ECE2020 Test 3 Summer 2013 GTL
July 17, 2013 Name: KEY

Calculators not allowed. **Show your work for any possible partial credit or in some cases for any credit at all** (5 pages plus a diagram page , 100 possible points).

1) (30 points) Memory Systems
Using SIMMs (which are one type of memory chip) that are 2 M**Bytes** (two million addresses by eight bit words):
Part A Suppose that these SIMMS are used to build a 8 million address memory system with 16 bit words. Answer the following questions about this memory system:
How many address lines does each SIMM require? $2\text{M BYTES} = 2 \cdot 2^{20} = 2^{21} \Rightarrow 21$

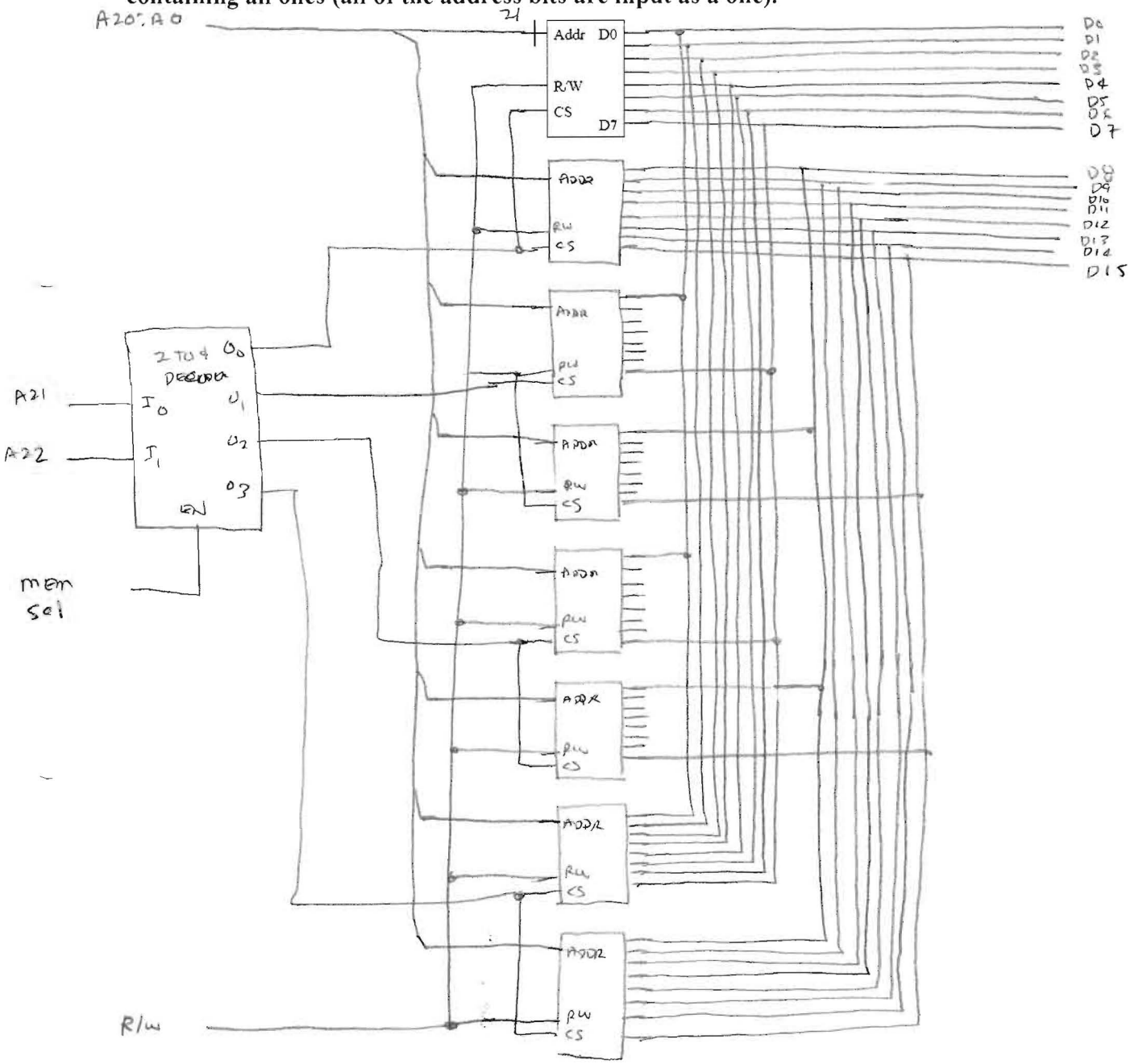
How many address lines does the entire memory system require? $8\text{M BYTES} = 2^3 \cdot 2^{20} \Rightarrow 2^{23} \Rightarrow 23$

How many SIMMS are require for the entire memory system? $\frac{8\text{M}}{2\text{M}} \times \frac{16 \text{ BITS}}{8 \text{ BITS}} = 4 \times 2 = 8$

What kind of address decoder is required? 2 TO 4

How many memory banks are needed? 4
(Continued next page)

Continued from previous page: Now using **however many** of these SIMMs you need, draw this same memory system. In the drawing you make below, add as many SIMM chips as you need and then be sure you label the memory system inputs, Addr, R/W, and Mem Sel, and the system's outputs D0, D1, D2, etc. Also label bus widths, and inputs and outputs of any required decoders. **Put a star on the chip(s) containing the memory location addressed by an address containing all ones (all of the address bits are input as a one).**



Reminder on the problem on the previous page: **Put a star on the chip(s) containing the memory location addressed by an address containing all ones (all of the address bits are input as a one).**

2) (15 points) Suppose the following inputs (in hexadecimal) are applied to the 32-bit shift unit used in the single cycle data path shown in the diagram on the last page of the exam. Determine the output of the shift unit (in hexadecimal). You may assume the "su en" is a one to allow the shift unit to output a result.

Shift Type (st)	Input Value	Shift Amount (count)	Output
rotate	0xBA24FC61	0xFFFFFFDC	A24FC61B
arithmetic	0xC57324BD	0x00000030	FFFF FFFF
logical	0x55332211	0x0000000C	0005 5332

Show your work here:

ROTATE: NEGATIVE so left rotate by FFFF FFDC.

1111 1101 1100
0000 0010 0011
+1
0010 0100 32+4 = -36

left rotate by $-36/4$ = 9 Hex digits

ARITHMETIC: positive so to right by 030 0000 0011 0000
32+16 = 48
$48/4$ = 12 Hex digits TO RIGHT WITH sign bit fill

logical: positive 0C 0000 1100 => $12/4$ = 3 hex digits TO RIGHT

3) Using the datapath we have been discussing in class and shown on the last page of the exam (you may tear off that last page) write the microcode fragments to accomplish the following procedures. Use an "X" when a value is a don't care. You may assume register 0 contains all zeros. For full credit complete the description field. **Less registers used and less steps in your answers earn more credit.** Add more step #s if you need more than is drawn below.

a) (10 points) Write a microinstruction to clear register 5 using only the logical unit.

#	X	Y	Z	rwe	im en	im va	au en	-a/s	lu en	lf	su en	st	st en	ld en	r/ -w	msel	description
1	5	X	5	1	0	0	0	X	1	0000	0	XX	0	0	X	0	R5 ← 0

b) (10 points) Write a microcode fragment (1 or more microinstructions) that writes the value that is in register 15 to memory location 0x0BDA. You may not need all the rows in the table. You may use other registers as needed.

#	X	Y	Z	rwe	im en	im va	au en	-a/s	lu en	lf	su en	st	st en	ld en	r/ -w	msel	description
1	X	X	1	1	1	0BDA	0	X	1	C	0	XX	0	0	X	0	R1 ← 0DBA
2	1	15	X	0	0	X	0	X	0	XXXX	0	XX	1	0	0	1	M[R1] ← R15
3																	
4																	
5																	

x	y	out
0	0	0
1	0	0
0	1	1
1	1	1

TO PASS THROUGH USE 1100 => C

c) (20 points) Using only registers 1, 2, 3, and 4 compute the expression (note NOR means nor operation).

$$M[R4] = \frac{8 \cdot R_2 + (R_3 \text{ NOR } R_1) - 5}{4}$$

#	X	Y	Z	rwe	im en	im va	au en	-a/s	lu en	lf	su en	st	st en	ld en	r/ -w	msel	description
1	3	1	3	1	0	X	0	X	1	0001	0	XX	0	0	X	0	R3 ← R3 NOR R1
2	2	X	2	1	1	-3	0	X	0	XXXX	1	01	0	0	X	0	R2 ← R2 shift left -3
3	3	X	3	1	1	5	1	1	0	XXXX	0	XX	0	0	X	0	R3 ← R3 - 5
4	2	3	1	1	0	X	1	0	0	XXXX	0	XX	0	0	X	0	R1 ← R2 + R3
5	1	X	1	1	1	2	0	X	0	XXXX	0	XX	0	0	X	0	R1 ← R1 shift +2
6	4	1	X	0	0	X	0	X	0	XXXX	0	XX	1	0	0	1	M[R4] ← R1

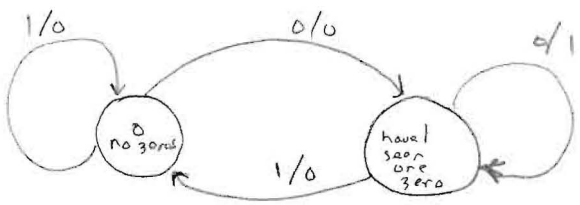
For NOR:

x	y	out
0	0	1
1	0	0
0	1	0
1	1	0

0001

4) (15 Points) Draw the state diagram (circles connected by arcs with input/output on the arcs) and then the state table (input, present state, next state, output) for a sequence detector. Do not design the hardware, you are only showing the state diagram and the state table. The sequence detector has a single bit input and a single bit output. The sequence detector will detect two consecutive zeros and output a 1 when it sees two consecutive 0's. As an example
first input is a 1, second input is a 0, third input is a 1, fourth input is a 0, fifth input is a 0...
example **inputs** and more repeated here are : 1 0 1 **0 0** 1 **0 0 0** 1 0 1 **0 0** 1
your state machine **outputs** should be: 0 0 0 0 1 0 0 1 1 0 0 0 0 1 0

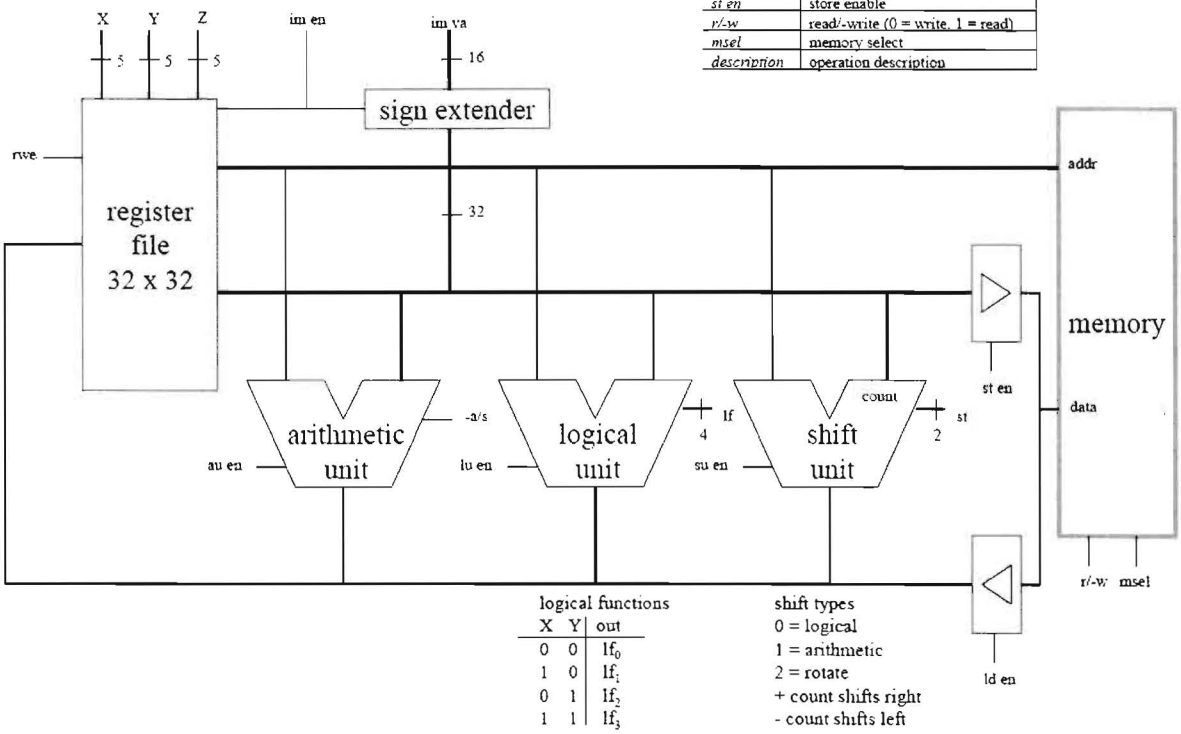
Hint: any time you are in the state where you just received a 0 as an input and then you receive another 0, you should output a one.



present state	input	next state	output
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

cycle	cycle number
X	register driven onto X bus
Y	register driven onto Y bus
Z	register written from Z bus
rwe	register write enable
im en	immediate enable on Y bus
im va	immediate value

au en	arithmetic unit enable
-a/s	-add / sub (0 = add, 1 = subtract)
lu en	logical unit enable
lf	logical function
su en	shift unit enable
st	shift type
ld en	load enable
st en	store enable
r/-w	read/-write (0 = write, 1 = read)
msel	memory select
description	operation description



logical functions

X	Y	out
0	0	lf_0
1	0	lf_1
0	1	lf_2
1	1	lf_3

shift types
0 = logical
1 = arithmetic
2 = rotate
+ count shifts right
- count shifts left