

EE2000 Logic Circuit Design

Number Systems & Arithmetic

1

Outline

- 1.1 Number Systems
 - Introduction to decimal, binary, octal, hexadecimal
 - Conversion between these number systems
- 1.2 Binary Arithmetic
 - Representation of negative numbers
 - Addition and subtraction of binary numbers
- 1.3 Binary Codes for Decimal Digits
 - BCD, Gray codes, ASCII
- 1.4 Parity and Error Correction

2

Number Systems

- Decimal (base 10)
E.g. 1, 2, 3, 4, 5, ... 10, 11, 12, ...
- Binary (base 2)
E.g. 000, 001, 010, 011, ...
- Octal (base 8)
E.g. 00, 01, 02, ..., 07, 10, 11, ...17, 20, ...
- Hexadecimal (base 16)
E.g. 1, 2, 3, ..., 9, A, B, C, D, E, F, 10, 11, ...

3

Number	Decimal	Binary	Octal	Hexadecimal
Zero	0	0	0	0
One	1	1	1	1
Two	2	10	2	2
Three	3	11	3	3
Four	4	100	4	4
Five	5	101	5	5
Six	6	110	6	6
Seven	7	111	7	7
Eight	8	1000	10	8
Nine	9	1001	11	9
Ten	10	1010	12	A
Eleven	11	1011	13	B
Twelve	12	1100	14	C
Thirteen	13	1101	15	D
Fourteen	14	1110	16	E
Fifteen	15	1111	17	F
Sixteen	16	10000	20	10
Seventeen	17	10001	21	11

4

Number with radix

Given a number with radix (or base) r (r is usually a positive integer, but not necessarily)

$$a_n a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m+1} a_{-m}$$

where $a_i = x$, $x \in (0, 1, \dots, r)$
its decimal value is given by

$$a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + \dots + a_2 r^2 + a_1 r + a_0$$

n : number of digits

r : radix (base)

a_n : coefficients (digits)

5

Polynomial Form

For decimal number, $r = 10$

e.g. $(7392)_{10}$

$$= 7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$$

For binary number, $r = 2$

e.g. $(1101.011)_2$

$$= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

For octal number, $r = 8$

e.g. $(527)_8$

$$= 5 \times 8^2 + 2 \times 8^1 + 7 \times 8^0$$

6

Conversion

- Binary to Decimal

For binary number, $r = 2$

e.g. $(1101.01)_2$

$$= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$= 8 + 4 + 0 + 1 + 0 + 0.25$$

$$= (13.25)_{10}$$

7

Conversion

- Decimal to Binary

- Method 1: Subtraction

- Subtract from the largest power of 2 that less than or equal to that number
- Put a 1 in the corresponding position of the binary equivalent
- Repeat the subtraction procedure with the difference until the difference becomes 0

- Method 2: Division

- Divide the number by 2
- The remainder (either 0 or 1) gives the least significant bit
- Divide the quotient by 2 repeatedly until the quotient becomes 0

8

Division Method Decimal to Binary

• $(98)_{10}$

$98 / 2 = 49$ Remainder = 0

$49 / 2 = 24$ Remainder = 1

$24 / 2 = 12$ Remainder = 0

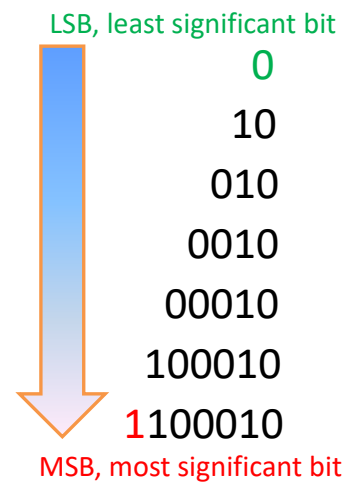
$12 / 2 = 6$ Remainder = 0

$6 / 2 = 3$ Remainder = 0

$3 / 2 = 1$ Remainder = 1

$1 / 2 = 0$ Remainder = 1

Ans = $(1100010)_2$



9

Division Method Decimal to Octal /hexadecimal

• $(611)_{10}$

$611 / 8 = 76$ Remainder = 3

$76 / 8 = 9$ Remainder = 4

$9 / 8 = 1$ Remainder = 1

$1 / 8 = 0$ Remainder = 1

Ans = $(1143)_8$

3
43
143
1143

$611 / 16 = 38$ Remainder = 3

$38 / 16 = 2$ Remainder = 6

$2 / 16 = 0$ Remainder = 2

Ans = $(263)_{16}$

3
63
263

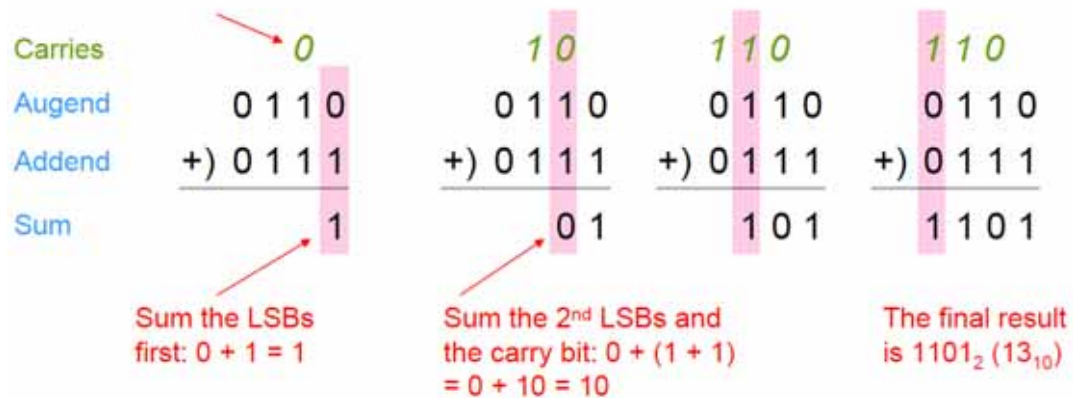
10

Binary Arithmetic

■ Compute the sum of $(0110)_2$ and $(0111)_2$

■ $(0110)_2 = (6)_{10}$

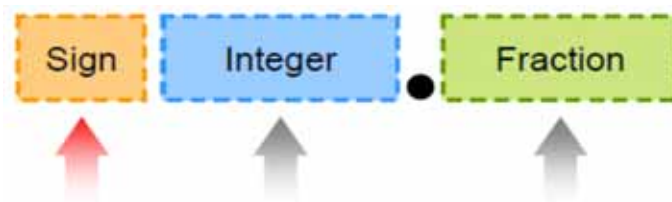
■ $(0111)_2 = (7)_{10}$



11

Signed Number Representation

- Negative binary number?
- 1) Sign-magnitude format



- 2) Complement format

12

Sign-Magnitude Binary

- The MSB be a sign bit
 - 0 means positive “+”
 - 1 means negative “-”

For example:

4-bit system (+5 = 0101; -5 = 1101)

Last 3 bits used for magnitude

13

Problems of Sign-Magnitude

- Problem 1) Result incorrect for addition of negative numbers
 - e.g. Compute the sum of +5 and -3 using 4-bit binary system
 - +5 = 0101₂
 - -3 = 1011₂
 - Result = 10000₂ (5 bits? 10000₂ ≠ 2₁₀) Overflow!!
- Problem 2) Two zeros
 - 0000 (positive zero)
 - 1000 (negative zero)
 - Any difference between them? Cause confusion!

14

Two's Complement

4-bit binary system

a	$0\ a$	a	$2^n - a$
Decimal	2's complement	Decimal	2's complement
0	0000		
+1	0001	-1	1111
+2	0010	-2	1110
+3	0011	-3	1101
+4	0100	-4	1100
+5	0101	-5	1011
+6	0110	-6	1010
+7	0111	-7	1001
		-8	1000

4-bit system

(+5 = 0101; -5 = 1101) Sign-Magnitude Binary

(+5 = 0101; -5 = 1011) Two's Complement Binary

15

-ve Dec $\rightarrow$ Bin (2's Complement)

- Negative number, $-a$, is stored as $2^n - a$
- Not convenient to compute $2^n - a$
- An easier way to find the 2's complement
 - Step 1) Find the binary equivalent of the magnitude
 - Step 2) Complement each bit (i.e. 0's to 1's, 1's to 0's) (this is 1's complement)
 - Step 3) Add 1 (this is 2's complement)

16

Positive to negative conversion

Find the 2's complement representation of -5 and -0 in a 4-bit binary system

	-5	-0
Step 1)	$5_{10} = 0101_2$	$0_{10} = 0000_2$
Step 2)	1010	1111
Step 3)	$\begin{array}{r} 1010 \\ +) \quad 1 \\ \hline 1011 \end{array}$	$\begin{array}{r} 1111 \\ +) \quad 1 \\ \hline 10000 \end{array}$
Answer =	1011_2	0000_2

The carry bit can be ignored.

17

Negative to positive conversion

Find the decimal value of $(1011)_2$ which is a 2's complemented binary number

	1011_2
Step 1)	0100_2
Step 2)	$\begin{array}{r} 0100 \\ +) \quad 1 \\ \hline 0101 \end{array}$
Step 3)	$0101_2 = 5_{10}$
Magnitude	5
Answer =	-5_{10}

18

Addition of 2's complement

- Now we know how to represent negative binary numbers
- How to do addition?
 - (a) $(-5) + 3$, (b) $(-5) + 7$

	(a)	(b)
Carries	0 1 1	1 1 1
Augend	- 5 1 0 1 1	- 5 1 0 1 1
Addend	+ 3 0 0 1 1	+ 7 0 1 1 1
Sum	- 2 1 1 1 0	2 ~~1~~ 0 0 1 0

The results are correct now!

The carry bit can be ignored.

19

Binary Subtraction

- $A - B = A + (-B) = A + (2\text{'s complement of } B)$

Compute the result of $5 - 7$ for a 4-bit binary system

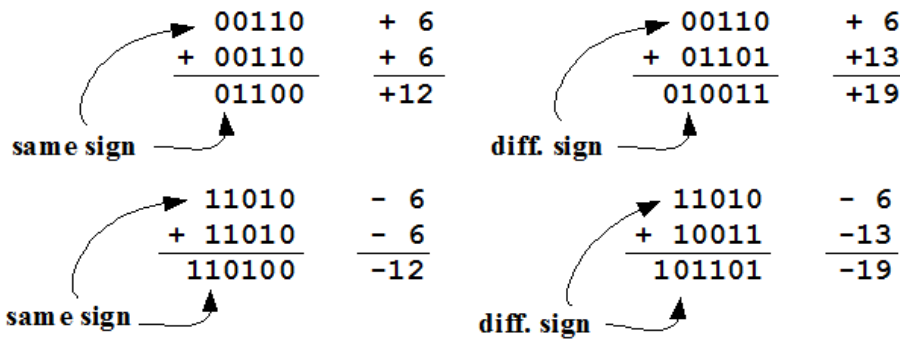
	5 - 7
Step 1)	5 + (-7)
Step 2)	$ \begin{array}{r} 0101 \\ +) 1001 \\ \hline 1110 \end{array} $
Step 3)	1110_2
Answer =	-2_{10}

20

Problem on Addition using 2's complement

- using 2's complement number for addition:
 - (+ve no) + (-ve no) $\Rightarrow$ overflow never occur
 - (+ve no) + (+ve no) or (-ve no) + (-ve no) $\Rightarrow$ overflow may occur

e.g. (note that the result is of the same bit length as the two numbers)



overflow does not occur

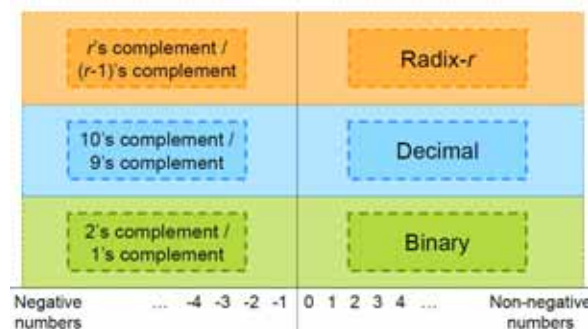
overflow occurs

Therefore, the addition of 2's complement is excellent for (+ve) and (-ve) numbers arithmetic.

21

Radix Complement

- There are 2 types of complement for any radix- r number system
 - r 's complement and $(r-1)$'s complement
 - For n -digit radix- r number system
 - The r 's complement number a is $r^n - a$
 - The $(r-1)$'s complement number a is $r^n - a - 1$



22

Binary Arithmetic

- Addition (+)
- Subtraction (-)
- Multiplication (x)
 - Series of addition
- Division ($\div$)
 - Series of subtraction

23

Binary Coded Decimal

- There are several BCD
 - 8421 code
 - 5421 code
 - 2421 code
 - Excess 3 code
 - 2-of-5 code

To perform arithmetic operations directly with BCD

When communicating with computer systems

Input: convert decimal no. to binary

Process: perform arithmetic calculations in binary

Output: convert binary result back to decimal

To develop code for human-computer communications

Digits: **Binary Coded Decimal (BCD)**, Gray Code

Alphanumeric: ASCII Code

24

8421 Code

- 8421 code is a kind of weighted code
- Use 4 bits to represent a decimal digit
- 8, 4, 2, 1 are the weight of the bits

- Code: $a_3a_2a_1a_0$

- Value: $(8 \times a_3) + (4 \times a_2) + (2 \times a_1) + (1 \times a_0)$

- Example

- Code: 0110

- Value: $(8 \times 0) + (4 \times 1) + (2 \times 1) + (1 \times 0)$

- $= 4 + 2 = 6$

Value (decimal digit)	8421 code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
unused	1010
	1011
	1100
	1101
	1110
	1111

25

8421 BCD Example

- 739 would be stored as

- 0111 0011 1001 (12 bits)
 - 7 3 9

- Note that BCD $\neq$ binary conversion

- For binary conversion

- 739 read as "seven hundred and thirty nine"

- $739_{10} = 1011100011_2$ (10 bits)

- For BCD conversion

- 739 read as seven-three-nine (3 separate digits)

26

Three Weighted Codes

Value (decimal digit)	8421 code	5421 code	2421 code
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111
unused	1010	0101	0101
	1011	0110	0110
	1100	0111	0111
	1101	1101	1000
	1110	1110	1001
	1111	1111	1010

The code are the same for the first 5 digits

$(5 \times 1) + (4 \times 0) + (2 \times 1) + (1 \times 1) = 5 + 2 + 1 = 8$

$(2 \times 1) + (4 \times 1) + (2 \times 1) + (1 \times 0) = 2 + 4 + 2 = 8$

4-bit codes have 16 combinations, but we just use 10 of them. The remaining 6 codes are unused and undefined

27

Properties of 2421 code

- A self-complementing code (9's complement)
 - The complement of 0 is 9 (0000 $\Leftrightarrow$ 1111)
 - The complement of 1 is 8 (0001 $\Leftrightarrow$ 1110)
 - The complement of 2 is 7 (0010 $\Leftrightarrow$ 1101)
 - The complement of 3 is 6 (0011 $\Leftrightarrow$ 1100)
 - The complement of 4 is 5 (0100 $\Leftrightarrow$ 1011)

28

Excess 3 Code (XS3)

Value (decimal digit)	8421code	Excess 3 code
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100
unused	1010	0000
	1011	0001
	1100	0010
	1101	1101
	1110	1110
	1111	1111

Excess 3 code is a shifted 8421 code

XS3's 0 = 8421 code's 3

XS3's 1 = 8421 code's 4

...

9's complement code:

Complement of 0 is 9 (0011 $\leftrightarrow$ 1100)

Complement of 1 is 8 (0100 $\leftrightarrow$ 1011)

...

Complement of 4 is 5 (0111 $\leftrightarrow$ 1000)

Evenly distribution of 1s and 0s

8421 code: the MSB has eight 0s,
only two 1s

XS3: each bit has exactly five 1s and
five 0s

29

2-of-5 code

Value (decimal digit)	2 of 5 Code
0	11000
1	10100
2	10010
3	10001
4	01100
5	01010
6	01001
7	00110
8	00101
9	00011
unused	the rest of the 22 patterns with 0, 1, 3, 4 or 5 1's

Use 5 bits to represent each digit

exactly two 1s and three 0s for all code

Advantage: capable for error detection

If an error is made in just one of the bits
during storage or transmission (the result
will contain either one or three 1's), can
be detected as an error

30

BCD Summary

- Weighted codes
 - 8421, 5421, 2421 codes
- Non-weight codes
 - XS3, 2-of-5 code
- Self-complement
 - 2421, XS3 codes
- Able to detect error
 - 2-of-5 code

31

BCD Drawbacks

- Inefficiency of storage
 - For weighted codes and XS3 code
 - Used 10 out of 16 combinations
 - 6 combinations waste
 - For 2-of-5 code
 - Used 10 out of 32 combinations
 - 22 combinations wasted
- Complex arithmetic calculations

32

8421 Code Addition

Compute the result of digit 7 plus digit 5 using 8421 code

	7 + 5	
Step 1)	$ \begin{array}{r} 0111 \\ +) 0101 \\ \hline 1100 \end{array} $	
Step 2)	1100 = unused code	
Step 3)	$ \begin{array}{r} 1100 \\ +) 0110 \\ \hline 0001\ 0010 \end{array} $	← Add 0110 to the sum if the result is an <u>unused code</u>
Answer =	1 2	Read as one-two, not twelve

Value (decimal digit)	8421 code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
unused	1010
	1011
	1100
	1101
	1110
	1111

33

Gray Code

2-bit gray code	3-bit gray code	4-bit gray code	decimal
00	000	0000	0
01	001	0001	1
11	011	0011	2
10	010	0010	3
00	110	0110	4
	111	0111	5
	101	0101	6
	100	0100	7
		1100	8
		1101	9
		1111	10
		1110	11
		1010	12
		1011	13
		1001	14
		1000	15

Gray code (after [Frank Gray](#)) originally designed to prevent spurious output from mechanical switches. Today, Gray codes are widely used for [error correction](#) in digital communications such as digital TV and cable TV systems

One bit difference in the next or adjacent code word independent of the direction taken in the code

34

Binary code to Gray code conversion

- i. Place a leading 0 before the MSB.
- ii. XOR adjacent bits together starting from the left hand side of this number will result in gray code.

e.g. $82_{10} \rightarrow 1010010_2$

0 1 0 1 0 0 1 0

∨ ∨ ∨ ∨ ∨ ∨ ∨

1 1 1 1 0 1 1 ← Gray code

XOR

00 → 0

01 → 1

10 → 1

11 → 0

Gray code to Binary code conversion

- i. copy the first 1 starting from left hand side and continue with 1 until the next 1 appears in the gray code.
- ii. change to 0 and continue with 0 until the next 1 appears in the gray code.
- iii. change to 1 and follow steps (i) & (ii).

10001011 ← gray code

11110010 ← binary code

ASCII Code

American Standard Code for Information Interchange

ASCII Code	Value
000 0000	NULL
...	...
010 0000	Space
010 0001	! (exclamation mark)
010 0010	" (double quote)
...	...
011 0000	0
011 0001	1
...	...
011 1010	:(colon)
...	...
100 0001	A
...	...
101 1010	Z
...	...
110 0001	a
...	...
111 1010	z
...	...

control signals

symbols

numeric characters

symbols

capital letters

symbols

small letters

symbols

The word *Logic* would be coded as:

100 1100 110 1111
L o
110 0111 110 1001 110 0011
g i c

739 would be coded as:

011 0111 011 0011 011 1001
7 3 9

(Please refer to the complete ASCII table in your book)

37

Code Summary

Digit	'0'	'1'	'2'	'3'	...	'7'	'8'	'9'
8421 BCD	0000	0001	0010	0011	...	0111	1000	1001
XS3 Code	0011	0100	0101	0110	...	1010	1011	1100
Gray Code	0000	0001	0011	0010	...	0100	1100	1101
ASCII	0110000	0110001	0110010	0110011	...	0110111	0111000	0111001

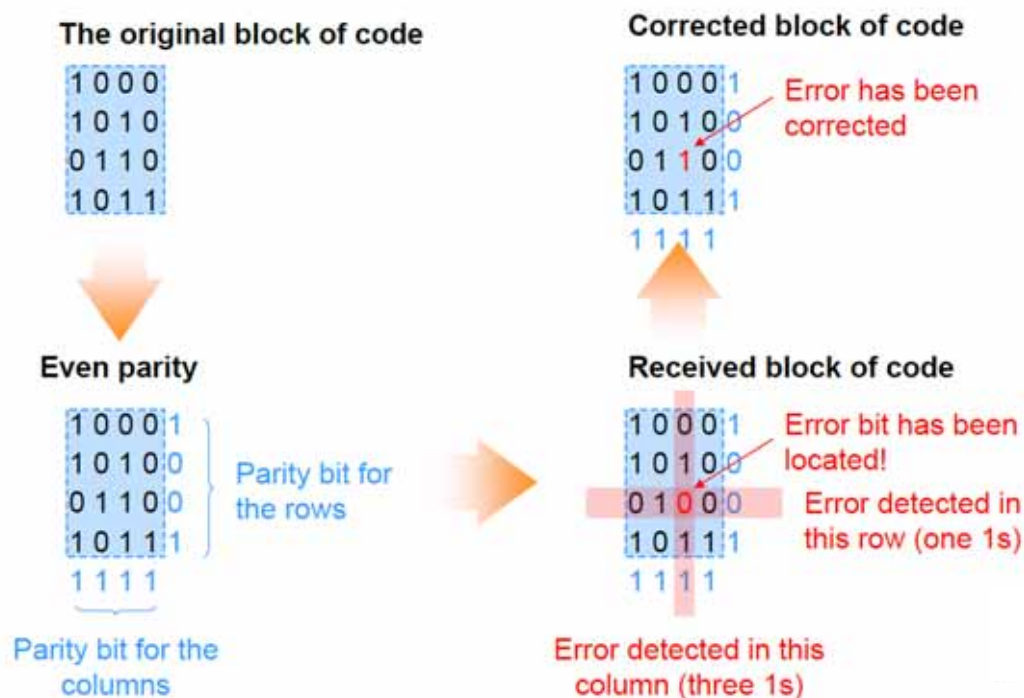
38

Parity and Error Correction

- The simplest method for **error detection** is using **parity bit**
 - An additional bit attaches to the original code
 - Two kinds of party bit (**even** or **odd** parities)
- The value of parity bit is defined by the total no. of “1” in the resulting codeword either even or odd

39

Example: Error Correction



40