ECE 223
Digital Circuits and Systems

Number System

M. Sachdev,
Dept. of Electrical & Computer Engineering
University of Waterloo

General Radix Representation

- **A decimal number such as 7392 represents**
  $7392 = 7x10^3 + 3x10^2 + 9x10^1 + 2x10^0$
- **It is practical to write only coefficients and deduce power of 10s from position**
  - In general, any radix (base) can be used
- Define coefficients a_i in radix r
  $0 <= a_i < r$
  $a_n r^n + a_{n-1} r^{n-1} \ldots a_0 r^0 + \ldots a_{-m} r^{-m}$
  Common radics – r = 2, 4, 8, 10, 16

General Radix Representation

r = 10 (Dec.)	r = 2 (Binary)	r = 8 (Octal)	r = 16 (Hex)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	L7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C

General Radix Representation

r = 10 (Dec.)	r = 2 (Binary)	r = 8 (Octal)	r = 16 (Hex)
13	1101	15	D
14	1110	16	E
15	1111	17	F

Usually, radix is shown as subscript

$(1234.4)_5 = 1x5^3 + 2x5^2 + 3x5^1\ 4x5^0 + 4x5^{-1} = (513.4)_{10}$

$(F75C.B)_{16} = 15x16^3 + 7x16^2 + 5x16^1 + 12x16^0 + 11x16^{-1} = (63{,}324.6875)_{10}$

Radix Conversion

- The integral part of a decimal number to radix r, repeatedly divide by r with reminders becoming a_i
  - Convert $(77)_{10}$ to binary

Integer	Remainder	Coefficient
77		
38	1	a0
19	0	a1
9	1	a2
4	1	a3
2	0	a4
1	0	a5
0	1	a6

- $(77)_{10} = (1001101)_2$

Examples

- Convert $(173)_{10}$ to r = 7

Integer	Remainder	Coefficient
173		
24	5	a0
3	3	a1
0	3	a2

$(173)_{10} = (335)_2$

- Converting from Binary to decimal

$(101101)_2 = 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 +0x2^1 +1x2^0 =$
$(45)_{10}$

Fraction Conversion

- To convert the fractional part of a number to radix r, repeatedly multiply by r; integral parts of products becoming a_i
- Convert $(0.7215)_{10}$ to binary

0.7215x2	= 1.443	$a_{-1} = 1$
0.443x2	= 0.866	$a_{-2} = 0$
0.866x2	= 1.772	$a_{-3} = 1$
0.772x2	= 1.544	$a_{-4} = 1$
0.544	= 1.088	$a_{-5} = 1$

- $(0.7215)_{10} = (0.10111..)_2$

7

Conversion between Binary/Octal/Hex

- Nice simple ways to convert between these three number systems, since all are a power of 2
- Binary to Octal simply requires grouping bits into groups of 3-bits and converting
- Binary to Hex simply requires grouping bits into groups of 4-bits and converting
- Going the other direction (Octal to Binary or Hex to Binary) should follow…
- Example $(100111100101)_2 = (4745)_8 = (9E5)_{16}$

8

Number Systems- Book Sections

- Number representations and conversions are covered in Chapter 1, Sections 1.1 through 1.4

9

Complements

- In computers, the representation and manipulation of –ve numbers is often performed using *complements*
- Complements for a radix come in two forms
  R's complement (Radix complement)
  (r-1)'s complement (Diminished radix complement)
- For binary system -- 2's complement, & 1's complement

10

r's Complement

- Given a +ve number *N* with *n* digits

  $N = (a_{n-1}a_{n-2} \ldots a_0)$
- r's complement is defined as

  $r^n - N$ for $N \neq 0$; zero otherwise
- Examples – 10's complement

  $(37218)_{10} = 10^5 - 37218 = 62782$

  $(0.12345)_{10} = 10^0 - 0.12345 = 0.87655$
- Examples – 2's complement

  $(101110)_2 = 2^6 - 101110 = 010010$

  $(0.0110)_2 = 2^0 - 0.0110 = 0.1010$

11

(r-1)'s Complement

- Given a +ve number *N* with *n* digit integer part & m digit fractional part

  $N = (a_{n-1}a_{n-2} \ldots a_0.a_{-1}a_{-2} \ldots a_{-m})$
- (r-1)'s complement is defined as

  $r^n - r^{-m} - N$
- Examples – 9's complement

  $(37218)_{10} = 10^5 - 1 - 37218 = 62781$

  $(0.12345)_{10} = 10^0 - 10^{-5} - 0.12345 = 0.87654$
- Examples – 1's complement

  $(101110)_2 = 2^6 - 2^0 - 101110 = 111111 - 101100 = 010011$

  $(0.0110)_2 = 2^0 - 2^{-4} - 0.0110 = 0.1111 - 0.0110 = 0.1001$

12

Interesting Facts

- The complement of a complement returns the original number
- Since we work with binary numbers a lot in digital systems, it is really worth nothing that:
  - The 1’s complement of a number is obtained by flipping bits
  - The 2’s complement of a number is obtained by flipping bits and adding 1

13

Arithmetic Operations

- Arithmetic operations - addition, subtraction, multiplication can be performed in any radix
- Example, r = 2

14

Addition (Unsigned Numbers)

- Binary addition of unsigned numbers is done just like in decimal. Add digits and generate carries

	1010101	85			1010101	85
+	0011001	+ 25		+	1011001	+ 89
0	1101110	110		1	0101110	174

- **Important:** We can get a non-zero ***carry out***; if we are limited to n-bits, this becomes an ***overflow*** condition

15

Subtraction (Unsigned Numbers)

- Binary subtraction is also similar to decimal subtraction. Use borrow when needed

	1010101	85			1010101	85
–	0011001	– 25		–	1011001	– 89
0	0111100	60			???????	– 4

- **Important:** Doesn't seems to work if second term is smaller!! A negative result is often considered an underflow

16

Subtraction (Unsigned Numbers) Using r's Complement

- Direct method (using borrows) is fine if done by hand, but a hassle in a digital system
  - Usage of complement makes subtraction easier to implement in hardware
- The subtraction of two numbers $(A-B)_r$ can be done as:
  - Add r's complement of B to A

  $Y = A + (r^n – B) = (A – B) + r^n$

17

Subtraction …

- If $A \geq B$ then carry out (r^n) occurs; ignore it, the rest is the subtracted result
- If $A < B$

  $Y = A + (r^n – B) = r^n - (A – B)$
- Note: (A – B) is a -ve number! Hence, take r's complement and treat it as –ve number

  $= r^n – Y = r^n – \{r^n - (A – B)\} = (A - B)$

18

Subtraction …Examples

- 10's complement subtraction

A = 72532, A' = 27468

B = 03250, B' = 96750

A – B = 72532		B – A = 03250	
+ 96750		+ 27468	
1 **69282**		30718	→ **- 69282**

19

Subtraction …Examples

- 2's complement subtraction

A = 1010100, A' = 0101100

B = 1000100, B' = 0111100

A – B =	1010100	B – A =	1000100
	+ 0111100	+	0101100
	1 **0010000**		1110000
		→	**-0010000**

20

Subtraction (Unsigned Numbers) Using (r-1)'s Complement

- Similar to r's complement subtraction
- If A> B

  $Y = A + \{(r^n – r^{-m}) – B\} = (A – B) + r^n – r^{-m}$

  If end carry (r^n) is discarded & least significant carry (r^{-m}) is added, result is the subtracted value
- If A = B

  $Y = A + \{(r^n – r^{-m}) – B\} = (A – B) + r^n – r^{-m} = 0 + r^n – r^{-m}$

21

Subtraction (Unsigned Numbers) Using (r-1)'s Complement

- If A < B

  $Y = A + \{(r^n – r^{-m}) – B\} = - (B – A) + r^n – r^{-m}$

  $= r^n – r^{-m} - (B – A)$
- We take (r-1)'s complement & treat it as a –ve number

  $r^n – r^{-m} –Y = r^n – r^{-m} – \{r^n – r^{-m} - (B – A)\} = (A - B)$

22

Subtraction …Examples

- 9's complement subtraction

A = 72532, A' = 27467

B = 03250, B' = 96749

A – B =	72532	B – A =	03250	
+	96749	+	27467	
1	69281		30717	→ **- 69282**
+	00001			
	69282			

23

Subtraction …Examples

- 1's complement subtraction

A = 1010100, A' = 0101011

B = 1000100, B' = 0111011

A – B =	1010100	B – A =	1000100	
+	0111011	+	0101011	
1	0010000		1101111	
	0000001		→	**- 0010000**
	0010000			

24

Signed Numbers

- Computers handle signed numbers as well
  - Numbers are represented in a fixed # of bits
- Often required to represent both +ve & -ve numbers in the same n-bit format
  - Left most bit (Most Significant Bit) represents the sign
    0 → +ve; 1 → -ve
- Three common representations
  - Sign & magnitude
  - Signed 1's complement
  - Signed 2's complement
- For +ve numbers, all three have same representation

Signed Numbers

- Sign & magnitude
  - MSB is the sign bit; rest is the magnitude
  - For 8 bit word

  + 9 → 0000 1001;   -9 → 1000 1001
- Signed 1's complement
  - MSB is the sign bit

    For +ve number → actual value

    For –ve number → 1's complement

  + 9 → 0000 1001;   -9 → 1111 0110
  - Has 2 representations for 0; not widely used

Signed Numbers

- Signed 2's complement
  - Common method of representing signed #s
  - Restrict the number range to (n-1) bits; Use 2's complement for –ve number representation
  - + 9 → 0000 1001; -9 → 1111 0111
- With 5 bit word size, in signed 2's complement we can represent number from -16 to +15

  $Y = (-a_{n-1})2^{n-1} + a_{n-2}2^{n-2} + \ldots\ldots a_0 2^0$
- Exercise – write all number from -16 to +15

27

Addition of Signed Binary Numbers

- 2's complement
  - Perform the addition, ignore the carry

	Binary	Decimal
	00000110	+ 6
+	00001101	+ 13
0	00010011	+ 19

	Binary	Decimal
	11111010	− 6
+	00001101	+ 13
1	00000111	+ 7

	Binary	Decimal
	00000110	+ 6
+	11110011	− 13
0	11111001	− 7

	Binary	Decimal
	11111010	− 6
+	11110011	− 13
1	11101101	− 19

28

Subtraction of Signed Binary Numbers

- 2's complement
  - Take 2's complement of the # to be subtracted

    (+/-A) – (+/-B) = (+/-A) + (-/+B)
  - A, B are in signed 2's complement
- Example, (-6) – (-13) = +7

(1111 1010) – (1111 0011)

= 1111 1010 + 0000 1101

= 1 0000 0111

- Ignore the carry out (9^{th} bit)

29

Codes

- Decimal numbers are coded using binary bit patterns

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

30

BCD Addition

- BCD addition can be carried out as follows:

4	0100	8	1000
+8	+1000	+9	+1001
12	1100	**17**	10001
	0110		0110
	1 0010		**1 0111**

31

Complements, Signed Arithmetic, Codes - Book Sections

- Complements and Signed Arithmetic are covered in Chapter 1, Sections 1.5 through 1.7

32