메모리 참조 명령어

입출력 명령어

명령어 종류의 결정

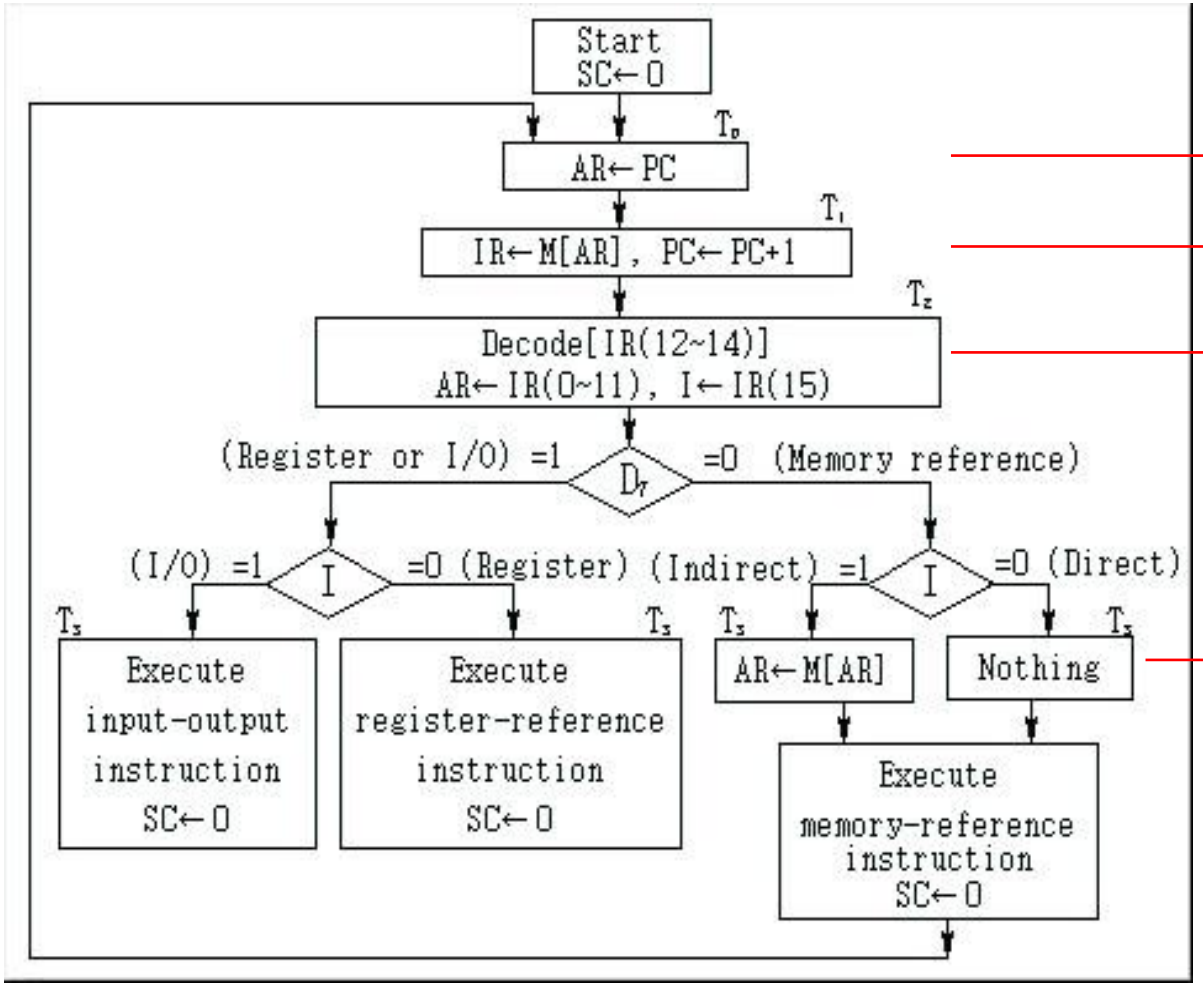
- T_3 : 제어 장치는 명령어의 종류를 결정

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

$D_7' I'\ T_3$: 아무런 일도 하지 않음

$D_7 I'\ T_3$: 레지스터 참조 명령어를 실행

$D_7 I\ T_3$: 입출력 명령어를 실행



T_0

T_1

T_2

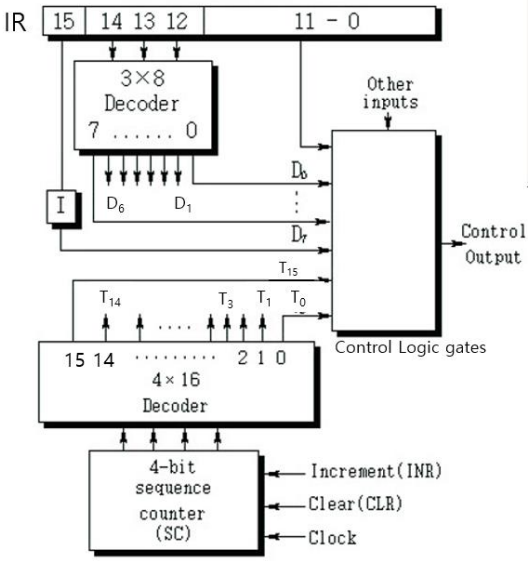
T_3

직접주소지정방식 명령어와 간접주소지정방식 명령어의 수행시간 동기화를 위해 직접주소지정방식 명령어는 T_3에 아무일도 않함

5.6 메모리 참조 명령어

- 각 명령은 D_i (i=0,1,2,3,4,5,6)에 의해 구별
- 피연산자에 대한 유효 주소는 T_2(I=0일 때)나 T_3(I=1일 때) AR 레지스터로 전송
- 명령의 수행은 T_4 에서 시작
- 레지스터 전송문을 이용한 7개의 메모리 참조 명령어

Symbol	Operation Decoder	Symbolic description
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR], E \leftarrow C_{out}$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC, PC \leftarrow AR+1$
ISZ	D_6	$M[AR] \leftarrow M[AR]+1,$ $If\ M[AR]+1 = 0\ then\ PC \leftarrow PC+1$



AND 명령어

$D_0T_4: DR \leftarrow M[AR]$
$D_0T_5: AC \leftarrow AC \wedge DR, \quad SC \leftarrow 0$

ADD 명령어

$D_1T_4: DR \leftarrow M[AR]$
$D_1T_5: AC \leftarrow AC + DR, \quad E \leftarrow C_{out}, \quad SC \leftarrow 0$

LDA 명령어

$D_2T_4: DR \leftarrow M[AR]$
$D_2T_5: AC \leftarrow DR, \quad SC \leftarrow 0$

M[AR]의 내용을 직접 AC로 가져오지 못하는 이유?

STA 명령어

$D_3T_4: M[AR] \leftarrow AC, \quad SC \leftarrow 0$

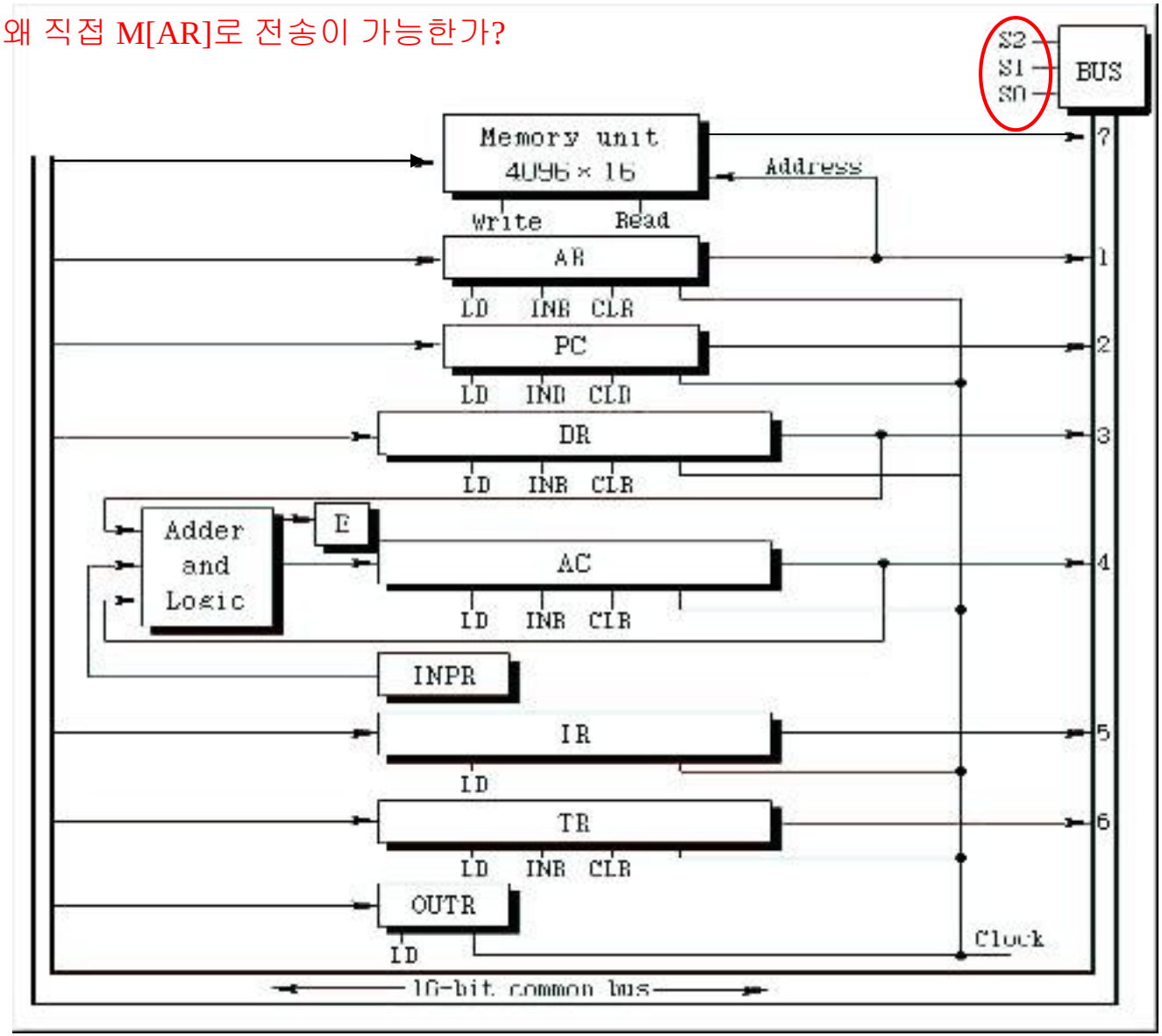
AC의 내용은 왜 직접 M[AR]로 전송이 가능한가?

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

Common bus system

M[AR]의 내용을 직접 AC로 가져오지 못하는 이유?

AC의 내용은 왜 직접 M[AR]로 전송이 가능한가?



BUN(Branch Unconditionary) 명령어

$$D_4T_4: PC \leftarrow AR, \quad SC \leftarrow 0$$

BSA(Branch and save return address) 명령어

$$D_5T_4: M[AR] \leftarrow PC, \quad AR \leftarrow AR+1$$
$$D_5T_5: PC \leftarrow AR, \quad SC \leftarrow 0$$

ISZ(Increment and Skip if zero) 명령어

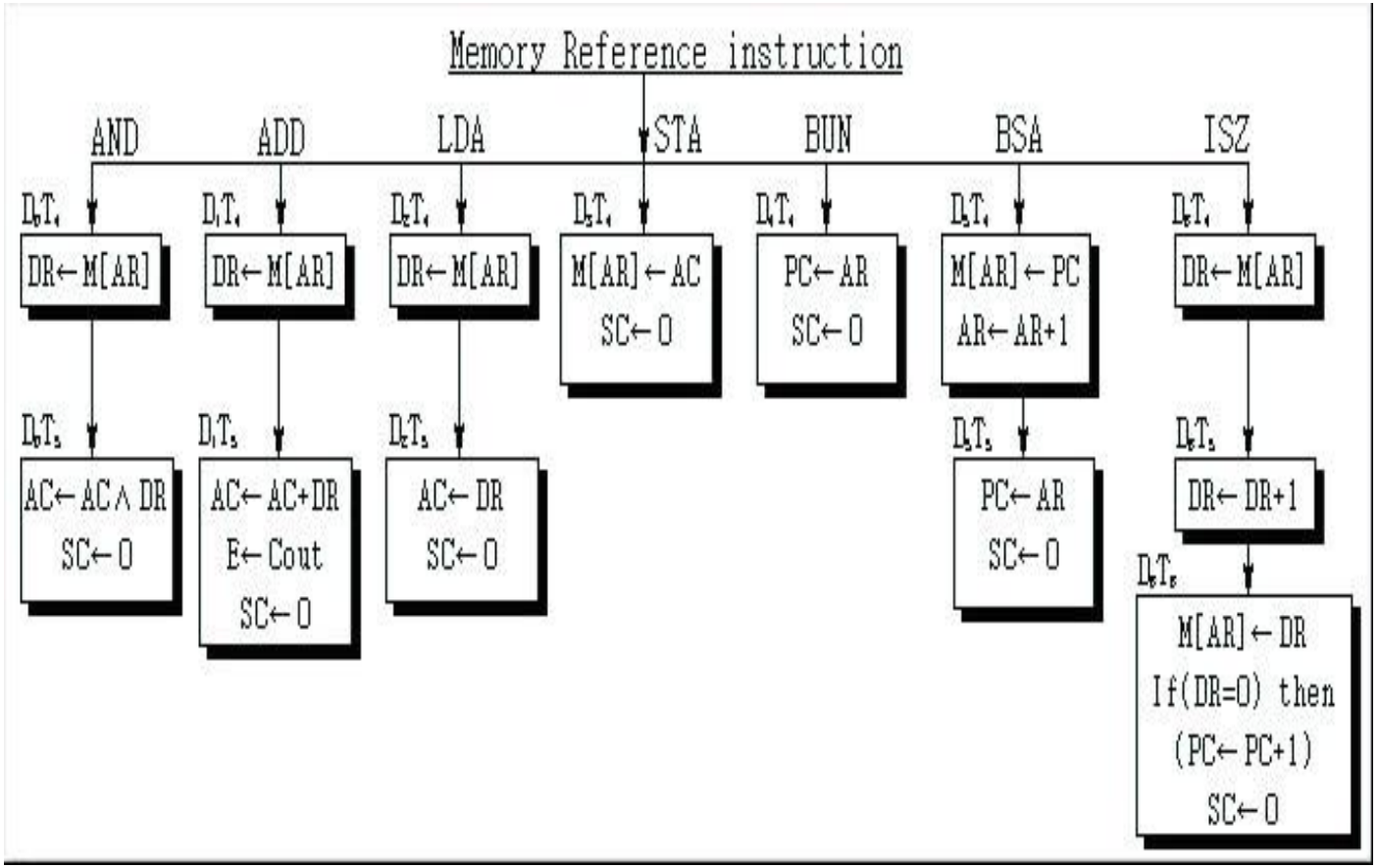
이 명령은 어떤 레지스터의 값을 증가시키는가?

$$D_6T_4: \quad DR \leftarrow M[AR]$$
$$D_6T_5: \quad DR \leftarrow DR+1$$
$$D_6T_6: \quad M[AR] \leftarrow DR, \ \text{if}\,(DR=0)\ then\,(PC \leftarrow PC+1), \quad SC \leftarrow 0$$

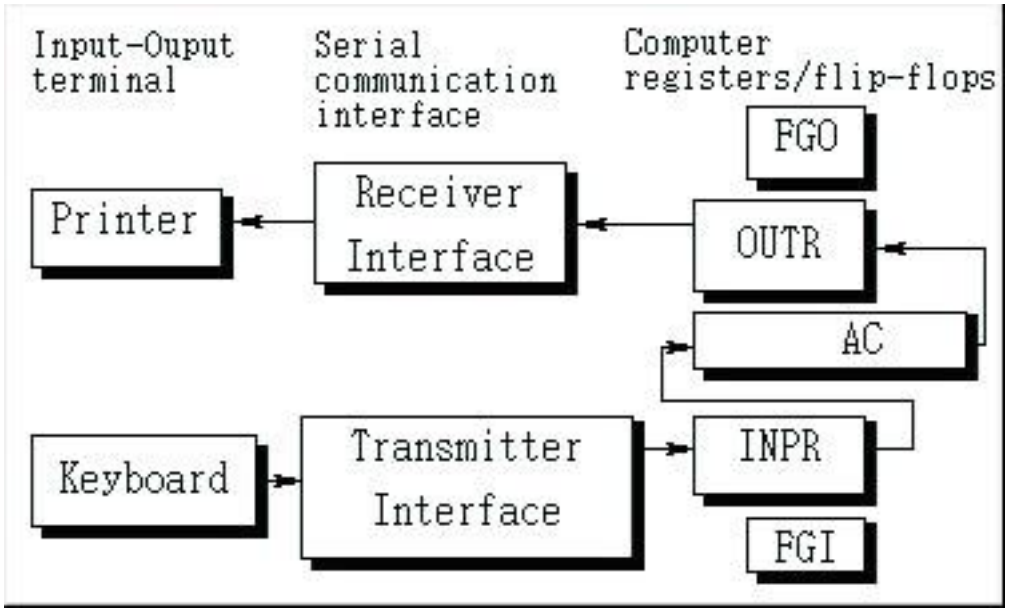
$$T_0: AR \leftarrow PC$$
$$T_1: IR \leftarrow M[AR], \quad PC \leftarrow PC+1$$
$$T_2: D_0, \cdots, D_7 \leftarrow Decode\ IR(12-14), \quad AR \leftarrow IR(0-11), \quad I \leftarrow IR(15)$$

제어흐름도

- 7가지 메모리 참조 명령어의 마이크로 연산 정리
- 가장 수행시간이 걸리는 명령어 ISZ은 T0~T6 까지 7개의 타이밍 클럭 필요
- 3비트 순차 카운터로 타이밍 시이퀀스 카운터 설계 가능



5.7 입출력과 인터럽트



FGI(FLAG Input) : 새로운 정보가 입력 장치에서 들어오면 **set** 되고 **INPR**에 들어온 정보가 **AC**로 전달되면 클리어 된다. **FGI**가 **1**인 동안은 다른 키를 쳐서 **INPR**의 정보를 바꿀 수 없다.

FGO(FLAG Out) : FGO가 **1**이면 **AC**로 부터 **OUTR**로 정보를 병렬전송 전송이 끝나면 **FGO**는 클리어되고 이때 출력장치는 **OUTR**로 부터 정보를 가져간다. 정보를 가져가면 다시 **FGO**는 클리어 된다.

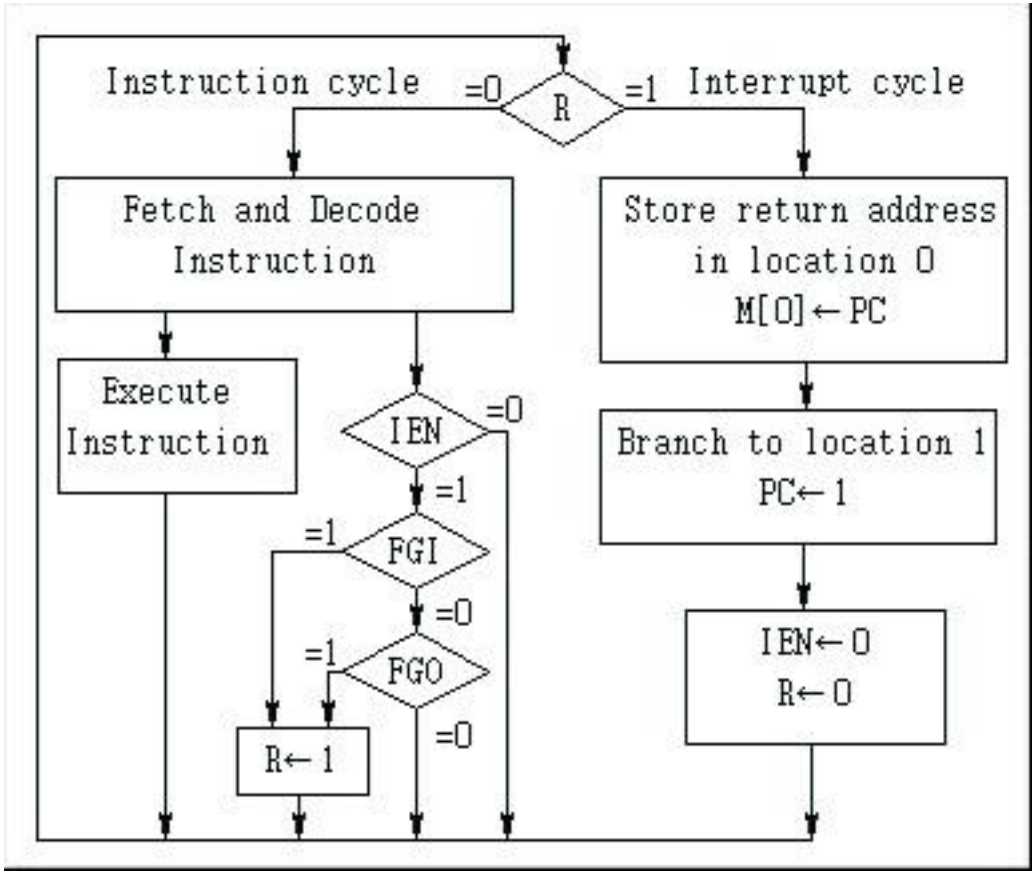
입출력 명령어

D_7IT_3=p(Common to all input-output instruction)

IR(i)=B_i [bit in IR(6~11) that specifies the instruction]

	P:	SC ← 0	Clear SC
INP	pB_{11}:	AC(0~7)←INPR, FGI←0	Input character
OUT	pB_{10}:	OUTR←AC(0~7), FGO←0	Output character
SKI	pB_9:	If(FGI=1) then (PC←PC+1)	Skip on input flag
SKO	pB_8:	If(FGO=1) then (PC←PC+1)	Skip on output flag
ION	pB_7:	IEN←1	Interrupt enable on
IOF	pB_6:	IEN←0	Interrupt enable off

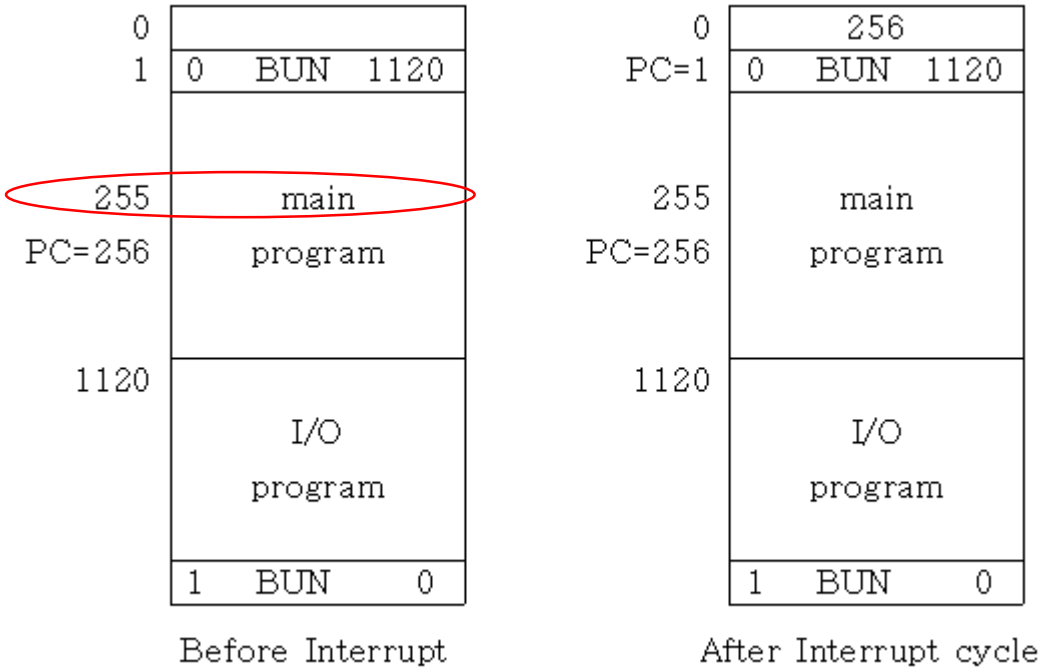
프로그램 인터럽트



R : 인터럽트 **Flip-flop**

앞서의 플래그를 검사하여 통신하는 방법을 프로그램 제어 전송이라고 하는데 이러한 방법은 일정 시간간격으로 플래그를 검사하여야 하기 때문에 비 효율적이다. 따라서 프로그램 수행중에는 플래그를 검사하지 않고 플래그가 **set**되면 자동으로 현재의 수행 프로그램을 종료하고 플래그가 **set**된 정보를 받아 입출력을 실행하는 방법 : 인터럽트

인터럽트 사이클의 수행 과정



인터럽트 서비스 루틴 (ISR)

메모리의 주소 255에 있는 명령이 수행될 때, R이 1이고 인터럽트가 발생했다고 가정
PC의 값 256이 메모리 주소 0에 저장되고 PC는 1로 세트되고, R은 클리어된다. 다음 명령 사이클은 주소 1에 저장되어 있는 명령을 가지고 수행됨으로 1120으로 분기함

인터럽트 플립플롭 R은 IEN=1이고 FGO 또는 FGI가 1일 때 1로 set된다. 이것은 타이밍 신호 T_0, T_1, T_2외의 어떠한 시간에도 수행되며, 레지스터 전송문으로 나타내면 다음과 같다.

$$T_0' T_1' T_2' (IEN)(FGI + FGO) : R \leftarrow 1$$

인터럽트 플립플롭 $R = 0$일 때만 명령 사이클이 수행되도록 수정

$R' T_0 : AR \leftarrow PC$
$R' T_1 : IR \leftarrow M[AR], PC \leftarrow PC + 1$
$R' T_2 : D_0, \cdots, D_7 \leftarrow Decode\ IR(12-14),\ AR <- IR(0-11), I <- IR(15)$

$R = 1$일 때 인터럽트 사이클이 수행되어 복귀주소를 메모리 주소 0에 저장하고 프로그램 수행을 주소1로 분기한 후 IEN, R, SC를 0으로 클리어

$RT_0 : AR \leftarrow 0, TR \leftarrow PC$
$RT_1 : M[AR] \leftarrow TR, PC \leftarrow 0$
$RT_2 : PC \leftarrow PC + 1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$

PC를 임시레지스터(TR)에 옮기고 T_1 사이클에 M[0]로 저장, 현재 인터럽트 상태이므로 다음 인터럽트를 받기 위해 IEN을 0으로 돌려놓는다 또한 명령 사이클을 받을 수 있도록 R은 0으로 환원

제어함수

Fetch	$R'\,T_0:$	$AR \leftarrow PC$
	$R'\,T_1:$	$IR \leftarrow M[AR],\ PC \leftarrow PC+1$
Decode	$R'\,T_2:$	$D_0,\ \cdots, D_7 \leftarrow Decode\ IR(12-14),$ $AR \leftarrow IR(0-11),\ I \leftarrow IR(15)$
Indirect	$D_7'\,I\,T_3:$	$AR \leftarrow M[AR]$
Interrupt		
$T_0'\,T_1'\,T_2'\,(IEN)\,(FGI+FGO):$		$R \leftarrow 1$
	$RT_0:$	$AR \leftarrow 0,\ TR \leftarrow PC$
	$RT_1:$	$M[AR] \leftarrow TR,\ PC \leftarrow 0$
	$RT_2:$	$PC \leftarrow PC+1,\ IEN \leftarrow 0,\ R \leftarrow 0,\ SC \leftarrow 0$
Memory-reference:		
AND	$D_0\,T_4:$	$DR \leftarrow M[AR]$
	$D_0\,T_5:$	$AC \leftarrow AC \wedge DR,\ SC \leftarrow 0$
ADD	$D_1\,T_4:$	$DR \leftarrow M[AR]$
	$D_1\,T_5:$	$AC \leftarrow AC+DR,\ E \leftarrow C_{out},\ SC \leftarrow 0$
LDA	$D_2\,T_4:$	$DR \leftarrow M[AR]$
	$D_2\,T_5:$	$AC \leftarrow DR,\ SC \leftarrow 0$
STA	$D_3\,T_4:$	$M[AR] \leftarrow AC,\ SC \leftarrow 0$
BUN	$D_4\,T_4:$	$PC \leftarrow AR,\ SC \leftarrow 0$
BSA	$D_5\,T_4:$	$M[AR] \leftarrow PC,\ AR \leftarrow AR+1$
	$D_5\,T_5:$	$PC \leftarrow AR,\ SC \leftarrow 0$
ISZ	$D_6\,T_4:$	$DR \leftarrow M[AR]$
	$D_6\,T_5:$	$DR \leftarrow DR+1$
	$D_6\,T_6:$	$M[AR] \leftarrow DR,\ \text{if}(DR=0)\,then\ (PC \leftarrow PC+1),\ SC \leftarrow 0$

Register-reference:		
	$D_7 I' T_3 = r$	(common to all register-reference instruction)
	$IR(i) = B_i$	$(i = 0,1,2,\cdots,11)$
	r:	$SC \leftarrow 0$
CLA	rB_{11}:	$AC \leftarrow 0$
CLE	rB_{10}:	$E \leftarrow 0$
CMA	rB_9:	$AC \leftarrow \overline{AC}$
CME	rB_8:	$E \leftarrow \overline{E}$
CIR	rB_7:	$AC \leftarrow shr\, AC, AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	rB_6:	$AC \leftarrow shl\, AC, AC(0) \leftarrow E, E \leftarrow AC(15)$
INC	rB_5:	$AC \leftarrow AC + 1$
SPA	rB_4:	if $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$
SNA	rB_3:	if $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$
SZA	rB_2:	if $(AC = 0)$ then $(PC \leftarrow PC + 1)$
SZE	rB_1:	if $(E = 0)$ then $(PC \leftarrow PC + 1)$
HLT	rB_0:	$S \leftarrow 0$ (S is a start-stop flip-flop)
Input-output:		
	$D_7 I T_3 = p$	(common to all input-output instruction)
	$IR(i) = B_i$	$(i = 6,7,8,9,10,11)$
	p:	$SC \leftarrow 0$
INP	pB_{11}:	$AC(0-7) \leftarrow INPR, FGI \leftarrow 0$
OUT	pB_{10}:	$OUTR \leftarrow AC(0-7), FGO \leftarrow 0$
SKI	pB_9:	$If\,(FGI = 1)$ then $(PC \leftarrow PC + 1)$
SKO	pB_8:	$If\,(FGO = 1)$ then $(PC \leftarrow PC + 1)$
ION	pB_7:	$IEN \leftarrow 1$
IOF	pB_6:	$IEN \leftarrow 0$

수고하셨습니다!