EEL3701

Menu

- Review of number systems
  - >Binary math
  - >Signed number systems

1

EEL3701

Review of Number Systems

Our decimal (**base** 10 or **radix** 10) number system is positional.

Ex: $9437_{10} = 9 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 7 \times 10^0$

We have a total of (R=10) digits, i.e., {0,1,2,3,4,5,6,7,8,9}; where R = radix

Similarly for R=2, 8, 16; $R=2^1$ is called **Binary**; $R=2^3$ is called **Octal**; $R=2^4$ is called **Hexadecimal** (or **Hex**)

For R > 10 we need additional symbols, e.g., for R=16 we need 6 additional symbols 0,1,2,3,4,5,6,7,8,9 and A,B,C,D,E,F for 10-15

2

EEL3701

Review of Number Systems

For example,

$123_{10} = 1x10^2 + 2x10^1 + 3x10^0$

$123_8 = 1x8^2 + 2x8^1 + 3x8^0$ or $123_8 = 83_{10}$

$123_{16} = 1x16^2 + 2x16^1 + 3x16^0$ or $123_{16} = 291_{10}$

EEL3701

Review of Number Systems

Fractions are also represented positionally as weighted negative powers of the radix or base.

Ex: $0.125_{10} = 0x10^0 + 1x10^{-1} + 2x10^{-2} + 5x10^{-3}$

Thus, for example,

$0.125_8 = 0x8^0 + 1x8^{-1} + 2x8^{-2} + 5x8^{-3}$

$0.125_{16} = 0x16^0 + 1x16^{-1} + 2x16^{-2} + 5x16^{-3}$

In general, if R>1, any **rational** number N can be represented in a power series given by:

$N = (d_4, d_3, d_2, d_1, d_0 , d_{-1} , d_{-2} , d_{-3})_R$

$N=d_4xR^4 + d_3xR^3 + d_2xR^2 + d_1xR^1 +d_0 + d_{-1}xR^{-1} + d_{-2}xR^{-2} + d_{-3}xR^{-3}$

$$N = \sum_{i=-k}^{m} d_i R^i$$

m= # of digits in the integer part - 1

k =# of digits in the fractional part

EEL3701

Decimal Numbers

$953.78_{10} = (d_2, d_1, d_0, d_{-1}, d_{-2})_{10}$ R=10; m=2; k=2

$953.78_{10} = 9x10^2 + 5x10^1 + 3x10^0 + 7x10^{-1} + 8x10^{-2}$

$d_i = (9,5,3,7,8)$

What are: $953.78_{10} = \_\_\_\_\_\__2$ $953.78_{10} = \_\_\_\_\_\__8$

$953.78_{10} = \_\_\_\_\_\__{16}$

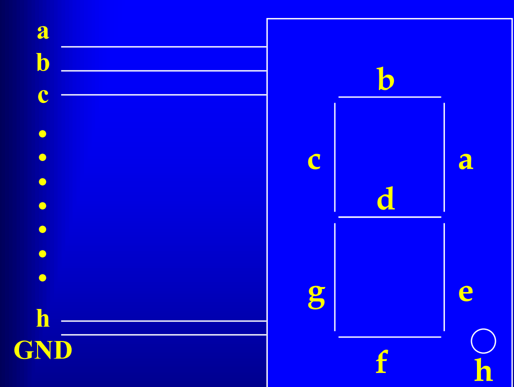
We'll get to this later.

An 7-segment display LED can represent all 16 hex symbols.

5

EEL3701

7-Segment LED



- Show 7-segment LED in LogicWorks, "7-Segment Display *"
- See also the LogicWorks 4-output device called "Hex Keyboard"
- See also the LogicWorks 4-input device called "Hex Display"

abc defg
A = 111 1101

abc defg
F = <u>011 1001</u>

6

EEL3701

How many bits?

Example: Let N=50 and R=2. We can use R=2 symbols, i.e., {0,1}.

$50_{10} = (d_5, d_4, d_3, d_2, d_1, d_0)_2$ Why?

$2^5 = 32$ and $2^6 = 64$ and $32 \le$ **50** < 64

$50_{10} = d_5 \times 2^5 + d_4 \times 2^4 + d_3 \times 2^3 + d_2 \times 2^2 + d_1 \times 2^1 + d_0$

Question: What is the range of unsigned numbers you can represent with ... ?

	Low	High	How Many?
4 bits	0	15	16
7 bits	0	127	128
8 bits	0	255	256

University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo

7

7

EEL3701

Decimal to Binary

Q: $50_{10} = ?_2$

- To obtain the binary digits (d_i's), use long division and save each remainder until the quotient equals 0.
- d_i's = the remainders in ***reverse*** order.

```
50÷2: q=25, r=0
25÷2: q=12, r=1
12÷2: q=6,  r=0
 6÷2: q=3,  r=0
 3÷2: q=1,  r=1
 1÷2: q=0,  r=1
```

d_i's = $\{110010\}_2$

$= 2^5 + 2^4 + 2^1$

$= 32 + 16 + 2 = 50$

STOP!

University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo

8

8

EEL3701

Decimal to Hex

- Hex is a grouping of 4 bits
- Octal is a grouping of 3 bits
- Ex: $10100110_2 = ?_{16} = ?_8$
  - >First regroup the binary as needed
  - >Then convert using table of hex or octal values
  - >For Hex (group in 4's):
    - – $1010\ 0110_2$ = **$A6_{16}$**
  - >For Octal (group in 3's):
    - – $10\ 100\ 110_2$ = **246_8**

Decimal	Binary	Hex	Octal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
7	111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
15	1111	F	17
16	1 0000	10	20
17	1 0001	11	21
32	10 0000	20	40
37	10 0101	25	45
42	10 1010	2A	52

EEL3701

Decimal to Hex

- Can convert to hex or octal using the <u>**same**</u> technique
  - $50_{10} = X_{16}$? R = 16; 50 ÷ 16: q = 3 , r = 2 , 3 ÷ 16: q= 0 , r = 3 So $50_{10} = 32_{16}$
  - Check: $3\times16^1 + 2\times16^0 = 48+2 = 50!$ ✓
  - $50_{10} = X_8$? R = 8; 50 ÷ 8: q = 6 , r = 2 , 6 ÷ 8: q= 0 , r = 6 . So $50_{10} = 62_8$
  - Check: 6 × 8 = 48 48+2 = 50! ✓
- Now $32_{16} = 0011\ 0010 = 1x2^5 + 1x2^4 + 0x2^3 + 0x2^2 + 1x2^1$ =32+16+2 = 50 √
- Grouping in 3 bits each $32_{16} = 00\ 110\ 010 = 062_8 = 62_8$
- Therefore converting **from Hex** to Octal or to Binary is <u>**Trivial!**</u>
- Example: 126_{16} = 0001 0010 0110 $_2$? = 446 $_8$?

EEL3701

Review of Number Systems

- Trivial Conversions: It is easy to convert from Binary/Hex/Octal to a Binary or Hex or Octal number
- Since $16 = 2^4$ and $8 = 2^3$, then to convert from
  - >Binary to Octal we group bits in groups of three
  - >Binary to Hex we group bits in groups of four

University of Florida, EEL 3701 – File **06**
© Drs. Schwartz & Arroyo
11

11

EEL3701

Review of Number Systems

Ex: $10101100_2 \rightarrow 10\ 101\ 100_2$
$\rightarrow 010\ 101\ 100_2 \rightarrow 254_8$

Ex: $10101100_2 \rightarrow 1010\ 1100_2 \rightarrow AC_{16}$

- It's easier to convert Dec→**Hex**→Bin then Dec→Bin directly

Ex: $100_{10} = \{100 \div 16 \rightarrow q=6;\ r=4$
$6 \div 16 \rightarrow q=0;\ r=6\} = 64_{16}$

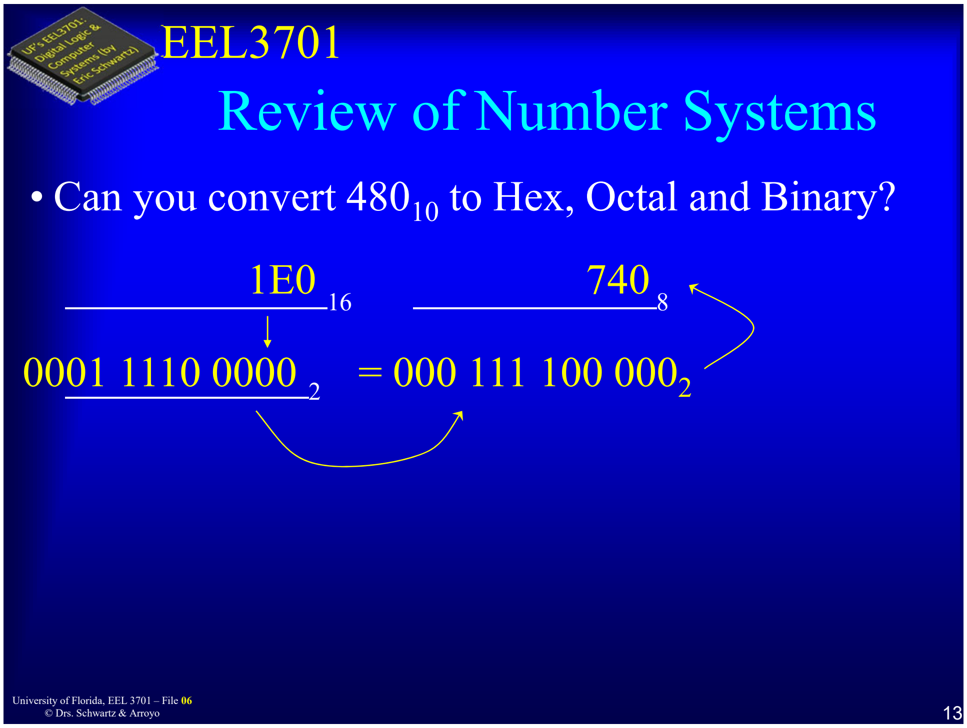
$64_{16} = 0110\ \ 0100_2 = 64 + 32 + 4 = 100$ ✓

$64_{16} = 01\ 100\ 100_2 = 144_8 = 64 + 4{*}8 + 4 = 100$ ✓

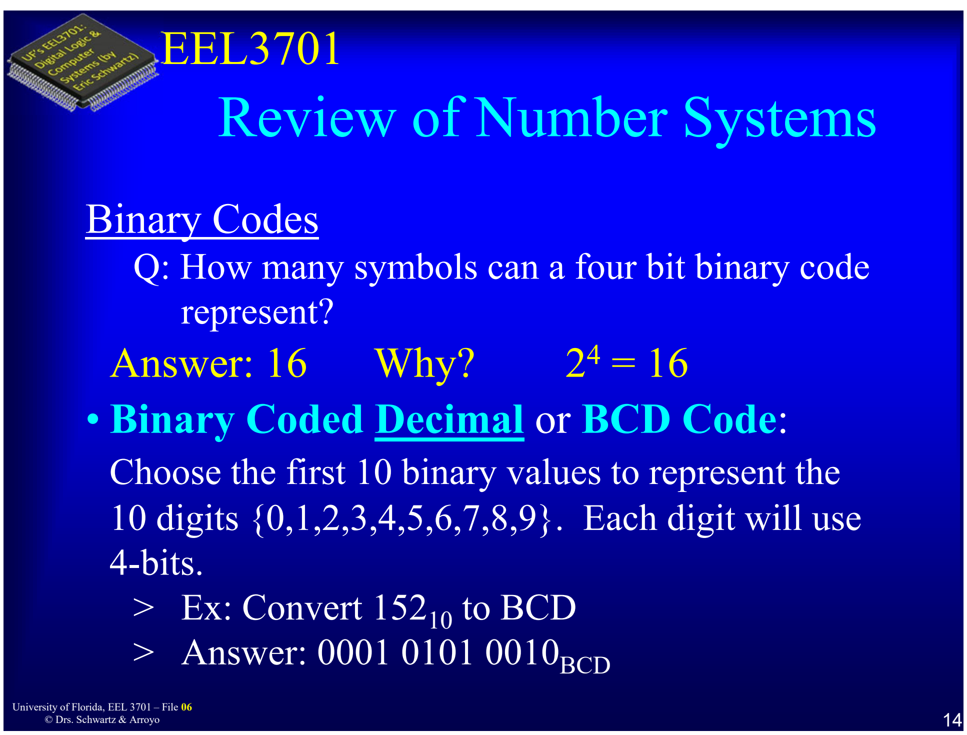
- From above you can also see that it may be easier to convert Bin→**Hex**→Dec (or Bin→**Oct**→Dec) then Bin→Dec directly

University of Florida, EEL 3701 – File **06**
© Drs. Schwartz & Arroyo
12

12


EEL3701
Review of Number Systems
• Can you convert 480$_{10}$ to Hex, Octal and Binary?
1E0$_{16}$
740$_{8}$
0001 1110 0000$_{2}$ = 000 111 100 000$_{2}$
University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo
13

13


EEL3701
Review of Number Systems
Binary Codes
Q: How many symbols can a four bit binary code represent?
Answer: 16 Why? $2^4 = 16$
• Binary Coded Decimal or BCD Code:
Choose the first 10 binary values to represent the 10 digits {0,1,2,3,4,5,6,7,8,9}. Each digit will use 4-bits.
> Ex: Convert 152$_{10}$ to BCD
> Answer: 0001 0101 0010$_{BCD}$
University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo
14

14

EEL3701

Review of Number Systems

- Ex: Convert 1010100101001_{BCD} to Decimal
  $1\ 0101\ 0010\ 1001_{BCD}$ = ?= $\mathbf{1529_{10}}$
- Ex: Convert 1010110101001_{BCD} to Decimal
  $1\ 0101\ \boxed{1010}\ 1001_{BCD}$ = ?= $\mathbf{\%!?\#_{10}}$
- Since the weights of the 4 digits are 8, 4, 2 and 1, BCD is often called 8-4-2-1 code.
- *Modern digital computers & micros include instructions to allow us to manipulate BCD numbers as if they were binary numbers.*
  - One common **Y2K** problem is strongly influenced by BCD

University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo
15

15

EEL3701

Addition, Subtraction, Multiplication

- Binary Math: We can manipulate Binary, Octal and Hexadecimal numbers like decimal numbers with respect to: addition, subtraction, multiplication and division.
- Examples:

```
           Dec              Dec                  Dec
   0101      5      1000     8         1010      10
 + 1101    +13     -0010    -2       × 0011     × 3
 1 0010     18      0110     6         1010      30
                                     1 010
                                     1 1110
```

University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo
16

16

EEL3701

Division with Binary Numbers

- Example: $11010_2 \div 100_2$ = ?

```
100/11010

      1
100/11010
   -100
    01010

     11
100/11010
   -100
     101
    -100
     010

    110
100/11010
   -100
     101
    -100
     010
```

Answer=110.1_2

Answer=110_2 R10_2

Dec

```
    6 R2
4/26
 -24
   2
```

26÷4=**6 R2**

EEL3701

Decimal Fraction Conversion

- To convert a decimal fraction to any other base, just multiply the decimal fraction by the radix and save the integer
- Ex: Convert 0.37_{10} to binary
  - >0.37 * 2 = **0**.74 => $0.\mathbf{0}_2$
  - >0.74 * 2 = **1**.48 => $0.0\mathbf{1}_2$
  - >0.48 * 2 = **0**.96 => $0.01\mathbf{0}_2$
  - >0.96 * 2 = **1**.92 => $0.010\mathbf{1}_2$
- Ex: Convert 0.37_{10} to hex
  - >0.37 * 16 = **5**.92 => $0.\mathbf{5}_{16}$
  - >0.92 * 16 = **14**.72 => $0.5\mathbf{E}_{16}$
  - >0.72 * 16 = **11**.52 => $0.5E\mathbf{B}_{16}$
  - >0.52 * 16 = **8**.32 => $0.5EB\mathbf{8}_{16}$

Note that conversion to hex also gives binary (or octal): 0.37_{10} = $0.0101\ 1110\ 1011\ 1000_2$

EEL3701

Signed Number Representations

- Negative Binary Numbers
  - >**Signed-Magnitude**: Treats the most significant bit (MSB) as the sign of the number (0 = +, 1 = –)
  - >**One's Complement**: Changes every bit from 0 to 1 and 1 to 0 (for negative numbers)
  - >**Two's Complement**: One's Complement + 1
- A **4-bit 2's complement number** $d_3d_2d_1d_0$ has value
  - >$(d_3d_2d_1d_0)_{\text{4-bit 2's comp}} = d_3\times-(2^3) + d_2\times2^2 + d_1\times2^1 + d_0\times2^0$
- A **3-bit 2's complement number** $d_2d_1d_0$ has value
  - >$(d_2d_1d_0)_{\text{3 bit 2's comp}} = d_2 \times -(2^2) + d_1 \times 2^1 + d_0 \times 2^0$

EEL3701

Signed Numbers

Examples of positive and negative number representations, **using only 4 bits**

Decimal	Sign Mag.	1's Compl.	2's Compl.
3	0011	0011	0011
-3	1011	1100	1101
7	0111	0111	0111
-7	1111	1000	1001
9	garbage	garbage	garbage

Note: Given an n-bit binary number N

- $N + N_{\text{1's compl}}$ = 1111 ... 111 = 2^n-1 (n ones)
- $N + N_{\text{2's compl}}$ = 1 0000 ... 000 = 2^n (a 1 followed by n zeros)

EEL3701 2's Comp Numbers

- To convert N to its 2's complement form, it may be easier to subtract it from 2^n or from $2^n - 1$ and add 1
- Example: Convert -25 to **8-bit** 2's complement:

  $25 = 16 + 8 + 1 = 0001\ 1001_2 = \$19$

  Tech 1: Complement then add 1

  1110 0110 + 1 or 1110 0111 or \$E7 ✓

  Tech 2: Subtract for 2^n

  256 - 25 = 231 = (128+64+32) + (7) = \$E7 ✓

  Tech 3: Subtract binary from 11…1, then add 1

  1111 1111 - 0001 1001 = \$FF - \$19 = \$E6

  \$E6 + 1 = \$E7 ✓

  No borrows in this subtraction, so easy

University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo

21

EEL3701 Subtraction and 2's complement

- Suppose you do the following: `6-7=?`
- You solve this by doing: `-(7-6)= -(1)= -1`
- In 2's comp. math we don't have to subtract
- Instead we can just **negate and add**
- Example: 6-7=? (using 4-bit 2's comp.)

  $6=0110_2$, $7=0111_2$, $-7=1001_{2(2's\ comp)}$

  $6-7 = 6+(-7) = \quad 0110_{2(2's\ comp)}$

  $+1001_{2(2's\ comp)}$

  2's comp. #'s → $\mathbf{1111_{2(2c)}} = -2^3+2^2+2^1+2^0 = -1$ √

- Alternative is $-(7-6) = -(0111_2 - 0110_2) = -(0001_2)$

  = 2's comp $(0001) = \mathbf{1111_{2(2c)}}$

University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo

22

EEL3701

Sign Extension

- When you want a two's complement signed number to have more bits, you use **Sign Extension**
  - > Keep the most significant bit for all of the added bits
  - > **Examples**:
    Make 5-bit 2's complement numbers into 8-bit 2's complement numbers
    - 1_{10} = $0\ 0001_{\text{5-bit 2's comp}}$ → $0000\ 0001_{\text{8-bit 2's comp}}$
    - -1_{10} = $1\ 1111_{\text{5-bit 2's comp}}$ → $1111\ 1111_{\text{8-bit 2's comp}}$
    - 12_{10} = $0\ 1100_{\text{5-bit 2's comp}}$ → $0000\ 1100_{\text{8-bit 2's comp}}$
    - -4_{10} = $1\ 1100_{\text{5-bit 2's comp}}$ → $1111\ 1100_{\text{8-bit 2's comp}}$

EEL3701

Signed Operations -vs- Representations

- 1's complement and 2's complement are both **operations** and **representations**

- Operations
  - > 4-bit 1's comp(1001)=0110
  - > 4-bit 2's comp(1001)=0111
- Representations
  - > $1001_{\text{4-bit 1's comp}}$
    - MSBit is 1 so negative
    - $1001_{\text{4-bit 1's comp}} = -(0110) = -6_{10}$
  - > $1001_{\text{4-bit 2's comp}}$ = ?
    - MSBit is 1 so negative
    - Complement: 0110;
    - Add 1: $0111 = 7_{10}$
    - $1001_{\text{4-bit 2's comp}} = -7_{10}$

- Operations
  - > 4-bit 1's comp(0110)=1001
  - > 4-bit 2's comp(0110)=1010
- Representations
  - > $0110_{\text{4-bit 1's comp}}$
    - MSBit is 0 so positive
    - $0110_{\text{4-bit 1's comp}} = +(0110) = 6_{10}$
  - > $0110_{\text{4-bit 2's comp}}$ = ?
    - MSBit is 0 so positive
    - $0110_{\text{4-bit 2's comp}} = 0110 = 6_{10}$

EEL3701

Overflow (2's Comp)

- An **overflow** occurs when
  - > Two positive numbers are added and the result appears to be negative
  - > Two negative numbers are added and the result appears to be positive
  - > Subtraction with the same strange results as above
- Example using 4-bit 2's complement numbers
  - > What is $3_{10} + 7_{10}$ in 4-bit 2's complement?
  - > $3_{10} = 0011_{\text{4-bit 2's comp}}$ and $7_{10} = 0111_{\text{4-bit 2's comp}}$

$$\begin{array}{r} 0011_{2(2c)} \\ +\ 0111_{2(2c)} \\ \hline 1010_{2(2c)} \end{array}$$

But $1010_{\text{4-bit 2's comp}}$ is negative (= -8 + 2 = -6)
==> **OVERFLOW**

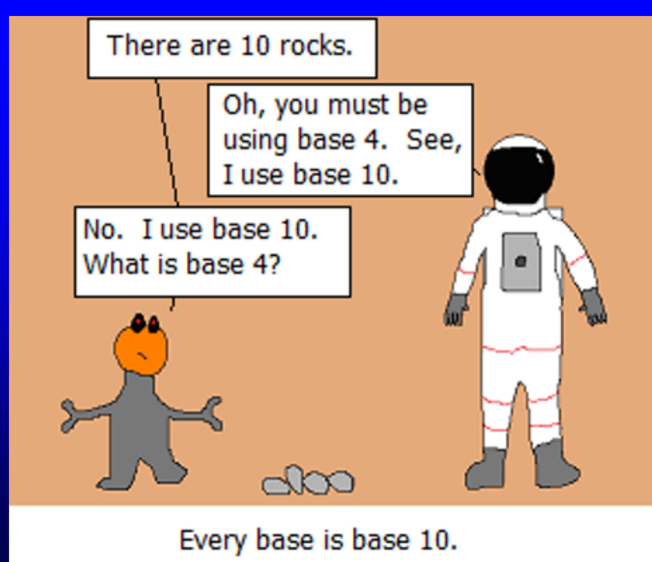
  - > Now try -3 + -7

University of Florida, EEL 3701 – File **06**
© Drs. Schwartz & Arroyo 25

25

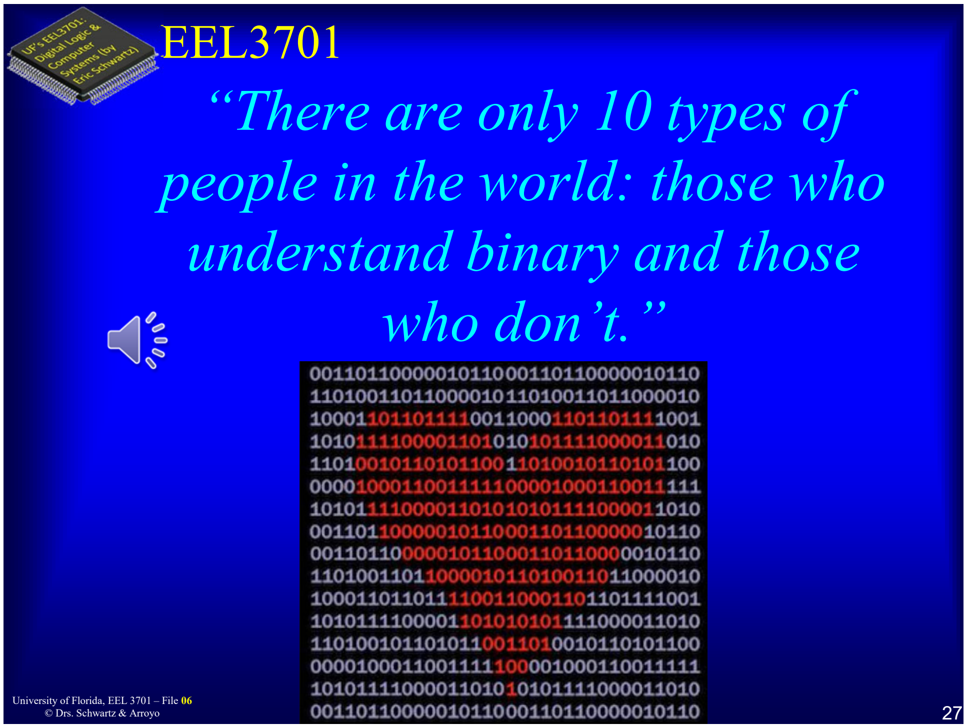
EEL3701

Alien Number Systems

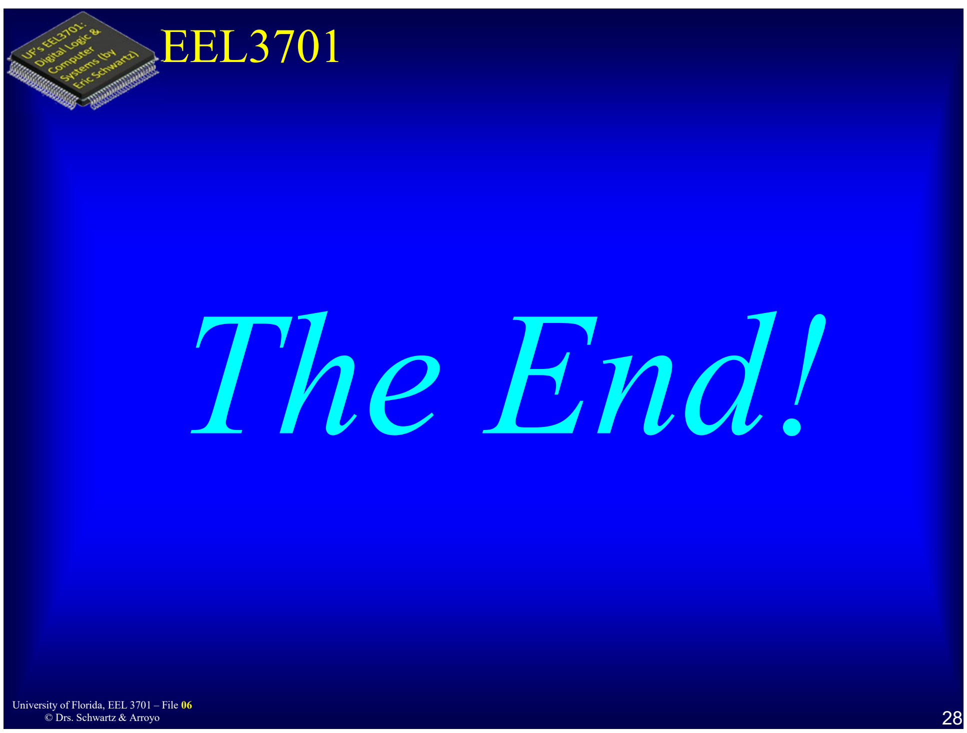


University of Florida, EEL 3701 – File **06**
© Drs. Schwartz & Arroyo 26

26


UF's EEL3701: Digital Logic & Computer Systems (by Eric Schwartz)
EEL3701
"There are only 10 types of people in the world: those who understand binary and those who don't."
University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo
27

27


UF's EEL3701: Digital Logic & Computer Systems (by Eric Schwartz)
EEL3701
The End!
University of Florida, EEL 3701 – File 06
© Drs. Schwartz & Arroyo
28

28