CS 2200 Spring 2011 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Problem	Points	Lost	Gained	Running Total	TA
1	1				
2	15				
3	5				
4	15				
5	14				
6	20				
7	5				
8	10				
9	15				
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, 1 min) (circle one)
Which of the following teams are in the semifinals of the **2011 World Cup Cricket** (circle all that apply)?

a) Australia b) South Africa c) India
d) New Zealand e) Pakistan f) USA
g) Sri Lanka h) England i) West Indies

CS 2200 Spring 2011 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Pipelining

2. (15 points, 10 min)

(a) (2 points) (Select one **correct** choice)

With reference to the 5-stage pipelined implementation of LC-2200 instruction set, a second ALU is needed in the EXEC stage of the pipeline

1) To make the stage symmetrical
2) To enable address calculation for LW/SW instructions
3) To enable address calculation for BEQ instruction
4) To enable multi-precision arithmetic for ADD instruction
5) To enable ADD and NAND instructions to be executed in parallel if they occur one after another

(b) (3 points)

What role does Branch Target Buffer play in a pipelined processor design?

- Remember the outcome of previously executed branches
- Use the past history in BTB to predict future branches

(+1.5 for each of the above points; -2 for not mentioning that it serves as a cache for recent branch results and targets of branches)

(c) (5 points)

I_1: R1 <- R2 + R3

I_2: R4 <- R1 + R5

If I_2 is immediately following I_1 in the pipeline with **no forwarding**, how many bubbles will be experienced by the pipeline? **You must explain your answer to get any credit**.

Cycle	IF	ID/RR	EX	MEM	WB	
1	**I2**	**I1**				
2		I2	I1			
3		I2	NOP	I1		
4		I2	NOP	NOP	I1	
5		I2	NOP	NOP	NOP	-> I1 (R1 written at the end of WB cycle)
6			I2	NOP	NOP	-> NOP

Three bubbles in the pipeline before I2 can read the registers and move to the next stage.

(-3 if the number of bubbles less than 3; -2 if reasoning is flawed but 3 bubbles mentioned)

CS 2200 Spring 2011 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(d) (5 points)

I_1: LW R1, 0(R2); load R1 from memory

I_2: R4 <- R1 + R5

If I_2 is immediately following I_1 in the pipeline with **data forwarding**, how many bubbles will be experienced by the pipeline? **You must explain your answer to get any credit**.

Cycle	IF	ID/RR	EX	MEM	WB	
1	**I2**	**I1**				
2		I2	I1			
3		I2	NOP	I1		(in MEM stage I1 can forward the data read from memory to I2)
4			I2	NOP	I1	

One bubble in the pipeline before I2 can read the registers and move to the next stage.

(-3 if the number of bubbles is different from 1; -2 if reasoning is flawed but 1 bubble mentioned)

Process Scheduling

3. (5 points, 5 min)
(Use each of the following terms EXACTLY ONCE: "FCFS", "waiting", "SJF", "ready", "PCB", "disk image", "round robin". NOTE: NOT ALL TERMS ARE USED.)

a) If process A is currently running and makes a request to read a large block of data from the disk, the OS puts this process in the ____**waiting**_____ state.

b) If process A is currently running and the OS preempts A since its time quantum is up to run a different process B, then the OS puts process A in the __**ready**_____ state.

c) To dispatch a process A to run on the processor, the OS must take register values from A's ____**PCB**_____ and load them into actual processor registers.

d) If fairness is more important than average waiting time experienced by the processes, we should use ___**FCFS**_____ non-preemptive scheduling algorithm.

e) If we want to optimize the average waiting time for processes, then we should use ____**SJF**_____ non-preemptive scheduling algorithm.

CS 2200 Spring 2011 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

4. (15 points, 15 min)
Recall that *Shortest Remaining Time First (SRTF)* is a variant of Shortest Job First (SJF) with preemption added in. Consider the following three processes vying for the CPU. The scheduler uses SRTF. The scheduler re-evaluates which process to run only upon the arrival of a new process into the scheduling queue, or the completion of a process. The table shows the arrival time of each process.

Process	**Arrival Time**	**Execution Time**
P1	T_0	2ms
P2	T_0+1ms	3ms
P3	T_0+2ms	1ms

The scheduling starts at time T_0.

(a) (6 points) Fill in the table below with the process that is executing on the processor during each time slot.

Interval T_0+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Running	**P1**	**P1**	**P3**	**P2**	**P2**	**P2**							

(+1 for each correct entry)
Use the following tables to show your work as to how you arrived at the above schedule.

Time T_0:

Process	Remaining time
P1	2ms
P2	Not arrived yet
P3	Not arrived yet

Time T_0+1: (2 points) (Process P2 arrives)
(+1 for each of P1 and P2)

Process	Remaining time
P1	1 ms
P2	3 ms
P3	Not arrived yet

Time T_0+2: (3 points) (Process P3 arrives)
(+1 for each of P1, P2, P3)

Process	Remaining time
P1	0
P2	3 ms
P3	1 ms

(b) (2 points) What is the total waiting time for all processes?
P1 waits for 0 ms
P2 waits for 2 ms
P3 waits for 0 ms
Total waiting time = 2 ms

(all or nothing)

(c) (2 points) What will be the total waiting time for all processes with an FCFS schedule?
P1 waits for 0 ms
P2 waits for 1 ms
P3 waits for 3 ms
Total waiting time = 4 ms

(all or nothing)

Memory Management and Virtual Memory

5. (14 points, 10 min)
(a) (2 points) (choose one of the following)
In a byte-addressed machine, if the pagesize is 8192 bytes, the **maximum** possible internal fragmentation is
1) 8192 bytes
2) 1 byte **(all or nothing)**
3) 8191 bytes
4) There is no internal fragmentation with paging

(b) (4 points)
Virtual address is 32 bits; pagesize 8192 bytes (8 Kbytes); How many entries are there in the page table?

Number of bits in offset:
$2^{\text{offset-bits}} = 8192$ **(-2 if this calculation is wrong)**
Number of offset bits = 13 bits
Number of bits in VPN = address bits – number of offsets bits
= 32 – 13
= 19 bits
(-2 if this calculation is wrong)
(no double jeopardy)

Number of entries in the page table = $2^{VPN} = 2^{19}$

(c) (4 points)
For the same memory system as in (b), the physical address is 24 bits. How many physical page frames does the memory system have?

Number of bits in PFN = physical address bits – number of offset bits
= 24 - 13
= 11 bits

CS 2200 Spring 2011 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Number of physical page frames = 2^{PFN} = 2^{11} = 2048

(-2 if this calculation is wrong)

(d) (2 points)
Total memory size is 24K. The current state of the memory allocation table for a variable-sized partition allocation scheme is as follows:

Start address	Size	Who has it?
0	4K	P1
4096	4K	P2
8192	4K	FREE
12288	2K	P3
14336	10K	FREE

What is the total amount of external fragmentation?
External fragmentation = sum of all the dis-contiguous FREE memories
= 4K + 10K = 14 K

(all or nothing)

(e) (2 points)
The necessary conditions for a paging memory system are (select one of the following choices)

1) The virtual and physical addresses have to be of the same size
2) The sizes of the virtual page and the physical frame have to be the same
3) The number of physical frames should be larger than the number of virtual pages
4) (1) and (2)
5) (2) and (3) **(all or nothing)**
6) (1) and (3)
7) (1), (2), and (3)

Working set, page replacement, TLB

6. (20 points, 15 min)
During the time interval t1 - t2, the following virtual page accesses are recorded for the process P1.

P1: 0, 10, 1, 0, 1, 2, 10, 2, 1, 1, 0

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

P1's working set = {0, 1, 2, 10} **(all or nothing)**

b) (3 points) What is the **memory pressure** on the system during this interval?

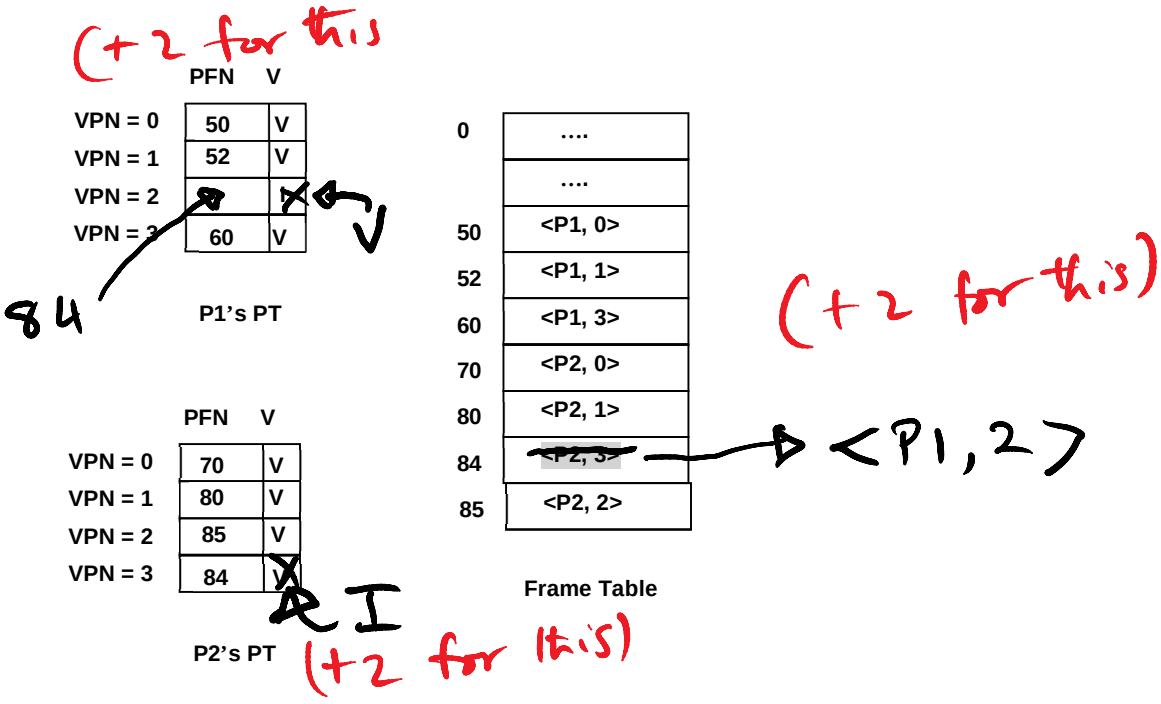
Memory pressure = P1's working set size = 4 **(all or nothing)**

CS 2200 Spring 2011 Test 2

Prism ID:_____

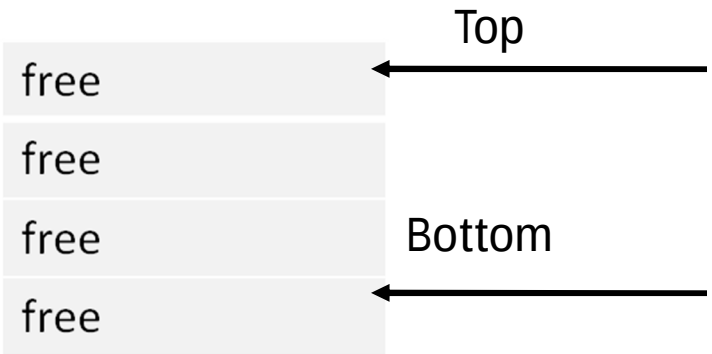
Name:_____Kishore_____ GTID#: 9_____

c) (6 points)
P1 incurs a page fault at VPN = 2. The victim page frame chosen by the page replacement algorithm is PFN = 84. The "before" picture of the page manager's data structure are shown below. Note that only relevant entries of the Frame Table are shown in the Figures below. Show the contents of the data structures "after" the page fault is serviced. **You can simply mark the changes in the figure below to show your answer.**



d) (9 points)
A processor has a 4-entry TLB that uses a least recently used (LRU) replacement policy. The program executed by this processor generates the following sequence of memory requests (only the page number of each request is shown). Assuming that the TLB is initially empty, which of these memory requests will result in TLB misses?

LRU stack



CS 2200 Spring 2011 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

State of TLB

Page #	TLB Miss?	T	B
0xF000	Y		
0xE000	Y		
0x10AB	Y	0x10AB, 0xE000, 0xF000, —	
0xF000	N	0xF000, 0x10AB, 0xE000, —	
0x2000	Y		
0x2001	Y	0x2001, 0x2000, 0xF000,	0x10AB
0x2000	N	0x2000, 0x2001, 0xF000,	0x10AB
0xF000	N	0xF000, 0x2000, 0x2001,	0x10AB
0x4000	Y		
0xE000	Y	0xE000, 0x4000, 0xF000,	0x2000
0x10AB	Y	0x10AB, 0xE000, 0x4000,	0xF000
0x2001	Y	0x2001, 0x1AB, 0xE000	0x4000
0x10AB	N	0x10AB, 0x2001, 0xE000,	0x4000
0x2000	Y	0x2000, 0x10AB, 0x2001,	0xE000
0x2001	N	0x2001, 0x2000, 0x10AB,	0xE000

Caching

7. (5 points, 5 min)
A pipelined processor has an average CPI of 1.2 not considering memory effects. On an average each instruction has an I-cache miss of 1%, and a D-cache miss of 1%. The miss penalty is 20 cycles. What is the effective CPI taking into account memory stalls?

$CPI_{effective} = CPI_{Avg}$ + memory stalls **(+1)**

Memory stalls = Memory stalls due to I-cache + Memory stalls due to D-cache
= 0.01 * 20 + 0.01 * 20
= 0.2 + 0.2
= 0.4 **(+3)**

$CPI_{effective}$ = 1.2 + 0.4
= 1.6 **(+1)**

8. (10 points, 10 min)
Consider the following memory hierarchy:
L1 cache: Access time = 2ns; hit rate = 99%
L2 cache: Access time = 10ns; hit rate = 95%
Main memory: Access time = 100ns
Compute the effective memory access time.

$EMAT_{L2} = T_{L2} + (1-h_{L2}) * T_{MEM}$
$= 10 + (1-0.95) * 100$
$= 10 + 5$
$= 15$ ns **(+5)**

$EMAT_{L1} = T_{l1} + (1-h_{L1}) * EMAT_{L2}$

```
      = 2 + (1-0.99) * 15
      = 2 + 0.15
      = 2.15                                    (+5)
```

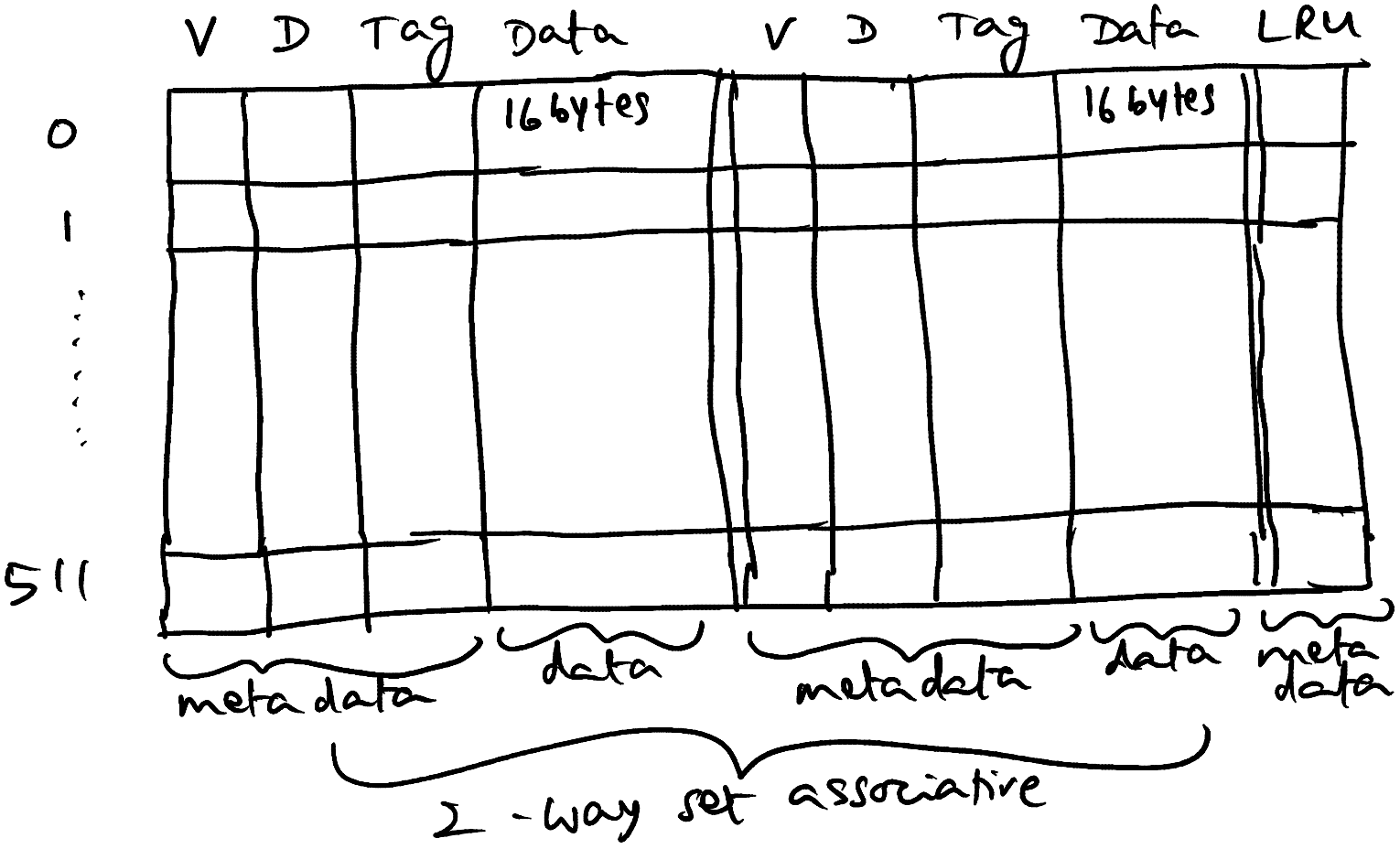
EMAT = $EMAT_{L1}$ = 2.15 ns

(-1 for minor math errors; -1 if hit-rate is used incorrectly in calculating EMAT; -4 if L2 or MEM access time calculation is flawed)

9. (15 points, 15 min)
Consider a 2-way set-associative cache for byte addressed processor.

- Data size of cache = 16KB.
- CPU address = 32 bits
- Memory word = 4 bytes.
- Cache block size = 16 bytes.
- Write policy is Write-back at the granularity of individual words.
- Cache replacement policy = LRU

a) (5 points) Show the layout of the cache, clearly showing the fields of the cache (the data and metadata). The sketch should show the number of lines in the cache, the amount of data in each line, and the fields of the metadata in each line.



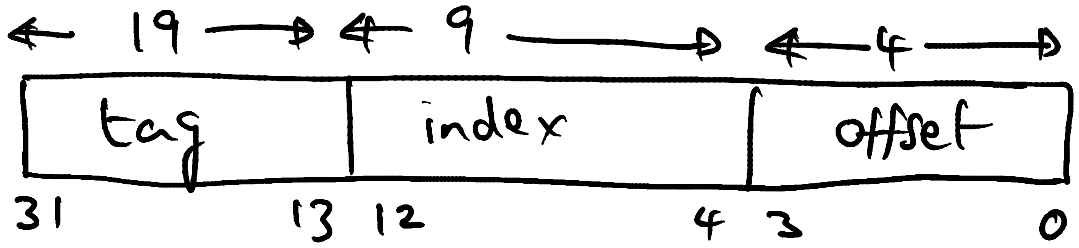
CS 2200 Spring 2011 Test 2

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(-1 if 2-way not shown; -1 if number of lines not 511; -1 if data size not shown; -2 if not all meta data shown)

b) (5 points) Show how the CPU interprets the memory address (i.e., which bits are used as offset, index, and tag).



(+2 for tag; +2 for index; +1 for offset)

c) (5 points) Compute the size of the metadata in **each line of the cache** (remember that this is a 2-way set-associative cache). Show your work to get partial credit.

Number of valid bits per line = 2 (1 bit for each of the two parallel caches)
(+1)

Number of dirty bits per line = 8 (4 for each of the two parallel caches)
(+1)

Number of LRU bits = 1 (to indicate which of the two caches is the LRU victim)
(+1)

Number of tag bits = 38 (19 for each of the two parallel caches)
(+1)

Total size of metadata in each line = 2 + 8 + 1 + 38 = 49 bits
(+1)