2015년 1학기 컴퓨터구조 중간고사

학과		학번		이름	

1. 현대의 컴퓨터들은 수를 표현하기 위하여 2의 보수(2's Complement)를 사용한다. 이에 대한 물음에 답하라. [총 15점]

1) 다음 두 십진수를 이진수로 표현하는데 필요한 비트수를 계산하고, 그 산출 근거를 간단하게 설명하라. (단, 부호비트를 비트수에 포함해서 계산) [각 1점]

십진수	비트수	산출근거
+64	8	n=8일 때 127까지의 양수 표현 가능
-39	7	n=7일 때 -64까지의 음수 표현 가능

2) 다음 십진수를 signed-magnitude, 1의 보수, 2의 보수로 표현하라. [각 1점]

십진수	signed-magnitude	1의 보수	2의 보수
-39	1100111	1011000	1011001

3) 1의 보수가 signed-magnitude보다 나은 점을 두 가지 제시하고, 그 이유를 간략하게 설명하라. [각 1점]

장점	이유
뺄셈연산이 간단함	덧셈 회로와 보수를 취하는 회로만으로 뺄셈이 가능
sign extension이 가능	양수는 msb왼편에 아무리 0을 추가해도 같은 양수, 음수는 msb왼편에 아무리 1을 추가해도 같은 음수가 되기 때문

4) 2의 보수가 1의 보수보다 나은 점을 두 가지 제시하고, 그 이유를 간략하게 설명하라. [각 1점]

장점	이유
표현범위가 넓음	0의 표현이 하나만 존재하기 때문
회로가 더 간단함	carry가 발생하더라도 추가적인 덧셈이 필요하지 않기 때문

2. Mano가 설계한 간단한 컴퓨터에는 8개의 레지스터(PC, AR, TR, AC, IR, DR, INPR, OUTR)와 4K X 16의 메모리를 버스로 연결하는 구조를 제안하였다. 또한

명령어 크기는 16비트로 정의하였고, 입출력은 8비트씩 처리된다고 가정하였다. 병렬전송이 가능한 버스구조라고 할 때, 다음 질문들에 답하라. [총 15점]

1) 아래 레지스터들의 기능을 간단하게 설명하고 크기를 비트로 제시하라. [각 1점]

Register	기능	크기
PC	next instruction의 address를 저장 한다	12
IR	instruction code를 저장 한다	16
AC	연산의 결과, 혹은 연산을 위한 데이터를 저장 한다	16

2) 버스를 사용하지 않는 경우 필요한 연결선의 개수를 산출하고, 버스를 사용하는 경우의 줄어드는 효과를 설명하라. [각 1점]

	연결선의 수	효과
직접 연결	(8x7/2)x16 = 448	모든 레지스터가 공통의 bus로 연결되기 때문에 필요한 연결선의 개수가 줄어든다.
버스 사용	16+16x8x2 = 272	

3) 버스제어를 MUX와 Decoder로 한다고 가정하였을 때, 필요한 MUX와 Decoder의 종류와 개수를 제시하라. [각 1점]

회로	종류	개수
MUX	8x1 MUX	16
Decoder	3x8 DEC	1

4) 버스제어를 3-state buffer gate('게이트'라 함)로 대치하는 경우 총 게이트의 수는 얼마나 필요하고, 추가로 필요한 회로는 무엇인가? [각 1점]

게이트 수	회로
256개	3x8 decoder

3. 현대 소프트웨어가 획기적으로 발전 할 수 있었던 배경에는 Indirect Addressing 기법이 큰 역할을 하였다. Mano의 컴퓨터에서 Memory Reference Instruction의 경우 direct와 indrect addressing을 구분하기 위하여 한 비트의 모드 비트를

두고, OPcode세 비트를 제외한 12비트를 address로 사용한다. [총 5점]

1) direct addressing과 indirect addressing으로 가능한 주소 공간(address space)을 구하여라. [각 1점]

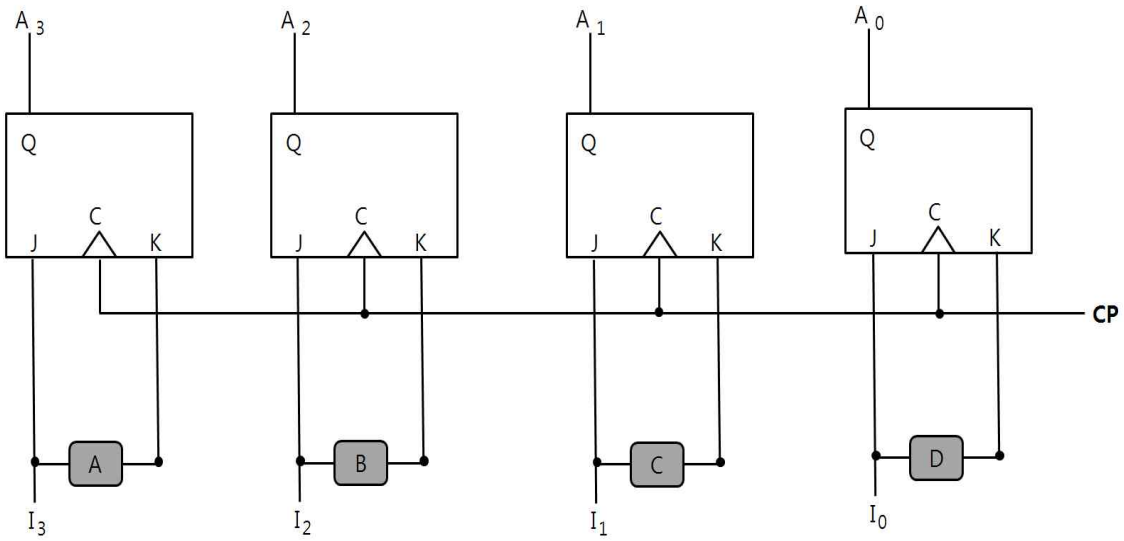
addressing	address space
Direct	2^12
Indirect	2^16

2) Indirect addressing을 사용하여서 가능하게 된 현대 프로그래밍이나 시스템상의 기능들을 세 가지 정도 기술하라. [각 1점]

더 넓은 address space 공간 사용 가능
subroutine call/return
pointer type in high level language

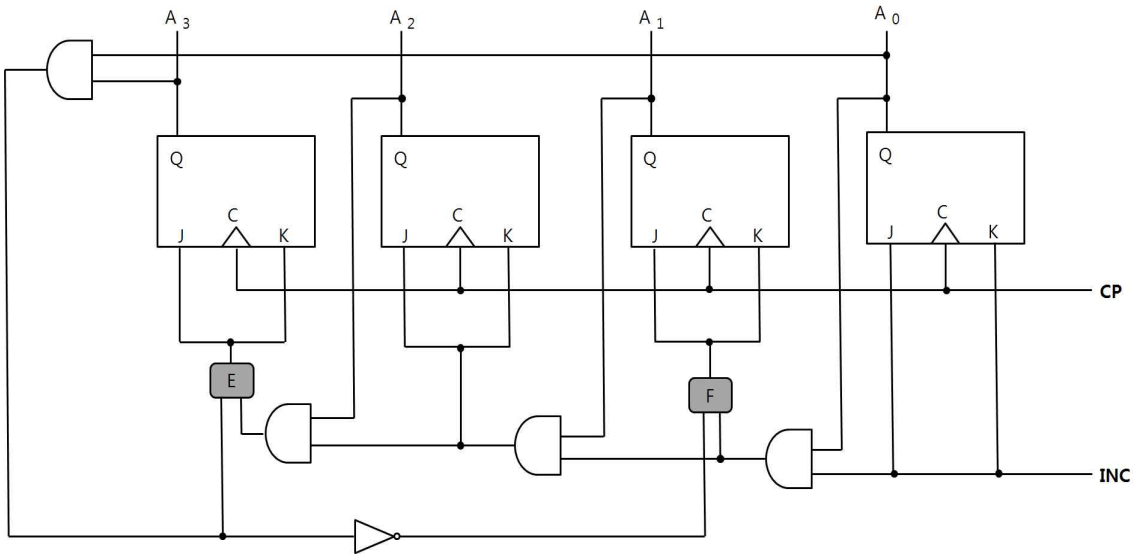
4. 4 비트크기의 레지스터를 설계하고자 한다. 다음 물음에 답하라. [총 20점]

1) 입력과 출력이 병렬적으로 가능한 레지스터를 JK flip-flop으로 설계할 때 빈칸에 들어갈 게이트들은 무엇인가? [각 1점]



A	NOT	B	NOT	C	NOT	D	NOT

2) 10진 Counter기능을 할 수 있는 레지스터로 설계할 때 빈칸에 들어갈 게이트는 무엇인가? [각 1점]

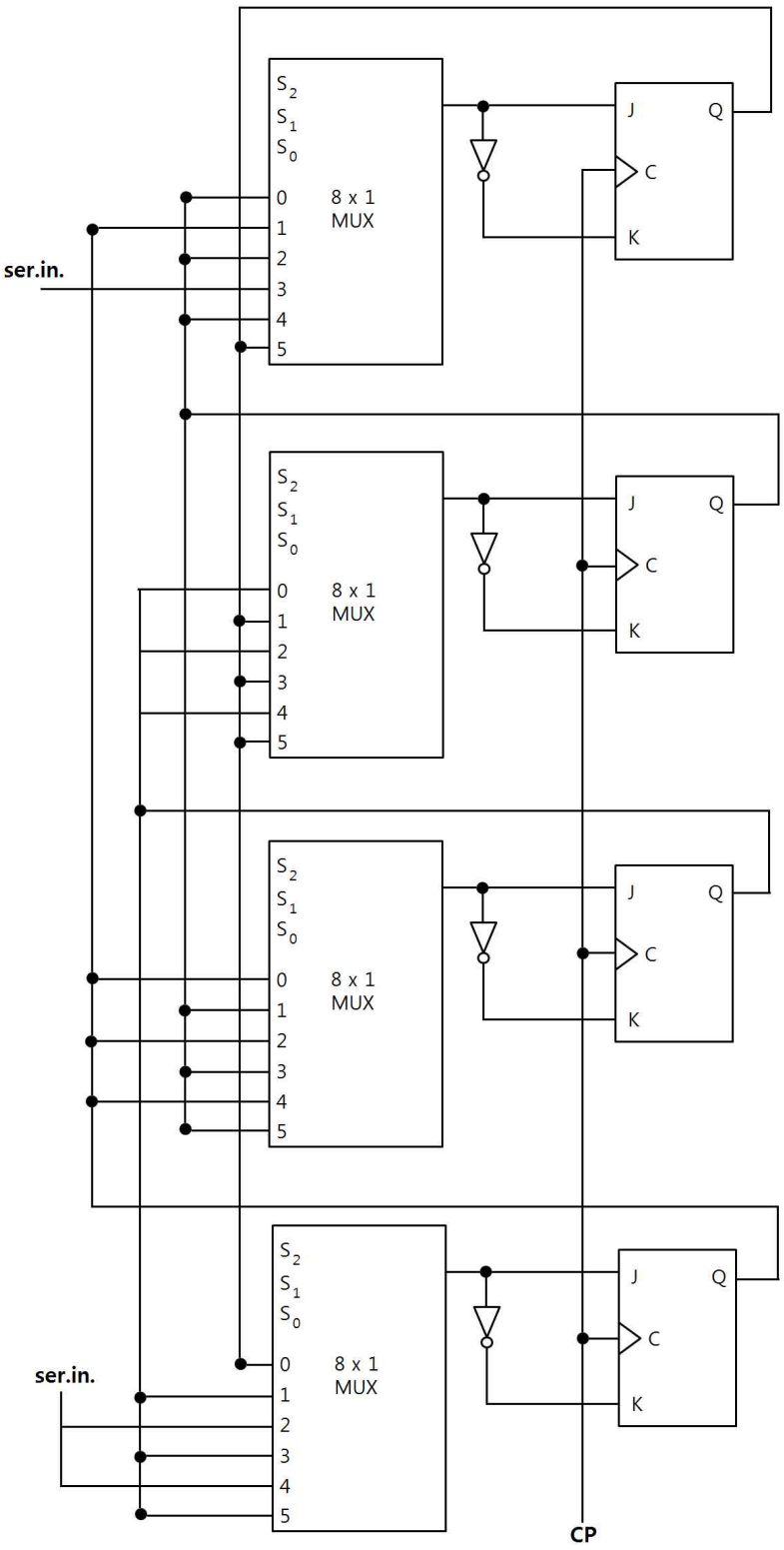


E	OR	F	AND

3) 4비트 레지스터가 저장하고 있는 값이 [1010]이라고 할 때, 다음 기능을 수행한 후의 결과를 써라. 단 최단 왼쪽비트를 부호 비트로 한다. [각 1점]

shift	left	right
circular	0101	0101
logical	0100	0101
arithmetic	0100혹은1100(overflow)	1101

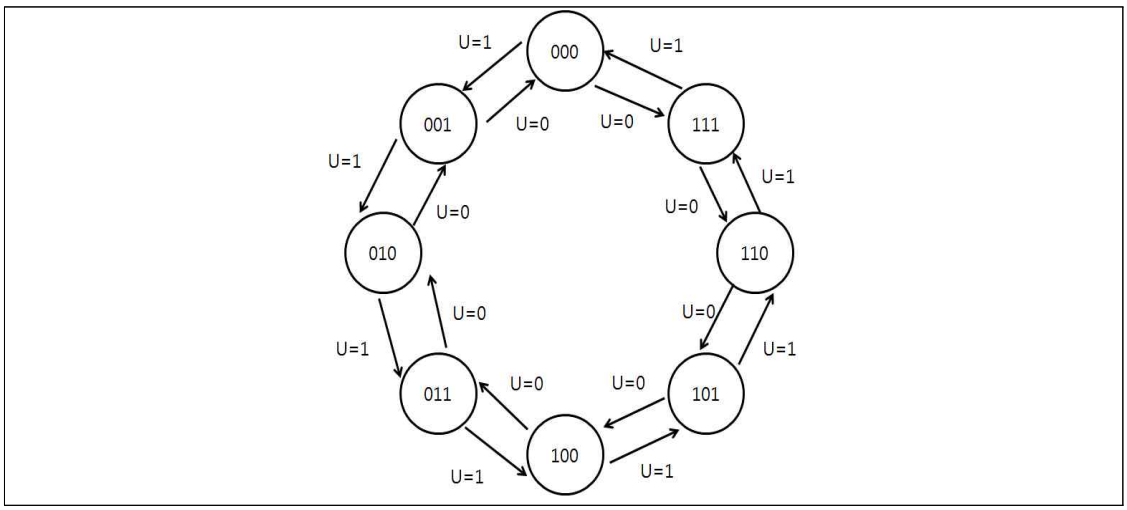
4) 위 3)번의 기능을 동시에 수행할 수 있는 레지스터를 MUX를 이용하여 설계하라. 단 각 기능을 선택하기 위한 제어선은 세 개(S0~S2) 있다. [8점]


S2
S1
S0
0
1
2
3
4
5
8 x 1
MUX
J
Q
C
K
ser.in.
ser.in.
CP

5. 본인이 하나의 문제를 만들고, 그에 대한 답을 작성하라 [5점]

6. T flip-flop과 external input을 이용하여 UP, DOWN이 가능한 modulo-8 counter를 설계하려고 한다. UP = 1일 때는 counter가 0,1,2,...,7을 카운트하고, UP = 0일 때는 counter가 7,6,5,...,0을 카운트 한다. 아래 과정을 통해 회로를 설계하라. [총 13점]

(1) modulo-8 with UP/DOWN counter를 위한 state diagram을 그려라. [2점]



(2) state transition table을 T flip-flop의 excitation table(여기표)를 이용하여 나타내라. [2점]

Present state			Input	Next state			Flip-flip inputs		
Q_1	Q_2	Q_3	U	Q_1	Q_2	Q_3	T_1	T_2	T_3
0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0	1
0	0	1	1	0	1	0	0	1	1
0	1	0	0	0	0	1	0	1	1
0	1	0	1	0	1	1	0	0	1
0	1	1	0	0	1	0	0	0	1
0	1	1	1	1	0	0	1	1	1
1	0	0	0	0	1	1	1	1	1
1	0	0	1	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0	1
1	0	1	1	1	1	0	0	1	1
1	1	0	0	1	0	1	0	1	1
1	1	0	1	1	1	1	0	0	1
1	1	1	0	1	1	0	0	0	1
1	1	1	1	0	0	0	1	1	1

(3) T flip-flop 설계를 위한 K-map을 그리고 식을 유도하여라. [각 2점]

Q_1Q_2 \ Q_3U	00	01	11	10
00	1	0	0	0
01	0	0	1	0
11	0	0	1	0
10	1	0	0	0

$T_1 = Q_2Q_3U + Q_2'Q_3'U'$

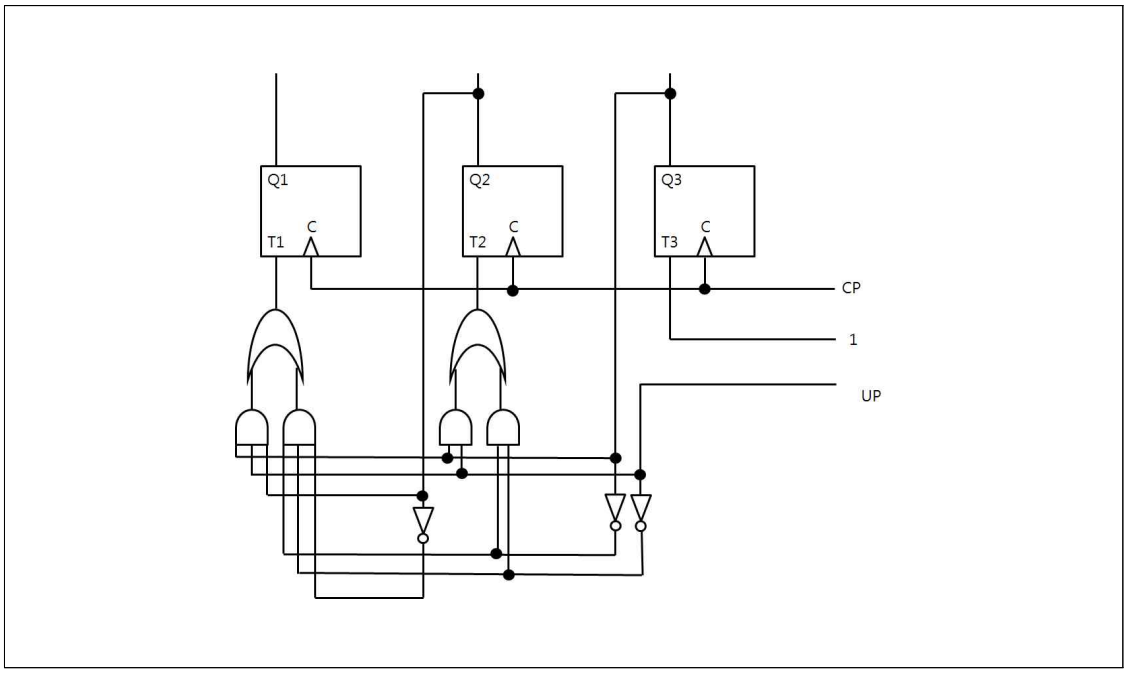
Q_1Q_2 \ Q_3U	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	1	0	1	0
10	1	0	1	0

$T_2 = Q_3U + Q_3'U' = (Q_3 \oplus U)'$

Q_1Q_2 \ Q_3U	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$T_3 = 1$

(4) 최종 설계를 diagram으로 표현하라. [3점]



7. functional completeness 이론에 의하면 operation들로 이루어진 집합은 어떤 조건을 만족할 때 functionally complete하다고 한다. [총 9점]

(1) operation을 원소로 가지는 집합 A가 functionally complete 하기 위한 조건은 무엇인가? [3점]

집합 A의 원소들로 모든 boolean function을 표현할 수 있어야 한다

(2) 집합 A가 functionally complete함을 증명하는 한 방법은 이미 functionally complete하다고 알려진 집합 B의 모든 원소들을 A집합의 원소들을 이용해 만들어낼 수 있음을 보이는 것이다. 집합 B의 예를 들고, 이를 이용해 {NOR} 집합이 functionally complete함을 보여라. [3점]

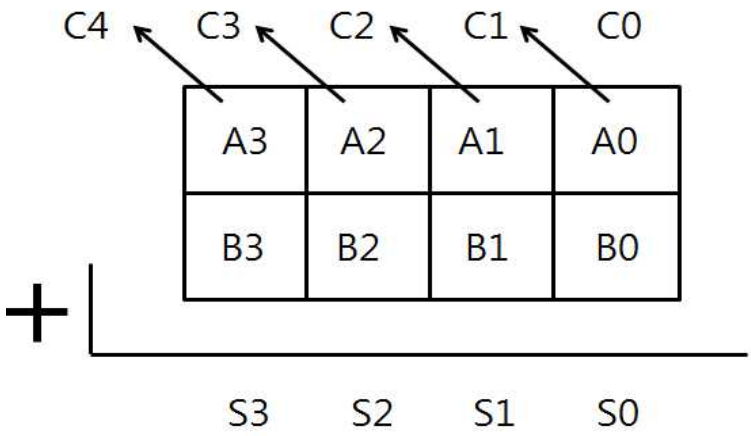
집합 B의 예 : {+, ·, ′}
x′ = x NOR x
x+y = ((x+y)′)′ = (x NOR y)′ = (x NOR y) NOR (x NOR y)
x·y = ((xy)′)′ = (x′+y′)′ = x′ NOR y′ = (x NOR x) NOR (y NOR y)

(3) {+, ′} 집합과 {·, ′} 집합이 functionally complete함을 각각 보여라. [3점]

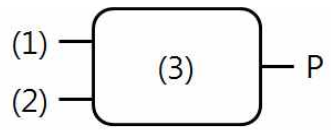
x·y = ((xy)′)′ = (x′+y′)′

x+y = ((x+y)′)′ = (x′·y′)′

8. 두 개의 binary number에 대한 덧셈을 수행할 때, overflow의 발생 여부는 더하려는 두 수가 signed number인지, unsigned number인지에 따라 달라질 수 있다. 다음 그림은 4bit 크기의 두 binary number A, B에 대한 덧셈을 수행하는 과정을 나타낸다. 3번 위치는 most significant bit 위치를 나타낸다. C1이 0번 위치에서 1번 위치로 올라가는 output carry를 나타낸다고 하였을 때, 다음 질문에 답하여라. [총 9점]

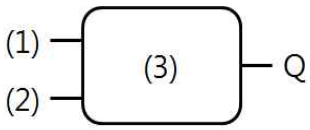


(1) 두 숫자가 unsigned number일 때, 덧셈을 수행한 후 overflow detection을 할 수 있는 회로를 설계한다고 하자. output P는 overflow가 발생했을 때 1을 출력한다. 빈칸을 채우시오. [각 1점]



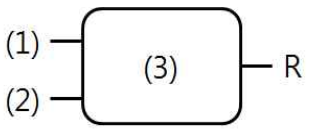
(1)	C4	(2)	C4	(3)	AND

(2) 두 숫자가 signed number일 때, 덧셈을 수행한 후 overflow detection을 할 수 있는 회로를 설계한다고 하자. output Q는 overflow가 발생했을 때 1을 출력한다. 빈칸을 채우시오. [각 1점]



(1)	C4	(2)	C3	(3)	XOR

(3) 현재 A가 4bit 크기의 레지스터 안에 저장되어 있다고 하자. A가 저장된 레지스터에 대해 Arithmetic Shift Left 연산을 수행할 때, overflow detection을 할 수 있는 회로를 설계하려고 한다. output R은 overflow가 발생했을 때 1을 출력 한다. 빈칸을 채우시오. (단, 해당 회로는 Arithmetic Shift Left 연산을 수행하기 전에 동작한다.) [각 1점]



(1)	A3	(2)	A2	(3)	XOR

9. 1024x16 크기의 RAM이 있다. 이 메모리에 access하기 위한 회로를 2x4 decoder, 3x8 decoder, 5x32 decoder를 이용해서 만들려고 한다. 5x32 decoder를 가장 적게 사용하고, 2x4 decoder를 가장 많이 사용해서 설계한다고 했을 때, 필요한 decoder의 개수를 각각 구하여라. (단 세 종류의 decoder 모두 최소 1번은 사용되어야 한다.) [총 9점]

2x4 DEC	$2^5 \cdot 2^3$ 개	3x8 DEC	2^5 개	5x32 DEC	1개