Logic Synthesis

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.iitb.ac.in

viren@cse.iitb.ac.in

EE-677: Foundations of VLSI CAD

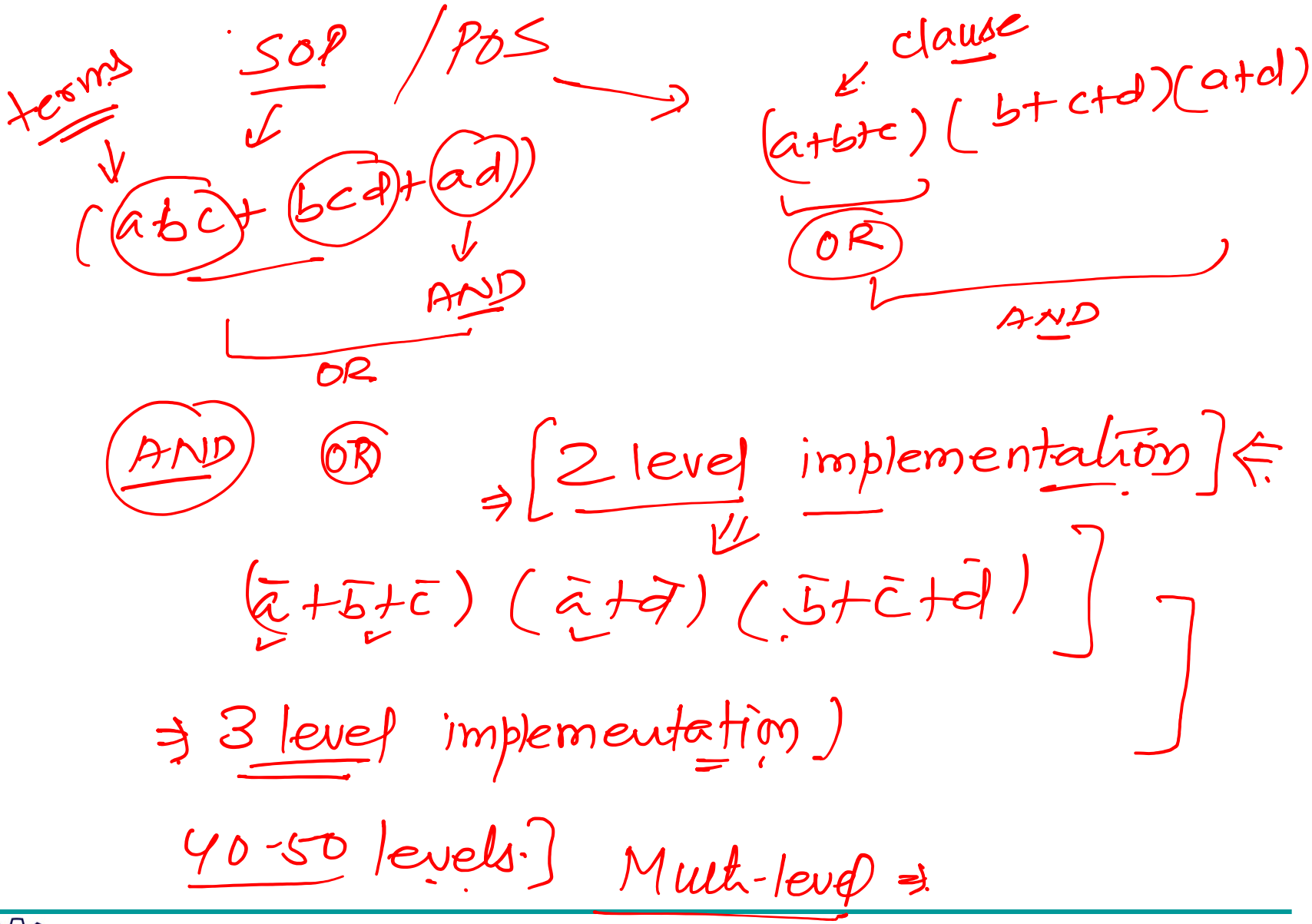
Lecture 18 on 20 Sep 2021

CADSL

Midsem =

Logic Design → Representation { Canonical, Non-canonical

Logic Design → Optimization { Area (Cost), Performance, Power


terms
SOP / POS
clause
(abc + bcd + ad)
(a+b+c)(b+c+d)(a+d)
AND
OR
OR
AND
AND
OR
⇒ [2 level implementation] ⇐
(ā+b+c̄)(ā+d)(b̄+c̄+d̄)
⇒ 3 level implementation)
40-50 levels.]
Multi-level ⇒

Two-level Logic Optimization Motivation

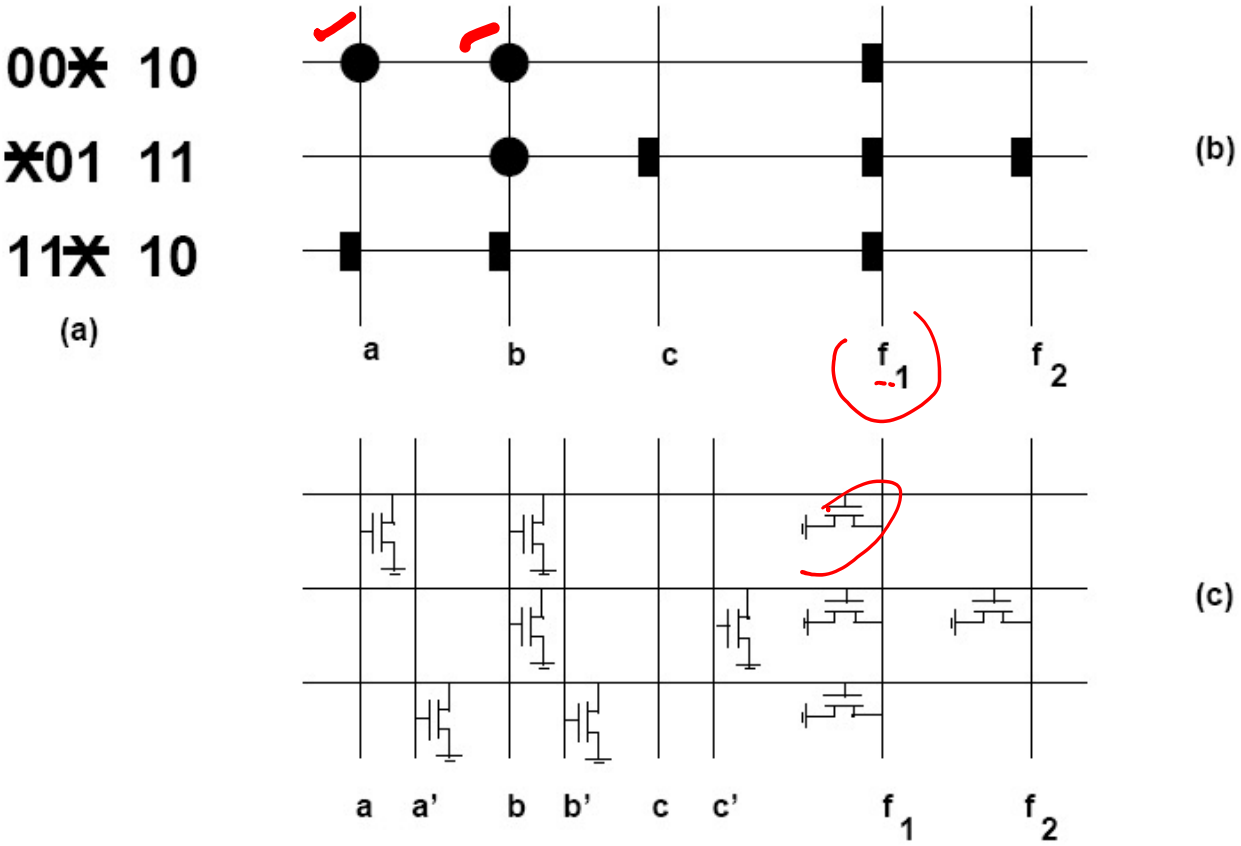
- Reduce size of the representation
- Direct implementation
  - PLAs reduce size and delay
- Other implementation styles
  - Reduce amount of information
  - Simplify local functions and connections

Programmable Logic Arrays

- Macro-cells with rectangular structure
  - Implement any multi-output function
  - Layout generated by module generators
  - Fairly popular in the seventies/eighties
- Advantages
  - Simple, predictable timing
- Disadvantages
  - Less flexible than cell-based realization
- Open issue
  - Will PLA structures be useful with new nanotechnologies? (e.g., nanowires)

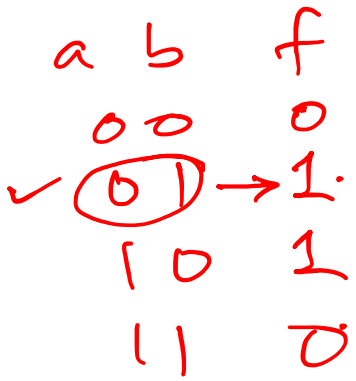
Programmable Logic Array

- $f_1 = a'b' + b'c + ab$; $f_2 = b'c$



Two-Level Minimization

- Assumptions
  - Primary goal is to reduce the number of implicants
  - All implicants have the same cost
  - Secondary goal is to reduce the number of literals
- Rationale
  - Implicants correspond to PLA rows
  - Literals correspond to transistors



Max 4 → ab + bcd + adef ✓ 9

Area (Cost) ✓ abd ab + bcd + aef 5.

↑ Max 3

= min (#literals + #terms)

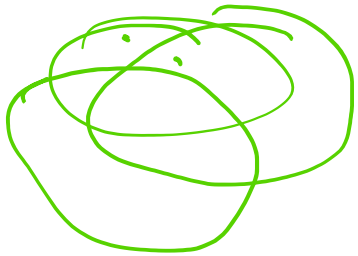
delay

= min (Max { literals in terms } + #terms)

min # terms ⇐ implicants

Definitions

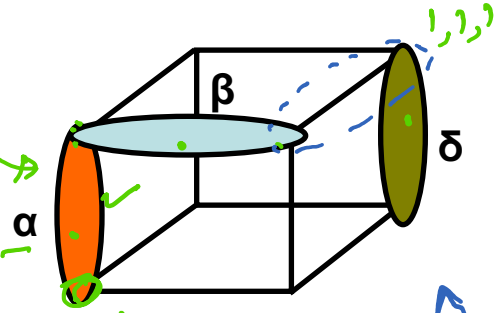
- Minimum cover
  - Cover of a function with minimum number of implicants
  - Global optimum
- Minimal cover or irredundant cover
  - Cover of the function that is not a proper superset of another cover
  - No implicant can be dropped
  - Local optimum
- Minimal w.r.to 1-implicant containment
  - No implicant contained by another one
  - Weak local optimum

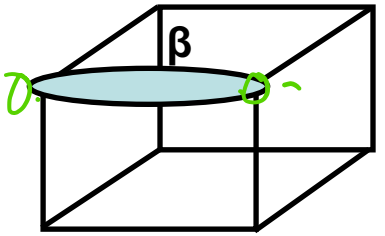


Example

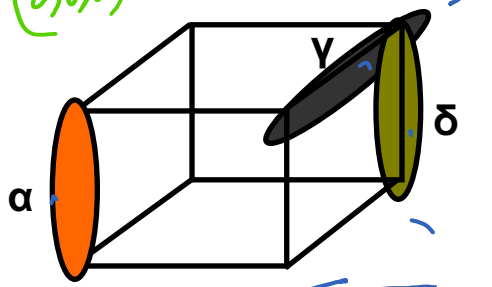
- $f_1 = a'b'c' + a'b'c + ab'c + abc + abc'$; $f_2 = a'b'c + ab'c$

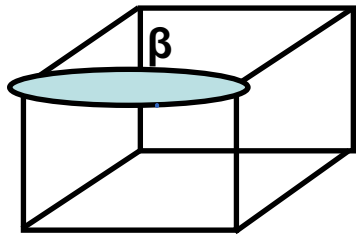
Minimum cover



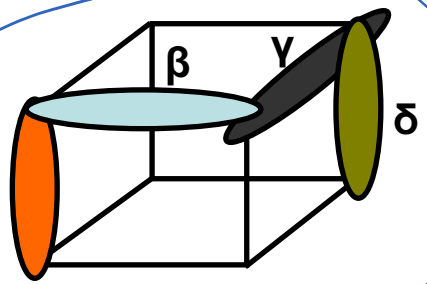


Irredundant cover

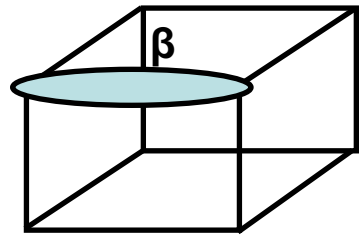




Minimal cover w.r. to single implicant containment



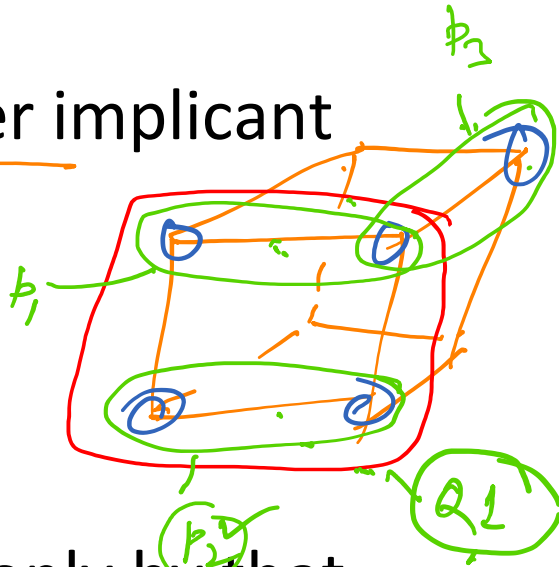
f1

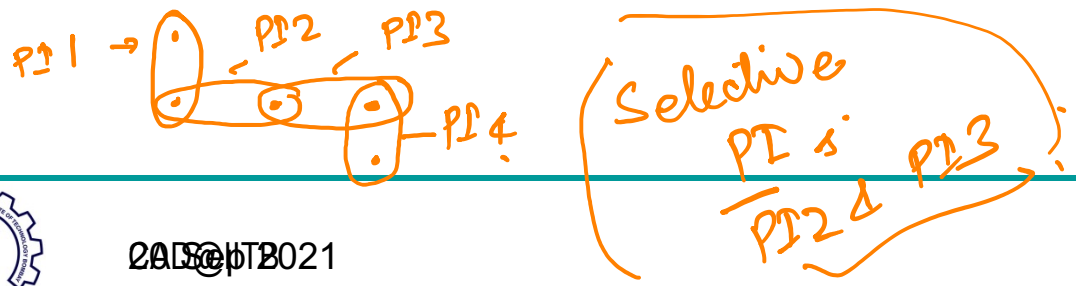


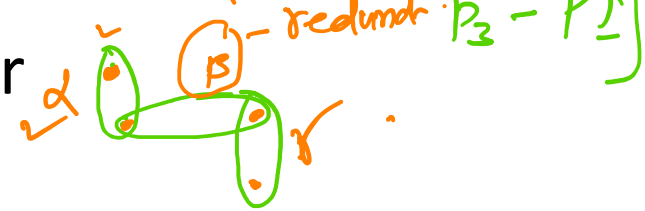
f2

Definitions

- Prime implicant
  - Implicant not contained by any other implicant
- Prime cover
  - Cover of prime implicants
- Essential prime implicant
  - There exist some minterm covered only by that prime implicant
  - Needs to be included in the cover







Two-level Logic Minimization

- Exact methods
  - Compute minimum cover
  - Often difficult/impossible for large functions
  - Based on Quine-McCluskey method
- Heuristic methods
  - Compute minimal covers (possibly minimum)
  - Large variety of methods and programs
    - MINI, PRESTO, ESPRESSO

Exact Logic Minimization

- Quine's theorem:
  - There is a minimum cover that is prime
- Consequence
  - Search for minimum cover can be restricted to prime implicants
- Quine-McCluskey method
  - Compute prime implicants
  - Determine minimum cover

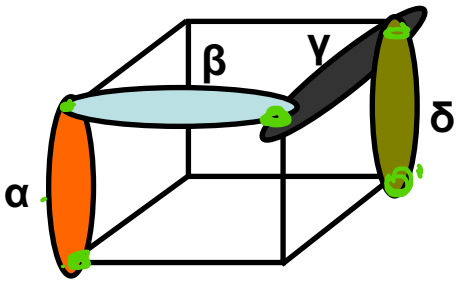
Prime Implicant Table

- Rows: minterms
- Columns: prime implicants
- Exponential size
  - 2^n minterms
- Remarks
  - Some functions have much fewer primes
  - Minterms can be grouped together
  - Implicit methods for implicant enumeration

Example

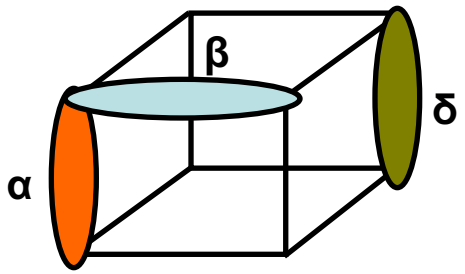
- f = a'b'c' + a'b'c + ab'c +abc +abc'
- Primes:
- Table:

α	00*	1
β	*01	1
γ	1*1	1
δ	11*	1



Prime implicants of f

	α	β	γ	δ
000	1	0	0	0
001	1	1	0	0
101	0	1	1	0
111	0	0	1	1
110	0	0	0	1



Minimum cover of f

Minimum Cover: Early Methods

- Reduce table
  - Iteratively identify essentials,
    save them in the cover.
    Remove covered minterms
- Petrick's method
  - Write covering clauses in *pos* form
  - Multiply out pos form into *sop* form
  - Select cube of minimum size
- Remark
  - Multiplying out clauses has exponential cost

Thank You