

Logic: Representation

Virendra Singh

Professor

Computer Architecture and Dependable Systems Lab

Department of Computer Science & Engineering, and

Department of Electrical Engineering

Indian Institute of Technology Bombay

<http://www.cse.iitb.ac.in/~viren/>

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

CS-230: Digital Logic Design & Computer Architecture



Lecture 8 (20 January 2022)

CADSL

Representations

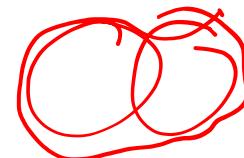
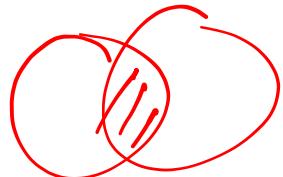
Truth Table

Sum Of Minterm. $\leftarrow \underline{\text{SOP}}$ ✓ true.

Product Of Maxterms $\leftarrow \underline{\text{POS}}$ ✓ false.

Canonical.

$x \cdot y$.



Union



Graphical Method: Binary Decision Diagram



Binary Decision Diagram

- ❖ BDD is canonical form of representation

- ❖ Shanon's expansion theorem

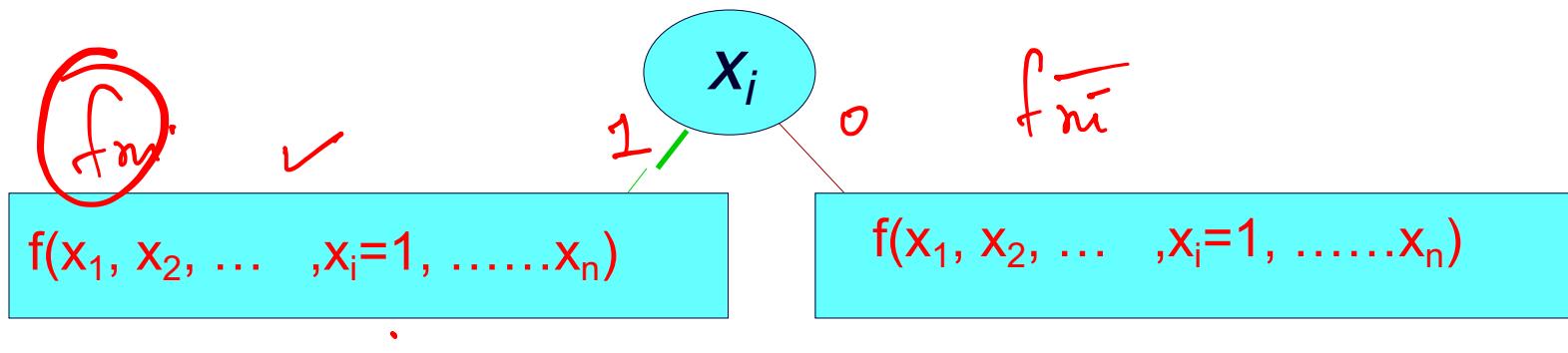
- ❖ $f(x_1, x_2, \dots, x_i, \dots, x_n) =$

$$x_i \cdot f(x_1, x_2, \dots, x_i=1, \dots, x_n) +$$

$$x_i' \cdot f(x_1, x_2, \dots, x_i=0, \dots, x_n)$$

$$\underline{x_i} \cdot \underline{f_{xi}} + \bar{\underline{x_i}} \cdot \underline{\bar{f}_{xi}}$$

$$f = \frac{f_{xi}}{f_{\bar{xi}}}$$

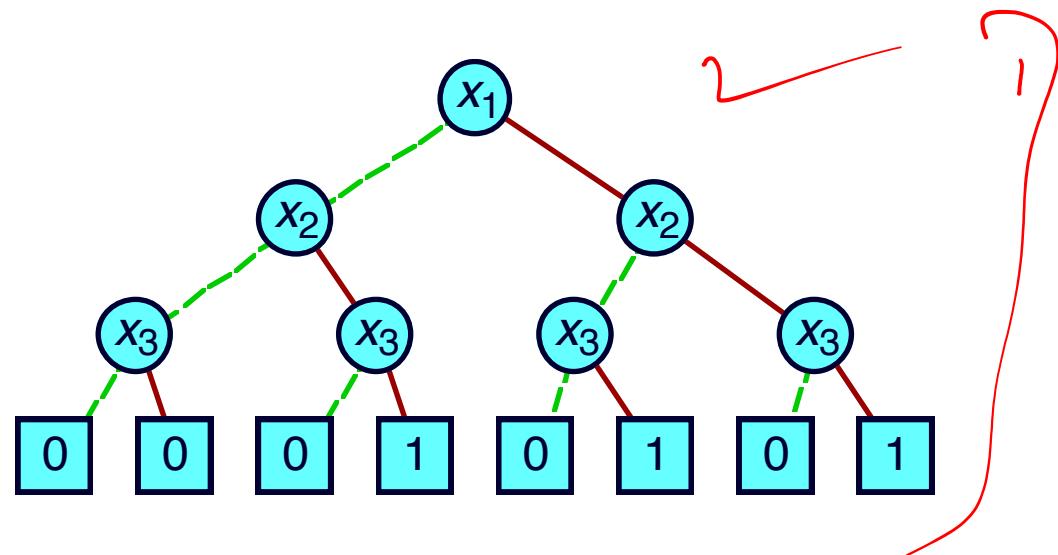


Decision Structures

Truth Table

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Decision Tree



- Vertex represents decision
- Follow green (dashed) line for value 0
- Follow red (solid) line for value 1
- Function value determined by leaf value.

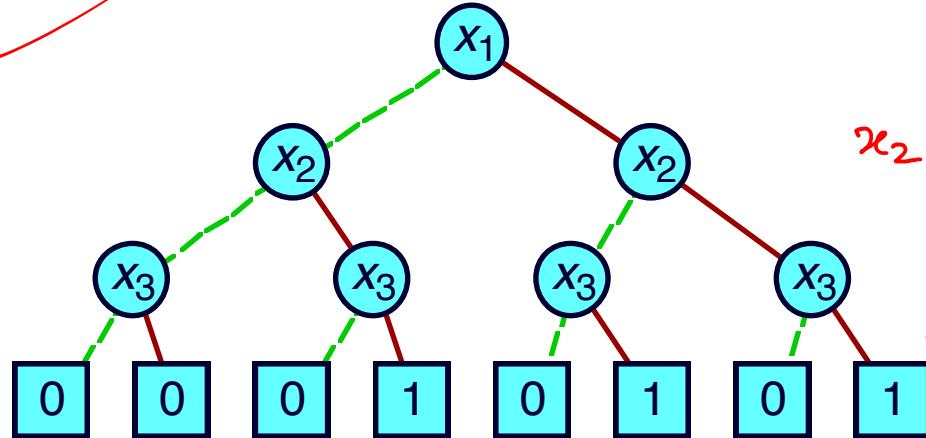


Example OBDD

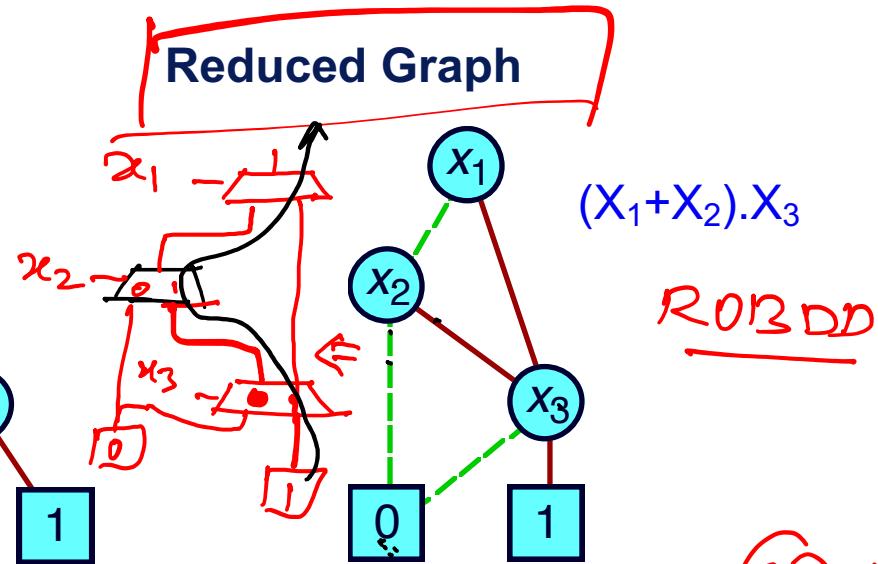
$$x_1 < x_2 < x_3$$

$$f_1 \equiv f_2$$

Initial Graph



Reduced Graph

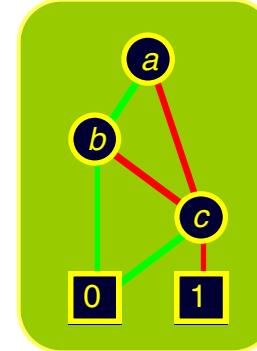
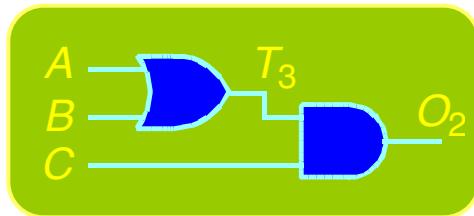
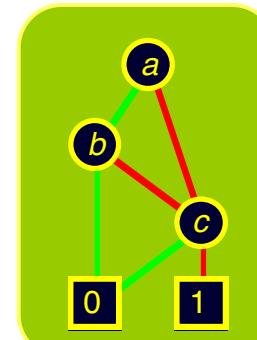
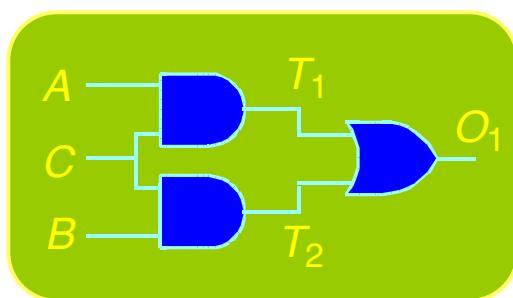


- Canonical representation of Boolean function
 - ❖ For given variable ordering
 - Two functions equivalent if and only if graphs isomorphic
 - o Can be tested in linear time
 - Desirable property: *simplest form is canonical.*



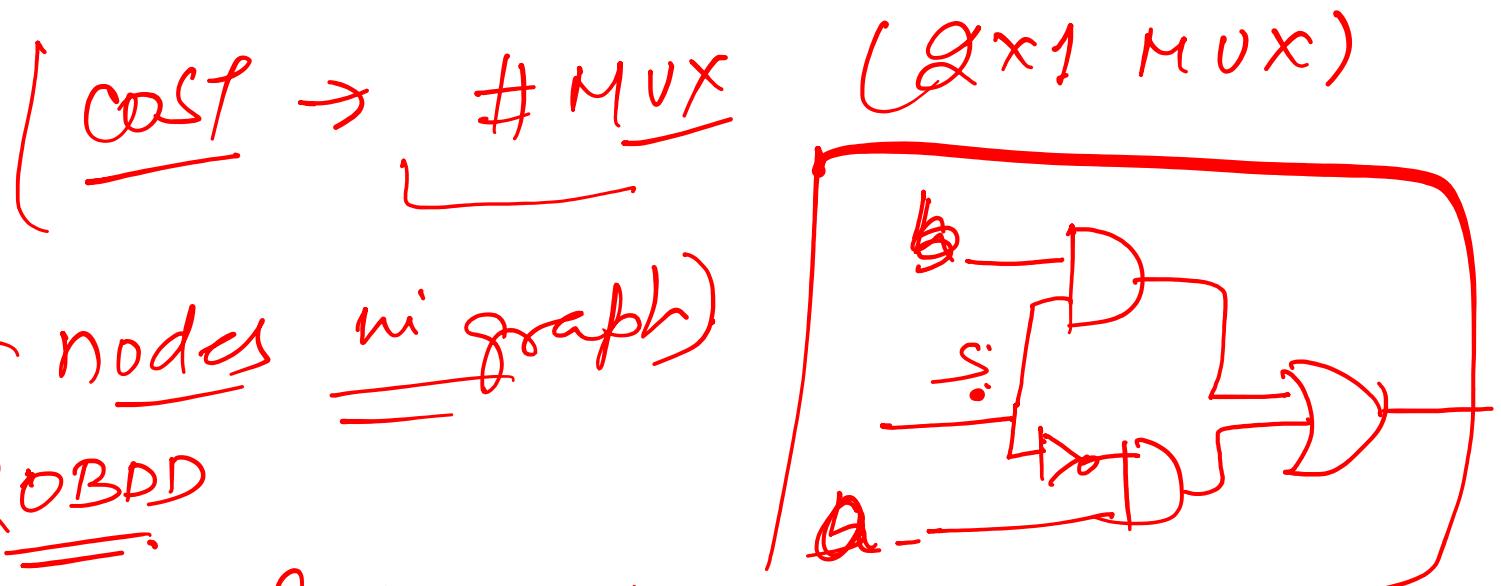
Binary Decision Diagram

- Generate Complete Representation of Circuit Function
 - Compact, canonical form



- Functions equal if and only if representations identical
- Never enumerate explicit function values
- Exploit structure & regularity of circuit functions





$$f = a \cdot \bar{s} + b \cdot s$$

$\underbrace{\text{Delay (Performance)}}$
 \downarrow (\downarrow longest path)

\Downarrow $\# \text{ MUX delay}$

Selector

$1 \text{ MUX delay} = t$

N.T.

$\approx \# \text{ MUX}$

bound



Shape & size of ROBDD

depends on Order of variable)

Th.

NP Complete

N nodes :

N' node =

$\lceil \frac{N'-N}{2} \rceil$

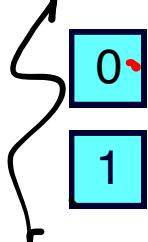
Synthesis ✓

VERIFICATION



Example Functions

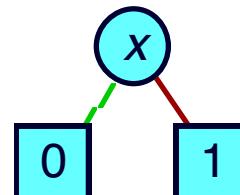
Constants



Unique unsatisfiable function

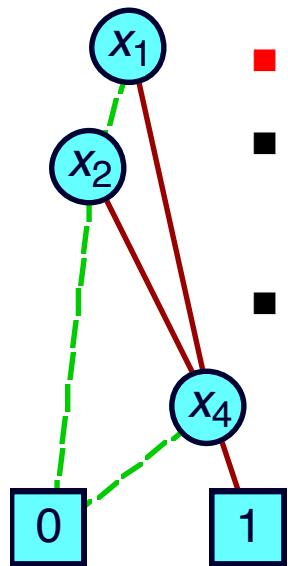
Unique tautology

Variable



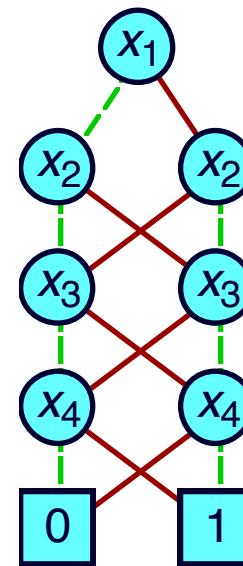
Treat variable as function

Typical Function



- $(x_1 + x_2) \cdot x_4$
- No vertex labeled x_3
 - ◆ independent of x_3
- Many subgraphs shared

Odd Parity



Linear representation

Operations with BDD

- ❖ Let v_1, v_2 denote root nodes of f_1, f_2 respectively , with $\text{var}(v_1) = x_1$ and $\text{var}(v_2) = x_2$
- ❖ If v_1 and v_2 are leafs, $f_1 \text{ OP } f_2$ is a leaf node with value $\text{val}(v_1) \text{ OP } \text{val}(v_2)$

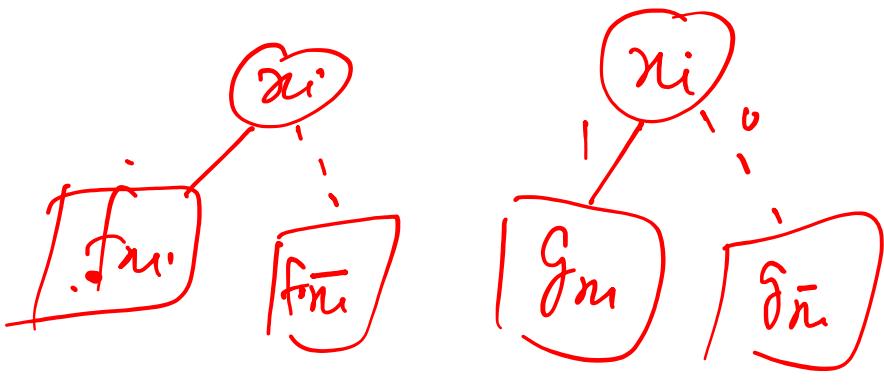


$$f(x_1, x_2 \dots x_i \dots x_n)$$

$$g(x_1, x_2 \dots x_i \dots x_n)$$

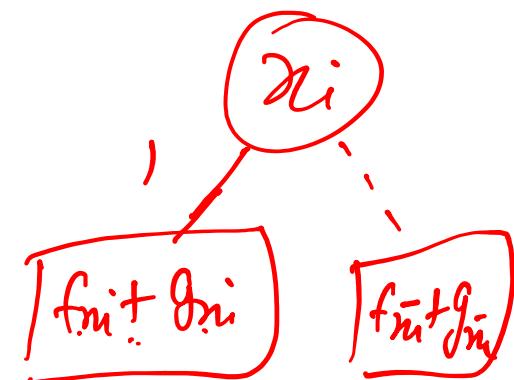
$$\Rightarrow x_i f_{xi} + \bar{x}_i \cdot f_{\bar{x}_i}$$

$$x_i g_{xi} + \bar{x}_i \cdot g_{\bar{x}_i}$$



OR

$$\underline{f+g}$$



$$x_i f_{xi} + \bar{x}_i f_{\bar{x}_i} + x_i g_{xi} + \bar{x}_i g_{\bar{x}_i}$$

$$\Rightarrow x_i (f_{xi} + g_{xi}) + \bar{x}_i (f_{\bar{x}_i} + g_{\bar{x}_i})$$

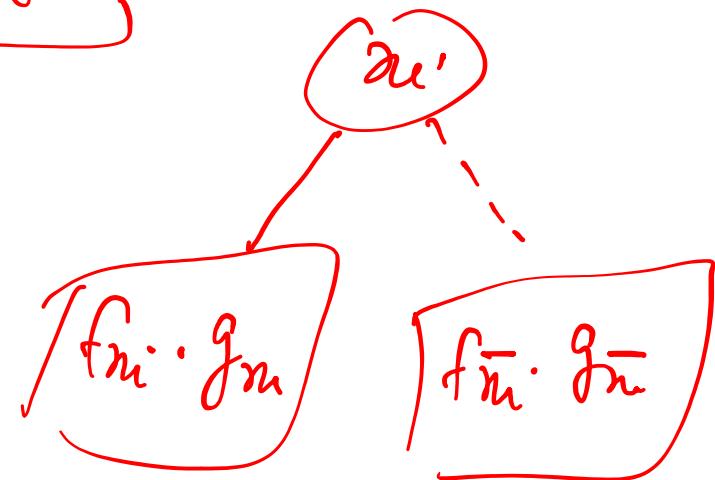


AND

$f \cdot g$

$$(\bar{x}_i f_{ni} + \bar{x}_i \bar{f}_{ni}) \cdot (\bar{x}_j g_{nj} + \bar{x}_j \bar{g}_{nj})$$

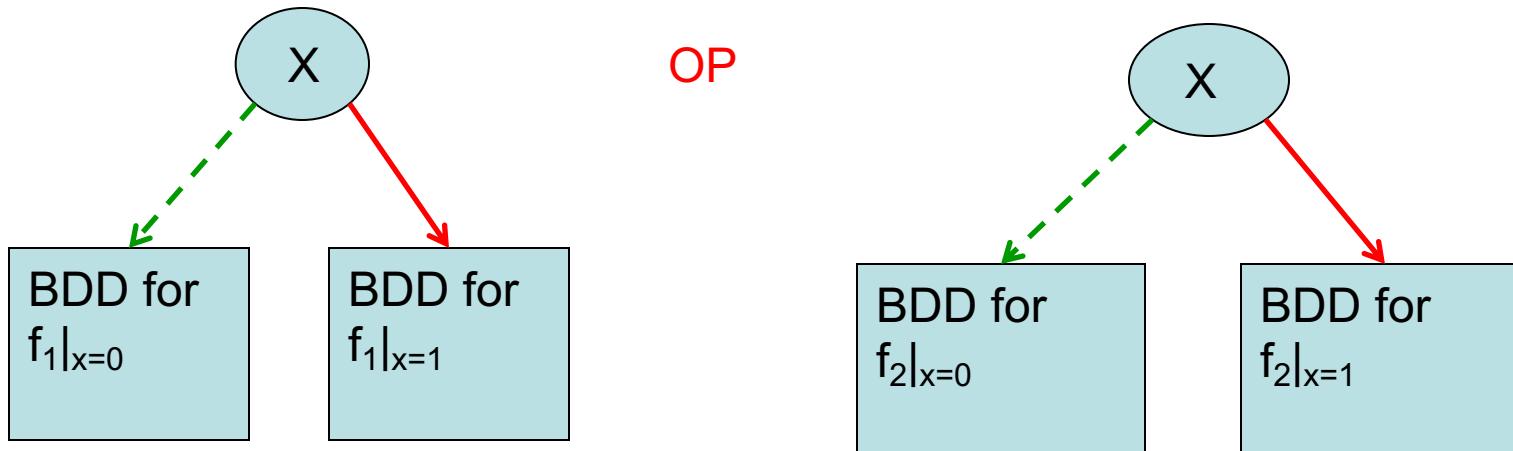
$$\bar{x}_i \underbrace{f_{ni} g_{nj}}_{\downarrow \cdot} + \bar{x}_i \bar{f}_{ni} \underbrace{g_{nj}}_{\downarrow}$$



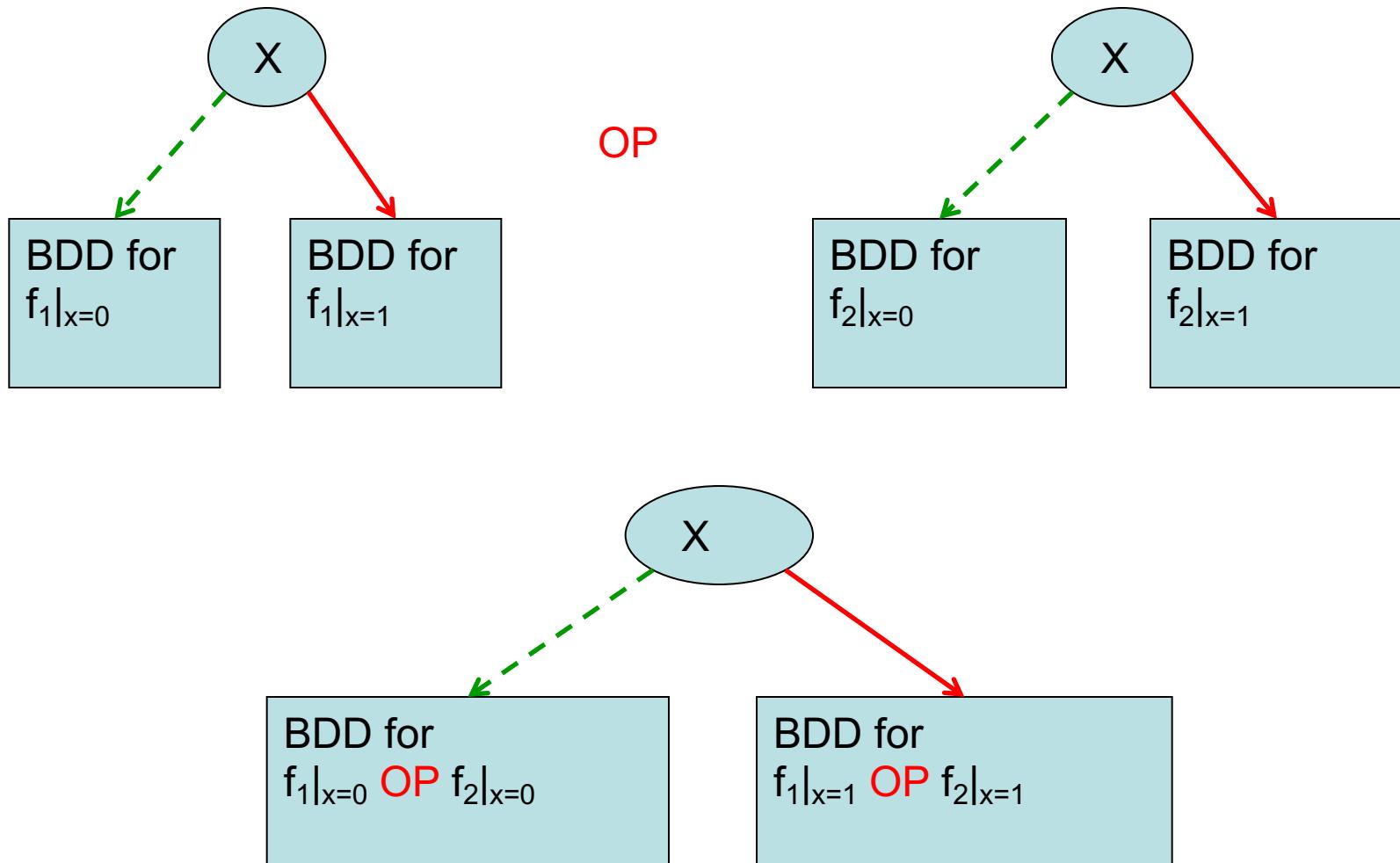
Operations with BDD

- ❖ If $x_1 = x_2 = x$, apply Shannon's expansion

$$f_1 \text{ OP } f_2 = x' \cdot (f_1|_{x=0} \text{ OP } f_2|_{x=0}) + x \cdot (f_1|_{x=1} \text{ OP } f_2|_{x=1})$$



Operations with BDD



$$f(x_1, x_2, \dots, x_n) \quad g(\underline{x_1 x_2 \dots x_n})$$

$$(x_1 \cdot f_m + \bar{x}_1 \bar{f}_{\bar{m}}) \cdot g.$$

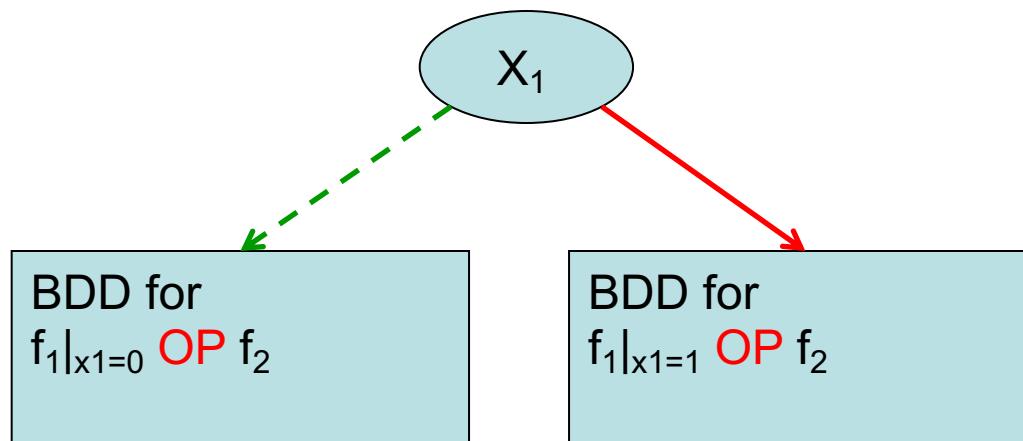
$$x_1 \cdot (f_m \cdot g) + \bar{x}_1 (\bar{f}_{\bar{m}} \cdot g)$$



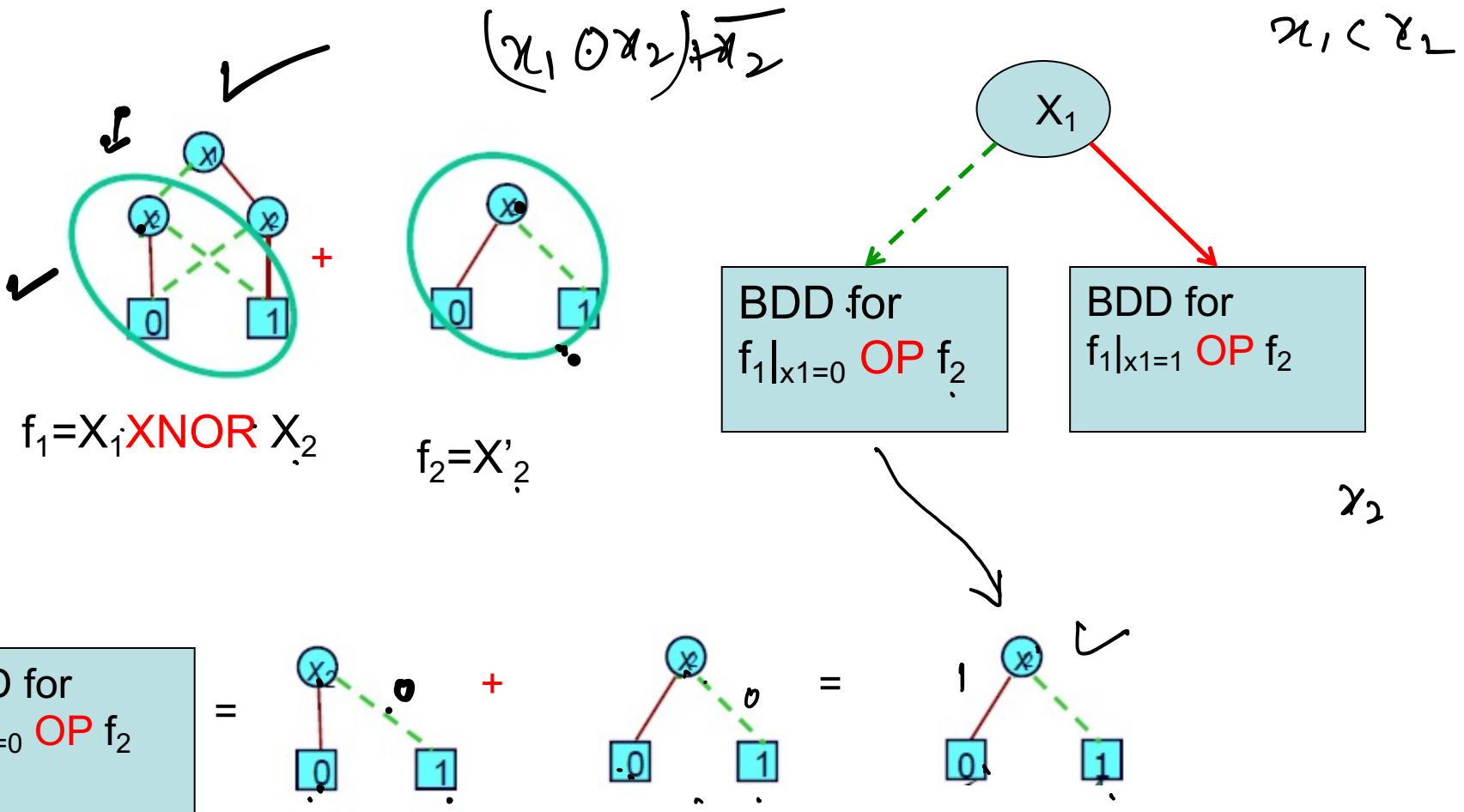
Operations with BDD

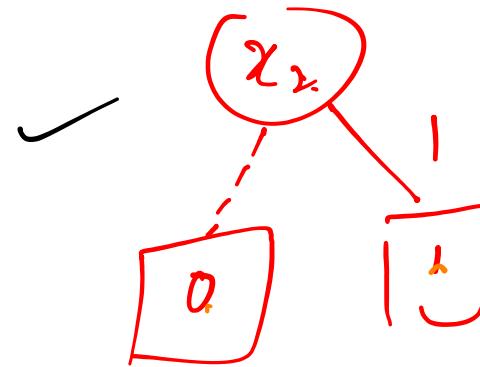
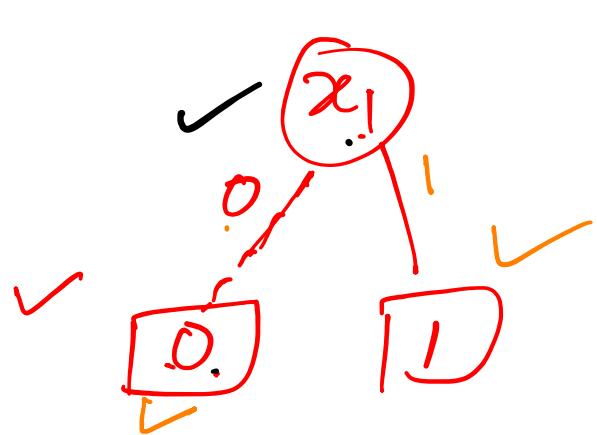
- ❖ Else suppose $x_1 < x_2 = x$, in variable order

$$f_1 \text{ OP } f_2 = x'_1 (f_1|_{x_1=0} \text{ OP } f_2) + x_1 (f_1|_{x_1=1} \text{ OP } f_2)$$



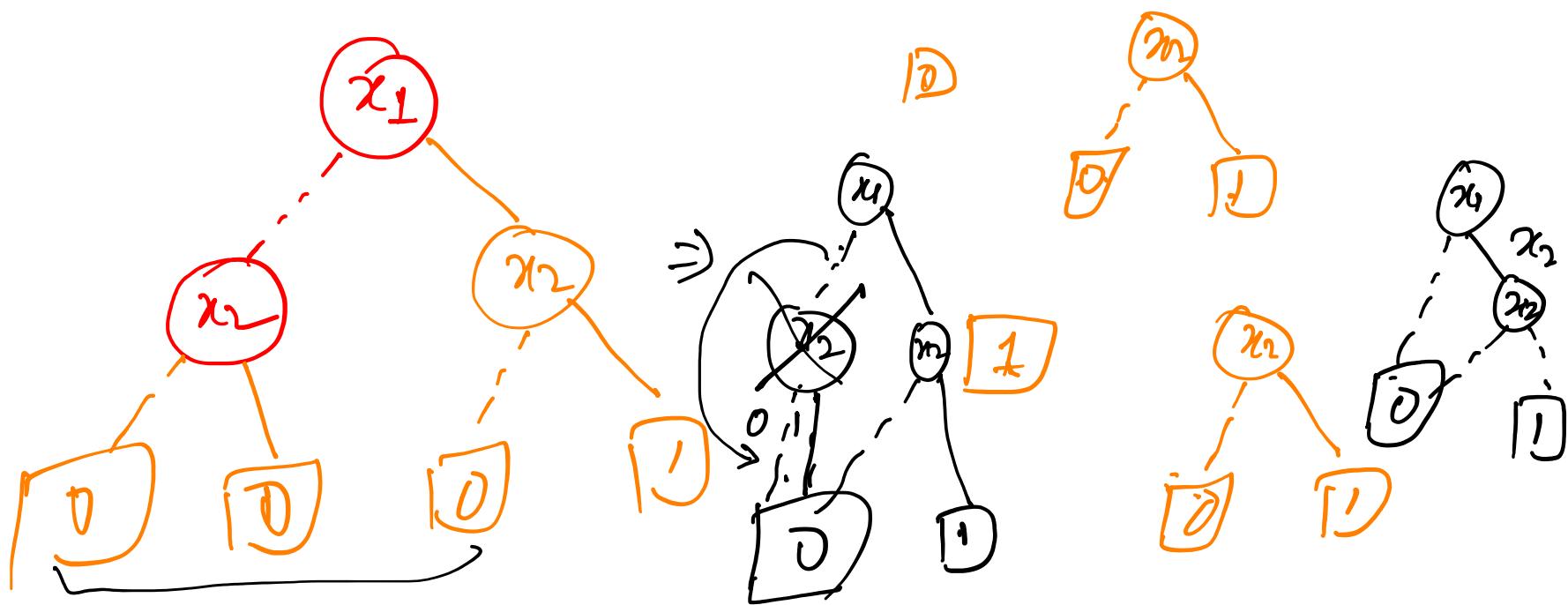
Operations with BDD: Example



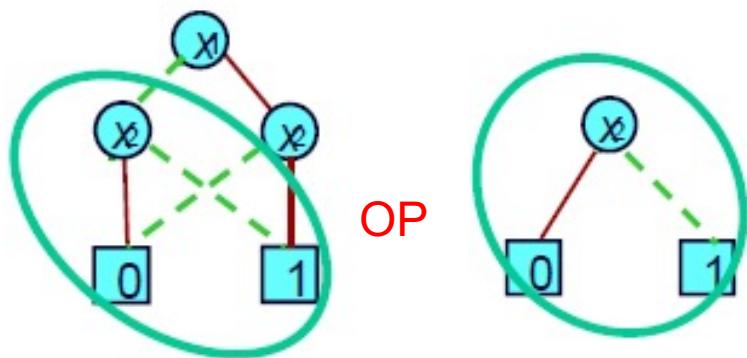


$$\frac{x_1 < x_2}{\text{order}}.$$

$$f = x_1 \cdot \overline{x_2}$$

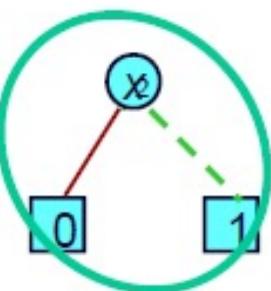


Operations with BDD: Example



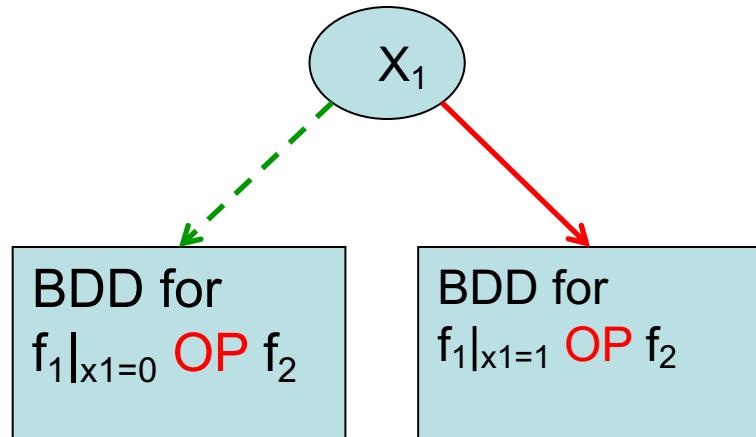
$$f_1 = X_1 \text{XNOR} X_2$$

OP



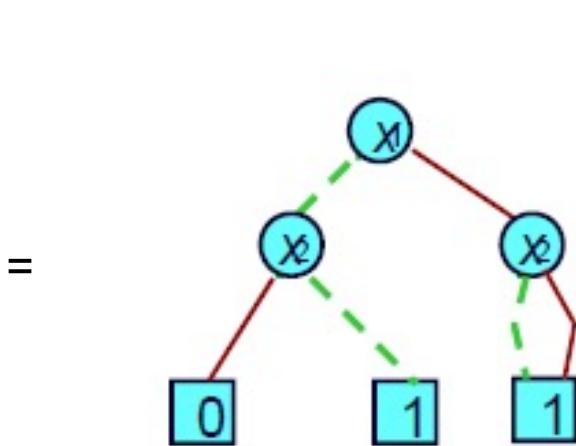
$$f_2 = \bar{X}_2$$

=

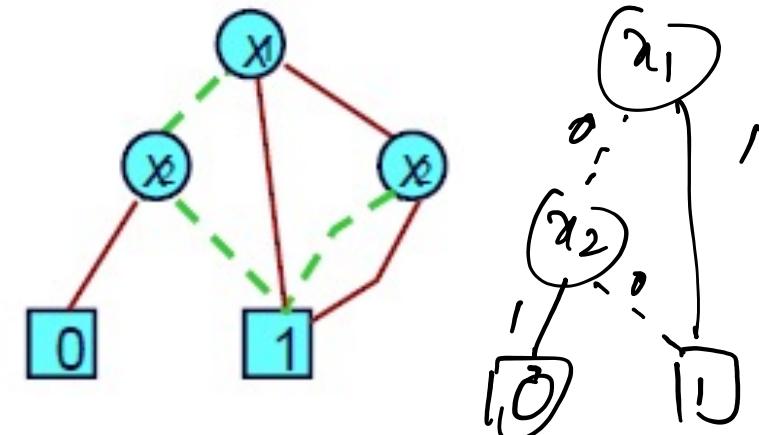


$$\text{BDD for } f_1|_{x_1=0} \text{ OP } f_2$$

$$\text{BDD for } f_1|_{x_1=1} \text{ OP } f_2$$

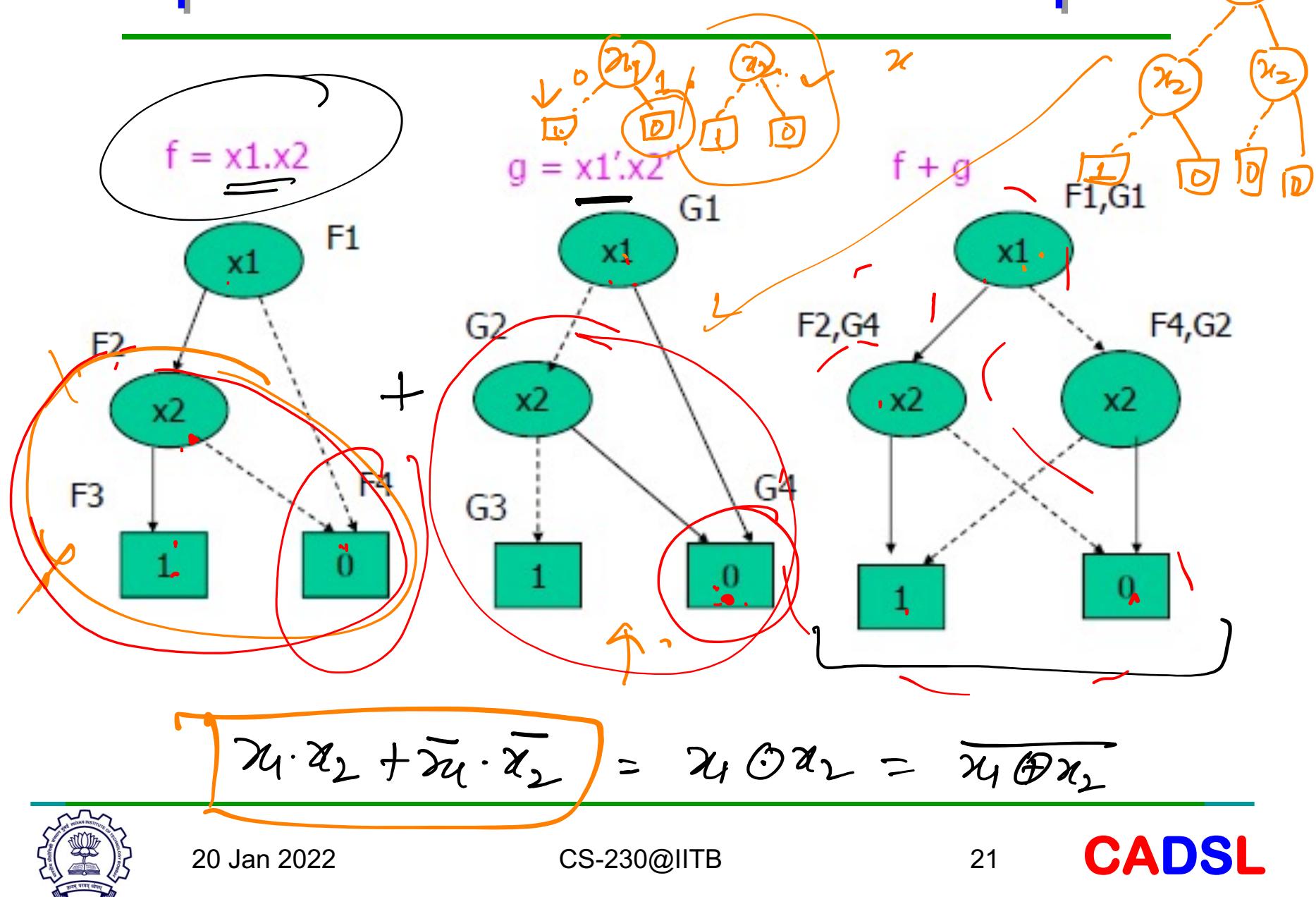


=

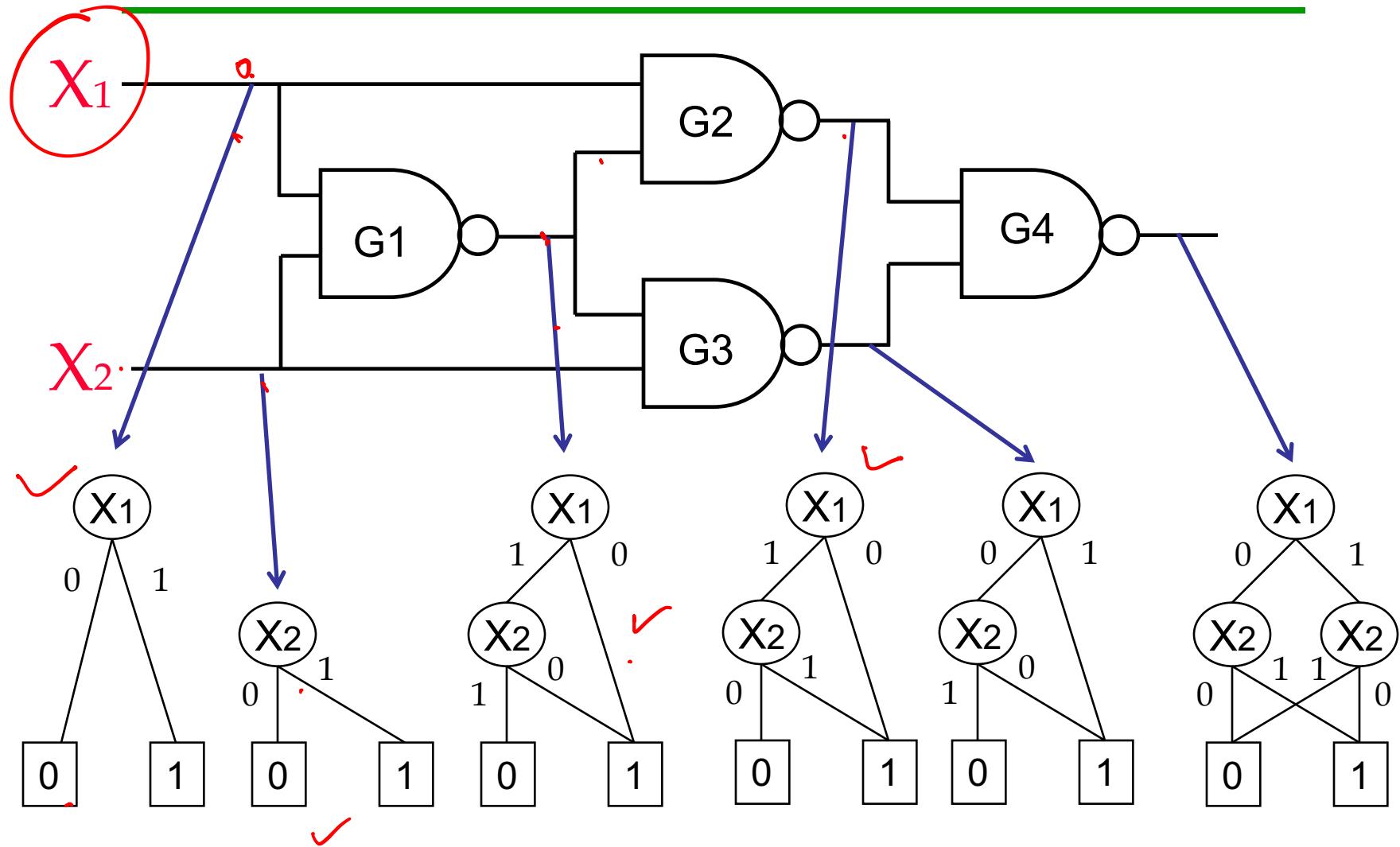


Operations with BDD: Example

$x_1 < x_2$



From Circuits to BDD

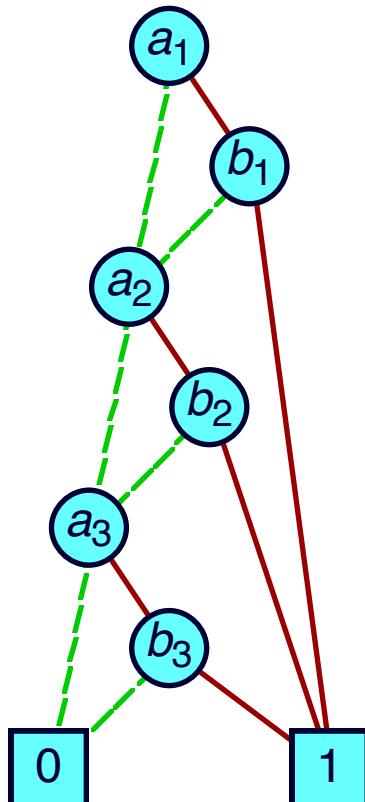


Effect of Variable Ordering

$$(a_1 \wedge b_1) \vee (a_2 \wedge b_2) \vee (a_3 \wedge b_3)$$

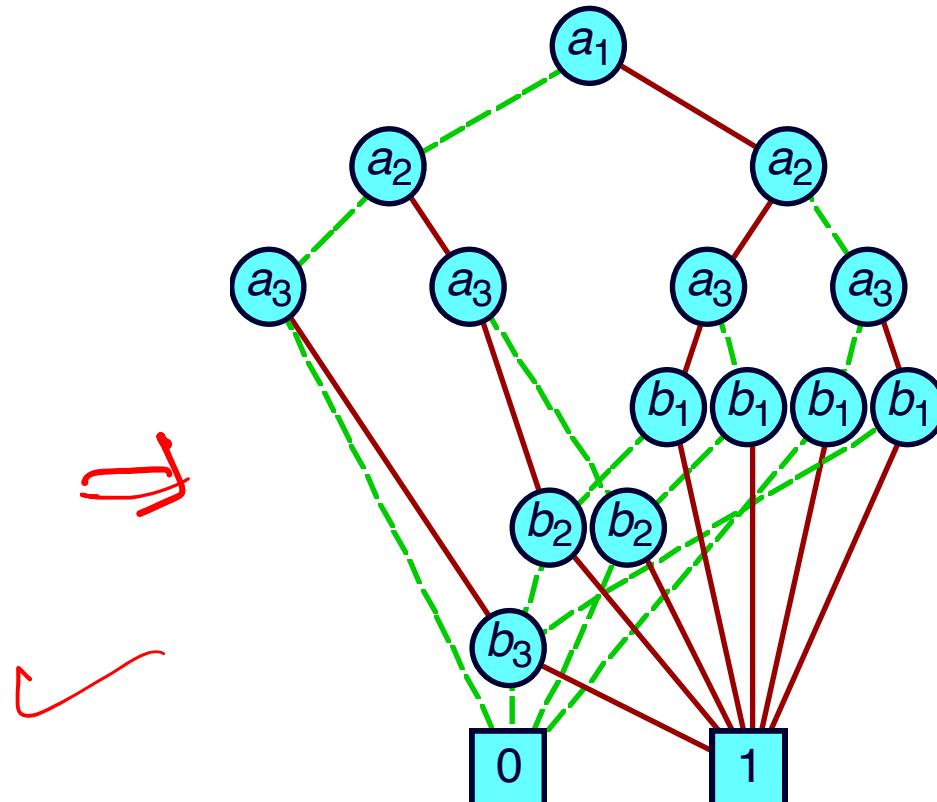
$a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3$

Good Ordering



Linear Growth

Bad Ordering



Exponential Growth



Selecting Good Variable Ordering

- Intractable Problem
 - Even when problem represented as OBDD
 - i.e., to find optimum improvement to current ordering
- Application-Based Heuristics
 - Exploit characteristics of application
 - e.g., Ordering for functions of combinational circuit
 - Traverse circuit graph depth-first from outputs to inputs
 - Assign variables to primary inputs in order encountered



Selecting Good Variable Ordering

🟡 Static Ordering

- Fan In Heuristic
- Weight Heuristic

🟡 Dynamic Ordering

- Variable Swap
- Window Permutation
- Sifting



Dynamic Variable Reordering



Richard Rudell, Synopsys

🟡 Periodically Attempt to Improve Ordering for All BDDs

- ❖ Move each variable through ordering to find its best location



Has Proved Very Successful

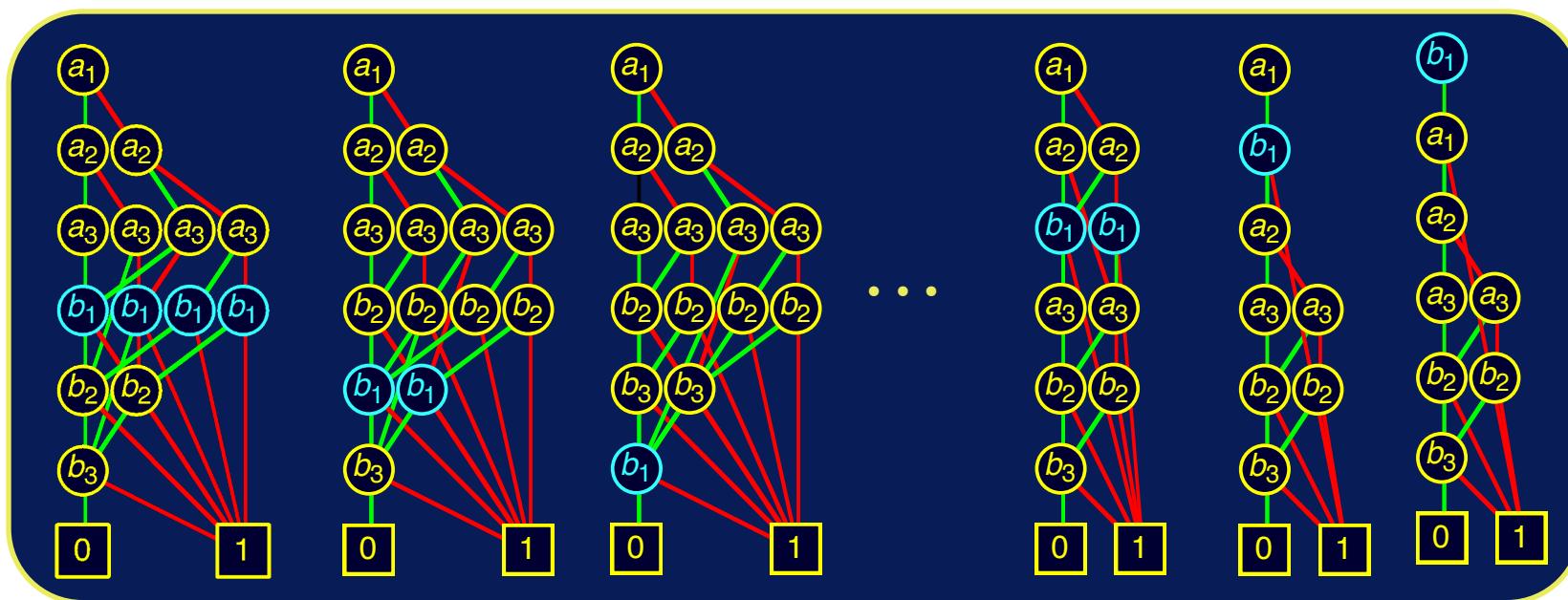
- ❖ Time consuming but effective



Dynamic Reordering By Sifting

- Choose candidate variable
- Try all positions in variable ordering
 - Repeatedly swap with adjacent variable
- Move to best position found

Best Choices



ROBDD Sizes & Variable Ordering

- Bad News 💣
 - Finding optimal variable ordering NP-Hard
 - Some functions have exponential BDD size for all orders e.g. multiplier
- Good News 😊
 - Many functions/tasks have reasonable size ROBDDs
 - Algorithms remain practical up to 1,000,000 node OBDDs
 - Heuristic ordering methods generally satisfactory
- What works in Practice 🤝
 - Application-specific heuristics e.g. DFS-based ordering for combinational circuits
 - Dynamic ordering based on variable sifting (*R. Rudell*)

WXFRHU



Thank You

