

# Biconditional BDD: A Novel Canonical BDD Enabling Efficient Direct Mapping of DG Controllable Polarity FETs

Luca Amarú, Pierre-Emmanuel Gaillardon, Giovanni De Micheli

Integrated Systems Laboratory (LSI), EPFL, Switzerland

Monday, March 25th, 2013



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

---

# Outline

- Introduction and motivation
- Novel representation form: Biconditional BDDs
- Direct mapping of BBDDs onto DG controllable polarity FETs
- Experimental results
- Conclusions

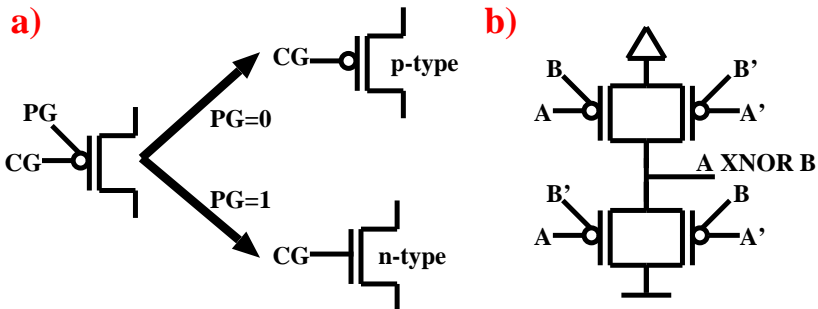
---

# Current Section

- Introduction and motivation
- Novel representation form: Biconditional BDDs
- Direct mapping of BBDDs onto DG controllable polarity FETs
- Experimental results
- Conclusions

## Functionality-Enhanced Devices $\Rightarrow$ Opportunities

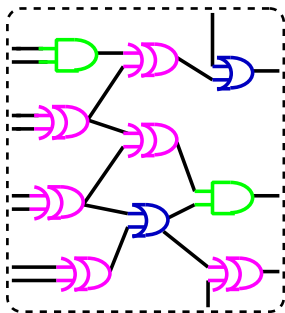
- Case study: Double-Gate **controllable polarity** FETs (e.g., SiNWFETs, CNTFETs etc.).



- **XOR/XNOR**-based logic is remarkably compact.
- Thanks to the **biconditional** (XNOR) connective embedded in the device operation: on state  $(PG=0 \cdot CG=0) + (PG=1 \cdot CG=1)$ .

# Exploit Controllable Polarity @ Logic Synthesis

- Controllable Polarity  $\Rightarrow$  **biconditional** (XNOR) connective.
- Highlight all the advantageous **XOR/XNOR** operations



- We need XOR-oriented representation form.
- We propose **Biconditional Binary Decision Diagrams** natively supporting the biconditional connective.

---

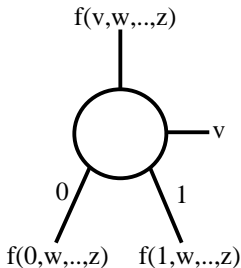
## Current Section

- Introduction and motivation
- Novel representation form: Biconditional BDDs
- Direct mapping of BBDDs onto DG controllable polarity FETs
- Experimental results
- Conclusions

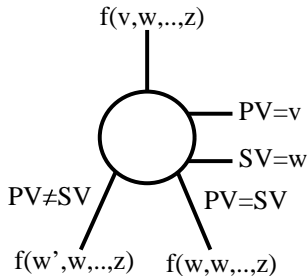
# Biconditional Binary Decision Diagrams (BBDDs)

- Biconditional BDDs: BDDs driven by a novel logic expansion.
- **Traditional BDDs**, Shannon's expansion:
- **Novel BBDDs**, biconditional expansion:

$$f(v, w, \dots, z) = v \cdot f(1, w, \dots, z) + v' \cdot f(0, w, \dots, z)$$

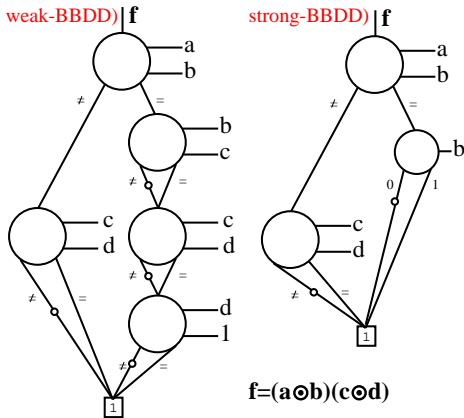


$$f(v, w, \dots, z) = (v \oplus w) \cdot f(w', w, \dots, z) + (v \odot w) \cdot f(w, w, \dots, z)$$



# BBDD Ordering and Reduction

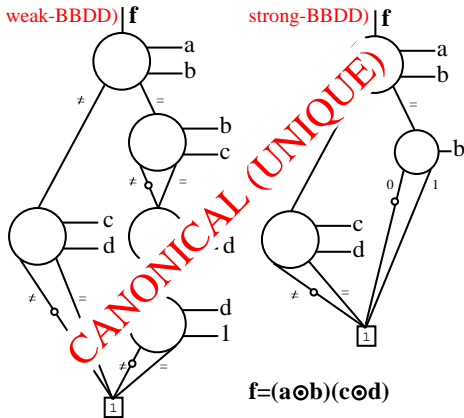
- Ordering: assign PV and SV by layers. Order  $\pi = \{a, b, c\} \Rightarrow PV, SV$  assignment  $\{PV_0, SV_0, PV_1, SV_1, PV_2, SV_2\} = \{a, b, b, c, c, 1\}$ .
- Reduction:
  - Traditional BDD reduction rules  $\Rightarrow$  **weak-reduction** of BBDDs.
  - Additional reduction rules  $\Rightarrow$  **strong-reduction** of BBDDs.



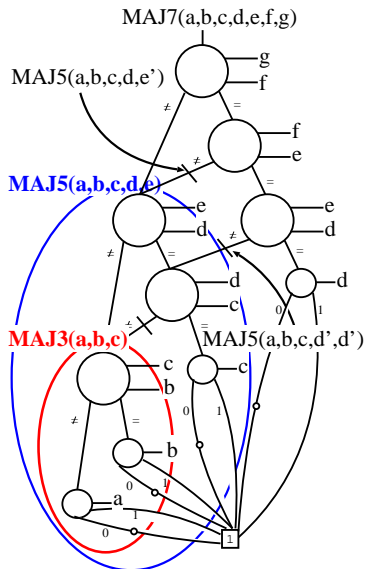


# BBDD Ordering and Reduction

- Ordering: assign PV and SV by layers. Order  $\pi = \{a, b, c\} \Rightarrow PV, SV$  assignment  $\{PV_0, SV_0, PV_1, SV_1, PV_2, SV_2\} = \{a, b, b, c, c, 1\}$ .
- Reduction:
  - Traditional BDD reduction rules  $\Rightarrow$  **weak-reduction** of BBDDs.
  - Additional reduction rules  $\Rightarrow$  **strong-reduction** of BBDDs.

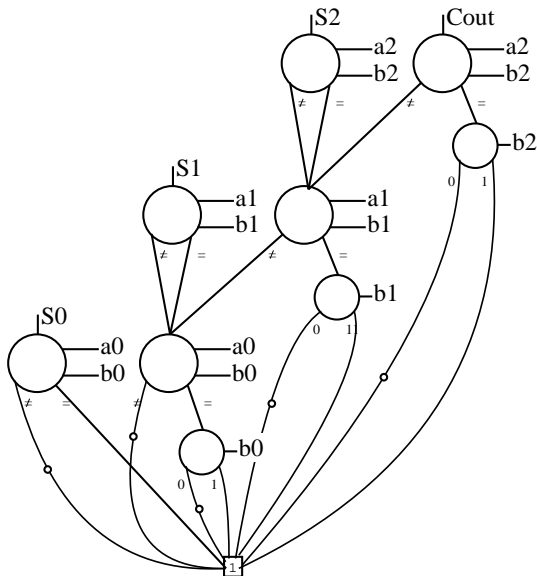


# BBDD for Majority Function



- Number of nodes for  $MAJ(n) = \frac{1}{8}n^2 + \frac{1}{2}n + \frac{11}{8}$
- $MAJ(3) = 4$
- $MAJ(5) = 7$
- $MAJ(7) = 11$
- ...
- Traditional BDDs:  
 $MAJ(n) = \lceil 0.5n \rceil (n - \lceil 0.5n \rceil + 1) + 1$

# BBDD for Adder Function



- Two binary numbers  $a, b$
- $ADD = a + b$ ,  
 $\text{length}(a) = \text{length}(b) = n$
- Number of nodes for  
 $ADD(n) = 3n + 1$
- Traditional BDDs:  
 $ADD(n) = 5n + 2$

## BBDD Representation Compactness Results

- BBDDs are canonical  $\Rightarrow$  enable efficient manipulation algorithms
- Construct BBDDs for general **MCNC benchmarks**
- Extended APPLY operator (typical bottom-up construction)
- Average results:
  - **BBDDs are 37% smaller** than traditional BDDs
- Note: **XOR-rich benchmarks** take **large advantage** of BBDDs

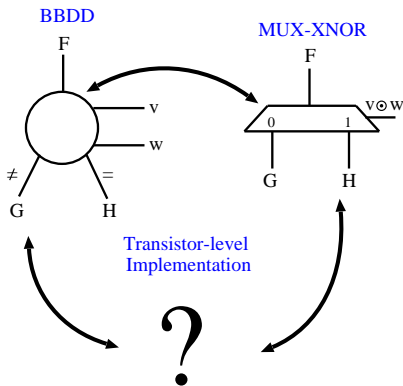
---

## Current Section

- Introduction and motivation
- Novel representation form: Biconditional BDDs
- Direct mapping of BBDDs onto DG controllable polarity FETs
- Experimental results
- Conclusions

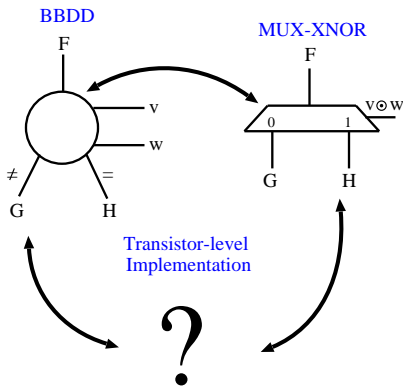
## Direct Mapping BBDDs onto (General) Logic

- BBDDs are remarkably compact
- We want to preserve such compactness in the circuit implementation
- Assign logic blocks to BBDDs constituents



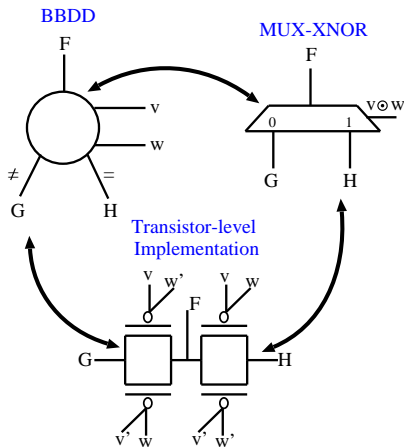
# BBDDs $\Leftrightarrow$ DG Controllable Polarity FETs

- BBDDs, DG Controllable Polarity FETs  $\Rightarrow$  driven by XNOR
- Share the same functionality at the logic level
- BBDD nodes *efficient* implementation



# BBDDs $\Rightarrow$ DG Controllable Polarity FETs

- BBDDs, DG Controllable Polarity FETs  $\Rightarrow$  driven by XNOR
- Share the same functionality at the logic level
- BBDD nodes *efficient* implementation





---

## Current Section

- Introduction and motivation
- Novel representation form: Biconditional BDDs
- Direct mapping of BBDDs onto DG controllable polarity FETs
- Experimental results
- Conclusions

# BBDD-based Direct Mapping Experiments

- **Reference** synthesis flow: commercial Synopsys **Design Compiler**.
- Logic circuit benchmarks: MCNC suite (medium/small circuits).
- Average results:
  - **Device count: -49.7%** w.r.t. Design Compiler.
  - **Logic depth: ~7x reduction** w.r.t. Design Compiler.
  - **Runtime: ~2x reduction** w.r.t. Design Compiler.

# Opportunities for Place&Route

- P&R is key at advanced technology nodes.
- BBDD-based direct mapping  $\Rightarrow$  simplifies P&R
- Why?
  - Edges (signal wires) connect only adjacent nodes (cells)
  - Branching variables (control wires) are local
  - Consecutive SV/PV are assigned to the same signal
- Together with *Sea of Tiles* (SoT) approach, it is possible to alleviate the interconnection issue in circuits based on DG devices.

---

## Current Section

- Introduction and motivation
- Novel representation form: Biconditional BDDs
- Direct mapping of BBDDs onto DG controllable polarity FETs
- Experimental results
- **Conclusions**

---

## Conclusions

- DG controllable polarity FETs  $\Rightarrow$  new logic synthesis opportunities
- We want an efficient logic representation form to harness controllable polarity functionality (XNOR)
- Biconditional (XNOR) Binary Decision Diagrams
- Direct mapping of BBDDs onto DG controllable polarity FETs natively preserves (and exploit) controllable polarity functionality

---

## Questions?

Thank you for your attention.

Come at my poster for more details!