

명령어 사이클

$$D_7 I' T_3 = r : SC \leftarrow 0$$

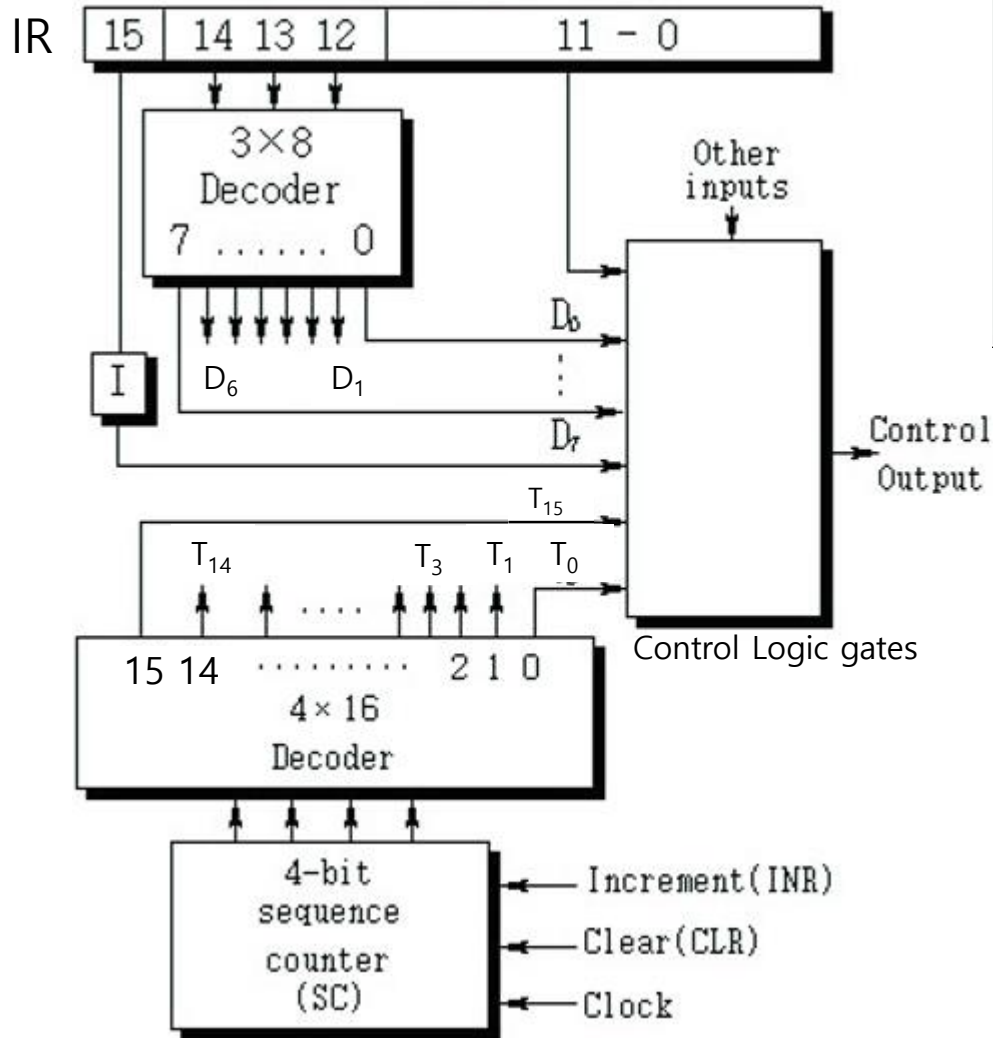
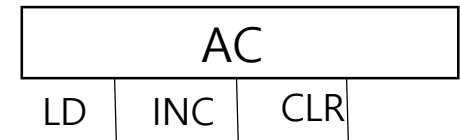
$$CLA \quad rB_{11} : AC \leftarrow 0$$


표5-2 컴퓨터 명령어

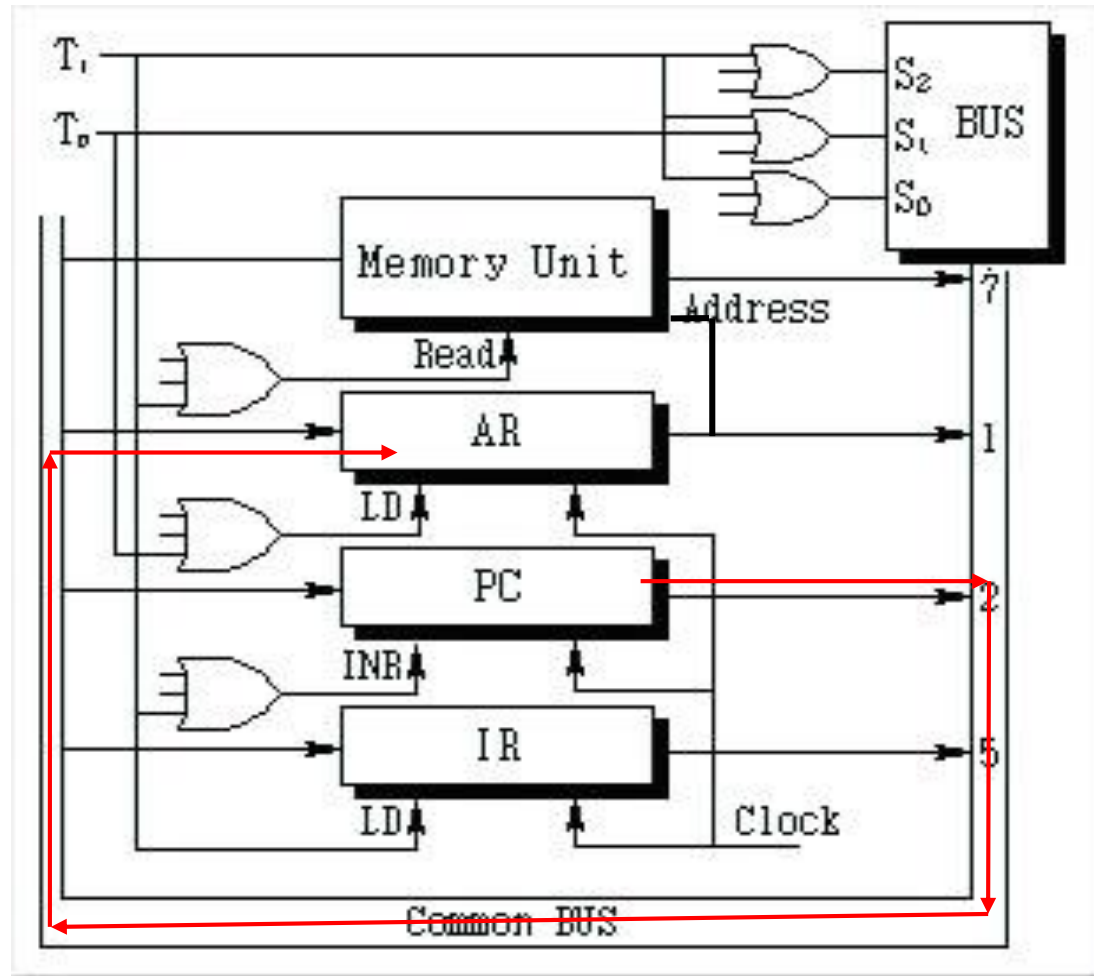
| Symbol | Hexadecimal code | | Description |
|--------|------------------|-------|--------------------------------------|
| | I = 0 | I = 1 | |
| AND | 0XXX | 8XXX | AND memory word to AC |
| ADD | 1XXX | 9XXX | Add memory word to AC |
| LDA | 2XXX | AXXX | Load memory word to AC |
| STA | 3XXX | BXXX | Store content of AC in memory |
| BUN | 4XXX | CXXX | Branch unconditionally |
| BSA | 5XXX | DXXX | Branch and save return address |
| ISZ | 6XXX | EXXX | Increment and skip if zero |
| CLA | 7800 | | Clear AC |
| CLE | 7400 | | Clear E |
| CMA | 7200 | | Complement AC |
| CME | 7100 | | Complement E |
| CIR | 7080 | | Circulate right AC and E |
| CIL | 7400 | | Circulate left AC and E |
| INC | 7020 | | Increment AC |
| SPA | 7010 | | Skip next instruction if AC positive |
| SNA | 7008 | | Skip next instruction if AC negative |
| SZA | 7004 | | Skip next instruction if AC zero |
| SZE | 7002 | | Skip next instruction if E 0 |
| HLT | 7001 | | Halt computer |
| INP | F800 | | Input character to AC |
| OUT | F400 | | Output character from AC |
| SKI | F200 | | Skip on input flag |
| SKO | F100 | | Skip on output flag |
| ION | F080 | | Interrupt on |
| IOF | F040 | | Interrupt off |



5.5 명령어 사이클(instruction cycle)

- 명령어를 메모리에서 가져온다. (fetch)
- 명령어를 decoding 한다.(decode)
- 메모리로부터 유효주소를 읽어온다.
- 명령어를 실행한다.(execute)

Fetch and decode



T_0 cycle

1. $S_2S_1S_0 = 010$ 으로 하여 PC 내용이 버스에 놓이도록 한다.
2. AR의 LD 입력을 인에이블

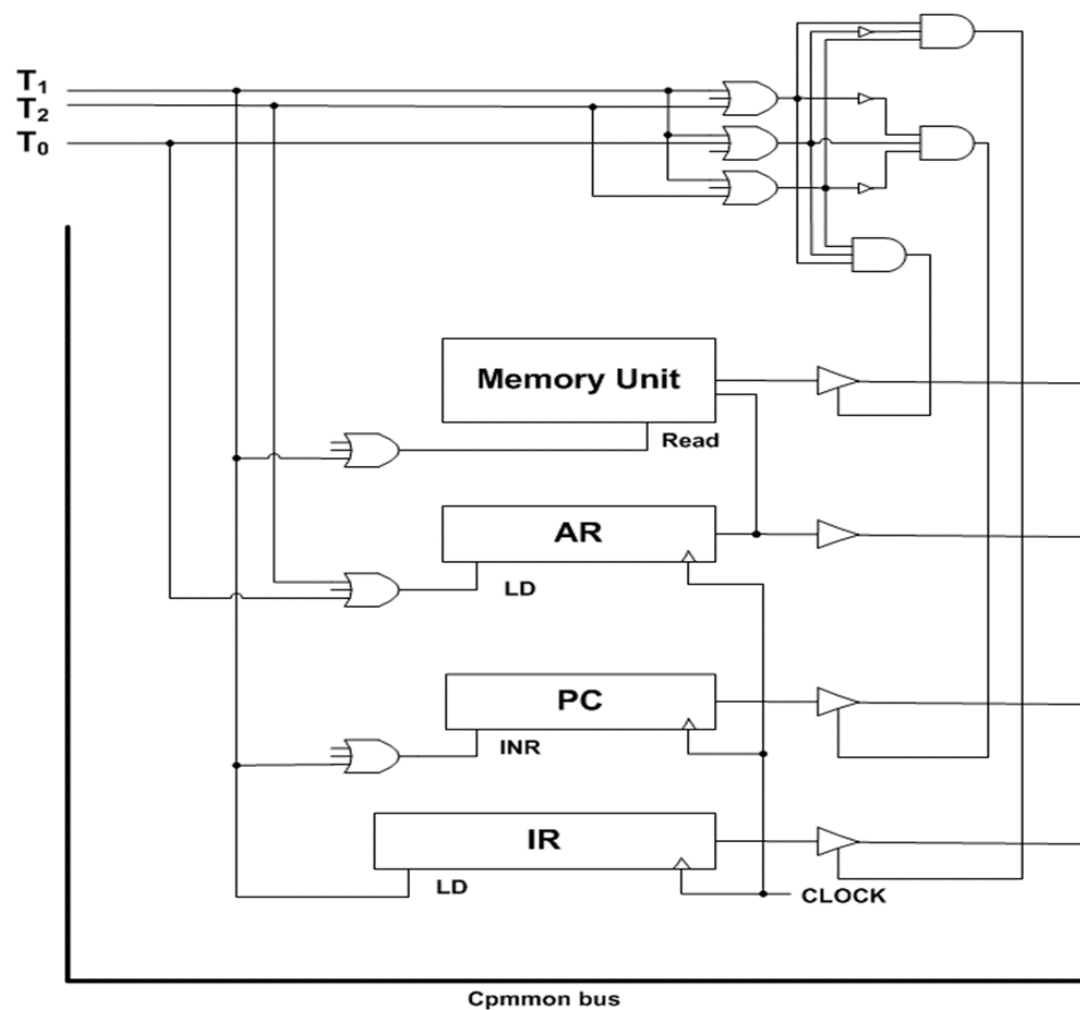
T_1 cycle

1. 메모리 읽기 입력을 인에이블
2. 메모리의 내용을 버스로
3. IR의 LD를 인에이블
4. PC의 INR 입력을 인에이블

$T_0: AR \leftarrow PC$

$T_1: IR \leftarrow M[AR], PC \leftarrow PC + 1$

$T_2: D_0, \dots, D_7 \leftarrow \text{Decode } IR(12-14), AR \leftarrow IR(0-11), I \leftarrow IR(15)$



$T_0: AR \leftarrow PC$
 $T_1: IR \leftarrow M[AR], PC \leftarrow PC + 1$
 $T_2: D_0, \dots, D_7 \leftarrow \text{Decode } IR(12-14), AR \leftarrow IR(0-11), I \leftarrow IR(15)$

명령어 종류의 결정

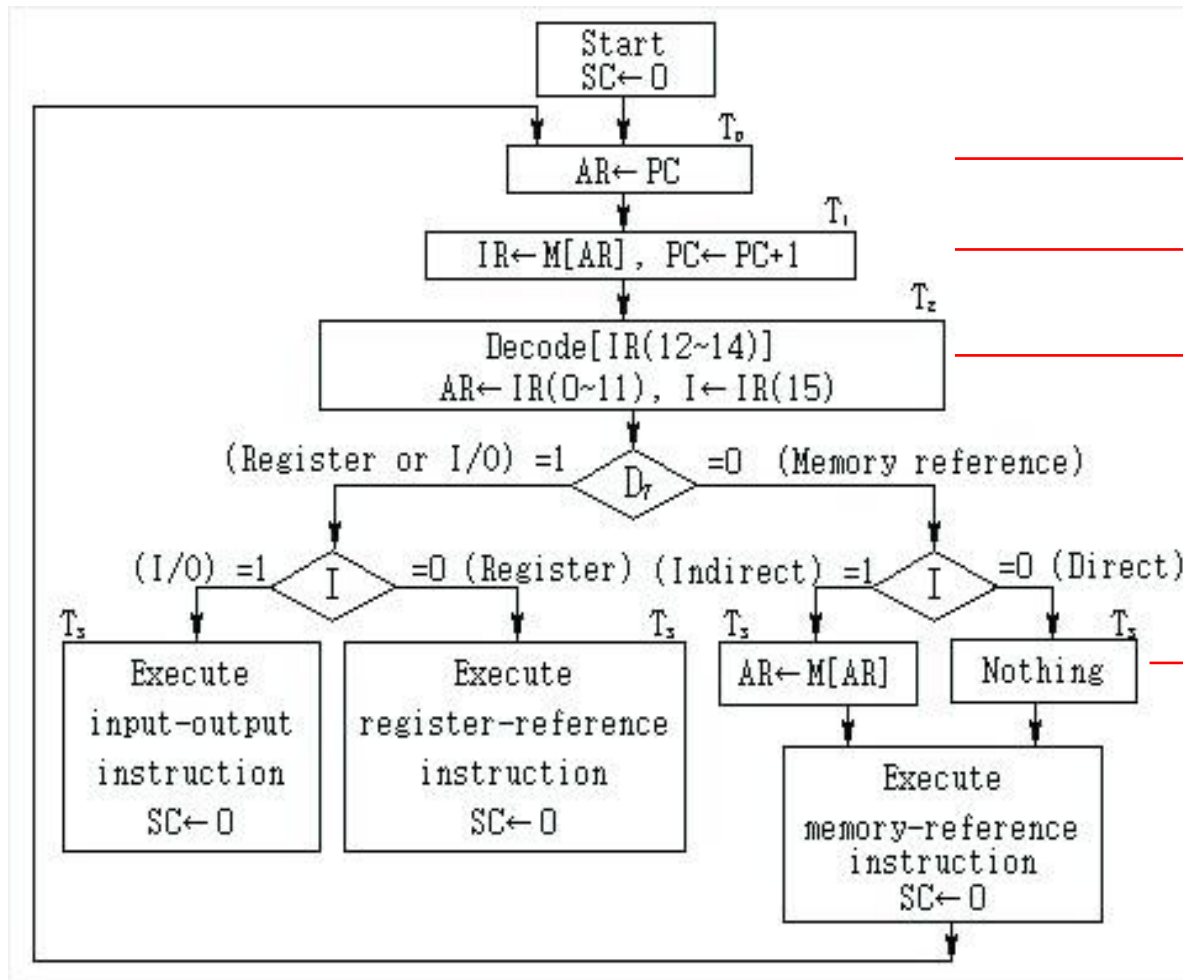
- T_3 : 제어 장치는 명령어의 종류를 결정

$D_7' I T_3 : AR \leftarrow M[AR]$

$D_7' I' T_3 : \text{아무런 일도 하지 않음}$

$D_7 I' T_3 : \text{레지스터 참조 명령어를 실행}$

$D_7 I T_3 : \text{입출력 명령어를 실행}$



T_0

T_1

T_2

T_3

직접주소지정방식 명령어와
간접주소지정방식 명령어의
수행시간 동기화를 위해
직접주소지정방식 명령어는
 T_3 에 아무일도 않함

레지스터 참조 명령어

표5-3 레지스터 참조 명령어의 수행

$D_7I'T_3=r$ (Common to all register-reference instruction)

$IR(i)=B_i$ [bit in $IR(0\sim11)$ that specifies the operation]

| | | | |
|-----|-------------|--|--------------------|
| | r : | $SC \leftarrow 0$ | Clear SC |
| CLA | rB_{11} : | $AC \leftarrow 0$ | Clear AC |
| CLE | rB_{10} : | $E \leftarrow 0$ | Clear E |
| CMA | rB_9 : | $AC \leftarrow AC'$ | Complement AC |
| CME | rB_8 : | $E \leftarrow E'$ | Complement E |
| CIR | rB_7 : | $AC \leftarrow shr\ AC, AC(15) \leftarrow E, EA \leftarrow C(0)$ | Circulate right AC |
| CIL | rB_6 : | $AC \leftarrow shr\ AC, AC(15) \leftarrow E, EA \leftarrow C(0)$ | Circulate left AC |
| INC | rB_5 : | $AC \leftarrow AC+1$ | Increment AC |
| SPA | rB_4 : | If $(AC(15)=0)$ then $(PC \leftarrow PC+1)$ | Skip if positive |
| SNA | rB_3 : | If $(AC(15)=1)$ then $(PC \leftarrow PC+1)$ | Skip if negative |
| SZA | rB_2 : | If $(AC=0)$ then $(PC \leftarrow PC+1)$ | Skip if AC zero |
| SZE | rB_1 : | If $(E=0)$ then $(PC \leftarrow PC+1)$ | Skip if positive |
| HLT | rB_0 : | $S \leftarrow 0$ (S is start-stop flip-flop) | Skip if positive |

수고하셨습니다!