Lecture 02
Codes and Complements

ECE09 – Digital Electronics 1: Logic Circuits and Switching Theory

Engr. Zoren P. Mabunga, M.Sc

Codes

- When numbers, letters or words are represented by a special group of symbols.
- Convenient in representing long and complicated numbers or words
  1. Straight Binary Coding
  2. Binary Coded Decimal (BCD)
  3. Gray Code
  4. Excess-3 Code
  5. American Standard Code for Information Interchange (ASCII)

Codes

- **Straight Binary Coding**
  - Decimal number is represented by its equivalent binary number.
- **Binary Coded Decimal (BCD)**
  - Each digit of decimal number is represented by its binary equivalent.
  - The main advantage of BCD system is that it is a fast and efficient system to convert the decimal numbers into binary numbers as compared to the pure binary system.

BCD to Decimal

Decimal digits	Weighted 4-bit BCD code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Codes

- **Excess-3 Code**
  - The Excess-3 (XS-3) BCD code does not use the principle of positional weights into consideration while converting the decimal numbers to 4-bit BCD system. Therefore, we can say that this code is a **non-weighted** BCD code.
  - In this code, the decimal number is transformed to the 4-bit BCD code by first adding 3 to all the digits of the number and then converting the excess digits, so obtained, into their corresponding 8421 BCD code.

Excess-3 to Decimal

Decimal digits	Excess-3 code
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Codes

- **Gray Code**
  - Also known as Reflected Binary Code
  - Named after Frank Gray
  - Unweighted Code
  - Largely used in mechanical switching systems.
  - Use to represent digital data that have been converted from analog.
  - Advantage of the Gray code is that only one bit in the code group changes when going from one step to the next.

Binary to Gray Code Conversion

STEP 1: Copy the MSB of the binary number.

STEP 2: Add the MSB to the next bit of the binary number, record the sum and neglect the carry.

STEP 3: Repeat the process.

Gray Code to Binary Conversion

STEP 1: Copy the MSB of the gray code.

STEP 2: Add the MSB of the binary number to the next bit of the gray code, record the sum and neglect the carry.

STEP 3: Repeat the process.

Gray Code to Decimal

Decimal Number	4-Bit Gray Code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101

Codes

- American Standard Code for Information Interchange (ASCII)
  - A 7 bit code
  - 128 possible characters

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
0	0	NUL	16	10	DLE	32	20	(space)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	

Binary Arithmetic

- Each binary arithmetic operation has an associated set of rules that should be adhered to while carrying out that operations.
- The binary arithmetic operations are usually simpler to carry out as compared to the decimal operations because one needs to deal with only two digits, 0 and 1, in the binary operations.
- The different binary arithmetic operations performed in a computer system are:

- Binary addition
- Binary multiplication
- Binary subtraction
- Binary division

Binary Addition

- Like decimal system, we can start the addition of two binary numbers column-wise from the right most bit and move towards the left most bit of the given numbers. However, we need to follow certain rules.

A	B	A+B	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

The carry, if it is generated, while performing the binary addition in a column would be forwarded to the next most significant column.

Binary Addition

- Examples
  - Perform the binary addition of the binary numbers 101010 and 010011:

$$\begin{array}{cccccccc} & 1 & 0 & 1 & 0 & 1 & 0 \\ + & 0 & 1 & 0 & 0 & 1 & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 & 1 \end{array}$$

Binary Addition

- We can also perform the binary addition on more than two binary numbers.

A	B	C	A+B+C	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- Perform the binary addition operation on the following three numbers: 0010, 0001, 0111.

$$
\begin{array}{rcccc}
 & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 1 \\
+ & 0 & 1 & 1 & 1 \\
\hline
 & 1 & 0 & 1 & 0
\end{array}
$$

Binary Multiplication

- The multiplication of two binary numbers can be carried out in the same manner as the decimal multiplication.
- Unlike decimal multiplication, only two values are generated as the outcome of multiplying the multiplication bit by 0 or 1 in the binary multiplication. These values are either 0 or 1.
- The binary multiplication can also be considered as repeated binary addition. Therefore, the binary multiplication is performed in conjunction with the binary addition operation.

Binary Multiplication

A	B	A×B
0	0	0
0	1	0
1	0	0
1	1	1

Perform the binary multiplication of the decimal numbers 12 and 10.

The equivalent binary representation of the decimal number 12 is 1100.
The equivalent binary representation of the decimal number 10 is 1010.

$$
\begin{array}{cccccccc}
 & & & 1 & 1 & 0 & 0 \\
 & & \times & 1 & 0 & 1 & 0 \\
\hline
 & & & 0 & 0 & 0 & 0 \\
 & & 1 & 1 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & & \\
1 & 1 & 0 & 0 & & & \\
\hline
1 & 1 & 1 & 1 & 0 & 0 & 0
\end{array}
$$

Binary Subtraction

- The binary subtraction is performed in the same way as the decimal subtraction. Like binary addition and binary multiplication, binary subtraction is also associated with a set of rules that need to be followed while carrying out the operation.

A	B	A-B	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Example

- Perform the binary subtraction of the following numbers: 10101 and 01110

		1	1	1	
	1	0	1	0	1
-	0	1	1	1	0
	0	0	1	1	1

Binary Division

- Binary division is also performed in the same way as we perform decimal division. Like decimal division, we also need to follow the binary subtraction rules while performing the binary division. The dividend involved in binary division should be greater than the divisor. The following are the two important points, which need to be remembered while performing the binary division.
- If the remainder obtained by the division process is greater than or equal to the divisor, put 1 in the quotient and perform the binary subtraction.
- If the remainder obtained by the division process is less than the divisor, put 0 in the quotient and append the next most significant digit from the dividend to the remainder.

Example

- Perform the binary division of the decimal numbers 18 and 8.

The equivalent binary representation of the decimal number 18 is 10010.

The equivalent binary representation of the decimal number 8 is 1000.

```
1 0 0 0 ) 1 0 0 1 0 ( 1 0  → Quotient
          1 0 0 0
          0 0 0 1 0
          0 0 0 0 0
          0 0 0 1 0  → Remainder
```

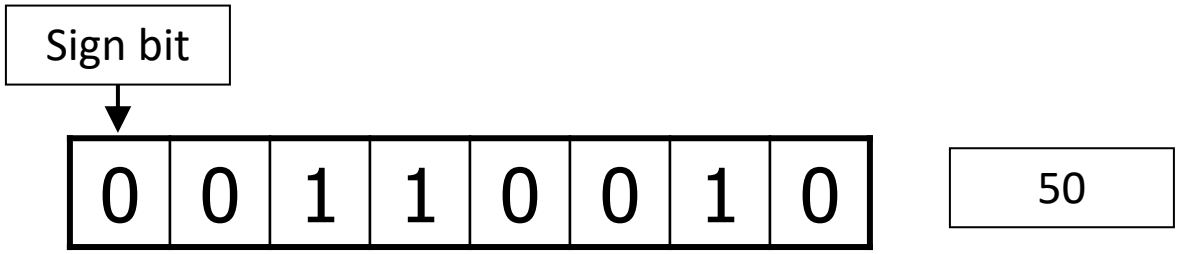
Signed/Unsigned Numbers

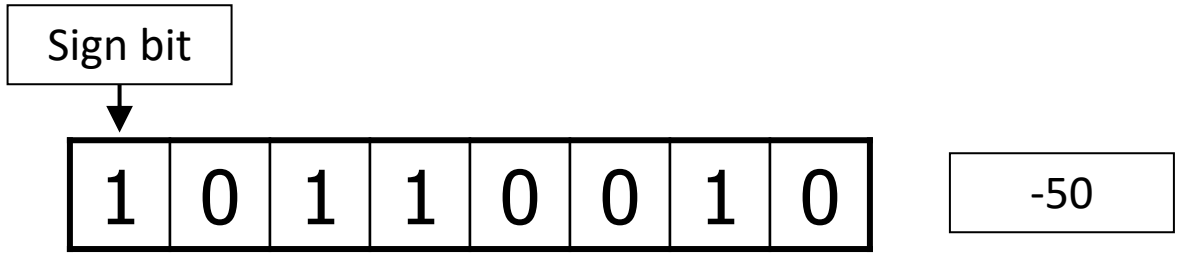
- The **unsigned binary number** is the number with a magnitude of either zero or greater than zero and are usually represented using the unsigned-magnitude representation, which only represents the **magnitude of the numbers**.
- This type of representation does not take the sign of the binary numbers into consideration while representing these numbers.

Signed/Unsigned Numbers

- The **signed binary numbers** are the numbers that are always associated with a sign. This sign helps in identifying whether the given binary number is a positive quantity or a negative quantity.
- Signed-magnitude representation is a method used in the computer system for representing the signed binary numbers. In this method, an **extra bit** called **sign bit** is associated with the magnitude of the given number. This sign bit is used to indicate whether the given binary number is positive or negative. The value of the sign bit is **0 for the positive** numbers and **1 for the negative** numbers.

Signed/Unsigned Numbers





Complements

- Used in digital computers to simplify the subtraction operation and for logical manipulation.

Two Types of Complements:

1. r's complement or radix complement
2. (r – 1)'s complement or diminished radix complement

Radix Complement

- The r's complement of an "n"-digit number "N" in base "r" is defined as $r^n - N$ for N not equal to zero and as 0 for N = 0.

- Examples:

1. Find the 10's complement of 2389
2. Find the 2's complement of 101100

Diminished Radix Complemenet

- Given a number “N” in base “r” having “n” digits, the (r-1)’s complements of N is defined as $(r^n - 1) – N$.

- Examples:

1. Find the 9’s complement of 546700
2. Find the 1’s complement of 1011000

Binary Subtraction Using 1’s Complement

- **STEP 1:** Convert number to be subtracted to its 1’s complement form.
- **STEP 2:** Perform the addition.
- **STEP 3:** If the final carry is 1, then add it to the result obtained in step 2. If the final carry is 0, the result of step 2 is negative and in 1’s complement form.

Example

- Perform the subtraction in binary of 12 and 5 using 1’s complement.

Example

- Perform the subtraction in binary of 5 and 12 using 1’s complement.

Binary Subtraction Using 2's Complement

- **STEP 1:** Convert number to be subtracted to its 2's complement form.
- **STEP 2:** Perform the addition.
- **STEP 3:** If the final carry is 1, then the result is positive and in its true form. If the final carry is 0, the result of step 2 is negative and in 2's complement form.

Example

- Perform the subtraction in binary of 9 and 4 using 2’s complement.

Example

- Perform the subtraction in binary of 6 and 11 using 2’s complement.