



# 디지털논리회로실습

## Lab 6. 실습설명

2017. 11. 08

HANYANG University

# Part II

## HEX 카운터 설계

- SW1 이 1일 때 KEY0을 누를 때 마다 숫자가 증가하는 16진수 카운터 설계

### ■ 설계 조건

- ✎ SW1 이 1이고 KEY0을 누를 때 마다 숫자가 증가
- ✎ SW1 이 0일 때는 아무 일도 일어나지 않음
- ✎ SW0 가 1일 때 Counter는 리셋

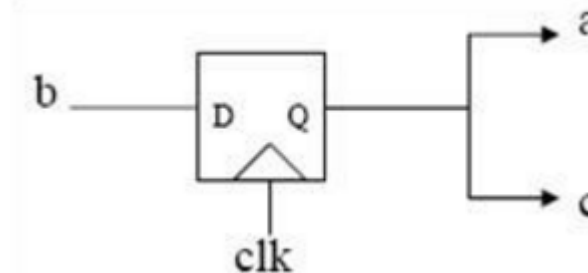
NOTE: KEY 는 누르지 않은 상태가 1임

# Part II

## Blocking Assignment (=)

- 계산과 동시에 저장이 이루어짐

```
reg [7:0] a,b,c; alwa  
ys@(clk) begin  
    b=1; a=b; c=a;  
end
```



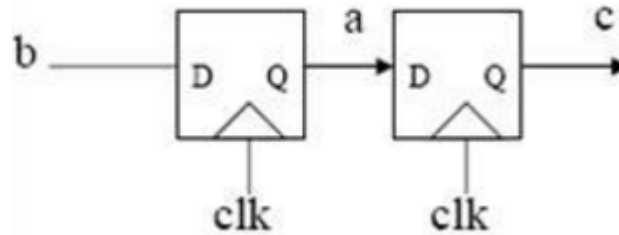
/tf_v/uut/b	15		1	2	3	4	5	6	7	8	9
/tf_v/uut/a	14		0	1	2	3	4	5	6	7	8
/tf_v/uut/c	14		0	1	2	3	4	5	6	7	8

# Part II

## Non-Blocking Assignment (<=)

- 선 계산, 후 저장으로 동작함

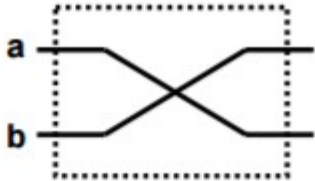
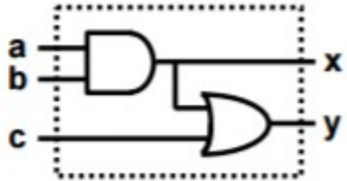
```
reg [7:0] a,b,c; always@(clk) begin  
    b<=b+1; a<=b; c<=a;  
end
```



/tf_v/uut/b	5	0	1	2	3	4	5	6	7
/tf_v/uut/a	4	0		1	2	3	4	5	6
/tf_v/uut/c	3	0			1	2	3	4	5

# Part II

## Blocking vs. Non-blocking assignments

		
<b>Blocking:</b> Evaluation and assignment are immediate	<del><math>a = b</math> <math>b = a</math></del>	$x = a \& b$ $y = x \mid c$
<b>Non-Blocking:</b> Assignment is postponed until all r.h.s. evaluations are done	$a \leq b$ $b \leq a$	<del><math>x \leq a \&amp; b</math> <math>y \leq x \mid c</math></del>
<b>When to use:</b> ( only in always blocks! )	Sequential Circuits	Combinational Circuits

# Part II

## Race condition (경쟁조건) 문제

- Always문에서 Blocking assignment를 사용하면 Race condition이 발생할 수 있음

```
a=1;  
b=2;
```

```
always @(posedge clk)  
a=b;
```

```
always @(posedge clk)  
b=a;
```

```
a=1;  
b=2;
```

```
always @(posedge clk)  
a<=b;
```

```
always @(posedge clk)  
b<=a;
```

실행되는 타이밍에 따라 a, b가 1이 되거나 2가 된다. (결과를 알 수 없음)

a=2 , b=1


**NOTE:** Always문에서는 **Non-blocking**을 사용하는 것을 권장함

# Part IV

## BCD 카운터 설계

- 50-MHz clock을 이용하여 1초당 1씩 증가하는 카운터를 설계

 NOTE : 50000000 을 저장하기 위해 26bit 가 필요함

 Hint : if(cnt==50000000) 를 이용

# Part V

## HELLO Shifter 설계

- DE2 보드의 8개 seg 에 HELLO가 이동하며 출력되는 HELLO Shifter 설계
- 설계 조건
  - 기본적으로 1초에 한번씩 왼쪽으로 이동하며 KEY0을 누르고 있을 때 이동방향이 오른쪽으로 바뀐다
  - KEY1을 누르고 있으면 이동하지 않는 상태가 되며 KEY1을 때면 다시 자동으로 문자가 이동함

Clock cycle	Displayed pattern
0	H E L L O
1	H E L L O
2	H E L L O
3	H E L L O
4	E L L O H
5	L L O H E
6	L O H E L
7	O H E L L
8	H E L L O
...	and so on

Table 1. Scrolling the word HELLO in ticker-tape fashion.