

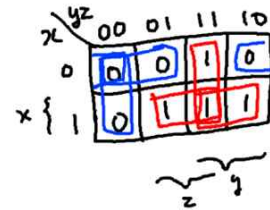
(답은 별도 배포한 답안지에 작성하시오. 별지 TABLE 5-6 참조)

1. $F(x, y, z) = \sum(3, 5, 6, 7)$ 일 때 Karnaugh map(K-map)을 이용하여 simplified Boolean expression을 유도한다. (20)

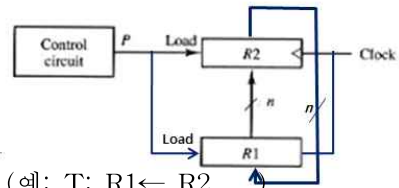
- (1) 해당 Karnaugh map을 그리시오.
- (2) sum-of-products 형태의 결과로 유도하시오.
- (3) product-of-sum 형태의 결과로 유도하시오.
- (4) 두 식이 동일한 Boolean expression임을 Boolean algebra를 이용하여 증명하시오

(1) (2) $F = xy + xz + yz$ (3) $F' = x'y' + x'z' + y'z'$
 $F = (x+y)(x+z)(y+z)$

		1	
1	1	1	1



(4) $F = (x+y)(y+z)(z+x)$
 $= (y+xz)(z+x) \quad \because x+yz = (x+y)(x+z)$
 $= xy + yz + zx$



2. 다음 물음에 답하시오. (30점)

- (1) 오른쪽 그림에 해당하는 RTL 문장을 적어보시오. (예: T: $R1 \leftarrow R2, \dots$)
 답: P : $R2 \leftarrow R1, R1 \leftarrow R2$

(2) 8-bit 내용 16진법 5A에 대해, logical shift right 한 후 rotate right를 한다.

- ① 그 결과를 16진법으로 표시하시오. 96
- ② 위 결과를 다시 arithmetic shift right 한 결과를 16진법으로 쓰시오. CB

답: ① 0101 1010 --0010 1101-- 1001 0110(96)
 ② 1100 1011 (CB)

(3) 5장의 Basic computer에서 direct와 indirect memory reference 명령어 중 무엇이 더 빠른 수행을 하는가와, 그 답에 대한 이유를 적으시오.

답: 같다. 이유: direct 경우 T_3 에서 아무것도 하지 않고 지나감

(4) Interrupt cycle에 들어가면 $IEN \leftarrow 0$ 로 하는 이유를 적으시오.

이전 프로그램의 레지스터 및 status를 save하는 동안 다른 인터럽트가 오지 못하도록 하여 프로그램 간 영키지 않게 함

3. 별첨에 복사된 교과서 Table 5-6을 참조하며 답하시오.

- (1) AR (address register)의 control logic은 다음과 같다.

DR 에 대해 control logic을 유도하시오.

$$LD(DR) = (D_0 + D_1 + D_2 + D_6)T_4$$

$$CLR(DR) =$$

$$INR(DR) = D_6T_5$$

- (2) PC 레지스터를 bus에 연결하기 위한 bus select 앞의 encoder 입력을 x_2 라 할 때, x_2 의 Boolean function을 구하시오. ($x_2 = \dots$ 형태로 표시)

$$(2) x_2 = T_0 + D_5T_4 (R'T_0 + RT_0 + D_5T_4 \text{ 경우 감점 -2})$$

4. 다음 4개 중 Mano's basic computer에서 single clock pulse 내에 수행될 수 없는 것을 모두 고르고, 고른 것을 여러 clock에 걸쳐 수행하기 위한 micro-operation sequence를 서술하시오. (주의: IR register에는 자체 increment기능(INR)이 없음) (10점)

- (1) $AC \leftarrow AC + DR$
 (2) $DR \leftarrow M[TR]$
 (3) $AC \leftarrow DR, DR \leftarrow AC$
 (4) $IR \leftarrow IR + 1$

$$(2) AR \leftarrow TR$$

$$DR \leftarrow M[AR]$$

$$(4) DR \leftarrow IR$$

$$DR \leftarrow DR + 1$$

$$IR \leftarrow DR$$

$D_2T_4: DR \leftarrow M[AR]$
 $D_2T_5: AC \leftarrow DR, DR \leftarrow AC, E \leftarrow 0$
 $D_2T_6: CIL$
 $D_2T_7: E \leftarrow 0$
 $D_2T_8: CIL$
 $D_2T_9: M[AR] \leftarrow AC, AC \leftarrow DR, SC \leftarrow 0$

5. 오른쪽과 같은 마이크로오페레이션들이 있다.

- (1) 무슨 작용을 하는 내용인지 한 줄의 symbolic description (designation) 으로 쓰시오.

(예: $AC \leftarrow AC \oplus M[AR]$ 등의 연산 형태로)

$$\text{답: } M[EA] \leftarrow M[EA] \times 4$$

(overflow 체크는 하지 않음)

- (2) 초기 값이 10진수로 $AC = 12$, $M[AR] = 8$ 이라 할때, 위의 명령어를 수행후의 각 AC , $M[AR]$ 의 값은 (십진수로)?

$$\text{답: } AC=12, M[EA] = 32$$

6. DEC라는 memory-reference instruction을 설계하려 한다. DEC는 주어진 메모리 주소에 있는 값을 1 감소시킨 다음 다시 저장된다. (단, 명령어 실행 후 본래 AC에 저장되어 있던 값이 변경되어서는 안 된다. op-code는 3-bit 110을 쓰고, T_4 부터 기술하시오.) (10)

$$D_6T_4: TR \leftarrow AC$$

$$D_6T_5: DR \leftarrow M[AR], AC \leftarrow 0$$

$$D_6T_6: AC \leftarrow AC$$

$$D_6T_7: AC \leftarrow \overline{AC} + DR$$

D₆T₈: M[AR] ← AC
D₆T₉: DR ← TR
D₆T₁₀: AC ← DR, SC ← 0

또는

D₆T₄: DR ← M[AR]
D₆T₅: AC ← DR, DR ← AC
D₆T₆: AC ← \overline{AC}
D₆T₇: AC ← AC + 1
D₆T₈: AC ← \overline{AC}
D₆T₉: M[AR] ← AC, AC ← DR, SC ← 0

7. Mano's Basic computer의 메모리에 다음과 같이 instruction이 저장되어 있다. 현재 PC의 값은 2이다. 표 안의 모든 숫자는 16진수(hexadecimal)이다.

- (1) 프로그램이 종료될 때까지 매 T₀일 때의 PC레지스터에서의 값을 모두 기술하시오.
PC값의 변화: 2 (CLA) → 3 (0 BSA 006) → 7 (0 ADD 006) → 8 (1 BUN 006) → 4 (0 STA 00A) → 5 (HLT)
(2) 프로그램 종료 후 메모리주소 A에 저장된 값을 기술하시오.
메모리 주소 A에 저장된 값: 0004

Location	Instruction
0	0000
1	4010
2	7800 CLA
3	5006 BSA 6
4	300A STA A
5	7001 HALT
6	0005 -
7	1006 ADD 6
8	C006 BUN 6 I
9	7001 HALT
A	7002
...	...

8. 오른쪽 그림과 같은 프로그램이 있다.

- (1) 첫 번째 인수와 두 번째 인수를 더하여 세 번째 인수에 결과를 쓰는 subprogram SUM를 작성하시오.
(2) 첫 번째 인수와 두 번째 인수를 주소로 보고, 두 주소가 가르키는 곳의 값을 더해서 세 번째 인수에 결과를 쓰는 subprogram SUM를 작성하시오.

7. (1)
SUM, HEX 0
LDA SUM I
ISZ SUM
ADD SUM I
ISZ SUM
STA SUM I
ISZ SUM

<u>calling program</u> BSA SUM HEX 100 HEX 200 HEX 0
--

```

        BUN SUM I
또는
    SUM, HEX 0
        LDA SUM I
        STA FST          변수 이름은 임의가능
        ISZ SUM
        LDA SUM I
        STA SCD          변수 이름은 임의가능
        ISZ SUM
        LDA FST
        ADD SCD
        STA SUM I
        ISZ SUM
        BUN SUM I
FST, HEX 0      (또는 FST, DEC 0)(또는 FST, ____ ) 변수 이름은 임의가능
SCD, HEX 0      (또는 SCD, DEC 0)(또는 FST, ____ ) 변수 이름은 임의가능

```

```

(2) SUM, HEX 0
    LDA SUM I
    STA FST          변수 이름은 임의가능
    ISZ SUM
    LDA SUM I
    STA SCD          변수 이름은 임의가능
    ISZ SUM
    LDA FST I
    ADD SCD I
    STA SUM I
    ISZ SUM
    BUN SUM I
FST, HEX 0      (또는 FST, DEC 0)(또는 FST, ____ )
SCD, HEX 0      (또는 SCD, DEC 0)(또는 FST, ____ )

```

학부

학번

이름

1.

(1)