BLE 201

DIGITAL ELECTRONICS

UNIT 4

COMBINATIONAL LOGIC DESIGN

LECTURE 2

Binary Coded Decimal (BCD)

- Binary coded decimal (BCD) is a way to express each of the decimal digits with a binary code.
- There are only ten code groups in the BCD system, so it is very easy to convert between decimal and BCD.
- The BCD code provides an excellent interface to binary systems. Examples of such interfaces are keypad inputs and digital readouts.
- The 8421 code is a type of **BCD** (binary coded decimal) code.
- Binary coded decimal means that each decimal digit, 0 through 9, is represented by a binary code of four bits.
- The designation 8421 indicates the binary weights of the four bits (2^3, 2^2, 2^1, 2^0).
- The 8421 code is the predominant BCD code, and when we refer to BCD, we always mean the 8421 code unless otherwise stated.

Binary Coded Decimal (BCD)

Decimal digit	BCD			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Binary Coded Decimal (BCD)

➢ Invalid Codes

❖ with four bits, sixteen numbers (0000 through 1111) can be represented but that, in the 8421 code, only ten of these are used.

❖ The six code combinations that are not used—1010, 1011, 1100, 1101, 1110, and 1111—are invalid in the 8421 BCD code.

❖ To express any decimal number in BCD, simply replace each decimal digit with the appropriate 4-bit code.

Binary Coded Decimal (BCD)

Example:

Convert each of the following decimal numbers to BCD:

(a) 35 **(b)** 98 **(c)** 170 **(d)** 2469

Solution

(a) $\overbrace{0011}^{3}\overbrace{0101}^{5}$

(b) $\overbrace{1001}^{9}\overbrace{1000}^{8}$

(c) $\overbrace{0001}^{1}\overbrace{0111}^{7}\overbrace{0000}^{0}$

(d) $\overbrace{0010}^{2}\overbrace{0100}^{4}\overbrace{0110}^{6}\overbrace{1001}^{9}$

Binary Coded Decimal (BCD)

- It is equally easy to determine a decimal number from a BCD number.
- Start at the right-most bit and break the code into groups of four bits.
- Then write the decimal digit represented by each 4-bit group.

Convert each of the following BCD codes to decimal:

(a) 10000110 **(b)** 001101010001 **(c)** 1001010001110000

Solution

(a) $\underbrace{1000}_{\downarrow\atop \mathbf{8}}\underbrace{0110}_{\downarrow\atop \mathbf{6}}$ **(b)** $\underbrace{0011}_{\downarrow\atop \mathbf{3}}\underbrace{0101}_{\downarrow\atop \mathbf{5}}\underbrace{0001}_{\downarrow\atop \mathbf{1}}$ **(c)** $\underbrace{1001}_{\downarrow\atop \mathbf{9}}\underbrace{0100}_{\downarrow\atop \mathbf{4}}\underbrace{0111}_{\downarrow\atop \mathbf{7}}\underbrace{0000}_{\downarrow\atop \mathbf{0}}$

Binary Coded Decimal (BCD)

Applications

- Digital clocks, digital thermometers, digital meters, and other devices with seven-segment displays typically use BCD code to simplify the displaying of decimal numbers.
- BCD is sometimes used for arithmetic operations in processors. To represent BCD numbers in a processor, they usually are "packed," so that eight bits have two BCD digits. Normally, a processor will add numbers as if they were straight binary. Special instructions are available for computer programmers to correct the results when BCD numbers are added or subtracted. For example, in Assembly Language, the programmer will include a DAA (Decimal Adjust for Addition)instruction to automatically correct the answer to BCD following an addition.

BCD Addition

- BCD is a numerical code and can be used in arithmetic operations.
- Addition is the most important operation because the other three operations (subtraction, multiplication, and division) can be accomplished by the use of addition.
- Here is how to add two BCD numbers:

  Step 1: Add the two BCD numbers, using the rules for binary addition.

  Step 2: If a 4-bit sum is equal to or less than 9, it is a valid BCD number.

  Step 3: If a 4-bit sum is greater than 9, or if a carry out of the 4-bit group is generated, it is an invalid result. Add 6 (0110) to the 4-bit sum in order to skip the six invalid states and return the code to 8421. If a carry results when 6 is added, simply add the carry to the next 4-bit group.
- An alternative method to add BCD numbers is to convert them to decimal, perform the addition, and then convert the answer back to BCD.

Example

Add the following BCD numbers:

(a) 0011 + 0100 (b) 00100011 + 00010101

(c) 10000110 + 00010011 (d) 010001010000 + 010000010111

Solution

The decimal number additions are shown for comparison.

(a)
$$
\begin{array}{r}
0011 \\
+\,0100 \\
\hline
\mathbf{0111}
\end{array}
\qquad
\begin{array}{r}
3 \\
+\,4 \\
\hline
7
\end{array}
$$

(b)
$$
\begin{array}{rr}
0010 & 0011 \\
+\,0001 & 0101 \\
\hline
\mathbf{0011} & \mathbf{1000}
\end{array}
\qquad
\begin{array}{r}
23 \\
+\,15 \\
\hline
38
\end{array}
$$

(c)
$$
\begin{array}{rr}
1000 & 0110 \\
+\,0001 & 0011 \\
\hline
\mathbf{1001} & \mathbf{1001}
\end{array}
\qquad
\begin{array}{r}
86 \\
+\,13 \\
\hline
\mathbf{99}
\end{array}
$$

(d)
$$
\begin{array}{rrr}
0100 & 0101 & 0000 \\
+\,0100 & 0001 & 0111 \\
\hline
\mathbf{1000} & \mathbf{0110} & \mathbf{0111}
\end{array}
\qquad
\begin{array}{r}
450 \\
+\,417 \\
\hline
\mathbf{867}
\end{array}
$$

Note that in each case the sum in any 4-bit column does not exceed 9, and the results are valid BCD numbers.

Example

Add the following BCD numbers:

(a) 1001 + 0100 **(b)** 1001 + 1001 **(c)** 00010110 + 00010101 **(d)** 01100111 + 01010011

Solution

The decimal number additions are shown for comparison.

$$
\begin{array}{llrll}
\textbf{(a)} & & 1001 & & 9 \\
 & & \underline{+\,0100} & & \underline{+4} \\
 & & 1101 & \text{Invalid BCD number } (>9) & 13 \\
 & & +\,0110 & \text{Add 6} & \\
\cline{2-3}
 & \underbrace{\mathbf{0001}}_{1} & \underbrace{\mathbf{0011}}_{3} & \text{Valid BCD number} &
\end{array}
$$

(b) 1001 + 1001

Solution

(b)

$$\begin{array}{rrlr}
 & 1001 & & 9\\
 & +\;1001 & & +\;9\\ \cline{2-2}\cline{4-4}
1 & 0010 & \text{Invalid because of carry} & 18\\
 & +\;0110 & \text{Add 6} & \\ \cline{1-2}
\underbrace{\mathbf{0001}}_{\downarrow\;1} & \underbrace{\mathbf{1000}}_{\downarrow\;8} & \text{Valid BCD number} &
\end{array}$$

(c) 00010110 + 00010101

Solution

(c)	0001	0110		16
	+ 0001	0101		+ 15
	0010	1011	Right group is invalid (>9), left group is valid.	31
		+ 0110	Add 6 to invalid code. Add carry, 0001, to next group.	
	$\underbrace{\mathbf{0011}}_{\downarrow \atop 3}$	$\underbrace{\mathbf{0001}}_{\downarrow \atop 1}$	Valid BCD number	

(d) 01100111 + 01010011

Solution

(d)		0110	0111		67
		+ 0101	0011		+ 53
		1011	1010	Both groups are invalid (>9)	120
		+ 0110	+ 0110	Add 6 to both groups	
	0001	**0010**	**0000**	Valid BCD number	
	1	2	0		

The Excess-3 Code

- Add 3 to each digit of decimal and convert to 4-bit binary form

Decimal	Binary	+3	Excess-3
0	0000	0011	0011
1	0001	0011	0100
2	0010	0011	0101
3	0011	0011	0110
4	0100	0011	0111
5	0101	0011	1000
6	0110	0011	1001
7	0111	0011	1010
8	1000	0011	1011
9	1001	0011	1100

Sample Problem:

Decimal	3	5	9
Excess-3	0110	1000	1100

Binary codes for the decimal digits

Decimal digit	(BCD) 8421	Excess-3	84-2-1	2421	(Biquinary) 5043210
0	0000	0011	0000	0000	0100001
1	0001	0100	0111	0001	0100010
2	0010	0101	0110	0010	0100100
3	0011	0110	0101	0011	0101000
4	0100	0111	0100	0100	0110000
5	0101	1000	1011	1011	1000001
6	0110	1001	1010	1100	1000010
7	0111	1010	1001	1101	1000100
8	1000	1011	1000	1110	1001000
9	1001	1100	1111	1111	1010000

The Gray Code

- The **Gray code** is unweighted and is not an arithmetic code; that is, there are no specific weights assigned to the bit positions.
- The important feature of the Gray code is that *it exhibits only a single bit change from one code word to the next in sequence.*
- This property is important in many applications, such as shaft position encoders, where error susceptibility increases with the number of bit changes between adjacent numbers in a sequence.

Four bit Gray code

Binary	Gray
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

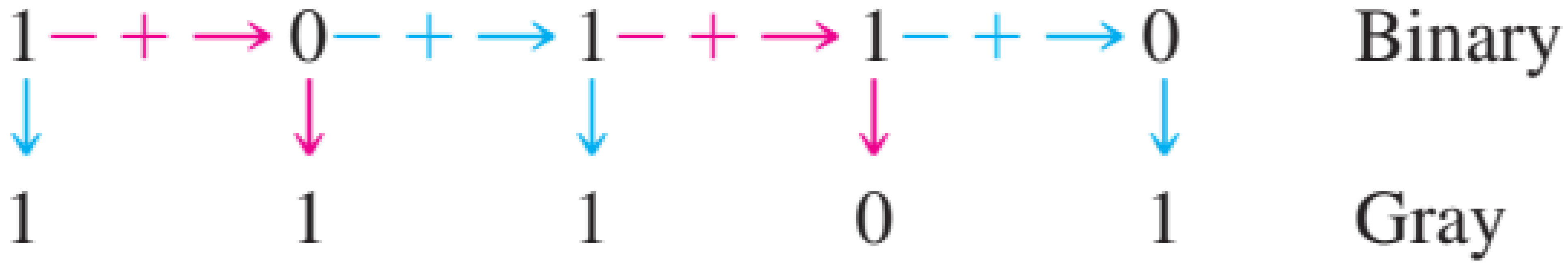
Binary-to-Gray Code Conversion

- Conversion between binary code and Gray code is sometimes useful.
- The following rules explain how to convert from a binary number to a Gray code word:

  1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.

  2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

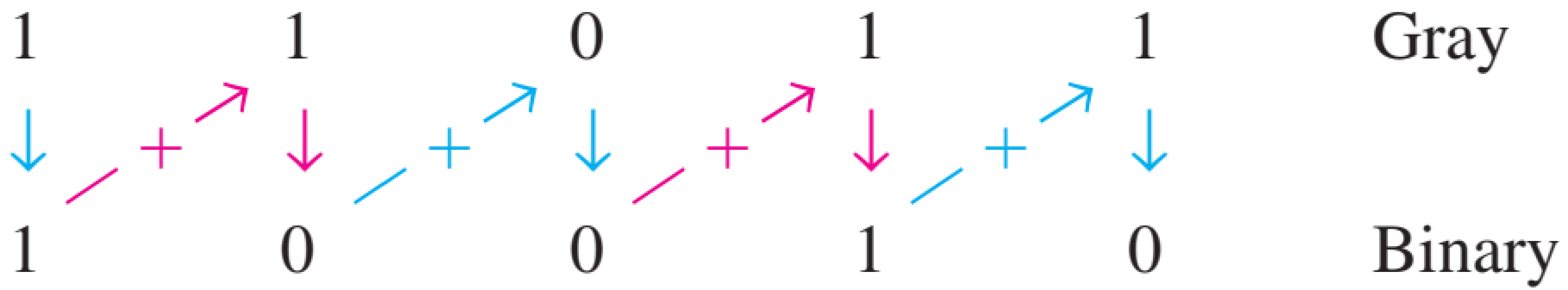
- For example, the conversion of the binary number 10110 to Gray code is as follows:



- The Gray code is 11101.

Gray-to-Binary Code Conversion

- To convert from Gray code to binary, use a similar method; however, there are some differences. The following rules apply:
  1. The most significant bit (left-most) in the binary code is the same as the corresponding bit in the Gray code.
  2. Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.
- For example, the conversion of the Gray code word 11011 to binary is as follows:



- The binary number is 10010.

Alphanumeric Codes

- In order to communicate, you need not only numbers, but also letters and other symbols.
- In the strictest sense, alphanumeric codes are codes that represent numbers and alphabetic characters (letters).
- Most such codes, however, also represent other characters such as symbols and various instructions necessary for conveying information.
- At a minimum, an alphanumeric code must represent 10 decimal digits and 26 letters of the alphabet, for a total of 36 items.
- This number requires six bits in each code combination because five bits are insufficient ($2^5 = 32$).
- There are 64 total combinations of six bits, so there are 28 unused code combinations.
- Obviously, in many applications, symbols other than just numbers and letters are necessary to communicate completely.
- You need spaces, periods, colons, semicolons, question marks, etc.
- You also need instructions to tell the receiving system what to do with the information.
- This should give you an idea of the requirements for a basic alphanumeric code. The ASCII is a common alphanumeric code.

ASCII – American Standard Code for Information Interchange

Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal
NUL (null)	0	0	Space	20	32	@	40	64	.	60	96
Start Heading	1	1	!	21	33	A	41	65	a	61	97
Start Text	2	2	“	22	34	B	42	66	b	62	98
End Text	3	3	#	23	35	C	43	67	c	63	99
End Transmit.	4	4	$	24	36	D	44	68	d	64	100
Enquiry	5	5	%	25	37	E	45	69	e	65	101
Acknowlege	6	6	&	26	38	F	46	70	f	66	102
Bell	7	7	`	27	39	G	47	71	g	67	103
Backspace	8	8	(	28	40	H	48	72	h	68	104
Horiz. Tab	9	9	)	29	41	I	49	73	i	69	105
Line Feed	A	10	*	2A	42	J	4A	74	j	6A	106
Vert. Tab	B	11	+	2B	43	K	4B	75	k	6B	107
Form Feed	C	12	,	2C	44	L	4C	76	l	6C	108
Carriage Return	D	13	-	2D	45	M	4D	77	m	6D	109
Shift Out	E	14	.	2E	46	N	4E	78	n	6E	110

Home Assignment

Write short note on the following codes:

- ASCII
- Extended ASCII Characters
- Unicode

END OF LECTURE