E&CE 223
Digital Circuits & Systems

Winter 2004

Lecture Transparencies
(Introduction)

M. Sachdev

Course Information: People

- **Instructor**
  - M. Sachdev, CEIT 4015, ext. 3370, msachdev@uwaterloo.ca
- **Lab Technologists**
  - Eric Praetzel, E2 - 2357, E.Praetzel@ece.uwaterloo.ca
- **Teaching Assistants**
  - Mohammed El Gebaly, mgebaly@vlsi.uwaterloo.ca
  - Shahab Ardalan ardalan@vlsi.uwaterloo.ca
  - Augstine Dominguez adominguez@engmail.uwaterloo.ca
  - Syed Iftekar Ali, sali@engmail.uwaterloo.ca

Course Information: Text

- **Text**
  - M. Marris Mano, Digital Design (3rd edition); Printice Hall
  - Lecture notes are at (http://ece.uwaterloo.ca/~msachdev)
- **Laboratory Manual**
  - download from http://ece.uwaterloo.ca:80/~ece223/
- **DC reserves**
  - M. Marris Mano, Digital Design (3rd edition); Printice Hall
- **Lectures**
  - **Lectures:** Mon, Wed, Fri 9.30am - 10.20 am, MC 1085
  - **Tutorials:** Tue @10.30 MC4041; Thurs@ 9.30 PHY235; @11.30 RCH 204

Course Information: Labs

- **Labs**
  - Three afternoons; 1.30 - 4.20; E2 2363
- **Labs**
  - **Lab 0 & 1:** individually; **Labs 2, 3 :** group of 2
  - For labs 2, 3; two 1.5 hr. slots will be available each lab day
  - You will be allowed to reserve four 1.5 hr. slots over each schedule lab period, with no more than 2 reservations in a single week. 4 slots include your DEMO period. Reservation is done electronically via Watstar
- **Marks**
  - **Final exam marks>= 50% :** Labs 30%, Midterm 20%, Final 50%
  - **Final exam marks < 50% :** Labs 0%, Midterm 0%, Final 100%

Coverage of Topics

- **Introduction [1]**

  This lecture

- **Number systems [2]**

  Radix, radix conversion, complements, subtraction, number representation, codes

- **Boolean algebra & logic circuits [8]**

  Boolean algebra, theorems, functions, minterms, maxterms, standard forms, Karnaugh maps, product of sum, sum of products, don't cares, prime implicants, multioutput circuits, Quine-McCluskey method

Coverage of Topics

- **Combinational logic design [8]**

  Design constraints, multi-level circuits, common term elimination, XOR circuits, adders & subtractors, iterative design, decoders/encoders, (de)multiplexers, programmed logic devices (ROM, PLA, FPGA)

- **Synchronous sequential circuits [8]**

  Sequential circuit types, flip-flops, triggering, analysis & design of clocked sequential circuits, transistion design, state reduction, design examples, registers, shift registers, special sequential circuits, timing considerations

- **Asynchronous sequential circuits [8]**

  fundamental mode systems with latches, design procedure, races, Moore & Mealy designs, hazards

Relationship to Future Courses

- **This course provides basis for higher order digital system courses**
  - E&CE 222 - Digital Computers
  - E&CE 324 - Microprocessor Systems & Interfacing
  - E&CE 427 - Digital Systems Engineering
  - E&CE 429 - Computer Structures

E&CE223 Assignments

- **Assignments 1**
  - Mano 1.3, 1.5, 1.6, 1.8, 1.15, 1.17, repeat 1.17 with 1's complements, 1.19, 1.23,
- **Assignment 2**
  - Mano 2.2, 2.3, 2.6, 2.7, 2.9, 2.11, 2.13, 2.14, 2.19
- **Assignment 3**
  - Mano 3.2, 3.3, 3.9, 3.10, 3.12, 3.23, 3.27
- **Assignment 4**
  - Mano 4.1, 4.7, 4.19, 4.28
- **Assignment 5**
  - 5.15, 5.16, 5.18, 5.24, 5.26, 5.27, 5.28, 5.32, 5.33, 5.34

E&CE223 Assignments

- **Assignment 6**
  - Mano 6.1, 6.2, 6.4, 6.9, 6.12, 6.21, 6.25
- **Assignment 7**
  - 7.4, 7.9, 7.17, 7.23, 7.27
- **Assignment 8**
  - 9.3, 9.5, 9.6, 9.12, 9.15, 9.18, 9.19
- ***Assignment problems and solutions are based on the 2nd edition of the book (see the course website for details)***

Lab and Tutorial Schedule (Tentative)

Week	Dates	Lab	Tutorial
1	Jan 5- 9	---	----
2	Jan 12-16	---	----
3	Jan 19 - 23	Lab0	Ass1
4	Jan 26- 30	Lab0	Ass2
5	Feb 2 - 6	-----	Lab1 Intro
6	Feb 9- 13	Lab1	Ass3
7	Feb 16 - 20	midterm	Ass4/Review
8	Feb 23- 27	Lab1	Ass5
9	Mar 1- Mar 5	Lab1	Lab2 Intro
10	Mar 8- 12	Lab2	Ass6
11	Mar 15 -19	Lab2	Lab3 Intro
12	Mar 22- 26	Lab3	Ass7
13	Mar 29 - 31	Lab3	Ass8

Section 1: Number Systems & Representations

- **Major topics**
  - Radix
  - r's complements
  - (r-1)'s complements
  - Subtraction using complements
  - Binary number representation
  - Codes

General Radix Representation

- **A decimal number such as 7392 actually represents**
  - $7392 = 7x10^3 + 3x10^2 +9x10^1 + 2x10^0$
  - $7.392 = 7x10^0 + 3x10^{-1} +9x10^{-2} + 2x10^{-3}$
- **It is practical to write only the coefficients and deduce necessary power of 10s from position**
  - In general, any radix (base) can be used
  - Define coefficients a_i in radix r

  $0 <= a_i < r$

  $a_n r^n + a_{n-1} r^{n-1} + ... + a_0 r^0 + a_{-1} r^{-1} + ... + a_{-m} r^{-m}$
- **Common radics include r = 2, 4, 8, 10, 16**

General Radix Representation

r =10 (dec.)	r=2(bin.)	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	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D

14	1110	16	E
15	1111	17	F

- **Notation**
  - Usually shows radix as a subscript

$(1234.4)_5 = 1x5^3 + 2x5^2 + 3x5^1 + 4 + 4x5^{-1} = (194.8)_{10}$

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

$= (63,324.6875)_{10}$

Radix Conversion

- **To convert the integral part of a number to radix r, repeatedly divide by r with the reminders becoming the a_i**
  - Convert $(77)_{10}$ to binary (r=2)

Integer	Remainder	Coefficient
77		
38	1	a_0
19	0	a_1
9	1	a_2
4	1	a_3
2	0	a_4
1	0	a_5
0	1	a_6

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

Radix Conversion

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

Integer	Remainder	Coefficient
173		
24	5	a_0
3	3	a_1
0	3	a_2

- $(173)_{10} = (335)_7$

Fraction Conversion

- **To convert the fractional part of a number to radix r, repeatedly multiply by r with the integral parts of the products becoming a_i**

- **Convert $(0.7215)_{10}$ to binary**

0.1715x2	= 1.443	a-1 = 1
0.443 x2	= 0.886	a-2 = 0
0.886 x2	= 1.772	a-3 = 1
0.772 x2	= 1.544	a-4 = 1
0.544 x2	= 1.088	a-5 = 1
0.088 x2	= 0.176	a-6 = 0
0.176 x2	= 0.352	a-7 = 0

- **$(0.7215)_{10} = (0.1011100\ ...)_2$**

Fraction Conversion

- Convert $(0.312)_{10}$ to r = 7

0.312 x7	= 2.184	a-1 = 2
0.184 x7	= 1.288	a-2 = 1
0.288 x7	= 2.016	a-3 = 2
0.016 x7	= 0.112	a-4 = 0

- $(0.312)_{10} = (0.2120....)_7$

Arithmetic Operations

- **The arithmetic operations of addition, subtraction, multiplication and division can be performed in any radix by using appropriate addition and multiplication tables**
  - Example, r =2

Complements

- **In a computer the representation and manipulation of negative numbers is usually performed using *complements***
- **Complements for a radix come in two forms**
  - r's complement
  - (r-1)'s complement

+	0	1
0	0	1
1	1	10

X	0	1
0	0	0
1	0	1

- **r's complement**
  - Given a positive number *N* which has n digit inte-

```
   1011
 +  101
 ------
  10000
```

```
    1011
 X   101
 -------
    1011
   000
  1011
 -------
  110111
```

ger part, the r's complement of N is defined as

$r^n - N$ for $N \neq 0$, zero otherwise

Complements

- **10's Complement**
  - 10's complement of $(37218)_{10} = 10^5 - 37218 = 62782$
  - 10's complement of $(0.12345)_{10} = 10^0 - 0.12345 = 0.87655$
- **2's Complement**
  - 2's complement of $(101110)_2 = 2^6 - (101110) = 010010$
  - 2's complement of $(0.0110)_2 = 2^0 - (0.0110) = 0.1010$

(r-1)'s Complement

- **Given a positive number N with integer part n bits, fractional part m bits, the (r-1)'s complement is defined as**

  $(r^n - n^{-m} - N)$ = rr......rr.rr.......rr - N

- **9's complement (r = 10)**

  9's complement of $(37218)_{10} = 10^5 - 1 - 37218 = 62781$

  9's complement of $(0.12345)_{10} = (10^0 - 10^{-5} - 0.12345)$

  $= 0.99999 - 0.12345 = 0.87654$

  9's complement 0f (25.639)10 = $(10^2 - 10^{-3} - 25.639) =$

  $= 99.999 - 25.639 = 74.360$

(r-1)'s Complement

- **1's complement (r = 2)**
  - 1's complement of $(101100)_2 = 2^6 - 1 - (101100)$
    $= 111111 - 101100 = 010011$
  - 1's complement of $(0.0110)_2 = 2^0 - 2^{-4} - (0.0110)$
    $= 0.1111 - 0.0110 = 0.1001$

r's Complement Subtraction

- **For positive numbers, A and B the r's complement subtraction (A-B) is performed as follows:**
  - Add the r's complement of B to A
  - If and end carry occurs, ignore it; else take the r's complement of the result and treat it as a negative number
  - Proof

  Let X = A - B, $A \geq 0, B \geq 0$

  Let Y = A + (r^n - B)

- **(1) If** $\boldsymbol{A \geq B}$

  Y = r^n + (A - B) = r^n + X

  The r^n is the carry out from (n-1)th (most significant digit). If this carry out is (end carry) is discarded, the value of Y is the true positive result X

- **If** $A < B$

  Y = r^n - (B - A) = r^n - (-X)

  Note that -X has a positive value. Hence Y is the r's complement of the negative of the true value.

  QED

- **10's complement subtraction**

  A = 72532 A' = 27468
  B = 03250 B' = 96750

```
A - B          72532
             + 96750
             ------------
             1  69282
B - A          03250
             + 27468
             ------------
               30718 --> - 69282
```

■ 2's complement subtraction

A = 1010100 A' = 0101100

B = 1000100 B' = 0111100

A - B

```
    1010100
+   0111100
---------------
1   0010000
```

B - A

```
     1000100
+    0101100
----------------
     1110000 --> - (0010000)2
```

$1110000 \rightarrow -(0010000)_2$

(r-1)'s Complement Subtraction

- **For positive numbers A and B the (r -1)'s complement subtraction (A-B) is performed as follows:**
  - Add the (r-1)'s complement of B to A
  - If an end carry occurs, add 1 to the least significant digit of the result (end-round-carry)
    else, take (r-1)'s complement of result and treat it as a negative number
  - Proof

  Let X = A - B, $A \geq 0, B \geq 0$

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

- **(1) If** $\boldsymbol{A > B}$

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

The r^n is the carry out from (n-1)th (most significant digit). If this carry out is (end carry) is discarded and r^{-m} (one in the least significant digit) is added to the result , the value of Y is the true positive result X

- **(2) A = B**

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

  The result is the (r-1)'s complement of the true result zero

- **(2)** $A < B$

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

  Note that (-X) is a positive value. Hence, Y is the (r-1)'s complement of the nagative of the true value

  QED

- **9's complement subtraction**

  A = 72532   A' = 27467
  B = 03250   B' = 96749

A - B

```
     72532
  +  96749
  ------------
  1  69281
     00001
  -----------
     69282
```

B - A

```
     03250
  +  27467
  ------------
     30717 --> - 69282
```

- **1's complement subtraction**

A = 1010100 A' = 0101011

B = 1000100 B' = 0111011

A - B

```
     1010100
+    0111011
---------------
1    0001111
     0000001
---------------
     0010000
```

B - A

```
     1000100
+    0101011
----------------
     1101111 --> - (0010000)2
```

Signed Binary Number Representation

- **In most computer applications, integers are represented in a fixed number of bits (fixed format)**
  - With n bits, positive numbers from 0 to 2^n -1 can be represented

00000	0	01011	11	10110	22
00001	1	01100	12	10111	23
00010	2	01101	13	11000	24
00011	3	01110	14	11001	25
00100	4	01111	15	11010	26
00101	5	10000	16	11011	27
00110	6	10001	17	11100	28
00111	7	10010	18	11101	29
01000	8	10011	19	11110	30
01001	9	10100	20	11111	31
01010	10	10101	21		

Signed Number Representation

- **It is often required to represent both positive & negative numbers in the same n-bit format; three common methods**
  - Sign & magnitude
  - Signed 1's complement
  - Signed 2's complement
- **In all cases, the leftmost bit is the sign: 0 --> positive 1 --> negative**
  - All three forms have the same representation for positive numbers
- **Sign & magnitude**
  - Most significant bit (MSB) is the sign and the rest is the magnitude

- Commonly used for fractional numbers (real, floating point) in computers

Signed 1's complement

- The MSB is the sign bit, the rest is
  -- the actual value for the positive numbers
  -- the 1's complement form for negative numbers
- Has two representations for zero!!
- No longer widely used

Signed 2's complement

- The common method of representing signed integers on computers
- Restrict the number range to (n-1) bits. Use 2's complement for negative number representation

  +M --> M (most significant bit is 0)

  -M --> 2^n - M (n-bit 2's complement)
  = 2^{n-1} + [2^{n-1} - M] = 2^{n-1} + [n-1 bit 2's complement]

Department of Electrical & Computer Engineering, University of Waterloo

- MSB indicates the sign of the number (sign bit)
- Numerical value of a number is given by

  $(-a_{n-1})2^{n-1} + a_{n-2}x2^{n-2} + \dots\dots\dots + a_1x2^1 + a_0xa$
- With a 5-bit 2's complement number, we can represent the range from -16 to +15

  $00110 \dashrightarrow (-0)2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 0xa^0 = 6$

  $10110 \dashrightarrow (-1)2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 0xa^0 = -10$
- 5-bit number in 2's complement

10000	-16	11011	-5	00110	6
10001	-15	11100	-4	00111	7
10010	-14	11101	-3	01000	8
10011	-13	11110	-2	01001	9
10100	-12	11111	-1	01010	10
10101	-11	00000	0	01011	11
10110	-10	00001	1	01100	12
10111	-9	00010	2	01101	13
11000	-8	00011	3	01110	14
11001	-7	00100	4	01111	15
11010	-6	00101	5		

Department of Electrical & Computer Engineering, University of Waterloo

Codes

- **Decimal numbers are coded using binary bit patterns**
  - Binary coded decimal (BCD): each decimal digit is represented by a 4-digit binary number; e.g. $(7\ 4\ 3)_{10} = (0111\ 0100\ 0011)_{BCD}$
  - Table of various binary codes for decimal digits

decimal	8421 (BCD)	84-2-1	excess-3	2 out of 5	gray
0	0000	0000	0011	00011	0000
1	0001	0111	0100	00101	0001
2	0010	0110	0101	00110	0011
3	0011	0101	0110	01001	0010
4	0100	0100	0111	01010	0110
5	0101	1011	1000	01100	1110
6	0110	1010	1001	10001	1010
7	0111	1001	1010	10010	1011
8	1000	1000	1011	10100	1001
9	1001	1111	1100	11000	1000

- **8421 and 84-2-1 are weighted codes. The latter is convenient for 9's complement operations**
- **Excess-3 is BCD plus 3. It is convenient 9's complement operations**
- **2 out of 5 is useful for error checking (exactly 2 bits are ones)**
- **In Gray code consecutive digits differ only by one bit. Useful for mechanical position decoding.**

ASCII Codes

- **There are several codes for representing textual information in computers. The most common is ASCII (American Standard Code for Information Interchange)**
  - 7 bit code
  - 95 printing characters, 33 control characters
  - Originally designed for punched paper tape and teletypes