Cpr E 281 HW06
SOLUTION

ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Arithmetic Circuits and Combinational-Circuit Building Blocks
Assigned Date: Seventh Week
Due Date: Monday, Oct. 10, 2016

P1. (20 points)

Consider the addition of the two ***n*-bit 2's complement** numbers:

$$X = x_{n-1}x_{n-2}\ldots x_1x_0$$

$$Y = y_{n-1}y_{n-2}\ldots y_1y_0$$

Suppose the sum is $S = s_{n-1}s_{n-2}\ldots s_1s_0$ and the carry is $C_n = c_nc_{n-1}c_{n-2}\ldots c_1c_0$.

a) (5 points) If X is positive, Y is negative, and c_{n-1}=0, what should be the values of c_n and s_{n-1} ? Will overflow occur?

b) (5 points) If X is negative, Y is negative, and c_{n-1}=0, what should be the values of c_n and s_{n-1} ? Will overflow occur?

c) (5 points) Following the idea in part (a) and (b), please construct a truth table for the values of c_n and s_{n-1} for all combinations of the sign of X, the sign of Y, and the value of c_{n-1}. For each combination, please also state if overflow occurs or not.

d) (5 points) Based on the truth table in part (c), prove that Overflow = $c_n \oplus c_{n-1}$.

Solution:

a)

If X is positive, then $x_{n-1} = 0$ in 2's complement.
If Y is negative, then $y_{n-1} = 1$ in 2's complement.
$c_{n-1} = 0$ is given.

$$\begin{array}{cccccc} c_n & 0 & c_{n-2} & \ldots & c_1 & c_0 \\ & 0 & x_{n-2} & \ldots & x_1 & x_0 \\ + & 1 & y_{n-2} & \ldots & y_1 & y_0 \\ \hline & s_{n-1} & s_{n-2} & \ldots & s_1 & s_0 \end{array}$$

$\Rightarrow$ $s_{n-1} = 1$ and $c_n = 0$. Overflow: NO

b)

If X is negative, then $x_{n-1} = 1$ in 2's complement.
If Y is negative, then $y_{n-1} = 1$ in 2's complement.
$c_{n-1} = 0$ is given.

$$\begin{array}{cccccc} c_n & 0 & c_{n-2} & \ldots & c_1 & c_0 \\ & 1 & x_{n-2} & \ldots & x_1 & x_0 \\ + & 1 & y_{n-2} & \ldots & y_1 & y_0 \\ \hline & s_{n-1} & s_{n-2} & \ldots & s_1 & s_0 \end{array}$$

$\Rightarrow$ $s_{n-1} = 0$ and $c_n = 1$. Overflow: YES c)

c_{n-1}	x_{n-1}	y_{n-1}	s_{n-1}	c_n	overflow?
0	0	0	0	0	NO
0	0	1	1	0	NO
0	1	0	1	0	NO
0	1	1	0	1	YES
1	0	0	1	0	YES
1	0	1	0	1	NO
1	1	0	0	1	NO
1	1	1	1	1	NO

d)

c_{n-1}	c_n	overflow?
0	0	NO
0	1	YES
1	0	YES
1	1	NO

$$\text{Overflow} = c_n \oplus c_{n-1}$$

P2. (10 points)
In class we learned that a carry-lookahead adder is faster than a ripple-carry adder. Could you explain why sometimes a designer might still choose a ripple-carry adder instead of a carry-lookahead adder?

Solution:

A carry-lookahead adder requires more gates to implement than a typical ripple-carry adder. If the area of chip is very limited and delays are not our first concern, a designer may choose to implement a ripple-carry adder over a carry-lookahead adder.

P3. (10 points)
Perform the following conversions.

a) (5 points) Decimal number 5.375 to fixed-point number.
b) (5 points) Fixed-point number 1101.011 to decimal number.

Solution:

a)

$(5)_{10} = (101)_2$
$0.375 \times 2 = \boxed{0}.75$
$0.75 \times 2 = \boxed{1}.5$
$0.5 \times 2 = \boxed{1}.0$

$\Rightarrow (5.375)_{10} = (101.011)_2$

b)

$(1101)_2 = (13)_{10}$
$.0111 = 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3} + 1\times2^{-4} = 0.4375$
$\Rightarrow (1101.0111)_2 = (13.4375)_{10}$

P4. (10 points)
Convert the decimal number 15.625 to IEEE 754 single-precision floating number format.

Solution:

$(15)_{10} = (1111)_2$
$(0.625)_{10} = (0.101)_2$

$(15.625)_{10} = (1111.101)_2 = 1.111101 \times 2^3 = \pm 1.M \times 2^{E-127}$

S = 0
E = 130 = $(10000010)_2$
M = 111101 00…..0

$\Rightarrow \underbrace{0}_{S}\ \underbrace{10000010}_{E}\ \underbrace{1111010\ 00000000\ 00000000}_{M}$

P5. (10 points)
Convert the following IEEE 754 single-precision floating number to decimal number.
1 01111110 0110000 00000000 00000000

Solution:

S = 1
E = $(01111110)_2 = 126$
M = 0110000 0…0

$\pm 1.M \times 2^{E-127} = -1.011 \times 2^{126-127}$ = - $(0.1011)_2$ = - ($1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4}$)

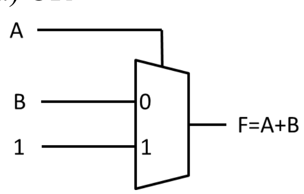
= - 0.6875

P6. (20 points)
Use only 2-to-1 multiplexer to implement each of the following functions:

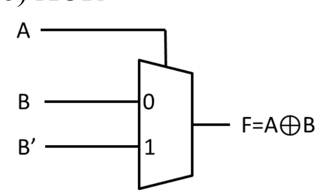
a) (5 points) $F(A,B) = A + B$ (OR)
b) (5 points) $F(A,B) = A \oplus B$ (XOR)
c) (5 points) $F(A,B) = \overline{A \bullet B}$ (NAND)
d) (5 points) $F(A,B) = \overline{A + B}$ (NOR)

Solution:

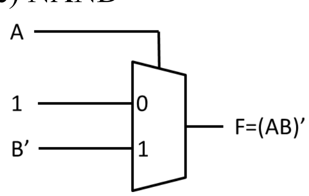
a) OR



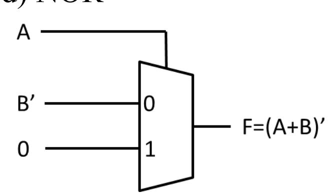
b) XOR



c) NAND



d) NOR



P7. (10 points)
Use only 2-to-1 multiplexers to implement the circuit for the following function:

$$F(A,B,C) = \prod M(1,2,4,5)$$

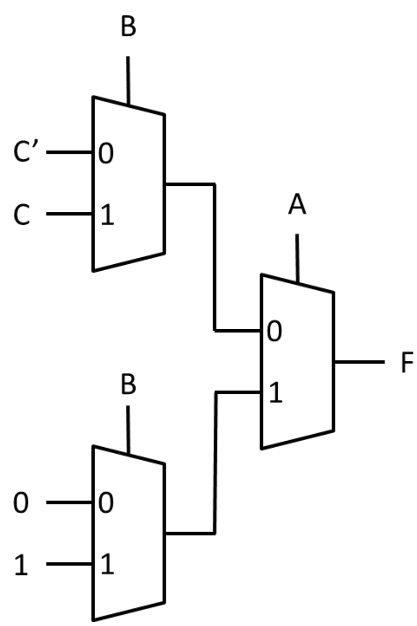
Assume the inverse of each input variable is available. (i.e., you can directly use the inverse of each input variable *A*, *B*, or *C*, in your answer.)

Solution:

Truth table (optional)

A	*B*	*C*	*F*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Circuit



P8. (10 points)
Repeat P7, but this time using only one 4-to-1 multiplexer.

Solution: