EE200 DIGITAL LOGIC CIRCUIT DESIGN

The material covered in this class will be as follows:

- Binary codes, BCD code, BCD addition
- Other decimal codes
- Gray code
- Error-detecting code (parity bit)
- Alpha-numeric codes, ASCII code

Binary codes:

- Digital systems and circuits work with signals that have only one of two states corresponding to digital 1 and 0.
- Any discrete element of information among a group of quantities (elements) can be represented by a binary code.
- One bit can represent up to two elements (1 or 0).
- A group of 2^n distinct elements requires a minimum of n bits.
- $\therefore$ to code a group of m elements, we need to use n bits such that: $2^n \geq m$.
- Examples:
    - o A group of four elements can be represented by two bit code [00, 01, 10, and 11].
    - o A binary code to represent the decimal digits [0-9], must contain at least four bits because

      $(2^4=16) \geq 10 \geq (2^3=8)$.

Decimal Codes:

Decimal Digit	BCD 8421	Excess-3	84-2-1	2421
0	0000	0011	0000	0000
1	0001	0100	0111	0001
2	0010	0101	0110	0010
3	0011	0110	0101	0011
4	0100	0111	0100	0100
5	0101	1000	1011	1011
6	0110	1001	1010	1100
7	0111	1010	1001	1101
8	1000	1011	1000	1110
9	1001	1100	1111	1111

BCD Addition:

The addition of two BCD digits with a possible carry from the previous less significant pair of digits results in a sum in the range 0 to a maximum of (9+9+1=19). There is a difference in the representation of the sum in binary and in BCD code.

1. If $0 \leq Sum \leq 9$ then, sum in BCD = sum in binary,
2. If $10 \leq Sum \leq 19$ then, sum in BCD consists of 8 bits which is not equal to the sum in binary. The correction needed in the sum to represent it in BCD is the addition of 6 to the binary sum.

4	0100	4	0100	8	1000
+5	0101	+8	1000	+9	1001
9	1001	12	1100	+17	10001
			+0110		+0110
			10010		10111

Signed BCD Addition

Signed decimal numbers in BCD are represented, in a similar way, to the signed numbers in binary. The plus sign is represented by four zeros and the minus by 1001. An example of signed BCD addition follows:

Add the signed decimal numbers (+375) + (-240) using BCD representation.
Solution:

```
+375  →   0000 0011 0111 0101    Add   0000 0011 0111 0101
+240  →   0000 0010 0100 0000
-240  →   1001 0111 0110 0000    +     1001 0111 0110 0000
                                       -------------------
                                        1001 1010 1101 0101
Correct the BCD digits by adding 0110 →      0110 0110
                                       -------------------
                                        1010 0001 0011 0101
Repeat correction                     → 0110
                                       -------------------
                                     1  0000 0001 0011 0101
```

The end carry is discarded and the result is the BCD equivalent of +135

Error Detection Code:

A common method to achieve error detection in binary information transmission from one location to another is by means of a parity bit. A parity bit is defined as an extra bit included with a message to make the total number of 1's transmitted either odd or even.

	Odd Parity		Even Parity
message	P	message	P
0000	1	0000	0
0001	0	0001	1
0010	0	0010	1
0011	1	0011	0
0100	0	0100	1
0101	1	0101	0
0110	1	0110	0
0111	0	0111	1
1000	0	1000	1
1001	1	1001	0
1010	1	1010	0
1011	0	1011	1
1100	1	1100	0
1101	0	1101	1
1110	0	1110	1
1111	1	1111	0

Gray Code:

Decimal equivalent	Binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

ASCII Code [American Standard Code for Information Interchange]:

- Used for handling numbers, letters, characters…etc.
- This type of code is called alpha-numeric code.
- To handle 10 decimal digits, 26 small letters, 26 capital letters, and at least 20 other characters, the code must cater for at least 82 elements. $(128=2^7) \geq 82 \geq (64=2^6)$.
- Therefore we need seven bits for the code.

The ASCII code is given in the following table.

	$b_7b_6b_5$							
$b_4b_3b_2b_1$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	-	o	DEL