ECE 2030 A 11:00am Computer Engineering Spring 2011
4 problems, 4 pages Exam Three Solutions 13 April 2011

Problem 1 (2 parts, 18 points) Datapath Elements

Part A (9 points) Consider the following inputs, shift types, and directions. Determine the resulting outs (in hexadecimal).

Input Value	Output Value	Shift Type	Shift Amount
0x87654321	0x65432187	rotate	left 8 bits
0x87654321	0xFFF87654	arithmetic	right 12 bits
0x87654321	0x10000000	logical	left 28 bits

Part B (9 points) Consider the following input and logical operation function codes. Determine the *logical function* and output value (in hexadecimal) for the operation.

X Input	Y Input	Output Value	Logical Function	Function Code
87654321	00FF00FF	0x87FF43FF	OR	E
87654321	00FF00FF	0xFF00FF00	$\overline{Y}$	3
87654321	00FF00FF	0xFF9AFFDE	NAND	7

Problem 2 (3 parts, 32 points) Memory Systems

Part A (12 points) Consider a DRAM chip organized as **512 million addresses** of **eight bit words**. Assume both the DRAM cell and the DRAM chip is 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**.

number of columns sqrt(2^{29}*2^{3}) = sqrt(2^{32}) = 2^{16} = 64K

column decoder required (*n* to *m*) 16 to 64K

type of mux required (*n* to *m*) 2^{16} / 2^{3} = 2^{13} = 8K to 1

number of muxes required 8

number of address lines in column number 16

number of address lines in column offset 13

Part B (10 points) Consider a **eight Gbyte** memory system with **two billion addresses** of **four byte words** using DRAM chips organized as **512 million** addresses by **eight bit words.**

word address lines for memory system Log2(2 billion) = 31

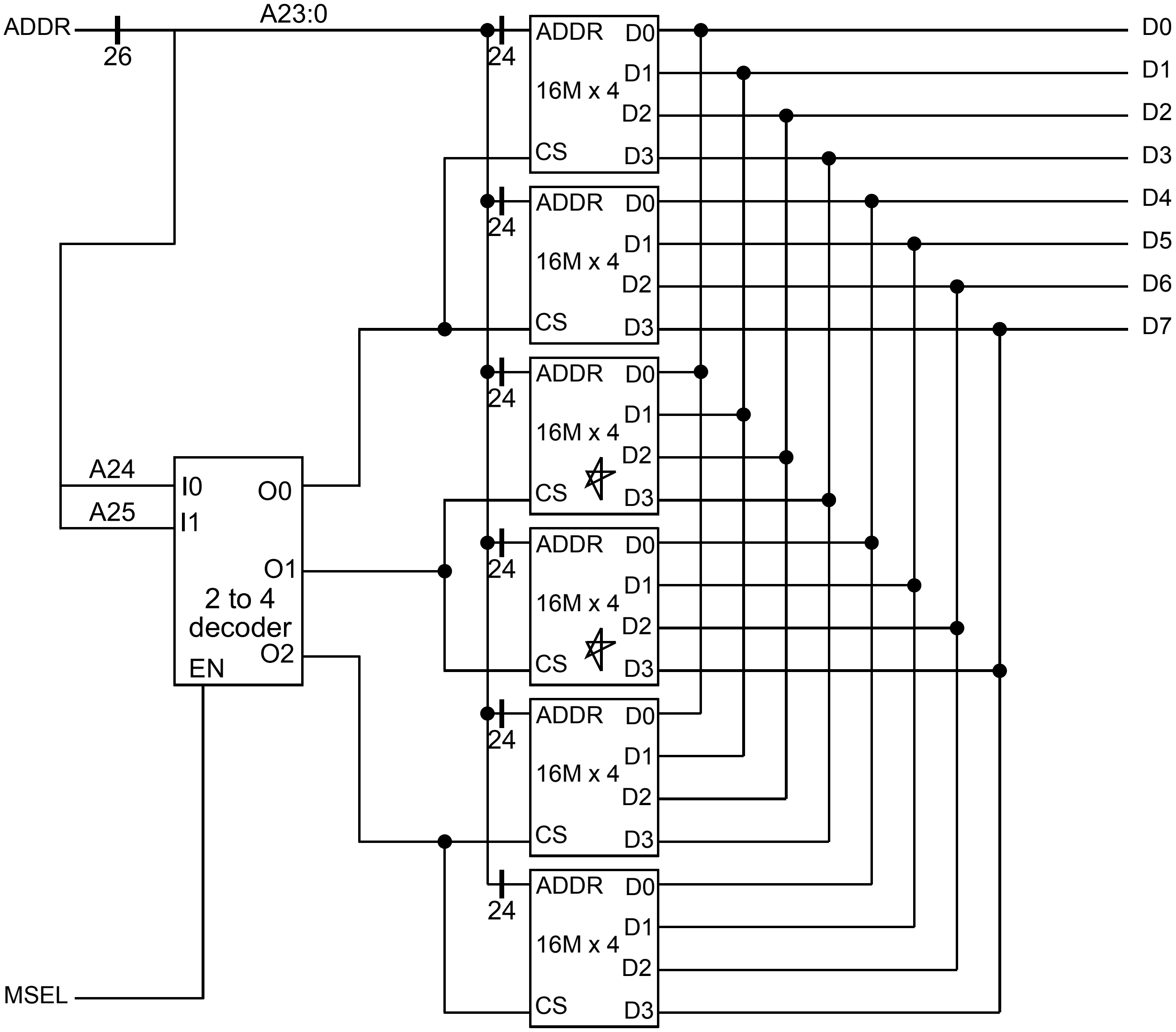
chips needed in one bank 4 bytes / 1 byte = 4 chips/bank

banks for memory system 2B / 512M = 2^{31}/2^{29} = 2^{2} = 4 banks

memory decoder required (*n* to *m*) 2 to 4

DRAM chips required 4 banks x 4 chip/bank = 16 chips

Part C (10 points) Design a 48 M address x 8 bit memory system with six 16 M address x 4 bit memory chips. **Label all busses and indicate bit width**. Assume R/W is connected and not shown here. Use a decoder if necessary. Place a star on the chip(s) that contain address 25,000,000.



Problem 3 (5 parts, 28 points) Microcode

Using the supplied datapath, write microcode fragments to accomplish the following procedures. Express all values in hexadecimal notation. Use 'X' when a value is don't cared. For shift operations, expression shift amount as unsigned value. The shift direction field (*dir*) is 1 for left shifts, 0 for right shifts. For maximum credit, complete the description field.

Part A (5 points) $7 ← $8 + $9. **Use only registers 7, 8, and 9.**

#	*X*	*Y*	*Z*	*rwe*	*im en*	*im va*	*au en*	*-a /s*	*lu en*	*lf*	*su en*	*st*	*dir*	*description*
1	8	9	7	1	0	X	1	0	0	X	0	X	X	R7 ← R8 + R9

Part B (5 points) Mask all but the ten least significant bits of $6. **Use only register 6.**

#	*X*	*Y*	*Z*	*rwe*	*im en*	*im va*	*au en*	*-a /s*	*lu en*	*lf*	*su en*	*st*	*dir*	*description*
1	6	X	6	1	1	3FF	0	X	1	8	0	X	X	R6 ← R6 & 0x3FF

Part C (5 points) Rotate register $3 by 16 bits. **Use only register 3.**

#	*X*	*Y*	*Z*	*rwe*	*im en*	*im va*	*au en*	*-a /s*	*lu en*	*lf*	*su en*	*st*	*dir*	*description*
1	3	X	3	1	1	10	0	X	0	X	1	2	0	R3 ← R3 >> 16 (rotate)

Part D (8 points) $5 ← 63 * $4. **Use only registers 4 and 5.**

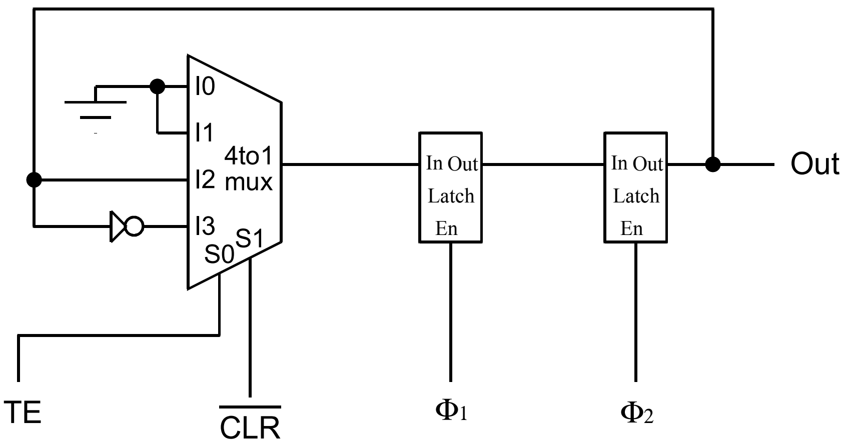
#	*X*	*Y*	*Z*	*rwe*	*im en*	*im va*	*au en*	*-a /s*	*lu en*	*lf*	*su en*	*st*	*dir*	*description*
1	4	X	5	1	1	6	0	X	0	X	1	1	1	R5 ← R4 x 64
2	5	4	5	1	0	X	1	1	0	X	0	X	X	R5 ← R5 - R4

Part E (5 points) $7 ← $5 xor $6. **Use only registers 5, 6, and 7.**

#	*X*	*Y*	*Z*	*rwe*	*im en*	*im va*	*au en*	*-a /s*	*lu en*	*lf*	*su en*	*st*	*dir*	*description*
1	5	6	7	1	0	X	0	X	1	6	0	X	X	R7 ← R5 XOR R6

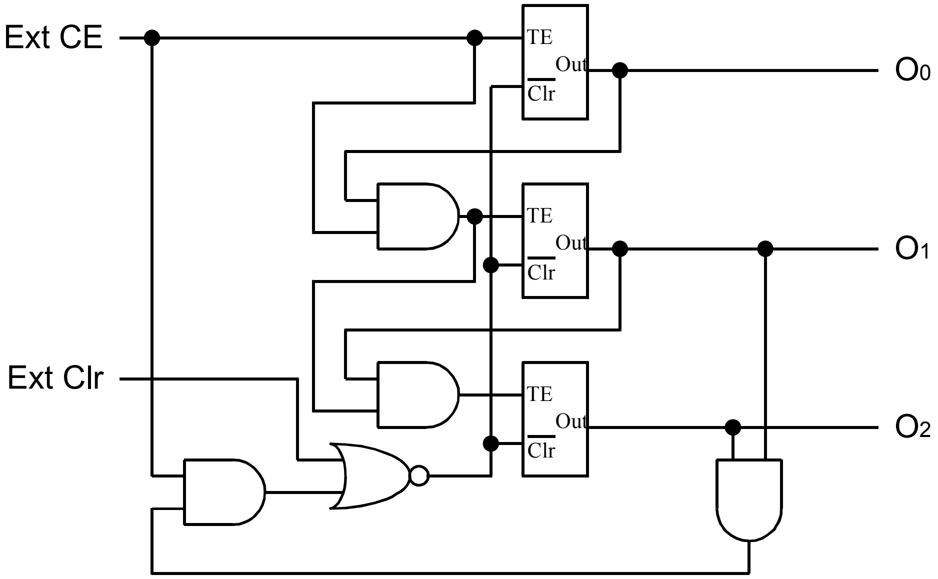
Problem 4 (2 parts, 22 points) Toggle Cells and Counters

Part A (11 points) Design a toggle cell using **two transparent latches**, **one 4to1 mux**, and **one inverter** (use icons, **labeling inputs & outputs**). Your toggle cell should have an active high toggle enable input **TE**, and an active low clear input **$\overline{\text{CLR}}$**, clock inputs Φ_1 and Φ_2, and an output **Out**. The **$\overline{\text{CLR}}$** signal has precedence over **TE**. Also complete the behavior table for the toggle cell.



TE	$\overline{\text{CLR}}$	CLK	Out
0	0	↑↓	0
1	0	↑↓	0
0	1	↑↓	Q_o
1	1	↑↓	$\overline{Q_o}$

Part B (11 points) Now combine these toggle cells to build a **divide by 7** counter. Your counter should have an external clear, external count enable, and three count outputs O_2, O_1, O_0. Use any basic gates (AND, OR, NAND, NOR, & NOT) you require. Assume clock inputs to the toggle cells are already connected. *Your design should support multi-digit systems*.


Ext CE
Ext Clr
TE
Out
$\overline{\text{Clr}}$
TE
Out
$\overline{\text{Clr}}$
TE
Out
$\overline{\text{Clr}}$
O_0
O_1
O_2