Problem 1 (3 parts, 30 points) Memory Systems

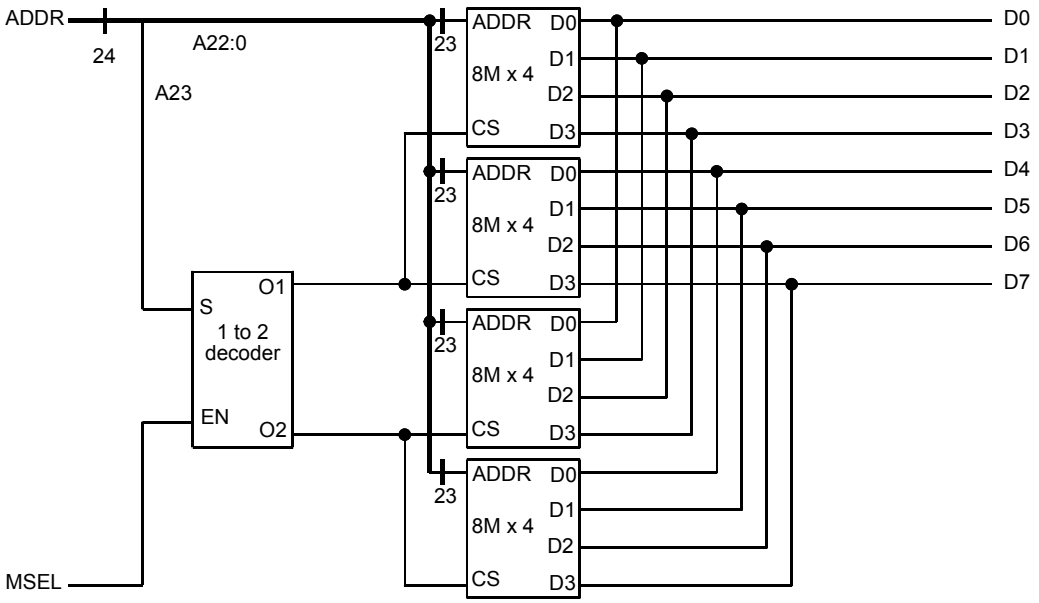
Part A (12 points) Consider a DRAM chip organized as **4 million addresses** of **16-bit words**. Assume both the DRAM cell and the DRAM chip are square. The column number and offset concatenate to form the memory address. Using the organization approach discussed in class, answer the following questions about the chip. *Express all answers in decimal (not powers of two)*.

total number of bits in address	**$\log_2(4M) = 22$**
number of columns	**$\text{sqrt}(2^{22}x2^4) = \text{sqrt}(2^{26}) = 2^{13} = 8K$**
column decoder required (*n* to *m*)	**13 to 8K**
number of words per column	**8K / 16 = 512**
type of mux required (*n* to *m*)	**512 to 1**
number of address lines in column offset	**9**

Part B (10 points) Consider a memory system with **256 million addresses** of **128-bit words** using **4 million** address by **16-bit word** memory DRAM chips.

word address lines for memory system	**$\log_2(256M) = 28$**
chips needed in one bank	**128/16 = 8**
banks for memory system	**256M / 4M = 64**
memory decoder required (*n* to *m*)	**6 to 64**
DRAM chips required	**8 x 64 = 512**

Part C (8 points) Design a **16 million address by 8 bit** memory system with **8 million x 4 bit** memory chips. *Label all busses and indicate bit width*. Assume R/W is connected and not shown here. Use a bank decoder if necessary. Be sure to include the address bus, data bus, and MSEL.



Problem 2 (3 parts, 23 points) Datapath Elements and State Machines

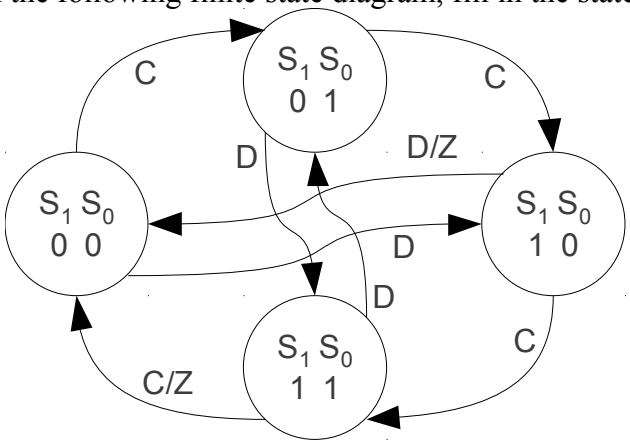
Part A (6 points) Suppose the following inputs (in `hexadecimal`) are applied to the 32-bit barrel shifter used in the datapath. Determine the output (in `hexadecimal`). Assume the shift amount is drawn from the 16-bit immediate value.

Shift Type	Shift Amount	Input Value	Output Value
logical	0x0010	BEAD2533	**0000BEAD**
arithmetic	0x0014	39317ACE	**00000393**
rotate	0xFFF8	FADE3650	**DE3650FA**

Part B (9 points) Consider the following input and output values for a logical operation. Determine the *logical function and function code* (in hexadecimal) required for the operation.

X Input	Y Input	Output	Logical Function	Function Code
DCBA4321	FF00FF00	FFBAFF21	OR	E
DCBA4321	FF00FF00	23FFBCFF	NAND	7
DCBA4321	FF00FF00	00FF00FF	$\overline{Y}$	3

Part C (8 points) Given the following finite state diagram, fill in the state table below.



S_1	S_0	C/$\overline{D}$	NS_1	NS_0	Z	S_1	S_0	C/$\overline{D}$	NS_1	NS_0	Z
0	0	0	**1**	**0**	**0**	1	0	0	**0**	**0**	**1**
0	0	1	**0**	**1**	**0**	1	0	1	**1**	**1**	**0**
0	1	0	**1**	**1**	**0**	1	1	0	**0**	**1**	**0**
0	1	1	**1**	**0**	**0**	1	1	1	**0**	**0**	**1**

Give the simplified Boolean expression for computing **Z** in terms of the current state and the input.

Z = $S1 \cdot \overline{S0} \cdot D + S1 \cdot S0 \cdot C = S1(\overline{S0} \cdot D + S0 \cdot C) = S1(\overline{S0 \text{ XOR } C})$.

Problem 3 (5 parts, 30 points) Microcode

For Parts A-C, using the supplied datapath, write microcode fragments to accomplish the following procedures. Express all values, except memory addresses, in hexadecimal notation. Use 'X' when a value is don't cared. For maximum credit, complete the description field.

Part A (5 points) $6 ← mem[$4]. **Use only registers 4 and 6**.

#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
1	4	X	6	1	0	X	0	X	0	X	0	X	1	0	1	1	$6 ← mem[$4]

Part B (8 points) Multiply $9 by 33 and put the result in $9. **Use only registers 7 and 9**.

#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
2	9	X	7	1	1	FFFB	0	X	0	X	1	1	0	0	X	0	$7 ← 32 * $9
3	9	7	9	1	0	X	1	0	0	X	0	X	0	0	X	0	$9 ← (32 + 1)$9

Part C (8 points) mem[0x2020] ← $5. **Use only registers 1 and 5**.

#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
4	X	X	1	1	1	2020	0	X	1	C	0	X	0	0	X	0	$1 ← 0x2020
5	1	5	X	0	0	X	0	X	0	X	0	X	0	1	0	1	mem[2020] ← $5

Part D (4 points) Write the MIPS instruction that is equivalent to the following microinstruction. (A summary of MIPS instructions is given on the next page.)

#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
6	2	8	7	1	0	x	0	x	1	6	0	x	0	0	x	0	

Equivalent MIPS Instruction: `xor $7, $2, $8 # R7 ← R2 xor R8`

Part E (5 points) Write the MIPS instruction that is equivalent to the following microinstruction. (A summary of MIPS instructions is given on the next page.)

#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
7	3	x	6	1	1	FFFD	0	x	0	x	1	0	0	0	x	0	

Equivalent MIPS Instruction: `sll $6, $3, 3 # R6 ← R3 << 3 (x8)`

Problem 4 (3 parts, 17 points) Assembly Programming

For maximum credit in each of the following problems, *use the fewest number of instructions*.

Part A (6 points) Write a MIPS code fragment that reads an integer value from memory location 1000 and puts it into register $6. **Use only registers $0 and $6.**

label	*instruction*	*comment*
	`addi $6, $0, 1000`	`# load address in R6`
	`lw $6, ($6)`	`# R6 ← mem[1000]`

Part B (7 points) Write a MIPS code fragment that branches to the label `Target` if register $3 ≥ $4. **Use only registers $3, $4, $2, and $0.**

label	*instruction*	*comment*
	`slt $2, $3, $4`	`# if R3 >= R4`
	`beq $2, $0, Target`	`# then branch to target`

Part C (4 points) Write a MIPS code fragment that divides the integer in register $7 by 32 and put the result in register $7. **Use only register $7.**

label	*instruction*	*comment*
	`sll $7, $7, 5`	`# R7 ← R7 >> 5`