Computer Architecture Lec 5a

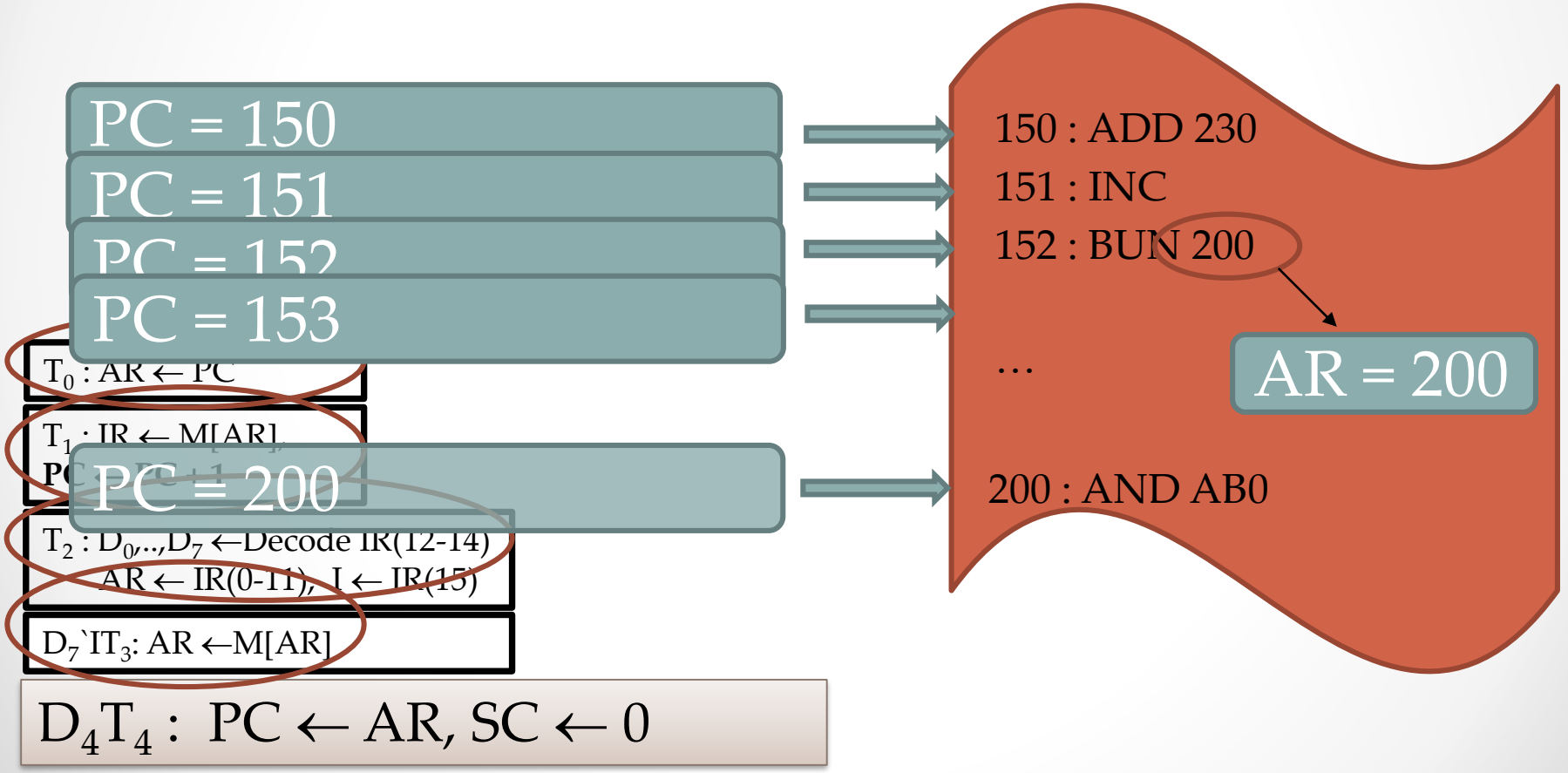
Dr. Esti Stein

(Partly taken from Dr. Alon Schclar slides)

Based on slides by:
Prof. Myung-Eui Lee
Korea University of Technology & Education
Department of Information & Communication

Taken from: **M. Mano/Computer Design and Architecture 3rd Ed.**

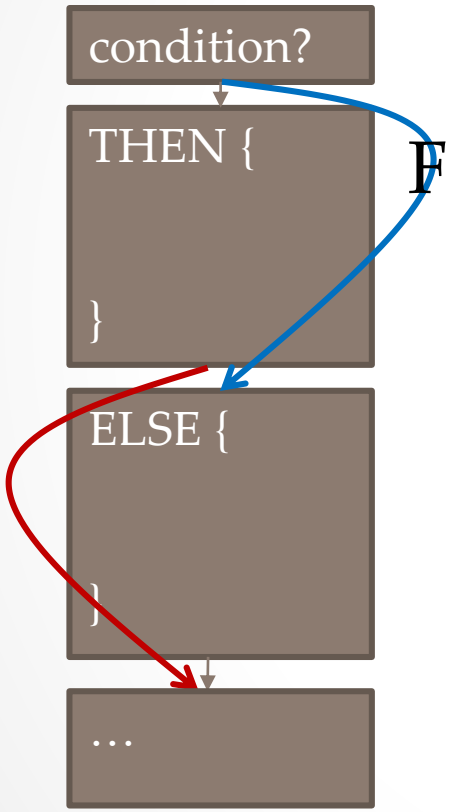
MRI - BUN


PC = 150
PC = 151
PC = 153
PC = 200
T_0 : AR ← PC
T_1 : IR ← M[AR]
T_2 : D_0,..,D_7 ←Decode IR(12-14)
AR ← IR(0-11), I ← IR(15)
D_7`IT_3: AR ←M[AR]
D_4T_4 : PC ← AR, SC ← 0
150 : ADD 230
151 : INC
152 : BUN 200
…
AR = 200
200 : AND AB0

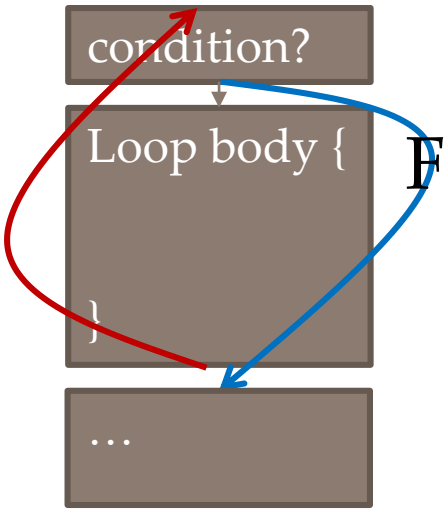
Conditional & Unconditional Jumps

conditional unconditional

If..then..else

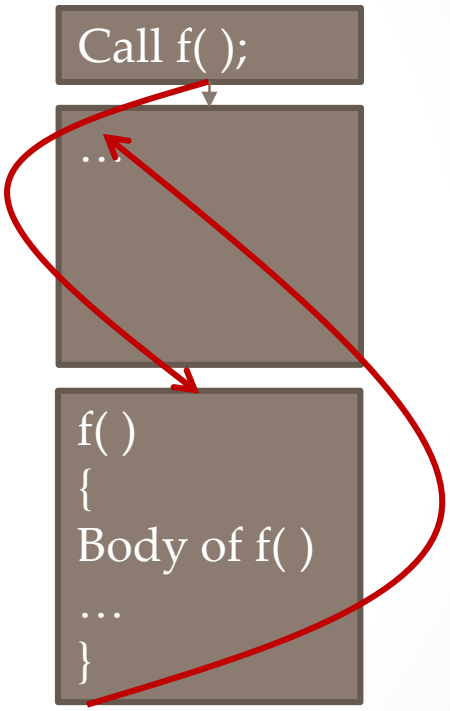


loops



for ( ),
while,
do .. while

subroutines



goto

switch / case

MRI - BSA

Branch to subroutine and save the return address

BSA : M[AR] ← PC, PC ← AR + 1

T_0 : AR ← PC

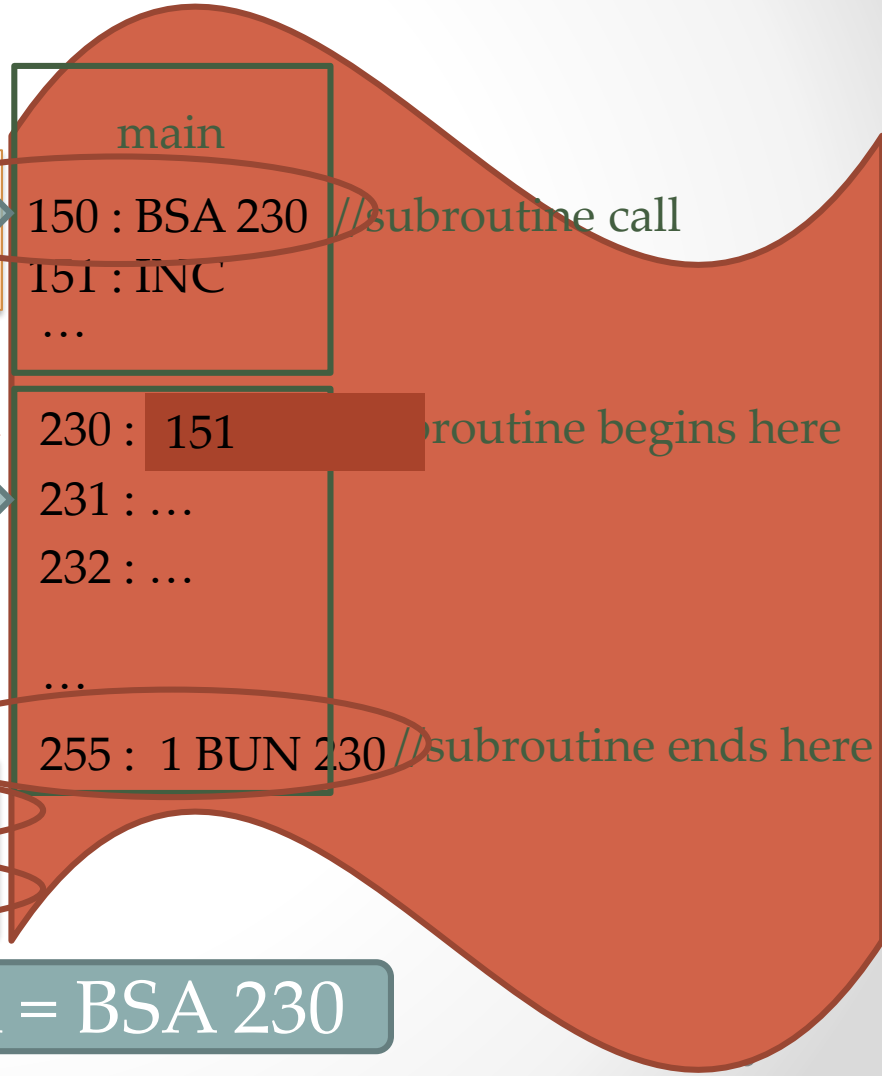
T_1 : IR ← M[AR], PC ← PC + 1

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

D_7`IT_3: AR ←M[AR]

D_5T_4 : M[AR] ← PC, AR ← AR + 1

D_5T_5 : PC ← AR, SC ← 0



PC = 231

AR = 231

IR = BSA 230

MRI – BUN (cont. BSA)

Branch to subroutine and save the return address

BSA : M[AR] ← PC, PC ← AR + 1

T_0 : AR ← PC

T_1 : IR ← M[AR], PC ← PC + 1

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

D_7`IT_3: AR ←M[AR]

D_4T_4 : PC ← AR, SC ← 0

main

150 : BSA 230 //subroutine call
151 : INC
…

230 : 151 broutine begins here
231 : …
232 : …

…

255 : 1 BUN 230 //subroutine ends here

PC = 151

AR = 151

IR = 1 BUN 230

I = 1

MRI - ISZ

Increment memory word specified by the effective address

- if the incremented value is equal to 0, PC is incremented by 1.

Useful for loop indices:

- Place a negative number in memory word
- Increment with each loop iteration
- eventually reaches the value of zero
- **At that time PC is incremented by one in order to skip the next instruction in the program.**

No single microoperation to increment a word inside the memory

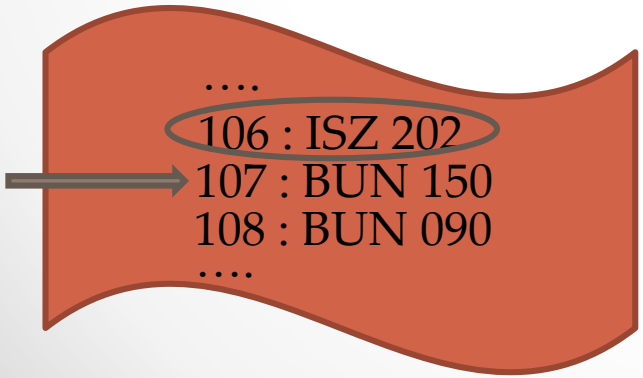
- First read the word into DR,
- increment DR,
- store the word back into memory

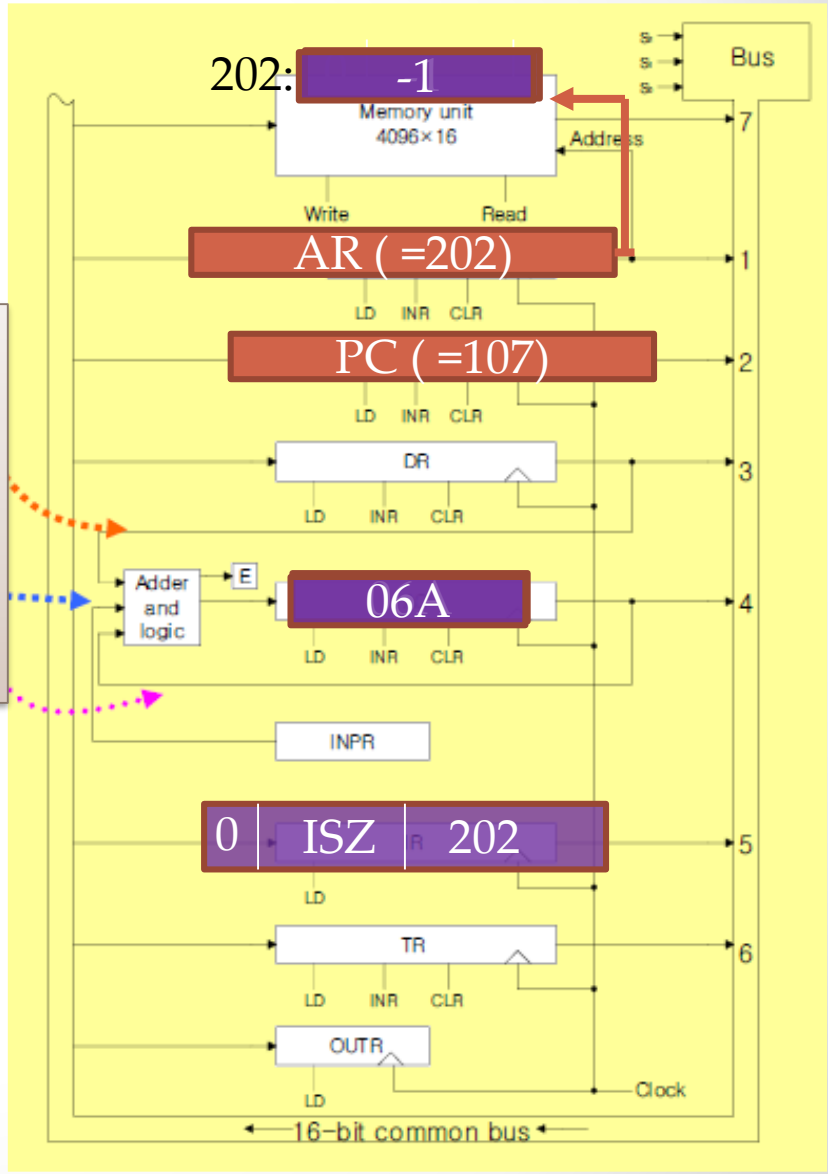
```
// set CTR to -100
LOP,    ...
        ...
        ISZ  CTR
        BUN  LOP
```

MRI - ISZ

Increment memory word and skip next instruction if memory word equals to zero.

D_6T_4 : DR ← M[AR]
D_6T_5 : DR ← DR + 1
D_6T_6 : M[AR] ← DR,
if (DR = 0) then PC ← PC + 1,
SC ← 0

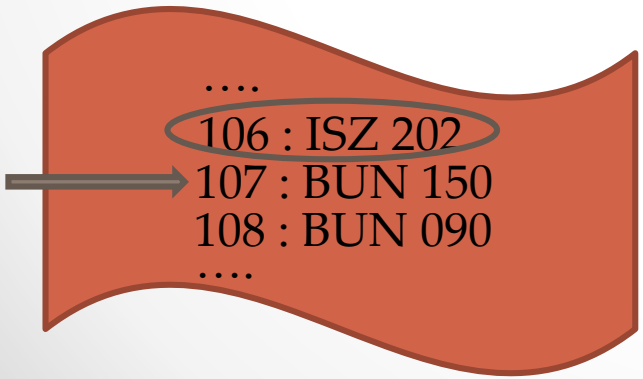


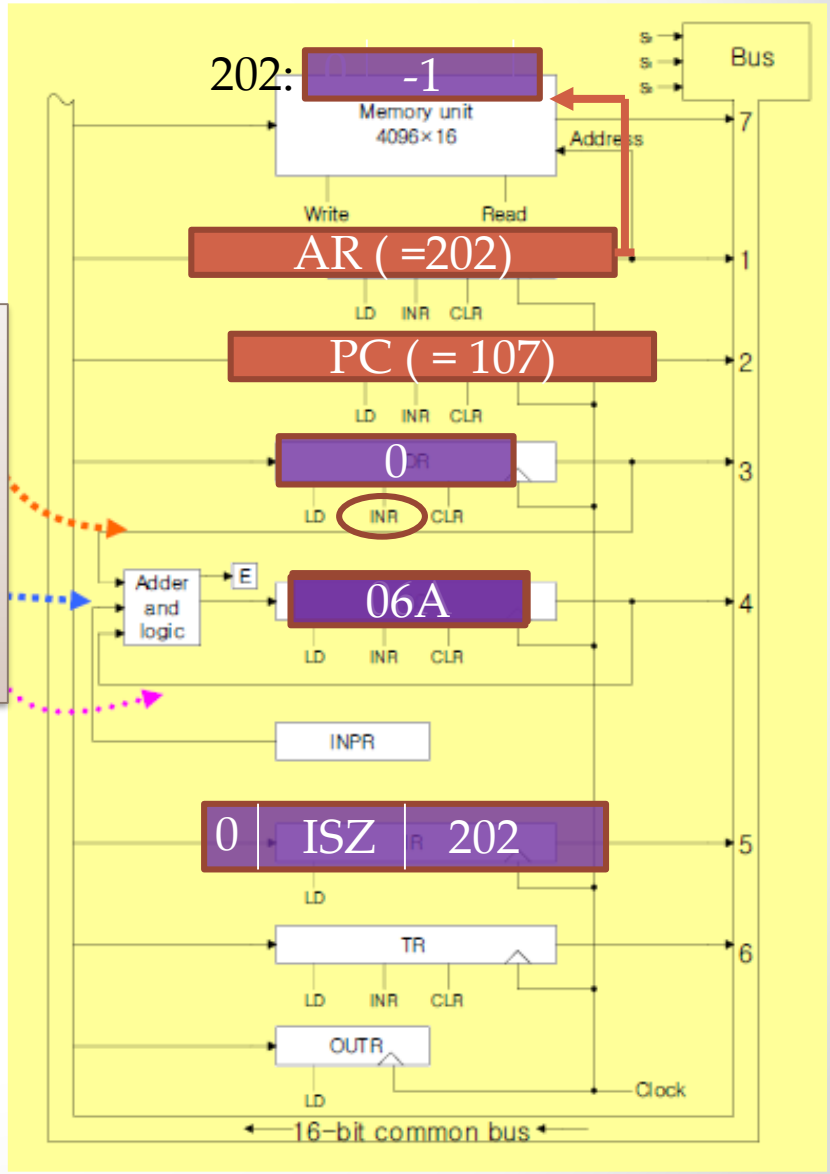


MRI - ISZ

Increment memory word and skip next instruction if memory word equals to zero.

D_6T_4 : DR ← M[AR]
D_6T_5 : DR ← DR + 1
D_6T_6 : M[AR] ← DR,
if (DR = 0) then PC ← PC + 1,
SC ← 0

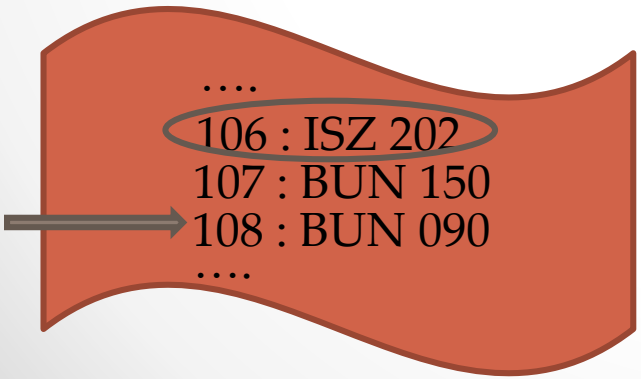


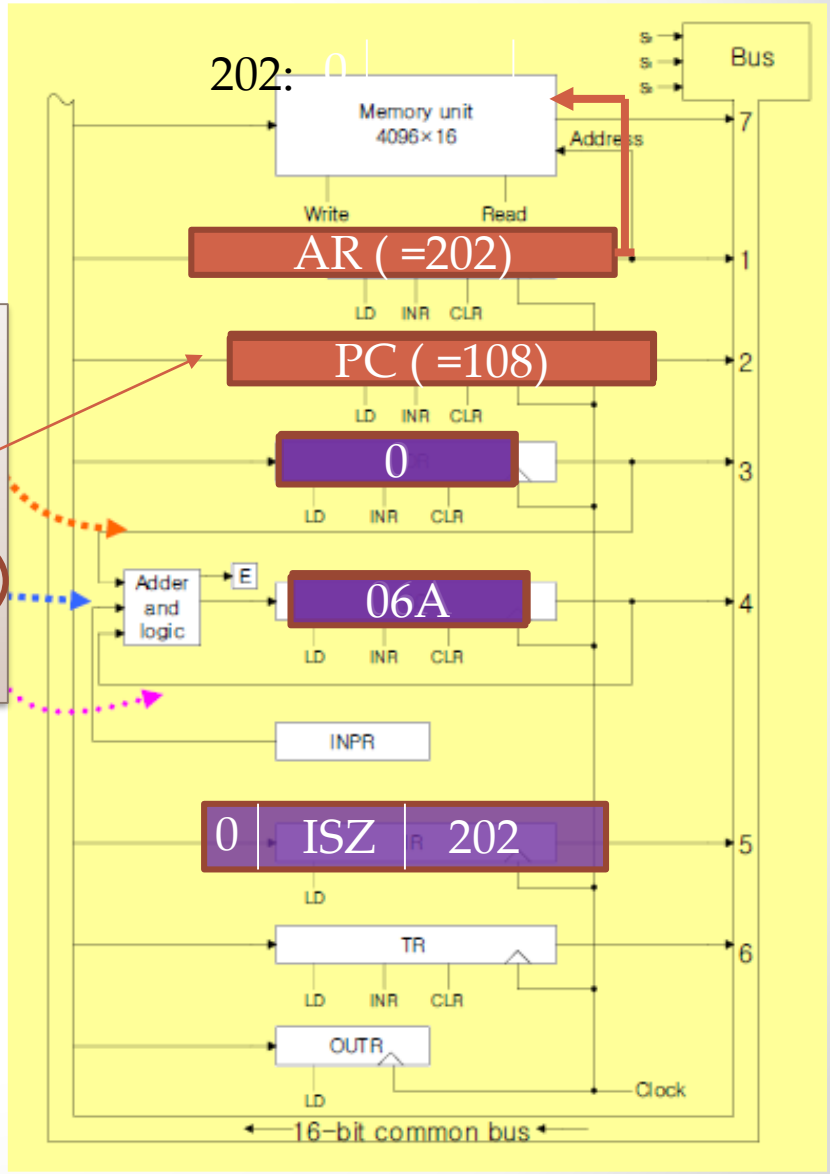


MRI - ISZ

Increment memory word and skip next instruction if memory word equals to zero.

D_6T_4 : DR ← M[AR]
D_6T_5 : DR ← DR + 1
D_6T_6 : M[AR] ← DR,
if (DR = 0) then PC ← PC + 1,
SC ← 0

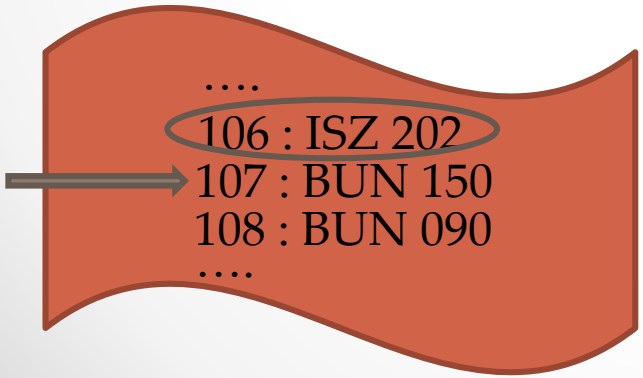


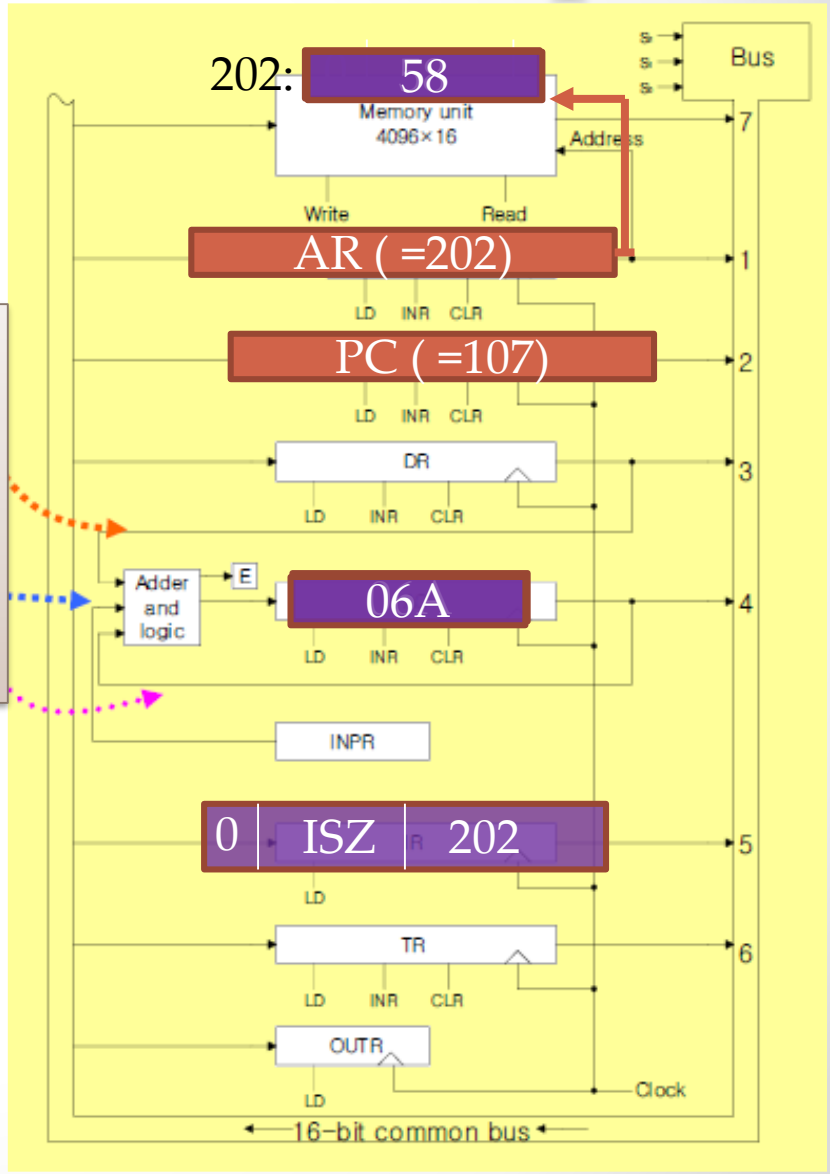


MRI – ISZ (another example)

Increment memory word and skip next instruction if memory word equals to zero.

D_6T_4 : DR ← M[AR]
D_6T_5 : DR ← DR + 1
D_6T_6 : M[AR] ← DR,
if (DR = 0) then PC ← PC + 1,
SC ← 0

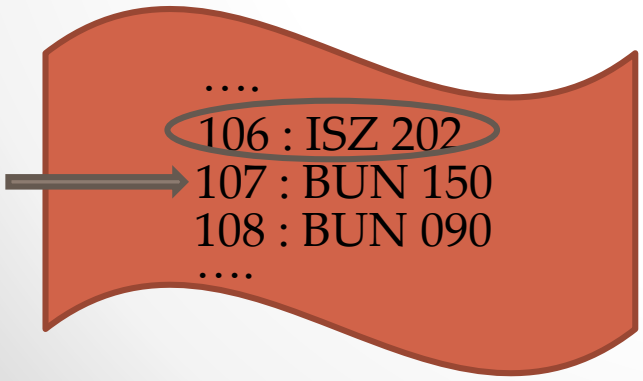


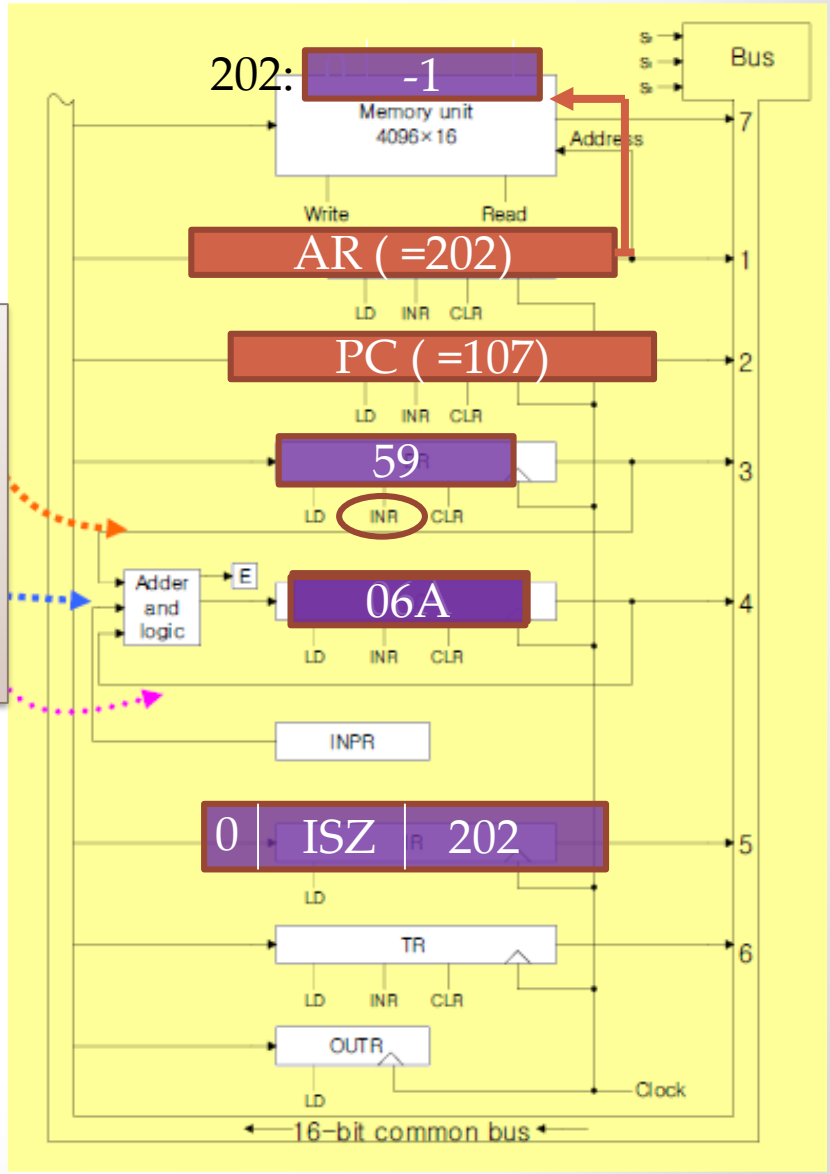


MRI - ISZ

Increment memory word and skip next instruction if memory word equals to zero.

D_6T_4 : DR ← M[AR]
D_6T_5 : DR ← DR + 1
D_6T_6 : M[AR] ← DR,
if (DR = 0) then PC ← PC + 1,
SC ← 0

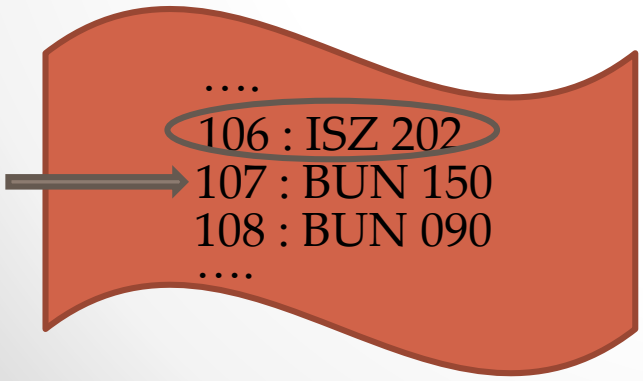


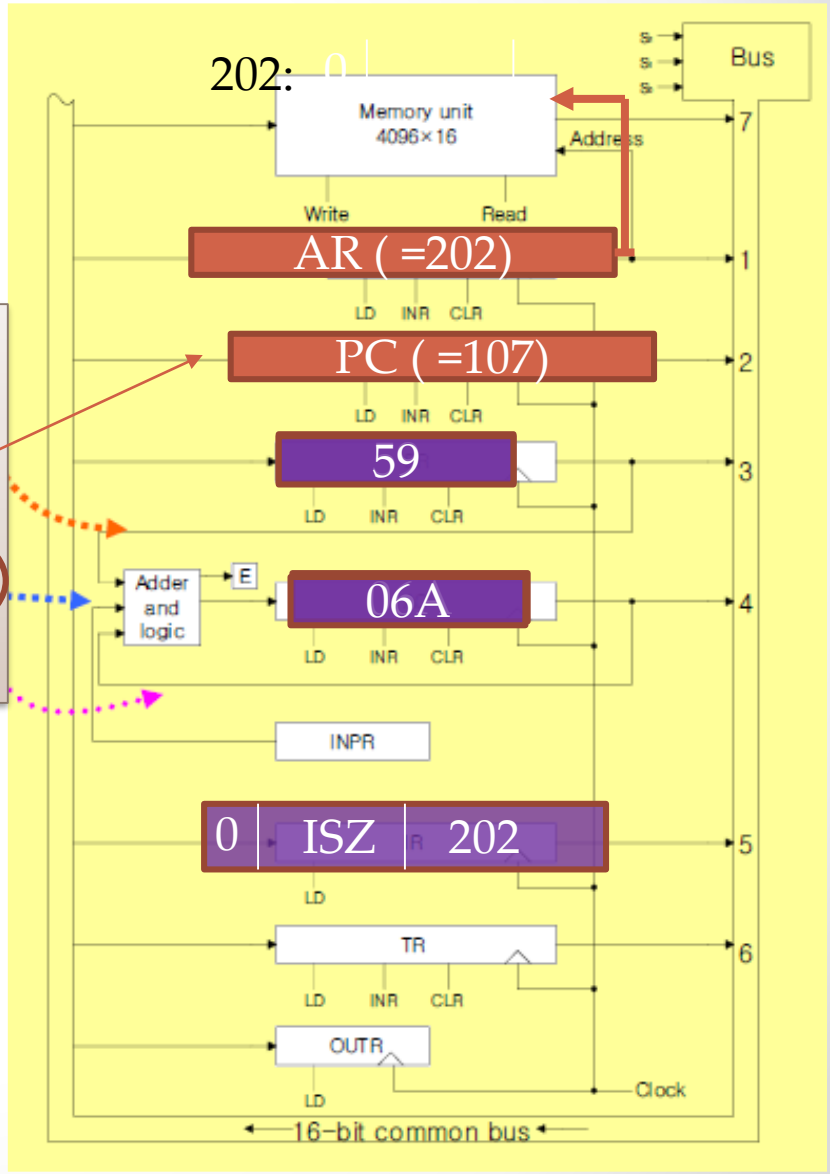


MRI - ISZ

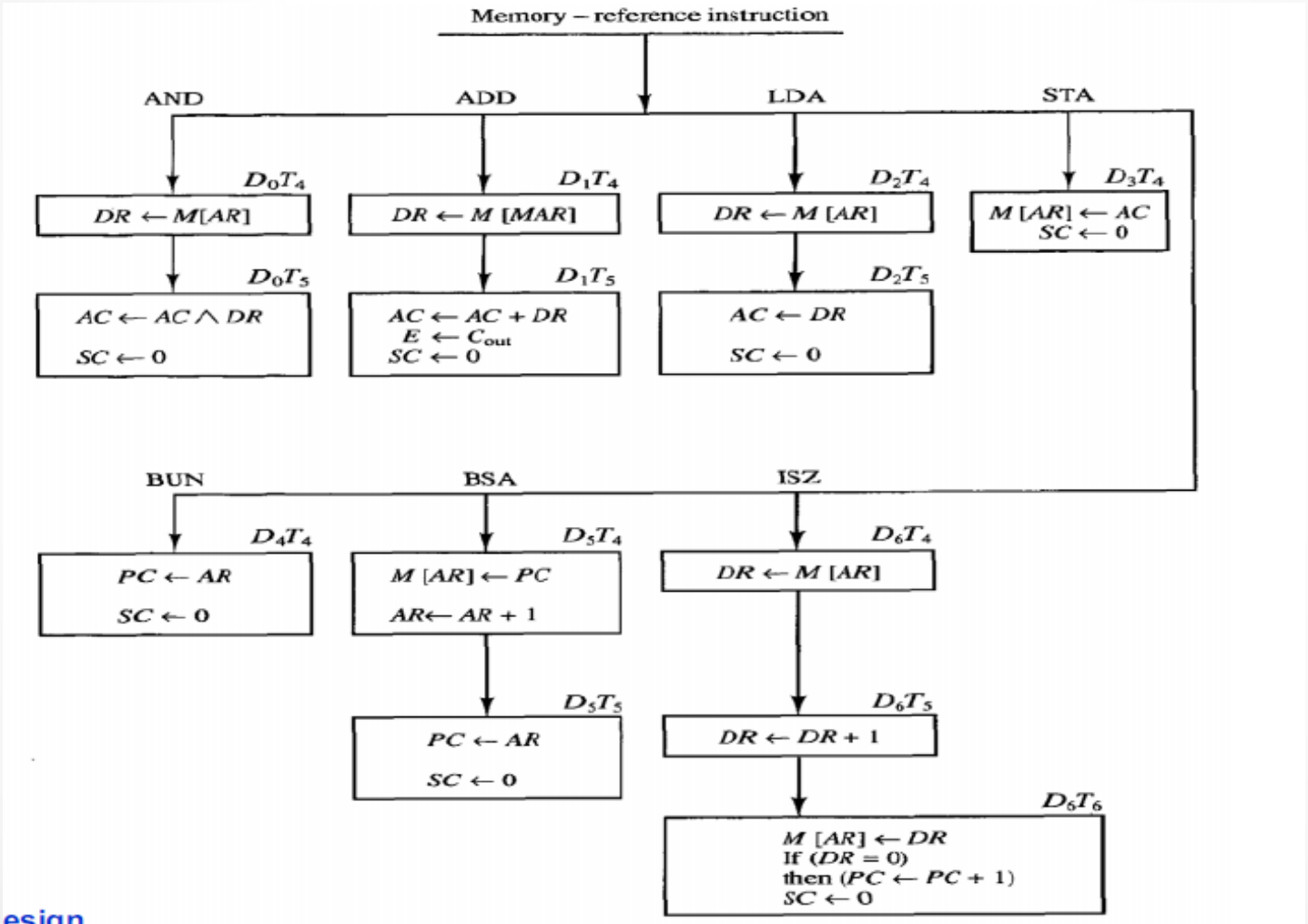
Increment memory word and skip next instruction if memory word equals to zero.

D_6T_4 : DR ← M[AR]
D_6T_5 : DR ← DR + 1
D_6T_6 : M[AR] ← DR,
if (DR = 0) then PC ← PC + 1,
SC ← 0





Memory Reference Flowchart



esign
rd Ed.

Figure 5-11 Flowchart for memory-reference instructions.

QUIZ7

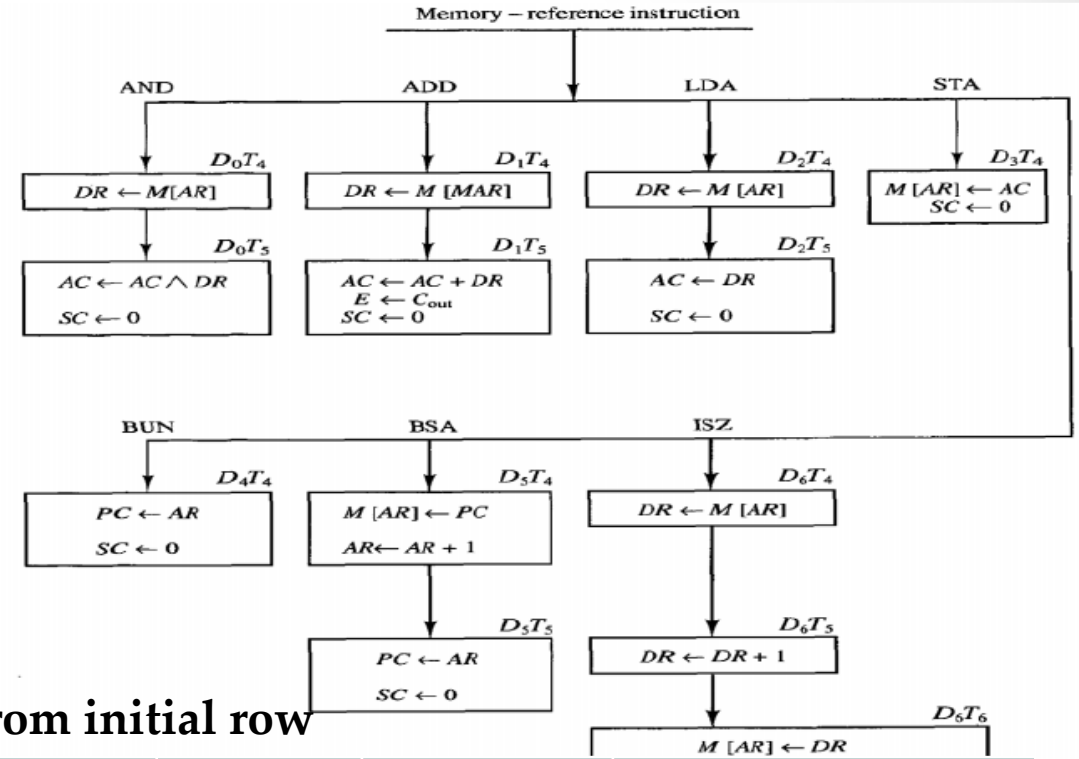
T_0 : AR ← PC

T_1 : IR ← M[AR], PC ← PC + 1

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

D_7`IT_3: AR ←M[AR]

083 : B8F2



For every instruction start from initial row

AC	E	PC	DR	AR	IR	instruction
A937	1	021	-	-	-	initial
						021: ADD 083
						021: LDA 083
						021: BSA 083

QUIZ8

1. What is the next instruction to be performed?
2. Fill the table.

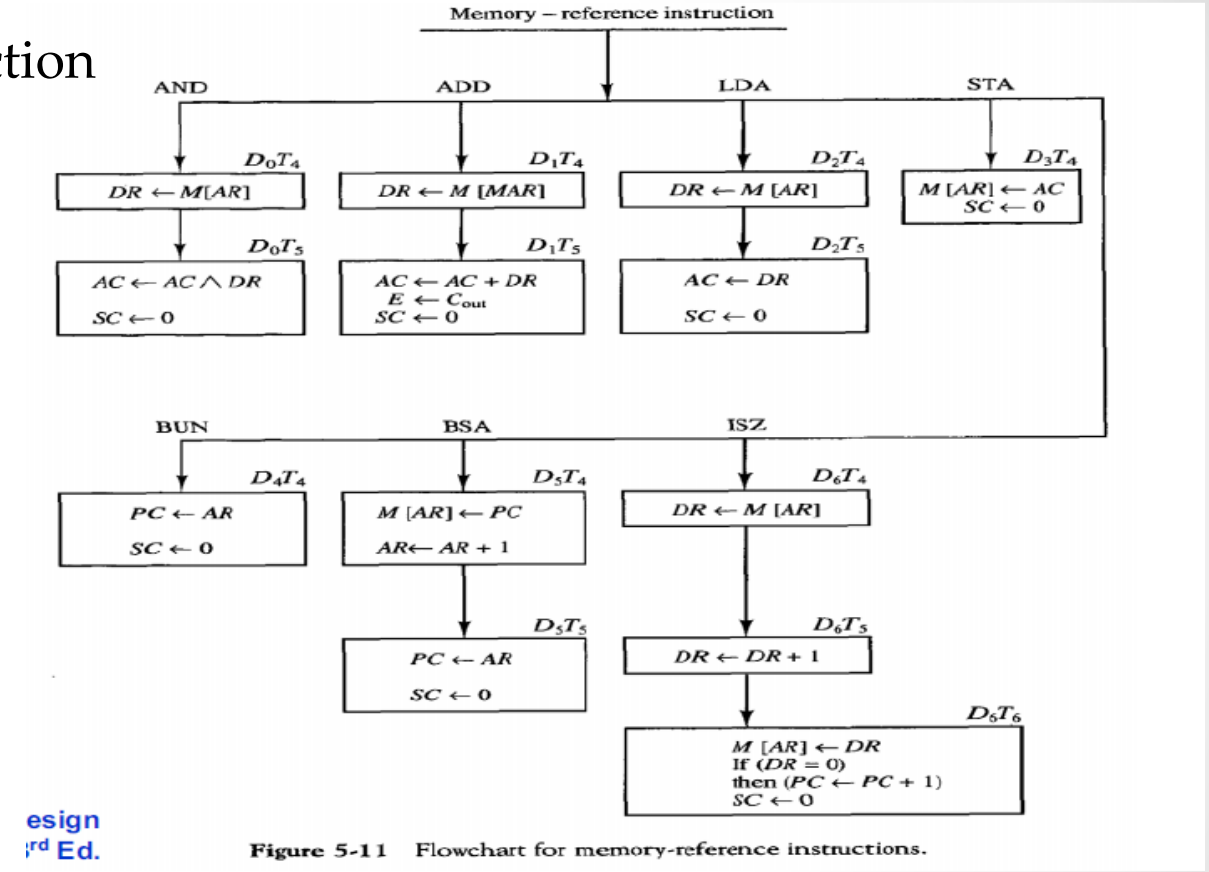
T_0 : AR ← PC

T_1 : IR ← M[AR], PC ← PC + 1

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

D_7`IT_3: AR ←M[AR]

32E : 09AC
…
3AF : 932E
…
9AC : 8B9F



esign
rd Ed.

Figure 5-11 Flowchart for memory-reference instructions.

AC	E	PC	DR	AR	IR	I	SC
7EC3	1	3AF	-	-	-	-	-