İTÜ Computer Engineering Department
April 22, 2021
Assoc.Prof. Feza BUZLUCA
Asst. Prof. Sanem KABADAYI

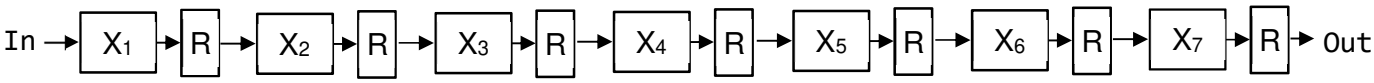
COMPUTER ARCHITECTURE MIDTERM EXAM QUESTION 1

Rules:

1. The normal end time of this question is **17:45**. However, because of 10 minutes buffer period the submission will be closed at **17:55**.
2. **Answer the question on its own sheet** and upload your files during the time allotted for that question, as explained in the file "Exam policies". Create your files in **PDF format**.
3. **You may not ask any questions during the exam.** State any assumptions you have to make.
4. Any cheating or any attempt to cheat will be subject to the University disciplinary proceedings.
5. Please **show ALL work**. Answers with no supporting explanations or work will be given no partial credit. If we cannot read or follow your solution, no partial credit will be given. PLEASE BE NEAT!

QUESTION 1: (30 Points)

- A task **T** that will execute on integers consists of seven suboperations X_i (i=1,2,…,7).
- To perform this task, we implement a pipeline P using digital combinational units (e.g., adders, multiplexers, logic gates) with the following propagation delays:
  X_1= (not given), X_2=20ns, X_3=20ns, X_4=15ns, X_5=10ns, X_6=10ns, X_7=10ns.
  The delay of each register (R) is 5 ns.
- The block diagram of the pipeline P is shown below.
- The suboperations must be performed in the given order.



- As the number of integers increases significantly ($n\to\infty$), the speedup achieved by this pipeline is 3 ($S_{n\to\infty} = 3$).
- The time required for a task <u>without pipelining</u> is the sum of delays of all digital combinational units ($t_n = \sum_{i=1}^{7} X_i$).

a) Calculate the delay of the first digital combinational unit (X_1 = ?). Show your work. (10 p)

b) The given pipeline P is actually not properly designed.

i. What are the drawbacks of the pipeline P? (10 p)

ii. Design a new pipeline P_{new} that avoids all disadvantages of pipeline P and yet maintains the speedup ($S_{n\to\infty} = 3$). Draw the block diagram of pipeline P_{new}. (10 p)

İTÜ Computer Engineering Department
April 22, 2021
Assoc.Prof. Feza BUZLUCA
Asst. Prof. Sanem KABADAYI

COMPUTER ARCHITECTURE MIDTERM EXAM QUESTION 2

Rules:

1. The normal end time of this question is **18:35**. However, because of 10 minutes extension the submission will be closed at **18:45**.
2. **Answer the question on its own sheet** and upload your files during the time allotted for that question, as explained in the file "Exam policies". Create your files in **PDF format**.
3. **You may not ask any questions during the exam.** State any assumptions you have to make.
4. Any cheating or any attempt to cheat will be subject to the University disciplinary proceedings.
5. Please **show ALL work**. Answers with no supporting explanations or work will be given no partial credit. If we cannot read or follow your solution, no partial credit will be given. PLEASE BE NEAT!

QUESTION 2: (35 Points)

Consider the exemplary RISC processor given in Section 2.4.2 of the lecture notes. **Differing** from lecture notes, suppose that the instruction pipeline is designed with four stages as explained below:

1. **Instruction Fetch and Decode (ID):** Instruction fetch and instruction decode.
2. **Read registers (RR):** Read registers (operands).
3. **Execute (EX):** Perform ALU operation, compute jump/branch targets, make branch decsions.
4. **Memory, Write back (MW):** Two stages in the original processor have been combined; the ME/WB register has been removed.

Assume the following:

- The register file access hazard is **NOT** fixed.
- The processor does **NOT** have any **forwarding (bypass)** connections.
- The internal structure of the execution stage is similar to the circuitry shown on slide 2.53 (latest version, 2021), i.e., **branch target address calculation and decision operations** are performed in the **EX** stage, and results are sent directly to the **ID** stage.

a) Draw the timing diagram for the piece of code given below to show the data and branch hazards, and software-based solutions that involve inserting NOOP instructions. (20 p)

```
                ……
1:              SUB  R1, R2, R1        ; R1 <- R1 + R2
2:              LDL  0(R1), R4         ; R4 <- M[R1]
3:              LDL  4(R1), R5         ; R5 <- M[R1+4]
4:              ADD  R6, R4, R6        ; R6 <- R6+R4
5:              ADD  R7, R5, R7        ; R7 <- R7+R5
6:              SUB  R6, R7, R7        ; R7 <- R7 – R6
7:              BNE  LABEL             ; Branch if not equal
8:              ADD  R6, #2, R8        ; R8 <- R6+2
9:              STL  8(R1), R8         ; M[R1+8]<-R8
                  :                    ; other instructions
                  :
10:  LABEL:  …..                       ; program continues
```

b) Assume that there is operand forwarding (bypassing) from the output of the EX stage to the inputs of the ALU. Other parts of the CPU are not modified. Consider only the lines between 1 and 6 (including 6) of the piece of code given in (a.i.). Draw the timing diagram for the piece of code (lines 1-6) to show hazards, and software-based solutions that involve inserting NOOP instructions. (15 p)

İTÜ Computer Engineering Department
April 22, 2021
Assoc.Prof. Feza BUZLUCA
Asst. Prof. Sanem KABADAYI

COMPUTER ARCHITECTURE MIDTERM EXAM QUESTION 2

Rules:

1. The normal end time of this question is **19:25**. However, because of 5 minutes extension the submission will be closed at **19:30**.
2. **Answer the question on its own sheet** and upload your files during the time allotted for that question, as explained in the file "Exam policies". Create your files in **PDF format**.
3. **You may not ask any questions during the exam.** State any assumptions you have to make.
4. Any cheating or any attempt to cheat will be subject to the University disciplinary proceedings.
5. Please **show ALL work**. Answers with no supporting explanations or work will be given no partial credit. If we cannot read or follow your solution, no partial credit will be given. PLEASE BE NEAT!

QUESTION 3: (35 Points)
Note that Parts (a) and (b) below are not related.

a) A CPU has an Interrupt Request input (**IRQ**) and an Interrupt Acknowledgement output (**INTA**). All signals are active at "1". If the CPU receives a request from the inputs (IRQ), it works with vectored interrupts and reads the interrupt vector number after the acknowledgement (INTA=1).
In this system, there are devices that work as interrupt sources. Each of these devices has an Interrupt Request output (**IRQ**), an Interrupt Acknowledgement input (**INTA**), and vector number output (**VN**). After the interrupt has been acknowledged (**INTA=1**) the device outputs its vector number at the **VN** and removes its request (**IRQ=0**).
We designed a digital combinational circuit DC with four inputs (I_1, I_2, I_3, I_4) and four outputs (O_1, O_2, O_3, O_4) to use it as priority interrupt controller. The logic expressions for the outputs are given below.

$$O_1 = I_2 + I_3 + I_4$$
$$O_2 = I_1 \cdot I_2$$
$$O_3 = I_1 \cdot I_3$$
$$O_4 = I_1 \cdot I_4$$

i. Since the circuit DC is not properly designed, it cannot be used as a priority interrupt controller. Explain the reason shortly. (10 p)

ii. Modify the expressions of the outputs, where it is necessary. Write which input/output of the DC will be connected to which input/output of the CPU or interrupt sources. (10 p)

b) The instruction cycle of a CPU has the following 4 states (cycles) with the given durations:
1. Instruction fetch 60 ns, 2. Decode and Operand fetch: 50 ns, 3. Execution and Operand write: 70 ns, 4. Interrupt (housekeeping): 200 ns.

- In this system, we have two interrupt sources (IS1, IS2). The duration of the interrupt service routine (ISR) including the RTI instruction for each source is 1000ns.
- The interrupts are disabled in the housekeeping operations and enabled again in the RTI instruction. Therefore, an ISR cannot be interrupted.
- The interrupt sources are connected to the CPU using Serial Priority Interrupt Hardware (Daisy Chain). The IS1 has higher priority than IS2.

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

- At Clock = 50ns, the device IS2 issues an interrupt request.
- At Clock = 120ns, the device IS1 issues an interrupt request.

i. Which ISR (for IS1 or IS2) will run **firstly** and when (Clock =?) will it start to run. Show your calculations (7 p)

ii. Which ISR (for IS1 or IS2) will run **secondly** and when (Clock =?) will it start to run. Show your calculations (8 p)