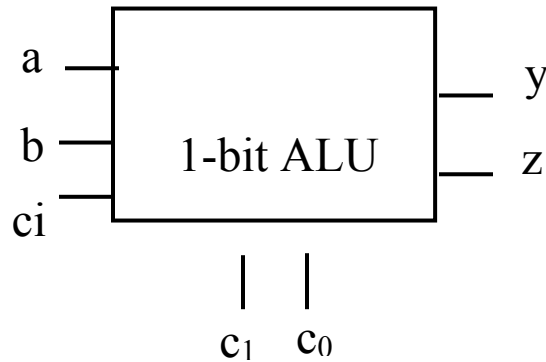


## Lab 10 (Team Project) เตรียมพร้อมการสอบ Verilog

จงออกแบบวงจรและเขียน Verilog เพื่อตรวจสอบการทำงาน

ชื่อ-นามสกุล..... รหัสนักศึกษา.....  
 ชื่อ-นามสกุล..... รหัสนักศึกษา.....  
 ชื่อ-นามสกุล..... รหัสนักศึกษา.....  
 ชื่อ-นามสกุล..... รหัสนักศึกษา.....  
 ชื่อ-นามสกุล..... รหัสนักศึกษา.....

1) ออกแบบ 1-bit ALU ( Arithmetic Logic Unit) ที่รับอินพุต a และ b และเลือกปฏิบัติการตามสัญญาณควบคุม  $c_1$  และ  $c_0$

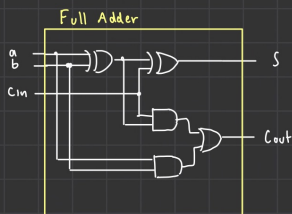


โดยมีตารางการปฏิบัติการดังนี้

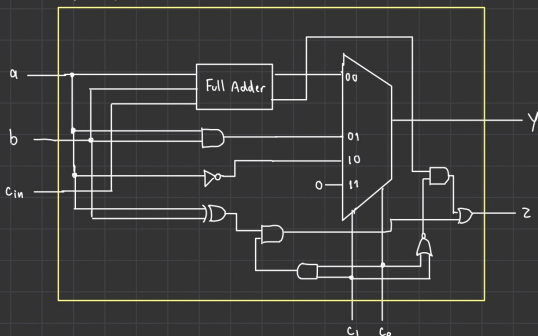
$c_1$	$c_0$	y	z
0	0	sum of (a, b, ci)	Carry out
0	1	a AND b	0
1	0	NOT a	0
1	1	0	“0” if a=b “1” if a!=b

- ออกแบบ โครงสร้างของ Verilog
- เขียน Code ด้วย Verilog ทั้งส่วน Design และ ส่วน Stimulus

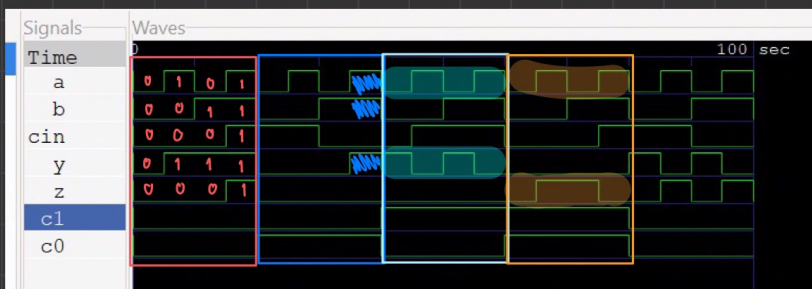
1.



1bit ALU



Handbook



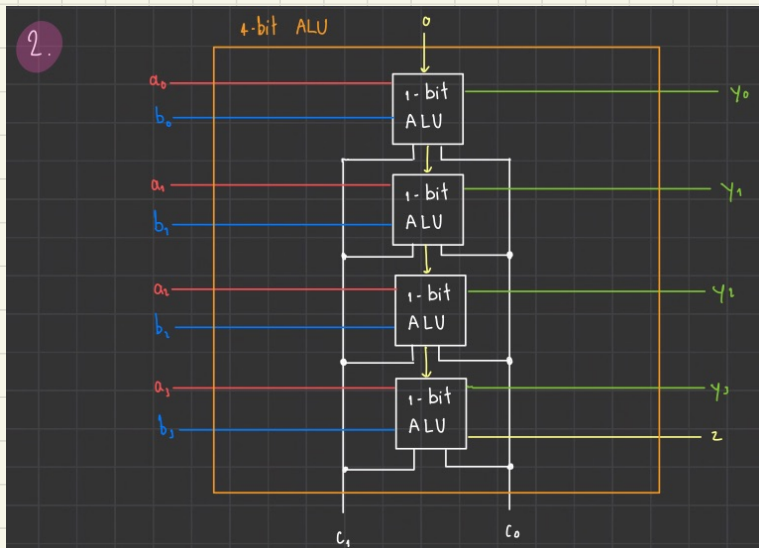
- ที่ select 00 จะเห็นว่า full adder นั้นทำงานได้ถูกต้องตามปกติ
- ที่ select 01 จะเห็นว่า y เป็น 1 ตามที่ควรจะเป็น a and b ซึ่งถูกต้อง
- ที่ select 10 จะเห็นว่า y เป็น not a ตามที่ถูกต้อง
- ที่ select 11 จะเห็นว่า y เป็น 0 ตามที่ควรจะเป็น z ได้ 1 เมื่อ a = b

- แสดงผลการทำงานที่ได้ และอธิบายว่า 1-bit ALU ที่ได้ทำงานถูกต้องอย่างไร

2) จากโมดูล 1-bit ALU (Arithmetic Logic Unit) ที่ได้ให้นำมาพัฒนาต่อเป็น 4-bit ALU โดยรับอินพุต a0-a3 และ b0-b3 โดยผลลัพธ์จะมี y0-y3 และ z ซึ่งเป็นผลการปฏิบัติการของอินพุต ซึ่งถูกเลือกปฏิบัติการตามสัญญาณควบคุม  $c_1$  และ  $c_0$

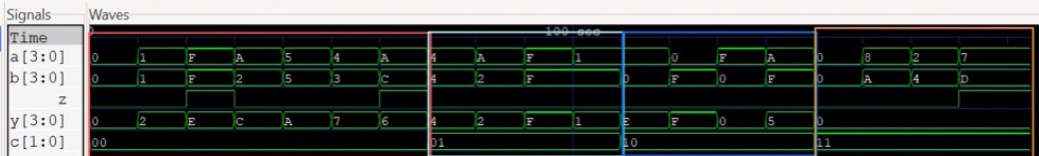
- ออกแบบ โครงสร้างของ Verilog
- เขียน Code ด้วย Verilog ทั้งส่วน Design และ ส่วน Stimulus
- แสดงผลการทำงานที่ได้ และอธิบายว่า 4-bit ALU ที่ได้ทำงานถูกต้องอย่างไร

2.



- select 10: 4-bit ALU  $y$  is 4-bit  $\neg a$  if  $0001 = 1110$ ,  $0000 = 1111$ ,  $1111 = 0000$  and  $z = 0$  correct ✓
- select 11: 4-bit ALU  $z$  is 1 if msb of  $a, b$  is 1  
if  $0111/1101 = 1$ ,  $0000/0000 = 0$  and  $y = 0$  correct ✓  
7 / D      0 / 0

### Waveform



- select 00: 4-bit ALU  $y$  is 4-bit  $a + b$  if  $1+1=02$ ,  $F+F=1E$ ,  $A+C=16$  last msb is  $z$ , lsb is  $y$  ✓
- select 01: 4-bit ALU  $y$  is 4-bit  $a, b$  and  $z$  if  $0100/0100 = 0100$ ,  $1010/0010 = 0010$   
4 / 4      4      A / 2      2 ✓

3) พัฒนางจรนับขึ้น,นับลง 2 บิต (00,01,10,11) โดยให้มีสัญญาณอินพุต (X) เป็นตัวควบคุมการนับขึ้นหรือนับลง โดยสัญญาณจะนับขึ้นก็ต่อเมื่อ X=0 และจะนับลงเมื่อ X=1 ออกแบบโดยใช้ Module D Flipflop

```
module D_FF(q,d,clk,reset);  
    output q;  
    input d,clk,reset;  
    reg q;  
    always @ (posedge reset or negedge clk)  
    if(reset)  
        q <= 1'b0;  
    else  
        q <= d;  
endmodule
```

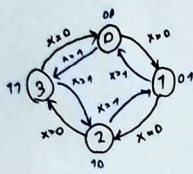
- ออกแบบ โครงสร้างของ Verilog
- เขียน Code ด้วย Verilog ทั้งส่วน Design และ ส่วน Stimulus
- แสดงผลการทำงานที่ได้ และอธิบายว่าวงจรนับที่ได้ทำงานถูกต้องอย่างไร

### ③. 2 Bit Count UP/Down (D Flip-Flop)

input x : x = 0 count up

x = 1 count down

#### ①. State Diagram



#### ②. Transition Table

X	$Q1_{t-1}$	$Q0_{t-1}$	$Q1_t$	$Q0_t$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

(multiple traits)  
K-map

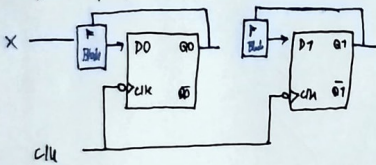
$Q1_t$  /  $Q1_{t-1}, Q0_{t-1}$

	00	01	11	10
0	X	0	1	0
1	1	0	1	0

$Q0_t$  /  $Q1_{t-1}, Q0_{t-1}$

	00	01	11	10
0	X	1	0	0
1	1	0	0	1

\* 74 D-Flip Flop 2 bits



#### ③. Design F Block

D1

X	$Q1_{t-1}, Q0_{t-1}$	00	01	11	10
0		0	1	0	1
1		1	0	1	0

$Q1_{t-1}$	$Q0_{t-1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

$$D1 = \bar{x}\bar{Q1}Q0 + \bar{x}Q1\bar{Q0} + x\bar{Q1}\bar{Q0} + xQ1Q0$$

$$= \bar{x}(Q1Q0 + Q1\bar{Q0})$$

$$+ x(\bar{Q1}\bar{Q0} + Q1Q0)$$

$$= \bar{x}(Q1 \oplus Q0)$$

$$+ x(Q1 \text{ XOR } Q0)$$

$$D1 = x \oplus Q1 \oplus Q0$$

D0

X	$Q1_{t-1}, Q0_{t-1}$	00	01	11	10
0		1	0	0	1
1		1	0	0	1

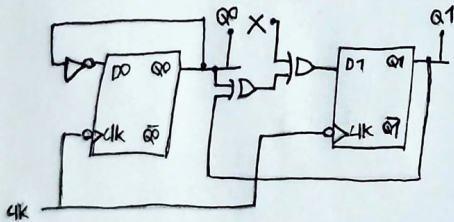
$\bar{Q1}\bar{Q0}$        $Q1\bar{Q0}$

$$D0 = \bar{Q1}\bar{Q0} + Q1\bar{Q0}$$

$$= \bar{Q0}(\bar{Q1} + Q1)$$

$$D0 = \bar{Q0}$$

#### ④. Logic Gate Diagram



$X = 0$  ဘာလဲလဲ

$X = 1$  ဘာလဲလဲ



```

0 output q = 0 (00) reset = 1 x = 0
25 output q = 0 (00) reset = 0 x = 0
30 output q = 1 (01) reset = 0 x = 0
40 output q = 2 (10) reset = 0 x = 0
50 output q = 3 (11) reset = 0 x = 0
60 output q = 0 (00) reset = 0 x = 0
70 output q = 1 (01) reset = 0 x = 0
80 output q = 2 (10) reset = 0 x = 0
90 output q = 3 (11) reset = 0 x = 0
100 output q = 0 (00) reset = 1 x = 1
125 output q = 0 (00) reset = 0 x = 1
130 output q = 3 (11) reset = 0 x = 1
140 output q = 2 (10) reset = 0 x = 1
150 output q = 1 (01) reset = 0 x = 1
160 output q = 0 (00) reset = 0 x = 1
170 output q = 3 (11) reset = 0 x = 1
180 output q = 2 (10) reset = 0 x = 1
190 output q = 1 (01) reset = 0 x = 1
200 output q = 0 (00) reset = 0 x = 1

```

နံပါတ်  $X = 0$  (စုံကလေးလဲ)

ယူကုတ် negative edge သို့မဟုတ် clk ဘာလဲလဲ state လဲလဲ q သိသိ

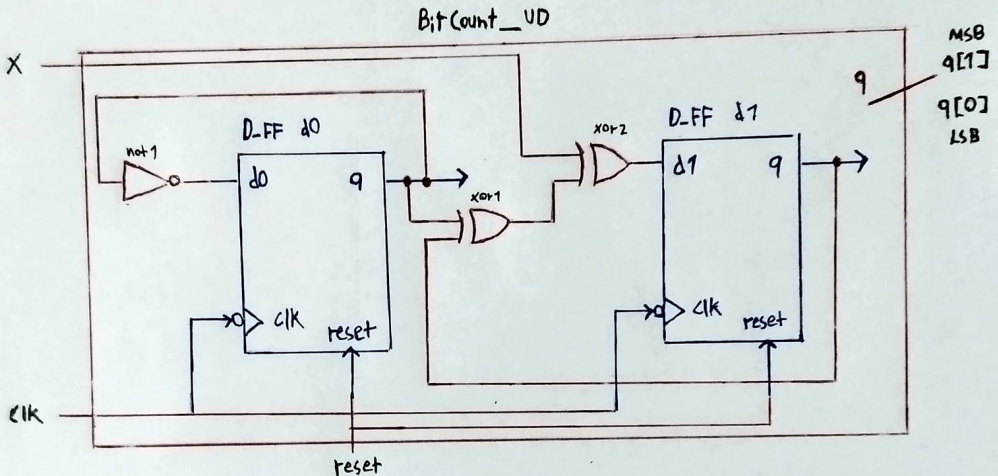
00, 01, 10, 11, 00, 01, 10, 11 သိသိလဲလဲ

နံပါတ်  $X = 1$  (စုံကလေးလဲ)

ယူကုတ် negative edge သို့မဟုတ် clk ဘာလဲလဲ state လဲလဲ q သိသိ

00, 11, 10, 01, 00, 11, 10, 01 ... သိသိလဲလဲ

ကွက်ကွက် Verilog



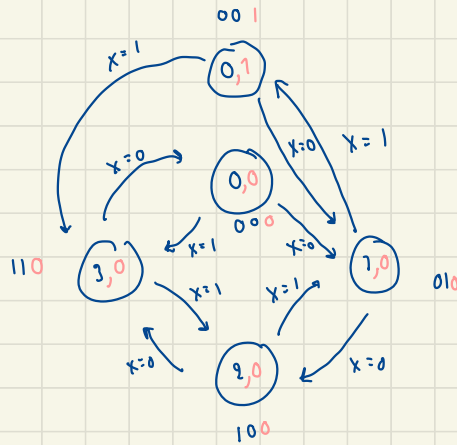
4) จากวงจรนับขึ้น, นับลง 2 บิต ที่ได้ในข้อ 3 ให้เพิ่มสัญญาณเอาต์พุตโดยให้สัญญาณเอาต์พุตเป็น 1 ก็ต่อเมื่ออยู่ที่สถานะ 01 และค่าอินพุต(x) เป็น 1 เท่านั้นในกรณีอื่นสัญญาณเอาต์พุตนี้จะเป็น 0

- ออกแบบ โครงสร้างของ Verilog
- เขียน Code ด้วย Verilog ทั้งส่วน Design และ ส่วน Stimulus
- แสดงผลการทำงานที่ได้ และอธิบายว่าวงจรนับที่ได้ทำงานถูกต้องอย่างไร



4

# State Machine



## transition table

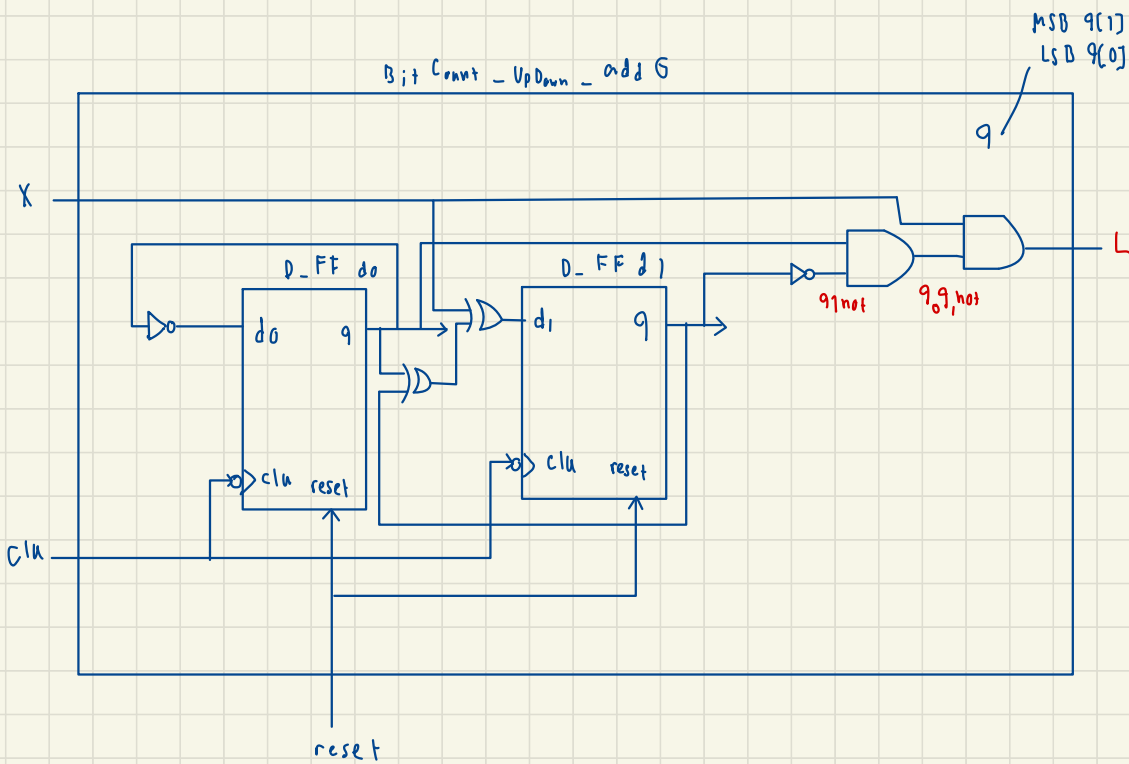
$x$	$Q1_{t-1}$	$Q0_{t-1}$	$Q1_t$	$Q0_t$	New Output
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	1	0	0

## K-map New output

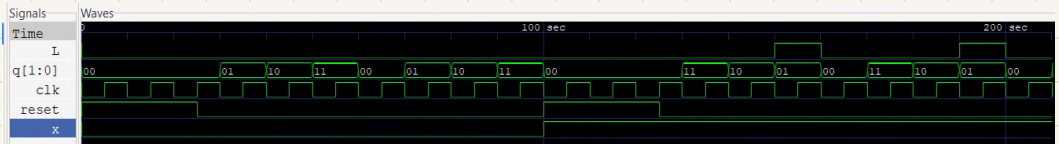
		New output					
		$Q1_{t-1}$	$Q0_{t-1}$	00	01	11	00
X	0	0	0	0	0	0	0
	1	0	1	0	0	0	0

$$\text{New output} = x \bar{Q1} Q0$$

# Assignment Verilog



לדונת מר 42



```

25 output L = 0 q = 0 (00) reset = 0 x = 0
30 output L = 0 q = 1 (01) reset = 0 x = 0
40 output L = 0 q = 2 (10) reset = 0 x = 0
50 output L = 0 q = 3 (11) reset = 0 x = 0
60 output L = 0 q = 0 (00) reset = 0 x = 0
70 output L = 0 q = 1 (01) reset = 0 x = 0
80 output L = 0 q = 2 (10) reset = 0 x = 0
90 output L = 0 q = 3 (11) reset = 0 x = 0
100 output L = 0 q = 0 (00) reset = 1 x = 1
125 output L = 0 q = 0 (00) reset = 0 x = 1
130 output L = 0 q = 3 (11) reset = 0 x = 1
140 output L = 0 q = 2 (10) reset = 0 x = 1
150 output L = 1 q = 1 (01) reset = 0 x = 1
160 output L = 0 q = 0 (00) reset = 0 x = 1
170 output L = 0 q = 3 (11) reset = 0 x = 1
180 output L = 0 q = 2 (10) reset = 0 x = 1
190 output L = 1 q = 1 (01) reset = 0 x = 1
200 output L = 0 q = 0 (00) reset = 0 x = 1

```

לדונת מר 42  
State 01 ו-10  
x ו-1