Number system and conversions (section 1.4 of textbook)

Vikas Dhiman for ECE275

September 2, 2022

6 Signed binary numbers

Signed numbers include both negative and positive numbers. There three common signed number representations

1. Sign magnitude representation
2. One's complement
3. Two's complement

6.1 Sign-magnitude representation

The Most significant (left most) *bit* (binary digit) represents sign (0 = + and 1 = −), the rest represent the magnitude. Example, a 5-bit number $(11010)_2$ in signed magnitude representation has the value of $(-1010)_2 = -10$. Note that +10 has to be represented by a leading 0 at the most significant bit (MSB) $+10 = (01010)_2$. Hence, the number of bits have to be specified.

Problem 5 • *Write down all possible 4-digit binary numbers and corresponding decimal values if they are in signed magnitude format? What is the minimum and maximum value?*

- *What is the minimum and maximum value of n-digit signed binary number in sign-magnitude format?*

4-digit

Binary		Decimal	
0000	=	+0	
0001	=	+1	
0010	=	+2	
0011	=	+3	
0100	=	+4	
0101	=	+5	
0110	=	+6	
0111	=	+7	← Max
1000	=	−0	
1001	=	−1	
1010	=	−2	
1011	=	−3	
1100	=	−4	
1101	=	−5	
1110	=	−6	
1111	=	−7	← Min

$(0000)_2 = (1000)_2$

Max: $2^3 - 1$ n-digit: $2^{n-1} - 1$

Min: $-(2^3 - 1)$ n-digit: $-(2^{n-1} - 1)$

6.2 One's complement negation

You can convert a positive number (say +10) to negative number by applying a negative sign in front of it (-(+10) = -10). It is more evident from taking negative of a negative number (-(-10) = +10). In case of sign-magnitude representation, the "negative operator" flips the sign bit. The next two signed number representations (1's complement and 2's complement) are designed around specific negative operator definitions.

Negate $13_{10} = 01101_2$ using 5-bit one's complement.

$11111 - 01101$

$-13_{10} = -(01101)_2$

$= 10010_2$

$13 - 3$

$13 + (-3)$

$= 10$

Negate -13_{10} using 5-bit one's complement.

$-(-13) = -(10010)_2$

$= 01101_2$

6.3 One's complement binary numbers

In one's complement representation, the negative operation is obtained by flipping all the bits of the binary number. Example, a 5-bit one's complement of $+10 = (01010)_2$ is $(10101)_2 = -10$. Note that flipping bits is equivalent to subtracting the number from $(11111)_2$, hence the name. You can also confirm that double negative operator yields back the same number.

Problem 6 • *Write down all possible 4-digit binary numbers and corresponding decimal values if they are in sign magnitude format? What is the minimum and maximum value?*

• *What is the minimum and maximum value of n-digit signed binary number in one's complement?*

4-bit 1's complement	Decimal	1's complement Binary
0000	0	
⋮		
0111	7 (2^3-1)	
1000	-7 $-(2^3-1)$	0111
1001	-6	0110
1010	-5	
1011	⋮	
1100		
1101		
1110		
1111	-0	0000

n-bit 1's compl

Min $-(2^{n-1}-1)$

Max $+(2^{n-1}-1)$

Problem 7 *Determine the decimal values of the following 1's complement ~~8-digit~~ 8-bit binary numbers :*

1. 01101110
2. 10101101

$= 64 + 16 + 2 = 82$

$= 2^6 + 2^4 + 2$

01010010

$-d$ $+d$

-82

Problem 8 *Convert the decimal numbers* $\boxed{-17}$ *and* $\boxed{+23}$ *into the 6-digit one's complement binary numbers and try adding them. What adjustments will you need to make to get the right result's (23-17=6) in binary representation.*

$17_{10} = 010001$

$-17 \longrightarrow 101110$

$23 \longrightarrow 010111$

$\underbrace{000101}_{5}$ $\xrightarrow{+1}$ 6

6 000110

-6 111001

23 010111

-23 101000

011110

$-\boxed{30}$

$-6 - 23 =$

-6 111001

-23 101000

100001 $\xrightarrow{+1}$ -29 $+1$

$2^N - 1$

6.4 Two's complement negation

In two's complement representation, the n-digit negative number is obtained by subtracting the positive number from 2^n. Example, two's complement of 5-digit binary number $+10 = (01010)_2$ is $2^5 - 10 = 22 = (11000)_2$. An easier algorithm to get two's complement goes via one's complement. Note that $(11111)_2 = 2^5 - 1$. We can get two's complement by adding 1 to one's complement. To get two's complement:

1. Flip all the bits. (Same as taking one's complement).
2. Add 1 to the number.

$$\frac{(2^5-1)-13}{+1}$$

Negate $13_{10} = 01101_2$ using 5-bit two's complement.

$$2^5 - 13$$

① Flip 10010

② Add 1 $(10011)_2 = -13$

Negate -13_{10} using 5-bit two's complement.

① Flip 01100

② Add 1 $(01101)_2 = +13$

6.5 Two's complement representation

4-bit binary	Flip	Add 1	Decimal
0000			0
0001			⋮
⋮			
0111			7 ← Max 2^3-1
1000	0111	1000	-8 ← weird 2's complement, Min -2^3
1001	0110	0111	-7
1010	0101	0110	-6
1011			⋮
1100			
1101			
1110			
1111	0000	0001	-1

n-bit 2's complement

Max $2^{n-1}-1$

Min -2^{n-1}

Problem 9 *Determine the decimal values of the following 2's complement 6-digit numbers :*

1. *01011110*
2. *10010111*

Problem 10 *Convert the decimal numbers -17 and +23 into the 6-digit two's complement binary numbers and try adding them. What adjustments will you need to make to get the right result's (23-17=6) in binary representation.*

$+17_{10} = 010001$

a) Flip 101110

b) Add 1 $\quad 1$

$-17_{10} = 101111$

$23_{10} = 010111$

$$\begin{array}{rcl} & & 1\,11111 \\ 23 & = & 010111 \\ -17 & = & 101111 \\ \hline & & 000110 \end{array} = +6$$

Problem 11 *Convert the decimal numbers 73, 23, -17, and -163 into signed 8-bit numbers in the following representations:*

1. *Sign and magnitude*
2. *1's complement*
3. *2's complement*

6.6 Arithmetic overflow

Problem 12 *Consider addition of 4-digit two's complement binary numbers*

Min = −8

Max = 7

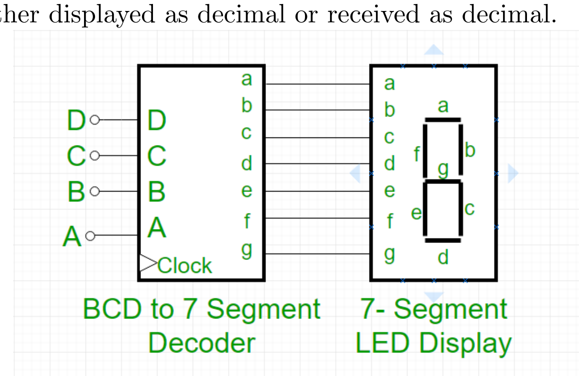
1. $1010_2 + 1101_2$
2. $1011_2 + 1100_2$

$$\begin{array}{r} -6 \\ -3 \\ \hline \end{array} \qquad \begin{array}{r} \boxed{1}\,0000 \\ 1010 \\ 1101 \\ \hline 0111 \end{array} \quad \begin{array}{l} \xrightarrow{\text{Flip}} 0101 \xrightarrow{+1} 0110 \\ \longrightarrow 0010 \xrightarrow{+1} 0011 \end{array} \qquad \begin{array}{r} \boxed{0}\,1000 \\ 0110 \\ 0011 \\ \hline 1001 \end{array}$$

In which of the two case overflow happens? Can you come up with a rule to "easily" detect overflow?

7 Binary coded decimal

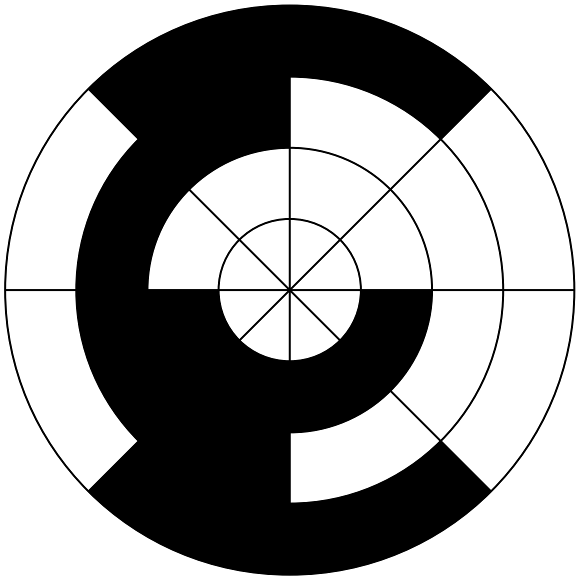
In Binary coded decimal (BCD), each decimal digit is represented by 4 bits. For example, 1047 = $(0001\_0000\_0100\_0111)_{\mathrm{BCD}}$. It is useful in input-output applications where the number has to be either displayed as decimal or received as decimal.



Problem 13 *Convert 11, 23, 35, 57 and 103897 to BCD?*

8 Gray code

A sequence of binary numbers where only one bit changes when the number increases by 1. It is helpful in applications like wheel encoders



Problem 14 *Write all possible 3-bit binary numbers in gray-code*