Digital Electronics

NUMBER SYSTEMS

- Four types of number systems are there. The no. systems are differentiated on the basis of the no. of symbols.

1. Decimal (10 symbols : 0 to 9)
2. Binary (2 symbols : 0 and 1)
3. Octal (8 symbols : 0 to 7)
4. Hexadecimal (16 symbols : 0 to F(16))

POSITIONAL NUMBER SYSTEM (RADIX / BASE)

- Radix = Base = No. of numerals in the system

eg: 132.694 → Radix point

P_2 P_1 P_0 . P_{-1} P_{-2} P_{-3}

d_2 d_1 d_0 . d_{-1} d_{-2} d_{-3}

P_2 P_1 P_0 P_{-1} P_{-2} P_{-3}

- When we write a no. from left to right, highest positional digit ie, greater than zero and end with lowest positional digit ie, greater than zero. So a no. can be represented with zeroes to the left of the highest positional digit and zeroes to the right of the lowest position digit greater than zero without effecting the value of the no.

BASE CONVERTION

Each position in a number contains a different weight based on its relative location to the radix point.

Weight of each position is based on the radix of the no. system being used.

$$\text{Weight} = \text{Radix}^{P}$$

$$\text{Decimal value} = \sum_{i=P_{min}}^{P_{max}} d_i (\text{radix})^i$$

eg:- a) 1 3 2 . 6 5 4
↓ ↓ ↓ ↓ ↓ ↓
2 1 0 -1 -2 -3 } Position
10^2 10^1 10^0 . 10^{-1} 10^{-2} 10^{-3} } weights

Decimal value = $\sum_{i=-3}^{2} d_i (radix)^i$

$= 4\times10^{-3} + 5\times10^{-2} + 6\times10^{-1} + 2\times10^{0} + 3\times10^{1} + 1\times10^{2}$

$= 0.004 + 0.05 + 0.6 + 2 + 30 + 100$

$= 132.654$

b) 101.11 : Binary

1 0 1 . 1 1

Position: 2 1 0 -1 -2

weights : 2^2 2^1 2^0 2^{-1} 2^{-2}

Decimal value = $\sum_{i=-2}^{2} d_i (2)^i$

$= 1\times2^{-2} + 1\times2^{-1} + 1\times2^{0} + 0\times2^{1} + 1\times2^{2}$

$= 0.25 + 0.5 + 1 + 0 + 4$

$= \underline{\underline{5.75}}$

15/7

c) 17.17_8 : Hexadecimal

1 7 . 1 7

Position: 1 0 . -1 -2

Weight : 8^1 8^0 8^{-1} 8^{-2}

Decimal value = $\sum_{i=-2}^{1} d_i (8)^i$

$= 7\times8^{-2} + 1\times8^{-1} + 7\times8^{0} + 1\times8^{1}$

$= 0.109 + 0.125 + 7 + 8$

$= \underline{\underline{15.234}}$

d) $1AB.EF \Rightarrow 15\times16^{-2} + 14\times16^{-1} + 11\times16^{0} + 10\times16^{1} + 1\times16^{2}$

$= \underline{\underline{427.933}}$

Converting from decimal

Converting the whole number part

1. Divide the whole number part of the decimal number with radix or base of the target number system.
2. Obtain the quotient and reminder from step 1. The remainder will be taken as the least significant bit.
3. Divide the quotient obtained from step 1 with the target number system. Continue this process until either the quotient is zero or an approximate limit is reached. Consider the last remainder as the most significant bit (msb)

eg: 11.375_{10}, target radix = 2

whole no. part = 11

	Q	R	
$2\overline{)11}$	5	1	→ LSB
$2\overline{)5}$	2	1	→ higher order bit
$2\overline{)2}$	1	0	→ "
$2\overline{)1}$	0	1	→ MSB

Converting the fractional part

1. Multiply the fractional part with the radix ~~of~~ or base of the target number system.
2. From the product obtained as the result, take the whole number part of the product as MSB.
3. Multiply the fractional part of the product with the radix or base of the target no. system.
4. Continue this process until the fractional part

of the number is either zero or approximately zero.

5. The whole number part of the last product will be taken as the LSB.

Eg: fractional part: .375

	Prad	whole
2(0.375) =	0.75	0
2(0.75) =	1.50	1
2(0.50) =	1.00	1

$\therefore 11.375_{10} \Rightarrow 1011.011$

- Binary, Hexa, Octal → Decimal
  $D = \sum_{i=P_{min}}^{P_{max}} di(radix)^i$
- Decimal → Binary, Hexa, Octal

Whole	Frac
r	r(f)

- Octal → Binary '3'
- Hexa → Binary '4'

Q. Convert $(10.4)_{10}$ to octal.

Whole number part: 10 ; Fraction

	Q	R	
8 | 10	1	2	→ LSB
8 | 1	0	1	→ MSB

	Prad	whole
8(.4) =	2.4	2
8(.8) =	1.6	1
8(.6) =	1.2	1
8(.2) =	0.4	

Fraction part: 0.4

8(0.4) =	3.2	3
8(.2) =	1.6	1
8(.6) =	4.8	4
8(.8) =	6.4	6
8(.4) =	4.0	4

$\therefore (10.4)_{10} = 12.31464$

Converting b/w 2^n bases

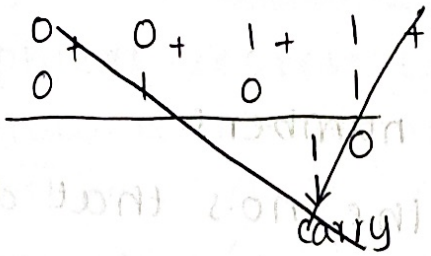
- Octal → Binary or vice versa.
- Hexadecimal → Binary or vice versa.

eg: 001 110 . 110 100 } divide into "3" Binary → octal

16/1

Binary Arithmetic

1. Addition (carry)



$$\begin{array}{r} 0+\\ 0\\ \hline 0 \end{array} \qquad \begin{array}{r} 0+\\ 1\\ \hline 1 \end{array} \qquad \begin{array}{r} 1+\\ 0\\ \hline 1 \end{array} \qquad \begin{array}{r} 1+\\ 1\\ \hline 10 \end{array}$$

↑ carry

eg: ~~1010.1~~ $\quad \overset{1}{1}\,0\,\overset{1}{1}\,0.1 +$

$$\begin{array}{r} 1110.1\\ \hline 11001.0 \end{array}$$

- Largest decimal sum that can result from the addition of two binary numbers is given by $2(2^n-1)$

eg: 2 8-bit numbers to be added would be represented with the highest decimal value 2^n-1 or $(255)_{10}$ (11111111_2)

- The sum of 255_{10} itself would result in 510_{10}. ie,

$$\begin{array}{r} 1\,1\,1\,1\,1\,1\,1\,1+\\ 1\,1\,1\,1\,1\,1\,1\,1\\ \hline 1\,1\,1\,1\,1\,1\,1\,1\,0 \end{array}$$

↑ carry bit

This proves that the largest sum achievable would only require one additional bit. ie, a single carry bit is sufficient to handle all possible magnitudes for binary addition.

2. Subtraction (borrow)

$$\begin{array}{r} 0-\\ 0\\ \hline 0 \end{array} \qquad \begin{array}{r} 10-\\ 1\\ \hline 1 \end{array} \qquad \begin{array}{r} 1-\\ 0\\ \hline 1 \end{array} \qquad \begin{array}{r} 1-\\ 1\\ \hline 0 \end{array}$$

borrow

$$\left\{\begin{array}{r} 1- \leftarrow \text{minuend}\\ 1 \leftarrow \text{subtrahend}\\ \hline \end{array}\right.$$

- Borrows are used as necessary when subtraction moves from lsb to msb.

- The msb requires a borrow, the event is called borrowin or subtraction is said to require a borrow.

3. Unsigned and signed numbers.

- Unsigned no.s are the no.s that doesnot allow -ve values and signed no.s allows both +ve and -ve values.
- Signed no.s requires a signed bit

- Range of an unsigned bit : $0 \leq N_{unsigned} \leq (2^n - 1)$
- In signed numbers, if the signed bit is 0; it is a +ve number and if it is 1; it is a -ve number.
- There are three ways to represent signed numbers:

i. Signed magnitude method
- msb is the signed bit and the rest of the bits are magnitude or absolute value of the number.
- range of a siagned magnitude number:

$$-(2^{n-1}-1) \leq N_{SM} \leq +(2^{n-1}-1) \quad (-127 \rightarrow 127)$$

ii. One's complement
- range is $-(2^{n-1}-1) \leq N_{one's\ compliment} \leq +(2^{n-1}-1)$

iii. Two's complement
- convert to one's compliment and then add 1.

17/7

Digital Circuitary & Interfacing

- Logic symbol

  Graphical representation of a symbol that shows how a circuit in a system interface with one another.

- Logic function / Logic Expression

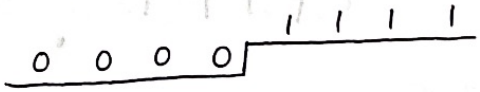
  It is an equation that provides functionality of the circuit.

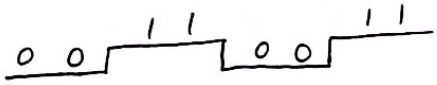
- Truth table

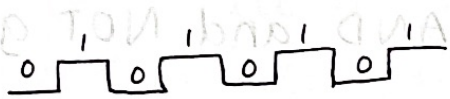
  - Lists possible input combinations and their corresponding outputs.
  - Input codes are listed in ascending order so that their row number can be given as their decimal equivalence.

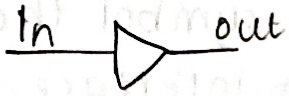
Row	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Logic symbols:

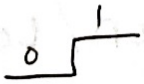
A: 

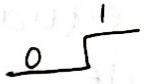
B: 

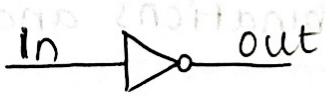
C: 

- Buffer
  - The first basic gate is buffer
  - The output of a buffer is its input
  - 

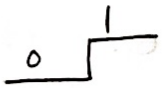
In	out
0	0
1	1

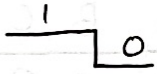
  - out = In
  - ~~sy~~ logic symbol : In : 

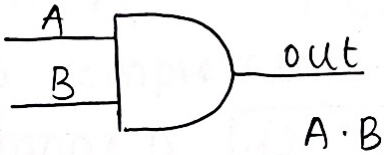
    out : 

- Inverter (NOT gate)
  - 

In	Out
0	1
1	0

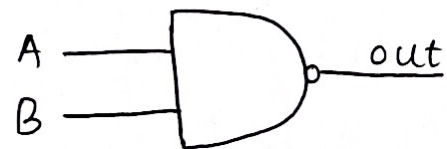
  - out = $\bar{In}$
  - wave form : In : 

    Out : 

- AND gate
  - The output of an AND gate will only be true, ie logic 1 only if both its inputs are true ~~or same~~.
  - 

    A·B

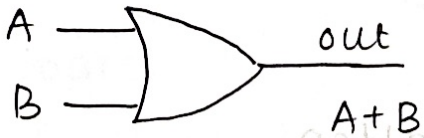
A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

- NAND gate
  - Combination of AND and NOT gate
  - 

A	B	Out (AND)	Out (NAND)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

- OR gate

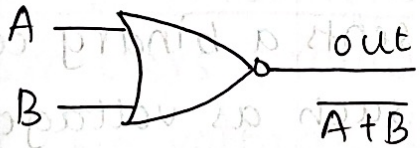
- The output of an OR gate will be true if any of the input is true.



A	B	out
0	0	0
0	1	1
1	0	1
1	1	1

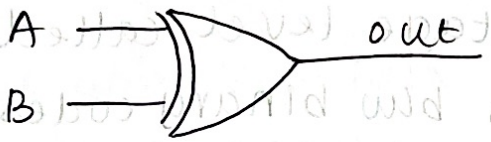
- NOR gate

- NOR gate is a combination of NOT and OR gate. The output is the opposite of the OR gate.



A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

- XOR gate

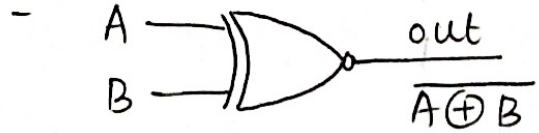


$out = A \oplus B$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

- This gate is also called exclusive or gate / difference gate. In this gate the o/p will be true when i/p codes differ from one another.

- XNOR gate

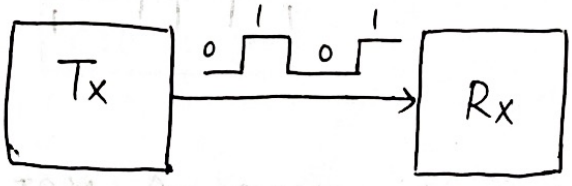


$out = \overline{A \oplus B}$

A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

- This is also known as equivalence gate because the o/p will be true when both the i/p codes are the same.

Digital circuit operation



Tx : Transmitter / driver (send signal)

Rx : Receiver.

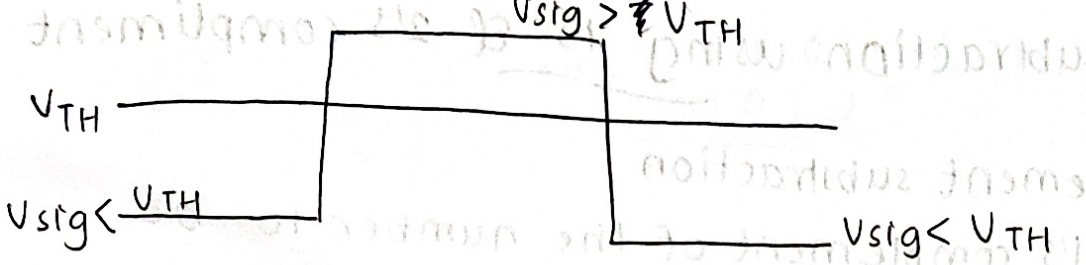
- Transmitter represents a binary code using an electrical quantity such as voltage.
- Receiving circuit observes the voltage and is able to decode the binary code values.

Logic Levels

Consider a voltage level called voltage threshold to switch b/w binary codes.

V_{TH} : Voltage threshold.

- If the signal is less than the threshold, the signal is known as logic low
- If the signal is higher than V_{TH}, the signal is known as logic high



Two types of digital assignments:

1. +ve logic

In +ve logic, the logic high represents 1 and logic low represents 0.

2. -ve logic

The logic high represents 0 and logic low represents 1.

Logic level	Logic value +ve	-ve
LOW	0	1
HIGH	1	0

Output DC specifications

- Transmitters provides specifications on the range of o/p voltages (V_o) that are guaranteed to provide a logic 1 or 0. Such specifications are known as DC o/p specifications and they are represented by V_{OH}-MAX, V_{OH}-MIN, V_{OL}-MAX, V_{OL}-MIN.

- V_{OH}-MAX and V_{OH}-MIN provide the range of voltages guaranteed to provide while outputting logic high / logic 1.
- V_{OL}-MAX and V_{OL}-MIN provide the range of voltages guaranteed to provide while outputting logic low / logic 0.

Binary Subtraction using 1's & 2's compliment

1's complement subtraction

1. Find 1's complement of the number to be subtracted (2nd number)
2. Perform addition of the first number with the ones complement of the 2nd number.
3. If carry is generated, add the carry to the result. If there is no carry, take 1's compliment of the result and assign -ve value.

2's compliment subtraction

1. Find 2's compliment of number to be ~~-~~ subtracted (2nd number)
2. Perform addition with 1st number.
3. If carry is generated, neglect the carry. Otherwise take 2's compliment of the result and add -ve sign.

eg:- • 1100 - 0101 → 0101 → 1010 + 1 = 1011 → 2's compliment

```
  1100 +
  1011
 ------
 (1)0111
  1000 +
     1
 ------
 -1001
```

• 1's compliment : 1001 - 0100 → 1011 (1's compliment)

```
  1001 +
  1011
 ------
 (1)0100
carry ↗  +1
 ------
  0101
```

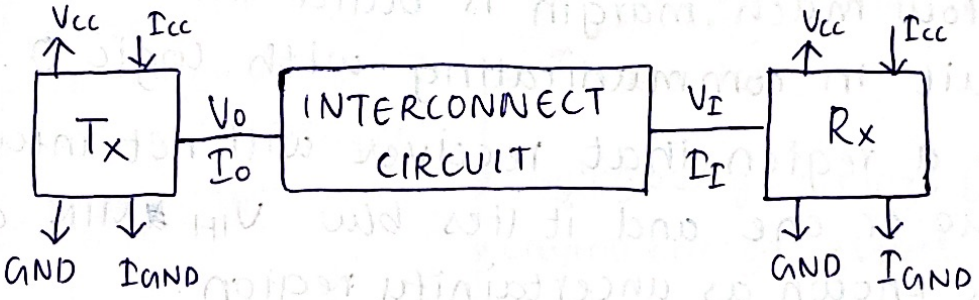
• 1001 - 1011 → 0100 : 1's compliment

$$\begin{array}{r} 1001 + \\ 0100 \\ \hline 1101 \end{array}$$ no carry

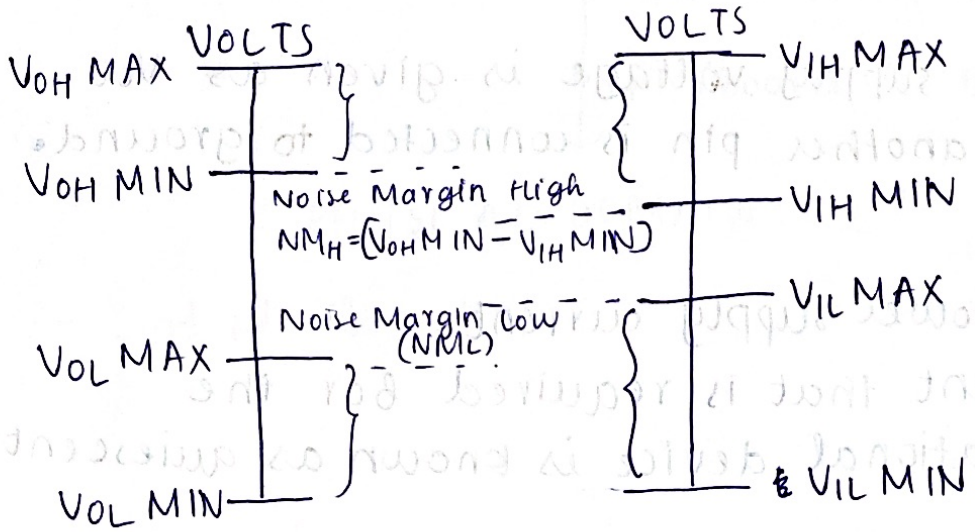
So, 1101 → 1's compliment → -0010

23/7

- The maximum amount of current that can flow through the ~~transist~~ transmitters output is specified as I_o.
- I_{OH}^{MAX}: Maximum output current from the transmitter for logic high.
- I_{OL} MAX : Maximum output current from the transmitter for logic low.
- Transmitter output providing current to the receiving circuit is known as sourcing current.
- Transmitter output drawing current from the receiving circuit is known as sinking current.



~~VOLTS~~ Noise Margin



- $NM_H = V_{OH}MIN - V_{IH}MIN$
- $NM_L = V_{IL}MAX - V_{OL}MAX$

- Receiving circuit provides input voltages (V_I) for logic high or low
- V_{IH} MAX, V_{IH} MIN, V_{IL} MAX & V_{IL} MIN
- V_{IH} MAX & V_{IH} MIN: The receiver interprets these voltages as logic high.
- V_{IL} MAX & V_{IL} MIN: The receiver interprets these voltages as logic low.

~~Noise Margin~~

- The best case scenerio for digital signaling is when the transmitter is sending the largest or smallest signal possible, ie V_{OH} MAX or V_{OH} MIN
- NM_H (noise margin high) is calculated ~~using~~ to signify how much margin is build into the Tx/Rx circuit will communicate logic 1
- NM_L: How much margin is build into the Tx/Rx circuit in communicating with logic 0.
- There is a region that receiver will not interpret as either zero or one and it lies b/w V_{IH} MIN and V_{IL} MAX. It is known as uncertainity region.

Power Supply

- The power supply voltage is given as Vcc (I_{CC} MAX) and another pin is connected to ground. (I_{GND})

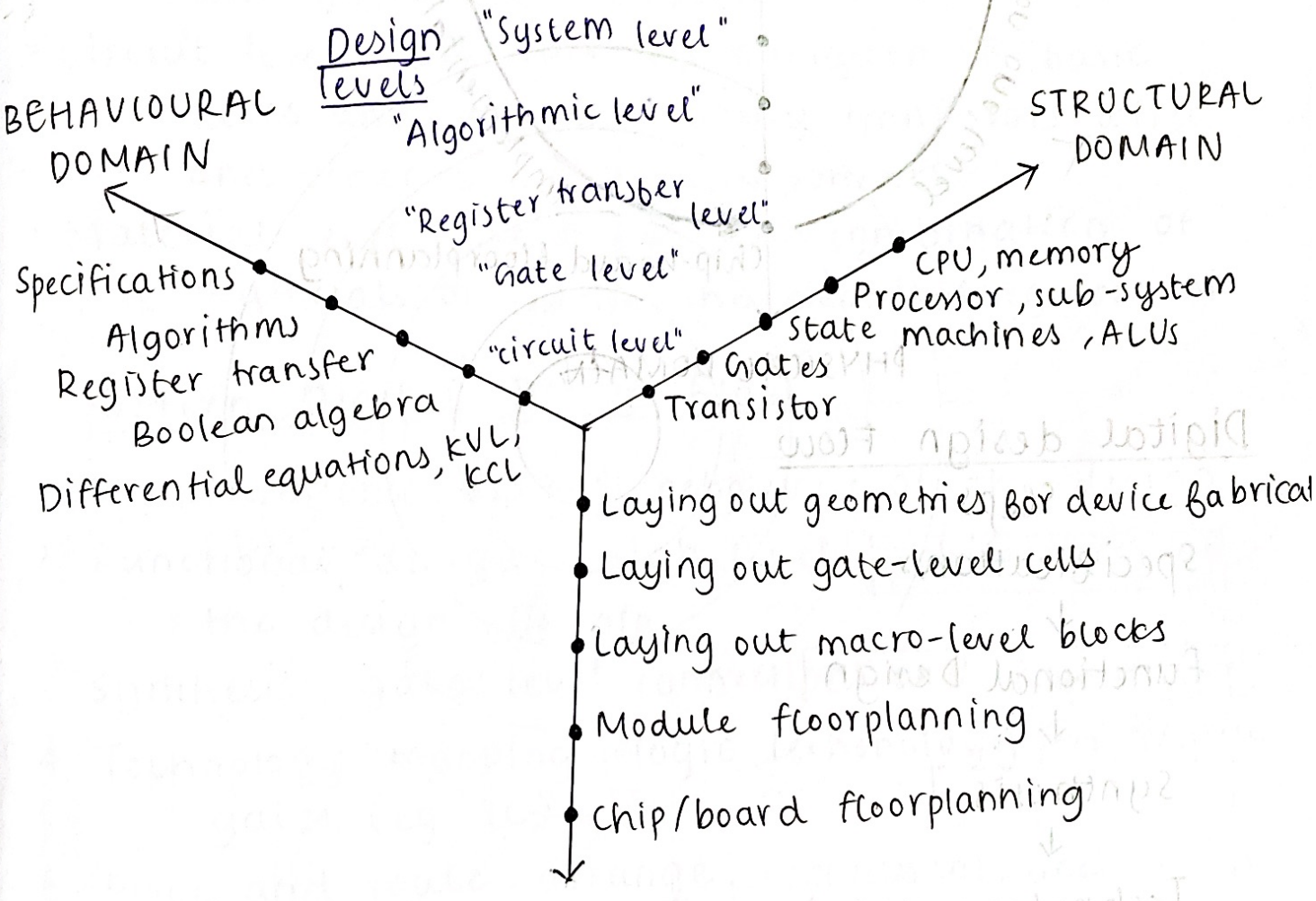
Components of power supply current

- The current that is required for the functional operational device is known as quiescent (I_Q).
- The second camponent of power supply is output current (I_O)

- GND : Transmitting circuit sources current to the receiver on its output pin.
The transmitter must receive the same amount of current on another pin. This is known as Vcc

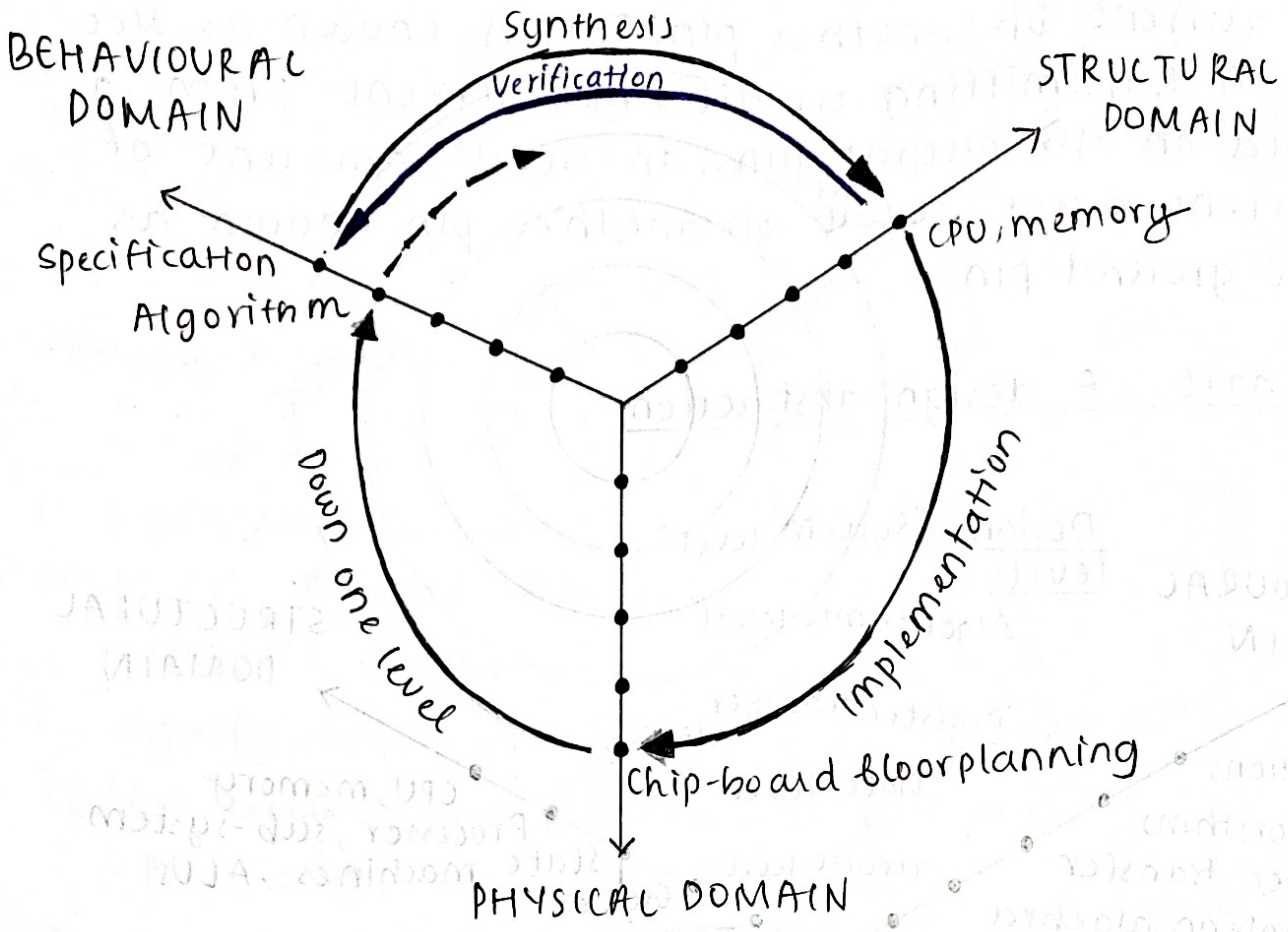
A transmitting circuit sinks current from a load on its output pin, an equal amount of current must exit on another pin known as the ground pin

Y chart of design abstraction.



- Gajski & Kuhn

Y Chart illustrating Top-down Design approach.



Digital design Flow

Steps

- Specifications
- ↓
- Functional Design
- ↓
- Synthesis
- ↓
- Technology mapping
- ↓
- Place and Route
- ↓
- Verification
- ↓
- Fabrication.

29/1 • HDL (hardware description language) Abstraction.

- Abstraction : to show only essential features.

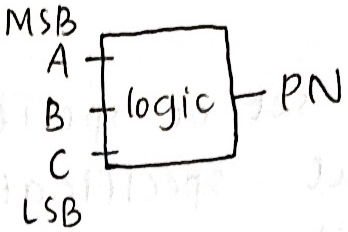
- System Level
  - Describe a set of specifications
- Algorithmic level : specifications are further divided into subsystems.
- Register transfer level : details how data is moved within and between subsystems.
- Gate level : in this, the design is described using basic gates and registers.
- Circuit level : describes the operation of basic gates and registers using transistors wires and other electrical components.
- Material level : describes the combination of materials to realise the electrical component.

Modern Digital Design Flow

1. Specifications : desired behaviour of the design
2. Functional design : high level architecture of the design (i/p, o/p
3. Synthesis : gate level connections and K map.
4. Technology mapping : logic technology to realise gates (eg: IC)
5. Place and route : arrange components and wires to minimise area and connections.
6. Verification : checks whether the design meets timing and power consumption requirements based on specifications.
7. Fabrication : Implement the final design.

Eg: Design a prime no. detector from the numbers 0_{10} upto 7_{10} with delay less than 200 ns

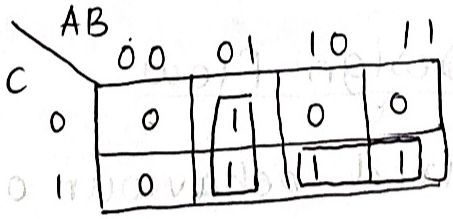
Functional design :



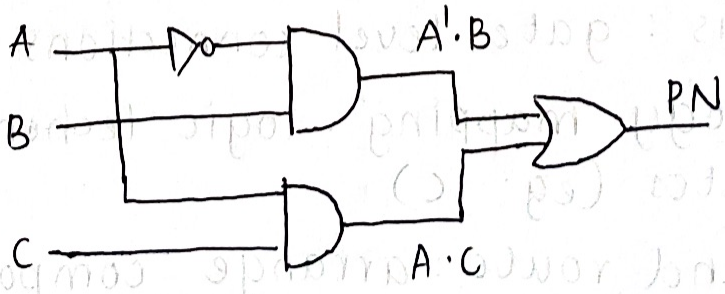
truth table :

A	B	C	PN	
0	0	0	0	
0	0	1	0	
0	1	0	1	→ 2
0	1	1	1	→ 3
1	0	0	0	
1	0	1	1	→ 5
1	1	0	0	
1	1	1	1	→ 7

Synthesis :



$A' \cdot B + A \cdot C$



30/4