Chapter 3
Digital Codes & Binary Arithmetic
SKEE1223 Digital Electronics

Mun'im/Arif

FKE, Universiti Teknologi Malaysia

September 24, 2015

Overview

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

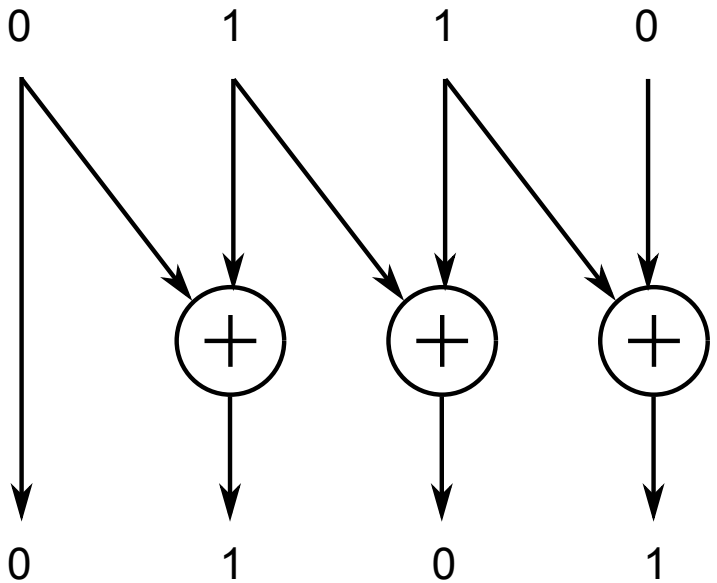
Decimal Value	BCD 8421	BCD 2421	XS-3
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

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

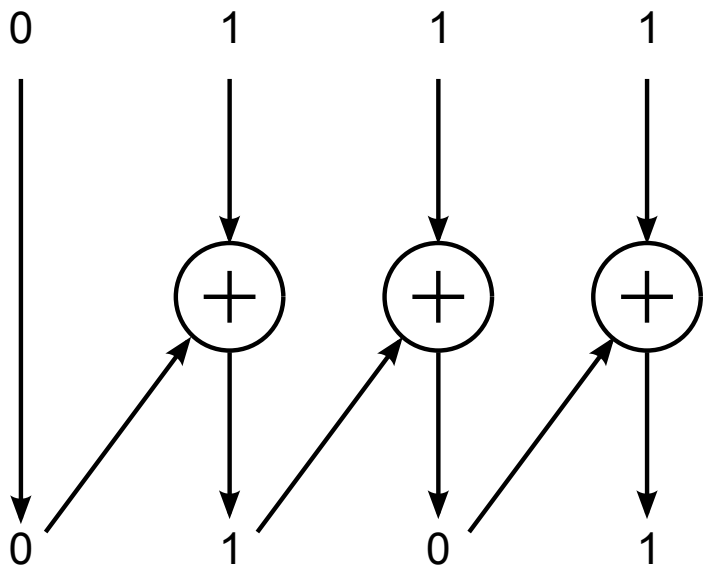
Binary → Gray Code

- 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

$b_7b_6b_5$ ASCII Table					$0\,0\,0$	$0\,0\,1$	$0\,1\,0$	$0\,1\,1$	$1\,0\,0$	$1\,0\,1$	$1\,1\,0$	$1\,1\,1$
b_4	b_3	b_2	b_1	상위 3비트 / 하위 4비트	0	1	2	3	4	5	6	7
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

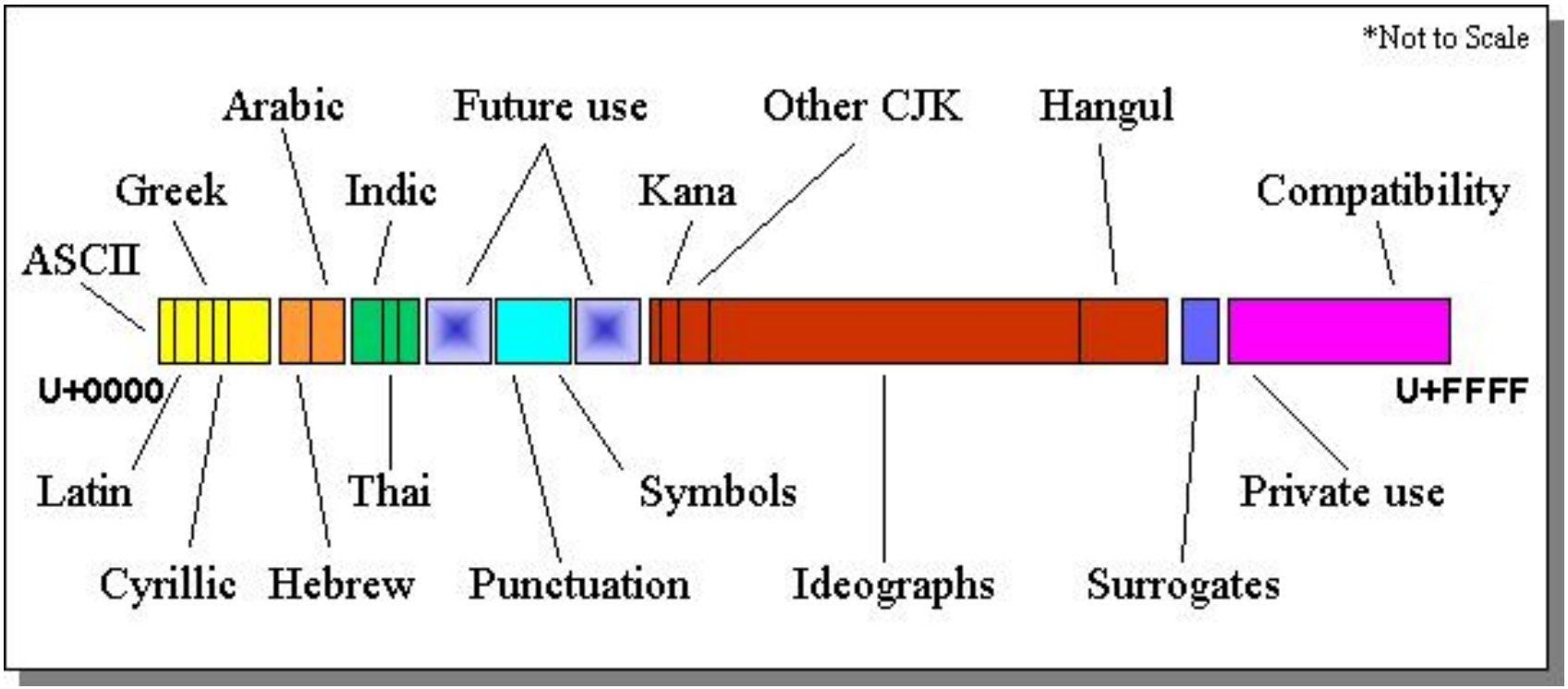
ASCII Example

- 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

6 + 3 in Binary

	1	*1*	*0*				*Carries*
	0	1	1	0	=	6_{10}	*Augend*
+	0	0	1	1	=	3_{10}	*Addend*
	1	0	0	1	=	9_{10}	*Sum*

Binary Subtraction Rules

A − *B*	*Borrow*	*Difference*
1 − 0	0	1
1 − 1	0	0
0 − 1	0	0
0 − 0	1	1

5 - 2 in Binary

	0	*1*	*0*				*Borrows*
	0	1	0	1	=	5_{10}	*Minuend*
+	0	0	1	0	=	2_{10}	*Subtrahend*
	0	0	1	1	=	3_{10}	*Difference*

Binary Multiplication Rules

$A \times B$	Product
0×0	0
0×1	0
1×0	0
1×1	1

7 x 5 in Binary

			1	1	1	=	7_{10}	*Multiplicand*
×			1	0	1	=	5_{10}	*Multiplier*
			1	1	1			
		0	0	0				*Partial products*
	1	1	1					
1	0	0	0	1	1	=	35_{10}	*Product*

31 ÷ 5 in Binary

Divisor								*Quotient*
1	0	1	1	1	1	1	1	*Dividend*
			1	0	1			
				1	0	1	1	
				1	0	1		
							1	*Remainder*

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