COMP2611: Computer Organization

Multi-cycle Datapath

Overview

- You will learn the following in this tutorial:
  - how an instruction is executed through the guidance of the control signals.
  - how the registers of the datapath change in each clock cycle as an instruction is executing.

Multi-cycle Datapath

Datapath implementation

- the updating of the datapath registers in each execution cycle
- the standard setting of the control signals
- exercise

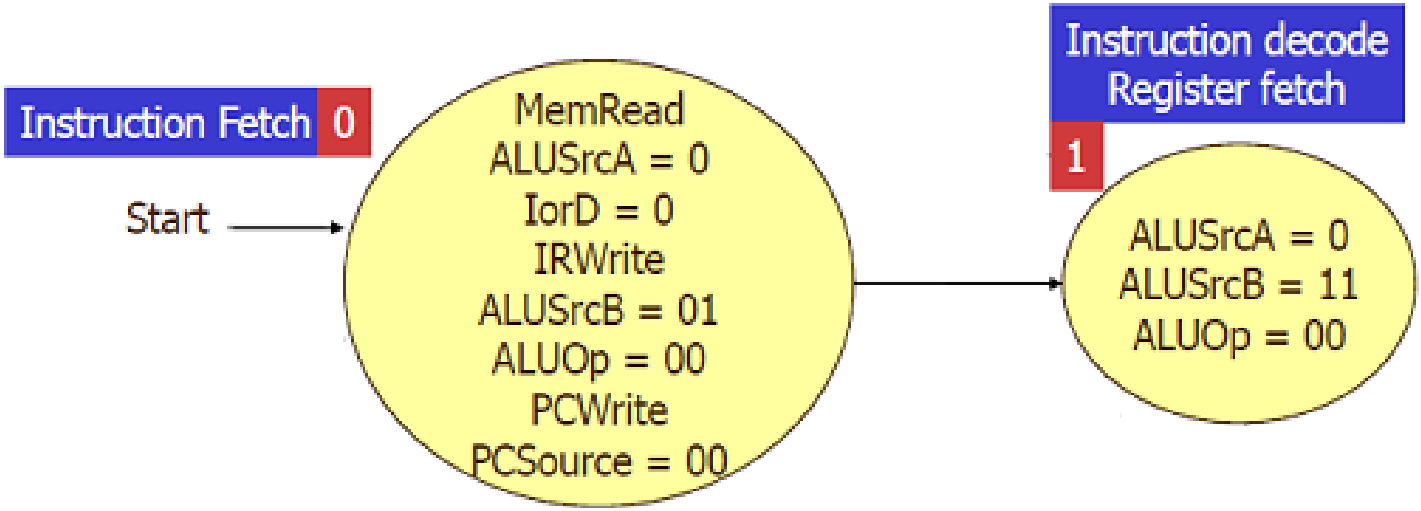
Exercises

Datapath implementation

- The clock cycle of the single-cycle datapath is often very long, leading to a low efficiency
- To improve efficiency, the multi-cycle datapath breaks an instruction execution into a series of steps (cycles):
  - Instruction fetch
  - Instruction decode and register reads
  - ALU operation
  - Memory access (when necessary)
  - Write data back (to a register)
- In each cycle of the execution, the execution results are always produced at the input of a register.
- The value of a register is not updated to its input in the current cycle until the next cycle starts (as it is often built using clock edge-trigger flip-flops).

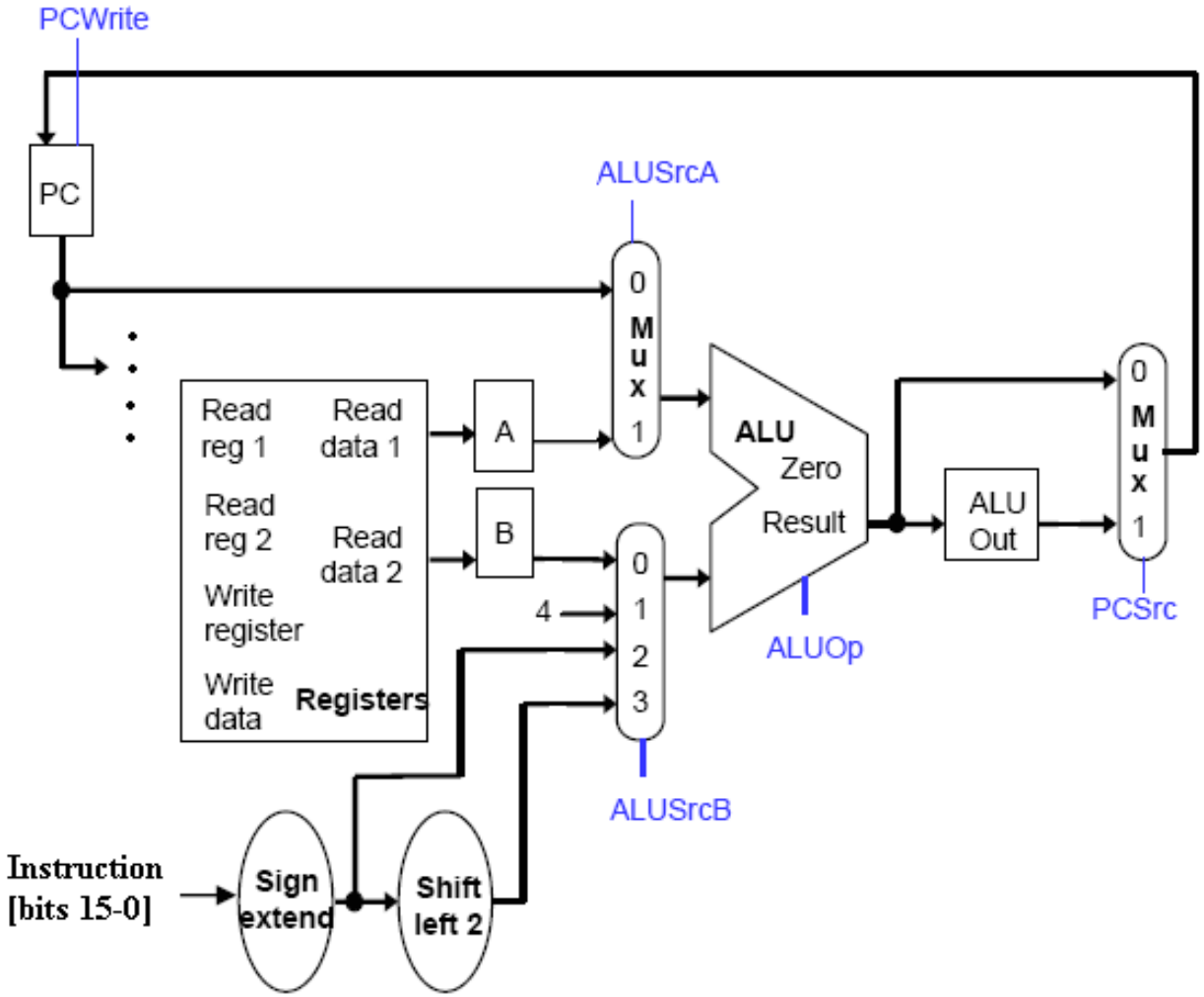
Control signals

- In the standard setting, the first two states of the control signals in the multi-cycle datapath are always these:



State 0

- In state 0, the addition PC + 4 is done and the result is put in the input of PC. Again, the value of PC is updated to it when the next cycle starts.

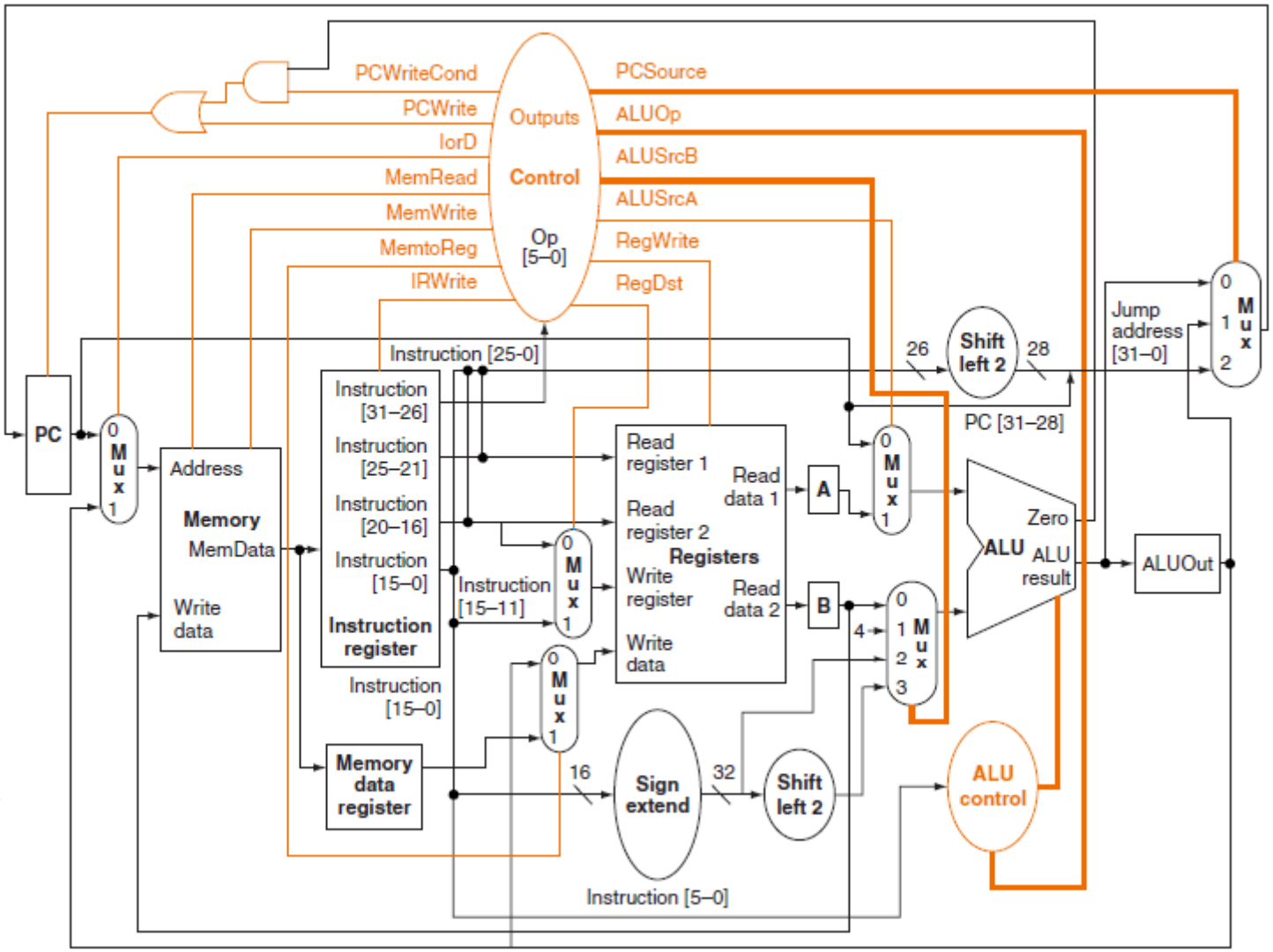


State 1

- Then, in state 1, PC has the original address in PC plus 4.
- The sign-extended and shifted version of the least significant 16 bits (bit 0 to bit 15) of the instruction is added to the current value of PC.
- If the instruction is a "branch" instruction, the result is used to update PC (in the next cycle). Otherwise, the result is not used (by setting the control signals accordingly).

Exercises

Question 1: Consider the following multi-cycle datapath for the instruction "jr".

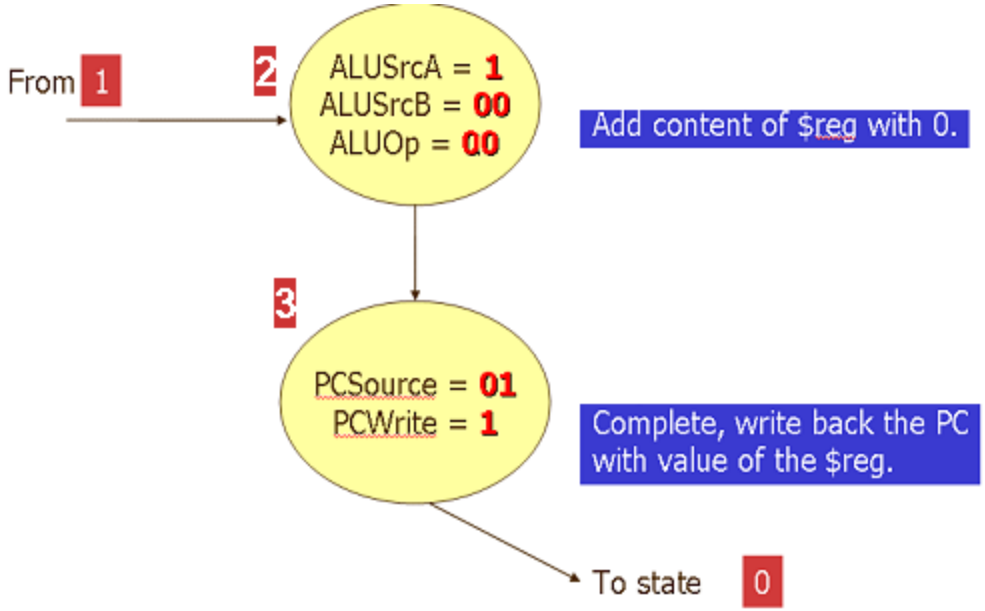


Exercise

(a) Draw the third and fourth states of the control signals for the execution of the following instruction:

Assembly code (jr $reg)	jr $t0
Machine code:	**000000 01000 00000 00000 00000 001000** **OpCode Rs Rt Rd Shift function**

Solution:



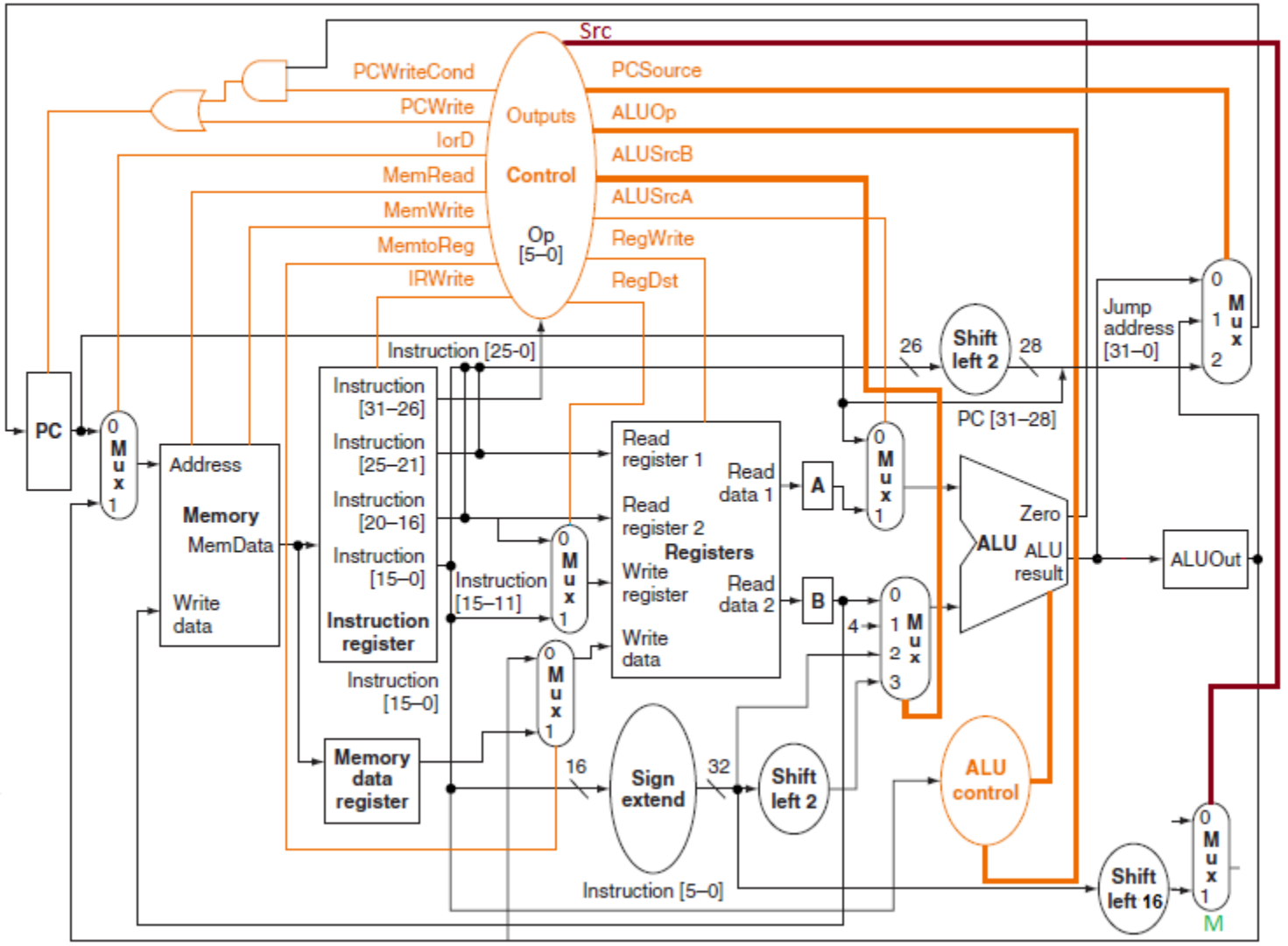
(b) Fill in the table below with the values (in hex. format) of the registers in the beginning of each execution cycle of "jr $t0", based on the values given in the first cycle. The value of $t0 is 0x000004F1. If a value is unknown, write down "x" instead.

Solution:

1st cycle	PC = 001f0000, IR = x, A = x, B = x, ALUOut = x, MDR (Memory Data Register) = x
2nd cycle	PC = 001f0004, IR = 01000008, A = x, B = x, ALUOut = 001f0004, MDR = 01000008
3rd cycle	PC = 001f0004, IR = 01000008, A = 000004f1, B = 00000000, ALUOut = 001f0024 (sign-extended 0008 to 32-bit, shifted left by 2 bits and then added with PC together), MDR = 01000008
4th cycle	PC = 001f0004, IR = 01000008, A = 000004f1, B = 00000000, ALUOut = 000004f1, MDR = 01000008
After 4th cycle	PC = 000004f1

Exercises

Question 2: Consider the following partial implementation of the multi-cycle datapath for the instruction "lui".



Exercises

(a) Complete the datapath to implement the instruction "lui" by making only the following changes:

1. Break up one of the existing data paths,

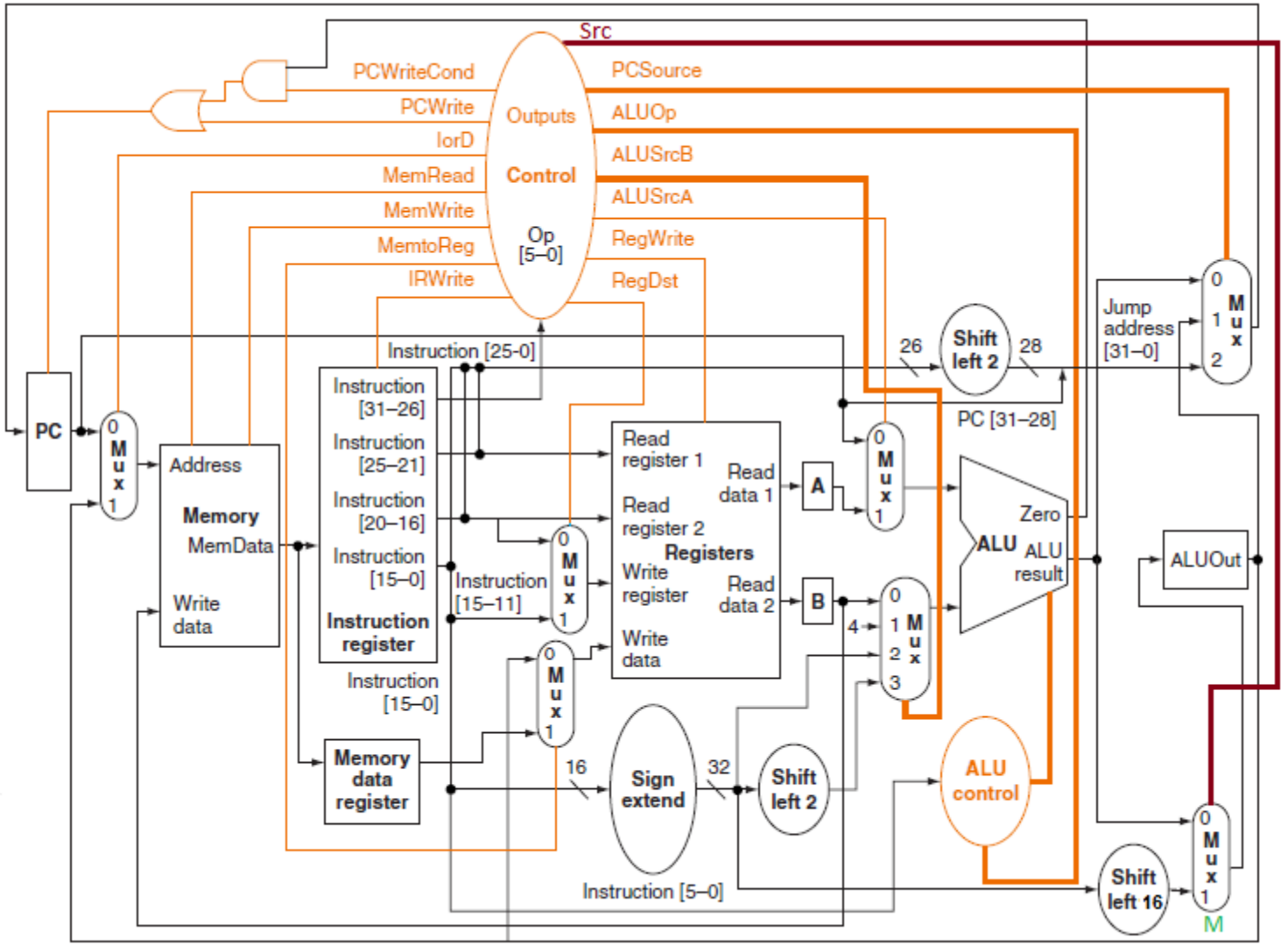
2. connect one of the two broken ends to the input 0 of the multiplexor M, and

3. connect the other broken end to the output of M.

Also, the standard setting of the control signals in the first two cycles (learnt from the lecture) must not have to be modified for your answer. The new signal Src for M must be set to 1 in only the third cycle of the execution.

Exercises

Solution:



(b) Fill in the table below with the control signals in each execution cycle of "lui". Your answer in this part must be in accord with that in part (a). If different values of a signal in a particular cycle do not change the execution in that cycle, write down "x" for it.

	PCWriteCond	PCWrite	IorD	MemRead	MemWrite	MemToReg	IRWrite
3rd cycle							
4th cycle							
5th cycle							

	Src	PCSource	ALUOp	ALUSrcB	ALUSrcA	RegWrite	RegDst
3rd cycle							
4th cycle							
5th cycle							

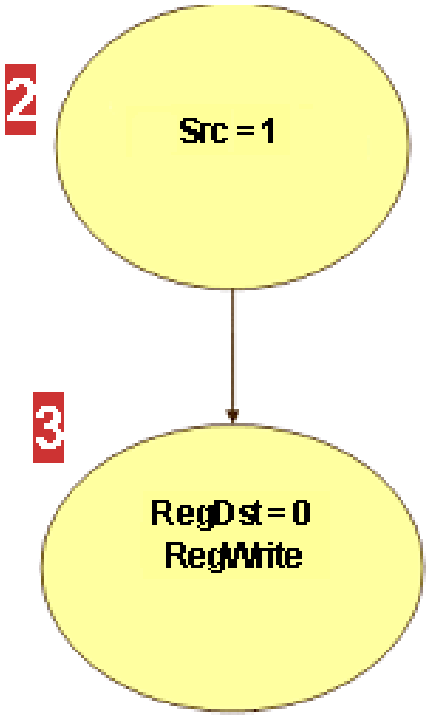
Solution:

	PCWriteCond	PCWrite	IorD	MemRead	MemWrite	MemToReg	IRWrite
3rd cycle	0	0	x	0	0	x	0
4th cycle	0	0	x	0	0	0	0
5th cycle							

	Src	PCSource	ALUOp	ALUSrcB	ALUSrcA	RegWrite	RegDst
3rd cycle	1	x	x	x	x	0	x
4th cycle	x	x	x	x	x	1	0
5th cycle							

(c) Draw all the states, after the state 0 and state 1, of the control signals according to your answer in part (a) only.

Solution:



Multi-cycle Datapath

Datapath implementation

- the updating of the datapath registers in each execution cycle
- the standard setting of the control signals
- exercises

Exercise

Exercises

Question 1: Redo all the parts of Question 2 of the previous exercise with this change: in part (a) the signal Src must be set to 1 in only the second cycle of the execution. The standard setting of the control signals in the first two cycles are still used.

Conclusion

- You have learnt:
  - how an instruction is executed through the guidance of the control signals.
  - how the registers of the datapath change in each clock cycle as an instruction is executing.