Digital Logic Design

(EL 114)

DR. RAJIB LOCHAN DAS

PhD: IIT Kharagpur

Research Interest: Adaptive and Digital Signal Processing,

Compressive Sensing and

Image Processing

Teaching Interest: Signals and Systems,

Digital Signal Processing,

Analog Electronics,

Circuit Theory,

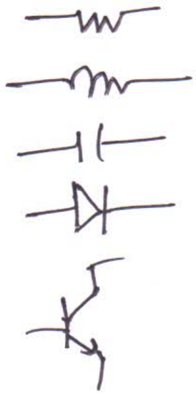
Digital Electronics,

Control Systems,

VLSI, ...

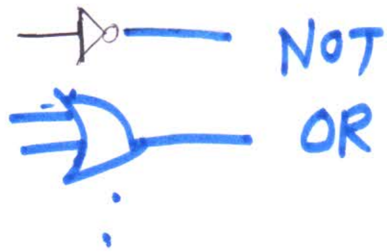
Electronics

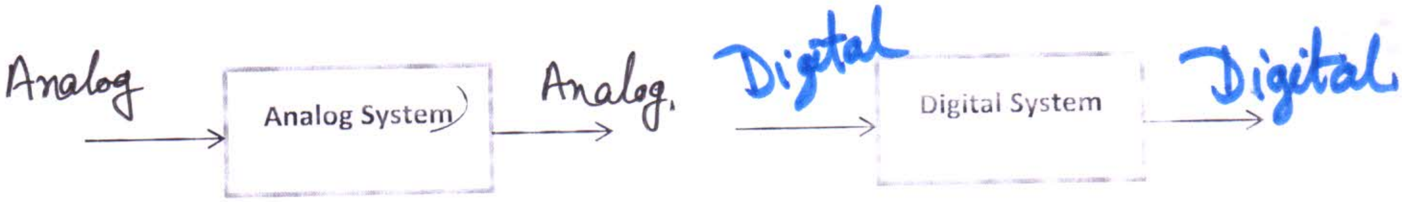
Analog Electronics

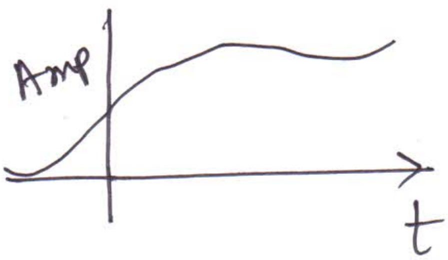


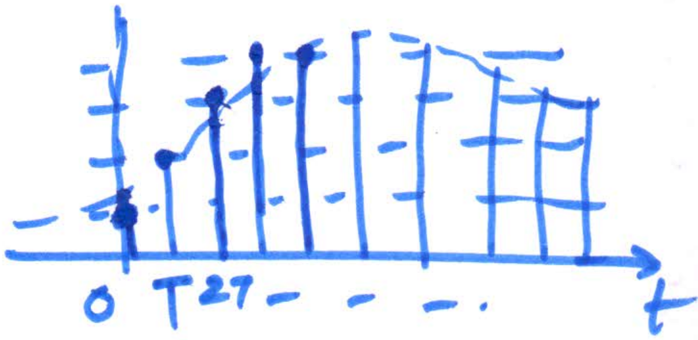
Digital Electronics

Logic gate



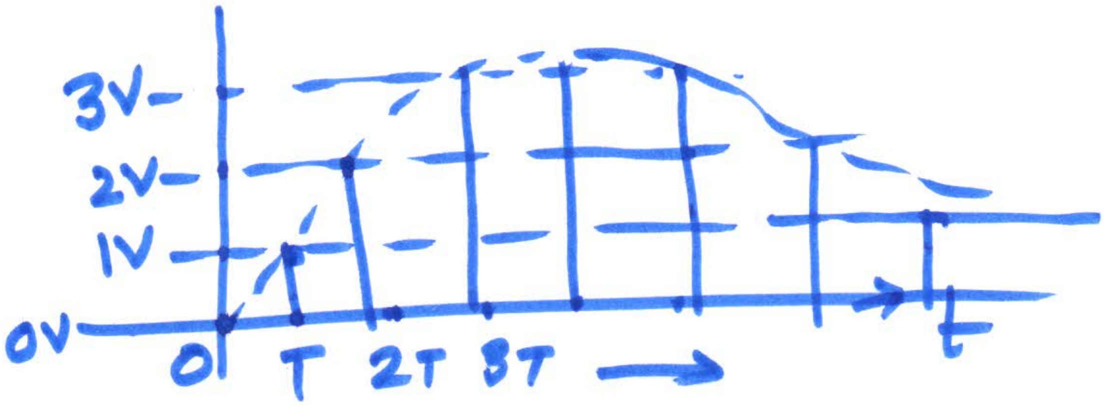






0.943

0.9432



$$\overbrace{\quad}^{2}$$

$$4\left\{\begin{array}{lcl} 0V & \to & 00 \\ 1V & \to & 01 \\ 2V & \to & 10 \\ 3V & \to & 11 \end{array}\right.$$

L - levels

$$\underline{\underline{L = 2^n}}$$

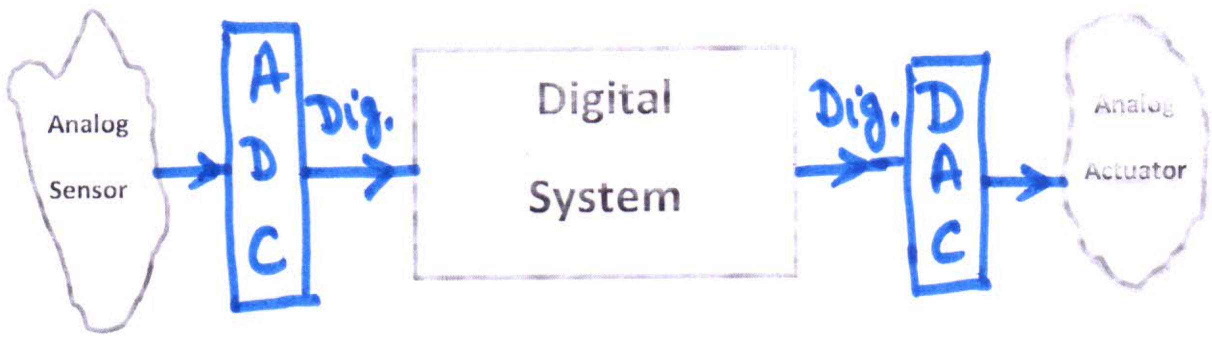
$$\underline{\underline{2^n \geq L}}$$

$$\begin{array}{lcl} 0 & \to & 000 \\ 1 & \to & 001 \\ 2 & \to & 010 \\ \vdots & \to & 011 \\ & & 100 \\ & & 101 \\ & & 110 \\ 7 & \to & 111 \end{array}$$

Advantages of Digital Systems:

1. Much easier to design
2. Higher accuracy
3. Programmable (may be)
4. Better noise immunity
5. Easier data storage

However, the real world is analog.



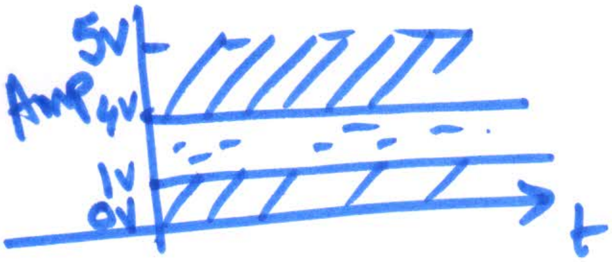
Generally, digital systems perform logical operations and arithmetic operations (computation) in binary number system.

In binary number system, we have only two valid symbols:

0 and 1 → bit

Why binary?

bit	Voltage	Logical
0	0v	F
1	5v	T



In decimal number system, we have ten symbols:

0,1,2,3,4,5,6,7,8,9

Numbers in decimal:

0, 1, . . . , 9, 10, 11, ... , 99, 100

Numbers in binary:

0, 1, 10, 11, 100, 101, 110, 111, 1000, ...

2 3 4 5 6 7 8

Binary	Decimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21

$2^n - 1$

Representation of positive numbers

1. Decimal Number System:

Ex. 5816

$= 5\times10^3 + 8\times10^2 + 1\times10 + 6$

5 4 3 2 1 0

Ex. 29380.71

$= 2\times10^4 + 9\times10^3 \times 3\times10^2 \times 8\times10^1 + 0 + 7\times10^{-1}$
$+ 1\times10^{-2}$

2. Binary Number System:

4 3 2 1 0

Ex. $(10101)_2 = (21)_{10}$

$= 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1$

$= 16 + 4 + 1 = 21$

Ex. $(1101.01)_2$

$= 8 + 4 + 1 + \frac{0}{2} + \frac{1}{4}$

$= 13 + 0.25 = 13.25$

In general, any number N= ($a_{n-1}\ a_{n-2}\ a_{n-3}\ a_{n-4}\ldots\ a_1\ a_0.\ a_{-1}\ a_{-2}\ldots\ a_{-m})_r$ with radix r ($0 \le a_i < r$, $-m \le i \le n-1$) can be written as

N =

$$a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + .. + a_1 r + a_0 + a_{-1} r^{-1} + ... + a_{-m} r^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i r^i$$

Commonly used number systems:

Radix r	Number System
2	Binary
8	Octal
10	Decimal
16	Hexadecimal

Octal and hexadecimal number systems are used as shorthand systems.

Octal Number System

Symbols: 0, 1, 2, 3, 4, 5, 6, 7

Ex. $(161)_8 = (113)_{10}$

$$= 1\times 8^2 + 6\times 8 + 1 = 64 + 48 + 1 = 113$$

Ex. $(34.4)_8 = (28.5)_{10}$

$$= 3\times 8 + 4 + \frac{4}{8} = 28.5$$

Hexadecimal Number System

Symbols: 0, 1, 2, . . . , 9, A, B, C, D, E, F

10 11 12 13 14 15

Ex. $(A1)_{16} = (161)_{10}$

$$= 10\times 16 + 1 = 161$$

Ex. $(2F.4)_{16} = (\cancel{3}4 7.25)_{10}$

$$= 2\times 16 + 15 + \frac{4}{16} = 47.25$$

Hexadecimal as shorthand for binary system

(usually referred as Hex)

Ex.

$$\begin{array}{c} 7\,6\,5\,4\,3\,2\,1\,0 \\ N = (\underbrace{1001}_{9}\underbrace{0110}_{6})_2 = (\underline{96})_{16} = 96\ H \end{array}$$

$$= \underline{1\times2^7 + 0\times2^6 + 0\times2^5 + 1\times2^4} + \underline{0\times2^3 + 1\times2^2 + 1\times2 + 0}$$

$$= 2^4(1\times2^3 + 0\times2^2 + 0\times2^1 + 1) + (\qquad\qquad)$$

$$= 16\times(\qquad\qquad) + (\quad)$$

Ex.

$$N = (10\underbrace{0110}_{6}\underbrace{0101}_{5}\underbrace{0010}_{2})_2 = 2652\ H$$

2 6 5 2

Conversion from decimal to other number systems:

1. First consider a positive decimal integer number N

MSD LSD

$$N = a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + \ldots + a_1 r + a_0$$

$$= r(\underline{a_{n-1} r^{n-2} + a_{n-2} r^{n-3} + \ldots + a_1}) + \underline{a_0}$$

$$= r \times \underline{Q} + \underset{\downarrow a_0}{R}$$

<u>Decimal to binary conversion</u>:

Ex. $(15)_{10} = (1111)_2$

$$\begin{array}{r|l} 2 & 15 \\ \hline 2 & 7 \\ \hline 2 & 3 \\ \hline 2 & 1 \\ \hline & 0 \end{array} \qquad \begin{array}{l} a_0 = 1 \\ a_1 = 1 \\ a_2 = 1 \\ a_3 = 1 \end{array} \qquad \begin{array}{ccccc|l} & 1 & 1 & 1 & 1 & \\ \hline 0 & 1 & 3 & 7 & 15 & 2 \end{array}$$

Ex. $(121)_{10} = (1111001)_2$

1 1 1 1 0 0 1

0 1 3 7 15 30 60 121 | 2

Decimal to octal conversion:

Ex. $(650)_{10} = (1212)_8$

1 2 1 2

0 1 10 81 650 | 8

Decimal to hexadecimal conversion:

Ex. $(500)_{10} = (1F4)_{16}$

1 F 4

0 1 31 500 | 16

2. Now consider positive fractional number N

$$N = (0.\, a_{-1}\, a_{-2}\, a_{-3} \ldots a_{-m})_r$$
$$= a_{-1}\, r^{-1} + a_{-2}\, r^{-2} + \ldots + a_{-m}\, r^{-m}$$

Multiply by r,

$$N \times r = a_{-1} + \underline{a_{-2}\, r^{-1} + \ldots + a_{-m}\, r^{-m+1}}$$

Ex. $(0.825)_{10} = (0.1101\ldots)_2$

a_{-1}

$0.825 \times 2 = 1.65$

a_{-2}

$0.65 \times 2 = 1.30$

a_{-3}

$0.30 \times 2 = 0.60$

$0.60 \times 2 = 1.20$

a_{-4}

Ex. $(15.825)_{10} = (\quad)_2$

$(1111.1101\ldots)_2$

Binary Addition

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 10

Sum of binary numbers:

Ex.

```
  111
  0101   →    5
  0011   →  + 3
  ----      ----
  1000        8
```

Ex.

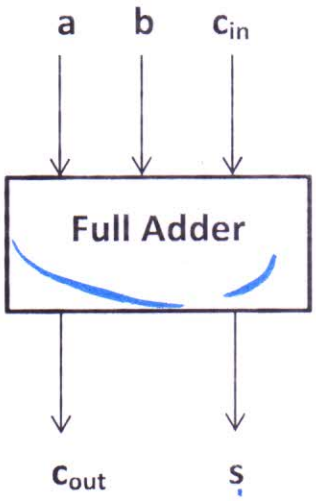
```
  111
  1011   →   11
  1101   →  +13
  ----      ----
 11000       24
```

Implementation of four-bit adder:

For one-bit adder-

c_{in}
a
b

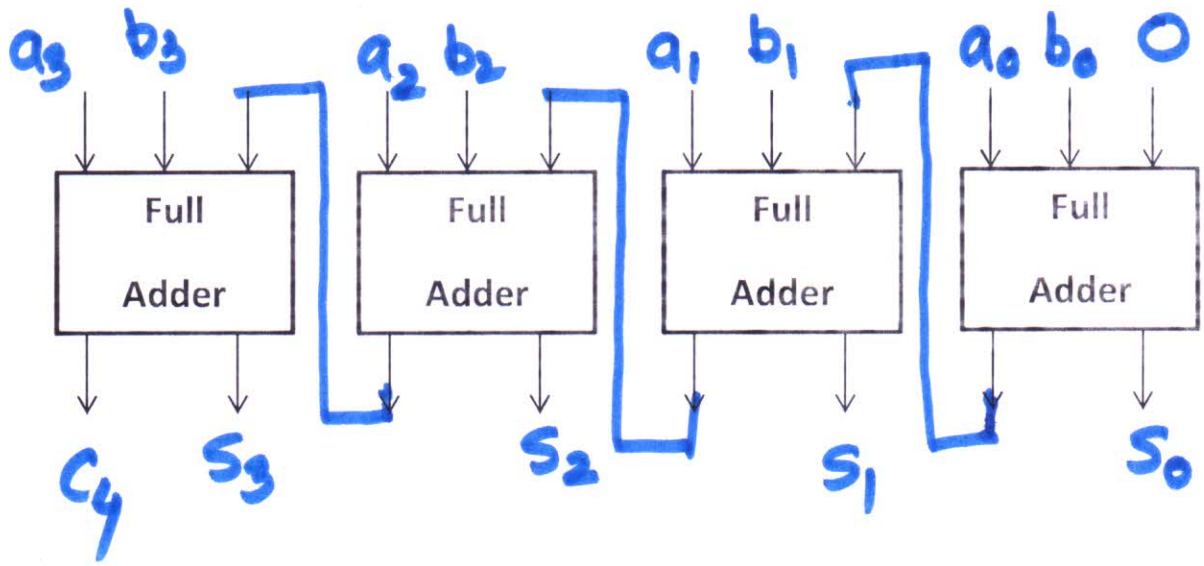
C_{out} S



a	b	c_{in}	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Four-bit adder:

$$\begin{array}{r} a_3\ a_2\ a_1\ a_0 \\ b_3\ b_2\ b_1\ b_0 \\ \hline c_4\ s_3\ s_2\ s_1\ s_0 \end{array}$$



Binary Subtraction

Ex.

$(15)_{10} - (6)_{10} = (1111)_2 - (0110)_2$

$= (15)_{10} + (-6)_{10}$ (Negative Number)

$= (1111)_2 + ?$

<u>**Binary representation of negative decimal numbers**</u>

Three ways:

1. **Sign-bit magnitude method**
2. **1's complement method and**
3. **2's complement method**

Sign-bit magnitude method:

The MSB (most significant bit) represents the 'sign', with a '0' denoting a plus sign and a '1' denoting a minus sign.

Ex. +5 = 0101 $n \rightarrow 2^n - 1$

-5 = 1101

$(n-1)$

$\underline{\cdot \; \cdot \; \cdot \; \cdot \; \cdot \; \cdot \; \cdot}$

$2^{n-1} - 1$

For n-bit representation-

Range: $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

For 8-bit representation-

Range: - 127 to 127

1's complement method:

Ex. +10 = 01010

-10 = 10101

$+5 \rightarrow 0101$

$-5 \rightarrow +1010$

1111

For n-bit representation-

Range: $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

For 8-bit representation-

Range: - 127 to 127

2's complement method:

Add one with 1's complement number.

Ex.

+9 = 01001

-9 = 10110 **(1's complement)**

+ 1

= 10111 **(2's complement)**

Ex.

2's complement of 01010 is ?

$$\begin{array}{r} 10101 \\ +\quad 1 \\ \hline 10110 \end{array}$$

Ex.

2's complement of 01100 is ? +12

$$\begin{array}{r} 10011 \\ +\quad 1 \\ \hline 10100 \end{array}$$

-12

$-(-12) = 12$

01100

Ex.

2's complement of 01000 is ?

$$\begin{array}{r} 10111 \\ +\quad 1 \\ \hline 11000 \end{array}$$

Ex.

2's complement 0000 is ? → +0

0000 → -0

0 0000 → +0
1 0000 → -0
Sign-Mag

Binary	Positive	Read in 2's complement
0000	0	0
0001	1	+1
0010	2	+2
0011	3	+3
0100	4	+4
0101	5	+5
0110	6	+6
0111	7	+7
1000	8	-8
1001	9	-7
1010	10	-6
1011	11	-5
1100	12	-4
1101	13	-3
1110	14	-2
1111	15	-1

0111 = 7

-8 to +7

For n-bit representation-

Range: $-(2^{n-1})$ to $(2^{n-1}-1)$

For 8-bit representation-

Range: - 128 to 127

Now, subtraction using 2's complement:

Ex. Compute 9-4

$$
\begin{array}{rr}
9: & 01001 \\
+\ (-4): & 11100 \\
\hline
\textcircled{1} & 00101 = +5
\end{array}
$$

Discard

01001
00100

Ex. Compute 4-9

$$
\begin{array}{rr}
4: & 00100 \\
(-9): & 10111 \\
\hline
-5 & 11011
\end{array}
$$

$= -(00101)_2$

$= -(5)_{10}$

Ex. Compute 5-5

$$\begin{array}{rl} +5: & 0101 \\ -5: & 1011 \\ \hline & \cancel{1}\ \underline{0000} \end{array}$$

Ex. Compute 14-10

$$\begin{array}{rl} 14: & 01110 \\ +10: & 01010 \end{array} \qquad \begin{array}{rl} 14: & 01110 \\ (-10): & 10110 \\ \hline +4\ \cancel{1} & \underline{00100} \end{array} \quad +4$$

Ex. Compute 10-14

$= \underline{-4}$

9345

1001 0011 0100 0101

16 bits

Binary Coded Decimal (BCD)

Each digit of a decimal number is replaced by a four digit binary numbers.

Weighted Code.

Decimal digit	8421 code	5421 code
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0010
4	0100	0100
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

0110

0+4+2+0   5+0+0+1

Ex. Representation of $(591)_{10}$ in BCD 8421 code:

BCD: 0101 1001 0001

Ex. Representation of $(804)_{10}$ in BCD 5421 code:

1011 0000 0100

469   $2^9 = 512$ $2^{10} = 1024$

Addition in BCD 8421 code:

Ex.	314:	0011	0001	0100
	+ 256:	0010	0101	0110
	570	0101	0110	1010
		5	7	+0110
				1 0000
				0

1010

1 0

1 0000

16

Add 0110 if number > 9.