Binary Codes

Objectives

In this lesson, you will study:

1. Several binary codes including
   - Binary Coded Decimal (BCD),
   - Error detection codes,
   - Character codes
2. Coding versus binary conversion.

Binary Codes for Decimal Digits

- Internally, digital computers operate on binary numbers.
- When interfacing to humans, digital processors, e.g. pocket calculators, communication is decimal-based.
- Input is done in decimal then converted to binary for internal processing.
- For output, the result has to be converted from its internal binary representation to a decimal form.
- To be handled by digital processors, the decimal input (output) must be coded in binary in a digit by digit manner.
- For example, to input the decimal number **957**, each *digit* of the number is individually *coded* and the number is stored as **1001_0101_0111.**
- Thus, we need a specific code for each of the 10 decimal digits. There is a variety of such decimal binary codes.
- The shown table gives several common such codes.
- One commonly used code is the *Binary Coded Decimal* (**BCD**) code which corresponds to the first 10 binary representations of the decimal digits 0-9.
- The BCD code requires 4 bits to represent the 10 decimal digits.
- Since 4 bits may have up to 16 different binary combinations, a total of 6 combinations will be unused.
- The position weights of the BCD code are 8, 4, 2, 1.
- Other codes (shown in the table) use position weights of 8, 4, -2, -1 and 2, 4, 2, 1.
- An example of a non-weighted code is the *excess-3 code* where digit codes is obtained from their binary equivalent after adding 3. Thus the code of a decimal 0 is 0011, that of 6 is 1001, etc.

Decimal Digit	BCD												Excess-3			
	8	4	2	1	8	4	-2	-1	2	4	2	1				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	0	1	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	1	1	1	0	1	1	1	0	0	0
6	0	1	1	0	1	0	1	0	1	1	0	0	1	0	0	1
7	0	1	1	1	1	0	0	1	1	1	0	1	1	0	1	0
8	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0
U	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0
N	1	0	1	1	0	0	1	0	0	1	1	0	0	0	0	1
U	1	1	0	0	0	0	1	1	0	1	1	1	0	0	1	0
S	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0	1
E	1	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0
D	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	1

Number Conversion versus Coding

- Converting a decimal number into binary is done by repeated division (multiplication) by 2 for integers (fractions) (see lesson 4).
- Coding a decimal number into its BCD code is done by replacing each decimal digit of the number by its equivalent 4 bit BCD code.

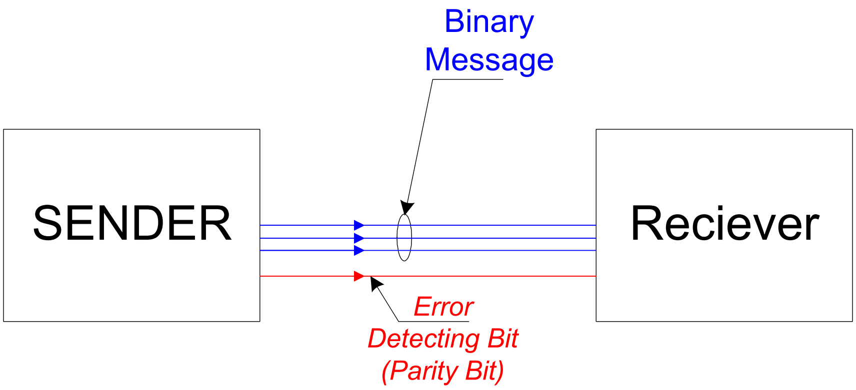
Example Converting $(13)_{10}$ into binary, we get **1101**, coding the same number into BCD, we obtain **00010011**.

Exercise: Convert $(95)_{10}$ into its binary equivalent value and give its BCD code as well.

Answer {$(1011111)_2$, and **10010101**}

Error-Detection Codes

- Binary information may be transmitted through some communication medium, e.g. using wires or wireless media.
- A corrupted bit will have its value changed from 0 to 1 or vice versa.
- To be able to detect errors at the receiver end, the sender sends an extra bit (parity bit) with the original binary message.



- A *parity bit* is an extra bit included with the *n-bit binary message* to make the total number of 1's in this message (*including the parity bit*) either odd or even.
- If the *parity bit* makes the total number of 1's an odd (even) number, it is called odd (even) parity.
- The table shows the ***required*** odd (***even***) ***parity*** for a 3-bit message.

Three-Bit Message			**Odd Parity Bit**	**Even Parity Bit**
X	**Y**	**Z**	**P**	**P**
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

- At the receiver end, an error is detected if the message does not match have the proper parity (odd/even).
- Parity bits can detect the occurrence 1, 3, 5 or any odd number of errors in the transmitted message.

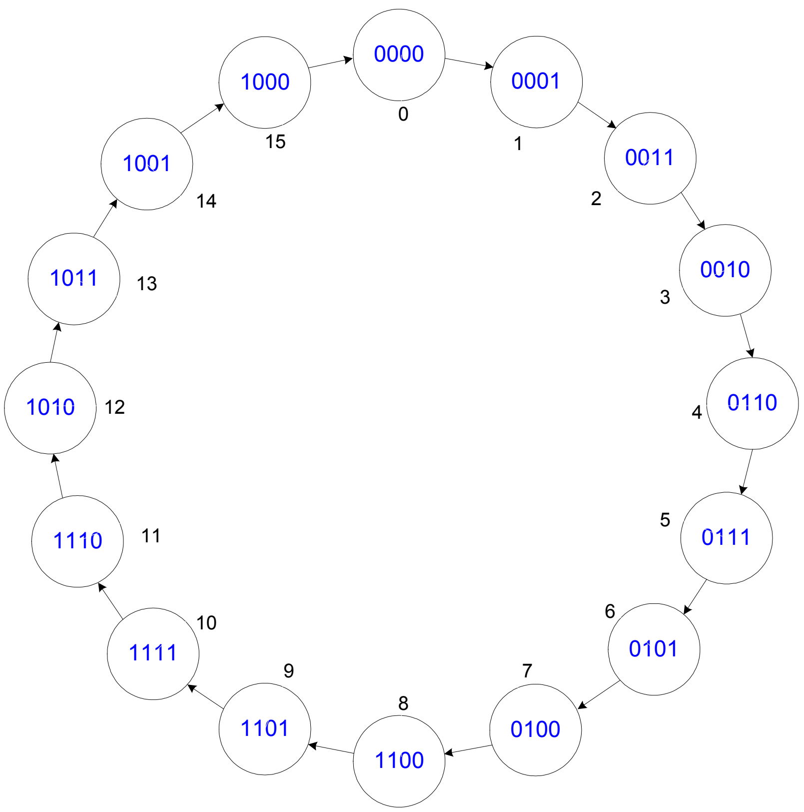
- No error is detectable if the transmitted message has 2 bits in error since the total number of 1's will remain even (or odd) as in the original message.
- In general, a transmitted message with even number of errors cannot be detected by the parity bit.

Error-Detection Codes

- Binary information may be transmitted through some communication medium, e.g. using wires or wireless media.
- Noise in the transmission medium may cause the transmitted binary message to be corrupted by changing a bit from 0 to 1 or vice versa.
- To be able to detect errors at the receiver end, the sender sends an extra bit (*parity bit*).

Gray Code

- The Gray code consist of 16 4-bit code words to represent the decimal Numbers 0 to 15.
- For Gray code, successive code words differ by only one bit from one to the next as shown in the table and further illustrated in the Figure.



Gray Code				Decimal Equivalent
0	0	0	0	0
0	0	0	1	1
0	0	1	1	2
0	0	1	0	3
0	1	1	0	4
0	1	1	1	5
0	1	0	1	6
0	1	0	0	7
1	1	0	0	8
1	1	0	1	9
1	1	1	1	10
1	1	1	0	11
1	0	1	0	12
1	0	1	1	13
1	0	0	1	14
1	0	0	0	15

Character Codes

ASCII Character Code

- ASCII code is a 7-bit code. Thus, it represents a total of 128 characters.

- Out of the 128 characters, there are 94 printable characters and 34 control (non- printable) characters.
- The printable characters include the upper and lower case letters (2*26), the 10 numerals (0-9), and 32 special characters, e.g. @, %, $, etc.
- For example, "A" is at $(41)_{16}$, while "a" is at $(61)\,)_{16}$.
- To convert upper case letters to lower case letters, add $(20)_{16}$. Thus "a" is at $(41)_{16} + (20)_{16} = (61)_{16}$.
- The code of the character "9" at position $(39)_{16}$ is different from the binary number 9 (0001001). To convert ASCII code of a numeral to its binary number value, subtract $(30)_{16}$.

00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o

NUL	Null	FF	Form feed	CAN	Cance
SOH	Start of heading	CR	Carriage return	EM	End o
STX	Start of text	SO	Shift out	SUB	Substi
ETX	End of text	SI	Shift in	ESC	Escap
EOT	End of transmission	DLE	Data link escape	FS	File se
ENQ	Enquiry	DC1	Device control 1	GS	Group
ACK	Acknowledge	DC2	Device control 2	RS	Recor
BEL	Bell	DC3	Device control 3	US	Unit se
BS	Backspace	DC4	Device control 4	SP	Space
HT	Horizontal tab	NAK	Negative acknowledge	DEL	Delete
LF	Line feed	SYN	Synchronous idle		
VT	Vertical tab	ETB	End of transmision block		

Unicode Character Code

- Unicode is a 16-bit character code that accommodates characters of various languages of the world.