

1. 다음 코드는 ASCII코드를 해밍코드로 변환한 것이다. 에러가 있는지 검사하고 에러가 있으면 정정하여 올바른 데이터가 무엇인지 알아내어라.

1 1 1 1 1 1 0 0 1 0 1 0

- 원래 데이터: 10101010
- 전송한 문자: *

p_1, p_2 p_4 p_8
 1 1 1 1 1 0 0 1 0 1 0

1 1 0 1 0 1 0 1 0

p_1 1 1 1 0 1 1 → 1

p_2 1 1 1 0 0 1 → 0

p_4 1 1 1 0 0 → 1

p_8 0 1 0 1 0 → 0

2×8
 10
 2 A

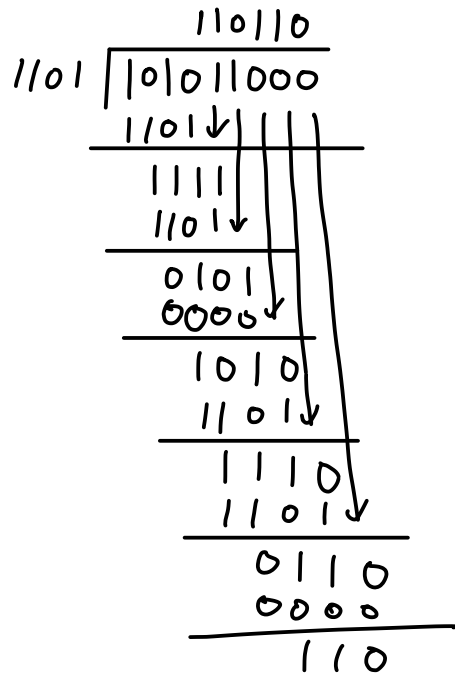
0 1 0 1

4 + 1 = 5

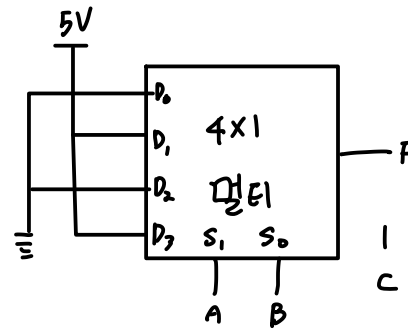
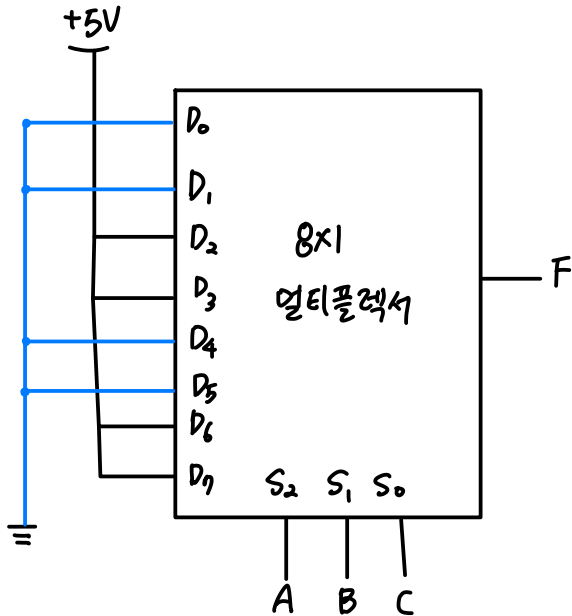
1 1 1 1 0 1 0 0 1 0 1 0
 1 0 1 0 1 0 1 0

2. 데이터가 101011이고, 키값이 1101인 경우 FCS를 계산하여라.

FCS = 110



3. $F(A,B,C) = \sum m(2,3,6,7)$ 을 8x1 멀티플렉서와 4x1 멀티플렉서로 각각 구현하고 회로도를 그려라.



4. JK 플립플롭을 사용하여 다음과 같이 동작하는 3비트 동기식 카운터를 설계하고 순서 논리 회로도를 그려라. (25점)

111 → 110 → 101 → 100 → 011 → 010 → 001 → 000 → 111 → ...

현재	다음	$J_c k_c$	$J_b k_b$	$J_a k_a$
111	110	x 0	x 0	x 1
110	101	x 0	x 1	1 x
101	100	x 0	0 x	x 1
100	011	x 1	1 x	1 x
011	010	0 x	x 0	x 1
010	001	0 x	x 0	1 x
001	000	0 x	0 x	x 1
000	111	1 x	1 x	1 x

$$\begin{array}{c|cccc} & 00 & 01 & 11 & 10 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 1 & x & x & x & x \end{array}$$

$$J_c = \overline{Q_B} \overline{Q_A}$$

$$\begin{array}{c|cccc} & 00 & 01 & 11 & 10 \\ \hline 0 & x & x & 0 & x \\ 1 & 1 & 0 & 0 & 0 \end{array}$$

$$k_c = \overline{Q_B} \overline{Q_A}$$

$$\begin{array}{c|cccc} & 00 & 01 & 11 & 10 \\ \hline 0 & 1 & 0 & x & x \\ 1 & 1 & 0 & x & x \end{array}$$

$$J_B = \overline{Q_A}$$

$$J_A = 1$$

$$\begin{array}{c|cccc} & 00 & 01 & 11 & 10 \\ \hline 0 & x & x & 0 & 1 \\ 1 & x & x & 0 & 1 \end{array}$$

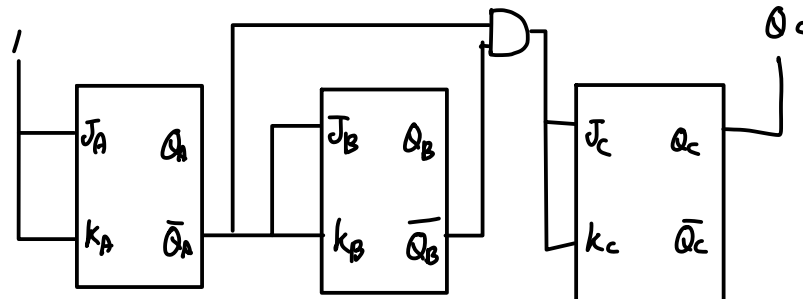
$$k_B = \overline{Q_A}$$

$$k_A = 1$$

$$J_c = \overline{Q_B} \overline{Q_A} \quad k_c = \overline{Q_B} \overline{Q_A}$$

$$J_B = \overline{Q_A} \quad k_B = \overline{Q_A}$$

$$J_A = 1 \quad k_A = 1$$



5. 다음 연산이 수행된 후 상태 레지스터의 다섯 가지 조건 플래그들(부호 플래그, 제로 플래그, 자리올림수, 패리티, 오버플로우)의 값을 구하라. 단, 패리티는 짝수로 맞춘다.

■ $0011_1010 + 0001_0111$

$$\begin{array}{r} \overset{0}{0} \overset{1}{0} \overset{1}{1} \overset{1}{1} \overset{1}{0} \overset{1}{1} \overset{0}{0} \\ 0011 \ 1010 \\ 0001 \ 0111 \\ \hline 0101 \ 0001 \end{array}$$

부호 플래그 : 0

제로 플래그 : 0

자리올림수 : 0

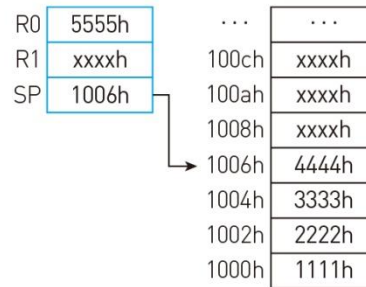
패리티 : 1

오버플로우 : 0

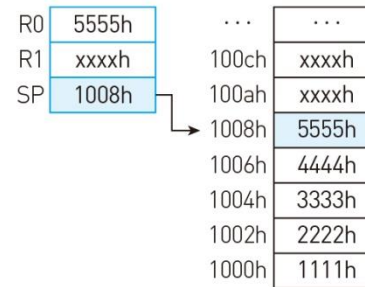
6. 컴퓨터의 상태가 옆의 그림과 같고, 스택은 아래와 같이 운영된다고 가정한다. 다음 명령어들이 연속적으로 실행된 후, 스택의 최종 상태를 오른쪽에 그려라.



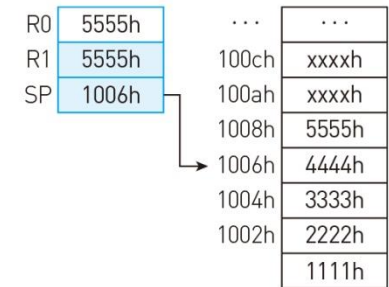
POP R3
PUSH R0
PUSH R2
PUSH R1
POP R0



(a) 초기 상태



(b) PUSH R0 실행 후



(c) POP R1 실행 후

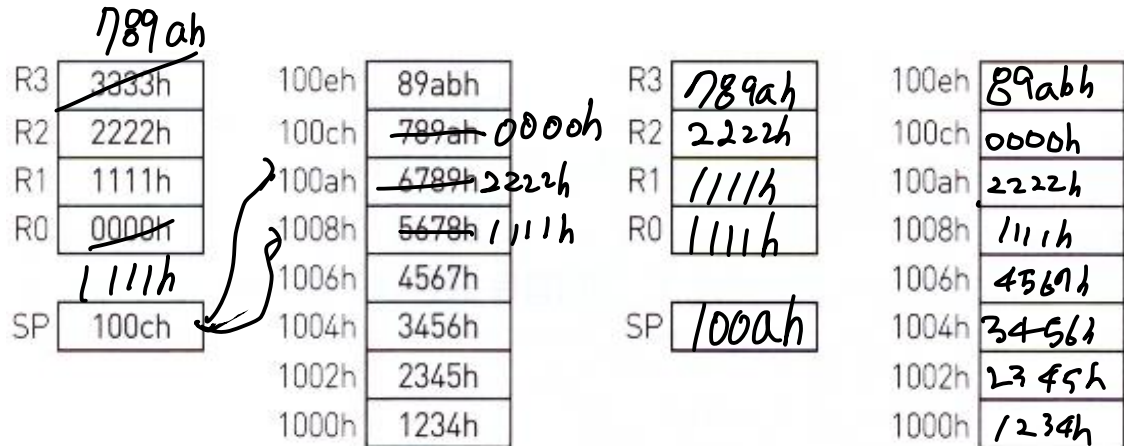
7. 컴퓨터의 상태가 아래와 같고, 스택은 다음과 같이 운영된다.

PUSH: $SP \leftarrow SP - [\text{단어의 크기}]$
 $\text{Mem}[SP] \leftarrow \text{오퍼랜드}$

POP: $\text{오퍼랜드} \leftarrow \text{Mem}[SP]$
 $SP \leftarrow SP + [\text{단어의 크기}]$

다음 명령어들이 연속적으로 실행된 후, 스택의 최종 상태를 오른쪽에 그려라.

POP R3
 PUSH R0
 PUSH R2
 PUSH R1
 POP R0



인출 → 해석 → 실행

8. 다음은 인터럽트 서비스 루틴의 마지막인 RETI 명령어를 인출한 후 레지스터와 스택의 상태이다. 이 명령어를 실행한 후 값이 변하는 레지스터의 값을 적어라. 스택은 다음과 같이 운영된다.

PUSH: $SP \leftarrow SP + [\text{단어의 크기}]$
 $\text{Mem}[SP] \leftarrow \text{오퍼랜드}$

POP: $\text{오퍼랜드} \leftarrow \text{Mem}[SP]$
 $SP \leftarrow SP - [\text{단어의 크기}]$

인터럽트 실행:

PUSH SR
 PUSH PC
 $PC \leftarrow \text{ISR}$

RETI 실행:
 POP PC
 POP SR

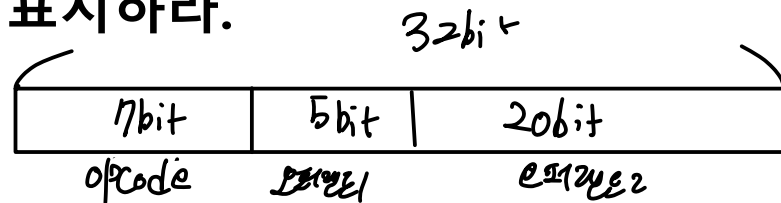
SP : 8006h
 PC : 1234h
 SR : 8042h

	1234h		
PC	08FEh
IR	RETI	800ch	4241h
R0	0000h	800ah	1234h
R1	1111h	8008h	8042h
R2	2222h	8006h	7542h
R3	3333h	8004h	4085h
SP	800ah	8002h	592ah
SR	8007h	8000h	4123h
레지스터		스택	

8042h

9. 32비트 명령을 실행하는 마이크로프로세서가 128가지의 연산들을 수행하며, 내부 레지스터의 수는 32개라고 하자. 이 CPU의 명령어 형식은 연산 코드 필드와 레지스터 번호를 나타내는 오퍼랜드1 및 나머지 비트들로 구성되는 주소 필드(오퍼랜드2)로 이루어진다.

(1) 명령어 형식을 표시하라.



(2) 이 명령어에 의해 직접 주소지정 될 수 있는 기억장치의 용량을 구하라. 단, 기억장치 주소는 바이트 단위로 지정된다고 가정한다.

$$2^{20} \times 1 \text{ byte} = 1 \text{ Mbyte}$$