Outline

- Review: clocked logic.
- 150 no longer a prerequisite! Quiz cancelled! New mini-labs!
- Review: clocked logic.

152 Project: Deeply Pipelined Processor

the big ideas

Datapaths
Control
Pipelining
Caching

Review: A Green Card View of MIPS

<table>
<thead>
<tr>
<th>NAME</th>
<th>MNE-MON-IC</th>
<th>FOR-MAT</th>
<th>OPERATION (in Verilog)</th>
<th>OPCODE/FUNCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>add</td>
<td>R</td>
<td>R[rd] = R[rs] + R[rt] (1)</td>
<td>0 / 20hex</td>
</tr>
<tr>
<td>Add Immediate</td>
<td>addi</td>
<td>I</td>
<td>R[rt] = R[rs] + SignExtimm (1)</td>
<td>8hex</td>
</tr>
<tr>
<td>Branch On Equal</td>
<td>beq</td>
<td>I</td>
<td>if(R[rs] == R[rt]) PC = PC+4+ BranchAddr (4)</td>
<td>4hex</td>
</tr>
</tbody>
</table>

152 Project: Deeply Pipelined Processor

Intel XScale ARM Pipeline, JSSC 36:11, November 2001

Schematic Diagrams
Timing Diagrams
Verilog
**Administrivia - Course Adjustments**

- 150 is no longer a prerequisite.
- Goal: In first month, cover the essential material for the Labs.
- Prerequisite quiz is cancelled!

**Administrivia -- Mini-Labs**

30 minute demo of CAD tools by a TA, followed by 90 minutes hands-on work.

In 119 Cory, in discussion section slots: Friday 12-2 and Friday 2-4.

- 9/3: Design Entry, Simulation
- 9/10: State Machine Debugging
- 9/17: Pushing designs to Xilinx

**Administrivia -- mini-Lab FAQ**

- Is there a pre-lab? Yes, a short list of questions, due at the start of lab.
- When is it due? End of lab, extensions by request.
- Is the first mini-lab on the website now? Yes!

**Administrivia -- other course changes**

- Lab 1 due date is still 9/13. Get started early!
- An extra week added to Lab 2 (workload is unchanged).
- After Lab 2 is completed, the course will be back on plan.

**Clocked Logic Review**
Review: Edge-Triggered D Flip Flops

- Value of D is sampled on positive clock edge.
- Q outputs sampled value for rest of cycle.

module ff(D, Q, CLK);
  input D, CLK;
  output Q;
  reg Q;
  always @ (posedge CLK)
    Q <= D;
endmodule

Logic styles used in CS 152 lectures

- All state elements in a design are edge-triggered, on the positive edge of a single global clock.
- or
- All state elements in a design are edge-triggered, on the negative edge of a single global clock.

State Machine Review

Specification: Traffic Light Controller

<table>
<thead>
<tr>
<th>CLK</th>
<th>Change</th>
<th>Rst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If Change == 1 on positive CLK edge traffic light changes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If Rst == 1 on positive CLK edge</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td>1 0 0</td>
</tr>
</tbody>
</table>
State Diagram: Traffic Light Controller

State Assignment: Traffic Light Controller

Next State Logic: Traffic Light Controller

State Verilog: Traffic Light Controller

Verilog: Edge-Triggered D Flip Flops

wire next_R, next_Y, next_G;
output R, Y, G;

Value of D is sampled on positive clock edge. Q outputs sampled value for rest of cycle.

module ff(D, Q, CLK);
input D, CLK;
output Q;
reg Q;

always @ (posedge CLK)
Q <= D;
endmodule
State Elements: Traffic Light Controller

wire  next_R, next_Y, next_G;
output  R, Y, G;

ff ff_R(R, next_R, CLK);
ff ff_Y(Y, next_Y, CLK);
ff ff_G(G, next_G, CLK);

Next State Logic: Traffic Light Controller

wire  next_R, next_Y, next_G;

assign  next_R =  rst ? 1'b1 : (change ? Y : R);
assign  next_Y =  rst ? 1'b0 : (change ? G : Y);
assign  next_G =  rst ? 1'b0 : (change ? R : G);

Verilog: Complete Traffic Light Controller

wire  next_R, next_Y, next_G;
output  R, Y, G;

assign  next_R =  rst ? 1'b1 : (change ? Y : R);
assign  next_Y =  rst ? 1'b0 : (change ? G : Y);
assign  next_G =  rst ? 1'b0 : (change ? R : G);

Logic Diagram: Traffic Light Controller

In conclusion -- Design Descriptions

- Schematics: visually coherent logic structure.
- Timing diagrams: logic in motion.
- Verilog: Precise semantics and structure.