CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Problem	(Points, Min)	Lost	Gained	Running Total	TA
1	(1, 0min)				
2	(15, 10min)				
3	(14, 10min)				
4	(15, 10min)				
5	(14, 10min)				
6	(16, 10min)				
7	(12, 15min)				
8	(13, 15min)				
Total	(100, 80min)				

- **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; you get a point regardless of correct/incorrect answer)
"It's tough to make predictions, especially about the future."
(a) Said Gubernatorial candidate Jason Carter on his chances of winning the election
(b) Said a famous computer scientist about choosing a victim page for replacement
(c) Said President Obama about Obamacare surviving after the elections
(d) Said Mahatma Gandhi about British rule in India prior to Independence
(e) Said Yogi Berra about life in general

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

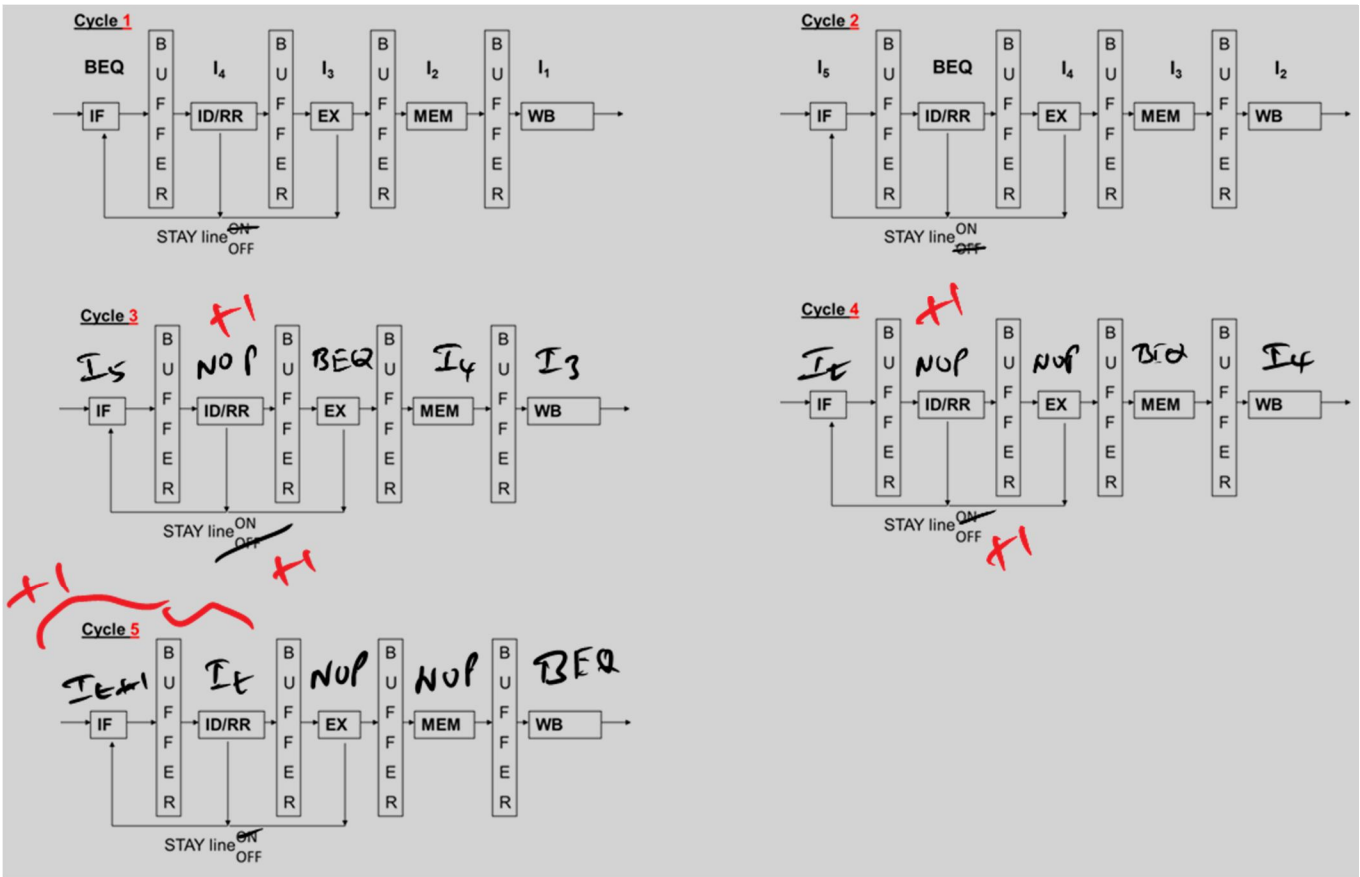
Name:_____Kishore_____ GTID#: 9_____

Pipelining

2. (15 points, 10 min)

(a) (5 points)consider a pipelined processor that uses **conservative handling of branch instruction**. For this question, assume there are NO data or control or structural hazards among any of the instructions shown in flight or that would follow the branch instruction. Let $\mathbf{I_t, I_{t+1}}$, … be the sequence of instructions from the target of the BEQ instruction. The "STAY line" can be turned ON or OFF by the ID/RR or the EX stage, to signal to the IF stage to NOT send new instructions down the pipeline (in this case, the IF stage will send NOP instructions to the next stage). **We have shown what happens in the first two cycles. Assuming that the branch is taken,** for EACH subsequent cycle show

- **Which** instructions are in the different stages of the pipeline
- **The state** (ON/OFF)of the "STAY line"



CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

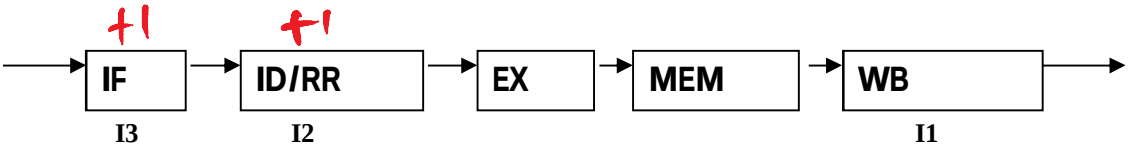
Name:_____Kishore_____ GTID#: 9_____

(b) Consider the following three consecutive instructions in a program in flight in a pipelined processor with **register forwarding**:

```
I1: R1 <---  R2 + R3
I2: R1 <---  R1 + R4
I3: R5 <---  R1 + R6
```

Partial credit Rubric:
-1 if WAW not recognized
-2 if WAW recognized but solution does not show how hardware disambiguates which stage supplies to I3

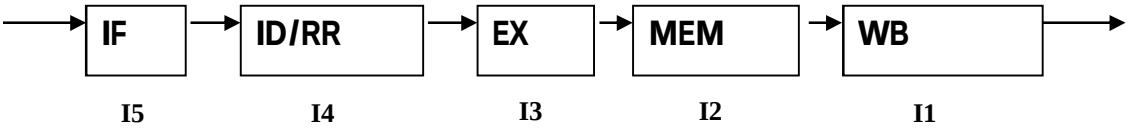
The state of the pipeline is:



(i) (2 points) Show in the **above figure**, **where I2** and **I3** are.
(ii) (3 points) Why are **I2 and I3** where you have shown them to be?

I2 has a WAW hazard with respect to I1 (same destination register R1). So I2 cannot progress beyond ID/RR despite register forwarding. Hence I3 is in the IF stage.

(c) (5 points) There are 5 instructions in flight as shown below. An external interrupt can occur asynchronously at any time.



Suggest a scheme for dealing with the external interrupt. Your answer should clearly state what happens to the instructions already in the above pipeline, and where the original program will be restarted when it is resumed.

- Ex stage of the pipeline samples the INT line
- If INT is on
  - Allow I1 and I2 to complete execution
  - Send bubbles (i.e., NOP instruction) down the pipe for 2 cycles to "drain" the pipe while I1 and I2 are moving through the pipeline
  - Squash I4 and I5; those stages also send bubbles instead of I4 and I5.
  - Load the PC value (corresponding to I3) into $K0 and enter the INT macro state
  - The program will resume execution upon return from the handler from I3 (i.e., the IF stage will fetch I3 and resume normal execution from thereon)

Partial credit Rubric:
-1 for not showing PC -> $K0
-1 for not being specific on which stage catches the interrupt

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Process Scheduling

3. (14 points, 10 min)

(a) (2 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

(b) (2 points) (**True/False with Justification**)

"The role of the dispatcher is to select a new process for running on the processor."

+1 False. Scheduler selects; dispatcher simply loads the volatile state of the process from PCB into the processor +1

(c) (2 points) (**True/False with Justification**)

"A non-preemptive scheduling algorithm will schedule a different process to run **ONLY** when the current one terminates."

+1 False. Also when the current process blocks on an I/O call. +1

(c) (2 points) (**True/False with Justification**)

"Upon context switch, the scheduler saves the volatile state of the current process in the system stack."

+1 False. In the PCB associated with the process. +1

(d) (2 points) (**True/False with Justification**)

"If you are the system administrator at a google data center, your raise most likely depends on the how good the CPU utilization of the servers are."

+1 False. System throughput. +1

(e) (2 points) (**True/False with Justification**)

"SJF will never result in starvation."

+1 False. Newly arriving "short" jobs can prevent a "long" job from ever getting a turn on the processor. +1

(f) (2 points) (**True/False with Justification**)

"It is impossible to implement a preemptive scheduling algorithm without a timer interrupt."

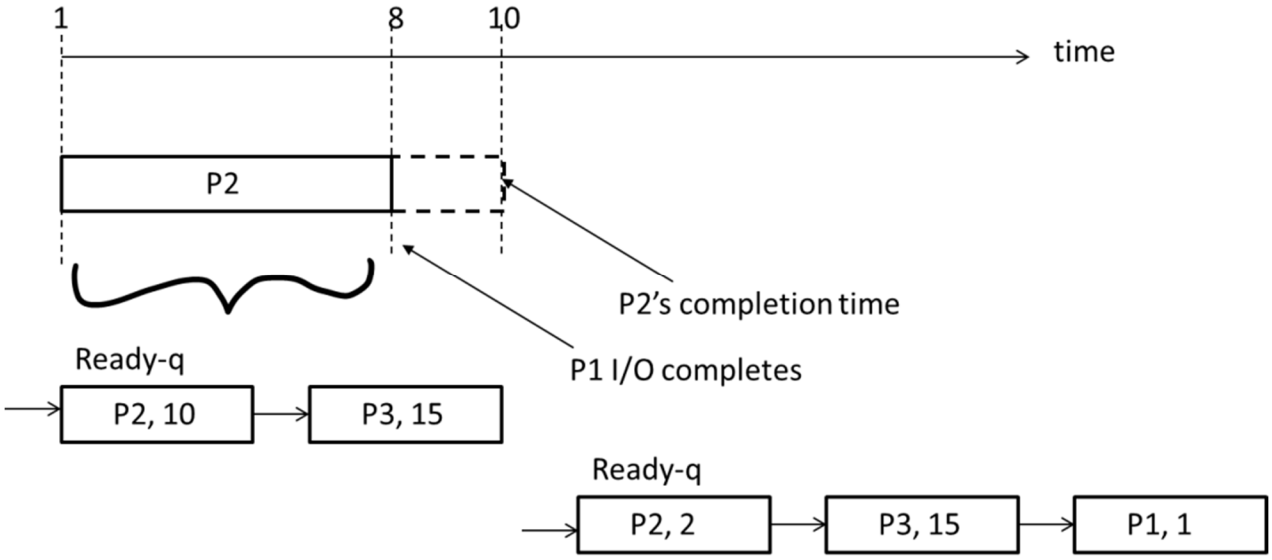
+1 False. At opportune moments (e.g., a process joining the ready queue upon I/O completion, or a new process startup) the scheduler can make a preemption decision commensurate with the scheduling policy. +1

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

4. (15 points, 10 min)
(a) (5 points) The scheduling discipline in use is Shortest Remaining Time First (SRTF). Given the timeline below:



- P2 is executing currently; (Ready queue as shown on left)
- P1 completes I/O at time = 8;
- Ready queue changes as shown on the right.

Explain what (if anything) will happen as a result of this change in the ready-queue?

+1 • When Pi rejoins the ready-q, the SRTF schedule re-evaluates who should run
+1 • P2's remaining time is 2 units (already completed 8 of its 10 unit burst)
+1 • But P1 needs only 1 unit of time
+1 • P2 is preempted at time = 8
+1 • P1 is scheduled to run at time = 8

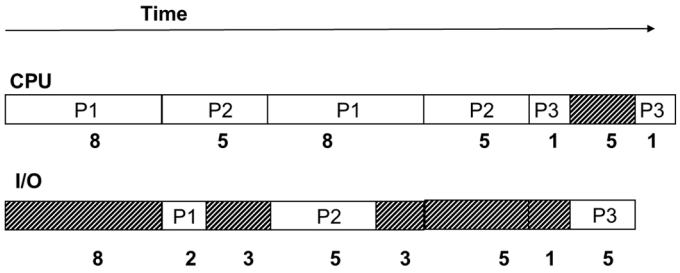
Partial credit Rubric:
-2 for not recognizing that SRTF is a preemptive scheduler

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(b) There are three processes all of which arrive in the order P1, P2, P3. All of them are ready to be scheduled at **time 0**. Given the schedule below:



(i) (2 points) What type of scheduler will result in the above schedule?

The fact that scheduler picked P1 to run first despite it being the longest job at time = 0, indicates this is a FCFS scheduler.

(ii) (2 points) What is the turnaround time of P1?
Turnaround time of P1 = running time + wait time = (8+2+8)+ (3) = 21

(ii) (2 points) What is the wait time of P2?
Wait time for P2 = Turnaround time – running time
= 26 – (5+5+5) = 11

(iii) (2 points) What is the observed throughput of the system?

Throughput = Number of processes completed/time taken
= 3/33 = 1/11 processes per unit time

All or nothing in each

(c) (2 points) (**True/False with Justification**)
"Linux scheduler does its work in O(1) time."

+1 True. There are finite number of priority levels (140). The head of each level is in kept in array. So in constant time (irrespective of the number of tasks at each level) the scheduler picks the next task to run by inspecting the array elements. +1

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

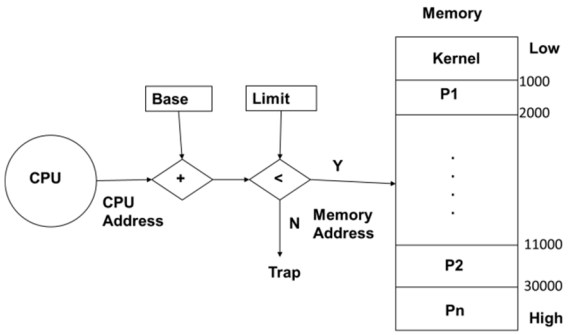
Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

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

5. (14 points, 5 min)

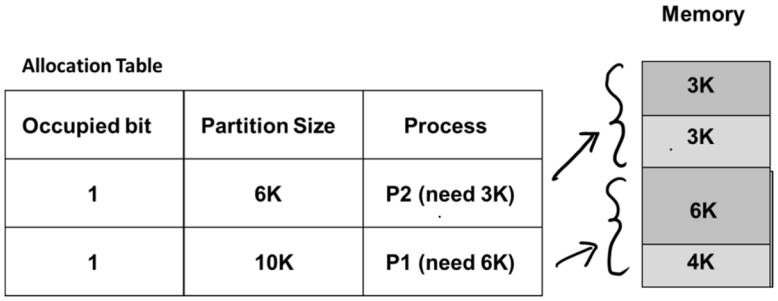
(a) (2 points) A given architecture uses "Base" and "Limit" registers for memory protection of individual processes. P1 is currently executing on the processor



What are the contents of Base and Limit registers?

Base = 1000 +1
Limit = 2000 +1

(b) Consider the following allocation table with fixed-size partition memory allocation.



(i) (2 points) How much is the total internal fragmentation?

Internal fragmentation = 7K (3 K wasted by P2 and 4K wasted by P1) All or nothing

(ii) (2 points) How much is the external fragmentation?

0

(c) (2 points)(**True/False with Justification**)
"Variable size partition" memory allocation suffers from internal fragmentation."

+1 False. Each process is given exactly what they need. +1

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(d) (3 points)
Virtual address is 64 bits; pagesize 8 Kbytes; How many entries are there in the page table? (show your work)

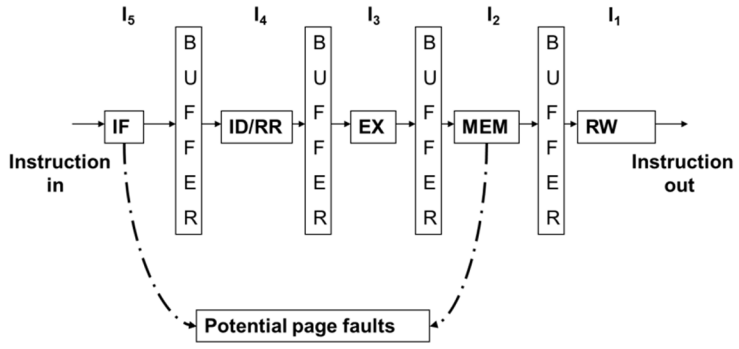
Number of bits in the VA for page offset = $\log_2 8K$ = 13 bits +1
So number of bits for VPN = 64-13 = 51 bits +1
So number of page table entries = 2^{51} +1

(e) (3 points)
For the same memory system as in (d), the physical address is 48 bits. How many physical page frames does the memory system have? (show your work)

The number of bits in the physical address for PFN = 48-13 = 35 +2
The maximum number of physical page frames = 2^{35} +1

Demand paging, Working set, page replacement

6. (16 points, 15 min)
(a) Consider a 5-stage piplelined processor, which uses demand-paged virtual memory management. **Instruction I2 incurs a page fault.**



(i) (2 points) What will happen to the instructions in flight in the processor?

I1 will complete; +1
I3, I4, and I5 will be squashed; +1

(ii) (2 points) At which instruction will the program be resumed?

I2, after page fault service is complete +2

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(iii) (2 points) Given VPN = 0x501 maps to PFN = 0x11E0, what is the physical address corresponding to the virtual address 0x5010F8E0

PA = 0x11E0 0F8E0 +2

(b) (2 points)
Working set of Process1 = {p1, p22, p53};
working set of Process2 = {p1, p2, p53, p104}, where pi refers to virtual page "i" of a given process.

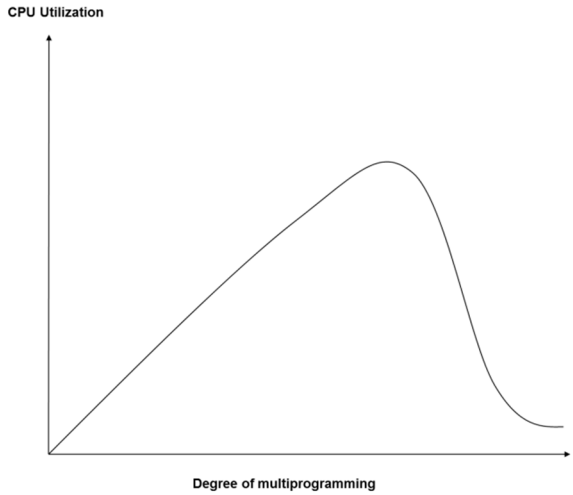
What is the current memory pressure of the system?

Memory pressure = WSS_{P1} + WSS_{P2} = 3 + 4 = 7 +2
(note that p1 of Process 1 is NOT same as p1 pf Process 2 since they are virtual pages)

(c) (2 points) (**True/False with Justification**)
"Clock sweep is just another name for FIFO page replacement algorithm."

+1 False. It is FIFO with a second change given to the FIFO candidate before being replaced. +1

(d) (2 points) (**True/False with Justification**)



"In the picture above, the reason for CPU utilization decrease as we increase the degree of multiprogramming beyond a certain point is due to all processes alternating between CPU bursts and I/O bursts."

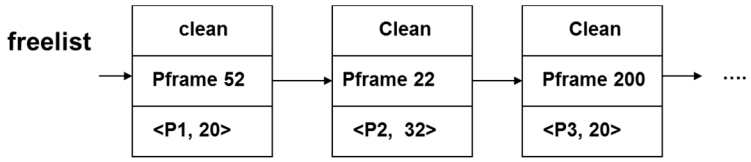
+1 False. It is due to the memory pressure increasing beyond the point of sustainability (i.e., the total memory pressure exceeds physical memory capacity of the system). +1

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(e) (4 points)
The freelist (as shown below) is a data structure of the memory manager that contains the pool of page frames that are available for allocation to satisfy page faults.



Each entry in the freelist, shows the page frame number that is available for allocation as well as the reverse mapping, i.e., the process-ID and the VPN of that process that was hosted in this page frame.

Process P3 is currently executing on the processor. It incurs a page fault and the faulting VPN is 20.

Explain what the memory manager would do to service this page fault.

+1 - Memory manager sees that pframe = 200 contains the contents of P3's VPN = 20
+1 - The node (pframe = 200) is delinked from the free list and freed up
+1 - PT of P3 at VPN = 20 is set to PFN = 200;
+1 - Valid bit of PT for P3 at VPN = 20 is set to VALID

TLB, Cache/Memory, Execution time, Cache Design (Note: K = 1024)

7. (12 points, 15 min)
(a) (2 points) (**True/False with Justification**)
"The principle of **Spatial locality** implies that once brought into the cache, we should keep the data around as long as possible."

+1 False. Spatial locality suggests bringing in adjacent memory locations to the current missing address. +1

(b) (2 points)(**True/False with Justification**)
"On a context switch from one process to another, the entire TLB has to be flushed."

+1 False. Only the User portion of the translations need to flushed since the kernel portion is common to all processes. +1

(c) (2 points) (**True/False with Justification**)
"The motivation for **virtually indexed physically tagged cache** is to parallelize the lookup of the TLB and the first-level cache."

+1 True. By using the unchanging portion of the VA and deriving the "index" for cache lookup from the unchanging portion of the VA, we can eliminate the serial nature of the address translation and cache lookup. +1

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

(d) (6 points)
In a pipelined processor

- **average CPI = 1.8** without accounting for memory stalls.
- I-Cache has a **hit rate** of 99%
- D-Cache has a **hit rate** of 99%
- Memory reference instructions account for 25% of all the instructions executed
- Out of these memory reference instructions 80% are loads and 20% are stores.
- Read-miss penalty (for instruction or data) is 25 cycles
- Write-miss penalty is 4 cycles.

Compute the effective CPI of the processor accounting for the memory stalls.

```
I-cache miss penalty cycles = I-cache miss rate * read miss penalty
                            = 0.01 * 25 = 0.25
Data read ratio
  = memory reference instructions * load instructions
  = 0.25 * 0.8 = 0.2
D-cache read miss penalty cycles
         = Data read ratio * D-cache miss rate * read miss penalty
         = 0.2 * 0.01 * 25 = 0.05
Data write ratio
  = memory reference instructions * store instructions
  = 0.25 * 0.20 = 0.05
D-cache write miss penalty cycles
         = Data write ratio * D-cache miss rate * write miss penalty
         = 0.05 * 0.01 * 4 = 0.002
D-cache miss penalty cycles
      = D-cache read miss penalty cycles + D-cache write miss penalty cycles
      = 0.05 + 0.002 = 0.052
Clock cycles due to memory stalls   =
                  = I-cache miss penalty cycles + D-cache miss penalty cycles
                  = 0.25 + 0.052 = 0.302
Effective CPI  = Average CPI + Clock cycles due to memory stalls
               = 1.8 + 0.302 = 2.102
```

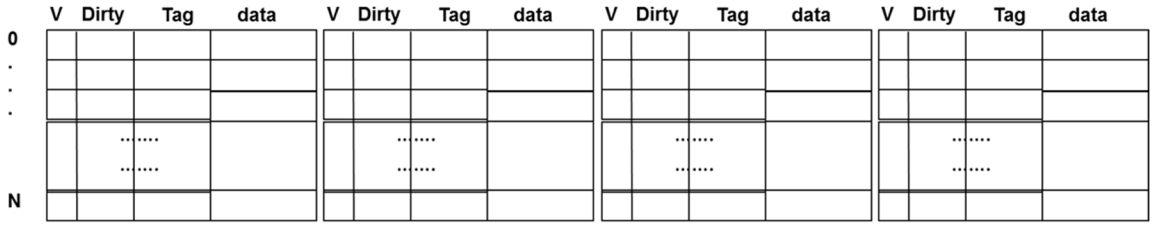
+1 +1 +1 +1 +1 +1

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

8. (13 points, 15 min)
Consider a 4-way set associative cache with the following parameters:



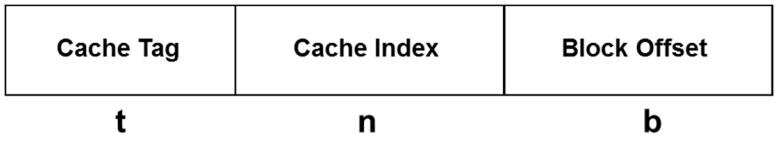
- Cache size (i.e, the amount of actual data it can hold) of **256 Kbytes**
- **64-bit byte-addressable** memory
- Each memory word contains **8 bytes**
- cache block size is **64 bytes**.
- **write-back** policy. One dirty bit per word.
- **one valid bit** per block.

(i) (4 points) How big is **N** in the above picture (show your work)?

Partial credit Rubric:
-2 for no 4-way
-2 for incorrect N

Number of data blocks = cache size / cache block size
= 256 Kbytes/64 bytes = 4096
With 4-way associativity, number of cache lines = 4096/4 = 1024
So,
N = 1023

(ii) (6 points) The 62-bit memory address is split into block offset, tag, and index as shown below:



What are the values of **t**, **n**, and **b** (show your work)?

b = $\log_2$(blocksize) = $\log_2$64 = 6 bits +2
n = $\log_2$(number of cache lines) = $\log_2$1024 = 10 bits +2
t = number of bits in memory address – (n+b)= 62 – 16 = 46 bits +2

(iii) (3 points) How many meta-data bits in **each** of the four parallel caches shown in the above picture? (show your work)
(Note: Remember that meta-data refers to all the other stuff (besides the data itself that goes into cache.)

Partial credit Rubric:
-1 for including index in meta data

8 words in each data block,
With one dirty bit per word,
The number of dirty bits per data block = 8 bits +1
Number of valid bits per data block = 1 bit +1
Number of tag bits each data block = 46 bits +1

Total number of metadata per data block = 55 bits

CS 2200 Fall 2014 Test 2 (MAX: 90 Mins)

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Acronyms

```
IF - instruction fetch (fetch into IR and increment PC)
ID/RR - instruction decode/register read (read register contents)
EX - execute (perform arithmetic/logic/address computation - maybe)
MEM - memory (fetch/store memory operand - maybe)
WB - write back to register file (may be)
TLB - Translation Look-aside Buffer
PTE - Page Table Entry
PFN - Physical Frame Number
VPN - Virtual Page Number
RLT - Reverse Lookup Table
PCB - Process Control Block
FCFS – First Come First Served
SJF – Shortest Job First
SRTF – Shortest Remaining Time First
```

Name	Notation	Units	Description
CPU Utilization	-	**%**	**Percentage of time the CPU is busy**
Throughput	**n/T**	**Jobs/s**	**System-centric metric quantifying the number of jobs *n* executed in time interval *T***
Avg. Turnaround time (t_{avg})	**$(t_1+t_2+\ldots+t_n)/n$**	**Secs**	**System-centric metric quantifying the average time it takes for a job to complete**
Avg. Waiting time (w_{avg})	**$(w_1+w_2+\ldots+w_n)/n$**	**Secs**	**System-centric metric quantifying the average waiting time that a job experiences**
Response time	**t_i**	**Secs**	**User-centric metric quantifying the turnaround time for a specific job *I***
Variance in Response time	**$E[(t_i - t_{avg})^2]$**	**$Secs^2$**	**User-centric metric that quantifies the statistical variance of the actual response time (t_i) experienced by a process (P_i) from the expected value (t_{avg})**
Starvation	-	-	**User-centric qualitative metric that signifies denial of service to a particular process or a set of processes due to some intrinsic property of the scheduler**
Convoy effect	-	-	**User-centric qualitative metric that results in a detrimental effect to some set of processes due to some intrinsic property of the scheduler**