Computer Architecture Recitation 3

Berkay Sancı - M. Alpaslan Tavukçu

08.05.2025

Question 1

Execution of an instruction for an experimental CPU has the following phases with the given durations:

1. Instruction fetch and decode: 50 ns (memory access needed),
2. Operand fetch: 60 ns (memory access needed),
3. Execution: 50 ns (memory access **not** needed),
4. Result write: 60 ns (memory access needed),
5. Interrupt housekeeping: 200 ns (memory access needed).

Memory access time and I/O interface access times are both 50 ns. Two vectored interrupt sources (devices) A and B are connected over a priority interrupt controller to the CPU. Device B has higher priority than device A. For The interrupt service routine (including the RTI (Return from Interrupt) instruction) for device A runs 1 instruction, also the interrupt service routine (ISR) for device B runs 1 instruction. There is a 2-wire DMAC that is configured to transfer words from the I/O interface to the memory using the **cycle-stealing** technique. The DMAC type is **fly-by (implicit)**. Data does not pass through DMAC.

Question 1

Assume that we start a clock (Clock = 0) when the CPU begins to run the main program.

- At Clock = 30 ns, the device A sends an interrupt request.
- At Clock = 40 ns, the DMAC attempts to start the 2 words data transfer from I/O interface to memory.
- At Clock = 170 ns, the device B sends an interrupt request. .

a) When (Clock =?) will the interrupt service routine (ISR) of the IS start to run? Why?

b) When (Clock =?) will the DMAC complete the transfer of the first, second, third, and fourth words? Why?

Solution 1

Question 2

The instruction cycle of a CPU has the following 5 states (cycles) with the given durations:

1. Instruction Fetch (IF): **40** ns,
2. Operand (Register) Read (OR): **30** ns,
3. Execution (EX): **30** ns,
4. Operand Write (OW) : **30** ns,
5. Interrupt (IR) (if necessary): **200** ns.

Note that the CPU accesses memory in instruction fetch and operand write cycles but not in the operand read and execution cycles. The CPU enters the interrupt cycle only if an interrupt request is accepted. The housekeeping operations in the interrupt cycle take **200** ns.
The memory access time and I/O interface access time are **50** ns each.
In this system, there are four 3-wire (BR, BG, BGACK) DMACs ($DMAC_1$, $DMAC_2$, $DMAC_3$, and $DMAC_4$.) which are connected to the 68000-like processor over a bus arbiter.
The precedence order is $DMAC_1 > DMAC_2 > DMAC_3 > DMAC_4$.

Question 2

For all four DMACs, the DMAC type is **flow-through (explicit)** (i.e., data passes through the DMAC). $DMAC_1$ and $DMAC_4$ are in cycle-stealing mode, and $DMAC_2$ and $DMAC_3$ are in burst mode. Assume that we start a clock (Clock = 0) when the CPU begins to run a program and all DMACs attempt to start transfer for 5 words when Clock = **20** ns.

a) When (Clock = ?) will $DMAC_1$ complete the transfer of the third word? Why?

b) When (Clock = ?) will the CPU complete the first instruction cycle? Why?

c) When (Clock = ?) will $DMAC_4$ complete the transfer of all 5 words? Why?

Solution 2

Time (ns)	**CPU**	**DMAC$_1$**	**DMAC$_2$**	**DMAC$_3$**	**DMAC$_4$**
40	IF				
100 (CPU) 140 (DMAC)	OR + EX	1. word			
240 340 440 540 640			1. word 2. word 3. word 4. word 5. word		
740		2. word			
840 940 1040 1140 1240				1. word 2. word 3. word 4. word 5. word	
1340 **(answer of a)**		3. word			
1440					1.word
1540		4. word			

Solution 2

Time (ns)	**CPU**	**DMAC$_1$**	**DMAC$_2$**	**DMAC$_3$**	**DMAC$_4$**
1640					2. word
1740		5. word			
1840					3. word
1870 (answer of b)	OW				
1970					4. word
2010	IF				
2070 (CPU) **2110 (DMAC) (answer of c)**	OR + EX				5. Word

Question 3

Consider a 5-stage, MIPS-like RISC processor. The details of the stages are as follows:

- **IF:** Fetch the instruction from instruction memory, and calculate the next PC
- **ID:** Decode the instruction, and read source operands from register file
- **EX:** Execute the instruction, depending on instruction type
  - if arithmetic instruction, then perform arithm. op. and update flags;
  - if memory operation, then calculate address;
  - if branch, then evaluate condition and compute target address
- **MEM:** Access memory (if the instruction is a memory operation)
- **WR:** Write to the register file (result from the ALU or data from memory is written to registers)

Assume the following:

- The processor **has full forwarding (bypass)** connections, meaning that forwarding **EX** to **EX**, **MEM** to **EX** and **MEM to MEM**.
- The CPU writes data to registers in the **first** half of the clock cycle (rising edge) and reads data from registers in the **second** half of the cycle (falling edge).
- **Branch** target address calculation and decision operations are performed in the **EX stage**, and results are sent directly to the **IF stage** in the same cycle.

```
      SHL R6, R6, 2          ; (I1) R6 = R6 << 2
      LD R10, 4(R6)          ; (I2) R10 = Mem[R6+4]
      BEQ R10, #4, LBL       ; (I3) Branch to LBL if (R10 == 4)
      ….
LBL:
```

Assume that the code given above is run on this processor.

Question 3 (cont.)

- Show the dependencies and dependency types (if any exists) between the instructions.
- Draw the timing diagram for the given instruction stream. Show the stall cycles (if there are any) on the diagram.
- Where will the data operands that are processed during the EX and MEM stages of the store (LD) instruction come from (which pipeline register, register file, or immediate)? Briefly explain your answer.

Question 3 Solution

Question 4

For this question, consider a pipelined 6-stage processor where each stage completes in a single cycle. The details of the stages are as follows:

- **IF:** Fetch the instruction from instruction memory, and calculate the next PC
- **ID:** Decode the instruction
- **OF:** Read source operands from register file
- **EX:** Execute the instruction, depending on instruction type
  - if arithmetic instruction, then perform arithm. op. and update flags;
  - if memory operation, then calculate address;
  - if branch, then evaluate condition and compute target address
- **MEM:** Access memory (if the instruction is a memory operation)
- **WR:** Write to the register file (result from the ALU or data from memory is written to registers)

Assume the following:

- The processor **has full forwarding (bypass)** connections.
- The CPU writes data to registers in the **first** half of the clock cycle (rising edge) and reads data from registers in the **second** half of the cycle (falling edge).
- **Branch** target address calculation and decision operations are performed in the **EX stage**, and results are sent directly to the **IF stage**.

```
      MOV R1, #1        // (I1): R1 = 1
L1:
      BEQ R1, #8, DONE  // (I2): Branch to DONE if R1 == 8
      ADD R1, R1, #1    // (I3): R1 = R1 + 1
      MOV R2, #4        // (I4): R2 = 4
L2:
      BEQ R2, #2, L1    // (I5): Branch to L1 if R2 == 2
      SHL R1, R1, #2    // (I6): R1 = R1 << 1
      SHR R2, R2, #1    // (I7): R2 = R2 >> 1
      B L2              // (I8): Unconditional branch to L2
DONE:
      ST R1, 0(R4)      // (I9): Mem[R4+0] <- R1
      LD R2, 4(R6)      // (I10): R2 = Mem[R6+4]
      ADD R1,R1,R2      // (I11): R1 = R1 + R2
```

Question 4 (cont.)

- Assume that the given pipeline stalls only on conditional branches and does not stall for any other reasons. What would be the CPI of the program given above when it runs to the completion? Show all of your work, and clearly explain your answer. Assume that there is no prediction mechanism.

Question 5

Consider a 5-stage processor. (IF,DR,EX,ME,WB) Assume that for conditional branches, branch prediction is made in the IF and results (prediction and the target address if taken) are sent directly to the IF stage. Assume one bit dynamic prediction strategy is used, with initial decision is not to take the branch. Draw the space-time diagram for the given instruction stream.

```
        MOV   R1, $01
        ADD   R1, $01, R1
        ADD   R1, $06, R2
LOOP:   ADD   R2, $02, R2
        SUB   R1, $01, R1
        BNZ   LOOP
        SUB   R2, $03, R2
        SHR   R2, $01
```

Question 5 Solution