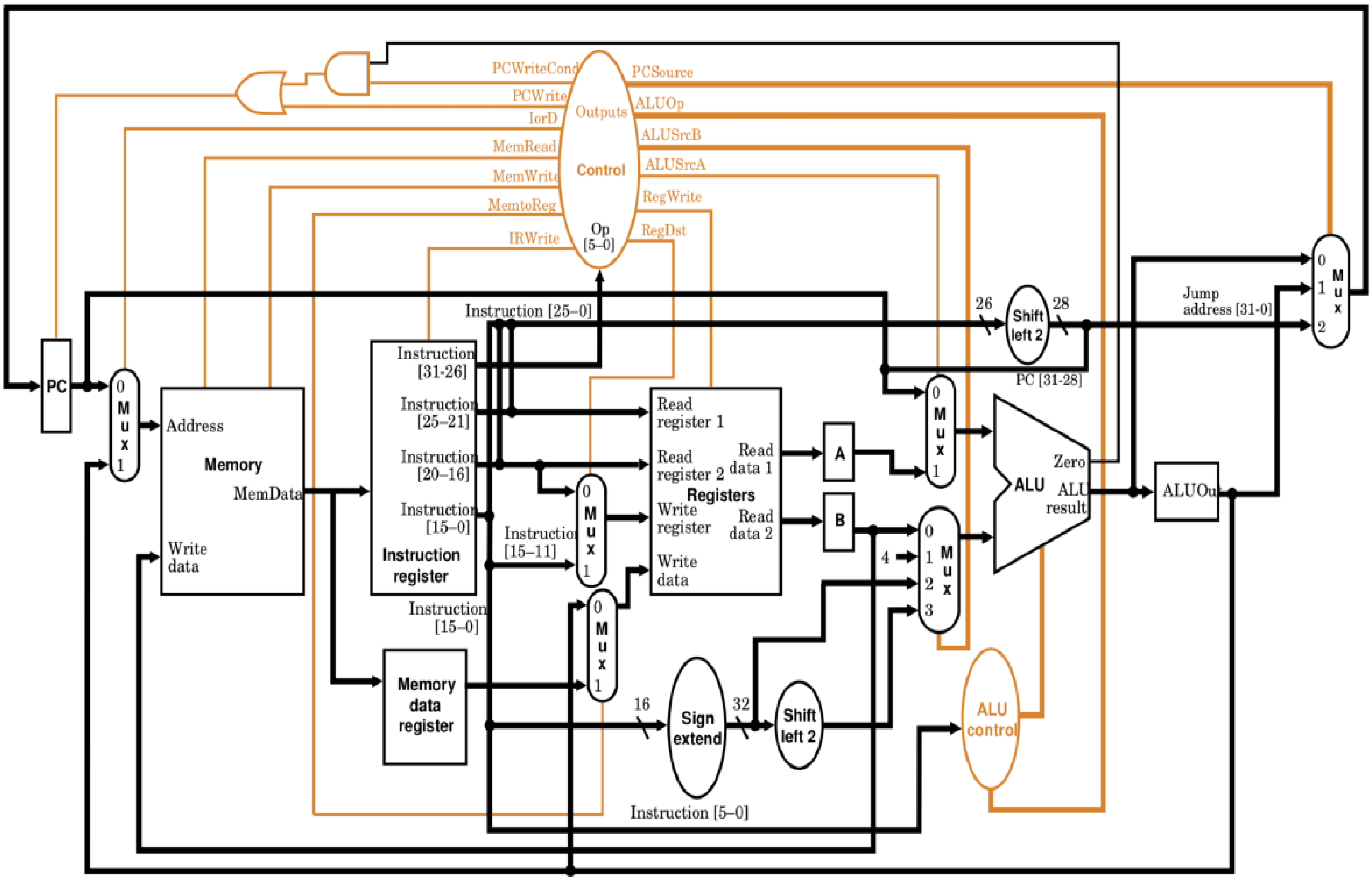
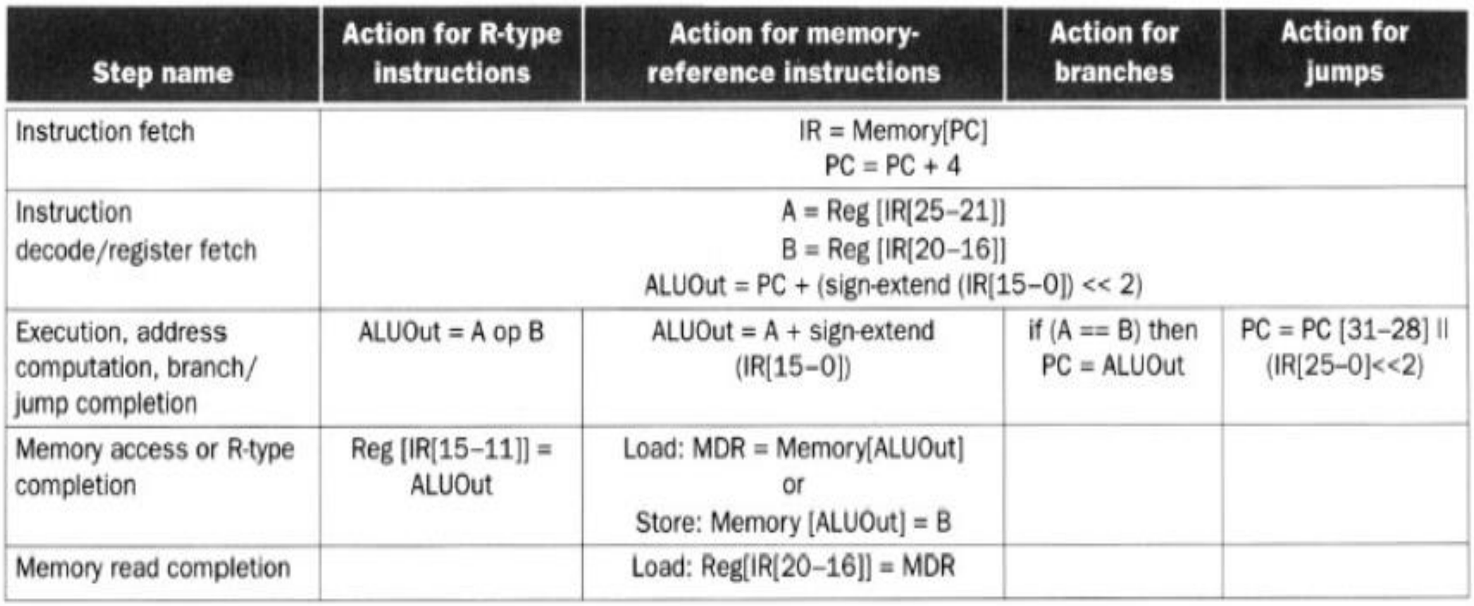
MULTI-CYCLE DATAPATH AND CONTROL



Multi-cycle datapath: summary



Step name	Action for R-type instructions	Action for memory-reference instructions	Action for branches	Action for jumps

Instruction fetch	IR = Memory[PC]
	PC = PC + 4			

Instruction decode/register fetch	A = Reg [IR[25–21]]
	B = Reg [IR[20–16]]
	ALUOut = PC + (sign-extend (IR[15–0]) << 2)			

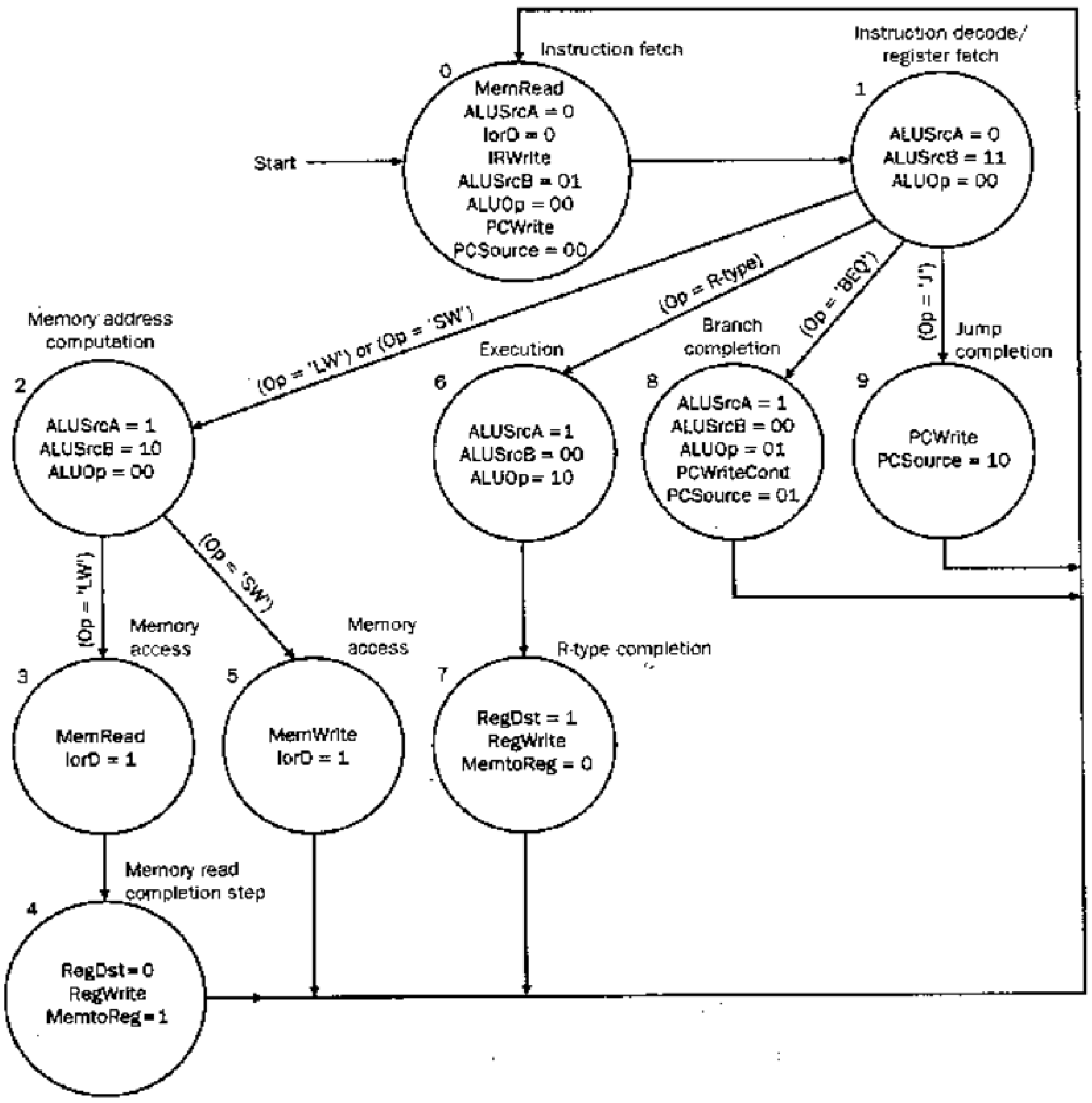
Execution, address computation, branch/jump completion	ALUOut = A op B	ALUOut = A + sign-extend (IR[15–0])	if (A == B) then PC = ALUOut	PC = PC [31–28] II (IR[25–0]<<2)
Memory access or R-type completion	Reg [IR[15–11]] = ALUOut	Load: MDR = Memory[ALUOut] or Store: Memory [ALUOut] = B		
Memory read completion		Load: Reg[IR[20–16]] = MDR		

Summary of execution steps

Instruction fetch, decode, register fetch same for all instructions

MULTI-CYCLE DATAPATH AND CONTROL

2-bit Signal	Value	Effect
ALUOp	00	The ALU performs an add operation.
	01	The ALU performs a subtract operation.
	10	The funct field of the instruction determines the operation.
ALUSrcB	00	The second input to ALU comes from the B register.
	01	The second input to ALU is 4.
	10	The second input to the ALU is the sign-extended, lower 16 bits of the Instruction Register (IR).
	11	The second input to the ALU is the sign-extended, lower 16 bits of the IR shifted left by 2 bits.
PCSource	00	Output of the ALU (PC+4) is sent to the PC for writing.
	01	The contents of ALUOut (the branch target address) are sent to the PC for writing.
	10	The jump target address (IR[25-0] shifted left 2 bits and concatenated with PC + 4[31-28]) is sent to the PC for writing.



Instruction fetch
0
MemRead
ALUSrcA = 0
IorD = 0
IRWrite
ALUSrcB = 01
ALUOp = 00
PCWrite
PCSource = 00
Start
Instruction decode/
register fetch
1
ALUSrcA = 0
ALUSrcB = 11
ALUOp = 00
(Op = 'LW') or (Op = 'SW')
(Op = R-type)
(Op = 'BEQ')
(Op = 'J')
Memory address
computation
2
ALUSrcA = 1
ALUSrcB = 10
ALUOp = 00
Execution
6
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 10
Branch
completion
8
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 01
PCWriteCond
PCSource = 01
Jump
completion
9
PCWrite
PCSource = 10
(Op = 'LW')
(Op = 'SW')
Memory
access
3
MemRead
IorD = 1
Memory
access
5
MemWrite
IorD = 1
R-type completion
7
RegDst = 1
RegWrite
MemtoReg = 0
Memory read
completion step
4
RegDst = 0
RegWrite
MemtoReg = 1