

Due Friday, October 18

The diagram illustrates the internal components and data flow of the MIPS processor:

- PC (Program Counter):** Provides the address for the Instruction Memory.
- Instruction Memory:** Receives the PC address and outputs the instruction. The instruction is split into fields: [31-0], [20-16], [9-5], and [4-0].
- Control Unit:** Receives the instruction fields and outputs control signals: *Reg2Loc*, *Uncondbranch*, *Branch*, *MemRead*, *MemtoReg*, *ALUOp*, *MemWrite*, *ALUSrc*, and *RegWrite*.
- Registers:** A set of 32 registers. The *Read register* field selects data from a register (Read data 1 or 2). The *Write register* field selects the register to be updated with *Write data*.
- ALU (Arithmetic Logic Unit):**
 - ALU control:** Receives the *ALUOp* signal and selects the ALU operation.
 - Sign-extend:** Takes the 32-bit *Instruction [31-0]* and extends the sign to 64 bits.
 - ALU Src:** Selects between the 64-bit sign-extended value and the *Read data 2* from the registers based on the *ALUSrc* control signal.
 - ALU result:** Performs the operation on the selected sources and outputs the 64-bit result.
 - Zero:** A flag that is set if the ALU result is zero.
- Branch and Jump Logic:**
 - Branch:** Uses *Read data 1* and *Read data 2* to calculate the branch target address.
 - Uncondbranch:** Uses *Read data 1* to calculate the branch target address.
 - Reg2Loc:** Uses *Read data 1* and *Read data 2* to calculate the branch target address.
- Multiplexers (Mux):**
 - Mux 0:** Selects between the PC and the branch target address to update the PC.
 - Mux 1:** Selects between the ALU result and the branch target address to update the PC.
- Memory:**
 - Read data:** Receives the address from the ALU result and outputs data from the memory.
 - Write data:** Receives the address from the ALU result and the *Write data* from the registers to store data in the memory.

[illegible]

(b) Using X's to indicate in the table below which blocks in the datapath diagram are used for each instruction

Instr.	Registers	Sign-Extend	Branch	ALU Control	Lower ALU	Upper ALU	Shift Left	Data Memory
add								
b								
cbz								
stur								

(c) Given the execution times for each of the blocks listed below, determine the total execution times for the instructions: add, b, cbz and stur. Note, this problem is challenging. You must determine the different components of the data path that each instruction uses, then find the time that signals require to complete the path, and then use the path that takes the longest to determine the total execution time for that instruction.

Instruction Memory	Add	Control	Mux	Registers	Sign-Extend	ALU	Data Memory	Shift Left 2	And Gate
280 ps	230 ps	70 ps	40 ps	170 ps	40 ps	240 ps	280 ps	12 ps	3 ps

2. (20 pts) The logic latencies for individual stages in a processor are listed in the following table.

IF	ID	EXE	MEM	WB
310 ps	340 ps	490 ps	325 ps	260 ps

(a) What is the minimum clock period for a pipelined and a non-pipelined processor using these parameters?

(b) What is total latency of an Arm ldur instruction in a pipelined processor? What is the throughput of a large series of ldur instructions with no stalls or hazards? Express your answer in millions of instructions per second (MIPs).

3. (30 pts) Do Zybooks 4.17.7: [20] <COD §4.4> in chapter 4.17, but with the following latencies:

Instruction Memory	Add	Mux	ALU	Registers	Data Memory	Sign Extend	Shift Left 2
250 ps	90 ps	45 ps	110 ps	130 ps	250 ps	26 ps	25 ps
