Lecture 2: Data Representation

Professor Peter Cheung
Department of EEE, Imperial College London

(Floyd 2.1-2.4, 2.8, 2.10-2.11)
(Tocci 2.1 – 2.8)

Points Addressed in this Lecture

- What do we mean by data?
- How can data be represented electronically?
- What number systems are often used and why?
- How do number systems of different bases work?
- How do you convert a number between binary and decimal?

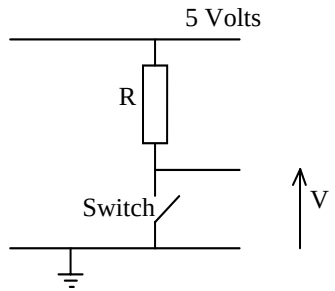
What do we mean by data?

  - Many definitions are possible depending on context

- We will say that:
  - data is a physical representation of information
- Data can be stored
  - e.g.: computer disk, cash till
- Data can be transmitted
  - e.g.: fax
- Data can be processed
  - e.g.: cash till

Electronic Representation of Data

- Information can be very complicated
  - e.g.:
    - Numbers Sounds
    - Pictures Codes
  - We need a simple electronic representation
- What can we do with electronics?
  - Set up voltages and currents
  - Change the voltages and currents
- A useful device is a switch
  - Switch Closed: V = 0 Volts
  - Switch Open: V = 5 Volts

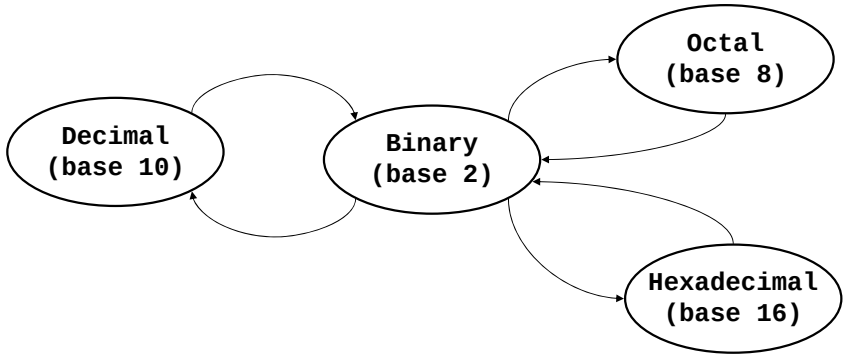


- Information can be represented by a voltage level
- The simplest information is TRUE/FALSE
  - This can be represented by two voltage levels:
    - 5 Volts for TRUE
    - 0 Volts for FALSE
- A voltage signal which has only two possibilities is a BIT
  - Bit stands for Binary Digit
- Binary means: only 2 possible values

FALSE	TRUE
(0)	(1)

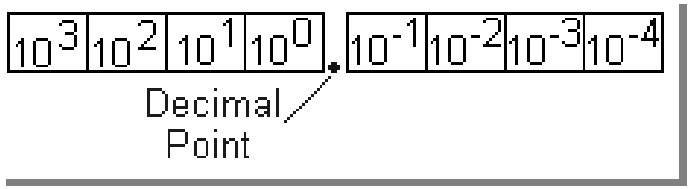
- Advantages of using binary representation
  - simple to implement in electronic hardware (switch)
  - good tolerance to noise

Number System Overview



Decimal Numbers

- The decimal number system has ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9
- The decimal numbering system has a base of 10 with each position weighted by a factor of 10:



Binary Numbers

- The binary number system has two digits: 0 and 1
- The binary numbering system has a base of 2 with each position weighted by a factor of 2:

POSITIVE POWERS OF TWO (WHOLE NUMBERS)									NEGATIVE POWERS OF TWO (FRACTIONAL NUMBER)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0.5	0.25	0.125	0.0625	0.03125	0.015625

Binary Number System

- Uses 2 symbols by our previous rule
  - 0 and 1
- Example: 10011 in binary is

$1 \times 2^4 + 1 \times 2^1 + 1 \times 2^0 = 19$

2^4	2^3	2^2	2^1	2^0
1	0	0	1	1

- Binary is the base 2 number system
- Most common in digital electronics

Integer and Fractional Parts

- Binary numbers can contain fractional parts as well as integer parts

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
1	0	0	1	1	0	1	1

Binary Point

$(19.375)_{10}$

- This 8-bit number is in Q3 format
  - 3 bits after the binary point
- How could 19.376 best be represented using an 8-bit binary number?
  - Quantization error

Conversion: decimal to binary (Method 1)

- The decimal number is simply expressed as a sum of powers of 2, and then 1s and 0s are written in the appropriate bit positions.

$$\begin{aligned} 50_{10} &= 32 + 18 \\ &= 32 + 16 + 2 \\ &= 1\times 2^5 + 1\times 2^4 + 1\times 2^1 \\ 50_{10} &= 110010_2 \end{aligned}$$

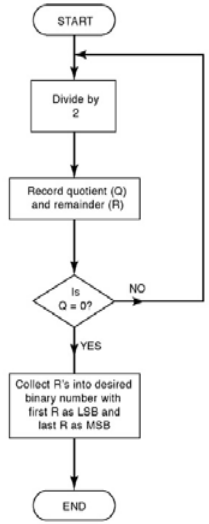
$$\begin{aligned} 346_{10} &= 256 + 90 \\ &= 256 + 64 + 26 \\ &= 256 + 64 + 16 + 10 \\ &= 256 + 64 + 16 + 8 + 2 \\ &= 1\times 2^8 + 1\times 2^6 + 1\times 2^4 + 1\times 2^3 + 1\times 2^1 \\ 346_{10} &= 101011010_2 \end{aligned}$$

Conversion: decimal to binary (method 2)

- Repeated division

		quotient	remainder	
50/2	=	25	0	LSB
25/2	=	12	1	
12/2	=	6	0	
6/2	=	3	0	
3/2	=	1	1	
1/2	=	0	1	MSB

$50_{10}=110010_2$



Conversion: binary to decimal

- The simplest way is to represent the binary number as
  $a_n \times 2^{n-1} + \ldots + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0$
- The conversion can be done by substituting the a's with the given bits then multiplying and adding:
  - eg: Convert $(1101)_2$ into decimal
  - $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (13)_{10}$
- Other algorithms can be used as alternatives if you prefer

Binary Addition

- First recall decimal addition

	1	1	1	
A	1	2	3	4
+ B		9	8	7
Sum	2	2	2	1

- In binary addition we follow the same pattern but
  - 0 + 0 = 0 carry-out 0
  - 0 + 1 = 1 carry-out 0
  - 1 + 0 = 1 carry-out 0
  - 1 + 1 = 0 carry-out 1
  - 1 + 1 + carry-in = 1 carry-out 1

		1		
A	0	1	1	1
+ B	0	1	1	0
Sum	1	1	0	1

- Note that we need to consider 3 inputs per bit of binary number
  - A, B and carry-in
- Each bit of binary addition generates 2 outputs
  - sum and carry-out

Hexadecimal Numbers

- Decimal, binary, and hexadecimal numbers

DECIMAL	BINARY	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Hexadecimal Numbers conversions

- **Binary-to-hexadecimal conversion**
  1. Break the binary number into 4-bit groups
  2. Replace each group with the hexadecimal equivalent

- **Hexadecimal-to-decimal conversion**
  1. Convert the hexadecimal to groups of 4-bit binary
  2. Convert the binary to decimal

- **Decimal-to-hexadecimal conversion**
  – Repeated division by 16

Binary Coded Decimal (BCD)

- Use 4-bit binary to represent one decimal digit
- Easy conversion
- Wasting bits (4-bits can represent 16 different values, but only 10 values are used)
- Used extensively in financial applications

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

Binary Coded Decimal (BCD)

- Convert 0110100000111001(BCD) to its decimal equivalent.

  0110 1000 0011 1001
  6 8 3 9

- Convert the BCD number 011111000001 to its decimal equivalent.

  0111 1100 0001
  7 ↑ 1

  The forbidden code group indicated an error

Putting it together

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

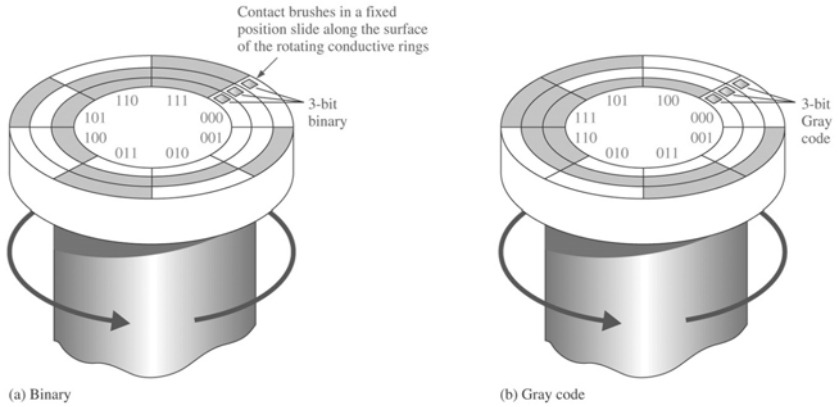
Gray Codes

- Only 1 bit changes in the count sequence
- Useful for industrial control

DECIMAL	BINARY	GRAY CODE
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001

Gray Codes

- Binary code results in glitches
- Gray code avoids glitches



ASCII code

- Codes representing letters of the alphabet, punctuation marks, and other special characters as well as numbers are called *alphanumeric* codes.
- The most widely used alphanumeric code is the American Standard Code for Information Interchange(ASCII). The ASCII (pronounced “askee”) code is a seven-bit code.

Character	Seven-Bit ASCII	Octal	Hex	Character	Seven-Bit ASCII	Octal	Hex
A	100 0001	101	41	Y	101 1001	131	59
B	100 0010	102	42	Z	101 1010	132	5A
C	100 0011	103	43	0	011 0000	060	30
D	100 0100	104	44	1	011 0001	061	31
E	100 0101	105	45	2	011 0010	062	32
F	100 0110	106	46	3	011 0011	063	33
G	100 0111	107	47	4	011 0100	064	34
H	100 1000	110	48	5	011 0101	065	35
I	100 1001	111	49	6	011 0110	066	36
J	100 1010	112	4A	7	011 0111	067	37
K	100 1011	113	4B	8	011 1000	070	38
L	100 1100	114	4C	9	011 1001	071	39
M	100 1101	115	4D	blank	010 0000	040	20
N	100 1110	116	4E	.	010 1110	056	2E
O	100 1111	117	4F	(	010 1000	050	28
P	101 0000	120	50	+	010 1011	053	2B
Q	101 0001	121	51	$	010 0100	044	24
R	101 0010	122	52	*	010 1010	052	2A
S	101 0011	123	53	)	010 1001	051	29
T	101 0100	124	54	—	010 1101	055	2D
U	101 0101	125	55	/	010 1111	057	2F
V	101 0110	126	56	,	010 1100	054	2C
W	101 0111	127	57	=	011 1101	075	3D
X	101 1000	130	58	⟨RETURN⟩	000 1101	015	0D
				⟨LINEFEED⟩	000 1010	012	0A