Phenix Muon Identifier
Address Decoder
Functional Description

Ganesh Rao
ORNL
7/22/1998

Introduction

The address decoder is a programmable CPLD device that is used to as an interface between the backplane buses and the circuitry on a Readout Card (ROC) or Front End Module (FEM). It connects the Arcnet and FEM buses to the internal circuitry on a ROC. It acts like a bus multiplexer and connects the input bus to the selected output bus and allows for direct access to ROC devices like the Glink, serial strings, FPGA programming pins and Detector ID. The Address Decoder is designed in VHDL and implemented in an Altera Max 7000S series EPM7128SQC160-15 device. This device will retain its program and will be operational at powerup. The device is compiled in Max+Plus II with the "FAST" synthesis style and automatic pin placement. If the output slew rate is too fast for the board, the "Slow Slew Rate" option can be selected to increase rise times and reduce switching noise. The pins have been locked to the ones that were auto-selected and then the diagnostic outputs were added. For the diagnostic pins, the compiler should be set to "NORMAL" mode to ensure a fit.

Inputs and Outputs

The following sections list the inputs and outputs of the Address Decoder. If the name of a pin is preceded by the letter "n" then it is active low and the "n" is similar to a "/." The diagnostic outputs are for testing purposes only and will not appear in the actual chip. The chip pinouts are given in Appendix A.

Inputs

GA[4..0]	Geographical Address (Inverted) from backplane
GAP	Geographical Address Parity (Inverted) from backplane
Detector_ID[15..0]	Detector ID from DIP switches
mclk	Arcnet Mode Clock
mode[1..0]	Arcnet Mode
AddrDat[4..0]	Arcnet Address/Data bus
fp_confdone[2..0]	CONFDONE FPGA programming pins
fp_nstatus[2..0]	nSTATUS FPGA programming pins
sdin	Serial Data from ROC/FEM board to be sent to bus
sdin2	Serial Data from ROC/FEM board to be sent to bus
sdin3	Serial Data from ROC/FEM board to be sent to bus
glrx_stat0	STAT0 line from Glink receiver
gltx_locked	LOCKED line from Glink transmitter
gltx_rfd	RFD (request for data) from Glink transmitter
FemAddr[4..0]	FEM Address bus
nALE	Address Latch Enable from FEM bus

Outputs

pargoodo	Diagnostic output: Parity OK?
enableo[8..0]	Diagnostic output: Which function is enabled?
replyo	Diagnostic output: Will the board enable outputs?
mp_dat[1..0]	Microprocessor data lines that send board data back to Arcnet.
nArcBuf_En	Enable the backplane drivers for the Arcnet lines
fp_dclk	FPGA programming clock
fp_d0	FPGA programming serial data
fp_nconfig[2..0]	nCONFIG line for FPGA programming
glrx_nrst	Glink receiver /RESET line
gltx_ed	Glink transmitter ED line
gltx_ndav	Glink transmitter /DAV line
gltx_nrst	Glink transmitter /RESET line
sclk	Serial string #1: Serial Clock
slatch	Serial string #1: Serial Latch
sdout	Serial string #1: Serial data out
rdback	Serial string #1: Read Back serial data.
nsreset	Serial string #1: Serial reset
sclk2	Serial string #2: Serial Clock
slatch2	Serial string #2: Serial Latch
sdout2	Serial string #2: Serial data out
rdback2	Serial string #2: Read Back serial data.

nsreset2	Serial string #2: Serial reset
sclk3	Serial string #3: Serial Clock
slatch3	Serial string #3: Serial Latch
sdout3	Serial string #3: Serial data out
rdback3	Serial string #3: Read Back serial data.
nsreset3	Serial string #3: Serial reset
nFemBuf_En	Enable the backplane drivers for the FEM bus lines
nRD_EN	Enable the 5-event FIFO to read data

Arcnet Bus Interface

The Arcnet bus is the slow controls link from the microprocessor on the FEM board. Arcnet controls several ROC/FEM systems. The Address decoder links this bus with the peripherals on both the ROC and FEM boards. The decoder's functions are:

- Read the geographical address and only enable the selected function if the board is being addressed.
- Control the backplane driver chips to eliminate bus contention while returning data (nArcBuf_En).
- Enable the 5-wire Phenix serial strings (3 available) for board programming.
- Enable FPGA programming pins (3 available) for downloading the FPGA code.
- Enable Glink transmitter/receiver reset controls and status lines.
- Allow serial readout of the Detector ID from FEM board DIP switches.

The AddrDat bus is used to send both address and data to the decoder. The mode of operation is controlled by *mclk* and *mode[1..0]*. Two transfers are required to load the address. The first is the board address. The second transfer specifies the peripheral function desired. Then, the data mode is enabled and the AddrDat lines are connected directly to the output lines and they assume the specified functionality. Modes are activated on the rising edge of *mclk*. The modes are listed in Table 1

Table 1: Arcnet transfer modes

mode1	mode0	Function after mclk -
0	0	Load board geographic address
0	1	Load board function / destination
1	0	Enable selected function
1	1	Disable functions

The board address that is loaded is compared to the backplane geographic address (0x01 to 0x15 or 1-21). If they do not match, then the address decoder does not enable any functions as it is not being addressed. If it does match, then the selected function is enabled. If the function calls for a return of data, the reply lines *mp_dat[1..0]* are enabled. This is true for a single board being addressed. There is also a reserved address for broadcasting information. The Broadcast Address is "b11101." If this address is used, all the receivers are enabled but no data is returned. This function can be used to reset all the Glinks and later read the status lines or to mass program the FPGAs.

The functions available in the Address Decoder are listed in Table 2. Table 3 shows how the AddrDat lines are redefined after the selected function is enabled after a rising edge on *mclk* with *mode* = "b10." This means that as soon as the function is enabled, the AddrDat lines are directly connected to the peripherals with the newly defined lines. If there is a desired default state for the lines, they can be assigned to AddrDat before the mode is enabled and before it is disabled with *mode* = "b11."

A more detailed description of the different functions is given below. The VHDL files are included in Appendix B for reference.

Table 2: Addressable decoder functions

Function Number	Description
1	Serial string #1
2	Serial string #2
3	Serial string #3
4	FPGA programming pins #1
5	FPGA programming pins #2
6	FPGA programming pins #3
7	Glink transmitter control
8	Glink receiver control
9	Detector ID serial readback

Table 3: Data line remapping by function

Fcn. No.	AddrDat4	AddrDat3	AddrDat2	AddrDat1	AddrDat0	mp_dat1	mp_dat0
1	/SRESET	RDBACK	SDATA	SLATCH	SCLK	---	SDIN
2	/SRESET2	RDBACK2	SDATA2	SLATCH2	SCLK2	---	SDIN2
3	/SRESET3	RDBACK3	SDATA3	SLATCH3	SCLK3	---	SDIN3
4	---	---	nCONFIG0	DATA0	DCLK	CONFDONE0	nSTATUS0
5	---	---	nCONFIG1	DATA0	DCLK	CONFDONE1	nSTATUS1
6	---	---	nCONFIG2	DATA0	DCLK	CONFDONE2	nSTATUS2
7	---	---	ED	ED_latch	/Reset	---	Locked
8	---	---	---	---	/Reset	---	STAT0
9	---	---	---	Synch. Load	SCLK	ID_valid	ID_out

Serial Strings

When one of the serial string functions is selected, the standard Phenix 5-wire protocol is implemented with the AddrDat lines. The *SDIN* line is the data from the peripheral and is returned on *mp_dat0*.

FPGA Programming

This mode allows Arcnet to configure the FPGAs in the system. The pin names listed are for configuring Altera Flex10K style parts but the functionality is identical to other chips that support JTAG programming or other serial schemes. For details on configuration of Flex10K devices, please refer to Altera application note AN59. The FPGA programming outputs of the Address Decoder should be pulled up with resistors. This is because the outputs are high impedance when not in use. This allows the programming pins to be shared with a bit-blaster serial download cable as a backup or test method of programming the chips. All the programmable parts share the DCLK and DATA0 outputs.

Glink Control

Functions seven and eight control the Glink transmitter and receiver respectively. The HDMP-1024 TTL Glink transmitter and HDMP-1022 TTL Glink receiver chips are assumed to be the peripherals.

For the transmitter, there are four lines. The active-low reset line (*AddrDat(0) = /Reset*) is connected to the transmitter's *RST** (chip reset) line. The *ED* (Enable Data) line is connected to the transmitter's *ED* input. This output is latched and *ED_latch* controls the latch. When *ED_latch* is low, the previous output is stored. When it is high, the latch is transparent. This way, Arcnet can set *ED* to a desired value and that value will be preserved even when other functions are addressed. The latch output, *ED*, is undefined at powerup. It must either be set soon after or the output pin must be pulled down. The transmitter's *LOCKED* line is returned on *mp_dat0*. All of the Glink transmitters can be reset using broadcast mode and then the status can be read back one board at a time. The *ED* line can be set at the same time. This should speed the process of bringing up the Glinks at startup. The decoder also contains the interface logic to control the Glink transmitters */DAV* line given the *RFD* input from the HDMP-1024 chip. The logic implemented is: */DAV* = not(*RFD* **AND** *ED*).

The Glink receiver control is limited to the active-low reset line. The status of the receiver, *STAT0*, is returned on *mp_dat0*.

Detector ID

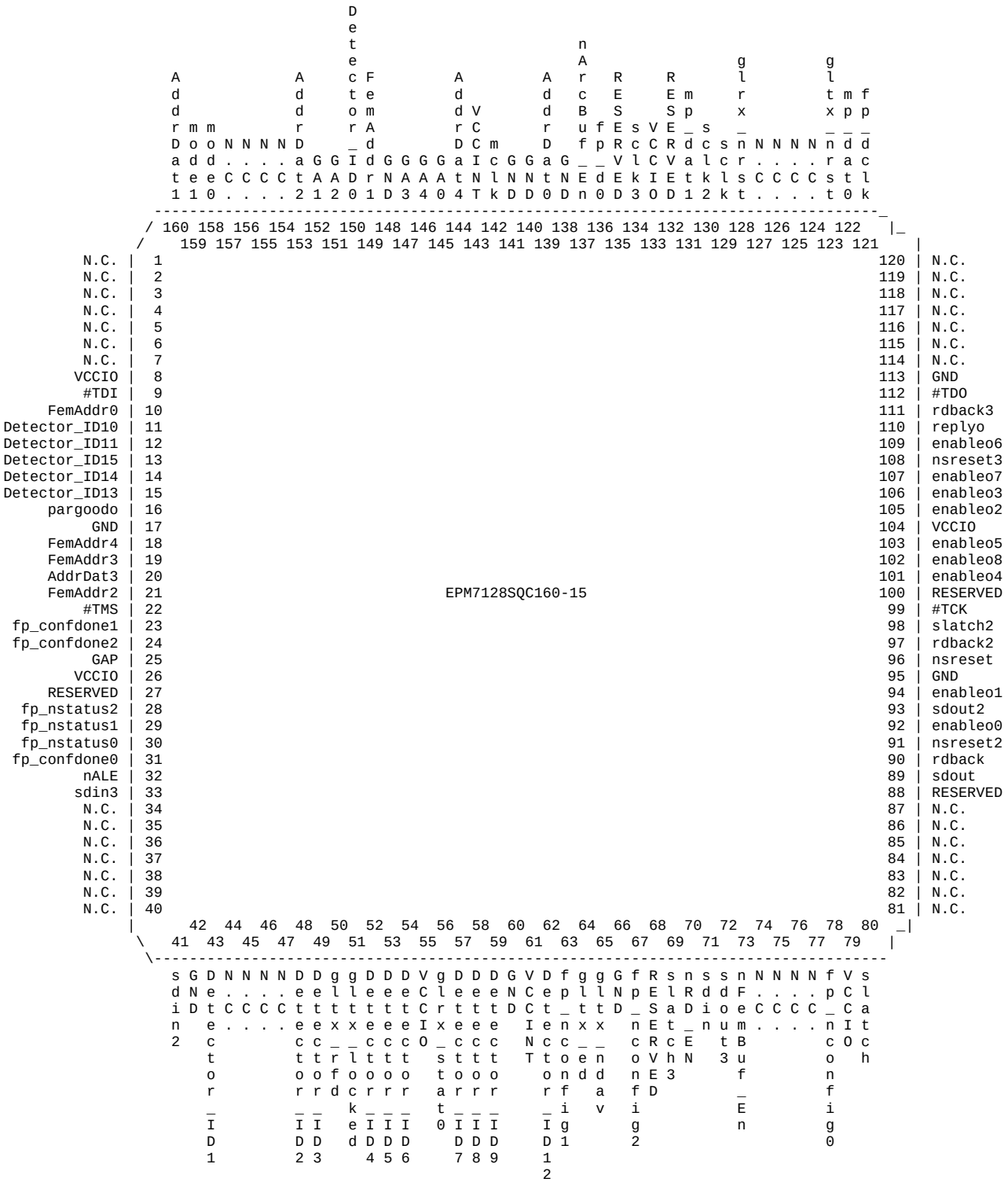
The Detector ID is a 16-bit number that is assigned to each Phenix subsystem. It will be predefined and can be hardwired on the board. On the Muon ID board, it will be set by DIP switches as the boards will also be used in Muon Tracker Anodes. This number will be set on the FEM board and will be available for Arcnet serial readout upon powerup. The data is sent out MSB first. i.e. the order is Detector_ID15 down to Detector_ID0.

The ID is stored in a shift register that must be loaded before readout can begin. The *AddrDat(0)* line is the clock, *AddrDat(1)* is the synchronous load line, *mp_dat0* is the serial data *(ID_out)* and *mp_dat1* is a handshaking line *(ID_valid)*. The Detector ID is valid if the *ID_valid* line is high. If *Synch*. *Load* is high when *AddrDat(0)* goes high, the shift register is loaded and the first data bit (the MSB) is sent out on *mp_dat0*. Each successive rising edge of *AddrDat(0)* will clock out the next bit if the Detector ID. After the last bit has been shifted out, the next clock will force *ID_valid* low. This indicates that all of the bits have been sent and that the register must be reloaded before being read again. All other bits read while *ID_valid* is low will be "0."

FEM Bus Interface

The FEM bus is a data link between the Data Formatter on the FEM and the 5-event FIFO on the ROC. The FEM must address each ROC individually to read the 16-bit data values from its 5-event FIFO. The FEM puts the ROC address on the *FemAddr[4..0]* lines and the address decoder checks this against its slot number on the falling edge of */ALE*. If the FEM address on the bus matches the geographic address, the decoder enables reading from the 5-event FIFO by driving */RD_EN* low. The buffer control line, *nFemBuf_En*, is also driven low to enable the VME backplane drivers on the ROC. Now, the data can be read from the FIFO by pulsing the */RD* line while keeping */ALE* low. The 16-bit data words are driven onto the backplane for the Data Formatter to read.

Appendix A



N.C. = Not Connected.
VCCINT = Dedicated power pin, which MUST be connected to VCC (5.0 volts).
VCCIO = Dedicated power pin, which MUST be connected to VCC (5.0 volts).
GND = Dedicated ground pin or unused dedicated input, which MUST be connected to GND.
RESERVED = Unused I/O pin, which MUST be left unconnected.

^ = Dedicated configuration pin.
+ = Reserved configuration pin, which is tri-stated during user mode.

* = Reserved configuration pin, which drives out in user mode.
PDn = Power Down pin.
@ = Special-purpose pin.
# = JTAG Boundary-Scan Testing/In-System-Programming or Configuration Pin. The JTAG inputs (TMS, TCK, TDI) should be tied to VCC when not in use.
& = JTAG pin used for I/O. When used as user I/O, JTAG pins must be kept stable before and during configuration. JTAG pin stability prevents accidental loading of JTAG instructions.

Appendix B

```
-- ******************************************************************
-- Oak Ridge National Laboratory
--
-- Project:      PHENIX Muon ID
-- Designer:  Ganesh Rao
-- File name: muid.vhd
-- Description:
-- Package for the Muon Identifier VHDL designs.
-- The address decoder is the design file using this package.
-- The package has constant declarations and function definitions
-- used in the Muon ID Address Decoder design. These constants can
-- be changed and the design recompiled for different configurations.
-- ******************************************************************

library ieee;
use ieee.std_logic_1164.all;

PACKAGE muid IS
   function parity(a: std_logic_vector) return std_logic;
   CONSTANT BCAST : std_logic_vector(4 downto 0) := "11101";
   COMPONENT arcnet is
      PORT(
         mclk             : IN   STD_LOGIC;
         AdrDat, BPadr    : IN   STD_LOGIC_VECTOR(4 downto 0);
         par_ok           : IN   STD_LOGIC;
         mode             : IN   STD_LOGIC_VECTOR(1 downto 0);
         en               : OUT  STD_LOGIC_VECTOR(8 downto 0);
         replyen             : OUT  STD_LOGIC);
      END COMPONENT;
END muid;

PACKAGE BODY muid IS

-- Generate ODD parity for 5-bit input vector
   function parity(a: std_logic_vector(4 downto 0)) return std_logic IS
      variable par: std_logic;
   begin
      par := not( a(0) xor a(1) xor a(2) xor a(3) xor a(4) );
      return par;
   end parity;
end muid;
```

```
-- ****************************************************************
-- Oak Ridge National Laboratory
--
-- Project:      PHENIX Muon ID
-- Designer:  Ganesh Rao
-- File name: arcnet.vhd
-- Description:
-- State machine for the ARCnet portion of the address decoder.
-- The mode inputs specify the mode of operation of the state machine.
-- mclk is the clock. AdrDat is the address/data bus and BPadr is the
-- backplane address translated from the geographic address of the crate.
-- par_ok is true (or '1') if the board address and the address parity
-- bit match and this should be checked at a higher level to allow
-- sharing of the parity checking function.
-- The AdrDat bus uses two transfers for the address. The first
-- (Alow) is the geographic address that must match BPadr or BCAST.
-- If it matches BCAST, ROC drivers should be disabled. The second
-- transfer or mode sends Ahi. Ahi is the function on the ROC that
-- should be enabled. Nine (9) are available.
-- ****************************************************************

library ieee;
use ieee.std_logic_1164.all;
library work;
use work.muid.all;

ENTITY arcnet IS
   PORT(
      mclk         : IN   STD_LOGIC;
      AdrDat, BPadr : IN   STD_LOGIC_VECTOR(4 downto 0);
      par_ok       : IN   STD_LOGIC;
      mode         : IN   STD_LOGIC_VECTOR(1 downto 0);
      en           : OUT  STD_LOGIC_VECTOR(8 downto 0);
      replyen          : OUT  STD_LOGIC);
END arcnet;

-- Modes:  00 = Load Board Address
--         01 = Load Board Function/destination
--         10 = Enable the selected outputs
--         11 = Disable functions

ARCHITECTURE a OF arcnet IS
   SIGNAL Alow          : STD_LOGIC_VECTOR(4 downto 0);
   SIGNAL Ahi       : STD_LOGIC_VECTOR(4 downto 0);

BEGIN
   PROCESS (mclk)
      VARIABLE adrvalid   : STD_LOGIC;
   BEGIN
      IF mclk'EVENT AND mclk = '1' THEN
         CASE mode IS
-- Load low data byte or board address
            WHEN "00" => Alow <= AdrDat;

-- Load high data byte or board function
            WHEN "01" => Ahi <= AdrDat;

-- Enable the appropriate lines, if any
-- If entered address = Backplane address, then enable reply lines.
-- If entered address = Broadcast address, disable reply lines.
-- If entered address is neither, it is not this board, disable lines.
-- Then check the board function lines and connect the data lines to
-- the appropriate output lines based on the addresss.
            WHEN "10" =>
               IF (Alow = BPadr AND par_ok = '1') THEN
                  IF ((Ahi > "00000") AND (Ahi <= "01001")) THEN replyen <= '1';
                  END IF;
                  adrvalid := '1';
               ELSIF (Alow = BCAST AND par_ok = '1') THEN
                  replyen <= '0';
                  adrvalid := '1';
               ELSE
                  replyen <= '0';
                  adrvalid := '0';
               END IF;

               IF (adrvalid = '1' and Ahi = "00001") THEN
                  en <= (0 => '1', others => '0');
               ELSIF (adrvalid = '1' and Ahi = "00010") THEN
                  en <= (1 => '1', others => '0');
               ELSIF (adrvalid = '1' and Ahi = "00011") THEN
                  en <= (2 => '1', others => '0');
               ELSIF (adrvalid = '1' and Ahi = "00100") THEN
                  en <= (3 => '1', others => '0');
               ELSIF (adrvalid = '1' and Ahi = "00101") THEN
                  en <= (4 => '1', others => '0');
               ELSIF (adrvalid = '1' and Ahi = "00110") THEN
```

```
                    en <= (5 => '1', others => '0');
                ELSIF (adrvalid = '1' and Ahi = "00111") THEN
                    en <= (6 => '1', others => '0');
                ELSIF (adrvalid = '1' and Ahi = "01000") THEN
                    en <= (7 => '1', others => '0');
                ELSIF (adrvalid = '1' and Ahi = "01001") THEN
                    en <= (8 => '1', others => '0');

-- In all other cases address invalid. Disable all.
                ELSE
                    en <= (others => '0');
                END IF;

-- In other modes, disable all.
             WHEN others =>
                    replyen <= '0';
                    en <= (others => '0');
          END CASE;
       END IF;
   END PROCESS;
END a;
```

```
-- ****************************************************************
-- Oak Ridge National Laboratory
--
-- Project:      PHENIX Muon ID
-- Designer:  Ganesh Rao
-- File name: addrdec.vhd
-- Description:
-- State machine for the  Address Decoder that connects the
-- data lines from the Arcnet processor to the appropriate
-- output lines. The outputs can be to program FPGAs, reset
-- the Glinks, or send a PHENIX serial string.
-- ****************************************************************

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
library work;
use work.muid.all;
library altera;
use altera.maxplus2.all;
LIBRARY lpm;
USE lpm.lpm_components.ALL;


ENTITY addrdec IS
   PORT (
      -- Diagnostic outputs. They can be removed later
      enableo                    : OUT  STD_LOGIC_VECTOR(8 downto 0);  -- Which fcn. enabled?
      replyo                  : OUT  STD_LOGIC;  -- Will board reply?
      pargoodo                : OUT  STD_LOGIC;  -- Is parity OK?

       -- Geographical Address and Parity (inverted and odd parity)
      GA                      : IN   STD_LOGIC_VECTOR(4 downto 0);  -- Inverted address
      GAP                     : IN   STD_LOGIC;  -- Inverted parity

      -- The Detector ID from DIP switches
      Detector_ID                : IN   STD_LOGIC_VECTOR(15 downto 0);

       -- Arcnet bus and associated serially controlled outputs
      mclk                    : IN   STD_LOGIC;  -- Mode Clock
      AddrDat                    : IN   STD_LOGIC_VECTOR(4 downto 0);  -- Arcnet Bus
      mode                    : IN   STD_LOGIC_VECTOR(1 downto 0);  -- Arcnet Modes
      mp_dat0, mp_dat1        : OUT  STD_LOGIC;   -- Return values
      gltx_nrst               : OUT  STD_LOGIC;   -- Reset Glink Tx active low
      gltx_ed                    : OUT  STD_LOGIC;   -- Enable Data to transmitter
      gltx_rfd                : IN   STD_LOGIC;   -- RFD from transmitter
      gltx_ndav               : OUT  STD_LOGIC;   -- /DAV to transmitter
      gltx_locked                : IN   STD_LOGIC;   -- Is Txmit locked?
      glrx_nrst               : OUT  STD_LOGIC;   -- Reset Glink Rx active low
      glrx_stat0              : IN   STD_LOGIC;   -- Is Rcvr locked?
      sdin,  sdin2,  sdin3    : IN   STD_LOGIC; -- Data from board to backplane bus
      sclk,  sclk2,  sclk3    : OUT  STD_LOGIC;   -- Serial clock to board
      sdout, sdout2,    sdout3      : OUT  STD_LOGIC;   -- Serial data to board
      slatch,   slatch2, slatch3 : OUT  STD_LOGIC;   -- Serial latch to board
      rdback,   rdback2, rdback3 : OUT  STD_LOGIC;   -- Serial readback enable to board
      nsreset,nsreset2,nsreset3  : OUT  STD_LOGIC;   -- Serial resests active low
      fp_dclk                    : OUT  STD_LOGIC;   -- Shared FPGA prog. pin (DCLK)
      fp_d0                   : OUT  STD_LOGIC;   -- Shared FPGA prog. pin (DATA0)
      fp_nstatus              : IN   STD_LOGIC_VECTOR(2 DOWNTO 0);  -- FPGA prog. pins
      fp_confdone                : IN   STD_LOGIC_VECTOR(2 DOWNTO 0);  -- FPGA prog. pins
      fp_nconfig              : OUT  STD_LOGIC_VECTOR(2 DOWNTO 0);  -- FPGA prog. pins

       -- FEM controlled bus lines
      FemAddr                    : IN   STD_LOGIC_VECTOR(4 DOWNTO 0);   -- FEM bus
      nALE                    : IN   STD_LOGIC;   -- Addr. Latch Enable
      nRD_EN                  : OUT  STD_LOGIC;   -- Read from 5-event FIFO

      -- Control the VME backplane drivers
      nFemBuf_En              : OUT  STD_LOGIC;-- Output Enable (active low) for FEM/ROC bus
      nArcBuf_En              : OUT  STD_LOGIC -- Output Enable (active low) for Arcnet bus

 );

END addrdec;

ARCHITECTURE a OF addrdec is
   SIGNAL enable           : STD_LOGIC_VECTOR(8 downto 0);  -- Internal enable lines
   SIGNAL reply            : STD_LOGIC;  -- Arcnet reply enable status
   SIGNAL en_reply         : STD_LOGIC_VECTOR(8 downto 0);  -- Arcnet reply per line
   SIGNAL    par              : STD_LOGIC;  -- Calculated parity of the board address
   SIGNAL  pargood            : STD_LOGIC;  -- Does calculated match backplane parity?
   SIGNAL boardnum         : STD_LOGIC_VECTOR(4 downto 0);  -- Geographic board address
   SIGNAL  boardpar        : STD_LOGIC;  -- Backplane address parity
   SIGNAL fpgaprog         : STD_LOGIC;  -- FPGA program mode?
   SIGNAL gltx_edlatch     : STD_LOGIC;  -- Latch ED line?
   SIGNAL gltx_ed_int         : STD_LOGIC;  -- Internal ED line.
   SIGNAL vcc, gnd         : STD_LOGIC;  -- Power and Ground for logic levels
```

```
   SIGNAL nFEM_reply       : STD_LOGIC;  -- FEM reply enable status (Active LOW)
   SIGNAL ID_out           : STD_LOGIC;  -- Serial data line for reading Detector ID
   SIGNAL ID_valid         : STD_LOGIC;  -- Is detector ID valid?
   SIGNAL loclk            : STD_LOGIC;  -- Allows AddrDat(0) to be used as a clock
   SIGNAL ID_cnt           : INTEGER RANGE 0 TO 15; -- Counts bits sent out
   SIGNAL ArcBuf_En        : STD_LOGIC;  -- Non-inverted internal signal
   TYPE   STATE_TYPE IS (CNT_STATE, NULL_STATE);  -- States for Detector ID readback
   SIGNAL    state            : STATE_TYPE;  -- State variable

BEGIN
-- Set diagnostic outputs. Can be commented out along with PORT definitions.
   enableo <= enable;
   replyo <= reply;
   pargoodo <= pargood;

   vcc <= '1';
   gnd <= '0';
   boardnum <= not(GA);        -- Board number from Geographic address
   boardpar <= not(GAP);          -- Geographic address parity
   par <= parity(boardnum);       -- Geographic address parity check
   pargood <=vcc WHEN par = boardpar ELSE
             gnd;

-- Check ARCnet bus against address map and enable appropriate lines.
   arcnetdec: arcnet PORT MAP (mclk, AddrDat, boardnum, pargood, mode, enable, reply);

-- Form the vector of reply enables from reply and the enable lines.
   PROCESS
   BEGIN
      FOR i IN en_reply'RANGE LOOP
         en_reply(i) <= reply AND enable(i);
      END LOOP;
   END PROCESS;

-- Now enable output on the proper lines. All lines are Active HIGH
-- unless otherwise specified.

-- ENABLE(0) = '1' enables serial string I/O
   sclk    <= enable(0) AND AddrDat(0);
   slatch  <= enable(0) AND AddrDat(1);
   sdout   <= enable(0) AND AddrDat(2);
   rdback  <= enable(0) AND AddrDat(3);
   nsreset    <= not( enable(0) AND not(AddrDat(4)) );  -- Active LOW

-- ENABLE(1) = '1' enables serial string I/O on alternate pins
   sclk2   <= enable(1) AND AddrDat(0);
   slatch2    <= enable(1) AND AddrDat(1);
   sdout2  <= enable(1) AND AddrDat(2);
   rdback2    <= enable(1) AND AddrDat(3);
   nsreset2 <= not( enable(1) AND not(AddrDat(4)) );  -- Active LOW

-- ENABLE(2) = '1' enables serial string I/O on second alternate pins
   sclk3   <= enable(2) AND AddrDat(0);
   slatch3    <= enable(2) AND AddrDat(1);
   sdout3  <= enable(2) AND AddrDat(2);
   rdback3    <= enable(2) AND AddrDat(3);
   nsreset3 <= not( enable(2) AND not(AddrDat(4)) );  -- Active LOW

-- ENABLE(3) or ENABLE(4) or ENABLE(5) = '1'
-- enables shared FPGA programming pins DCLK and D0
-- FPGA programming pins are high impedance when not addressed.
-- Pulled high externally to allow BitBlaster use.
   fpgaprog <= enable(3) OR enable(4) OR enable(5);
   WITH fpgaprog SELECT
   fp_dclk   <=  fpgaprog AND AddrDat(0) when '1',
             'Z'                  when others;

   WITH fpgaprog SELECT
   fp_d0  <=  fpgaprog AND AddrDat(1) when '1',
             'Z'                  when others;

-- ENABLE(3) = '1' enables FPGA #1 programming pins
   WITH enable(3) SELECT
   fp_nconfig(0) <= not( enable(3) AND not(AddrDat(2)) )  when '1',  -- Active LOW
                  'Z'                              when others;

-- ENABLE(4) = '1' enables FPGA #2 programming pins
   WITH enable(4) SELECT
   fp_nconfig(1) <= not( enable(4) AND not(AddrDat(2)) )  when '1',  -- Active LOW
                  'Z'                              when others;

-- ENABLE(5) = '1' enables FPGA #3 programming pins
   WITH enable(5) SELECT
   fp_nconfig(2) <= not( enable(5) AND not(AddrDat(2)) )  when '1',  -- Active LOW
                  'Z'                              when others;

-- ENABLE(6) = '1' enables the GLINK transmitter pins.
```

```
-- AddrDat(1) is ED Latch and AddrDat(2) is ED.
-- When EDLATCH is high, the latch is transparent. When low,
-- the previous value is preserved at the output.
   gltx_nrst <= not( enable(6) AND not(AddrDat(0)) );     -- Active LOW
   gltx_edlatch <= enable(6) AND AddrDat(1);
   latchgltx3: latch PORT MAP (AddrDat(2), gltx_edlatch, gltx_ed_int);
   gltx_ed <= gltx_ed_int;  -- Internal line necessary for other logic

-- ENABLE(7) = '1' enables the GLINK receiver pins.
   glrx_nrst <= not( enable(7) AND not(AddrDat(0)) );     -- Active LOW


-- ENABLE(8) = '1' enables Arcnet to read the Detector ID serially from
-- a shift register. The Detector ID is a 16-bit constant that is arcnet
-- readable. The data comes out on mp_dat0 on rising edges of AddrDat(0).
-- Data is sent out MSB first (i.e. Detector ID 15 to Detector ID 0).
-- AddrDat(1) is the LOAD line and it synchronously reloads the register
-- with the Detector ID. This *MUST* be done before reading. The function
-- must be enabled for anything to happen.
   id_shiftreg: lpm_shiftreg
      GENERIC MAP ( LPM_WIDTH => 16 )
      PORT MAP (data => Detector_ID, clock => AddrDat(0), enable => enable(8),
                shiftin => gnd,  load => AddrDat(1),
                shiftout => ID_out);

   loclk <= AddrDat(0);
   PROCESS (loclk)
   BEGIN
      IF loclk'EVENT AND loclk = '1' THEN
         CASE state IS
            WHEN NULL_STATE =>
               IF enable(8) = '1' AND AddrDat(1) = '1' THEN
                  state <= CNT_STATE;
                  ID_cnt <= 0;
               END IF;

            WHEN CNT_STATE =>
               IF enable(8) = '1' AND AddrDat(1) = '1' THEN
                  ID_cnt <= 0;
               ELSIF ID_cnt < 15 AND enable(8) = '1' THEN
                  ID_cnt <= ID_cnt + 1;
               ELSE
                  state <= NULL_STATE;
               END IF;
         END CASE;
      END IF;
   END PROCESS;

   WITH state SELECT
      ID_valid      <=  '0' WHEN   NULL_STATE,
                        '1' WHEN   CNT_STATE;


-- Enable the appropriate lines back to the ARCnet microprocessor.
   WITH en_reply SELECT
   mp_dat0 <= sdin          when "000000001",
              sdin2         when "000000010",
              sdin3         when "000000100",
              fp_nstatus(0)    when "000001000",
              fp_nstatus(1)    when "000010000",
              fp_nstatus(2)    when "000100000",
              gltx_locked   when "001000000",
              glrx_stat0    when "010000000",
              ID_out        when "100000000",
              '0'           when others;

   WITH en_reply SELECT
   mp_dat1 <= fp_confdone(0)   when "000001000",
              fp_confdone(1)   when "000010000",
              fp_confdone(2)   when "000100000",
              ID_valid      when "100000000",
              '0'           when others;


-- This section is the interface to the 5-event FIFO FPGA. It checks the
-- Fem Address bus to see if this board is being addressed and if so, it
-- asserts (active low) the nRD_EN line which enables reads of the 5-event
-- FIFO in the FPGA. nALE must be low for the full duration of the read
-- cycle or nRD_EN will go high.
PROCESS (nALE)
BEGIN
   IF not(FemAddr = boardnum AND pargood = '1') THEN
      nFEM_reply <= '1';
   ELSIF nALE'EVENT AND nALE = '0' THEN
         nFEM_reply <= '0';   -- Ready to read data
   END IF;
   nRD_EN <= nFEM_reply OR nALE;
```

```
END PROCESS;


-- This section is the interface to the TTL Glink Transmitter.
gltx_ndav <= not(gltx_rfd AND gltx_ed_int);   -- Active LOW

-- These lines control the VME backplane drivers. They enable the
-- drivers (turns them on) when the mode says that the FEM is
-- expecting data from the 5-event FIFO or when Arcnet needs data.
-- The lcell was required to prevent NOT gate push-back that caused
-- the output to have an initial output state of '0', which is very bad.
nFemBuf_En <= nFEM_reply OR nALE;     -- Active LOW
keepcell: lcell PORT MAP (reply, ArcBuf_En);
nArcBuf_En <= NOT ArcBuf_En;          -- Active LOW

END a;
```