An Electronic Interface to High Speed Pattern Recognition System

Andrea Morris
Madonna Catholic Secondary School

The XFT Track Processor

The XFT track processer searches for track patterns

- Energetic particles are created in matter-antimatter collisions
- 48 measurements are made of their position
- XFT is a high-speed track processor
  - Searches for patterns of “hits”
  - Operates at 33 MHz
  - Composed to two special circuits
    1. FINDER
    2. LINKER

FINDER prototype being developed at U of T

- Requires test circuit to act as interface between FINDER and computer

Goal of project is to design interface circuit

The FINDER ASIC

The FINDER ASIC finds track segment patterns at very high speed

- A “real” track must have 4 segments
- FINDER searches for segments 10 million times per second
- Produces a list of track segments for LINKER
  - LINKER finds the 4 segments that match

Each FINDER

- Requires 35 input signals (for data and control information)
- Produces 50 output signals to feed data to the LINKER chips
- Contains about 50,000 logic gates

FINDER Interface must:

- Supply the input signals
- Record the output signals
- Control the chip under computer supervision

Design of FINDER Interface

Steps to design Interface using VERILOG:

1. Learned how to design in VERILOG using simple examples
2. Created a data flow diagram for interface
3. Divided interface into separate modules
   - Identified functionality of each module
   - Organised input and output signals
4. Began coding each module in VERILOG

Functionality of FINDER Interface:

- Have to set the control parameters
- Bring information to and from VME crate backplane
- Send commands to FINDER chip, eg.
  - Start and Stop
  - Read out memory on chip

Current status of interface design:

- Clock generator works!
- Buffer, Control, Readout and FIFO modules coded
  - Need to be tested
- Testbench programme to test interface needs to be designed
- Start and Stop commands have to be implemented

Outlook for Project

- VERILOG programming will be shortly completed
- Then can proceed with testbench programming and testing
- Will then make all changes required to ensure that interface works correctly

```
/* The FIFO module takes in info and outputs it in the same order
   (First In First Out) so that Vme has access to the patterns found.
   Vme can input and output to this module. All of the info is
   received from Finder.   */

module    FIFO(vme_data_bus[31:0],as,ds,read,vme_address[1:0],
               pixel[24:0], clk_33,inhibitor[1:0],fifo_enable);

output   [31:0] vme_data_bus;

input           as,ds,read;
input   [1:0]   vme_address;
input   [24:0]  pixel;
input           clk_33;
input   [1:0]   inhibitor;
input           fifo_enable;

reg     [24:0]  fifo_info;
wire    [4:0]   local_strobe;
reg     [31:0]  vme_data_bus;

assign
local_strobe  = as && ds && read  && fifo_enable;

always @ (negedge clk_33)
        begin
          if (!inhibitor[1])
           assign
             fifo_info = pixel[24:0];
        end

always @ (local_strobe)
        vme_data_bus[31:0] = inhibitor[1]?{{7{fifo_info[24]}},
                                        {fifo_info[24:0]}}:32'bz;

endmodule
```