Instruction Cycle

Instruction Cycle

- 1) Instruction Fetch from Memory
- 2) Instruction Decode
- 3) Read Effective Address(if indirect addressing mode)
- 4) Instruction Execution
- 5) Go to step 1) : Next Instruction[PC + 1]

Instruction Fetch : T0, T1

- T0 = 1
  - » 1) Place the content of PC onto the bus by making the bus selection inputs $S_2S_1S_0$=010
  - » 2) Transfer the content of the bus to AR by enabling the LD input of AR

$$T_0 : AR \leftarrow PC$$
$$T_1 : IR \leftarrow M[AR],\ PC \leftarrow PC + 1$$

$$T_0 : AR \leftarrow PC$$

- T1 = 1 $T_1 : IR \leftarrow M[AR],\ PC \leftarrow PC + 1$
  - » 1) Enable the read input memory
  - » 2) Place the content of memory onto the bus by making $S_2S_1S_0$= 111
  - » 3) Transfer the content of the bus to IR by enable the LD input of IR
  - » 4) Increment PC by enabling the INR input of PC

◆ Instruction Decode : T2

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

Op.code Address Di/Indirect

- **IR(12-14)에 따라 *Fig. 5-6* 에서 D0 - D7 출력**

◆ Instruction Execution : T3, T4, T5, T6

$IR(12-14) = 111$

D_7=1 { Register(I=0) → $D_7I'T_3$(Execute)
I/O (I=1) → D_7IT_3 (Execute)

D_7=0 : Memory Ref. { Indirect(I=1) → $D_7'IT_3$($AR \leftarrow M[AR]$)
Direct (I=0) → nothing in T_3

Read effective Address

- **Register 와 I/O 명령은 T3에서 실행되며 Memory Ref. 명령은 T3에서 Operand의 effective address를 읽음**
- ***Memory Ref. 명령은 종류에 따라 T4, T5, T6을 갖음*** :

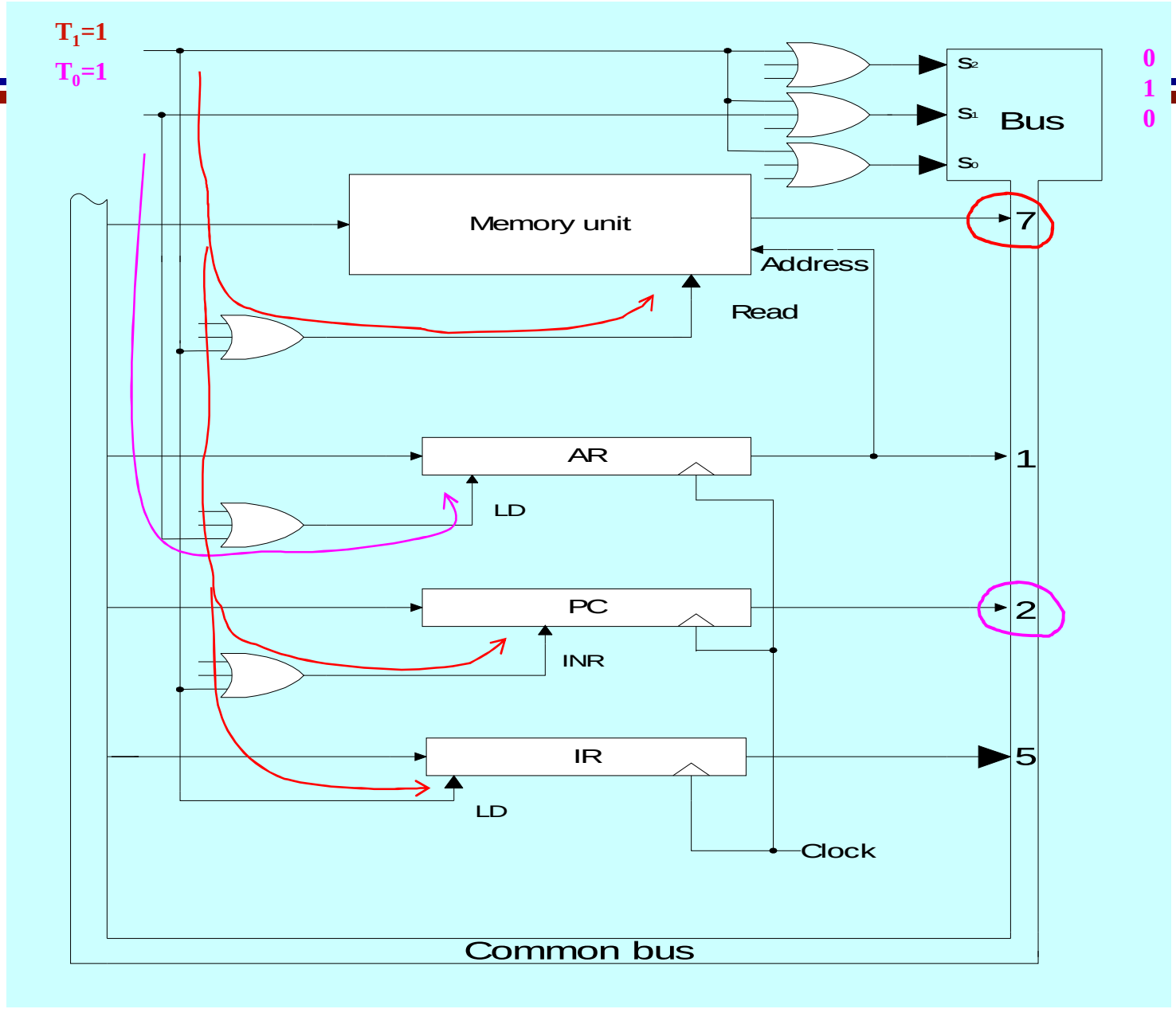
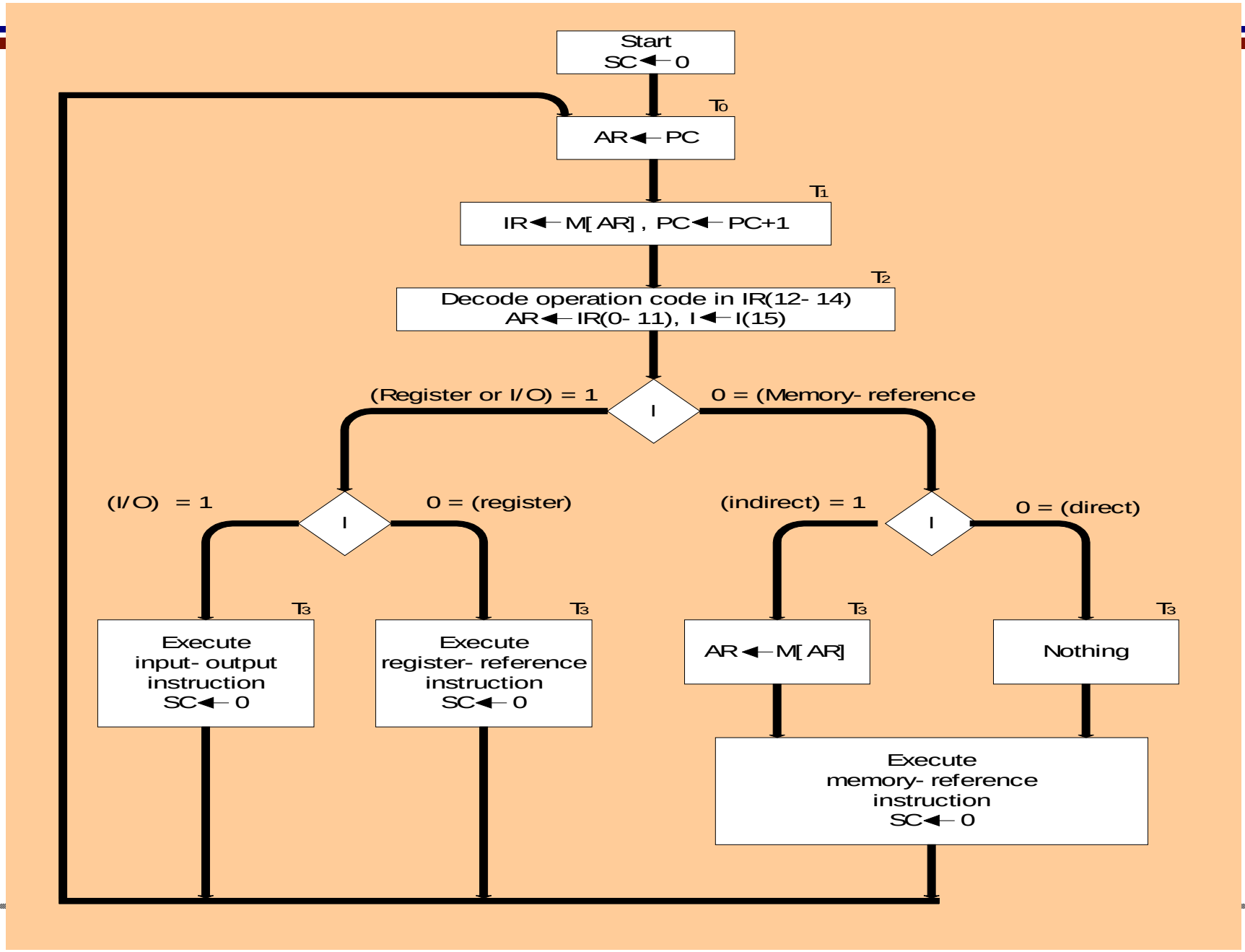

T1=1
T0=1
S2
S1
S0
Bus
0
1
1
0
Memory unit
7
Address
Read
AR
1
LD
PC
2
INR
IR
5
LD
Clock
Common bus

Fig. Flowchart for instruction cycle(initial)



- Register Ref. Instruction
  - r = $D_7 I' T_3$: 공통항
  - IR(i) = B_i ← IR(0 -11)

Address 로 사용되지 않음

  - ***B_0 - B_{11}*** : **12 개의 Register Ref. Instruction (*Tab. 5-3)***

5-6 Memory Ref. Instruction

D_7 : Register or I/O = 1

IR(12,13,14) = 111

D_6 - D_0 : **7 개의 Memory Ref. Instruction*(Tab. 5-4)***

- AND to AC

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

- ADD to AC

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

- LDA : memory read

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

- STA : memory write

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

- BUN : branch unconditionally

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

- BSA : branch and save return address

$D_5T_4 : M[AR] \leftarrow PC, AR \leftarrow AR + 1$

$D_5T_5 : PC \leftarrow AR, SC \leftarrow 0$

  - Return Address : save return address (135 ← 21)
  - Subroutine Call : ***Fig. 5-10***

$D_5T_4 : M[135] \leftarrow 21(PC), 136(AR) \leftarrow 135 + 1$

$D_5T_5 : 136(PC) \leftarrow 136(AR), SC \leftarrow 0$

- ISZ : increment and skip if zero

$D_6T_4 : DR \leftarrow M[AR]$

$D_6T_5 : DR \leftarrow DR + 1$

$D_6T_6 : M[AR] \leftarrow DR, if\ (DR = 0)\ then\ (PC \leftarrow PC + 1), SC \leftarrow 0$

- Control Flowchart : ***Fig. 5-11***
  - Flowchart for the 7 memory reference instruction
    - » The longest instruction : ISZ(T6)
    - » 따라서 3 bit Sequence Counter로 구현가능(현재 4 비트는 확장에 대비함)

Fig. 5-10 Example of BSA