Lecture-2   7/8/2019   Page-1

- Digital design ~~basic~~ : (Review of concept)

Design specification:-

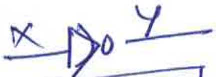
Digital logic operations

NOT | AND | OR

Binary variable - {0, or 1}

NOT

I/p x	y O/P
0	1
1	0



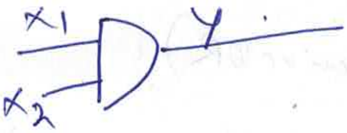
$\boxed{y = \bar{x}}$ → Boolean Expression

↳ Truth table (one of the ways of specifying a design)

AND

I/p		O/p
X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

$y = x_1 \cdot x_2$ → Boolean Exp.

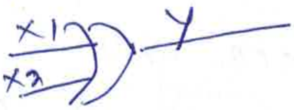


Properties of AND gate
- controlling input
- Masking (AND)

OR

I/p		O/p
x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

$y = x_1 + x_2$ → Boolean Exp.



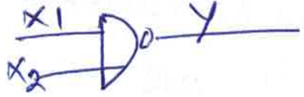
Properties of OR gate
- controlling input
- Masking (OR)

Universal gate :

NAND

I/P		O/P
X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$Y = \overline{X_1 \cdot X_2}$$



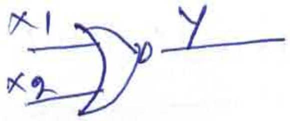
Implementation of other gate
- Inverter
- AND
- OR

De-morgan's law

NOR gate

I/P		O/P
X_1	X_2	Y
0	0	~~0~~ 1
0	1	0
1	0	0
1	1	0

$$Y = \overline{X_1 + X_2}$$



Implementation of other gate
- Inverter
- AND
- OR

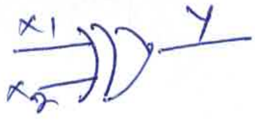
By the use of De-morgan's law.

Derived gate :-

- XOR - (Exclusive OR)

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$Y = X_1 \oplus X_2$$



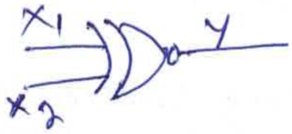
properties of XOR gate

- Comparator / Equivalence chec
- Used as a buffer/inverter

– XNOR –

x_1	x_2	Y
0	0	1
0	1	0
1	0	0
1	1	1

$y = \overline{x_1 \oplus x_2}$



Large function (Large no. of variables) -

Three variable

$y = f(x_1, x_2, x_3)$

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	
0	0	1	
1	1	1	

$\boxed{8 = 2^3}$

– Ten variable – $y = f(x_1, x_2, --- x_{10})$

Table size – 11 columns

– 2^{10} rows.

– n-variable

Table size = $(n+1)$ columns

– 2^n ~~row~~ rows.

→ What is the total number of possible function with

How to specify a complex design?

———— × ————

Example - Intel i-5 — i7 processor family has around <u>1000 – 2000</u> pins <u>input</u>/output

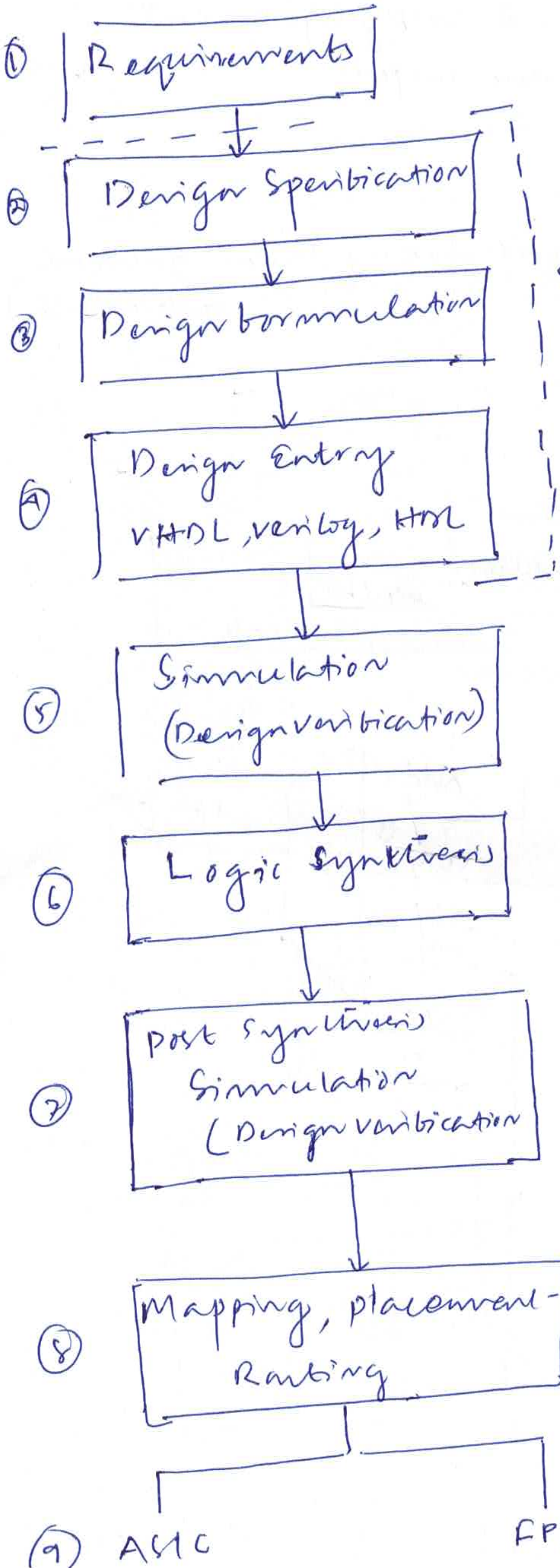
<u>Use of programming language</u> -

<u>Hardware description "language"</u> - (HDL)

<u>Design specification</u>

- Level of abstraction (Identifying the level of abstraction)
  - Architectural level
  - Behavior level
  - Structural level
- Design specification using language
  - Architecture level language
    - Blue spec
    - System C
    - System verilog
    - (Use of python kind language)
  - ~~RTL level~~
    Behavior level / Structural level
    ~~RTL~~ Register Transfer level
    - VHDL
    - Verilog
    - HDL (we will be covering in our class)

Modern Digital System Design Flow —



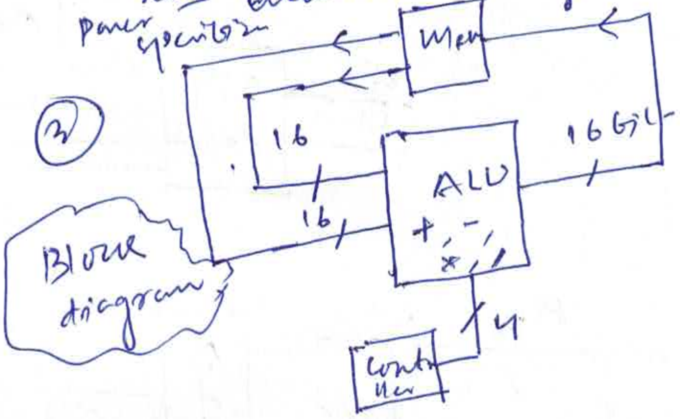
Example

① Design of an auto pilot micro controller
- Altitude
- Temp
- pressure
- Light
- Weather predict

← Design specification

② Altitude - 30,000 ft
ALU Temp - 50°c to -10°c
16 bit size pressure - ~~[illegible]~~
1 milli sensor Light -
power specification Weather - Rainy, cloud

③



④ VHDL, verilog, HDL

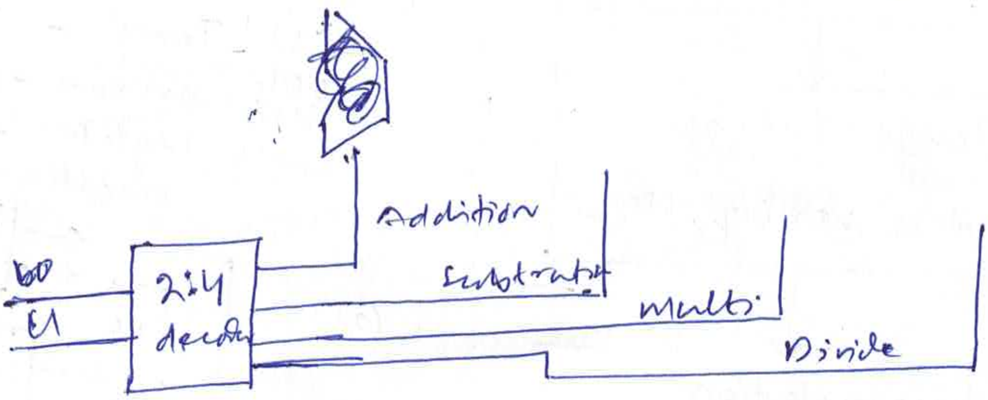
```
module adder16 (input a16,
                b16
input control = 2'b000; output c16
~~⑤~~ always @ (posedge clock)
    begin
      if control = 2'b000
        begin
          c16 = a16 + b16

      end
end module.
```

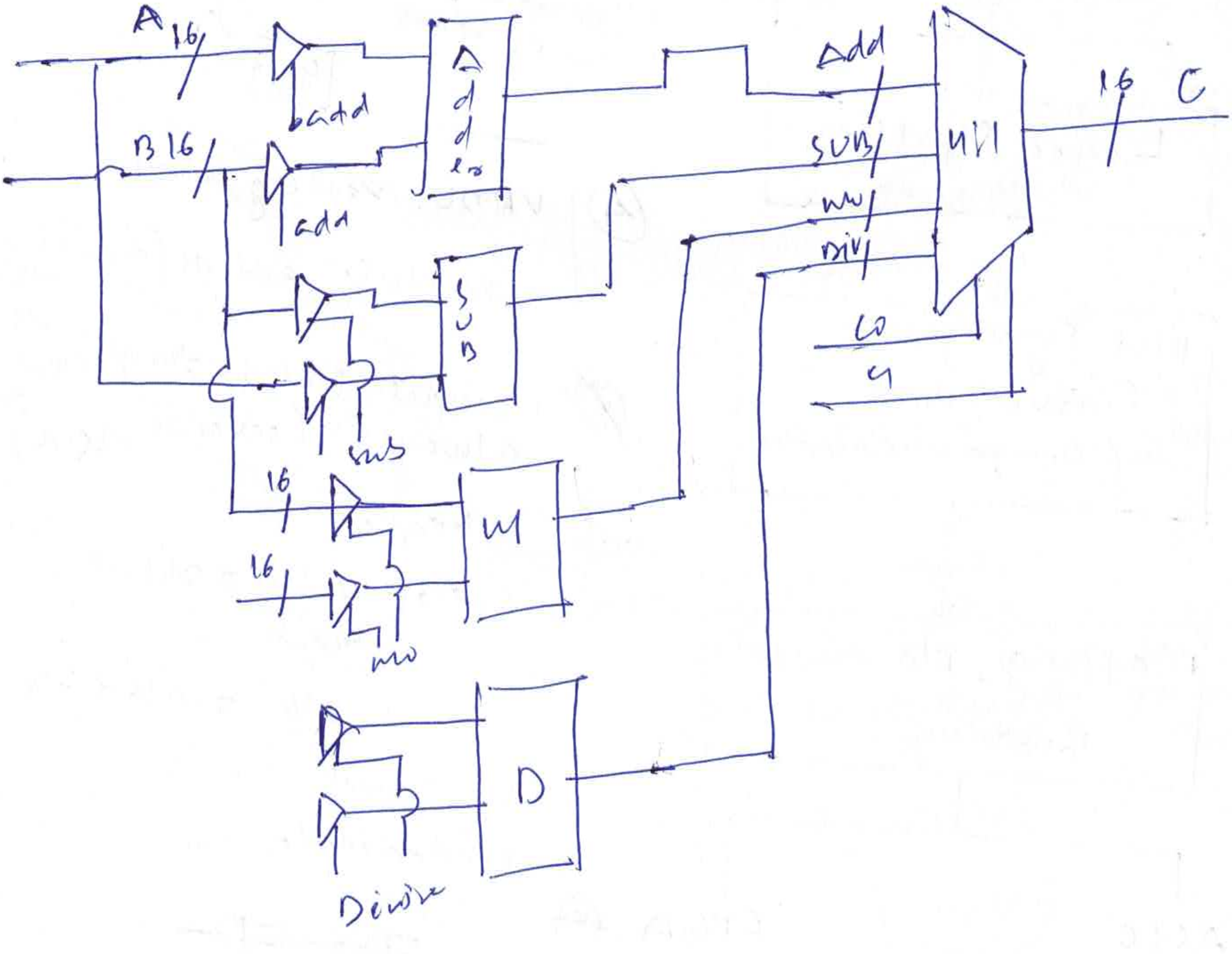
⑤ Simmulation to varify whoather the design is correct or buggay

Test benches are used to supply input and check the ~~design~~ desired output

⑥ Logic synthesis :-

this gives rise to gate level design (called gatelevel ~~gate~~ netlist)


E0
E1
2:4 decoder
Addition
Substraction
Multi
Divide


A 16
B 16
badd
add
Adder
SUB
sub
16
16
M
mo
D
Divide
Add
SUB
MU
DIV
MUX
16
C
C0
C1