Outline of Lecture

- The ALU Control Operation & Design
- The Datapath Control Operation & Design

ALU Control

- After the design of partial single MIPS datapath, we need to add the *control unit* that controls the whole operation of the datapath (generatse appropriate signals for the operation of the datapath).

- From the previous chapter - when we designed the ALU - these are the ALU control signals that we came up with:

ALU Control lines	Function
0 00	AND
0 01	OR
0 10	ADD
1 10	SUB
1 11	SLT

- For `load` and `store`, we need to calculate the memory address by addition.

- For R-type instructions, the ALU needs to perform one of five operations (`Subtract, add, AND, OR, SLT`) - depending on the value of the 6-bit funct filed.

ALU Control

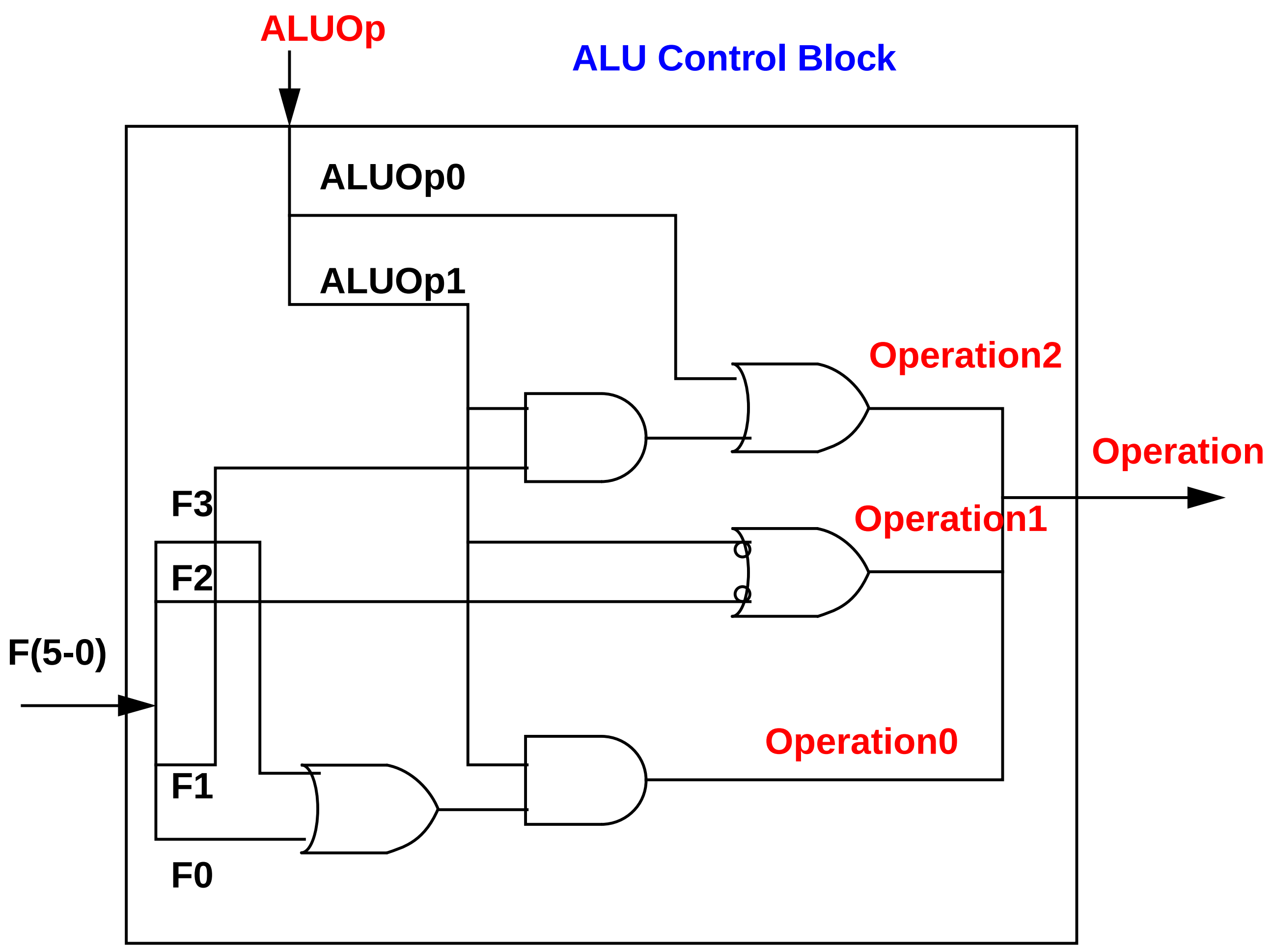
- For `branch equal`, the ALU must perform a subtraction.
- By using the *function field* (6 bits for MIPS), the table above can be expanded to cover a wider range of instructions.

Instruction opcode	ALUOp	Instruction operation	Function code	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	010
SW	00	store word	XXXXXX	add	010
Branch equal	01	branch equal	XXXXXX	subtract	110
R-type	10	add	100000	add	010
R-type	10	subtract	100010	subtract	110
R-type	10	AND	100100	and	000
R-type	10	OR	100101	or	001
R-type	10	SLT	101010	slt	111

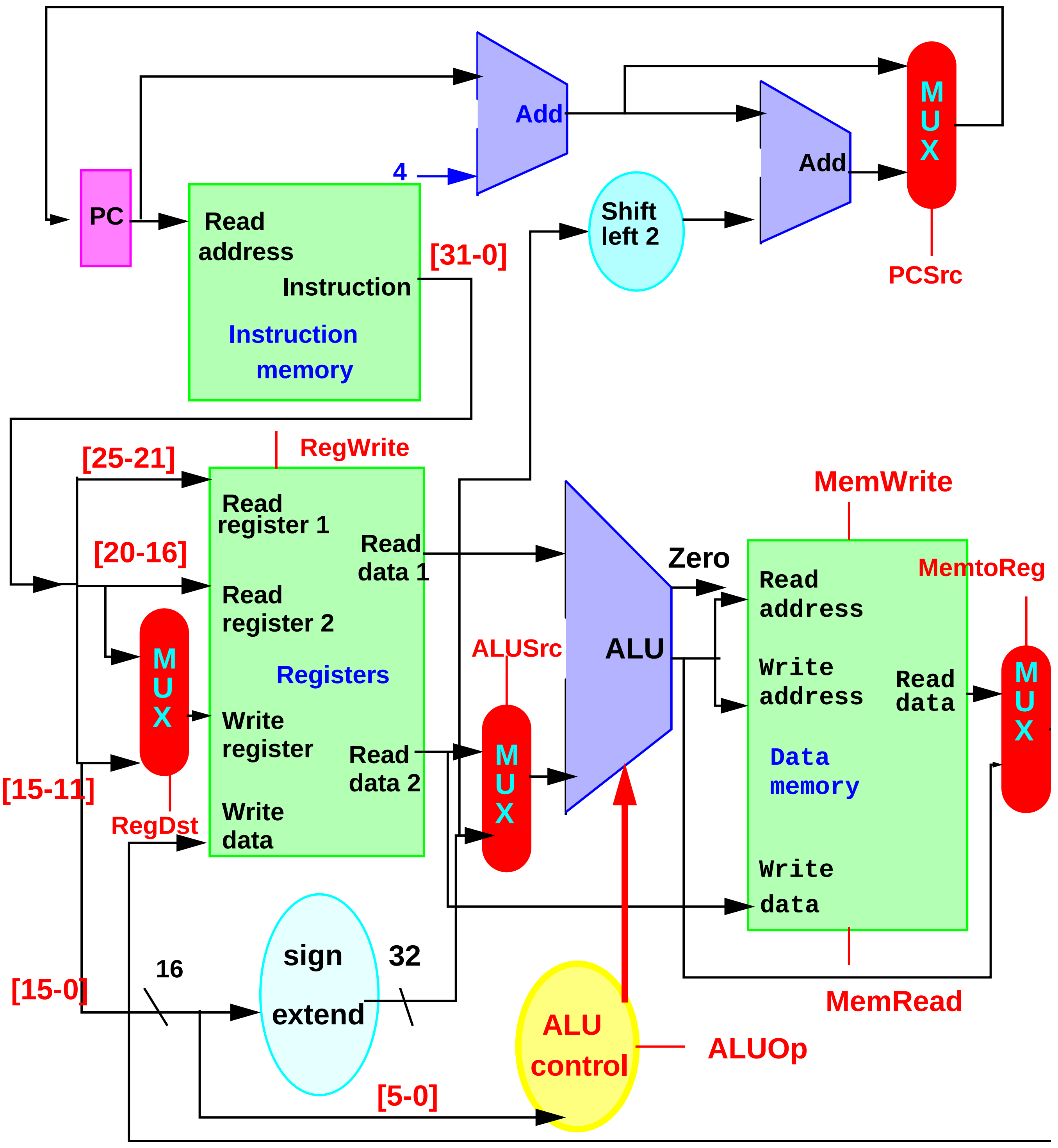
- In order to design the logic for this ALU control unit, we create a truth table.
- The input of the truth table is a 2-bit ALUOp and a 6-bit Function code, and the output is a 3-bit ALU control signals

ALUOp		Function code						Operation
ALUOp1	ALUOp2	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	010
X	1	X	X	X	X	X	X	110
1	X	X	X	0	0	0	0	010
1	X	X	X	0	0	1	0	110
1	X	X	X	0	1	0	1	001
1	X	X	X	1	0	1	0	111

- **Using the above truth table, we can design the ALU control unit:**



Adding the ALU control unit to the Datapath



The function of each of the control signals in the data-path is given in the following table:

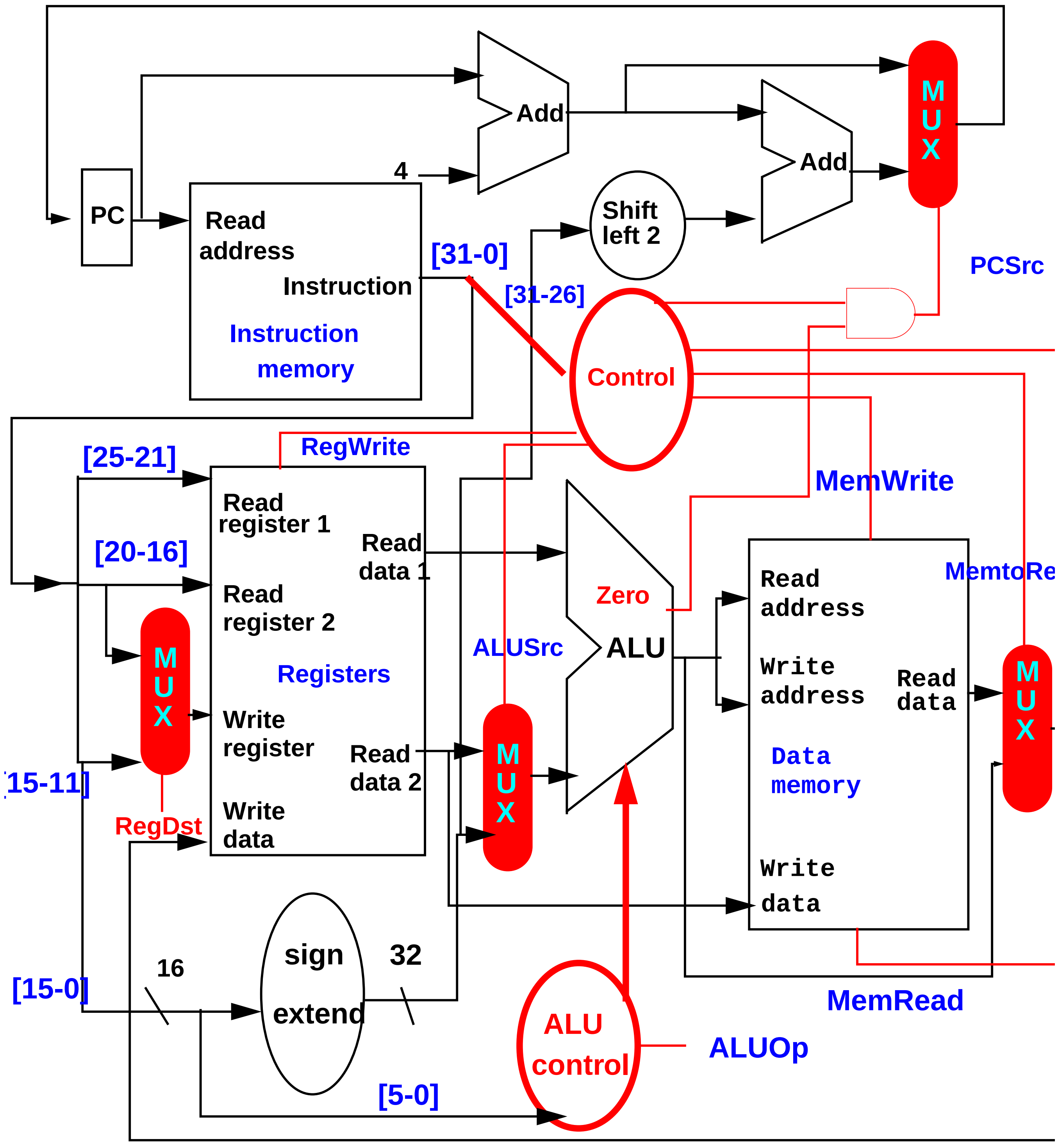
Signal name	Effect when deasserted	Effect when asserted
MemRead	None	Data memory contents at the read address are put on read data output
MemWrite	None	Data memory contents at address given by write address is replaced by value on write data input
ALUSrc	The second ALU operand comes from the second register file output	The second ALU operand is the sign-extended lower 16-bits of the instruction
RegDst	The register destination number for the write register comes from the rt field	The register destination number for the write register comes from the rd field
RegWrite	None	The register on the write register input is written into with the value on the write data input
PCSrc	The PC is replaced by the output of the adder that computes the value of PC+4	The PC is replaced by the output of the adder that computes the branch target
MemtoReg	The Value fed to the register write data input comes from the ALU	The value fed to the register write data input comes from the data memory

The control signals can be set by the datapath control unit based only on the *opcode field of the instruction*.

The only exception is the ***PCSrc*** control line. It should be set if the instruction is a branch on equal and the Zero output of the ALU is true.

To generate the PRSrc signal, we will AND toegther a signal from the control unit, called `branch`, with the Zero signal out of the ALU.

The Whole Datapath including all the control signals will be as follows:


PC
Read address
Instruction
Instruction memory
4
Add
Add
Shift left 2
M U X
PCSrc
[31-0]
[31-26]
Control
RegWrite
[25-21]
[20-16]
[15-11]
M U X
RegDst
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
ALUSrc
M U X
Zero
ALU
MemWrite
Read address
Write address
Data memory
Write data
Read data
MemtoRe
M U X
MemRead
[15-0]
16
sign extend
32
[5-0]
ALU control
ALUOp

The logic circuits for the datapath control unit.

The setting of the control lines is completely determined by the opcode fields of the instruction.

Instruction	Reg Dst	ALU Src	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALU Op1	ALU Op2
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

The input to the datapath control unit:

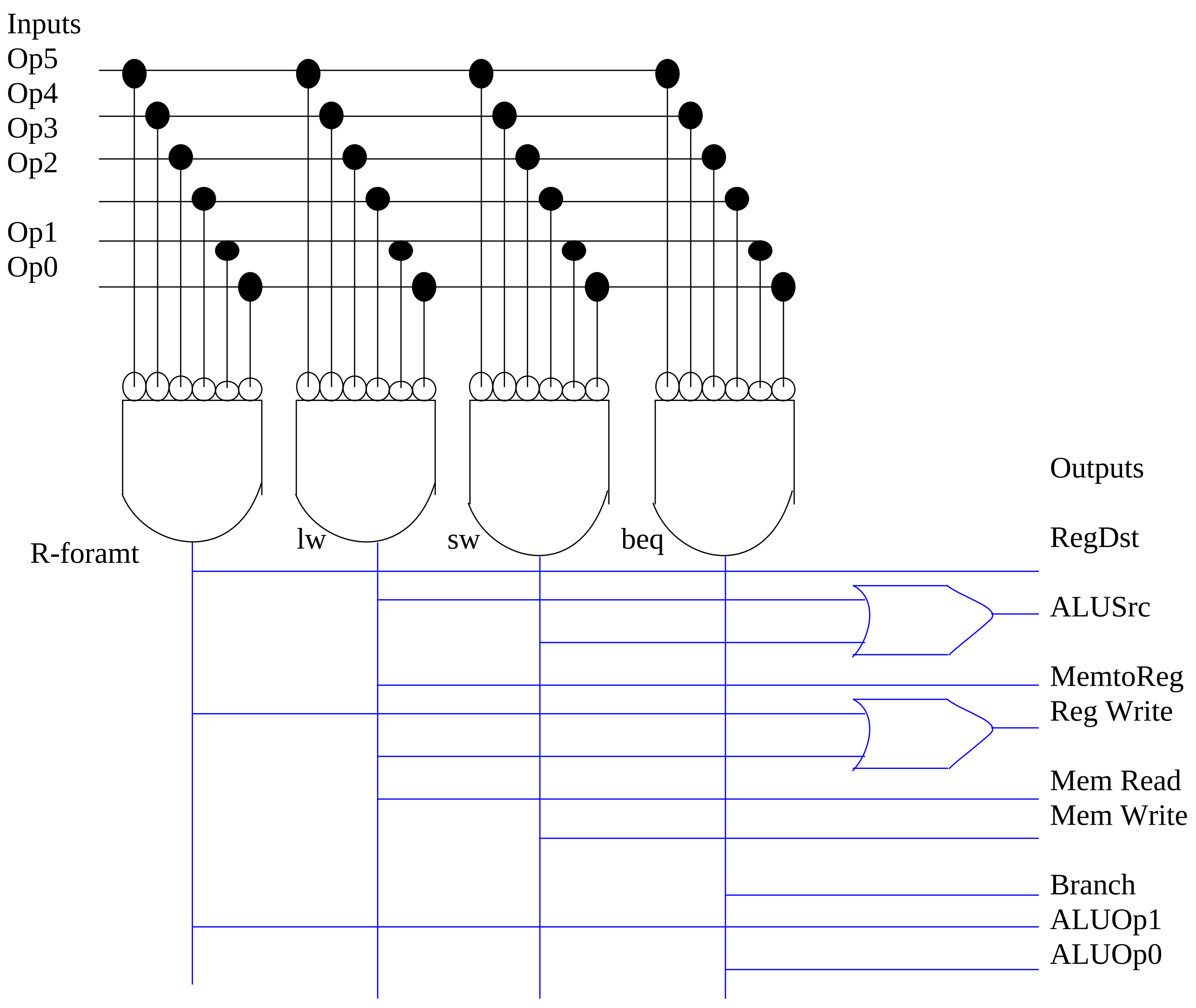
Name	Opcode in decimal	Opcode in binary					
		Op5	Op4	Op3	Op2	Op1	Op0
R-format	0	0	0	0	0	0	0
lw	35	1	0	0	0	1	1
sw	43	1	0	1	0	1	1
beq	4	0	0	0	1	0	0

Truth Table of the Datapath Control Unit

		R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp2	0	0	0	1

Implementation

- We can easily implement the above truth table using simple gates. If we used all possible instructions, then a PLA would be more appropriate.



Further Reading

Chapter 5 & Appendix B. David A. Patterson and John L. Hennessy. *Computer Organization & Design: The Hardware / Software Interface*. Morgan Kaufman Publishers, 1998 (page 350-370).