

5.72.2.2 `enable_alignment()`

```
def ru_transceiver.GthFrontend.enable_alignment (
    self,
    enable = True,
    commitTransaction = True )
```

Enable alignment flag for Transceivers. Does not touch transceivers not in list

5.72.2.3 enable_data()

```
def ru_transceiver.GthFrontend.enable_data (
    self,
    enable = True,
    commitTransaction = True )
```

Enable data flag for Transceivers. Does not touch transceivers not in list

5.72.2.4 enable_prbs()

```
def ru_transceiver.GthFrontend.enable_prbs (
    self,
    enable = True,
    commitTransaction = True )
```

Enable PRBS checker for Transceivers. Does not touch transceivers not in list

5.72.2.5 get_transceivers()

```
def ru_transceiver.GthFrontend.get_transceivers (
    self )
```

Return board transceivers

5.72.2.6 initialize()

```
def ru_transceiver.GthFrontend.initialize (
    self,
    commitTransaction = True,
    check_reset_done = True,
    max_retries = 10 )
```

Initialise Transceiver: Reset GTH block, enable alignment

5.72.2.7 is_aligned()

```
def ru_transceiver.GthFrontend.is_aligned (
    self )
```

Return lock alignment status of each transceivers as array

5.72.2.8 is_cdr_locked()

```
def ru_transceiver.GthFrontend.is_cdr_locked (
    self )
```

Return cdr status for each transceiver as array

5.72.2.9 is_reset_done()

```
def ru_transceiver.GthFrontend.is_reset_done (
    self )
```

Return Reset status of GTH block

5.72.2.10 read_drp()

```
def ru_transceiver.GthFrontend.read_drp (
    self,
    address,
    commitTransaction = True,
    transceiver = None )
```

Read from the drp port. Set Transceiver via transceiver setting; defaults to self.transceivers[0]

5.72.2.11 read_prbs_counter()

```
def ru_transceiver.GthFrontend.read_prbs_counter (
    self,
    reset = False )
```

Read / log the total amount of Prbs errors since the last check

5.72.2.12 readback_all_drp()

```
def ru_transceiver.GthFrontend.readback_all_drp (
    self )
```

Read back all Valid DRP ports of all registered transceivers

5.72.2.13 reset_prbs_counter()

```
def ru_transceiver.GthFrontend.reset_prbs_counter (
    self,
    commitTransaction = True )
```

Reset internal Transceiver PRBS counter

5.72.2.14 reset_receivers()

```
def ru_transceiver.GthFrontend.reset_receivers (
    self,
    commitTransaction = True )
```

Reset Receivers paths for Transceivers

5.72.2.15 set_transceivers()

```
def ru_transceiver.GthFrontend.set_transceivers (
    self,
    transceivers )
```

Set array of transceivers to be addressed

5.72.2.16 write_drp()

```
def ru_transceiver.GthFrontend.write_drp (
    self,
    address,
    data,
    commitTransaction = True,
    transceiver = None )
```

Write to the drp port. Set Transceiver via transceiver setting; defaults to self.transceivers[0]

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_transceiver.py

5.73 hameg.Hameg Class Reference

Public Member Functions

- def **__init__** (self, port=DEFAULT_PORT, reset=True)
- def **setup_standalone** (self)
- def **setup_logging** (self)
- def **get_model** (self)
- def **configure_channel** (self, channel, voltage, current)
- def **get_channel_config** (self, channel)
- def **get_channel_config_all** (self)
- def **activate_channel** (self, channel)
- def **deactivate_channel** (self, channel)
- def **activate_channels** (self, channels)
- def **deactivate_channels** (self, channels)
- def **activate_output** (self, activate=True)
- def **deactivate_output** (self)
- def **is_channel_on** (self, channel)
- def **get_fuse_triggered** (self, channel)
- def **get_current** (self, channel)
- def **get_voltage** (self, channel)
- def **get_power** (self, channel)
- def **get_power_all** (self)

Public Attributes

- **logger**
- **link**
- **number_channels**
- **logdir**

5.73.1 Detailed Description

Control of HAMEG power supply

5.73.2 Member Function Documentation

5.73.2.1 activate_channel()

```
def hameg.Hameg.activate_channel (
    self,
    channel )
```

Activate given channel

5.73.2.2 activate_channels()

```
def hameg.Hameg.activate_channels (
    self,
    channels )
```

Activate a list of given channels

5.73.2.3 activate_output()

```
def hameg.Hameg.activate_output (
    self,
    activate = True )
```

Activates voltage output

5.73.2.4 configure_channel()

```
def hameg.Hameg.configure_channel (
    self,
    channel,
    voltage,
    current )
```

Configure channel N with given voltage, current. Fuses are on and linked to the same channel. Channel gets activated

5.73.2.5 deactivate_channel()

```
def hameg.Hameg.deactivate_channel (
    self,
    channel )
```

Deactivate given channel

5.73.2.6 deactivate_channels()

```
def hameg.Hameg.deactivate_channels (
    self,
    channels )
```

Deactivate a list of given channels

5.73.2.7 deactivate_output()

```
def hameg.Hameg.deactivate_output (
    self )
```

Deactivates voltage output

5.73.2.8 get_channel_config()

```
def hameg.Hameg.get_channel_config (
    self,
    channel )
```

Get the channel configuration

5.73.2.9 get_channel_config_all()

```
def hameg.Hameg.get_channel_config_all (
    self )
```

Get configuratoin for all channels

5.73.2.10 get_current()

```
def hameg.Hameg.get_current (
    self,
    channel )
```

Get the current of a given channel in mA

5.73.2.11 get_fuse_triggered()

```
def hameg.Hameg.get_fuse_triggered (
    self,
    channel )
```

Returns whether or not a fuse was triggered

5.73.2.12 get_model()

```
def hameg.Hameg.get_model (
    self )
```

Return model and type

5.73.2.13 get_power()

```
def hameg.Hameg.get_power (
    self,
    channel )
```

Return Voltage, Current, Power of channel

5.73.2.14 get_power_all()

```
def hameg.Hameg.get_power_all (
    self )
```

Gets the voltage, current and power of all channels

5.73.2.15 `get_voltage()`

```
def hameg.Hameg.get_voltage (
    self,
    channel )
```

Get the voltage of a given channel in V

5.73.2.16 `is_channel_on()`

```
def hameg.Hameg.is_channel_on (
    self,
    channel )
```

Reports if a channel is active

5.73.2.17 `setup_logging()`

```
def hameg.Hameg.setup_logging (
    self )
```

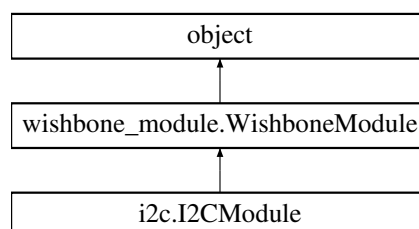
Sets up the logging

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/hameg.py`

5.74 `i2c.I2CModule` Class Reference

Inheritance diagram for `i2c.I2CModule`:



Public Member Functions

- def `__init__` (self, moduleid, board_obj)
- def `dump_config` (self)

Additional Inherited Members

5.74.1 Detailed Description

Class to handle I2C modules

5.74.2 Member Function Documentation

5.74.2.1 `dump_config()`

```
def i2c.I2CModule.dump_config (  
    self )
```

Dump the modules state and configuration as a string

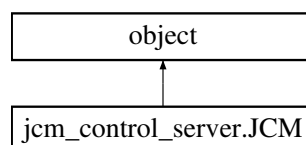
Reimplemented from [wishbone_module.WishboneModule](#).

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/i2c.py`

5.75 jcm_control_server.JCM Class Reference

Inheritance diagram for `jcm_control_server.JCM`:



Public Member Functions

- `def __init__ (self)`
- `def setup_logging (self)`
- `def start_calibrate_device (self)`
- `def start_full_configure (self, bitfile=None)`
- `def is_stopped (self)`
- `def start_readback (self)`
- `def rename_readback_file (self, file_name=None, restore=False)`
- `def remove_readback_files (self)`
- `def compare_readback (self, basename1, basename2)`
- `def start_blind_scrubber (self)`
- `def stop_blind_scrubber (self)`
- `def start_random_fault_injection (self, delay, n_faults, delay_after, correction)`
- `def stop_random_fault_injection (self)`
- `def read_messages (self)`
- `def test_function (self)`

Public Attributes

- `message_queue`
- `process`
- `message_thread`
- `logger`

5.75.1 Detailed Description

Class JCM - executed on JCM

5.75.2 Member Function Documentation

5.75.2.1 is_stopped()

```
def jcm_control_server.JCM.is_stopped (  
    self )
```

Check if self.process is stopped or still running

5.75.2.2 read_messages()

```
def jcm_control_server.JCM.read_messages (  
    self )
```

Read data from message_queue and return as msg

5.75.2.3 rename_readback_file()

```
def jcm_control_server.JCM.rename_readback_file (
    self,
    file_name = None,
    restore = False )
```

Renames the readback file on the JCM device
if restore is False, it renames the standard readBack.data to file_name in the /tmp folder,
if restore is True, it renames the file_name in /tmp to the standard readBack.data file

5.75.2.4 start_blind_scrubber()

```
def jcm_control_server.JCM.start_blind_scrubber (
    self )
```

Start blind scrubbing

5.75.2.5 start_calibrate_device()

```
def jcm_control_server.JCM.start_calibrate_device (
    self )
```

Start JTAG calibration

5.75.2.6 start_full_configure()

```
def jcm_control_server.JCM.start_full_configure (
    self,
    bitfile = None )
```

Start full FPGA configuration

5.75.2.7 start_random_fault_injection()

```
def jcm_control_server.JCM.start_random_fault_injection (
    self,
    delay,
    n_faults,
    delay_after,
    correction )
```

Start random fault injection of n_faults,

* @param delay The delay in milliseconds between injection and repair.

* @param delay_after The delay in milliseconds between repairs and next injection (or between to injections if

5.75.2.8 start_readback()

```
def jcm_control_server.JCM.start_readback (
    self )
```

Readback configuration of FPGA

5.75.2.9 stop_blind_scrubber()

```
def jcm_control_server.JCM.stop_blind_scrubber (
    self )
```

Send stop signal to jcm_blind_scrubber

5.75.2.10 stop_random_fault_injection()

```
def jcm_control_server.JCM.stop_random_fault_injection (
    self )
```

Send stop signal to jcm_random_fault_injection

5.75.2.11 test_function()

```
def jcm_control_server.JCM.test_function (
    self )
```

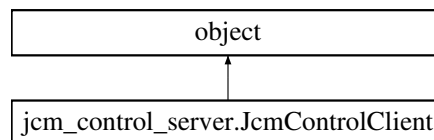
Test functions

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/jcm/jcm_control_server.py

5.76 jcm_control_server.JcmControlClient Class Reference

Inheritance diagram for jcm_control_server.JcmControlClient:



Public Member Functions

- def **__init__** (self, address, port)
- def [start_calibrate_device](#) (self)
- def [start_full_configure](#) (self, bitfile=None)
- def [is_stopped](#) (self)
- def [read_calibrate_device_results](#) (self)
- def [read_full_configure_results](#) (self)
- def [start_readback](#) (self)
- def [is_readback_finished](#) (self)
- def [rename_readback_file](#) (self, file_name=None, restore=False)
- def [remove_readback_files](#) (self)
- def [compare_readback](#) (self, basename1, basename2)
- def [start_blind_scrubber](#) (self)
- def [read_start_blind_scrubber_results](#) (self)
- def [stop_blind_scrubber](#) (self)
- def [start_random_fault_injection](#) (self, delay, n_faults, delay_after=0, correction=True)
- def [stop_random_fault_injection](#) (self)
- def [read_fault_injection_results](#) (self)
- def [read_messages](#) (self)
- def [test_function](#) (self)

Public Attributes

- **logger**

5.76.1 Detailed Description

Remote JCM client, running on a beam test notebook

5.76.2 Member Function Documentation

5.76.2.1 `is_readback_finished()`

```
def jcm_control_server.JcmControlClient.is_readback_finished (
    self )
```

Read readback results

5.76.2.2 `is_stopped()`

```
def jcm_control_server.JcmControlClient.is_stopped (
    self )
```

Check if self.process is stopped or still running

5.76.2.3 `read_calibrate_device_results()`

```
def jcm_control_server.JcmControlClient.read_calibrate_device_results (
    self )
```

Read calibrate device results

5.76.2.4 `read_fault_injection_results()`

```
def jcm_control_server.JcmControlClient.read_fault_injection_results (
    self )
```

Read All messages from fault injection
return tuple (successful_injected_faults,messages)

5.76.2.5 read_full_configure_results()

```
def jcm_control_server.JcmControlClient.read_full_configure_results (
    self )
```

Read full configure results

5.76.2.6 read_messages()

```
def jcm_control_server.JcmControlClient.read_messages (
    self )
```

Read messages

5.76.2.7 read_start_blind_scrubber_results()

```
def jcm_control_server.JcmControlClient.read_start_blind_scrubber_results (
    self )
```

Read All messages from start_blind_scrubber.
return tuple (success, messages)

5.76.2.8 start_blind_scrubber()

```
def jcm_control_server.JcmControlClient.start_blind_scrubber (
    self )
```

Start blind scrubber

5.76.2.9 start_calibrate_device()

```
def jcm_control_server.JcmControlClient.start_calibrate_device (
    self )
```

Start device calibration

5.76.2.10 start_full_configure()

```
def jcm_control_server.JcmControlClient.start_full_configure (
    self,
    bitfile = None )
```

Start full configuration

5.76.2.11 start_random_fault_injection()

```
def jcm_control_server.JcmControlClient.start_random_fault_injection (
    self,
    delay,
    n_faults,
    delay_after = 0,
    correction = True )
```

Start random fault injection of n_faults

5.76.2.12 start_readback()

```
def jcm_control_server.JcmControlClient.start_readback (
    self )
```

Start readback of configuration

5.76.2.13 stop_blind_scrubber()

```
def jcm_control_server.JcmControlClient.stop_blind_scrubber (
    self )
```

Send stop signal to jcm_blind_nscrubber

5.76.2.14 stop_random_fault_injection()

```
def jcm_control_server.JcmControlClient.stop_random_fault_injection (
    self )
```

Send stop signal to jcm_random_fault_injection

5.76.2.15 test_function()

```
def jcm_control_server.JcmControlClient.test_function (
    self )
```

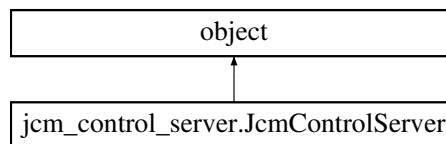
Test function

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/jcm/jcm_control_server.py

5.77 jcm_control_server.JcmControlServer Class Reference

Inheritance diagram for jcm_control_server.JcmControlServer:



Public Member Functions

- def [__init__](#) (self, address, port)
- def **start** (self)
- def **close** (self)

5.77.1 Detailed Description

RPC server, running on JCM

5.77.2 Constructor & Destructor Documentation

5.77.2.1 __init__()

```
def jcm_control_server.JcmControlServer.__init__ (
    self,
    address,
    port )
```

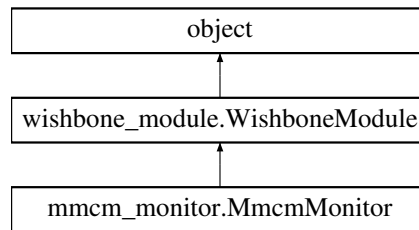
Initialize server with given interface address and port

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/jcm/jcm_control_server.py

5.78 mmcm_monitor.MmcmMonitor Class Reference

Inheritance diagram for mmcm_monitor.MmcmMonitor:



Public Member Functions

- def `__init__` (self, moduleid, board_obj, index)
- def `get_counters` (self, verbose=True)
- def `get_lock_counter` (self, verbose=False)
- def `get_mismatch` (self, verbose=False)
- def `dump_config` (self)

Public Attributes

- **index**

Static Public Attributes

- int **LOCK_COUNTER** = 7

5.78.1 Detailed Description

wishbone slave instantiated inside the wishbone mmcm monitor handling the DP1, DP0 errors

5.78.2 Constructor & Destructor Documentation

5.78.2.1 `__init__()`

```
def mmcm_monitor.MmcmMonitor.__init__ (
    self,
    moduleid,
    board_obj,
    index )
```

init

Reimplemented from [wishbone_module.WishboneModule](#).

5.78.3 Member Function Documentation

5.78.3.1 dump_config()

```
def mmcm_monitor.MmcmMonitor.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.78.3.2 get_counters()

```
def mmcm_monitor.MmcmMonitor.get_counters (
    self,
    verbose = True )
```

gets the values of the free running counters

5.78.3.3 get_lock_counter()

```
def mmcm_monitor.MmcmMonitor.get_lock_counter (
    self,
    verbose = False )
```

gets the values of the lock counter

5.78.3.4 get_mismatch()

```
def mmcm_monitor.MmcmMonitor.get_mismatch (
    self,
    verbose = False )
```

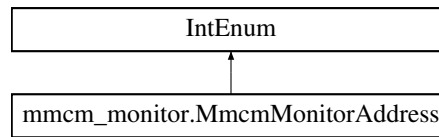
gets the actual mismatch status

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/mmcm_monitor.py

5.79 mmcm_monitor.MmcmMonitorAddress Class Reference

Inheritance diagram for mmcm_monitor.MmcmMonitorAddress:



Static Public Attributes

- int **GET_LOCK_COUNTER** = 0
- int **MISMATCH_LSB** = 1
- int **MISMATCH_MSB** = 2
- int **MISMATCH_2ND_LSB** = 3
- int **MISMATCH_2ND_MSB** = 4
- int **MISMATCH_COUNTER** = 5
- int **MISMATCH_2ND_COUNTER** = 6
- int **LATCH_COUNTERS** = 7
- int **RESET_COUNTERS** = 8
- int **GET_COUNTER_CLK_0_FIRST** = RESET_COUNTERS + 1
- int **GET_COUNTER_CLK_0_LAST** = GET_COUNTER_CLK_0_FIRST + 3
- int **GET_COUNTER_CLK_N_FIRST** = GET_COUNTER_CLK_0_FIRST + 3*6
- int **GET_COUNTER_CLK_N_LAST** = GET_COUNTER_CLK_0_FIRST + 3*7 -1

5.79.1 Detailed Description

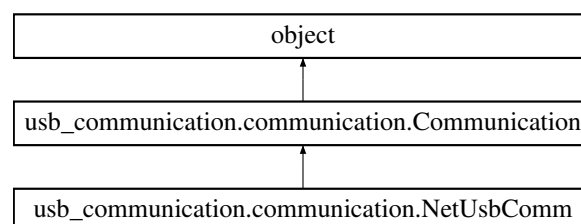
memory mapping for the mmcm_monitor got from mmcm_monitor_pkg.vhd

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/mmcm_monitor.py

5.80 usb_communication.communication.NetUsbComm Class Reference

Inheritance diagram for usb_communication.communication.NetUsbComm:



Public Member Functions

- `def __init__ (self, ctlOnly=False, HOSTNAME='127.0.0.1', PORT_CONTROL=30000, PORT_DATA0=30001, PORT_DATA1=30002, Timeout=None, enable\_rderr\_exception=False)`
- `def stop (self)`
- `def set\_server (self, server)`
- `def open\_connection (self, ctlOnly=False)`
- `def close\_connections (self)`

Public Attributes

- **`socketControl`**
- **`timeout`**
- **`hostname`**
- **`port_control`**
- **`port_data0`**
- **`port_data1`**
- **`ctlOnly`**
- **`write_retries`**
- **`server`**

Static Public Attributes

- **`socketData0`**
- **`socketData1`**

5.80.1 Detailed Description

Network usb communication class.

This class implements the Communication class by implementing a network communication with the UsbComm C++ program.

5.80.2 Member Function Documentation

5.80.2.1 `close_connections()`

```
def usb_communication.communication.NetUsbComm.close_connections (
    self )
```

Close all open sockets

5.80.2.2 stop()

```
def usb_communication.communication.NetUsbComm.stop (  
    self )
```

Perform any operation on close

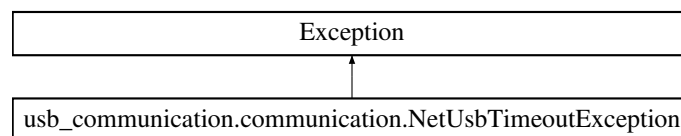
Reimplemented from [usb_communication.communication.Communication](#).

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py

5.81 usb_communication.communication.NetUsbTimeoutException Class Reference

Inheritance diagram for usb_communication.communication.NetUsbTimeoutException:



Public Member Functions

- def **__init__** (self, signum=0)
- def **__signum__** (self)

Public Attributes

- **signum**

5.81.1 Detailed Description

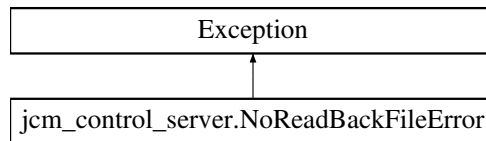
basic class to handle netusbcomm timeout exception

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py

5.82 jcm_control_server.NoReadBackFileError Class Reference

Inheritance diagram for jcm_control_server.NoReadBackFileError:



Public Member Functions

- `def __init__(self, message)`
- `def __srt__(self)`
- `def log\_info(self)`

Public Attributes

- `logger`
- `message`

5.82.1 Detailed Description

basic class to define the lack of a readback file in the JCM error exception

5.82.2 Member Function Documentation

5.82.2.1 `log_info()`

```
def jcm_control_server.NoReadBackFileError.log_info (  
    self )
```

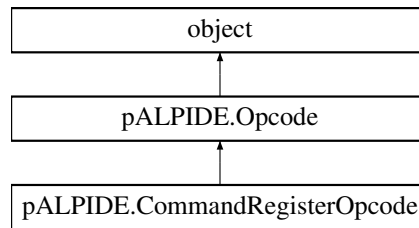
returns the args of the exception

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/software/py/jcm/jcm_control_server.py`

5.83 pALPIDE.Opcode Class Reference

Inheritance diagram for pALPIDE.Opcode:



Static Public Attributes

- int **GRST** = 0x00D2
- int **PRST** = 0x00E4
- int **PULSE** = 0x0078
- int **BCRST** = 0x0036
- int **RORST** = 0x0063
- int **DEBUG** = 0x00AA
- int **TRIGGER** = 0x00B1
- int **TRIGGER1** = 0x0055
- int **TRIGGER2** = 0x00C9
- int **TRIGGER3** = 0x002D
- int **WROP** = 0x009C
- int **RDOP** = 0x004E

5.83.1 Detailed Description

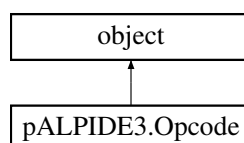
Opcodes supported by ALPIDE

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/pALPIDE.py

5.84 pALPIDE3.Opcode Class Reference

Inheritance diagram for pALPIDE3.Opcode:



Static Public Attributes

- int **GRST** = 0x00D2
- int **PRST** = 0x00E4
- int **PULSE** = 0x0078
- int **BCRST** = 0x0036
- int **RORST** = 0x0063
- int **DEBUG** = 0x00AA
- int **TRIGGER** = 0x00B1
- int **TRIGGER1** = 0x0055
- int **TRIGGER2** = 0x00C9
- int **TRIGGER3** = 0x002D
- int **WROP** = 0x009C
- int **RDOP** = 0x004E
- int **CMUCLRERR** = 0xFF00
- int **FIFOTEST** = 0xFF01
- int **LOADOBDEFCFG** = 0xFF02
- int **XOFF** = 0xFF10
- int **XON** = 0xFF11
- int **ADCMEASURE** = 0xFF20

5.84.1 Detailed Description

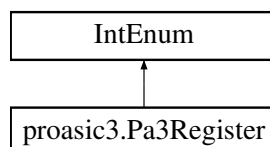
Opcodes supported by ALPIDE

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/pALPIDE3.py

5.85 proasic3.Pa3Register Class Reference

Inheritance diagram for proasic3.Pa3Register:



Static Public Attributes

- int **MINOR_VERSION** = 0x0
- int **MAJOR_VERSION** = 0x1
- int **CLOCK_STATUS** = 0x3
- int **CLOCK_CONFIG** = 0x4
- int **CC_CMD** = 0x8
- int **CC_STATUS** = 0x9
- int **CC_ACTIVE_STATE** = 0xA
- int **CC_MINOR_SCRUB_CNT** = 0xB
- int **CC_MAJOR_SCRUB_CNT** = 0xC
- int **CC_CTRL** = 0xD
- int **RC_CMD** = 0x10
- int **RC_STATUS** = 0x11
- int **RC_FLASHPAGE1** = 0x12
- int **RC_FLASHPAGE2** = 0x13
- int **RC_FLASHPAGE3** = 0x14
- int **SMAP_CMD** = 0x18
- int **SMAP_DATA_TX** = 0x19
- int **SMAP_DATA_RX** = 0x1A
- int **SMAP_STATUS** = 0x1B
- int **FLASH_CMD** = 0x20
- int **FLASH_STATUS** = 0x21
- int **FLASH_PATTERN** = 0x22
- int **FLASH_MIS_COUNT** = 0x23
- int **FLASH_PAGE_ADDR** = 0x24
- int **FLASH_MINOR_BLOCK_ADDR** = 0x25
- int **FLASH_MAJOR_BLOCK_ADDR** = 0x26
- int **FLASH_TRX_SIZE_MINOR** = 0x27
- int **FLASH_TRX_SIZE_MAJOR** = 0x28
- int **FLASH_STATUS_WORD** = 0x29
- int **FLASH_SELECT_IC** = 0x2A
- int **FLASH_MAJOR_TRX_CNT** = 0x2B
- int **FLASH_MINOR_TRX_CNT** = 0x2C
- int **FLASH_STATE_0** = 0x46
- int **FLASH_STATE_1** = 0x47
- int **FLASH_STATE_2** = 0x48
- int **FLASH_STATE_3** = 0x4D
- int **FLASH_ST_WRD** = 0x4E
- int **FIFO_RX_DATA** = 0x30
- int **FIFO_TX_DATA** = 0x31
- int **FIFO_STATUS** = 0x32
- int **FIFO_WRITER_CMD** = 0x33
- int **FIFO_WRITER_STATUS** = 0x34
- int **ECC_CMD** = 0x38
- int **ECC_STATUS** = 0x39
- int **ECC_SB_ERROR_CNT** = 0x3A
- int **ECC_FIFO_TMR_ERROR** = 0x3B
- int **DIPSWITCH1** = 0x40
- int **DIPSWITCH2** = 0x41
- int **PUSHBUTTON** = 0x42
- int **CTRL_LEDS** = 0x43
- int **CRC_0** = 0x49
- int **CRC_1** = 0x4A
- int **CRC_2** = 0x4B

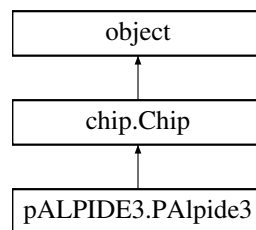
- int **CRC_3** = 0x4C

The documentation for this class was generated from the following file:

- /media/sf_proj/RUV1_Test/modules/board_support_software/software/py/proasic3.py

5.86 pALPIDE3.PAlpide3 Class Reference

Inheritance diagram for pALPIDE3.PAlpide3:



Public Member Functions

- def **__init__** (self, board, chipid, readback=False, log=False, commitTransaction=True)
- def **reset** (self)
- def **initialize** (self, disable_manchester=1, grst=True)
- def **configure_dtu** (self, PLLDAC=None, DriverDAC=None, PreDAC=None, PLLDelayStages=None, PllBandwidthControl=None, LockWaitCycles=None, UnlockWaitCycles=None, commitTransaction=None, log=None, readback=None)
- def **initialize_readout** (self, PLLDAC=None, DriverDAC=None, PreDAC=None, PLLDelayStages=None, PllBandwidthControl=None, commitTransaction=None, log=None, readback=None)
- def **reset_pll** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **pll_check_lock** (self)
- def **trigger** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **propagate_comma** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **propagate_data** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **propagate_prbs** (self, commitTransaction=None, readback=None, log=None, verbose=False)
- def **propagate_clock** (self, acommitTransaction=None, readback=None, log=None, verbose=False)
- def **set_xoff** (self, XOff=1, commitTransaction=None, readback=None, log=None, verbose=False)
- def **dump_config** (self)
- def **init_pixel_matrix** (self, readback=None, log=None, commitTransaction=None)
- *MATRIX #.*
- def **set_default_variables** (self, commitTransaction, log, readback=None)
- def **setreg_cmd** (self, Command=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def **getreg_cmd** (self, commitTransaction=None, log=None, verbose=False)
- def **setreg_periphery_ctrl** (self, ChipModeSelector=None, ClusteringEnabled=None, StartMemSelfTest=None, MatrixROSpeed=None, ForceBusy=None, ForceBusyValue=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def **getreg_periphery_ctrl** (self, commitTransaction=None, log=None, verbose=False)
- def **setreg_disable_regions_lsbs** (self, RegionDisable=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def **getreg_disable_regions_lsbs** (self, commitTransaction=None, log=None, verbose=False)

- def [setreg_disableof_regions_msbs](#) (self, RegionDisable=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_disableof_regions_msbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_cfg_1](#) (self, MEBMask=None, EnInternalGeneration=None, EnBusyMonitoring=None, TPulseMode=None, EnTestStrobe=None, SMState=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_cfg_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_cfg_2](#) (self, FrameDuration=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_cfg_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_pulsing1](#) (self, TPulseDelay=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_pulsing1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fromu_pulsing_2](#) (self, TPulseDuration=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fromu_pulsing_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fromu_status_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fromu_status_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dac_settings_dclk_and_mclk_io_buffers](#) (self, DCLKReceiver=None, DCLKDriver=None, MCLKReceiver=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dac_settings_dclk_and_mclk_io_buffers](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dac_settings_cmu_io_buffers](#) (self, DCTRLReceiver=None, DCTRLDriver=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dac_settings_cmu_io_buffers](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_cmu_and_dmu_cfg](#) (self, PreviousChipID=None, DMUXOff=None, InitialToken=None, DisableManchester=None, EnableDDR=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_cmu_and_dmu_cfg](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_cmu_errors_counter](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_cfg](#) (self, VcoDelayStages=None, PIIBandwidthControl=None, PILOffSignal=None, SerPhase=None, PLLReset=None, LoadENStatus=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_cfg](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_dacs](#) (self, PLLDAC=None, DriverDAC=None, PreDAC=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_dacs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_dtu_pll_lock_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_pll_lock_2](#) (self, LockWaitCycles=None, UnlockWaitCycles=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_pll_lock_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_test_1](#) (self, TestEN=None, IntPatternEN=None, PrbsEN=None, Bypass8b10b=None, BDIN8b10b0=None, BDIN8b10b1=None, BDIN8b10b2=None, K0=None, K1=None, K2=None, ForceSetLoadEN=None, ForceUnsetLoadEN=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_test_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_test_2](#) (self, DIN0=None, DIN1=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_test_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_dtu_test_3](#) (self, DIN2=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_dtu_test_3](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_busy_min_width](#) (self, BusyMinWidth=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_busy_min_width](#) (self, commitTransaction=None, log=None, verbose=False)

- def [setreg_fuses_write_lsbs](#) (self, FusesWriteLSBs=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fuses_write_lsbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_fuses_write_msbs](#) (self, FusesWriteMSBs=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_fuses_write_msbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fuses_read_lsbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_fuses_read_msbs](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_temperature_sensor](#) (self, commitTransaction=None, log=None, verbose=False)
- def [getreg_dmu_and_tru_state](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_pixel_cfg_1](#) (self, RowSel=None, SetsAllRows=None, PIXCNFG_REGSEL=None, PIXCNFG_DATA=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_pixel_cfg_1](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_pixel_cfg_2](#) (self, ColSel=None, SetsAllColumns=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_pixel_cfg_2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_pixel_cfg_3](#) (self, MatrixLatchEn=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_pixel_cfg_3](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_analog_monitor_and_override](#) (self, VoltageDACSel=None, CurrentDACSel=None, SWCNTL_DACMONI=None, SWCNTL_DACMONV=None, IRefBufferCurrent=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_analog_monitor_and_override](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VRESETP](#) (self, VRESETP=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VRESETP](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VRESETD](#) (self, VRESETD=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VRESETD](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCASP](#) (self, VCASP=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VCASP](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCASN](#) (self, VCASN=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VCASN](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VPULSEH](#) (self, VPULSEH=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VPULSEH](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VPULSEL](#) (self, VPULSEL=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VPULSEL](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCASN2](#) (self, VCASN2=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VCASN2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VCLIP](#) (self, VCLIP=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VCLIP](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_VTEMP](#) (self, VTEMP=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_VTEMP](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_IAUX2](#) (self, IAUX2=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_IAUX2](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_IRESET](#) (self, IRESET=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_IRESET](#) (self, commitTransaction=None, log=None, verbose=False)
- def [setreg_IDB](#) (self, IDB=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def [getreg_IDB](#) (self, commitTransaction=None, log=None, verbose=False)

- def `setreg_IBIAS` (self, IBIAS=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_IBIAS` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_ITHR` (self, ITHR=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_ITHR` (self, commitTransaction=None, log=None, verbose=False)
- def `setreg_buffer_bypass` (self, VCASNBYPASS=None, VCASN2BYPASS=None, VCASPBYPASS=None, VCLIPBYPASS=None, IRESETBYPASS=None, IBIASBYPASS=None, ITHRBYPASS=None, IDBBYPASS=None, SS=None, commitTransaction=None, log=None, readback=None, verbose=False)
- def `getreg_buffer_bypass` (self, commitTransaction=None, log=None, verbose=False)

Public Attributes

- `modulename`
- `logger`
- `PLLLockCounter`

5.86.1 Detailed Description

pALPIDE3 chip

agreed basic function structure

```
setreg_<REGNAME>(self, <list of parameters=ManualDefaults>,
                 commitTransaction=None, log=None,
                 readback=None, verbose=False):
    \"\"\"DOCSTRING\"\"\"
    <Input Assertions>
    if commitTransaction == None:
        commitTransaction = self.commitTransaction
    if log == None:
        log = self.log
    if readback == None:
        readback = self.readback

    dataawrite = <parameter assignment>
    self.write_reg(address=address,
                   data=dataawrite,
                   readback=readback,
                   log=log,
                   commitTransaction=commitTransaction,
                   verbose=verbose)

getreg_<REGNAME>(self, commitTransaction=None, log=None, verbose=False):
    \"\"\"DOCSTRING\"\"\"
    if commitTransaction == None:
        commitTransaction = self.commitTransaction
    if log == None:
        log = self.log

    dataread = self.read_reg(address=address,
                             log=log,
                             commitTransaction=commitTransaction,
                             verbose=verbose)

    <dataread unwrap in dictionary>
    ret = {"ret": dataread, ...:...}
```

5.86.2 Member Function Documentation

5.86.2.1 dump_config()

```
def pALPIDE3.PAlpide3.dump_config (
    self )
```

Dump chip configuration

Reimplemented from [chip.Chip](#).

5.86.2.2 getreg_analog_monitor_and_override()

```
def pALPIDE3.PAlpide3.getreg_analog_monitor_and_override (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for Analog Monitor and Override Register
Register fields

```
=====
VoltageDACSel          [3:0] : 0x0
CurrentDACSel          [6:4] : 0x0
SWCNTL_DACMONI         [7] : 0x0
SWCNTL_DACMONV         [8] : 0x0
IRefBufferCurrent      [10:9] : 0x2
```

Orgininal docstring

```
=====
```

Analog Monitor and Override Register

DACMONI current selection:

```
0  IRESET
1  IAUX2
2  IBIAS
3  IDB
4  IREF
5  ITHR
6  IREFBuffer
```

DACMONV voltage selection:

```
0  VCASN
1  VCASP
2  VPULSEH
3  VPULSEL
4  VRESETP
5  VRESETD
6  VCASN2
7  VCLIP
8  VTEMP
```

IRefBuffer current setting:

```
0  0,25uA
1  0,75uA
2  1,00uA (default)
3  1,25uA
```

This is an autogenerated function. Do not modify directly.

5.86.2.3 getreg_buffer_bypass()

```
def pALPIDE3.PAlpide3.getreg_buffer_bypass (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Buffer Bypass Register
Register fields
=====
VCASNBYPASS          [0] : 0x0
VCASN2BYPASS         [1] : 0x0
VCASPBYPASS          [2] : 0x0
VCLIPBYPASS          [3] : 0x0
IRESETBYPASS         [4] : 0x0
IBIASBYPASS          [5] : 0x0
ITHRBYPASS           [6] : 0x0
IDBBYPASS            [7] : 0x0

Original docstring
=====
Buffer Bypass Register
```

This is an autogenerated function. Do not modify directly.

5.86.2.4 getreg_busy_min_width()

```
def pALPIDE3.PAlpide3.getreg_busy_min_width (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for BUSY min width
Register fields
=====
BusyMinWidth          [4:0] : 0x8

Original docstring
=====
BUSY min width
```

This is an autogenerated function. Do not modify directly.

5.86.2.5 getreg_cmd()

```
def pALPIDE3.PAlpide3.getreg_cmd (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for Command Register.

Register fields

=====

Command [15:0] : 0x0

Original docstring

=====

Command Register.

Valid opcodes:

16'hFF00	CMUCLRERR
16'h00D2	GRST
16'h00E4	PRST
16'h0078	PULSE
16'h0036	BCRST
16'h0063	RORST
16'h00B1,0055,00C9,002D	TRIGGER
16'h009C	WROP
16'h004E	RDOP

This is an autogenerated function. Do not modify directly.

5.86.2.6 getreg_cmu_and_dmu_cfg()

```
def pALPIDE3.PAlpide3.getreg_cmu_and_dmu_cfg (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for CMU and DMU Configuration Register

Register fields

=====

PreviousChipID	[3:0] : 0xf
DMUXOff	[4] : 0x0
InitialToken	[5] : 0x0
DisableManchester	[6] : 0x0
EnabledDDR	[7] : 0x1

Original docstring

=====

CMU and DMU Configuration Register

This is an autogenerated function. Do not modify directly.

5.86.2.7 getreg_cmu_errors_counter()

```
def pALPIDE3.PAlpide3.getreg_cmu_errors_counter (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for CMU Errors Counter
 Register fields
 =====

CMUErrorsCounter [11:0] : 0x0

Original docstring
 =====
 CMU Errors Counter

This is an autogenerated function. Do not modify directly.

5.86.2.8 getreg_dac_settings_cmu_io_buffers()

```
def pALPIDE3.PAlpide3.getreg_dac_settings_cmu_io_buffers (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DAC settings for CMU I/O buffers
 Register fields
 =====

DCTRLReceiver [3:0] : 0xa
 DCTRLDriver [7:4] : 0xa

Original docstring
 =====
 DAC settings for CMU I/O buffers

This is an autogenerated function. Do not modify directly.

5.86.2.9 getreg_dac_settings_dclk_and_mclk_io_buffers()

```
def pALPIDE3.PAlpide3.getreg_dac_settings_dclk_and_mclk_io_buffers (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DAC settings for DCLK and MCLK I/O buffers
 Register fields
 =====

DCLKReceiver [3:0] : 0xa
 DCLKDriver [7:4] : 0xa
 MCLKReceiver [11:8] : 0xa

Original docstring
 =====
 DAC settings for DCLK and MCLK I/O buffers

This is an autogenerated function. Do not modify directly.

5.86.2.10 getreg_disable_regions_lsbs()

```
def pALPIDE3.PAlpide3.getreg_disable_regions_lsbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Disable of regions 0-15
Register fields
=====
    RegionDisable          [15:0] :    0x0

Origininal docstring
=====
Disable of regions 0-15

This is an autogenerated function. Do not modify directly.
```

5.86.2.11 getreg_disableof_regions_msbs()

```
def pALPIDE3.PAlpide3.getreg_disableof_regions_msbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Disableof regions 16-31
Register fields
=====
    RegionDisable          [15:0] :    0x0

Origininal docstring
=====
Disableof regions 16-31

This is an autogenerated function. Do not modify directly.
```

5.86.2.12 getreg_dmu_and_tru_state()

```
def pALPIDE3.PAlpide3.getreg_dmu_and_tru_state (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DMU and TRU State
Register fields

=====

TRUSMState	[2:0] :	0x0
DMUSMState	[5:3] :	0x0
BusyMismatchError	[6] :	0x0
BusyFifoFullError	[7] :	0x0

Original docstring

=====

DMU and TRU State

Reset on GRST or RORST

Reset on GRST or RORST

This is an autogenerated function. Do not modify directly.

5.86.2.13 getreg_dtu_cfg()

```
def pALPIDE3.PAlpide3.getreg_dtu_cfg (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU Configuration Register

Register fields

=====

VcoDelayStages	[1:0] :	0x1
PllBandwidthControl	[2] :	0x1
PllOffSignal	[3] :	0x1
SerPhase	[7:4] :	0x8
PLLReset	[8] :	0x0
LoadENStatus	[12] :	0x0

Original docstring

=====

DTU Configuration Register

LoadENStatus is read-only

This is an autogenerated function. Do not modify directly.

5.86.2.14 getreg_dtu_dacs()

```
def pALPIDE3.PAlpide3.getreg_dtu_dacs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU DACs Register

Register fields

=====

PLLDAC	[3:0] :	0x8
DriverDAC	[7:4] :	0x8
PreDAC	[11:8] :	0x0

Original docstring

=====

DTU DACs Register

This is an autogenerated function. Do not modify directly.

5.86.2.15 getreg_dtu_pll_lock_1()

```
def pALPIDE3.PAlpide3.getreg_dtu_pll_lock_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU PLL Lock Register 1

Register fields

=====

LockCounter	[7:0] :	0x0
LockFlag	[8] :	0x0
LockStatus	[9] :	0x0

Original docstring

=====

DTU PLL Lock Register 1

This is an autogenerated function. Do not modify directly.

5.86.2.16 getreg_dtu_pll_lock_2()

```
def pALPIDE3.PAlpide3.getreg_dtu_pll_lock_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU PLL Lock Register 2

Register fields

=====

LockWaitCycles	[7:0] :	0x0
UnlockWaitCycles	[15:8] :	0x0

Original docstring

=====

DTU PLL Lock Register 2

This is an autogenerated function. Do not modify directly.

5.86.2.17 getreg_dtu_test_1()

```
def pALPIDE3.PAlpide3.getreg_dtu_test_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for DTU Test Register 1
Register fields
=====
TestEN                [0] : 0x0
IntPatternEN          [1] : 0x0
PrbsEN                [2] : 0x0
Bypass8b10b          [3] : 0x0
BDIN8b10b0            [5:4] : 0x0
BDIN8b10b1            [7:6] : 0x0
BDIN8b10b2            [9:8] : 0x0
K0                    [10] : 0x0
K1                    [11] : 0x0
K2                    [12] : 0x0
ForceSetLoadEN        [13] : 0x0
ForceUnsetLoadEN      [14] : 0x0

Original docstring
=====
DTU Test Register 1

This is an autogenerated function. Do not modify directly.
```

5.86.2.18 getreg_dtu_test_2()

```
def pALPIDE3.PAlpide3.getreg_dtu_test_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for DTU Test Register 2
Register fields
=====
DIN0                  [7:0] : 0x0
DIN1                  [15:8] : 0x0

Original docstring
=====
DTU Test Register 2

This is an autogenerated function. Do not modify directly.
```

5.86.2.19 getreg_dtu_test_3()

```
def pALPIDE3.PAlpide3.getreg_dtu_test_3 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 3
Register fields

=====

DIN2	[7:0] :	0x0
------	---------	-----

Original docstring

=====

DTU Test Register 3

This is an autogenerated function. Do not modify directly.

5.86.2.20 getreg_fromu_cfg_1()

```
def pALPIDE3.PAlpide3.getreg_fromu_cfg_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 1
Register fields

=====

MEBMask	[2:0] :	0x0
EnInternalGeneration	[3] :	0x0
EnBusyMonitoring	[4] :	0x1
TPulseMode	[5] :	0x0
EnTestStrobe	[6] :	0x0
SMState	[12:8] :	0x0

Original docstring

=====

FROMU Configuration Register 1

EnBusyMonitoring default = 1

SMState is Read only

This is an autogenerated function. Do not modify directly.

5.86.2.21 getreg_fromu_cfg_2()

```
def pALPIDE3.PAlpide3.getreg_fromu_cfg_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 2
Register fields

=====

FrameDuration [15:0] : 0x0

Original docstring

=====

FROMU Configuration Register 2

This is an autogenerated function. Do not modify directly.

5.86.2.22 getreg_fromu_pulsing1()

```
def pALPIDE3.PAlpide3.getreg_fromu_pulsing1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for FROMU Pulsing Register1
Register fields

=====

TPulseDelay [15:0] : 0x0

Original docstring

=====

FROMU Pulsing Register1

This is an autogenerated function. Do not modify directly.

5.86.2.23 getreg_fromu_pulsing_2()

```
def pALPIDE3.PAlpide3.getreg_fromu_pulsing_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for FROMU Pulsing Register 2
Register fields

=====

TPulseDuration [15:0] : 0x0

Original docstring

=====

FROMU Pulsing Register 2

This is an autogenerated function. Do not modify directly.

5.86.2.24 getreg_fromu_status_1()

```
def pALPIDE3.PAlpide3.getreg_fromu_status_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Status Register 1
Register fields
=====
    StrobeCounter          [15:0] :    0x0

Original docstring
=====
FROMU Status Register 1

This is an autogenerated function. Do not modify directly.
```

5.86.2.25 getreg_fromu_status_2()

```
def pALPIDE3.PAlpide3.getreg_fromu_status_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for FROMU Status register 2
Register fields
=====
    BunchCounter           [11:0] :    0x0
    EventCounter            [13:12] :    0x0

Original docstring
=====
FROMU Status register 2

This is an autogenerated function. Do not modify directly.
```

5.86.2.26 getreg_fuses_read_lsbs()

```
def pALPIDE3.PAlpide3.getreg_fuses_read_lsbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Fuses Read LSBs
Register fields
=====
    FusesReadLSBs          [15:0] :    0x0

Original docstring
=====
Fuses Read LSBs

This is an autogenerated function. Do not modify directly.
```

5.86.2.27 getreg_fuses_read_msbs()

```
def pALPIDE3.PAlpide3.getreg_fuses_read_msbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Fuses Read MSBs
Register fields
=====
    FusesReadMSBs          [7:0] :    0x0

Origininal docstring
=====
Fuses Read MSBs

This is an autogenerated function. Do not modify directly.
```

5.86.2.28 getreg_fuses_write_lsbs()

```
def pALPIDE3.PAlpide3.getreg_fuses_write_lsbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Fuses Write LSBs
Register fields
=====
    FusesWriteLSBs         [15:0] :    0x0

Origininal docstring
=====
Fuses Write LSBs

This is an autogenerated function. Do not modify directly.
```

5.86.2.29 getreg_fuses_write_msbs()

```
def pALPIDE3.PAlpide3.getreg_fuses_write_msbs (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Fuses Write MSBs
Register fields
=====
    FusesWriteMSBs         [7:0] :    0x0

Origininal docstring
=====
Fuses Write MSBs

This is an autogenerated function. Do not modify directly.
```

5.86.2.30 getreg_IAUX2()

```
def pALPIDE3.PAlpide3.getreg_IAUX2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    IAUX2                [7:0] :   0x0

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.31 getreg_IBIAS()

```
def pALPIDE3.PAlpide3.getreg_IBIAS (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    IBIAS                [7:0] :  0x40

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.32 getreg_IDB()

```
def pALPIDE3.PAlpide3.getreg_IDB (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    IDB                  [7:0] :  0x40

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.33 getreg_IRESET()

```
def pALPIDE3.PAlpide3.getreg_IRESET (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
    IRESET                                [7:0] : 0x32
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.34 getreg_ITHR()

```
def pALPIDE3.PAlpide3.getreg_ITHR (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
    ITHR                                7:0 : 0x33
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.35 getreg_periphery_ctrl()

```
def pALPIDE3.PAlpide3.getreg_periphery_ctrl (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for Periphery Control register
Register fields

=====

ChipModeSelector	[1:0] :	0x0
ClusteringEnabled	[2] :	0x0
StartMemSelfTest	[3] :	0x0
MatrixROSpeed	[5] :	0x1
ForceBusy	[6] :	0x0
ForceBusyValue	[7] :	0x0

Origininal docstring

=====

Periphery Control register

Chip Mode Selector values:

0 Cfg Mode

1 Readout Triggered Mode

2 Readout Continuous Mode

MatrixROSpeed 0=10MHz, 1=20MHz (default)

This is an autogenerated function. Do not modify directly.

5.86.2.36 getreg_pixel_cfg_1()

```
def pALPIDE3.PAlpide3.getreg_pixel_cfg_1 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

Autogenerated function for Pixel CFG Register 1

Register fields

=====

RowSel	[8:0] :	0x0
SetsAllRows	[9] :	0x0
PIXCNFG_REGSEL	[10] :	0x0
PIXCNFG_DATA	[11] :	0x0

Origininal docstring

=====

Pixel CFG Register 1

This is an autogenerated function. Do not modify directly.

5.86.2.37 getreg_pixel_cfg_2()

```
def pALPIDE3.PAlpide3.getreg_pixel_cfg_2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )
```

```

Autogenerated function for Pixel CFG Register 2
Register fields
=====
    ColSel          [9:0] :    0x0
    SetsAllColumns  [10] :    0x0

Original docstring
=====
Pixel CFG Register 2

This is an autogenerated function. Do not modify directly.

```

5.86.2.38 getreg_pixel_cfg_3()

```

def pALPIDE3.PAlpide3.getreg_pixel_cfg_3 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Pixel CFG Register 3
Register fields
=====
    MatrixLatchEn    [0] :    0x0

Original docstring
=====
Pixel CFG Register 3

This is an autogenerated function. Do not modify directly.

```

5.86.2.39 getreg_temperature_sensor()

```

def pALPIDE3.PAlpide3.getreg_temperature_sensor (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for Temperature Sensor
Register fields
=====
    TempSensor       [15:0] :    0x0

Original docstring
=====
Temperature Sensor

This is an autogenerated function. Do not modify directly.

```

5.86.2.40 getreg_VCASN()

```
def pALPIDE3.PAlpide3.getreg_VCASN (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCASN                [7:0] : 0x39

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.41 getreg_VCASN2()

```
def pALPIDE3.PAlpide3.getreg_VCASN2 (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCASN2                [7:0] : 0x40

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.42 getreg_VCASP()

```
def pALPIDE3.PAlpide3.getreg_VCASP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCASP                [7:0] : 0x56

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.43 getreg_VCLIP()

```
def pALPIDE3.PAlpide3.getreg_VCLIP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VCLIP                      [7:0] :   0x0

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.44 getreg_VPULSEH()

```
def pALPIDE3.PAlpide3.getreg_VPULSEH (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VPULSEH                   [7:0] :  0xff

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.45 getreg_VPULSEL()

```
def pALPIDE3.PAlpide3.getreg_VPULSEL (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VPULSEL                   [7:0] :   0x0

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.46 getreg_VRESETD()

```
def pALPIDE3.PAlpide3.getreg_VRESETD (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VRESETD                [7:0] :    0x0

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.47 getreg_VRESETP()

```
def pALPIDE3.PAlpide3.getreg_VRESETP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VRESETP                [7:0] :   0x75

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.48 getreg_VTEMP()

```
def pALPIDE3.PAlpide3.getreg_VTEMP (
    self,
    commitTransaction = None,
    log = None,
    verbose = False )

Autogenerated function for
Register fields
=====
    VTEMP                  [7:0] :    0x0

Origininal docstring
=====
```

This is an autogenerated function. Do not modify directly.

5.86.2.49 `init_pixel_matrix()`

```
def pALPIDE3.PAlpide3.init_pixel_matrix (
    self,
    readback = None,
    log = None,
    commitTransaction = None )
```

MATRIX #.

init the matrix by unmasking and disabling pulse

5.86.2.50 `initialize()`

```
def pALPIDE3.PAlpide3.initialize (
    self,
    disable_manchester = 1,
    grst = True )
```

Perform chip initialization procedure ([grst], manchester, RORST)

5.86.2.51 `initialize_readout()`

```
def pALPIDE3.PAlpide3.initialize_readout (
    self,
    PLLDAC = None,
    DriverDAC = None,
    PreDAC = None,
    PLLDelayStages = None,
    PLLBandwidthControl = None,
    commitTransaction = None,
    log = None,
    readback = None )
```

Perform initialization sequenca for readout

Reimplemented from [chip.Chip](#).

5.86.2.52 propagate_clock()

```
def pALPIDE3.PAlpide3.propagate_clock (
    self,
    acommitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate 600 MHz clock from chip

Reimplemented from [chip.Chip](#).

5.86.2.53 propagate_comma()

```
def pALPIDE3.PAlpide3.propagate_comma (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate COMMA pattern from chip (for realignment)

Reimplemented from [chip.Chip](#).

5.86.2.54 propagate_data()

```
def pALPIDE3.PAlpide3.propagate_data (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate Data from chip

Reimplemented from [chip.Chip](#).

5.86.2.55 propagate_prbs()

```
def pALPIDE3.PAlpide3.propagate_prbs (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Propagate PRBS test pattern from chip

Reimplemented from [chip.Chip](#).

5.86.2.56 reset()

```
def pALPIDE3.PAlpide3.reset (
    self )
```

Reset the chip

Reimplemented from [chip.Chip](#).

5.86.2.57 reset_pll()

```
def pALPIDE3.PAlpide3.reset_pll (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Reset the pll

Reimplemented from [chip.Chip](#).

5.86.2.58 set_default_variables()

```
def pALPIDE3.PAlpide3.set_default_variables (
    self,
    commitTransaction,
    log,
    readback = None )
```

Set default values for commitTransaction, log and readback. If values are set, the function values are used, if not, the class default is used

5.86.2.59 `set_xoff()`

```
def pALPIDE3.PAlpide3.set_xoff (
    self,
    XOff = 1,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Send XOFF to the chip. 1 -> Xoff, 0 -> Xon

Reimplemented from [chip.Chip](#).

5.86.2.60 `setreg_analog_monitor_and_override()`

```
def pALPIDE3.PAlpide3.setreg_analog_monitor_and_override (
    self,
    VoltageDACSel = None,
    CurrentDACSel = None,
    SWCNTL_DACMONI = None,
    SWCNTL_DACMONV = None,
    IRefBufferCurrent = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Analog Monitor and Override Register

Register fields

=====

VoltageDACSel	[3:0]	: 0x0
CurrentDACSel	[6:4]	: 0x0
SWCNTL_DACMONI	[7]	: 0x0
SWCNTL_DACMONV	[8]	: 0x0
IRefBufferCurrent	[10:9]	: 0x2

Original docstring

=====

Analog Monitor and Override Register

DACMONI current selection:

```
0 IRESET
1 IAUX2
2 IBIAS
3 IDB
4 IREF
5 ITHR
6 IREFBuffer
```

DACMONV voltage selection:

```
0 VCASN
1 VCASP
2 VPULSEH
3 VPULSEL
4 VRESETP
5 VRESETD
6 VCASN2
7 VCLIP
8 VTEMP
```

IRefBuffer current setting:

```

0    0,25uA
1    0,75uA
2    1,00uA (default)
3    1,25uA

```

This is an autogenerated function. Do not modify directly.

5.86.2.61 setreg_buffer_bypass()

```

def pALPIDE3.PAlpide3.setreg_buffer_bypass (
    self,
    VCASNBYPASS = None,
    VCASN2BYPASS = None,
    VCASPBYPASS = None,
    VCLIPBYPASS = None,
    IRESETBYPASS = None,
    IBIASBYPASS = None,
    ITHRBYPASS = None,
    IDBBYPASS = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

```

Autogenerated function for Buffer Bypass Register

Register fields

=====

VCASNBYPASS	[0] :	0x0
VCASN2BYPASS	[1] :	0x0
VCASPBYPASS	[2] :	0x0
VCLIPBYPASS	[3] :	0x0
IRESETBYPASS	[4] :	0x0
IBIASBYPASS	[5] :	0x0
ITHRBYPASS	[6] :	0x0
IDBBYPASS	[7] :	0x0

Original docstring

=====

Buffer Bypass Register

This is an autogenerated function. Do not modify directly.

5.86.2.62 setreg_busy_min_width()

```

def pALPIDE3.PAlpide3.setreg_busy_min_width (
    self,
    BusyMinWidth = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

```

Autogenerated function for BUSY min width
 Register fields
 =====
 BusyMinWidth [4:0] : 0x8

Orgininal docstring
 =====
 BUSY min width

This is an autogenerated function. Do not modify directly.

5.86.2.63 setreg_cmd()

```
def pALPIDE3.PAlpide3.setreg_cmd (
    self,
    Command = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Command Register.
 Register fields
 =====
 Command [15:0] : 0x0

Orgininal docstring
 =====
 Command Register.
 Valid opcodes:
 16'hFF00 CMUCLRERR
 16'h00D2 GRST
 16'h00E4 PRST
 16'h0078 PULSE
 16'h0036 BCRST
 16'h0063 RORST
 16'h00B1,0055,00C9,002D TRIGGER
 16'h009C WROP
 16'h004E RDOP

This is an autogenerated function. Do not modify directly.

5.86.2.64 setreg_cmu_and_dmu_cfg()

```
def pALPIDE3.PAlpide3.setreg_cmu_and_dmu_cfg (
    self,
    PreviousChipID = None,
    DMUXOff = None,
    InitialToken = None,
    DisableManchester = None,
    EnableDDR = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for CMU and DMU Configuration Register
Register fields

=====

PreviousChipID	[3:0] :	0xf
DMUXOff	[4] :	0x0
InitialToken	[5] :	0x0
DisableManchester	[6] :	0x0
EnableDDR	[7] :	0x1

Original docstring

=====

CMU and DMU Configuration Register

This is an autogenerated function. Do not modify directly.

5.86.2.65 setreg_dac_settings_cmu_io_buffers()

```
def pALPIDE3.PAlpide3.setreg_dac_settings_cmu_io_buffers (
    self,
    DCTRLReceiver = None,
    DCTRLDriver = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DAC settings for CMU I/O buffers
Register fields

=====

DCTRLReceiver	[3:0] :	0xa
DCTRLDriver	[7:4] :	0xa

Original docstring

=====

DAC settings for CMU I/O buffers

This is an autogenerated function. Do not modify directly.

5.86.2.66 setreg_dac_settings_dclk_and_mclk_io_buffers()

```
def pALPIDE3.PAlpide3.setreg_dac_settings_dclk_and_mclk_io_buffers (
    self,
    DCLKReceiver = None,
    DCLKDriver = None,
    MCLKReceiver = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DAC settings for DCLK and MCLK I/O buffers

Register fields

=====

```
DCLKReceiver          [3:0] :    0xa
DCLKDriver            [7:4] :    0xa
MCLKReceiver          [11:8] :   0xa
```

Original docstring

=====

DAC settings for DCLK and MCLK I/O buffers

This is an autogenerated function. Do not modify directly.

5.86.2.67 setreg_disable_regions_lsbs()

```
def pALPIDE3.PAlpide3.setreg_disable_regions_lsbs (
    self,
    RegionDisable = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Disable of regions 0-15

Register fields

=====

```
RegionDisable          [15:0] :   0x0
```

Original docstring

=====

Disable of regions 0-15

This is an autogenerated function. Do not modify directly.

5.86.2.68 setreg_disableof_regions_msbs()

```
def pALPIDE3.PAlpide3.setreg_disableof_regions_msbs (
    self,
    RegionDisable = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Disableof regions 16-31

Register fields

=====

```
RegionDisable          [15:0] :   0x0
```

Original docstring

=====

Disableof regions 16-31

This is an autogenerated function. Do not modify directly.

5.86.2.69 setreg_dtu_cfg()

```
def pALPIDE3.PAlpide3.setreg_dtu_cfg (
    self,
    VcoDelayStages = None,
    PllBandwidthControl = None,
    PllOffSignal = None,
    SerPhase = None,
    PLLReset = None,
    LoadENStatus = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU Configuration Register

Register fields

=====

VcoDelayStages	[1:0] :	0x1
PllBandwidthControl	[2] :	0x1
PllOffSignal	[3] :	0x1
SerPhase	[7:4] :	0x8
PLLReset	[8] :	0x0
LoadENStatus	[12] :	0x0

Orgininal docstring

=====

DTU Configuration Register

LoadENStatus is read-only

This is an autogenerated function. Do not modify directly.

5.86.2.70 setreg_dtu_dacs()

```
def pALPIDE3.PAlpide3.setreg_dtu_dacs (
    self,
    PLLDAC = None,
    DriverDAC = None,
    PreDAC = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU DACs Register

Register fields

=====

PLLDAC	[3:0] :	0x8
DriverDAC	[7:4] :	0x8
PreDAC	[11:8] :	0x0

Orgininal docstring

=====

DTU DACs Register

This is an autogenerated function. Do not modify directly.

5.86.2.71 setreg_dtu_pll_lock_2()

```
def pALPIDE3.PAlpide3.setreg_dtu_pll_lock_2 (
    self,
    LockWaitCycles = None,
    UnlockWaitCycles = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU PLL Lock Register 2
Register fields

=====

LockWaitCycles	[7:0] :	0x0
UnlockWaitCycles	[15:8] :	0x0

Original docstring

=====

DTU PLL Lock Register 2

This is an autogenerated function. Do not modify directly.

5.86.2.72 setreg_dtu_test_1()

```
def pALPIDE3.PAlpide3.setreg_dtu_test_1 (
    self,
    TestEN = None,
    IntPatternEN = None,
    PrbsEN = None,
    Bypass8b10b = None,
    BDIN8b10b0 = None,
    BDIN8b10b1 = None,
    BDIN8b10b2 = None,
    K0 = None,
    K1 = None,
    K2 = None,
    ForceSetLoadEN = None,
    ForceUnsetLoadEN = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for DTU Test Register 1
Register fields

=====

TestEN	[0] :	0x0
IntPatternEN	[1] :	0x0
PrbsEN	[2] :	0x0
Bypass8b10b	[3] :	0x0
BDIN8b10b0	[5:4] :	0x0
BDIN8b10b1	[7:6] :	0x0
BDIN8b10b2	[9:8] :	0x0
K0	[10] :	0x0
K1	[11] :	0x0
K2	[12] :	0x0

```

ForceSetLoadEN          [13] :   0x0
ForceUnsetLoadEN        [14] :   0x0

```

Original docstring

=====

DTU Test Register 1

This is an autogenerated function. Do not modify directly.

5.86.2.73 setreg_dtu_test_2()

```

def pALPIDE3.PAlpide3.setreg_dtu_test_2 (
    self,
    DIN0 = None,
    DIN1 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

```

Autogenerated function for DTU Test Register 2

Register fields

=====

```

DIN0                      [7:0] :   0x0
DIN1                      [15:8] :   0x0

```

Original docstring

=====

DTU Test Register 2

This is an autogenerated function. Do not modify directly.

5.86.2.74 setreg_dtu_test_3()

```

def pALPIDE3.PAlpide3.setreg_dtu_test_3 (
    self,
    DIN2 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

```

Autogenerated function for DTU Test Register 3

Register fields

=====

```

DIN2                      [7:0] :   0x0

```

Original docstring

=====

DTU Test Register 3

This is an autogenerated function. Do not modify directly.

5.86.2.75 setreg_fromu_cfg_1()

```
def pALPIDE3.PAlpide3.setreg_fromu_cfg_1 (
    self,
    MEBMask = None,
    EnInternalGeneration = None,
    EnBusyMonitoring = None,
    TPulseMode = None,
    EnTestStrobe = None,
    SMState = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 1

Register fields

=====

MEBMask	[2:0]	: 0x0
EnInternalGeneration	[3]	: 0x0
EnBusyMonitoring	[4]	: 0x1
TPulseMode	[5]	: 0x0
EnTestStrobe	[6]	: 0x0
SMState	[12:8]	: 0x0

Original docstring

=====

FROMU Configuration Register 1

EnBusyMonitoring default = 1

SMState is Read only

This is an autogenerated function. Do not modify directly.

5.86.2.76 setreg_fromu_cfg_2()

```
def pALPIDE3.PAlpide3.setreg_fromu_cfg_2 (
    self,
    FrameDuration = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for FROMU Configuration Register 2

Register fields

=====

FrameDuration	[15:0]	: 0x0
---------------	--------	-------

Original docstring

=====

FROMU Configuration Register 2

This is an autogenerated function. Do not modify directly.

5.86.2.77 setreg_fromu_pulsing1()

```
def pALPIDE3.PAlpide3.setreg_fromu_pulsing1 (
    self,
    TPulseDelay = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for FROMU Pulsing Register1
Register fields
=====
    TPulseDelay          [15:0] :    0x0

Origininal docstring
=====
FROMU Pulsing Register1

This is an autogenerated function. Do not modify directly.
```

5.86.2.78 setreg_fromu_pulsing_2()

```
def pALPIDE3.PAlpide3.setreg_fromu_pulsing_2 (
    self,
    TPulseDuration = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )

Autogenerated function for FROMU Pulsing Register 2
Register fields
=====
    TPulseDuration       [15:0] :    0x0

Origininal docstring
=====
FROMU Pulsing Register 2

This is an autogenerated function. Do not modify directly.
```

5.86.2.79 setreg_fuses_write_lsbs()

```
def pALPIDE3.PAlpide3.setreg_fuses_write_lsbs (
    self,
    FusesWriteLSBs = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Fuses Write LSBs
Register fields

=====

FusesWriteLSBs [15:0] : 0x0

Original docstring

=====

Fuses Write LSBs

This is an autogenerated function. Do not modify directly.

5.86.2.80 setreg_fuses_write_msbs()

```
def pALPIDE3.PAlpide3.setreg_fuses_write_msbs (
    self,
    FusesWriteMSBs = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Fuses Write MSBs
Register fields

=====

FusesWriteMSBs [7:0] : 0x0

Original docstring

=====

Fuses Write MSBs

This is an autogenerated function. Do not modify directly.

5.86.2.81 setreg_IAUX2()

```
def pALPIDE3.PAlpide3.setreg_IAUX2 (
    self,
    IAUX2 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

IAUX2 [7:0] : 0x0

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.86.2.82 setreg_IBIAS()

```
def pALPIDE3.PAlpide3.setreg_IBIAS (
    self,
    IBIAS = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
    IBIAS                                [7:0] : 0x40
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.83 setreg_IDB()

```
def pALPIDE3.PAlpide3.setreg_IDB (
    self,
    IDB = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
    IDB                                [7:0] : 0x40
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.84 setreg_IRESET()

```
def pALPIDE3.PAlpide3.setreg_IRESET (
    self,
    IRESET = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

 IRESET [7:0] : 0x32

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.86.2.85 setreg_ITHR()

```
def pALPIDE3.PAlpide3.setreg_ITHR (
    self,
    ITHR = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

 ITHR [7:0] : 0x33

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.86.2.86 setreg_periphery_ctrl()

```
def pALPIDE3.PAlpide3.setreg_periphery_ctrl (
    self,
    ChipModeSelector = None,
    ClusteringEnabled = None,
    StartMemSelfTest = None,
    MatrixROSpeed = None,
    ForceBusy = None,
    ForceBusyValue = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Periphery Control register
Register fields

=====

 ChipModeSelector [1:0] : 0x0
 ClusteringEnabled [2] : 0x0
 StartMemSelfTest [3] : 0x0

MatrixROSpeed	[5] :	0x1
ForceBusy	[6] :	0x0
ForceBusyValue	[7] :	0x0

Orgininal docstring

=====

Periphery Control register

Chip Mode Selector values:

0 Cfg Mode

1 Readout Triggered Mode

2 Readout Continuous Mode

MatrixROSpeed 0=10MHz, 1=20MHz (default)

This is an autogenerated function. Do not modify directly.

5.86.2.87 setreg_pixel_cfg_1()

```
def pALPIDE3.PAlpide3.setreg_pixel_cfg_1 (
    self,
    RowSel = None,
    SetsAllRows = None,
    PIXCNFG_REGSEL = None,
    PIXCNFG_DATA = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Pixel CFG Register 1

Register fields

=====

RowSel	[8:0] :	0x0
SetsAllRows	[9] :	0x0
PIXCNFG_REGSEL	[10] :	0x0
PIXCNFG_DATA	[11] :	0x0

Orgininal docstring

=====

Pixel CFG Register 1

This is an autogenerated function. Do not modify directly.

5.86.2.88 setreg_pixel_cfg_2()

```
def pALPIDE3.PAlpide3.setreg_pixel_cfg_2 (
    self,
    ColSel = None,
    SetsAllColumns = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Pixel CFG Register 2
Register fields

=====

ColSel	[9:0] :	0x0
SetsAllColumns	[10] :	0x0

Original docstring

=====

Pixel CFG Register 2

This is an autogenerated function. Do not modify directly.

5.86.2.89 setreg_pixel_cfg_3()

```
def pALPIDE3.PAlpide3.setreg_pixel_cfg_3 (
    self,
    MatrixLatchEn = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for Pixel CFG Register 3
Register fields

=====

MatrixLatchEn	[0] :	0x0
---------------	-------	-----

Original docstring

=====

Pixel CFG Register 3

This is an autogenerated function. Do not modify directly.

5.86.2.90 setreg_VCASN()

```
def pALPIDE3.PAlpide3.setreg_VCASN (
    self,
    VCASN = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VCASN	[7:0] :	0x39
-------	---------	------

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.86.2.91 setreg_VCASN2()

```
def pALPIDE3.PAlpide3.setreg_VCASN2 (
    self,
    VCASN2 = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
VCASN2                                [7:0] : 0x40
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.92 setreg_VCASP()

```
def pALPIDE3.PAlpide3.setreg_VCASP (
    self,
    VCASP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
VCASP                                [7:0] : 0x56
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.93 setreg_VCLIP()

```
def pALPIDE3.PAlpide3.setreg_VCLIP (
    self,
    VCLIP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VCLIP [7:0] : 0x0

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.86.2.94 setreg_VPULSEH()

```
def pALPIDE3.PAlpide3.setreg_VPULSEH (
    self,
    VPULSEH = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VPULSEH [7:0] : 0xff

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.86.2.95 setreg_VPULSEL()

```
def pALPIDE3.PAlpide3.setreg_VPULSEL (
    self,
    VPULSEL = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

=====

VPULSEL [7:0] : 0x0

Original docstring

=====

This is an autogenerated function. Do not modify directly.

5.86.2.96 setreg_VRESETD()

```
def pALPIDE3.PAlpide3.setreg_VRESETD (
    self,
    VRESETD = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
VRESETD                [7:0] :    0x0
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.97 setreg_VRESETP()

```
def pALPIDE3.PAlpide3.setreg_VRESETP (
    self,
    VRESETP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields

```
=====
VRESETP                [7:0] :    0x75
```

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.98 setreg_VTEMP()

```
def pALPIDE3.PAlpide3.setreg_VTEMP (
    self,
    VTEMP = None,
    commitTransaction = None,
    log = None,
    readback = None,
    verbose = False )
```

Autogenerated function for
Register fields
=====

VTEMP [7:0] : 0x0

Original docstring
=====

This is an autogenerated function. Do not modify directly.

5.86.2.99 trigger()

```
def pALPIDE3.PAlpide3.trigger (
    self,
    commitTransaction = None,
    readback = None,
    log = None,
    verbose = False )
```

Send a trigger to the chip

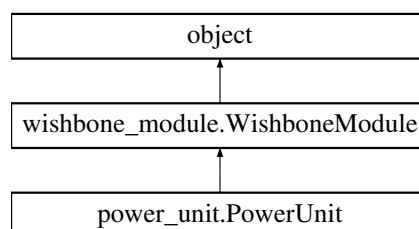
Reimplemented from [chip.Chip](#).

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/pALPIDE3.py

5.87 power_unit.PowerUnit Class Reference

Inheritance diagram for power_unit.PowerUnit:



Public Member Functions

- def **__init__** (self, board_obj, moduleid)
- def **initialize** (self)
- def **setup_power_IB** (self, dvdd=1.9, dvdd_current=1.5, avdd=1.9, avdd_current=1.5, bb=0.0, bb_↵
current=None, module=0)
- def **setup_power_IBs** (self, dvdd=1.9, dvdd_current=1.5, avdd=1.9, avdd_current=1.5, bb=0, bb_↵
current=None, module_list=[0, 1])
- def **log_enable_status** (self)
- def **power_on_IB** (self, module, backbias_en=0)
- def **power_on_IBs** (self, module_list=[0, 1], backbias_en=0)
- def **power_off_all** (self)
- def **get_values_IB** (self, module)
- def **get_values_IBs** (self, module_list=[0, 1])
- def **log_values_IB** (self, module)
- def **log_values_IBs** (self, module_list=[0, 1])
- def [RaiseThresholdsToMax](#) (self, commitTransaction=True)
- def [LowerThresholdsToMin](#) (self, commitTransaction=True)
- def [SetThreshold](#) (self, channel, value, commitTransaction=True)
- def [SetThresholdWithMask](#) (self, mask, value, commitTransaction=True)
- def [SetThresholdAll](#) (self, value, commitTransaction=True)
- def [EnablePowerAll](#) (self, commitTransaction=True)
- def [EnableBiasAll](#) (self, commitTransaction=True)
- def [EnablePower](#) (self, channel, commitTransaction=True)
- def [EnablePowerWithMask](#) (self, mask, commitTransaction=True)
- def [EnableBiasWithMask](#) (self, mask, commitTransaction=True)
- def [DisablePowerAll](#) (self, commitTransaction=True)
- def [DisableBiasAll](#) (self, commitTransaction=True)
- def [SetPowerVoltage](#) (self, channel, voltagecode, commitTransaction=True)
- def [SetPowerVoltageAll](#) (self, voltagecode, commitTransaction=True)
- def [SetBiasVoltage](#) (self, voltagecode, commitTransaction=True)
- def [ConfigurePowerADC](#) (self, commitTransaction=True)
- def [ConfigureBiasADC](#) (self, commitTransaction=True)
- def [get_power_enable_status](#) (self, commitTransaction=True)
- def [get_bias_enable_status](#) (self, commitTransaction=True)
- def [ReadPowerADC](#) (self)
- def [ReadBiasADC](#) (self)
- def **get_power_adc_values** (self, module)
- def [ReadPowerADCChannel](#) (self, channel)
- def [ReadBiasADCChannel](#) (self, channel)
- def [ConfigureRTD](#) (self, SensorID, commitTransaction=True)
- def [ReadRTD](#) (self, SensorID, commitTransaction=True)

Public Attributes

- **ThresCurrAddress**
- **PotPowerAddress**
- **PotBiasAddress**
- **ADCAddressSetup**
- **ADCBiasAddressSetup**
- **ADCAddress**
- **ADCBiasAddress**
- **IOExpanderBiasAddress**
- **IOExpanderPowerAddress**
- **TempThreshConfigAddress**
- **TempThreshRdAddress**

Static Public Attributes

- int **CONFIG_RTD_BYTE3_60HZ** = 0xc2
- int **CONFIG_RTD_BYTE3_50HZ** = 0xc3

5.87.1 Detailed Description

Class for performing I2C transactions with the Power Unit on RUv1

5.87.2 Member Function Documentation

5.87.2.1 ConfigureBiasADC()

```
def power_unit.PowerUnit.ConfigureBiasADC (
    self,
    commitTransaction = True )
```

Configure control registers of all Bias ADCs

5.87.2.2 ConfigurePowerADC()

```
def power_unit.PowerUnit.ConfigurePowerADC (
    self,
    commitTransaction = True )
```

Configure control registers of all Power ADCs

5.87.2.3 ConfigureRTD()

```
def power_unit.PowerUnit.ConfigureRTD (
    self,
    SensorID,
    commitTransaction = True )
```

Configure temperature sensor with ID SensorID

5.87.2.4 DisableBiasAll()

```
def power_unit.PowerUnit.DisableBiasAll (
    self,
    commitTransaction = True )
```

Disable each bias channel

5.87.2.5 DisablePowerAll()

```
def power_unit.PowerUnit.DisablePowerAll (
    self,
    commitTransaction = True )
```

Disable each power channel

5.87.2.6 EnableBiasAll()

```
def power_unit.PowerUnit.EnableBiasAll (
    self,
    commitTransaction = True )
```

Enable all bias channels

5.87.2.7 EnableBiasWithMask()

```
def power_unit.PowerUnit.EnableBiasWithMask (
    self,
    mask,
    commitTransaction = True )
```

Enable each bias for channels in bit position of mask

5.87.2.8 EnablePower()

```
def power_unit.PowerUnit.EnablePower (
    self,
    channel,
    commitTransaction = True )
```

Enable power channel 'channel'

5.87.2.9 EnablePowerAll()

```
def power_unit.PowerUnit.EnablePowerAll (
    self,
    commitTransaction = True )
```

Enable all power channels

5.87.2.10 EnablePowerWithMask()

```
def power_unit.PowerUnit.EnablePowerWithMask (
    self,
    mask,
    commitTransaction = True )
```

Enable each power channel of bitposition in 'mask'

5.87.2.11 get_bias_enable_status()

```
def power_unit.PowerUnit.get_bias_enable_status (
    self,
    commitTransaction = True )
```

Read enable status of all bias channels as a mask

5.87.2.12 `get_power_enable_status()`

```
def power_unit.PowerUnit.get_power_enable_status (
    self,
    commitTransaction = True )
```

Read enable status of all power channels as a mask

5.87.2.13 `LowerThresholdsToMin()`

```
def power_unit.PowerUnit.LowerThresholdsToMin (
    self,
    commitTransaction = True )
```

Set current thresholds of all channels to 0x0000

5.87.2.14 `RaiseThresholdsToMax()`

```
def power_unit.PowerUnit.RaiseThresholdsToMax (
    self,
    commitTransaction = True )
```

Set current thresholds of all channels to 0xffff

5.87.2.15 `ReadBiasADC()`

```
def power_unit.PowerUnit.ReadBiasADC (
    self )
```

Trigger ADC conversion and readout of all Bias ADC channels

5.87.2.16 `ReadBiasADCChannel()`

```
def power_unit.PowerUnit.ReadBiasADCChannel (
    self,
    channel )
```

Trigger ADC conversion and readout of a Bias ADC channel

5.87.2.17 ReadPowerADC()

```
def power_unit.PowerUnit.ReadPowerADC (
    self )
```

Trigger ADC conversion and readout of all Power ADC channels

5.87.2.18 ReadPowerADCChannel()

```
def power_unit.PowerUnit.ReadPowerADCChannel (
    self,
    channel )
```

Trigger ADC conversion and readout a Power ADC channel

5.87.2.19 ReadRTD()

```
def power_unit.PowerUnit.ReadRTD (
    self,
    SensorID,
    commitTransaction = True )
```

Read temperature Sensor with ID SensorID

5.87.2.20 SetBiasVoltage()

```
def power_unit.PowerUnit.SetBiasVoltage (
    self,
    voltagecode,
    commitTransaction = True )
```

Set output voltage setting of bias channels to 'voltage'

5.87.2.21 SetPowerVoltage()

```
def power_unit.PowerUnit.SetPowerVoltage (
    self,
    channel,
    voltagecode,
    commitTransaction = True )
```

Set output voltage setting of power 'channel' to 'voltage'

5.87.2.22 SetPowerVoltageAll()

```
def power_unit.PowerUnit.SetPowerVoltageAll (
    self,
    voltagecode,
    commitTransaction = True )
```

Set output voltage setting of all power channels to 'voltage'

5.87.2.23 SetThreshold()

```
def power_unit.PowerUnit.SetThreshold (
    self,
    channel,
    value,
    commitTransaction = True )
```

Set threshold of channel 'channel' to 'value'

```
16 channels
0 + 2*module: analog
1 + 2*module: digital
```

5.87.2.24 SetThresholdAll()

```
def power_unit.PowerUnit.SetThresholdAll (
    self,
    value,
    commitTransaction = True )
```

Set threshold of all channels to 'value'

5.87.2.25 SetThresholdWithMask()

```
def power_unit.PowerUnit.SetThresholdWithMask (
    self,
    mask,
    value,
    commitTransaction = True )
```

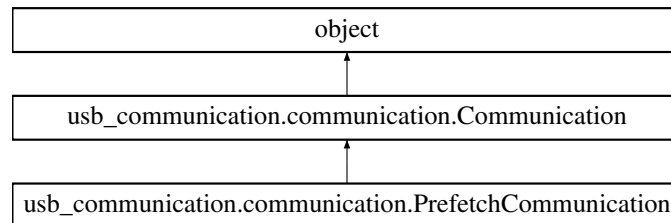
Set threshold of each channel of bitposition in 'mask' to 'value'

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_unit.py

5.88 usb_communication.communication.PrefetchCommunication Class Reference

Inheritance diagram for usb_communication.communication.PrefetchCommunication:



Public Member Functions

- def `__init__` (self, comm, [enable_rderr_exception](#)=False)
- def [stop](#) (self)
- def [discardall_dp1](#) (self, maxReads=10)
- def [discardall_dp2](#) (self, maxReads=10)
- def [discardall_dp3](#) (self, maxReads=10)
- def [start_recording](#) (self)
- def [stop_recording](#) (self)
- def [prefetch](#) (self)
- def [load_sequence](#) (self, sequence)
- def [stop_prefetch_mode](#) (self, checkEmpty=True)
- def [get_sequence](#) (self)

Public Attributes

- **comm**
- **logger**
- **recording**
- **prefetch_mode**
- **sequence**
- **prefetched_commands**
- **prefetched_results**
- **prefetch_read_buffer**
- **prefetched_read_buffer**

5.88.1 Detailed Description

Prefetch Communication controller.

Optimized communication control which provides functions to record a set of wishbone transactions, and later prefetch the same set of wishbone transactions in a single usb packet.

5.88.2 Constructor & Destructor Documentation

5.88.2.1 `__init__()`

```
def usb_communication.communication.PrefetchCommunication.__init__ (
    self,
    comm,
    enable_r derr_exception = False )
```

Initialize with given comm object as actual communication controller

5.88.3 Member Function Documentation

5.88.3.1 `get_sequence()`

```
def usb_communication.communication.PrefetchCommunication.get_sequence (
    self )
```

Return the current sequence of commands

5.88.3.2 `load_sequence()`

```
def usb_communication.communication.PrefetchCommunication.load_sequence (
    self,
    sequence )
```

Load given sequence of commands to communication (no call to usb yet)

5.88.3.3 prefetch()

```
def usb_communication.communication.PrefetchCommunication.prefetch (  
    self )
```

Prefetch mode. Load sequence of commands, send to usb, store results

5.88.3.4 start_recording()

```
def usb_communication.communication.PrefetchCommunication.start_recording (  
    self )
```

Start recording wishbone access commands

5.88.3.5 stop()

```
def usb_communication.communication.PrefetchCommunication.stop (  
    self )
```

Perform any operation on close

Reimplemented from [usb_communication.communication.Communication](#).

5.88.3.6 stop_recording()

```
def usb_communication.communication.PrefetchCommunication.stop_recording (  
    self )
```

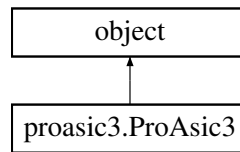
Finish recording wishbone access commands. Return the sequence of commands generated so far

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py

5.89 proasic3.ProAsic3 Class Reference

Inheritance diagram for proasic3.ProAsic3:



Public Member Functions

- def **__init__** (self, sca)
- def **initialize** (self)
- def **set_i2c_channel** (self, channel)
- def **write_reg** (self, address, value)
- def **write_fifo_reg** (self, address, data)
- def **read_reg** (self, address)
- def **read_fifo_reg** (self, address, length)
- def **set_reset** (self, value)
- def **set_start_program** (self, value)
- def **get_version** (self)
- def **get_clock_status** (self)
- def **set_clock_config** (self, lcl_clk_in_se, lcl_clk_cs, lcl_clk_rst)
- def **get_clock_config** (self)
- def **set_cc_command_register** (self, opcode, execute=1)
- CC.*
- def **get_cc_command_register** (self)
- def **get_program_done** (self)
- def **reprogram_ultrascale** (self)
- def **start_blind_scrubbing** (self)
- def **stop_blind_scrubbing** (self)
- def **trigger_single_scrub** (self)
- def **clear_scrubbing_counter** (self)
- def **get_cc_status** (self)
- def **get_scrubbing_counter** (self)
- def **get_crc** (self)
- def **is_idle** (self)
- def **is_init_config_done** (self)
- def **is_scrubbing** (self)
- def **get_cc_active_state** (self)
- def **set_rc_command_register** (self, opcode, execute=1)
- RC.*
- def **get_rc_command_register** (self)
- def **get_rc_current_page** (self)
- def **get_rc_active_state** (self)
- def **set_smmap_command_register** (self, opcode, execute=1)
- SMAP.*
- def **get_smmap_command_register** (self)
- def **set_smmap_byte_tx** (self, byte)
- def **get_smmap_byte_tx** (self)
- def **get_smmap_byte_rx** (self)

- def **get_smap_status** (self)
- def **get_fifo_rx_data** (self)
 - FIFO.*
- def **set_fifo_tx_data** (self, data)
- def **get_fifo_status** (self)
- def **set_fifo_writer_command_register** (self, opcode, execute=1)
- def **get_fifo_writer_command_register** (self)
- def **get_fifo_writer_status** (self)
- def **set_flash_command_register** (self, opcode, execute=1)
 - FLASH.*
- def **get_flash_command_register** (self)
- def **set_flash_address** (self, block_address, page_address)
- def **get_flash_byte_counter** (self)
- def **set_flash_select_ic** (self, value)
- def **get_flash_select_ic** (self)
- def **get_flash_status_word** (self)
- def **set_pattern_checker** (self, value)
- def **get_flash_state** (self)
- def **get_flash_page_size** (self)
- def **set_flash_page_size** (self, page_size)
- def **set_ecc_command_register** (self, enable, clear_status=0)
 - ECC.*
- def **get_ecc_command_register** (self)
- def **enable_ecc** (self)
- def **disable_ecc** (self)
- def **reset_ecc_counters_and_status** (self)
- def **clear_ecc_status** (self)
- def **get_ecc_sb_error_counter** (self)
- def **get_ecc_fifo_tmr_error** (self)
- def **get_dipswitches** (self)
 - DEBUG.*
- def **get_led** (self)
- def **set_led** (self, value)
- def **dump_config** (self, filename=None)
 - DUMP.*
- def **flash_read_page** (self, page_address)
- def **flash_dump_page** (self, page_address, to_screen=False, path='.', file_base_name='page')
- def **flash_write_parameter_page** (self, fw_start_page, fw_size_pages, scrub_start_page, scrub_size_pages, verify=False, to_file=False)
- def **flash_write_firmware** (self, fw_filename, fw_start_block=0x100, scrub_filename="", scrub_start_block=0x200, verify=False, param_only=False, to_file=True)

Public Attributes

- **name**
- **logger**
- **sca**
- **sca_channel_used**

5.89.1 Detailed Description

Class to handle PA3 transactions

5.89.2 Member Function Documentation

5.89.2.1 dump_config()

```
def proasic3.ProAsic3.dump_config (
    self,
    filename = None )
```

DUMP.

Print (or if filename is set, log to file) all wishbone registers of the PA3

5.89.2.2 flash_dump_page()

```
def proasic3.ProAsic3.flash_dump_page (
    self,
    page_address,
    to_screen = False,
    path = '.',
    file_base_name = 'page' )
```

Dumps the selected page to file.
if to_screen is True it writes it into the logfile

5.89.2.3 flash_read_page()

```
def proasic3.ProAsic3.flash_read_page (
    self,
    page_address )
```

Reads a page from the flash and returns it as a bytearray

5.89.2.4 flash_write_firmware()

```
def proasic3.ProAsic3.flash_write_firmware (
    self,
    fw_filename,
    fw_start_block = 0x100,
    scrub_filename = "",
    scrub_start_block = 0x200,
    verify = False,
    param_only = False,
    to_file = True )
```

Write firmware for the UltraScale (and optionally a scrubbing image) to the flash

If param_only is set to true, the parameter page is updated based on file sizes and start blocks, but no firmware/scrub image is actually written to the flash.

Code based on flashWritePageI2C() from Gitle Mikkelsen's repo:

https://gitlab.cern.ch/alice-its-wp10-firmware/CRU_ITS/blob/GM_dev/software/py/PA3.py

5.89.2.5 flash_write_parameter_page()

```
def proasic3.ProAsic3.flash_write_parameter_page (
    self,
    fw_start_page,
    fw_size_pages,
    scrub_start_page,
    scrub_size_pages,
    verify = False,
    to_file = False )
```

Write parameter page to FLASH

scrub_start_page and scrub_size_pages parameters can be omitted if there is no scrub image. When verify is set to true, the parameter page will be read back from flash and its contents verified.

5.89.2.6 get_cc_command_register()

```
def proasic3.ProAsic3.get_cc_command_register (
    self )
```

Reads the CC_CMD register

5.89.2.7 get_clock_config()

```
def proasic3.ProAsic3.get_clock_config (
    self )
```

Gets the clock configuration

5.89.2.8 get_clock_status()

```
def proasic3.ProAsic3.get_clock_status (
    self )
```

Get Status of clocks

5.89.2.9 get_version()

```
def proasic3.ProAsic3.get_version (
    self )
```

Gets the PA3 FPGA design version

5.89.2.10 read_fifo_reg()

```
def proasic3.ProAsic3.read_fifo_reg (
    self,
    address,
    length )
```

Reads up to 16 bytes to a the same PA3 address, ie. fifo mode, using SCA-I2C multi byte mode. Returns a bytearray with the data

5.89.2.11 read_reg()

```
def proasic3.ProAsic3.read_reg (
    self,
    address )
```

Reads a PA3 register using SCA-I2C single byte mode

5.89.2.12 set_cc_command_register()

```
def proasic3.ProAsic3.set_cc_command_register (
    self,
    opcode,
    execute = 1 )
```

CC.

Writes the CC_CMD register

5.89.2.13 set_clock_config()

```
def proasic3.ProAsic3.set_clock_config (
    self,
    lcl_clk_in_se,
    lcl_clk_cs,
    lcl_clk_rst )
```

Set the clock configuration

5.89.2.14 set_ecc_command_register()

```
def proasic3.ProAsic3.set_ecc_command_register (
    self,
    enable,
    clear_status = 0 )
```

ECC.

bit 1 is pulsed

5.89.2.15 set_i2c_channel()

```
def proasic3.ProAsic3.set_i2c_channel (
    self,
    channel )
```

Changes the active SCA-I2C channel used

5.89.2.16 `set_reset()`

```
def proasic3.ProAsic3.set_reset (
    self,
    value )
```

Sets the RESET pin of the PA3 FPGA to value

5.89.2.17 `set_start_program()`

```
def proasic3.ProAsic3.set_start_program (
    self,
    value )
```

Sets the START_PROGRAM pin of the PA3 FPGA to value

5.89.2.18 `write_fifo_reg()`

```
def proasic3.ProAsic3.write_fifo_reg (
    self,
    address,
    data )
```

Writes up to 16 bytes to a the same PA3 address, ie. fifo mode, using SCA-I2C multi byte mode. Expects a bytearray as input

5.89.2.19 `write_reg()`

```
def proasic3.ProAsic3.write_reg (
    self,
    address,
    value )
```

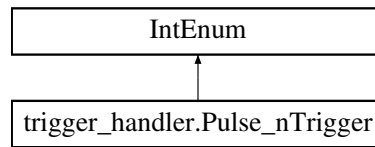
Writes to a PA3 register using SCA-I2C single byte mode

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py`

5.90 trigger_handler.Pulse_nTrigger Class Reference

Inheritance diagram for trigger_handler.Pulse_nTrigger:



Static Public Attributes

- int **Trigger** = 0x00
- int **Pulse** = 0x01

5.90.1 Detailed Description

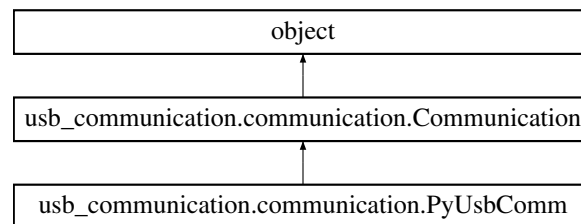
Pulse or trigger selection

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/trigger_handler.py

5.91 usb_communication.communication.PyUsbComm Class Reference

Inheritance diagram for usb_communication.communication.PyUsbComm:



Public Member Functions

- def **__init__** (self, VID=0x04b4, PID=0x0008, IF=2, PacketSize=1024, serialNr=None, enable_r derr_↔ exception=False)

Public Attributes

- **dev**
- **epo**
- **epi**
- **dp2**
- **dp3**

5.91.1 Detailed Description

Implementation of Communication class.

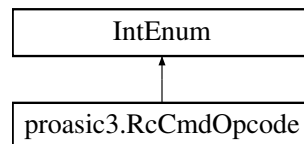
This implements the communication class by establishing a direct Connection to the usb port via PyUsb.

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py

5.92 proasic3.RcCmdOpcode Class Reference

Inheritance diagram for proasic3.RcCmdOpcode:



Static Public Attributes

- int **READ_PARAMETER** = 0x01
- int **READ_CONFIG** = 0x02
- int **READ_SCRUB** = 0x04
- int **START_PAGE** = 0x08
- int **STOP_PAGE** = 0x10

5.92.1 Detailed Description

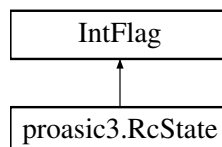
Read controller opcode mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.93 proasic3.RcState Class Reference

Inheritance diagram for proasic3.RcState:



Static Public Attributes

- int **BUSY** = 0
- int **WAIT_FOR_FLASH** = 1
- int **READING_BITFILE** = 2
- int **READING_PARAMETERS** = 3
- int **CONFIG_PARAMETERS_OK** = 4
- int **BS_PARAMETERS_OK** = 5
- int **PARAMETER_ERROR** = 6
- int **CONFIG_WITH_ECC_NON_VALID** = 7

5.93.1 Detailed Description

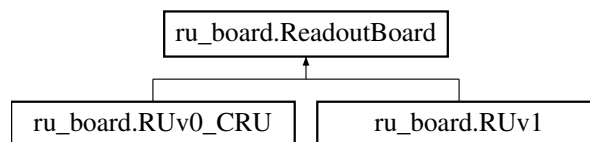
Read controller FSM state

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.94 ru_board.ReadoutBoard Class Reference

Inheritance diagram for ru_board.ReadoutBoard:



Public Member Functions

- def **__init__** (self, comm)
- def **initialize** (self)
- def **write** (self, mod, addr, data, commitTransaction=True)
- def **read** (self, mod, addr, commitTransaction=True)
- def **flush** (self)
- def **read_all** (self)

Public Attributes

- **comm**

5.94.1 Member Function Documentation

5.94.1.1 flush()

```
def ru_board.ReadoutBoard.flush (
    self )
```

Flush communication buffer.

5.94.1.2 read()

```
def ru_board.ReadoutBoard.read (
    self,
    mod,
    addr,
    commitTransaction = True )
```

Read from specific register on the board

5.94.1.3 read_all()

```
def ru_board.ReadoutBoard.read_all (
    self )
```

Read all pending results.

5.94.1.4 write()

```
def ru_board.ReadoutBoard.write (
    self,
    mod,
    addr,
    data,
    commitTransaction = True )
```

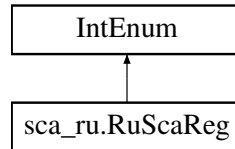
Write to specific register on the board

The documentation for this class was generated from the following file:

- /media/sf_proj/RUV1_Test/modules/board_support_software/software/py/ru_board.py

5.95 sca_ru.RuScaReg Class Reference

Inheritance diagram for sca_ru.RuScaReg:



Static Public Attributes

- int **CTRL_STATUS** = 0
- int **TRID_CHANNEL** = 1
- int **LENGTH_COMMAND** = 2
- int **LENGTH_ERROR** = 2
- int **DATA_LSB** = 3
- int **DATA_MSB** = 4

5.95.1 Detailed Description

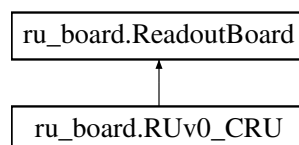
SCA wishbone module register mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/sca_ru.py

5.96 ru_board.RUv0_CRU Class Reference

Inheritance diagram for ru_board.RUv0_CRU:



Public Member Functions

- def **__init__** (self, comm)
- def **initialize** (self, reenable_forward_to_usb=1)
- def **initialize_gbt_x** (self, commitTransaction=True)
- def **is_gbt_rx_ready** (self)
- def **send_gbt_x_data_frame** (self, data, commitTransaction=True)
- def **send_trigger** (self, triggerType=0x10, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_start_of_triggered** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_end_of_triggered** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_start_of_continuous** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_heartbeat** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_end_of_continuous** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **write_gbt_x** (self, module, address, data, commitTransaction=True)
- def **read_gbt_x** (self, module, address, commitTransaction=True)
- def **read_gbt_x_data_from_usb** (self, size, rawoutfile=None)
- def **wait** (self, wait_value, commitTransaction=True)
- def **read_counters** (self)
- def **reset_counters** (self)
- def **set_gbt_x_forward_to_usb** (self, value, commitTransaction=True)
- def **get_gbt_x_forward_to_usb** (self)
- def **get_adc_names** (self)
- def **read_adcs** (self)
- def **read_adcs_conv** (self)
- def **dump_config** (self)
- def **check_git_hash_and_date** (self, expected_git_hash=None)
- def **get_dna_value** (self, verbose)
- def **get_microprocessor_counters** (self, reg_num, reg_wid=32)

Public Attributes

- **logger**
- **comm**
- **wait_module**
- **status**
- **master_monitor**
- **sca**
- **pa3**

Static Public Attributes

- int **CRU_OFFSET** = 0x40
- int **MASTER_MONITOR_MODULE** = CRU_OFFSET + 0
- int **STATUS_MODULE** = CRU_OFFSET + 1
- int **SCA_MODULE** = CRU_OFFSET + 2
- int **GBT_FPGA_MODULE** = CRU_OFFSET + 3
- int **WAIT_MODULE** = CRU_OFFSET + 4
- list **SCA_ADC_COUNTERS**

5.96.1 Detailed Description

Board implementation for Readout unit

5.96.2 Member Function Documentation

5.96.2.1 check_git_hash_and_date()

```
def ru_board.RUv0_CRU.check_git_hash_and_date (
    self,
    expected_git_hash = None )
```

gets git hash and fw date

5.96.2.2 get_dna_value()

```
def ru_board.RUv0_CRU.get_dna_value (
    self,
    verbose )
```

Gets the FPGA DNA value

5.96.2.3 get_microprocessor_counters()

```
def ru_board.RUv0_CRU.get_microprocessor_counters (
    self,
    reg_num,
    reg_wid = 32 )
```

Gets the register values from SPI

5.96.2.4 read_gbtx_data_from_usb()

```
def ru_board.RUv0_CRU.read_gbtx_data_from_usb (
    self,
    size,
    rawoutfile = None )
```

size is expressed in gbt packages

5.96.2.5 send_end_of_continuous()

```
def ru_board.RUv0_CRU.send_end_of_continuous (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a EOC trigger

5.96.2.6 send_end_of_triggered()

```
def ru_board.RUv0_CRU.send_end_of_triggered (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a EOT trigger

5.96.2.7 send_heartbeat()

```
def ru_board.RUv0_CRU.send_heartbeat (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a SOC trigger

5.96.2.8 send_start_of_continuous()

```
def ru_board.RUv0_CRU.send_start_of_continuous (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a SOC trigger

5.96.2.9 send_start_of_triggered()

```
def ru_board.RUv0_CRU.send_start_of_triggered (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a SOT trigger

5.96.2.10 send_trigger()

```
def ru_board.RUv0_CRU.send_trigger (
    self,
    triggerType = 0x10,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a trigger to the RUv1 through the GBTx

NOTE: orbit is 32 bit in ITS upgrade but is limited to 31 bits for the CRU design

5.96.2.11 wait()

```
def ru_board.RUv0_CRU.wait (
    self,
    wait_value,
    commitTransaction = True )
```

Implements the wait function of the dctrl wishbone slave

5.96.2.12 write_gbtx()

```
def ru_board.RUv0_CRU.write_gbtx (
    self,
    module,
    address,
    data,
    commitTransaction = True )
```

Writes to the gbtx interface on the RDO board

5.96.3 Member Data Documentation

5.96.3.1 SCA_ADC_COUNTERS

```
list ru_board.RUv0_CRU.SCA_ADC_COUNTERS [static]
```

Initial value:

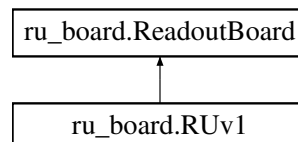
```
= [
    (0x00, "I_MGT"),
    (0x01, "I_INT"),
    (0x02, "I_1V2"),
    (0x03, "I_1V5"),
    (0x04, "I_1V8"),
    (0x05, "I_2V5"),
    (0x06, "I_3V3"),
    (0x07, "I_IN "),
    (0x08, "V_MGT"),
    (0x09, "V_INT"),
    (0x0A, "V_1V2"),
    (0x0B, "V_1V5"),
    (0x0C, "V_1V8"),
    (0x0D, "V_2V5"),
    (0x0E, "V_3V3"),
    (0x0F, "V_IN "),
    (0x17, "T1  "),
    (0x18, "T2  ")
]
```

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_board.py

5.97 ru_board.RUv1 Class Reference

Inheritance diagram for ru_board.RUv1:



Public Member Functions

- `def __init__(self, comm)`
- `def initialize(self)`
- `def gbt0_update_delay(self, delay=0x8)`
- `def set_chip2connector_lut(self, lut)`
- `def get_chip2connector_lut(self)`
- `def enable_chip2connector_lut(self, enable=True)`
- `def write_chip_reg(self, address, data, chipid, commitTransaction=True, readback=False)`
- `def read_chip_reg(self, address, chipid, commitTransaction=True)`
- `def write_chip_opcode(self, opcode, commitTransaction=True)`
- `def power_off_chip(self)`
- `def power_on_chip(self, avdd=1.8, dvdd=1.8, backbias=None)`
- `def set_chip_voltage(self, avdd=1.8, dvdd=1.8, backbias=None)`
- `def log_currents(self)`
- `def wait(self, wait_value, commitTransaction=True)`
- `def dump_config(self)`
- `def check_git_hash_and_date(self, expected_git_hash=None)`
- `def get_dna_value(self)`

Public Attributes

- **logger**
- **comm**
- **drp_bridge**
- **gth**
- **datapathmon**
- **gpio**
- **datapathmon_gpio**
- **status**
- **master_monitor**
- **wait_module**
- **dctrl**
- **i2c_gbtx**
- **gbtx01**
- **gbtx2**
- **radmon**
- **sysmon**
- **gbtx_flow_monitor**
- **master_usb**
- **usb_if**
- **gbt_packer_gth**
- **gbt_packer_gpio**
- **gbt_packer_monitor_gth**
- **gbt_packer_monitor_gpio**
- **trigger_handler**
- **trigger_handler_monitor**
- **powerunit_1**
- **powerunit_2**
- **wb2fifo**
- **chip2connector_lut**
- **chip2connector_enabled**

Static Public Attributes

- **int MASTER_MONITOR_MODULE = 0**
- **int STATUS_MODULE = 1**
- **int GTH_FRONTEND_MODULE = 2**
- **int DATAPATH_MONITOR_MODULE = 3**
- **int ALPIDE_MODULE = 4**
- **int I2C_GBT_MODULE = 5**
- **int I2C_PU1_MODULE = 6**
- **int I2C_PU2_MODULE = 7**
- **int GBTX0_MODULE = 8**
- **int GBTX2_MODULE = 9**
- **int WAIT_MODULE = 10**
- **int RADMON_MODULE = 11**
- **int SYSMON_MODULE = 12**
- **int GBTX_FLOW_MONITOR_MODULE = 13**
- **int USB_IF = 14**
- **int MASTER_USB = 15**
- **int TRIGGER_HANDLER = 16**
- **int TRIGGER_HANDLER_MONITOR = 17**
- **int GPIO_CONTROL = 18**

- int **DATAPATH_MONITOR_GPIO_1** = 19
- int **DATAPATH_MONITOR_GPIO_2** = 20
- int **GBT_PACKER_GTH** = 21
- int **GBT_PACKER_GPIO** = 22
- int **GTH_DRP** = 23
- int **GBT_PACKER_MONITOR_GTH** = 24
- int **GBT_PACKER_MONITOR_GPIO** = 25
- int **WB2FIFO_MODULE** = -1

5.97.1 Detailed Description

Board implementation for Readout unit

5.97.2 Member Function Documentation

5.97.2.1 `check_git_hash_and_date()`

```
def ru_board.RUv1.check_git_hash_and_date (
    self,
    expected_git_hash = None )
```

gets git hash and fw date

5.97.2.2 `enable_chip2connector_lut()`

```
def ru_board.RUv1.enable_chip2connector_lut (
    self,
    enable = True )
```

Enable/disable automatic chip to connector mapping

5.97.2.3 `get_chip2connector_lut()`

```
def ru_board.RUv1.get_chip2connector_lut (
    self )
```

Return the mapping between chip and its connector

5.97.2.4 get_dna_value()

```
def ru_board.RUv1.get_dna_value (
    self )
```

Gets the FPGA DNA value

5.97.2.5 set_chip2connector_lut()

```
def ru_board.RUv1.set_chip2connector_lut (
    self,
    lut )
```

Set the mapping between chip and its connector at the RU in form of a Dictionary {chipid:connector}

5.97.2.6 wait()

```
def ru_board.RUv1.wait (
    self,
    wait_value,
    commitTransaction = True )
```

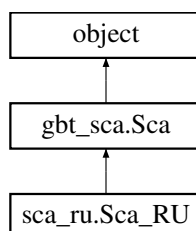
Implements the wait function of the dctrl wishbone slave

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_board.py

5.98 gbt_sca.Sca Class Reference

Inheritance diagram for gbt_sca.Sca:



Public Member Functions

- `def __init__ (self, use_adc=True, use_gpio=True, use_pa3_i2c=True, use_pa3_i2c_2=True, use_us_↵ i2c=True)`
- `def initialize (self)`
- `def initialize_gpio_channel (self)`
- `def init_communication (self)`
- `def enable_channel (self, SPI=0, GPIO=0, I2C_PA3_1=0, I2C_PA3_2=0, I2C_VTTX1=0, I2C_VTTX2=0, I2C_VTRX2=0, I2C_US=0, I2C_GBTX=0, JTAG=0, ADC=0)`
- `def initialize_i2c_channel (self, channel, speed=ScaI2cSpeed.f1MHz)`
- `def set_i2c_w_ctrl_reg (self, channel, speed, nbytes, sclmode)`
- `def initialize_adc_channel (self)`
- `def set_adc_channel (self, channel, commitTransaction=True)`
- `def read_adc_channel (self, channel)`
- `def read_gpio (self)`
- `def write_gpio (self, value)`
- `def set_gpio_direction (self, value)`
- `def set_gpio (self, index, value)`
- `def set_xcku_reset (self, value)`
- `def read_gbtx_register (self, register, gbtx=0)`
- `def write_gbtx_register (self, register, value, gbtx=0)`

Public Attributes

- `use_adc`
- `use_gpio`
- `use_pa3_i2c`
- `use_pa3_i2c_2`
- `use_us_i2c`

5.98.1 Detailed Description

Class to handle SCA transactions

5.98.2 Member Function Documentation

5.98.2.1 init_communication()

```
def gbt_sca.Sca.init_communication (
    self )
```

Implementation to initialize communication with SCA

Reimplemented in [sca_ru.Sca_RU](#).

5.98.2.2 initialize_i2c_channel()

```
def gbt_sca.Sca.initialize_i2c_channel (
    self,
    channel,
    speed = ScaI2cSpeed.f1MHz )
```

Sets the I2C transactions speed to 1MHz,
with no multi byte operation and
no scl mode

5.98.2.3 read_adc_channel()

```
def gbt_sca.Sca.read_adc_channel (
    self,
    channel )
```

Expected wait time for ADC read: 9000 cycles

5.98.2.4 read_gbt_x_register()

```
def gbt_sca.Sca.read_gbt_x_register (
    self,
    register,
    gbt_x = 0 )
```

Read GBTx register "register" from GBTx "gbtx"

5.98.2.5 set_adc_channel()

```
def gbt_sca.Sca.set_adc_channel (
    self,
    channel,
    commitTransaction = True )
```

Sets the adc channel to the correct MUX channel

5.98.2.6 set_gpio()

```
def gbt_sca.Sca.set_gpio (
    self,
    index,
    value )
```

overwrites the value in the index with the value in the SCA gpio

5.98.2.7 set_gpio_direction()

```
def gbt_sca.Sca.set_gpio_direction (
    self,
    value )
```

Sets the direction for the GPIO pin of the SCA.
value is a one-hot bit array with position n indicating GPIO n
if the bit is set to 1, the pin is an output,
if the bit is set to 0, the pin is an input

5.98.2.8 set_i2c_w_ctrl_reg()

```
def gbt_sca.Sca.set_i2c_w_ctrl_reg (
    self,
    channel,
    speed,
    nbytes,
    sclmode )
```

GBT sca manual page 23 v8.2

5.98.2.9 set_xcku_reset()

```
def gbt_sca.Sca.set_xcku_reset (
    self,
    value )
```

XCKU reset via SCA

5.98.2.10 write_gbt_x_register()

```
def gbt_sca.Sca.write_gbt_x_register (
    self,
    register,
    value,
    gbt_x = 0 )
```

Write "value" (8bit) to register "register" of GBTx "gbtx"

5.98.2.11 write_gpio()

```
def gbt_sca.Sca.write_gpio (
    self,
    value )
```

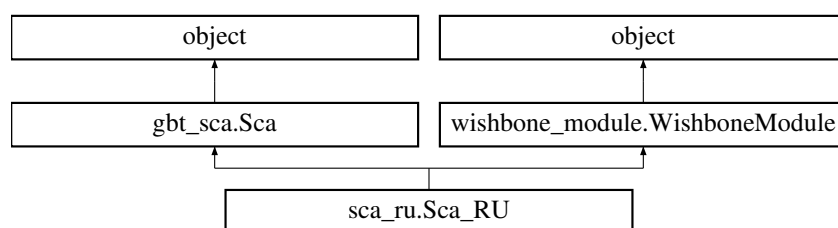
Writes GPIOs set to output

The documentation for this class was generated from the following file:

- /media/sf_proj/RUV1_Test/modules/board_support_software/software/py/gbt_sca.py

5.99 sca_ru.Sca_RU Class Reference

Inheritance diagram for sca_ru.Sca_RU:



Public Member Functions

- def **__init__** (self, moduleid, board_obj, use_adc=True, use_gpio=True, use_pa3_i2c=True, use_us_↵ i2c=True)
- def **sca_write_check_results** (self, trid=0x12)
- def **init_communication** (self)
- def **HDLC_reset** (self, commitTransaction=True)
- def **HDLC_connect** (self, commitTransaction=True)
- def **get_status_latch** (self, commitTransaction=True)
- def **clear_status_latch** (self, commitTransaction=True)

Additional Inherited Members

5.99.1 Detailed Description

Class to implement SCA GBT transactions

5.99.2 Member Function Documentation

5.99.2.1 `clear_status_latch()`

```
def sca_ru.Sca_RU.clear_status_latch (
    self,
    commitTransaction = True )
```

Reads the status
latch ignoring the return value to clear it

5.99.2.2 `get_status_latch()`

```
def sca_ru.Sca_RU.get_status_latch (
    self,
    commitTransaction = True )
```

Returns the value of the status latch,
ignore the value for clear

5.99.2.3 `init_communication()`

```
def sca_ru.Sca_RU.init_communication (
    self )
```

Implementation to initialize communication with SCA

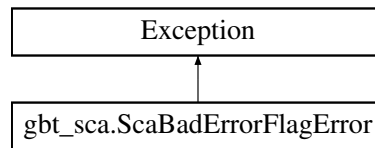
Reimplemented from [gbt_sca.Sca](#).

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/sca_ru.py`

5.100 gbt_sca.ScaBadErrorFlagError Class Reference

Inheritance diagram for gbt_sca.ScaBadErrorFlagError:



Public Member Functions

- `def __init__ (self, status=None)`
- `def __sts__ (self)`
- `def __srt__ (self)`
- `def get_error_message (self)`
- `def log_info (self)`

Public Attributes

- `logger`
- `status`
- `error_message`

5.100.1 Detailed Description

basic class to define an SCA error flag exception

5.100.2 Member Function Documentation

5.100.2.1 log_info()

```
def gbt_sca.ScaBadErrorFlagError.log_info (  
    self )
```

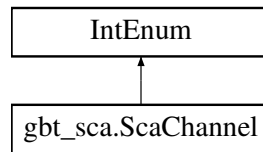
returns the args of the exception

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py`

5.101 gbt_sca.ScaChannel Class Reference

Inheritance diagram for gbt_sca.ScaChannel:



Static Public Attributes

- int **CTRL** = 0x00
- int **SPI** = 0x01
- int **GPIO** = 0x02
- int **I2C0** = 0x03
- int **I2C1** = 0x04
- int **I2C2** = 0x05
- int **I2C3** = 0x06
- int **I2C4** = 0x07
- int **I2C5** = 0x08
- int **I2C6** = 0x09
- int **I2C7** = 0x0A
- int **I2C8** = 0x0B
- int **I2C9** = 0x0C
- int **I2CA** = 0x0D
- int **I2CB** = 0x0E
- int **I2CC** = 0x0F
- int **I2CD** = 0x10
- int **I2CE** = 0x11
- int **I2CF** = 0x12
- int **JTAG** = 0x13
- int **ADC** = 0x14
- int **DAC** = 0x15

5.101.1 Detailed Description

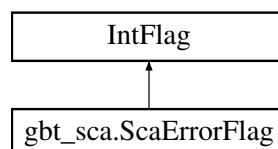
Copied from p17 of the GBT-SCA-UserManual.pdf v8.2
(<https://espace.cern.ch/GBT-Project/GBT-SCA/Manuals/Forms/AllItems.aspx>)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py

5.102 gbt_sca.ScaErrorFlag Class Reference

Inheritance diagram for gbt_sca.ScaErrorFlag:



Static Public Attributes

- int **GENERIC_ERROR_FLAG** = 0
- int **INVALID_CHANNEL_REQUEST** = 1
- int **INVALID_COMMAND_REQUEST** = 2
- int **INVALID_TRANSACTION_REQUEST** = 3
- int **INVALID_LENGTH** = 4
- int **CHANNEL_NOT_ENABLED** = 5
- int **CHANNEL_CURRENTLY_BUSY** = 6
- int **COMMAND_IN_TREATMENT** = 7
- int **NUM_BITS** = 8

5.102.1 Detailed Description

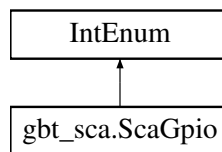
Sca error flag bit mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py

5.103 gbt_sca.ScaGpio Class Reference

Inheritance diagram for gbt_sca.ScaGpio:



Static Public Attributes

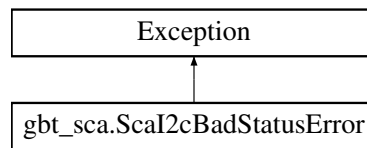
- int **XCKU_TMR_RESET** = 0
- int **XCKU_TMR_RESERVED1** = 1
- int **XCKU_TMR_RESERVED2** = 2
- int **XCKU_TMR_RESERVED3** = 3
- int **XCKU_RESERVED4** = 4
- int **XCKU_RESERVED5** = 5
- int **XCKU_RESERVED6** = 6
- int **XCKU_RESERVED7** = 7
- int **XCKU_RESERVED8** = 8
- int **XCKU_RESERVED9** = 9
- int **XCKU_RESERVED10** = 10
- int **XCKU_RESERVED11** = 11
- int **PA3_START_PROGRAM** = 12
- int **PA3_PROGRAM_DONE** = 13
- int **PA3_RESET** = 14
- int **PA3_ALIVE_COUNTER_b0** = 15
- int **PA3_ALIVE_COUNTER_b1** = 16
- int **PA3_ALIVE_COUNTER_b2** = 17
- int **PA3_ALIVE_COUNTER_b3** = 18
- int **PA3_ALIVE_COUNTER_b4** = 19
- int **SCA_GPIO_NUM** = 20

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py

5.104 gbt_sca.ScaI2cBadStatusError Class Reference

Inheritance diagram for gbt_sca.ScaI2cBadStatusError:



Public Member Functions

- `def __init__(self, status=None)`
- `def __sts__(self)`
- `def __srt__(self)`
- `def get_error_message(self)`
- `def log_info(self)`

Public Attributes

- `logger`
- `status`
- `error_message`

5.104.1 Detailed Description

basic class to define an SCA I2C read error exception

5.104.2 Member Function Documentation

5.104.2.1 log_info()

```
def gbt_sca.ScaI2cBadStatusError.log_info (  
    self )
```

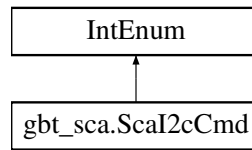
returns the args of the exception

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py`

5.105 gbt_sca.Scal2cCmd Class Reference

Inheritance diagram for gbt_sca.Scal2cCmd:



Static Public Attributes

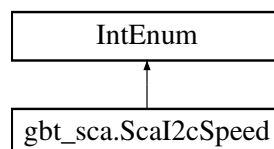
- int I2C_W_CTRL = 0x30
- int I2C_R_CTRL = 0x31
- int I2C_R_STR = 0x11
- int I2C_W_MSK = 0x20
- int I2C_R_MSK = 0x21
- int I2C_W_DATA0 = 0x40
- int I2C_R_DATA0 = 0x41
- int I2C_W_DATA1 = 0x50
- int I2C_R_DATA1 = 0x51
- int I2C_W_DATA2 = 0x60
- int I2C_R_DATA2 = 0x61
- int I2C_W_DATA3 = 0x70
- int I2C_R_DATA3 = 0x71
- int I2C_S_7B_W = 0x82
- int I2C_S_7B_R = 0x86
- int I2C_S_10B_W = 0x8A
- int I2C_S_10B_R = 0x8E
- int I2C_M_7B_W = 0xDA
- int I2C_M_7B_R = 0xDE
- int I2C_M_10B_W = 0xE2
- int I2C_M_10B_R = 0xE6
- int I2C_RMW_OR = 0xC6
- int I2C_RMW_XOR = 0xCA

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py

5.106 gbt_sca.Scal2cSpeed Class Reference

Inheritance diagram for gbt_sca.Scal2cSpeed:



Static Public Attributes

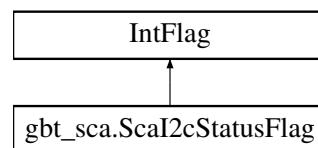
- int **f100kHz** = 0
- int **f200kHz** = 1
- int **f400kHz** = 2
- int **f1MHz** = 3

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py

5.107 gbt_sca.ScaI2cStatusFlag Class Reference

Inheritance diagram for gbt_sca.ScaI2cStatusFlag:



Static Public Attributes

- int **RESERVED0** = 0
- int **RESERVED1** = 1
- int **SUCC** = 2
- int **LEVERR** = 3
- int **EMPTY4** = 4
- int **INVCOM** = 5
- int **NOACK** = 6
- int **EMPTY7** = 7
- int **NUM_BITS** = 8

5.107.1 Detailed Description

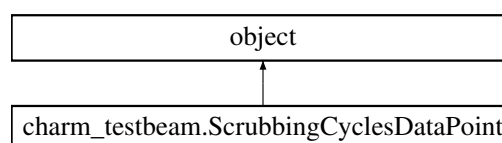
SCA I2C status bit mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbt_sca.py

5.108 charm_testbeam.ScrubbingCyclesDataPoint Class Reference

Inheritance diagram for charm_testbeam.ScrubbingCyclesDataPoint:



Public Member Functions

- `def __init__(self)`

Public Attributes

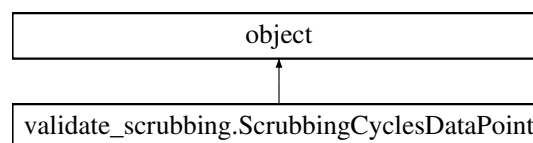
- **start_timestamp**
- **stop_timestamp**

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/software/py/charm_testbeam.py`

5.109 validate_scrubbing.ScrubbingCyclesDataPoint Class Reference

Inheritance diagram for `validate_scrubbing.ScrubbingCyclesDataPoint`:



Public Member Functions

- `def __init__(self)`

Public Attributes

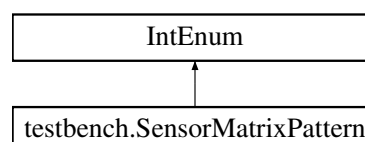
- **start_timestamp**
- **stop_timestamp**

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/software/py/validate_scrubbing.py`

5.110 testbench.SensorMatrixPattern Class Reference

Inheritance diagram for `testbench.SensorMatrixPattern`:



Static Public Attributes

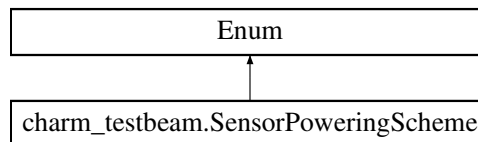
- int **EMPTY** = 0
- int **IMAGE** = 1

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/testbench.py

5.111 charm_testbeam.SensorPoweringScheme Class Reference

Inheritance diagram for charm_testbeam.SensorPoweringScheme:



Static Public Attributes

- int **HAMEG** = 0
- int **POWERUNIT** = 1

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam.py

5.112 simulation_if.SimulationServer Class Reference

Public Member Functions

- def **__init__** (self)
- def **run** (self)
- def **send_close** (self)
- def **stop** (self)
- def **start** (self)

Public Attributes

- **server**
- **thrd**
- **stop_server**

Static Public Attributes

- bool **stop_server** = False

5.112.1 Detailed Description

RPC server handling communication between Python and Systemverilog

5.112.2 Member Function Documentation

5.112.2.1 start()

```
def simulation_if.SimulationServer.start (
    self )
```

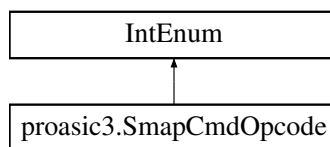
Start RPC server

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/simulation_if.py

5.113 proasic3.SmapCmdOpcode Class Reference

Inheritance diagram for proasic3.SmapCmdOpcode:



Static Public Attributes

- int **INIT_XILINX** = 0x01
- int **STARTUP** = 0x02
- int **WRITE_ONE_BYTE** = 0x04
- int **READ_ONE_BYTE** = 0x08
- int **ABORT** = 0x10
- int **WR_FINISHED** = 0x20
- int **CLEAR_STATUS_REGISTER** = 0x40

5.113.1 Detailed Description

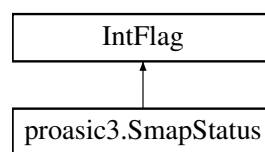
Smmap controller opcode mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.114 proasic3.SmapStatus Class Reference

Inheritance diagram for proasic3.SmapStatus:



Static Public Attributes

- int **INIT_B_DID_NOT_RESPOND_TO_PROG_B_EQ_0** = 0
- int **INIT_B_DID_NOT_GO_HIGH_AFTER_PROG_B** = 1
- int **DONE_NEVER_GOES_HIGH** = 2
- int **XILINX_DONE_PIN** = 3
- int **IF_BUSY** = 4

5.114.1 Detailed Description

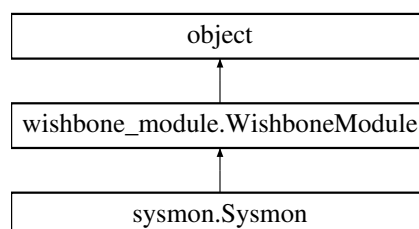
Smmap controller Status Register

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/proasic3.py

5.115 sysmon.Sysmon Class Reference

Inheritance diagram for sysmon.Sysmon:



Public Member Functions

- `def __init__ (self, moduleid, board_obj)`
- `def set_data (self, address, value, commitTransaction=True)`
- `def get_data (self, address, commitTransaction=True)`
- `def set_drp_address (self, value, commitTransaction=True)`
- `def get_drp_data (self, commitTransaction=True)`
- `def set_drp_data (self, value, commitTransaction=True)`
- `def get_temperature (self)`
- `def get_vcc_int (self)`
- `def get_vcc_aux (self)`
- `def get_vref_p (self)`
- `def get_vref_n (self)`
- `def get_vcc_bram (self)`
- `def disable_ot_protection (self, commitTransaction=True)`
- `def enable_ot_protection (self, commitTransaction=True)`
- `def get_ot_status (self)`
- `def get_tmr_mismatch (self, commitTransaction=True)`
- `def get_vcc_alpide_3v3 (self)`
- `def get_vcc_sca_1v5 (self)`
- `def get_mismatch (self)`
- `def dump_config (self)`

Public Attributes

- `verbose`

5.115.1 Detailed Description

System Monitor Wishbone Slave

5.115.2 Member Function Documentation

5.115.2.1 dump_config()

```
def sysmon.Sysmon.dump_config (  
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.115.2.2 `get_drp_data()`

```
def sysmon.Sysmon.get_drp_data (
    self,
    commitTransaction = True )
```

Returns data present in the DATA Register

5.115.2.3 `get_mismatch()`

```
def sysmon.Sysmon.get_mismatch (
    self )
```

gets the actual mismatch status

5.115.2.4 `set_drp_address()`

```
def sysmon.Sysmon.set_drp_address (
    self,
    value,
    commitTransaction = True )
```

Set the ADDR Register for next transaction

5.115.2.5 `set_drp_data()`

```
def sysmon.Sysmon.set_drp_data (
    self,
    value,
    commitTransaction = True )
```

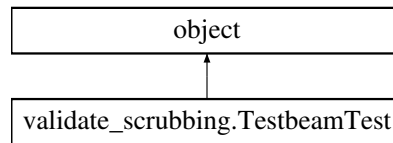
Set the DATA Register

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/sysmon.py`

5.116 validate_scrubbing.TestbeamTest Class Reference

Inheritance diagram for validate_scrubbing.TestbeamTest:



Public Member Functions

- def **__init__** (self)
- def **setup_power** (self)
- def **poweron_RUv0** (self, force_powercycle=False)
- def **poweroff_RUv0** (self)
- def **connect_jcm** (self)
- def **readback_jcm** (self, readback_name, empty_folder=False, backup_readback=False)
- def **analyse_readback_files** (self)
- def **restore_readback_file** (self)
- def **single_scrub_cycle** (self)
- def **program_RUv1** (self)
- def **store_configuration** (self)
- def **on_test_stop** (self)
- def **check_powerstate_RUv1** (self)
- def **powercycle_RUv1** (self)
- def **powercycle_stave** (self)
- def **check_power** (self)
- def **setup_comms** (self)
- def **tearDown** (self)
- def **cru_reads** (self, dp)
- def **rdo_reads** (self, dp)
 - RDO radiation monitor instantaneous values.*
- def **read_values** (self, recordPrefetch=True)
- def **save_datapoint** (self, dp)
- def **stop** (self)
- def **check_status** (self)
- def **setup_logging** (self)
- def **test_routine** (self)
- def **log_powersupply_values** (self)

Public Attributes

- **hameg**
- **testbench**
- **logger**
- **logdir**
- **datafile**
- **datafilepath**
- **sequence_cru**
- **sequence_rdo**

- `jcm_control_client`
- `faults_injected`
- `manual_scrubbing_info`
- `tot_read_counter`
- `pre_injection_file`
- `after_injection_file`
- `after_scrubbing_file`
- `readback_bkp`
- `testrun_exit_status_info`
- `runlog`
- `run_logger`
- `start_time`

5.116.1 Member Function Documentation

5.116.1.1 `check_status()`

```
def validate_scrubbing.TestbeamTest.check_status (
    self )
```

Check status of run. Decide if the run should stop

5.116.1.2 `readback_jcm()`

```
def validate_scrubbing.TestbeamTest.readback_jcm (
    self,
    readback_name,
    empty_folder = False,
    backup_readback = False )
```

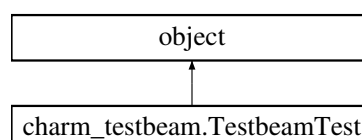
Stores the readback files in the `/tmp/read_data_<readback_name>.rdb`

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/software/py/validate_scrubbing.py`

5.117 `charm_testbeam.TestbeamTest` Class Reference

Inheritance diagram for `charm_testbeam.TestbeamTest`:



Public Member Functions

- def **__init__** (self)
 - def **setup_power** (self)
 - def **setup_sensors_power_source_on_hameg** (self)
 - def **setup_powerunit** (self)
 - def **poweron_RUv0** (self, force_powercycle=False)
 - def **poweroff_RUv0** (self)
 - def **connect_jcm** (self)
 - def **setup_beam_shutter** (self)
 - def **beam_on** (self)
 - def **beam_off** (self)
 - def **program_RUv1** (self)
 - def **store_configuration** (self)
 - def **on_test_start** (self)
 - def **on_test_stop** (self)
 - def **manual_single_scrub** (self)
 - def **start_random_fault_injection** (self, delay, nr_faults, delay_after, make_correction=True)
 - def **stop_random_fault_injection** (self)
 - def **clean_readbacks_in_jcm** (self)
 - def **readback_xcku_configuration** (self, basename="", max_tries=10)
 - def **compare_xcku_readbacks** (self, basename1, basename2)
 - def **log_jcm_readback_status** (self, are_equal)
 - def **jcm_download_readback_files** (self)
 - def **check_powerstate_RUv1** (self)
 - def **powercycle_RUv1** (self)
 - def **check_powerstate_powerunit** (self)
 - def **powercycle_powerunit** (self)
 - def **force_cut_sensor_power** (self)
 - def **poweroff_powerunit** (self)
 - def **poweroff_stave** (self)
 - def **powercycle_stave** (self)
 - def **check_power** (self)
 - def **setup_comms** (self)
 - def **start_datataking** (self)
 - def **tearDown** (self)
 - def **sensor_reads** (self, dp)
 - def **cru_reads** (self, dp)
 - def **rdo_reads** (self, dp)
- RDO radiation monitor instantaneous values.*
- def **print_values** (self, last_read, dp)
 - def **read_values** (self, recordPrefetch=False)
 - def **save_datapoint** (self, dp)
 - def **readback_drp** (self, filename)
 - def **stop** (self)
 - def **send_stimuli** (self)
 - def **check_status** (self)
 - def **setup_logging** (self)
 - def **test_routine** (self)
 - def **return_exit_status_over_tg** (self, exitstatus)
 - def **handle_failure** (self, run_error)
 - def **log_powersupply_values** (self)

Public Attributes

- hameg
- testbench
- logger
- logdir
- dump_process
- read_process
- datafile
- datafilepath
- sequence_cru
- sequence_rdo
- jcm_control_client
- faults_injected
- beam_shutter
- beam_shutter_info
- manual_scrubbing_info
- tot_read_counter
- testrun_exit_status_info
- runlog
- run_logger
- start_time

5.117.1 Member Function Documentation

5.117.1.1 check_status()

```
def charm_testbeam.TestbeamTest.check_status (
    self )
```

Check status of run. Decide if the run should stop

5.117.1.2 manual_single_scrub()

```
def charm_testbeam.TestbeamTest.manual_single_scrub (
    self )
```

manual, slow scrubbing routine

5.117.1.3 send_stimuli()

```
def charm_testbeam.TestbeamTest.send_stimuli (
    self )
```

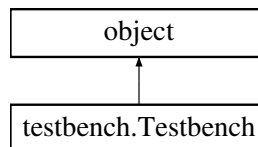
In main loop, send stimuli to board (run each time main loop)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam.py

5.118 testbench.Testbench Class Reference

Inheritance diagram for testbench.Testbench:



Public Member Functions

- def **__init__** (self, use_usb_comm=USE_USB_COMM)
- def **tg_notification** (self, message)
- def **version** (self, get_pa3=False)
- def **setup_standalone** (self)
- def **setup_logging** (self)
- def **setup_comms** (self, cru_ctlOnly=False)
- def **setup_comm_cru** (self, ctlOnly=False)
- def **setup_comm_rdo** (self, ctlOnly=True)
- def **stop** (self)
- def **setup_cru** (self)
- def **setup_rdo** (self, connector_nr=4)
- def **set_dctrl_connector** (self, connector_nr, force=False)
- def **setup_powerunit** (self, i2c_connector_on_ruv1=1)
- def **setup_boards** (self)
- def **initialize_boards** (self)
- def **initialize_for_regression_UT** (self)
- def **initialize_for_regression** (self)
- def **gpio_subset** (self, transceivers)
- def **gth_subset** (self, transceivers)
- def **gbt_packer_clean_fifos** (self)
- def **test_readout_gpio_setup_sensors** (self, connector=0, scan_idelay=False)
- def **test_readout_gpio_routine** (self, connector=0, scan_idelay=False, nr_triggers=10)
- def **setup_sensors** (self, enable_strobe_generation=0, LinkSpeed=3, disable_manchester=1, pattern=Sensor↔MatrixPattern.EMPTY)
- def **setup_sensor_matrix** (self, ch, pattern=SensorMatrixPattern.EMPTY)
- def **setup_sensor_matrix_image** (self, ch, img_path)

- def **setup_pulser** (self, [chip](#)=0x8)
- def **clear_pulser** (self)
- def **setup_prbs_test** (self, chips)
- def **setup_prbs_test_gpio** (self, SENSORS)
- def **scan_idelay_gpio** (self, stepsize=10, waittime=1, set_optimum=True)
- def **chips_propagate_clock** (self, first=0, last=8)
- def **reset_counters** (self, commitTransaction=True)
- def **setup_readout** (self, max_retries=10)
- def **setup_readout_gpio** (self)
- def **setup_trigger_handler_continuous** (self, trigger_frequency=10000, send_pulses=False)
- def [chip](#) (self, nr)
- def [test_chips](#) (self, nrfirst=0, nrlast=8, nrtests=100, verbose=True)
- def [test_chips_fast](#) (self, nrfirst=0, nrlast=8, nrtests=100)
- def [test_chips_continuous](#) (self, nrfirst=0, nrlast=8)
- def **test_dclk_phases** (self, nrtests=1000, connector=None, manchester_tx_en=False)
- def **test_prbs** (self, runtime=10)
- def **test_prbs_gpio** (self, runtime=10)
- def **test_readout** (self, nr_triggers=10, dump_data=False)
- def **test_readout_gpio** (self, nr_triggers=10, dump_data=False)
- def **test_usb_performance_read** (self, packets_per_train=1000, nr_trains=1000, test_cru=True, test_↵
rdo=True)
- def **test_usb_performance_write** (self, packets_per_train=1000, nr_trains=1000, test_cru=True, test_↵
rdo=True)
- def **check_event_readout** (self, nr_events, nr_empty_triggers, lanes, verbose=False, raw_data_file=None)
- def **test_usb_endurance** (self)
- def **monitor_gbtx_lock_loss** (self, intervall=600, Test=False)
- def **test_usb_rdo_read** (self)
- def **test_pa3_endurance** (self)
- def **test_pa3_read_error_rate** (self)
- def **test_pa3_write_error_rate** (self)
- def **test_pa3_dump_config** (self)
- def **test_pa3_dump_config_2_i2c** (self)
- def **test_sca_endurance** (self)
- def **test_microprocessor_arch** (self)
- def **calibrate_gbtx01_bitslips** (self)

Public Attributes

- **logger**
- **serv**
- **comm_cru**
- **comm_rdo**
- **cru**
- **rdo**
- **powerunit**
- **use_usb_comm**
- **chips**
- **logdir**

5.118.1 Detailed Description

Testbench for executing different tests

5.118.2 Member Function Documentation

5.118.2.1 chip()

```
def testbench.Testbench.chip (
    self,
    nr )
```

Return a chip with given number

5.118.2.2 setup_boards()

```
def testbench.Testbench.setup_boards (
    self )
```

Setup board classes

5.118.2.3 setup_comm_cru()

```
def testbench.Testbench.setup_comm_cru (
    self,
    ctlOnly = False )
```

Setup Communication interfaces for CRU

5.118.2.4 setup_comm_rdo()

```
def testbench.Testbench.setup_comm_rdo (
    self,
    ctlOnly = True )
```

Setup Communication interfaces for RDO

5.118.2.5 setup_comms()

```
def testbench.Testbench.setup_comms (
    self,
    cru_ctlOnly = False )
```

Setup Communication interfaces for CRU and RDO

5.118.2.6 test_chips()

```
def testbench.Testbench.test_chips (
    self,
    nrfirst = 0,
    nrlast = 8,
    nrtests = 100,
    verbose = True )
```

Perform a chip test on chips [first,last]

5.118.2.7 test_chips_continuous()

```
def testbench.Testbench.test_chips_continuous (
    self,
    nrfirst = 0,
    nrlast = 8 )
```

Perform a chip test on chips [first,last]

5.118.2.8 test_chips_fast()

```
def testbench.Testbench.test_chips_fast (
    self,
    nrfirst = 0,
    nrlast = 8,
    nrtests = 100 )
```

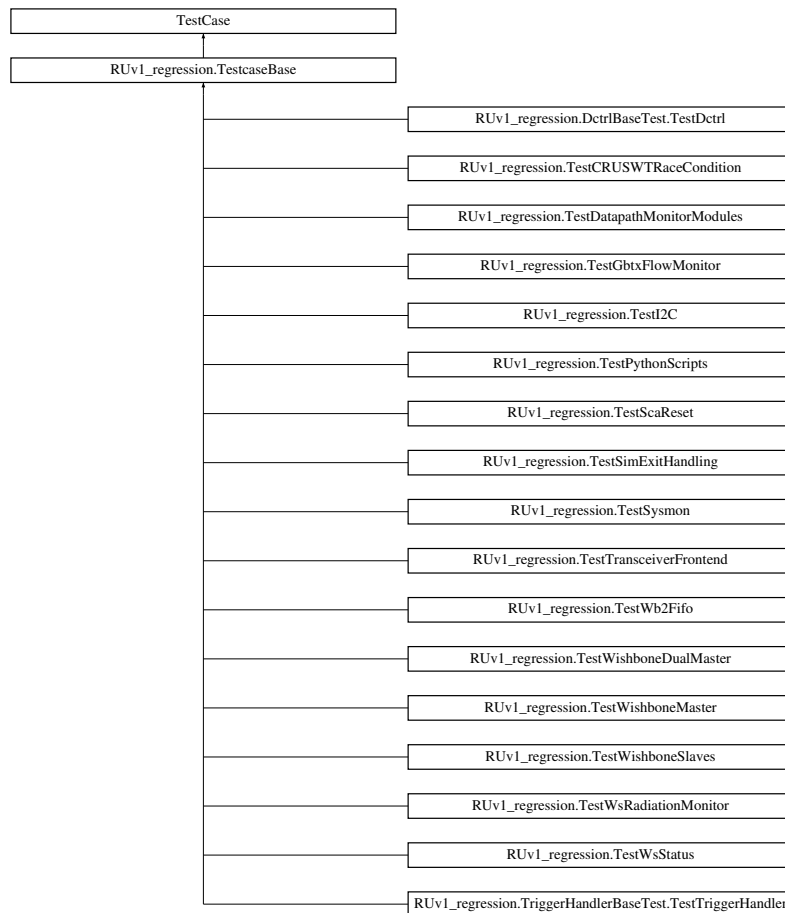
Perform a chip test on chips [first,last]

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/testbench.py

5.119 RUv1_regression.TestcaseBase Class Reference

Inheritance diagram for RUv1_regression.TestcaseBase:



Public Member Functions

- def **setUp** (self)
- def **setup_connection_lut** (self)
- def **tearDown** (self)
- def **send_trigger** (self, triggerType=0x10, bc=0xabc, orbit=0x43215678, force_cru=False, commit↵Transaction=True)
- def **send_idle** (self, value=1, commitTransaction=False)
- def **send_start_of_triggered** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_end_of_triggered** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_start_of_continuous** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **send_end_of_continuous** (self, bc=0xabc, orbit=0x43215678, commitTransaction=True)
- def **sync** (self)

Public Attributes

- **board**
- **board_usb**
- **cru**
- **ltu**

Static Public Attributes

- **connector** = None

5.119.1 Member Function Documentation

5.119.1.1 send_end_of_continuous()

```
def RUv1_regression.TestcaseBase.send_end_of_continuous (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a EOC trigger

5.119.1.2 send_end_of_triggered()

```
def RUv1_regression.TestcaseBase.send_end_of_triggered (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a EOT trigger

5.119.1.3 send_idle()

```
def RUv1_regression.TestcaseBase.send_idle (
    self,
    value = 1,
    commitTransaction = False )
```

waits for 25 ns in sim or hw

NOTE: commit transaction is false by default

5.119.1.4 send_start_of_continuous()

```
def RUv1_regression.TestcaseBase.send_start_of_continuous (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a SOC trigger

5.119.1.5 send_start_of_triggered()

```
def RUv1_regression.TestcaseBase.send_start_of_triggered (
    self,
    bc = 0xabc,
    orbit = 0x43215678,
    commitTransaction = True )
```

Sends a SOT trigger

5.119.1.6 sync()

```
def RUv1_regression.TestcaseBase.sync (
    self )
```

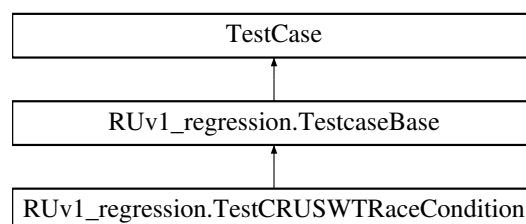
Reads a board register to align simulator and software

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.120 RUv1_regression.TestCRUSWTRaceCondition Class Reference

Inheritance diagram for RUv1_regression.TestCRUSWTRaceCondition:



Public Member Functions

- def **setUp** (self)
- def **tearDown** (self)
- def **test_set_dctrl_mask** (self)
- def **test_set_dclk_mask** (self)
- def **test_gbt2_idelay** (self)

Public Attributes

- **logger**

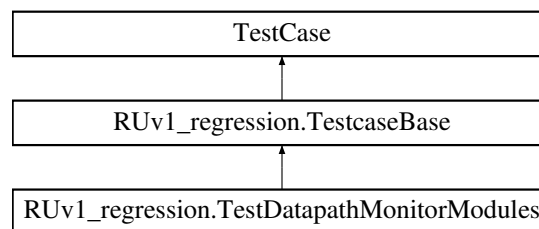
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.121 RUv1_regression.TestDatapathMonitorModules Class Reference

Inheritance diagram for RUv1_regression.TestDatapathMonitorModules:



Public Member Functions

- def **test_dpmon_gth_all** (self)
- def **test_dpmon_gpio_all** (self)
- def **test_dpmon_gth_lane** (self)
- def **test_dpmon_gpio_lane_first** (self)
- def **test_dpmon_gpio_lane_last** (self)

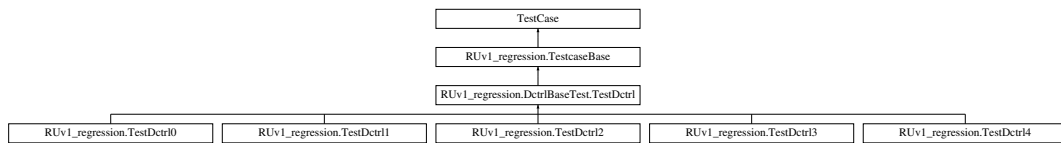
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.122 RUv1_regression.DctrlBaseTest.TestDctrl Class Reference

Inheritance diagram for RUv1_regression.DctrlBaseTest.TestDctrl:



Public Member Functions

- def **configure_test** (self, connector, chipid)
- def **setUp** (self)
- def **tearDown** (self)
- def **test_dctrl_dac_range** (self)
- def **test_manchester_tx_on_rx_on** (self)
- def **test_manchester_tx_on_rx_off** (self)
- def **test_manchester_tx_off_rx_on** (self)
- def **test_manchester_tx_off_rx_off** (self)
- def **test_idelay** (self)
- def **test_idelays_readback** (self)
- def **test_multiple_reads_rx_manchester** (self)
- def **test_multiple_reads_rx_no_manchester** (self)
- def **test_dclk_phases** (self)
- def **test_CMU_error_status** (self)
- def **test_counters** (self)
- def [test_readback_wait_cycles](#) (self)

Public Attributes

- **connector**
- **chipid**
- **ch**

Additional Inherited Members

5.122.1 Member Function Documentation

5.122.1.1 test_readback_wait_cycles()

```
def RUv1_regression.DctrlBaseTest.TestDctrl.test_readback_wait_cycles (
    self )
```

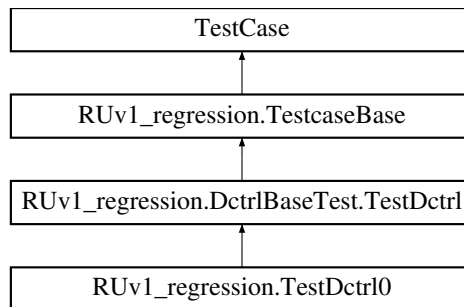
Test readback of initial wait register

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.123 RUv1_regression.TestDctrl0 Class Reference

Inheritance diagram for RUv1_regression.TestDctrl0:



Public Member Functions

- def **setUp** (self)

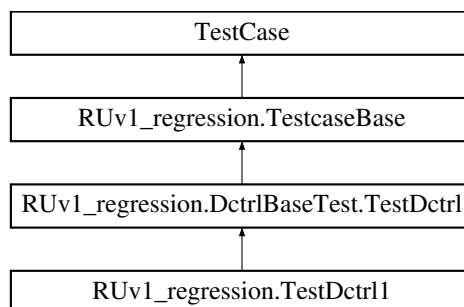
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.124 RUv1_regression.TestDctrl1 Class Reference

Inheritance diagram for RUv1_regression.TestDctrl1:



Public Member Functions

- def **setUp** (self)

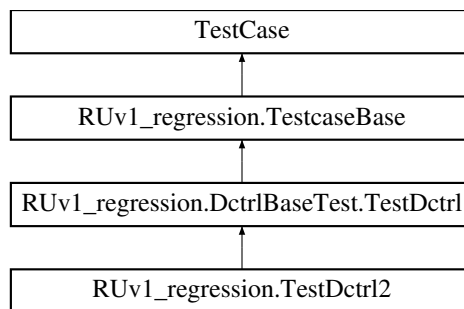
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.125 RUv1_regression.TestDctrl2 Class Reference

Inheritance diagram for RUv1_regression.TestDctrl2:



Public Member Functions

- def **setUp** (self)

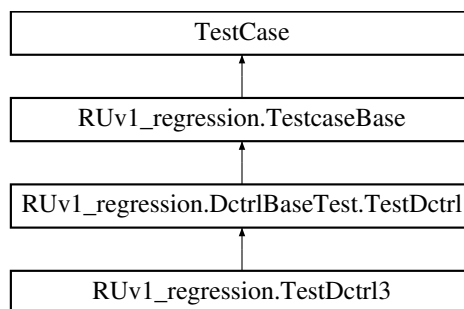
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.126 RUv1_regression.TestDctrl3 Class Reference

Inheritance diagram for RUv1_regression.TestDctrl3:



Public Member Functions

- def **setUp** (self)

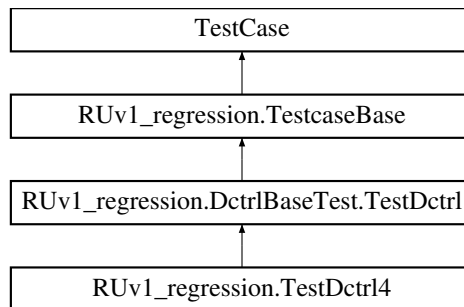
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.127 RUv1_regression.TestDctrl4 Class Reference

Inheritance diagram for RUv1_regression.TestDctrl4:



Public Member Functions

- def **setUp** (self)

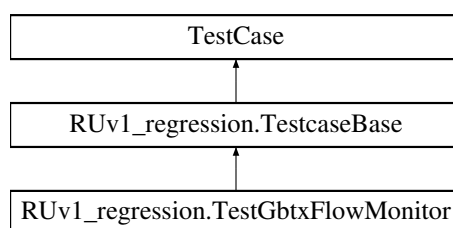
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.128 RUv1_regression.TestGbtxFLOWMonitor Class Reference

Inheritance diagram for RUv1_regression.TestGbtxFLOWMonitor:



Public Member Functions

- def **setUp** (self)
- def **test_swt** (self)
- def **test_trigger** (self)

Public Attributes

- **chips**

Additional Inherited Members

5.128.1 Detailed Description

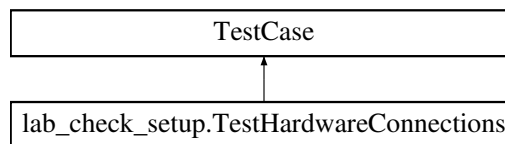
Class to verify the behaviour of the gbtX flow monitor wishbone slave

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.129 lab_check_setup.TestHardwareConnections Class Reference

Inheritance diagram for lab_check_setup.TestHardwareConnections:



Public Member Functions

- def **setUp** (self)
- def **tearDown** (self)
- def **test_board_connection** (self)
- def **single_sensor_dctr** (self, sensor=0)
- def **test_single_sensor_dctr0** (self)
- def **test_single_sensor_dctr1** (self)
- def **test_single_sensor_dctr2** (self)
- def **test_single_sensor_dctr3** (self)
- def **test_single_sensor_dctr4** (self)
- def **test_single_sensor_dctr5** (self)
- def **test_single_sensor_dctr6** (self)
- def **test_single_sensor_dctr7** (self)
- def **test_single_sensor_dctr8** (self)
- def **test_sensor_connection** (self)

Public Attributes

- **cru**
- **rdo**
- **logger**

5.129.1 Detailed Description

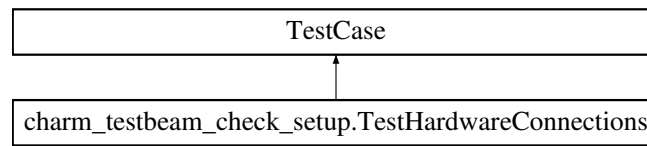
Test to run for verifying proper connections between all devices

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/lab_check_setup.py

5.130 charm_testbeam_check_setup.TestHardwareConnections Class Reference

Inheritance diagram for charm_testbeam_check_setup.TestHardwareConnections:



Public Member Functions

- def **setUp** (self)
- def **tearDown** (self)
- def **test_board_connection** (self)
- def **test_powerboard_connection** (self)
- def **test_sensor_connection** (self)

Public Attributes

- **cru**
- **rdo**
- **powerboard**
- **logger**

5.130.1 Detailed Description

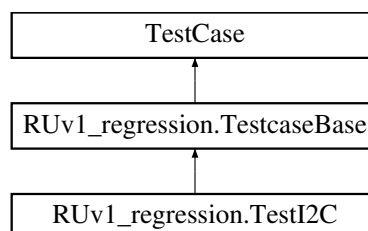
Test to run for verifying proper connections between all devices

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam_check_setup.py

5.131 RUv1_regression.TestI2C Class Reference

Inheritance diagram for RUv1_regression.TestI2C:



Public Member Functions

- def **setUp** (self)
- def **test_i2c_gbtx** (self)
- def **test_i2c_pu1** (self)
- def **test_i2c_pu2** (self)

Public Attributes

- **i2cmod_pu1**
- **i2cmod_pu2**

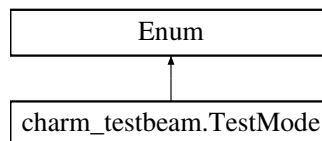
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.132 charm_testbeam.TestMode Class Reference

Inheritance diagram for charm_testbeam.TestMode:



Static Public Attributes

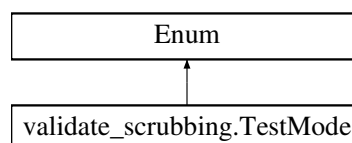
- int **DRY_RUN_JCM** = 0
- int **DRY_RUN_PA3** = 1
- int **JCM_FAULT_INJECTION** = 2
- int **BEAM_JCM** = 3
- int **BEAM_PA3** = 4

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam.py

5.133 validate_scrubbing.TestMode Class Reference

Inheritance diagram for validate_scrubbing.TestMode:



Static Public Attributes

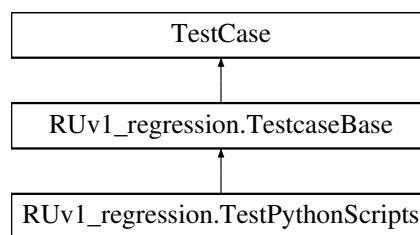
- int **JCM** = 0
- int **PA3** = 1

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/validate_scrubbing.py

5.134 RUv1_regression.TestPythonScripts Class Reference

Inheritance diagram for RUv1_regression.TestPythonScripts:



Public Member Functions

- def **test_githash** (self)
- def **test_prefetch_githash** (self)
- def **test_prefetch_counters** (self)
- def **test_prefetch_basic_communication** (self)
- def **test_prefetch_sensor_datareadout** (self)

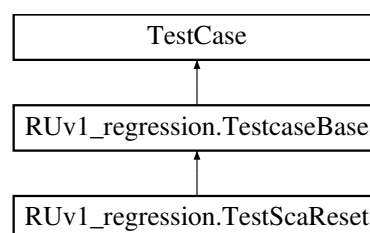
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.135 RUv1_regression.TestScaReset Class Reference

Inheritance diagram for RUv1_regression.TestScaReset:



Public Member Functions

- def **setUp** (self)
- def [test_reset](#) (self)

Additional Inherited Members

5.135.1 Member Function Documentation

5.135.1.1 test_reset()

```
def RUv1_regression.TestScaReset.test_reset (
    self )
```

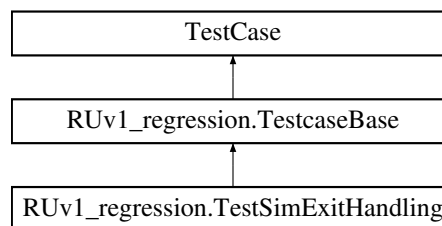
Asserts reset via SCA, it tries to read, expect fail, deassert reset, expect correct read

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.136 RUv1_regression.TestSimExitHandling Class Reference

Inheritance diagram for RUv1_regression.TestSimExitHandling:



Public Member Functions

- def **setUp** (self)
- def **test_1_pass** (self)
- def **test_2_fail** (self)

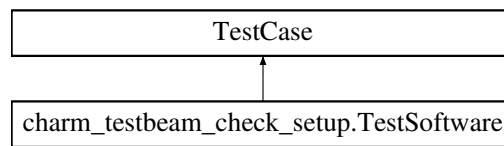
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.137 charm_testbeam_check_setup.TestSoftware Class Reference

Inheritance diagram for charm_testbeam_check_setup.TestSoftware:



Public Member Functions

- def **setUp** (self)
- def **test_program_rdo** (self)

Public Attributes

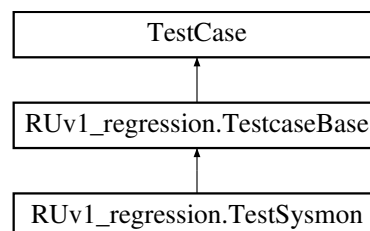
- **cru**
- **rdo**
- **powerboard**
- **logger**

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/charm_testbeam_check_setup.py

5.138 RUv1_regression.TestSysmon Class Reference

Inheritance diagram for RUv1_regression.TestSysmon:



Public Member Functions

- def **test_getTemperature** (self)
- def **test_getVccInt** (self)
- def **test_getVccAux** (self)
- def **test_getVccBram** (self)
- def **test_getOtStatus** (self)
- def **test_getTmrMismatch** (self)
- def **test_enableOtProtection** (self)
- def **test_disableOtProtection** (self)
- def **test_getVcc_ALPIDE_3v3** (self)
- def **test_getVcc_SCA_1v5** (self)

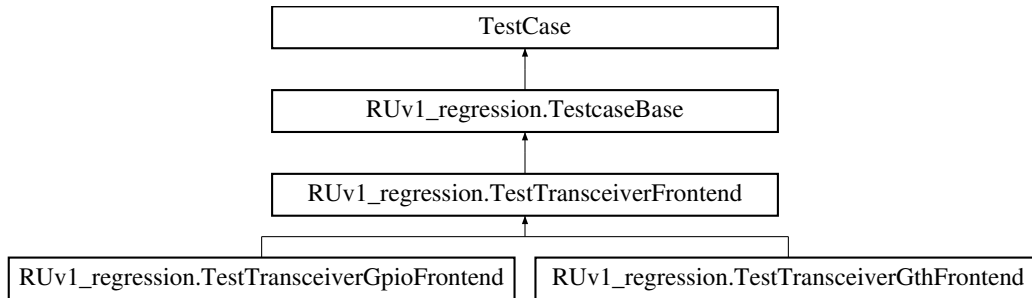
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.139 RUv1_regression.TestTransceiverFrontend Class Reference

Inheritance diagram for RUv1_regression.TestTransceiverFrontend:



Public Member Functions

- def **setUp** (self)
- def **setup_chips** (self, IBSerialLinkSpeed=2)
- def **setup_chips_static** (cls, chips, IBSerialLinkSpeed=2)
- def **check_event_readout** (self, nr_events, nr_noevent_triggers, lanes)

Public Attributes

- **chips**

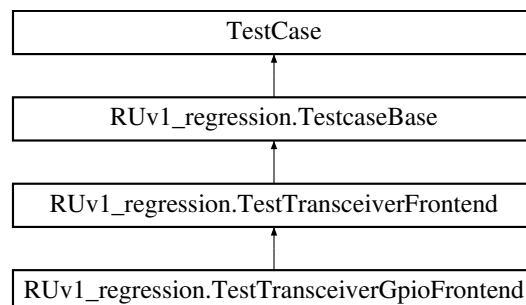
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.140 RUv1_regression.TestTransceiverGpioFrontend Class Reference

Inheritance diagram for RUv1_regression.TestTransceiverGpioFrontend:



Public Member Functions

- def **setUpClass** (cls)
- def **setUp** (self)
- def **tearDown** (self)
- def **test_idelay** (self)
- def **test_data** (self)
- def **test_prbs400** (self)

Public Attributes

- **gpio_idelays**
- **logger**
- **connector_lut**
- **trigger_min_dist**

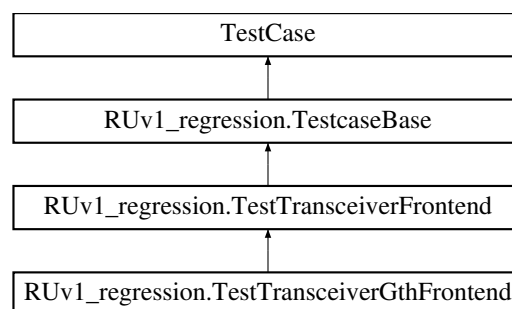
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.141 RUv1_regression.TestTransceiverGthFrontend Class Reference

Inheritance diagram for RUv1_regression.TestTransceiverGthFrontend:



Public Member Functions

- def **test_eyescan** (self)
- def **setUp** (self)
- def **tearDown** (self)
- def **test_drp** (self)
- def **test_data** (self)
- def **test_comma1200** (self)
- def **test_locks** (self)
- def **test_prbs1200** (self)

Public Attributes

- **logger**
- **connector_lut**
- **trigger_min_dist**

Additional Inherited Members

5.141.1 Member Function Documentation

5.141.1.1 test_drp()

```
def RUv1_regression.TestTransceiverGthFrontend.test_drp (
    self )
```

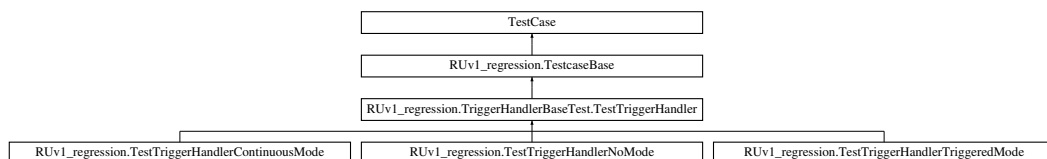
Test DRP interface

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.142 RUv1_regression.TriggerHandlerBaseTest.TestTriggerHandler Class Reference

Inheritance diagram for RUv1_regression.TriggerHandlerBaseTest.TestTriggerHandler:



Public Member Functions

- def **setUp** (self)
- def **tearDown** (self)

Public Attributes

- **gbt_packer_wd_timeout_gth**
- **gbt_packer_wd_timeout_gpio**
- **mode**

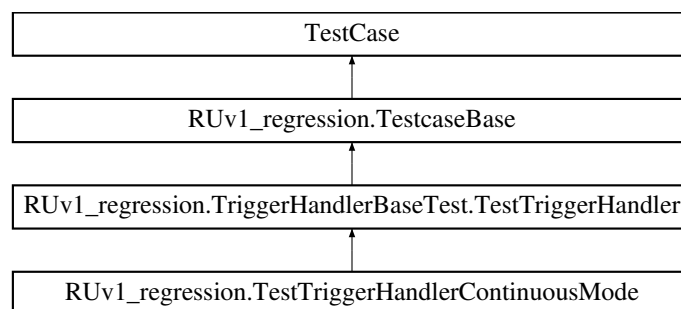
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.143 RUv1_regression.TestTriggerHandlerContinuousMode Class Reference

Inheritance diagram for RUv1_regression.TestTriggerHandlerContinuousMode:



Public Member Functions

- def **setUp** (self)
- def **tearDown** (self)
- def **test_send_trigger** (self)

Public Attributes

- **mode**

Additional Inherited Members

5.143.1 Detailed Description

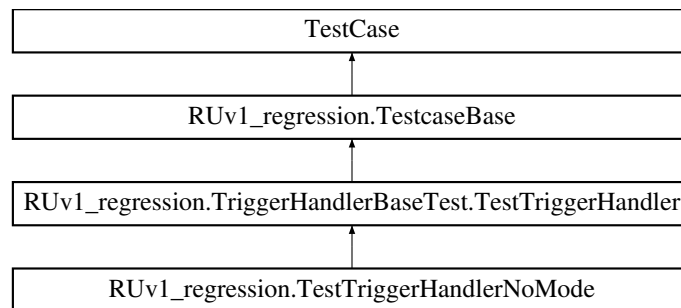
Tests relative to the trigger handler in continuous mode

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.144 RUv1_regression.TestTriggerHandlerNoMode Class Reference

Inheritance diagram for RUv1_regression.TestTriggerHandlerNoMode:



Public Member Functions

- def **test_send_trigger_reject** (self)
- def [test_mode_status_continuous](#) (self)
- def [test_mode_status_triggered](#) (self)

Additional Inherited Members

5.144.1 Detailed Description

Tests relative to the trigger handler in no mode (i.e. inactive)

5.144.2 Member Function Documentation

5.144.2.1 test_mode_status_continuous()

```
def RUv1_regression.TestTriggerHandlerNoMode.test_mode_status_continuous (
    self )
```

Checks if the fsm enters correctly in continuous mode

5.144.2.2 test_mode_status_triggered()

```
def RUv1_regression.TestTriggerHandlerNoMode.test_mode_status_triggered (
    self )
```

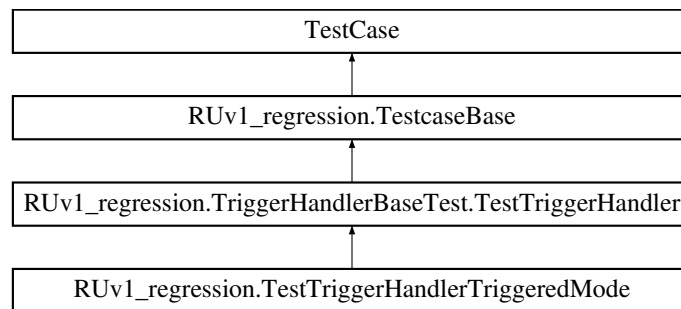
Checks if the fsm enters correctly in triggered mode

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.145 RUv1_regression.TestTriggerHandlerTriggeredMode Class Reference

Inheritance diagram for RUv1_regression.TestTriggerHandlerTriggeredMode:



Public Member Functions

- def **setUp** (self)
- def **tearDown** (self)
- def **test_send_trigger** (self)
- def **test_send_trigger_cru** (self)
- def **test_trigger_too_close_reject** (self)
- def **test_send_multiple_triggers** (self, trigger_nr=10)
- def **test_send_pulse** (self)
- def **test_trigger_gating** (self)

Public Attributes

- **mode**

Additional Inherited Members

5.145.1 Detailed Description

Tests relative to the trigger handler in triggered mode

5.145.2 Member Function Documentation

5.145.2.1 test_send_multiple_triggers()

```
def RUv1_regression.TestTriggerHandlerTriggeredMode.test_send_multiple_triggers (
    self,
    trigger_nr = 10 )
```

Sends multiple triggers in triggered mode properly spaced

5.145.2.2 test_send_pulse()

```
def RUv1_regression.TestTriggerHandlerTriggeredMode.test_send_pulse (
    self )
```

Test sending a pulse

5.145.2.3 test_trigger_gating()

```
def RUv1_regression.TestTriggerHandlerTriggeredMode.test_trigger_gating (
    self )
```

test sending a trigger when gating, expect no trigger sent

5.145.2.4 test_trigger_too_close_reject()

```
def RUv1_regression.TestTriggerHandlerTriggeredMode.test_trigger_too_close_reject (
    self )
```

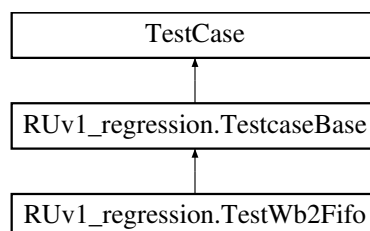
Sends multiple triggers in triggered mode too close in time,
with a different bc, should not generate a trigger to sensor

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.146 RUv1_regression.TestWb2Fifo Class Reference

Inheritance diagram for RUv1_regression.TestWb2Fifo:



Public Member Functions

- def **setUp** (self)
- def **test_pagewrite** (self)
- def **test_performance** (self)
- def **test_validate_counter** (self)
- def **test_counter_read** (self)
- def **test_bulk_write** (self)

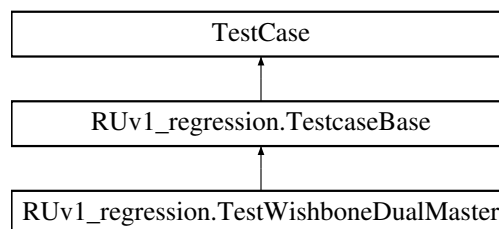
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.147 RUv1_regression.TestWishboneDualMaster Class Reference

Inheritance diagram for RUv1_regression.TestWishboneDualMaster:



Public Member Functions

- def **setUp** (self)
- def **test_multiaccess_write** (self)
- def **test_interleaved_reads** (self)
- def **test_timeouts** (self)

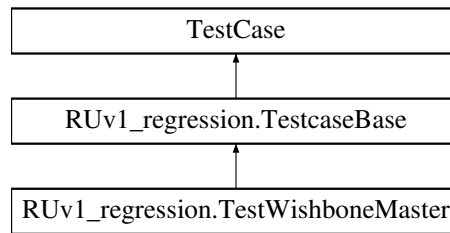
Additional Inherited Members

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.148 RUv1_regression.TestWishboneMaster Class Reference

Inheritance diagram for RUv1_regression.TestWishboneMaster:



Public Member Functions

- def [test_access_nonexist](#) (self)

Additional Inherited Members

5.148.1 Detailed Description

Test functions related to the wishbone master

5.148.2 Member Function Documentation

5.148.2.1 test_access_nonexist()

```
def RUv1_regression.TestWishboneMaster.test_access_nonexist (
    self )
```

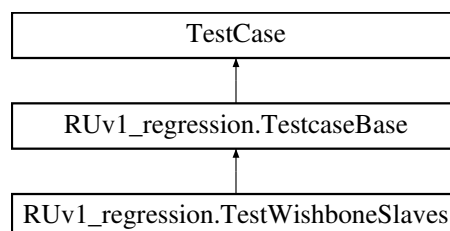
Communicates with the last valid address of the RUv1

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.149 RUv1_regression.TestWishboneSlaves Class Reference

Inheritance diagram for RUv1_regression.TestWishboneSlaves:



Public Member Functions

- def [test_wsmstr](#) (self)
- def [test_gbt_x_flow_monitor](#) (self)
- def [test_sysmon](#) (self)
- def [test_gth_frontend](#) (self)
- def [test_gpio_frontend](#) (self)
- def [test_datapath_monitor](#) (self)
- def [test_datapath_monitor_GPIO_1](#) (self)
- def [test_datapath_monitor_GPIO_2](#) (self)
- def [test_gbt_packer_monitor_gth](#) (self)
- def [test_gbt_packer_monitor_gpio](#) (self)
- def [test_wsstatus](#) (self)
- def [test_radiation_monitor](#) (self)
- def [test_dctrl](#) (self)
- def [test_gbt_x0](#) (self)
- def [test_gbt_x2](#) (self)
- def [test_trigger_handler](#) (self)
- def [test_trigger_handler_monitor](#) (self)
- def [test_gbt_packer_gth](#) (self)
- def [test_gbt_packer_gpio](#) (self)

Public Attributes

- [chip](#)

Static Public Attributes

- **WB_MASTER_MONITOR** = ru_board.RUv1.MASTER_MONITOR_MODULE
- **WB_STATUS** = ru_board.RUv1.STATUS_MODULE
- **WB_DCTRL** = ru_board.RUv1.ALPIDE_MODULE
- **WB_I2C_GBT** = ru_board.RUv1.I2C_GBT_MODULE
- **WB_I2C_PU1** = ru_board.RUv1.I2C_PU1_MODULE
- **WB_I2C_PU2** = ru_board.RUv1.I2C_PU2_MODULE
- **WB_GTH_FRONTEND** = ru_board.RUv1.GTH_FRONTEND_MODULE
- **WB_DATAPATH_MON** = ru_board.RUv1.DATAPATH_MONITOR_MODULE
- **WB_GBTX0** = ru_board.RUv1.GBTX0_MODULE
- **WB_GBTX2** = ru_board.RUv1.GBTX2_MODULE
- **WB_WAIT** = ru_board.RUv1.WAIT_MODULE
- **WB_RADMON** = ru_board.RUv1.RADMON_MODULE
- **WB_SYSMON** = ru_board.RUv1.SYSMON_MODULE
- **WB_GBTX_FLOW_MONITOR** = ru_board.RUv1.GBTX_FLOW_MONITOR_MODULE
- **WB_TRIGGER_HANDLER** = ru_board.RUv1.TRIGGER_HANDLER
- **WB_TRIGGER_HANDLER_MONITOR** = ru_board.RUv1.TRIGGER_HANDLER_MONITOR
- **WB_GPIO_CONTROL** = ru_board.RUv1.GPIO_CONTROL
- **WB_DATAPATH_MON_GPIO_1** = ru_board.RUv1.DATAPATH_MONITOR_GPIO_1
- **WB_DATAPATH_MON_GPIO_2** = ru_board.RUv1.DATAPATH_MONITOR_GPIO_2

5.149.1 Detailed Description

Test functions related to wishbone slaves (not directly the slave functionality)

5.149.2 Member Function Documentation

5.149.2.1 test_gbtx_flow_monitor()

```
def RUv1_regression.TestWishboneSlaves.test_gbtx_flow_monitor (  
    self )
```

Read/write from gbtx_flow_monitor slave

5.149.2.2 test_sysmon()

```
def RUv1_regression.TestWishboneSlaves.test_sysmon (  
    self )
```

Read/write from wsmstr slave

5.149.2.3 test_wsmstr()

```
def RUv1_regression.TestWishboneSlaves.test_wsmstr (  
    self )
```

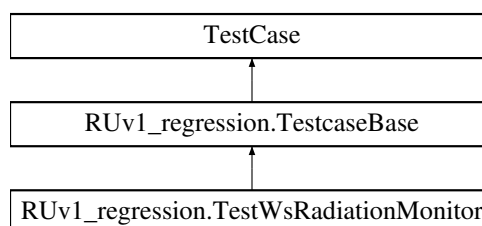
Read/write from ws_master_monitor slave

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.150 RUv1_regression.TestWsRadiationMonitor Class Reference

Inheritance diagram for RUv1_regression.TestWsRadiationMonitor:



Public Member Functions

- def **test_radmon** (self)

Additional Inherited Members

5.150.1 Detailed Description

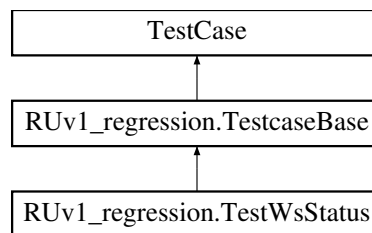
Class to verify the behaviour of the radiation monitor wishbone slave

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.151 RUv1_regression.TestWsStatus Class Reference

Inheritance diagram for RUv1_regression.TestWsStatus:



Public Member Functions

- def **test_read_gitghash** (self)
- def **test_read_gitdate** (self)
- def **test_os** (self)
- def **test_dipswitch** (self)
- def **test_dna** (self)

Additional Inherited Members

5.151.1 Detailed Description

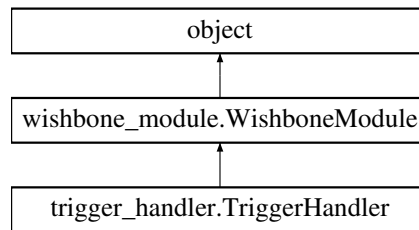
Class to verify the behaviour of the status wishbone slave

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.152 trigger_handler.TriggerHandler Class Reference

Inheritance diagram for trigger_handler.TriggerHandler:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def **enable** (self, commitTransaction=True)
- def **disable** (self, commitTransaction=True)
- def **is_enable** (self, commitTransaction=True)
- def **set_trigger_period** (self, value, commitTransaction=True)
- def **get_trigger_period** (self, commitTransaction=True)
- def **get_pulse_ntrigger** (self, commitTransaction=True)
- def **configure_to_send_pulses** (self, commitTransaction=True)
- def **configure_to_send_triggers** (self, commitTransaction=True)
- def **set_trigger_minimum_distance** (self, value, commitTransaction=True)
- def **get_trigger_minimum_distance** (self, commitTransaction=True)
- def **get_operating_mode** (self, commitTransaction=True)
- def **set_opcode_gating** (self, value, commitTransaction=True)
- def **get_opcode_gating** (self, commitTransaction=True)
- def **get_mismatch** (self, commitTransaction=True)
- def **set_trigger_source** (self, value, commitTransaction=True)
- def **get_trigger_source** (self, commitTransaction=True)
- def **dump_config** (self)

Additional Inherited Members

5.152.1 Detailed Description

trigger handler wishbone slave

5.152.2 Member Function Documentation

5.152.2.1 dump_config()

```
def trigger_handler.TriggerHandler.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.152.2.2 get_opcode_gating()

```
def trigger_handler.TriggerHandler.get_opcode_gating (
    self,
    commitTransaction = True )
```

returns the current setting of the opcode gating register

5.152.2.3 get_operating_mode()

```
def trigger_handler.TriggerHandler.get_operating_mode (
    self,
    commitTransaction = True )
```

Returns the operating mode of the trigger_handler

5.152.2.4 get_trigger_minimum_distance()

```
def trigger_handler.TriggerHandler.get_trigger_minimum_distance (
    self,
    commitTransaction = True )
```

Gets the trigger minimum_temporal distance in units of 6.25ns

5.152.2.5 get_trigger_period()

```
def trigger_handler.TriggerHandler.get_trigger_period (
    self,
    commitTransaction = True )
```

Gets the continuous mode trigger period in units of 6.25ns

5.152.2.6 get_trigger_source()

```
def trigger_handler.TriggerHandler.get_trigger_source (
    self,
    commitTransaction = True )
```

returns the current setting of the trigger_source register

5.152.2.7 set_opcode_gating()

```
def trigger_handler.TriggerHandler.set_opcode_gating (
    self,
    value,
    commitTransaction = True )
```

Allows gating the trigger/pulse signal to the alpine control generating an empty gbt packet

5.152.2.8 set_trigger_minimum_distance()

```
def trigger_handler.TriggerHandler.set_trigger_minimum_distance (
    self,
    value,
    commitTransaction = True )
```

Sets the trigger minimum_temporal distance in units of 6.25ns

5.152.2.9 `set_trigger_period()`

```
def trigger_handler.TriggerHandler.set_trigger_period (
    self,
    value,
    commitTransaction = True )
```

Sets the continuous mode trigger period in units of 6.25ns

5.152.2.10 `set_trigger_source()`

```
def trigger_handler.TriggerHandler.set_trigger_source (
    self,
    value,
    commitTransaction = True )
```

Allows selecting if triggers should come from gbtx0 or gbtx2

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/trigger_handler.py

5.153 `RUv1_regression.TriggerHandlerBaseTest` Class Reference

Classes

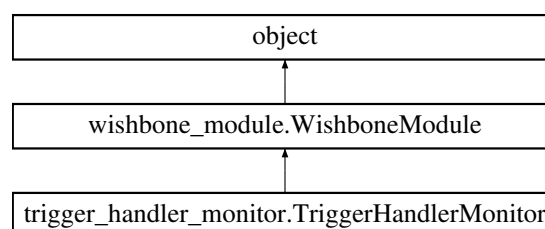
- class [TestTriggerHandler](#)

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/RUv1_regression.py

5.154 `trigger_handler_monitor.TriggerHandlerMonitor` Class Reference

Inheritance diagram for `trigger_handler_monitor.TriggerHandlerMonitor`:



Public Member Functions

- `def __init__ (self, moduleid, board_obj)`
- `def latch_counters (self, commitTransaction=True)`
- `def reset_counters (self, commitTransaction=True)`
- `def get_counters (self, reset_after=False, commitTransaction=True)`
- `def dump_config (self)`

Additional Inherited Members

5.154.1 Detailed Description

trigger handler wishbone slave

5.154.2 Member Function Documentation

5.154.2.1 dump_config()

```
def trigger_handler_monitor.TriggerHandlerMonitor.dump_config (  
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.154.2.2 get_counters()

```
def trigger_handler_monitor.TriggerHandlerMonitor.get_counters (  
    self,  
    reset_after = False,  
    commitTransaction = True )
```

Reads all counters and returns dictionary

5.154.2.3 latch_counters()

```
def trigger_handler_monitor.TriggerHandlerMonitor.latch_counters (  
    self,  
    commitTransaction = True )
```

Latches all the counters

5.154.2.4 reset_counters()

```
def trigger_handler_monitor.TriggerHandlerMonitor.reset_counters (
    self,
    commitTransaction = True )
```

Resets all the counters

The documentation for this class was generated from the following file:

- /media/sf_proj/RUV1_Test/modules/board_support_software/software/py/trigger_handler_monitor.py

5.155 tti.TTI Class Reference

Public Member Functions

- def **__init__** (self, port=DEFAULT_PORT, reset=True)
- def **setup_standalone** (self)
- def **setup_logging** (self)
- def **get_model** (self)
- def **configure_channel** (self, channel, voltage, current, ovp_ratio=1.05, ocp_ratio=1.05)
- def **get_channel_config** (self, channel)
- def **get_channel_config_all** (self)
- def **activate_output_channel** (self, channel)
- def **deactivate_output_channel** (self, channel)
- def **activate_output_channels** (self, channels)
- def **deactivate_output_channels** (self, channels)
- def **activate_output_all_channels** (self, activate=True)
- def **deactivate_output_all_channels** (self)
- def **is_channel_on** (self, channel)
- def **get_fuse_triggered** (self, channel)
- def **clear_triggered_fuse** (self)
- def **get_current** (self, channel)
- def **get_voltage** (self, channel)
- def **get_power** (self, channel)
- def **get_power_all** (self)

Public Attributes

- **logger**
- **link**
- **number_channels**
- **logdir**

5.155.1 Detailed Description

Control of TTI power supply

5.155.2 Member Function Documentation

5.155.2.1 activate_output_all_channels()

```
def tti.TTI.activate_output_all_channels (
    self,
    activate = True )
```

Activates output of all channels

5.155.2.2 activate_output_channel()

```
def tti.TTI.activate_output_channel (
    self,
    channel )
```

Activate output of given channel

5.155.2.3 activate_output_channels()

```
def tti.TTI.activate_output_channels (
    self,
    channels )
```

Activate the output of a list of given channels

5.155.2.4 clear_triggered_fuse()

```
def tti.TTI.clear_triggered_fuse (
    self )
```

Attempts to clear triggered fuses, cannot recover LSR bit 6

5.155.2.5 `configure_channel()`

```
def tti.TTI.configure_channel (
    self,
    channel,
    voltage,
    current,
    ovp_ratio = 1.05,
    ocp_ratio = 1.05 )
```

Configure channel N with given voltage, current. Ratio for the trip point of the over voltage and over current can be set.

5.155.2.6 `deactivate_output_all_channels()`

```
def tti.TTI.deactivate_output_all_channels (
    self )
```

Deactivate output of all channels

5.155.2.7 `deactivate_output_channel()`

```
def tti.TTI.deactivate_output_channel (
    self,
    channel )
```

Deactivate output of given channel

5.155.2.8 `deactivate_output_channels()`

```
def tti.TTI.deactivate_output_channels (
    self,
    channels )
```

Deactivate the output of a list of given channels

5.155.2.9 get_channel_config()

```
def tti.TTI.get_channel_config (
    self,
    channel )
```

Get the channel configuration

5.155.2.10 get_channel_config_all()

```
def tti.TTI.get_channel_config_all (
    self )
```

Get configuratoin for all channels

5.155.2.11 get_current()

```
def tti.TTI.get_current (
    self,
    channel )
```

Get the current of a given channel in mA

5.155.2.12 get_fuse_triggered()

```
def tti.TTI.get_fuse_triggered (
    self,
    channel )
```

Returns whether or not a fuse was triggered

5.155.2.13 get_model()

```
def tti.TTI.get_model (
    self )
```

Return model and type

5.155.2.14 get_power()

```
def tti.TTI.get_power (
    self,
    channel )
```

Return Voltage, Current, Power of channel

5.155.2.15 get_power_all()

```
def tti.TTI.get_power_all (
    self )
```

Gets the voltage, current and power of all channels

5.155.2.16 get_voltage()

```
def tti.TTI.get_voltage (
    self,
    channel )
```

Get the voltage of a given channel in V

5.155.2.17 is_channel_on()

```
def tti.TTI.is_channel_on (
    self,
    channel )
```

Reports if a channel is active

5.155.2.18 setup_logging()

```
def tti.TTI.setup_logging (
    self )
```

Sets up the logging

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/tti.py

5.156 usb_communication.communication.UsbCommServer Class Reference

Public Member Functions

- def `__init__` (self, executable='../usb_comm/build/usb_comm', serial=None)
- def `start` (self)
- def `stop` (self)
- def `is_stopped` (self)
- def `read_messages` (self)

Public Attributes

- `executable`
- `process`
- `message_queue`
- `message_thread`
- `logger`
- `serial`

5.156.1 Detailed Description

Run usb_comm Server in subprocess.

Convenience class to start / stop the usb_comm software from python

5.156.2 Member Function Documentation

5.156.2.1 is_stopped()

```
def usb_communication.communication.UsbCommServer.is_stopped (
    self )
```

Check if the server is stopped or still running

5.156.2.2 read_messages()

```
def usb_communication.communication.UsbCommServer.read_messages (
    self )
```

Read all messages sent by the usb comm server program.

5.156.2.3 start()

```
def usb_communication.communication.UsbCommServer.start (  
    self )
```

Start the USB Comm Server program

5.156.2.4 stop()

```
def usb_communication.communication.UsbCommServer.stop (  
    self )
```

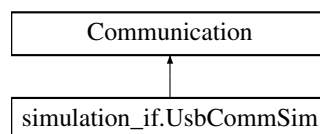
Send stop signal to USB Comm Server program

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py

5.157 simulation_if.UsbCommSim Class Reference

Inheritance diagram for simulation_if.UsbCommSim:



Public Member Functions

- def **__init__** (self, server)
- def **open** (self)
- def **close** (self)
- def **flush_dp0** (self)
- def **clear** (self)

Public Attributes

- **max_retries**
- **dp0_queue**
- **mutex**
- **files**
- **server**

5.157.1 Detailed Description

Usb Communication wrapper connecting to simulation.

This wrapper writes read/write actions to a set of fifos, which will be checked by a simulation testbench and then forwarded to the simulation environment

5.157.2 Member Function Documentation

5.157.2.1 clear()

```
def simulation_if.UsbCommSim.clear (  
    self )
```

Clear buffers: Close and reopen files

5.157.2.2 flush_dp0()

```
def simulation_if.UsbCommSim.flush_dp0 (  
    self )
```

Flush DP0: send data to fifo

5.157.2.3 open()

```
def simulation_if.UsbCommSim.open (  
    self )
```

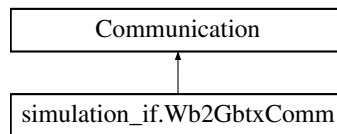
(re)open buffer files for data exchanges

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/simulation_if.py

5.158 simulation_if.Wb2GbtComm Class Reference

Inheritance diagram for simulation_if.Wb2GbtComm:



Public Member Functions

- def **__init__** (self, gbt_sim, create_raw_data_files=True)
- def **__del__** (self)
- def **send_idle** (self, numToSend=1)
- def **send_trigger** (self, triggerType=0x10, bc=0xabc, orbit=0x43215678)
- def **discardall_dp1** (self, maxReads=10)
- def **discardall_dp2** (self, maxReads=10)
- def **discardall_dp3** (self, maxReads=10)

Public Attributes

- **gbt_sim**
- **dp2_file**

5.158.1 Detailed Description

Usb Communication wrapper connecting to GbtLink of Simulation.

This wrapper writes read/write actions to a set of fifos, which will be checked by a simulation testbench and then forwarded to the simulation environment

5.158.2 Member Function Documentation

5.158.2.1 send_trigger()

```
def simulation_if.Wb2GbtComm.send_trigger (
    self,
    triggerType = 0x10,
    bc = 0xabc,
    orbit = 0x43215678 )
```

Provides trigger to the RUv1

Inputs:
 trigger type: 12 bit
 bc: 12 bit
 orbit: 32 bit (only 31 used on the RUv0_CRU)

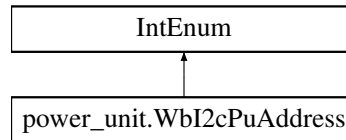
TODO: update with proper documentation

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/simulation_if.py

5.159 power_unit.Wbl2cPuAddress Class Reference

Inheritance diagram for power_unit.Wbl2cPuAddress:



Static Public Attributes

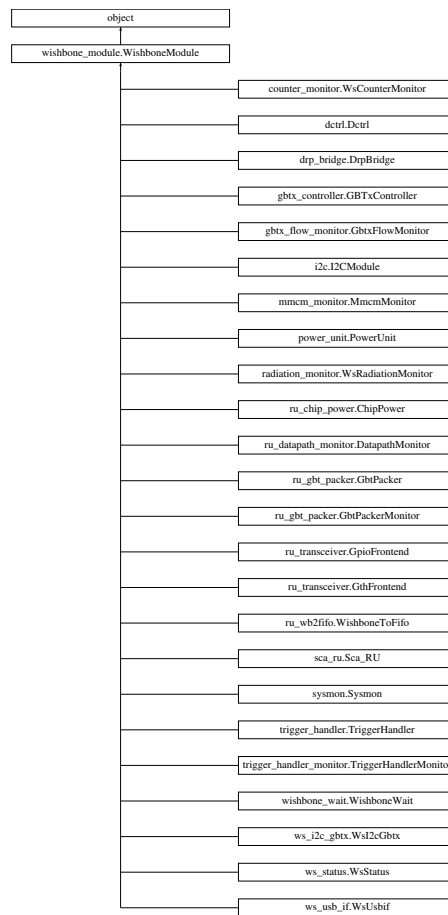
- int **InternalRegister** = 0x0
- int **TempThreshConfigAddress** = 0x1
- int **ThresCurrAddress_0** = 0x2
- int **ThresCurrAddress_1** = 0x3
- int **ThresCurrAddress_2** = 0x4
- int **ThresCurrAddress_3** = 0x5
- int **PotPowerAddress_0** = 0x6
- int **PotPowerAddress_1** = 0x7
- int **PotPowerAddress_2** = 0x8
- int **PotPowerAddress_3** = 0x9
- int **PotBiasAddress** = 0xA
- int **ADCAddressSetup_0** = 0xB
- int **ADCAddressSetup_1** = 0xC
- int **ADCAddressSetup_2** = 0xD
- int **ADCAddressSetup_3** = 0xE
- int **ADCBiasAddressSetup** = 0xF
- int **ADCAddress_0** = 0x10
- int **ADCAddress_1** = 0x11
- int **ADCAddress_2** = 0x12
- int **ADCAddress_3** = 0x13
- int **ADCBiasAddress** = 0x14
- int **TempThreshRdAddress** = 0x15
- int **IOExpanderBiasAddress** = 0x16
- int **IOExpanderPowerAddress_0** = 0x20
- int **IOExpanderPowerAddress_1** = 0x21

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/power_unit.py

5.160 wishbone_module.WishboneModule Class Reference

Inheritance diagram for wishbone_module.WishboneModule:



Public Member Functions

- `def __init__(self, moduleid, name, board_obj)`
- `def write(self, addr, data, commitTransaction=True)`
- `def read(self, addr, commitTransaction=True)`
- `def flush(self)`
- `def firmware_wait(self, wait_value, commitTransaction=True)`
- `def read_all(self)`
- `def dump_config(self)`

Public Attributes

- **moduleid**
- **board**
- **comm**
- **logger**

5.160.1 Detailed Description

Abstract wishbone module providing basic communication functions

5.160.2 Member Function Documentation

5.160.2.1 dump_config()

```
def wishbone_module.WishboneModule.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented in [dctrl.Dctrl](#), [sysmon.Sysmon](#), [ru_chip_power.ChipPower](#), [gbtx_controller.GBTxController](#), [ws_usb_if.WsUsbif](#), [ru_gbt_packer.GbtPacker](#), [mmcm_monitor.MmcmMonitor](#), [trigger_handler.TriggerHandler](#), [ws_status.WsStatus](#), [gbtx_flow_monitor.GbtxFLOWMonitor](#), [trigger_handler_monitor.TriggerHandlerMonitor](#), [radiation_monitor.WsRadiationMonitor](#), [ws_i2c_gbt.WsI2cGbt](#), [wishbone_wait.WishboneWait](#), [ws_master_monitor.WsMasterMonitor](#) and [i2c.I2CModule](#).

5.160.2.2 firmware_wait()

```
def wishbone_module.WishboneModule.firmware_wait (
    self,
    wait_value,
    commitTransaction = True )
```

Execute a wait of wait_value*6.25 ns

5.160.2.3 flush()

```
def wishbone_module.WishboneModule.flush (
    self )
```

Flush all pending transactions.

5.160.2.4 read()

```
def wishbone_module.WishboneModule.read (
    self,
    addr,
    commitTransaction = True )
```

Read from a specific address from the module. Optionally commit transaction

5.160.2.5 read_all()

```
def wishbone_module.WishboneModule.read_all (
    self )
```

Read all results after a flush()

5.160.2.6 write()

```
def wishbone_module.WishboneModule.write (
    self,
    addr,
    data,
    commitTransaction = True )
```

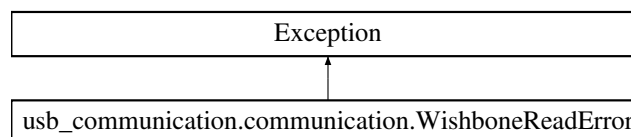
Write to a specific address of the module. Optionally commit transaction

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/wishbone_module.py

5.161 usb_communication.communication.WishboneReadError Class Reference

Inheritance diagram for usb_communication.communication.WishboneReadError:



Public Member Functions

- def **__init__** (self, value="", data=0, address=0, rderr_flag=0)
- def **__str__** (self)
- def **__data__** (self)
- def **__address__** (self)
- def **__rderr_flag__** (self)
- def **print_info** (self)

Public Attributes

- **value**
- **data**
- **address**
- **rderr_flag**

5.161.1 Detailed Description

basic class to define a wishbone read error exception

5.161.2 Member Function Documentation

5.161.2.1 print_info()

```
def usb_communication.communication.WishboneReadError.print_info (
    self )
```

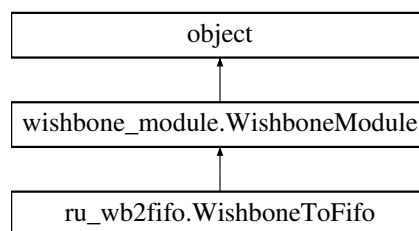
returns the args of the exception

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/usb_if/software/usb_communication/communication.py

5.162 ru_wb2fifo.WishboneToFifo Class Reference

Inheritance diagram for ru_wb2fifo.WishboneToFifo:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def **page_write** (self, data)
- def **write_bitfile** (self, filename, timeout_count=100)
- def **bulk_write** (self, data, chunks=256 *10)
- def **counter_test** (self, max_val, chunks=256 *10)
- def **read_write_counter** (self)
- def **read_read_counter** (self)
- def **read_overflow_counter** (self)
- def **read_validate_read_counter** (self)
- def **read_fifo_status** (self)

Additional Inherited Members

5.162.1 Detailed Description

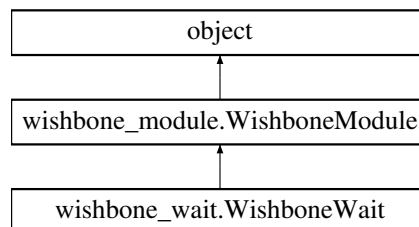
Send data to wishbone fifo slave

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ru_wb2fifo.py

5.163 wishbone_wait.WishboneWait Class Reference

Inheritance diagram for wishbone_wait.WishboneWait:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def **rst_ctrl_cntrs** (self)
- def **get_counters** (self)
- def **wait** (self, wait_value, commitTransaction=True)
- def **dump_config** (self)

Public Attributes

- **verbose**

5.163.1 Detailed Description

wishbone_wait wishbone slave

5.163.2 Member Function Documentation

5.163.2.1 dump_config()

```
def wishbone_wait.WishboneWait.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.163.2.2 get_counters()

```
def wishbone_wait.WishboneWait.get_counters (
    self )
```

latches and reads all the counters

5.163.2.3 rst_ctrl_cntrs()

```
def wishbone_wait.WishboneWait.rst_ctrl_cntrs (
    self )
```

resets all the counters

5.163.2.4 wait()

```
def wishbone_wait.WishboneWait.wait (
    self,
    wait_value,
    commitTransaction = True )
```

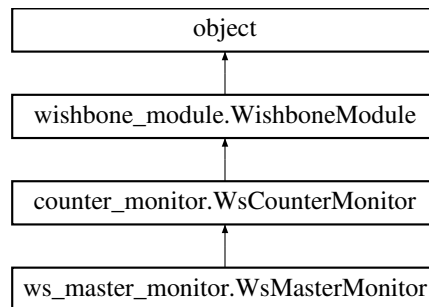
Implements the wait function of the wait wishbone slave

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/wishbone_wait.py

5.164 counter_monitor.WsCounterMonitor Class Reference

Inheritance diagram for counter_monitor.WsCounterMonitor:



Public Member Functions

- `def __init__ (self, moduleid, name, board_obj)`
- `def reset_counter (self, register, commitTransaction=True)`
- `def reset_all_counters (self, commitTransaction=True)`
- `def latch_all_counters (self, commitTransaction=True)`

Public Attributes

- **LATCH_CMD**
- **RESET_ONE_CMD**
- **RESET_ALL_CMD**

5.164.1 Detailed Description

Parent class for counter monitor slaves that use the counter_monitor.vhd WB slave

5.164.2 Member Function Documentation

5.164.2.1 latch_all_counters()

```
def counter_monitor.WsCounterMonitor.latch_all_counters (
    self,
    commitTransaction = True )
```

Latches values of all counters into WB register

5.164.2.2 reset_all_counters()

```
def counter_monitor.WsCounterMonitor.reset_all_counters (
    self,
    commitTransaction = True )
```

Resets all counters

5.164.2.3 reset_counter()

```
def counter_monitor.WsCounterMonitor.reset_counter (
    self,
    register,
    commitTransaction = True )
```

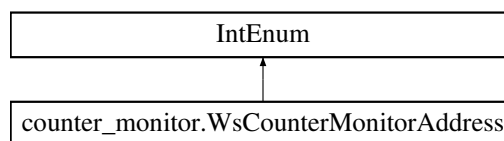
Resets counter specified in 'register'

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/counter_monitor.py

5.165 counter_monitor.WsCounterMonitorAddress Class Reference

Inheritance diagram for counter_monitor.WsCounterMonitorAddress:



Static Public Attributes

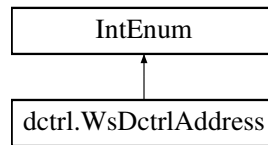
- int **LATCH_COUNTERS** = 0x00
- int **RESET_COUNTERS** = 0x01

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/counter_monitor.py

5.166 dctrl.WsDctrlAddress Class Reference

Inheritance diagram for dctrl.WsDctrlAddress:



Static Public Attributes

- int **WRITE_CTRL** = 0x00
- int **WRITE_ADDRESS** = 0x01
- int **WRITE_DATA** = 0x02
- int **PHASE_FORCE** = 0x03
- int **READ_STATUS** = 0x04
- int **READ_DATA** = 0x05
- int **LATCH_CTRL_CNTRS** = 0x06
- int **RST_CTRL_CNTRS** = 0x07
- int **READ_BROADCAST_CNTR** = 0x08
- int **READ_WRITE_CNTR** = 0x09
- int **READ_READ_CNTR** = 0x0A
- int **READ_OPCODE_CNTR_LSB** = 0x0B
- int **READ_TRIGGER_SENT_CNTR_LSB** = 0x0C
- int **READ_TRIGGER_NOT_SENT_CNTR** = 0x0D
- int **MASK_BUSY_REG** = 0x0E
- int **BUSY_TRANSCEIVER_MASK_LSB** = 0x0F
- int **READ_BUSY_TRANSCEIVER_STATUS_LSB** = 0x10
- int **SET_DCTRL_INPUT** = 0x11
- int **SET_DCTRL_TX_MASK** = 0x12
- int **BUSY_TRANSCEIVER_MASK_MSB** = 0x13
- int **READ_BUSY_TRANSCEIVER_STATUS_MSB** = 0x14
- int **SET_DCLK_TX_MASK** = 0x15
- int **MANCHESTER_TX_EN** = 0x16
- int **SET_IDELAY_VALUE0** = 0x17
- int **SET_IDELAY_VALUE1** = 0x18
- int **SET_IDELAY_VALUE2** = 0x19
- int **SET_IDELAY_VALUE3** = 0x1A
- int **SET_IDELAY_VALUE4** = 0x1B
- int **GET_IDELAY_VALUE0** = 0x1C
- int **GET_IDELAY_VALUE1** = 0x1D
- int **GET_IDELAY_VALUE2** = 0x1E
- int **GET_IDELAY_VALUE3** = 0x1F
- int **GET_IDELAY_VALUE4** = 0x20
- int **AUTO_PHASE_OFFSET** = 0x21
- int **SET_DCLK_PARALLEL_0** = 0x22
- int **SET_DCLK_PARALLEL_1** = 0x23
- int **SET_DCLK_PARALLEL_2** = 0x24
- int **SET_DCLK_PARALLEL_3** = 0x25
- int **SET_DCLK_PARALLEL_4** = 0x26
- int **WAIT_CYCLES** = 0x27

- int **MANCHESTER_RX_DETECTED** = 0x28
- int **READ_PULSE_SENT_CNTR_LSB** = 0x29
- int **READ_PULSE_SENT_CNTR_MSB** = 0x2A
- int **READ_OPCODE_REJECTED_CNTR_LSB** = 0x2B
- int **READ_OPCODE_REJECTED_CNTR_MSB** = 0x2C
- int **READ_OPCODE_CNTR_MSB** = 0x2D
- int **READ_TRIGGER_SENT_CNTR_MSB** = 0x2E
- int **MISMATCH** = 0x2F
- int **NUM_REG** = 0x30

5.166.1 Detailed Description

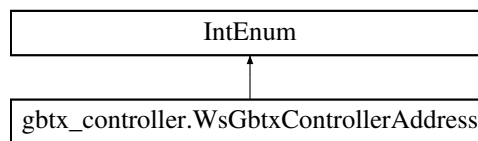
memory mapping for dctrl/ctrl module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/dctrl.py

5.167 gbt_x_controller.WsGbt_xControllerAddress Class Reference

Inheritance diagram for gbt_x_controller.WsGbt_xControllerAddress:



Static Public Attributes

- int **GBTX_CONTROLLER_ENABLE** = 0x00
- int **GBTX_RESET** = 0x01
- int **SET_IDELAY_VALUE0** = 0x02
- int **SET_IDELAY_VALUE1** = 0x03
- int **SET_IDELAY_VALUE2** = 0x04
- int **SET_IDELAY_VALUE3** = 0x05
- int **SET_IDELAY_VALUE4** = 0x06
- int **SET_IDELAY_VALUE5** = 0x07
- int **SET_IDELAY_VALUE6** = 0x08
- int **SET_IDELAY_VALUE7** = 0x09
- int **SET_IDELAY_VALUE8** = 0x0A
- int **SET_IDELAY_VALUE9** = 0x0B
- int **GET_IDELAY_VALUE0** = 0x0C
- int **GET_IDELAY_VALUE1** = 0x0D
- int **GET_IDELAY_VALUE2** = 0x0E
- int **GET_IDELAY_VALUE3** = 0x0F
- int **GET_IDELAY_VALUE4** = 0x10
- int **GET_IDELAY_VALUE5** = 0x11
- int **GET_IDELAY_VALUE6** = 0x12

- int **GET_IDELAY_VALUE7** = 0x13
- int **GET_IDELAY_VALUE8** = 0x14
- int **GET_IDELAY_VALUE9** = 0x15
- int **IDELAY_LOAD** = 0x16
- int **BITSLIP_RX_VALUE** = 0x17
- int **BITSLIP_TX_VALUE** = 0x18
- int **BITSLIP_LOAD** = 0x19
- int **TX_PATTERN_SELECTION** = 0x1A
- int **TX1_PATTERN_SELECTION** = 0x1B
- int **MISMATCH** = 0x1C
- int **RXRDY_LOST_FLAG** = 0x1D
- int **NUM_REGS** = 0x1E

5.167.1 Detailed Description

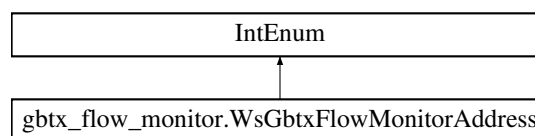
memory mapping for gbtX controller module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbtX_controller.py

5.168 gbtX_flow_monitor.WsGbtXFlowMonitorAddress Class Reference

Inheritance diagram for gbtX_flow_monitor.WsGbtXFlowMonitorAddress:



Static Public Attributes

- int **LATCH_COUNTERS** = 0x00
- int **RESET_COUNTERS** = 0x01
- int **READ_COUNTER_SWT_DOWNLINK_LSB** = 0x02
- int **READ_COUNTER_TRG_DOWNLINK_LSB** = 0x03
- int **READ_COUNTER_SWT_UPLINK_LSB** = 0x04
- int **READ_COUNTER_SOP_UPLINK_LSB** = 0x05
- int **READ_COUNTER_EOP_UPLINK_LSB** = 0x06
- int **READ_COUNTER_OVERFLOW_DOWNLINK_LSB** = 0x07
- int **READ_COUNTER_OVERFLOW_UPLINK_LSB** = 0x08
- int **READ_COUNTER_SWT_DOWNLINK_MSB** = 0x09
- int **READ_COUNTER_TRG_DOWNLINK_MSB** = 0x0A
- int **READ_COUNTER_SWT_UPLINK_MSB** = 0x0B
- int **READ_COUNTER_SOP_UPLINK_MSB** = 0x0C
- int **READ_COUNTER_EOP_UPLINK_MSB** = 0x0D
- int **READ_COUNTER_OVERFLOW_DOWNLINK_MSB** = 0x0E
- int **READ_COUNTER_OVERFLOW_UPLINK_MSB** = 0x0F

5.168.1 Detailed Description

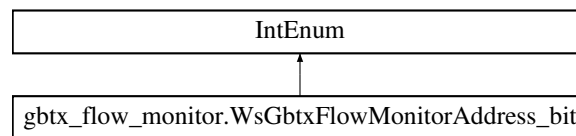
memory mapping for sca controller module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbtx_flow_monitor.py

5.169 gbtx_flow_monitor.WsGbtxFLOWMonitorAddress_bit Class Reference

Inheritance diagram for gbtx_flow_monitor.WsGbtxFLOWMonitorAddress_bit:



Static Public Attributes

- int **SWT_DOWNLINK** = 0
- int **TRG_DOWNLINK** = 1
- int **SWT_UPLINK** = 2
- int **SOP_UPLINK** = 3
- int **EOP_UPLINK** = 4
- int **OVERFLOW_DOWNLINK** = 5
- int **OVERFLOW_UPLINK** = 6
- int **NUM_COUNTERS** = 7

5.169.1 Detailed Description

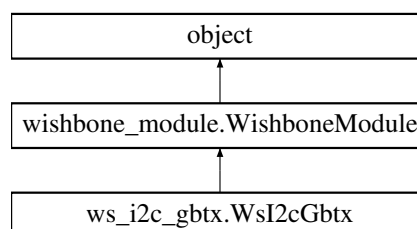
Bit mapping of the {RESET|LATCH}_COUNTERS registers

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/gbtx_flow_monitor.py

5.170 ws_i2c_gbtx.WsI2cGbtX Class Reference

Inheritance diagram for ws_i2c_gbtx.WsI2cGbtX:



Public Member Functions

- def `__init__` (self, moduleid, board_obj)
- def `set_address` (self, address, gbtx=0, commitTransaction=True)
- def `get_address` (self, gbtx=0, commitTransaction=True)
- def `write_data` (self, gbtx, address, data, commitTransaction=True)
- def `read_data` (self, gbtx, address, commitTransaction=True)
- def `dump_gbtx_config` (self, gbtx)
- def `dump_config` (self)
- def `gbtx_config` (self, filename, gbtx=0)

Additional Inherited Members

5.170.1 Detailed Description

wishbone slave instatiated inside the usb_if executing the control over the DP2, DP3 fifos and the DP1, DP0 errors

5.170.2 Constructor & Destructor Documentation

5.170.2.1 `__init__()`

```
def ws_i2c_gbtx.WsI2cGbtx.__init__ (
    self,
    moduleid,
    board_obj )
```

init

5.170.3 Member Function Documentation

5.170.3.1 `dump_config()`

```
def ws_i2c_gbtx.WsI2cGbtx.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.170.3.2 dump_gbtx_config()

```
def ws_i2c_gbtx.WsI2cGbtx.dump_gbtx_config (
    self,
    gbtx )
```

Dump the modules state and configuration as a string

5.170.3.3 gbtx_config()

```
def ws_i2c_gbtx.WsI2cGbtx.gbtx_config (
    self,
    filename,
    gbtx = 0 )
```

Write GBTx xml configuration data in file "filename" to GBTx "gbtx"

5.170.3.4 get_address()

```
def ws_i2c_gbtx.WsI2cGbtx.get_address (
    self,
    gbtx = 0,
    commitTransaction = True )
```

Gets the address for the next I2C transaction

5.170.3.5 read_data()

```
def ws_i2c_gbtx.WsI2cGbtx.read_data (
    self,
    gbtx,
    address,
    commitTransaction = True )
```

Reads the data to the corresponding GBTx register

5.170.3.6 set_address()

```
def ws_i2c_gbtx.WsI2cGbtx.set_address (
    self,
    address,
    gbtx = 0,
    commitTransaction = True )
```

Sets the address for the next I2C transaction in GBTx gbtx

5.170.3.7 write_data()

```
def ws_i2c_gbtx.WsI2cGbtx.write_data (
    self,
    gbtx,
    address,
    data,
    commitTransaction = True )
```

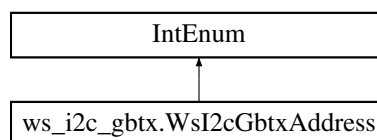
Writes the data to the corresponding GBTx register

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_i2c_gbtx.py

5.171 ws_i2c_gbtx.WsI2cGbtxAddress Class Reference

Inheritance diagram for ws_i2c_gbtx.WsI2cGbtxAddress:



Static Public Attributes

- int **ADDRESS_GBTX0** = 0
- int **ADDRESS_GBTX1** = 1
- int **ADDRESS_GBTX2** = 2
- int **DATA** = 3

5.171.1 Detailed Description

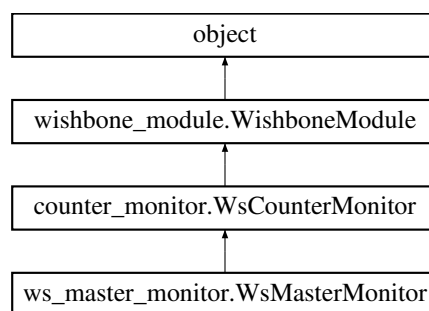
memory mapping for the ws_i2gbtx got from ?

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_i2c_gbtx.py

5.172 ws_master_monitor.WsMasterMonitor Class Reference

Inheritance diagram for ws_master_monitor.WsMasterMonitor:



Public Member Functions

- def **__init__** (self, moduleid, board_obj)
- def [read_counters](#) (self)
- def [dump_config](#) (self)
- def [get_mismatch](#) (self)
- def [dp23_monitor_latch_counters](#) (self)
- def [dp23_monitor_rst_counters](#) (self)
- def **read_dp23_status** (self)
- def [dp23_monitor_get_register](#) (self, verbose=True)
- def [get_dp_data_producer](#) (self, verbose=True)
- def [set_dp_data_producer](#) (self, DP2=None, DP3=None, verbose=True, commitTransaction=True)

Additional Inherited Members

5.172.1 Detailed Description

Wishbone slave instantiated inside the wishbone master handling the DP1, DP0 errors

5.172.2 Member Function Documentation

5.172.2.1 dp23_monitor_get_register()

```
def ws_master_monitor.WsMasterMonitor.dp23_monitor_get_register (
    self,
    verbose = True )
```

gets all the registers in the dp23 monitor registers.

5.172.2.2 dp23_monitor_latch_counters()

```
def ws_master_monitor.WsMasterMonitor.dp23_monitor_latch_counters (
    self )
```

latches all the counters in the dp23 fifos monitor

5.172.2.3 dp23_monitor_rst_counters()

```
def ws_master_monitor.WsMasterMonitor.dp23_monitor_rst_counters (
    self )
```

resets all the counters in the dp23 fifos monitor

5.172.2.4 dump_config()

```
def ws_master_monitor.WsMasterMonitor.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.172.2.5 get_dp_data_producer()

```
def ws_master_monitor.WsMasterMonitor.get_dp_data_producer (
    self,
    verbose = True )
```

returns the data producer address for the DP2 and DP3

5.172.2.6 get_mismatch()

```
def ws_master_monitor.WsMasterMonitor.get_mismatch (
    self )
```

gets the actual mismatch status

5.172.2.7 read_counters()

```
def ws_master_monitor.WsMasterMonitor.read_counters (
    self )
```

Reads all the values of the error counters in the wishbone master

5.172.2.8 set_dp_data_producer()

```
def ws_master_monitor.WsMasterMonitor.set_dp_data_producer (
    self,
    DP2 = None,
    DP3 = None,
    verbose = True,
    commitTransaction = True )
```

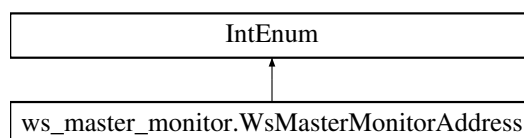
selects the DP2 and DP3 transceivers to be used

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_master_monitor.py

5.173 ws_master_monitor.WsMasterMonitorAddress Class Reference

Inheritance diagram for ws_master_monitor.WsMasterMonitorAddress:



Static Public Attributes

- int **LATCH_COUNTERS** = 0x00
- int **RESET_COUNTERS** = 0x01
- int **READ_WBM_WRERRCNTR** = 0x02
- int **READ_WBM_RDERRCNTR** = 0x03
- int **READ_WBM_SEEERRCNTR** = 0x04

5.173.1 Detailed Description

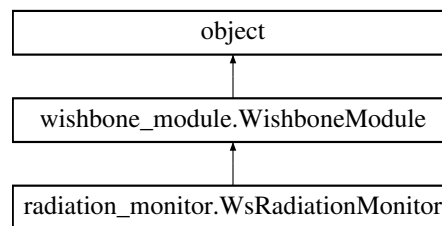
Memory mapping for the `ws_master_monitor` taken from `ws_master_pkg.vhd`

The documentation for this class was generated from the following file:

- `/media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_master_monitor.py`

5.174 radiation_monitor.WsRadiationMonitor Class Reference

Inheritance diagram for `radiation_monitor.WsRadiationMonitor`:



Public Member Functions

- `def __init__(self, moduleid, board_obj)`
- `def get_counters(self, module)`
- `def get_all(self)`
- `def dump_config(self)`
- `def dump_register_mapping(self)`

Public Attributes

- **verbose**
- **read_range_lut**

5.174.1 Detailed Description

`radiation monitor wishbone slave`

5.174.2 Member Function Documentation

5.174.2.1 dump_config()

```
def radiation_monitor.WsRadiationMonitor.dump_config (  
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.174.2.2 get_all()

```
def radiation_monitor.WsRadiationMonitor.get_all (  
    self )
```

returns the counters as a dictionary of tuples,
the key is the address of the registers

5.174.2.3 get_counters()

```
def radiation_monitor.WsRadiationMonitor.get_counters (  
    self,  
    module )
```

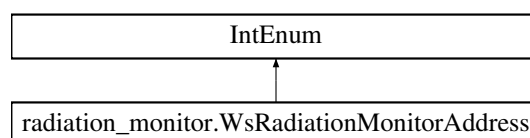
returns the counters for the given module as a dictionary of tuples,
the key is the address of the registers

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/radiation_monitor.py

5.175 radiation_monitor.WsRadiationMonitorAddress Class Reference

Inheritance diagram for radiation_monitor.WsRadiationMonitorAddress:



Static Public Attributes

- int **MODULE_0_FIRST** = 0x00
- int **MODULE_0_LAST** = MODULE_0_FIRST + WsRadiationMonitorSizes.MODULE_0_WIDTH -1
- int **MODULE_1_FIRST** = MODULE_0_LAST + 1
- int **MODULE_1_LAST** = MODULE_1_FIRST + WsRadiationMonitorSizes.MODULE_1_WIDTH -1
- int **MODULE_2_FIRST** = MODULE_1_LAST + 1
- int **MODULE_2_LAST** = MODULE_2_FIRST + WsRadiationMonitorSizes.MODULE_2_WIDTH -1
- int **MODULE_3_FIRST** = MODULE_2_LAST + 1
- int **MODULE_3_LAST** = MODULE_3_FIRST + WsRadiationMonitorSizes.MODULE_3_WIDTH -1
- int **MODULE_4_FIRST** = MODULE_3_LAST + 1
- int **MODULE_4_LAST** = MODULE_4_FIRST + WsRadiationMonitorSizes.MODULE_4_WIDTH -1
- int **MODULE_5_FIRST** = MODULE_4_LAST + 1
- int **MODULE_5_LAST** = MODULE_5_FIRST + WsRadiationMonitorSizes.MODULE_5_WIDTH -1
- int **MODULE_6_FIRST** = MODULE_5_LAST + 1
- int **MODULE_6_LAST** = MODULE_6_FIRST + WsRadiationMonitorSizes.MODULE_6_WIDTH -1
- int **MODULE_7_FIRST** = MODULE_6_LAST + 1
- int **MODULE_7_LAST** = MODULE_7_FIRST + WsRadiationMonitorSizes.MODULE_7_WIDTH -1
- int **MODULE_DUMMY_FIRST** = MODULE_7_LAST + 1
- int **MODULE_DUMMY_LAST** = MODULE_DUMMY_FIRST + WsRadiationMonitorSizes.MODULE_DUM←
MY_WIDTH -1
- int **NUM_REGS** = MODULE_DUMMY_LAST + 1

5.175.1 Detailed Description

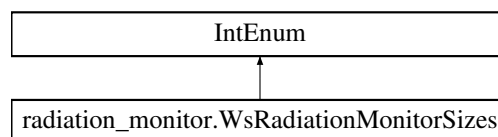
memory mapping for radiation monitor module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/radiation_monitor.py

5.176 radiation_monitor.WsRadiationMonitorSizes Class Reference

Inheritance diagram for radiation_monitor.WsRadiationMonitorSizes:



Static Public Attributes

- int **MODULE_0_WIDTH** = 11
- int **MODULE_1_WIDTH** = 5
- int **MODULE_2_WIDTH** = 5
- int **MODULE_3_WIDTH** = 7
- int **MODULE_4_WIDTH** = 4
- int **MODULE_5_WIDTH** = 1
- int **MODULE_6_WIDTH** = 10
- int **MODULE_7_WIDTH** = 7
- int **MODULE_DUMMY_WIDTH** = 1
- int **MODULE_NUM** = 8

5.176.1 Detailed Description

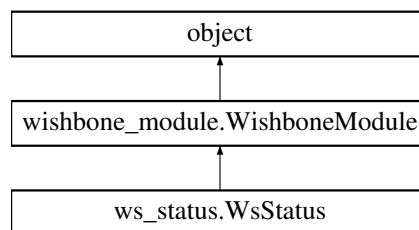
Used for memory mapping

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/radiation_monitor.py

5.177 ws_status.WsStatus Class Reference

Inheritance diagram for ws_status.WsStatus:



Public Member Functions

- def `__init__` (self, moduleid, board_obj)
- def `get_git_hash` (self)
- def `get_date` (self)
- def `get_os` (self)
- def `get_dipswitch` (self)
- def `get_dna_value` (self)
- def `check_git_hash_and_date` (self, expected_git_hash=None)
- def `dump_config` (self)

Additional Inherited Members

5.177.1 Detailed Description

wishbone slave used to identify the firmware and the FPGA

5.177.2 Constructor & Destructor Documentation

5.177.2.1 `__init__()`

```
def ws_status.WsStatus.__init__ (
    self,
    moduleid,
    board_obj )
```

init

5.177.3 Member Function Documentation

5.177.3.1 `check_git_hash_and_date()`

```
def ws_status.WsStatus.check_git_hash_and_date (
    self,
    expected_git_hash = None )
```

gets git hash and git date

5.177.3.2 `dump_config()`

```
def ws_status.WsStatus.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.177.3.3 `get_date()`

```
def ws_status.WsStatus.get_date (
    self )
```

Returns the git date

5.177.3.4 get_dipswitch()

```
def ws_status.WsStatus.get_dipswitch (
    self )
```

Returns value of RU DIPSWITCH

5.177.3.5 get_dna_value()

```
def ws_status.WsStatus.get_dna_value (
    self )
```

Returns the unique DNA value of the FPGA, in simulation it returns 0x76543210FEDCBA9876543210

5.177.3.6 get_git_hash()

```
def ws_status.WsStatus.get_git_hash (
    self )
```

Gets git hash

5.177.3.7 get_os()

```
def ws_status.WsStatus.get_os (
    self )
```

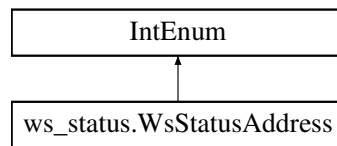
Returns the OS code
0: Unix
1: Windows
2: others
0xaffe: simulation

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_status.py

5.178 ws_status.WsStatusAddress Class Reference

Inheritance diagram for ws_status.WsStatusAddress:



Static Public Attributes

- int **GITHASH_LSB** = 0x00
- int **GITHASH_MSB** = 0x01
- int **DATE_LSB** = 0x02
- int **DATE_CSB** = 0x03
- int **DATE_MSB** = 0x04
- int **OS_LSB** = 0x05
- int **DIPSWITCH_VAL** = 0x06
- int **DNA_DO_READ** = 0x07
- int **DNA_CHUNK_0** = 0x08
- int **DNA_CHUNK_1** = 0x09
- int **DNA_CHUNK_2** = 0x0A
- int **DNA_CHUNK_3** = 0x0B
- int **DNA_CHUNK_4** = 0x0C
- int **DNA_CHUNK_5** = 0x0D

5.178.1 Detailed Description

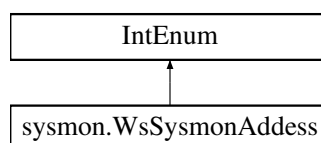
memory mapping for the ws_status got from ws_status_pkg.vhd

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_status.py

5.179 sysmon.WsSysmonAddress Class Reference

Inheritance diagram for sysmon.WsSysmonAddress:



Static Public Attributes

- int **TMR_MISC** = 0x00
- int **DRP_ADDR** = 0x01
- int **DRP_DATA** = 0x02

5.179.1 Detailed Description

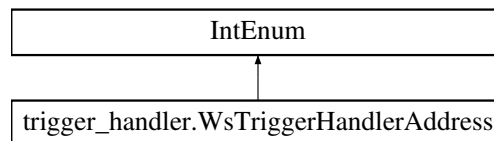
Wishbone Register mapping for SYSMON

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/sysmon.py

5.180 trigger_handler.WsTriggerHandlerAddress Class Reference

Inheritance diagram for trigger_handler.WsTriggerHandlerAddress:



Static Public Attributes

- int **ENABLE** = 0x00
- int **TRIGGER_PERIOD** = 0x01
- int **PULSE_nTRIGGER** = 0x02
- int **TRIGGER_MIN_DISTANCE** = 0x03
- int **MODE** = 0x04
- int **OPCODE_GATING** = 0x05
- int **MISMATCH_WB** = 0x06
- int **TRIG_SOURCE** = 0x07

5.180.1 Detailed Description

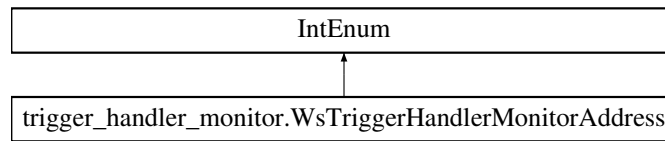
memory mapping for trigger handler module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/trigger_handler.py

5.181 trigger_handler_monitor.WsTriggerHandlerMonitorAddress Class Reference

Inheritance diagram for trigger_handler_monitor.WsTriggerHandlerMonitorAddress:



Static Public Attributes

- int **CNTR_RST** = 0x00
- int **CNTR_LATCH** = 0x01
- int **CNTR_TRIGGER_SENT_LSB** = 0x02
- int **CNTR_TRIGGER_SENT_MSB** = 0x03
- int **CNTR_TRIGGER_NOT_SENT** = 0x04
- int **CNTR_TRG_FIFO_GTH_FULL** = 0x05
- int **CNTR_TRG_FIFO_GTH_OVFL** = 0x06
- int **CNTR_TRG_FIFO_GPIO_FULL** = 0x07
- int **CNTR_TRG_FIFO_GPIO_OVFL** = 0x08
- int **CNTR_RX_FIFO_DATA_AVAILABLE_LSB** = 0x09
- int **CNTR_RX_FIFO_DATA_AVAILABLE_MSB** = 0x0A
- int **CNTR_TRIGGER_GATED_LSB** = 0x0B
- int **CNTR_TRIGGER_GATED_MSB** = 0x0C

5.181.1 Detailed Description

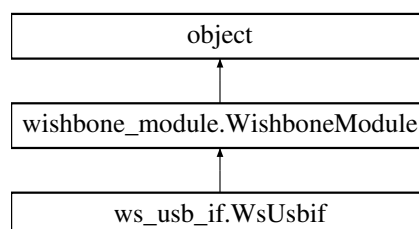
memory mapping for trigger handler module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/trigger_handler_monitor.py

5.182 ws_usb_if.WsUsbif Class Reference

Inheritance diagram for ws_usb_if.WsUsbif:



Public Member Functions

- `def __init__ (self, moduleid, board_obj)`
- `def dp23_monitor_latch_counters (self)`
- `def dp23_monitor_rst_counters (self)`
- `def read_dp23_status (self)`
- `def dp23_monitor_get_register (self, verbose=True)`
- `def get_dp_data_producer (self, verbose=True)`
- `def set_dp_data_producer (self, DP2=None, DP3=None, verbose=True, commitTransaction=True)`
- `def dump_config (self)`

Public Attributes

- `transceiver`
- `dp_map`

5.182.1 Detailed Description

wishbone slave instantiated inside the usb_if executing the control over the DP2, DP3 fifos and the DP1, DP0 errors

5.182.2 Constructor & Destructor Documentation

5.182.2.1 __init__()

```
def ws_usb_if.WsUsbif.__init__ (
    self,
    moduleid,
    board_obj )
```

init

5.182.3 Member Function Documentation

5.182.3.1 dp23_monitor_get_register()

```
def ws_usb_if.WsUsbif.dp23_monitor_get_register (
    self,
    verbose = True )
```

gets all the registers in the dp23 monitor registers.
Note that a fetch command is required before executing this command

5.182.3.2 dp23_monitor_latch_counters()

```
def ws_usb_if.WsUsbif.dp23_monitor_latch_counters (
    self )
```

latches all the counters in the dp23 fifos monitor

5.182.3.3 dp23_monitor_rst_counters()

```
def ws_usb_if.WsUsbif.dp23_monitor_rst_counters (
    self )
```

resets all the counters in the dp23 fifos monitor

5.182.3.4 dump_config()

```
def ws_usb_if.WsUsbif.dump_config (
    self )
```

Dump the modules state and configuration as a string

Reimplemented from [wishbone_module.WishboneModule](#).

5.182.3.5 get_dp_data_producer()

```
def ws_usb_if.WsUsbif.get_dp_data_producer (
    self,
    verbose = True )
```

returns the data producer address for the DP2 and DP3

5.182.3.6 set_dp_data_producer()

```
def ws_usb_if.WsUsbif.set_dp_data_producer (
    self,
    DP2 = None,
    DP3 = None,
    verbose = True,
    commitTransaction = True )
```

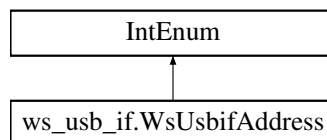
selects the DP2 and DP3 transceivers to be used

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_usb_if.py

5.183 ws_usb_if.WsUsbifAddress Class Reference

Inheritance diagram for ws_usb_if.WsUsbifAddress:



Static Public Attributes

- int **DP23_CMD** = 0x00
- int **READ_DP23_STS_DP2_WRDS_MSB** = 0x01
- int **READ_DP23_STS_DP2_WRDS_LSB** = 0x02
- int **READ_DP23_STS_DP2_OVFL** = 0x03
- int **READ_DP23_STS_DP3_WRDS_MSB** = 0x04
- int **READ_DP23_STS_DP3_WRDS_LSB** = 0x05
- int **READ_DP23_STS_DP3_OVFL** = 0x06
- int **CFG_DP23_PRODUCER_ADDRESS** = 0x07
- int **READ_DP23_STS_DP2_RDWRDS_MSB** = 0x08
- int **READ_DP23_STS_DP2_RDWRDS_LSB** = 0x09
- int **READ_DP23_STS_DP2_FULL** = 0x0A
- int **READ_DP23_STS_DP3_RDWRDS_MSB** = 0x0B
- int **READ_DP23_STS_DP3_RDWRDS_LSB** = 0x0C
- int **READ_DP23_STS_DP3_FULL** = 0x0D

5.183.1 Detailed Description

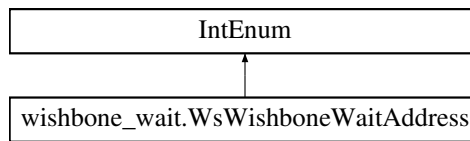
memory mapping for the ws_usbif got from ws_usbif_pkg.vhd

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/ws_usb_if.py

5.184 wishbone_wait.WsWishboneWaitAddress Class Reference

Inheritance diagram for wishbone_wait.WsWishboneWaitAddress:



Static Public Attributes

- int **WAIT_VALUE** = 0
- int **RST_CTRL_CNTRS** = 1
- int **READ_WAIT_EXEC_CNTR** = 2

5.184.1 Detailed Description

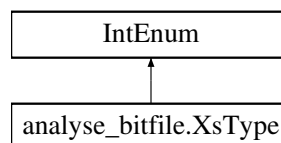
memory mapping for wishbone_wait/ctrl module

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/modules/board_support_software/software/py/wishbone_wait.py

5.185 analyse_bitfile.XsType Class Reference

Inheritance diagram for analyse_bitfile.XsType:



Static Public Attributes

- int **PRAGUE** = 0
- int **OXFORD** = 1

The documentation for this class was generated from the following file:

- /media/sf_proj/RUv1_Test/software/py/flash_analysis/analyse_bitfile.py

Index

- `__init__`
 - `analyse_bitfile.BitFileAnalyser`, 95
 - `beam_shutter.BeamShutter`, 93
 - `flash_test.FlashChecker`, 160
 - `jcm_control_server.JcmControlServer`, 205
 - `mmcm_monitor.MmcmMonitor`, 206
 - `usb_communication.communication.PrefetchCommunicationCtrl.Dctrl`, 133
 - 268
 - `ws_i2c_gbtx.WsI2cGbtx`, 368
 - `ws_status.WsStatus`, 377
 - `ws_usb_if.WsUsbif`, 383
- `activate_channel`
 - `hameg.Hameg`, 193
- `activate_channels`
 - `hameg.Hameg`, 193
- `activate_output`
 - `hameg.Hameg`, 193
- `activate_output_all_channels`
 - `tti.TTI`, 347
- `activate_output_channel`
 - `tti.TTI`, 347
- `activate_output_channels`
 - `tti.TTI`, 347
- `align_transceivers`
 - `ru_transceiver.GpioFrontend`, 184
 - `ru_transceiver.GthFrontend`, 188
- `analyse_bitfile`
 - `analyse_bitfile.BitFileAnalyser`, 96
- `analyse_bitfile.BitFileAnalyser`, 95
- `__init__`, 95
 - `analyse_bitfile`, 96
- `analyse_bitfile.XsType`, 386
- `analyse_block`
 - `flash_test.FlashChecker`, 160
- `beam_off`
 - `beam_shutter.BeamShutter`, 93
 - `beam_shutter.DummyBeamShutter`, 146
- `beam_on`
 - `beam_shutter.BeamShutter`, 93
 - `beam_shutter.DummyBeamShutter`, 146
- `beam_shutter.BeamShutter`, 92
 - `__init__`, 93
 - `beam_off`, 93
 - `beam_on`, 93
 - `beam_shutter_2_off`, 94
 - `beam_shutter_2_on`, 94
- `beam_shutter.DummyBeamShutter`, 145
 - `beam_off`, 146
 - `beam_on`, 146
 - `beam_shutter_2_off`
 - `beam_shutter.BeamShutter`, 94
 - `beam_shutter_2_on`
 - `beam_shutter.BeamShutter`, 94
 - `busy_force`
 - `dctrl.Dctrl`, 133
 - `busy_force_release`
 - `dctrl.Dctrl`, 133
- `charm_testbeam`, 11
 - `charm_testbeam.BeamShutterDataPoint`, 94
 - `charm_testbeam.DataPoint`, 128
 - `charm_testbeam.ExitStatus`, 151
 - `charm_testbeam.ScrubbingCyclesDataPoint`, 300
 - `charm_testbeam.SensorPoweringScheme`, 302
 - `charm_testbeam.TestbeamTest`, 308
 - `check_status`, 310
 - `manual_single_scrub`, 310
 - `send_stimuli`, 310
 - `charm_testbeam.TestMode`, 325
 - `charm_testbeam_check_setup.TestHardwareConnections`, 324
 - `charm_testbeam_check_setup.TestSoftware`, 328
 - `charm_trigger.CharmTrigger`, 98
 - `get_last_spill`, 99
 - `check_git_hash_and_date`
 - `ru_board.RUv0_CRU`, 283
 - `ru_board.RUv1`, 288
 - `ws_status.WsStatus`, 378
 - `check_status`
 - `charm_testbeam.TestbeamTest`, 310
 - `validate_scrubbing.TestbeamTest`, 308
- `chip`, 12
 - `testbench.Testbench`, 313
- `chip.Chip`, 99
 - `configure_dtu`, 100
 - `dump_config`, 101
 - `initialize`, 101
 - `initialize_readout`, 101
 - `is_inner_barrel`, 101
 - `is_outer_barrel_master`, 102
 - `is_outer_barrel_slave`, 102
 - `propagate_clock`, 102
 - `propagate_comma`, 102
 - `propagate_data`, 103
 - `propagate_prbs`, 103
 - `read_reg`, 103
 - `read_region_reg`, 104
 - `reset`, 104

- reset_pll, [104](#)
- set_xoff, [104](#)
- trigger, [105](#)
- write_opcode, [105](#)
- write_read_test, [105](#)
- write_reg, [106](#)
- write_region_reg, [106](#)
- chunks
 - simulation_if, [22](#)
- clear
 - simulation_if.GbtXSim, [181](#)
 - simulation_if.UsbCommSim, [353](#)
- clear_cmueerrors
 - pALPIDE.Alpide, [37](#)
- clear_gbt_cru
 - simulation_if.GbtCruGbtXBridge, [169](#)
- clear_gbtX
 - simulation_if.GbtCruGbtXBridge, [169](#)
- clear_status_latch
 - sca_ru.Sca_RU, [294](#)
- clear_triggered_fuse
 - tii.TTI, [347](#)
- close_connections
 - usb_communication.communication.NetUsbComm, [209](#)
- config, [13](#)
- config.Analysis, [91](#)
 - set_basepath, [92](#)
- configure_channel
 - hameg.Hameg, [193](#)
 - tii.TTI, [347](#)
- configure_dtu
 - chip.Chip, [100](#)
 - pALPIDE.Alpide, [37](#)
- ConfigureBiasADC
 - power_unit.PowerUnit, [261](#)
- ConfigurePowerADC
 - power_unit.PowerUnit, [261](#)
- ConfigureRTD
 - power_unit.PowerUnit, [261](#)
- counter_monitor, [13](#)
- counter_monitor.WsCounterMonitor, [362](#)
 - latch_all_counters, [362](#)
 - reset_all_counters, [362](#)
 - reset_counter, [363](#)
- counter_monitor.WsCounterMonitorAddress, [363](#)
- dctrl, [13](#)
- dctrl.Dctrl, [132](#)
 - busy_force, [133](#)
 - busy_force_release, [133](#)
 - disable_manchester_tx, [133](#)
 - dump_config, [134](#)
 - enable_manchester_tx, [134](#)
 - force_phase, [134](#)
 - get_auto_phase_offset, [134](#)
 - get_busy_transceiver_mask, [135](#)
 - get_busy_transceiver_status, [135](#)
 - get_counters, [135](#)
 - get_dclk_mask, [135](#)
 - get_dclk_parallel, [135](#)
 - get_dctrl_mask, [136](#)
 - get_idelays, [136](#)
 - get_input, [136](#)
 - get_manchester_rx_detected, [136](#)
 - get_manchester_tx, [136](#)
 - get_mismatch, [137](#)
 - get_phase, [137](#)
 - get_wait_cycles, [137](#)
 - read_chip_reg, [137](#)
 - readreg_write_phase, [137](#)
 - release_phase_force, [138](#)
 - reset_counters, [138](#)
 - set_auto_phase_offset, [138](#)
 - set_busy_transceiver_mask, [138](#)
 - set_dclk_mask, [138](#)
 - set_dclk_parallel, [139](#)
 - set_dctrl_mask, [139](#)
 - set_idelays, [139](#)
 - set_input, [139](#)
 - set_wait_cycles, [140](#)
 - write_chip_opcode, [140](#)
 - write_chip_reg, [140](#)
- dctrl.WsDctrlAddress, [364](#)
- deactivate_channel
 - hameg.Hameg, [193](#)
- deactivate_channels
 - hameg.Hameg, [194](#)
- deactivate_output
 - hameg.Hameg, [194](#)
- deactivate_output_all_channels
 - tii.TTI, [348](#)
- deactivate_output_channel
 - tii.TTI, [348](#)
- deactivate_output_channels
 - tii.TTI, [348](#)
- diagnose_read_results
 - usb_communication.communication.Communication, [117](#)
- disable_controller
 - gbtx_controller.GBTXController, [174](#)
- disable_log
 - usb_communication.communication.Communication, [117](#)
- disable_manchester_tx
 - dctrl.Dctrl, [133](#)
- disable_rderr_exception
 - usb_communication.communication.Communication, [117](#)
- DisableBiasAll
 - power_unit.PowerUnit, [261](#)
- DisablePowerAll
 - power_unit.PowerUnit, [262](#)
- dp23_monitor_get_register
 - ws_master_monitor.WsMasterMonitor, [371](#)
 - ws_usb_if.WsUsbif, [383](#)
- dp23_monitor_latch_counters

- ws_master_monitor.WsMasterMonitor, 372
- ws_usb_if.WsUsbif, 383
- dp23_monitor_rst_counters
 - ws_master_monitor.WsMasterMonitor, 372
 - ws_usb_if.WsUsbif, 384
- drp_bridge, 14
- drp_bridge.DrpBridge, 141
 - read_drp, 141
 - write_drp, 141
- dump_config
 - chip.Chip, 101
 - dctrl.Dctrl, 134
 - gbtx_controller.GBTxController, 175
 - gbtx_flow_monitor.GbtFlowMonitor, 179
 - i2c.I2CModule, 197
 - mmcm_monitor.MmcmMonitor, 207
 - pALPIDE.Alpide, 38
 - pALPIDE3.PAlpide3, 218
 - proasic3.ProAsic3, 272
 - radiation_monitor.WsRadiationMonitor, 375
 - ru_chip_power.ChipPower, 108
 - ru_gbt_packer.GbtPacker, 171
 - sysmon.Sysmon, 305
 - trigger_handler.TriggerHandler, 341
 - trigger_handler_monitor.TriggerHandlerMonitor, 345
 - wishbone_module.WishboneModule, 357
 - wishbone_wait.WishboneWait, 360
 - ws_i2c_gbtX.WsI2cGbtX, 368
 - ws_master_monitor.WsMasterMonitor, 372
 - ws_status.WsStatus, 378
 - ws_usb_if.WsUsbif, 384
- dump_gbtX_config
 - ws_i2c_gbtX.WsI2cGbtX, 368
- enable_alignment
 - ru_transceiver.GpioFrontend, 184
 - ru_transceiver.GthFrontend, 188
- enable_chip2connector_lut
 - ru_board.RUv1, 288
- enable_controller
 - gbtx_controller.GBTxController, 175
- enable_data
 - ru_transceiver.GpioFrontend, 184
 - ru_transceiver.GthFrontend, 188
- enable_log
 - usb_communication.communication.Communication, 117
- enable_manchester_tx
 - dctrl.Dctrl, 134
- enable_prbs
 - ru_transceiver.GpioFrontend, 184
 - ru_transceiver.GthFrontend, 189
- enable_rdr_exception
 - usb_communication.communication.Communication, 118
- enable_realign
 - ru_transceiver.GpioFrontend, 185
- enable_voltages
 - ru_chip_power.ChipPower, 109
- EnableBiasAll
 - power_unit.PowerUnit, 262
- EnableBiasWithMask
 - power_unit.PowerUnit, 262
- EnablePower
 - power_unit.PowerUnit, 262
- EnablePowerAll
 - power_unit.PowerUnit, 263
- EnablePowerWithMask
 - power_unit.PowerUnit, 263
- enqueue_output
 - usb_communication.communication, 26
- events.Event, 148
- events.EventDecoder, 149
 - get_eventcount_map, 149
 - process, 149
 - set_max_errors, 150
- events.EventPlotter, 150
- eye_scan
 - ru_eyescan.EyeScan, 153
- eye_scan_adaptive
 - ru_eyescan.EyeScan, 154
- firmware_wait
 - wishbone_module.WishboneModule, 357
- flash_dump_page
 - proasic3.ProAsic3, 272
- flash_read_page
 - proasic3.ProAsic3, 272
- flash_test.FlashChecker, 159
 - __init__, 160
 - analyse_block, 160
 - set_first_byte_bug, 160
 - set_is_inverted, 160
- flash_write_firmware
 - proasic3.ProAsic3, 272
- flash_write_parameter_page
 - proasic3.ProAsic3, 273
- flush
 - ru_board.ReadoutBoard, 279
 - usb_communication.communication.Communication, 118
 - wishbone_module.WishboneModule, 357
- flush_dp0
 - simulation_if.UsbCommSim, 353
- flush_gbt_cru
 - simulation_if.GbtCruGbtXBridge, 169
- flush_gbtX
 - simulation_if.GbtCruGbtXBridge, 169
 - simulation_if.GbtXSim, 181
- force_phase
 - dctrl.Dctrl, 134
- formatter
 - RUv1_regression, 20
- formatter_ch
 - RUv1_regression, 20
- gbt_sca, 14

- gbt_sca.Sca, 289
 - init_communication, 290
 - initialize_i2c_channel, 290
 - read_adc_channel, 291
 - read_gbt_x_register, 291
 - set_adc_channel, 291
 - set_gpio, 291
 - set_gpio_direction, 292
 - set_i2c_w_ctrl_reg, 292
 - set_xcku_reset, 292
 - write_gbt_x_register, 292
 - write_gpio, 293
- gbt_sca.ScaBadErrorFlagError, 295
 - log_info, 295
- gbt_sca.ScaChannel, 296
- gbt_sca.ScaErrorFlag, 296
- gbt_sca.ScaGpio, 297
- gbt_sca.ScaI2cBadStatusError, 298
 - log_info, 298
- gbt_sca.ScaI2cCmd, 299
- gbt_sca.ScaI2cSpeed, 299
- gbt_sca.ScaI2cStatusFlag, 300
- gbtx_config
 - ws_i2c_gbt_x.WsI2cGbt_x, 369
- gbtx_controller, 14
- gbtx_controller.GBTxController, 174
 - disable_controller, 174
 - dump_config, 175
 - enable_controller, 175
 - get_bitslip_rx, 175
 - get_bitslip_tx, 175
 - get_idelay, 175
 - get_loss_lock_flag, 176
 - get_mismatch, 176
 - get_tx1_pattern, 176
 - get_tx_pattern, 176
 - reset_gbt_x_asic, 176
 - rst_loss_lock_flag, 177
 - set_bitslip_rx, 177
 - set_bitslip_tx, 177
 - set_idelay, 177
 - set_tx1_pattern, 178
 - set_tx_pattern, 178
- gbtx_controller.WsGbt_xControllerAddress, 365
- gbtx_flow_monitor, 15
- gbtx_flow_monitor.Gbt_xFlowMonitor, 178
 - dump_config, 179
 - latch_counters, 179
 - read_counters, 179
 - reset_counters, 180
- gbtx_flow_monitor.WsGbt_xFlowMonitorAddress, 366
- gbtx_flow_monitor.WsGbt_xFlowMonitorAddress_bit, 367
- get_address
 - ws_i2c_gbt_x.WsI2cGbt_x, 369
- get_all
 - radiation_monitor.WsRadiationMonitor, 375
- get_auto_phase_offset
 - dctrl.Dctrl, 134
- get_bias_enable_status
 - power_unit.PowerUnit, 263
- get_bitslip_rx
 - gbtx_controller.GBTxController, 175
- get_bitslip_tx
 - gbtx_controller.GBTxController, 175
- get_busy_transceiver_mask
 - dctrl.Dctrl, 135
- get_busy_transceiver_status
 - dctrl.Dctrl, 135
- get_cc_command_register
 - proasic3.ProAsic3, 273
- get_channel_config
 - hameg.Hameg, 194
 - tii.TTI, 348
- get_channel_config_all
 - hameg.Hameg, 194
 - tii.TTI, 349
- get_chip2connector_lut
 - ru_board.RUv1, 288
- get_clock_config
 - proasic3.ProAsic3, 273
- get_clock_status
 - proasic3.ProAsic3, 274
- get_counters
 - dctrl.Dctrl, 135
 - mmcm_monitor.MmcmMonitor, 207
 - radiation_monitor.WsRadiationMonitor, 375
 - trigger_handler_monitor.TriggerHandlerMonitor, 345
 - wishbone_wait.WishboneWait, 361
- get_current
 - hameg.Hameg, 194
 - tii.TTI, 349
- get_data
 - simulation_if.Gbt_xSim, 181
- get_date
 - ws_status.WsStatus, 378
- get_dclk_mask
 - dctrl.Dctrl, 135
- get_dclk_parallel
 - dctrl.Dctrl, 135
- get_dctrl_mask
 - dctrl.Dctrl, 136
- get_dipswitch
 - ws_status.WsStatus, 378
- get_dna_value
 - ru_board.RUv0_CRU, 283
 - ru_board.RUv1, 288
 - ws_status.WsStatus, 379
- get_dp_data_producer
 - ws_master_monitor.WsMasterMonitor, 372
 - ws_usb_if.WsUsbif, 384
- get_drp_data
 - sysmon.Sysmon, 305
- get_eventcount_map
 - events.EventDecoder, 149

- get_fuse_triggered
 - hameg.Hameg, 195
 - tti.TTI, 349
- get_git_hash
 - ws_status.WsStatus, 379
- get_idelay
 - gbtx_controller.GBTxController, 175
- get_idelays
 - dctrl.Dctrl, 136
- get_input
 - dctrl.Dctrl, 136
- get_last_spill
 - charm_trigger.CharmTrigger, 99
- get_lock_counter
 - mmcm_monitor.MmcmMonitor, 207
- get_loss_lock_flag
 - gbtx_controller.GBTxController, 176
- get_manchester_rx_detected
 - dctrl.Dctrl, 136
- get_manchester_tx
 - dctrl.Dctrl, 136
- get_microprocessor_counters
 - ru_board.RUv0_CRU, 283
- get_mismatch
 - dctrl.Dctrl, 137
 - gbtx_controller.GBTxController, 176
 - mmcm_monitor.MmcmMonitor, 207
 - sysmon.Sysmon, 306
 - ws_master_monitor.WsMasterMonitor, 372
- get_model
 - hameg.Hameg, 195
 - tti.TTI, 349
- get_opcode_gating
 - trigger_handler.TriggerHandler, 342
- get_operating_mode
 - trigger_handler.TriggerHandler, 342
- get_os
 - ws_status.WsStatus, 379
- get_phase
 - dctrl.Dctrl, 137
- get_power
 - hameg.Hameg, 195
 - tti.TTI, 349
- get_power_all
 - hameg.Hameg, 195
 - tti.TTI, 350
- get_power_enable_status
 - power_unit.PowerUnit, 263
- get_sequence
 - usb_communication.communication.PrefetchCommunication, 268
- get_status_latch
 - sca_ru.Sca_RU, 294
- get_swt
 - simulation_if.GbtxSim, 182
- get_transceivers
 - ru_transceiver.GpioFrontend, 185
 - ru_transceiver.GthFrontend, 189
- get_trigger_minimum_distance
 - trigger_handler.TriggerHandler, 342
- get_trigger_period
 - trigger_handler.TriggerHandler, 342
- get_trigger_source
 - trigger_handler.TriggerHandler, 343
- get_tx1_pattern
 - gbtx_controller.GBTxController, 176
- get_tx_pattern
 - gbtx_controller.GBTxController, 176
- get_version
 - proasic3.ProAsic3, 274
- get_vertical_range
 - ru_eyescan.EyeScanGth, 156
- get_voltage
 - hameg.Hameg, 195
 - tti.TTI, 350
- get_wait_cycles
 - dctrl.Dctrl, 137
- getreg_adc_avdd_value
 - pALPIDE.Alpide, 38
- getreg_adc_avss_value
 - pALPIDE.Alpide, 38
- getreg_adc_bg2v_value
 - pALPIDE.Alpide, 39
- getreg_adc_calibration_value
 - pALPIDE.Alpide, 39
- getreg_adc_ctrl
 - pALPIDE.Alpide, 39
- getreg_adc_dac_input_value
 - pALPIDE.Alpide, 40
- getreg_adc_dvdd_value
 - pALPIDE.Alpide, 41
- getreg_adc_dvss_value
 - pALPIDE.Alpide, 41
- getreg_adc_iaux2_value
 - pALPIDE.Alpide, 41
- getreg_adc_ibias_value
 - pALPIDE.Alpide, 42
- getreg_adc_idb_value
 - pALPIDE.Alpide, 42
- getreg_adc_iref_value
 - pALPIDE.Alpide, 42
- getreg_adc_ireset_value
 - pALPIDE.Alpide, 43
- getreg_adc_ithr_value
 - pALPIDE.Alpide, 43
- getreg_adc_t2v_value
 - pALPIDE.Alpide, 43
- getreg_adc_vcasn2_value
 - pALPIDE.Alpide, 44
- getreg_adc_vcasn_value
 - pALPIDE.Alpide, 44
- getreg_adc_vcasp_value
 - pALPIDE.Alpide, 44
- getreg_adc_vclip_value
 - pALPIDE.Alpide, 45
- getreg_adc_vpulseh_value

pALPIDE.Alpide, [45](#)
 getreg_adc_vpulsel_value
 pALPIDE.Alpide, [45](#)
 getreg_adc_vresetd_value
 pALPIDE.Alpide, [46](#)
 getreg_adc_vresetp_value
 pALPIDE.Alpide, [46](#)
 getreg_adc_vtemp_value
 pALPIDE.Alpide, [46](#)
 getreg_analog_monitor_and_override
 pALPIDE.Alpide, [47](#)
 pALPIDE3.PAlpide3, [219](#)
 getreg_buffer_bypass
 pALPIDE.Alpide, [48](#)
 pALPIDE3.PAlpide3, [219](#)
 getreg_busy_min_width
 pALPIDE.Alpide, [48](#)
 pALPIDE3.PAlpide3, [220](#)
 getreg_cmd
 pALPIDE.Alpide, [48](#)
 pALPIDE3.PAlpide3, [220](#)
 getreg_cmu_and_dmu_cfg
 pALPIDE.Alpide, [49](#)
 pALPIDE3.PAlpide3, [221](#)
 getreg_cmu_and_dmu_status
 pALPIDE.Alpide, [49](#)
 getreg_cmu_errors_counter
 pALPIDE3.PAlpide3, [221](#)
 getreg_dac_settings_cmu_io_buffers
 pALPIDE.Alpide, [50](#)
 pALPIDE3.PAlpide3, [222](#)
 getreg_dac_settings_dclk_and_mclk_io_buffers
 pALPIDE.Alpide, [50](#)
 pALPIDE3.PAlpide3, [222](#)
 getreg_disable_regions_lsbs
 pALPIDE.Alpide, [51](#)
 pALPIDE3.PAlpide3, [222](#)
 getreg_disable_regions_msbs
 pALPIDE.Alpide, [51](#)
 getreg_disableof_regions_msbs
 pALPIDE3.PAlpide3, [223](#)
 getreg_dmu_and_tru_state
 pALPIDE3.PAlpide3, [223](#)
 getreg_dmu_data_fifo_lsbs
 pALPIDE.Alpide, [51](#)
 getreg_dmu_data_fifo_msbs
 pALPIDE.Alpide, [52](#)
 getreg_dtu_cfg
 pALPIDE.Alpide, [52](#)
 pALPIDE3.PAlpide3, [224](#)
 getreg_dtu_dacs
 pALPIDE.Alpide, [53](#)
 pALPIDE3.PAlpide3, [224](#)
 getreg_dtu_pll_lock_1
 pALPIDE.Alpide, [53](#)
 pALPIDE3.PAlpide3, [225](#)
 getreg_dtu_pll_lock_2
 pALPIDE.Alpide, [53](#)
 pALPIDE3.PAlpide3, [225](#)
 getreg_dtu_test_1
 pALPIDE.Alpide, [54](#)
 pALPIDE3.PAlpide3, [225](#)
 getreg_dtu_test_2
 pALPIDE.Alpide, [54](#)
 pALPIDE3.PAlpide3, [226](#)
 getreg_dtu_test_3
 pALPIDE.Alpide, [55](#)
 pALPIDE3.PAlpide3, [226](#)
 getreg_fromu_cfg_1
 pALPIDE.Alpide, [55](#)
 pALPIDE3.PAlpide3, [227](#)
 getreg_fromu_cfg_2
 pALPIDE.Alpide, [56](#)
 pALPIDE3.PAlpide3, [227](#)
 getreg_fromu_cfg_3
 pALPIDE.Alpide, [56](#)
 getreg_fromu_pulsing1
 pALPIDE.Alpide, [56](#)
 pALPIDE3.PAlpide3, [228](#)
 getreg_fromu_pulsing_2
 pALPIDE.Alpide, [57](#)
 pALPIDE3.PAlpide3, [228](#)
 getreg_fromu_status_1
 pALPIDE.Alpide, [57](#)
 pALPIDE3.PAlpide3, [228](#)
 getreg_fromu_status_2
 pALPIDE.Alpide, [57](#)
 pALPIDE3.PAlpide3, [229](#)
 getreg_fromu_status_3
 pALPIDE.Alpide, [58](#)
 getreg_fromu_status_4
 pALPIDE.Alpide, [58](#)
 getreg_fromu_status_5
 pALPIDE.Alpide, [58](#)
 getreg_fuses_read_lsbs
 pALPIDE3.PAlpide3, [229](#)
 getreg_fuses_read_msbs
 pALPIDE3.PAlpide3, [229](#)
 getreg_fuses_write_lsbs
 pALPIDE3.PAlpide3, [230](#)
 getreg_fuses_write_msbs
 pALPIDE3.PAlpide3, [230](#)
 getreg_IAUX2
 pALPIDE.Alpide, [59](#)
 pALPIDE3.PAlpide3, [230](#)
 getreg_IBIAS
 pALPIDE.Alpide, [59](#)
 pALPIDE3.PAlpide3, [231](#)
 getreg_IDB
 pALPIDE.Alpide, [59](#)
 pALPIDE3.PAlpide3, [231](#)
 getreg_IRESET
 pALPIDE.Alpide, [60](#)
 pALPIDE3.PAlpide3, [231](#)
 getreg_ITHR
 pALPIDE.Alpide, [60](#)

- pALPIDE3.PAlpide3, 232
- getreg_mode_ctrl
 - pALPIDE.Alpide, 60
- getreg_periphery_ctrl
 - pALPIDE3.PAlpide3, 232
- getreg_pixel_cfg
 - pALPIDE.Alpide, 61
- getreg_pixel_cfg_1
 - pALPIDE3.PAlpide3, 233
- getreg_pixel_cfg_2
 - pALPIDE3.PAlpide3, 233
- getreg_pixel_cfg_3
 - pALPIDE3.PAlpide3, 234
- getreg_temperature_sensor
 - pALPIDE3.PAlpide3, 234
- getreg_VCASN
 - pALPIDE.Alpide, 61
 - pALPIDE3.PAlpide3, 234
- getreg_VCASN2
 - pALPIDE.Alpide, 62
 - pALPIDE3.PAlpide3, 235
- getreg_VCASP
 - pALPIDE.Alpide, 62
 - pALPIDE3.PAlpide3, 235
- getreg_VCLIP
 - pALPIDE.Alpide, 62
 - pALPIDE3.PAlpide3, 235
- getreg_VPULSEH
 - pALPIDE.Alpide, 63
 - pALPIDE3.PAlpide3, 236
- getreg_VPULSEL
 - pALPIDE.Alpide, 63
 - pALPIDE3.PAlpide3, 236
- getreg_VRESETD
 - pALPIDE.Alpide, 63
 - pALPIDE3.PAlpide3, 236
- getreg_VRESETP
 - pALPIDE.Alpide, 64
 - pALPIDE3.PAlpide3, 237
- getreg_VTEMP
 - pALPIDE.Alpide, 64
 - pALPIDE3.PAlpide3, 237
- GPIOSensorMask
 - testbench, 23
- GPIOSubsetMap
 - RUv1_regression, 21
 - testbench, 23
- hameg.Hameg, 192
 - activate_channel, 193
 - activate_channels, 193
 - activate_output, 193
 - configure_channel, 193
 - deactivate_channel, 193
 - deactivate_channels, 194
 - deactivate_output, 194
 - get_channel_config, 194
 - get_channel_config_all, 194
 - get_current, 194
 - get_fuse_triggered, 195
 - get_model, 195
 - get_power, 195
 - get_power_all, 195
 - get_voltage, 195
 - is_channel_on, 196
 - setup_logging, 196
- i2c, 15
- i2c.I2CModule, 196
 - dump_config, 197
- init_ADC
 - ru_chip_power.ChipPower, 109
- init_communication
 - gbt_sca.Sca, 290
 - sca_ru.Sca_RU, 294
- init_pixel_matrix
 - pALPIDE.Alpide, 64
 - pALPIDE3.PAlpide3, 237
- initialize
 - chip.Chip, 101
 - pALPIDE.Alpide, 65
 - pALPIDE3.PAlpide3, 238
 - ru_chip_power.ChipPower, 109
 - ru_eyescan.EyeScan, 154
 - ru_gbt_packer.GbtPacker, 171
 - ru_transceiver.GpioFrontend, 185
 - ru_transceiver.GthFrontend, 189
- initialize_i2c_channel
 - gbt_sca.Sca, 290
- initialize_readout
 - chip.Chip, 101
 - pALPIDE.Alpide, 65
 - pALPIDE3.PAlpide3, 238
- initialize_readout_slave
 - pALPIDE.Alpide, 65
- is_aligned
 - ru_transceiver.GpioFrontend, 185
 - ru_transceiver.GthFrontend, 189
- is_cdr_locked
 - ru_transceiver.GthFrontend, 190
- is_channel_on
 - hameg.Hameg, 196
 - tti.TTI, 350
- is_inner_barrel
 - chip.Chip, 101
- is_outer_barrel_master
 - chip.Chip, 102
- is_outer_barrel_slave
 - chip.Chip, 102
- is_readback_finished
 - jcm_control_server.JcmControlClient, 202
- is_reset_done
 - ru_transceiver.GthFrontend, 190
- is_stopped
 - jcm_control_server.JCM, 198
 - jcm_control_server.JcmControlClient, 202
 - usb_communication.communication.UsbCommServer, 351

- jcm_control_server.JCM, 197
 - is_stopped, 198
 - read_messages, 198
 - rename_readback_file, 198
 - start_blind_scrubber, 199
 - start_calibrate_device, 199
 - start_full_configure, 199
 - start_random_fault_injection, 199
 - start_readback, 200
 - stop_blind_scrubber, 200
 - stop_random_fault_injection, 200
 - test_function, 200
- jcm_control_server.JcmControlClient, 201
 - is_readback_finished, 202
 - is_stopped, 202
 - read_calibrate_device_results, 202
 - read_fault_injection_results, 202
 - read_full_configure_results, 202
 - read_messages, 203
 - read_start_blind_scrubber_results, 203
 - start_blind_scrubber, 203
 - start_calibrate_device, 203
 - start_full_configure, 203
 - start_random_fault_injection, 204
 - start_readback, 204
 - stop_blind_scrubber, 204
 - stop_random_fault_injection, 204
 - test_function, 204
- jcm_control_server.JcmControlServer, 205
 - __init__, 205
- jcm_control_server.NoReadBackFileError, 211
 - log_info, 211
- lab_check_setup.TestHardwareConnections, 323
- latch_all_counters
 - counter_monitor.WsCounterMonitor, 362
- latch_counters
 - gbtx_flow_monitor.GbtxFlowMonitor, 179
 - ru_datapath_monitor.DatapathMonitor, 125
 - ru_datapath_monitor.DatapathMonitorDual, 127
 - ru_gbt_packer.GbtPackerMonitor, 173
 - trigger_handler_monitor.TriggerHandlerMonitor, 345
- load_sequence
 - usb_communication.communication.PrefetchCommunication, 268
- log_currents
 - ru_chip_power.ChipPower, 109
- log_info
 - gbt_sca.ScaBadErrorFlagError, 295
 - gbt_sca.Sca2cBadStatusError, 298
 - jcm_control_server.NoReadBackFileError, 211
- LowerThresholdsToMin
 - power_unit.PowerUnit, 264
- manual_single_scrub
 - charm_testbeam.TestbeamTest, 310
- mask_all_pixels
 - pALPIDE.Alpide, 65
- mask_row
 - pALPIDE.Alpide, 66
- mmcm_monitor, 15
 - mmcm_monitor.MmcmMonitor, 206
 - __init__, 206
 - dump_config, 207
 - get_counters, 207
 - get_lock_counter, 207
 - get_mismatch, 207
 - mmcm_monitor.MmcmMonitorAddress, 208
- module_includes, 15
- modules_path
 - testbench, 24
- open
 - simulation_if.GbtCruGbtXBridge, 169
 - simulation_if.GbtXSim, 182
 - simulation_if.UsbCommSim, 353
- pALPIDE.Addr, 29
- pALPIDE.Alpide, 33
 - clear_cmuerors, 37
 - configure_dtu, 37
 - dump_config, 38
 - getreg_adc_avdd_value, 38
 - getreg_adc_avss_value, 38
 - getreg_adc_bg2v_value, 39
 - getreg_adc_calibration_value, 39
 - getreg_adc_ctrl, 39
 - getreg_adc_dac_input_value, 40
 - getreg_adc_dvdd_value, 41
 - getreg_adc_dvss_value, 41
 - getreg_adc_iaux2_value, 41
 - getreg_adc_ibias_value, 42
 - getreg_adc_idb_value, 42
 - getreg_adc_iref_value, 42
 - getreg_adc_ireset_value, 43
 - getreg_adc_ithr_value, 43
 - getreg_adc_t2v_value, 43
 - getreg_adc_vcasn2_value, 44
 - getreg_adc_vcasn_value, 44
 - getreg_adc_vcasp_value, 44
 - getreg_adc_vclip_value, 45
 - getreg_adc_vpulseh_value, 45
 - getreg_adc_vpulsel_value, 45
 - getreg_adc_vresetd_value, 46
 - getreg_adc_vresetp_value, 46
 - getreg_adc_vtemp_value, 46
 - getreg_analog_monitor_and_override, 47
 - getreg_buffer_bypass, 48
 - getreg_busy_min_width, 48
 - getreg_cmd, 48
 - getreg_cmu_and_dmu_cfg, 49
 - getreg_cmu_and_dmu_status, 49
 - getreg_dac_settings_cmu_io_buffers, 50
 - getreg_dac_settings_dclk_and_mclk_io_buffers, 50
 - getreg_disable_regions_lsbs, 51
 - getreg_disable_regions_msbs, 51

- getreg_dmu_data_fifo_lsbs, 51
- getreg_dmu_data_fifo_msbs, 52
- getreg_dtu_cfg, 52
- getreg_dtu_dacs, 53
- getreg_dtu_pll_lock_1, 53
- getreg_dtu_pll_lock_2, 53
- getreg_dtu_test_1, 54
- getreg_dtu_test_2, 54
- getreg_dtu_test_3, 55
- getreg_fromu_cfg_1, 55
- getreg_fromu_cfg_2, 56
- getreg_fromu_cfg_3, 56
- getreg_fromu_pulsing1, 56
- getreg_fromu_pulsing_2, 57
- getreg_fromu_status_1, 57
- getreg_fromu_status_2, 57
- getreg_fromu_status_3, 58
- getreg_fromu_status_4, 58
- getreg_fromu_status_5, 58
- getreg_IAUX2, 59
- getreg_IBIAS, 59
- getreg_IDB, 59
- getreg_IRESET, 60
- getreg_ITHR, 60
- getreg_mode_ctrl, 60
- getreg_pixel_cfg, 61
- getreg_VCASN, 61
- getreg_VCASN2, 62
- getreg_VCASP, 62
- getreg_VCLIP, 62
- getreg_VPULSEH, 63
- getreg_VPULSEL, 63
- getreg_VRESETD, 63
- getreg_VRESETP, 64
- getreg_VTEMP, 64
- init_pixel_matrix, 64
- initialize, 65
- initialize_readout, 65
- initialize_readout_slave, 65
- mask_all_pixels, 65
- mask_row, 66
- pattern30, 66
- propagate_clock, 66
- propagate_comma, 66
- propagate_data, 67
- propagate_prbs, 67
- pulse, 67
- pulse_all_pixels_disable, 68
- pulse_all_pixels_enable, 68
- pulse_pixel_enable, 68
- pulse_row_disable, 68
- pulse_row_enable, 69
- region_control_register_mask_all_double_columns, 69
- region_control_register_pulse_gating, 69
- region_control_register_unmask_all_double_columns, 69
- reset, 70
- reset_pll, 70
- set_default_variables, 70
- set_xoff, 70
- setreg_adc_ctrl, 71
- setreg_adc_dac_input_value_, 72
- setreg_analog_monitor_and_override, 72
- setreg_buffer_bypass, 73
- setreg_busy_min_width, 74
- setreg_cmd, 74
- setreg_cmu_and_dmu_cfg, 75
- setreg_dac_settings_cmu_io_buffers, 75
- setreg_dac_settings_dclk_and_mclk_io_buffers, 76
- setreg_disable_regions_lsbs, 76
- setreg_disable_regions_msbs, 77
- setreg_dtu_cfg, 77
- setreg_dtu_dacs, 78
- setreg_dtu_pll_lock_2, 78
- setreg_dtu_test_1, 79
- setreg_dtu_test_2, 79
- setreg_dtu_test_3, 80
- setreg_fromu_cfg_1, 80
- setreg_fromu_cfg_2, 81
- setreg_fromu_cfg_3, 81
- setreg_fromu_pulsing1, 82
- setreg_fromu_pulsing_2, 82
- setreg_IAUX2, 83
- setreg_IBIAS, 83
- setreg_IDB, 83
- setreg_IRESET, 84
- setreg_ITHR, 84
- setreg_mode_ctrl, 85
- setreg_pixel_cfg, 85
- setreg_VCASN, 86
- setreg_VCASN2, 86
- setreg_VCASP, 87
- setreg_VCLIP, 87
- setreg_VPULSEH, 87
- setreg_VPULSEL, 88
- setreg_VRESETD, 88
- setreg_VRESETP, 89
- setreg_VTEMP, 89
- trigger, 89
- unmask_all_pixels, 90
- unmask_col, 90
- unmask_pixel, 90
- unmask_reg, 90
- unmask_row, 91
- write_read_test, 91
- pALPIDE.CommandRegisterOpcode, 115
- pALPIDE Opcode, 212
- pALPIDE3.Addr, 31
- pALPIDE3 Opcode, 212
- pALPIDE3.PAlpide3, 215
- dump_config, 218
- getreg_analog_monitor_and_override, 219
- getreg_buffer_bypass, 219
- getreg_busy_min_width, 220

- getreg_cmd, [220](#)
- getreg_cmu_and_dmu_cfg, [221](#)
- getreg_cmu_errors_counter, [221](#)
- getreg_dac_settings_cmu_io_buffers, [222](#)
- getreg_dac_settings_dclk_and_mclk_io_buffers, [222](#)
- getreg_disable_regions_lsbs, [222](#)
- getreg_disableof_regions_msbs, [223](#)
- getreg_dmu_and_tru_state, [223](#)
- getreg_dtu_cfg, [224](#)
- getreg_dtu_dacs, [224](#)
- getreg_dtu_pll_lock_1, [225](#)
- getreg_dtu_pll_lock_2, [225](#)
- getreg_dtu_test_1, [225](#)
- getreg_dtu_test_2, [226](#)
- getreg_dtu_test_3, [226](#)
- getreg_fromu_cfg_1, [227](#)
- getreg_fromu_cfg_2, [227](#)
- getreg_fromu_pulsing1, [228](#)
- getreg_fromu_pulsing_2, [228](#)
- getreg_fromu_status_1, [228](#)
- getreg_fromu_status_2, [229](#)
- getreg_fuses_read_lsbs, [229](#)
- getreg_fuses_read_msbs, [229](#)
- getreg_fuses_write_lsbs, [230](#)
- getreg_fuses_write_msbs, [230](#)
- getreg_IAUX2, [230](#)
- getreg_IBIAS, [231](#)
- getreg_IDB, [231](#)
- getreg_IRESET, [231](#)
- getreg_ITHR, [232](#)
- getreg_periphery_ctrl, [232](#)
- getreg_pixel_cfg_1, [233](#)
- getreg_pixel_cfg_2, [233](#)
- getreg_pixel_cfg_3, [234](#)
- getreg_temperature_sensor, [234](#)
- getreg_VCASN, [234](#)
- getreg_VCASN2, [235](#)
- getreg_VCASP, [235](#)
- getreg_VCLIP, [235](#)
- getreg_VPULSEH, [236](#)
- getreg_VPULSEL, [236](#)
- getreg_VRESETD, [236](#)
- getreg_VRESETP, [237](#)
- getreg_VTEMP, [237](#)
- init_pixel_matrix, [237](#)
- initialize, [238](#)
- initialize_readout, [238](#)
- propagate_clock, [238](#)
- propagate_comma, [239](#)
- propagate_data, [239](#)
- propagate_prbs, [239](#)
- reset, [240](#)
- reset_pll, [240](#)
- set_default_variables, [240](#)
- set_xoff, [240](#)
- setreg_analog_monitor_and_override, [241](#)
- setreg_buffer_bypass, [242](#)
- setreg_busy_min_width, [242](#)
- setreg_cmd, [243](#)
- setreg_cmu_and_dmu_cfg, [243](#)
- setreg_dac_settings_cmu_io_buffers, [244](#)
- setreg_dac_settings_dclk_and_mclk_io_buffers, [244](#)
- setreg_disable_regions_lsbs, [245](#)
- setreg_disableof_regions_msbs, [245](#)
- setreg_dtu_cfg, [245](#)
- setreg_dtu_dacs, [246](#)
- setreg_dtu_pll_lock_2, [246](#)
- setreg_dtu_test_1, [247](#)
- setreg_dtu_test_2, [248](#)
- setreg_dtu_test_3, [248](#)
- setreg_fromu_cfg_1, [248](#)
- setreg_fromu_cfg_2, [249](#)
- setreg_fromu_pulsing1, [249](#)
- setreg_fromu_pulsing_2, [250](#)
- setreg_fuses_write_lsbs, [250](#)
- setreg_fuses_write_msbs, [251](#)
- setreg_IAUX2, [251](#)
- setreg_IBIAS, [251](#)
- setreg_IDB, [252](#)
- setreg_IRESET, [252](#)
- setreg_ITHR, [253](#)
- setreg_periphery_ctrl, [253](#)
- setreg_pixel_cfg_1, [254](#)
- setreg_pixel_cfg_2, [254](#)
- setreg_pixel_cfg_3, [255](#)
- setreg_VCASN, [255](#)
- setreg_VCASN2, [255](#)
- setreg_VCASP, [256](#)
- setreg_VCLIP, [256](#)
- setreg_VPULSEH, [257](#)
- setreg_VPULSEL, [257](#)
- setreg_VRESETD, [257](#)
- setreg_VRESETP, [258](#)
- setreg_VTEMP, [258](#)
- trigger, [259](#)
- pattern30
 - pALPIDE.Alpide, [66](#)
- power_on
 - power_supplies.ChipPowerSupply, [111](#)
 - power_supplies.ChipPowerSupplyHameg, [112](#)
 - power_supplies.ChipPowerSupplyHamegStave, [113](#)
 - power_supplies.ChipPowerSupplyInternal, [114](#)
 - power_supplies.ChipPowerSupplyMixed, [115](#)
 - power_supplies.CompositeChipPowerSupply, [121](#)
 - power_supplies.DummyPowerSupply, [147](#)
- power_supplies.ChipPowerSupply, [111](#)
 - power_on, [111](#)
- power_supplies.ChipPowerSupplyHameg, [111](#)
 - power_on, [112](#)
- power_supplies.ChipPowerSupplyHamegStave, [112](#)
 - power_on, [113](#)
- power_supplies.ChipPowerSupplyInternal, [113](#)
 - power_on, [114](#)

- power_supplies.ChipPowerSupplyMixed, 114
 - power_on, 115
- power_supplies.CompositeChipPowerSupply, 121
 - power_on, 121
- power_supplies.DummyPowerSupply, 146
 - power_on, 147
- power_unit, 16
- power_unit.PowerUnit, 259
 - ConfigureBiasADC, 261
 - ConfigurePowerADC, 261
 - ConfigureRTD, 261
 - DisableBiasAll, 261
 - DisablePowerAll, 262
 - EnableBiasAll, 262
 - EnableBiasWithMask, 262
 - EnablePower, 262
 - EnablePowerAll, 263
 - EnablePowerWithMask, 263
 - get_bias_enable_status, 263
 - get_power_enable_status, 263
 - LowerThresholdsToMin, 264
 - RaiseThresholdsToMax, 264
 - ReadBiasADC, 264
 - ReadBiasADCChannel, 264
 - ReadPowerADC, 264
 - ReadPowerADCChannel, 265
 - ReadRTD, 265
 - SetBiasVoltage, 265
 - SetPowerVoltage, 265
 - SetPowerVoltageAll, 266
 - SetThreshold, 266
 - SetThresholdAll, 266
 - SetThresholdWithMask, 266
- power_unit.Wbl2cPuAddress, 355
- prefetch
 - usb_communication.communication.PrefetchCommunication, 268
- print_info
 - usb_communication.communication.WishboneReadError, 359
- proasic3, 16
- proasic3.CcActiveState, 97
- proasic3.CcCmdOpcode, 97
- proasic3.CcStatus, 98
- proasic3.EccCmdFlags, 147
- proasic3.EccStatus, 148
- proasic3.FifoState, 157
- proasic3.FifoWriterCmdOpcode, 158
- proasic3.FifoWriterStatus, 158
- proasic3.FlashCmdOpcode, 161
- proasic3.FlashFsmBlockEraseControllerState, 161
- proasic3.FlashFsmNandControllerState, 163
- proasic3.FlashFsmPageReadController, 163
- proasic3.FlashFsmPageWriteController, 165
- proasic3.FlashFsmReadIdControllerState, 166
- proasic3.FlashSelectICOpcode, 167
- proasic3.FlashStatus, 167
- proasic3.Pa3Register, 213
- proasic3.ProAsic3, 270
 - dump_config, 272
 - flash_dump_page, 272
 - flash_read_page, 272
 - flash_write_firmware, 272
 - flash_write_parameter_page, 273
 - get_cc_command_register, 273
 - get_clock_config, 273
 - get_clock_status, 274
 - get_version, 274
 - read_fifo_reg, 274
 - read_reg, 274
 - set_cc_command_register, 274
 - set_clock_config, 275
 - set_ecc_command_register, 275
 - set_i2c_channel, 275
 - set_reset, 275
 - set_start_program, 276
 - write_fifo_reg, 276
 - write_reg, 276
- proasic3.RcCmdOpcode, 278
- proasic3.RcState, 278
- proasic3.SmapCmdOpcode, 303
- proasic3.SmapStatus, 304
- process
 - events.EventDecoder, 149
- propagate_clock
 - chip.Chip, 102
 - pALPIDE.Alpide, 66
 - pALPIDE3.PAlpide3, 238
- propagate_comma
 - chip.Chip, 102
 - pALPIDE.Alpide, 66
 - pALPIDE3.PAlpide3, 239
- propagate_data
 - chip.Chip, 103
 - pALPIDE.Alpide, 67
 - pALPIDE3.PAlpide3, 239
- propagate_prbs
 - chip.Chip, 103
 - pALPIDE.Alpide, 67
 - pALPIDE3.PAlpide3, 239
- pulse
 - pALPIDE.Alpide, 67
- pulse_all_pixels_disable
 - pALPIDE.Alpide, 68
- pulse_all_pixels_enable
 - pALPIDE.Alpide, 68
- pulse_pixel_enable
 - pALPIDE.Alpide, 68
- pulse_row_disable
 - pALPIDE.Alpide, 68
- pulse_row_enable
 - pALPIDE.Alpide, 69
- radiation_monitor, 17
- radiation_monitor.WsRadiationMonitor, 374
 - dump_config, 375
 - get_all, 375

- get_counters, 375
- radiation_monitor.WsRadiationMonitorAddress, 375
- radiation_monitor.WsRadiationMonitorSizes, 376
- RaiseThresholdsToMax
 - power_unit.PowerUnit, 264
- read
 - ru_board.ReadoutBoard, 280
 - wishbone_module.WishboneModule, 357
- read_adc_channel
 - gbt_sca.Sca, 291
- read_all
 - ru_board.ReadoutBoard, 280
 - wishbone_module.WishboneModule, 357
- read_all_counters
 - ru_datapath_monitor.DatapathMonitor, 125
 - ru_datapath_monitor.DatapathMonitorDual, 127
- read_calibrate_device_results
 - jcm_control_server.JcmControlClient, 202
- read_chip_reg
 - dctrl.Dctrl, 137
- read_counters
 - gbtx_flow_monitor.GbtxFlowMonitor, 179
 - ru_datapath_monitor.DatapathMonitor, 125
 - ru_datapath_monitor.DatapathMonitorDual, 127
 - ru_gbt_packer.GbtPackerMonitor, 173
 - ws_master_monitor.WsMasterMonitor, 373
- read_data
 - ws_i2c_gbtx.Wsl2cGbtX, 369
- read_dp2
 - usb_communication.communication.Communication, 118
- read_dp3
 - usb_communication.communication.Communication, 118
- read_drp
 - drp_bridge.DrpBridge, 141
 - ru_transceiver.GthFrontend, 190
- read_fault_injection_results
 - jcm_control_server.JcmControlClient, 202
- read_fifo_reg
 - proasic3.ProAsic3, 274
- read_full_configure_results
 - jcm_control_server.JcmControlClient, 202
- read_gbtx_data_from_usb
 - ru_board.RUv0_CRU, 283
- read_gbtx_register
 - gbt_sca.Sca, 291
- read_messages
 - jcm_control_server.JCM, 198
 - jcm_control_server.JcmControlClient, 203
 - usb_communication.communication.UsbCommServer, 351
- read_prbs_counter
 - ru_transceiver.GpioFrontend, 185
 - ru_transceiver.GthFrontend, 190
- read_reg
 - chip.Chip, 103
 - proasic3.ProAsic3, 274
- usb_communication.communication.Communication, 118
- read_region_reg
 - chip.Chip, 104
- read_results
 - usb_communication.communication.Communication, 119
- read_start_blind_scrubber_results
 - jcm_control_server.JcmControlClient, 203
- read_values
 - ru_chip_power.ChipPower, 110
- read_values_raw
 - ru_chip_power.ChipPower, 110
- readback_all_drp
 - ru_transceiver.GthFrontend, 190
- readback_jcm
 - validate_scrubbing.TestbeamTest, 308
- ReadBiasADC
 - power_unit.PowerUnit, 264
- ReadBiasADCChannel
 - power_unit.PowerUnit, 264
- readFromGbt
 - simulation_if.GbtxSim, 182
- ReadPowerADC
 - power_unit.PowerUnit, 264
- ReadPowerADCChannel
 - power_unit.PowerUnit, 265
- readreg_powerbank_switch
 - ru_chip_power.ChipPower, 110
- readreg_write_phase
 - dctrl.Dctrl, 137
- ReadRTD
 - power_unit.PowerUnit, 265
- region_control_register_mask_all_double_columns
 - pALPIDE.Alpide, 69
- region_control_register_pulse_gating
 - pALPIDE.Alpide, 69
- region_control_register_unmask_all_double_columns
 - pALPIDE.Alpide, 69
- register_read
 - usb_communication.communication.Communication, 119
- register_write
 - usb_communication.communication.Communication, 119
- release_phase_force
 - dctrl.Dctrl, 138
- rename_readback_file
 - jcm_control_server.JCM, 198
- reset
 - chip.Chip, 104
 - pALPIDE.Alpide, 70
 - pALPIDE3.PAlpide3, 240
- reset_all_counters
 - counter_monitor.WsCounterMonitor, 362
- reset_counter
 - counter_monitor.WsCounterMonitor, 363
- reset_counters

- dctrl.Dctrl, 138
- gbtx_flow_monitor.GbtxFlowMonitor, 180
- ru_datapath_monitor.DatapathMonitor, 125
- ru_datapath_monitor.DatapathMonitorDual, 127
- ru_gbt_packer.GbtPackerMonitor, 173
- trigger_handler_monitor.TriggerHandlerMonitor, 345
- reset_gbtx_asic
 - gbtx_controller.GBTxController, 176
- reset_lane_counters
 - ru_datapath_monitor.DatapathMonitor, 125
 - ru_datapath_monitor.DatapathMonitorDual, 128
- reset_pll
 - chip.Chip, 104
 - pALPIDE.Alpide, 70
 - pALPIDE3.PAlpide3, 240
- reset_prbs_counter
 - ru_transceiver.GpioFrontend, 186
 - ru_transceiver.GthFrontend, 191
- reset_receivers
 - ru_transceiver.GthFrontend, 191
- resume_eye_scan
 - ru_eyescan.EyeScan, 154
- rst_ctrl_cntrs
 - wishbone_wait.WishboneWait, 361
- rst_loss_lock_flag
 - gbtx_controller.GBTxController, 177
- ru_board, 17
- ru_board.ConfigurationMemoryMapping, 122
- ru_board.ReadoutBoard, 279
 - flush, 279
 - read, 280
 - read_all, 280
 - write, 280
- ru_board.RUv0_CRU, 281
 - check_git_hash_and_date, 283
 - get_dna_value, 283
 - get_microprocessor_counters, 283
 - read_gbtx_data_from_usb, 283
 - SCA_ADC_COUNTERS, 286
 - send_end_of_continuous, 283
 - send_end_of_triggered, 284
 - send_heartbeat, 284
 - send_start_of_continuous, 284
 - send_start_of_triggered, 284
 - send_trigger, 285
 - wait, 285
 - write_gbtx, 285
- ru_board.RUv1, 286
 - check_git_hash_and_date, 288
 - enable_chip2connector_lut, 288
 - get_chip2connector_lut, 288
 - get_dna_value, 288
 - set_chip2connector_lut, 289
 - wait, 289
- ru_chip_power, 17
- ru_chip_power.ChipPower, 108
 - dump_config, 108
 - enable_voltages, 109
 - init_ADC, 109
 - initialize, 109
 - log_currents, 109
 - read_values, 110
 - read_values_raw, 110
 - readreg_powerbank_switch, 110
 - set_voltages, 110
- ru_chip_power.ConversionParameter, 122
 - step_to_value, 122
 - value_to_step, 123
- ru_datapath_monitor, 18
- ru_datapath_monitor.DatapathMonitor, 124
 - latch_counters, 125
 - read_all_counters, 125
 - read_counters, 125
 - reset_counters, 125
 - reset_lane_counters, 125
 - wb_register_mapping, 126
- ru_datapath_monitor.DatapathMonitorDual, 126
 - latch_counters, 127
 - read_all_counters, 127
 - read_counters, 127
 - reset_counters, 127
 - reset_lane_counters, 128
- ru_eyescan.EyeScan, 153
 - eye_scan, 153
 - eye_scan_adaptive, 154
 - initialize, 154
 - resume_eye_scan, 154
- ru_eyescan.EyeScanGth, 155
 - get_vertical_range, 156
- ru_eyescan.EyeScanGtx, 156
- ru_gbt_packer, 18
- ru_gbt_packer.GbtPacker, 170
 - dump_config, 171
 - initialize, 171
 - update_masks, 171
- ru_gbt_packer.GbtPackerMonitor, 172
 - latch_counters, 173
 - read_counters, 173
 - reset_counters, 173
 - wb_register_mapping, 173
- ru_transceiver, 18
- ru_transceiver.GpioFrontend, 183
 - align_transceivers, 184
 - enable_alignment, 184
 - enable_data, 184
 - enable_prbs, 184
 - enable_realign, 185
 - get_transceivers, 185
 - initialize, 185
 - is_aligned, 185
 - read_prbs_counter, 185
 - reset_prbs_counter, 186
 - scan_idelays, 186
 - set_lane_chip_mask, 186
 - set_transceivers, 186

- ru_transceiver.GthFrontend, 187
 - align_transceivers, 188
 - enable_alignment, 188
 - enable_data, 188
 - enable_prbs, 189
 - get_transceivers, 189
 - initialize, 189
 - is_aligned, 189
 - is_cdr_locked, 190
 - is_reset_done, 190
 - read_drp, 190
 - read_prbs_counter, 190
 - readback_all_drp, 190
 - reset_prbs_counter, 191
 - reset_receivers, 191
 - set_transceivers, 191
 - write_drp, 191
- ru_wb2fifo.WishboneToFifo, 359
- RUv1_regression, 19
 - formatter, 20
 - formatter_ch, 20
 - GPIO_SUBSET_MAP, 21
 - SIMULATION, 21
- RUv1_regression.DctrlBaseTest, 141
- RUv1_regression.DctrlBaseTest.TestDctrl, 319
 - test_readback_wait_cycles, 319
- RUv1_regression.TestcaseBase, 315
 - send_end_of_continuous, 316
 - send_end_of_triggered, 316
 - send_idle, 316
 - send_start_of_continuous, 316
 - send_start_of_triggered, 317
 - sync, 317
- RUv1_regression.TestCRUSWTRaceCondition, 317
- RUv1_regression.TestDatapathMonitorModules, 318
- RUv1_regression.TestDctrl0, 320
- RUv1_regression.TestDctrl1, 320
- RUv1_regression.TestDctrl2, 321
- RUv1_regression.TestDctrl3, 321
- RUv1_regression.TestDctrl4, 322
- RUv1_regression.TestGbtXFlowMonitor, 322
- RUv1_regression.TestI2C, 324
- RUv1_regression.TestPythonScripts, 326
- RUv1_regression.TestScaReset, 326
 - test_reset, 327
- RUv1_regression.TestSimExitHandling, 327
- RUv1_regression.TestSysmon, 328
- RUv1_regression.TestTransceiverFrontend, 329
- RUv1_regression.TestTransceiverGpioFrontend, 329
- RUv1_regression.TestTransceiverGthFrontend, 330
 - test_drp, 331
- RUv1_regression.TestTriggerHandlerContinuousMode, 332
- RUv1_regression.TestTriggerHandlerNoMode, 333
 - test_mode_status_continuous, 333
 - test_mode_status_triggered, 333
- RUv1_regression.TestTriggerHandlerTriggeredMode, 334
 - test_send_multiple_triggers, 334
 - test_send_pulse, 334
 - test_trigger_gating, 335
 - test_trigger_too_close_reject, 335
- RUv1_regression.TestWb2Fifo, 335
- RUv1_regression.TestWishboneDualMaster, 336
- RUv1_regression.TestWishboneMaster, 337
 - test_access_nonexist, 337
- RUv1_regression.TestWishboneSlaves, 337
 - test_gbtX_flow_monitor, 339
 - test_sysmon, 339
 - test_wsmstr, 339
- RUv1_regression.TestWsRadiationMonitor, 339
- RUv1_regression.TestWsStatus, 340
- RUv1_regression.TriggerHandlerBaseTest, 344
- RUv1_regression.TriggerHandlerBaseTest.TestTriggerHandler, 331
- SCA_ADC_COUNTERS
 - ru_board.RUv0_CRU, 286
- sca_ru, 21
- sca_ru.RuScaReg, 281
- sca_ru.Sca_RU, 293
 - clear_status_latch, 294
 - get_status_latch, 294
 - init_communication, 294
- scan_idelays
 - ru_transceiver.GpioFrontend, 186
- send_end_of_continuous
 - ru_board.RUv0_CRU, 283
 - RUv1_regression.TestcaseBase, 316
- send_end_of_triggered
 - ru_board.RUv0_CRU, 284
 - RUv1_regression.TestcaseBase, 316
- send_heartbeat
 - ru_board.RUv0_CRU, 284
- send_idle
 - RUv1_regression.TestcaseBase, 316
- send_start_of_continuous
 - ru_board.RUv0_CRU, 284
 - RUv1_regression.TestcaseBase, 316
- send_start_of_triggered
 - ru_board.RUv0_CRU, 284
 - RUv1_regression.TestcaseBase, 317
- send_stimuli
 - charm_testbeam.TestbeamTest, 310
- send_trigger
 - ru_board.RUv0_CRU, 285
 - simulation_if.Wb2GbtXComm, 354
- set_adc_channel
 - gbt_sca.Sca, 291
- set_address
 - ws_i2c_gbtX.Wsl2cGbtX, 369
- set_auto_phase_offset
 - dctrl.Dctrl, 138
- set_basepath
 - config.Analysis, 92
- set_bitslip_rx
 - gbtX_controller.GBTXController, 177

- set_bitslip_tx
 - gbtx_controller.GBTxController, 177
- set_busy_transceiver_mask
 - dctrl.Dctrl, 138
- set_cc_command_register
 - proasic3.ProAsic3, 274
- set_chip2connector_lut
 - ru_board.RUV1, 289
- set_clock_config
 - proasic3.ProAsic3, 275
- set_dclk_mask
 - dctrl.Dctrl, 138
- set_dclk_parallel
 - dctrl.Dctrl, 139
- set_dctrl_mask
 - dctrl.Dctrl, 139
- set_default_variables
 - pALPIDE.Alpide, 70
 - pALPIDE3.PAlpide3, 240
- set_dp_data_producer
 - ws_master_monitor.WsMasterMonitor, 373
 - ws_usb_if.WsUsbif, 384
- set_drp_address
 - sysmon.Sysmon, 306
- set_drp_data
 - sysmon.Sysmon, 306
- set_ecc_command_register
 - proasic3.ProAsic3, 275
- set_first_byte_bug
 - flash_test.FlashChecker, 160
- set_gpio
 - gbt_sca.Sca, 291
- set_gpio_direction
 - gbt_sca.Sca, 292
- set_i2c_channel
 - proasic3.ProAsic3, 275
- set_i2c_w_ctrl_reg
 - gbt_sca.Sca, 292
- set_idelay
 - gbtx_controller.GBTxController, 177
- set_idelays
 - dctrl.Dctrl, 139
- set_input
 - dctrl.Dctrl, 139
- set_is_inverted
 - flash_test.FlashChecker, 160
- set_lane_chip_mask
 - ru_transceiver.GpioFrontend, 186
- set_max_errors
 - events.EventDecoder, 150
- set_opcode_gating
 - trigger_handler.TriggerHandler, 343
- set_reset
 - proasic3.ProAsic3, 275
- set_start_program
 - proasic3.ProAsic3, 276
- set_transceivers
 - ru_transceiver.GpioFrontend, 186
- ru_transceiver.GthFrontend, 191
- set_trigger_minimum_distance
 - trigger_handler.TriggerHandler, 343
- set_trigger_period
 - trigger_handler.TriggerHandler, 343
- set_trigger_source
 - trigger_handler.TriggerHandler, 344
- set_tx1_pattern
 - gbtx_controller.GBTxController, 178
- set_tx_pattern
 - gbtx_controller.GBTxController, 178
- set_voltages
 - ru_chip_power.ChipPower, 110
- set_wait_cycles
 - dctrl.Dctrl, 140
- set_xcku_reset
 - gbt_sca.Sca, 292
- set_xoff
 - chip.Chip, 104
 - pALPIDE.Alpide, 70
 - pALPIDE3.PAlpide3, 240
- SetBiasVoltage
 - power_unit.PowerUnit, 265
- SetPowerVoltage
 - power_unit.PowerUnit, 265
- SetPowerVoltageAll
 - power_unit.PowerUnit, 266
- setreg_adc_ctrl
 - pALPIDE.Alpide, 71
- setreg_adc_dac_input_value_
 - pALPIDE.Alpide, 72
- setreg_analog_monitor_and_override
 - pALPIDE.Alpide, 72
 - pALPIDE3.PAlpide3, 241
- setreg_buffer_bypass
 - pALPIDE.Alpide, 73
 - pALPIDE3.PAlpide3, 242
- setreg_busy_min_width
 - pALPIDE.Alpide, 74
 - pALPIDE3.PAlpide3, 242
- setreg_cmd
 - pALPIDE.Alpide, 74
 - pALPIDE3.PAlpide3, 243
- setreg_cmu_and_dmu_cfg
 - pALPIDE.Alpide, 75
 - pALPIDE3.PAlpide3, 243
- setreg_dac_settings_cmu_io_buffers
 - pALPIDE.Alpide, 75
 - pALPIDE3.PAlpide3, 244
- setreg_dac_settings_dclk_and_mclk_io_buffers
 - pALPIDE.Alpide, 76
 - pALPIDE3.PAlpide3, 244
- setreg_disable_regions_lsbs
 - pALPIDE.Alpide, 76
 - pALPIDE3.PAlpide3, 245
- setreg_disable_regions_msbs
 - pALPIDE.Alpide, 77
- setreg_disableeof_regions_msbs

- pALPIDE3.PAlpide3, [245](#)
- setreg_dtu_cfg
 - pALPIDE.Alpide, [77](#)
 - pALPIDE3.PAlpide3, [245](#)
- setreg_dtu_dacs
 - pALPIDE.Alpide, [78](#)
 - pALPIDE3.PAlpide3, [246](#)
- setreg_dtu_pll_lock_2
 - pALPIDE.Alpide, [78](#)
 - pALPIDE3.PAlpide3, [246](#)
- setreg_dtu_test_1
 - pALPIDE.Alpide, [79](#)
 - pALPIDE3.PAlpide3, [247](#)
- setreg_dtu_test_2
 - pALPIDE.Alpide, [79](#)
 - pALPIDE3.PAlpide3, [248](#)
- setreg_dtu_test_3
 - pALPIDE.Alpide, [80](#)
 - pALPIDE3.PAlpide3, [248](#)
- setreg_fromu_cfg_1
 - pALPIDE.Alpide, [80](#)
 - pALPIDE3.PAlpide3, [248](#)
- setreg_fromu_cfg_2
 - pALPIDE.Alpide, [81](#)
 - pALPIDE3.PAlpide3, [249](#)
- setreg_fromu_cfg_3
 - pALPIDE.Alpide, [81](#)
- setreg_fromu_pulsing1
 - pALPIDE.Alpide, [82](#)
 - pALPIDE3.PAlpide3, [249](#)
- setreg_fromu_pulsing_2
 - pALPIDE.Alpide, [82](#)
 - pALPIDE3.PAlpide3, [250](#)
- setreg_fuses_write_lsbs
 - pALPIDE3.PAlpide3, [250](#)
- setreg_fuses_write_msbs
 - pALPIDE3.PAlpide3, [251](#)
- setreg_IAUX2
 - pALPIDE.Alpide, [83](#)
 - pALPIDE3.PAlpide3, [251](#)
- setreg_IBIAS
 - pALPIDE.Alpide, [83](#)
 - pALPIDE3.PAlpide3, [251](#)
- setreg_IDB
 - pALPIDE.Alpide, [83](#)
 - pALPIDE3.PAlpide3, [252](#)
- setreg_IRESET
 - pALPIDE.Alpide, [84](#)
 - pALPIDE3.PAlpide3, [252](#)
- setreg_ITHR
 - pALPIDE.Alpide, [84](#)
 - pALPIDE3.PAlpide3, [253](#)
- setreg_mode_ctrl
 - pALPIDE.Alpide, [85](#)
- setreg_periphery_ctrl
 - pALPIDE3.PAlpide3, [253](#)
- setreg_pixel_cfg
 - pALPIDE.Alpide, [85](#)
- setreg_pixel_cfg_1
 - pALPIDE3.PAlpide3, [254](#)
- setreg_pixel_cfg_2
 - pALPIDE3.PAlpide3, [254](#)
- setreg_pixel_cfg_3
 - pALPIDE3.PAlpide3, [255](#)
- setreg_VCASN
 - pALPIDE.Alpide, [86](#)
 - pALPIDE3.PAlpide3, [255](#)
- setreg_VCASN2
 - pALPIDE.Alpide, [86](#)
 - pALPIDE3.PAlpide3, [255](#)
- setreg_VCASP
 - pALPIDE.Alpide, [87](#)
 - pALPIDE3.PAlpide3, [256](#)
- setreg_VCLIP
 - pALPIDE.Alpide, [87](#)
 - pALPIDE3.PAlpide3, [256](#)
- setreg_VPULSEH
 - pALPIDE.Alpide, [87](#)
 - pALPIDE3.PAlpide3, [257](#)
- setreg_VPULSEL
 - pALPIDE.Alpide, [88](#)
 - pALPIDE3.PAlpide3, [257](#)
- setreg_VRESETD
 - pALPIDE.Alpide, [88](#)
 - pALPIDE3.PAlpide3, [257](#)
- setreg_VRESETP
 - pALPIDE.Alpide, [89](#)
 - pALPIDE3.PAlpide3, [258](#)
- setreg_VTEMP
 - pALPIDE.Alpide, [89](#)
 - pALPIDE3.PAlpide3, [258](#)
- SetThreshold
 - power_unit.PowerUnit, [266](#)
- SetThresholdAll
 - power_unit.PowerUnit, [266](#)
- SetThresholdWithMask
 - power_unit.PowerUnit, [266](#)
- setup_boards
 - testbench.Testbench, [313](#)
- setup_comm_cru
 - testbench.Testbench, [313](#)
- setup_comm_rdo
 - testbench.Testbench, [313](#)
- setup_comms
 - testbench.Testbench, [313](#)
- setup_logging
 - hameg.Hameg, [196](#)
 - tti.TTI, [350](#)
- SIMULATION
 - RUv1_regression, [21](#)
- simulation_if, [21](#)
 - chunks, [22](#)
- simulation_if.GbtCruGbtXBridge, [168](#)
 - clear_gbt_cru, [169](#)
 - clear_gbtX, [169](#)
 - flush_gbt_cru, [169](#)

- flush_gbt, 169
- open, 169
- simulation_if.GbtSim, 180
 - clear, 181
 - flush_gbt, 181
 - get_data, 181
 - get_swt, 182
 - open, 182
 - readFromGbt, 182
 - writeToGbt, 182
- simulation_if.SimulationServer, 302
 - start, 303
- simulation_if.UsbCommSim, 352
 - clear, 353
 - flush_dp0, 353
 - open, 353
- simulation_if.Wb2GbtComm, 354
 - send_trigger, 354
- single_read
 - usb_communication.communication.Communication, 119
- single_write
 - usb_communication.communication.Communication, test_access_nonexist 120
- start
 - simulation_if.SimulationServer, 303
 - usb_communication.communication.UsbCommServer, 351
- start_blind_scrubber
 - jcm_control_server.JCM, 199
 - jcm_control_server.JcmControlClient, 203
- start_calibrate_device
 - jcm_control_server.JCM, 199
 - jcm_control_server.JcmControlClient, 203
- start_full_configure
 - jcm_control_server.JCM, 199
 - jcm_control_server.JcmControlClient, 203
- start_random_fault_injection
 - jcm_control_server.JCM, 199
 - jcm_control_server.JcmControlClient, 204
- start_readback
 - jcm_control_server.JCM, 200
 - jcm_control_server.JcmControlClient, 204
- start_recording
 - usb_communication.communication.PrefetchCommunication, 269
- step_to_value
 - ru_chip_power.ConversionParameter, 122
- stop
 - usb_communication.communication.Communication, test_send_pulse 120
 - usb_communication.communication.NetUsbComm, 209
 - usb_communication.communication.PrefetchCommunication, 269
 - usb_communication.communication.UsbCommServer, 352
- stop_blind_scrubber
 - jcm_control_server.JCM, 200
 - jcm_control_server.JcmControlClient, 204
- stop_random_fault_injection
 - jcm_control_server.JCM, 200
 - jcm_control_server.JcmControlClient, 204
- stop_recording
 - usb_communication.communication.PrefetchCommunication, 269
- sync
 - RUv1_regression.TestcaseBase, 317
- sysmon, 22
 - sysmon.DrpSysmonAddress, 142
 - sysmon.DrpSysmonConfigMasks, 144
 - sysmon.DrpSysmonFlagMasks, 145
 - sysmon.Sysmon, 304
 - dump_config, 305
 - get_drp_data, 305
 - get_mismatch, 306
 - set_drp_address, 306
 - set_drp_data, 306
 - sysmon.WsSysmonAddress, 380
- test_access_nonexist
 - RUv1_regression.TestWishboneMaster, 337
- test_chips
 - testbench.Testbench, 314
- test_chips_continuous
 - testbench.Testbench, 314
- test_chips_fast
 - testbench.Testbench, 314
- test_drp
 - RUv1_regression.TestTransceiverGthFrontend, 331
- test_function
 - jcm_control_server.JCM, 200
 - jcm_control_server.JcmControlClient, 204
- test_gbt_flow_monitor
 - RUv1_regression.TestWishboneSlaves, 339
- test_mode_status_continuous
 - RUv1_regression.TestTriggerHandlerNoMode, 333
- test_mode_status_triggered
 - RUv1_regression.TestTriggerHandlerNoMode, 333
- test_readback_wait_cycles
 - RUv1_regression.DctrlBaseTest.TestDctrl, 319
- test_reset
 - RUv1_regression.TestScaReset, 327
- test_send_multiple_triggers
 - RUv1_regression.TestTriggerHandlerTriggeredMode, 334
- test_send_pulse
 - RUv1_regression.TestTriggerHandlerTriggeredMode, 334
- test_sysmon
 - RUv1_regression.TestWishboneSlaves, 339
- test_trigger_gating
 - RUv1_regression.TestTriggerHandlerTriggeredMode, 335
- test_trigger_too_close_reject

- RUv1_regression.TestTriggerHandlerTriggeredMode, 335
- test_wsmstr
 - RUv1_regression.TestWishboneSlaves, 339
- testbench, 22
 - GPIO_SENSOR_MASK, 23
 - GPIO_SUBSET_MAP, 23
 - modules_path, 24
 - USB_COMM_EXEC, 24
- testbench.SensorMatrixPattern, 301
- testbench.Testbench, 311
 - chip, 313
 - setup_boards, 313
 - setup_comm_cru, 313
 - setup_comm_rdo, 313
 - setup_comms, 313
 - test_chips, 314
 - test_chips_continuous, 314
 - test_chips_fast, 314
- trigger, 24
 - chip.Chip, 105
 - pALPIDE.Alpide, 89
 - pALPIDE3.PAlpide3, 259
- trigger.BitMap, 96
- trigger_handler, 24
- trigger_handler.Pulse_nTrigger, 277
- trigger_handler.TriggerHandler, 341
 - dump_config, 341
 - get_opcode_gating, 342
 - get_operating_mode, 342
 - get_trigger_minimum_distance, 342
 - get_trigger_period, 342
 - get_trigger_source, 343
 - set_opcode_gating, 343
 - set_trigger_minimum_distance, 343
 - set_trigger_period, 343
 - set_trigger_source, 344
- trigger_handler.WsTriggerHandlerAddress, 381
- trigger_handler_monitor, 25
- trigger_handler_monitor.CountersBitMapping, 123
- trigger_handler_monitor.TriggerHandlerMonitor, 344
 - dump_config, 345
 - get_counters, 345
 - latch_counters, 345
 - reset_counters, 345
- trigger_handler_monitor.WsTriggerHandlerMonitorAddress, 382
- tti.TTI, 346
 - activate_output_all_channels, 347
 - activate_output_channel, 347
 - activate_output_channels, 347
 - clear_triggered_fuse, 347
 - configure_channel, 347
 - deactivate_output_all_channels, 348
 - deactivate_output_channel, 348
 - deactivate_output_channels, 348
 - get_channel_config, 348
 - get_channel_config_all, 349
 - get_current, 349
 - get_fuse_triggered, 349
 - get_model, 349
 - get_power, 349
 - get_power_all, 350
 - get_voltage, 350
 - is_channel_on, 350
 - setup_logging, 350
- unmask_all_pixels
 - pALPIDE.Alpide, 90
- unmask_col
 - pALPIDE.Alpide, 90
- unmask_pixel
 - pALPIDE.Alpide, 90
- unmask_reg
 - pALPIDE.Alpide, 90
- unmask_row
 - pALPIDE.Alpide, 91
- update_masks
 - ru_gbt_packer.GbtPacker, 171
- USB_COMM_EXEC
 - testbench, 24
- usb_communication.communication, 25
 - enqueue_output, 26
- usb_communication.communication.AddressMismatchError, 32
- usb_communication.communication.Communication, 116
 - diagnose_read_results, 117
 - disable_log, 117
 - disable_rdrerr_exception, 117
 - enable_log, 117
 - enable_rdrerr_exception, 118
 - flush, 118
 - read_dp2, 118
 - read_dp3, 118
 - read_reg, 118
 - read_results, 119
 - register_read, 119
 - register_write, 119
 - single_read, 119
 - single_write, 120
 - stop, 120
 - write_reg, 120
- usb_communication.communication.NetUsbComm, 208
 - close_connections, 209
 - stop, 209
- usb_communication.communication.NetUsbTimeoutException, 210
- usb_communication.communication.PrefetchCommunication, 267
 - __init__, 268
 - get_sequence, 268
 - load_sequence, 268
 - prefetch, 268
 - start_recording, 269
 - stop, 269
 - stop_recording, 269

- usb_communication.communication.PyUsbComm, 277
- usb_communication.communication.UsbCommServer, 351
 - is_stopped, 351
 - read_messages, 351
 - start, 351
 - stop, 352
- usb_communication.communication.WishboneReadError, 358
 - print_info, 359
- userdefinedexceptions, 26
- userdefinedexceptions.ChipError, 107
- userdefinedexceptions.ChipidMismatchError, 107
- userdefinedexceptions.DataReadbackMismatchError, 131
- userdefinedexceptions.ExternalShutdownError, 152
- validate_scrubbing, 26
- validate_scrubbing.DataPoint, 130
- validate_scrubbing.ExitStatus, 151
- validate_scrubbing.ScrubbingCyclesDataPoint, 301
- validate_scrubbing.TestbeamTest, 307
 - check_status, 308
 - readback_jcm, 308
- validate_scrubbing.TestMode, 325
- value_to_step
 - ru_chip_power.ConversionParameter, 123
- wait
 - ru_board.RUv0_CRU, 285
 - ru_board.RUv1, 289
 - wishbone_wait.WishboneWait, 361
- wb_register_mapping
 - ru_datapath_monitor.DatapathMonitor, 126
 - ru_gbt_packer.GbtPackerMonitor, 173
- wishbone_module, 27
- wishbone_module.WishboneModule, 356
 - dump_config, 357
 - firmware_wait, 357
 - flush, 357
 - read, 357
 - read_all, 357
 - write, 358
- wishbone_wait, 27
- wishbone_wait.WishboneWait, 360
 - dump_config, 360
 - get_counters, 361
 - rst_ctrl_cntrs, 361
 - wait, 361
- wishbone_wait.WsWishboneWaitAddress, 386
- write
 - ru_board.ReadoutBoard, 280
 - wishbone_module.WishboneModule, 358
- write_chip_opcode
 - dctrl.Dctrl, 140
- write_chip_reg
 - dctrl.Dctrl, 140
- write_data
 - ws_i2c_gbtx.Wsl2cGbtx, 370
- write_drp
 - drp_bridge.DrpBridge, 141
 - ru_transceiver.GthFrontend, 191
- write_fifo_reg
 - proasic3.ProAsic3, 276
- write_gbtx
 - ru_board.RUv0_CRU, 285
- write_gbtx_register
 - gbt_sca.Sca, 292
- write_gpio
 - gbt_sca.Sca, 293
- write_opcode
 - chip.Chip, 105
- write_read_test
 - chip.Chip, 105
 - pALPIDE.Alpide, 91
- write_reg
 - chip.Chip, 106
 - proasic3.ProAsic3, 276
 - usb_communication.communication.Communication, 120
- write_region_reg
 - chip.Chip, 106
- writeToGbt
 - simulation_if.GbtSim, 182
- ws_i2c_gbtx, 28
- ws_i2c_gbtx.Wsl2cGbtx, 367
 - __init__, 368
 - dump_config, 368
 - dump_gbtx_config, 368
 - gbtx_config, 369
 - get_address, 369
 - read_data, 369
 - set_address, 369
 - write_data, 370
- ws_i2c_gbtx.Wsl2cGbtxAddress, 370
- ws_master_monitor, 28
- ws_master_monitor.WsMasterMonitor, 371
 - dp23_monitor_get_register, 371
 - dp23_monitor_latch_counters, 372
 - dp23_monitor_rst_counters, 372
 - dump_config, 372
 - get_dp_data_producer, 372
 - get_mismatch, 372
 - read_counters, 373
 - set_dp_data_producer, 373
- ws_master_monitor.WsMasterMonitorAddress, 373
- ws_status, 28
- ws_status.WsStatus, 377
 - __init__, 377
 - check_git_hash_and_date, 378
 - dump_config, 378
 - get_date, 378
 - get_dipswitch, 378
 - get_dna_value, 379
 - get_git_hash, 379
 - get_os, 379
- ws_status.WsStatusAddress, 380

`ws_usb_if`, [28](#)
`ws_usb_if.WsUsbif`, [382](#)
 `__init__`, [383](#)
 `dp23_monitor_get_register`, [383](#)
 `dp23_monitor_latch_counters`, [383](#)
 `dp23_monitor_rst_counters`, [384](#)
 `dump_config`, [384](#)
 `get_dp_data_producer`, [384](#)
 `set_dp_data_producer`, [384](#)
`ws_usb_if.WsUsbifAddress`, [385](#)