CS152 – Computer Architecture and Engineering

Lecture 2 – Clocked Logic Review

2004-09-02

Dave Patterson
(www.cs.berkeley.edu/~patterson)

John Lazzaro
(www.cs.berkeley.edu/~lazzaro)

www-inst.eecs.berkeley.edu/~cs152/
### Review: A Green Card View of MIPS

<table>
<thead>
<tr>
<th>NAME</th>
<th>MNE-MON-IC</th>
<th>FORMAT</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>0/20_{\text{hex}}</td>
</tr>
<tr>
<td>Add Immediate</td>
<td>addi</td>
<td>I</td>
<td>(R[rt] = R[rs] + \text{SignExtImm}) (1)(2)</td>
<td>8_{\text{hex}}</td>
</tr>
<tr>
<td>Branch On Equal</td>
<td>beq</td>
<td>I</td>
<td>if((R[rs] == R[rt])) (\text{PC=PC+4+BranchAddr}) (4)</td>
<td>4_{\text{hex}}</td>
</tr>
</tbody>
</table>
Outline

- Review: clocked logic.
- 150 no longer a prerequisite! Quiz cancelled! New mini-labs!
- Review: clocked logic.
152 Project: Deeply Pipelined Processor

Intel XScale ARM Pipeline,
JSSC 36:11, November 2001
152 Project: Deeply Pipelined Processor

the big ideas

Datapaths
Control
Pipelining
Caching

Intel XScale ARM Pipeline,
JSSC 36:11, November 2001
152 Project: Deeply Pipelined Processor

Schematic Diagrams

Timing Diagrams

Verilog

Intel XScale ARM Pipeline,
JSSC 36:11, November 2001
152 Project: Deeply Pipelined Processor

Test Plan

Xilinx

CAD flows

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

CLK

D

Q
Review: Edge-Triggering in Verilog

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;
    always @ (CLK)
        Q <= D;
endmodule
```
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

Correct?
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

- **If Change == 1 on positive CLK edge traffic light changes**
  - R (red)
  - Y (yellow)
  - G (green)

- **If Rst == 1 on positive CLK edge**
  - R Y G = 1 0 0
State Diagram: Traffic Light Controller

Rst == 1

R Y G 1 0 0

Change == 1

R Y G 0 0 1

Change == 1

R Y G 0 1 0

R Y G 1 0 0

R Y G 0 0 1

R Y G 0 1 0
Timing Diagram: Traffic Light Controller

Rst == 1

Change == 1

R Y G 1 0 0 → R Y G 0 0 1 → R Y G 0 1 0

CLK

Change

R Y G 1 0 0 → 0 0 1 → 0 1 0 → 1 0 0
State Assignment: Traffic Light Controller

"One-Hot Encoding"
Next State Logic: Traffic Light Controller

Rst == 1
Change == 1

R Y G 1 0 0
Change == 1
Change == 1

R Y G 0 0 1
Change == 1

R Y G 0 1 0

Next State Combinational Logic

D Q R

Change

D Q G

D Q Y
State Verilog: Traffic Light Controller

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

???
Verilog: 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
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);

ff ff_R(R, next_R, CLK);
ff ff_Y(Y, next_Y, CLK);
ff ff_G(G, next_G, CLK);
Logic Diagram: Traffic Light Controller

Rst == 1

Next State Combinational Logic

D Q R

D Q G

D Q Y

Change == 1

Change == 1

Change == 1

Change
In conclusion -- Design Descriptions

*Schematics: visually coherent logic structure.*

*Timing diagrams: logic in motion.*

*Verilog: Precise semantics and structure.*