CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Problem	Points	Lost	Gained	Running Total	TA
1	1				
2	20				
3	9				
4	10				
5	10				
6	10				
7	10				
8	10				
9	10				
10	10				
Total	100				

- **You may ask for clarification but you are ultimately responsible for the answer you write on the paper.**
- **Illegible answers are wrong answers.**
- **Please do not discuss this test by any means (until 5 pm today)**
- **Please look through the entire test before starting. WE MEAN IT!!!**

Illegible answers are wrong answers.
Good luck!

1. (1 point, 0 min) (circle one)
Your favorite spring break destination

a) Orlando b) Daytona beach c) Cornfields of Illinois
d) Seoul e) Bangalore f) CCB 16 !!
g) Write in your own _____

CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Pipelining

2. (20 points, 10 min)

(a) (4 points)
Each entry in the register file in a pipelined processor with register forwarding has two distinguished bits "B" (busy) and "RP" (read pending).

Given:
I1: R1 ← R2 + R3
I2: R4 ← R1 + R5

And the state of the pipeline:

I2 I1

Assuming no other instructions are in flight, fill in the state of the B and RP bits in the register file.

-1 for each wrong placement of 'i'
-1 for each missing 'i'

		B	RP
R0		0	0
R1		1	0
R2		0	0
R3		0	0
R4		1	0
R5		0	0
R6		0	0
R7		0	0

CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(b) (8 points)
Given:
I1: R1 ⟵ R2 + R3
…..
…..
I2: R4 ⟵ R1 + R5

+0.5 for each correct entry

Fill in the table below:

Number of unrelated instructions between I1 and I2	Number of bubbles without register forwarding	Number of bubbles with register forwarding
0	3	0
1	2	0
2	1	0
3 or more	0	0

Given:
I1: R1 ⟵ Memory[R2+offset]; load R1 with contents of memory at R2+offset
…..
…..
I2: R4 ⟵ R1 + R5

Fill in the table below:

Number of unrelated instructions between I1 and I2	Number of bubbles without register forwarding	Number of bubbles with register forwarding
0	3	1
1	2	0
2	1	0
3 or more	0	0

(c) (8 points)
One conservative way of handling branches is to stop new instructions from entering the pipeline when the decode stage encounters a branch instruction. Once the branch is resolved, normal pipeline execution can resume, either along the sequential path of control or along the target of the branch. Recall that for a BEQ instruction, the outcome of the branch is known only at the end of the EX cycle.

Given the following sequence of instructions:

```
      BEQ      L1
      ADD
      LW
      ….
L1    NAND
      SW
```

Using conservative approach and **assuming branch is taken**, what is the observed CPI for the 3 instructions (BEQ, NAND, SW)?

-3 if pipelining concept not understood
-2 for conceptual errors (missing/extra NOPs)

CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

You may use the following table to chart out the passage of instructions through the pipeline

Cycle number	IF	ID/RR	EX	MEM	WB
1	BEQ				
2	ADD	BEQ			
3	ADD+	NOP	BEQ		
4	NAND	NOP	NOP	BEQ	
5	SW	NAND	NOP	NOP	BEQ
6		SW	NAND	NOP	NOP
7			SW	NAND	NOP
8				SW	NAND
9					SW
10					

+: ADD instruction is stuck in IF stage until BEQ is resolved
Total number of cycles taken = 9
Observed CPI = 9/3 = 3

-1 for minor error (extra cycle)

Process Scheduling

3. (9 points, 5 min)
(a) (3 points) (Select one correct choice)
A program in execution has
1. Exactly one active entity called a process
2. Multiple active entities called threads
3. Two active entities, one a thread and the other a process
4. Three active entities, one a task, the second a thread, and third a process
5. Four active entities, one a job, the second a task, the third a thread, and the fourth a process

(b) (3 points) (Select one correct choice)
One of the following is **NOT** part of the state of a running program
1. General Purpose Registers that are visible to the instruction set
2. Program counter and the register that represents the stack pointer
3. Layout of the program in memory
4. Priority information
5. Internal registers in the datapath of the processor

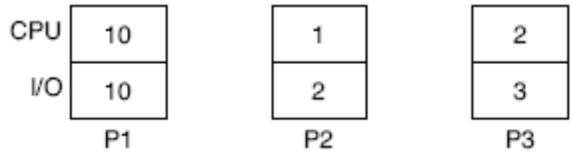
(c) (3 points) (Select one correct choice)
To implement a preemptive scheduling algorithm we need
1. A trap instruction
2. An external interrupt
3. The currently running process to terminate
4. The currently running process to make an I/O request

CS 2200 Spring 2012 Test 2

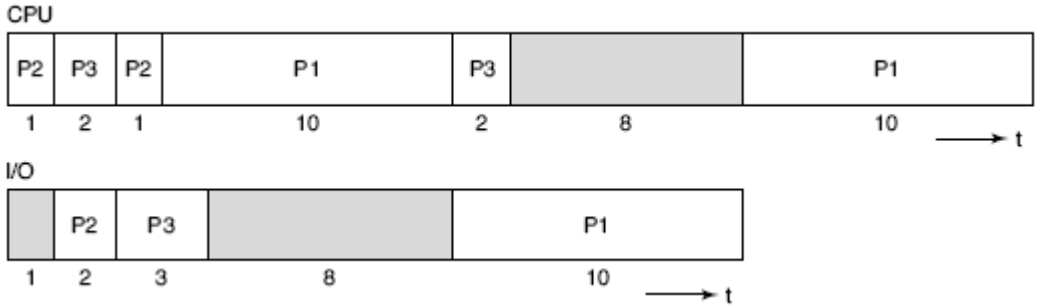
Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

4. (10 points, 5 min)
Consider the following three processes in the scheduling queue. Each process does one CPU burst, followed by one I/O burst, and completes its execution with one more CPU burst. The processes arrive in the order P1, P2, P3. P3 has the highest static priority, followed by P2, and P1 has the lowest priority.



Given the following schedule:



(a) (4 points) (Select one of the following)
The above schedule represents
1. FCFS
2. SJF
3. SJF with preemption
4. FCFS with preemption
5. Static Priority
6. Static Priority with preemption

(b) (3 points) (Select one of the following)
The throughput of the system is
1. 1/11 processes/unit-time
2. 3/24 processes/unit-time
3. 3/34 processes/unit-time

(c) (3 points)
What is the waiting time experienced by each for P1, P2, and P3?

Tp1 = 34; Ep1 = 30; Wp1 = Tp1 - Ep1 = 34-30 = 4
Tp2 = 4; Ep2 = 4; Wp2 = Tp2 - Ep2 = 4-4 = 0
Tp3 = 16; Ep1 = 7; Wp3 = Tp3 - Ep3 = 16-7 = 9

+1 for each correct Wpi

CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Memory Management and Virtual Memory (Note: K = 1024)

5. (10 min, 5 min)
(a) (5 points)
Consider a fixed size partition memory management scheme implemented for a byte-addressed machine. The memory manager supports

- 10 partitions of 1 Kbytes
- 3 partitions of 3 Kbytes
- 1 partition of 5 Kbytes

What is the largest internal fragmentation that can occur with this allocation scheme?

Maximum internal fragmentation possible = 5 Kbytes – 1 = 5120 – 1 = 5119

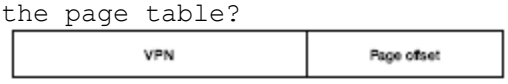
(b) (2 points) (Select one correct choice)
With variable size partition memory management scheme there can be

1. No external fragmentation
2. No internal fragmentation
3. No fragmentation
4. Both internal and external fragmentation

(c) (3 points) (Select one correct choice)
In a byte-addressed machine, if the pagesize is 8192 bytes, the **maximum** possible internal fragmentation is

1) 8192 bytes
2) 1 byte
3) 8191 bytes
4) There is no internal fragmentation with paging

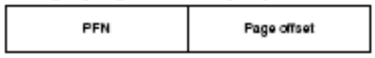
6. (10 points, 5 min)
(a) (3 points)
Virtual address is 32 bits; pagesize 4Kbytes; How many entries are there in the page table?



Bits for page offset = $\log_2$ (pagsize) = 12
Bits for VPN = 32-12 = 20
Number of PTEs = 2^{20}

+1 for each line
or
+3 for 2^{20}

(b) (3 points)
For the same memory system as in (b), the physical address is 28 bits. How many physical page frames does the memory system have?



Bits in page offset = 12
Bits in PFN = 28-12 = 16
Number of physical page frames = 2^{16}

+1 for each line
or
+3 for 2^{16}

(c) (2 points)
In a computer with the above memory system (32-bit virtual address; 28 bit physical address; 4 Kbyte page size), currently 8 processes are executing. How many page tables are in memory?

Number of page tables = degree of multiprogramming = 8

(d) (2 points)
What function does the Page Table Base Register (PTBR) serve in the CPU?

Points to the starting address of the page table for the currently executing process.

Demand paging, Working set, page replacement, TLB

7. (10 points, 5 min)

Give a one line description of each of these data structures used by a demand paged memory manager

(a) (2 points)
Freelist: List of unallocated physical frames in memory

(b) (2 points)
Frametable: Given a PFN, returns <PID, VPN>

(c) (2 points)
Disk Map: Given a <PID, VPN> returns the disk location for the page

(d) (2 points)
Page table: Given a VPN, returns the PFN for a given process

(e) (2 points)
Process Control Block: One entry in the PCB, PTBR contains the base address of the page table for this process, loaded into the PTBR register of the CPU when this process is scheduled to run.

CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

8. (10 points, 5 min)

(a) (2 points)(Select one correct choice)
The reference bit for supporting an approximate LRU page replacement scheme is implemented by associating
1. One bit per page table entry
2. One bit per memory location
3. One bit per physical page frame
4. One bit per process
5. One bit for the entire physical memory
6. One bit for the entire page table

(b) (2 points) (Select one correct choice)
The size of LRU stack for a TRUE LRU scheme is
1. 3
2. Equal to the number of virtual pages
3. Equal to the number of physical frames
4. Equal to the size of the virtual address space

(c) During the time interval t1 - t2, the following virtual page accesses are recorded for the processes P1 and P2.

P1: 0, 0, 1, 2, 22, 2, 0, 0, 1, 1, 2, 0
P2: 0, 0, 10, 1, 2, 0, 1, 12, 20

(2 points) What is the **working set** for P1 in this time interval?

WSp1 = {0, 1, 2, 22}

(2 points) What is the working set for P2 in this time interval?

WSp2 = {0, 1, 2, 10, 12, 20}

(2 points) What is the total **memory pressure** on the system during this interval?

Total memory pressure = WSSp1 + WSSp2 = 4+6 = 10

Caching

9. (10 points, 5 min)
(a) (2 points) (Select one correct choice)
Spatial locality suggests that
1. Once brought into the cache, we should keep the data around as long as possible
2. On a miss, we should bring in adjacent memory locations into the cache
3. The memory location being brought in due to a miss is not likely to be referenced in the future

CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(b) (2 points) (Select one correct choice)
A direct-mapped cache

1. Has a one-to-one mapping between a memory location and a cache location
2. Allows a memory location to be en-cached wherever there is space in the cache
3. Is so called because there is a directory associated with the contents of the cache
4. Is usually much smaller than any other type of cache organization

(c) (2 points) (Select one correct choice)
In an 8-way set associative cache with 64 Kbytes of data, 64 bytes per block and with a t-bit tag

1. There are **four t-bit** tag comparators
2. There are **eight t-bit** tag comparators
3. There are **64 t-bit** tag comparators
4. There are **1K t-bit** tag comparators
5. There are **125 t-bit** tag comparators
6. There is one **t-bit** tag comparator for the **entire cache**

(d) (4 points)
A pipelined processor has an average CPI of 1.1 not considering memory effects. On an average each instruction has an I-cache miss of 0.5%, and a D-cache miss of 0.5%. The miss penalty is 100 cycles. What is the effective CPI taking into account memory stalls?

Average memory stalls due to cache misses
= (I-cache miss rate + D-cache miss rate) * miss penalty } +1 for each line
= (0.005+0.005) * 100
= 1
Effective CPI
= Average CPI without memory effects + Average memory stalls } +1
= 1.1+1
= 2.1

CS 2200 Spring 2012 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

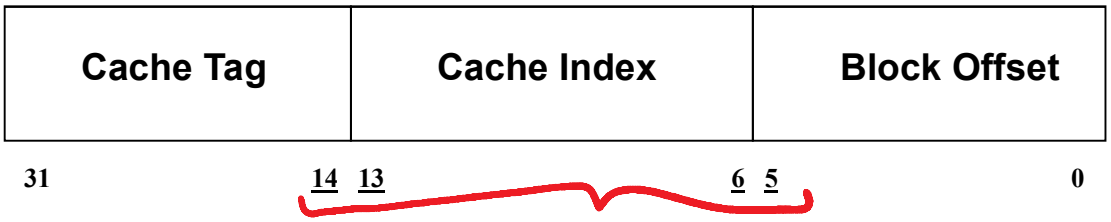
10. (10 points, 5 min)

Consider a 4-way set-associative cache for byte addressed processor.

- Data size of cache = 64KB.
- CPU address = 32 bits
- Memory word = 4 bytes.
- Cache block size = 64 bytes.
- Write policy is Write-back at the granularity of individual words.
- Cache replacement policy = LRU

a) (5 points)

The memory address is interpreted as follows:



Fill in the blanks above to indicate how the memory address is interpreted for looking up the cache.

-1 for each wrong entry

b) (5 points)

Show one cache line clearly indicating the size of the metadata (valid, dirty, and tag bits) and the size of the data fields.

There are four parallel caches. One row of each cache has the following structure:

-1 for each wrong entry

Meta data			Datablock
V	D1-D16	Tag	data
1 bit	16 bits	18 bits	64 bytes

An entire cache line is cumulatively the data and meta data in a given row of the four parallel caches.

* Note: We don't count off if you do not show LRU meta data. An approximate LRU can be implemented with 2-bits for each cache line. A true LRU will require 5 bits per cache line and a state machine to remember the access order for each cache line.