Homework 5

Graded

Student

Vidit Dharmendra Pokharna

Total Points

100 / 100 pts

Question 1

Pipeline Buffers **15** / 15 pts

1.1 **DBUF Contents** **5** / 5 pts

✔ **– 0 pts** Correct

– 2 pts Does not include Opcode

– 1 pt Does not include Contents in $Rx

– 1 pt Does not include Contents in $Ry

– 1 pt Does not include Immediate Value

– 1 pt Includes one extra answer choice

– 2 pts Includes two extra answer choices

– 3 pts Includes 3 extra answer choices

– 5 pts Incorrect

1.2 **EBUF Contents** **5** / 5 pts

✔ **– 0 pts** Correct

– 1 pt Does not include Contents in $Rx

– 2 pts Does not include Opcode

– 2 pts Does not include Result of ALU operation

– 1 pt Includes one extra component

– 2 pts Includes two extra components

– 3 pts Includes three extra components

– 5 pts Incorrect

1.3 **MBUF Contents** **5** / 5 pts

✔ **– 0 pts** Correct

– 5 pts Does not include Opcode

– 1 pt Includes one extra component

– 2 pts Includes two extra components

– 3 pts Includes three extra components

– 5 pts Incorrect

Question 2

Pipeline Branching **20** / 20 pts

2.1 **Conservative Branching** **5** / 5 pts

– **0 pts** Correct (ADDI)

✔ – **0 pts** Correct

– **5 pts** Incorrect

2.2 **Branch Prediction** **5** / 5 pts

✔ – **0 pts** Correct (3)

– **0 pts** Correct

– **5 pts** Incorrect

2.3 **Branch Mis-prediction** **10** / 10 pts

✔ – **0 pts** Correct (BTB, Any heuristic that increases prediction chances)

– **2 pts** Insufficient description for any mechanism

– **5 pts** Only mentions 1 correct mechanism to reduce branch mis-prediction (branching heuristic or BTB)

– **10 pts** Does not mention 2 mechanism to reduce branch mis-prediction

– **10 pts** Blank/no answer

Question 3

Hazards **15** / 15 pts

3.1 **Number of Bubbles** **5** / 5 pts

– **0 pts** Correct (1)

✔ – **0 pts** Correct

– **5 pts** Incorrect

3.2 **Type of Bubbles** **5** / 5 pts

– **0 pts** Correct (RAW Hazard)

✔ – **0 pts** Correct

– **5 pts** Incorrect

3.3 **Deeper Pipeline Bubbles** **5** / 5 pts

+ **5 pts** Correct (two or more bubbles)

✔ + **5 pts** Correct

+ **0 pts** Incorrect

Question 4

Processor Scheduling Concepts **4** / 4 pts

4.1 **Round Robin** **1** / 1 pt

✔ **+ 1 pt** Correct

+ 0 pts Incorrect

4.2 **I/O Queue** **1** / 1 pt

✔ **+ 1 pt** Correct

+ 0 pts Incorrect

4.3 **FCFS** **1** / 1 pt

✔ **+ 1 pt** Correct

+ 0 pts Incorrect

4.4 **Priority Scheduler** **1** / 1 pt

✔ **+ 1 pt** Correct

+ 0 pts Incorrect

Question 5

Shortest Remaining Time First (SRTF) **30** / 30 pts

✔ **– 0 pts** Correct

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CPU Burst	P1	P1	P0	P0	P2	P1	P1	P2	P2	P0	P0	P2	P2			P2	P2	P2	P2	P2
I/O Burst			P1	P1	P1	P0	P0	P0	P0					P2	P2					

– 5 pts Scheduled the wrong process once (P2 at t = 3 instead of P0, etc.)

– 10 pts Scheduled the wrong process twice

– 15 pts Performs SJF (i.e. Does not preempt a running process)

– 15 pts Preempted IO Queue

– 30 pts Blank/no answer/Incorrect

Question 6

Waiting and Turnaround Times **16** / 16 pts

6.1 **Turnaround Time** **8** / 8 pts

✔ **– 0 pts** Correct (12.67)

– 0 pts Correct

– 0 pts Correct (from wrong schedule in Q5)

– 2 pts Incorrect P0 turnaround time (Expected: 11)

– 2 pts Incorrect P1 turnaround time (Expected: 7)

– 2 pts Incorrect P2 turnaround time (Expected: 20)

– 2 pts Error in averaging / result off by 1 (12.67 +- 1)

– 8 pts Wrong formula / calculated something other than turnaround time

– 8 pts Incorrect

6.2 **Waiting Time** **8** / 8 pts

✔ **– 0 pts** Correct (3.67)

– 0 pts Correct

– 0 pts Correct (from wrong schedule in Q5)

– 2 pts Incorrect P0 waiting time (3)

– 2 pts Incorrect P1 waiting time (0)

– 2 pts Incorrect P2 waiting time (8)

– 2 pts Error in averaging / result off by 1 (3.67 +- 1)

– 8 pts Wrong formula / calculated something other than waiting time

– 8 pts Incorrect

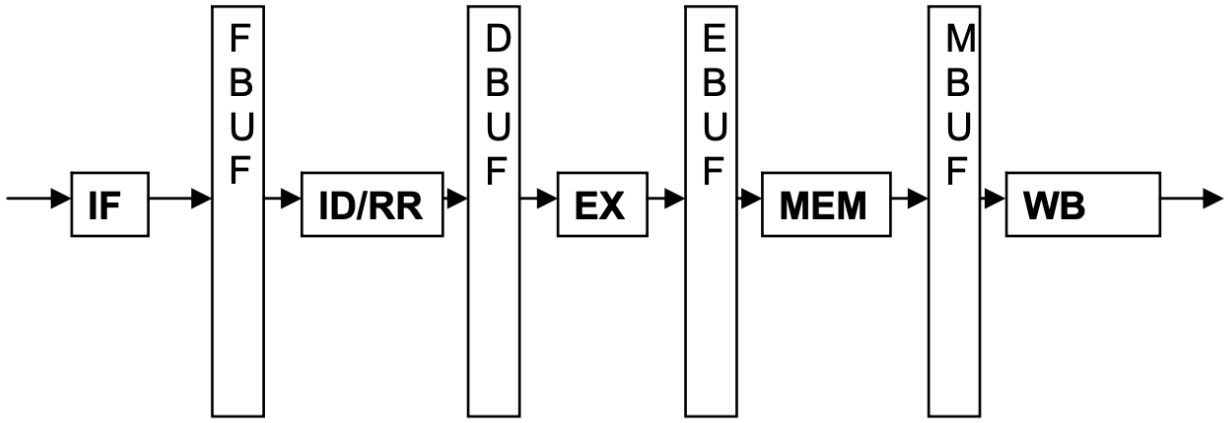
6.3 **Work** **0** / 0 pts

✔ **+ 0 pts** Correct

Q1 Pipeline Buffers

15 Points

Suppose we want to implement the LC-2222 Instruction `SW SR, BaseR, offset20`. According to the pipelined processor architecture described in the textbook, what are the **minimum** contents that must be placed in each pipeline buffer to support this instruction? Select the corresponding contents that are stored in each buffer (buffers pictured below). Assume we don't need to handle program discontinuities here.



Q1.1 DBUF Contents
5 Points

DBUF Contents (Select all that apply):

- [ ] PC
- [x] Contents in `$Rx`
- [x] Contents in `$Ry`
- [ ] Contents in `$Rz`
- [ ] Register number of `$Rx`
- [ ] Register number of `$Ry`
- [ ] Register number of `$Rz`
- [x] Opcode
- [ ] Result of ALU operation
- [x] Immediate Value
- [ ] Value read from memory
- [ ] None of the above

Q1.2 EBUF Contents
5 Points

EBUF Contents (Select all that apply):

- [ ] PC
- [x] Contents in `$Rx`
- [ ] Contents in `$Ry`
- [ ] Contents in `$Rz`
- [ ] Register number of `$Rx`
- [ ] Register number of `$Ry`
- [ ] Register number of `$Rz`
- [x] Opcode
- [x] Result of ALU operation
- [ ] Immediate Value
- [ ] Value read from memory
- [ ] None of the above

Q1.3 MBUF Contents
5 Points

MBUF Contents:

- [ ] PC
- [ ] Contents in `$Rx`
- [ ] Contents in `$Ry`
- [ ] Contents in `$Rz`
- [ ] Register number of `$Rx`
- [ ] Register number of `$Ry`
- [ ] Register number of `$Rz`
- [x] Opcode
- [ ] Result of ALU operation
- [ ] Immediate Value
- [ ] Value read from memory
- [ ] None of the above

Q2 Pipeline Branching

20 Points

Suppose we have a pipeline running the LC-2222 ISA and a set of instructions to run. We want to find the costs of different approaches to branching.

Branch instructions can disturb the sequential execution of a program, leading to problems in the pipeline.

Consider the following instructions for 4.1 - 4.3

```
START: ADDI
    LEA
    ...
    BGT START
    BEQ START
    LW
    NAND
    SW
```

The chart below shows the progression of the instructions through the pipeline with register forwarding.

Cycle Number	IF	ID/RR	EX	MEM	WB
1	**BGT**	-	-	-	-
2	**?**	**BGT**	-	-	-
3	**?**	**?**	**BGT**	-	-
4	**?**	**?**	**?**	**BGT**	-
5	**?**	**?**	**?**	**?**	**BGT**
6	**?**	**?**	**?**	**?**	**?**
7	**?**	**?**	**?**	**?**	**?**

Q2.1 Conservative Branching
5 Points

Suppose the pipeline is implemented using the **conservative approach** and that **BGT is not taken** but **BEQ is taken**. What instruction is in the pipeline in the **ID/RR stage** in **cycle 7**?

- ○ BEQ
- ○ SW
- ◉ ADDI
- ○ LEA

Q2.2 Branch Prediction
5 Points

Now suppose that the pipeline is implemented using **branch prediction**. It predicts that **BGT will be taken** and **BEQ will not be taken**. If **BGT is not taken** and **BEQ is taken**, how many NOPs will be in the pipeline in **cycle 7**?

3

Q2.3 Branch Mis-prediction
10 Points

In the previous problem, we saw how branch mis-prediction can cost us NOPs in the pipeline. Describe **two** mechanisms to **reduce branch mis-prediction** in the pipeline.

Two strategies that can help reduce branch misprediction in the pipeline are Branch Target Buffers (BTB) and Branch History Tables (BHT).

The BTB is a mechanism that maintains a record of the target address associated with a previously encountered branch instruction. This is advantageous for unconditional branches, like jumps, where the outcome is certain and the branch instruction always leads to a specific address.

On the other hand, the BHT is designed to store information about the historical behavior of branch instructions. It helps predict a branch's outcome by keeping track of whether the branch was typically taken or not in past occurrences. This historical context aids in making more accurate predictions, especially for conditional branches where the outcome may vary based on previous executions.

Q3 Hazards

15 Points

Dealing with hazards in a pipeline leads to bubbles depending on the implementation.

Q3.1 Number of Bubbles

5 Points

Consider the following LC2200 code.

```
one: .fill 1
LW $t0, 0($zero)
ADDI $t0, $t0, 1
ADDI $t1, $t0, 3
BEQ $t1, $t0, halt
ADDI $t1, $t1, 1
ADD $t0, $t0, $t1
halt: HALT
```

Suppose the 5-stage pipelined version of the LC 2200 from the textbook implements **data forwarding**, and it handles control hazards using **branch prediction**. Assume there's a table entry for line 3 (BEQ instruction) in the Branch Target Buffer (BTB) which states 'branch not taken.'
Assume that the code block above starts at memory address 0x0000, where the label `one` is placed.
What is the total number of bubbles generated in the pipeline after executing this code?

- ○ 0
- ◉ 1
- ○ 2
- ○ 3

Q3.2 Type of Bubbles
5 Points

Consider the following LC2200 code.

```
ADDI $t0, $zero, 1
ADDI $t1, $t1, 1
ADD $s1, $s0, $s0
ADDI $s2, $t0, 1
BEQ $zero, $t0, label
ADD $t2, $s2, $s1
...
label: ADD $t0, $zero, 1
...
```

Suppose the pipeline **doesn't implement** data forwarding, and it handles control hazards using the **conservative approach**. The **first bubble** generated will be a result of which of the following hazards?

- (●) RAW (Data) Hazard
- ( ) WAR (Data) Hazard
- ( ) WAW (Data) Hazard
- ( ) Control Hazard
- ( ) There is no hazard.

Q3.3 Deeper Pipeline Bubbles
5 Points

Consider a pipeline with nine stages instead of five like our LC-2200. We know we must insert two bubbles into the LC-2200 pipeline when branch mis-prediction occurs. How many bubbles do you think we would need to inject into our nine-stage pipeline in the same situation?

- ( ) fewer than two bubbles
- ( ) exactly two bubbles
- (●) two or more bubbles
- ( ) there is no way to tell

Q4 Processor Scheduling Concepts

4 Points

Below are some true/false conceptual questions on processor scheduling.

Q4.1 Round Robin

1 Point

Round Robin is a preemptive algorithm:

- (●) True
- ( ) False

Q4.2 I/O Queue

1 Point

The I/O queue is a priority queue that prioritizes on shorter jobs:

- ( ) True
- (●) False

Q4.3 FCFS

1 Point

One may experience a higher variance in response time for FCFS:

- (●) True
- ( ) False

Q4.4 Priority Scheduler
1 Point

Priority Scheduler prevents long-running computations from blocking shorter jobs:

- ○ True
- ◉ False

Q5 Shortest Remaining Time First (SRTF)
30 Points

Assume the scheduler uses the Shortest Remaining Time First (SRTF) algorithm. Consider the following three processes which each require **one** CPU burst followed by **one** I/O burst, followed by **one** final CPU burst. The scheduling starts at time T0.
Note: if there is a **tie** between the shortest times, please pick the processes with **the highest process ID**.

Note: When calculating time remaining for a process, use the sum of all remaining CPU time for that process.

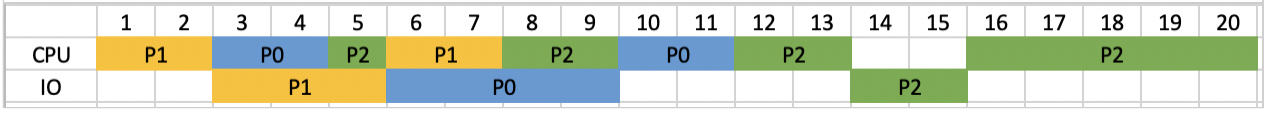
Process ID	CPU burst #1 and #2	I/O burst
P0	2	4
P1	2	3
P2	5	2

Create a process diagram showing the current process on the CPU and the current process performing I/O at each unit of time. **Download the template from Canvas, "Homework 5" assignment, under Modules** and fill out the process diagram. Attach **a screenshot (PNG/JPG)** of the complete template to the file attachment tab below.

For an example of what your table should look like, see "Homework 5".

Answer (screenshot of table)

▾ **Screen Shot 2023-10-11 at 9.55.38 PM.png** Download



Q6 Waiting and Turnaround Times
16 Points

The next question is regarding the Waiting Time and Turnaround Time for the Shortest Remaining Time First (SRTF) algorithm shown in Question 5.

Q6.1 Turnaround Time
8 Points

What is the average turnaround time for the system in Question 5 (Round to 2 decimal places if needed)?

12.67

Q6.2 Waiting Time
8 Points

What is the average waiting time for the system in Question 5 (Round to 2 decimal places if needed)?

3.67

Q6.3 Work
0 Points

Show your work for Q6.1 and Q6.2 here.

6.1:
P0 = 11, P1 = 7, P1 = 20
(11+7+20)/3 = 38/3 = 12.67

6.2:
P0 = 2+1 = 3, P1 = 0, P2 = 4+2+2=8
(3+0+8)/3 = 11/3 = 3.67