VLSI SYSTEMS LAB

Multi-Level Programmable Array

20023179 Kim, Hyung ock

KAIST

Inroduction

- Regular Structures
  - Why? Easy to P&R(almost no need to P&R)
  - Examples
    - PLA – like
    - Binary Tree base
    - Lattice Diagram
      - Better solution than UAA
      - UAA is treated as attempt to combine PLA-like and tree-like

Inroduction

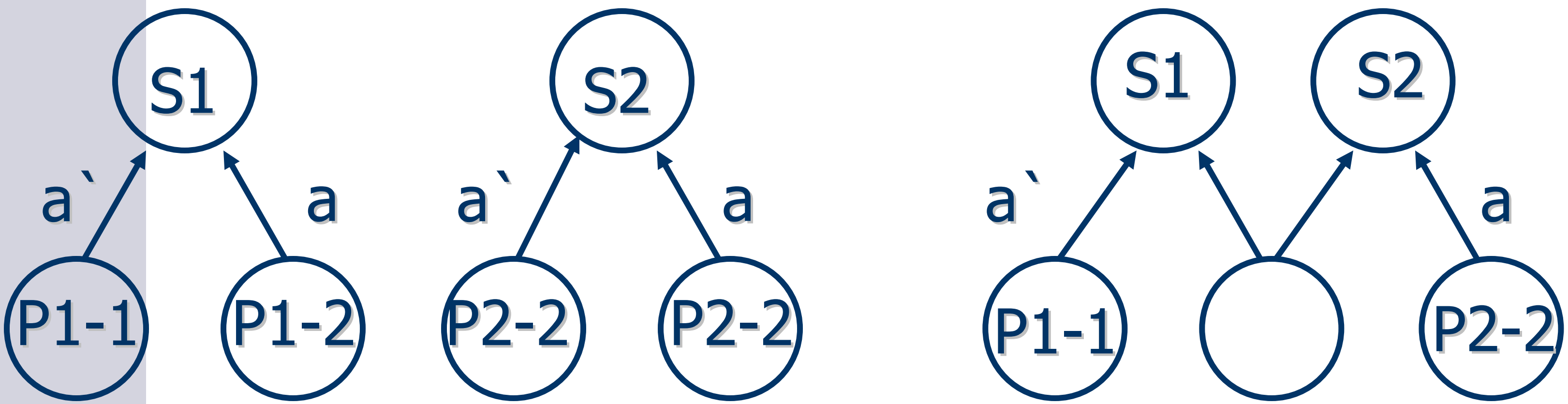
- This Presentation is composed as following
  - Intro to Lattice Diagram
  - MOPS for multiple-out Lattice Diagram
  - Generalized architecture for MOPS

1. Intro to Lattice Diagram

- Chacteristics
  - Like Tree and similar to BDD.
  - BDD has combined predecessors if and only if predecessors in the same level is equal.
  - But Lattice Diagram has always combine neighbor predecessors by some Rule. It occurs repetition of control variables.
  - Although BDD grows horizontally, Lattice grows vertically by the repetition of variables…
  - BDD and Lattice Diagram is made of MUX.

1. Intro to Lattice Diagram

- Combining Rule
  - Basic rule is the combining of two predecessors by XOR



(a AND P1-2) XOR (a’ AND P2-1)

  - More rule and method are introduced in “LATTICE DIAGRAMS USING REED-MULLER LOGIC” by Perkowski

2. MOPS for multiple-out Lattice

- Some problems in Lattice Diagram
  - Repetition of control variable
    - It increases vertical depth.
    - This problem controlled by variable ordering.
  - In the case of multi-output func
    - Ordering is not easy to be performed
    - There is quite waste for one block
    - And Partitions generate big empty subareas
      - Not good method, it leads to horizontal growth.

2. MOPS for multiple-out Lattice

- Functional Decomposition
  - Basic conception is to divide function to sub-functions
  - There are some decomposition methods
    - AND Decomposition, OR ~, Decomposition with Mux
  - Multi-output func can be decomposed by symmetric func
    - Multi-output func can be composed of Boolean operation(AND, OR, EXOR) of symmetric funcs.
    - Because of no repetition of variable in symmetric func, this method is very nice to reduce vertical depth.

2. MOPS for multiple-out Lattice

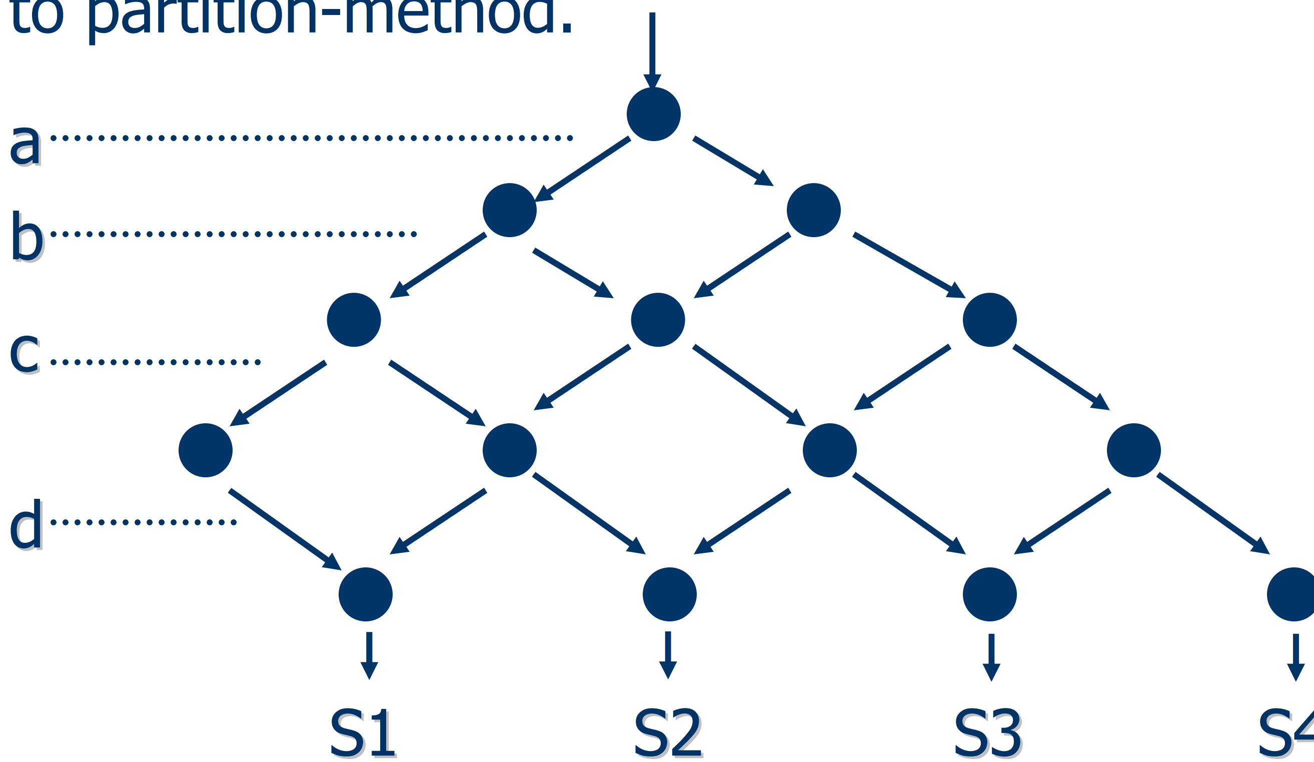
- What is symmetric func?
  - All minterms that have same number of ones in their binary number have same value(zero, or one).
  - Eg

F(a,b,c,d)
: It polarity 1111
= S3,4

a,b \ c,d	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	0

2. MOPS for multiple-out Lattice

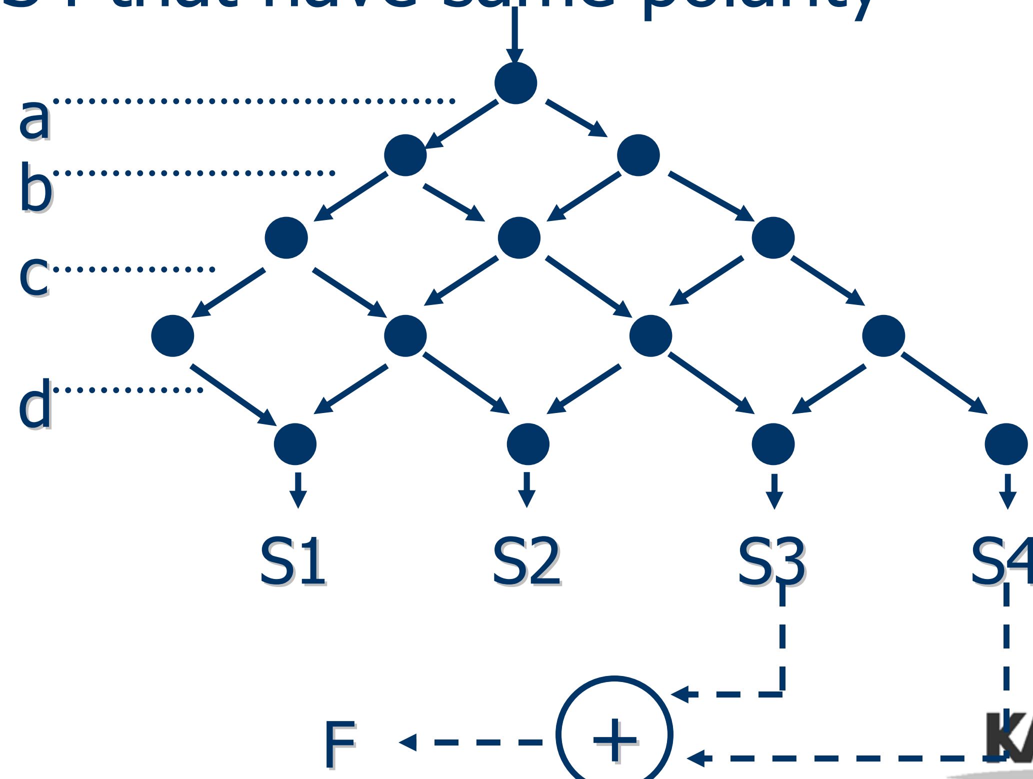
- MOPS for 4-variables
  - MOPS is one diagram but it can express all symmetric func which has same polarity
    - So that reason, it reduces horizontal width compare to partition-method.



2. MOPS for Multiple-out Lattice

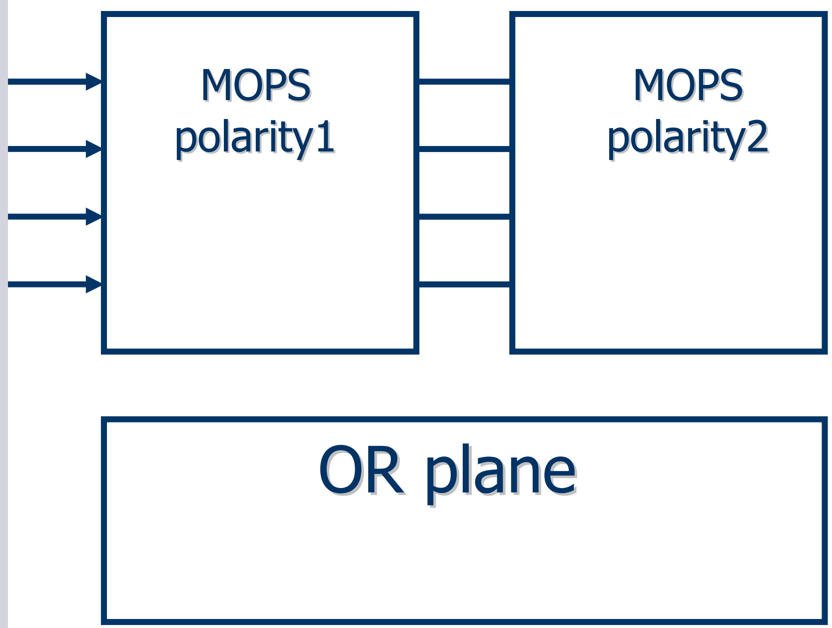
- Examples of using MOPS
  - F = (~b XOR ~d) OR (a XOR c) OR (abcd)
    ➔ It is decomposed to two symmetric functions S3, S4 that have same polarity

a,b \ c,d	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	0



3.Generalized architecture for MOPS

- Every multi-output Boolean func can be decomposed to vector-OR of symmetric func of variable polarity
  - Each MOPS has same control variable but different polarity
  - Outputs of two MOPSes are combined in OR plane



3.Generalized architecture for MOPS

- Every multi-output func with subset SVi, i= 1~k of mutual symmetric variables can be decompsed to serial composition of K MOPS arrays followed by AND/OR plane.
  - F(SV) = f1(SV1) OR f2(SV2) … OR fk(SVk)
  - Each fi(SVi) is symmetric , it can be expressed by one MOPS