1. (11 pts) For each question, state which answer is the most apropriate. The first one is done for you.

Questions:

u What is this section of the test?

___ What is a Page Table Cache?

___ What is a flip-flop?

___ Which devices have to worry about head crashes?

___ A multiprogrammed operating system has what goal?

___ What structure uses a dirty bit?

___ What is synchronous timing?

___ What does Virtual Memory do?

___ What is a base page table register?

___ What is a parity bit?

___ What do all caches need?

___ What is the goal of the memory heirarchy?

Answers:

a) Memory that retains its contents when the power is turned off.
b) A small fast memory holding recently accessed data and/or instructions.
c) Spatial and Temporal locality.
d) A technique used in CD-ROM Drives to increase storage density.
e) A setup that does not require a clock.
f) A structure that holds recent mappings of virtual to physical addresses.
g) A pointer to the beginning of a page table.
h) The ability of an I/O device to read from and write to memory without processor assistance
i) Hard disk drives.
j) A setup that requires the use of a clock.
k) Memory that loses its values when the power is turned off.
l) A circuit that exhibits purely sequential behavior.
n) A binary digit appended to a group of binary digits to make the sum of all the digits an even number.
o) A write-back cache.
p) To maximize the efficient use of an expensive resource (the CPU).
q) no-write-allocate
r) To make memory perform like it is built of fast memory but cost like it is built out of cheap memory.
s) Replacement policies.
t) Maps from one space to another.
u) Gradeable. :-)

2. (7 pts) Circle the right answer.

The contents of Volatile memory (remain / disappear) after the power is removed.

Dynamic RAM is (hotter / cooler) than Static RAM.

Write-back caches use (write-allocate / no-write-allocate) policies.

Segment tables (do / do not) contain protection information.

All caches (are / are not) actually Set Associative caches.

A flip-flop (is / is not) the same as a gated latch.

Segmented paged Virtual Memory is an OS feature only, and doesn't require hardware support. (True / False)

3. (6 pts) Here is a 12-bit Error Correction code format (same one used in class):

$$d_8 \;\; d_7 \;\; d_6 \;\; d_5 \;\; C_4 \;\; d_4 \;\; d_3 \;\; d_2 \;\; C_3 \;\; d_1 \;\; C_2 \;\; C_1$$

a. Given the *data* bit pattern

1 1 0 1 1 1 0 1

in a machine using the above ECC code, what bit pattern gets sent to memory? (No credit will be given without work being shown.)

4. (6 pts) In this same machine, the following bit pattern is retrieved from memory:

0 1 0 0 1 0 1 0 1 0 1 0

Assuming the above Error Correction code format, identify and correct any errors that may have occurred during transmission or storage. (No credit will be given without work being shown.)

5. (5) There is one principle that programs exhibit that makes caches feasible. What is it? Give the two types of this principle, and explain what each exploit.

6. (4) Caches can be either Virtually or Physically Addressed. Explain the difference, and give one advantage and one disadvantage to using Physically addressed caches.

7. (3) Page tables can be extremely large. Describe one technique we discussed in class that allows only a subset of the page table to be permanently resident in memory. (Using pictures here is a good idea.)

8. (3) What is the Worst Case Path? Why does it matter? (Why is it important?)

9. (7) If a byte-addressable machine generates 21-bit logical addresses and has 128Kbytes of physical memory,

    a. How big is the physical address space?

    b. How big is the virtual address space?

    If a page size is 8K-bytes:

    c. How many page frames are there?

    d. How many pages?

    e. How many bits wide is the page table?

    If the page size is 4K-bytes,

    f. How many page frames are there?

    g. How many pages?

    h. How many bits wide is the page table?

    If the memory is expanded to 1Megabyte, and pages are 2K bytes long,

    i. How many frames are there?

    j. How many pages?

    k. How many bits wide is the page table?

10. (3pts) Assume a task is divided into 8 equal-sized segments, and page tables have 4 entries. Thus, the system has a combination of segmentation and paging. Assume also that the page size is 2K bytes.

    a. What is the maximum size of each segment?

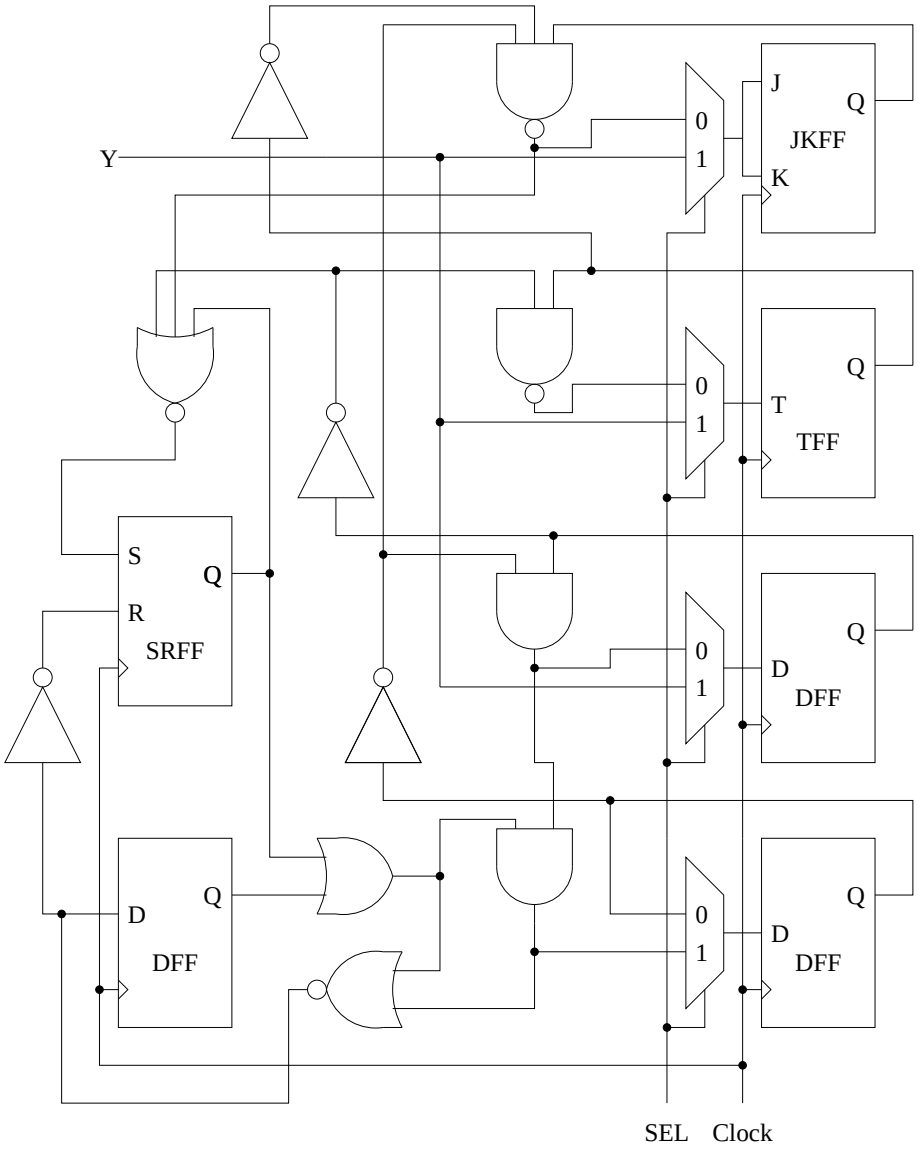
    b. What is the maximum logical address space for the task?

    c. Show how an address is partitioned (what bits are what).

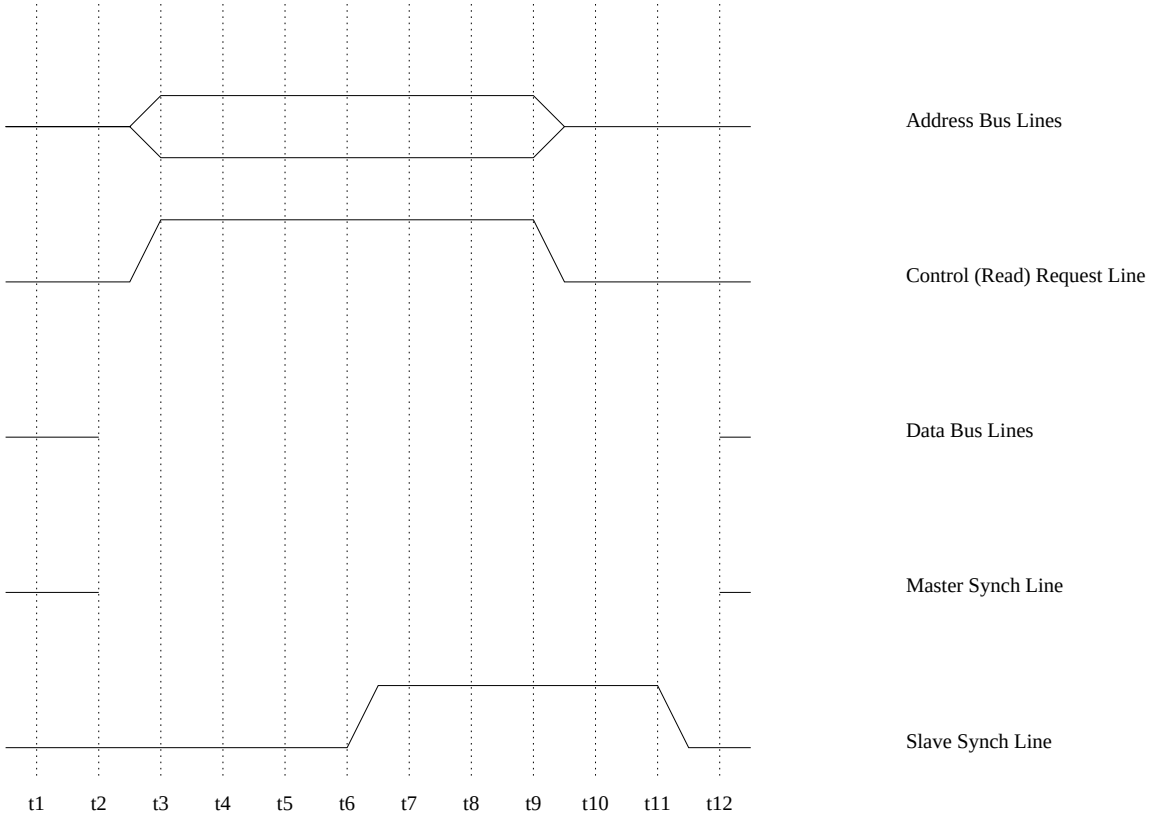
11. (13 pts) Assuming rising edge-triggered flipflops, what is the maximum clock frequency possible for the following circuit? (In other words, what is the maximum clock frequency that will still guarantee correct behavior?) Use the following delay values, and assume all input signals become valid at time 0. (Tprop is the propagation time for the flipflop, the time it takes from the rising edge of the clock until the output of the FF is valid.)

AND: 4ns OR: 3ns NAND: 6ns NOR: 5ns NOT: 1ns MUX: 4ns
Tprop (TFF): 11ns Tsetup (TFF): 3ns Thold (TFF): 1ns
Tprop (DFF): 7ns Tsetup (DFF): 4ns Thold (DFF): 1ns
Tprop (JKFF): 10ns Tsetup (JKFF): 3ns Thold (JKFF): 1ns
Tprop (SRFF): 8ns Tsetup (SRFF): 4ns Thold (SRFF): 1ns

Note: You must show the path in order to get credit.

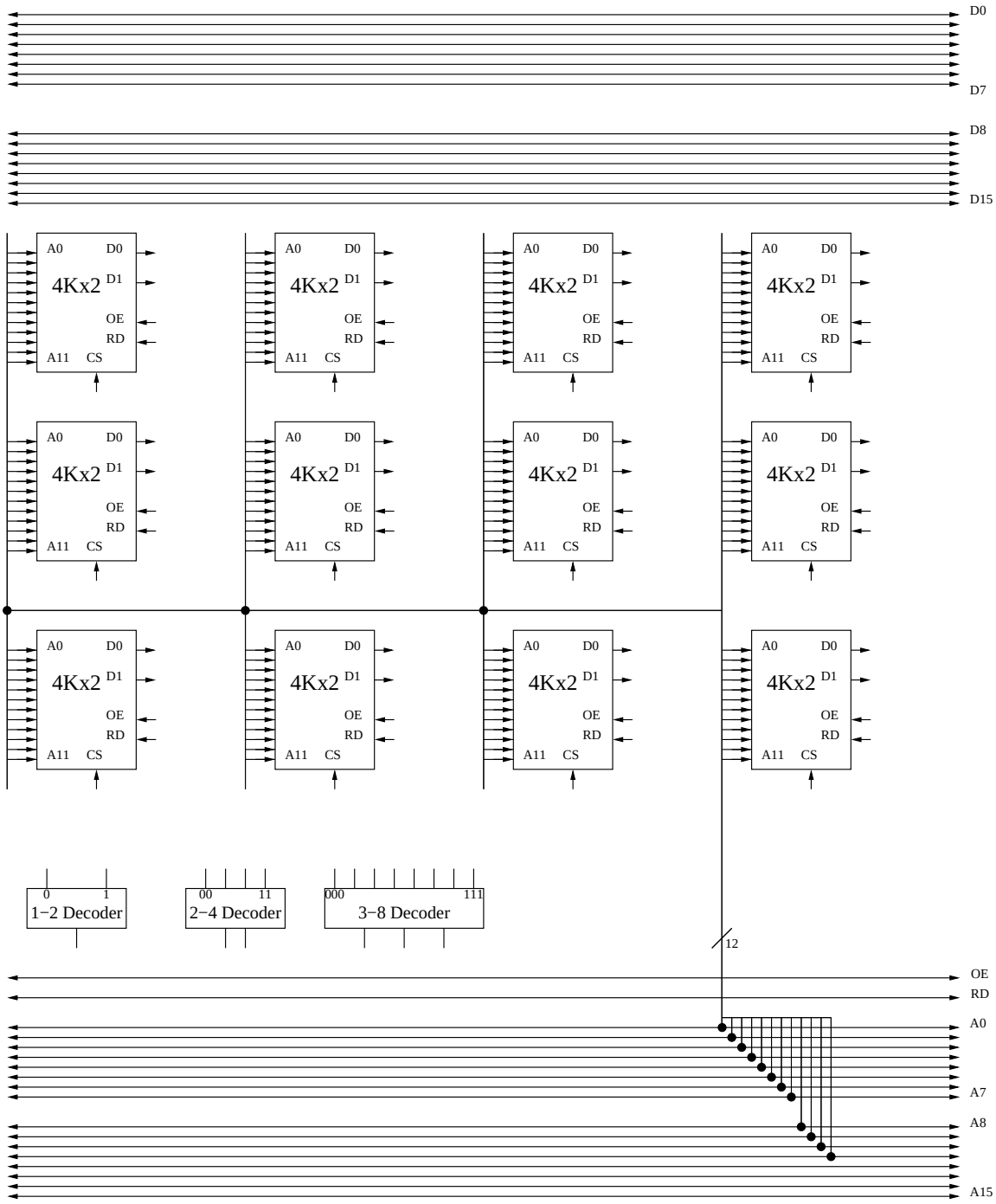


12. (9 pts) A Master I/O device wishes to do a **read** from a slave device. In the figure below, we see that at time *t3* the Master asserts the Address Bus lines and the Read Request line. We also see that at time *t9* the Master deasserts those lines. You are to draw in the Data Bus Lines and the Master Synch line, and add arrows to indicate which transition causes which. Also, explain in words what is happening during the handshaking.



13. (8 pts) Add the connections to the following diagram necessary to create a 8Kx8 memory. Not all of the hardware shown is required to perform this task.

CS - Chip Select
OE - Output Enable
RD - Read (Read/Write, technically)



14. (13) Assume a byte-addressable computer with a 32-bit word size and 256 bytes of memory. In this machine accessing main memory takes 10 clock cycles (in addition to the time necessary to do a cache lookup), and the bus between main memory and the processor is 16-bits wide. This machine also has a 64-byte physically addressed Direct-Mapped cache with a line size of 2 words and an access time of 1 cycle. Given the following address reference sequence (in Hex):

0xB5,0xB7,0x37,0x38,0x39

a) Write down how you are partitioning each address (which bits are the Tag, offset, etc.)

b) In the table below, fill in the proposed Cache's Tag values after each memory reference has been processed. If it is a hit, mark the entry number to indicate this, and if it is a miss enter what the new tag should be. (X indicates the entry is invalid). There may be more Tag Array entries than you need.

Tag Array	Contents of Tag Array after processing address (Time ->)					
Entry Number	Initial Contents	0xB5 (10110101)	0xB7 (10110111)	0x37 (00110111)	0x38 (00111000)	0x39 (00111001)
0	X					
1	X					
2	X					
3	X					
4	X					
5	X					
6	X					
7	X					
8	X					
9	X					
10	X					
11	X					
12	X					
13	X					
14	X					
15	X					

What set of memory addresses are sent to memory on the first miss?

15. (13) What if a 48-byte 3-way Set Associative Cache (instead of the Direct-mapped Cache) with a line size of 1 word is used instead? Remember, this is a byte-addressable machine with a 32-bit word size, a 16-bit bus between processor and memory, and a Main Memory access time of 10 cycles (in addition to the time necessary to to a cache lookup). The Cache access time is still 1 cycle. Given the same address reference sequence (in Hex) as before:

0xB5,0xB7,0x37,0x38,0x39

a) Write down how you are partitioning each address (which bits are the Tag, offset, etc.)

b) In the table below, fill in the proposed Cache's Tag values after each memory reference has been processed. If it is a hit, put an "H" in the tag field, and if it is a miss write down what the new tag should be. Use an LRU replacement scheme, and after each address is processed be sure to indicate the age of the references. There may be more entries than you need. MRU = Most Recently Used, LRU = Least Recently Used.

Tag Array				Contents of Tag Array after processing address (Time ->)									
Set #	Entry #	Initial contents		0xB5 (10110101)		0xB7 (10110111)		0x37 (00110111)		0x38 (00111000)		0x39 (00111001)	
		Age	Tag	Age	Tag	Age	Tag	Age	Tag	Age	Tag	Age	Tag
0	0	MRU	0 0 1 1										
	1	LRU	1 1 1 0										
	2		1 0 0 0										
1	0		0 1 0 0										
	1	MRU	0 0 1 1										
	2	LRU	1 1 0 0										
2	0	LRU	1 1 0 0										
	1		0 1 0 1										
	2	MRU	0 1 1 0										
3	0	LRU	0 0 1 0										
	1		0 1 1 1										
	2	MRU	0 1 1 0										

16. (8 pts) The following tables contain some of the information about a segmented, paged virtual memory system and certain select memory locations. Total physical memory size is 32K bytes, and the page size is 512 bytes. All numbers in this table are in decimal unless otherwise noted. (**Note:** The maximum number of entries in the page tables is significant, but the number of entries in the Segment table is not.)

Segment Table

Entry Number	Presence Bit	Page Table
0	1	5
1	1	2
2	1	0
3	0	7
7	1	5
12	1	3
13	1	1
15	1	4

Page Table 0

Entry Number	Present? (1=Yes)	Disk Addr	Frame Number
0	1	1234123	0x4
2	0	0893748	0x7
4	1	2489567	0x1
8	1	9623873	0x17
16	1	B0F6BD3	0x23
25	0	32829AA	0xA
29	1	56D87AC	0xC
31	1	10A876D	0x6

Page Table 2

Entry Number	Present? (1=Yes)	Disk Addr	Frame Number
0	1	1234123	0xF
1	0	0893748	0x11
2	1	2489567	0x14
3	1	9623873	0x27
4	1	BC56BD3	0x29
6	0	832759E	0x15
10	1	46B37AC	0x24
31	1	810476D	0x16

Memory

Address	Contents
0x00A4	0x76
0x01A4	0x73
0x02A4	0x32
0x03A4	0x46
0x04A4	0x30
0x06A4	0xa9
0x0AA4	0x05
0x31A4	0x74
0x62A4	0x29

Page Table 5

Entry Number	Present? (1=Yes)	Disk Addr	Frame Number
1	1	1234123	0xD
3	0	0893748	0x13
9	0	2489567	0x19
15	1	9623873	0x20
18	1	AE76BD3	0x18
22	0	328759A	0xE
25	1	11D87BE	0x12
31	1	91C875D	0x0

Page Table 7

Entry Number	Present? (1=Yes)	Disk Addr	Frame Number
0	1	1234123	0x5
1	0	0893748	0x26
2	1	2489567	0x21
3	1	9623873	0x2
4	1	AE76BD3	0x1A
5	1	328759A	0x10
6	1	56D87AC	0x3
7	1	10A876D	0x8

For each of the following convert the virtual address into a physical address (if possible) and write down the value of the memory location corresponding to the address. If it is not possible to do so, explain why.

0x3EA4 (**0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0** in binary).

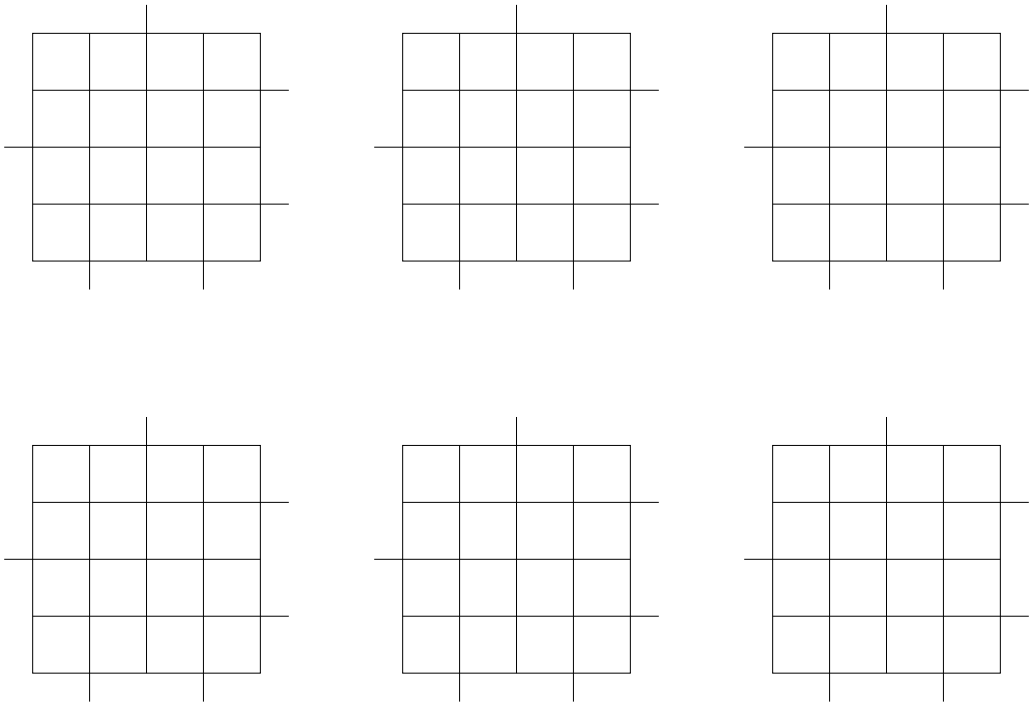
0x4CA4 (**0 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0** in binary).

0x89A4 (**1 0 0 0 1 0 0 1 1 0 1 0 0 1 0 0** in binary).

0xEFA4 (**1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0** in binary).

17. (16) Given the following table, draw the Karnaugh maps for Y1', Y2', and Y3' and Z in terms of X, Y1, Y2 and Y3, and then write **minimum** boolean equations for each. Do not worry if the state transition table takes you to impossible states, or gets stuck in a single state - fixing that is somebody else's problem. :-)

Present State (Y1 Y2 Y3)	Next State X=0 (Y1' Y2' Y3')	Next State X=1 (Y1' Y2' Y3')	Output X=0	Output X=1
001	011	010	0	0
011	010	011	0	0
100	011	010	0	1
101	101	100	0	0
110	000	001	0	1
111	100	101	0	0



18. (15 pts) Given the following Karnaugh maps, implement the sequential machine using an SR FF for Y1, a JK FF for Y2, and a T FF for Y3. You do not need to draw the gates, but you do need to write down the **minimized** input equations for each of the inputs of each of the Flip Flops in the circuit.

Y1'

Y2 \ Y3X	00	01	11	10
00		1		d
01	1	1		1
11	1	1		1
10				d

Y1

Y2'

Y2 \ Y3X	00	01	11	10
00	1	1		d
01	1	1	d	
11	1	1	1	
10		1	1	

Y1

Y3'

Y2 \ Y3X	00	01	11	10
00	1	1		d
01	d	1	1	1
11	1	1	1	1
10	1	1		d

Y1