

Single Cycle Processor Examples

Last updated 4/14/21

Single Cycle Processor - Examples

- Code the instruction for `add RA, RB, RA` $RA \leftarrow RA + RB$

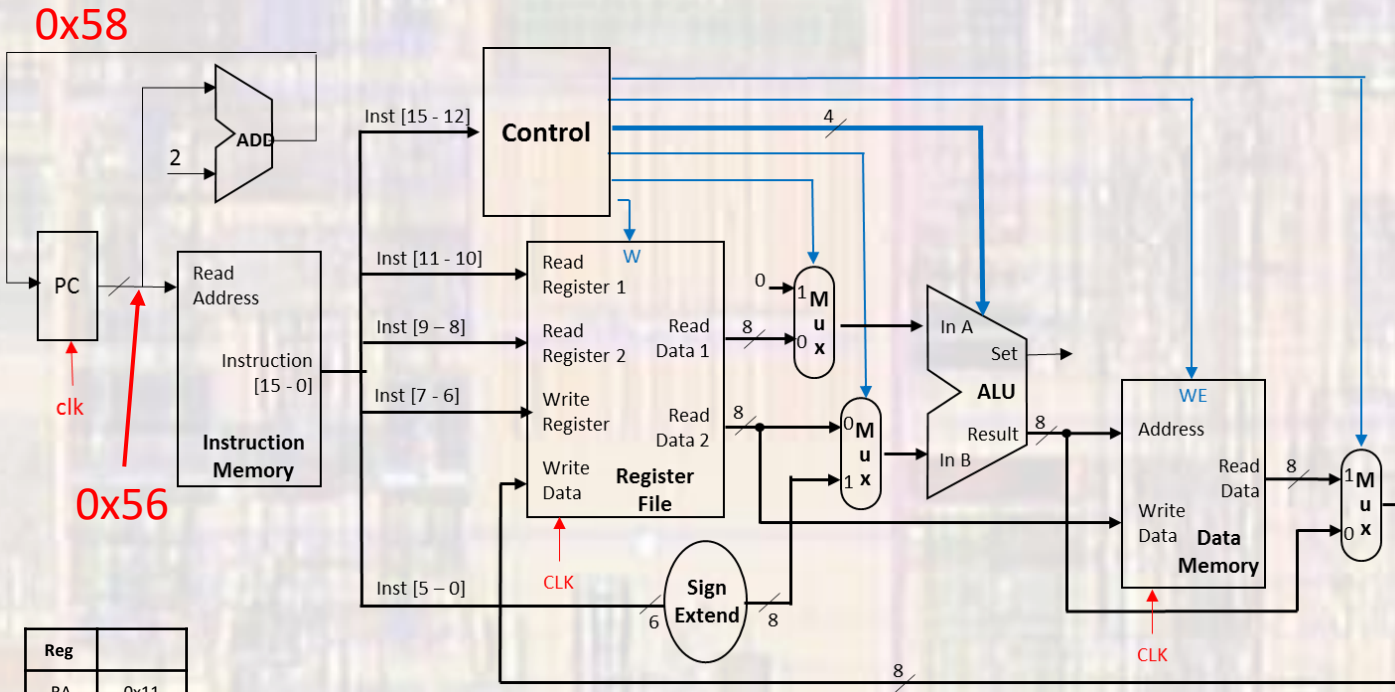
or	0000			
and	0001			
nor	0010			
nand	0011			
add	0100			
sub	0101			
slt	0110	00 - A	00 - A	00 - A
ld	1000	01 - B	01 - B	01 - B
st	1001	10 - C	10 - C	10 - C
ldi	1100	11 - D	11 - D	11 - D

Technically these are
don't care
but
we will always code them as 0s

Instruction				Reg 1		Reg 2		W Reg		Immediate Value						
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: **add RA, RB, RA** located in program memory at 0x56



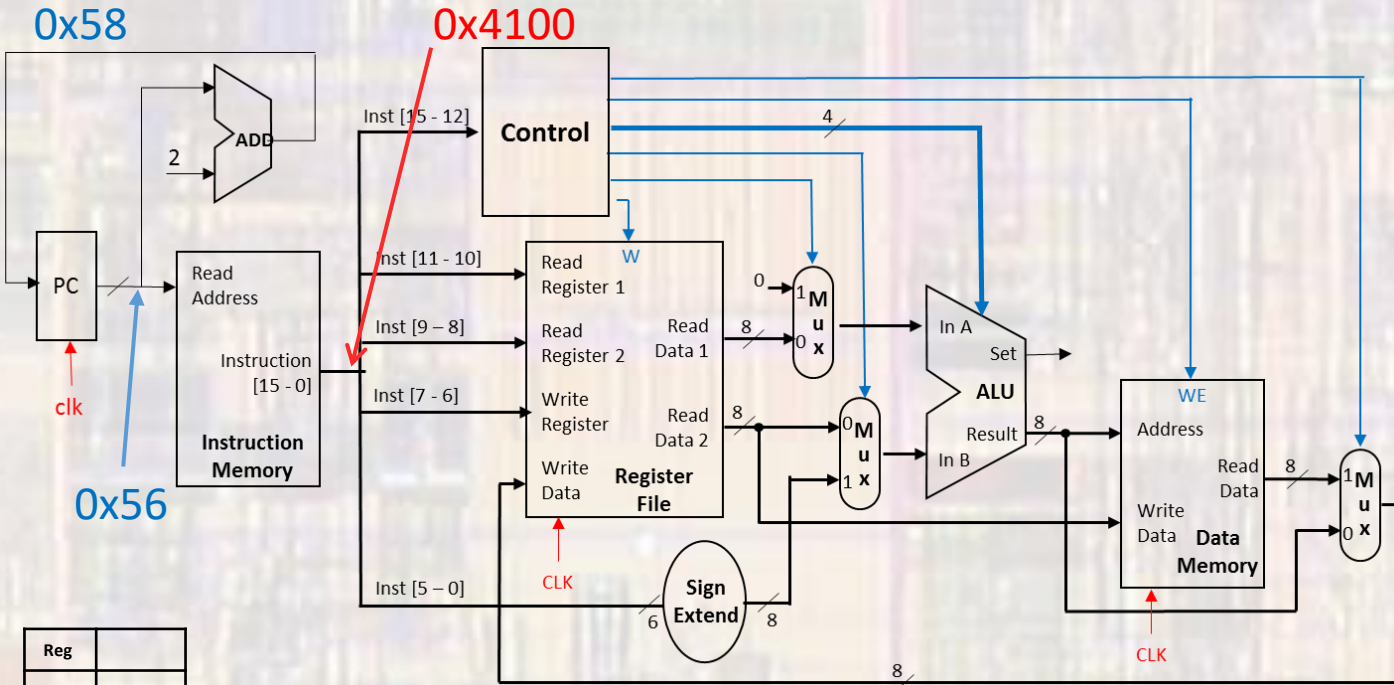
Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **add RA, RB, RA** located in program memory at 0x56

Instruction		Reg 1		Reg 2		W Reg		Immediate Value								
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

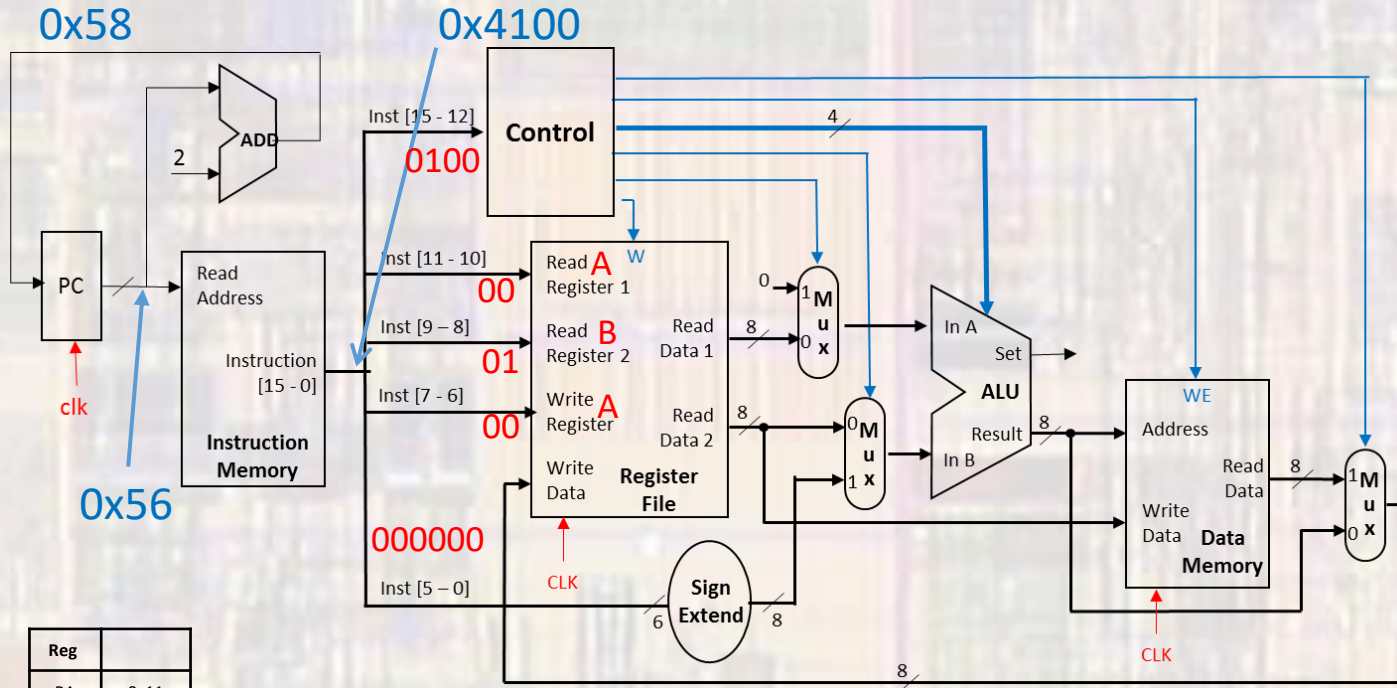


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **add RA, RB, RA** located in program memory at 0x56

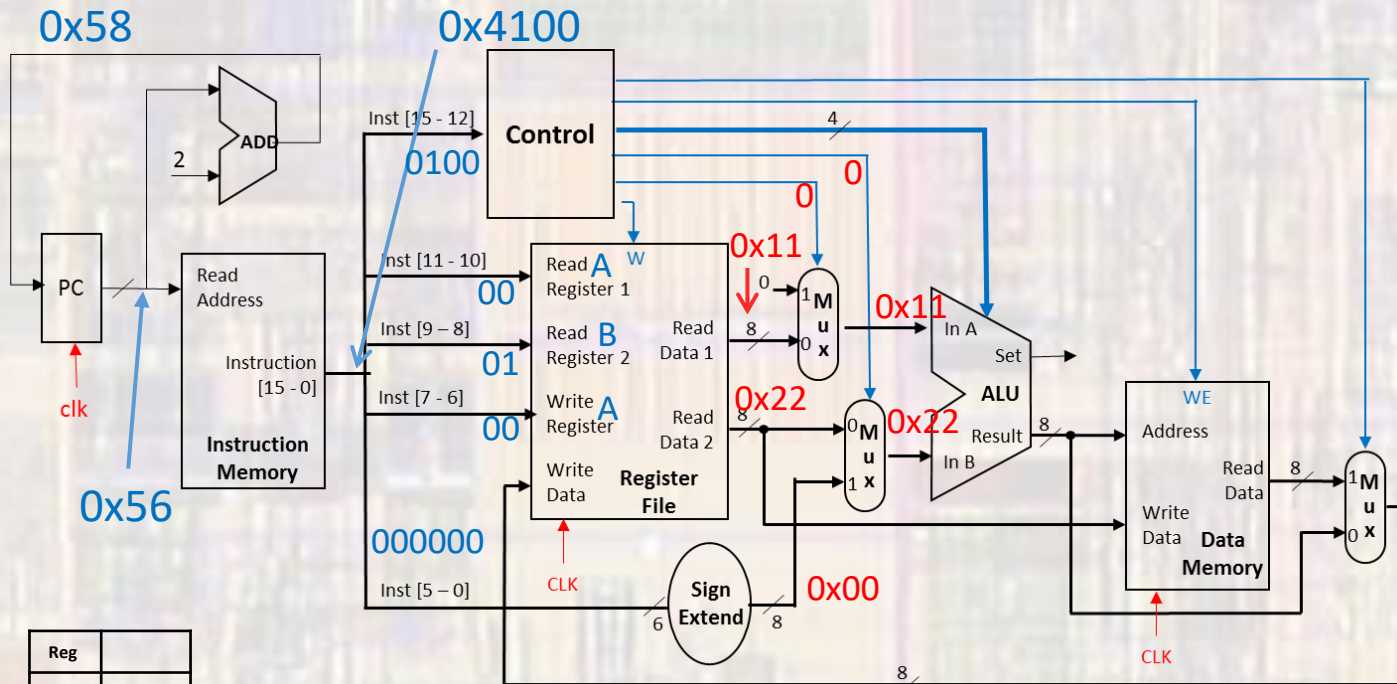


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

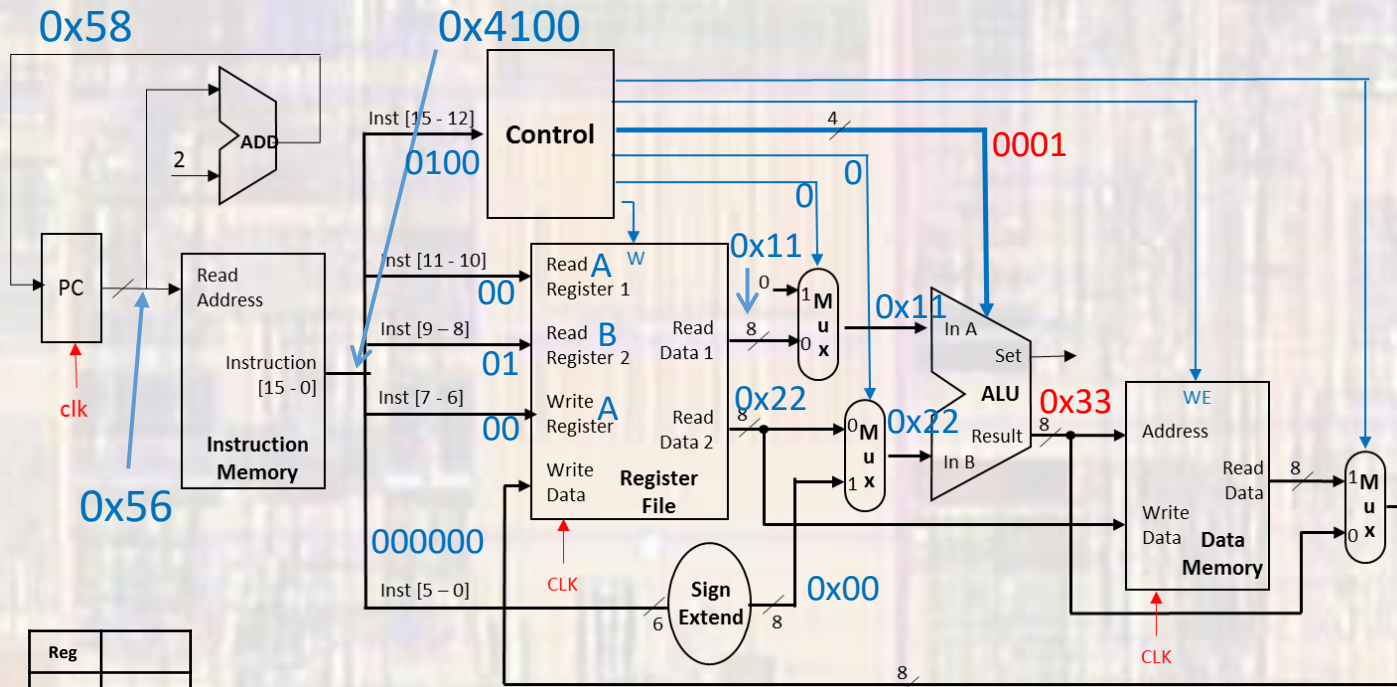
Provide the values for each signal after executing the instruction: `add RA, RB, RA` located in program memory at 0x56



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

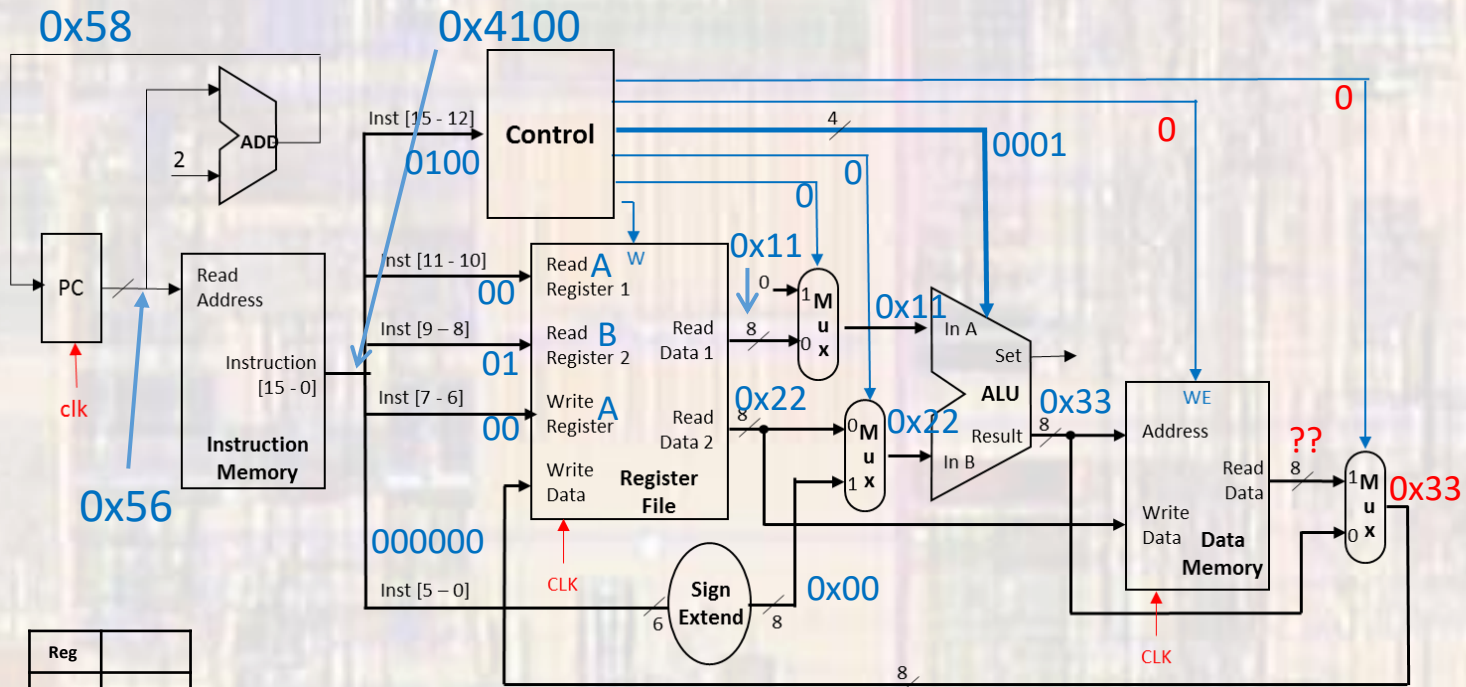
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `add RA, RB, RA` located in program memory at 0x56



Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: `add RA, RB, RA` located in program memory at 0x56

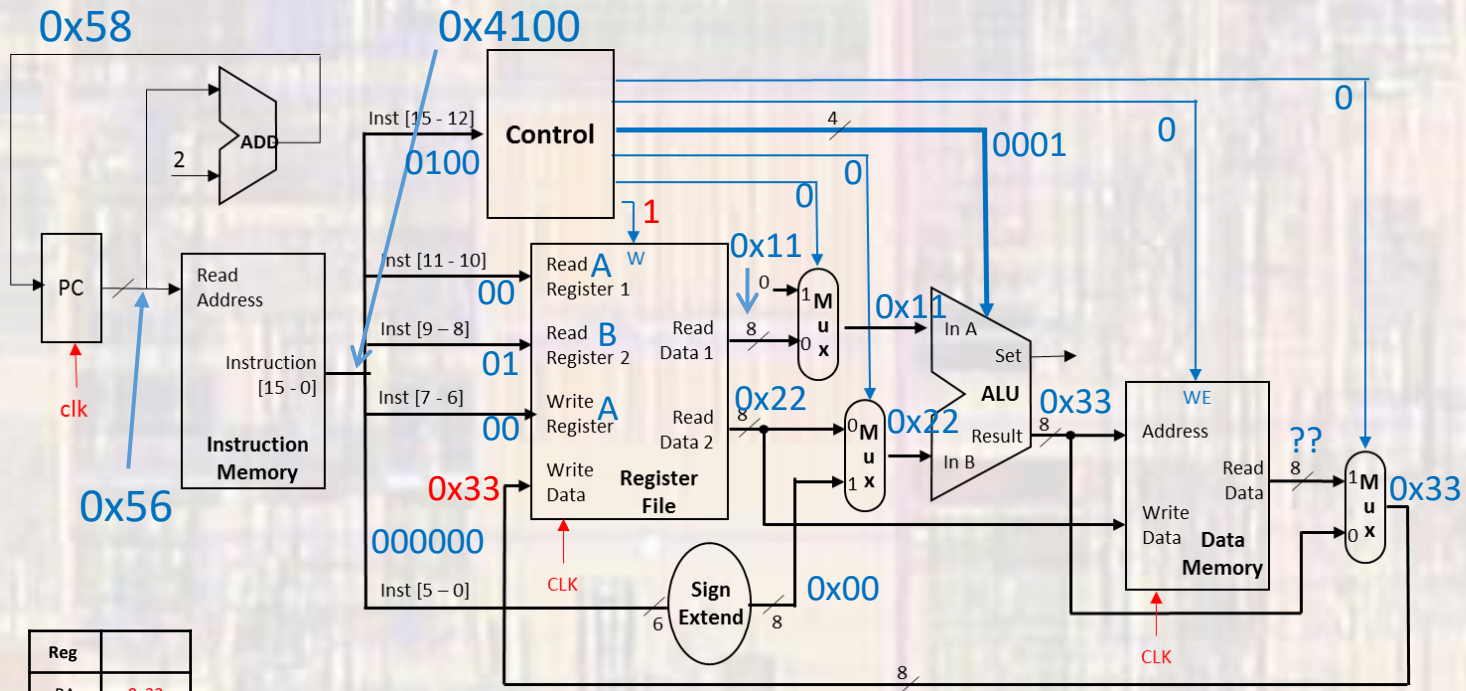


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: `add RA, RB, RA` located in program memory at 0x56

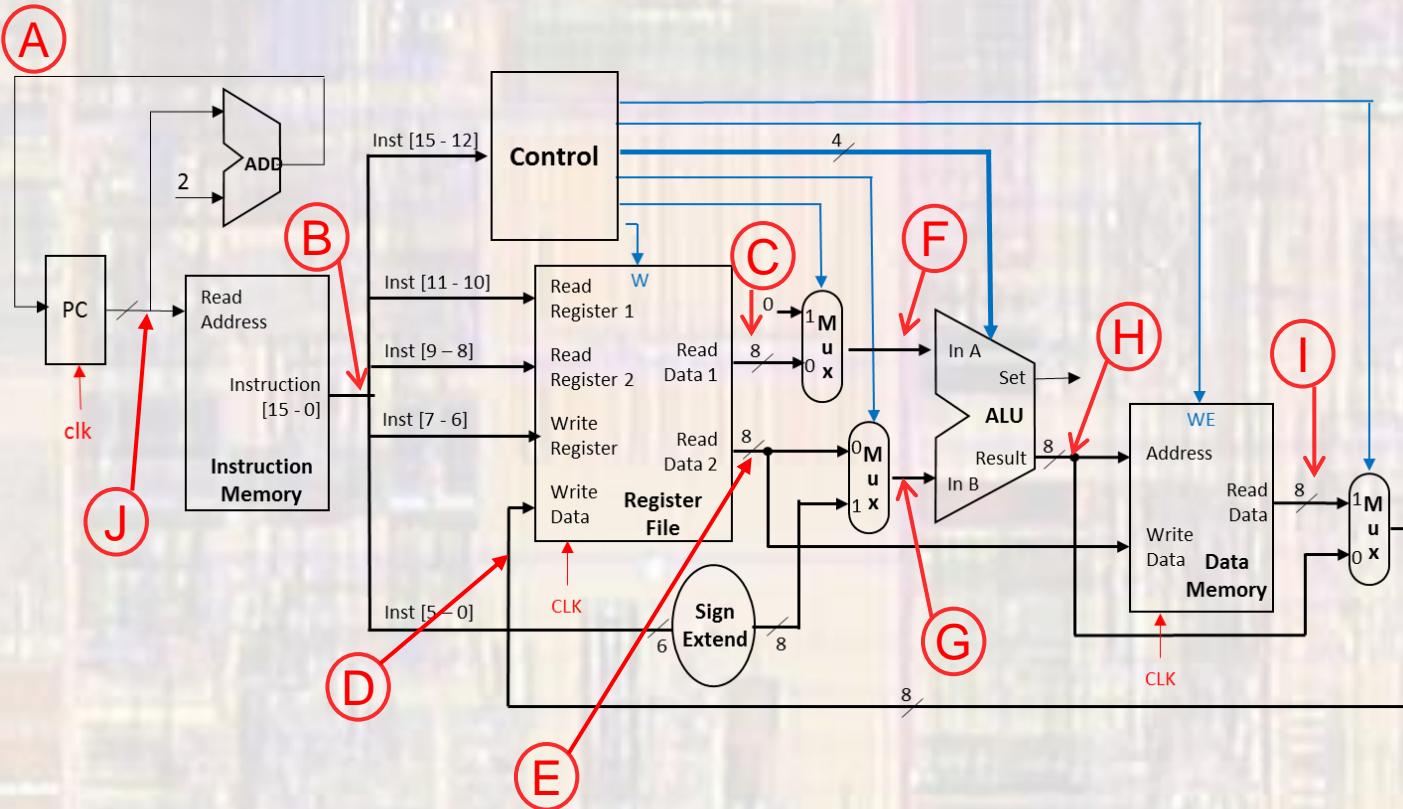


At the next rising clock edge $A \leftarrow 0x33$

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: `add RA, RB, RA` located in program memory at 0x56



Node	Value (hex)
A	0x58
B	0x4100
C	0x11
D	0x33
E	0x22
F	0x11
G	0x22
H	0x33
I	??
J	0x56

Single Cycle Processor - Examples

- Code the instruction for **ldi RD, 24**

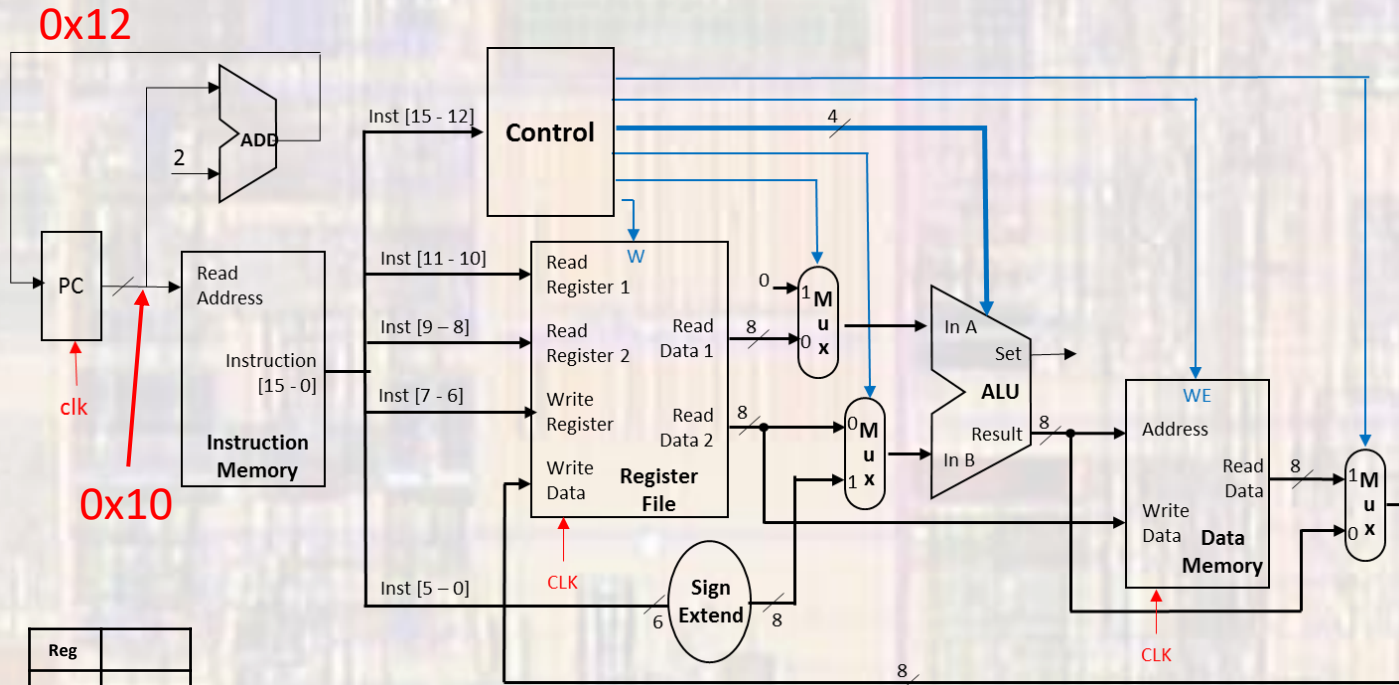
RD ← 24 (0x18)

or	0000			
and	0001			
nor	0010			
nand	0011			
add	0100			
sub	0101			
slt	0110	00 – A	00 – A	00 – A
ld	1000	01 – B	01 – B	01 – B
st	1001	10 – C	10 – C	10 – C
ldi	1100	11 – D	11 – D	11 – D

Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
1	1	0	0	x	x	x	x	1	1	0	1	1	0	0	0

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: **ldi RD, 24** located in program memory at 0x10



0x12

0x10

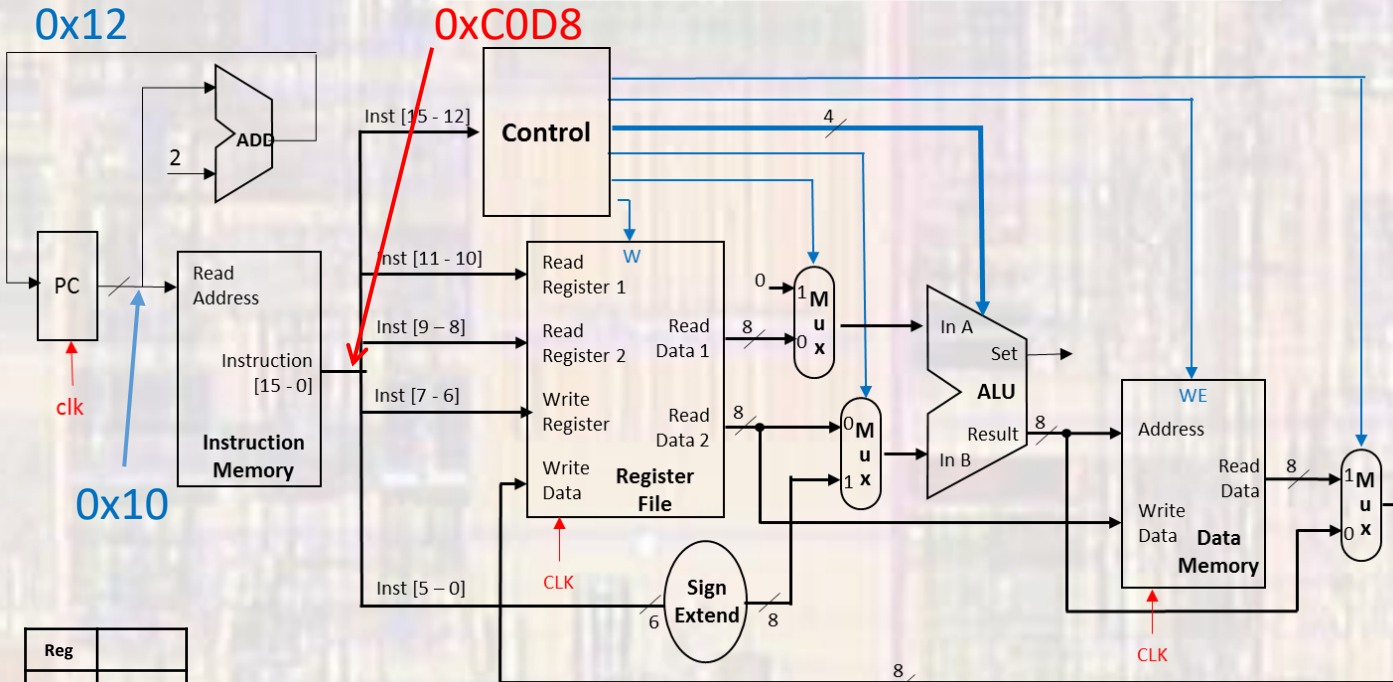
Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: `ldi RD, 24` located in program memory at 0x10

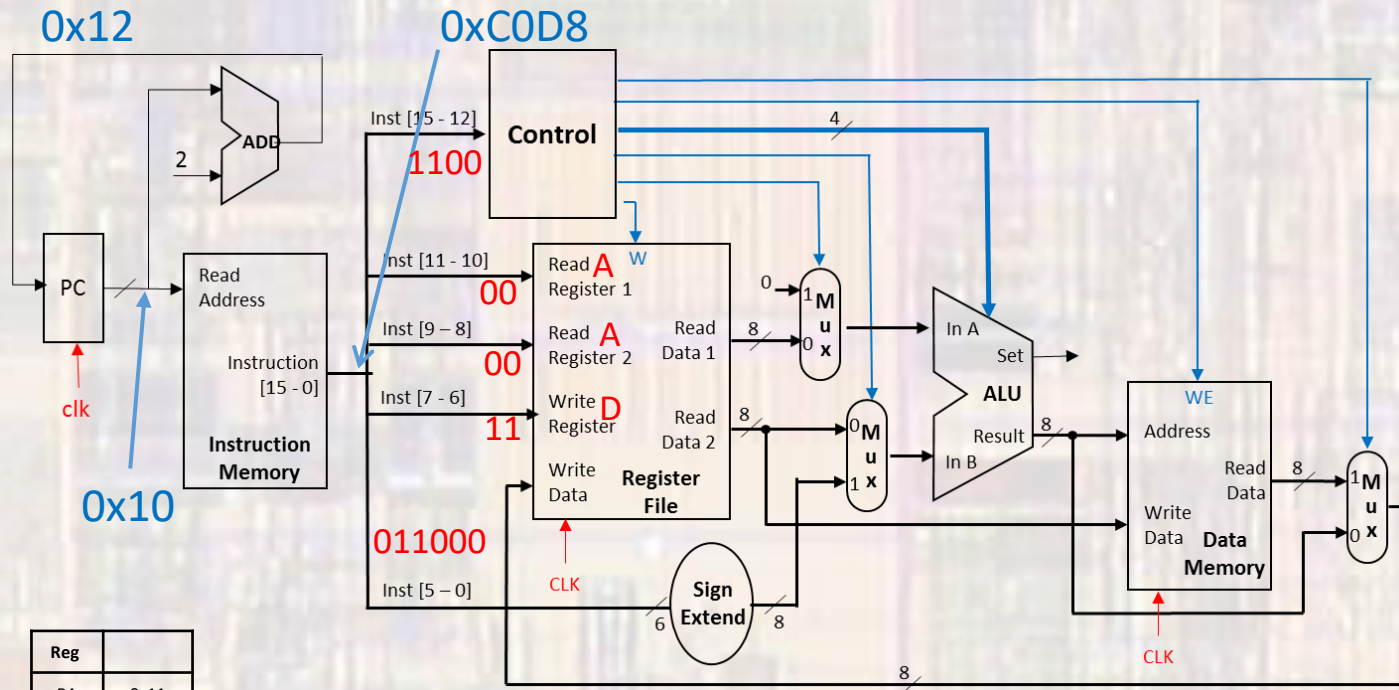
Instruction				Reg 1		Reg 2		W Reg			Immediate Value			
1	1	0	0	x	x	x	x	1	1	0	1	1	0	0



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

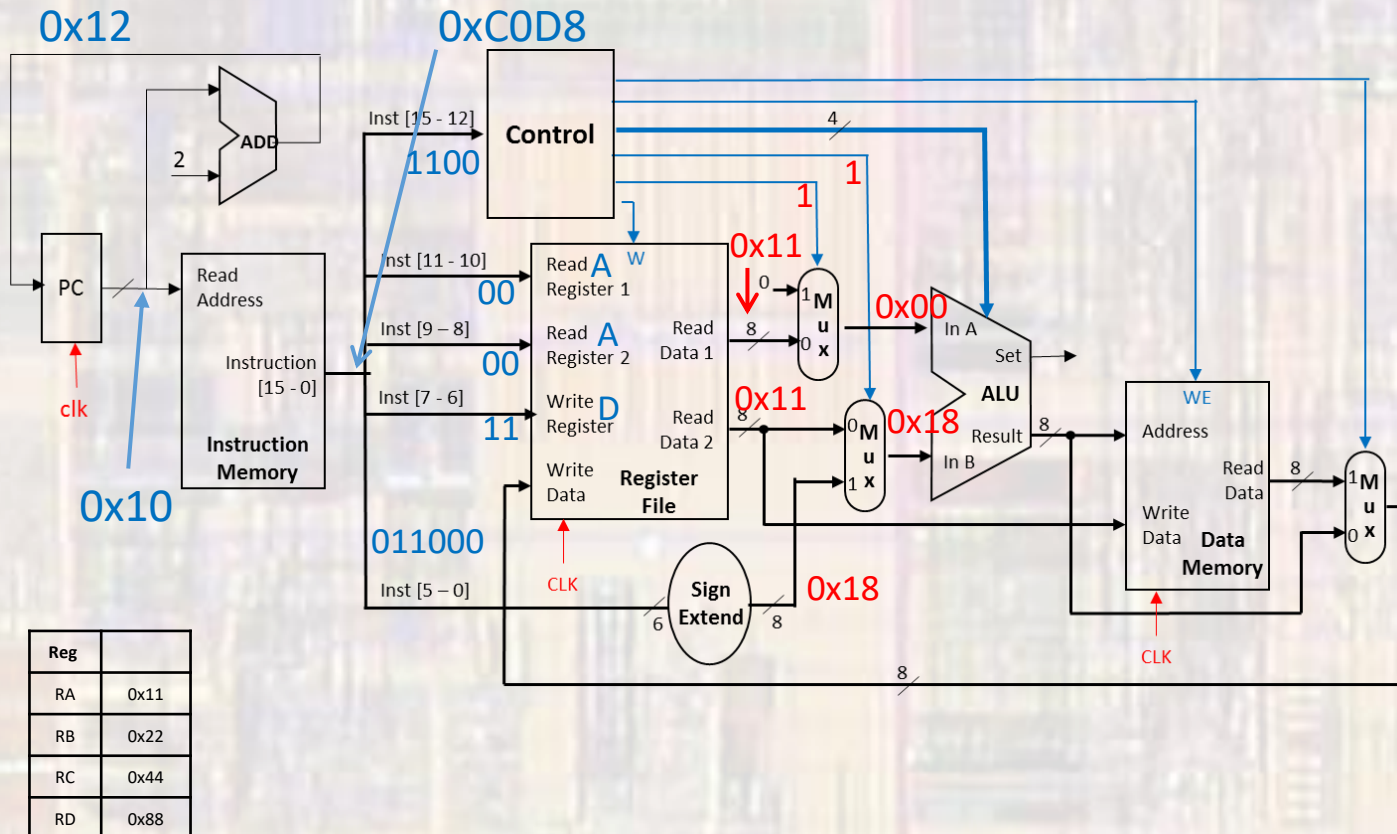
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `ldi RD, 24` located in program memory at 0x10



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

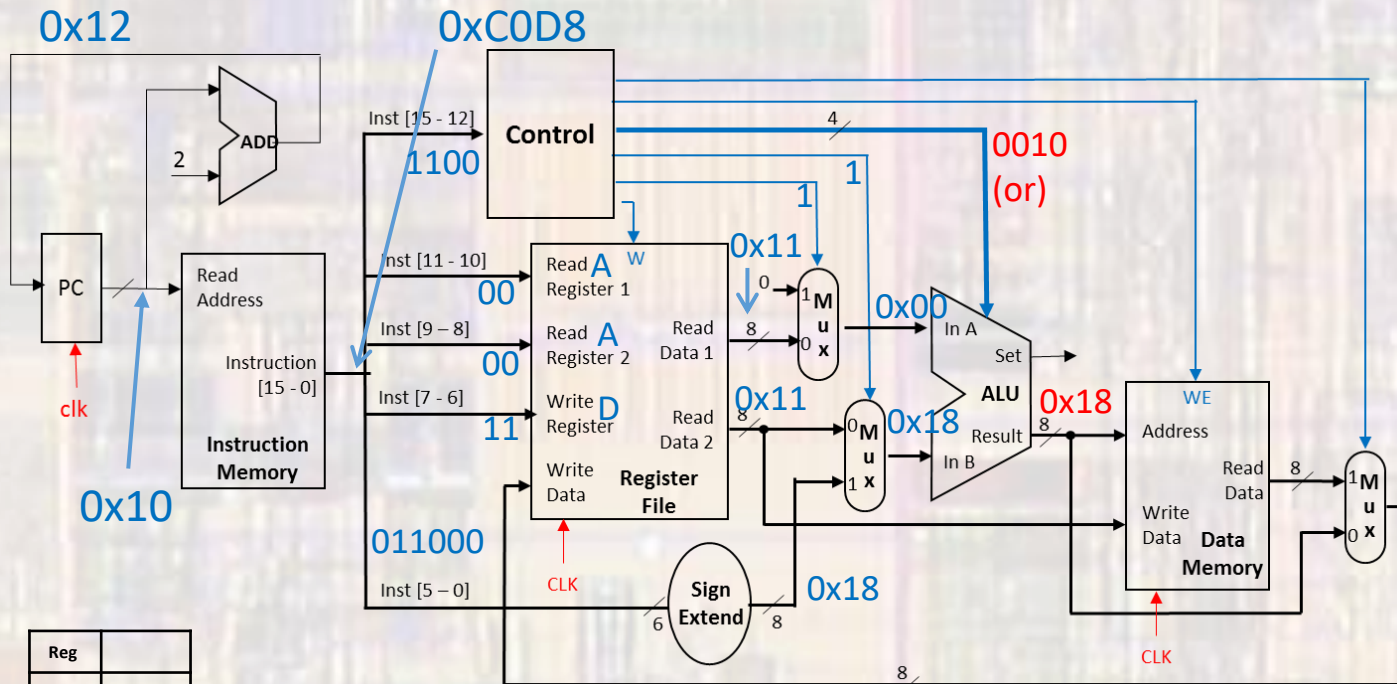
Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `ldi RD, 24` located in program memory at 0x10



Single Cycle Processor - Examples

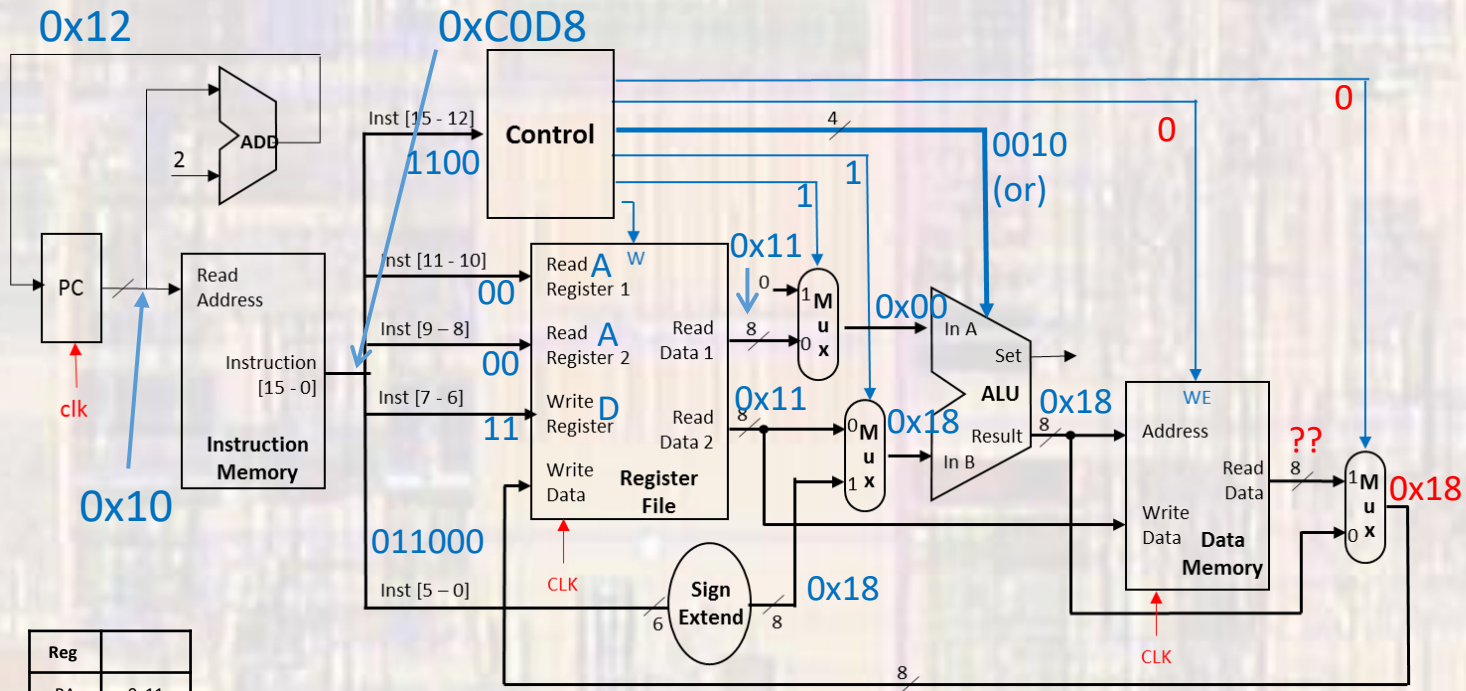
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `ldi RD, 24` located in program memory at 0x10



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

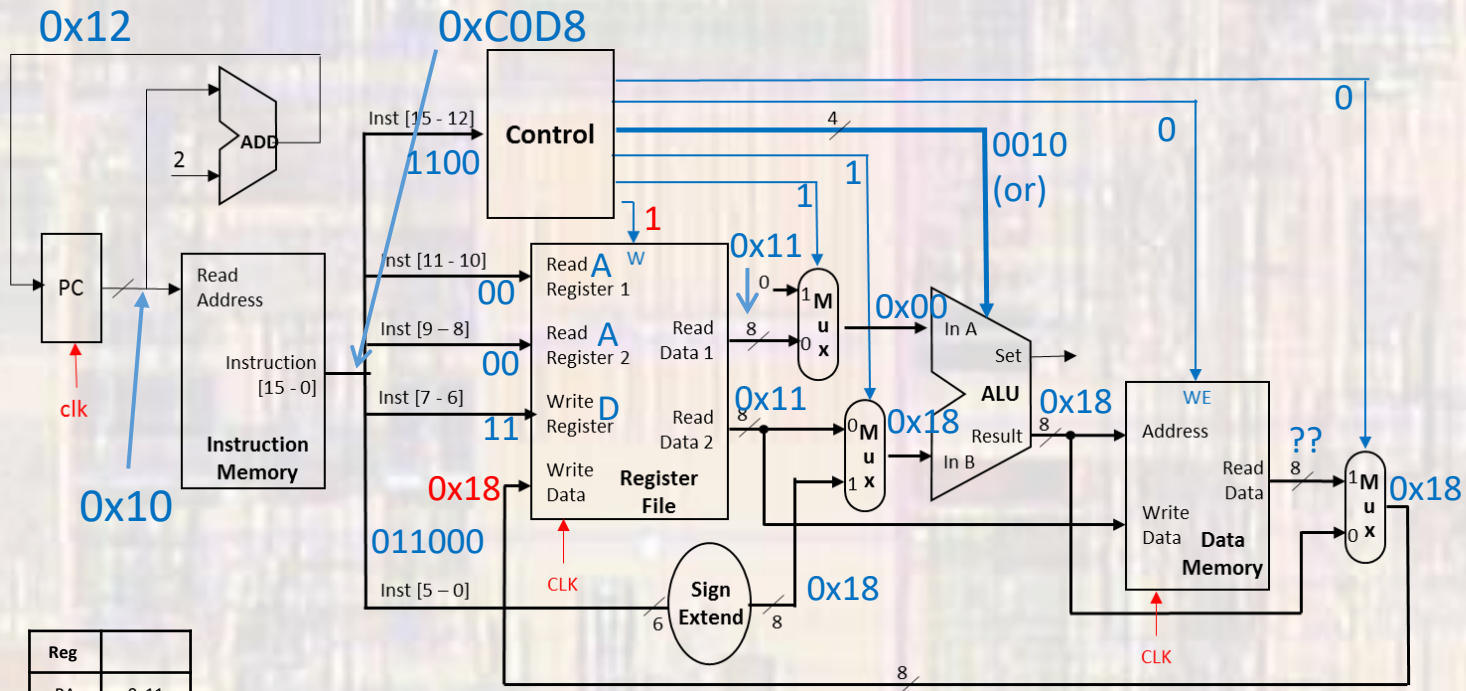
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: **ldi RD, 24** located in program memory at 0x10



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `ldi RD, 24` located in program memory at 0x10



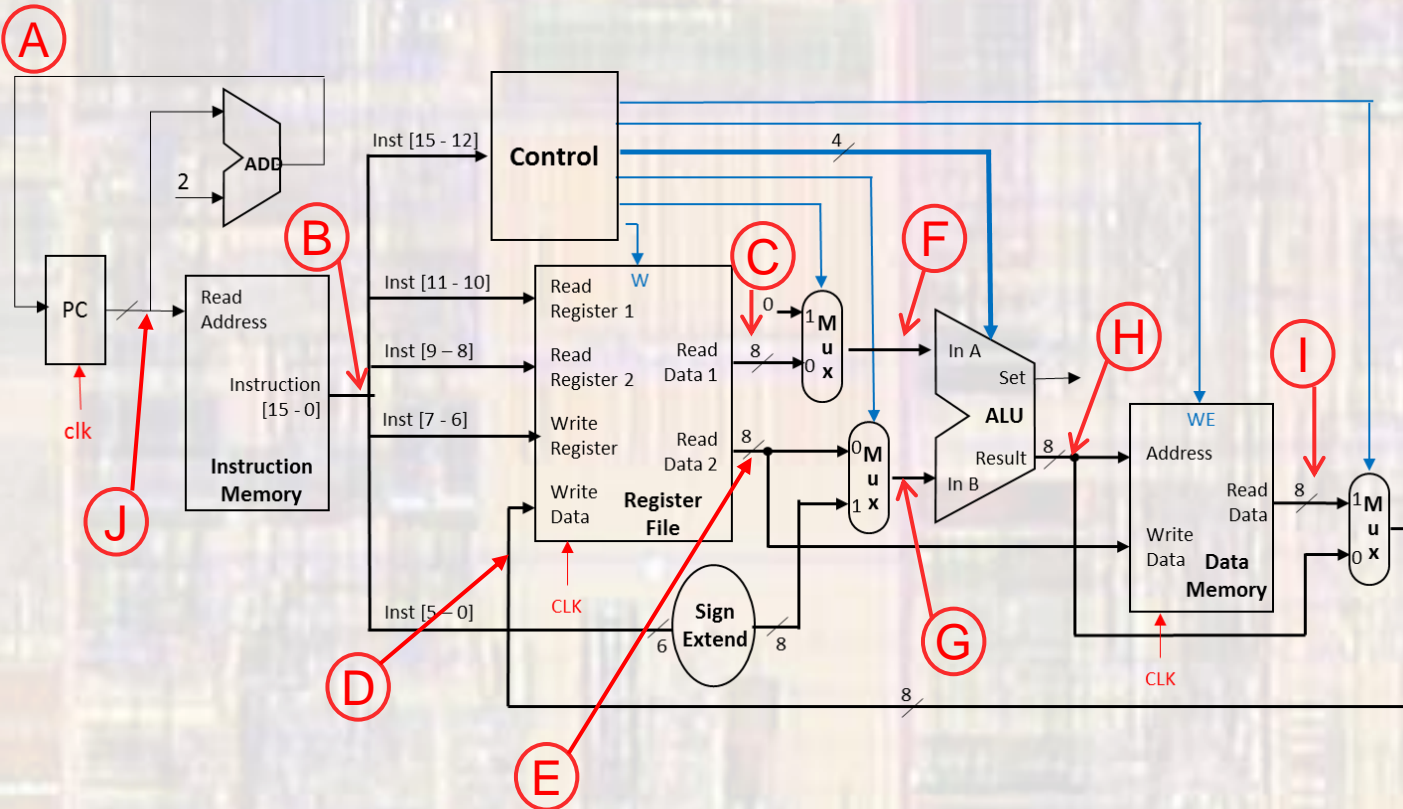
Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x18

At the next rising clock edge $D \leftarrow 0x18$

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: `ldi RD, 24` located in program memory at 0x10



Node	Value (hex)
A	0x12
B	0xC0D8
C	0x11
D	0x18
E	0x11
F	0x00
G	0x18
H	0x18
I	??
J	0x10

Single Cycle Processor - Examples

- Code the instruction for `st RC, RB`

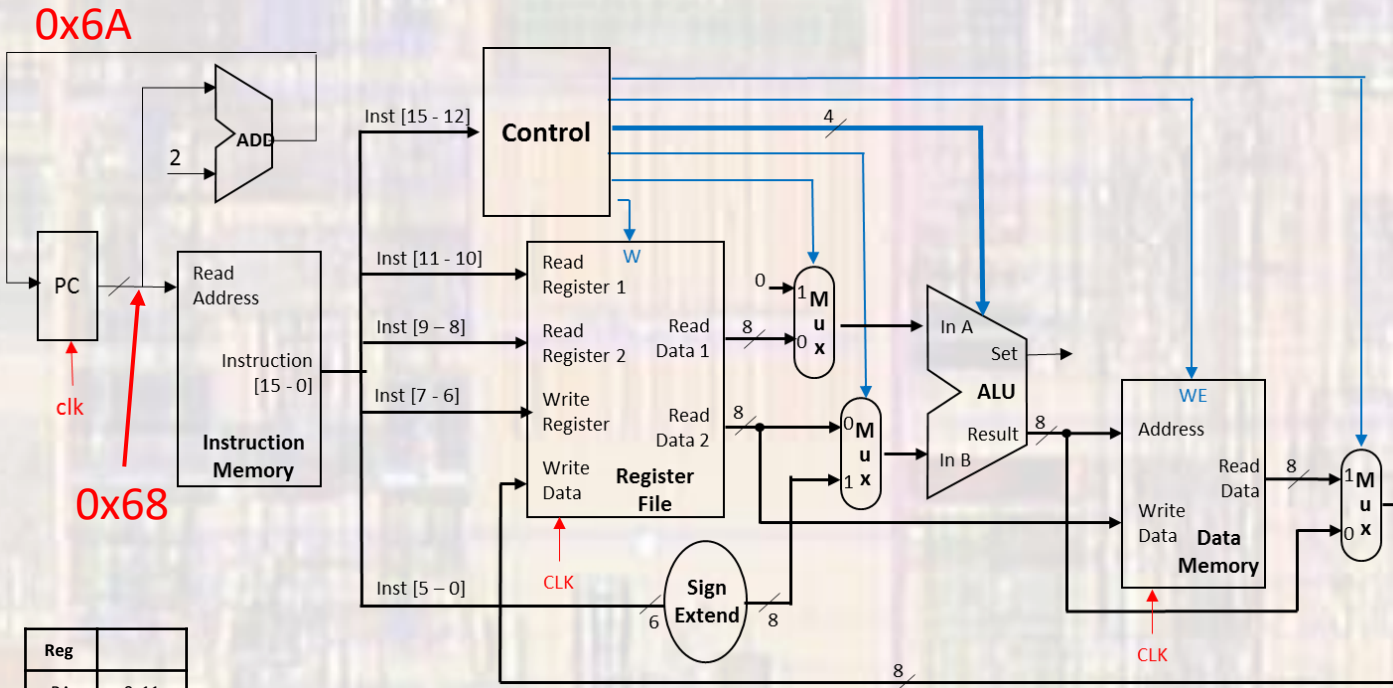
$\text{mem}(\text{RC}) \leftarrow \text{RB}$

or	0000			
and	0001			
nor	0010			
nand	0011			
add	0100			
sub	0101			
slt	0110	00 - A	00 - A	00 - A
ld	1000	01 - B	01 - B	01 - B
st	1001	10 - C	10 - C	10 - C
ldi	1100	11 - D	11 - D	11 - D

Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
1	0	0	1	1	0	0	1	x	x	0	0	0	0	0	0

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: **st RC, RB** located in program memory at 0x68



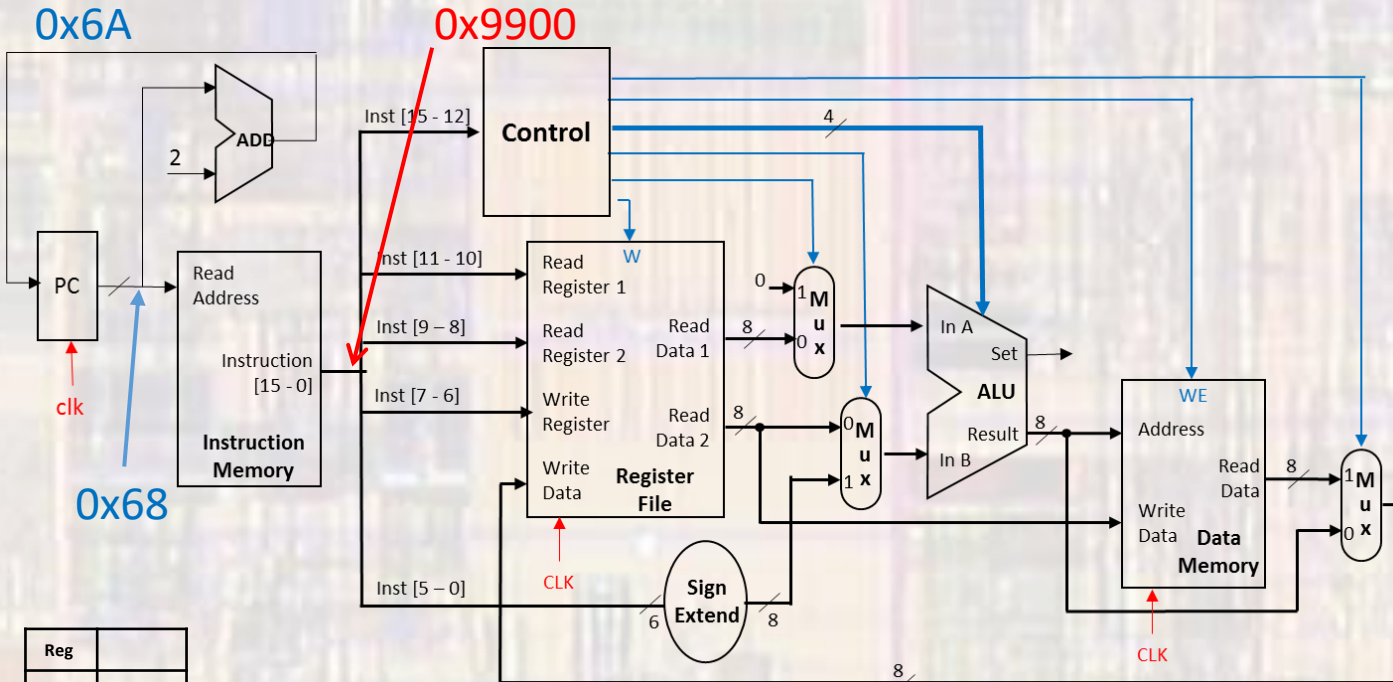
Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: `st RC, RB` located in program memory at 0x68

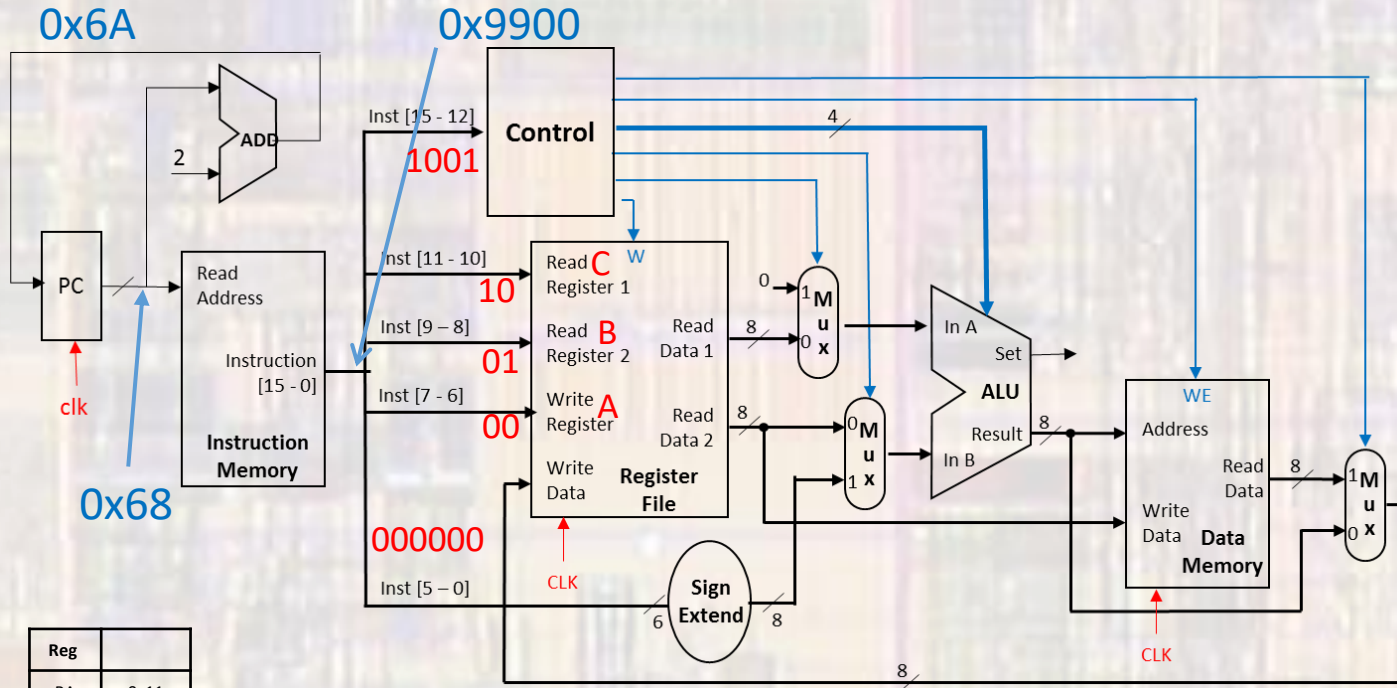
Instruction			Reg 1		Reg 2		W Reg		Immediate Value							
1	0	0	1	1	0	0	1	x	x	0	0	0	0	0	0	0



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

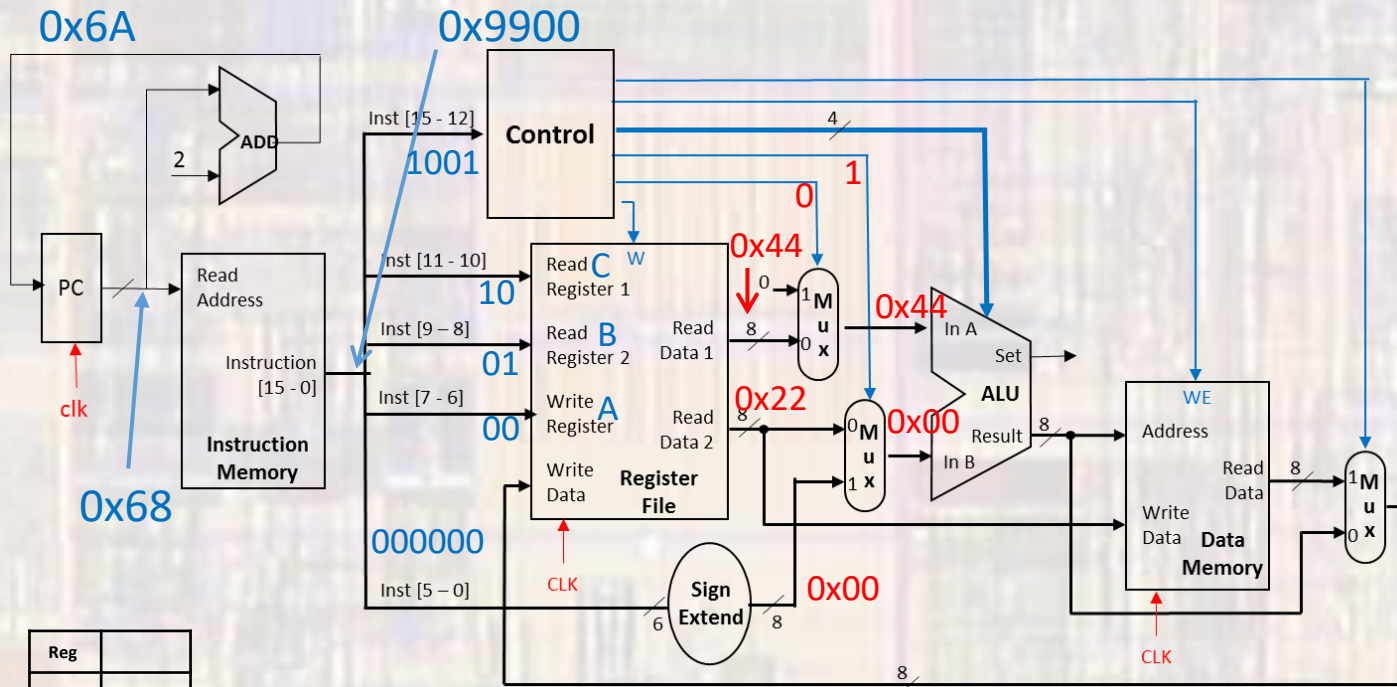
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `st RC, RB` located in program memory at 0x68



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

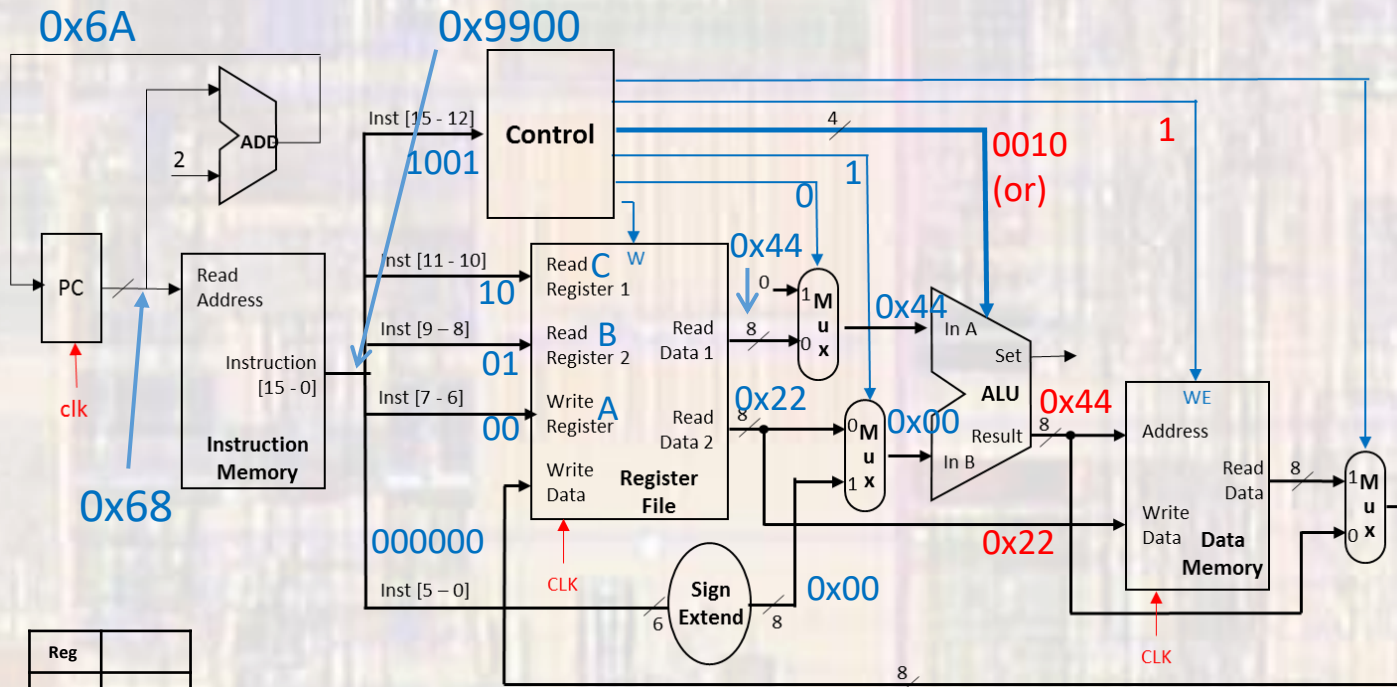
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `st RC, RB` located in program memory at 0x68



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

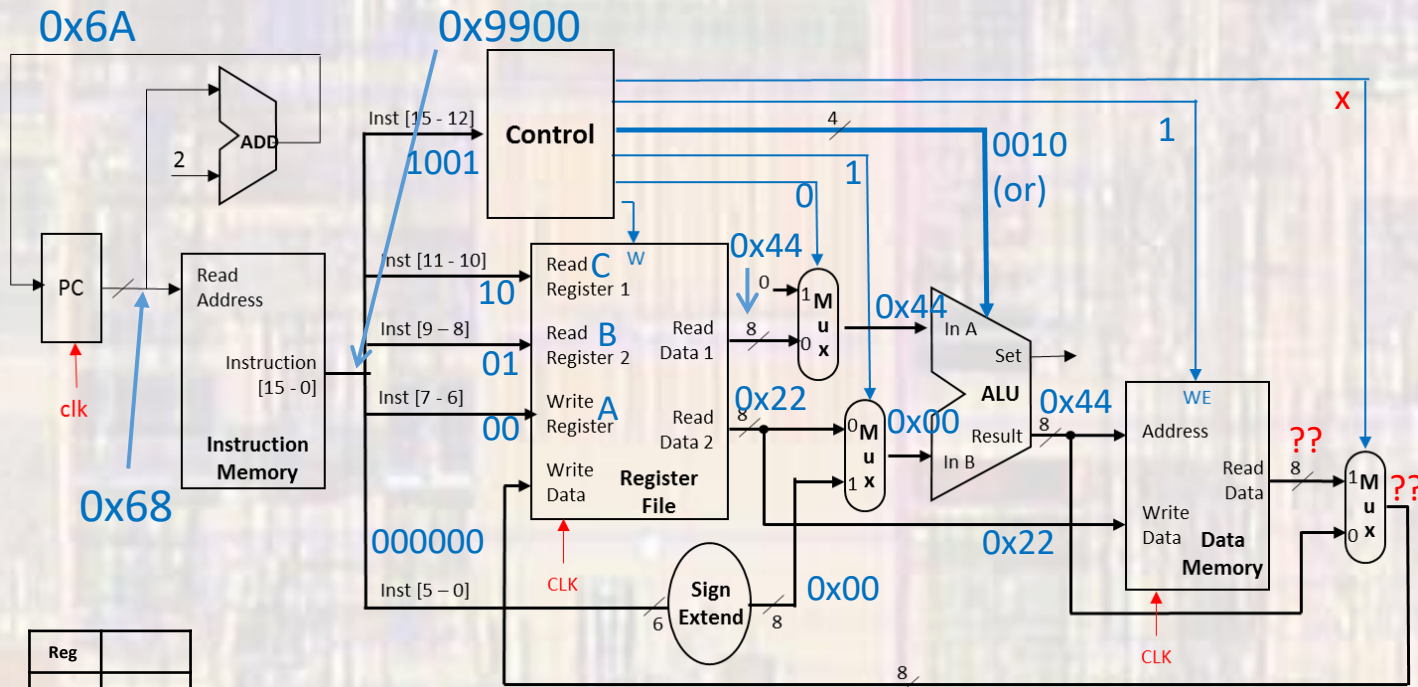
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `st RC, RB` located in program memory at 0x68



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

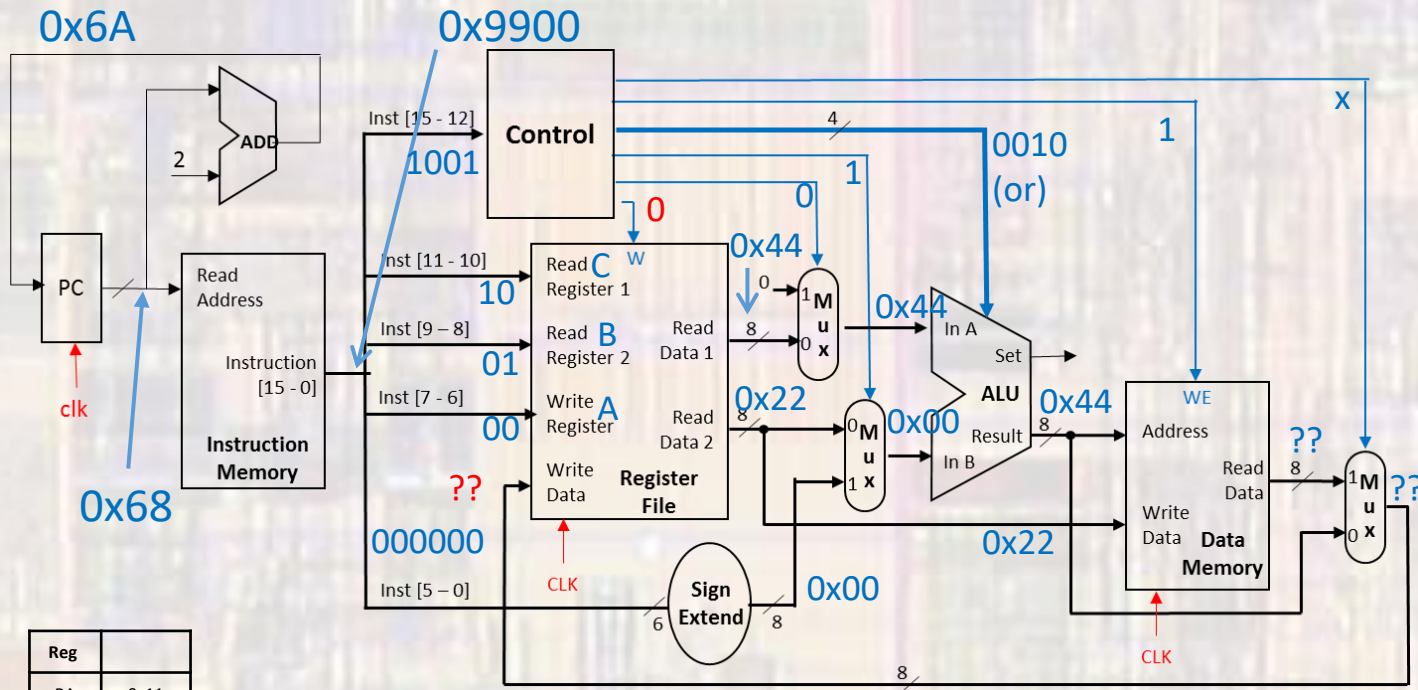
- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `st RC, RB` located in program memory at 0x68



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
- Provide the values for each signal after executing the instruction: `st RC, RB` located in program memory at 0x68



Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

No reg write

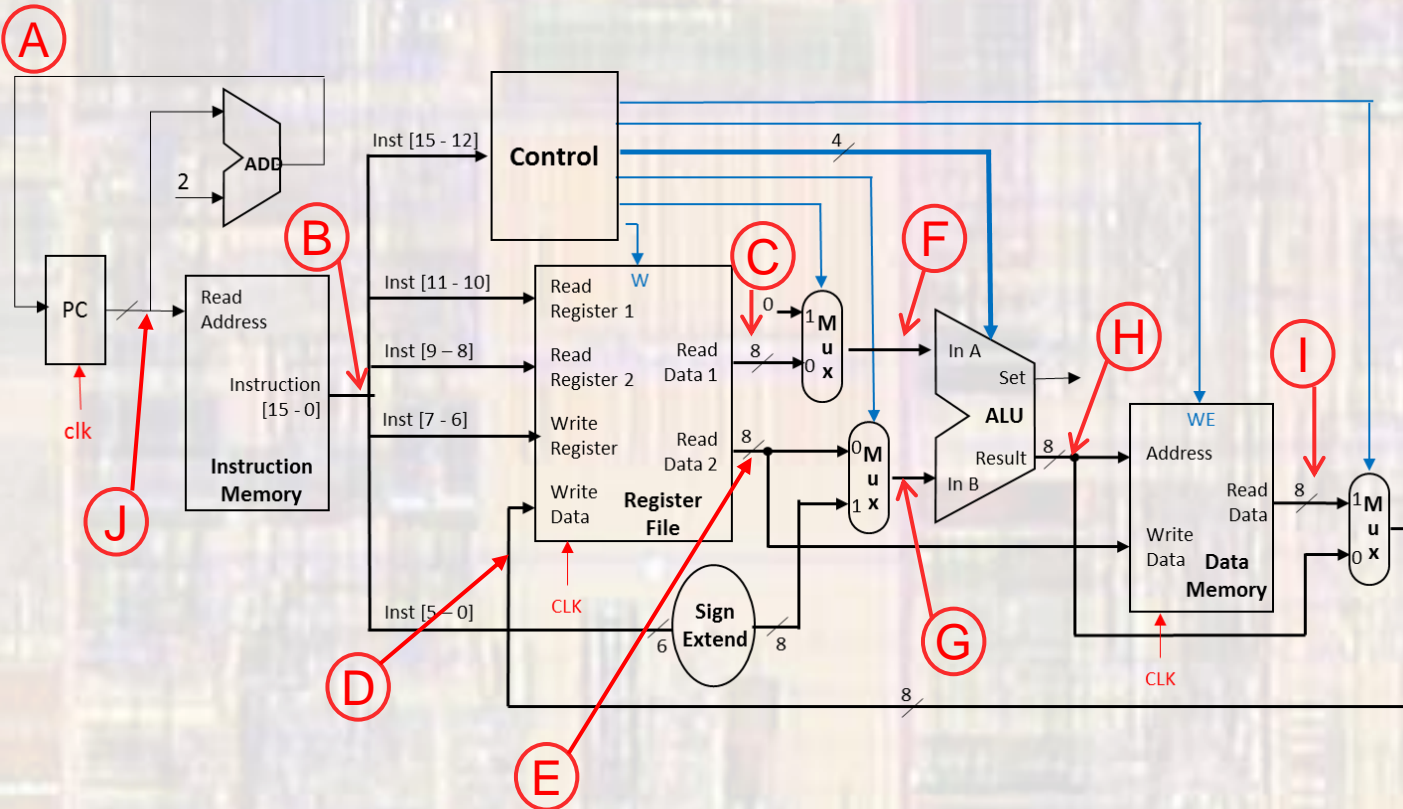
At the next rising clock edge

mem(0x44) ← 0x22

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: `st RC, RB` located in program memory at 0x68



Node	Value (hex)
A	0x6A
B	0x9900
C	0x44
D	??
E	0x22
F	0x44
G	0x00
H	0x44
I	??
J	0x68

Single Cycle Processor - Examples

- Code the instruction for **ld RC, RD**

RD \leftarrow mem(RC)

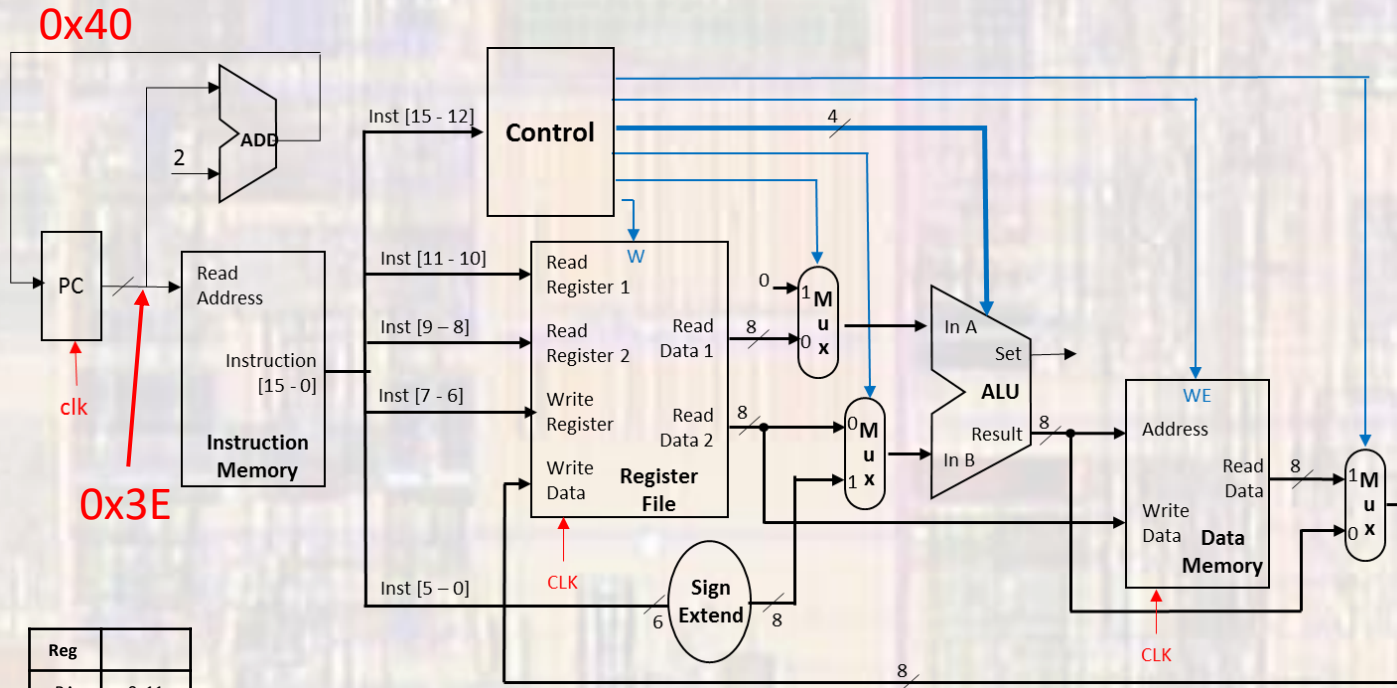
or	0000			
and	0001			
nor	0010			
nand	0011			
add	0100			
sub	0101			
slt	0110	00 - A	00 - A	00 - A
ld	1000	01 - B	01 - B	01 - B
st	1001	10 - C	10 - C	10 - C
ldi	1100	11 - D	11 - D	11 - D

Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
1	0	0	0	1	0	x	x	1	1	0	0	0	0	0	0

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$



0x40

0x3E

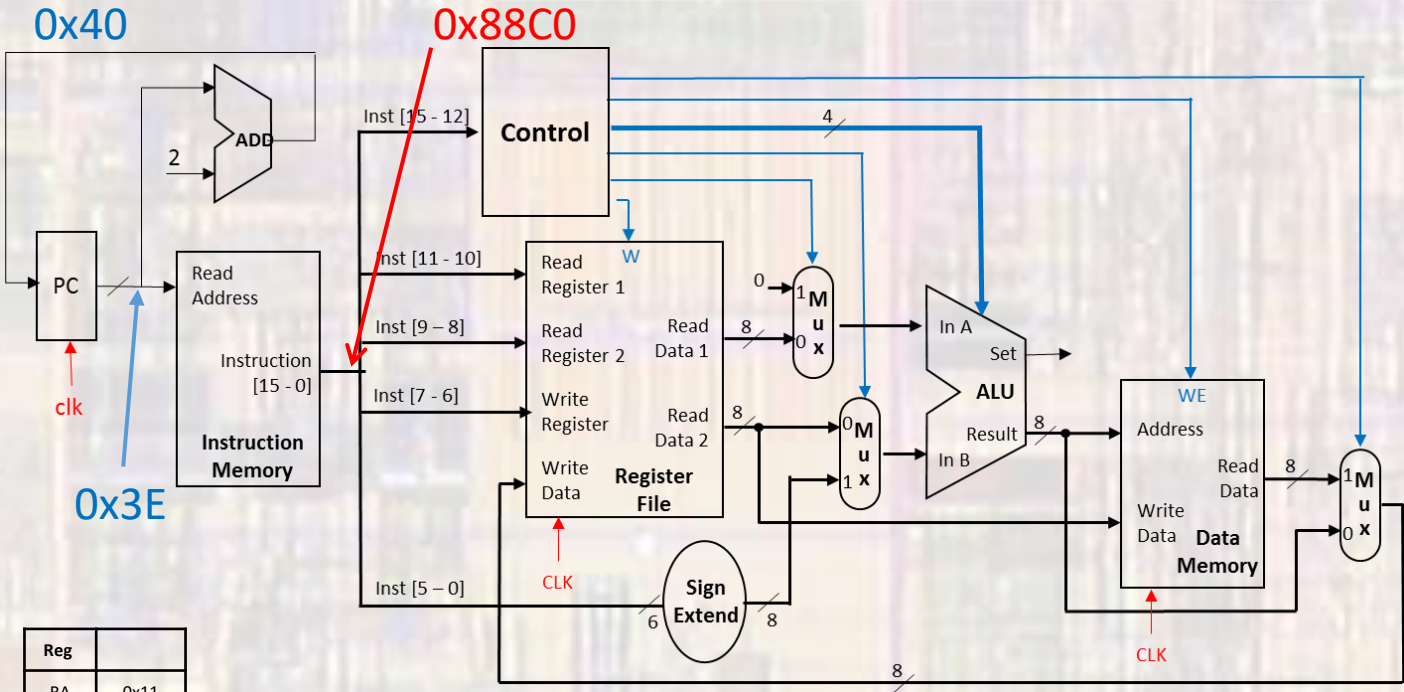
Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$

Instruction		Reg 1		Reg 2		W Reg		Immediate Value							
1	0	0	0	1	0	x	x	1	1	0	0	0	0	0	0

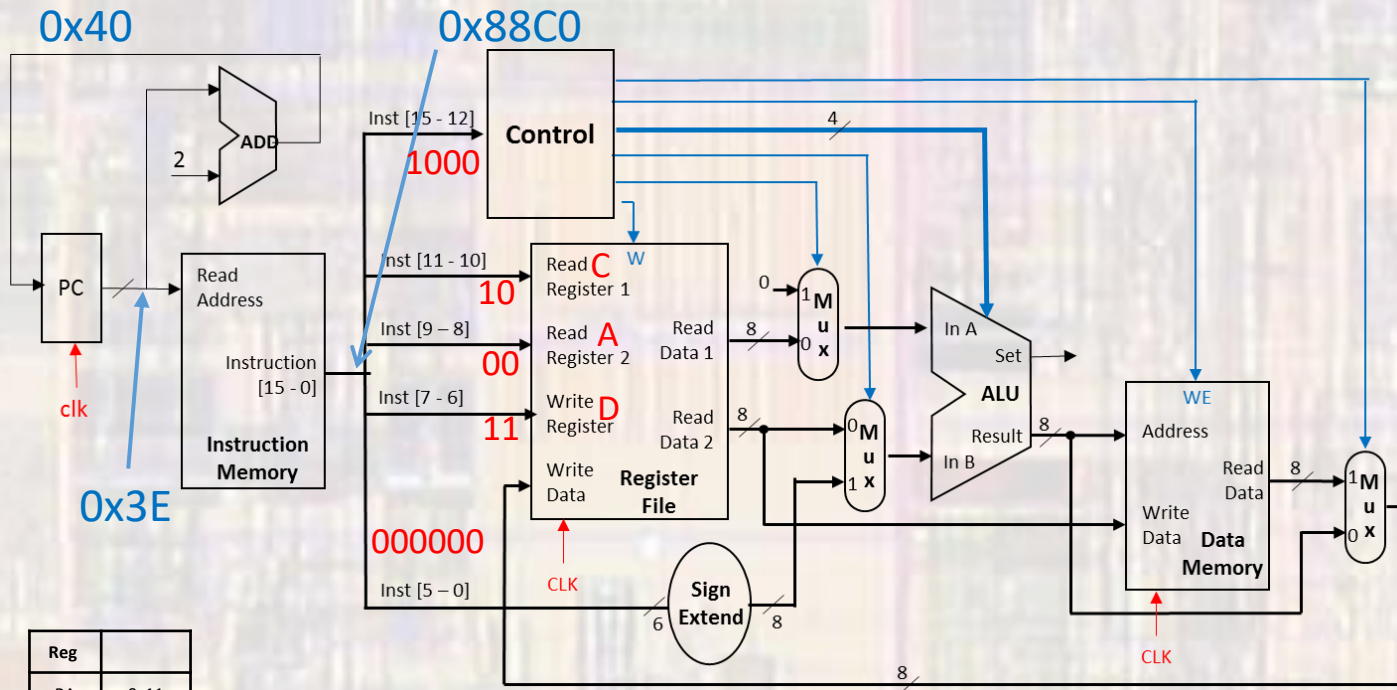


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$

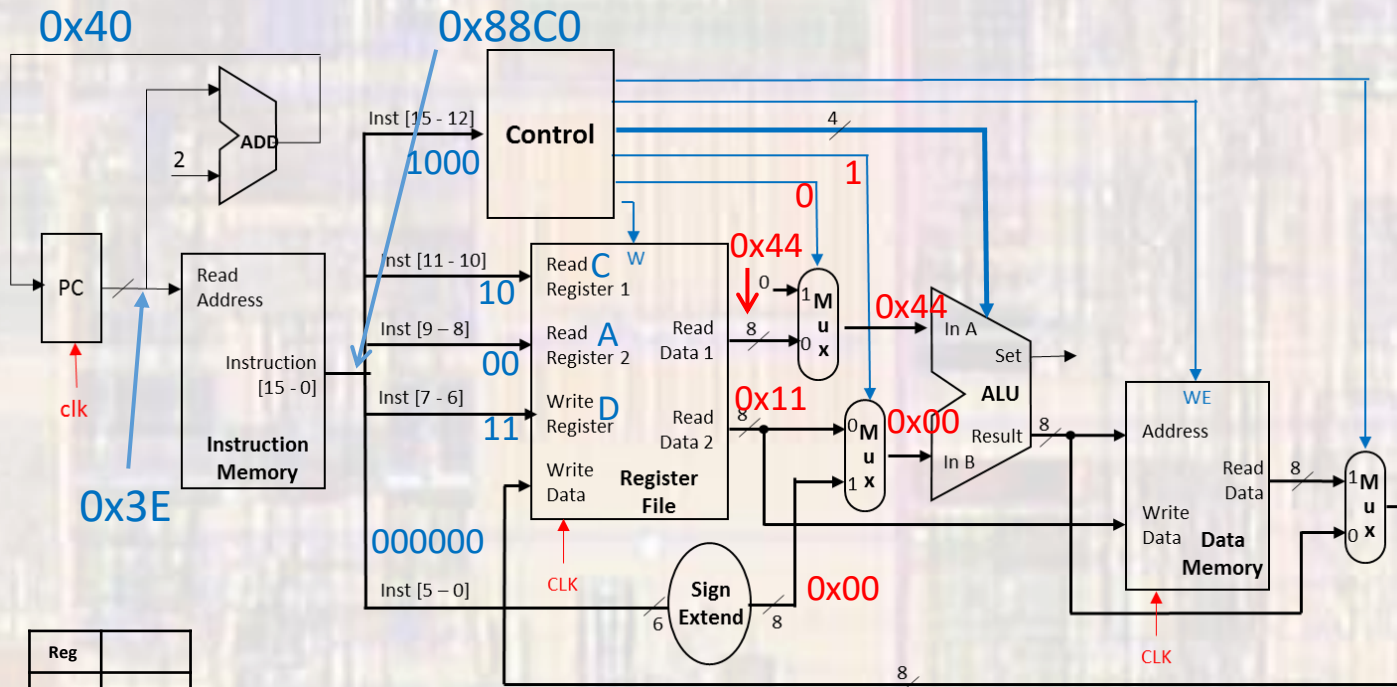


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$

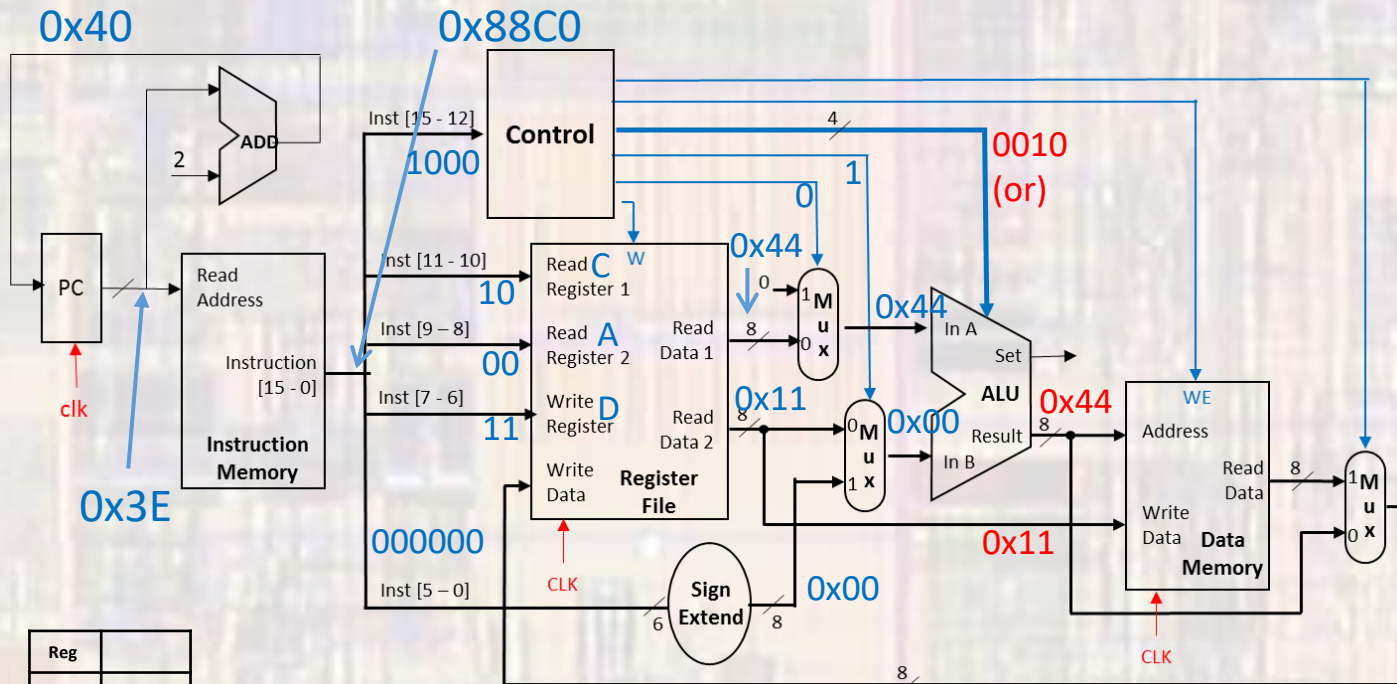


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$

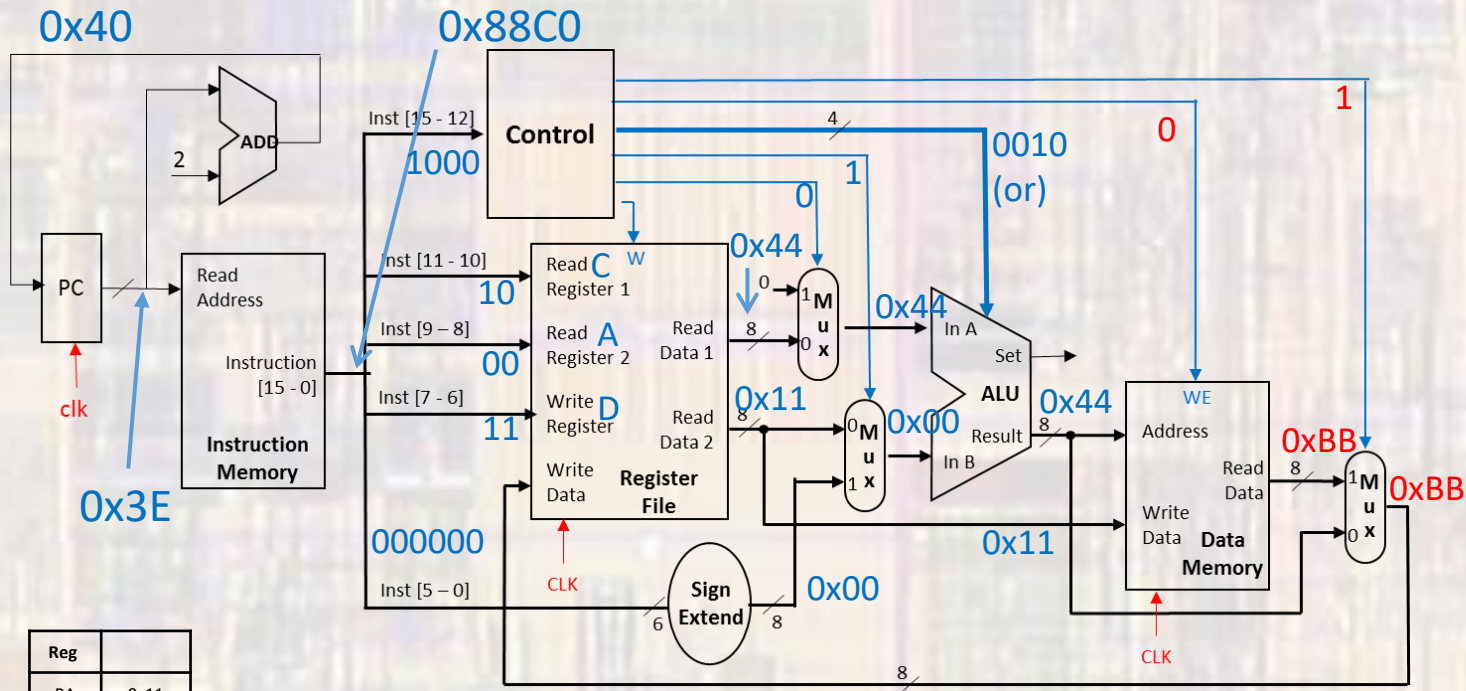


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$

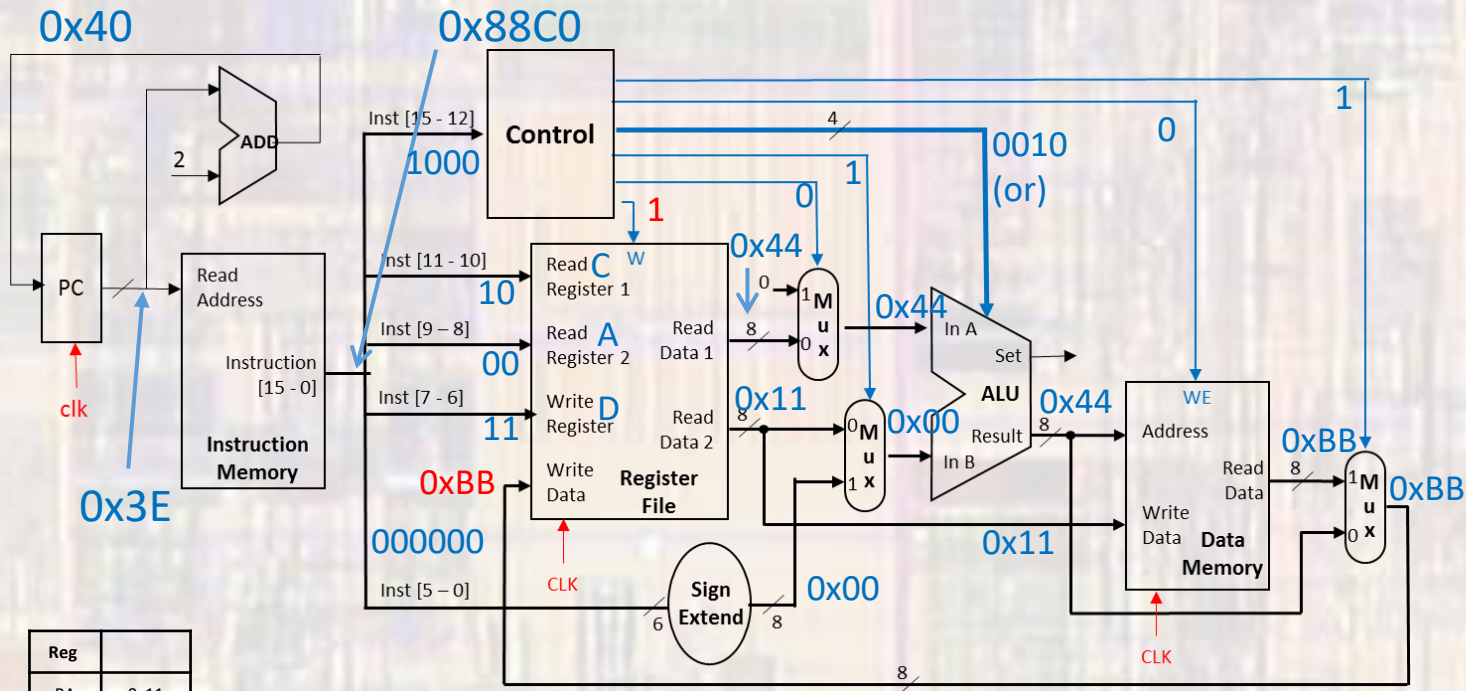


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$



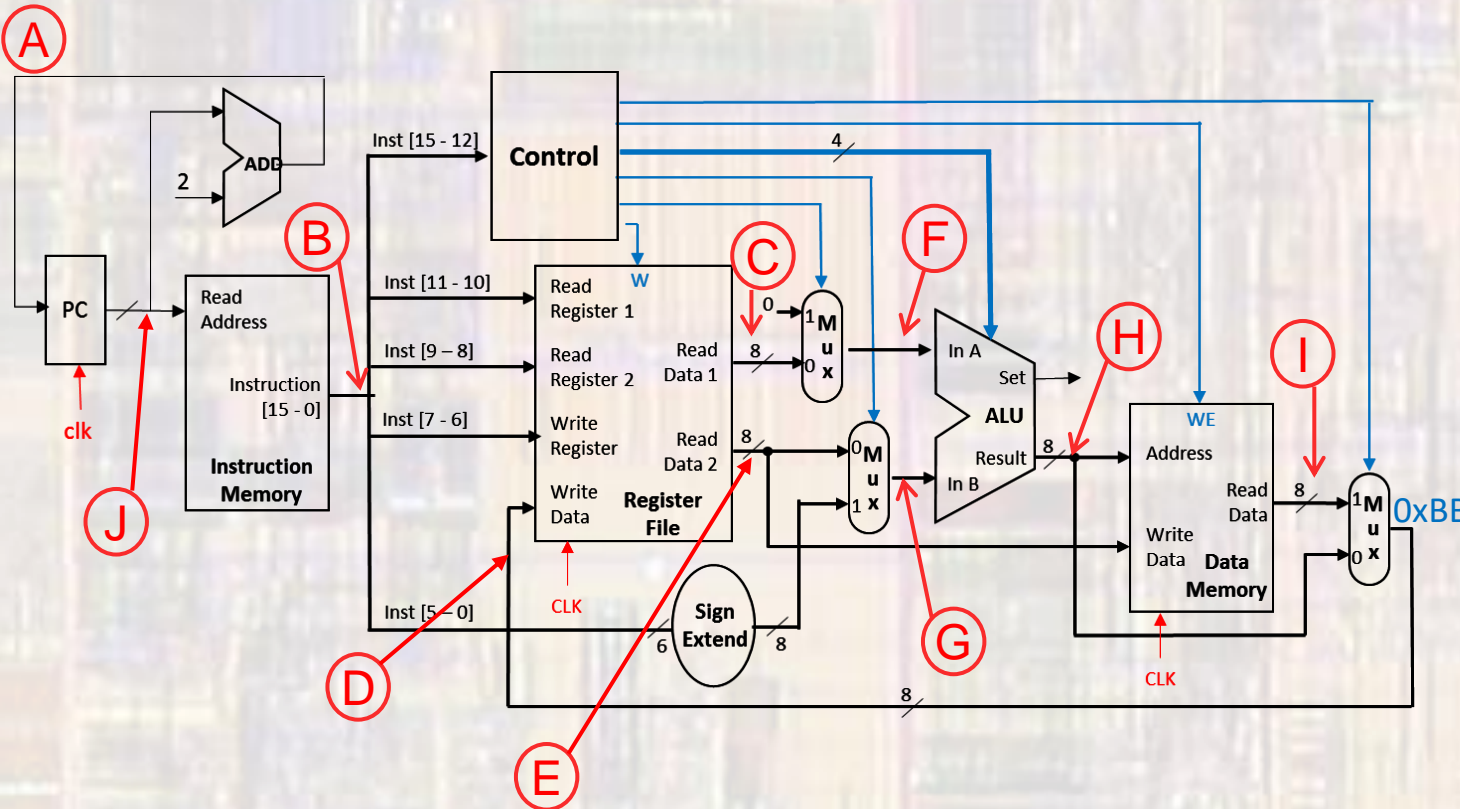
Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0xBB

At the next rising clock edge **RD ← 0xBB**

Single Cycle Processor - Examples

- Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Provide the values for each signal after executing the instruction: **ld RC, RD** located in program memory at **0x3E** assume $\text{mem}(0x43) = 0xAA$, $\text{mem}(0x44) = 0xBB$, $\text{mem}(0x45) = 0xCC$



Node	Value (hex)
A	0x40
B	0x88C0
C	0x44
D	0xBB
E	0x11
F	0x44
G	0x00
H	0x44
I	0xBB
J	0x3E