XOR Gates in Emerging Technologies

Valentina Ciriani

UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI INFORMATICA

Outline

- XOR's Role in New Technologies
- XOR Based Logic Synthesis
  - Two Level Logic
    - ESOP
  - Bounded Multilevel Logic
    - SPP
    - ESPP
  - Unbounded Multilevel Logic
    - BBDD (Biconditional BDD)
    - XAIG
  - Secure Two Parties Computation
    - Example of Logic Synthesis Problem with Free XORs

Post-CMOS nanotechnologies

- Role of Logic Synthesis as

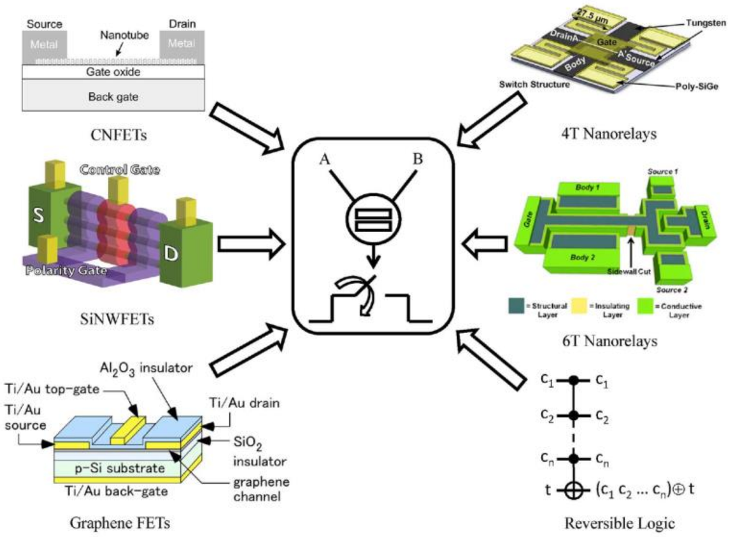
"enabler in the selection of post-CMOS technologies"

L.G. Amarù, P.-E. Gaillardon, S. Mitra, G. De Micheli. New Logic Synthesis as Nanotechnology Enabler. *Proceedings of the IEEE*, 2015

- Emerging nanothecnologies:
  - Graphene
  - Silicon nanowires
  - Carbon nanotubes
  - Organic FETs
  - Reversible logic
  - …
- New computational paradigms:
  - Lattices
  - Quantum computing
  - Adiabatic computation
  - …

XORs in new technologies

- CMOS technology:
  - NAND, NOR, INV (negative unate)
  - XORs are expensive gates
- New technologies:
  - Boolean comparator (XOR)
  - Majority voter
  - Lattices
  - …



[L.G. Amarù, P.-E. Gaillardon, S. Mitra, G. De Micheli. New Logic Synthesis as Nanotechnology Enabler. *Proceedings of the IEEE*, 2015]

ESOP

- An Exclusive-Sum-Of-Products (ESOP) is an Exclusive-OR of products of literals
- Example:

$$\bar{x}_1 x_2 \oplus x_2 x_3 \oplus x_4 \bar{x}_5 x_6$$

- Several heuristic synthesis methods have been proposed
  - EXORCISM
  - EXMIN

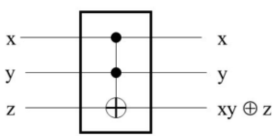
ESOP

- SOP (Sum of Product) covering:
  - each vector x such that f(x)=1 is covered by at least one product
  - each vector x such that f(x)=0 is not covered

- ESOP covering:
  - each vector x such that f(x)=1 is covered by an odd number of products
  - each vector x such that f(x)=0 is covered by an even number of products

ESOP for quantum computing

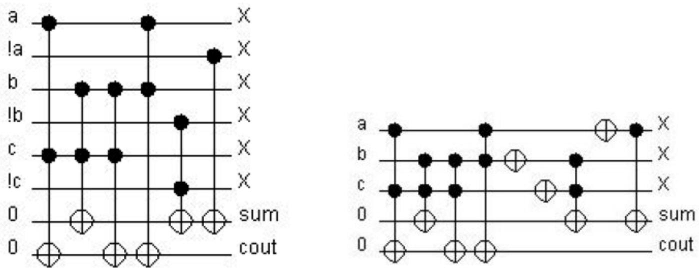
- ESOP covering:
  - Can be used as the starting expression to generate a cascade of reversible Toffoli gates



- Example (full adder)

```
1-1 01
-11 11
11- 01
-00 10
0-- 10
```

ESOP form



Cascate of Toffoli gates

[K. Fazel, M. A. Thornton, J. E. Rice, ESOP-based Toffoli Gate Cascade Generation, 2007]

SPP forms

- Sum of Pseudoproducts (SPP)

$$\underbrace{(x_1 \oplus x_2 \oplus x_3)\,(x_1 \oplus \overline{x_4})\,x_3}_{\text{Pseudoproduct}} + (x_1 \oplus x_2 \oplus x_3 \oplus \overline{x_4})\,\overline{x_5} + x_1$$

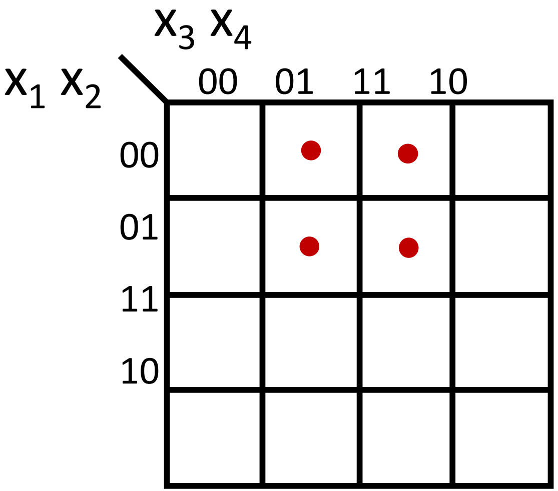
Pseudoproduct

(AND of XORs of literals)

- An SPP form is an OR of ANDs of XORs of literals
- The SPP problem: find an SPP form, covering a function F, with the minimum number of literals/pseudoproducts

Cubes

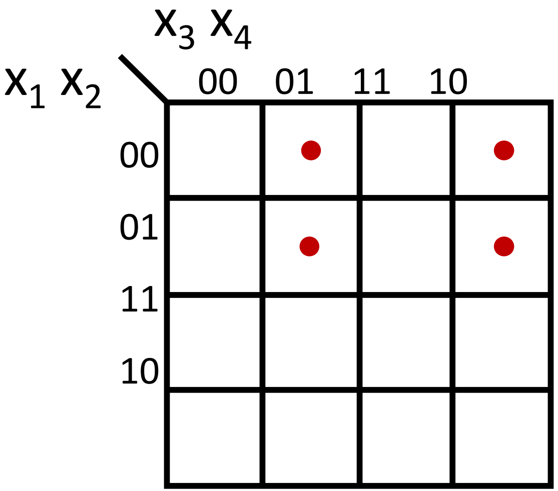
x_1	x_2	x_3	x_4
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1



Product: $\overline{x_1}\, x_4$

Pseudocubes

x_1	x_2	x_3	x_4
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0



pseudoproduct: $\overline{x_1}(x_3 \oplus x_4)$

Pseudocubes and Affine Spaces

- Theorem:

$$\text{Pseudocubes (SPP)} \Leftrightarrow \text{Affine Spaces}$$

- Corollary:

$$\text{Cubes (SOP)} \subseteq \text{Affine Spaces}$$

- SPPs are a direct generalization of SOP forms

SPP forms

Advantages

- Compact expressions
- Good testability of EXORs
- Three levels of logic

Disadvantages

- Unbounded fan-in EXORs
- Impractical for many technologies
- Huge minimization time

Solving the Disadvantages of SPP

2-SPP forms:

- are OR of ANDs of 2-EXORs of literals:

$$(x_1 \oplus x_2)\,(x_1 \oplus \overline{x_5})\,x_3 + (x_1 \oplus \overline{x_4})\,\overline{x_5} + x_1$$

- are still very compact
  - Only 4% more literals than SPP expressions
- have a reduced minimization time (heuristic)
  - 92% less time than SPP synthesis
- are practical
  - EXOR gates with fan-in 2 are typically easier to implement

2-SPP Minimization Problem

Problem: Find a sum of 2-pseudoproducts (2-SPP form) that is minimal w.r.t. the number of literals/products

- Exact minimization: Similar to Quine-McCluskey algorithm for SOPs
- Heuristic minimization: direct generalizations of classical two-level heuristic minimization
  - MERGE
  - EXPAND
  - EXOR-EXPAND
  - IRREDUNDANT
  - REDUCE

ESPP forms

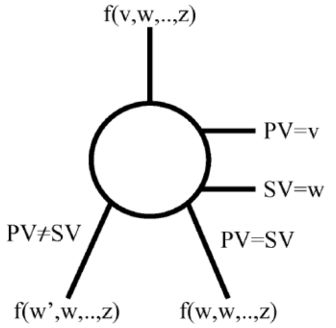
- Exclusive Sum of Pseudoproducts (ESPP) forms are :

$$\underbrace{(x_1 \oplus x_2 \oplus x_3)\,(x_1 \oplus \overline{x_4})\,x_3}_{\text{Pseudoproduct}} \oplus (x_1 \oplus x_2 \oplus x_3 \oplus \overline{x_4})\,\overline{x_5} \oplus x_1$$

- An ESPP form is a XOR of ANDs of XORs of literals

BBDD

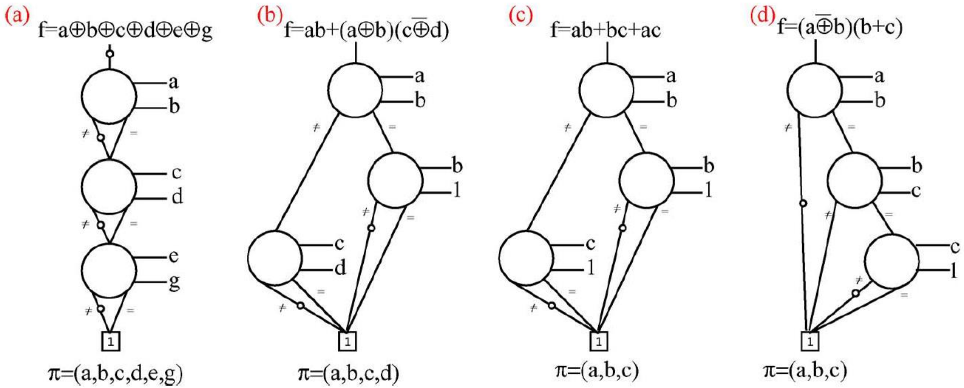
- Biconditional Binary Decision Diagrams:
  - $f(v,w,.., z) = (v \oplus w)\, f(w',w,.., z) + (\overline{v \oplus w})\, f(w,w,.., z)$



[Amarù et al. Proceedings of the IEEE, 2015]

- The reduction rules are a generalization of the ones for ROBDDs
- Under ordering and reduction rules, ROBBDDs are unique (canonical)
- Efficient manipulation of ROBBDDs, based on the biconditional expansion
- A ROBBDD can be transformed in a diagram of MUXs controlled by XNORs of variables

BBDD

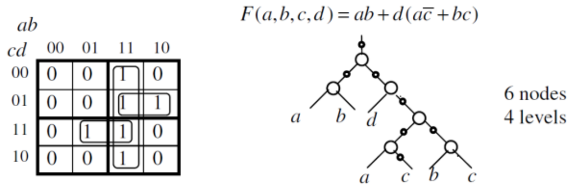


[L.G. Amarù, P.-E. Gaillardon, S. Mitra, G. De Micheli. New Logic Synthesis as Nanotechnology Enabler. *Proceedings of the IEEE*, 2015]

AND Inverter Graphs

- AIG is an acyclic combinational Boolean network composed of
  - 2-AND gates (internal nodes)
  - inverters (edges)

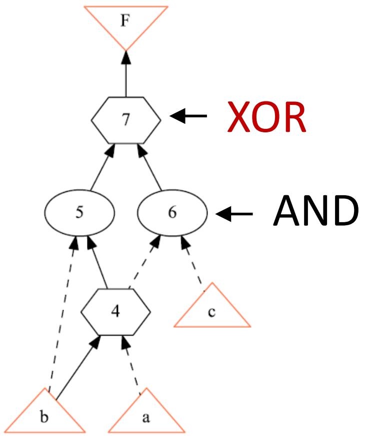
- Very used representation for Boolean functions
- ABC (R. Brayton, A. Mishchenko)



[R. Brayton, A. Mishchenko, ABC: An Academic Industrial-Strength Verification Tool, CAV 2010]

XAIG

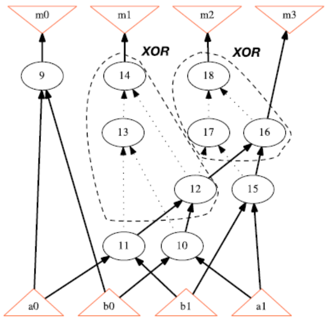
- XAIG is an acyclic combinational Boolean network composed of
  - 2-AND gates (internal nodes)
  - 2-XOR gates (internal nodes)
  - inverters (edges)



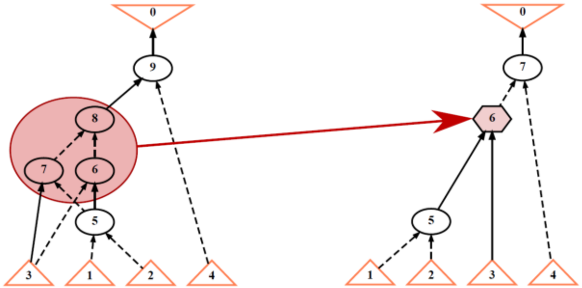
[I. Halecek, P. Fiser, J. Schmidt, Towards AND/XOR balanced synthesis: Logic circuits rewriting with XOR, Microelectronics Reliability, 2018]

XAIG: Algebraic rewriting

- Algebraic rewriting approaches:
  - $A \oplus B = !(A \wedge B) \wedge !(!A \wedge !B)$
  - $!(A \oplus B) = !(!A \wedge B) \wedge !(!A \wedge B)$



[C. Yu, M. Ciesielski, and A. Mishchenko Fast Algebraic Rewriting Based on And-Inverter Graphs, IEEE TCAD 2018]



[I. Halecek, P. Fiser, J. Schmidt, Towards AND/XOR balanced synthesis: Logic circuits rewriting with XOR, Microelectronics Reliability, 2018]

XAIG: Boolean approach

- Let G be a two-input gate, a similar gate to G is a two-input gate G_s (e.g., EXOR):

$$G(x, y) = G_s (x, y) \text{ for all } (x, y) \text{ but one}$$

- A gate G in a circuit C is swappable into a gate G_s, if
  - G is similar to G_s (the different input is (x,y))
  - the input configuration (x,y) never occurs as an input to G in C

A	B	**A⊕B**	!(A∧B)	A∧!B	!A∧B	!(!A∧!B)
0	0	**0**	1	0	0	0
0	1	**1**	1	0	1	1
1	0	**1**	1	1	0	1
1	1	**0**	0	0	1	1

XAIG: Boolean approach

- Let G be a two-input gate, a similar gate to G is a two-input gate G_s (e.g., EXOR):

  $$G(x, y) = G_s(x, y) \text{ for all } (x, y) \text{ but one}$$

- A gate G in a circuit C is swappable into a gate G_s, if
  - G is similar to G_s (the different input is (x,y))
  - the input configuration (x,y) never occurs as an input to G in C

A	B	**A⊕B**	!(A∧B)	A∧!B	!A∧B	!(!A∧!B)
0	0	**0**	-	0	0	0
0	1	**1**	1	-	1	1
1	0	**1**	1	1	-	1
1	1	**0**	0	0	1	-

Satisfiability don't cares (SDCs)

Boolean test (BDDs)

XAIG: future direction

- Proposed methods (algebraic and Boolean) are:
  - Rewriting techniques
  - Postprocessing algorithms on a given AIG
- Future direction:
  - Direct minimization method for XAIG

Secure two-party computation

Secure two-party computation protocols:

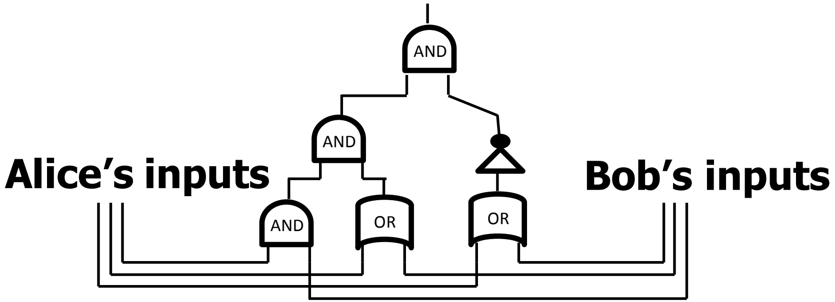
- allow two parties to compute any function *F* on their respective inputs
- while maintaining the privacy of their input values:
  - such that only the resulting output is shared among the parties
  - and nothing is known about the other party's input

Example: Millionaire

- Alice and Bob
  - are millionaires
  - wish to determine who has more money
  - don’t wish to reveal her or his precise wealth to the other
- Inputs:
  - Alice $ 2,000,003
  - Bob $ 2,000,002
- Output: Alice

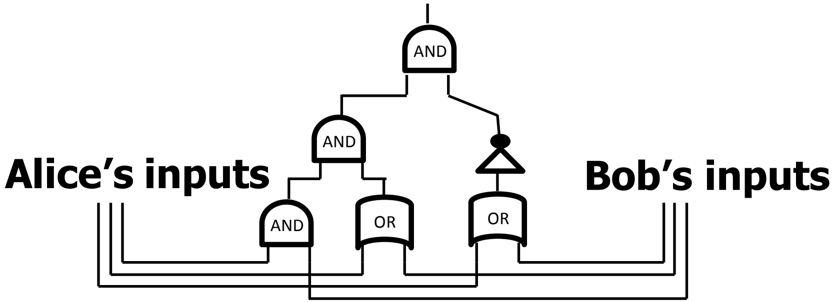
Yao's protocol

- Convert the function into a Boolean circuit



Yao's protocol

- Bob ad Alice cooperate in defining the solution by:
  - Exchanging limited information (communication protocol)
  - Computing the output of each gate



EXOR-free protocol

- Kolesnikov and Schneider (2008) show that 2-input EXOR gates can be computed for "free":
  - 2-EXORs evaluated without the communication protocol
  - Bob computes the result by simply performing the 2-EXOR of the encrypted input values

- Problem: find a Circuit with a minimum number of non-XOR gates
  - ESOP forms
  - XAIG
  - Multivalued circuits

Conclusion

- Emerging technologies need new logic synthesis methods defined on new models of logic devices
- Since several technologies rely on comparators:
  - XOR gates should be taken into consideration in the new logic synthesis methods
- New nice problems to solve!

Thanks!