

Chapter 3

Digital Codes & Binary Arithmetic

SKEE1223 Digital Electronics

Mun'im/Arif

FKE, Universiti Teknologi Malaysia

September 24, 2015

Overview

- 1 Binary Codes
 - BCD
 - Gray Code
- 2 Alphanumeric Codes
 - ASCII
 - Unicode
- 3 Error Detecting Code
- 4 Other Codes
- 5 Arithmetic
 - Binary Addition
 - Binary Subtraction
 - Binary Multiplication

Binary Coded Decimal (BCD)

- Each decimal digit (0 to 9) is represented by 4 bit binary

Binary	Decimal
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9

Variations of BCD

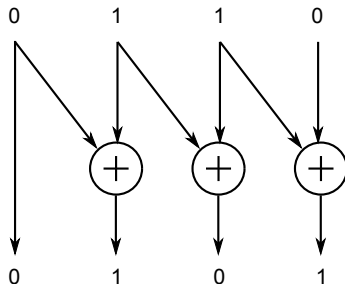
Decimal Value	BCD 8421	BCD 2421	XS-3
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

Gray Code

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

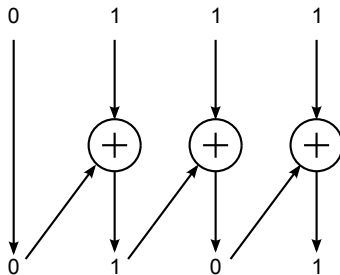
Binary → Gray Code

- MSB of Gray Code is the same MSB in binary
- From left to right, add each adjacent pair of binary code, discard carry



Gray Code to Binary Conversions

- MSB of binary is the same MSB in Gray Code
- From left to right, add each generated binary code with adjacent Gray Code, discard carry



ASCII

- American Standard Code for Information Interchange
- 128 characters, represented by 8-bit binary code with MSB '0'
- The 8-bit code runs from 00_{16} to $7F_{16}$
- The first 32 ASCII characters used for controls such as ESC, new line, space, start of text, etc
- Other characters include letters (upper and lower case), decimal digits, and symbols

ASCII Table

ASCII Table					0_00_0	0_00_1	0_10_0	0_10_1	1_00_0	1_00_1	1_10_0	1_10_1
				상위 3비트	0	1	2	3	4	5	6	7
$b_7b_6b_5$	b_4	b_3	b_2	b_1	하위 4비트							
0	0	0	0	0	NUL ₀	DLE ₁₆	SP ₃₂	0 ₄₈	@ ₆₄	P ₈₀	` ₉₆	p ₁₁₂
0	0	0	1	1	SOH ₁	DC1 ₁₇	! ₃₃	1 ₄₉	A ₆₅	Q ₈₁	a ₉₇	q ₁₁₃
0	0	1	0	2	STX ₂	DC2 ₁₈	" ₃₄	2 ₅₀	B ₆₆	R ₈₂	b ₉₈	r ₁₁₄
0	0	1	1	3	ETX ₃	DC3 ₁₉	# ₃₅	3 ₅₁	C ₆₇	S ₈₃	c ₉₉	s ₁₁₅
0	1	0	0	4	EOT ₄	DC4 ₂₀	\$ ₃₆	4 ₅₂	D ₆₈	T ₈₄	d ₁₀₀	t ₁₁₆
0	1	0	1	5	ENQ ₅	NAK ₂₁	% ₃₇	5 ₅₃	E ₆₉	U ₈₅	e ₁₀₁	u ₁₁₇
0	1	1	0	6	ACK ₆	SYN ₂₂	& ₃₈	6 ₅₄	F ₇₀	V ₈₆	f ₁₀₂	v ₁₁₈
0	1	1	1	7	BEL ₇	ETB ₂₃	' ₃₉	7 ₅₅	G ₇₁	W ₈₇	g ₁₀₃	w ₁₁₉
1	0	0	0	8	BS ₈	CAN ₂₄	(₄₀	8 ₅₆	H ₇₂	X ₈₈	h ₁₀₄	x ₁₂₀
1	0	0	1	9	HT ₉	EM ₂₅) ₄₁	9 ₅₇	I ₇₃	Y ₈₉	i ₁₀₅	y ₁₂₁
1	0	1	0	A	LF ₁₀	SUB ₂₆	* ₄₂	: ₅₈	J ₇₄	Z ₉₀	j ₁₀₆	z ₁₂₂
1	0	1	1	B	VT ₁₁	ESC ₂₇	+ ₄₃	; ₅₉	K ₇₅	[₉₁	k ₁₀₇	{ ₁₂₃
1	1	0	0	C	FF ₁₂	FS ₂₈	, ₄₄	< ₆₀	L ₇₆	\ ₉₂	l ₁₀₈	₁₂₄
1	1	0	1	D	CR ₁₃	GS ₂₉	- ₄₅	= ₆₁	M ₇₇] ₉₃	m ₁₀₉	} ₁₂₅
1	1	1	0	E	SO ₁₄	RS ₃₀	. ₄₆	> ₆₂	N ₇₈	^ ₉₄	n ₁₁₀	~ ₁₂₆
1	1	1	1	F	SI ₁₅	US ₃₁	/ ₄₇	? ₆₃	O ₇₉	_ ₉₅	o ₁₁₁	DEL ₁₂₇

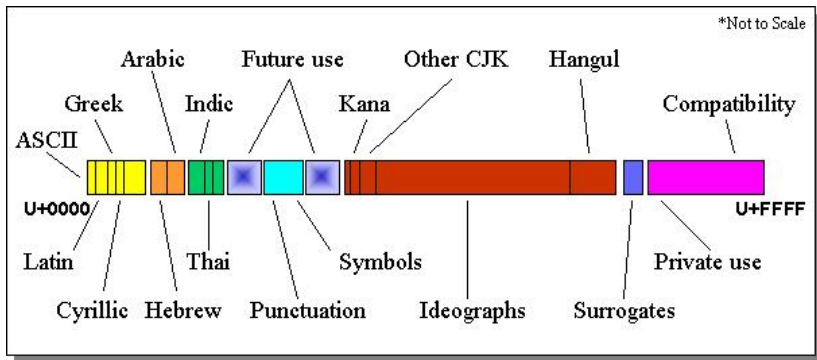
ASCII Example

- Find the ASCII equivalent "ab.12" in binary
- $0110\ 0001\ 0110\ 0010\ 0010\ 1110\ 0011\ 0001\ 0011\ 0010_2$
- A receiver receives the bit sequence:
- $504D544B_{16}$
 - Find the ASCII characters corresponding the transmitted data
 - PMTK

Unicode

- ASCII serves U.S. basic needs only
- Unicode covers all language in all platforms
- 21 bits
- Over one million *codepoints*
- Comes in 3 versions: UTF-8, UTF-16 and UTF-32

Unicode Encoding Layout



UTF-8

- Uses 1, 2, 3, or 4 bytes to encode a character
- “A” is 41 (same as ASCII!)
- Alpha is CE 91
- Katakana “A” is E3 82 A2
- Gothic Ahsa is F0 90 8C B0

Parity

- Simplest form of error correction
- A bit string has odd (even) parity of the number of 1s in the string is odd (even)

Decimal Value	8 b_3	4 b_2	2 b_1	1 b_0	Parity p
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0

Even Parity Example for 7-bit Data

Character	ASCII	ASCII with even parity
S	0110101	00110101
a	1100001	11100001
l	1101100	01101100
a	1100001	11100001
m	1101101	11101101
!	0100001	00100001

Redundant Decimal Codes

- 4 bits is enough to represent 0..9
- Sometimes, it better to “waste” bit to improve error detection or simplify hardware circuits

Decimal Value	2-out-of-5	Biquinary 5043210
0	00011	0100001
1	00101	0100010
2	00110	0100100
3	01001	0101000
4	01010	0110000
5	01100	1000001
6	10001	1000010
7	10010	1000100
8	10100	1001000
9	11000	1010000

Other Codes

- **Morse code**: actually a trinary (3 valued) code, used for ham radio
- **Baudot code**: 5-bit binary code also used for ham radio.
- **Braille**: for the visually impaired
- **EBCDIC** (Extended Binary Coded Decimal Interchange Code): 8-bit code used in IBM mainframes
- **ECC** (Error Correcting Code): General term for many advanced techniques that correct rather than just detect errors (<http://www.eccpage.com/>)

Binary Addition Rules

$A + B$	Carry	Sum
$0 + 0$	0	0
$0 + 1$	0	1
$1 + 0$	0	1
$1 + 1$	1	0

Binary Subtraction Rules

<i>A – B</i>	<i>Borrow</i>	<i>Difference</i>
1 – 0	0	1
1 – 1	0	0
0 – 1	0	0
0 – 0	1	1

Binary Multiplication Rules

$A \times B$	<i>Product</i>
0×0	0
0×1	0
1×0	0
1×1	1

7 x 5 in Binary

$$\begin{array}{r}
 \times \quad \quad \quad 1 \ 1 \ 1 \\
 \quad \quad \quad 1 \ 0 \ 1 \\
 \hline
 \quad \quad \quad 1 \ 1 \ 1 \\
 \quad \quad 0 \ 0 \ 0 \\
 \quad 1 \ 1 \ 1 \\
 \hline
 1 \ 0 \ 0 \ 0 \ 1 \ 1
 \end{array}
 = 35_{10}$$

Multiplicand
Multiplier
Partial products
Product



<https://www.openlearning.com/courses/SKEE1223x>