سیستم‌های دیجیتال ۱

دکتر شهرزاد عجبی

۳ واحد نظری همنیاز : اصول الکترونیک

معرفی درس

1. سیستم اعداد ۱ نمره
2. دروازه ها و ساختارهای مدارهای منطقی ، جبر بول ، جدول کارنو، تحقق توابع دو سطحی ۲ نمره
3. مدارهای ترکیبی (طراحی و تحلیل) : مبدل کد، جمع کننده، ضرب کننده، کد بردار؛ کد گذار ، مالی پلکسر ۶ نمره
4. مدارهای ترتیبی(طراحی و تحلیل) : معادلات حالت و تحلیل مدارهای ترتیبی، فلیپ فلاپها، شیفت رجیسترها، ثباتها، شمارنده ها ۶ نمره
5. آرایه های برنامه پذیر PLD (PAL,PLA), CPLD,FPGA ۱ نمره
6. حافظه ها(ROM, RAM) ۱ نمره

- زبان توصیف سخت افزاری

سرفصل (آموزش عالی)

- هدف: آشنایی با روشهای تحلیل و طراحی مدارهای منطقی و سیستم‌های دیجیتال
- شرح درس:
  - سیستم اعداد، جبر بول ، جدول کارنو
  - دروازه های ، ساختارهای مدارهای منطقی
  - آرایه های برنامه پذیر **PLD (PAL,PLA), CPLD,FPGA**
  - زبان توصیف سخت افزاری
  - مدارهای ترکیبی: مبدل کد، جمع کننده، ضرب کننده، کد بردار؛ کد گذار ، مالی پلکسر
  - مدارهای ترتیبی: معادلات حالت و تحلیل مدارهای ترتیبی، فلیپ فلاپها، شیفت رجیسترها، ثباتها، شمارنده ها
  - حافظه ها(**ROM, RAM**)
  - **ALU**، معماری کامپیوتر

مراجع :

۱. م. تابنده و س. م. مگی، مدارهای منطقی و سخت‌افزارهای کامپیوتر، مؤسسه انتشارات علمی دانشگاه صنعتی شریف، ۱۳۹۰.

2. M.M. Mano, and M. D. Cilleti, Digital Design, 4th ed., Prentice – Hall, 2006.

3. V.P. Nelson,et al., Digital logic Circuit Analysis and Design, Prentice – Hall , 1995.

4. M.M. Mano, Computer System Architecture, 2nd Custom ed., Pearson Custom Publishing, 2005.

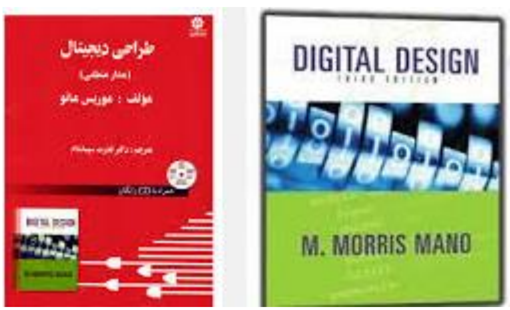
5. C. H. Roth, Fundamentals of Logic Design, 6th ed., Cengage Learning Press, 2010.

6. John F. Wakerly, Digital Design: Principles and Practices, 4th ed., 2005.

مراجع

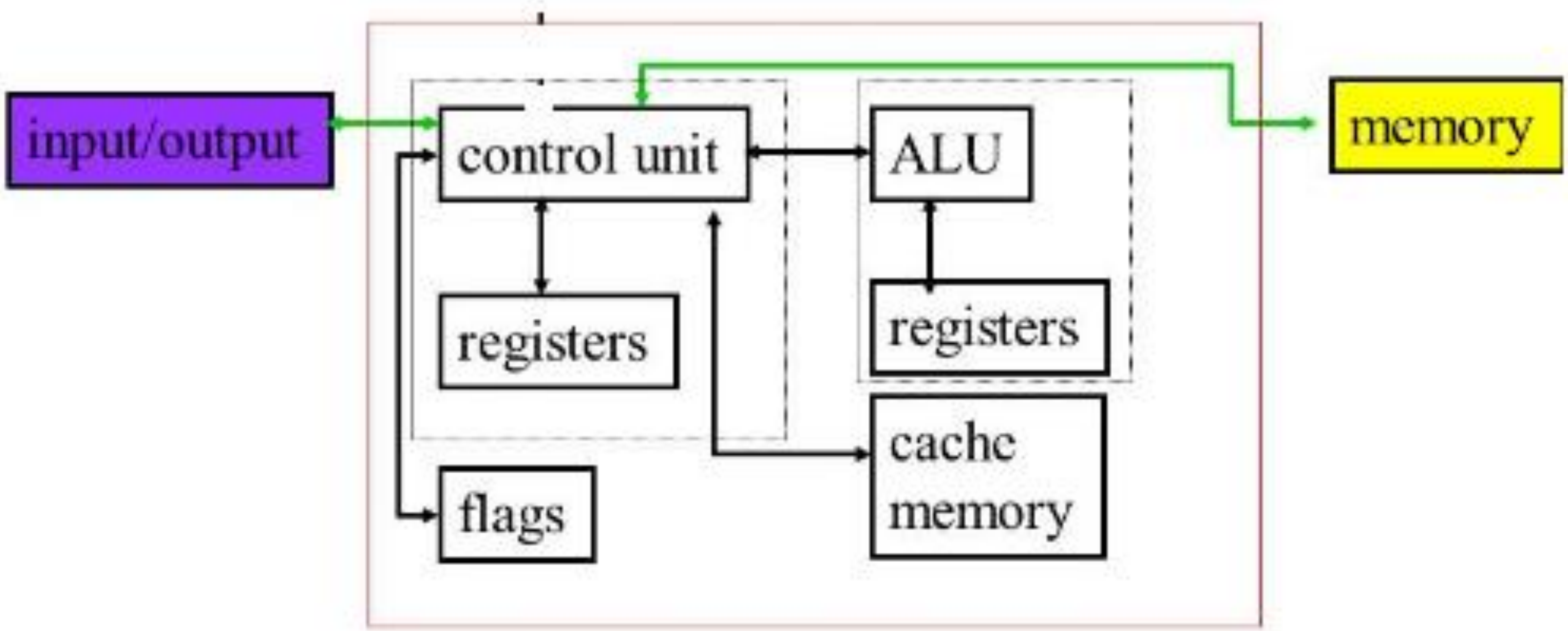
م تابنده، س. م. مکی؛ مدارهای منطقی و سخت افزارهای کامپیوتر، موسسه انتشارات علمی دانشگاه صنعتی شریف، ۱۳۹۰

موریس مانو؛ مدارهای منطقی



- جزوه دستنویس خودتان کافی است!

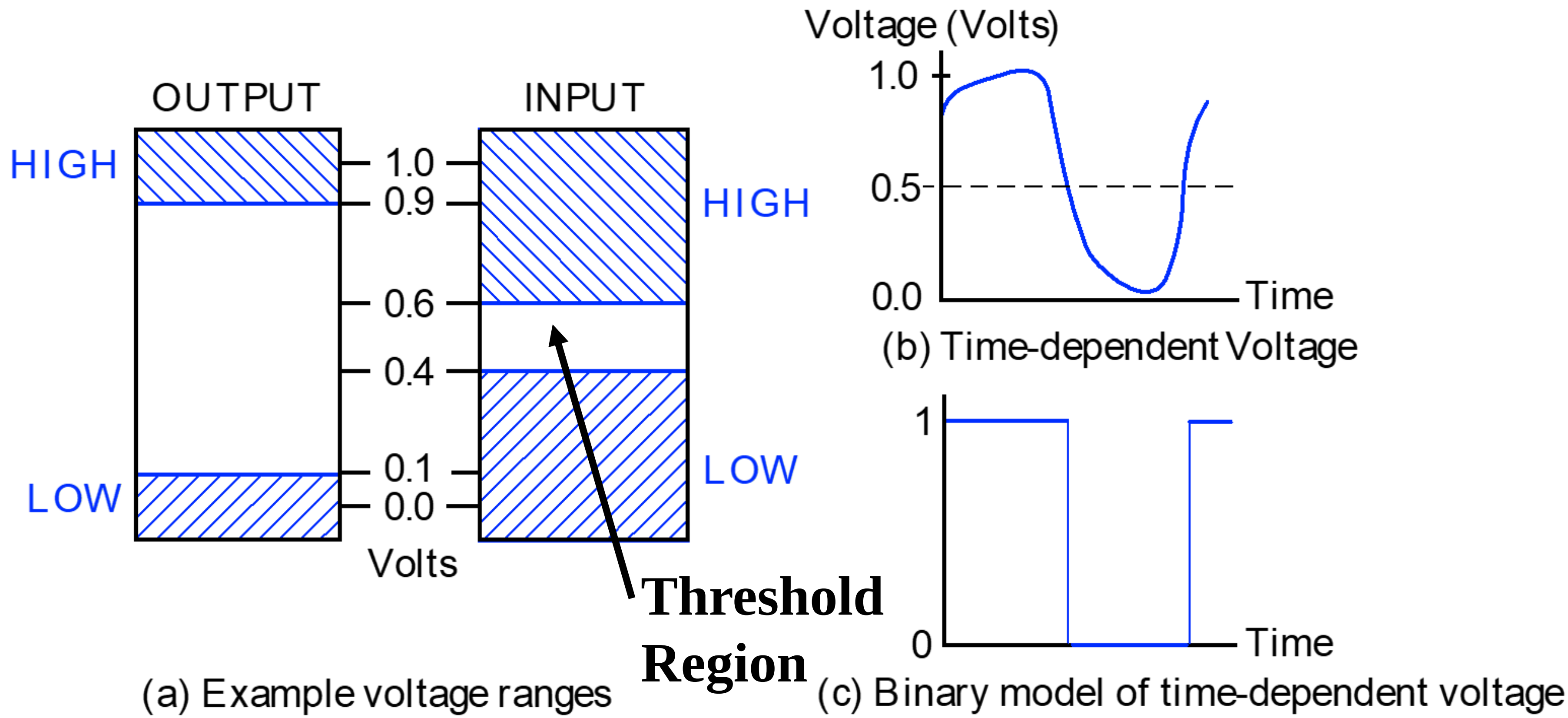
▪Why Study Digital Design?



سیستم‌های کامپیوتر دیجیتال

- سیستم‌های دیجیتال مقادیر گسسته داده را در نظر می‌گیرند.
- مثل:
  - ۲۶ حرف الفبای انگلیسی
  - ۱۰ رقم دسیمال
- کامپیوتر با صفر و یک کار می‌کند.
- دلیل: مقادیر باینری به راحتی با جریان و ولتاژ قابل نمایش هستند.
- با مدار قابل تحقق هستند.

Signal Example – Physical Quantity: Voltage



سیستم اعداد

- مبناهای { دو، هشت، ۱۶ }
  [binary, octal and hexadecimal]
- عملیات ریاضی
- تبدیل مبنا
- کدهای دهدهی [Binary Coded Decimal (BCD)]
- کدهای الفبایی
- کد گری
- کد پریتی

سیستم اعداد: یادآوری فهم دسیمال

- اعداد دسیمال با ارقام ساخته می‌شوند. (0,1,2,3,4,5,6,7,8,9)

- $8653 = 8*10^3 + 6*10^2 + 5*10^1 + 3*10^0$
- $97654.35 = 9*10^4 + 7*10^3 + 6*10^2 + 5*10^1 + 4*10^0 + 3*10^{-1} + 5*10^{-2}$
- نحوه نمایش پایه عدد $(97654.35)_{10}$

سیستم‌های عددی – نمایش

- عدد در مبنای r
- یک عدد در مبنای r به صورت یه رشته از ارقام (کاراکتر) نمایش داده می شود:

$$\boldsymbol{a}_{n-1}\boldsymbol{a}_{n-2}\cdots\ \boldsymbol{a}_1\boldsymbol{a}_0\ .\ \boldsymbol{a}_{-1}\ \boldsymbol{a}_{-2}\cdots\ \boldsymbol{a}_{-m+1}\ \boldsymbol{a}_{-m}$$

که در آن $0 \le a_i < r$ و . ممیز است.

- هر موقعیت در رشته دارای وزنی است که توانی از پایه است:

$$(\text{Number})_r = \left(\sum_{i=0}^{i=n-1} a_i \cdot r^i\right) + \left(\sum_{j=-m}^{j=-1} a_j \cdot r^j\right)$$

(Integer Portion) + (Fraction Portion)

مبنای هشت

- نمایش عدد در مبنای هشت یا Octal با ارقام :
(0,1,2,3,4,5,6,7)
- مثال

• $(4536)_8 = 4*8^3 + 5*8^{2\,+}\,3*8^{1\,+}\,6*8^0 = (1362)_{10}$

- مثال

• $(465.27)_8 = 4*8^{2\,+}\,6*8^{1\,+}\,5*8^0 + 2*8^{-1\,+}\,7*8^{-2}$

- چرا مبنای هشت را بیاموزیم؟

مبنا های متداول

Name	Radix	Digits
Binary	2	0,1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

- **The six letters (in addition to the 10 integers) in hexadecimal represent:**

شمارش

Decimal (Base 10)	Binary (Base 2)	Octal (Base 8)	Hexadecimal (Base 16)
00	00000	00	00
01	00001	01	01
02	00010	02	02
03	00011	03	03
04	00100	04	04
05	00101	05	05
06	00110	06	06
07	00111	07	07
08	01000	10	08
09	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

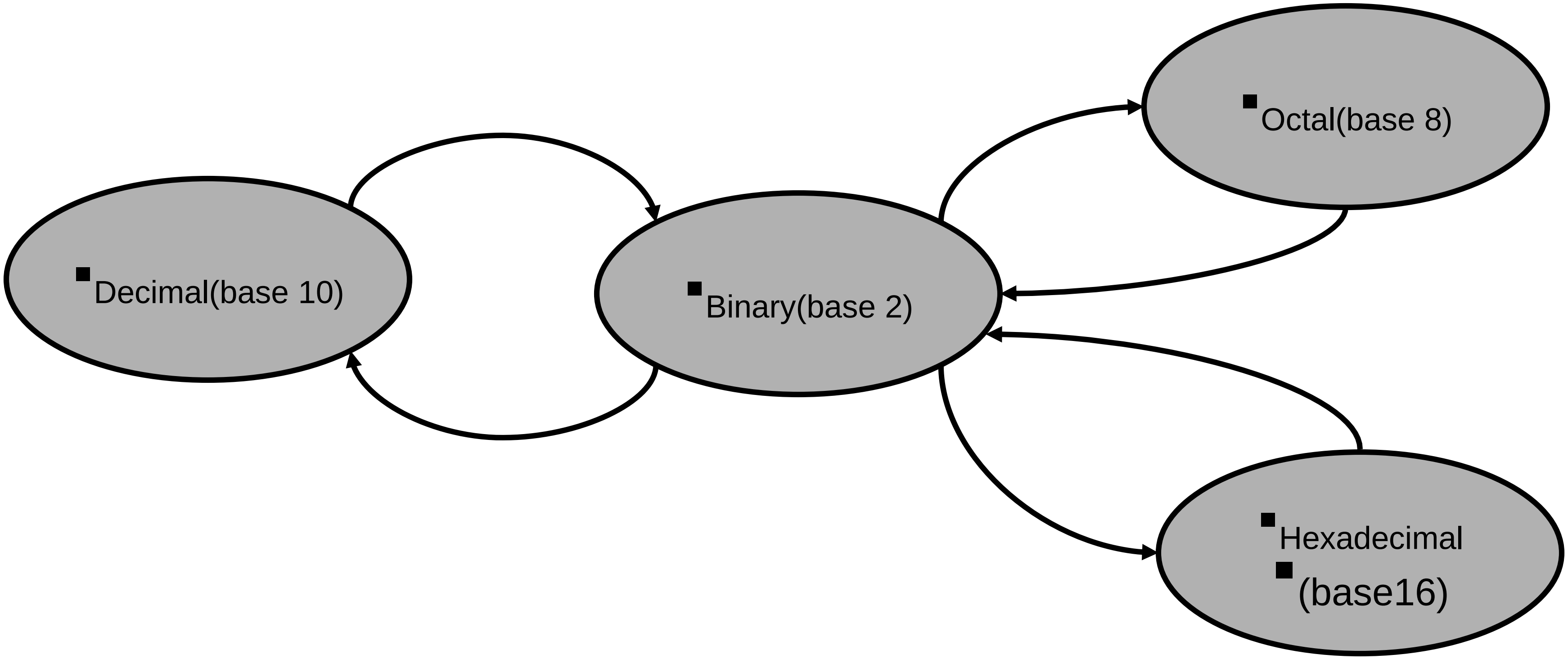
سیستم های عددی

	General	Decimal	Binary
Radix (Base)	r	10	2
Digits	0 => r - 1	0 => 9	0 => 1
0	r^0	1	1
1	r^1	10	2
2	r^2	100	4
3	r^3	1000	8
Powers of Radix 4	r^4	10,000	16
5	r^5	100,000	32
-1	r^{-1}	0.1	0.5
-2	r^{-2}	0.01	0.25
-3	r^{-3}	0.001	0.125
-4	r^{-4}	0.0001	0.0625
-5	r^{-5}	0.00001	0.03125

اعداد باینری یا مبنای دو

- Binary numbers are made of <u>b</u>inary dig<u>it</u>s (bits):
  - 0 and 1
  - مثال
  - $(1011)_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = (11)_{10}$
- مثال اعشاری
  - $(110.10)_2 = 1x2^2 + 1x2^1 + 0x2^0 + 1x2^{-1} + 0x2^{-2}$
- معرفی بایت و نیبل
- یک عدد باینری شامل ۸ بیت *byte*
  - $(11001001)_2$
- یک عدد باینری ۴ بیتی: *nibble*
  - $(1101)_2$

تبدیل از مبنا های مختلف



خلاصه تبدیل مبنا

		To			
From		Dec	Bin	Octal	Hex
	Dec	-	Repeated / or *	تقسیم و یا ضرب متوالی	تقسیم و یا ضرب متوالی
	Bin	استفاده از وزن بیت ها	-	Convert every 3 digits	Convert every 4 digits
	Octal	Add digit*weight	Split every digit to 3	-	Thru Bin
	Hex	Add digit*weights	Split every digit to 4	Thru Bin	-

مثال تبدیل عدد صحیح

- For each digit position:

1. **Divide decimal number by the base (e.g. 2)**
2. **The *remainder* is the lowest-order digit**
3. **Repeat first two steps until no *divisor* remains.**

- Example for $(13)_{10}$:

مثال تبدیل عدد صحیح

- For each digit position:

1. **Divide decimal number by the base (e.g. 2)**
2. **The *remainder* is the lowest-order digit**
3. **Repeat first two steps until no *divisor* remains.**

- Example for (13)**10**:

Integer Quotient		Remainder	Coefficient
13/2 = 6	+	½	a0 = 1
6/2 = 3	+	0	a1 = 0
3/2 = 1	+	½	a2 = 1
1/2 = 0	+	½	a3 = 1

- Answer (13)**10 = (**a3 a2 a1 a0)2 = (1101)2

مثال تبدیل بخش اعشاری

- For each digit position:

1. **Multiply decimal number by the base (e.g. 2)**
2. **The *integer* is the highest-order digit**
3. **Repeat first two steps until fraction becomes zero.**

- Example for $(0.625)_{10}$:

مثال تبدیل بخش اعشاری

- For each digit position:

1. **Multiply decimal number by the base (e.g. 2)**
2. **The *integer* is the highest-order digit**
3. **Repeat first two steps until fraction becomes zero.**

- Example for $(0.625)_{10}$:

0.625 * 2 =	1	+	0.25	a-1 = 1
0.250 * 2 =	0	+	0.50	a-2 = 0
0.500 * 2 =	1	+	0	a-3 = 1

- Answer $(0.625)_{10} = (0.a_{-1}\, a_{-2}\, a_{-3})_2 = (0.101)_2$

توان های مفید از ۲ برای تبدیل مبنای ۲

- Useful for Base Conversion

Exponent	Value
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

Exponent	Value
11	2,048
12	4,096
13	8,192
14	16,384
15	32,768
16	65,536
17	131,072
18	262,144
19	524,288
20	1,048,576
21	2,097,152

توانهای خاص ۲

- 2^{10} (1024) is Kilo, denoted "K"
- 2^{20} (1,048,576) is Mega, denoted "M"
- 2^{30} (1,073, 741,824) is Giga, denoted "G"
- 2^{40} (1,099,511,627,776) is Tera, denoted “T"

مثال: اکتال

- For each digit position:

1. **Divide decimal number by the base (8)**
2. **The *remainder* is the lowest-order digit**
3. **Repeat first two steps until no *divisor* remains.**

- Example for $(175)_{\mathbf{10}}$:

مثال: اكتال

- For each digit position:

1. **Divide decimal number by the base (8)**
2. **The *remainder* is the lowest-order digit**
3. **Repeat first two steps until no *divisor* remains.**

- Example for (175)**10**:

$175/8 =$	21	$+$	$7/8$	$a_0 = 7$
$21/8 =$	2	$+$	$5/8$	$a_1 = 5$
$2/8 =$	0	$+$	$2/8$	$a_2 = 2$

- Answer $(175)_{\mathbf{10}}$ **= (**$a2\ a1\ a0)_2 = (257)_8$

مثال اكتال

- For each digit position:

1. **Multiply decimal number by the base (e.g. 8)**
2. **The *integer* is the highest-order digit**
3. **Repeat first two steps until fraction becomes zero.**

- Example for (0.3125)**10**:

	Integer		Fraction	Coefficient
0.3125 x 8 =	2	+	5	a-1 = 2
0.5000 x 8 =	4	+	0	a-2 = 4

- Answer (0.3125)**10** = (0.24)8

مثال تبدیل مبنا

مثال تبدیل مبنا

Binary Arithmetic عملیات ریاضی-

- Single Bit Addition with Carry
- Multiple Bit Addition
- Single Bit Subtraction with Borrow
- Multiple Bit Subtraction
- Multiplication
- BCD Addition

جمع تک بیتی همراه با رقم نقلی

Given two binary digits (X,Y), a carry in (Z) we get the following sum (S) and carry (C):

Carry in (Z) of 0:

Z	0	0	0	0
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 0	0 1	0 1	1 0

Carry in (Z) of 1:

Z	1	1	1	1
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 1	1 0	1 0	1 1

Multiple Bit Binary Addition

- مثال

نقلی	<u>0</u>	<u>0</u>
مضاف الیه	01100	10110
مضاف	<u>+ 10001</u>	<u>+ 10111</u>
مجموع		

- نقلی ورودی پیش فرض صفر است.

تفریق باینری تک بیتی همراه با قرضی

- **Given two binary digits (X,Y), a borrow in (Z) we get the following difference (S) and borrow (B):**
- **Borrow in (Z) of 0:**

Z	**0**	**0**	**0**	**0**
X	**0**	**0**	**1**	**1**
- Y	**-0**	**-1**	**-0**	**-1**
BS	**0 0**	**1 1**	**0 1**	**0 0**

- **Borrow in (Z) of 1:**

Z	**1**	**1**	**1**	**1**
X	**0**	**0**	**1**	**1**
- Y	**-0**	**-1**	**-0**	**-1**
BS	**11**	**1 0**	**0 0**	**1 1**

تفریق باینری چند بیتی

- دو مثال تفریق چند بیتی

Borrows	0	0
Minuend	10110	10110
Subtrahend	- 10010	- 10011
Difference		

- فرض : مثال های برای پاسخ مثبت طرح شده اند.

ضرب باینری

```
          1  0  1  1  1
X            1  0  1  0
-----------------------
          0  0  0  0  0
       1  0  1  1  1
    0  0  0  0  0
 1  0  1  1  1
-----------------------
 1  1  1  0  0  1  1  0
```

نحوه نمایش اعداد علامت دار

- روی کاغذ از علامت + و – برای نمایش علامت استفاده می شود.
- برای کامپیوتر نمایش علامت باید با بیت ۰ یا ۱ باشد.
- سه نوع نمایش برای اعداد منفی وجود دارد:
- **signed magnitude, 1's complement, 2's complement**
- نکته: مستقل از روش نمایش علامت ، بیت سمت چپ، بی علامت نامیده می شود. برای اعداد منفی بیت علامت ۱ و برای اعدادمثبت این بیت صفر است.

Consider ***signed magnitude***:

00001100b = 12d

Sign bit　　Magnitude

10001100b = -12d

Sign bit　　Magnitude

One’s Complement

- **The one’s complement of a binary number involves inverting all bits.**
  - **1’s comp of 00110011 is 11001100**
  - **1’s comp of 10101010 is 01010101**
- برای عددی با **n** رقم صحیح و **m**رقم اعشار مکمل یک آن با رابطه زیر بدست می‌آید.
- **1’s complement is $(2^{n}-2^{-m}) - N$.**
- برای پیدا کردن منفی یا قرینه یک عدد از آن مکمل یک می گیریم.

00001100b = 12d

Sign bit Magnitude

11110011b = -12d

Sign bit Magnitude

Two's Complement

- The two's complement of a binary number involves inverting all bits and adding 1.
  - 2's comp of 00110011 is 11001101
  - 2's comp of 10101010 is 01010110
- For an n bit number N the 2's complement is
- $2^n - N$.
- To find negative of 2's complement number take the 2's complement.

00001100b = 12d

Sign bit Magnitude

11110100b = -12d

Sign bit Magnitude

مکمل دو یک عدد بدون کمک فرمول

- Algorithm 1 – Simply complement each bit and then add 1 to the result.
  - Finding the 2's complement of $(01100101)_2$ and of its 2's complement…

N = 01100101   [N] = 10011011

```
N   = 01100101        [N] =   10011011
        10011010                 01100100
   +           1      +            1
   --------------      --------------
        10011011                 01100101
```

- Algorithm 2 – Starting with the least significant bit, copy all of the bits up to and including the first 1 bit and then complementing the remaining bits.
  - N = 0 1 1 0 0 1 0 1

    [N] = 1 0 0 1 1 0 1 1

نکته:

ماشین با تعداد محدود بیت ظرفیت اعداد محدودی را دارد

- Machines that use 2’s complement arithmetic can represent integers in the range

  $$-2^{n-1} <= N <= 2^{n-1}-1$$

  where n is the number of bits available for representing N. Note that $2^{n-1}-1 = (011..11)_2$ and $-2^{n-1} = (100..00)_2$

- For 2’s complement more negative numbers than positive.
- For 1’s complement two representations for zero.

مثال

1’s Complement جمع در سیستم اعداد علامت دار

- Using 1’s complement numbers, adding numbers is easy.
- For example, suppose we wish to add $+(1100)_2$ and $+(0001)_2$.
- Let’s compute $(12)_{10} + (1)_{10}$.
  - $(12)_{10} = +(1100)_2 = 01100_2$ in 1’s comp.
  - $(1)_{10} = +(0001)_2 = 00001_2$ in 1’s comp.

- Step 1: Add binary numbers
- Step 2: Add carry to low-order bit

```
Add            0 1 1 0 0
            +  0 0 0 0 1
            ---------------
             0 0 1 1 0 1
Add carry    └──────────→ 0
            ---------------
Final Result   0 1 1 0 1
```

1’s Complement تفریق در سیستم اعداد علامت دار

- Using 1’s complement numbers, subtracting numbers is also easy.
- For example, suppose we wish to subtract $+(0001)_2$ from $+(1100)_2$.
- Let’s compute $(12)_{10} - (1)_{10}$.
  - $(12)_{10} = +(1100)_2 = 01100_2$ in 1’s comp.
  - $(-1)_{10} = -(0001)_2 = 11110_2$ in 1’s comp.

Step 1: Take 1's complement of 2nd operand

Step 2: Add binary numbers

Step 3: Add carry to low order bit

```
              0 1 1 0 0
-             0 0 0 0 1
---------------
1's comp
              0 1 1 0 0
Add    +      1 1 1 1 0
---------------
          1   0 1 0 1 0
Add carry             1
---------------
              0 1 0 1 1   Final Result
```

2's Complement Addition

- Using 2's complement numbers, adding numbers is easy.
- For example, suppose we wish to add $+(1100)_2$ and $+(0001)_2$.
- Let's compute $(12)_{10} + (1)_{10}$.
  - $(12)_{10} = +(1100)_2 = 01100_2$ in 2's comp.
  - $(1)_{10} = +(0001)_2 = 00001_2$ in 2's comp.

Step 1: Add binary numbers

Step 2: Ignore carry bit

```
            0 1 1 0 0
Add   +     0 0 0 0 1
      --------------
Final     0 0 1 1 0 1
Result
          Ignore
```

2's Complement Subtraction

- Using 2's complement numbers, follow steps for subtraction
- For example, suppose we wish to subtract $+(0001)_2$ from $+(1100)_2$.
- Let's compute $(12)_{10}$ - $(1)_{10}$.
  - $(12)_{10}$ = $+(1100)_2$ = 01100_2 in 2's comp.
  - $(-1)_{10}$ = $-(0001)_2$ = 11111_2 in 2's comp.

```
        0 1 1 0 0
  -   [ 0 0 0 0 1 ]
      -------------
```

2's comp

```
        0 1 1 0 0
  +     1 1 1 1 1
      -------------
      1 [ 0 1 0 1 1 ]
```

Add

Final Result

Ignore Carry

Step 1: Take 2's complement of 2nd operand

Step 2: Add binary numbers

Step 3: Ignore carry bit

2's Complement Subtraction: Example #2

- Let's compute $(13)_{10} - (5)_{10}$.
  - $(13)_{10} = +(1101)_2 = (01101)_2$
  - $(-5)_{10} = -(0101)_2 = (11011)_2$
- Adding these two 5-bit codes…

```
          0 1 1 0 1
+         1 1 0 1 1
-------------------
  1       0 1 0 0 0
carry
```

- Discarding the carry bit, the sign bit is seen to be zero, indicating a correct result. Indeed,

$(01000)_2 = +(1000)_2 = +(8)_{10}$.

2's Complement Subtraction: Example #3

- Let's compute $(5)_{10} - (12)_{10}$.
  - $(-12)_{10} = -(1100)_2 = (10100)_2$
  - $(5)_{10} = +(0101)_2 = (00101)_2$
- Adding these two 5-bit codes…
- Here, there is no carry bit and the sign bit is 1. This indicates a negative result, which is what we expect. $(11001)_2 = -(7)_{10}$.

```
      0 0 1 0 1
+     1 0 1 0 0
    -----------
      1 1 0 0 1
```

اعداد و کدهای باینری

- انواع اطلاعات
  - عددی
    - کدها باید رابطه با عدد داشته باشند.
    - معمولا امکان محاسبات ریاضی مطلوب است.
  - غیر عددی
    - چون نیازی به محاسبات نیست، انعطاف بیشتری دارند.
    - نیازی به رابطه ریاضی با اطلاعات کد شده وجود ندارد.

مثال کد باینری غیر عددی

- یک کد باینری **n** بیتی برای نمایش مجموعه ای از عناصر به کار می‌رود:
- مثلا برای هفت رنگ رنگین کمان نیاز به یک کد ۳ بیتی است، در جدول زیر هر رنگ با یک عدد باینری ۳ بیتی کد شده یا هر کد به رنگی تخصیص داده شده است .
- کد **100** بدون استفاده است.

Color	Binary Number
Red	000
Orange	001
Yellow	010
Green	011
Blue	101
Indigo	110
Violet	111

تعداد بیت مورد نیاز

- تعداد M عنصر و تعداد n بیت لازم برای کد کردن در رابطه زیر صدق می کند.

$$2^{n} \geq M > 2^{(n-1)}$$

$$x = \log_2 M$$

$n =$ عدد صحیح بزرگتر از x

رابطه تعداد عنصر کد شده با تعداد بیت کد **n**

- Given n digits in radix r, there are r^n distinct elements that can be represented.
- But, you can represent m elements, $m < r^n$

Examples:

- You can represent 4 elements in radix $r = 2$ with $n = 2$ digits: (00, 01, 10, 11).
- You can represent 4 elements in radix $r = 2$ with $n = 4$ digits: (0001, 0010, 0100, 1000).
- This second code is called a "one hot" code.

Binary Coded Decimal

Digit	BCD Code	Digit	BCD Code
0	**0000**	**5**	**0101**
1	**0001**	**6**	**0110**
2	**0010**	**7**	**0111**
3	**0011**	**8**	**1000**
4	**0100**	**9**	**1001**

- Binary coded decimal (BCD) represents each decimal digit with four bits
  - Ex. 0011 0010 1001 = 329_{10}

- This is ***NOT*** the same as 001100101001_2
- Why do this? Because people think in decimal.

مقایسه BCD با کد باینری

Decimal	Binary	Octal	Hexadecimal	BCD
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0100
15	1111	17	F	0001 0101

- **BCD not very efficient**
- **Used in early computers (40s, 50s)**
- **Used to encode numbers for seven-segment displays.**
- **Easier to read?**

کدهای دسیمال: کدهای باینری برای ارقام دسیمال:

Binary Codes for Decimal Digits

- ۸۰۰۰ روش برای انتخاب ۱۰ کد از ۱۶ عدد۴ بیتی وجود دارد. تعداد کمی مفید هستند:

کدهای وزنی، کدهای افزونی ۳، کد گری

Decimal	8,4,2,1	Excess3	8,4,-2,-1	Gray
0	0000	0011	0000	0000
1	0001	0100	0111	0001
2	0010	0101	0110	0011
3	0011	0110	0101	0010
4	0100	0111	0100	0110
5	0101	1000	1011	0111
6	0110	1001	1010	
7	0111	1010	1001	
8	1000	1011	1000	
9	1001	1100	1111	

Binary Coded Decimal (BCD)

- **The BCD code is the 8,4,2,1 code.**
- **8, 4, 2, and 1 are weights**
- **BCD is a *weighted* code**
- **This code is the simplest, most intuitive binary code for decimal digits and uses the same powers of 2 as a binary number, but only encodes the first ten values from 0 to 9.**
- **Example: 1001 (9) = 1000 (8) + 0001 (1)**
- **How many "invalid" code words are there?**
- **What are the "invalid" code words?**

GRAY CODE – Decimal

- What special property does the Gray code have in relation to adjacent decimal digits?

کد گری

- نکته: این کد می‌تواند برای اعدادبزرگتر از ۹ نیز بدست بیاید.
- به عنوان مثال برای عدد ۳۲ داریم:
  - کد BCD Gray :
    - BCD of 32= 0011 0010
    - Gray BCD of 32=0010   0011
  - کد Gray :
    - Binary of 32 = 100000
    - Gray code of 32= 110000

تبدیل به باینری و کد کردن

- **تذکر: معادل باینری یک عدد با کد معادل آن متفاوت است.**

- $\mathbf{13_{10} = 1101_2}$ (This is <u>conversion</u>)
- $\mathbf{13 \Leftrightarrow 0001|0011}$ (This is <u>coding</u>)

BCD Arithmetic

- **Given a BCD code, we use binary arithmetic to add the digits:**

8	**1000**	**Eight**
+5	**+0101**	**Plus 5**
13	**1101**	**is 13 (> 9)**

- **Note that the result is MORE THAN 9, so must be represented by two digits!**
- **To correct the digit, subtract 10 by adding 6 modulo 16.**

8	**1000**	**Eight**
+5	**+0101**	**Plus 5**
13	**1101**	**is 13 (> 9)**
	+0110	**so add 6**
	carry = 1 0011	**leaving 3 + cy**
	0001 | 0011	**Final answer (two digits)**

- **If the digit sum is > 9, add one to the next significant digit**

مثال جمع BCD

- **Add 2905_{BCD} to 1897_{BCD} showing carries and digit corrections.**

$$
\begin{array}{rcccc}
 & 0001 & 1000 & 1001 & 0111 \\
+ & \underline{0010} & \underline{1001} & \underline{0000} & \underline{0101} \\
 & \rule{2em}{1pt} & \rule{2em}{1pt} & \rule{2em}{1pt} & \rule{2em}{1pt}
\end{array}
$$

تمرین:

- برای تفریق و جمع BCD چهار مثال مختلف بزنید. در هر مثال یک حالت خطا موجود باشد.

ALPHANUMERIC CODES - ASCII Character Codes

- **American Standard Code for Information Interchange**

- **This code is a popular code used to represent information sent as character-based data. It uses 7-bits to represent:**
  - **94 Graphic printing characters.**
  - **34 Non-printing characters**
- **Some non-printing characters are used for text format (e.g. BS = Backspace, CR = carriage return)**

ASCII Code

	$B_7B_6B_5$							
$B_4B_3B_2B_1$	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

ASCII Properties

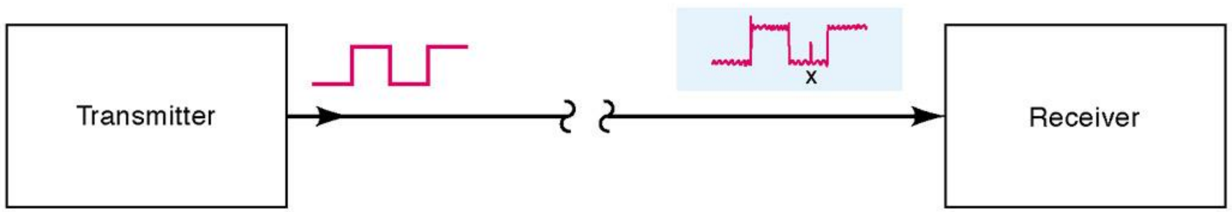
ASCII has some interesting properties:

- **Digits 0 to 9 span Hexadecimal values 30_{16} to 39_{16}.**
  - **Upper case A-Z span 41_{16} to $5A_{16}$.**
  - **Lower case a-z span 61_{16} to $7A_{16}$.**
    - **Lower to upper case translation (and vice versa) occurs by flipping bit 6.**

UNICODE

- **UNICODE extends ASCII to 65,536 universal characters codes**
  - **For encoding characters in world languages**
  - **Available in many modern applications**
  - **2 byte (16-bit) code words**
  - **See Reading Supplement – Unicode on the Companion Website http://www.prenhall.com/mano**

ASCII Codes and Data Transmission



- ASCII Codes
  - A – Z (26 codes), a – z (26 codes)
  - 0-9 (10 codes), others (@#$%^&*….)
  - Complete listing in Mano text
- Transmission susceptible to noise
- Typical transmission rates (1500 Kbps, 56.6 Kbps)
  - How to keep data transmission accurate?

کدهای تشخیص خطا:
Parity Codes

- کدهای پریتی از افزودن یک بیت به نام بیت پریتی به کلمه کد ایجاد می شوند.
- دو نوع کد پریتی وجود دارد: زوج و فرد

- In an *odd-parity code*, the parity bit is specified so that the total number of ones is odd.
- In an even*-parity code*, the parity bit is specified so that the total number of ones is even.

P	Information Bits

1 1 0 0 0 0 1 1
↑
Added even parity bit

0 1 0 0 0 0 1 1
↑
Added odd parity bit

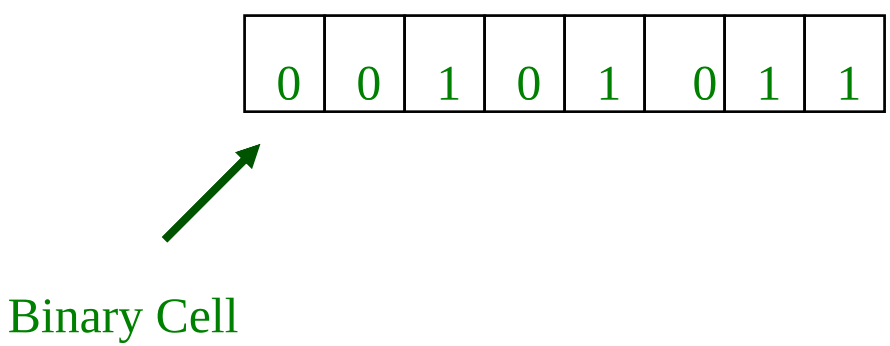
مثال کد پریتی

- Concatenate a parity bit to the ASCII code for the characters 0, X, and = to produce both odd-parity and even-parity codes.

Character	ASCII	Odd-Parity ASCII	Even-Parity ASCII
0	0110000	10110000	00110000
X	1011000	01011000	11011000
=	0111100	10111100	00111100

ذخیره داده های باینری

- **Binary cells store individual bits of data**
- **Multiple cells form a register.**
- **Data in registers can indicate different values**
  - **Hex (decimal)**
  - **BCD**
  - **ASCII**



ثبات یا رجیستر : انتقال و پردازش

- داده های درون ثبات ها قابل انتقال به ثبات دیگر هستند.
- مدارهای منطقی پردازش داده را به عهده دارند.
- هدف این درس آشنایی با این مدارها و نیز روش طراحی و تحلیل آن ها است.