Week 3 Status Update

CSE 141L
Oct. 17, 2008

Announcements

• No office hours today (go to Y! Hack Week stuff)
• Lab 2, Part B is now available
  – Due in 2 weeks (Oct. 31… scaaary)
  – Fetch unit control and testing
• Need to pick partners
  – E-mail me (members and team name) by next Friday or…
• Lab 1 grades are coming soon…
  – Most people did very well
Why ‘pc_prev_r’?

Question 5 – Cycle 0
Question 5 – Cycle 1

Question 5 – Cycle 2
Question 5 – Cycle 3

Datapath

Question 5 – Cycle 4

Datapath
Verilog Refactoring – adder.v

module adder#(parameter WIDTH=32)
  
  input  [WIDTH-1:0] din0,
  input  [WIDTH-1:0] din1,
  output [WIDTH-1:0] dout
;
  reg  [WIDTH-1:0] outReg;
  wire  dout_w = outReg;
  assign dout = dout_w;

  always @( * )
  begin
    outReg = din0 + din1;
  end

endmodule

• Simplify, if possible
• Ports are already wires
Verilog Refactoring – mux.v

```
module my_mux#(parameter WIDTH=10)
(
    input    sel,
    input    [WIDTH-1:0] din0,
    input    [WIDTH-1:0] din1,
    output   [WIDTH-1:0] dout
);
reg    [WIDTH-1:0] regin0, regin1;
reg    [WIDTH-1:0] regout;
assign dout = regout;
always@(*)
    begin
        case(sel)
            1'd0 : regout <= regin0;
            1'd1 : regout <= regin1;
        endcase
    end
always@(*)
    begin
        regin1<= din1;
        regin0<= din0;
    end
endmodule
```

```
module my_mux#(parameter WIDTH=10)
(
    input    sel,
    input    [WIDTH-1:0] din0,
    input    [WIDTH-1:0] din1,
    output   [WIDTH-1:0] dout
);
assign dout = sel? din1 : din0;
endmodule
```

• Avoid ‘<=’ in combinational logic

Before we move on…

• Any other questions about part A stuff?
Lab 2, Part 2 Overview

• 4 Steps
  – Implement Control for Fetch Unit
  – Use given test benches
  – Create your own test bench
  – Performance evaluation
• Keep your Verilog code as clean as possible

Step 1: Control Unit

• Webpage fully specifies behavior of control unit
• Three types of signals:
  – Input (e.g. restart, load_store_valid)
  – Internal (e.g. sel_mux, fifo_clear)
  – Output (instr_valid, load_data_valid)
• Subtle point:
  – Restart should clear FIFO on the next cycle, regardless of pre-emption… Why?
Testing your Fetch Unit

• We give you two testbenches
  – Simple restart/dequeue/load example
  – Branching/looping program
• You will need to create one testbench on your own
  – Test simultaneous restart and load/store
• Can use the GUI TB editor or do it manually

Manual Testbench Setup

`timescale 1ns/1ns // Add this to the top of your file to set time scale
module testbench();
  reg [3:0] A, B;
  reg C0;
  wire [3:0] S;
  wire C4;
  4bitFA uut (.B(B), .A(A), .cin(C0), .S(S), .cout(C4)); // instantiate adder

initial // initial blocks run only at the beginning of simulation (only use in testbenches)
begin
  $monitor($time,"A=%b,B=%b, c_in=%b, c_out=%b, sum = %b\n",A,B,C0,C4,S);
end
initial
begin
  A = 4'd0; B = 4'd0; C0 = 1'b0;
  // wait 50 ns before next assignment
  #50 A = 4'd3; B = 4'd4;
  #50 A = 4'b0001; B = 4'b0010; // don’t use \#n outside of testbenches
end
endmodule
Go Get ‘Em!