COMP2611: Computer Organization

Control of the single cycle datapath

Overview

- You will learn the following in this tutorial:
  - the hardware control units of the single cycle datapath,
  - the effects of the individual control signals,
  - how an instruction is executed through the guidance of the control signals.

Single Cycle Datapath

Review of the control signals

- the two hardware control units
- the effects of the control signals

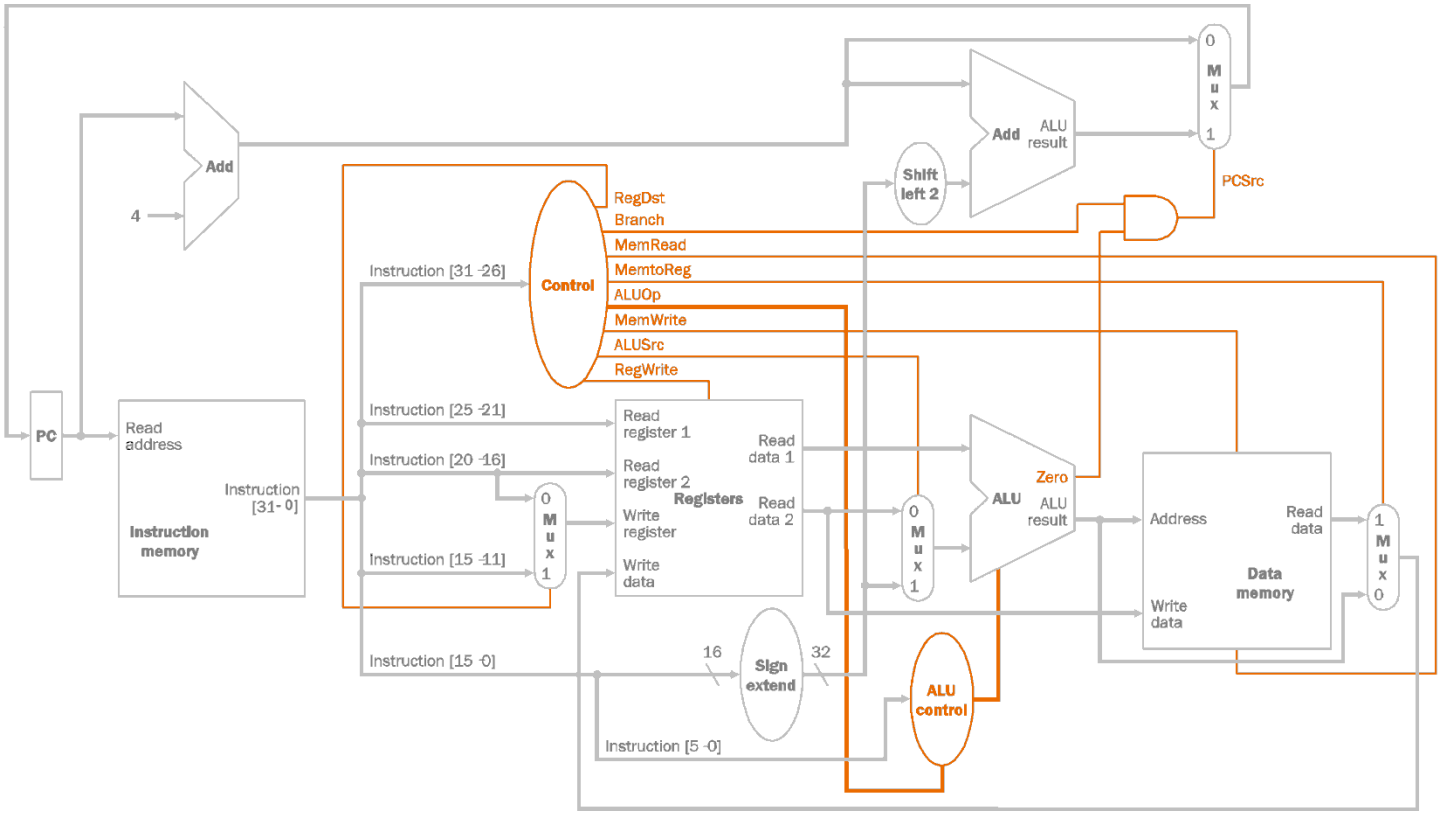
Sample execution of an instruction

- instruction fetch, decoding, executing, memory referencing and writing back
- Exercises

Exercises

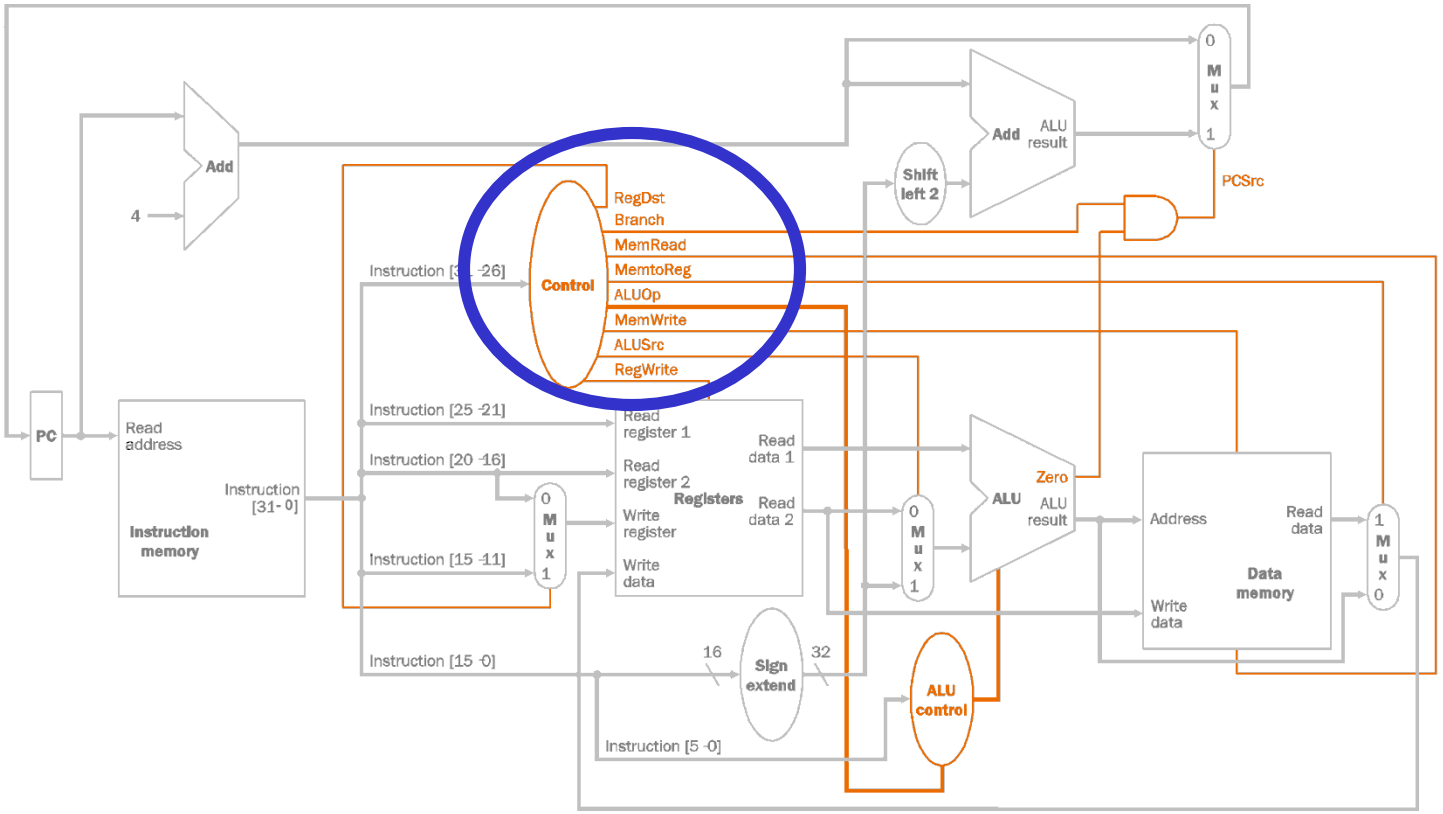
Review of the Control signals

- The control units of the datapath are responsible for setting all the control signals for the instructions.
- Two hardware units:
  - Control – 6 bits of input (bits [31-26]),
  - ALU Control – 2+6 bits of input (ALUOp from control, bits [5-0])



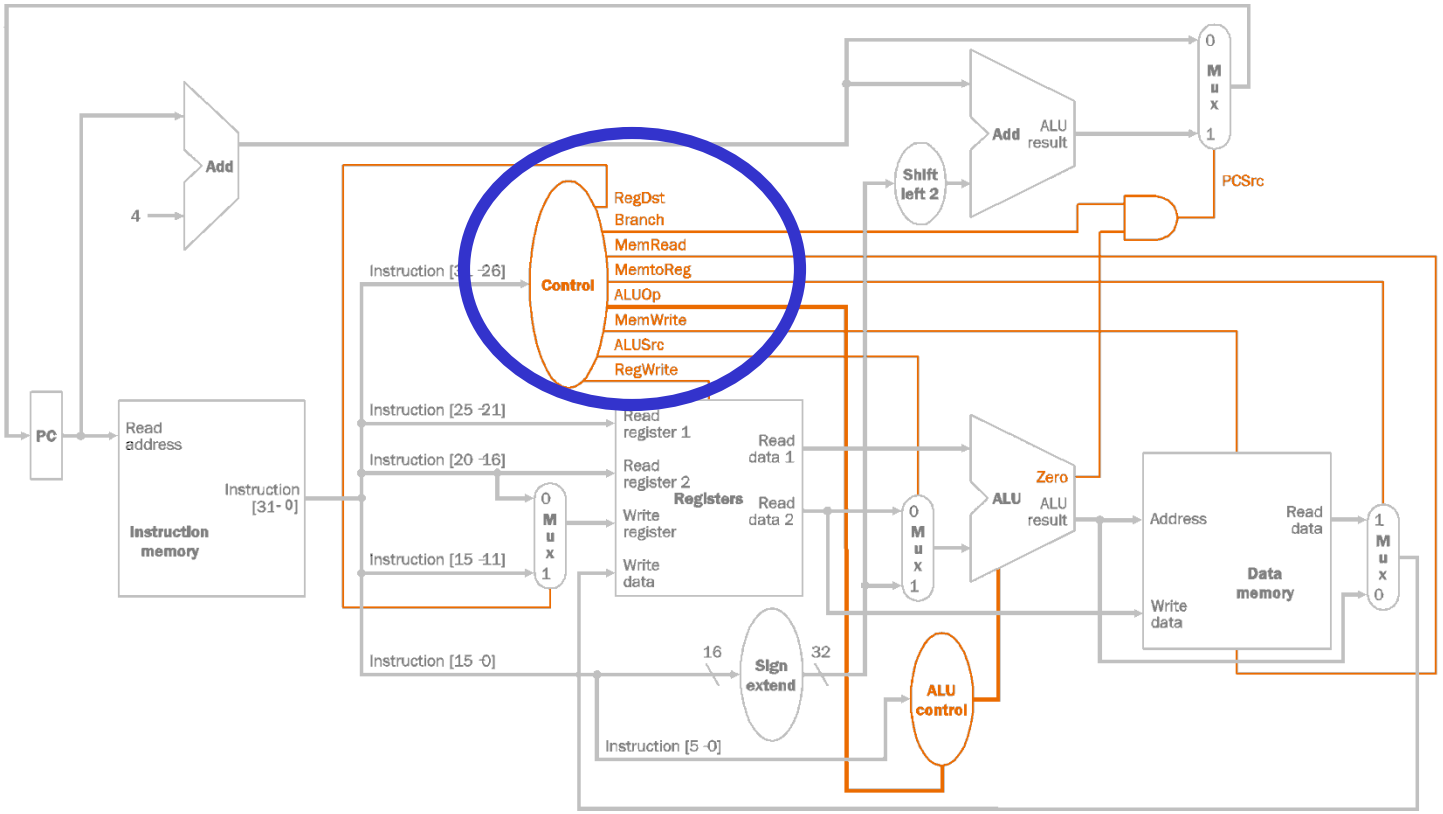
Review of the Control signals

Signal name	Effect when deasserted	Effect when asserted
RegDst	The register destination number for the write register comes from rt field (bits 20-16)	The register destination number for the write register comes from rd field (bits 15-11)
Branch + Zero	When both signals are asserted, PCSrc will be asserted	When either one is deasserted, PCSrc will be deasserted
MemRead	None	Enable read from memory. Memory contents designated by the address are put on the read data output
MemtoReg	Feed the write data input of the register file with the output from the ALU	Feed the write data input of the register file with output from the memory



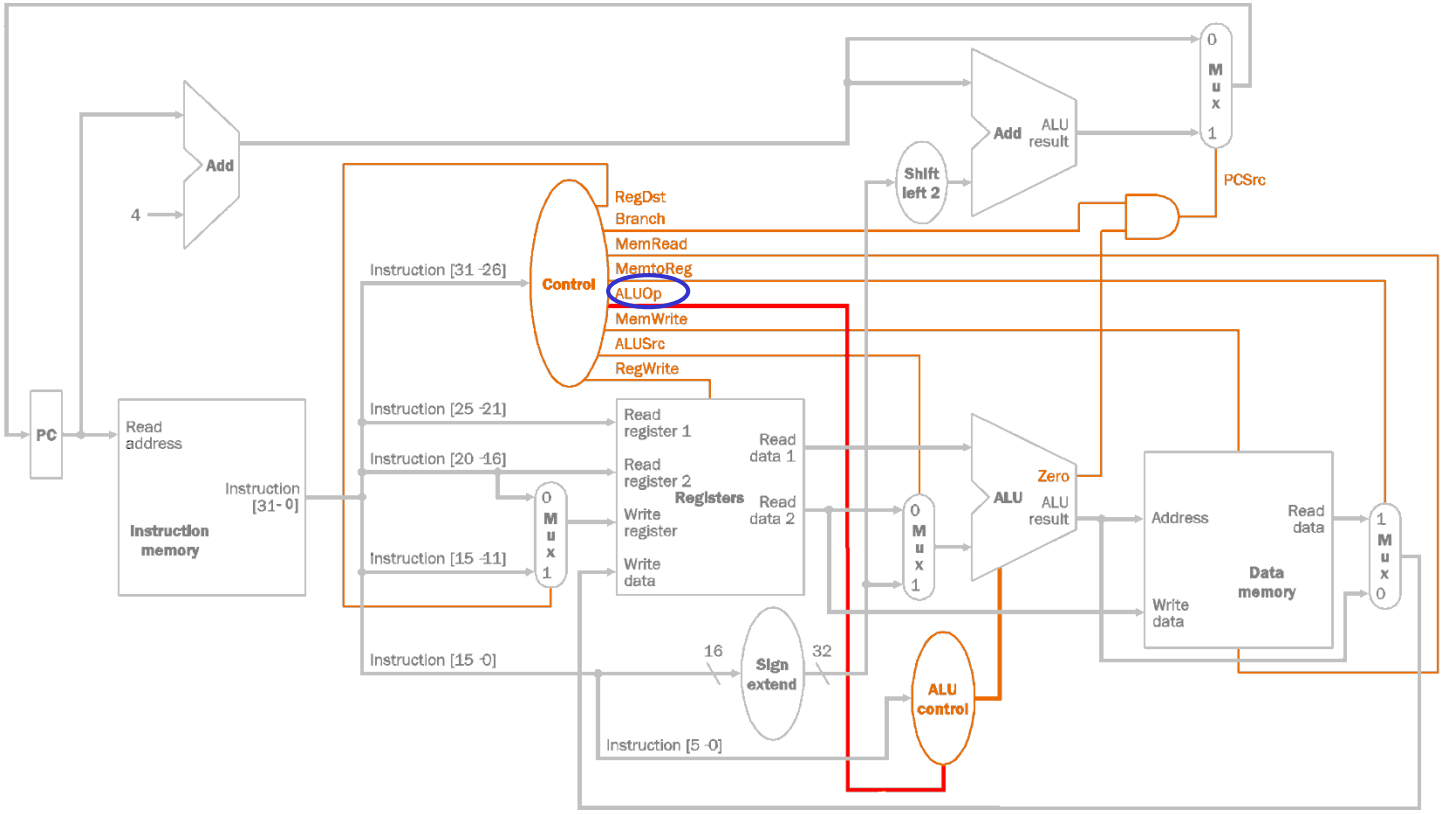
Review of the Control signals

Signal name	Effect when deasserted	Effect when asserted
MemWrite	None	Enable write to the memory. Overwrite the memory contents designated by the address with the value on the write data input
ALUSrc	The Second ALU operand comes from the second register file output (read data port 2)	The second ALU operand is the sign-extended 16-bit data (bits 15-0)
PCSrc	The next PC picks up from the output of the adder that computes PC+4	The next PC picks up the output of from the adder that computes the branch target
RegWrite	None	Enable data write to the register specified by the register destination number



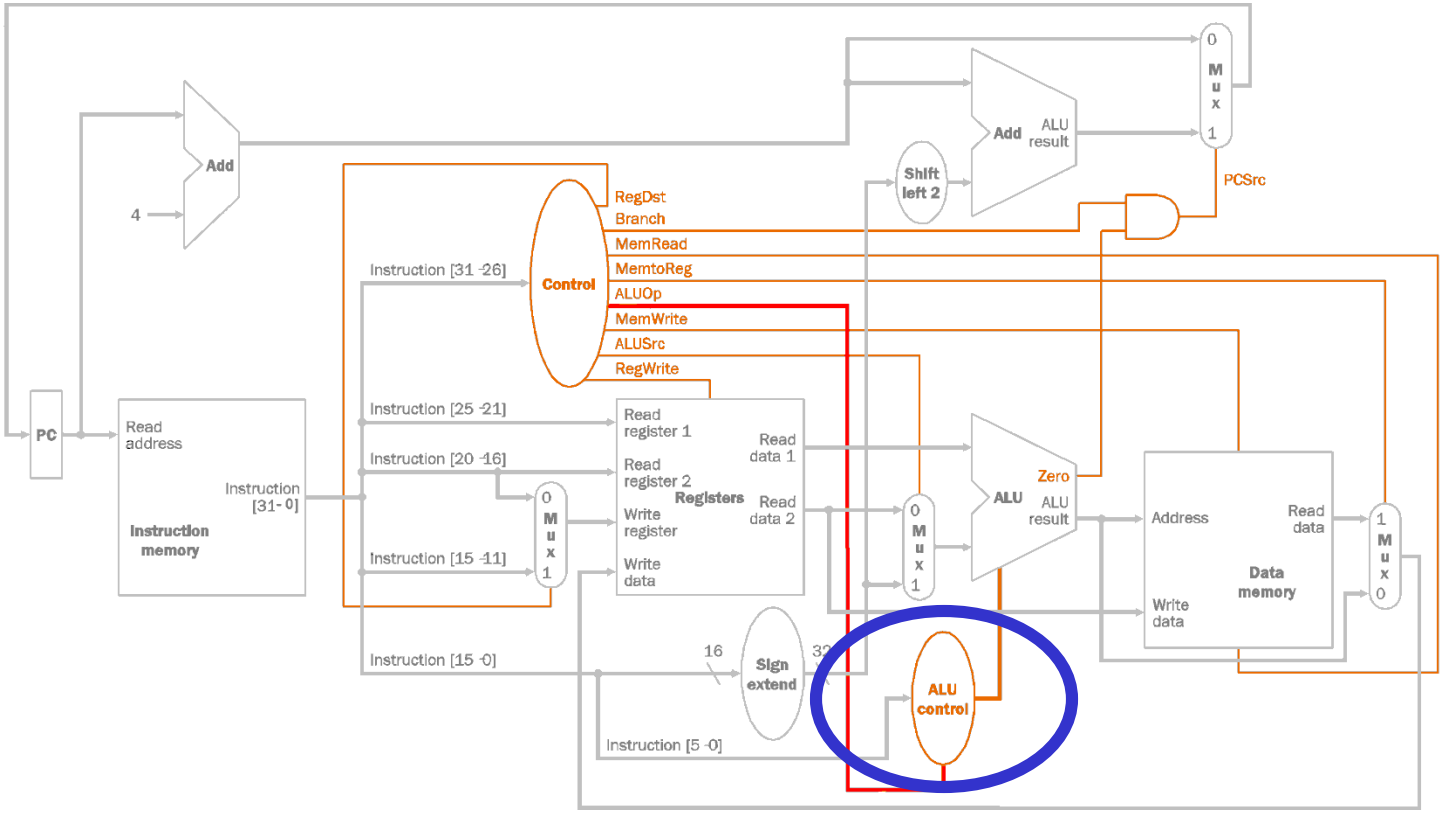
Review of the Control signals

ALUOp signal (2-bit)	Effect
00	ADD operation at the ALU for Load/Store instructions
01	SUB operation at the ALU for BEQ
10	ALU operation determined by the funct field (bits 5-0)



Review of the Control signals

ALU control signal (4-bit)	Effect
0000	The ALU will perform the AND operation
0001	The ALU will perform the OR operation
0010	The ALU will perform the ADD operation
0110	The ALU will perform the SUB operation
0111	The ALU will perform the Set on Less Than (SLT) operation
1100	The ALU will perform the NOR operation



Single Cycle Datapath

Review of the control signals

- the two hardware control units
- the effects of the control signals

Sample execution of an instruction

- instruction fetch, decoding, executing, memory referencing and writing back
- Exercises

Exercises

Execution of an instruction (with control signals)

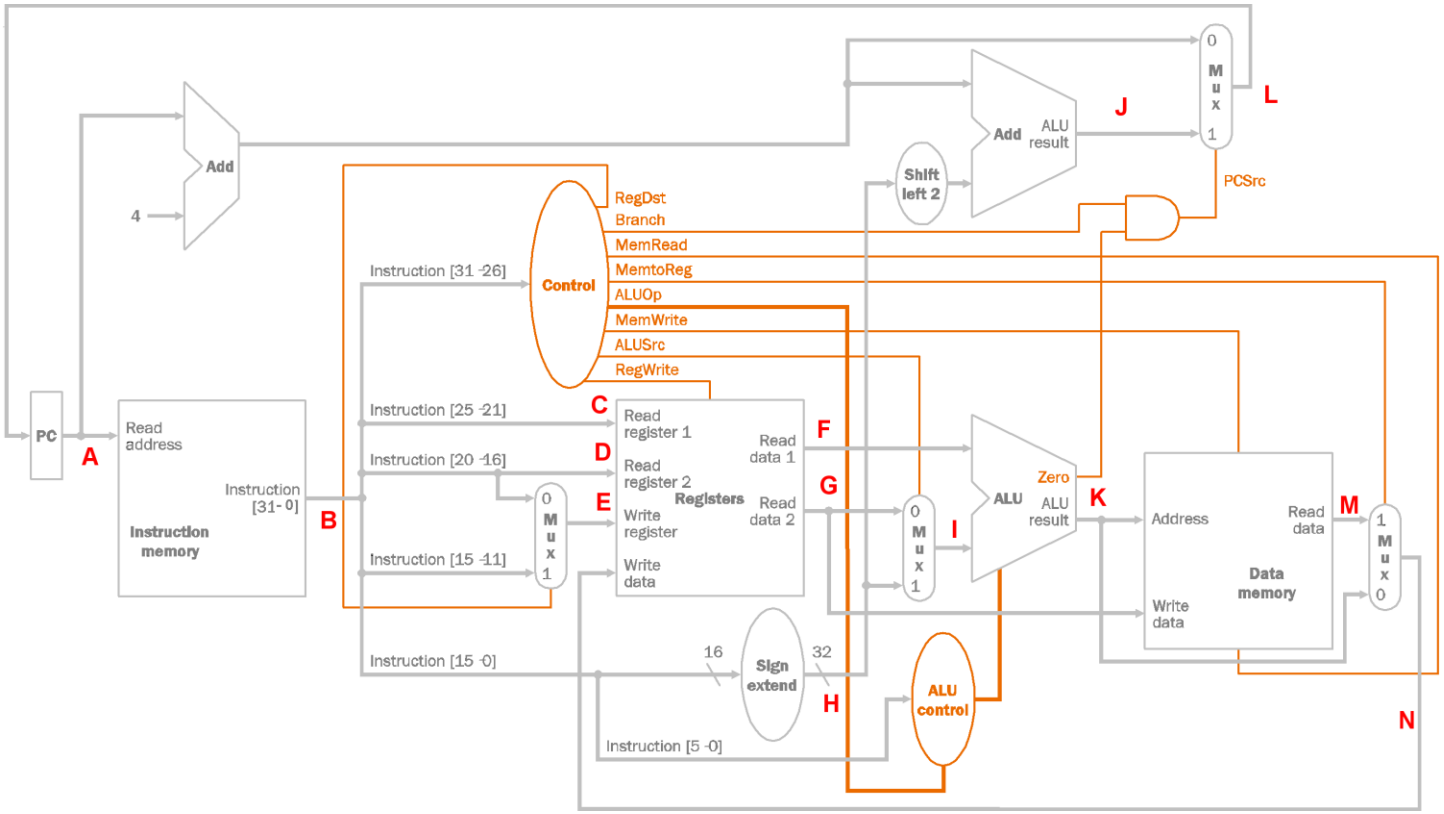
- Execution of the R-type instruction "AND"

Description	Bitwise AND operation on the values of the registers Rs and Rt. Store the result in Rd.
Operation	$Rd=$Rs & $Rt; PC=PC+4
Syntax	AND $Rd, $Rs, $Rt
Encoding	000000 sssss ttttt ddddd 00000 100100 OpCode Rs Rt Rd Shift function

- Example: AND $t0, $t1, $t2
  - Encoding 000000 01001 01010 01000 00000 100100

Execution of "AND $t0, $t1, $t2"

- Step 1 (fetch):
  - A= address of the instruction
  - B= 000000 01001 01010 01000 00000 100100
  - Control signals not yet generated



Execution of "AND $t0, $t1, $t2"

- Step 2 (decode):

  C = 01001, D = 01010, E = 01000

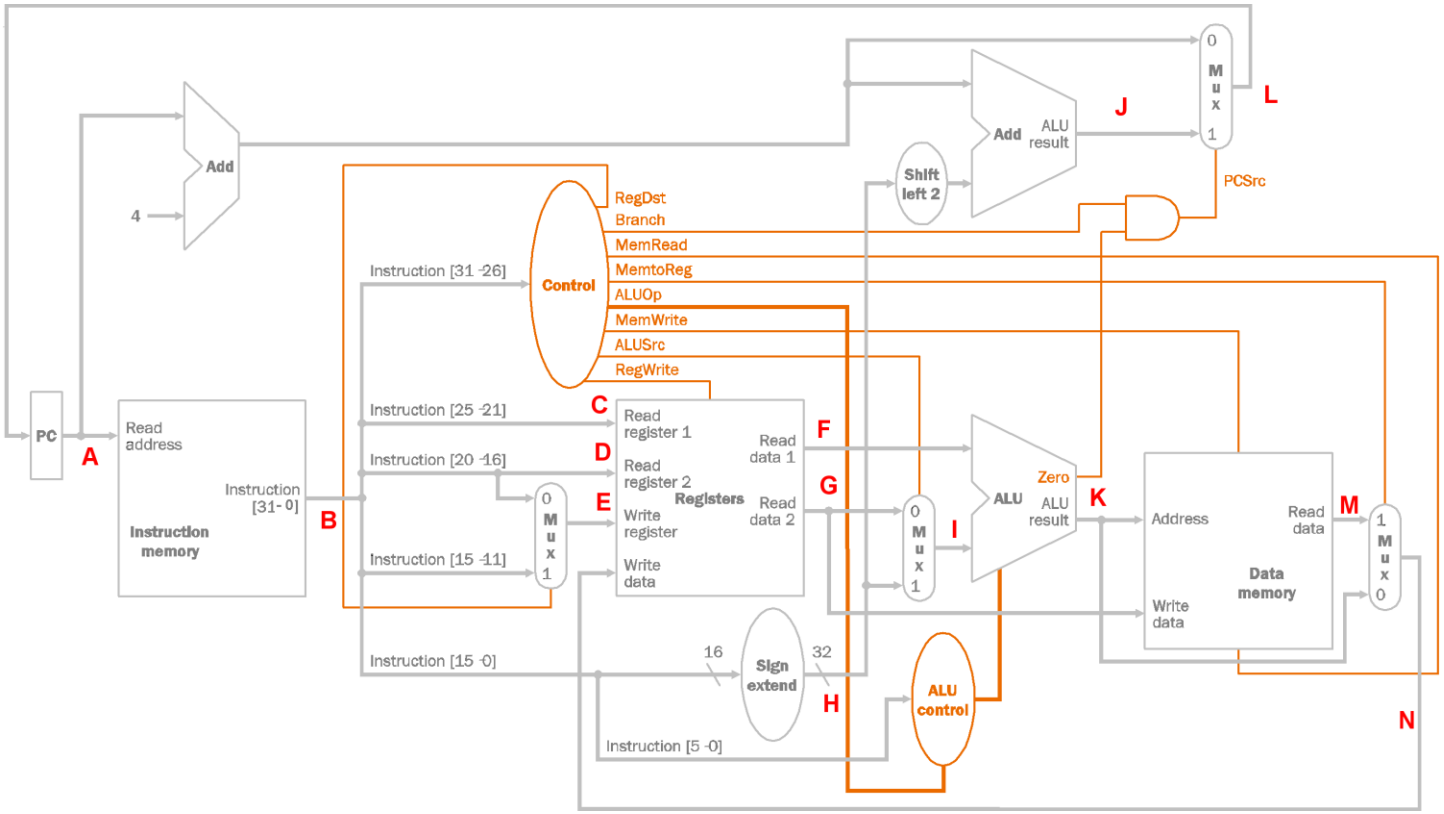
  F = value of Rs

  G = value of Rt, H= 0000 0000 0000 0000 01000 00000 100100

Instruction fetched:

000000 01001 01010 01000 00000 100100

OpCode Rs Rt Rd Shift function



Execution of "AND $t0, $t1, $t2"

- Step 2 (control signals)

Instruction fetched:

000000 01001 01010 01000 00000 100100

OpCode Rs Rt Rd Shift function

Control signal	Value
RegDst	1 (R-type instruction)
Branch	0 (not a branch instruction i.e. BEQ)
MemRead	0 (not the LW instruction)
MemtoReg	0 (not the LW instruction)
ALUOp	10 (R-type instruction)
MemWrite	0 (not the SW instruction)
ALUSrc	0 (R-type instruction)
RegWrite	1 (R-type instruction needs to write back)

Inputs		Outputs
Funct (bits 5:0)	ALUOp	ALU Control
100 100	10	0000 (AND operation)

Execution of "AND $t0, $t1, $t2"

Instruction fetched:
000000 01001 01010 01000 00000 100100

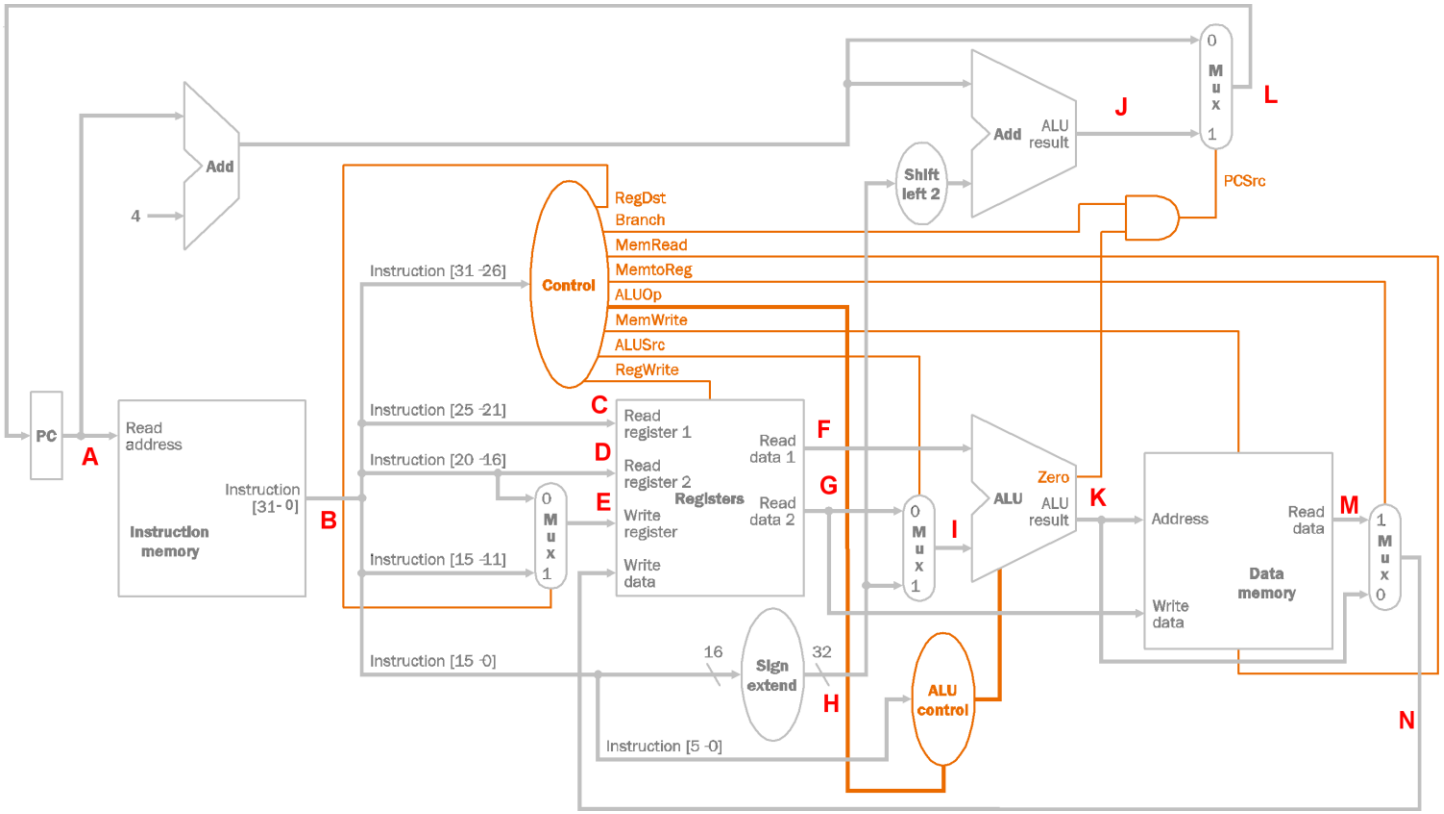
- Step 3 (execution):

ALUSrc = 0, therefore I= value of Rt

J = 0000 0000 0000 0001 0000 0000 1001 0000+PC+4

ALU Control = 0000, therefore K = (value of Rs) bitwise_AND (value of Rt),

Branch =0 => **PCSrc=0**, therefore L = PC+4

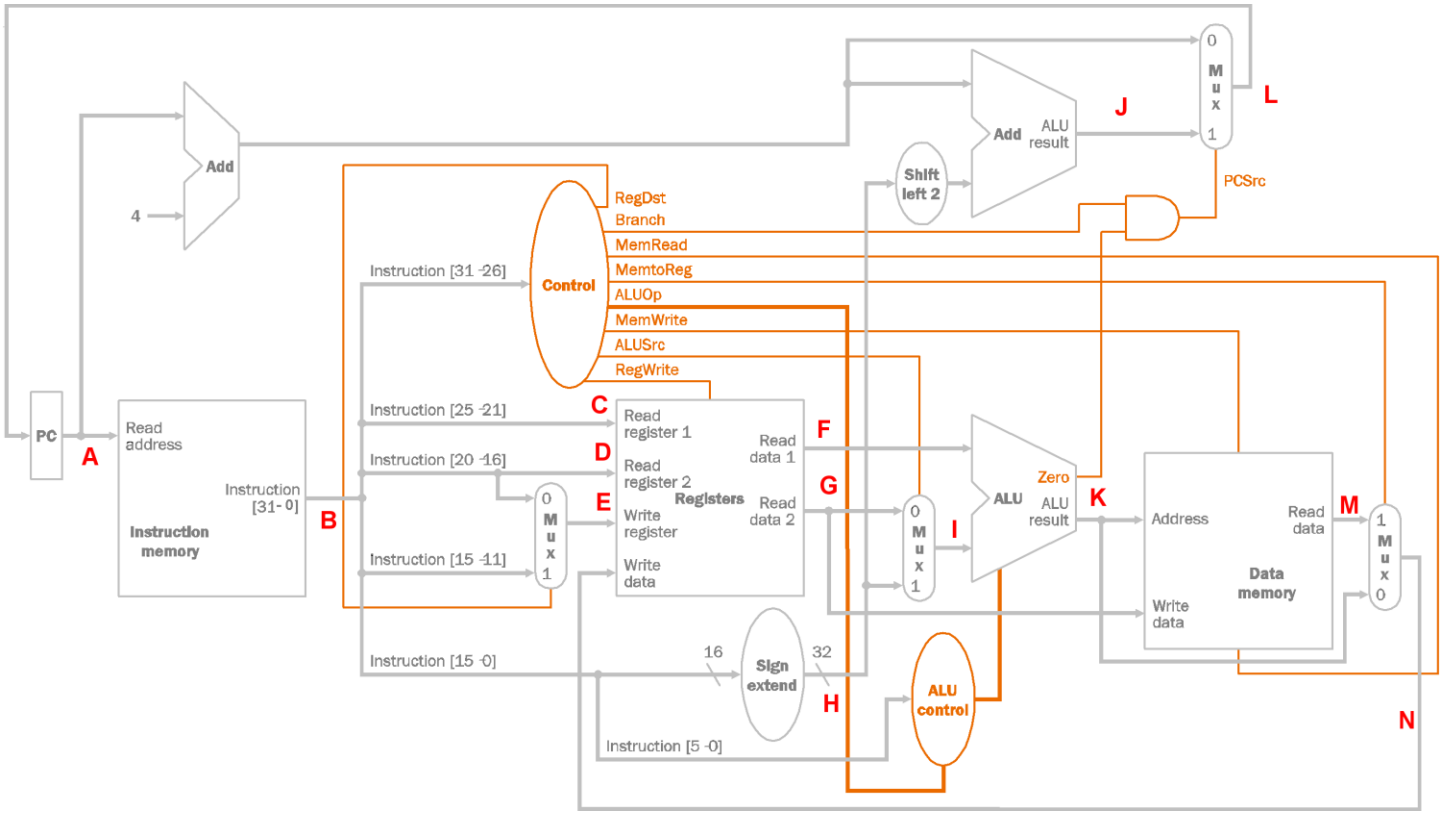


Execution of "AND $t0, $t1, $t2"

- Step 4 (memory)

Instruction fetched:
000000 01001 01010 01000 00000 100100

MemRead=0, therefore M = Empty (can't read)



Execution of "AND $t0, $t1, $t2"

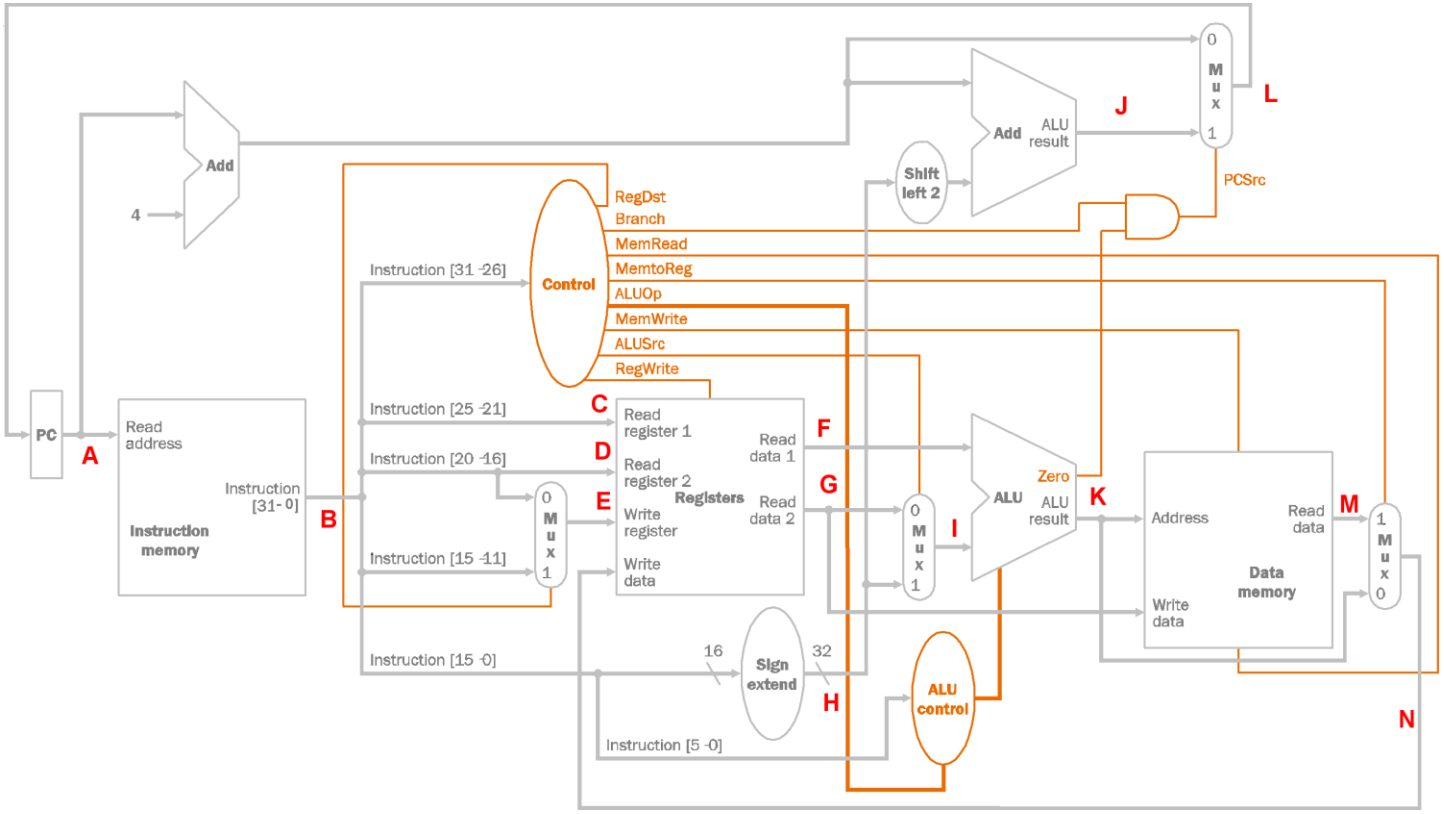
- Step 5 (write back):

Instruction fetched:
000000 01001 01010 01000 00000 100100

MemtoReg=0, therefore N= Results of the ALU

= (value of Rs) bitwise_AND (value of Rt)

RegWrite=1, RegDst=1, therefore N is written back to register 01000 ($t0)



Single Cycle Datapath

Review of the control signals

- the two hardware control units
- the effects of the control signals

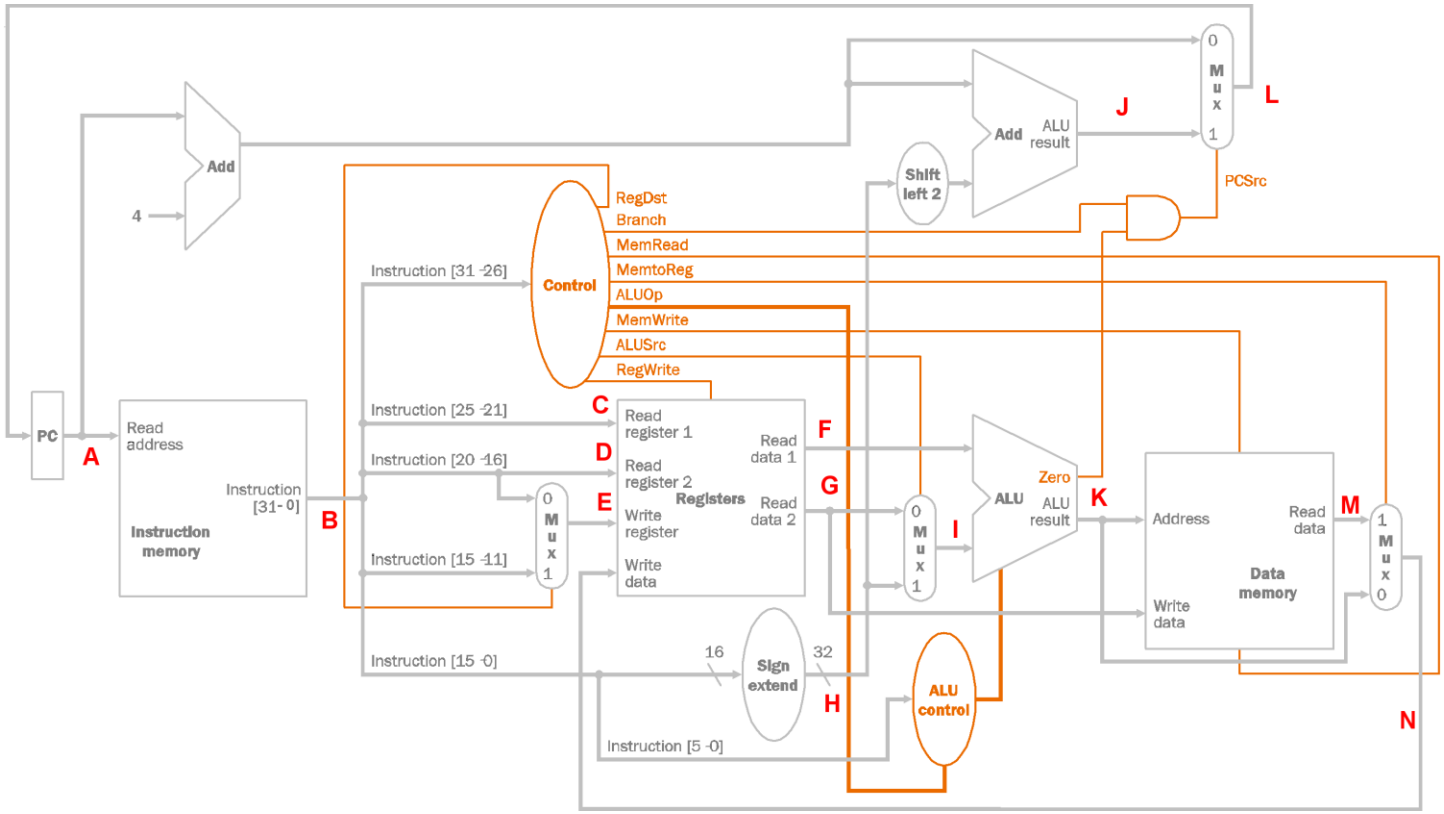
Sample execution of an instruction

- instruction fetch, decoding, executing, memory referencing and writing back
- Exercises

Exercises

Exercises

Question 1: Repeat the execution of the I-type instruction "LW $t0, 8($t1)" by filling the values for A-N and also the control signals.



Exercises

Question 1: Repeat the execution of the I-type instruction "LW $t0, 8($t1)" by filling the values for A-N and also the control signals.

Solution:

lw $t0, 8($t1) = 100011 01001 01000 0000 0000 0000 1000

Step 1

A = address of the instruction

B = 100011 01001 01000 0000 0000 0000 1000

Step 2

C = 01001, D = 01000, E = 01000, F = value of $t1, G = value of $t0

H = 0000 0000 0000 0000 0000 0000 0000 1000

Controls: RegDst=0, Branch=0, MemRead=1, MemtoReg=1, ALUop=00, MemWrite=0, ALUSrc=1, RegWrite=1, ALU Control=0010

Exercises

Step 3

I = 0000 0000 0000 0000 0000 0000 0000 1000

J = 0000 0000 0000 0000 0000 0000 0010 0000+PC+4

K = value of $t1+0000 0000 0000 0000 0000 0000 0000 1000, L=PC+4

Step 4

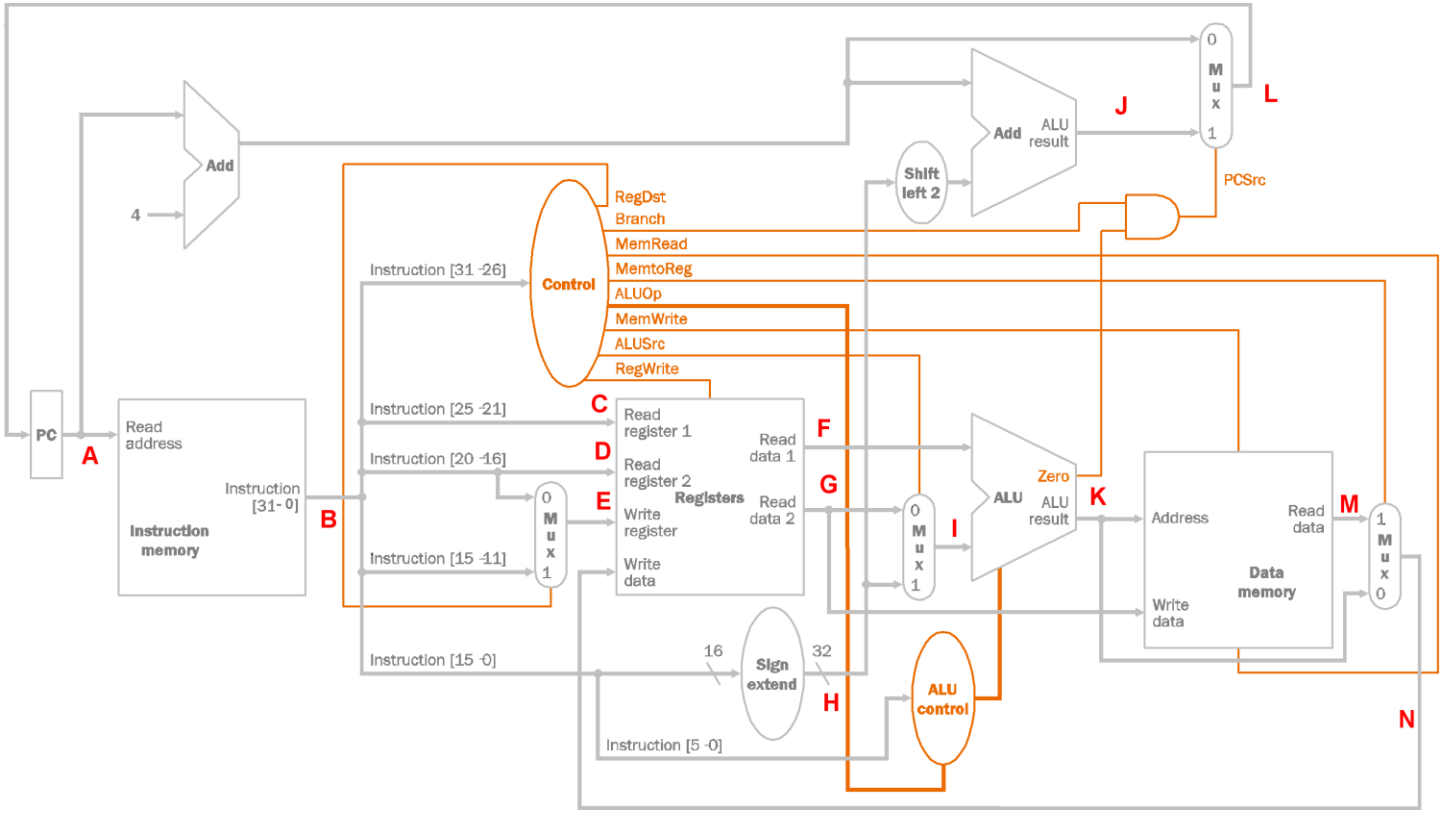
M = mem[value of $t1+0000 0000 0000 0000 0000 0000 0000 1000]

Step 5

N = mem[value of $t1+0000 0000 0000 0000 0000 0000 0000 1000]

Exercises

Question 2: Repeat the execution of the I-type instruction "BEQ $t0, $t1,100" by filling the values for A-N and also the control signals. Assume $t0==$t1.



Question 2: Repeat the execution of the I-type instruction "BEQ $t0, $t1,100" by filling the values for A-N and also the control signals. Assume $t0==$t1.

Solution:

beq $t0, $t1, 100 = 000100 01000 01001 0000 0000 0110 0100

Step1

A = address of the instruction

B = 000100 01000 01001 0000 0000 0110 0100

Step2

C = 01000, D = 01001, E = either 01001 or 00000, F = value of $t0, G = value of $t1
H = 0000 0000 0000 0000 0000 0000 0110 0100

Controls: RegDst=X, Branch=1, MemRead=0, MemtoReg=X, ALUop=01, MemWrite=0, ALUSrc=0, RegWrite=0, ALU Control=0110

Exercises

Step3

I=value of $t1

J= 00 0000 0000 0000 0000 0000 0110 010000+PC+4

K=0

Zero=1 (because $t0==$t1)

L= 00 0000 0000 0000 0000 0000 0110 010000+PC+4 (because branch=1 and Zero=1)

Step4

M=Empty (can't read)

Step5

MemtoReg=X, so N can't be determined

Single Cycle Datapath

Review of the control signals

- the two hardware control units
- the effects of the control signals

Sample execution of an instruction

- instruction fetch, decoding, executing, memory referencing and writing back
- Exercises

Exercises

Exercises

Question 1: Repeat the execution of the following instructions by filling the values for A-N and also the control signals :

1) SW $t0, 8($t1)
2) ADD $t0, $t1, $t2
3) OR $t0, $t1, $t2

Exercises

Question 2: Modify the datapath on slide 11 to support the execution of an I-type instruction "LUI $t0, 100". The instruction shifts the immediate value by 16 bits to the left (lower 16 bits are filled with zeroes) and stores the shifted value to the register. Include all the control signals.

(encoding 000111 00000 ttttt iiiiiiiiiiiiiiii, t: destination register, i:immediate).

Conclusion

- Today we have reviewed:
  - the control units of the single cycle datapath,
  - the effects of the individual control signals,
  - how an instruction is executed through the guidance of the control signals.