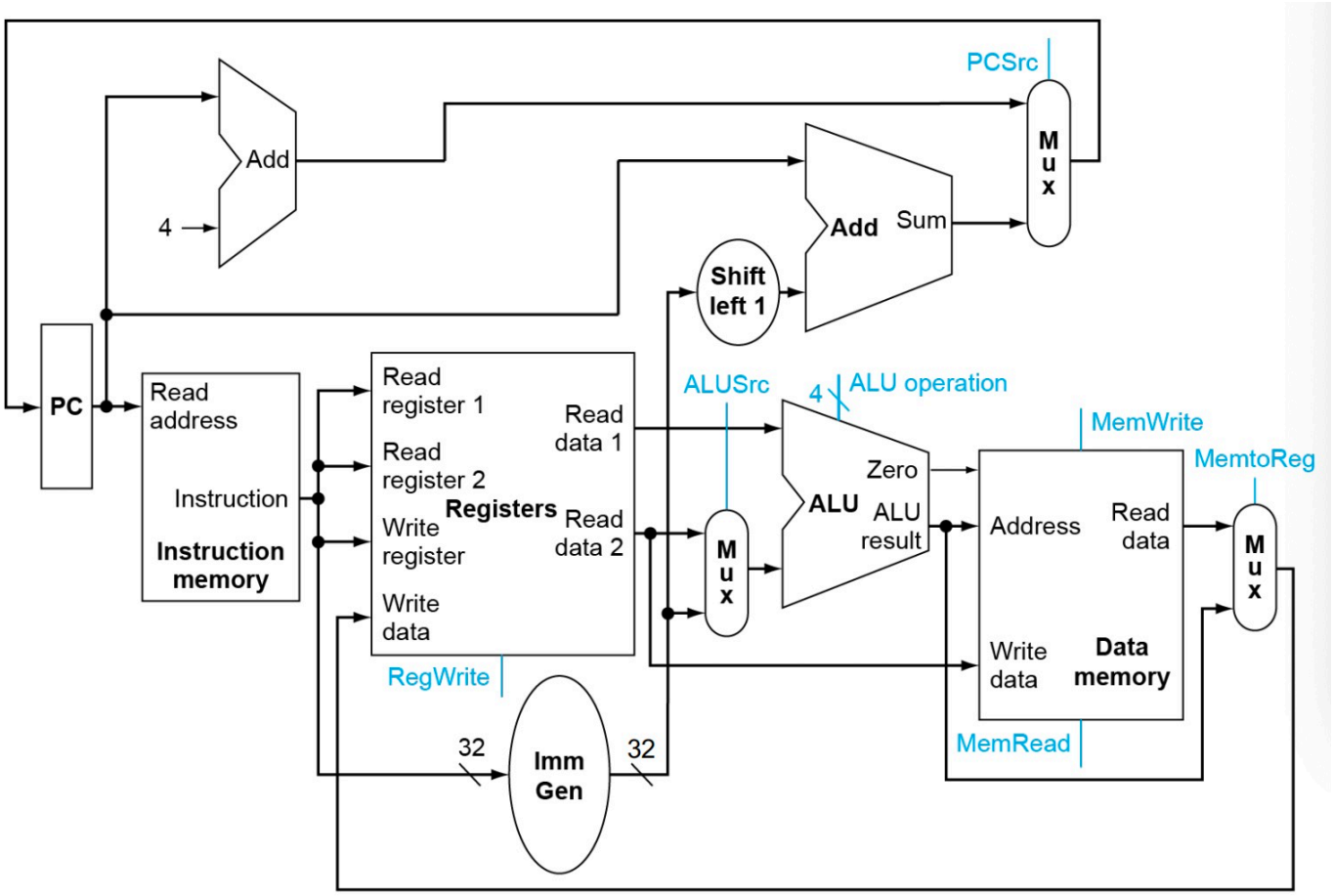
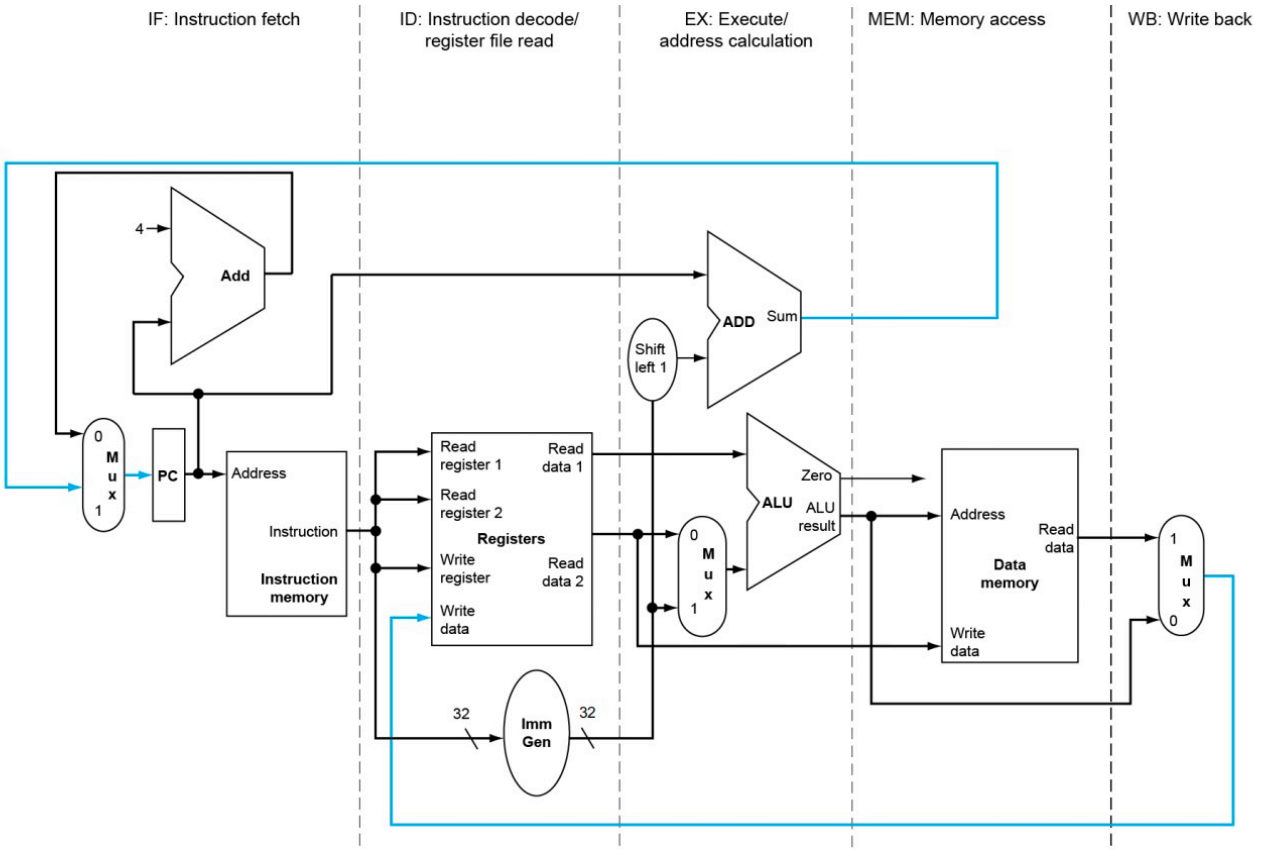
VE370 Mid Review

Pipelined Processor

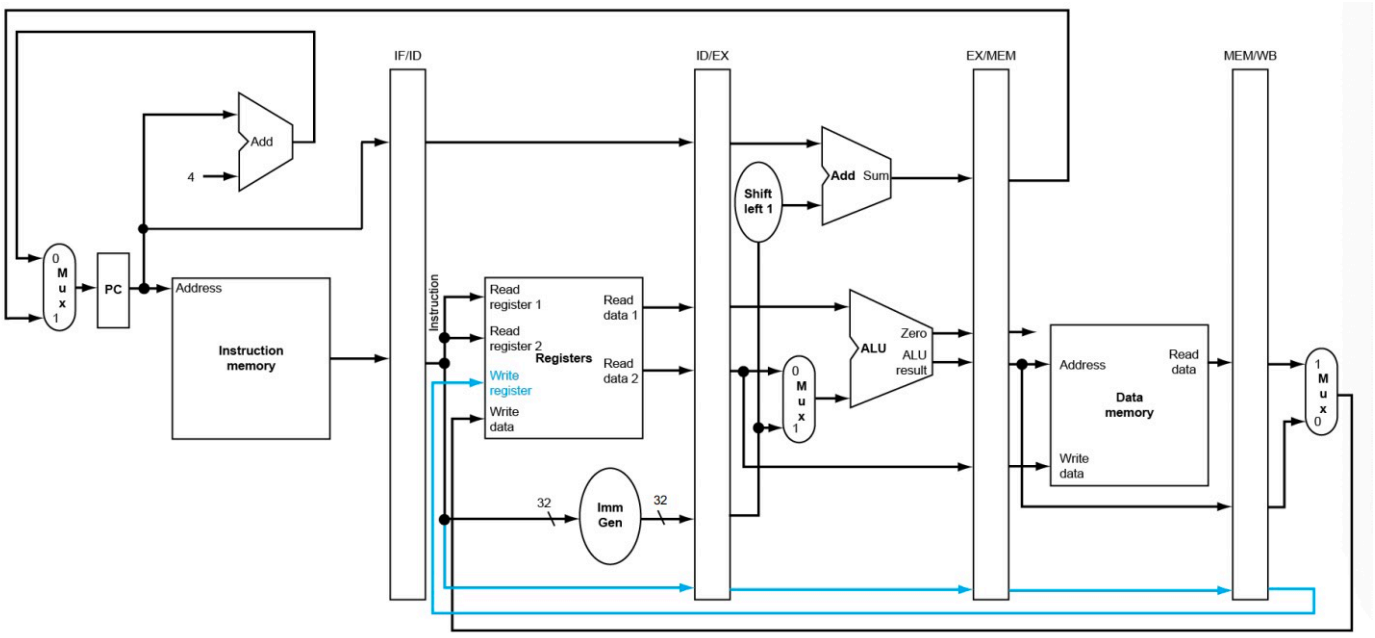


- Motivation: have a faster system
- Observation: Many components in idle while others are working
- Solution: separate into several stages

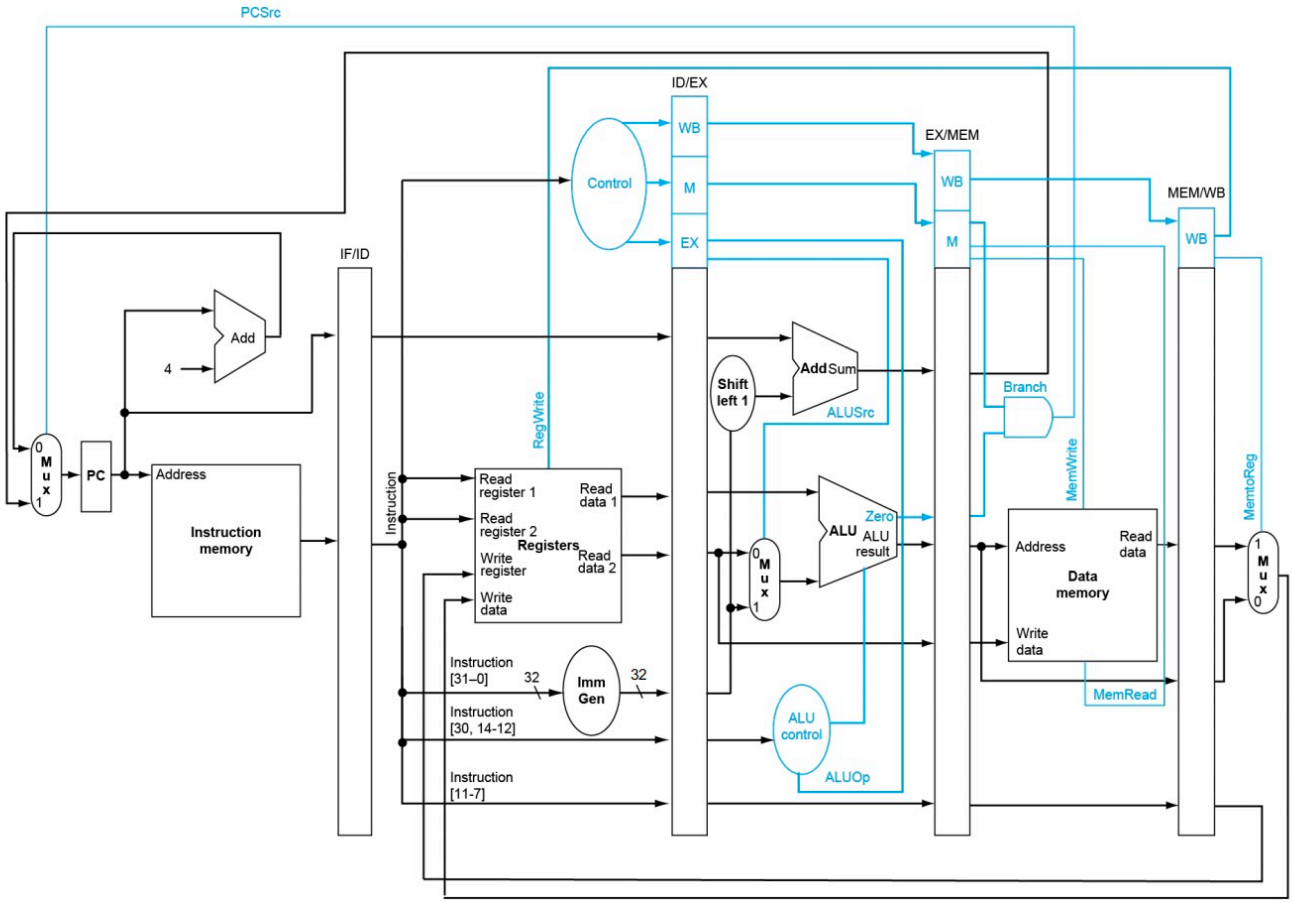


Two tips:

- Write register source of the register file should change.



- Control signals should pass through stages.



IF/ID	PC	ID/EX	PC	RegWrite	EX/MEM	PC+ immediate	ALU result	MEM/WB	RegWrite
	Instruction code		Branch	Instruction [30]		Branch	read data 2		MemtoReg
			MemRead	funct3		MemRead	write register		write
			MemtoReg	immediate		MemtoReg			write register
			ALUOp	write register		MemWrite			ALU result
			MemWrite	read data 1		RegWrite			read data from memory
			ALUSrc	read data 2		zero			

1. (15 points) Given this instruction:
   `sw x5, -4(x2)`

   As the instruction goes through the pipeline, what will be stored in the pipeline registers:
   IF: what's in PC
   ID: what's in IF/ID
   EX: what's in ID/EX?
   MEM: what's in EX/MEM
   WB: what's in MEM/WB?

Performance:

$$T_{tot} = IC \times CPT \times T_{CC}$$

2. (20 points) Assume that individual stages of the RISC-V pipelined datapath have the following latencies:

IF	ID	EX	MEM	WB
300 ps	150 ps	250 ps	300 ps	150 ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU/Logic	Jump/Branch	Load	Store
45%	15%	20%	20%

(1) What is the clock cycle time? (2 points)
(2) What is the execution time of a sw instruction in the pipelined processor? (3 points)
(3) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor? (5 points)
(4) Using the processor to run a program of 1,000 instructions, what is the total execution time? What is the average CPI? (10 points)