Welcome to The Hardware Lab!

Fall 2016

Lab2: Data Flow and Behavioral Modeling

Prof. Chun-Yi Lee

Department of Computer Science
National Sting Hua University

Verilog Lab 2

- **Six basic questions** (50%)
  - Demonstration on **10/20/2016 (Thu)**, **in class**
- **Lab 2 report** (30%)
  - **Circuit** or **block diagram** of your design, **and explain how it works**
  - Answer the questions listed in the handout (if any)
  - Contribution of each team member
  - What you have learned from Lab 2
- **Two optional questions** (20%, 10% each)
  - Challenge it if you have extra time
  - Include in your Lab 2 report if you can make it
  - **Describe your design and how it works**

Verilog Lab 2 Rules

- You can use **ANY** modeling techniques except for Q4 ~ Q6
  - Gate-level modeling
  - Data flow modeling
  - Behavioral modeling (combinational)
  - Only gate-level is allowed for Q4 ~ Q6
- Demonstrate your work by **waveforms**
  - Edit your own testbench
- Please follow the **module names** and **I/O ports** of the provided templates
  - **Please keep in mind that we will deduct points if the module names and I/O port mismatch with the template**

Verilog Lab 2 Submission

- **Demonstration** and **code/report** submission due date & time:
  5:30pm, 10/20/2016 (Thu)
  This is a HARD deadline

- Please submit your report and Verilog codes to **ILMS**
  - You codes have to include your **design** and **testbench**
  - Format: **Lab2_Team1_Codes.v, Lab2_Team1_Tb.v, Lab2_Team1_Report.pdf**
  - **We will deduct points if the submitted files mismatch the requirements**
  - **Please don't uncomment the modules in the template file**

- We will test your codes by our own testbench

Question 1

- 4-bit **Comparator**
  - The 4-bits are **unsigned numbers**
  - **You can use data flow or behavioral modeling techniques**

A_lt_B	Condition
1'b1	A[3:0] < B[3:0]
1'b0	Otherwise

A_gt_B	Condition
1'b1	A[3:0] > B[3:0]
1'b0	Otherwise

A_eq_B	Condition
1'b1	A[3:0] == B[3:0]
1'b0	Otherwise

Question 2

- Binary code to Grey code
- Din[3:0] = Binary code; Dout[3:0] = Grey code

Decimal value	Binary code	Grey code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

Decimal value	Binary code	Grey code
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Question 3

- Instruction decoding and execution (**for unsigned numbers**)
  - **Inputs**: Op_Code (3 bits), Rs (4 bits), and Rt (4 bits)
  - **Output**: Rd (4 bits)

Instruction	OP_Code	Function
SUB	000	Rd = Rs - Rt (Rd = 0 if Rs - Rt is negative)
ADD	001	Rd = Rs + Rt
MUL	010	Rd = Rs * Rt
BITWISE NAND	011	Rd = Rs (bitwise nand) Rt
BITWISE NOR	100	Rd = Rs (bitwise nor) Rt
LOGICAL SHIFT LEFT	101	Rd = Rs << Rt
LOGICAL SHIFT RIGHT	110	Rd = Rs >> Rt
EQUAL	111	Rd = (Rs==Rt) ? Rs : Rt

Question 4

- Use **NAND gates only** to realize the following functions
  - **NOT, NOR, AND, OR, XOR, XNOR**
  - **Input**: A (1bit), B (1bit), Sel (3 bits)
  - **Output**: Out (1 bit)
  - **Please draw your circuits in your report (you can use a Mux)**

Sel [2:0]	Out
000	Out = !A
001	Out = A nor B
010	Out = A and B
011	Out = A or B
100	Out = A xor B
101	Out = A xnor B
110 & 111	Out = A nand B

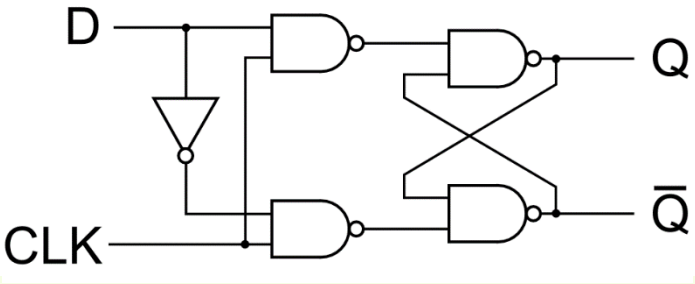
Question 5

- Use **NOR gates only** to realize the following functions
  - **NOT, NAND, AND, OR, XOR, XNOR**
  - **Input**: A (1bit), B (1bit), Sel (3 bits)
  - **Output**: Out (1 bit)
  - **Please draw your circuits in your report (you can use a Mux)**

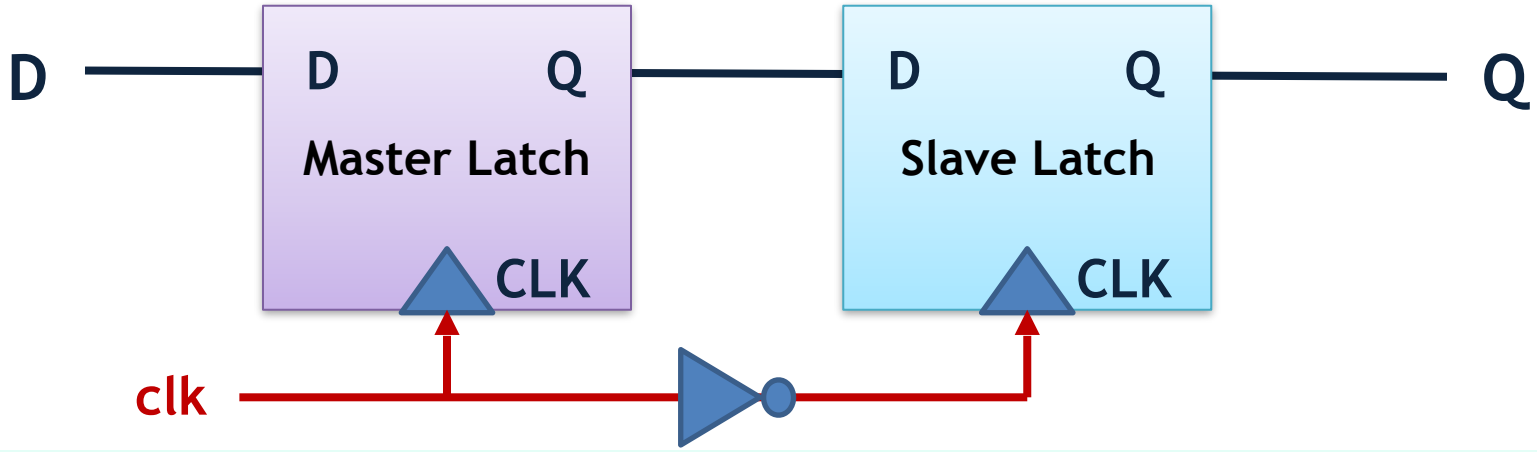
Sel [2:0]	Out
000	Out = !A
001	Out = A nand B
010	Out = A and B
011	Out = A or B
100	Out = A xor B
101	Out = A xnor B
110 & 111	Out = A nor B

Question 6

- Design a **latch** module as follows:



- Then design a **clk negative** edge trigger **flip-flop** module as:

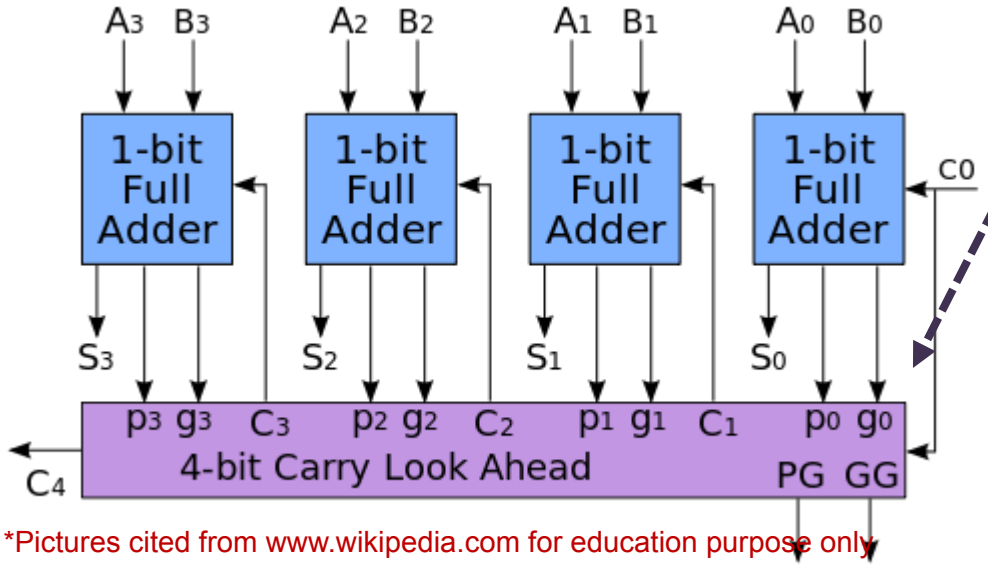


- Test your **latch** and **flip-flop** by your testbench

Optional Question 1

- Design a 4-bit Carry-Lookahead (CLA) Adder
  - Using your 1-bit Full Adder module in Lab 1 and logic gates
  - Please explain the benefits of a Carry-Lookahead Adder
  - Please explain the how it works and **the circuit of your CLA**

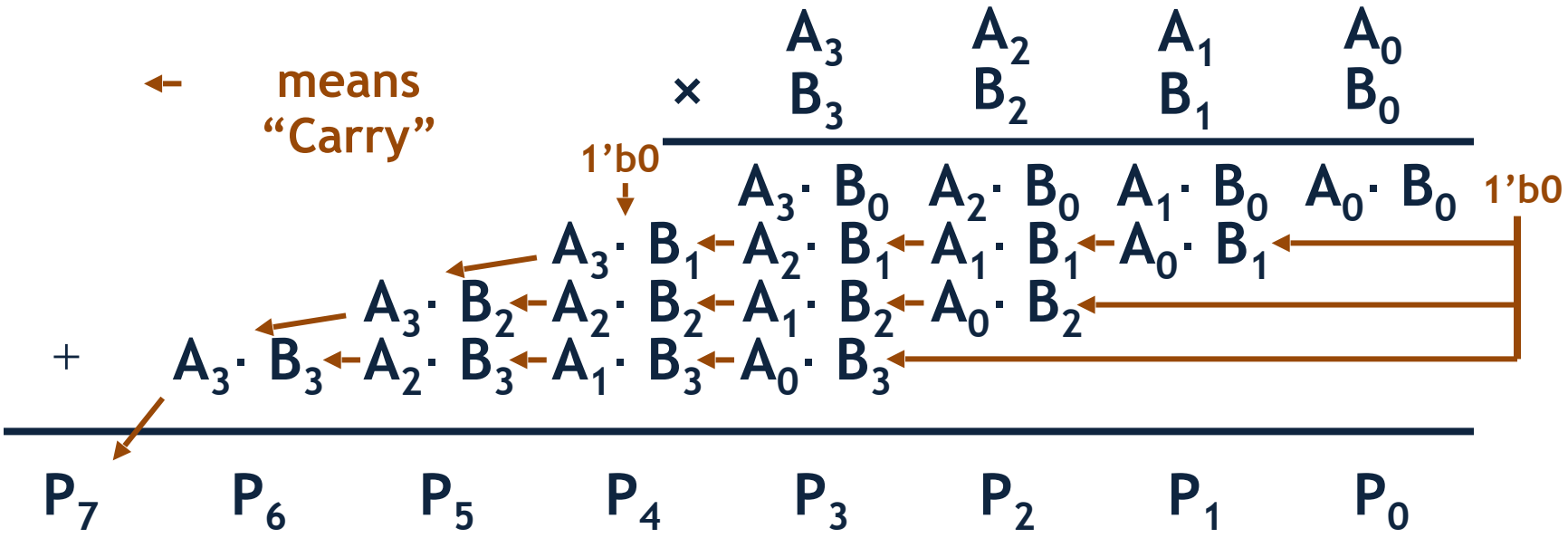
- Go to Wikipedia to check out the details of it
  - http://ppt.cc/Q3av5



*Pictures cited from www.wikipedia.com for education purpose only

Optional Question 2

- Design a 4-bit unsigned **Multiplier**
  - Using your 1-bit Full Adder module in Lab 1 and logic gates
  - Please explain the how it works
  - Please draw your block diagram using **Full Adders** and **logic gates**
- **Inputs**: A[3:0] and B[3:0]; **Output**: P[7:0]



Thank you for your attention!

*Yosemite Valley view taken at Glacier Point, Yosemite National Park, CA.
This picture is taken by Chun-Yi Lee himself, who is also a fan of photography