VLSI Signal Processing Group
Tay-Jyi Lin tjlin@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

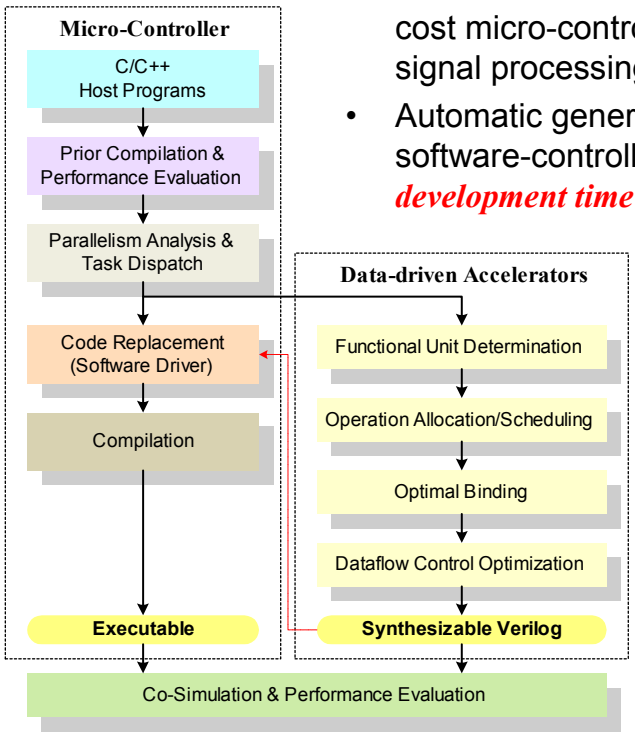
Formal Equivalence Checking of Folded Architectures

Tay-Jyi Lin *and Chein-Wei Jen*

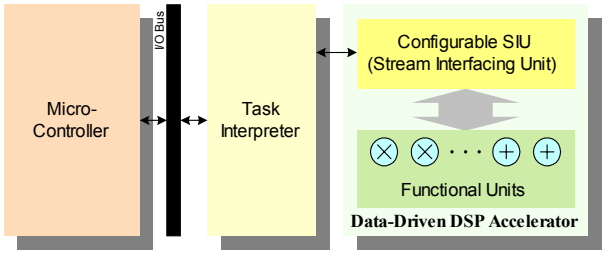
VLSI Signal Processing Group
Department of Electronics Engineering
National Chiao Tung University, Taiwan
{tjlin,cwjen}@ee.nctu.edu.tw

VLSI Signal Processing Group
Tay-Jyi Lin tjlin@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

CASCADE (Configurable And SCAlable Dsp Environment)



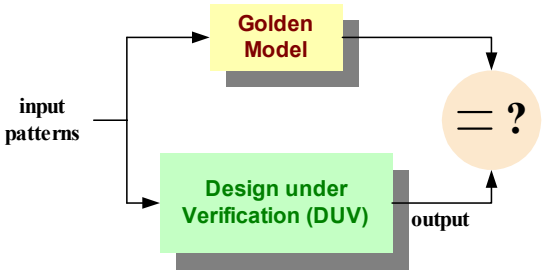
- The data-driven accelerators boost the performance of low-cost micro-controllers or simple DSP processors for video signal processing to ***lengthen their product life span***.
- Automatic generation of the accelerators with simple software-controlled interfacing dramatically ***reduces the development time***.



- This paper describes the ***formal verification*** on the auto-generated data-driven accelerators.

Equivalence Checking (EC)

- Simulation-based EC
  - equivalence is only guaranteed to some extent that the test suite exercises the design



- Formal EC
  - verification by formal (mathematical) methods

(1) combinational EC

check the identity of the corresponding ***canonical representations***

e.g. ROBDD (Reduced Ordered Binary Decision Diagram) and BMD (Binary Moment Diagram), etc

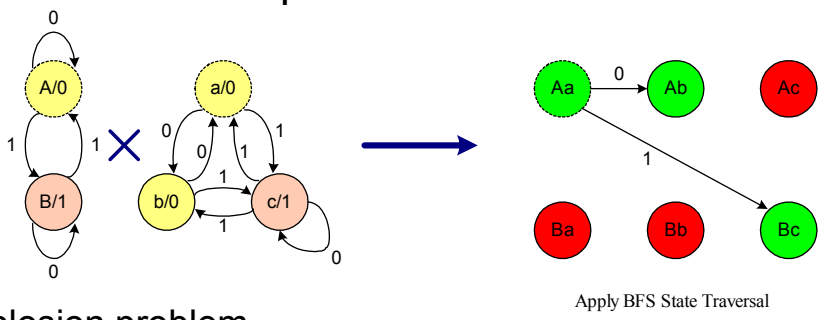
(2) sequential EC

state traversal on the ***product machine***

VLSI Signal Processing Group
Tay-Jyi Lin tjlin@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Sequential EC

- State traversal on the product machine

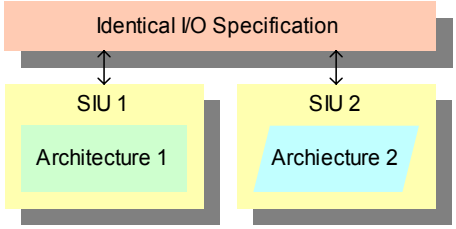


(1) State explosion problem

- ***symbolic state traversal*** with BDD-represented transition functions

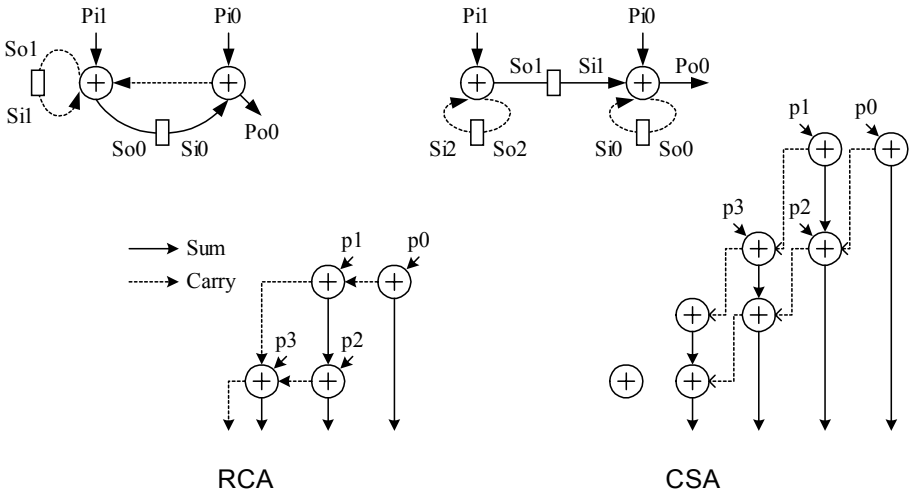
(2) The golden model must have ***identical synchronous behaviors*** with DUV

- SIU encapsulation, but verification on the SIU itself is another problem
- combinational EC



Sequential EC with Combinational Algorithms

- For the folded architectures with ***acyclic DFG*** -represented computations, all registers can be removed by ***unfolding transforms*** with proper ***retiming***. More efficient combinational EC algorithms can thus be applied.

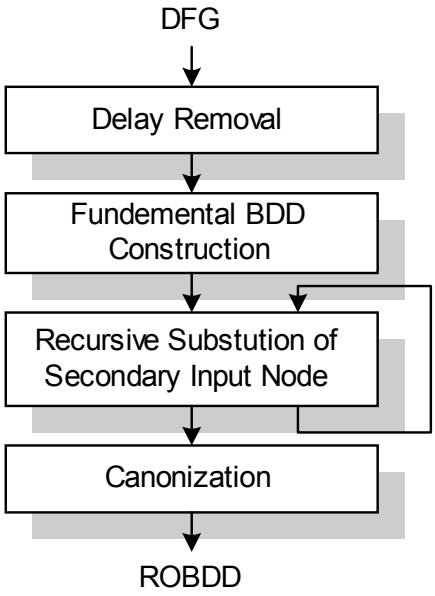


- For the folded architectures with feedback, ***robust register mapping*** is required for the loop synchronization delay elements (registers).

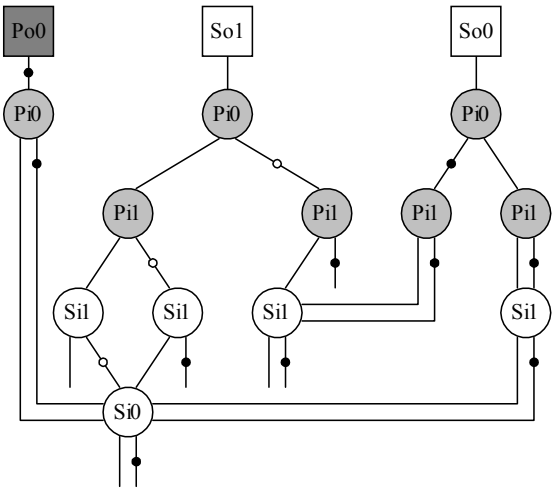
VLSI Signal Processing Group
Tay-Jyi Lin tjlin@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Iterative ROBDD Construction

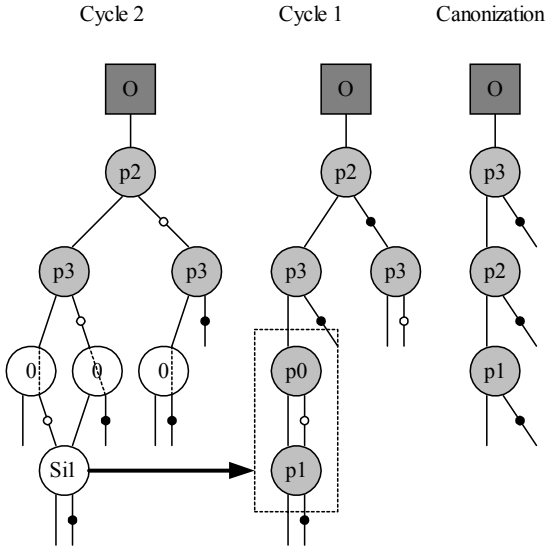
- First, we build the canonical representation (here, ROBDD as an example) for the combinational part in the folded architectures, which performs an identical function for every cycle.
- Our proposed algorithm constructs the ROBDD of the folded architectures with ***implicit*** unfolding & retiming transforms.
- The identical testing on partially-built ROBDD is possible with our proposed scheme.



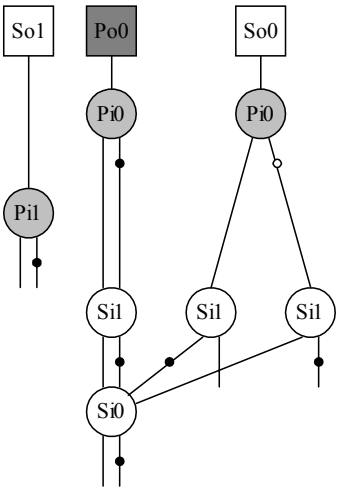
Example (1/2) - RCA-based Serial Multiplier



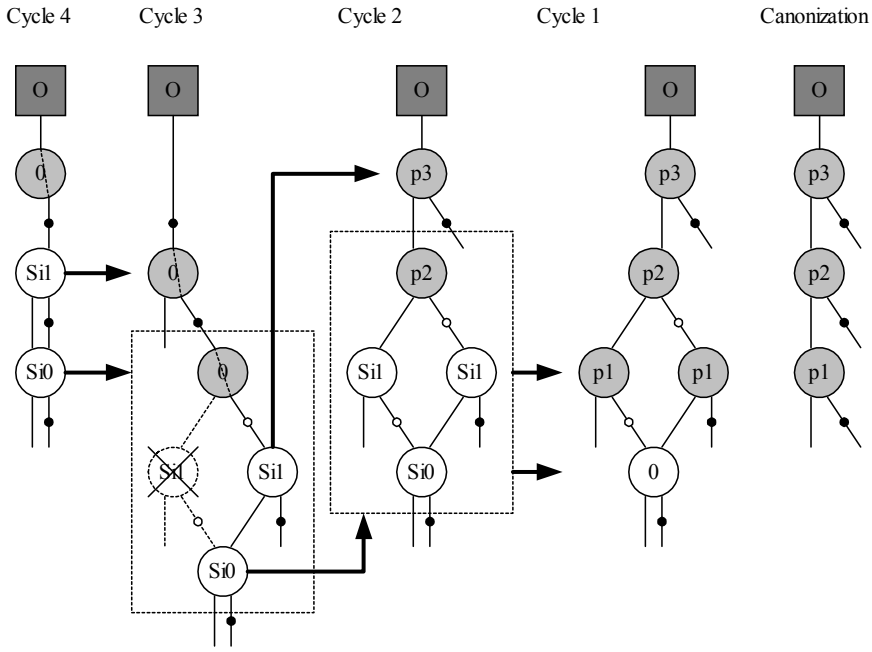
Fundamental BDD



Example (2/2) - CSA-based Serial Multiplier



Fundamental BDD



Department of Electronics Engineering, National Chiao Tung University
VLSI Signal Processing Group
Tay-Jyi Lin tjlin@ee.nctu.edu.tw

Conclusion

- Combinational EC algorithms can solve more complex and even unsolved (with distinct I/O sequencing or timing, such as the folded architectures) sequential EC problems by the ***unfolding transform*** with proper ***retiming*** to completely remove the pipeline registers and ***robust register mapping*** for loop synchronization delay elements.
- Our proposed unified algorithm constructs the ROBDD direct from the folded architecture with ***implicit*** unfolding and retiming.
- The complex ROBDD construction is only required for the combinational part of the folded architecture, which is much smaller than the flattened unfolded representation. So, the proposed algorithm is ***computation-effective***.
- Identity testing on partial-built ROBDD is possible with our proposed iterative ROBDD construction scheme. Thus, the proposed algorithm is ***memory-effective***.