1. (12 pts) For each question, state which answer is the most apropriate. The first one is done for you.

Questions:

1z. What is this section of the test? **u**

1a. What is a flip-flop?

1b. What is DMA?

1c. Which devices have to worry about head crashes?

1d. A multiprogrammed operating system has what goal?

1e. What is an interrupt?

1f. What is a dynamic RAM?

1g. Constant Linear Velocity is what?

1h. What is synchronous timing?

1i. What is Non-Volatile Memory?

1j. What is a Page Table Cache?

1k. What is a parity bit?

1l. What is a cache?

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) Life, the universe, and everything.
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) An unscheduled subroutine call.
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) Because it uses a blue laser.
p) To maximize the efficient use of an expensive resource (the CPU).
q) An integral part of an adder circuit.
r) An atomic unit.
s) A type of memory that uses capacitors to store data.
t) A technique used in disk drives to reduce seek time.
u) Gradeable. :-)

2. (3) In the ALU you designed in the homework, what bits were set or cleared in order to indicate that the values were to be added instead of subtracted? Why did this work so well?

3. (4) What is Cache Coherence? Is it a concern only in parallel computer systems? Why or why not?

4. (5) What is the goal of the memory heirarchy? What principle makes it possible to achieve this goal? Give the two types, and explain what they are.

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

6. (4) 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.)

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

8. (4) Given a logical 22-bit address and a 256Kbyte physical memory for a byte-addressable machine,

How big is the physical address space?

How big is the virtual address space?

Assuming 8K-byte pages, how many page frames are there? How many pages? How many bits wide is the page table?

Assuming 2K-byte pages, how many page frames are there? How many pages? How many bits wide is the page table?

9. (5 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

0 0 1 0 1 0 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.)

10. (5 pts) In this same machine, the following bit pattern is retrieved from memory:

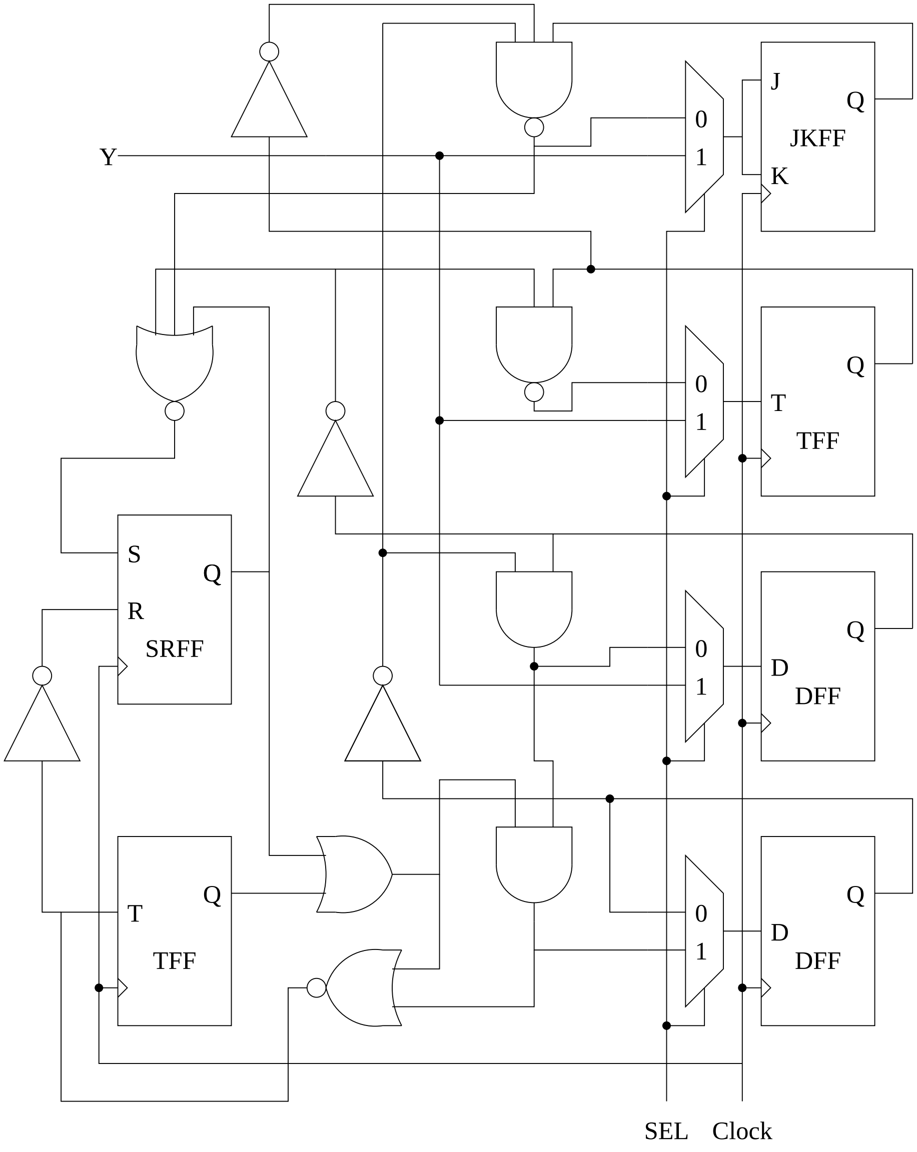
0 1 0 0 1 0 1 0 1 0 1 0

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.)

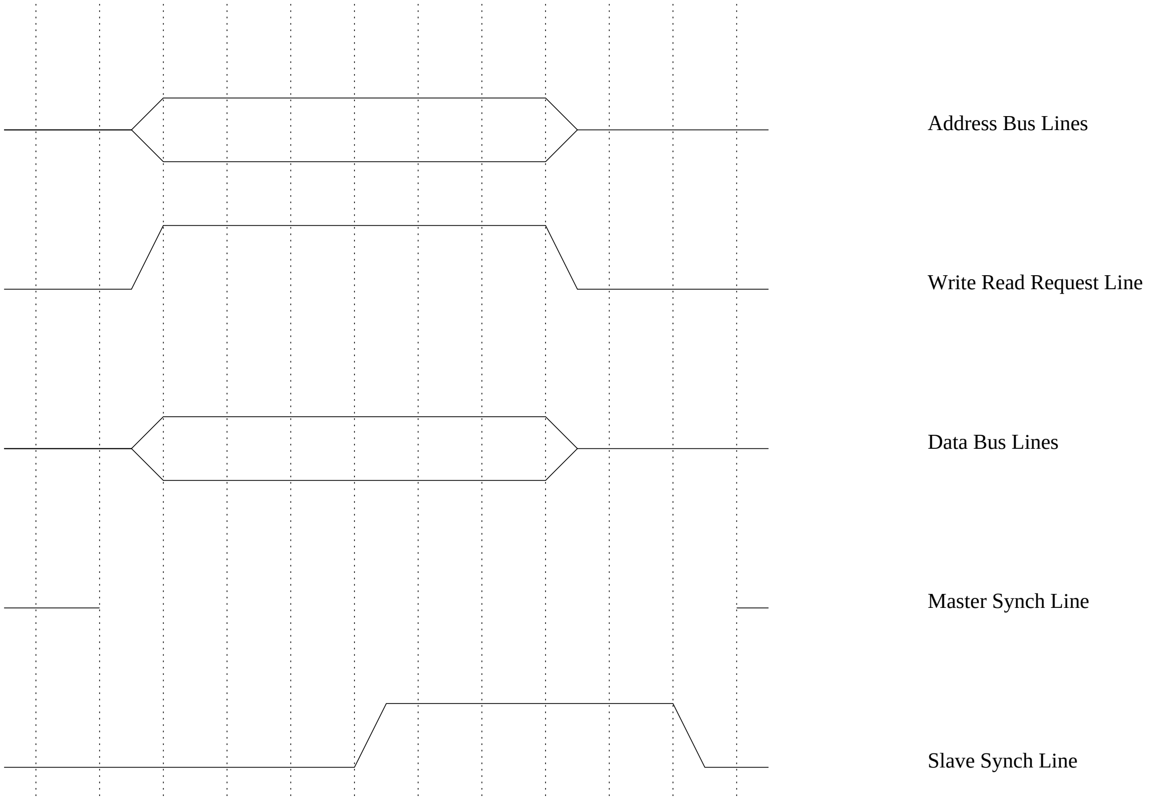
11. (12 pts) Assuming rising edge-triggered flipflops, what is the maximum clock frequency possible for the following circuit? (In other words, what is the maximum clock frequency that will still guarantee correct behavior?) Use the following delay values, and assume all input signals become valid at 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: 2ns NAND: 6ns NOR: 5ns NOT: 1ns MUX: 5ns
Tprop (TFF): 10ns Tsetup (TFF): 4ns Thold (TFF): 1ns
Tprop (DFF): 7ns Tsetup (DFF): 3ns Thold (DFF): 1ns
Tprop (JKFF): 8ns Tsetup (JKFF): 4ns Thold (JKFF): 1ns
Tprop (SRFF): 6ns Tsetup (SRFF): 2ns Thold (SRFF): 1ns

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

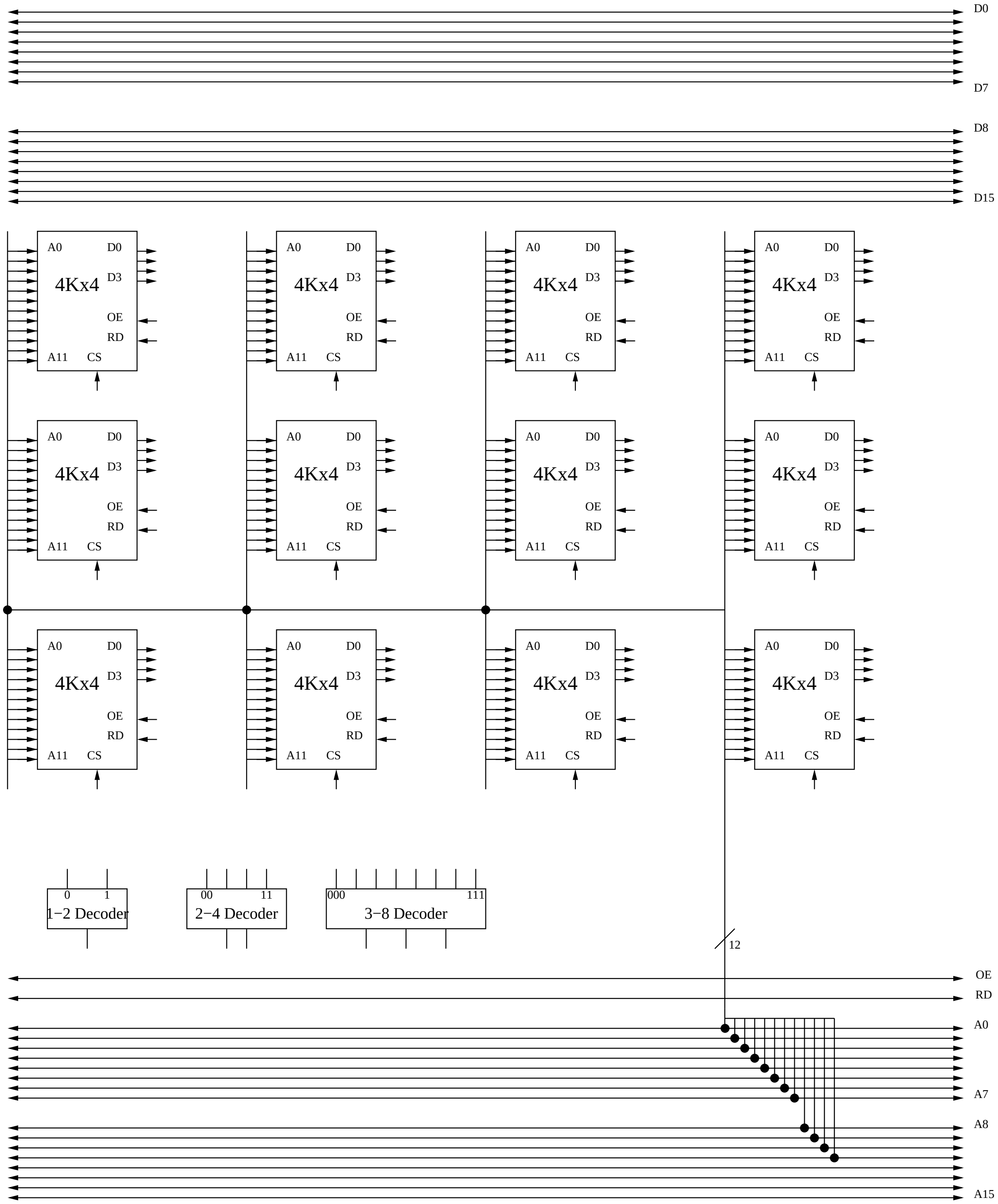


12. (9 pts) A Master I/O device wishes to do a write to a slave device. In the figure below, we see the timing for the Master asserting and releasing the Address Bus lines, the Data Bus lines, and the Write Request line. You are to draw in the Master Synch line, arrows indicating which transition causes which, and then explain in words what is happening during the handshaking.



13. (9 pts) Add the connections to the following diagram necessary to create a 8Kx16 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. (15) Assume a byte-addressable computer with a 32-bit word size and 256 bytes of memory. In this machine accessing main memory takes 4 clock cycles (in addition to the time necessary to do a cache lookup), and the bus between main memory and the processor is 8-bits wide. This machine also has a 128-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):

0xB5,0x35,0x37,0xB7,0x38

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)	0x35 (00110101)	0x37 (00110111)	0xB7 (10110111)	0x38 (00111000)
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?

What is the Average Memory access time for this sequence of references?

15. (15) What if a 24-byte 3-way Set Associative Cache (instead of the Direct-mapped Cache) with a line size of 2 words is used instead? Remember, this is a byte-addressable machine with a 32-bit word size, an 8-bit bus between processor and memory, and a Main Memory access time of 4 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:

0xB5,0x35,0x37,0xB7,0x38

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)		0x35 (00110101)		0x37 (00110111)		0xB7 (10110111)		0x38 (00111000)	
		Age	Tag	Age	Tag	Age	Tag	Age	Tag	Age	Tag	Age	Tag
0	0	MRU	0 0 0 1 1										
	1	LRU	0 1 1 1 0										
	2		0 1 0 0 0										
1	0		0 0 1 0 0										
	1	MRU	0 0 0 0 1										
	2	LRU	1 1 1 0 0										
2	0	LRU	1 0 1 0 0										
	1		0 1 0 0 1										
	2	MRU	1 0 1 1 0										
3	0	LRU	0 0 0 1 0										
	1		0 0 1 1 1										
	2	MRU	0 0 1 1 0										

What is the Average Memory access time for this sequence of references?

16. (9 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
8	1	2
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	0x57
16	1	B0F6BD3	0x23
25	0	32829AA	0x1
29	1	56D87AC	0x0
31	1	10A876D	0x6

Page Table 2

Entry Number	Present? (1=Yes)	Disk Addr	Frame Number
0	1	1234123	0x1
1	0	0893748	0x3
2	1	2489567	0x5
3	1	9623873	0x7
4	1	BC56BD3	0x9
6	0	832759E	0x2
11	1	46B37AC	0x4
31	1	810476D	0x6

Memory

Address	Contents
0x00A4	0x76
0x01A4	0x73
0x02A4	0x32
0x03A4	0x46
0x04A4	0x30
0x62A4	0x29
0x06A4	0xa9
0x09A4	0x74
0x0AA4	0x05

Page Table 5

Entry Number	Present? (1=Yes)	Disk Addr	Frame Number
1	1	1234123	0x5
3	0	0893748	0x3
9	0	2489567	0x4
15	1	9623873	0x31
18	1	AE76BD3	0x6
22	0	328759A	0x7
25	1	11D87BE	0x2
31	1	91C875D	0x0

Page Table 7

Entry Number	Present? (1=Yes)	Disk Addr	Frame Number
0	1	1234123	0x5
1	0	0893748	0x6
2	1	2489567	0x1
3	1	9623873	0x2
4	1	AE76BD3	0x4
5	1	328759A	0x0
6	1	56D87AC	0x3
7	1	10A876D	0x6

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.

0xBAA4 (**1 0 1 1 1 0 1 0 1 0 1 0 0 1 0 0** in binary).

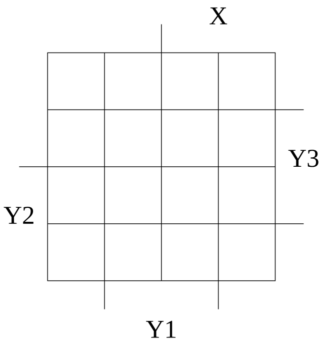
0x4CA4 (**0 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0** in binary).

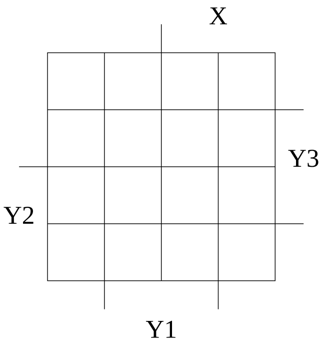
0xD4A4 (**1 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0** in binary).

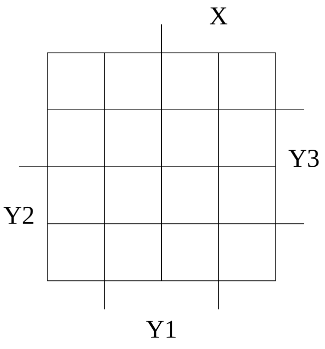
0x1EA4 (**0 0 0 1 1 1 1 0 1 0 1 0 0 1 0 0** in binary).

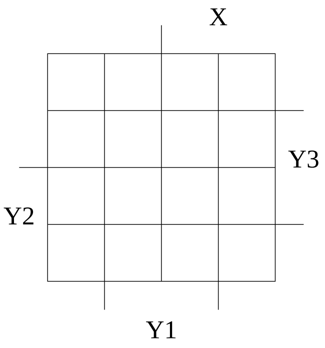
17. (16) Given the following table, draw the Karnaugh maps for Y1’, Y2’, and Y3’ and Z in terms of X, Y1, Y2 and Y3, and then write **minimum** boolean equations for each. Do not worry if the state transition table takes you to impossible states - fixing that is somebody else’s problem. :-)

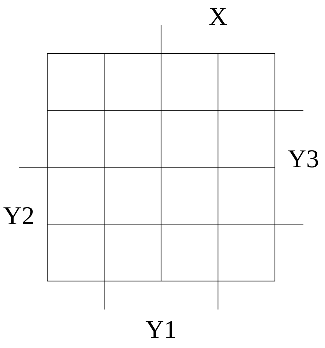
Present	Next State		Output	
State	X=0	X=1	X=0	X=1
(Y1 Y2 Y3)	(Y1’ Y2’ Y3’)	(Y1’ Y2’ Y3’)		
001	100	100	0	0
010	110	001	0	1
011	110	010	0	1
100	001	001	0	0
101	001	001	0	0
110	010	001	0	1
111	010	010	0	1

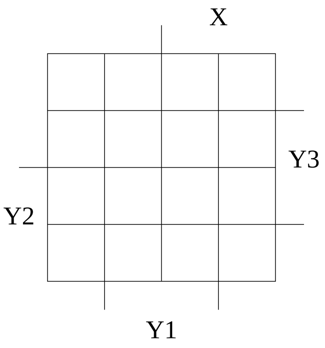












16. (15 pts) Given the following Karnaugh maps, implement the sequential machine using an SR FF for Y1, a JK FF for Y2, and a T FF for Y3. You do not need to draw the gates, but you do need to write down the **minimized** input equations for each of the inputs of each of the Flip Flops in the circuit.

Y1’

			X	**X**	
	1	**1**		**d**	
	d	**1**		**1**	**Y3**
Y2		**1**			**Y3**
Y2		**1**		**d**	
		Y1	**Y1**		

Y2’

			X	**X**	
	1	**1**	**d**	**1**	
					Y3
Y2	**1**	**1**	**1**		**Y3**
Y2		**1**	**1**		
		Y1	**Y1**		

Y3’

			X	**X**	
				d	
	1	**1**	**1**	**1**	**Y3**
Y2	**1**			**1**	**Y3**
Y2		**1**	**1**	**d**	
		Y1	**Y1**		

			X	X	
					Y3
Y2					Y3
Y2					
		Y1	Y1		

			X	X	
					Y3
Y2					Y3
Y2					
		Y1	Y1		

			X	X	
					Y3
Y2					Y3
Y2					
		Y1	Y1		

			X	X	
					Y3
Y2					Y3
Y2					
		Y1	Y1		

			X	X	
					Y3
Y2					Y3
Y2					
		Y1	Y1		

			X	X	
					Y3
Y2					Y3
Y2					
		Y1	Y1		