VLSI SYSTEMS LAB

Layout-Driven Synthesis For Submicron Technology : Mapping Expansions To Regular Lattices

High Level Synthesis Homework #2

2002/10/08
SeonPil Kim

KAIST

Components of Paper

1. Expansion of Function

   creates several successor nodes of this node. Functions f corresponds to the initial node in the lattice, initially a tree.

2. Joining (Reverse Expansion)

   joins several nodes of a bottom of the lattice . This is in a sense a reverse operation to expansion.

3. Regular Geometry

   to which the nodes are mapped, this geometry guides which nodes of the level are to be joined.

4. Process of Lattice Creation

5. Conclusion

Expansion - introduction

- □ What is Expansion?
  - The fundament of out approach
  - Expansion is operators that transform a function to few simpler function.

- □ Classification of Expansion
  - In Uniqueness
    - Canonical (such as Shannon Expansion)
    - Non-Canonical (such as SOP Expansion)
  - In Implement type
    - Maximum type – Generalization of Shannon, Post (Multi-Valued Shannon), SOP Expansions
    - Linear Independent type – Generalization of Davio expansions (This Expansion is usually for arbitrary algebra that have at least one Linear Operation)

Expansion – nodes of Shannon and pD,nD

- Figure 1. present different expansion nodes for various kinds of expansions

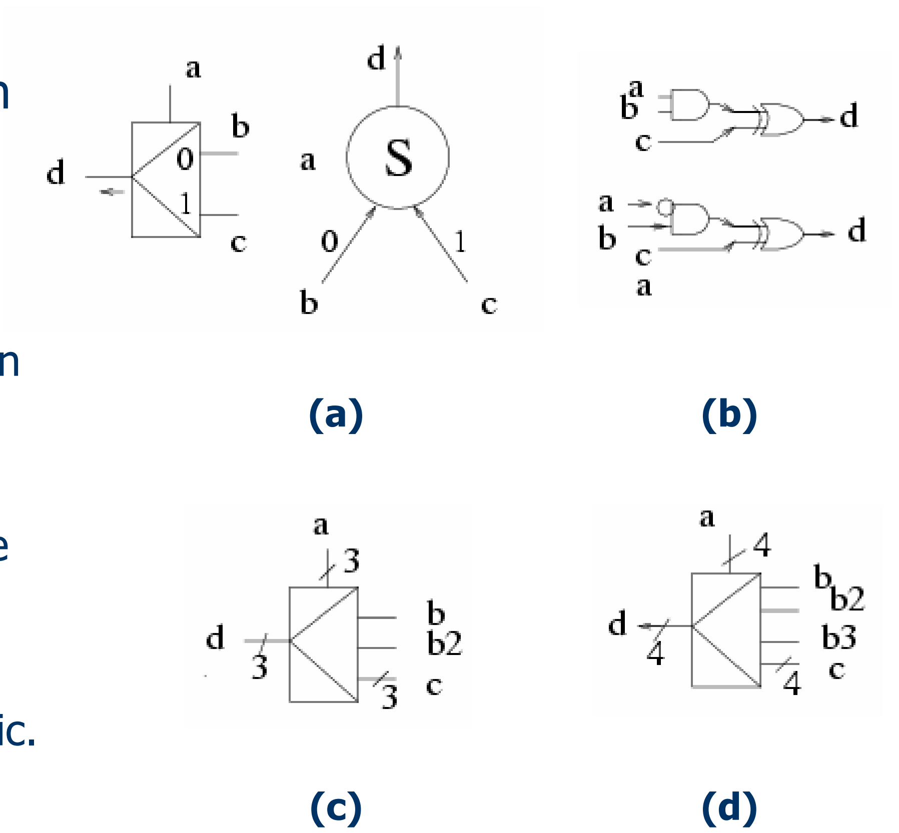
- Figure 1-(a) :

  shows two views of a cell for Shannon (S) Expansion.
- Figure 1-(b) :

  shows the positive Davio (pD) and the negative Davio (nD)
- Figure 1-(c) :

  shows Shannon node for 3-valued logic.
- Figure 1-(d) :

  shows node for 4-valued logic.



(a) (b) (c) (d)

Figure 1.

Expansion – nodes of SOP

□ SOP Expansion

$$f = l_j f_{l_j} + l_r f_{l_r} \ldots + l_s f_{l_s}.$$

- In Binary SOP expansions a branching from node f is for any subset of literals li that their union covers the node function f.

- Trees, diagrams, lattices based on SOP (binary and MV) expansion are not ordered and not canonical.

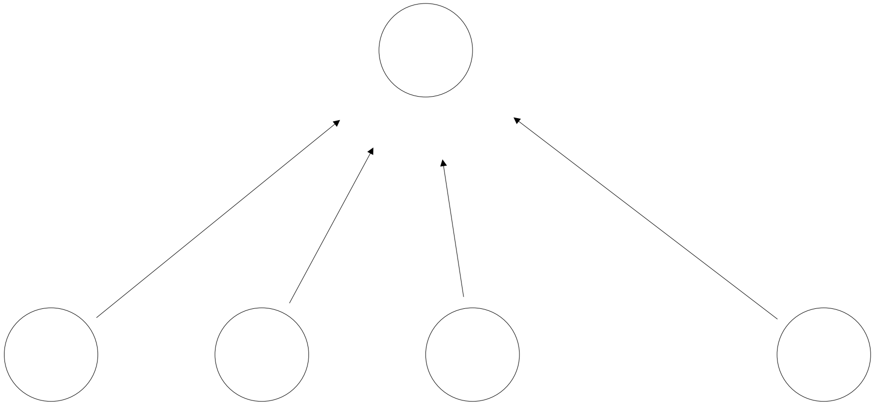


Figure 2. SOP

Expansion – cofactor

- □ Cofactors
  - Each binary function f is represented by pair

    fa = [ON(fa) , OFF(fa)]
  - Every Cofactor fa for the product a of an (in)complete function f can be interpreted as intersecting f with a and replacing all K-map cells outside product a with don't cares.

- □ Standard Cofactor
  - A standard cofactor fx whew x is variable does not depend on this variable.
  - Standard cofactors are in general not disjoint

- □ Vacuous Cofactor (v-factor)
  - fx is still a function of all variable including x , but as a result of cofactoring the variable x becomes vacuous.

Joining – v-cofactor case

- □ Vacuous cofactor Expansion
  - For any two disjoint products a1 and a2, the v-cofactor fa1 and ga2 are disjoint.
  - Then, fa1 and ga2 are in incomplete tautology relation.

    -> functions f and g are not changed when fa1 and ga2 are joined (OR-ed) to create a new function :

$$a_1 f_{a_1} \;+\; \overline{a_2} g_{a_2}$$

  - This way ENTIRE lattice is created level-by-level.
  - Functions in lattice nodes become more and more unspecified when variables in levels are repeated.

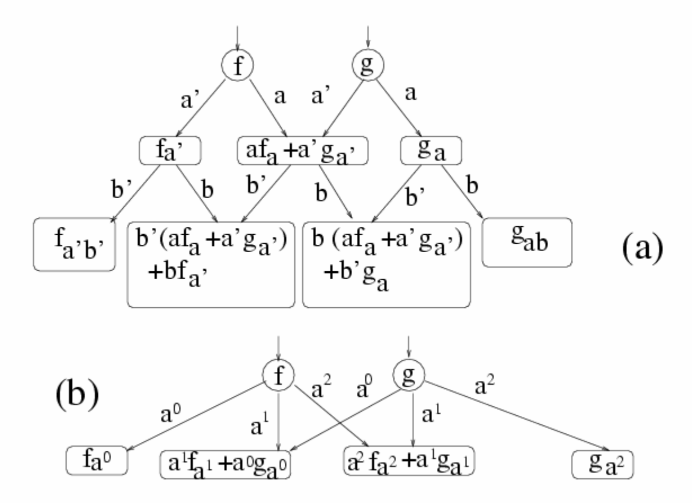


Figure 3. Expansion and Joining (a) Shannon (b) Ternary Post

Joining– not v-cofactor case

- When g and h is not disjoint

<Example – Figure 2.>

- g2 node and h0 node are combined to a new node ag2 XOR h0
- Correction Term ah0 and ag2 are propagated to left and right, respectably.

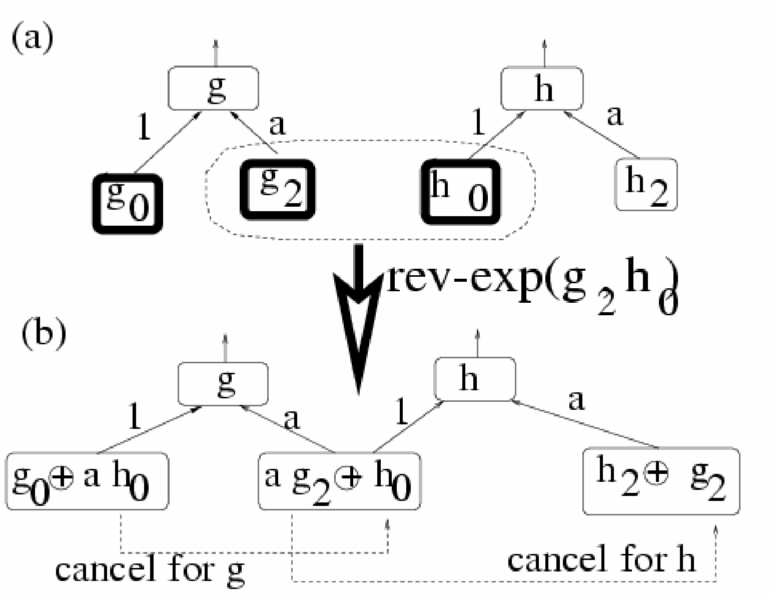


Figure 2.Creation of a Positive Davio level in a Regular Diagram: (a) two expanded node before reverse expansion, (b) layer of regular diagram after reverse expansion of node g2 and h0

(this expansion is based on EXOR-base)

Joining – about cofactor

- Every variable cuts a K-map into two disjoint parts.

- Thus ,arbitrary two functions f and g can always be expanded together to a Shannon Lattice with OR-ing as a join operation.
  - The same variable xi is used in the same level.
  - All expansions use negated literal in the left and positive literal of the variable in left

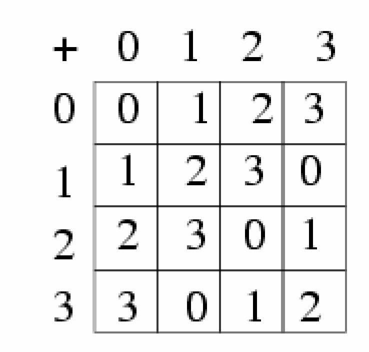
- This process can increase the number of nodes in comparison with a shared OBDD of these functions. But a regular structure is created, thus simplifying layout and making delay predictable.

- When fa1 and ga2 is not disjoint, new functions in levels are created by rearranging the cofactor in joinings. But when fa1 and ga2 is disjoint, there is no need to rearrange the f and g functions. In other words, fa1 and ga2 is OR-ing without changing f and g.

Regular Geometry – Galois Field

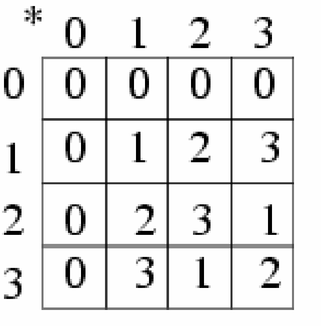
□ Galois Field

- Galois Field is based on Algebra of Finite Field that has finite elements.
- Digital logic is in GF(2) because all values of signal in logic are in two values 0,1.
- For Multi-valued logic, it is needed to have truth table of all operations that are used in logic completely.



+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Figure 5-1. GF(4) Add operation



*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Figure 5-2. GF(4) Multiplication operation

Regular Geometry

□ Symmetry

It can be shown in professor’s paper, every function that is not symmetric can be symmetrized by repeating variables in the lattice layers. So Non-symmetric function will not be cared.

□ In case of 4 Neighbors

- Input :North and East (GF(2))
- Output :South and West

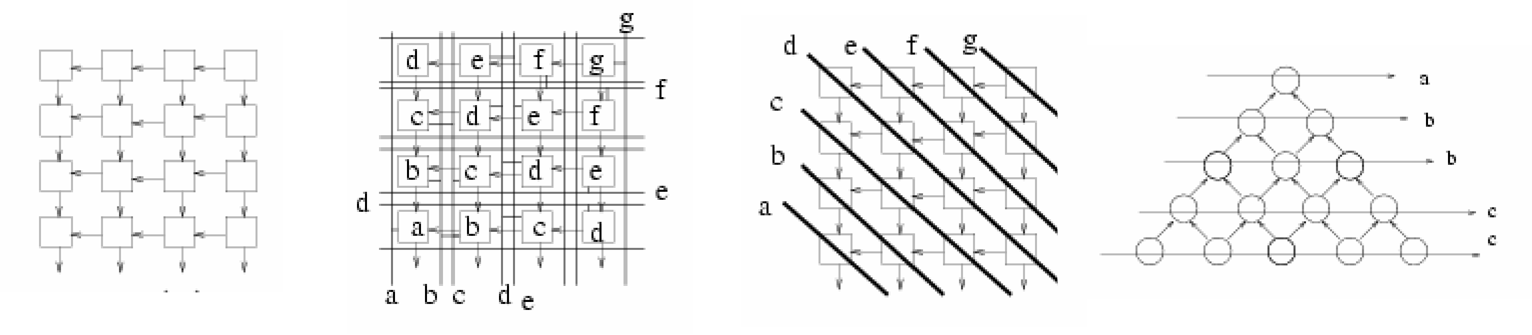
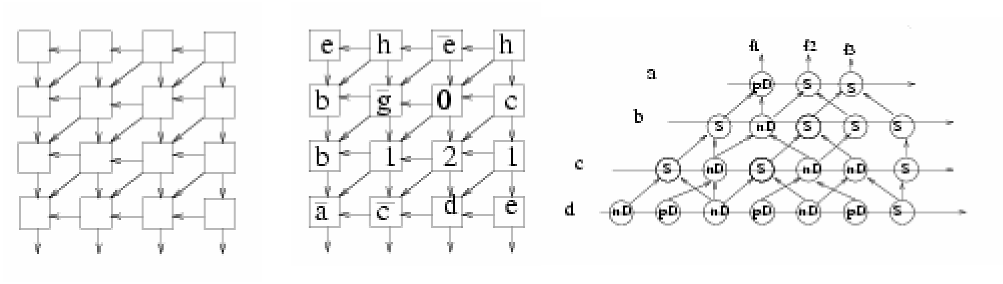


Figure . Regular Lattice for 4 Neighbors

Regular Geometry

- In case of 6 Neighbors
  - Inputs : (N,NE and E) GF(3)
  - Outputs : (W,SW and S)



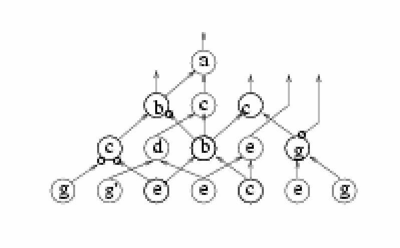


Figure . Regular Lattice for 6 Neighbors

- In case of 8 Neighbors
  - Inputs : (N,NE, NW and E) GF(4)
  - Outputs : (W,SW, SE and S)

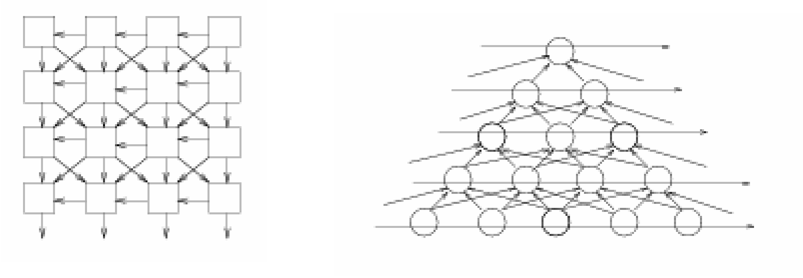


Figure . Regular Lattice for 8 Neighbors

Process of Lattice Creation

- EXAMPLE
  - 2X2 Regular LAYOUT Geometries
    - In case of 4 neighbors, 2X2 cells, the lattice is planar and it is based on a rectangular grid.
    - Cell has two inputs and two outputs.
    - The structure generalize the known Switch realization of symmetric binary functions, based on Shannon expansion
    - The same structure for Positive and Negative Davio expansions, negated variables and constants as control variables of the nodes.
    - Theorem Every non-symmetric function can be symmetrized by repeating variables.
  - 3X3 Regular LAYOUT Geometries
    - Three Inputs (N,NE,E) and three outputs (W,SW,S)
    - Generalized ternary diagrams for binary EXOR.
    - Arbitrary expansion-based Post Logic
    - GF(3)

Conclusion

- □ Starting from all possible neighbor geometries in two and three dimensional spaces, we create all possible regular structures.
- □ This extends previous planar geometries.
- □ Next we design arbitrary expansions for any of the structures.
- □ New expansion can be constructed based on the Linearly Independent function theory, or any other canonical or non-canonical function expansions.
- □ There exist a very high number of various new expansions.
- □ This LAYOUT-driven synthesis approaches are created for various function.
- □ This approach generalizes and unifies many known expansions, decision diagrams, and regular layout geometries.