

학과		학번		이름	
----	--	----	--	----	--

1. 다음의 두 boolean expression에 대하여 물음에 답하라. [총 15점]

$$(x + y) \cdot (x + z) = x + y \cdot z$$

1) 두 식이 동일한 boolean expression임을 truth table을 이용하여 증명하라. [3점]

x	y	z	$(x+y)(x+z)$	$x+yz$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

2) 두 식이 동일한 boolean expression임을 boolean algebra를 이용하여 증명하라. [3점]

$$\begin{aligned}
 (x+y)(x+z) &= xx+xz+xy+yz \\
 &= x+xz+xy+yz \\
 &= x+xy+yz \\
 &= x+yz
 \end{aligned}$$

3) K-map을 이용하여 좌변 수식으로부터 우변 수식을 도출하라. [3점]

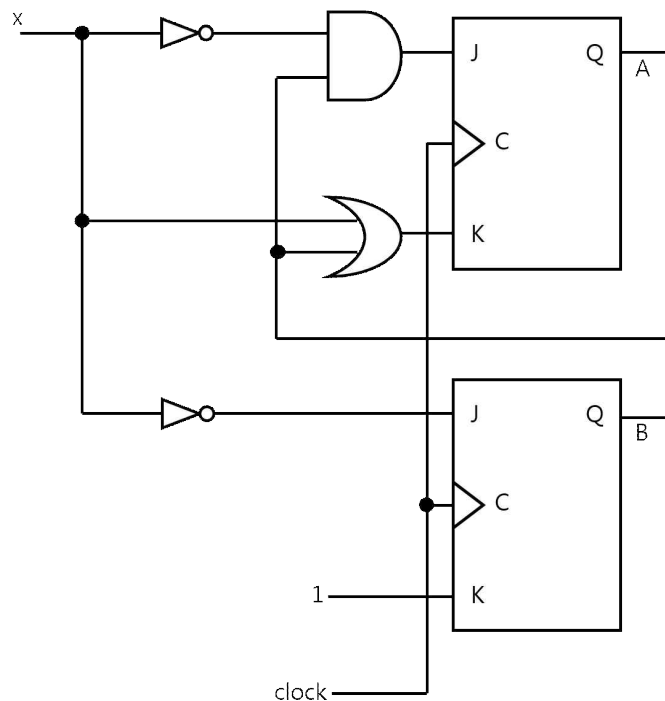
x \ yz	00	01	11	10
0	0	0	1	0
1	1	1	1	1

$$F = x + yz$$

4) 좌변과 같은 수식을 우변과 같은 수식으로 최적화하는 장점을 6가지 제시하라. [각 1점]

회로의 복잡성이 낮아진다	연산의 수행 횟수가 줄어든다
더 적은 개수의 게이트로 회로 구현 가능	연산 처리 속도 향상
회로 설계 비용을 줄일 수 있다	대량 생산에 용이

2. 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. 물음에 답하여라. [총 20점]



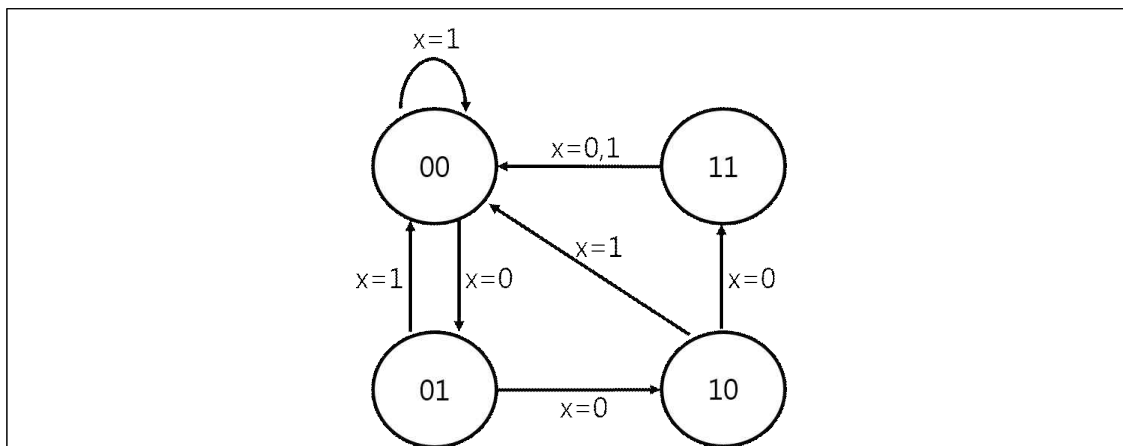
- 1) 문제에 주어진 회로에서 flip-flop A, B의 입력 값 J, K를 boolean function 식으로 각각 표현하라. [각 2점]

J_A	$x'B$
K_A	$x+B$
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	Next state		Flip-flop inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	1
0	0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	1	1
0	1	1	0	0	0	1	0	1
1	0	0	1	1	0	0	1	1
1	0	1	0	0	0	1	0	1
1	1	0	0	0	1	1	1	1
1	1	1	0	0	0	1	0	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]

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

3. 부동소수점(Floating point) 표현법은 사인부(Sign field), 지수부(Exponent Field), 유효숫자부(Mantissa Field) 세 개의 부분으로 이루어진다. [총 15점]

- 1) 사인부의 값을 s , 지수부의 값을 e , 유효숫자부의 값을 M 으로 가정하였을 때, 세 변수(s , e , M)를 이용하여 부동소수점 수를 표현하라. 단 지수부의

베이스는 2로 한다. [3점]

$$(-1)^s \times M \times 2^e$$

2) 각 부분의 크기가 늘어나는 경우, 발생하는 효과를 기술하라. [각 2점]

지수부	더 넓은 범위의 수를 나타낼 수 있음
유효숫자부	실수 표현의 정확도가 높아짐

3) 0.125와 0.625를 1)의 부동소수점 형태로 표현하라. 단, 유효숫자부는 반드시 정규화(normalization)된 상태이어야 한다. [각 2점]

0.125	$(-1)^0 \times 0.1 \times 2^{-2}$
0.625	$(-1)^0 \times 0.101 \times 2^0$

4) 유효숫자부의 크기를 8비트라고 하였을 때 표현 가능한 mantissa의 범위를 16진법의 숫자로 표현하라. [2점]

$$0.80_{16} \leq \text{Mantissa} \leq 0.FF_{16}$$

5) 이 표현법으로 나타낼 수 없는 십진수는 무엇인가? [2점]

$$0$$

4. 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) 아래의 표는 부호를 포함하여 3 비트로 표현할 수 있는 수의 경우의 수를 나타낸다. 빈칸을 채우시오. [각 열 1점]

이진표현	signed magnitude	1의 보수	2의 보수
000	0	0	0
001	1	1	1
010	2	2	2
011	3	3	3
100	-0	-3	-4
101	-1	-2	-3
110	-2	-1	-2
111	-3	-0	-1

- 2) 2의 보수 체계를 사용하여 아래 십진수를 주어진 크기에 따라 16진수로 표현하라. (표현 범위를 벗어나는 경우 X로 표기) [각 1점]

십진수	8비트	12비트
255	X	0FF
-33	DF	FDF

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

<p>뺄셈연산이 간단함 (0101-0010의 경우 1의 보수 방식은 0101과 1101을 더한 뒤 발생하는 캐리를 다시 더해주면 0011이 되지만 signed-magnitude의 경우에는 0101과 1010을 단순히 더하는 방식으로 뺄셈을 수행할 수 없음)</p>
<p>sign extension이 가능 (4비트 값 1101을 6비트로 확장한다고 했을 때 1의 보수 방식에서는 단순히 msb 왼편에 모두 1을 붙여 111101로 표현이 되지만 signed-magnitude의 경우에는 부호를 표현하는 msb값이 달라져야함)</p>

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

<p>0의 표현이 오직 하나만 존재 (n=3일 때 2의 보수는 000으로 0을 표현하지만 1의 보수는 000과 111로 0을 표현함)</p>
<p>덧셈 연산 시 발생하는 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 발생한 캐리를 다시 더해야함)</p>

- 5) 3비트 크기의 이진보수 두 수를 덧셈할 수 있는 Carry-Lookahead Adder를 설계할 때, 미리 생성되는 Carry를 두 수 $A:A_2A_1A_0$ 와 $B:B_2B_1B_0$, C_0 를 이용해 표현하라. [각 2점]

C_1	$A_0B_0 + (A_0 + B_0)C_0$
C_2	$A_1B_1 + (A_1 + B_1)A_0B_0 + (A_1 + B_1)(A_0 + B_0)C_0$
C_3	$A_2B_2 + (A_2 + B_2)A_1B_1 + (A_2 + B_2)(A_1 + B_1)A_0B_0 + (A_2 + B_2)(A_1 + B_1)(A_0 + B_0)C_0$

- 6) 위 5번의 CLA를 설계할 때, 어떤 종류의 PLD(Programmable Logic Device)가 가장 효율적일 지를 쓰고, 그 이유를 작성하라. [각 2점]

종류	PAL(Programmable Array Logic)
이유	PAL의 경우 AND plane만 프로그래밍 가능하고, OR plane은 고정되어 있기 때문에 PLA보다 더 빠른 속도로 덧셈 연산을 처리할 수 있음. 따라서 CLA 특성에 더 적합함

5. 4개의 register가 common bus로 연결된 디지털 컴퓨터가 있다. 각 register는 16 bit 크기를 가지며 common bus는 MUX(multiplexer)와 디코더로 구성되어 있다. [총 10점]

- (1) 각 MUX에는 몇 개의 selection input이 필요한가? [2점]
- (2) 어떤 종류의 MUX가 필요한가? (예를 들어, 16×1) [2점]
- (3) Common bus를 구성하기 위해 몇 개의 MUX가 필요한가? [2점]
- (4) 어떤 종류의 디코더가 필요한가? [2점]
- (5) 마이크로프로세서가 생기면서 상대적으로 간단하게 버스시스템을 구성할 수 있다. 이 경우 어떤 방법으로 버스시스템을 만들 수 있는가? [2점]

(1)	$\log_2 4 = 2$	(2)	4×1	(3)	16개
(4)	2×4	(5)	MUX 대신 three-state buffer gate 사용		

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

- 1) 3비트 데이터 x_0-x_2 에 대하여 생성되는 odd 패리티 비트 P와 오류를 검출하기 위한 C (=1이면 오류 발생)를 계산하기 위한 수식을 작성하라. [각 2점]

P	$(x_0 \oplus x_1 \oplus x_2)'$
C	$(x_0 \oplus x_1 \oplus x_2 \oplus P)'$

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

${}_4C_1 \lambda^1 (1-\lambda)^3 + {}_4C_3 \lambda^3 (1-\lambda)^1$

- 3) 1)의 3비트 데이터에 대해 Hamming이 고안한 Error Correction Code를 설계하고자 한다. 이 때 필요한 추가 코드는 최소 3비트(c_0-c_2)이다. 그 이유를 적으시오. [2점]

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) 3비트의 추가 Code 생성을 위한 Boolean식을 완성하라. [각 2점]

$c_0 =$	$(x_0 \oplus x_1)'$
$c_1 =$	$(x_0 \oplus x_2)'$
$c_2 =$	$(x_1 \oplus x_2)'$

5) Receiver가 받은 c_0-c_2 와 전달된 x_0-x_2 를 이용하여 다시 계산한 $c_0^* - c_2^*$ 값을 비교하였을 때, 아래의 결과는 어떤 의미인가? [각 2점]

c_0-c_2	$c_0^*-c_2^*$	의미
1,1,0	1,1,0	에러가 발생하지 않았다
0,0,0	0,1,1	6번째 위치의 데이터비트(x_2)에 오류 발생
1,1,1	0,0,1	3번째 위치의 데이터비트(x_0)에 오류 발생