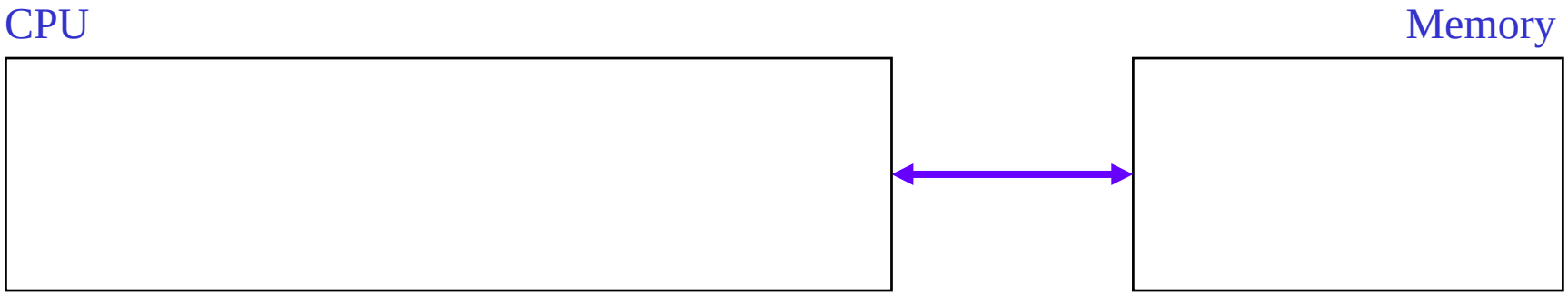
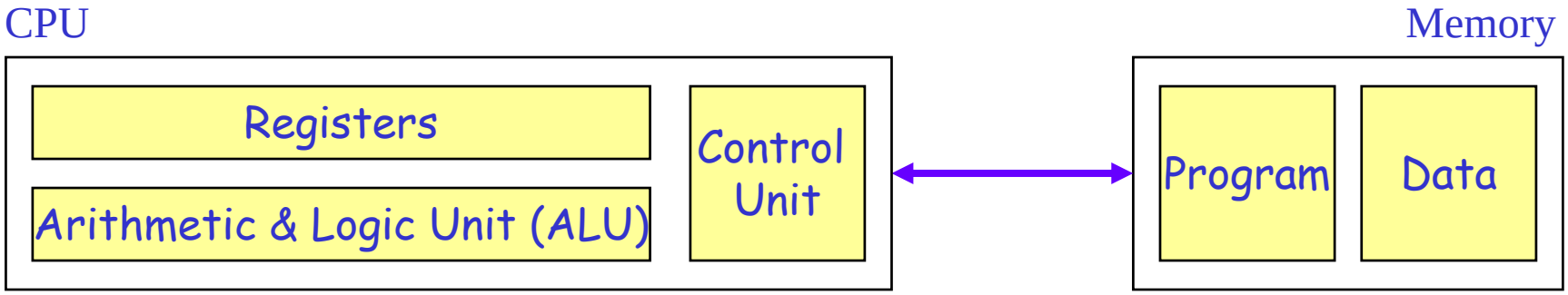
Story so far

- simplest description of computer?
- what makes it general purpose?



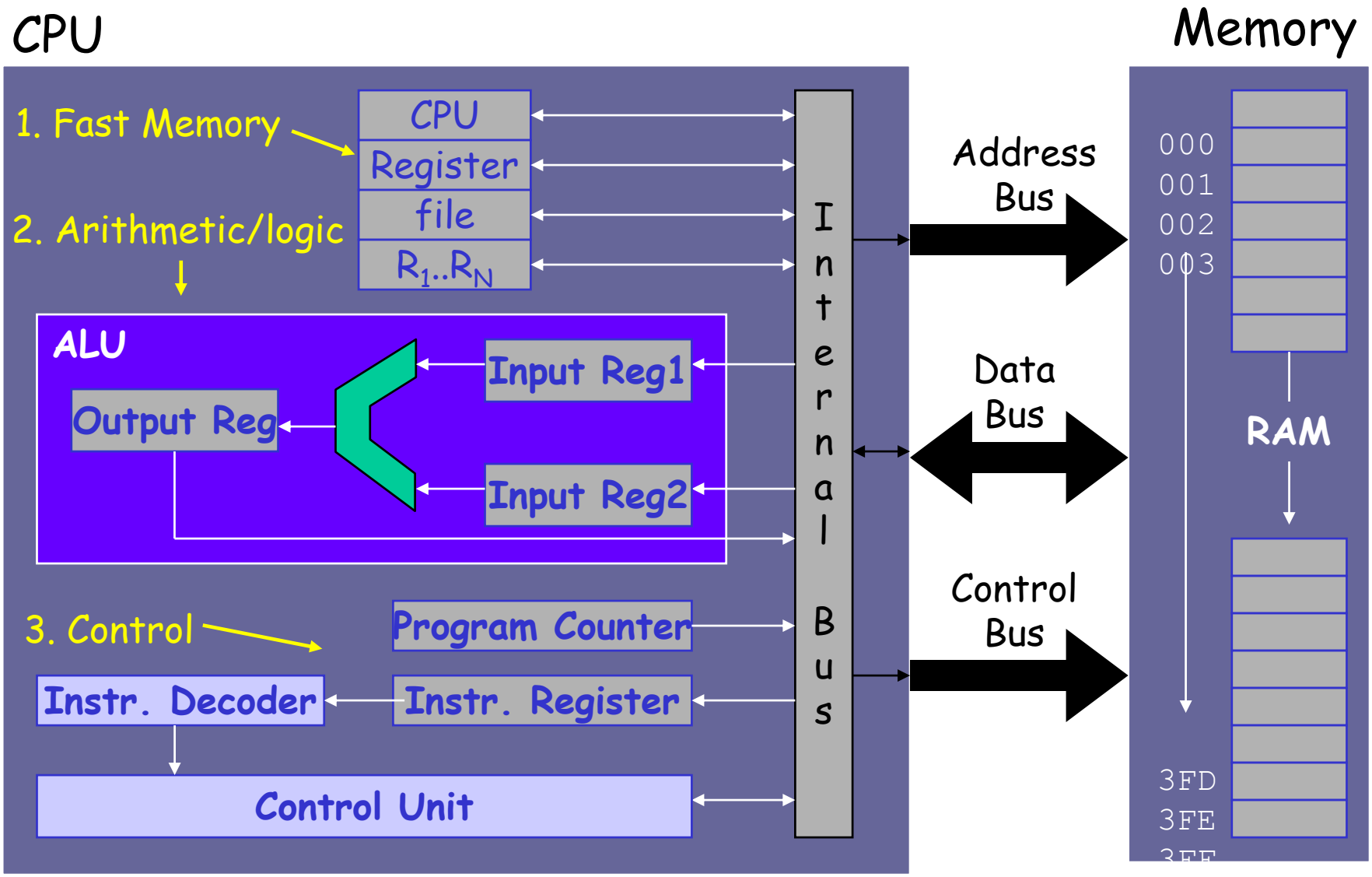
Story so far

- simplest description of computer?
- what makes it general purpose?

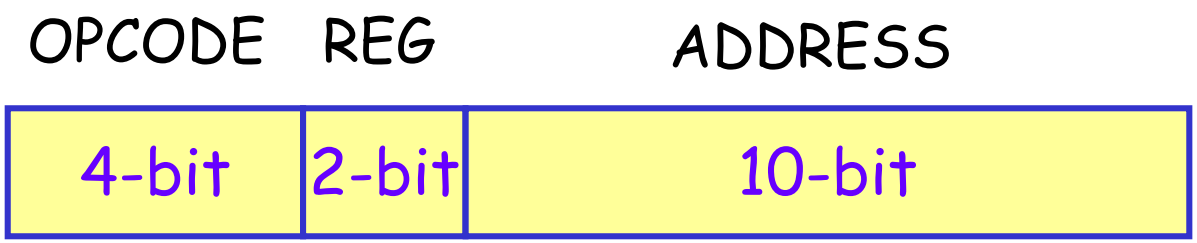


- this lecture: internal structure of the CPU
  - how it relates to the instruction set

CPU organisation: overview



TOY1 instruction set



Opcode	Assembly Instruction	Action
0000	STOP	Stop Program Execution
0001	LOAD Rn, [Addr]	Rn = Memory [Addr] (a)
0010	STORE Rn, [Addr]	Memory [Addr] = Rn
0011	ADD Rn, [Addr]	Rn = Rn + Memory [Addr] (b)
0100	SUB Rn, [Addr]	Rn = Rn - Memory [Addr]
0101	GOTO Addr	PC = Addr
0110	IFZER Rn, Addr	IF Rn == 0 THEN PC = Addr
0111	IFNEG Rn, Addr	IF Rn < 0 THEN PC = Addr

if opcode=LOAD
then Rn = (a)
if opcode=ADD
then Rn = (b)

Micro-steps: LOAD R2,[201]

Control Unit Action FETCH INSTRUCTION[12]				
PC to Address Bus[13]	080H	→	080H	Address Bus
0 to Control Bus[14]	0	→	0	Control Bus
Address Bus to Memory	080H	→	080H	Memory
Control Bus to Memory	0	READ →	0	Memory
Increment PC[15]	080	INC →	081H	PC becomes PC+1[16]
Memory [080H] to Data Bus	1A01H	→	1A01H	Data Bus
Data Bus to Instruction Register	1A01H	→	1A01H	Instruction Register

DECODE INSTRUCTION				
IR to Instruction Decoder	1A01H	→	1A01H	Instruction Decoder
Instruction Decoder to Control Unit[17]	1, 2, 201H	→	1, 2, 201H	Control Unit

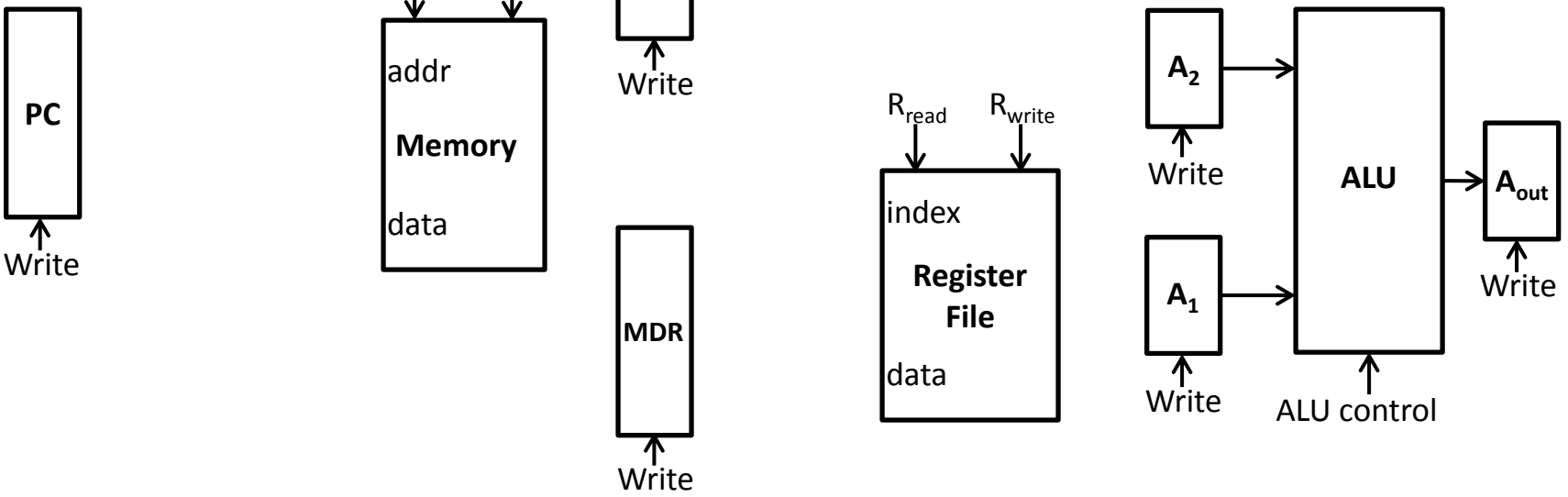
EXECUTE INSTRUCTION[18]				
Control Unit to Address Bus	201H	→	201H	Address Bus
0 to Control Bus	0	→	0	Control Bus
Address Bus to Memory	201H	→	201H	Memory
Control Bus to Memory	0	READ →	0	Memory
Memory [201H] to Data bus	0009H	→	0009H	Data Bus
Data Bus to Register 2	0009H	→	0009H	Register 2

Understand CPU structure

- step by step
  - step 1: just main elements, no connections, no control
  - step 2: add control elements, e.g. multiplexers (mux)
  - step 3: derive control signals: next lecture
- Register Transfer Level (RTL) description: simple
  - 6 registers: PC, IR, A_1, A_2, A_{out}, MDR (memory data register)
  - 2 components respond in same cycle: UX, ALU, combinational
  - 2 components have Read and Write signals: Memory, Register file
- circuit diagram (see notes on *basic logic design*)
  - with or without control elements (e.g. mux)
  - with or without control unit

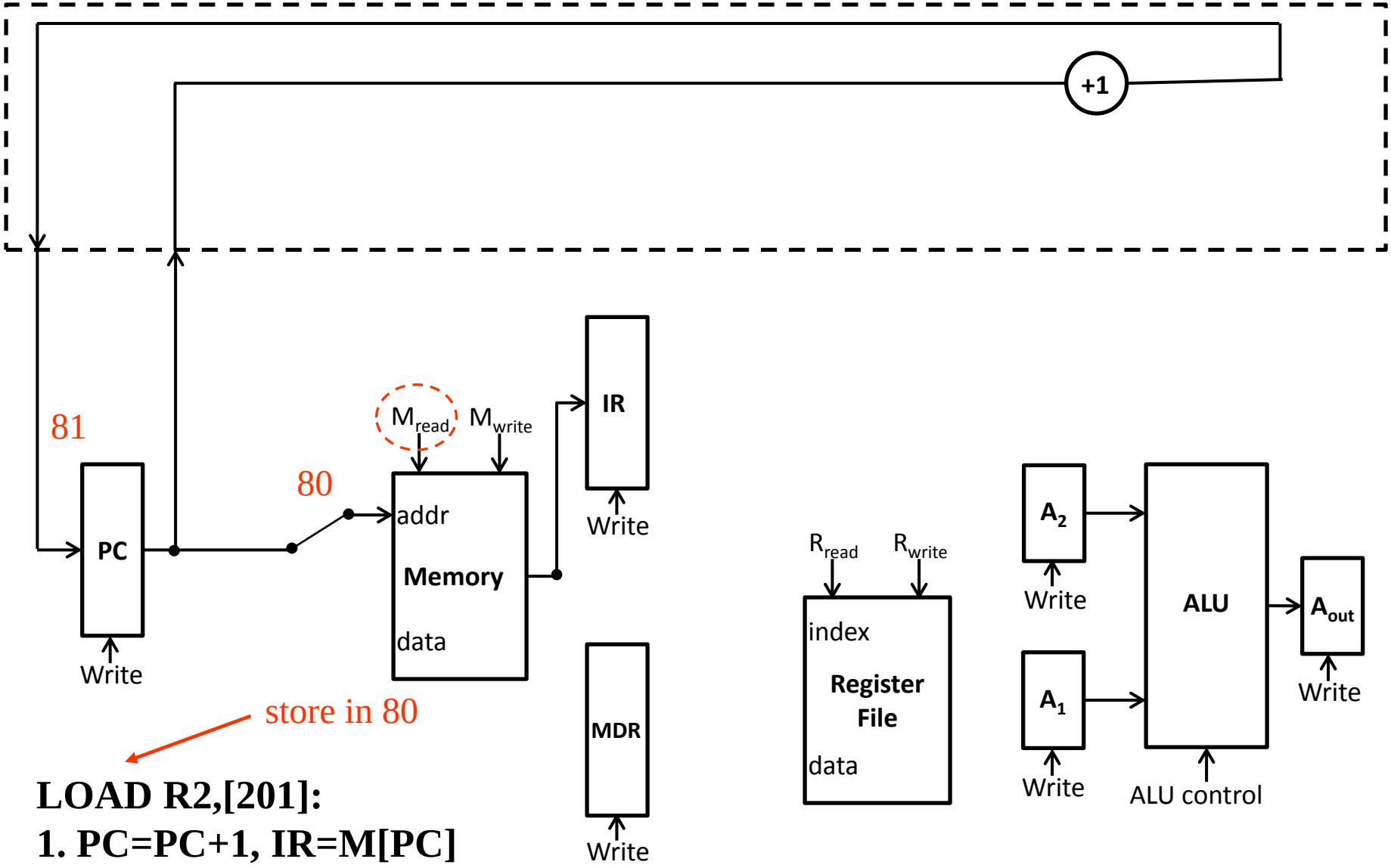
Abstraction: include the minimum

Control and next instruction logic



if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal



LOAD R2,[201]:

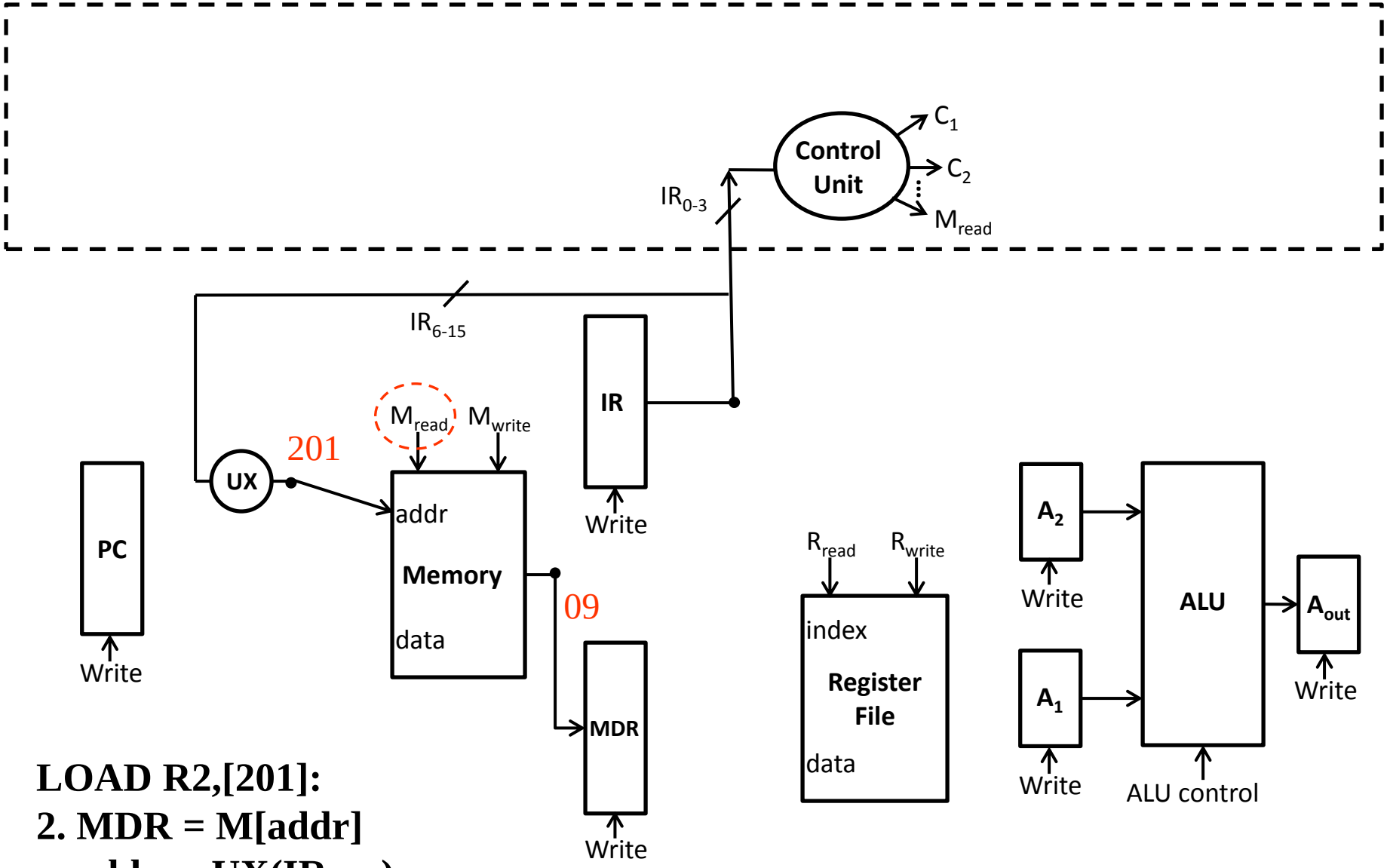
1. PC=PC+1, IR=M[PC]

if ALU Control = 0 then $A_{out} = A_1 + A_2$

if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals

1 ALU Control signal



LOAD R2,[201]:

2. MDR = M[addr]

addr = UX($IR_{6..15}$)

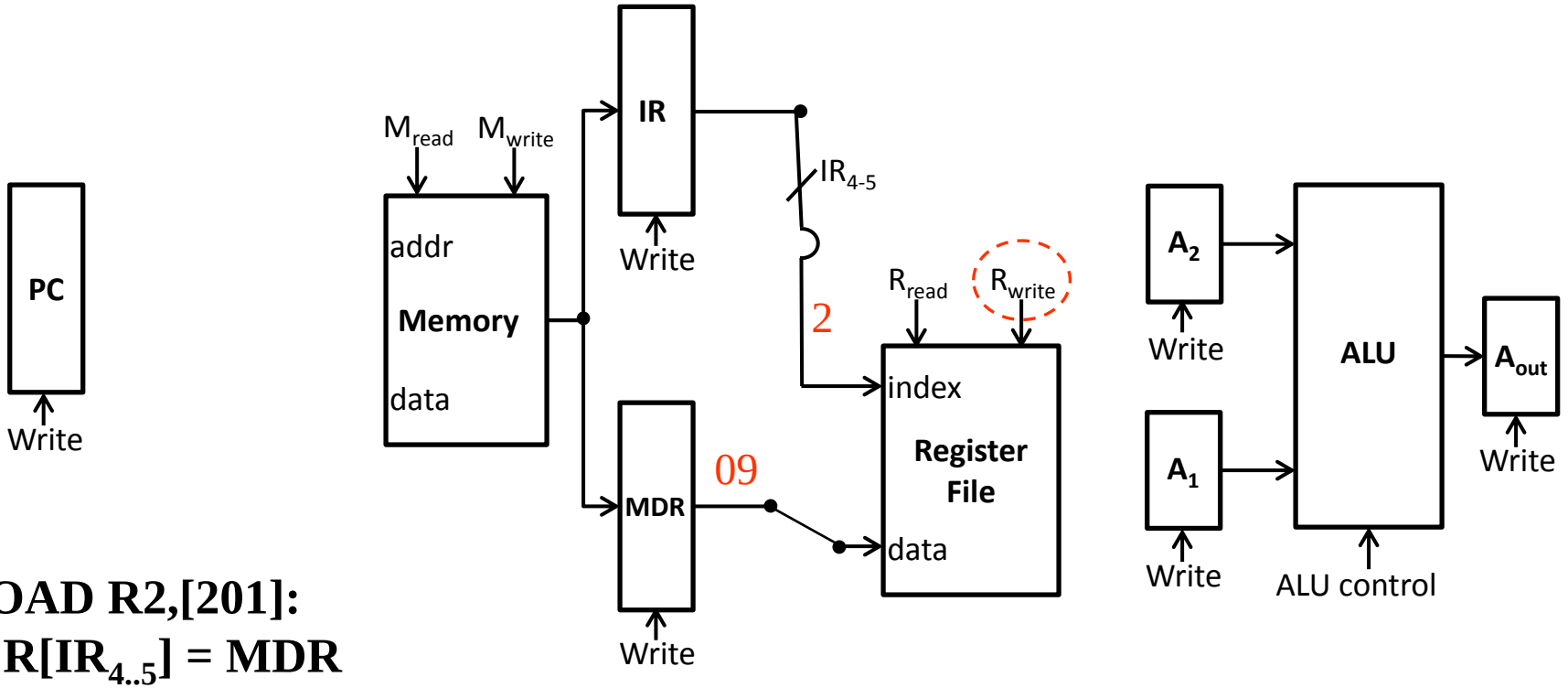
UX: Unsign eXtension from 10 to 16 bits

if ALU Control = 0 then $A_{out} = A_1 + A_2$

if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals

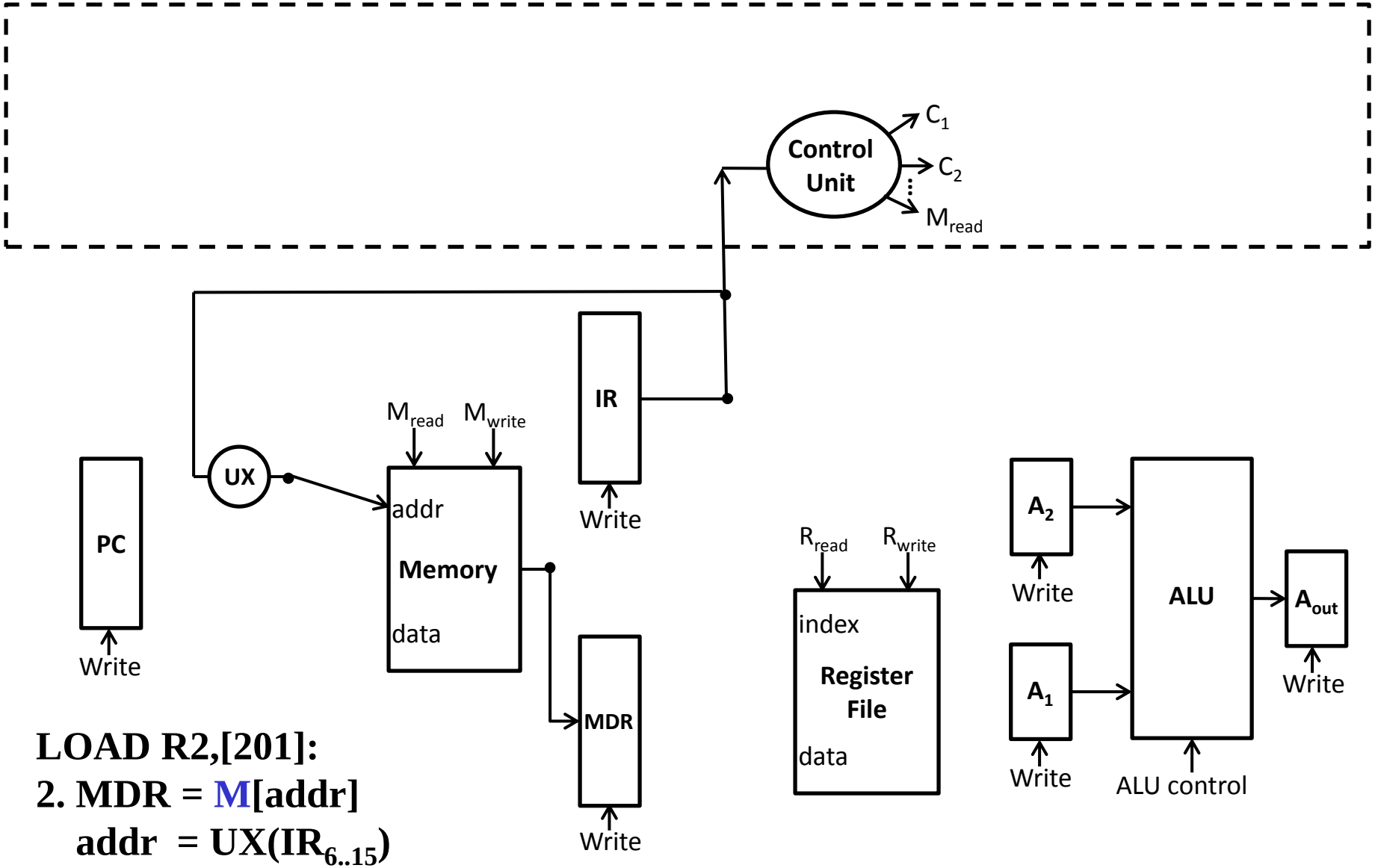
1 ALU Control signal



LOAD R2,[201]:
3. R[$IR_{4..5}$] = MDR

if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal



LOAD R2,[201]:

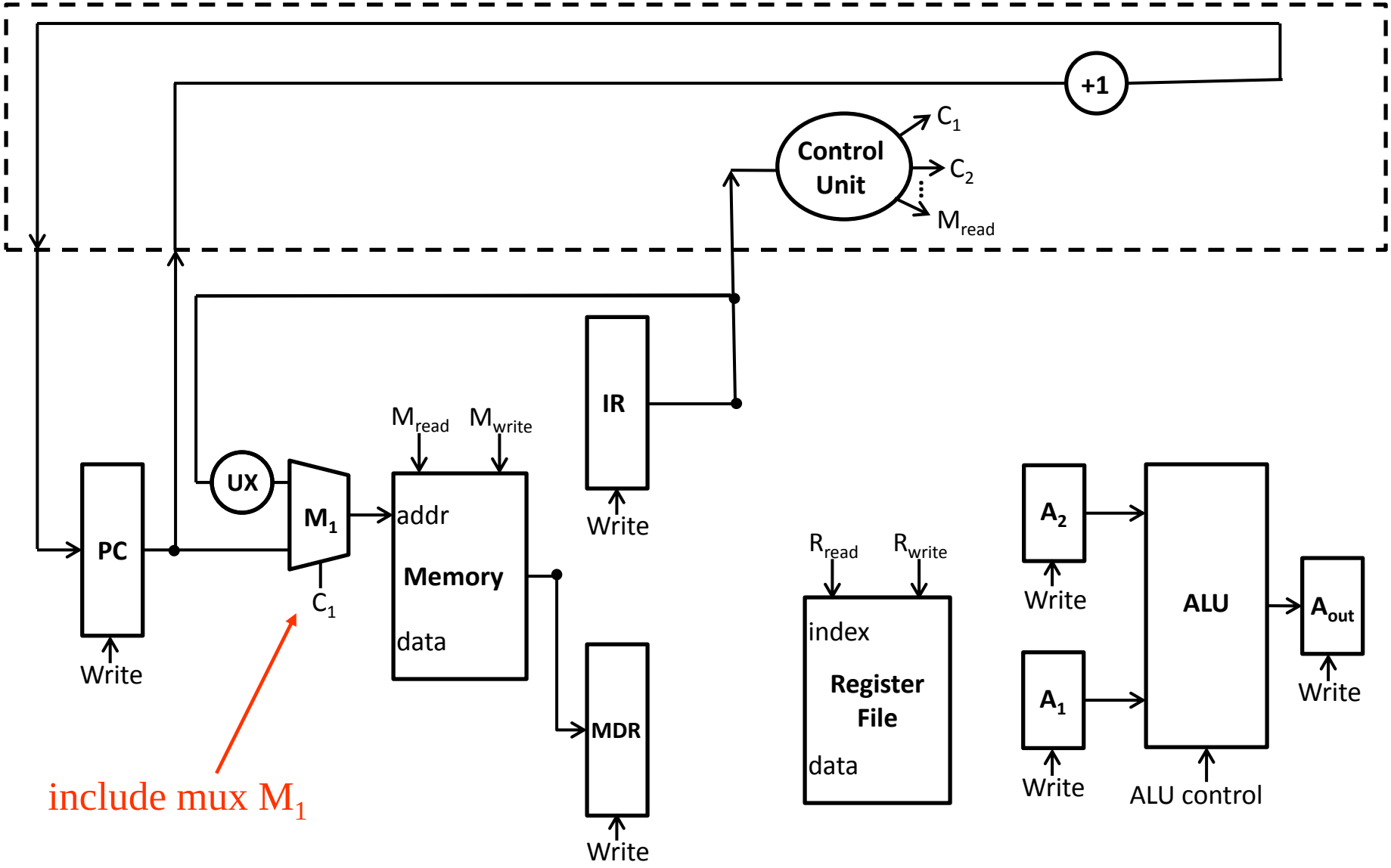
2. MDR = M[addr]

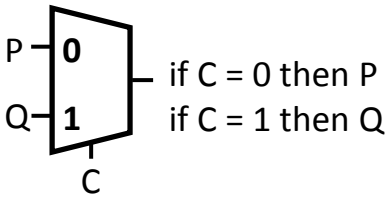
addr = UX($IR_{6..15}$)

but step 1: IR = M[PC]
so need mux

if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal





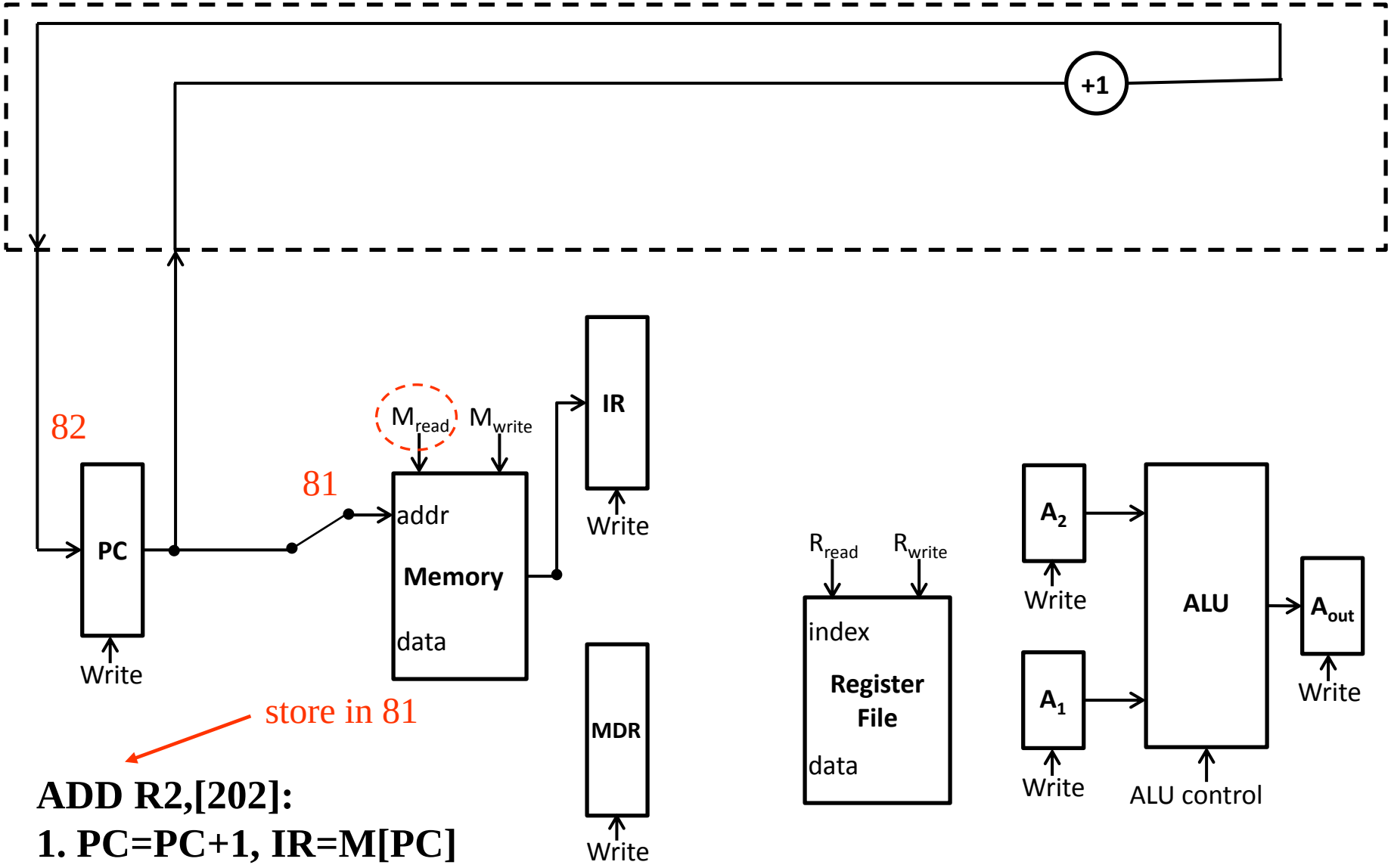
if C = 0 then P
if C = 1 then Q

if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal
1 multiplexer signal

Micro-steps: ADD R2,[202]

Control Unit Action				
FETCH INSTRUCTION				
PC to Address Bus	081H	→	081H	Address Bus
0 to Control Bus	0	→	0	Control Bus
Address Bus to Memory	081H	→	081H	Memory
Control Bus to Memory	0	READ →	0	Memory
Increment PC	081H	INC →	082H	PC becomes PC+1
Memory [081H] to Data Bus	3A02H	→	3A02H	Data Bus
Data Bus to Instruction Register	3A02H	→	3A02H	Instruction Register
DECODE INSTRUCTION				
IR to Instruction Decoder	3A02H	→	3A02H	Instruction Decoder
Instruction Decoder to Control Unit	3, 2, 202H	→	3, 2, 202H	Control Unit
EXECUTE INSTRUCTION				
Register 2 to ALU Input Reg 1	0009	→	0009	ALU Input Reg 1
Control Unit to Address Bus	202H	→	202H	Address Bus
0 to Control Bus	0	→	0	Control Bus
Address Bus to Memory	202H	→	202H	Memory
Control Bus to Memory	0	READ →	0	Memory
Memory [202H] to Data bus	0006H	→	0006H	Data Bus
Data Bus to ALU Input Reg 2	0006H	→	0006H	ALU Input Reg 2
Control Unit to ALU		ADD →	000FH	Output Register
ALU Output Reg to Register 2	000F	→	000FH	Register 2



ADD R2,[202]:

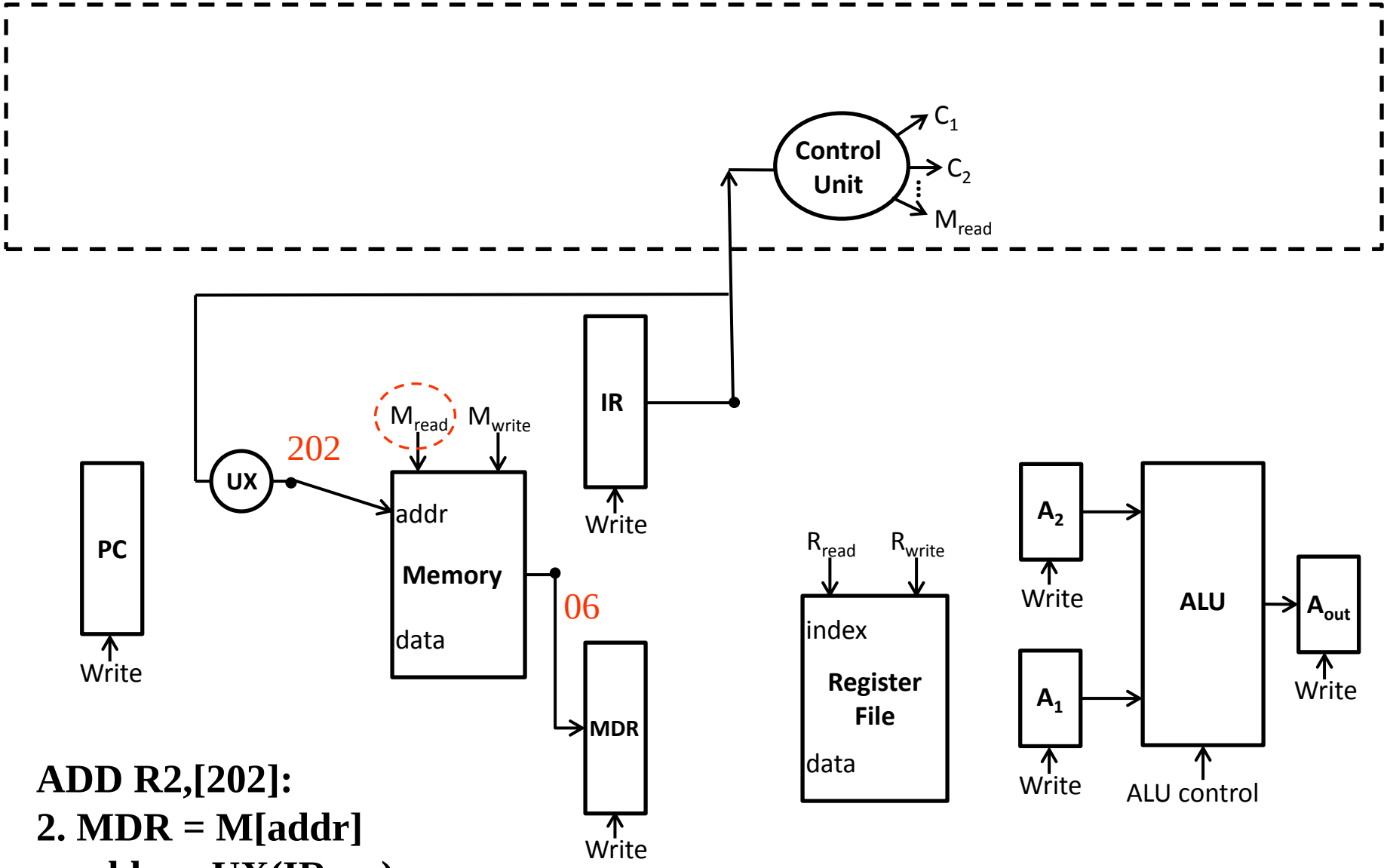
1. PC=PC+1, IR=M[PC]

if ALU Control = 0 then $A_{out} = A_1 + A_2$

if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals

1 ALU Control signal



ADD R2,[202]:

2. MDR = M[addr]

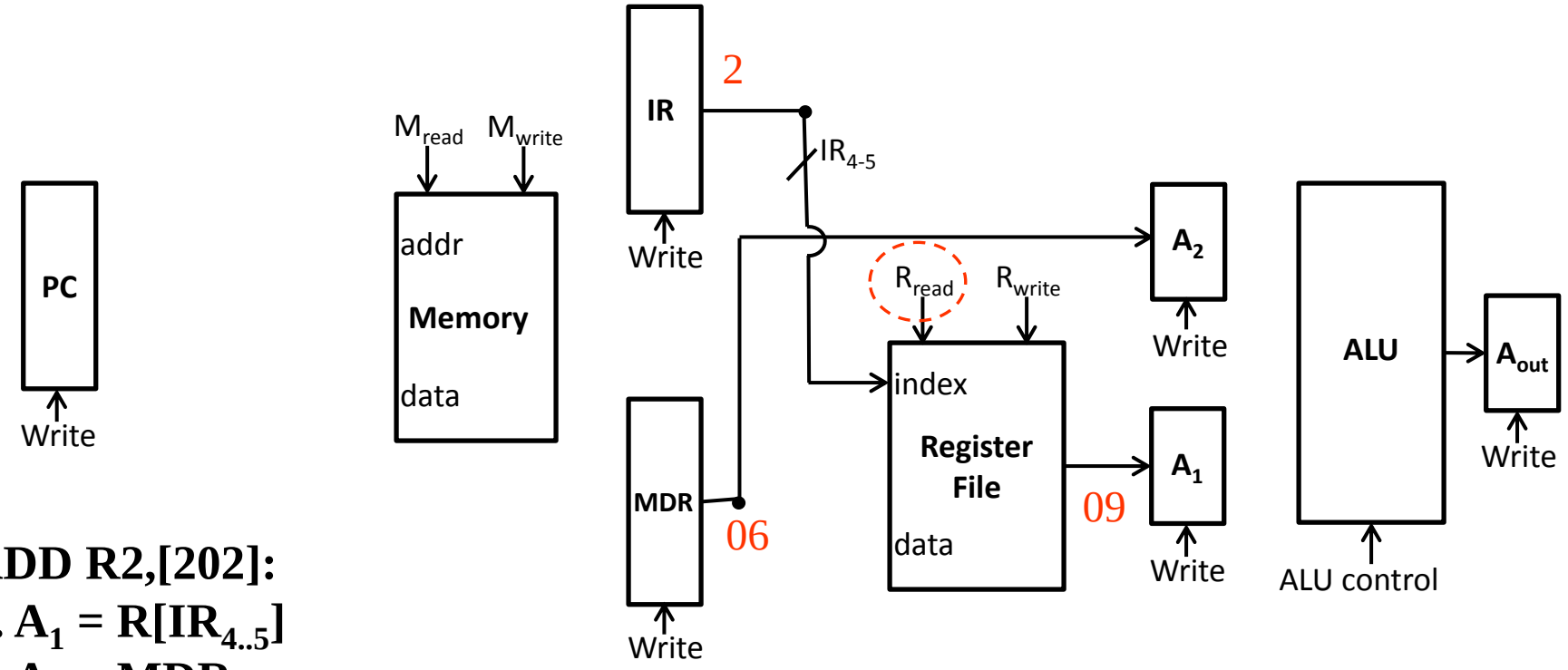
addr = UX($IR_{6..15}$)

if ALU Control = 0 then $A_{out} = A_1 + A_2$

if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals

1 ALU Control signal



ADD R2,[202]:

3. $A_1 = R[IR_{4..5}]$

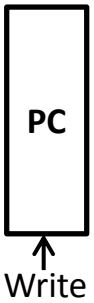
$A_2 = MDR$

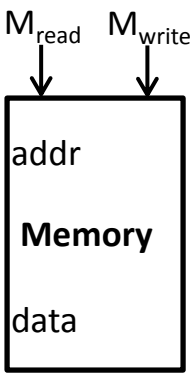
if ALU Control = 0 then $A_{out} = A_1 + A_2$

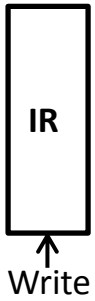
if ALU Control = 1 then $A_{out} = A_1 - A_2$

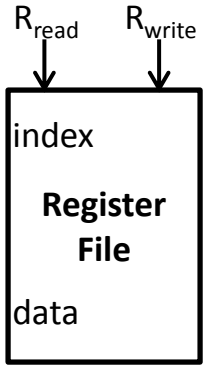
10 read/write signals

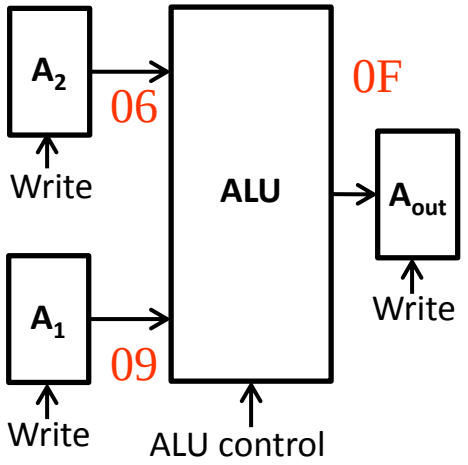
1 ALU Control signal









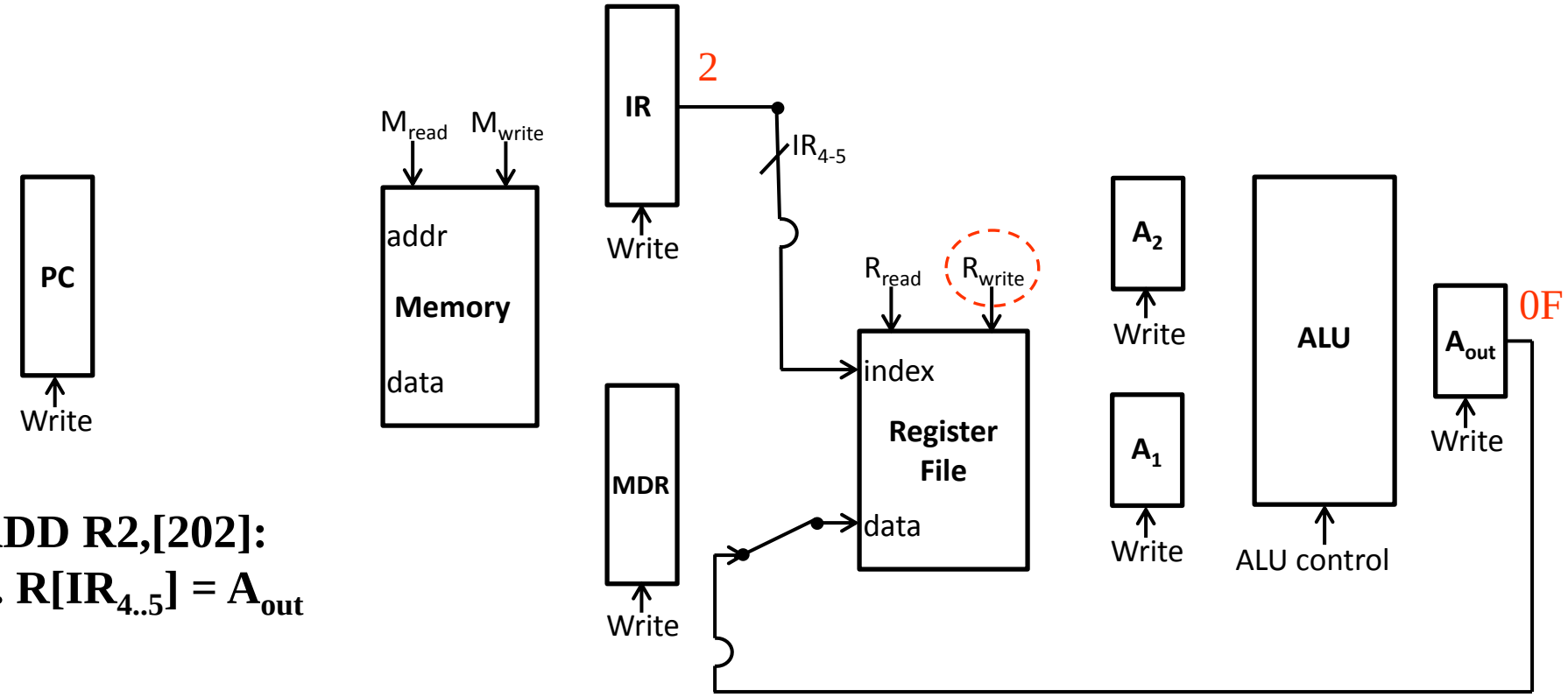


ADD R2,[202]:

4. $A_{out} = A_1 + A_2$

if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal



ADD R2,[202]:

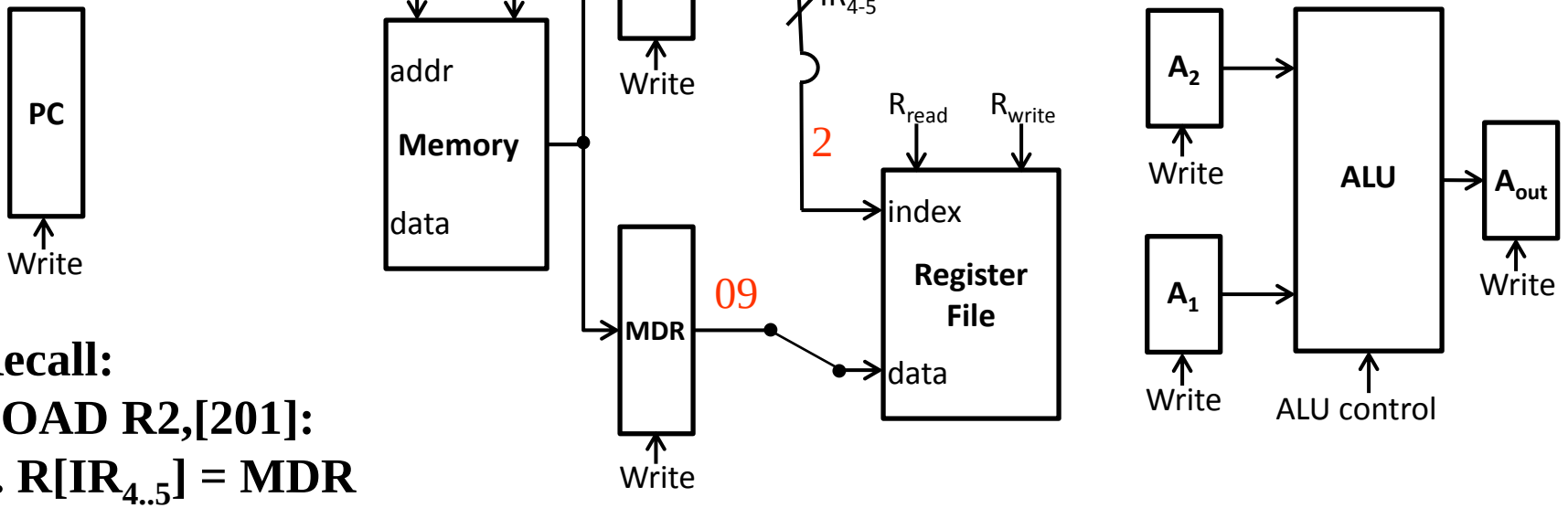
5. $R[IR_{4..5}] = A_{out}$

if ALU Control = 0 then $A_{out} = A_1 + A_2$

if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals

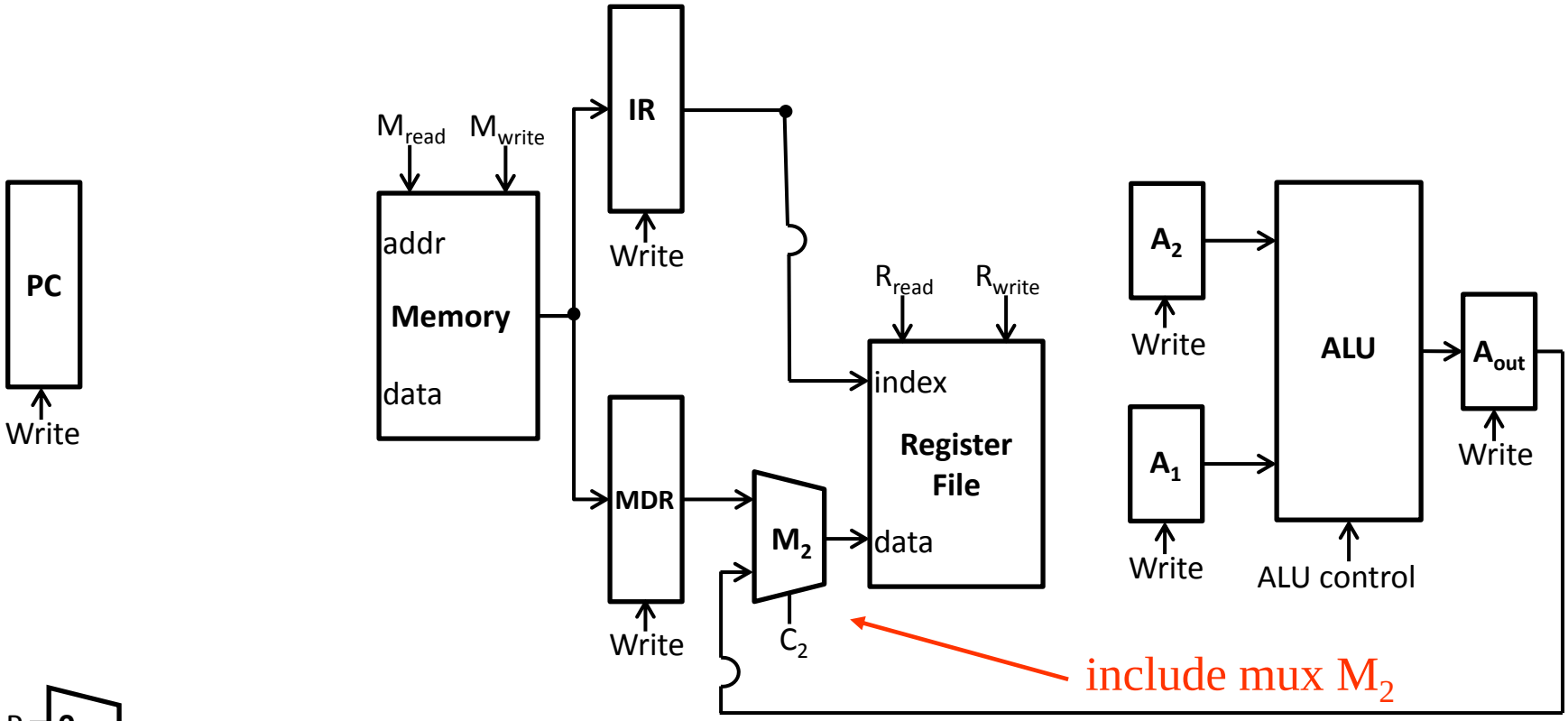
1 ALU Control signal

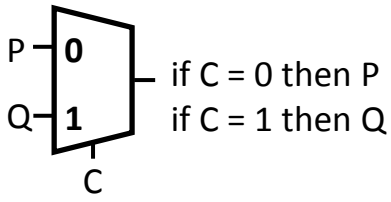


Recall:
LOAD R2,[201]:
3. R[$IR_{4..5}$] = MDR
and
ADD R2,[202]:
5. R[$IR_{4..5}$] = A_{out}

if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal

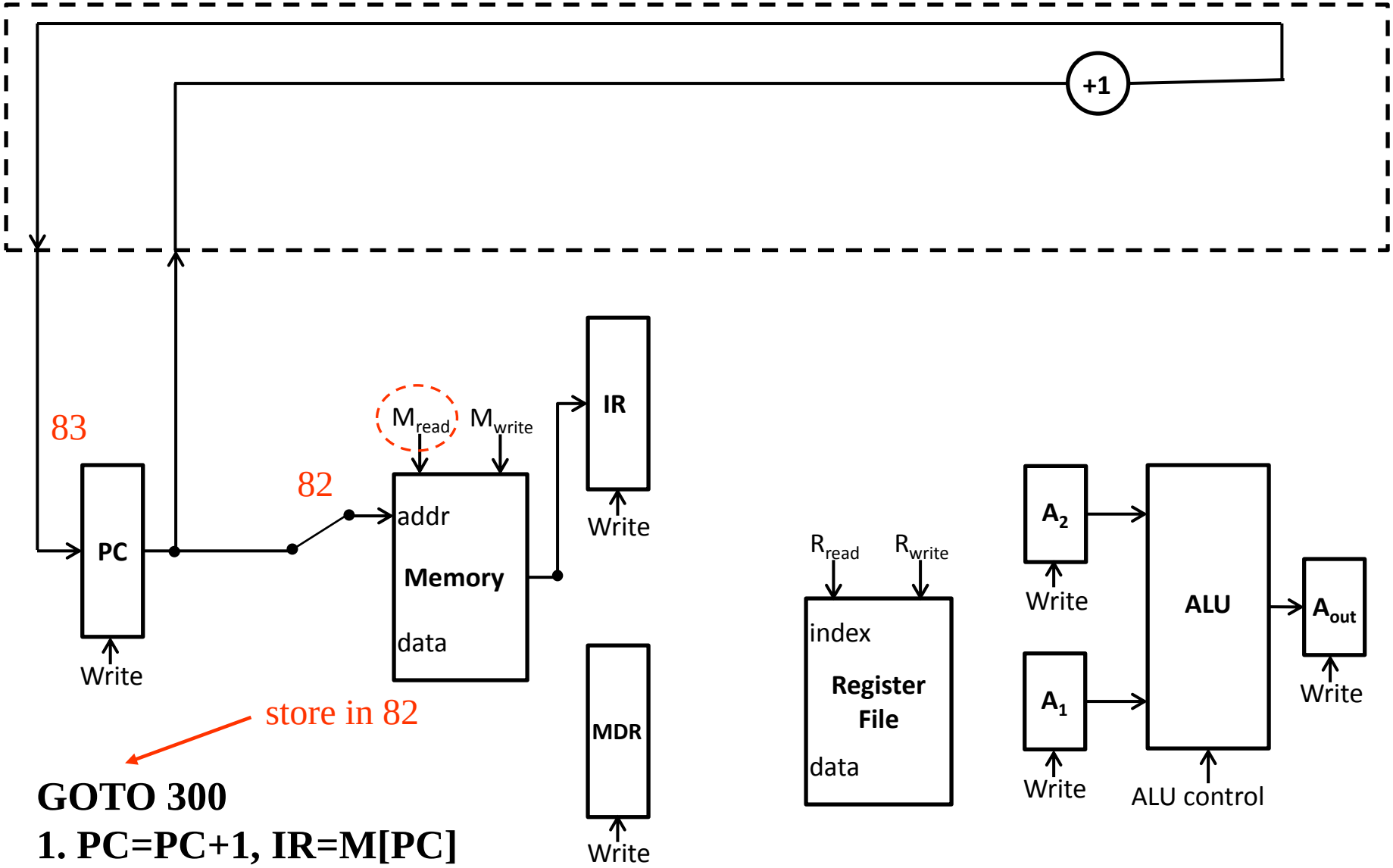




if C = 0 then P
if C = 1 then Q

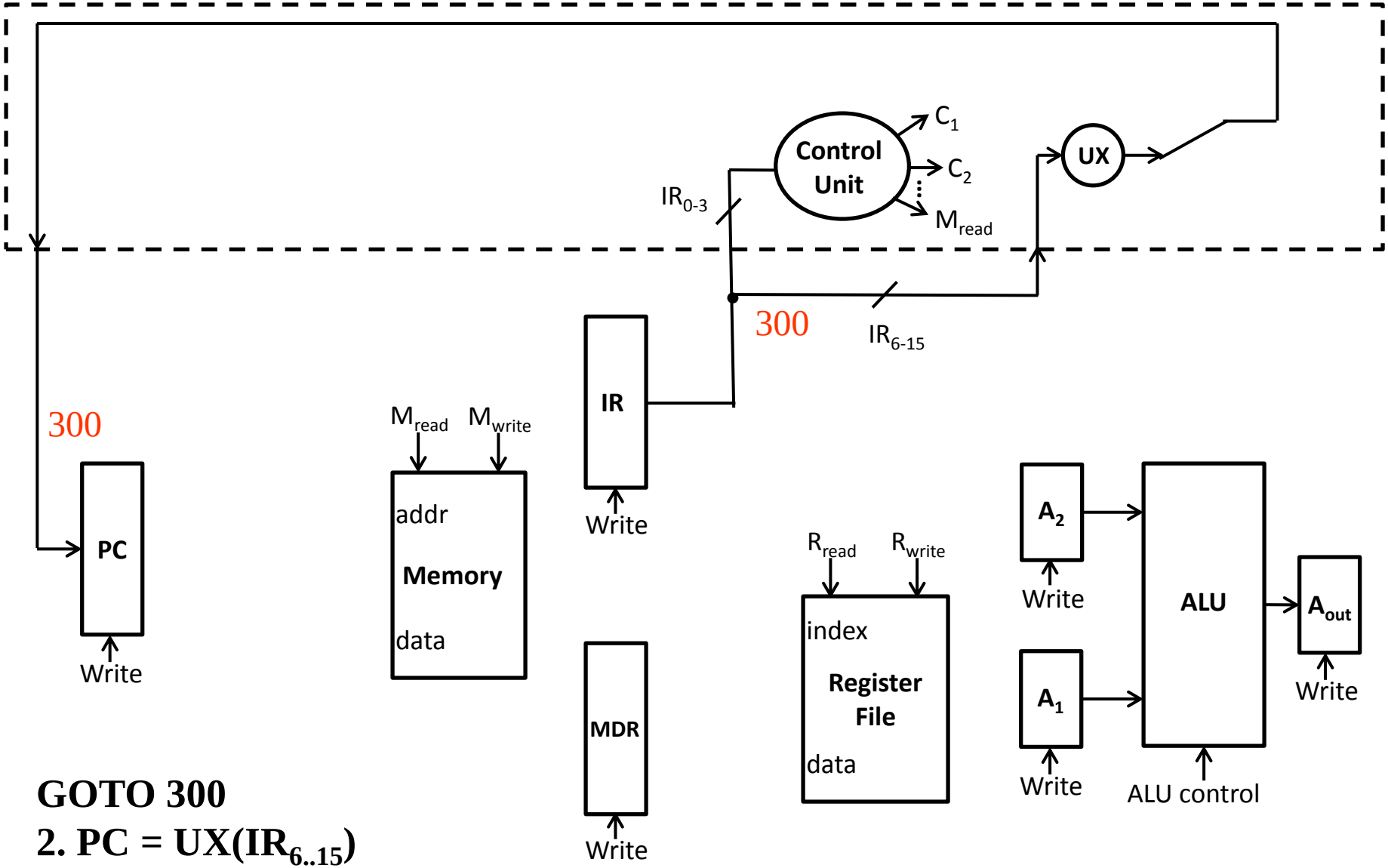
if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal
1 multiplexer signal



if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal



GOTO 300

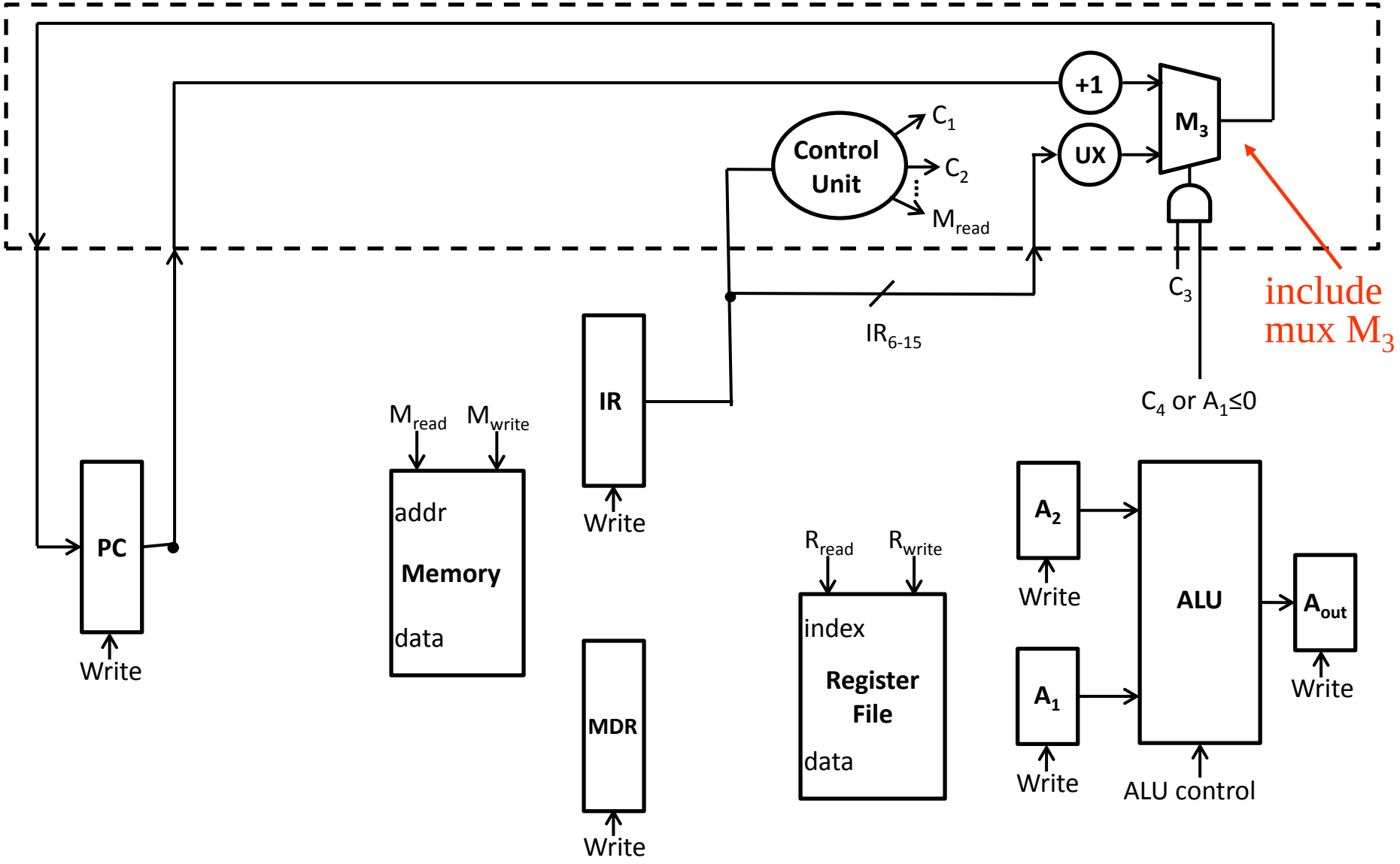
2. PC = UX($IR_{6..15}$)

if ALU Control = 0 then $A_{out} = A_1 + A_2$

if ALU Control = 1 then $A_{out} = A_1 - A_2$

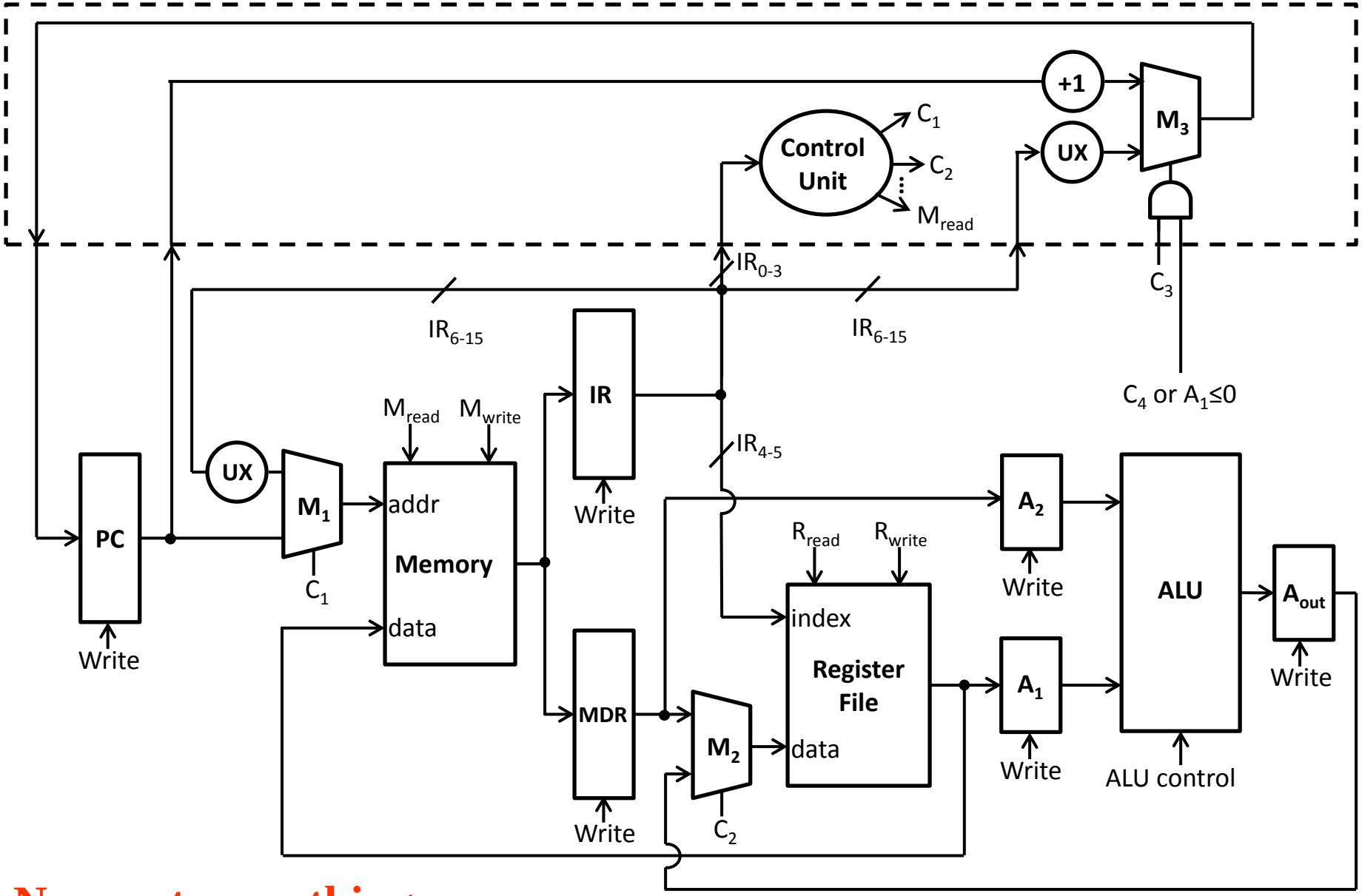
10 read/write signals

1 ALU Control signal



if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal
2 multiplexer signals



Now put everything together!

if ALU Control = 0 then $A_{out} = A_1 + A_2$
if ALU Control = 1 then $A_{out} = A_1 - A_2$

10 read/write signals
1 ALU Control signal
4 multiplexer signals