İTÜ Computer Engineering Department
01.07.2020
Assoc.Prof. Feza BUZLUCA
Asst. Prof. Sanem KABADAYI

COMPUTER ARCHITECTURE FINAL EXAMINATION

Regulations:

1. The exam duration is **3 hours**. The exam itself should take **a maximum of 3 hours** to complete. You may use additional time to scan your papers properly and submit your file.
2. In case of unexpected technical problems, such as power outages, Internet unavailability, software failures, you may upload your files to Ninova until 23:00.
3. Read the file "exam policies" in the "class-files" directory in Ninova carefully, and submit your solutions to Ninova as explained in this file. **You must submit 5 zip files.**
4. Do not send your solutions by e-mail. We will only accept files that have been uploaded to the official Ninova e-learning system before the deadline.
5. There are **5 questions**; you must **answer each question on its own sheet** as explained in the file "exam policies". Create a separate zip file for each question.
6. **You may not ask any questions during the exam.** If you think something is missing in a question, explain it, make the necessary assumption, and solve the question.
7. Any cheating or any attempt to cheat will be subject to University disciplinary proceedings.

QUESTION 1: (15 Points)

We designed a pipeline P to perform task **T** on the elements of an array.
The total time required to complete the 5th task using pipeline P is $\mathbf{T}_5$ = 180 ns.
The total time required to complete the 10th task using pipeline P is $\mathbf{T}_{10}$ = 280 ns.

a) The time required for task **T** without pipelining is $\mathbf{t}_n$ = 80ns.
Calculate the speedup achieved by pipeline P if the number of tasks increases significantly ($n\rightarrow\infty$) because of a large array ($S_{n\rightarrow\infty}$ = ?). (5p)

b) We design a new pipeline $\mathbf{P_{new}}$ for the same task **T**.
The total time required to complete the 5th task using the new pipeline, P_{new}, is $\mathbf{T}_5$ = 150 ns. Other times (e.g., $\mathbf{T}_{10}$) are not given.
Can we now definitely conclude (claim) that the speedup ($S_{new\infty}$) achieved by the new pipeline P_{new} as the number of tasks increases significantly ($n\rightarrow\infty$) is higher than the speedup (S_∞) of the original pipeline P ($S_{new\infty} > S_\infty$?). Explain briefly. (10p)

QUESTION 2: (25 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 three stages as explained below:

1. **Instruction Fetch (IF):** Same as in lecture notes
2. **Instruction Decode, Read registers, Execute (DRE):** Two stages in the original processor have been combined; the DR/EX register has been removed.
3. **Memory, Write back (MWB):** Two stages in the original processor have been combined; the ME/WB register has been removed.

The register file access hazard is not fixed, and the processor does not contain any forwarding (bypass) connections.
The internal structure of the execution circuitry is as shown on slide 2.30, i.e., the branch penalty is not reduced.

a) Draw the timing diagram for a piece of code with two instructions to show the data dependency problem in the given pipeline with three stages. Explain the problem briefly. (5 p)

b) Draw the timing diagram for the same code that shows the hardware-based solution with stalling to the data dependency problem. What is the penalty for this solution? (10 p)

c) Draw the timing diagram for a piece of code to show the conditional branch hazard in the given pipeline with three stages. Explain the problem briefly. (10 p)

QUESTION 3: (30 Points)
A single-CPU computer system has

- **1Mi bytes** of main memory and
- cache memory that can store **8 Ki bytes** of data.

Prefixes Mi and Ki represent Megabinary and Kilobinary respectively, as explained in Chapter 1 of lecture notes.

Data transfer between main memory and cache is in blocks of **8 bytes**.
The cache control unit uses **2-way** *set associative mapping*.
For write operations, **Flagged Write Back** (FWB) with **Write Allocate** (WA) is used.
When necessary, **LRU** is used as the replacement algorithm. If a set is empty, assume that frame_0 is older than frame _1.
Cache memory is used only for data, not for instructions.

```
LOOP1:
For i = 0 to 4
   B[i] = B[i]+A[i];
End of For
LOOP2:
For i = 0 to 4
   C[i] = A[i];
End of For
```

The CPU runs the piece of pseudocode given on the right.

- In the first loop, five elements of array B are added to the elements of array A, and the results are copied into array B. Details of this operation are: read B[i], read A[i], add the two data, write the result to B[i].
- In the second loop, five elements of array A are copied into array C.
- Each element is one byte.
- The starting addresses of the arrays are given below:
  A: $00008
  B: $0100E
  C: $0200A

Initially, cache memory is empty, and none of the frames are dirty.

a) Consider LOOP1. (15 p)

i) Which set/frames of cache memory does the cache control unit place arrays A and B into? Give your answers in decimal. (Example: "**set number: 73, frame: 0**" or "**set number 85, frame: 1**")

ii) How many read misses, read hits, write misses, write hits, write-to-main-memory operations, and block transfers occur during the run of the given loop?

b) Consider LOOP2. (15 p)

i) Which set/frames of cache memory does the cache control unit place the array C? Explain your answer briefly. Give your answers in decimal. (Example: "**set number: 73, frame: 0**" or "**set number 85, frame: 1**").

ii) How many read misses, read hits, write misses, write hits, write-to-main-memory operations, and block transfers occur during the run of the given loop?

QUESTION 4: (15 Points)

For single-error correction in memory, we construct a 4:7 Hamming code as given below:

$$p_0 = d_0 \oplus d_1 \oplus d_3$$
$$p_1 = d_0 \oplus d_2 \oplus d_3$$
$$p_2 = d_1 \oplus d_2 \oplus d_3$$

d_i represents data bits, p_i represents parity bits, and $\oplus$ is the logical exclusive-OR (XOR) function.

a) Draw the **syndrome impact table** for the given code, and show that it is not properly designed because it cannot correct single-bit errors. Explain your answer briefly. (10 p)

b) Change only the given equation for p_2 to construct a real Hamming code that can correct single-bit errors. (5 p)

QUESTION 5: (15 Points)
In a symmetric multiprocessor (SMP) system with a shared bus, there are three CPUs (CPU1, CPU2, and CPU3) that have local cache memories. The system does not have a shared L2 cache.
To provide cache coherence, the snoopy **MESI** protocol with **Write Back** (WB) is used.

a) The MESI state of a particular frame in cache memory of CPU1 is given below. What are the MESI states of the corresponding frames in cache memories of CPU2 and CPU3, and is the data in main memory valid or not for the given cases below? Answer for (i), (ii), and (iii) separately. If there is more than one possibility, give all of them. (5 p)

	CPU1:	CPU2:	CPU3:	Main Memory:
i)	Shared			
ii)	Exclusive			
iii)	Modified			

b) Assume that a particular frame in cache memory of CPU1 is in state Exclusive. In this case, CPU2 attempts to write to the corresponding frame in its cache memory. Which control messages are sent by the MESI cache controllers during this operation? Write the messages in the order they are sent. What are the states of the corresponding frames in cache memories of the CPUs (CPU1, CPU2, and CPU3), and is the data in main memory valid or not after the write operation? (10 p)