Word Level Functional Coverage Computation

Bijan Alizadeh

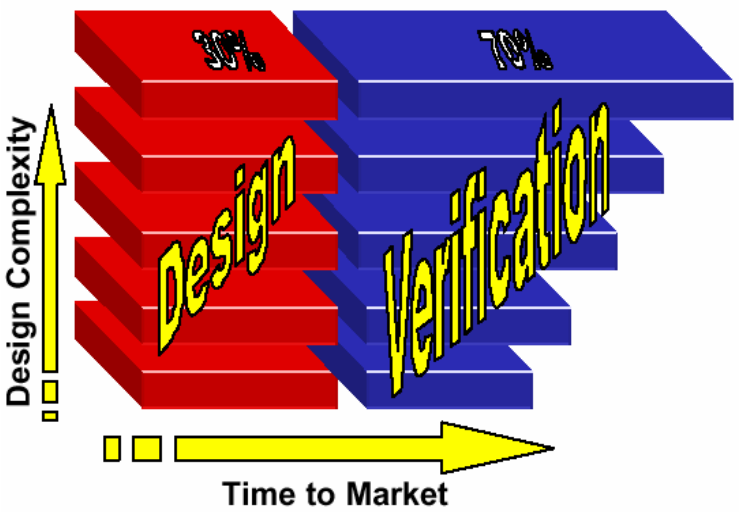
Microelectronic Research and Development Center of IRAN (MERDCI)

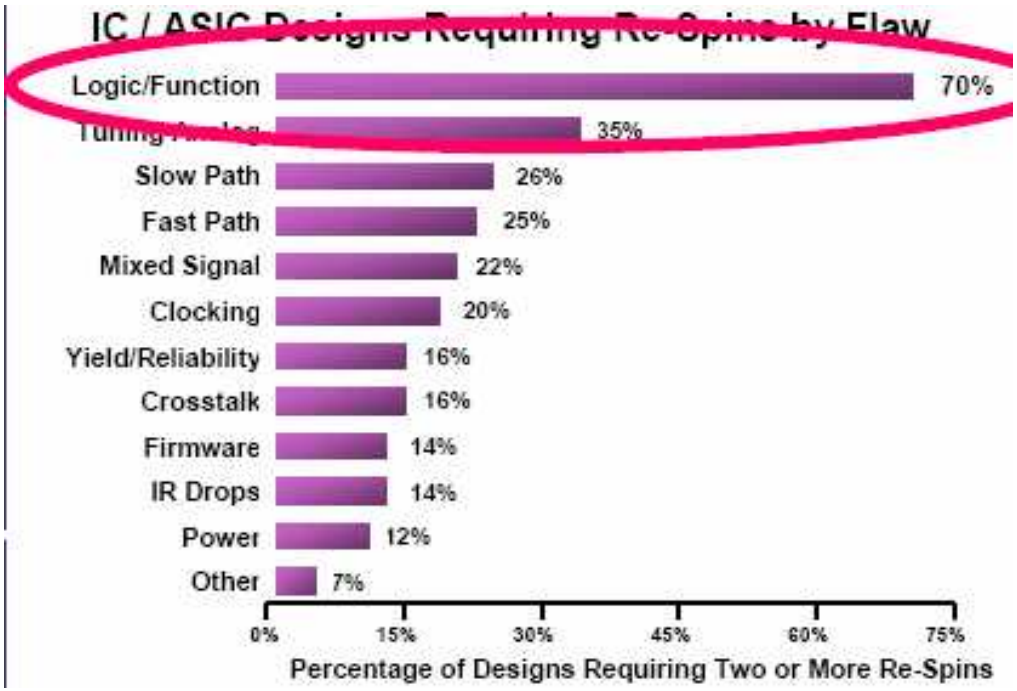
ASP-DAC 2006

Agenda

1. Introduction
2. Word-level Structures
3. Proposed Environment
4. Coverage Computation
5. Experimental Results
6. Future Works

1. Introduction
Industry Forces





Time-to-market

Design Complexity

Shortage of Skilled Engineers

2. Word-level Structures
Taylor Expansion Diagram (TED)

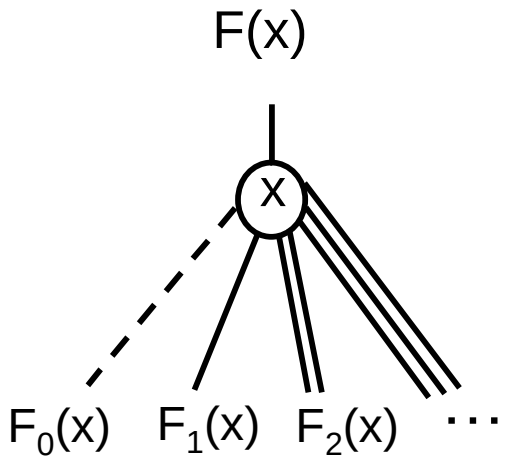
Arithmetic Function (F: Integer → Integer)

- Treat F as a continuous function
- Taylor expansion (around X=0):

$$F(X) = F(0) + X*F'(0) + \tfrac{1}{2}*X^2*F''(0) + \ldots$$

Notation

- $F_0(x)$ = F(x=0) 0-child - - - - - -
- $F_1(x)$ = F'(x=0) 1-child ----------
- $F_2(x)$ = ½ F''(x=0) 2-child ======
- etc.



2. Word-level Structures

TED Example

$$F = A^2B + 2C + 3$$

Ⓐ -- $F_0(A) = F_{|A=0} = 2C + 3$

— $F_1(A) = F'_{|A=0} = 2AB_{|A=0} = 0$

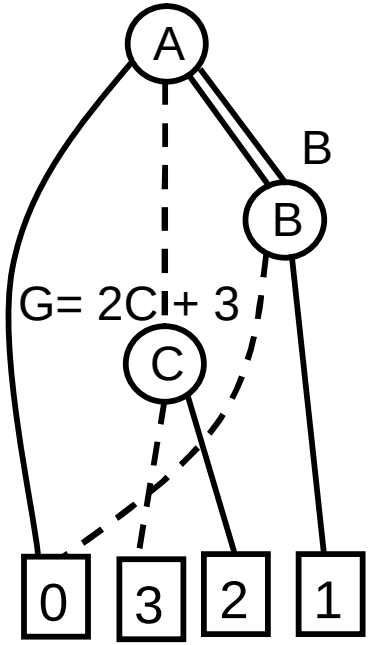
═ $F_2(A) = ½\ F''_{|A=0} = B$

Ⓑ -- $B_0 = B(0) = 0$

— $B_1 = B' = 1$

Ⓒ -- $G_0(C) = (2C+3)_{|C=0} = 3$

— $G_1(C) = (2C+3)' = 2$



(without normalization)

2. Word-level Structures

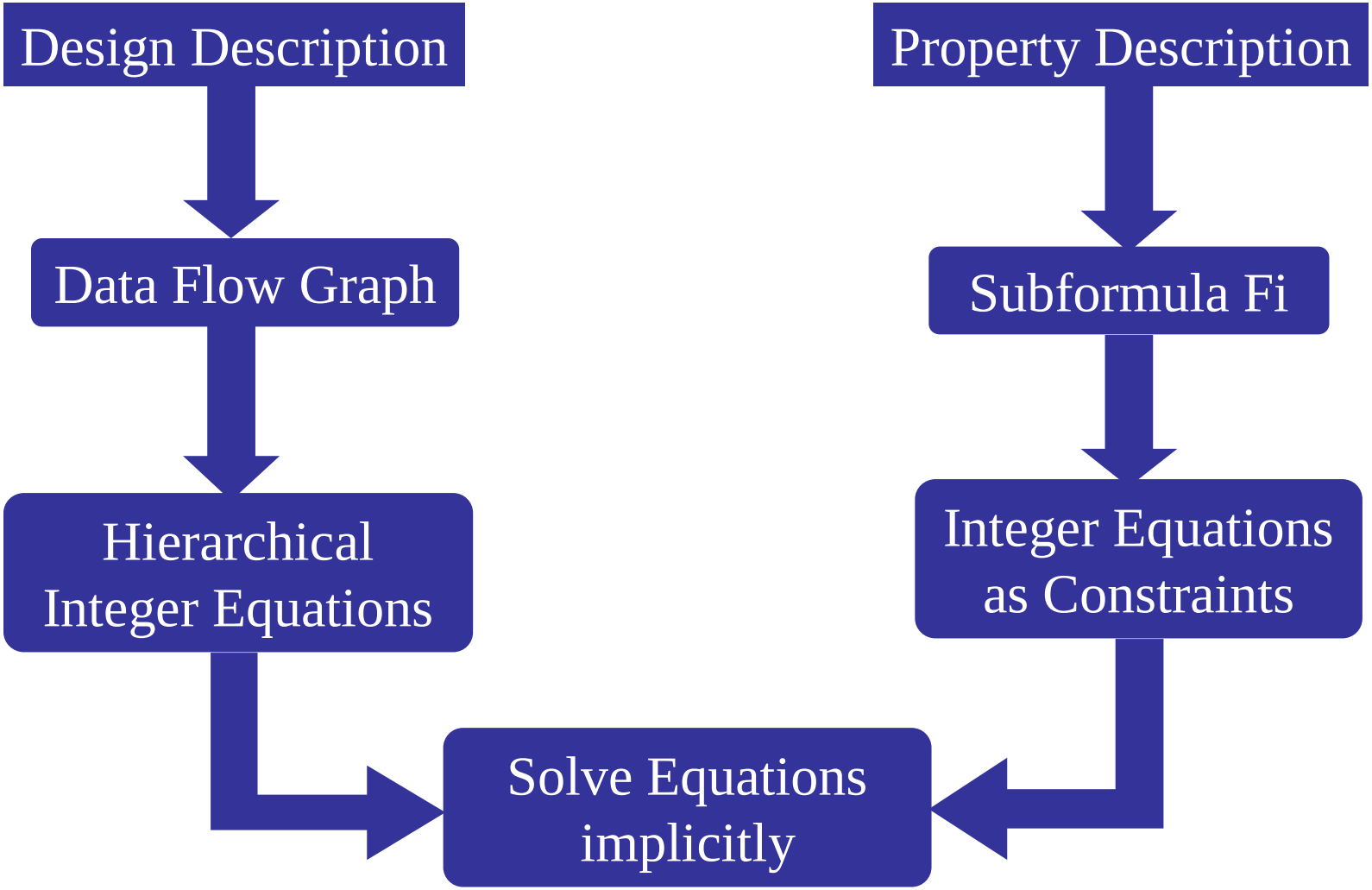
TED Limitations

- Represent relational expressions (X < Y) by using bit expansion

- Normalization: More complicated than in *BMD

3. Proposed Environment

Hierarchical Integer Equations



3. Proposed Environment
DFG Extraction

- A table based representation of the design

Target	Value	Condition	FlipFlop Flag

List of Next
Value of
aft
Specify some cond
until Target
Specify whether the related signal is output of a Flip-Flop

3. Proposed Environment

New Word-level Structure: Linear TED

Algebraic Expression

F(x,y,…) = constant part + x*(linear part)

Where x is top variable of F(x,y,…)

Six Nodes

Constant (C)

Variable (V)

Relational Variable (RV)

Branch node (BR) is described as follows:

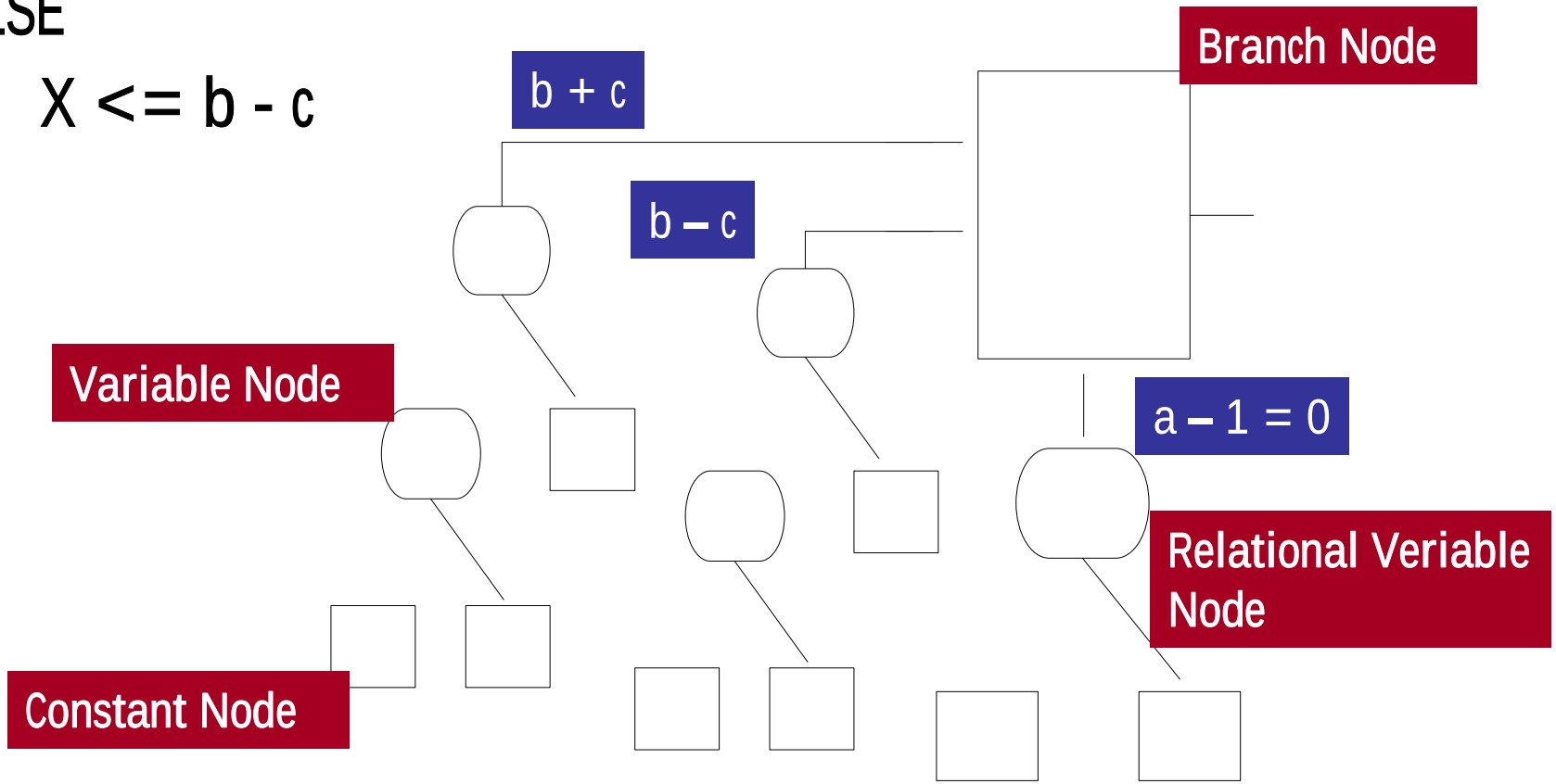
F = Select.InOne + Not(Select).InZero

Union (U) and Intersect (I) nodes are the same as Disjunction and Conjunction operations

3. Proposed Environment

LTED Example

```
IF (a) THEN
    X <= b + c
ELSE
    X <= b - c
```



3. Proposed Environment

Union and Intersect Operations

I: $X + 2Y - 1 > 0$ → $a0 = 1; b0 = 2; c0 = -1$

II: $2X + 4Y - 4 > 0$ → $a1 = 2; b1 = 4; c1 = -4$

Parallel

$a0*b1 = a1*b0$

Both Upward direction

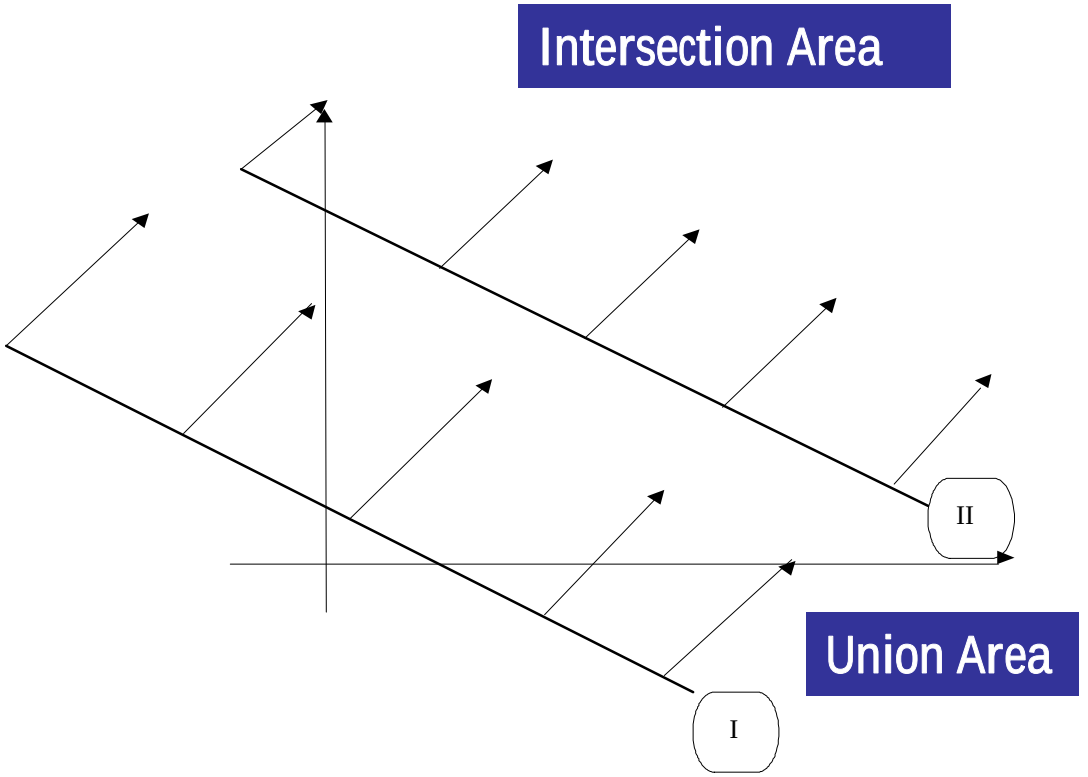
Greater Than Operator

$a0*a1 + b0*b1 > 0$

$b0 > 0$

(I) is below (II)

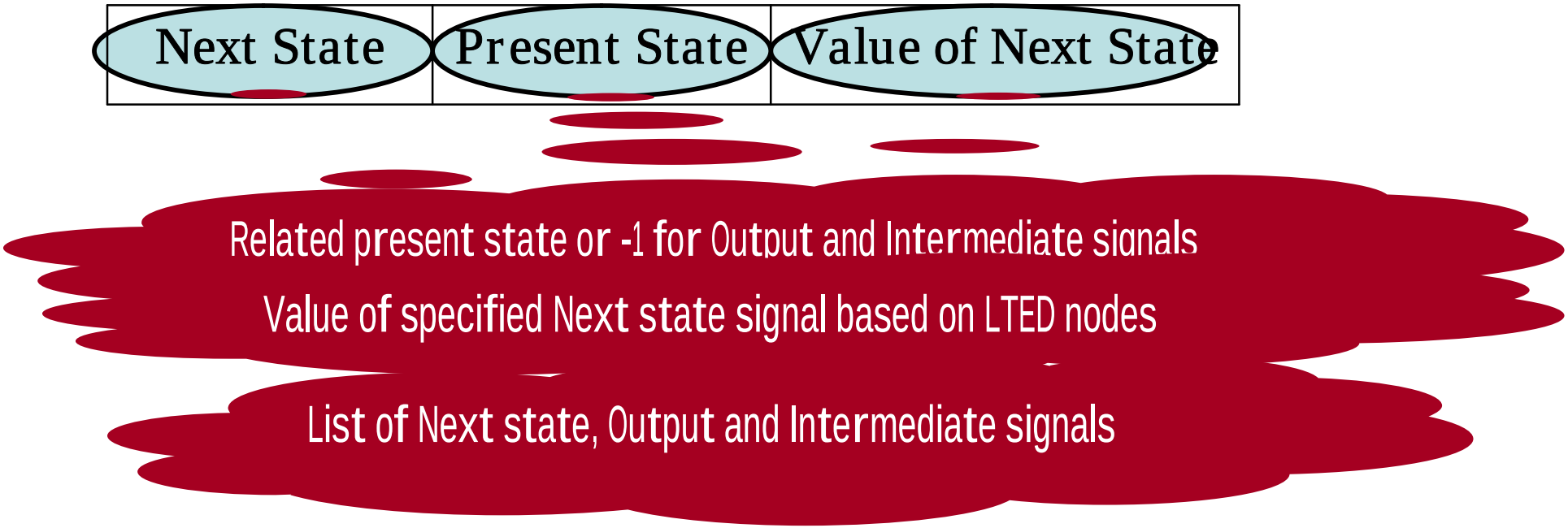
$c0*b1 - c1*b0 > 0$



3. Proposed Environment

Hierarchical Integer Equation Representation

- DFG to Integer Equations



3. Proposed Environment

Property Description

Property (*Q*)

Linear time logic: Q ::= {P1=>P2}

Q Subdivides into

1) Assumption Part (P1): can be specified at different times

P::=(P) | P∧P | ¬P | P=P | P>P | P>=P | P≠P | time=i, P | time=[i to j], P | Variable | IntegerValue

2) Commitment Part (P2)

P::=(P) | P∧P | ¬P | P=P | P>P | P>=P | P≠P | time=i, P | time=[i to j], P | Variable | IntegerValue

3. Proposed Environment

Implicit Solving Method

1. Propagation Phase

Extract LTED value of specified signal from table-based representation of the design

Propagate constant value and relational operator into the LTED structure of the signal (new LTED)

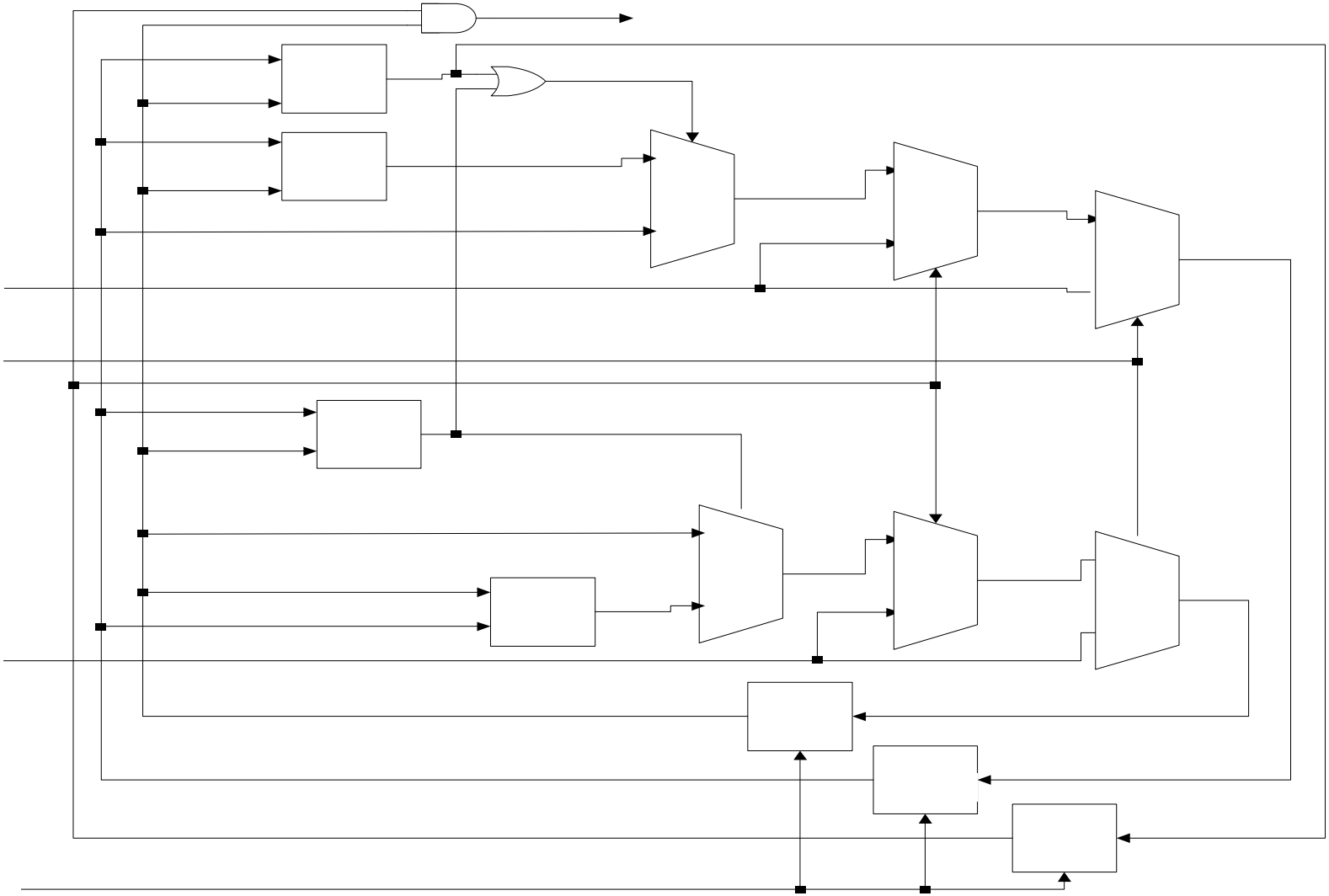
2. Simplification Phase

Apply assumptions to new LTED

Simplify it

3. Proposed Environment

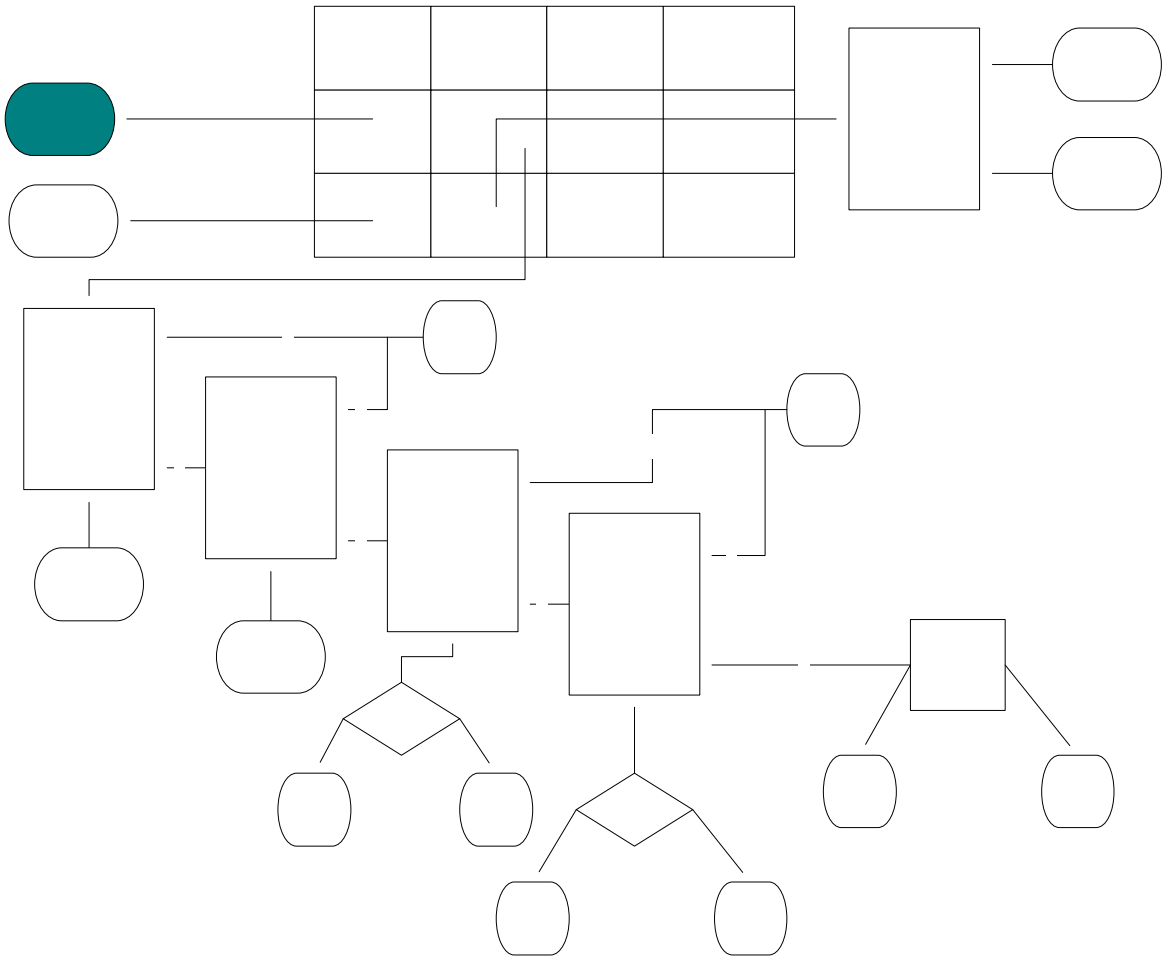
Example: Greatest Common Divisor



3. Proposed Environment

First Step: DFG Extraction

Data Flow Graph of nxtX & X signals



3. Proposed Environment

Second Step: Integer Equations Conversion

List of Next state, Output and Intermediate signals

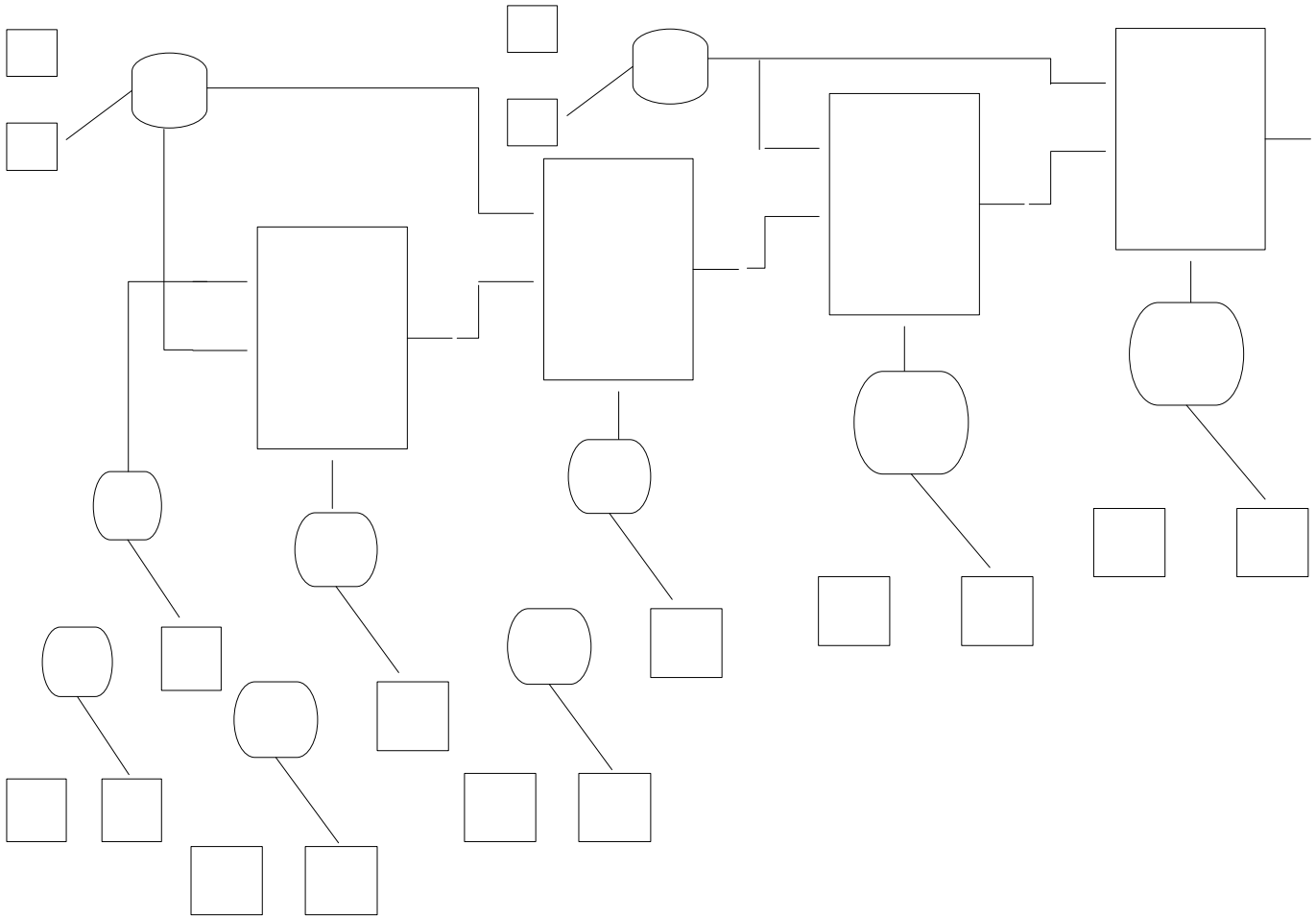
Related present state or -1 for Output and Intermediate signals

Next State	Present State	Value of Next State
ID of nxtX	ID of X	See next slide
ID of nxtY	ID of Y	LTED Node
ID of nxtReset	ID of Reset	LTED Node
ID of Out	-1	LTED Node

3. Proposed Environment

Second Step: Integer Equations Conversion

Integer Equation form of nxtX signal



4. Coverage Computation

Example: Property Checking

Property:

{time=1, Start=0 & Reset=0 & X=10 & Y=5
=> time=2, X=5}

1. Assumptions (*time=1*):

Start=0 & Reset=0 & X=10 & Y=5

2. Commitment (time=2): X=5

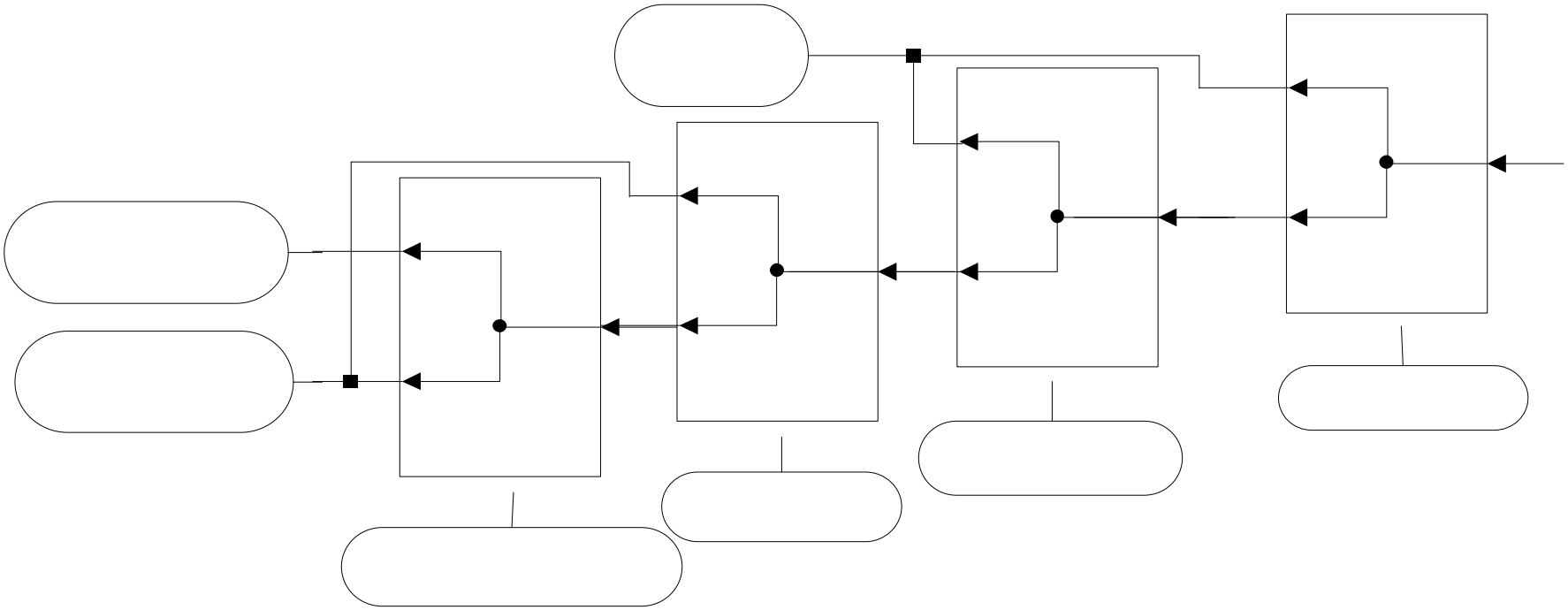
time=2, X=5 means nxtX=5 ---> nxtX-5=0

4. Coverage Computation

Example: Property Checking

1. Extract value of nxtX signal from the table

2. Propagation Phase

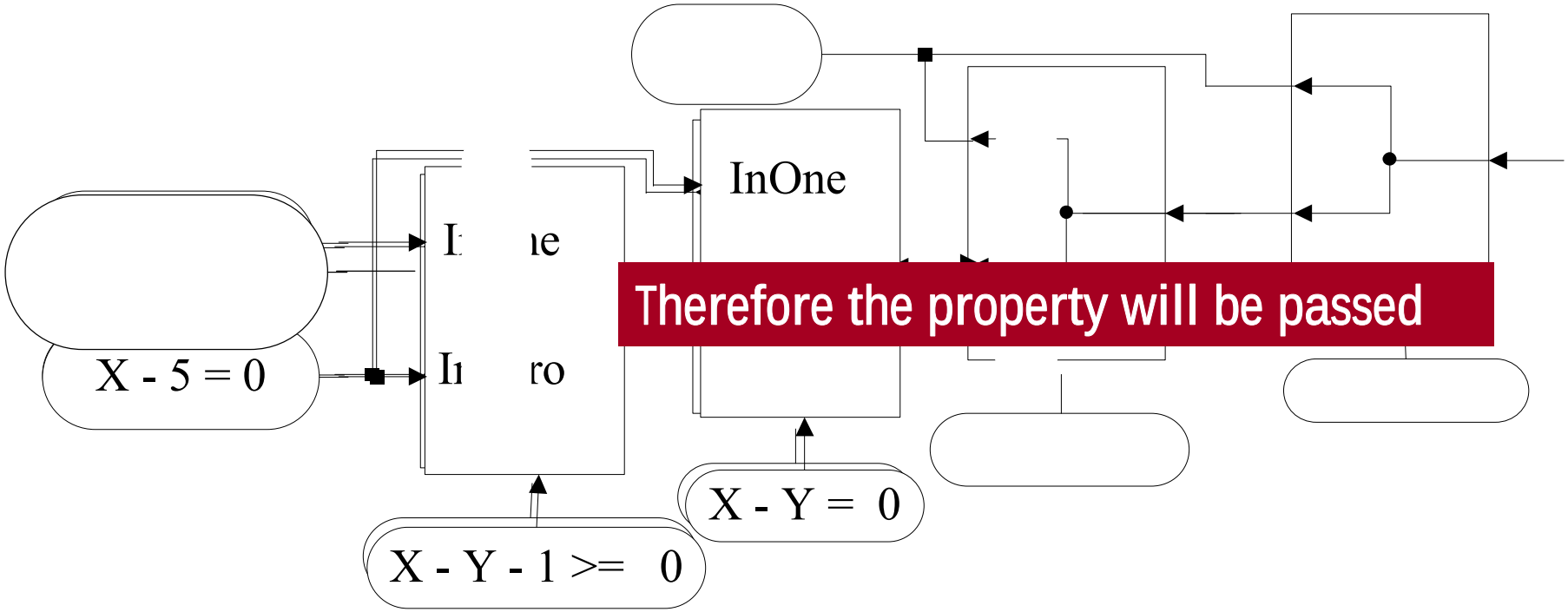


4. Coverage Computation

Example: Property Checking

3. Simplification

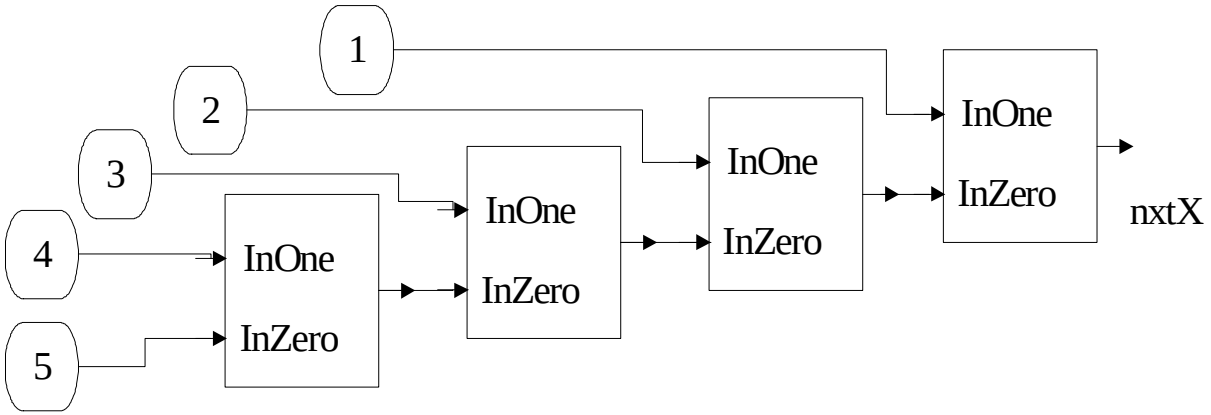
Start=0; Reset=0; X=10; Y=5



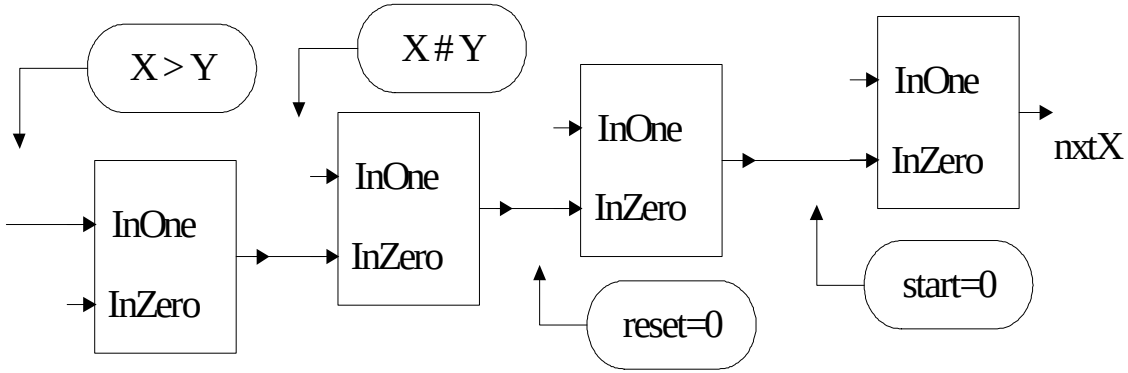
Therefore the property will be passed

4. Coverage Computation
Parameters

1. All Paths



2. Covered Paths



4. Coverage Computation
Definition

Coverage =

Number of Covered Paths/ Number of All Paths

```
int NumberofAllPaths(LTED In)
if(In->type==Branch)
   if(pIn1->type!=Branch)
      if(pIn0->type!=Branch)  return 2;
      else return NumberofAllPaths(pIn0)+1;
   else if(pIn0->type!=Branch)
            return NumberofAllPaths(pIn1)+1;
         else
     returnNumberofAllPaths(pIn0)+NumberofAllPaths(pIn1)+1;
```

5. Experimental Results

Circuit		TLC	GCD	SAYEH	EL	2CA
SN		hwyl	X	DataBus	door	cntl1
P3	WLM	12.1	0.03	11.6	0.21	0.1
	PC	20%	25%	10%	33.3%	25%
	VIS	19.2	0.13	NS	4.7	0.9
P4	WLM	0.4	0.03	12.1	---	0.01
	PC	20%	25%	20%	---	25%
	VIS	0.9	0.14	39.8	---	0.1
TC		80%	100%	55%	100%	100%
N	WLM	60	32	1612	87	62
	VIS	974	968442	419062	20418	39381
M	WLM	5.3	4.5	10.3	4.1	5.1
	VIS	10.1	36	26.48	5.2	5.5

SN: Signal Name

P3: Cpu Time of Property3 (Sec.)
P4: Cpu Time of Property4 (Sec.)

WLM: our Word Level Method

PC: %Property Coverage

TC: %Total Coverage

N: Number of Nodes (LTED,BDD)

M: Memory Usage (MegaByte)

6. Future Works

- More Metrics
- Efficient Implementation
- Support Higher Order Expressions

7. Good bye

Thanks for your attention

Questions?