Homework #3

Steven Lei

Out: January 24, 2023
Due: January 28, 2023
Points: 100

For this assignment you must do the problems "by hand". That is, you may **not** use a calculator or program to do the conversions or the binary arithmetic.

Conversions (20 points)

Complete the following table of equivalent values (assuming 8-bit values). For example, in the first row, binary 01110110, unsigned decimal 118, 2's complement +118, 1's complement +118, Excess-128 –10, and BCD 76 all have the same 8-bit pattern. Note that not all binary values are valid BCD values.

Invert +1
0 → 0 0→127 +0 20 0→127
1 → 0 -1→-128 1 → 0 -0→-127

Binary	Unsigned Decimal	Signed Decimal (2's Complement)	Signed Decimal (1's Complement)	Signed Decimal (Excess-128)	BCD
01110110	118	+118	+118	–10	76
00000000	0	+0	+0	-128	+0
11111111	255	-1	-0	127	NA
01111111	127	+127	+127	-1	NA
10000000	128	-128	-127	0	+80
01001001	73	+73	+73	-55	+49
10010100	148	-108	-107	20	+94

$2^8 - 1$
$2^7 - 1$
2^7
$64 + 2^3 + 2^{\square}$
$2^2 + 2^4 + 2^7$
$4 + 16 + 127$

1101011
→ 1
-1101100 = 108

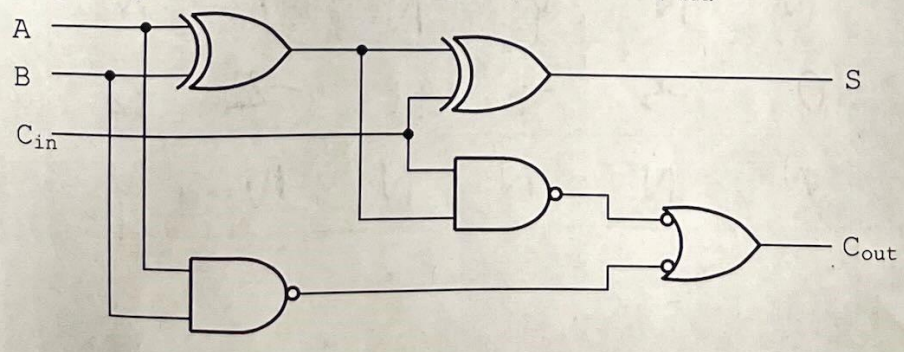
1101011

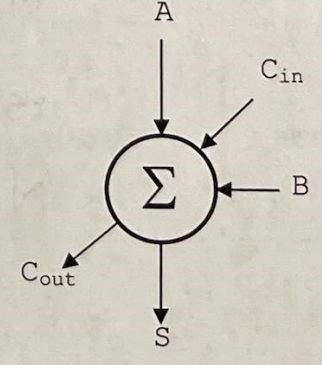
Binary Arithmetic (40 points)

The attached page contains a table of values to add and subtract in binary. The values to add and subtract (A and B) are 4-bits and the resulting sums and differences must also be 4-bits. For A, B, A + B, and A – B the equivalent unsigned decimal and 2's complement signed decimal values must be computed. Additionally it must be indicated if there is a carry out of the addition, a borrow in to the subtraction, and whether the addition or subtraction overflowed when treated as unsigned values and as signed values.

Adder/Subtracter Circuit (40 points)

The circuit for a 1-bit full adder was given in lecture and is repeated below. The symbol on the right is often used to represent this circuit. For this problem, design an 8-bit adder/subtracter by using eight (8) full adders and any other necessary logic. Note that you may use the full adder symbol on the right for ease of notation. The circuit has as its inputs two 8-bit values A (A_7 to A_0, with A_0 being the low order bit) and B (B_7 to B_0, with B_7 being the high order bit), and an add/subtract signal called Add/Sub which is high (1) to indicate subtraction (A – B) and low (0) to indicate addition (A + B). The circuit should output an 8-bit result R (R_7 to R_0, with R_7 being the high order bit) and an active high carry/borrow out (C_{out}).





2s Complement

A	Binary Value	0101	0110	0000	1000	1111	1110	1100	0100	1001
	Unsigned Decimal Value	5	6	0	8	15	14	12	4	9
	Signed Decimal Value	+5	+6	0	-8	-1	-2	-4	+4	-7
B	Binary Value	1101	0001	1000	1000	0100	1111	1011	0110	0111
	Unsigned Decimal Value	13	1	8	8	4	15	11	6	7
	Signed Decimal Value	-3	+1	-8	-8	+4	-1	-5	+6	+7
A ? B	Unsigned < = >	<	>	<	=	>	<	>	<	>
	Signed < = >	>	>	>	=	<	<	>	<	<
A + B	Binary Value	0010	0111	1000	0000	0011	1101	0111	1010	0000
	Carry	Y	N	N	Y	Y	Y	Y	N	Y
	Unsigned Decimal Value	2	7	8	0	3	13	7	10	0
	Unsigned Overflow	Y	N	N	Y	Y	Y	Y	N	Y
	Signed Decimal Value	+2	+7	-8	0	+3	-3	+7	-6	0
	Signed Overflow	N	N	N	Y	N	N	Y	N	N
A − B	Binary Value	1000	0101	1000	0000	1011	1111	0001	1110	0010
	Borrow	Y	N	Y	N	N	Y	N	Y	N
	Unsigned Decimal Value	8	5	8	0	11	15	1	14	2
	Unsigned Overflow	Y	N	Y	N	N	Y	N	Y	N
	Signed Decimal Value	-8	+5	-8	0	-5	-1	+1	-2	+2
	Signed Overflow	Y	N	N	N	N	N	N	N	Y

Adder/subtracter 8 bit