A Binding Algorithm in High-level Synthesis for Path Delay Testability

Yuki Yoshikawa

Kure National College of Technology

2013.1.24

Outline

- Background
- Related work
    - ✓ High level synthesis for delay testability [Sying’07]
- Research objective
    - ✓ Two pattern testable paths
- The proposed binding method
- Experimental results
- Conclusion

Background

High quality delay testing is an important issue for recent high performance and high speed LSIs

Approaches to delay testing

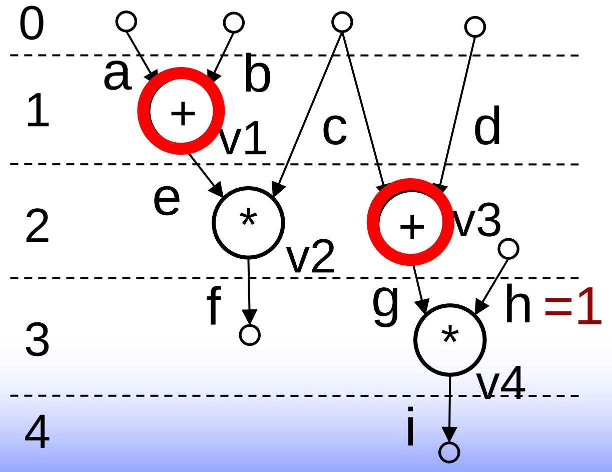
- Design for testability (DFT)
  - Scan design with LoS or LoC test application [Savir'93,94]
  - Non scan design with hierarchical test generation [Amin'02]
- High level synthesis for testability (SFT)
  - Resource binding for applying two-pattern tests to operational modules [Sying'09]

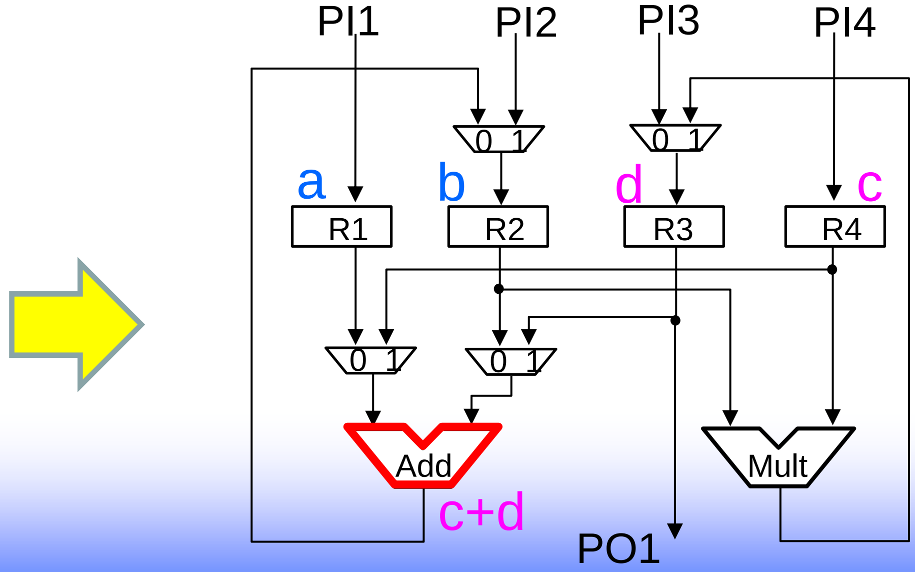
A resource binding which approaches to path delay faults

Related work

A resource binding method to test delay faults in operational modules was proposed [Sying’09]

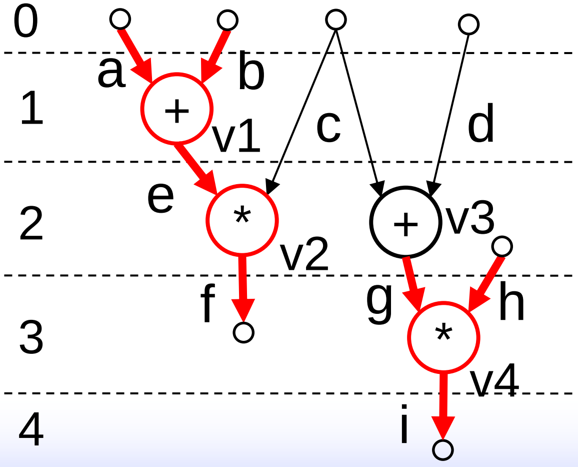
- Two consecutive operations are bind to one operational module
- The inputs of the operations are controllable and its second output is observable

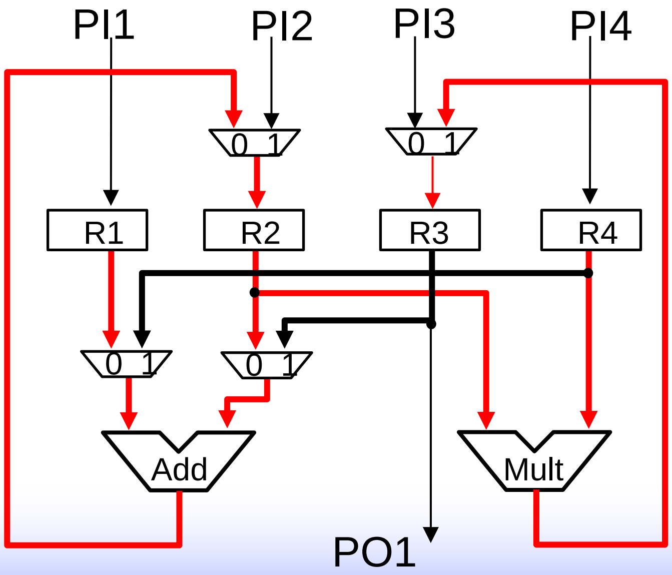




Research objective

We propose a binding method that makes all the single cycle paths in the synthesized datapath testable

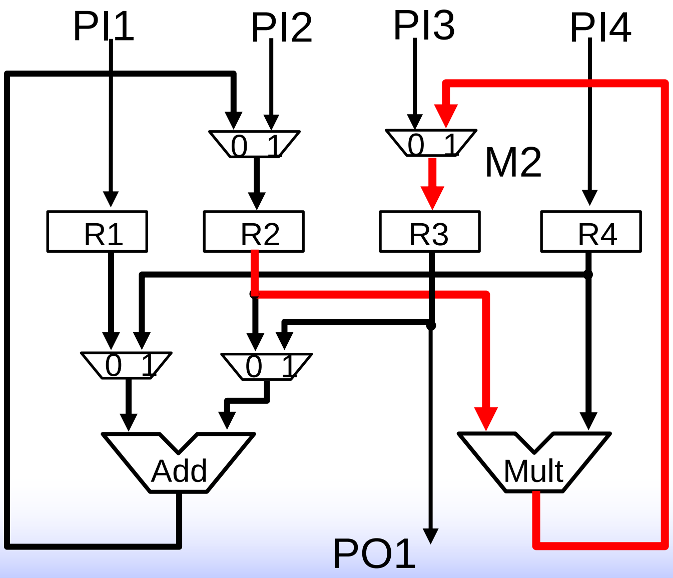




Paths at RTL

Structural path (S-path)

- A path from a register to a register passing through only combinational modules
- For example, path R2 – Mult – Mux2 – R3

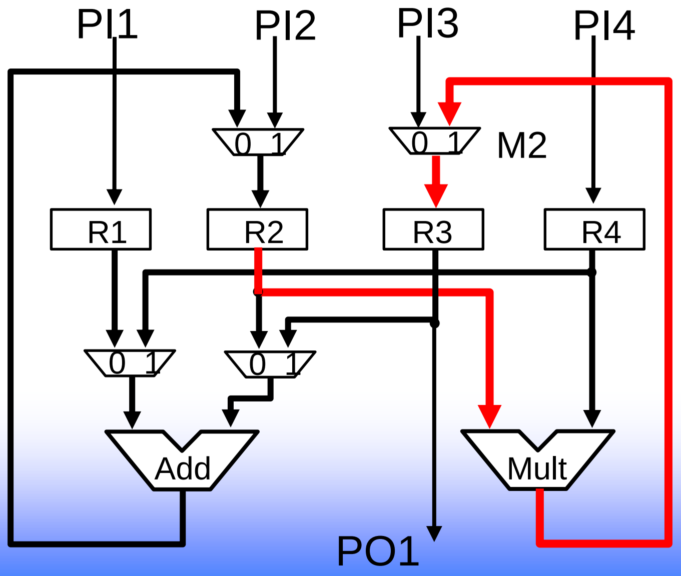


Two pattern testable S-path

A control sequence for a pair of two-pattern tests

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V2q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

- The first vector V1=(V1p, V1q)
- The second vector V2=(V2p, V2q)



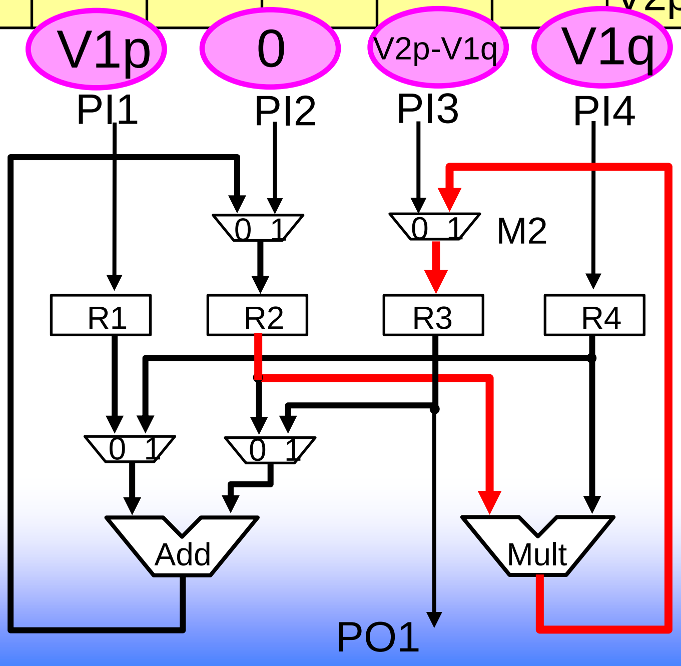
Two pattern testable S-path

A control sequence for a pair of two-pattern tests

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V2q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

Step T0

- V1p, 0, V2p-V1q and V1q are set to PI1, PI2, PI3 and PI4, respectively



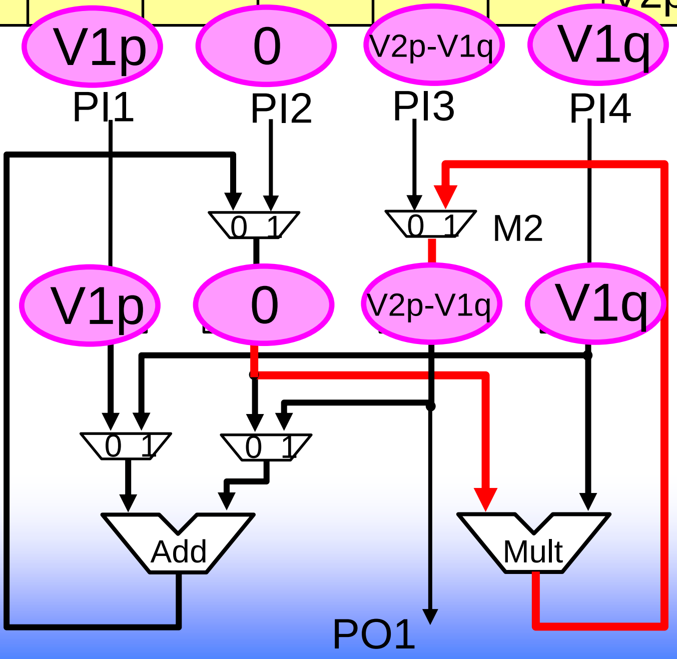
Two pattern testable S-path

A control sequence for a pair of two-pattern tests

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V2q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

Step T1

- R1, R2, R3 and R4 capture V1p, 0, V2p-V1q and V1q, respectively



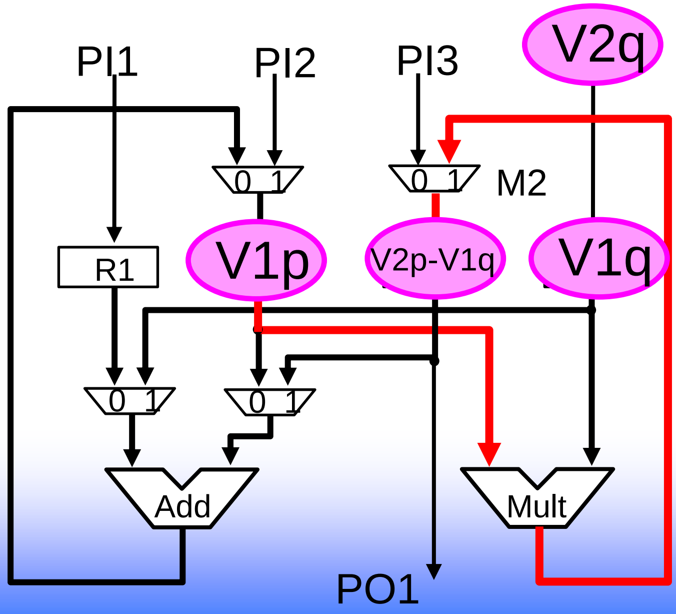
Two pattern testable S-path

A control sequence for a pair of two-pattern tests

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V2q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

Step T2

- R2 captures V1p by calculating V1p + 0
- R3 and R4 keep their values by using the hold function
- Then, V1p and V1q become the first test pattern



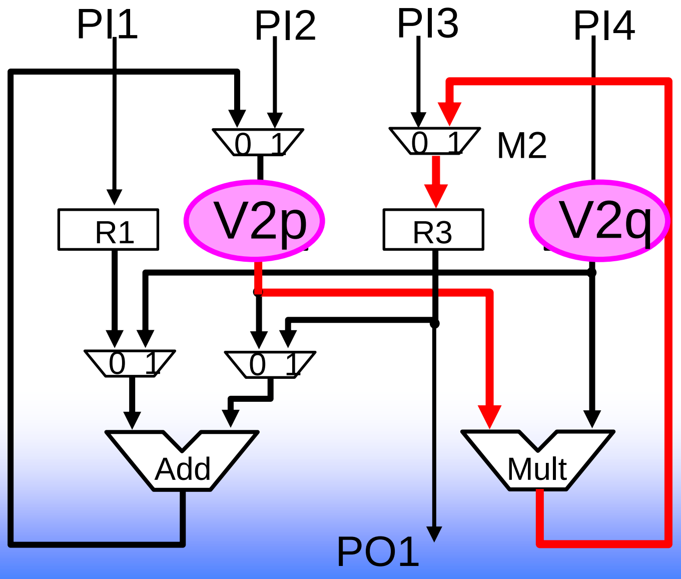
Two pattern testable S-path

A control sequence for a pair of two-pattern tests

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V1q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

Step T3

- R2 captures V2p by calculating (V2p-V1q) + V1q
- R4 captures V2q from PI4
- Then, V2p and V2q become the second test pattern



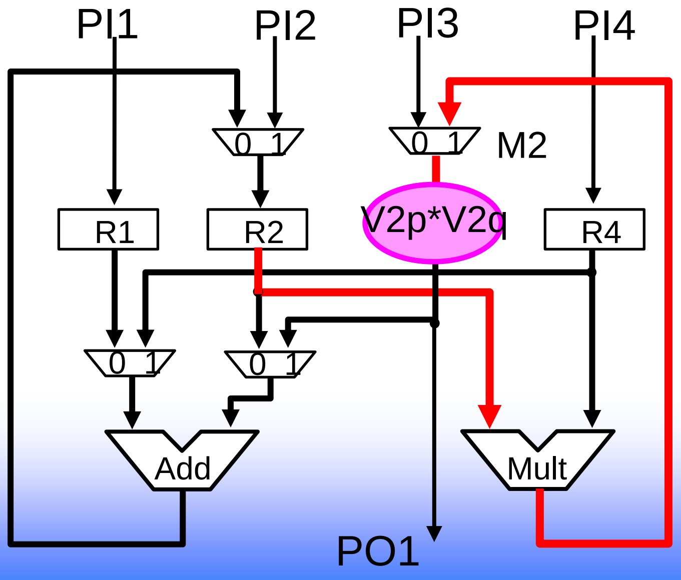
Two pattern testable S-path

A control sequence for a pair of two-pattern tests

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V2q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

Step T4

- R3 captures the output V2p*V2q of the multiplier



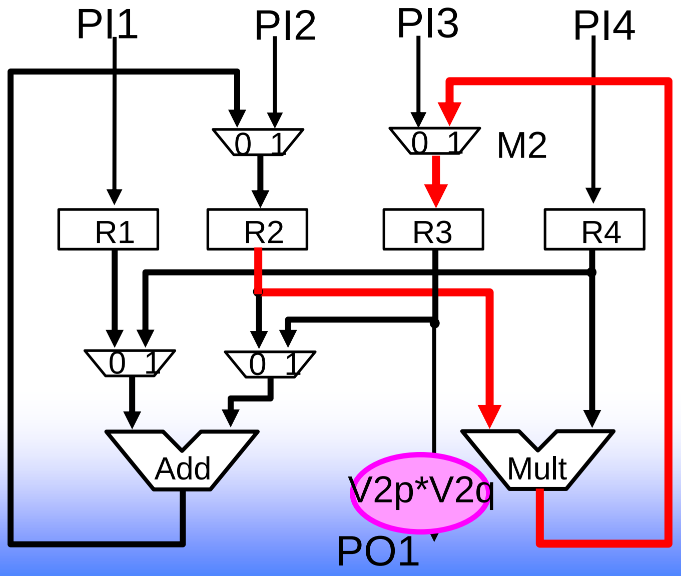
Two pattern testable S-path

A control sequence for a pair of two-pattern tests

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V2q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

Step T5

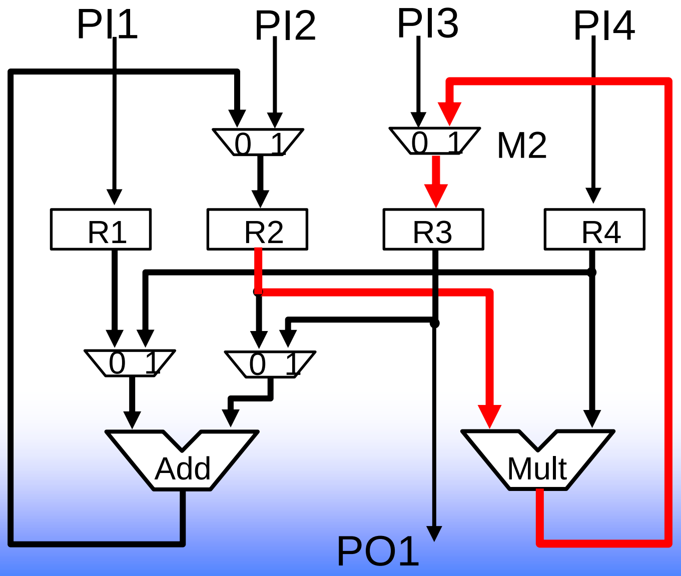
- Finally, the output is observed at PO1



Path delay test environment (PDTE)

	PI1	PI2	PI3	PI4	Reg1	Reg2	Reg3	Reg4	Mux1	Mux2	Mux3	Mux4	PO
T0	V1p	0	V2p-V1q	V1q									
T1					V1p	0	V2p-V1q	V1q	1	0			
T2				V2q		V1p	V2p-V2q	V1q	0		0	0	
T3						V2p		V2q	0	1	1	1	
T4							V2p*V2q			1			
T5													V2p*V2q

This control sequence is called path delay test environment (PDTE)



Synthesis of an s-path which has a PDTE

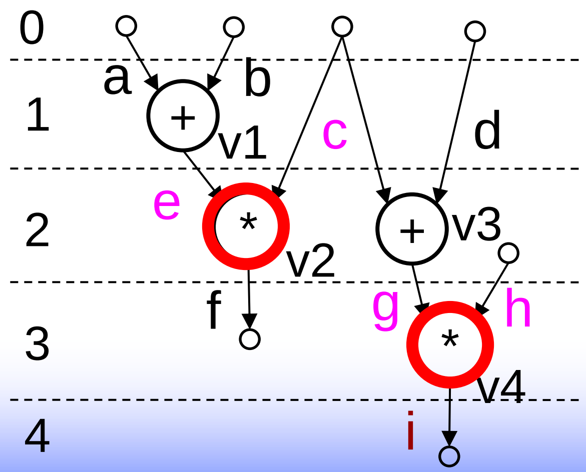
The key conditions

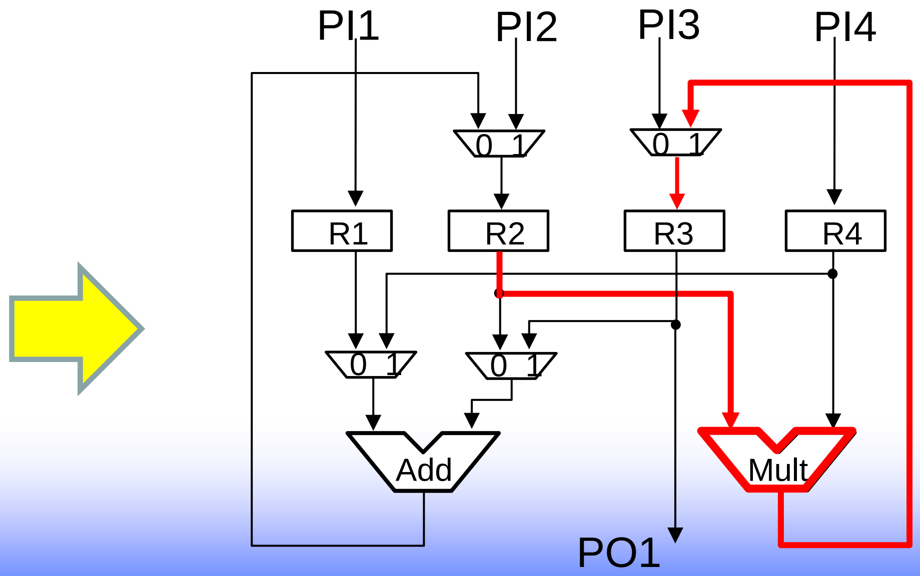
- Two consecutive operations are bind to one operational module
- The inputs of the operations are controllable and its second output is observable
- The corresponding inputs between two operations are bind to one input register, respectively
- The outputs of the two operations are bind to one output register

Synthesis of an s-path which has a PDTE

The key conditions

- Two consecutive operations are bind to one operational module
- The inputs of the operations are controllable and its second output is observable

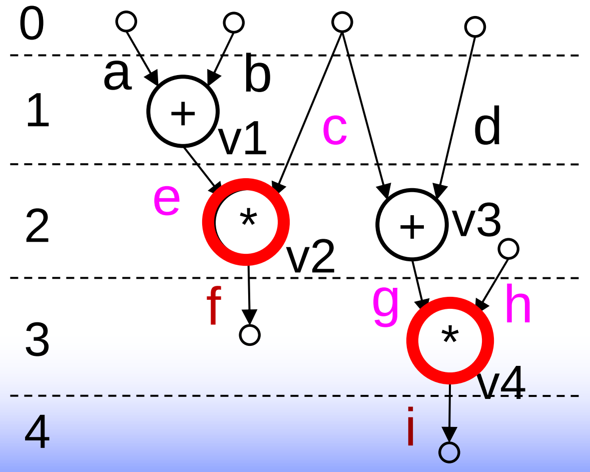


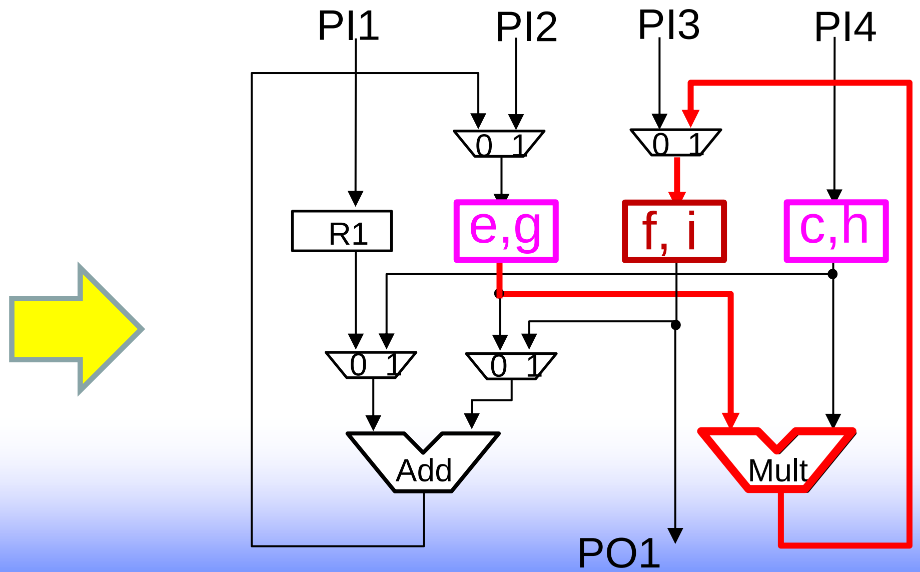


Synthesis of an s-path which has a PDTE

The key conditions

- The corresponding inputs between two operations are bind to one input register, respectively
- The outputs of the two operations are bind to one output register





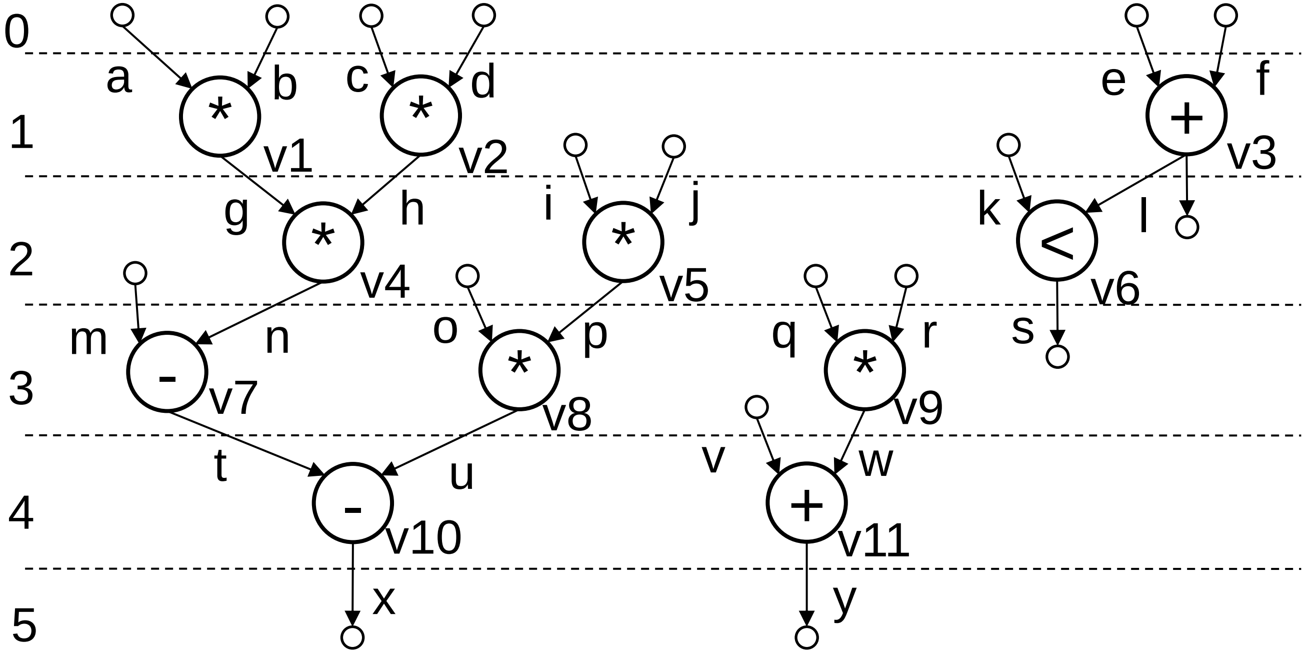
Resource binding algorithm

Step 1: Resource binding

1. Maximization of the number of two-pattern testable paths (First priority)
2. Minimization of resources

Step 2: Addition of test modes

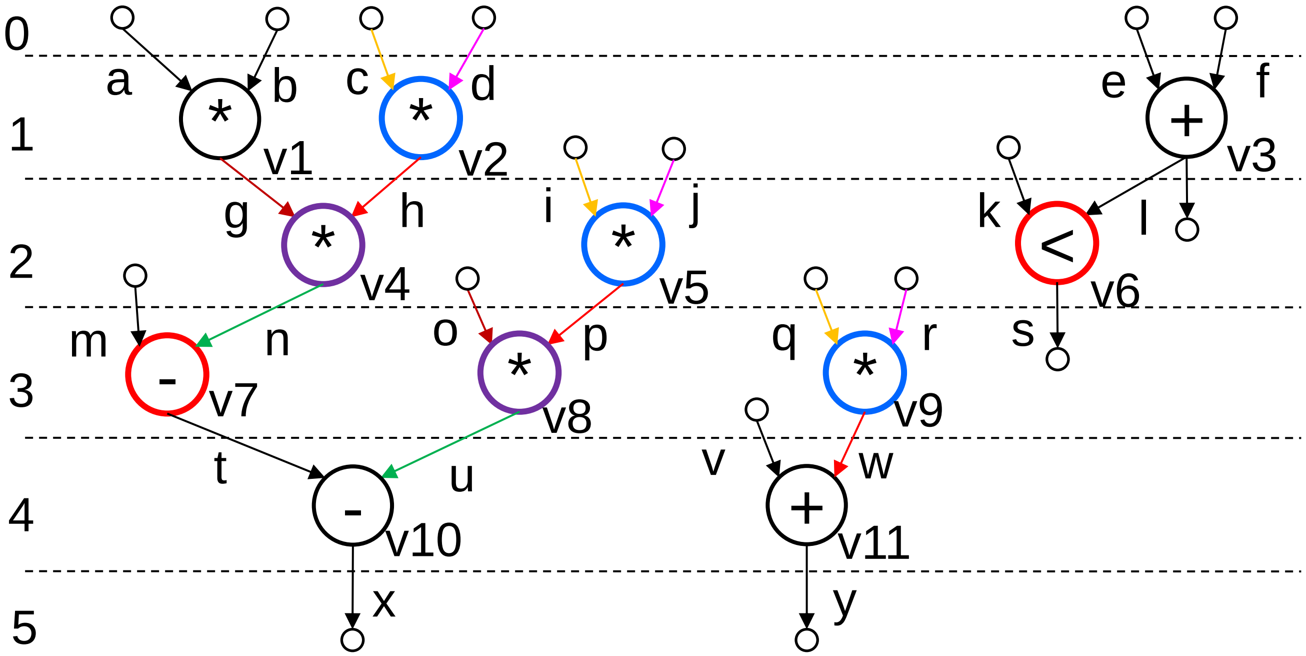
Maximization of TP testable paths



An operation which has a few testable pair is preferentially shared because it has a few candidates

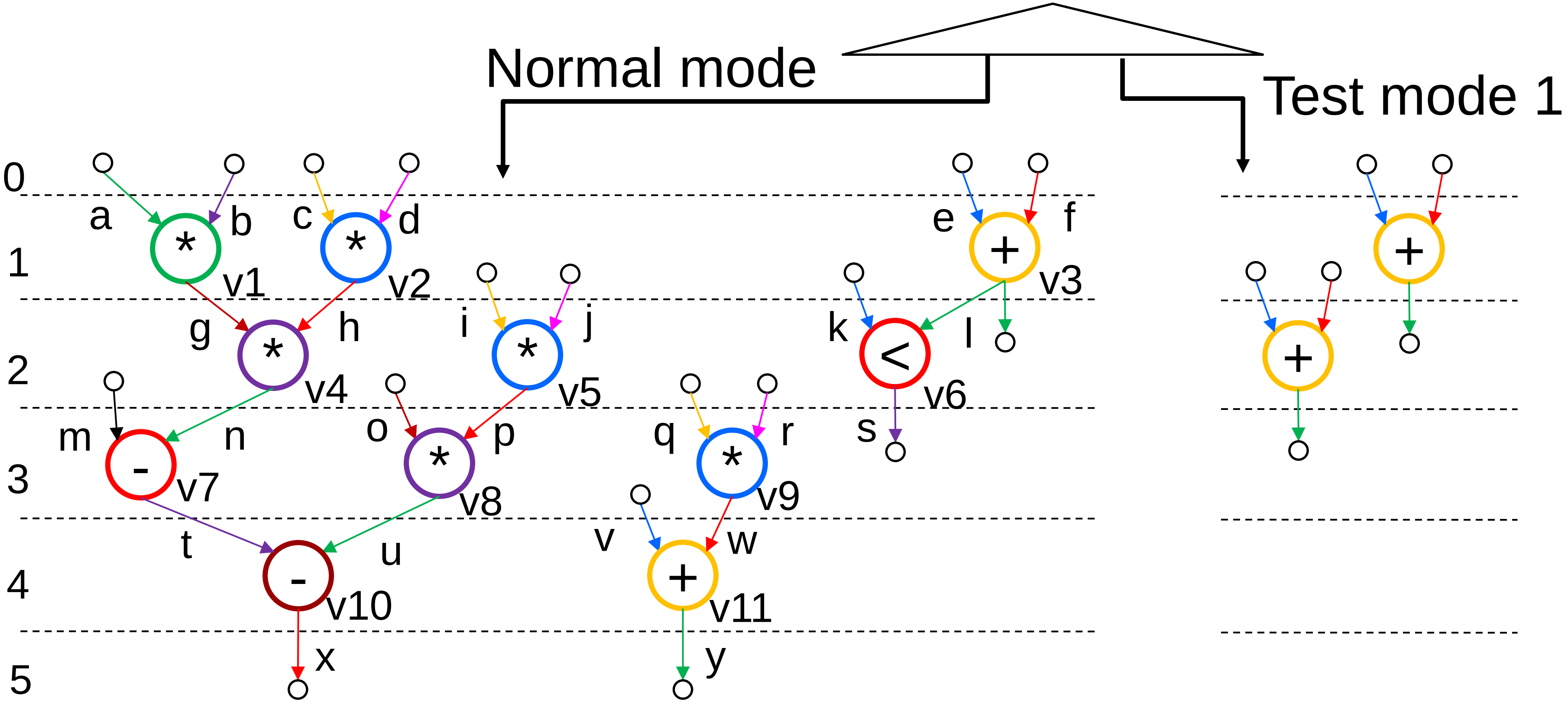
- V4 has two sharable pairs v8 and v9
- V8 has only one sharable pair V4

Minimization of resources



If there is no sharable pair, for the remaining operations and variables, a binding for minimizing resources is applied

Addition of test modes



For the operations which have no testable pair, a test mode is added

- The add module that v3 and v11 are assigned is tested by Test mode 1

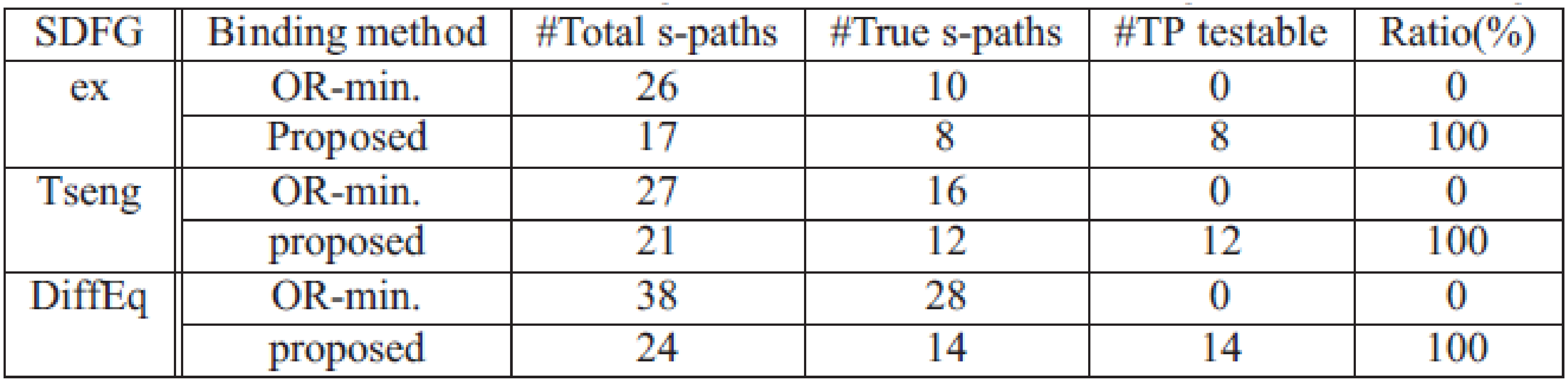
Experimental results

For three SDFGs, Ex, Tseng, DiffEq,

we applied two synthesis methods

- Binding for path delay testability (Proposed)
- Binding for minimizing the number of operational modules and registers (OR-min)

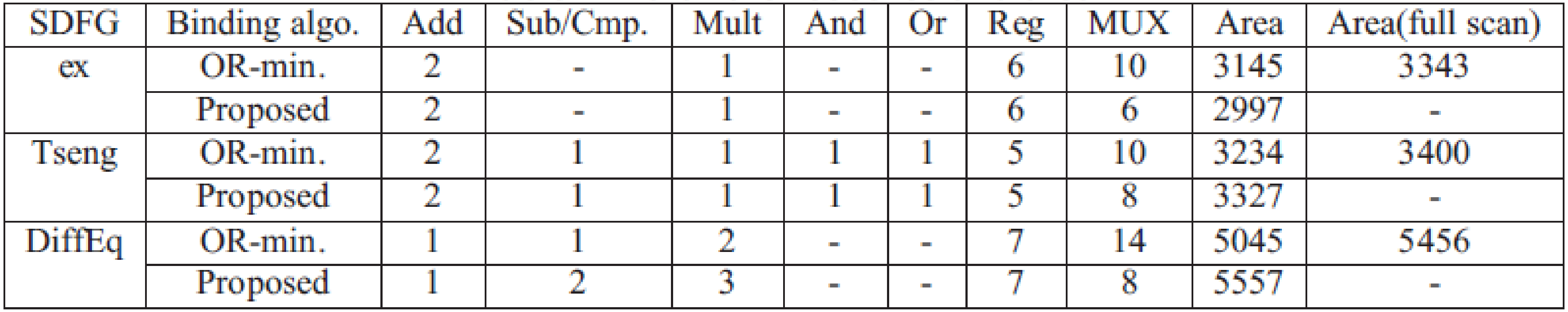
Evaluation of path delay testability



SDFG	Binding method	#Total s-paths	#True s-paths	#TP testable	Ratio(%)
ex	OR-min.	26	10	0	0
	Proposed	17	8	8	100
Tseng	OR-min.	27	16	0	0
	proposed	21	12	12	100
DiffEq	OR-min.	38	28	0	0
	proposed	24	14	14	100

- #Total s-paths synthesized by the proposed binding is less than that synthesized by OR-min
- For the proposed binding method, all the ture paths are two pattern testable
- For OR-min, there is no two pattern testable paths

Evaluation of hardware resource and area



SDFG	Binding algo.	Add	Sub/Cmp.	Mult	And	Or	Reg	MUX	Area	Area(full scan)
ex	OR-min.	2	-	1	-	-	6	10	3145	3343
	Proposed	2	-	1	-	-	6	6	2997	-
Tseng	OR-min.	2	1	1	1	1	5	10	3234	3400
	Proposed	2	1	1	1	1	5	8	3327	-
DiffEq	OR-min.	1	1	2	-	-	7	14	5045	5456
	Proposed	1	2	3	-	-	7	8	5557	-

- For Tseng and DiffEq, each circuit area synthesized by OR-min is smaller than that by the proposed binding
- For Tseng the area of the proposed binding is smaller than that of full scan design
- For DiffEq, both areas are almost the same

Conclusion

We have proposed a binding method in high-level synthesis for path delay testability

- All the single cycle structural paths that are active in the normal function are two-pattern testable
- The area synthesized by the proposed binding is smaller than that synthesized by OR-min

Future works

- More and more experiments for bigger SDFGs.
- A scheduling algorithm for path delay testability