Basic CPU – Data Path

Based on 컴퓨터구조론 (개정 6 판), 김종현 , 생능출판사

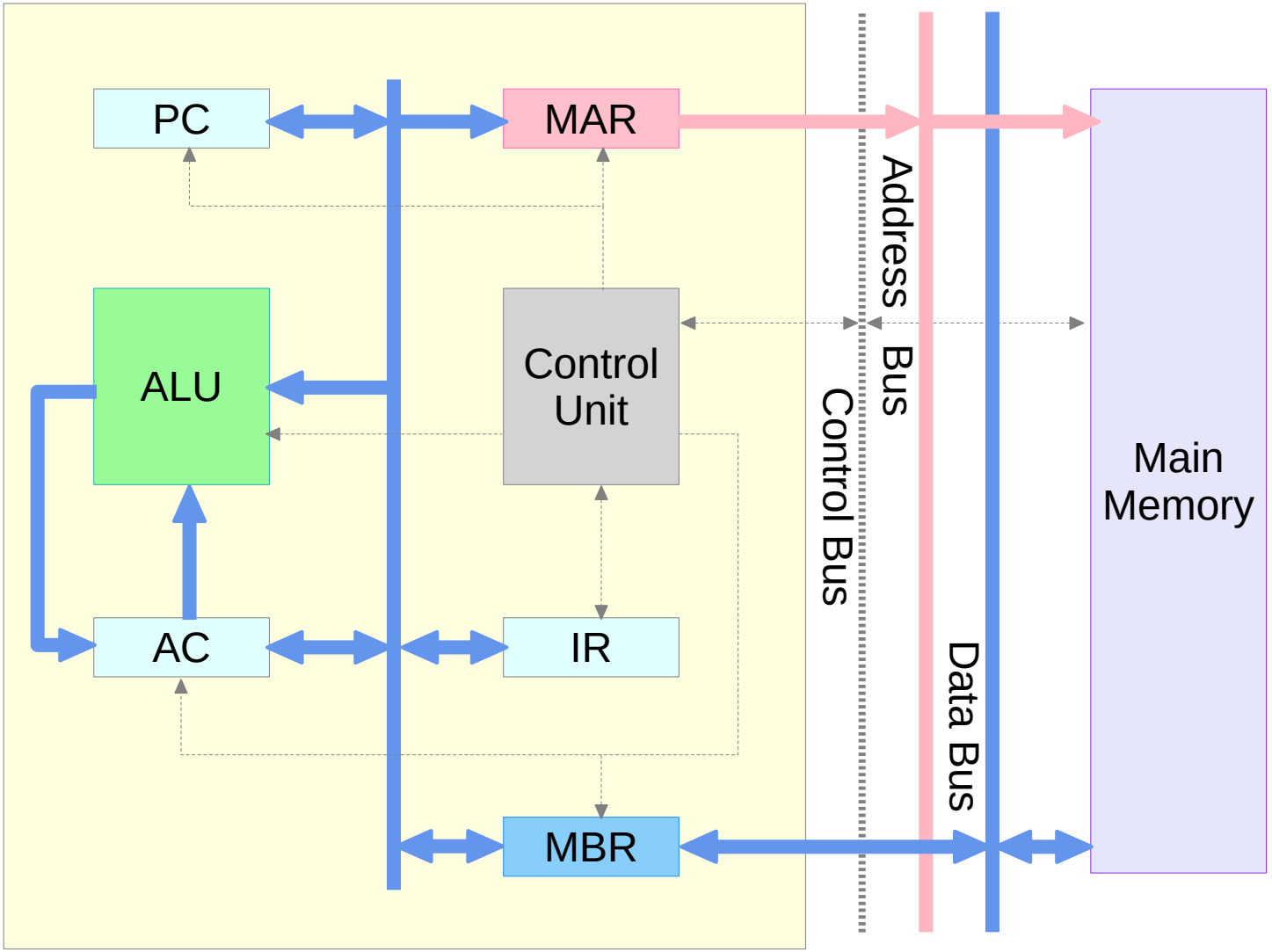
(Computer Architecture, 6th ed, Jonghyun Kim, Life & Power Press, Korea)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

Data Path

Basic CPU



Instruction Cycles

FETCH

t_0 : $MAR \leftarrow PC$

t_1 : $MBR \leftarrow M[MAR]$, $PC \leftarrow PC + 1$

t_2 : $IR \leftarrow MBR$

EXECUTION : *LOAD addr*

t_0 : $MAR \leftarrow IR(addr)$

t_1 : $MBR \leftarrow M[MAR]$

t_2 : $AC \leftarrow MBR$

EXECUTION : *ADD addr*

t_0 : $MAR \leftarrow IR(addr)$

t_1 : $MBR \leftarrow M[MAR]$

t_2 : $AC \leftarrow AC + MBR$

EXECUTION : *STA addr*

t_0 : $MAR \leftarrow IR(addr)$

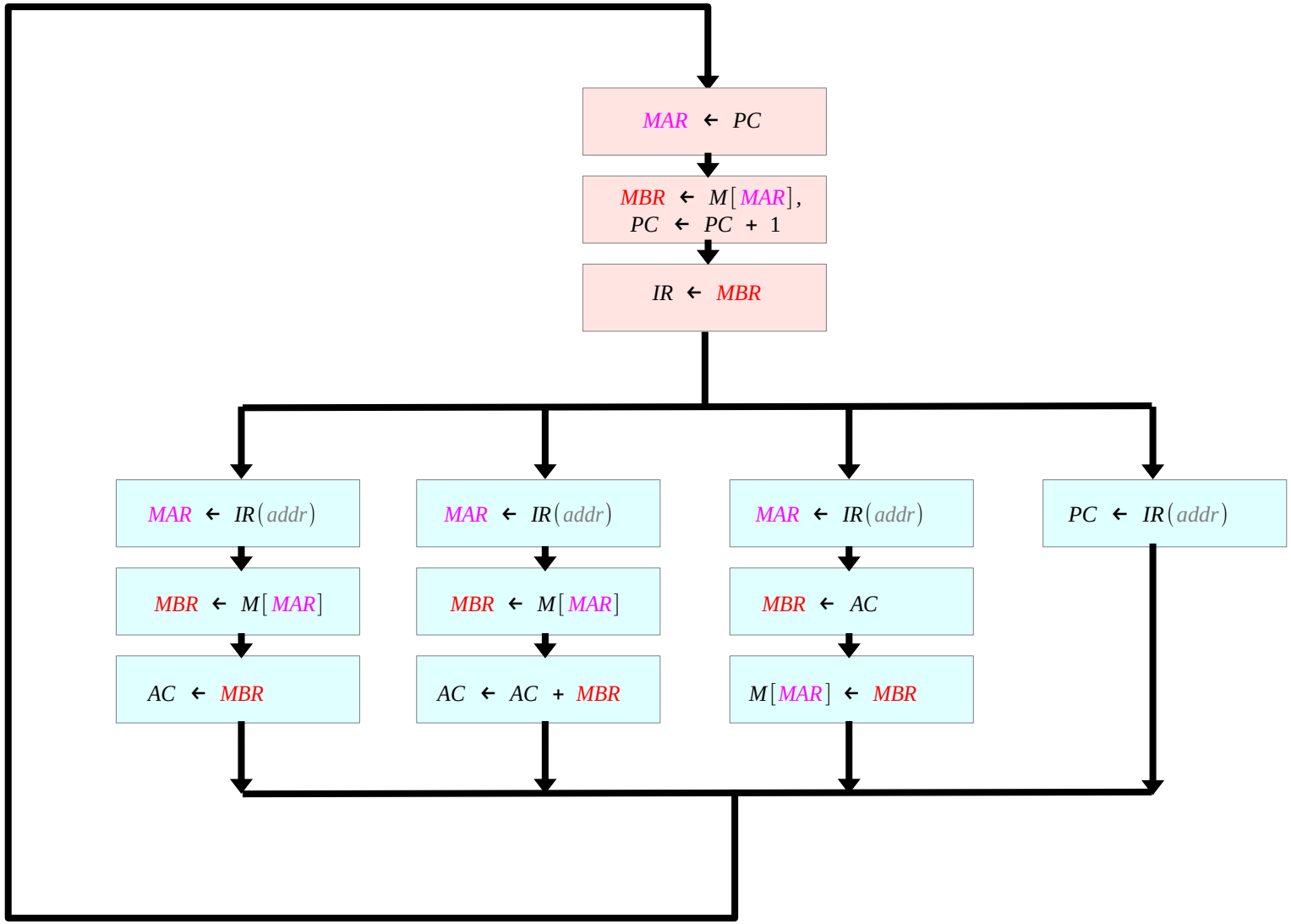
t_1 : $MBR \leftarrow AC$

t_2 : $M[MAR] \leftarrow MBR$

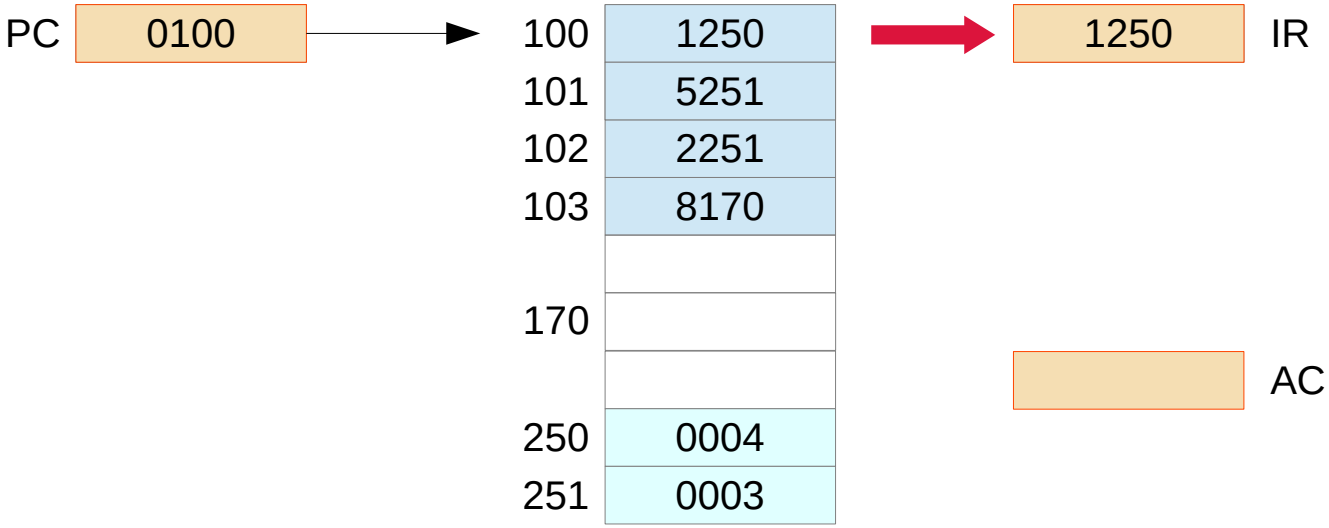
EXECUTION : *JUMP addr*

t_0 : $PC \leftarrow IR(addr)$

Data Path

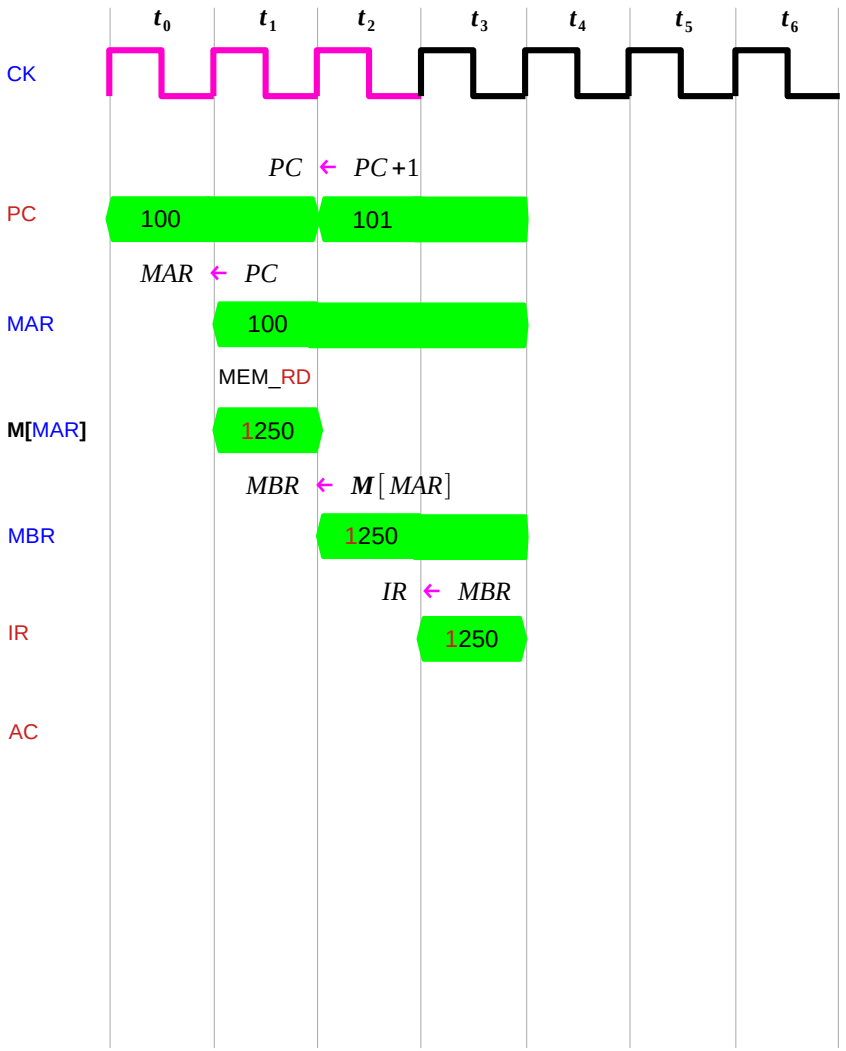

MAR ← PC
MBR ← M[MAR],
PC ← PC + 1
IR ← MBR
MAR ← IR(addr)
MBR ← M[MAR]
AC ← MBR
MAR ← IR(addr)
MBR ← M[MAR]
AC ← AC + MBR
MAR ← IR(addr)
MBR ← AC
M[MAR] ← MBR
PC ← IR(addr)
t0
t1
t2
t3
t4
t5

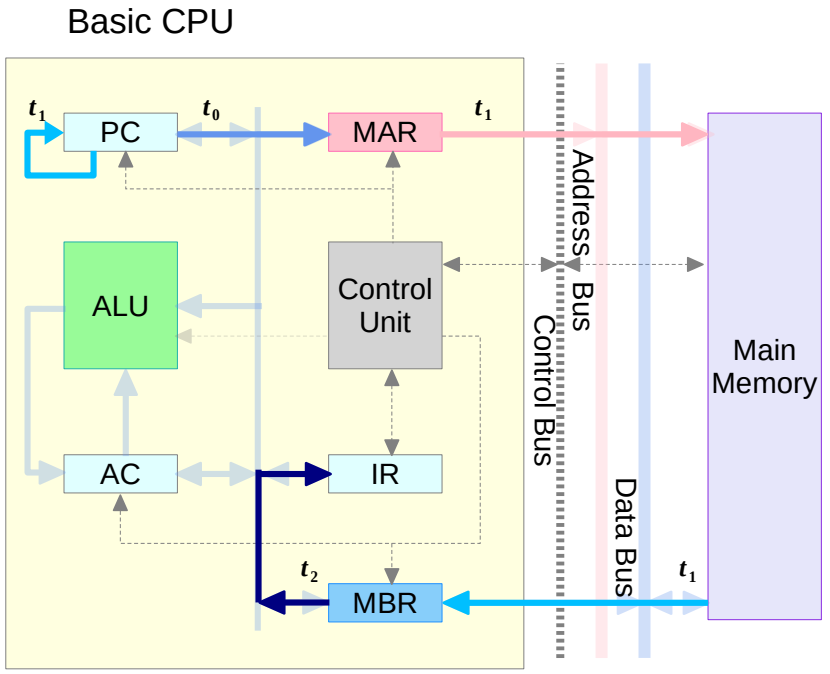
[1] LOAD 250 – (1) Fetch Cycle



LOAD ***250*** ; AC ← M[*250*]

[1] LOAD 250 – (2) Fetch Cycle

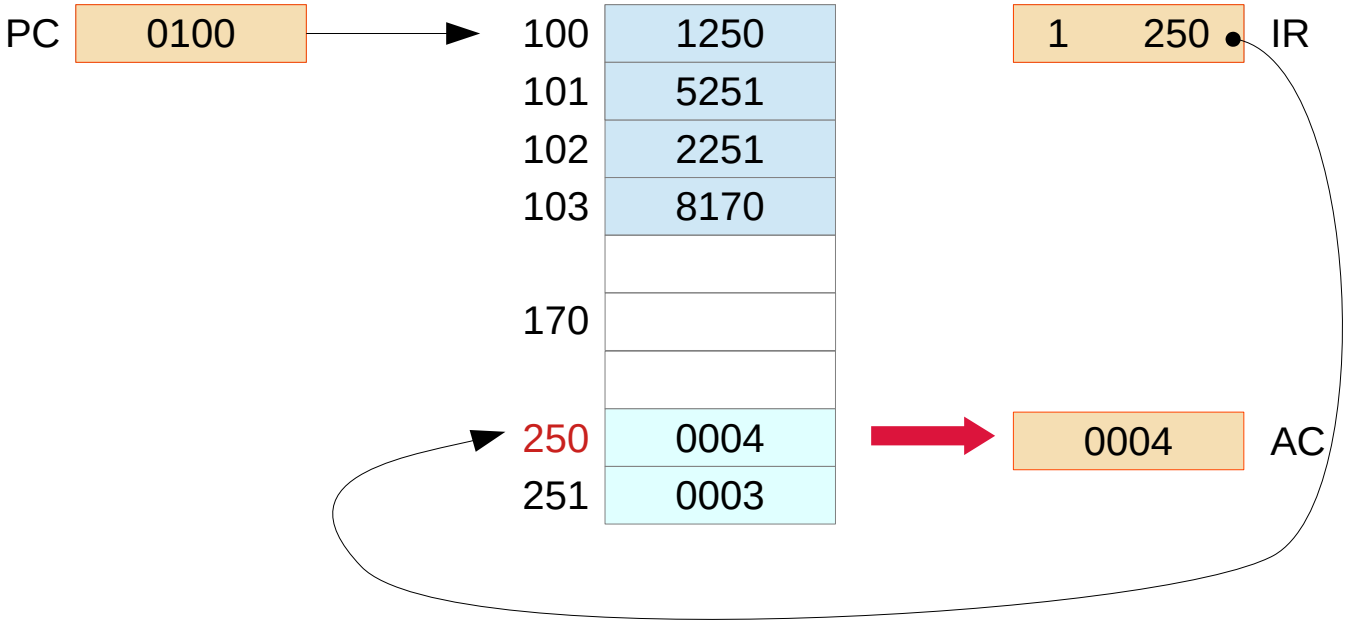




100	1250	**LOAD** ***250***	; AC ← M[*250*]
101	5251	**ADD** ***251***	; AC ← AC + M[*251*]
102	2251	**STA** ***251***	; M[*251*] ← AC
103	8170	**JUMP** ***170***	; PC ← *170*

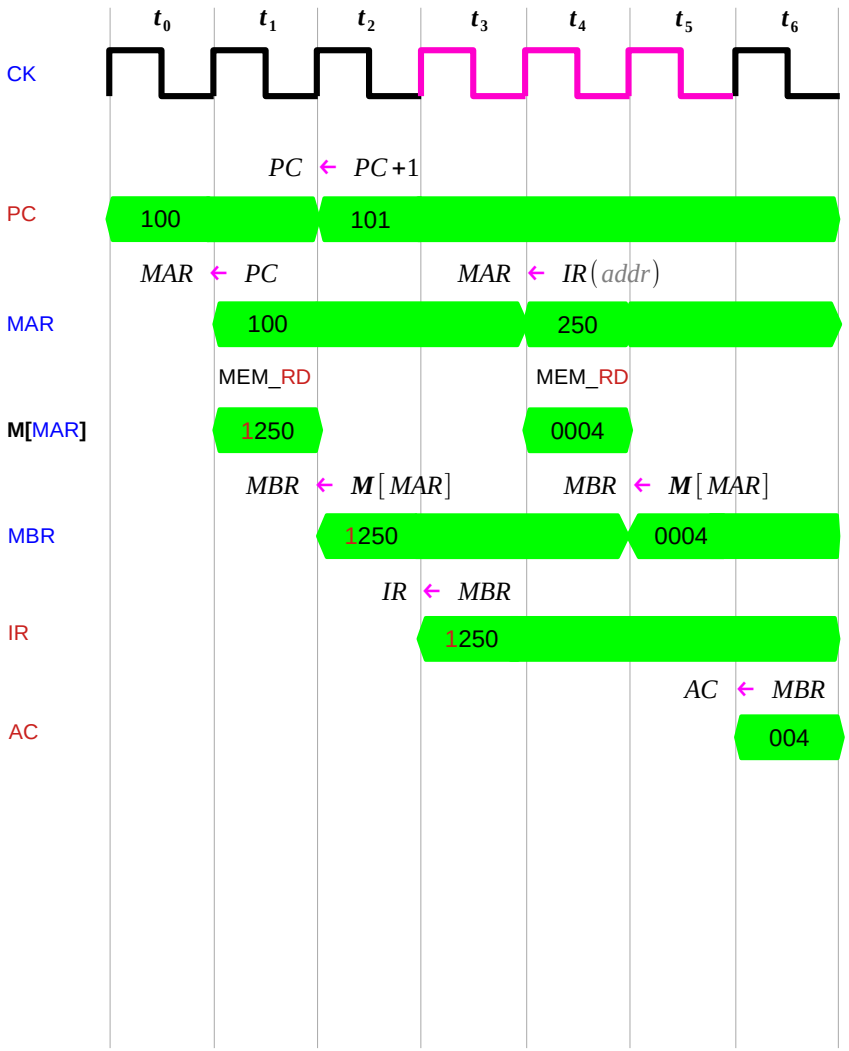
FETCH : *LOAD addr*
t_0 : *MAR* ← *PC*
t_1 : *MBR* ← *M*[*MAR*], *PC* ← *PC* + 1
t_2 : *IR* ← *MBR*

[1] LOAD 250 – (3) Execution Cycle

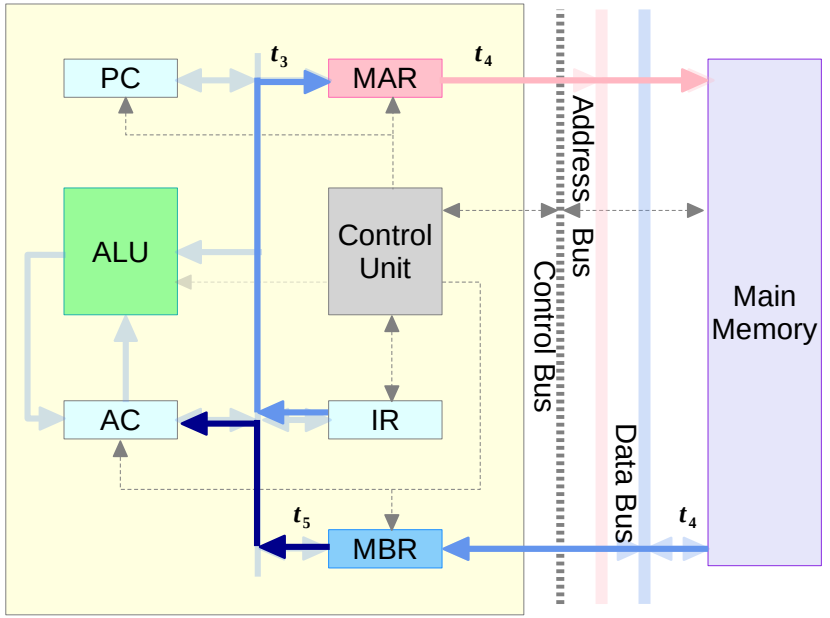


LOAD ***250*** ; AC ← M[*250*]

[1] LOAD 250 – (4) Execution Cycle



Basic CPU



100	1250	**LOAD** ***250***	; AC ← M[*250*]
101	5251	**ADD** ***251***	; AC ← AC + M[*251*]
102	2251	**STA** ***251***	; M[*251*] ← AC
103	8170	**JUMP** ***170***	; PC ← *170*

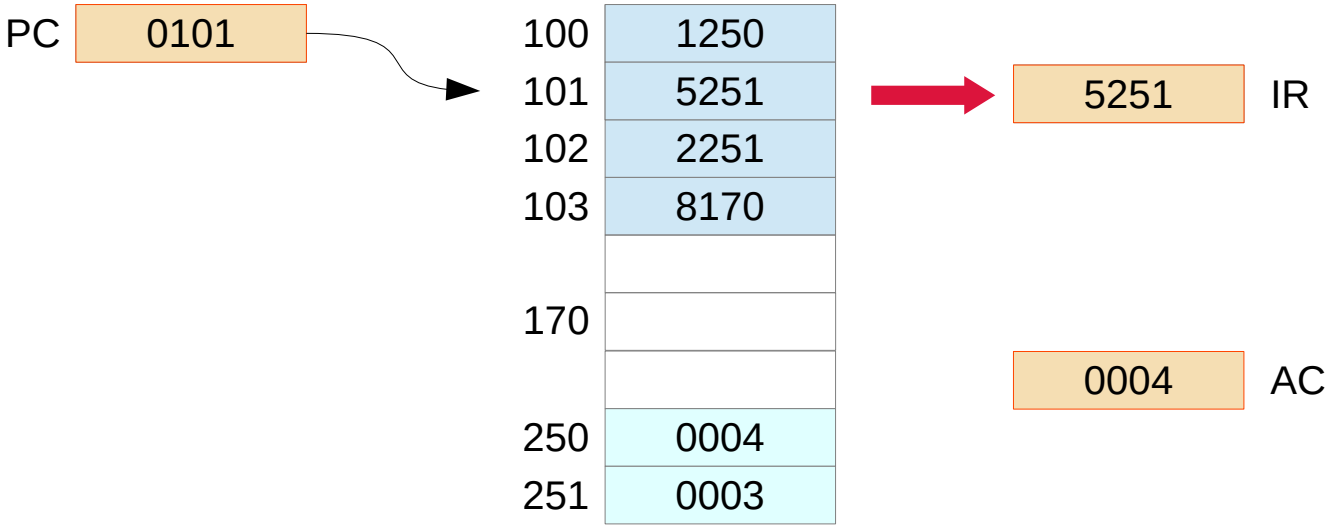
EXECUTION : *LOAD addr*

t_3 : $MAR \leftarrow IR(addr)$

t_4 : $MBR \leftarrow M[MAR]$

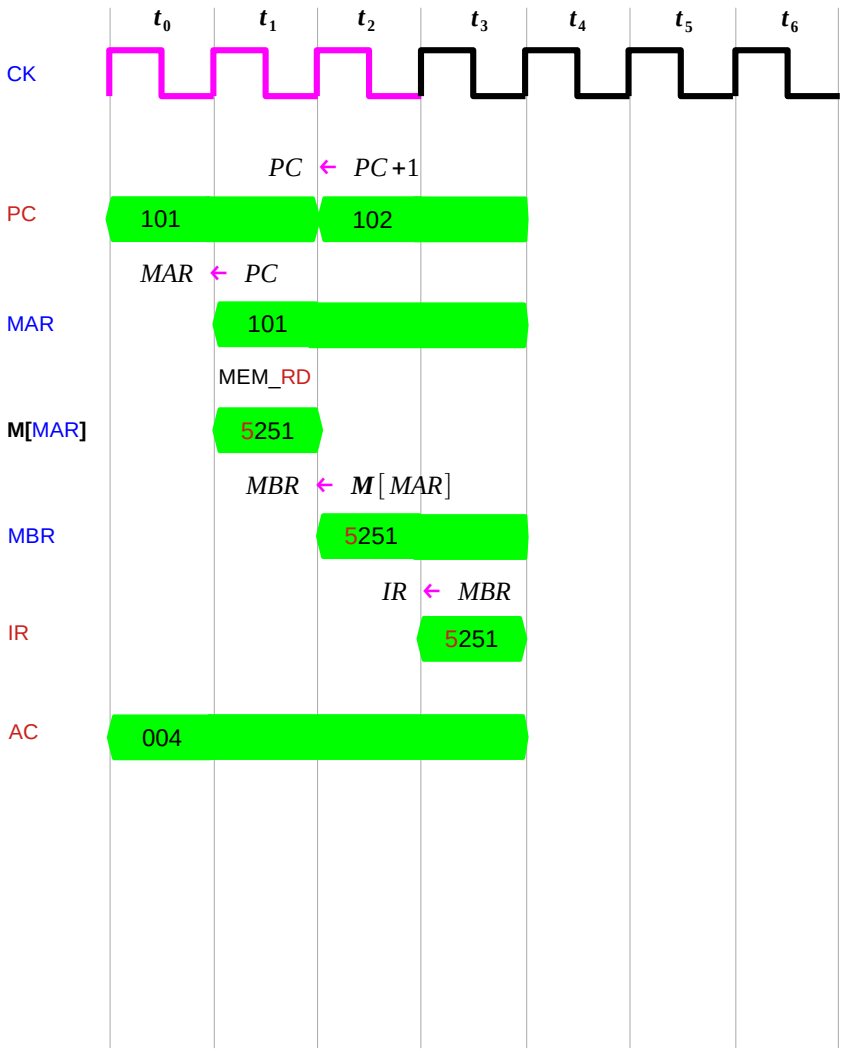
t_5 : $AC \leftarrow MBR$

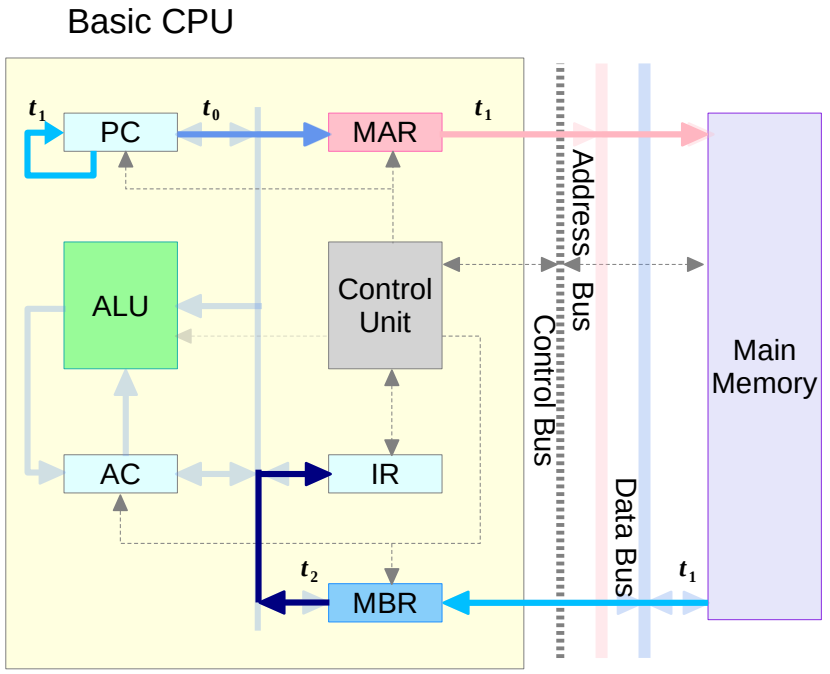
[2] ADD 251 – (1) Fetch Cycle



ADD ***251*** ; AC ← AC + M[*251*]

[2] ADD 251 – (2) Fetch Cycle





100	1250	**LOAD** ***250***	; AC ← M[*250*]
101	5251	**ADD** ***251***	; AC ← AC + M[*251*]
102	2251	**STA** ***251***	; M[*251*] ← AC
103	8170	**JUMP** ***170***	; PC ← *170*

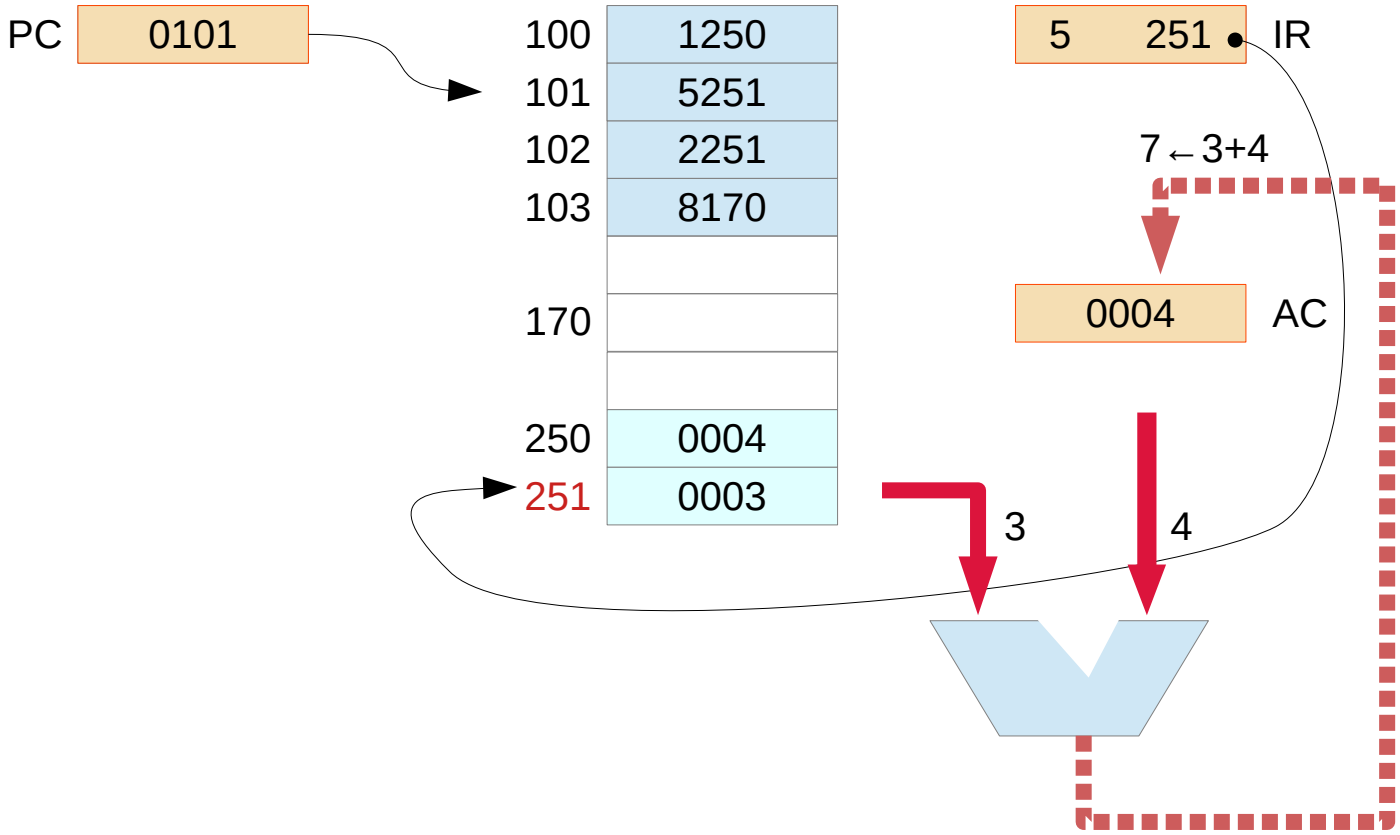
FETCH : *LOAD addr*

t_0 : $MAR \leftarrow PC$

t_1 : $MBR \leftarrow M[MAR],\ PC \leftarrow PC + 1$

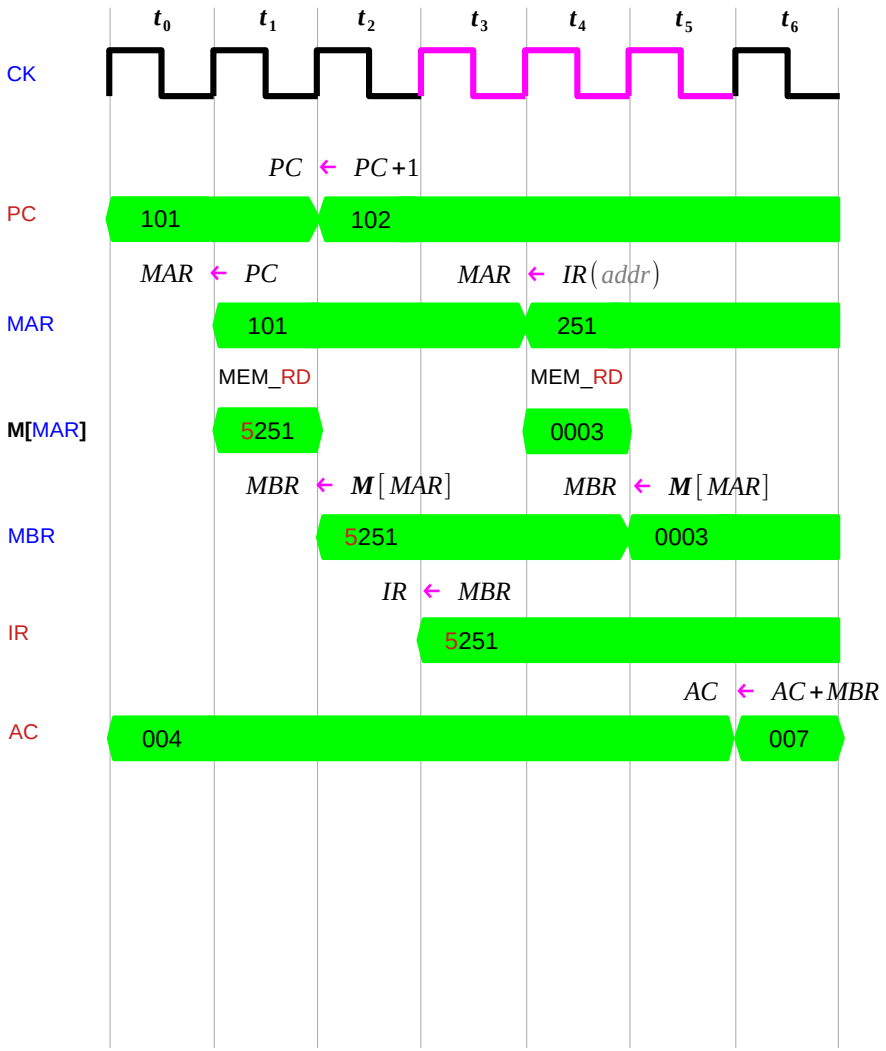
t_2 : $IR \leftarrow MBR$

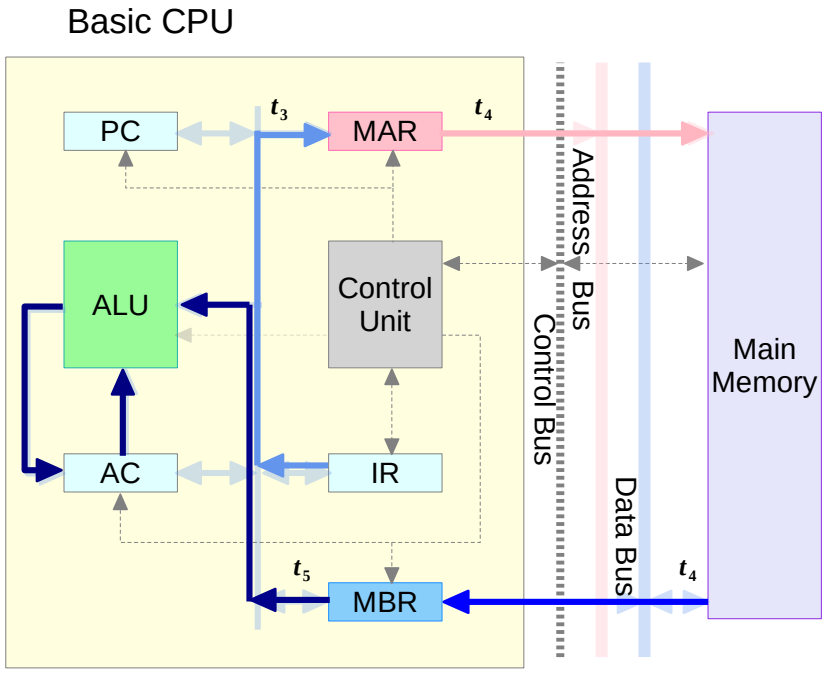
[2] ADD 251 – (3) Execution Cycle



ADD ***251*** ; AC ← AC + M[*251*]

[2] ADD 251 – (4) Execution Cycle





100	1250	**LOAD** ***250*** ; AC ← M[*250*]
101	5251	**ADD** ***251*** ; AC ← AC + M[*251*]
102	2251	**STA** ***251*** ; M[*251*] ← AC
103	8170	**JUMP** ***170*** ; PC ← *170*

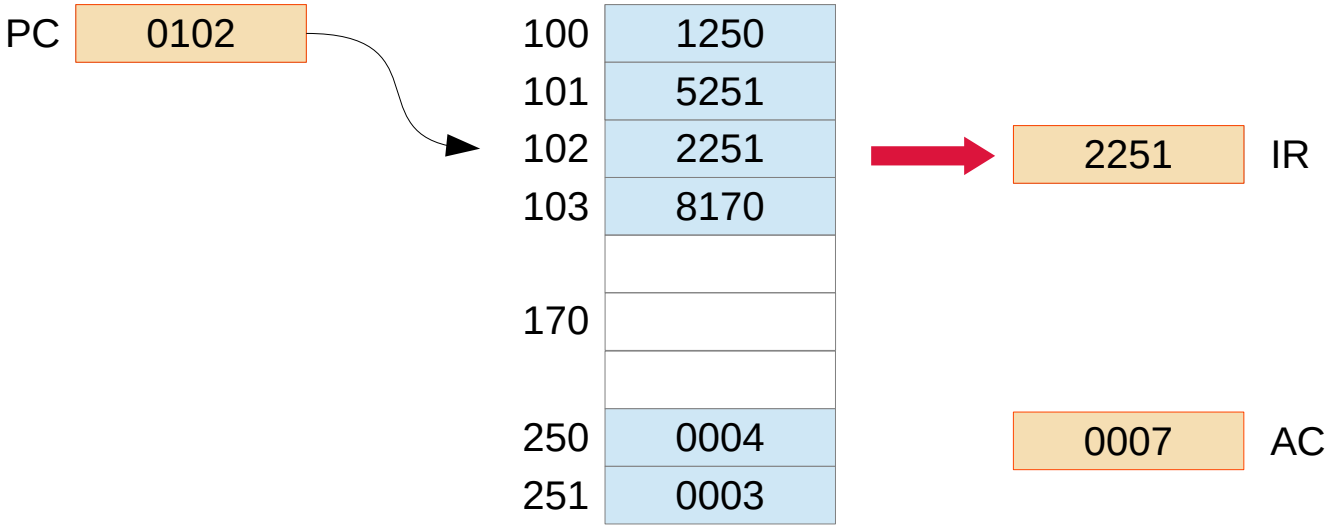
EXECUTION : *ADD addr*

t_3 : $MAR \leftarrow IR(addr)$

t_4 : $MBR \leftarrow M[MAR]$

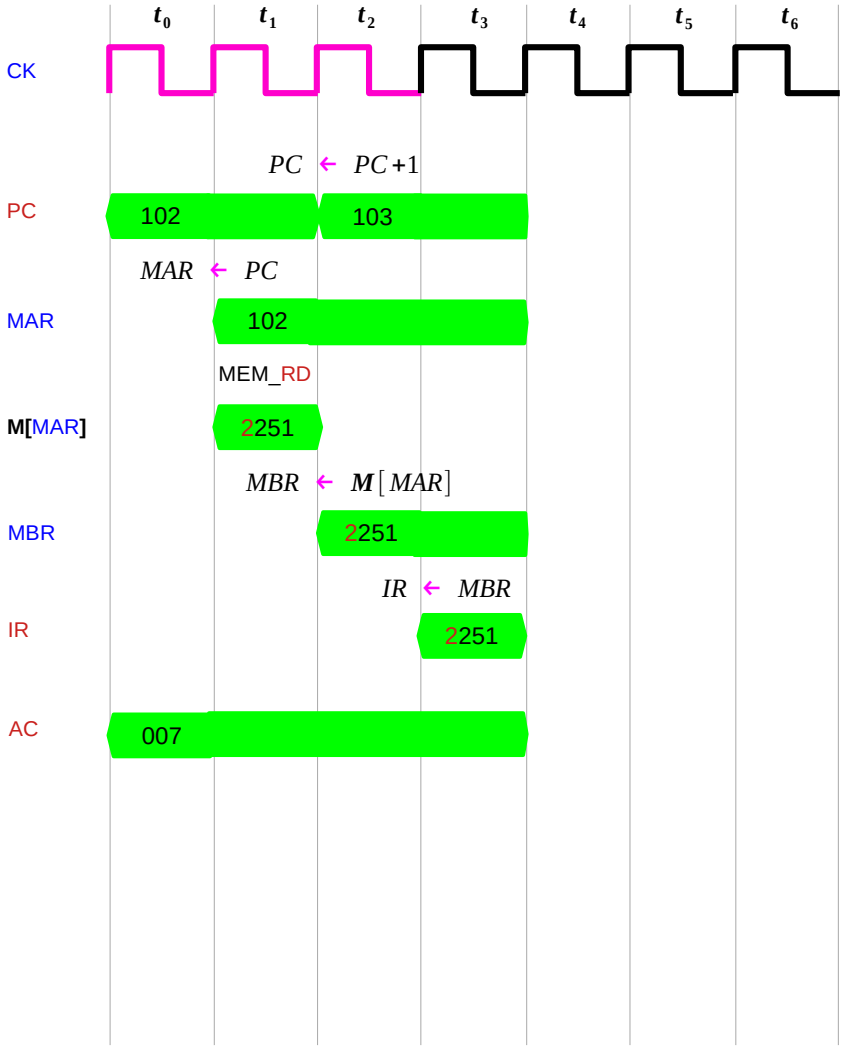
t_5 : $AC \leftarrow AC + MBR$

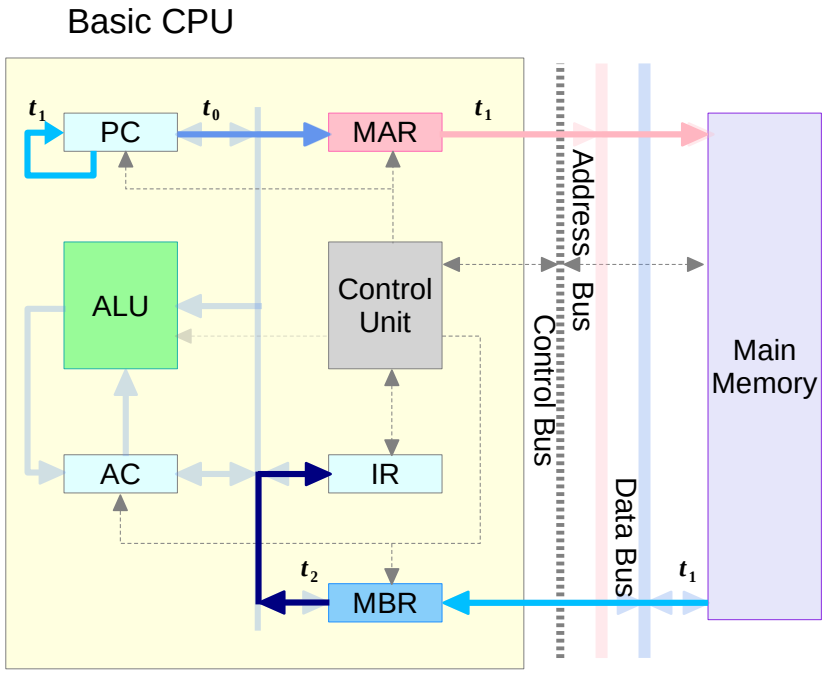
[3] STA 251 – (1) Fetch Cycle



STA ***251*** ; M[*251*] ← AC

[3] STA 251 – (2) Fetch Cycle





100	1250	**LOAD** ***250*** ; AC ← M[*250*]
101	5251	**ADD** ***251*** ; AC ← AC + M[*251*]
102	2251	**STA** ***251*** ; M[*251*] ← AC
103	8170	**JUMP** ***170*** ; PC ← *170*

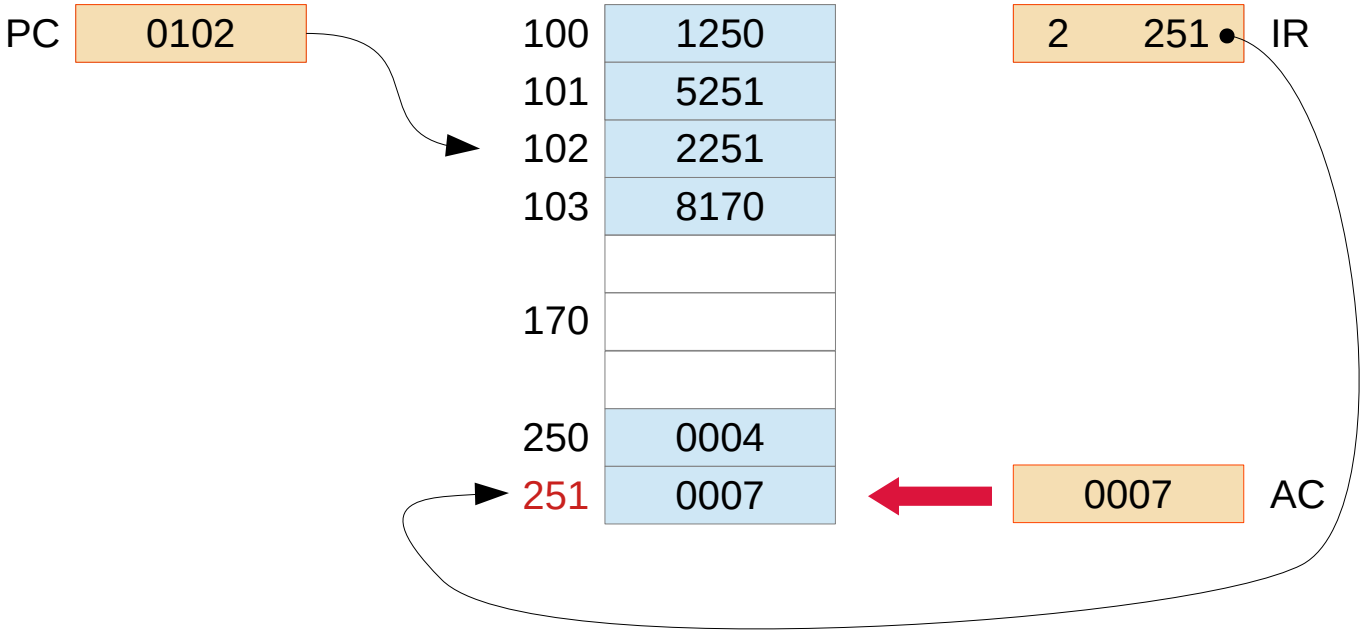
FETCH : *LOAD addr*

t_0 : $MAR \leftarrow PC$

t_1 : $MBR \leftarrow M[MAR],\ PC \leftarrow PC + 1$

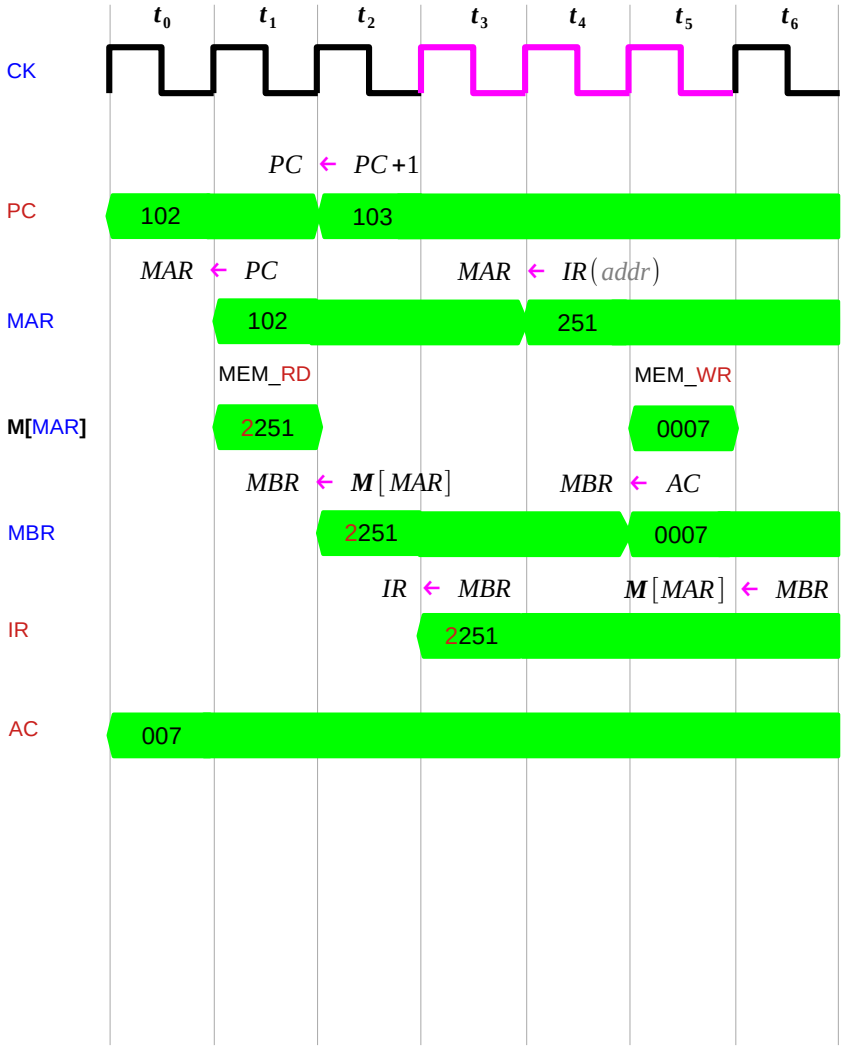
t_2 : $IR \leftarrow MBR$

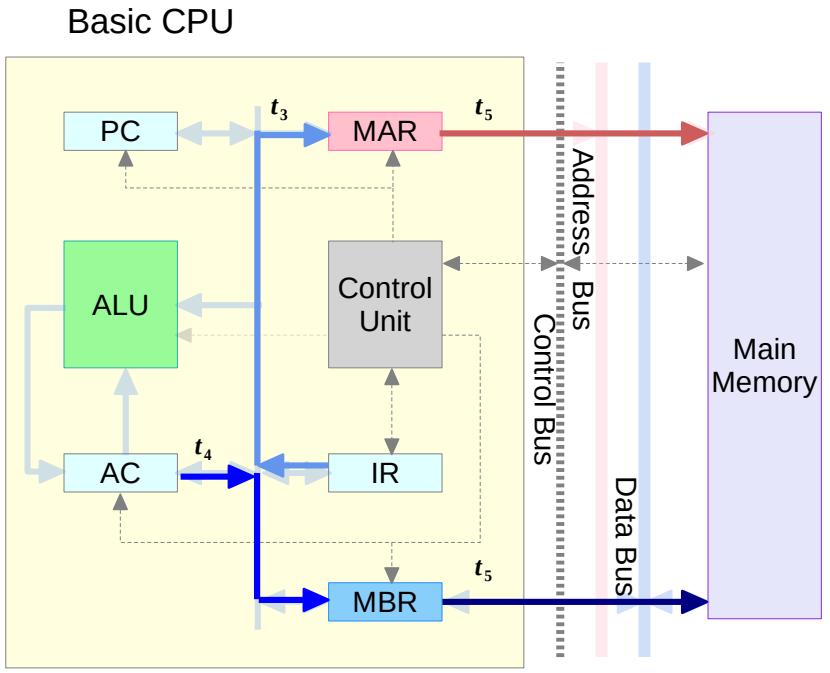
[3] STA 251 – (3) Execution Cycle



STA ***251*** ; M[*251*] ← AC

[3] STA 251 – (4) Execution Cycle





100	1250	**LOAD** ***250*** ; AC ← M[*250*]
101	5251	**ADD** ***251*** ; AC ← AC + M[*251*]
102	2251	**STA** ***251*** ; M[*251*] ← AC
103	8170	**JUMP** ***170*** ; PC ← *170*

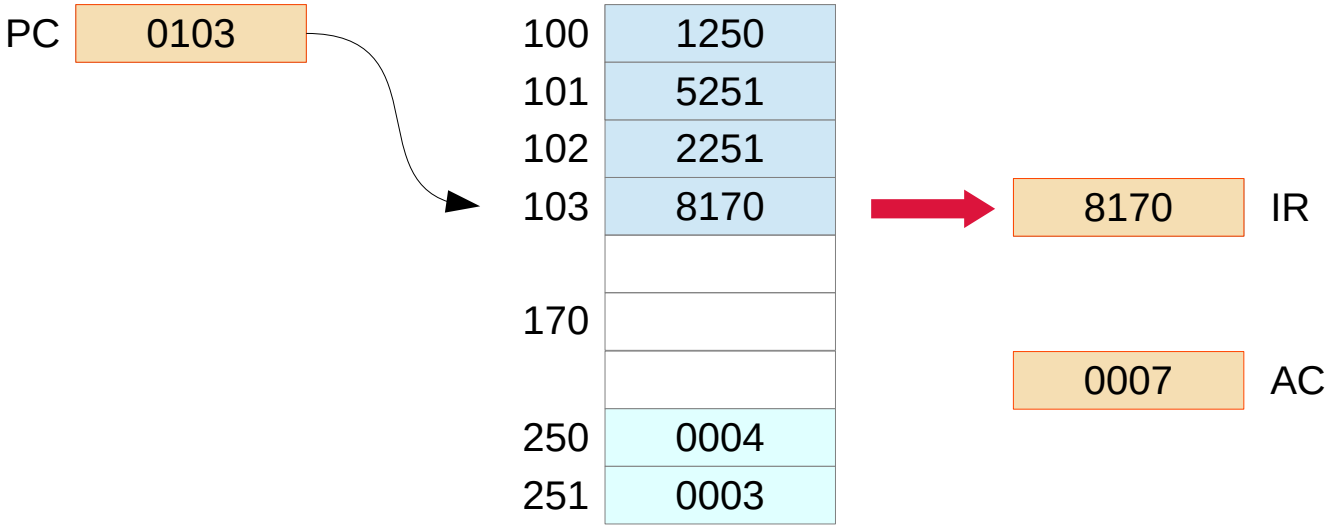
EXECUTION : *STA addr*

t_3 : $MAR \leftarrow IR(addr)$

t_4 : $MBR \leftarrow AC$

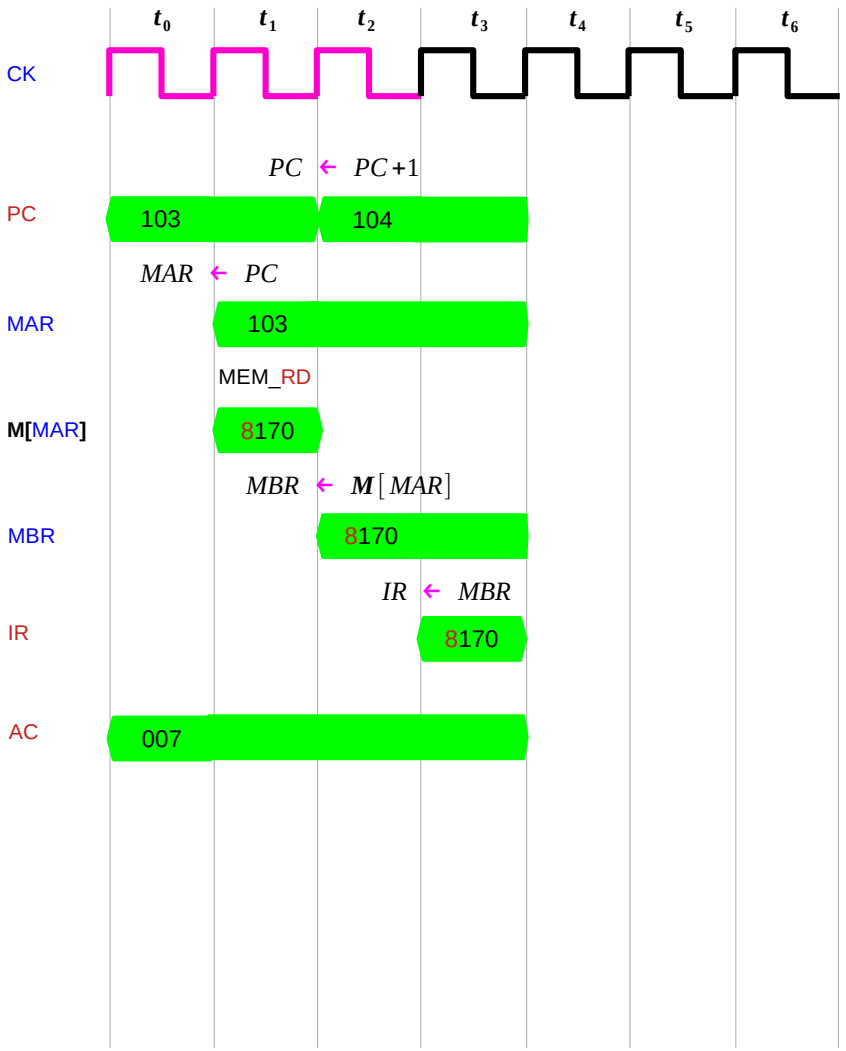
t_5 : $M[MAR] \leftarrow MBR$

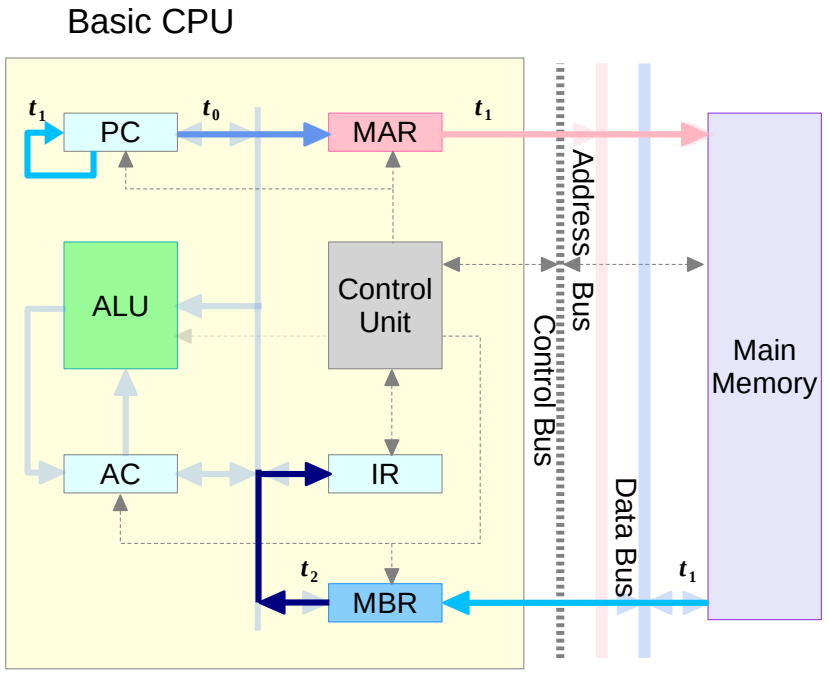
[4] JUMP 170 – (1) Fetch Cycle



JUMP ***170*** ; PC ← *170*

[4] JUMP 170 – (2) Fetch Cycle





100	1250	**LOAD** ***250***	; AC ← M[*250*]
101	5251	**ADD** ***251***	; AC ← AC + M[*251*]
102	2251	**STA** ***251***	; M[*251*] ← AC
103	8170	**JUMP** ***170***	; PC ← *170*

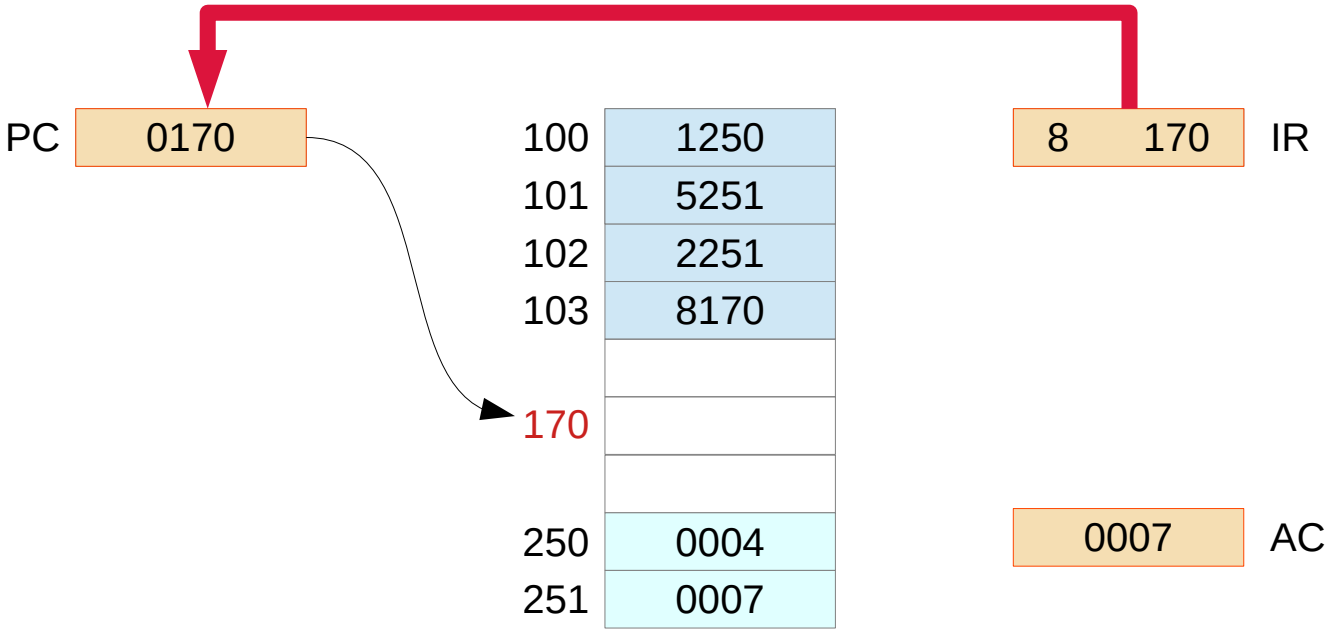
FETCH : *LOAD addr*

t_0 : $MAR \leftarrow PC$

t_1 : $MBR \leftarrow M[MAR],\ PC \leftarrow PC + 1$

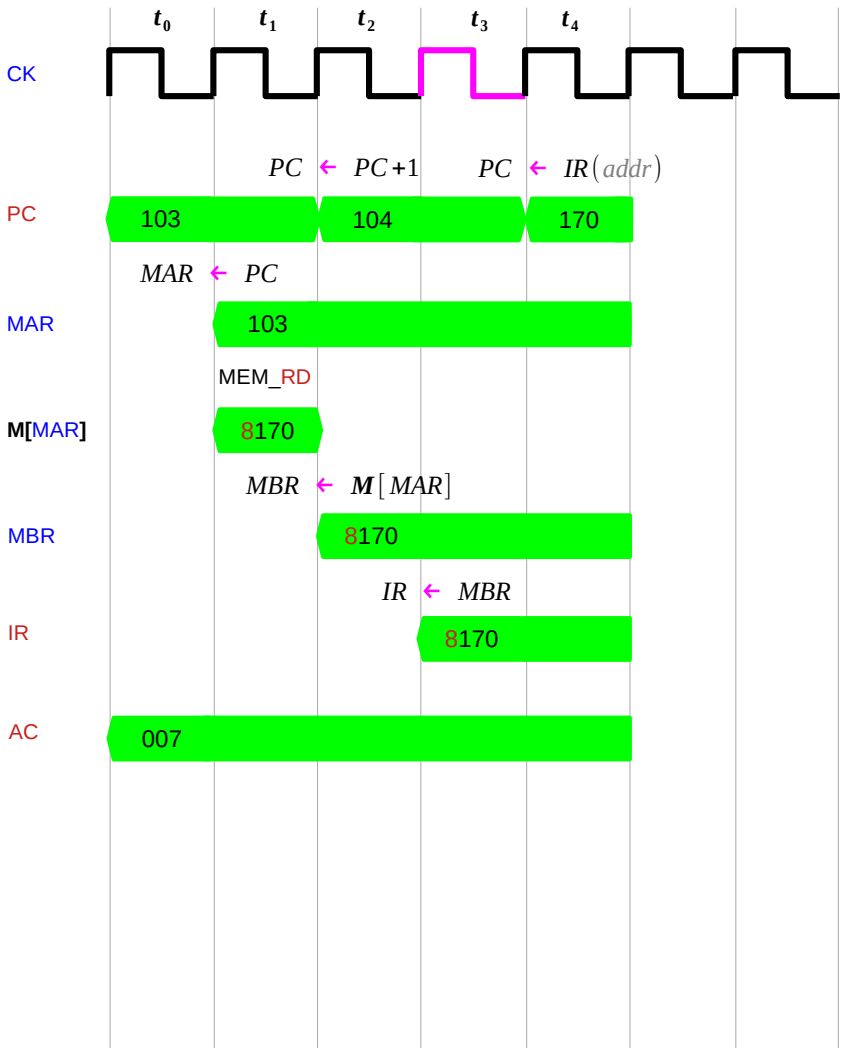
t_2 : $IR \leftarrow MBR$

[4] JUMP 170 – (3) Execution Cycle

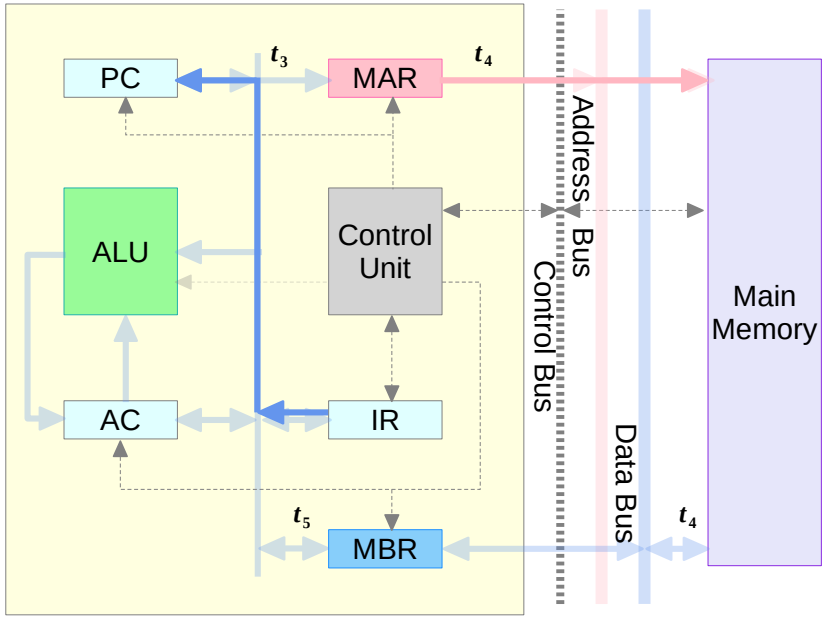


JUMP ***170*** ; PC ← *170*

[4] JUMP 170 – (4) Execution Cycle



Basic CPU



100	1250	**LOAD** ***250***	; AC ← M[*250*]
101	5251	**ADD** ***251***	; AC ← AC + M[*251*]
102	2251	**STA** ***251***	; M[*251*] ← AC
103	8170	**JUMP** ***170***	; PC ← *170*

EXECUTION : *JUMP addr*
t_3 : $PC \leftarrow IR(addr)$

References

[1] http://en.wikipedia.org/
[2] https://en.wikiversity.org/wiki/The_necessities_in_SOC_Design
[3] https://en.wikiversity.org/wiki/The_necessities_in_Digital_Design
[4] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Design
[5] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Architecture
[6] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Organization
[7] https://en.wikiversity.org/wiki/Understanding_Embedded_Software
[8] Digital Systems, Hill, Peterson, 1987
[9] http://en.wikipedia.org/
[10] http://www.ele.uri.edu/Courses/ele306/f01/Tinydoc.pdf