CPSC 218, 03S Term 2 — Midterm Exam — Solution

1. **(10 marks)** Circle all of the **zero or more** correct answers for each of the following questions.

1a. How many unique 4-variable Boolean functions are there?

(a) 4 **(b)** 16 **(c)** 64 **(d)** 256 **(e)** 65536

> **(e)**: 4-variable truth table has 2^4 rows. Viewing the output function-value column as a 2^4 digit binary number, there are 2^{2^4} unique values for this column and thus $2^{2^4} = 2^{16} = 65536$ unique truth tables.

1b. Which of the following are minimal, functionally complete sets of Boolean operators? (Minimal means that no subset is complete.)

(a) $\{$NAND$\}$ **(b)** $\{$XOR$\}$ **(c)** $\{$AND,NOT,OR$\}$ **(d)** $\{$AND,OR$\}$

> **(a)**: NAND is AND+NOT, which is complete. Neither XOR nor AND+OR are complete and AND+NOT+OR is not minimal.

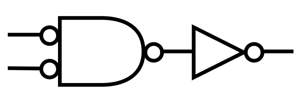
1c. Which operations require different hardware implementations for unsigned and signed-twos-complement numbers?

(a) add, ignoring overflow **(b)** add, testing for overflow **(c)** shift right

(d) inequality compare **(e)** load from memory

> **(b), (c), (d) and (e)**: Add, ignoring overflow, is the same for both. For the rest: unsigned overflow detected by $C_{out} = 1$, signed overflow detected by sign flip when both operands have same sign; unsigned shift right should set high-order-bit to 0, signed shift right should set high-order-bit to value of high-order-bit before shift; unsigned inequality compare uses magnitude, signed compare ensure that negative numbers are smaller than positive number and otherwise uses magnitude; unsigned load of byte (8 bits) or short (16 bits) requires that high-order bits be *zero extended*, signed load requires that high-order bits be *sign extended*.

1d. What Boolean operator(s) are equivalent to this logic diagram?



(a) AND **(b)** OR **(c)** NAND **(d)** NOR

> **(d)**: Move AND-gate-input circles to the left, changing AND to OR and leaving two circles on OR's output. Move one of the OR-output circles to right canceling INVERTER's circle, changing it into a BUFFER. You're left with a NOR plus a BUFFER which has the same truth table as a NOR.

1e. What boolean functions are equivalent to $f = ab + ab'$?

(a) a **(b)** b **(c)** $a + b$ **(d)** $(a'b' + a'b)'$ **(e)** $((a' + b')(a' + b))'$

> **(a), (d), (e)**: Using distributive, complement and identity axioms, $ab + ab' = a(b + b') = a(1) = a$. Starting from $ab + ab' = a$ and using identity, complement, distributive, complement and DeMorgan's axioms, $a = a + 0 = a + bb' = (a + b)(a + b') = ((a + b)(a + b'))'' = (a'b' + a'b)'$. Finally, using complement and DeMorgan's axiom, $ab + ab' = (ab + ab')'' = ((a' + b')(a' + b))'$.

2. **(6 marks)** Consider the following k-map.

wx \ yz	00	01	11	10
00	1	0	0	1
01	1	x	0	1
11	x	1	x	0
10	1	1	0	x

wx \ yz	00	01	11	10
00	1	0	0	1
01	1	x	0	1
11	x	1	x	0
10	1	1	0	x

2a. Give the simplest boolean function derived directly from this k-map, in sum-of-products form.

$f = w'z' + wy'$

2b. Give the simplest boolean function derived directly from this k-map, in product-of-sums form.

$f = (w'z + wy)' = (w + z')(w' + y')$

3. **(5 marks)** What is the main advantage and disadvantage of DRAM compared to SRAM?

3a. Advantage of DRAM:

Small and thus cheap to build. One transistor per bit.

3b. Disadvantage of DRAM:

Slow compared to static RAM due primarily to the fact that reading a bit requires that its entire row first be *pre-charged*.

4. **(9 marks)** Consider **six-bit**, signed numbers whose values range from -32 to 31.

4a. Use hexadecimal notation to give the representation of -17 in each of the following encodings.

i. sign magnitude:

0x31

ii. two's complement:

0x2f

iii. biased:

0x0f

4b. What is the hexadecimal value of the six-bit, two's complement representation of -17 after it has been shifted to the right twice?

0x3b

4c. Which of the following six-bit (unsigned or two's complement signed) operations result in overflow? Justify your answer.

i. unsigned 0x2a + 0x16?

overflow; carry out of 1

ii. signed 0x2a + 0x16?

no overflow; negative and positive number

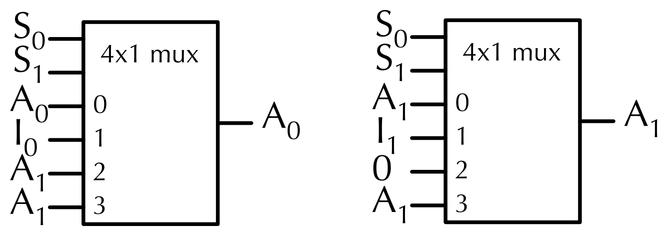
iii. signed 0x2a + 0x26?

overflow; sign change

iv. signed 0x2a + 0x36?

no overflow

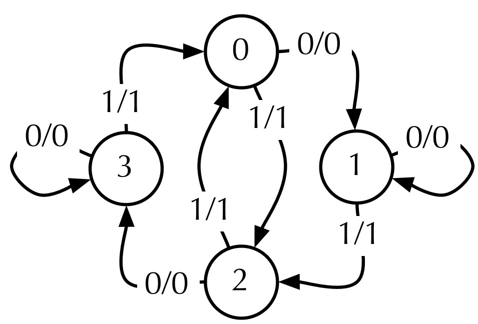
5. **(8 marks)** Using packages such as decoders, encoders, multiplexers and D flip flops, draw the logic diagram of a 2-bit, divide-by-two register. Use control inputs of 0 to leave the value unchanged, 1 to load a new value, 2 to divide an unsigned number by 2 and 3 to divide a signed, two's complement number by 2. Be sure to carefully label packages and their inputs and outputs. (Hint: remember that there is an easy way to divide an integer by two.)



6. **(12 marks)** Design a two-bit counter with input C and output E. When $C = 0$, the counter moves to the closest *odd* number *greater than or equal to* its current value (e.g., 2 moves to 3; 3 stays at 3). When $C = 1$, the counter moves to the closest *even* number *greater than* its current value — or to 0 if the current value is 2 or 3 — (e.g., 0 moves to 2; 1 moves to 2); consider 0 to be even. The output E should be 1 when the counter value is even and 0 otherwise. Use D flip flops (recall that the next state of a D flip flop is equal to its input).

6a. Draw the state diagram.

6b. Draw the excitation table.



Q(t)			Q(t+1)				
A	B	C	A	B	E	D_A	D_B
0	0	0	0	1	0	0	1
0	0	1	1	0	1	1	0
0	1	0	0	1	0	0	1
0	1	1	1	0	1	1	0
1	0	0	1	1	0	1	1
1	0	1	0	0	1	0	0
1	1	0	1	1	0	1	1
1	1	1	0	0	1	0	0

6c. Draw all required k-maps and boolean functions.

D_A

A \ BC	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$D_A = A'C + AC' = A \oplus C$

D_B

A \ BC	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$D_B = C'$

E

A \ BC	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$E = C$

6d. Draw the circuit.