

학과		학번		이름	
----	--	----	--	----	--

1. 십진수를 이진수로 표현할 수 있는 방법은 ① signed-magnitude ② 1's complement(1의 보수) ③ 2's complement(2의 보수)등 세 가지가 있다. [총 20점]

- 1) 부호를 포함하여 n 비트로 표현할 수 있는 수의 범위를 나타내라. [각 2점]

방법	범위
①	$-(2^{n-1} - 1) \sim 2^{n-1} - 1$
②	$-(2^{n-1} - 1) \sim 2^{n-1} - 1$
③	$-2^{n-1} \sim 2^{n-1} - 1$

- 2) n = 8로 가정하여 다음 십진수를 signed-magnitude, 1의 보수, 2의 보수로 표현하라. [각 2점]

십진수	signed-magnitude	1의 보수	2의 보수
-64	11000000	10111111	11000000

- 3) 1의 보수가 signed-magnitude보다 나은 점을 두 가지 제시하고, 실제 숫자를 예로 들어 설명하라. [각 2점]

뺄셈연산이 간단함 (0101-0010의 경우 1의 보수 방식은 0101과 1101을 더한 뒤 발생하는 캐리를 다시 더해주면 00110이 되지만 signed-magnitude의 경우에는 0101과 1010을 단순히 더하는 방식으로 뺄셈을 수행할 수 없음) sign extension이 가능 (4비트 값 1101을 6비트로 확장한다고 했을 때 1의 보수 방식에서는 단순히 msb 원편에 모두 1을 붙여 111101로 표현이 되지만 signed-magnitude의 경우에는 부호를 표현하는 msb값이 달라져야함)
--

- 4) 2의 보수가 1의 보수보다 나은 점을 두 가지 제시하고, 실제 숫자를 예로 들어 설명하라. [각 2점]

0의 표현이 오직 하나만 존재

($n=3$ 일 때 2의 보수는 000으로 0을 표현하지만 1의 보수는 000과 111로 0을 표현함)

덧셈 연산 시 발생되는 캐리를 신경 쓰지 않아도 됨

(0101-0010의 경우 2의 보수 방식은 $0101+1110 = 0011$ 로 계산되나 1의 보수 방식은 $0101+1101$ 을 수행하여 나온 값 0010에 발생된 캐리를 다시 더해야함)

2. 오늘날 컴퓨터는 데이터를 전송하는 과정에서 외부의 환경 요인에 의한 오류를 검출하기 위하여 패리티비트를 사용한다. 다음 질문들에 답하라. [총 20점]

- 1) 4비트 데이터 x, y, z, w 에 대하여 생성되는 odd 패리티 비트 P 와 오류를 검출하기 위한 C (=1이면 오류 발생)를 계산하기 위한 수식을 작성하라. [각 2점]

P	$(x \oplus y \oplus z \oplus w)'$
C	$(x \oplus y \oplus z \oplus w \oplus P)'$

- 2) 위와 같은 경우, 오류를 검출할 수 없는 확률을 계산하라. 전송 중 비트 하나가 변할 확률은 λ 이다. [3점]

$$_5 C_2 \lambda^2 (1 - \lambda)^3 + _5 C_4 \lambda^4 (1 - \lambda)$$

- 3) 1)의 4비트 데이터에 대해 오류가 발생한 비트의 위치를 검출하여 수정할 수 있는 해밍코드를 만들려고 한다. 이 때 패리티 비트는 최소한 3비트가 필요하다. 이유를 적으시오. [3점]

데이터비트의 개수 = n , 패리티비트의 개수 = m 이라고 했을 때
 m 개의 패리티비트를 사용하여 $(2^m) - m - 1$ 개의 데이터비트의 오류를 검출할 수 있기 때문에 $n=4$ 일 때 필요한 m 의 개수는 3이 된다.

- 4) 추가된 3개의 패리티 비트 I, m, n 을 계산하는 수식을 작성하라. [각 2점]

I	$(x \oplus y \oplus w)'$
m	$(x \oplus z \oplus w)'$
n	$(y \oplus z \oplus w)'$

- 5) 추가된 3개의 패리티 비트 I, m, n 을 이용하여 오류의 위치를 검출하는 방법을 설명하라. [4점]

수신자가 전송받은 데이터 비트를 x, y, z, w 라 하고 전송받은 패리티 비트를 I, m, n 이라고 하자. 이 때 오류 검출을 위해 아래와 같이 l^*, m^*, n^* 을 계산한다.

$$l^* = (x \oplus y \oplus w \oplus I)'$$

$$m^* = (x \oplus z \oplus w \oplus m)'$$

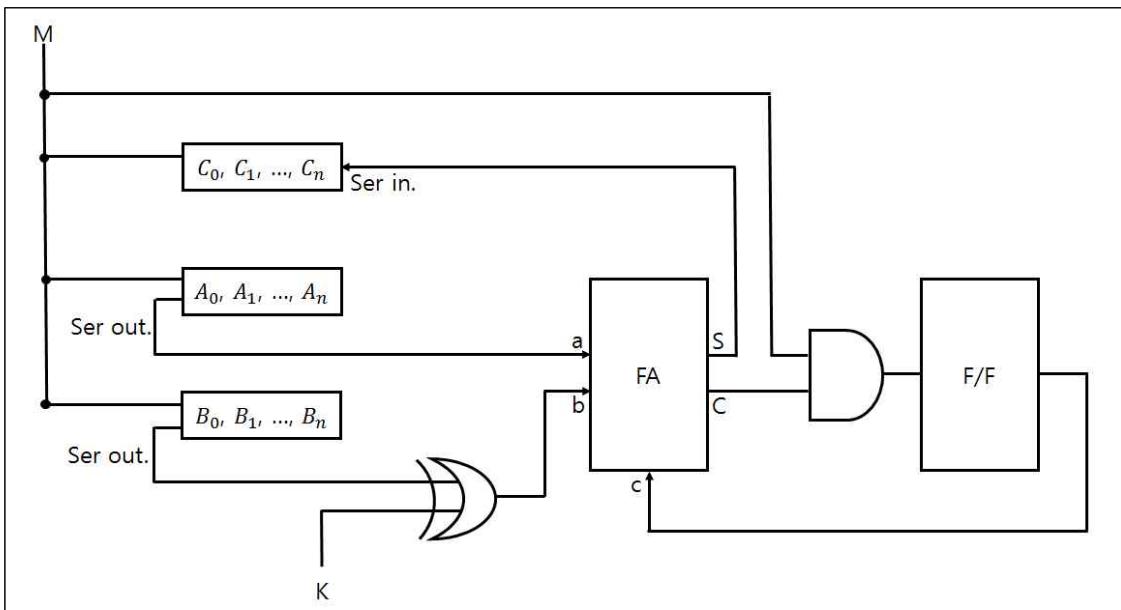
$$n^* = (y \oplus z \oplus w \oplus n)'$$

l^*, m^*, n^* 의 값 및 대응되는 오류 bit에 대한 truth table은 아래와 같다.

n^*	m^*	l^*	error
0	0	0	
0	0	1	I
0	1	0	m
0	1	1	x
1	0	0	n
1	0	1	y
1	1	0	z
1	1	1	w

3. 두 개의 shift register(A, B)에 저장된 수에 대하여 덧셈과 뺄셈을 수행하여 결과를 shift register(C)에 저장하는 Binary Serial Adder(BSA)를 Full Adder와 Flip-Flop을 이용하여 만들려고 한다. 입력 값 M은 shift left 기능에 대한 제어선을 의미하며, 입력 값 K는 0이면 덧셈, 1이면 뺄셈을 수행하게 하는 제어선을 의미한다. [총 15점]

- 1) 아래 배치도에서 비어 있는 연결 부분을 완성하라. [4점]



2) 두 개의 빈칸(X, Y)에 들어갈 게이트 명을 써라. [각 2점]

X	XOR	Y	AND
---	-----	---	-----

3) BSA가 제대로 수행되려면 Flip-Flop의 초기 값에 대한 하나의 가정이 필요하다. 무엇인가? [3점]

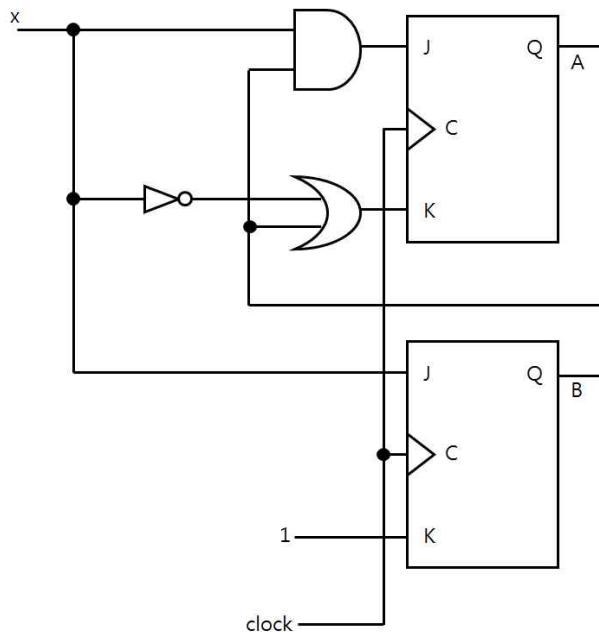
플립플롭의 초기 값은 k이어야 한다

4) Full Adder의 출력 값 S와 C를 계산하는 식을 작성하라. (입력 값은 a, b, c로 표현) [각 2점]

S	$a \oplus b \oplus c$
C	$ab + (a \oplus b)c$

4. 아래에 주어진 그림은 두 개의 JK flip-flop A, B와 external input x를 이용하여 어떠한 동작을 수행하는 회로를 나타낸다. 이 회로가 어떻게 동작하는지를 조사하기 위해 sequential circuit을 설계하는 과정을 거꾸로 따라가 보려고 한다. 물

음에 답하여라. [총 20점]



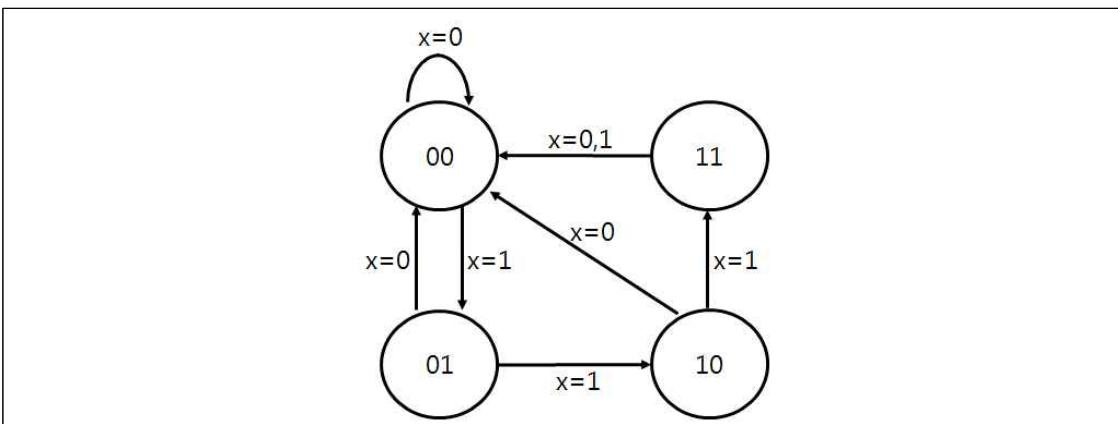
- 1) 문제에 주어진 회로에서 flip-flop A, B의 입력 값 J , K 를 boolean function 식으로 각각 표현하라. [각 2점]

J_A	Bx
K_A	$B+x'$
J_B	x
K_B	1

- 2) 1)에서 도출한 boolean function 식을 근거로 하여 아래에 주어진 state transition table의 next state 부분과 flip-flop inputs 부분을 채우시오. [3점]

Present state		Input x	Next state		Flip-flop inputs			
A	B		A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	1	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	0	0	1	0	1
0	1	1	1	0	1	1	1	1
1	0	0	0	0	0	1	0	1
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	0	1
1	1	1	0	0	1	1	1	1

- 3) 2)의 table에서 도출한 present state와 next state, x 정보를 참고로 하여 해당 회로의 state diagram을 그리시오. [3점]



- 4) 결론적으로 문제에서 주어진 회로는 어떠한 기능을 수행하는가? $x = 0$ 일 때와 $x = 1$ 일 때 회로가 수행하는 기능을 아래 빈 칸에 작성하시오. [각 3점]

$x = 0$	clear 기능을 수행
$x = 1$	increment 기능을 수행(00, 01, 10, 11, 00, ...)

5. 아래 물음에 답하시오. [총 10점]

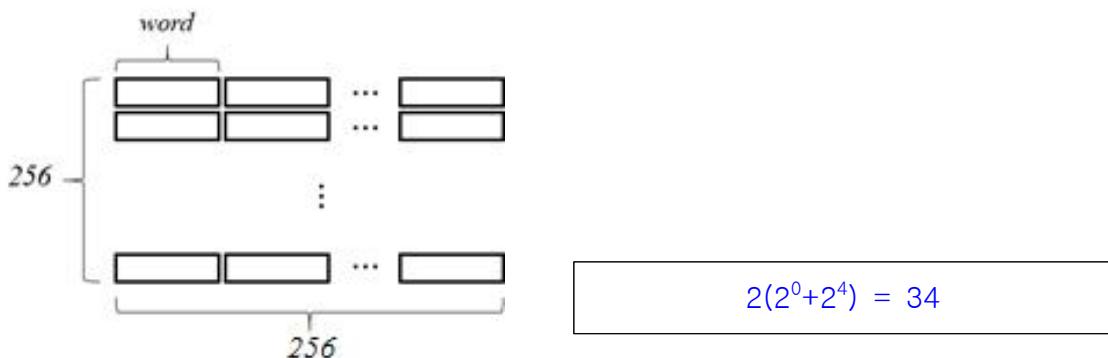
- 1) 64K-word RAM($2^{16} \times 16$)을 위한 주소는 최소 몇 비트가 필요한가? [2점]

16-bit

- 2) 위 RAM에 direct(random) access 하기 위한 회로를 4x16 decoder만으로 만들기 위해서는 총 몇 개의 4x16 decoder가 필요한가? [3점]

$$2^0 + 2^4 + 2^8 + 2^{12} = 4369$$

- 3) 아래 그림과 같이 특정 word의 주소가 (x, y)의 형태를 지니는 2차원 addressing 구조로 위 RAM을 설계하였을 때, 총 몇 개의 4x16 decoder가 필요한가? (단, 4x16 decoder만을 사용한다고 가정) [2점]



- 4) word의 주소가 (x, y, z)로 표현되는 3차원 addressing 구조로 위 RAM을 설계하고자 한다. 필요한 4x16 decoder의 개수가 최소가 되도록 RAM을 구성한다고 했을 때, 주소 비트 x, y, z의 크기는 얼마가 되는가? (단, 4x16 decoder만을 사용한다고 가정하며, $x \geq y \geq z$ 가 되도록 한다) [각 1점]

x	8	y	4	z	4
---	---	---	---	---	---

6. Boolean function F가 다음과 같이 주어졌을 때, 아래 그림의 5-variable K-map을 이용하여 F에 대한 간소화된 식을 도출하시오. [5점]

$$F(A, B, C, D, E) = \sum(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$$

		CDE	000	001	011	010
AB	00		1	1	1	
AB	01			1		
AB	11					
AB	10		1	1		

		CDE	100	101	111	110
AB	00		1	1		
AB	01		1			
AB	11		1	1		
AB	10				1	

$$F = B'C'E + A'C'DE + A'B'C'D + A'B'E + BCD'E + ACDE$$

7. 아래의 문장들에 대하여 O/X로 답하라. [각 2점] (단, 틀리면 1점 감점이 있음)

- 1) BCD코드를 8421코드라고 부르기도 한다.
- 2) SR, JK, D, T flip-flop들 중 오직 D flip-flop만이 present state와 상관없이 입력으로 들어오는 값에 의해 flip-flop의 next state가 결정된다.
- 3) NAND게이트만을 사용하여 6번 문제에서 도출된 boolean function 식에 해당하는 회로를 만들 수 있다.
- 4) NAND게이트만을 사용하여 임의의 boolean function 식에 해당하는 회로를 만들 수 있다.
- 5) XOR게이트만을 사용하여 임의의 boolean function 식에 해당하는 회로를 만들 수 있다.

1	2	3	4	5
O	O	O	O	X