CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________

Problem	Points	Lost	Gained	Running Total	TA
1	1 (+1)				
2	18				
3	22				
4	20				
5	20				
6	19				
Total	100				

You may ask for clarification but you are ultimately responsible for the answer you write on the paper. If you make any assumptions state them.

Please look through the entire test before starting. WE MEAN IT!!!

NOTE: M = 10^6 K = 10^3 Mi = 2^{20} Ki = 2^{10}
Illegible answers are wrong answers.

Show your work in the space provided to get any credit for problem-oriented questions.

Good luck!

1. (1 point, 1 min) (don’t worry you get 1 point regardless of your answer! You get 1 extra credit point if you get it right!!)

Given:

```
   5+3+2 = 151022
   9+2+4 = 183652
   8+6+3 = 482466
   5+4+5 = 202541
```

What is

```
   7+2+5 = 143547
```

CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________

Q2

Cache
2. (18 points, 20 min)
a) (6 points)
Given:

C0	C1

- 8 total cache blocks
- 2-way set-associative organization
- Cache initially empty
- LRU replacement policy
- Memory blocks L, M, N and O all map to the **same** cache line.
- The processor performs a total of eight accesses, to memory blocks O, N, M, L, O, O, N, and M in that order. For each of these accesses, specify (by filling in the table below) whether it is a cache hit or a cache miss, type of miss (cold/capacity/conflict), and the memory block evicted (if any). **Note: capacity miss dominates over conflict; cold dominates over capacity.**

Memory Access	Hit/miss	Type of miss	Block evicted from cache
O	miss	Cold +0.5	-
N	miss	Cold +0.5	-
M	miss	Cold	O +0.5
L	miss	cold	N +0.5
O	Miss	Conflict +0.5	M +0.5
O	Hit +0.5	-	- +0.5
N	Miss	Conflict +0.5	L +0.5
M	Miss	Conflict +0.5	O +0.5

(Area for rough work)

b) (2 points) In a 4-way set associative cache with 64K bytes of data, 64 bytes per block and with a t-bit tag (circle the right choice)

[✓ (i) There are four t-bit tag comparators]

(ii) There are 64 t-bit tag comparators

(iii) There are 1K t-bit tag comparators

(iv) There is one t-bit tag comparator for the entire cache

c) (2 points) Virtually indexed and physical tagged cache is attractive because (circle the correct choice)

(i) it results in a better cache hit ratio for a given cache organization

(ii) it eliminates the memory aliasing problem with physically indexed physically tagged caches

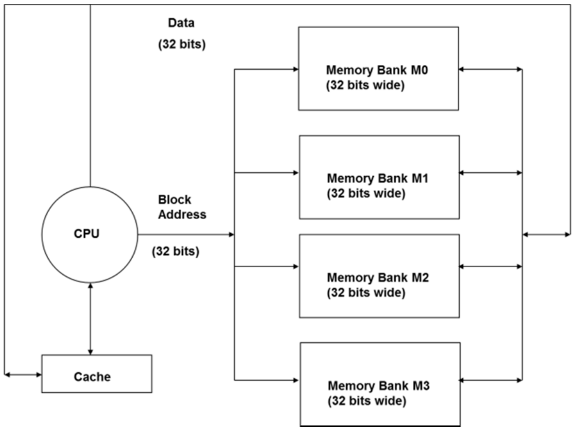
(iii) it enables building bigger first level caches than physically indexed and physically tagged caches

[✓ (iv) it removes the address translation through the TLB out of the critical path of the cache access]

d) (4 points) Consider a

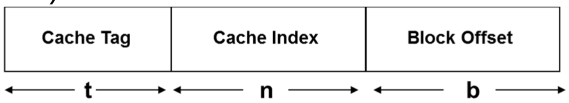
- 4-way interleaved memory system
- DRAM access time: 70 cycles
- bus cycle time for address or data transfer from/to the CPU/memory: 5 cycles
- Block size: 4 words (each 32 bits)

Compute the block transfer time.



Answer:
Address from CPU to memory = 5 cycles (all memory banks receive the address) **(+1)**
DRAM access time = 70 cycles (all four banks retrieve in parallel their respective words of the block) **(+1)**
Data transfer from memory to CPU = 4 * 5 cycles (memory banks take turns sending their respective word of the block to the CPU **(+1)**
Total time = 5 + 70 + 4*5 = 95 cycles **(+1)**

e) (4 points) Given



Let **a** be the number of bits in the memory address, **S** be the total size of the cache in bytes, and **B** the block size in bytes. Assuming a direct-mapped cache, answer the following:

(i) **$b = \log_2 B$** **(+1)** **(ii)** **$n = \log_2(S/B)$** **(+1.5)**

(iii) **$t = a - (b+n)$** **(+1.5)**

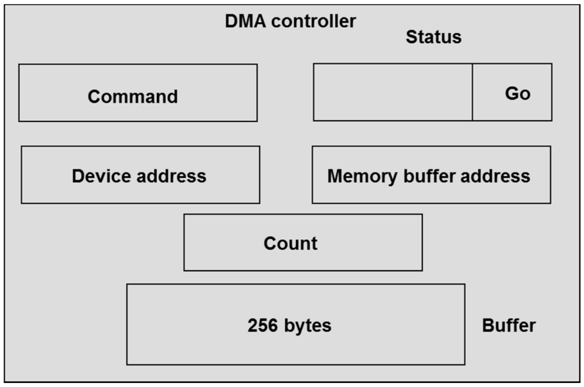
CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Q3

Name:_____Kishore_______GT Number: _______________

Input/Output and Disk

3. (22 points, 20 min)



a) (2 points) To transfer N bytes of data from a memory buffer at address M to the device at address D, the CPU executes the following instructions to the DMA controller:

1. Store N into the Count register.
2. Store M into the Memory buffer address register.
3. Set the Go bit in the Status register.
4. Store D into the Device address register.
5. Store “write to the device” command into the Command register.

What (if any) is wrong with the above code sequence?

Line 3 above; Setting the “Go” bit should be the last instruction.
(all or nothing)

b) (2 points) What is the role of the buffer in a DMA controller? (select ALL that apply: +1 for correct choice; -1 for incorrect choice)

(i) Asynchronously transfer data to/from the device

[✓ (ii) Synchronously transfer data to/from the device]

[✓ (iii) Asynchronously transfer data to/from the memory]

(iv) Synchronously transfer data to/from the memory

c) (2 points) Memory mapped I/O is a … (circle the right choice)

(i) technique for interfacing slow speed devices to the CPU

(ii) technique for interfacing high speed devices to the CPU

[✓ (iii) technique that allows the CPU to use Load/Store instructions to access the device registers]

(iv) technique that allows the CPU to quickly find the location of the handler code for a device

d) Given the following specification for a disk drive:

Average seek time	=	4 ms
Rotational speed	=	6000 RPM
Platters	=	3
Surface per platter	=	2
Tracks per surface	=	5000
Sectors per track	=	1024
Recording density	=	256 bytes per sector

(i) (2 point)How much time is needed to get to a random sector on the disk?

Time to get to a random track = average seek time = 4 ms (1) **(+0.5)**
Time to get to the random sector on the track
= average rotational latency = (1/6000)/2 min = 5 ms (2) **(+0.5)**
Time to get a random sector on the disk = (1) + (2) = 9 ms **(+1)**

(ii) (2 point) How much time is needed to read one random sector from the disk when the head is already positioned on the desired sector?

Time to read one sector
= rotational latency/number of sectors per track **(+1)**
= 10 ms / 1024 = 0.00977 ms **(+1)**

(iii) (2 points) If the disk gets a request to read 6 random sectors, how much total time will that request take to complete?

Time to read one random sector
= time to get to the sector + sector read time
= 9 ms + 0.00977 ms = 9.00977 ms **(+1)**

Time to read 6 random sectors = 6 * 9.00977 = 54.05862 ms **(+1)**

(iv) (2 points) If the disk gets a request to read 6 consecutive sectors, how much total time will that request take to complete?

Time to read first sector = 9.00977 ms **(+1)**
Time to read the next 5 consecutive sectors = 5 * .00977 = 0.04885 ms
Time to read 6 consecutive sectors
= 9.00977 + 0.04885 ms = 9.05862 ms **(+1)**

(v) (2 points) What is the transfer rate of the disk?

Amount of data in one track = 1024 * 256 bytes = 262,144 bytes **(+1)**
Data transfer rate = data in one track/rotational latency **(+1)**
= (262,144/10)*1000 bytes/sec = 26,214,400 bytes/sec

CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________

e) (4 points) Consider a disk using zoned-bit recording with the following parameters:

***p* – number of platters,**
***n* – number of surfaces per platter,**
z – number of zones,
t_{zi} – number of tracks at zone z_i,
s_{zi} – number of sectors per track at zone z_i,
b - number of bytes per sector

Give an expression for the total capacity of the disk.

$$\boxed{\text{Capacity} = (\underbrace{p * n}_{+1} * \underbrace{(\Sigma\, (t_{zi} * s_{zi}), \text{ for } 1 \le i \le z)}_{+2} \underbrace{* \, b}_{+1})\text{ bytes}}$$

f) (1 point) Which of the following disk scheduling algorithms are prone to starvation? (circle **all** the right choices)

(i) FCFS

(ii) SCAN

[✓ (iii) SSTF]

(iv) LOOK

(1 point) Explain why.

A track that is far away from the current head position may end up not getting service if subsequent requests are close to the current head position. (+1)

File Systems (see last page for a cheat sheet on Unix FS commands)

Q4

4. (20 points, 20 min)
a) (2 points)
The current state of a file "foo" is as follows:

```
   -r-xr-x--x   3 rama           0 Apr 27 21:01 foo
```

After executing the following commands:

```
   chmod u+w foo
   chmod g-x foo
```

The new access rights are (circle the right choice):

[✓ (i) -rwxr----x]

(ii) -rwxr-x---

(iii) -rwxrwx---

(iv) -rwxr---w-

CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________

b) (8 points) In the following table, assume **none of the files exist to start with** in the current directory. Fill in the table. The reference count in the table pertains to the i-node that is affected by the command in that row. If a new i-node is created, show the old reference count for that i-node as 0.

Command	New i-node created (yes/no)	Reference count	
		old	new
touch f1	Yes	0	1
touch f2	Yes	0	1
ln f2 f3	No	1	2
ln –s f1 f4	Yes	0	1
ln f4 f5	No	1	2
ln -s f4 f6	Yes	0	1
rm f6	No	1	0
rm f3	No	2	1

Use this area for rough work for this question

+1 for each correct row

– 0.5 for each incorrect entry in the table

c) (2 points) Hybrid indexed allocation results in (circle ONE choice that captures ALL the TRUE statements in the following list)

(i) External Fragmentation

(ii) Internal fragmentation

(iii) Ability to grow the file easily

(iv) Ability to handle small files efficiently

(v) {i and ii}

(vi) {ii and iii}

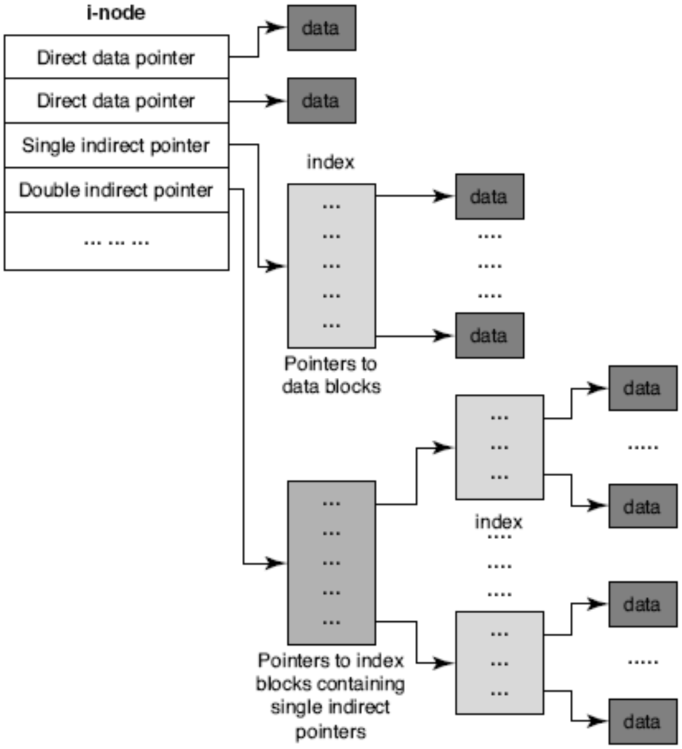
[✓ (vii) {ii, iii, and iv}]

(viii) None of the above

d) (**NOTE Mi = 2^{20}**)

Given the following:
Size of index block = 128 bytes
Size of Data block = 2048 bytes
Size of pointer = 8 bytes (to index or data blocks)
The i-node consists of
2 direct data block pointers,
1 single indirect pointer, and
1 double indirect pointer.

Note that the index blocks and data blocks are allocated on a need basis. An index block is used for the top-level i-node as well as for the index blocks that store pointers to other index blocks and data blocks (see Figure).



i. (2 points) How many pointers does each index block contain?

Number of pointers in each index block
= size of index block / size of pointer (+1)
= 128/8 = 16 pointers (+1)

ii. (2 points) How many data blocks are used to store a 40 KiB file?

Number of data blocks needed for 40 KiB file = 40 KiB / 2KiB = 20 (+2)

iii. (2 points) How many index blocks (including the i-node for the file) are needed to store a 256 KiB file?

Number of data blocks needed for a 256 KiB file = 128
To get 128 data blocks
1 i-node gives 2 direct data blocks
1 single indirect index block gives 16 data blocks
1 double indirect index block gives
7 single indirect index blocks gets us remaining 96 + 14 data blocks (+1)
Therefore totally, we need 10 index blocks for this file. (+1)

iv. (2 points) What is the largest file size that can be supported in this file system?

Max number of data blocks
= direct data blocks + data blocks via single indirect + data blocks with 2 levels of indirection = 2 + 16 + 16*16 = 274 (+1)

Max file size = 274 * 2 KiB = 548 KiB (+1)

CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________ Q5

Parallel Systems

5. (20 points, 20 mins)

a) (1 point) Assume that there are four threads in a process. Assume that the threads are scheduled on a uniprocessor. Once spawned, each thread prints its thread-id and terminates. Assuming a non-preemptive thread scheduler, how many different executions are possible? (circle the right choice)

(i) 4

(ii) 1

(iii) 16

[✓ (iv) 4!]

(1 point) Explain why.

The threads are independent of one another. Therefore, the operating system may schedule them in any order. The number of possible executions of the 4 threads is 4!.

b) (1 points) Given below are two threads and their execution history on an SMP. Assume that the execution of each instruction is atomic. Assume that **Mem[x] = 0** initially.

Thread 1 (T1)	**Thread 2 (T2)**
Time 0: R1 <- Mem[x]	**Time 1: R2 <- Mem[x]**
Time 2: R1 <- R1+2	**Time 3: R2 <- R2+1**
Time 4: Mem[x] <- R1	**Time 5: Mem[x] <- R2**

what is the final value in memory location **x**?

(i) 0

[✓ (ii) 1]

(iii) 2

(iv) 3

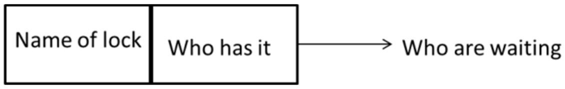
(2 point) Explain why.

Each of the threads T1 and T2 load a memory location, add a value to it, and write it back. Given the timeline, both T1 and T2 get the old value of memory location x and do processing shown in the timeline. Therefore, when both threads complete execution, x will contain the value written to it by the last store operation. Since T2 is the last to complete its store operation, the final value in x is 1. (+1) (+1)

CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________

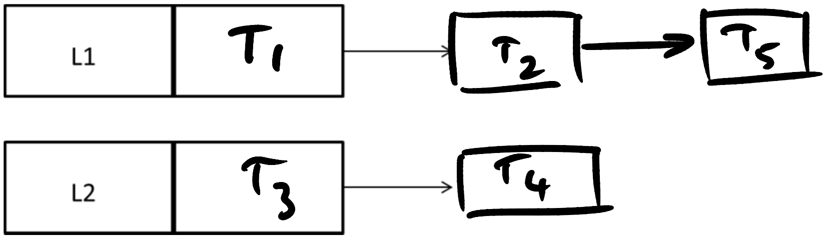
c) (5 points) The internal representation in the thread library for a LOCK is shown below:



Assume that the following events happen in the order shown (T1–T5 are threads of the same process):

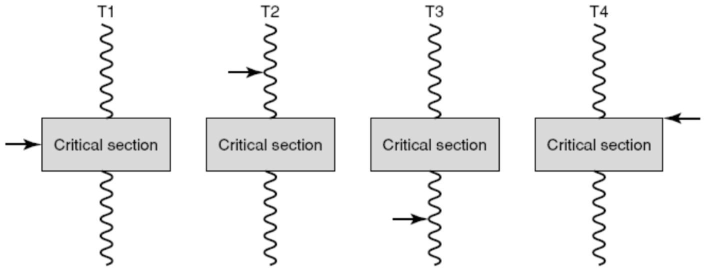
- T1 executes thread_mutex_lock(L1);
- T2 executes thread_mutex_lock(L1);
- T3 executes thread_mutex_lock(L2);
- T4 executes thread_mutex_lock(L2);
- T5 executes thread_mutex_lock(L1);

Assuming that there have been no other calls to the threads library prior to this, show the state of the internal data structures of L1 and L2 after the above five calls (by filling in the figures below).



+1 for each entry

d) (4 points)
Shown in the figure below are the points of execution (indicated by the arrows) of four threads (T1-T4) of the same process. Assume that the critical sections are mutually exclusive (i.e., they are governed by the same lock).



For each thread, state if it is **active** or **blocked**, and why.
(+0.5 for each correct entry below)

T1 is **active** because **it has lock and executing inside critical section**

T2 is **active** because **it is executing outside the critical section**

T3 is **active** because **it is executing outside the critical section**

T4 is **blocked** because **it is waiting for lock to enter critical section**

e) (2 points) Deadlock (circle the right choice)

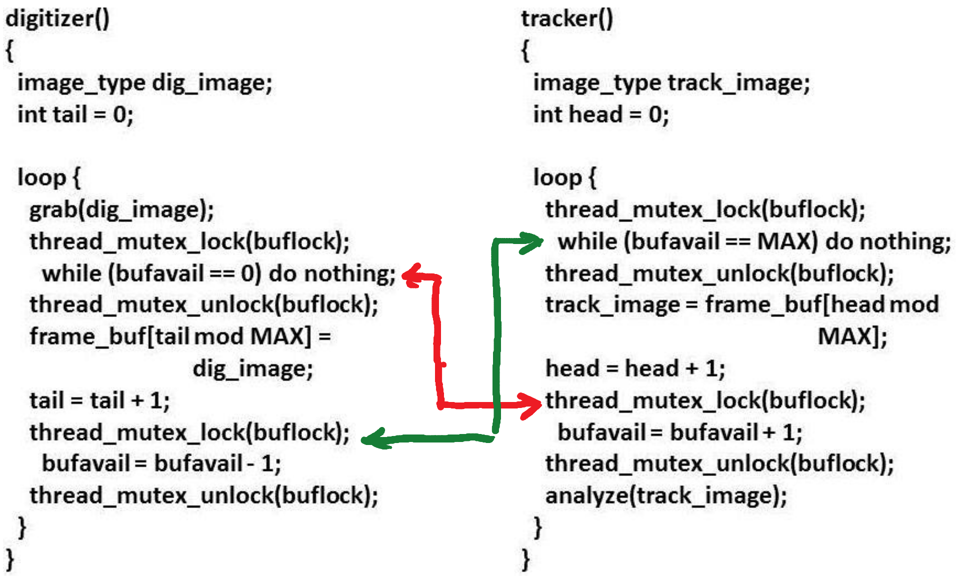
(i) Is a condition where threads are not using mutex locks

(ii) Is a condition where all the locks variables are in use

(iii) A lock variable that is dead

[✓ (iv) Is a condition where one or more threads are waiting for an event that will never happen]

f) (4 points) In the following code, buflock, bufavail,and frame_buf are shared variables. Each of the functions (digitizer and tracker) are executed by two distinct threads. What is the problem (if any) with the following code?



```
digitizer()                                   tracker()
{                                             {
 image_type dig_image;                         image_type track_image;
 int tail = 0;                                 int head = 0;

 loop {                                        loop {
  grab(dig_image);                              thread_mutex_lock(buflock);
  thread_mutex_lock(buflock);                    while (bufavail == MAX) do nothing;
   while (bufavail == 0) do nothing;            thread_mutex_unlock(buflock);
  thread_mutex_unlock(buflock);                 track_image = frame_buf[head mod
  frame_buf[tail mod MAX] =                                               MAX];
                 dig_image;                     head = head + 1;
  tail = tail + 1;                              thread_mutex_lock(buflock);
  thread_mutex_lock(buflock);                    bufavail = bufavail + 1;
   bufavail = bufavail - 1;                     thread_mutex_unlock(buflock);
  thread_mutex_unlock(buflock);                 analyze(track_image);
 }                                             }
}                                             }
```

Digitizer thread when it wants to decrement bufavail, may not be able to get the lock if the tracker is in the while loop; (+2)

Tracker thread when it wants to increment bufavail, may not be able to get the lock if the digitizer is in the while loop; (+2)

(Just saying deadlock/livelock without explanation gets 2 points MAX)

CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________ Q6

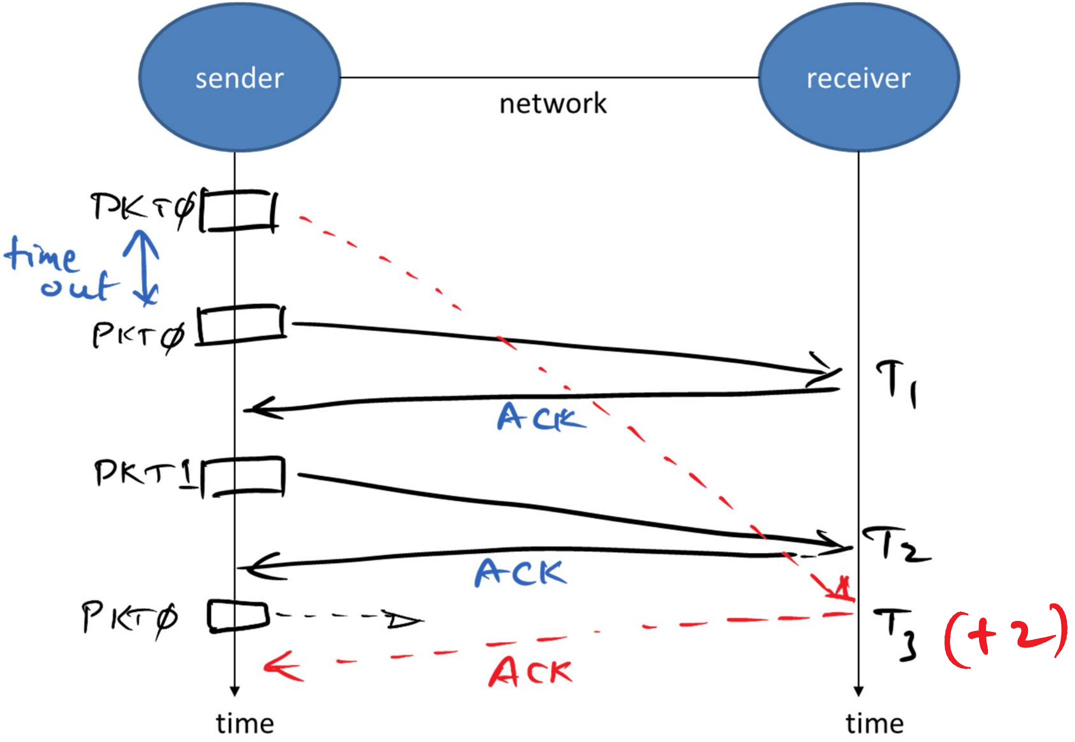
Networking

6. (19 points, 20 min)

a) (3 points) The one-bit sequence number requirement for the "Stop-and-wait protocol (aka alternating bit protocol)" is predicated on the following assumption:

- packets do not get reordered in the network OR arbitrarily delayed

Show using the space-time diagram below what can go wrong if this assumption does not hold.



- **Sender sends PKT0**
- **Sender times out; retransmits PKT0; Receiver ACKS PKT0**
- **Receiver is now awaiting sequence number 1**
- **Sender sends PKT1; Receiver ACKS PKT1**
- **Receiver is now awaiting sequence number 0**
- **The original PKT0 arrives LATE (time T3) at receiver**
- **Receiver accepts it as a new packet and ACKs it**
- **Obviously this is in error**

(+1 grace point) (+2 for showing anomaly like above)

b) (2 points) Why you need MAC address when each node on the Internet has an IP address?

IP address is a virtual address used by the network layer of the protocol (+1) **stack; MAC address is the real address used by the NIC (which is at the link layer) to receive a packet on the wire intended for this node.** (+1)

CS 2200 Fall 2014 Final Exam 8 AM to 10 AM

Name:_____Kishore_______GT Number: _______________

c) (2 points) The sequence number in a packet … (circle the right choice)

(i) Gives the destination address

[✓ (ii) Is needed for message reconstruction at the destination]

(iii) Assures the integrity of the packet

(iv) Is computed using cyclic redundancy check (CRC) algorithm

(v) Is the same for every packet in a given message

d) (2 points) In the sliding window protocol,

(i) when is the window size decreased?

When the transport layer notices congestion on the network due to frequent retransmissions of packets

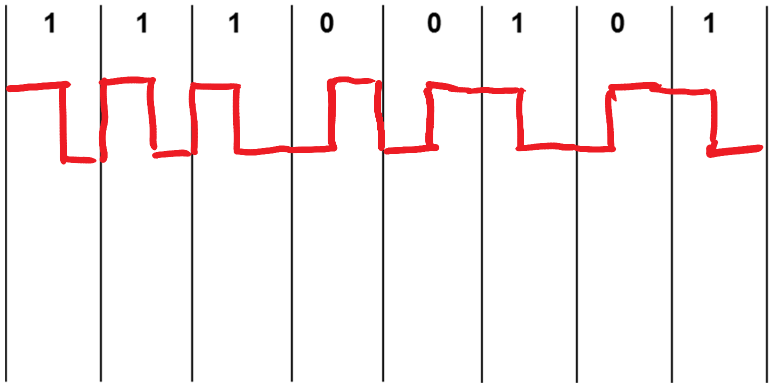
+1

(ii) when is the window size increased?

When the transport layer notices that packets are getting through without much retransmission of packets (an indication that there is no congestion)

+1

e) (4 points) Show the wave form for the following packet with Manchester encoding (the space between the vertical lines represent time per bit):



(+0.5 for each bit)

f) (2 points) Given:

- Message size = 100,000 bytes;
- Header size per packet = 100 bytes
- Packet size = 1100 bytes
- Packet loss = 10%

The number of packets to transmit the message (ignoring fractional packet loss): (circle the correct choice)

(i) 100

(ii) 91

(iii) 1000

[✓ (iv) 111]

(v) 110

(vi) Cannot be determined with the data given

g) (2 points) A packet header consists of the following fields:
destination_address
source_address
num_packets
sequence_number
packet_size
checksum

Assume that each of these fields occupies 4 bytes. Given that the packet size is 1500 bytes, compute the payload in each packet.

Size of the header = 6*4 = 24 bytes **(+1)**
Payload in each packet = packet size – header size
= 1500 – 24 = 1476 bytes **(+1)**

h) (2 points)
Given the following:
Message size = 10,000 bits
Bandwidth on the wire = 100,000 bits/sec
Time of flight = 5 msecs
Sender overhead = 2 ms
Receiver overhead = 1 ms

Compute the throughput.

Transmission delay = Tw = (10000/100000) * 1000 ms = 100 ms **(+1)**
Time for message transmission = S + Tw + Tf + R
= 2 + 100 + 5 + 1 = 108 ms **(+0.5)**
Throughput = message size/time for transmission
= (10000/108) * 1000 bits/sec
= 93 Kbits/sec **(+0.5)**