Logic Synthesis

2 Level Exact Synthesis

→ SUNDAY A/N

Virendra Singh

Professor

Department of Electrical Engineering &

Dept. of Computer Science & Engineering

Indian Institute of Technology Bombay

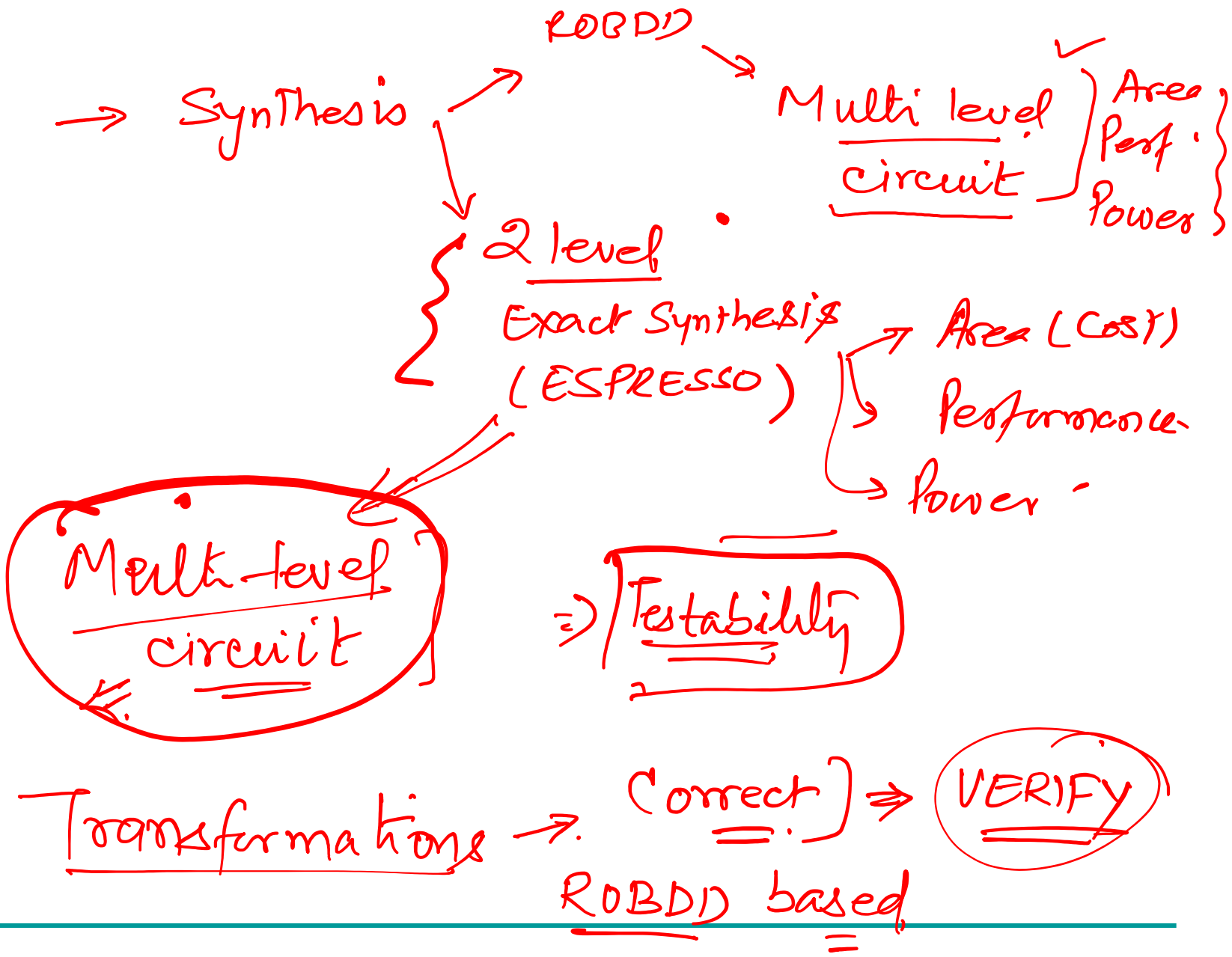
http://www.ee.iitb.ac.in/~viren/

E-mail: viren@{ee, cse}.iitb.ac.in

EE-677: Foundations of VLSI CAD

Lecture 21 on 28 Sep 2021

CADSL


→ Synthesis
ROBDD
Multi level circuit
Area
Perf.
Power
2 level
Exact Synthesis
(ESPRESSO)
Area (Cost)
Performance
Power
Multi-level circuit
⇒ Testability
Transformations →
(Correct) ⇒ VERIFY
ROBDD based

{ SAT based Satisfiability]

clause

$(a+b)\,(b+\bar{a}+c)\,(a+c+d) = 1$

$c_1 \quad c_2 \quad c_3$

SAT { Under what assignments to variables (literals) a, b, c, d. the expression evaluates to true]

{ a = 1 b = 1 c = 1 d = 1 }

$c_1 \cdot c_2 \cdot c_3 = 1 \Rightarrow$ $c_1 = 1,\ c_2 = 1,\ c_3 = 1$

$c_1 = 1,\ c_2 = 0,\ c_3 = 1$

0

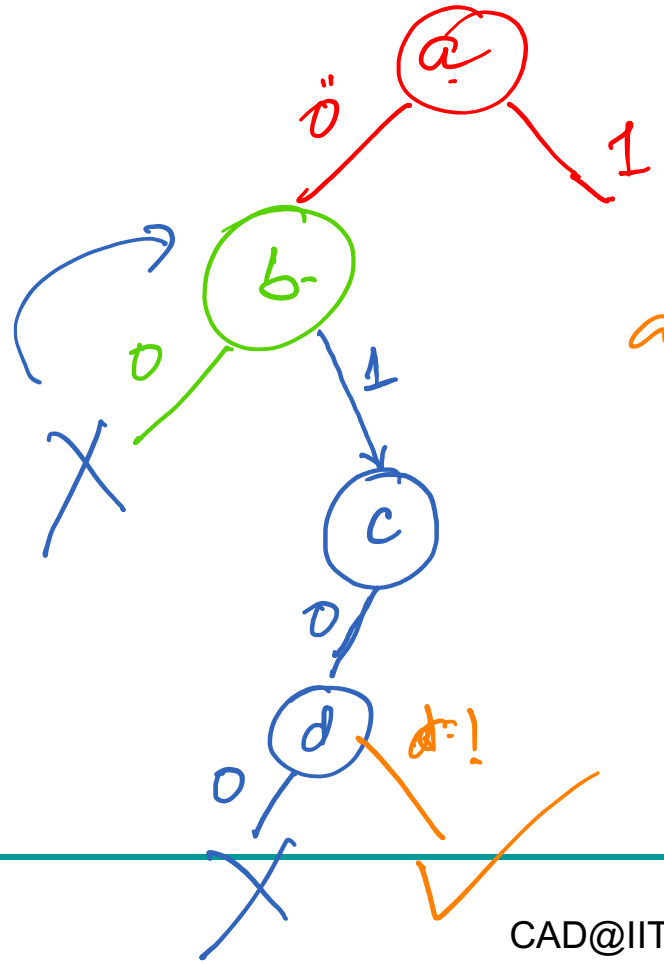
How to find the assignments?

$(a+b)$ $(\bar{a}+b+c)$ $(a+c+d)$ c_3

$(a+b)$ c_2

Decision Tree

$(a < b < c < d)$



$c_1 = 0$

$a=0$ c_2 ✓

c_1 c_3

$b=1$ c_2 ✓

c_1 ✓

c_3

$c=1$ c_3 ✓

c_1 ✓

c_3

$d=0$ c_2 ✓

c_1 ✓

c_3 ✗

$d=1$ c_2 ✓

c_1 ✓

c_3 ✓

$a=0$

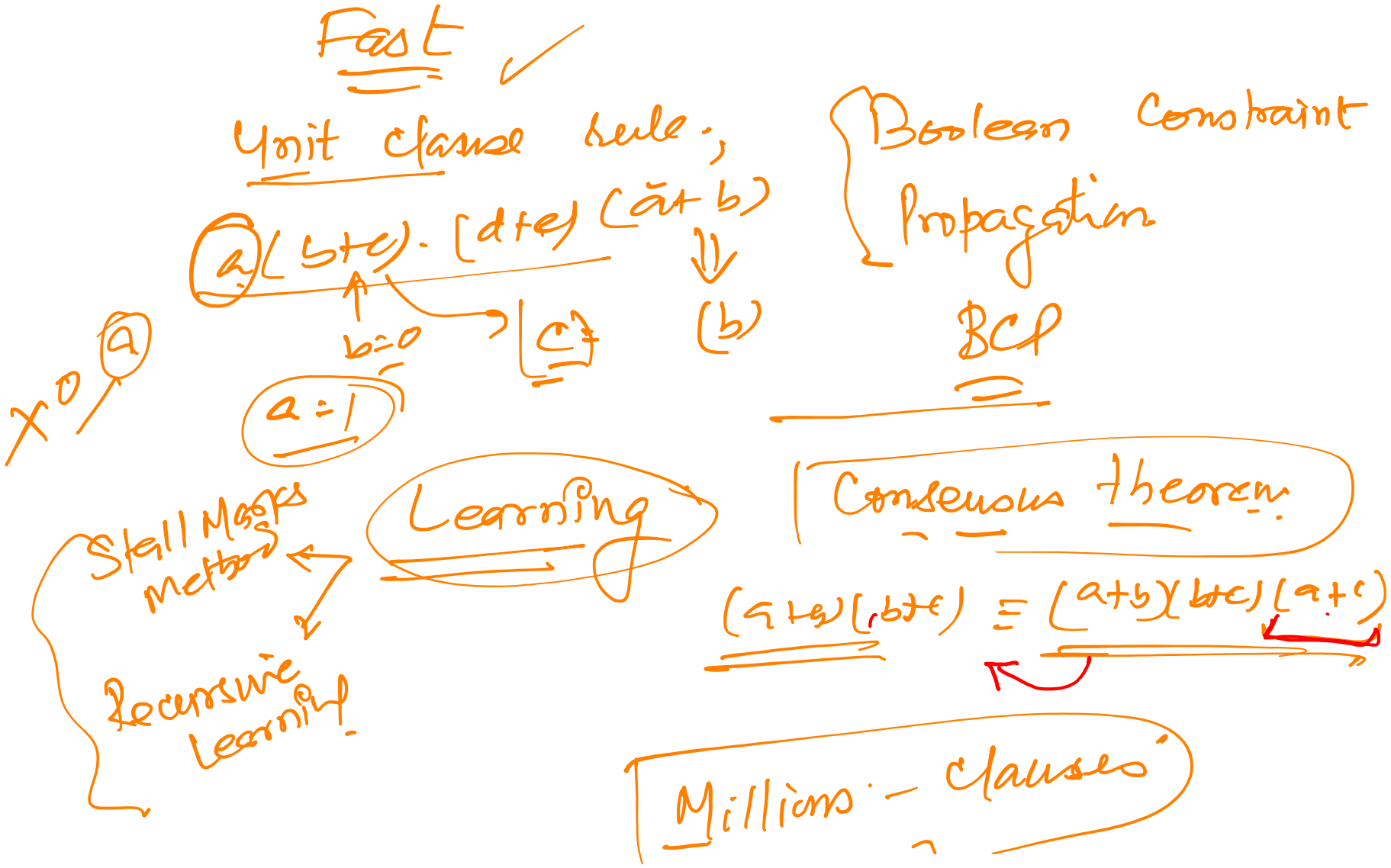
$b=1$

$c=0$

$d=1$

1960.

DPLL


Fast
Unit clause rule;
Boolean Constraint Propagation
BCP
Learning
Stall Maps method
Recursive Learning
Consensus theorem
Millions - clauses

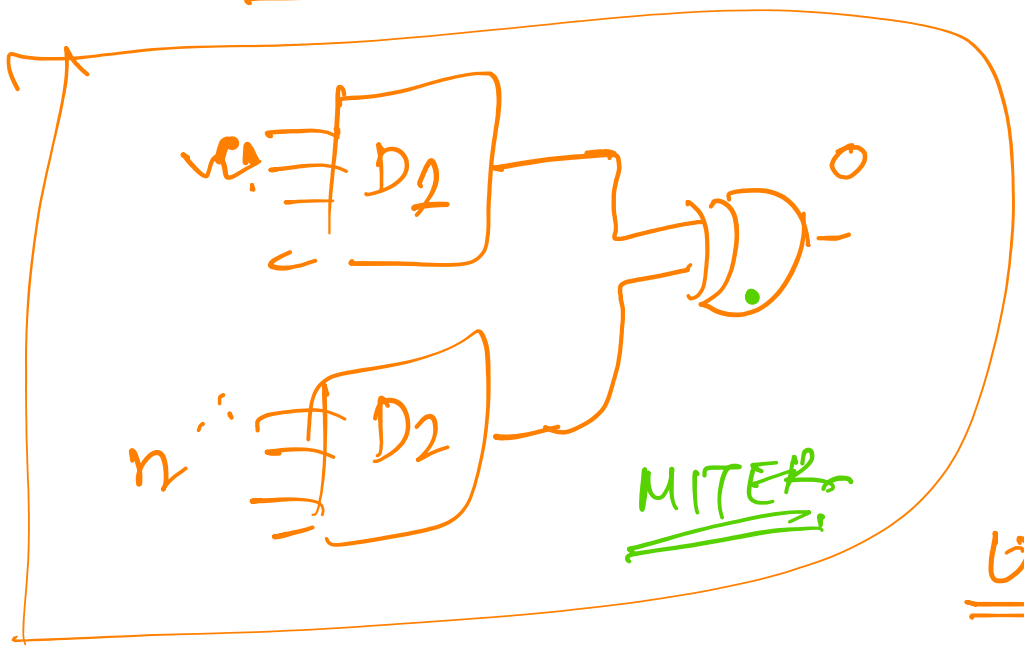
VERIFICATION

D1 D2

D1 ✓

D2

$D_1 \equiv D_2$



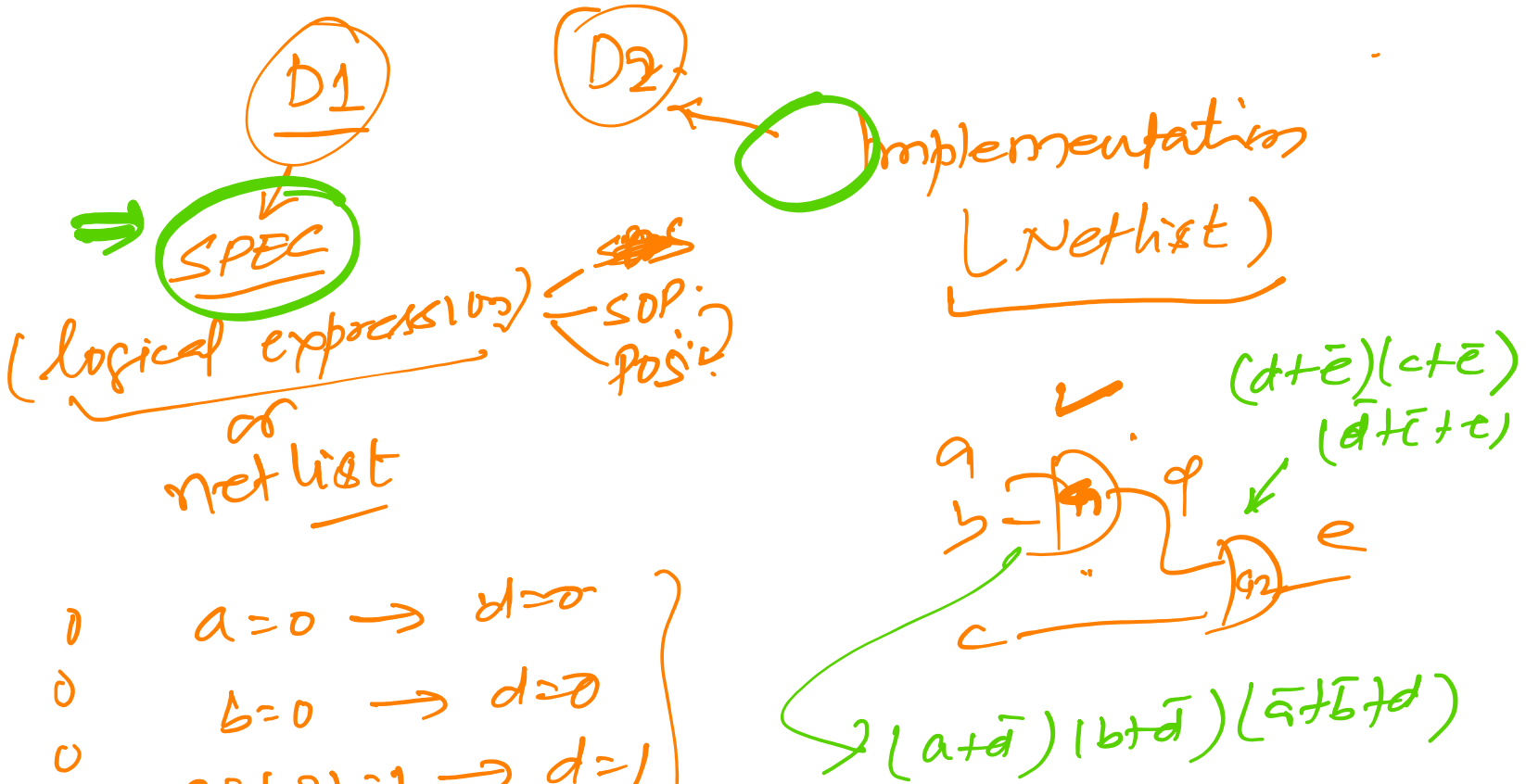
SAT expression

Should never satisfy if $D_1 \equiv D_2$

UNSAT problem

if any assignment exists $\Rightarrow D_1 \not\equiv D_2$

a=1 b=1 c=1 d=1 ⇐ Counter Example.


D1
D2
Implementation
(Netlist)
SPEC
(logical expressions)
of
netlist
SOP
POS
(d+ē)(c+ē)
(d̄+c̄+c)
a
b
c
q
e
(a+d̄)(b+d̄)(ā+b̄+d)
0 0 0
0 1 0
1 0 0
1 1 1
a=0 → d=0
b=0 → d=0
a=1 & b=1 → d=1
AND
a → d ⇒ (ā+d) ≡ (a+d̄)
b → d ⇒ (b+d̄)
(a·b → d) ⇒ (ā·b̄+d) ⇒ (ā+b̄+d)

{ Equivalence checking }

SAT

Thank You