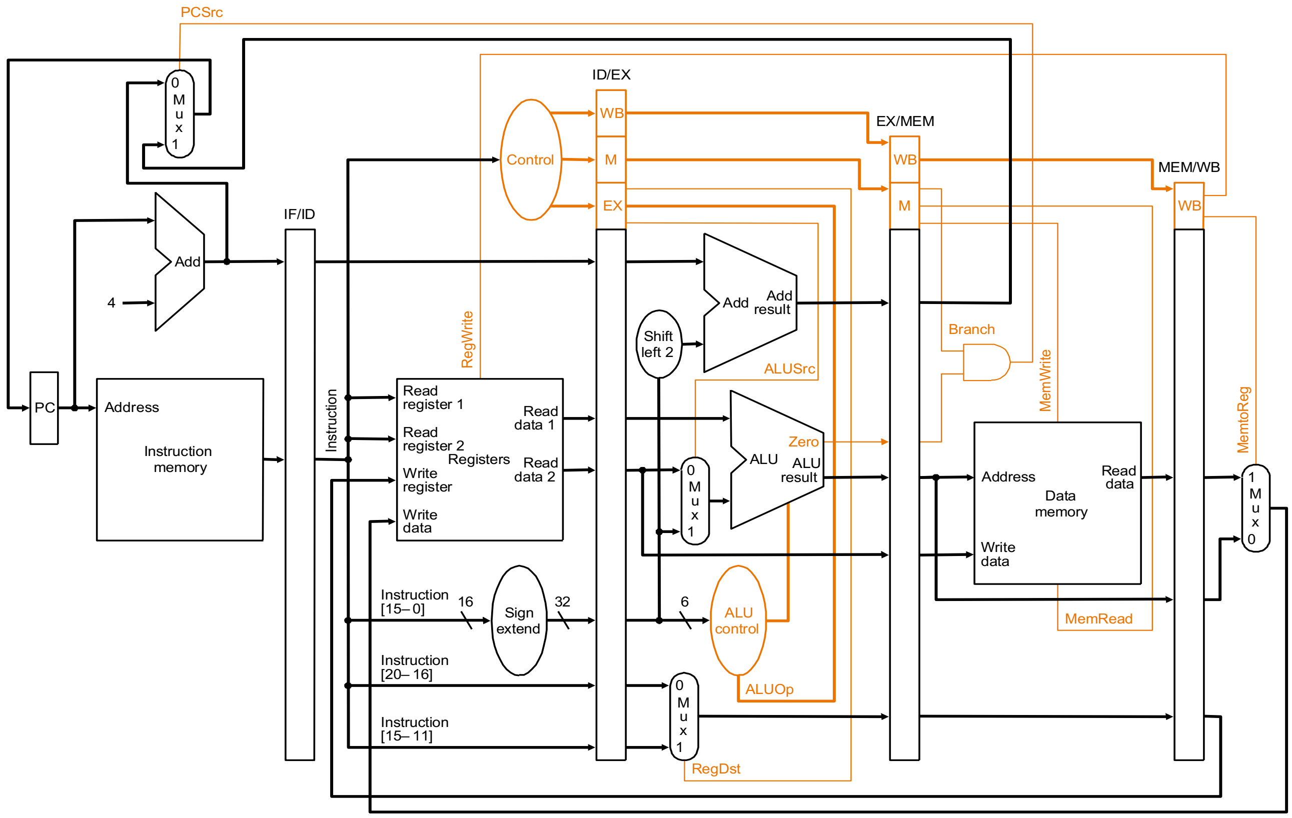
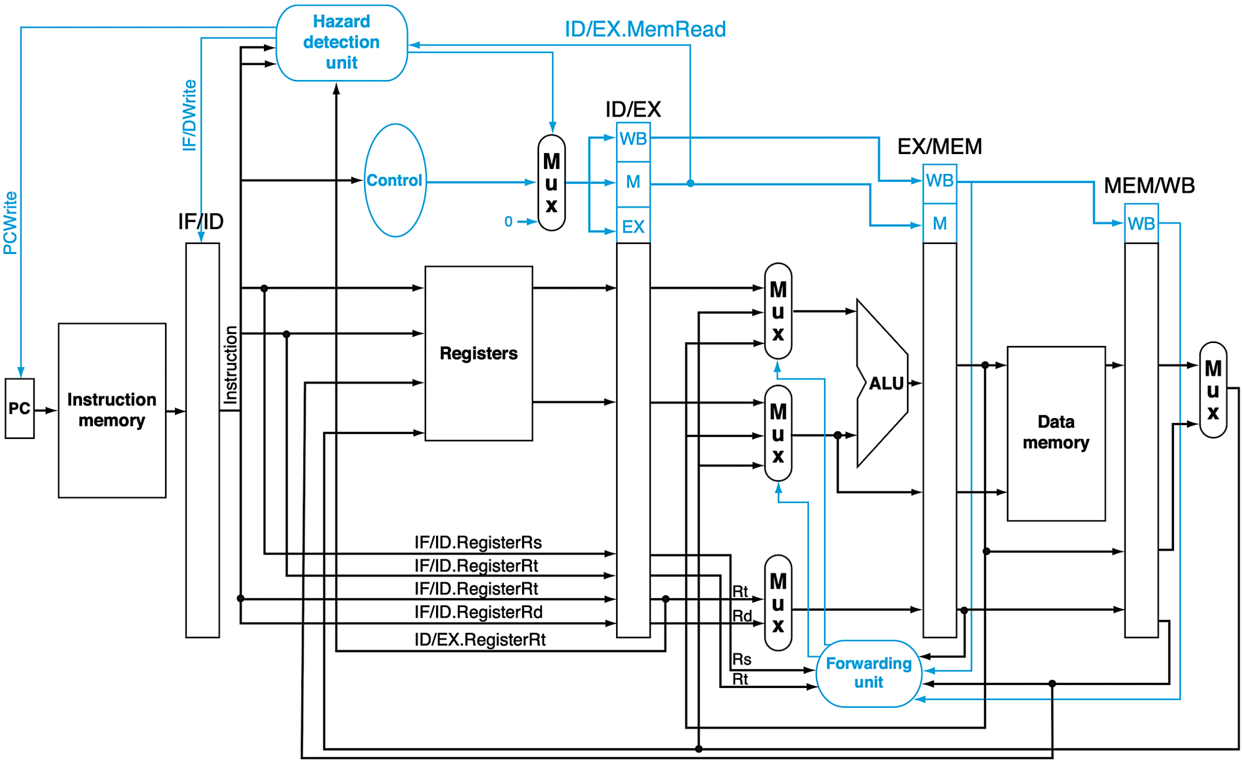
Pipeline Model Datapath and Control



Inst	EX Stage				MEM Stage			WB Stage	
	RegDst	ALUSrc	ALUop		Mem Read	Mem Write	Branch	MemToReg	Reg Write
			op1	op0					
R-type	1	0	1	0	0	0	0	0	1
lw	0	1	0	0	1	0	0	1	1
sw	X	1	0	0	0	1	0	X	0
beq	X	0	0	1	0	0	1	X	0



Hazard Detection Unit

- IF/ID.RegisterRs and IF/ID.RegisterRt, the source registers for the current instruction.
- ID/EX.MemRead and ID/EX.RegisterRt, to determine if the previous instruction is LW

  if so, which register it will write to
- PCWrite, IF/ID Write: determine whether the pipeline stalls or continues.
- A mux select for a new multiplexer, which forces control signals for the current EX and future MEM/WB stages to 0 in case of a stall.

Basically,

- MUX = 1 prevent inst. In ID come to EX, it will be substitute using a "nop"
- IF/ID Write = 0 prevent inst. In IF come to ID
- PC Write prevent the next inst. come to IF stage.

Forwarding Unit will generate signal MUX control to select the operand 1 and 2 for ALU.

Mux control	Source	Explanation
ForwardA = 00	ID/EX	The first ALU operand comes from the register file.
ForwardA = 10	EX/MEM	The first ALU operand is forwarded from the prior ALU result.
ForwardA = 01	MEM/WB	The first ALU operand is forwarded from data memory or an earlier ALU result.
ForwardB = 00	ID/EX	The second ALU operand comes from the register file.
ForwardB = 10	EX/MEM	The second ALU operand is forwarded from the prior ALU result.
ForwardB = 01	MEM/WB	The second ALU operand is forwarded from data memory or an earlier ALU result.

MISC (From PYP)

- Predict not taken is easier to implement, as PC+4 is already computed.
- To implement predict taken, we need to compute the target branch address quickly **in the first cycle**. That requires additional hardware.