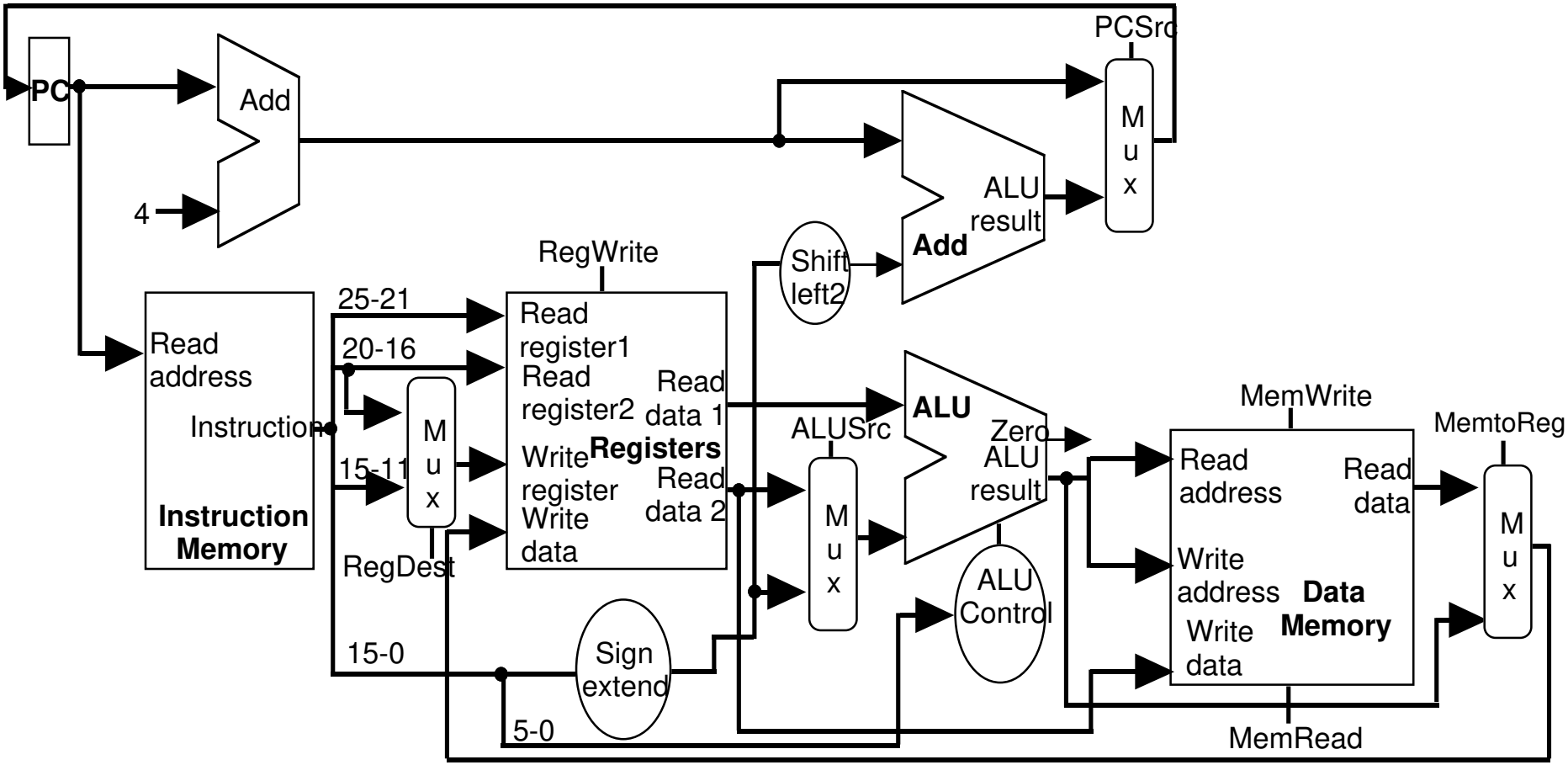
Controlling the Datapath

For the datapath to execute instructions properly, control signals must be presented to the datapath components. Generally, the control signals come from the opcode, but for R-type instructions they also come from the function field. Also, for branching instructions, the result of the comparison is emitted from the ALU, as the ZERO signal.

@1997 LS All Rights Reserved

Datapath Schematic

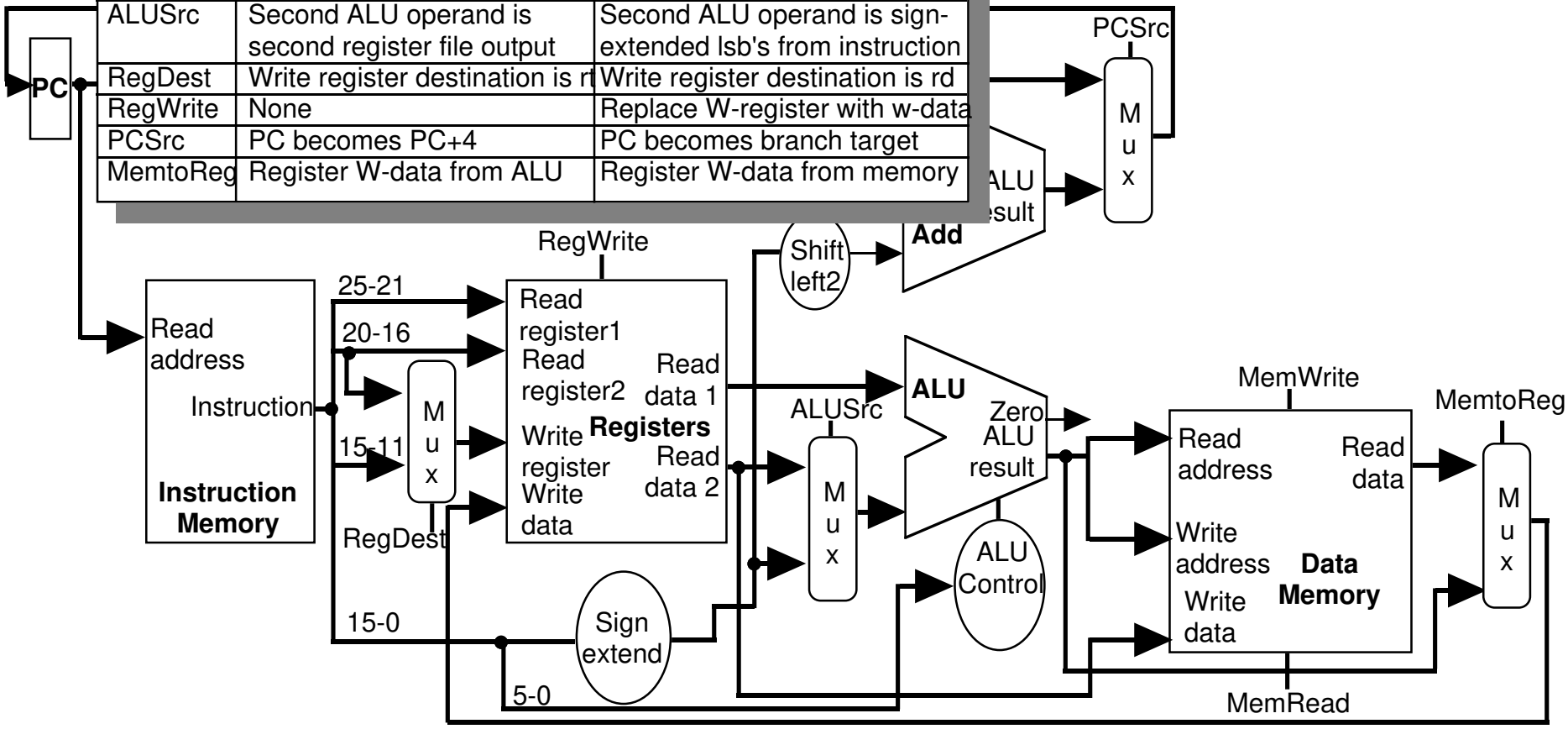
Control lines determine the operation of the datapath components



@1997 LS All Rights Reserved

Control Signal Behavior

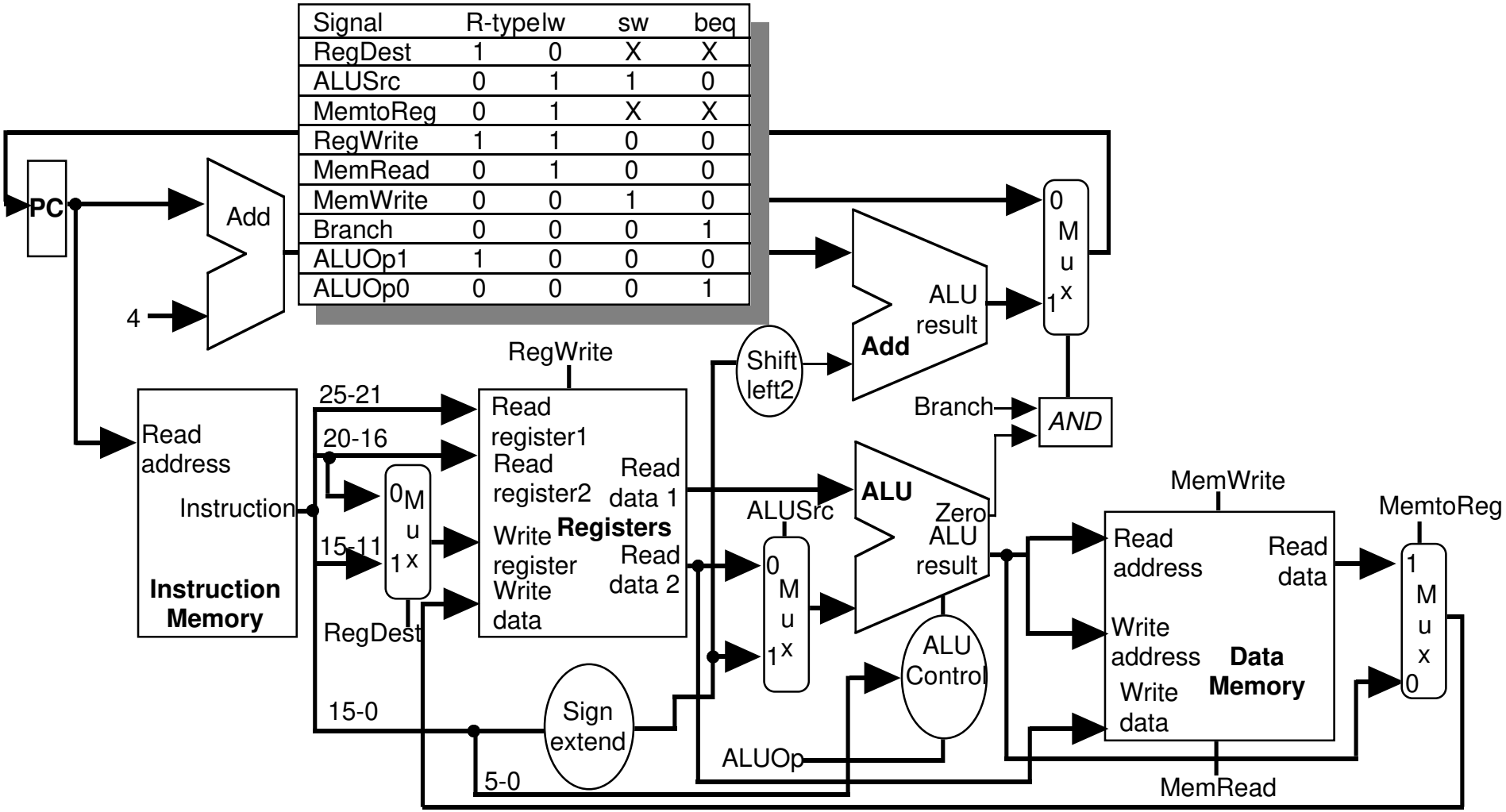
Signal	Effect when Deasserted	Effect when Asserted
MemRead	None	Emit mem contents from R-addressed location as R-data
MemWrite	None	Replace mem contents at W-addressed location as W-data
ALUSrc	Second ALU operand is second register file output	Second ALU operand is sign-extended lsb's from instruction
RegDest	Write register destination is rt	Write register destination is rd
RegWrite	None	Replace W-register with w-data
PCSrc	PC becomes PC+4	PC becomes branch target
MemtoReg	Register W-data from ALU	Register W-data from memory



@1997 LS All Rights Reserved

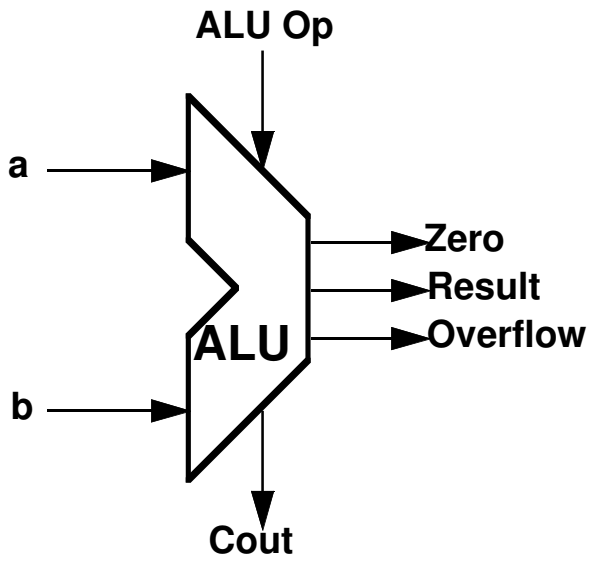
Control Line Settings

Signal	R-type	lw	sw	beq
RegDest	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
ALUOp0	0	0	0	1



@1997 LS All Rights Reserved

Controlling the ALU

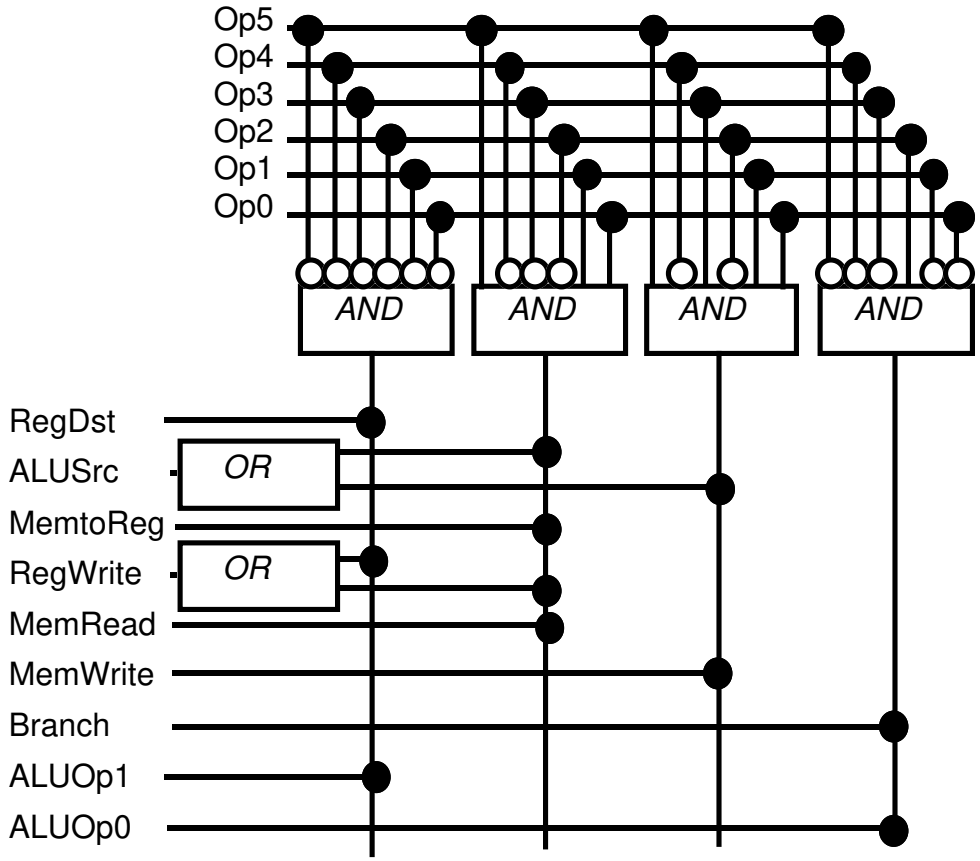


Opcode	ALUop	Funct	ALU Action	ALU Control
lw	00	XXXXX	add	010
sw	00	XXXXX	add	010
beq	01	XXXXX	subtract	110
add	10	100000	add	010
subtract	10	100010	subtract	110
and	10	100100	and	000
or	10	100101	or	001
slt	10	101010	set-on-lt	111

@1997 LS All Rights Reserved

Computing the Control

Signal	R-type	lw	sw	beq
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
RegDest	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
ALUOp0	0	0	0	1



@1997 LS All Rights Reserved

Computing Cycle Time

Suppose the following times apply ...

- Memory units: 10ns
- ALU and adders: 10ns
- Register file ref: 5ns
- All other operations are 0ns.

Charges for instructions are ...

R-format:	30ns
Load inst:	40ns
Store inst:	35ns
Branch:	25ns
Jump:	10ns

Instruction Mix GCC	
22%	Load
11%	Store
49%	R-type
16%	Branch
2%	Jump

CPU Clock Cycle = 40x.22+35x.11+30x.49+25x.16+10x.02=31.6

CPU Perfv/CPU Perfs = 40/31.6=1.27

@1997 LS All Rights Reserved