Faults and High-Level Decision Diagrams

RTL-statement:

K: (*If* T,C) $R_D \leftarrow F(R_{S1}, R_{S2}, \ldots R_{Sm}), \rightarrow N$

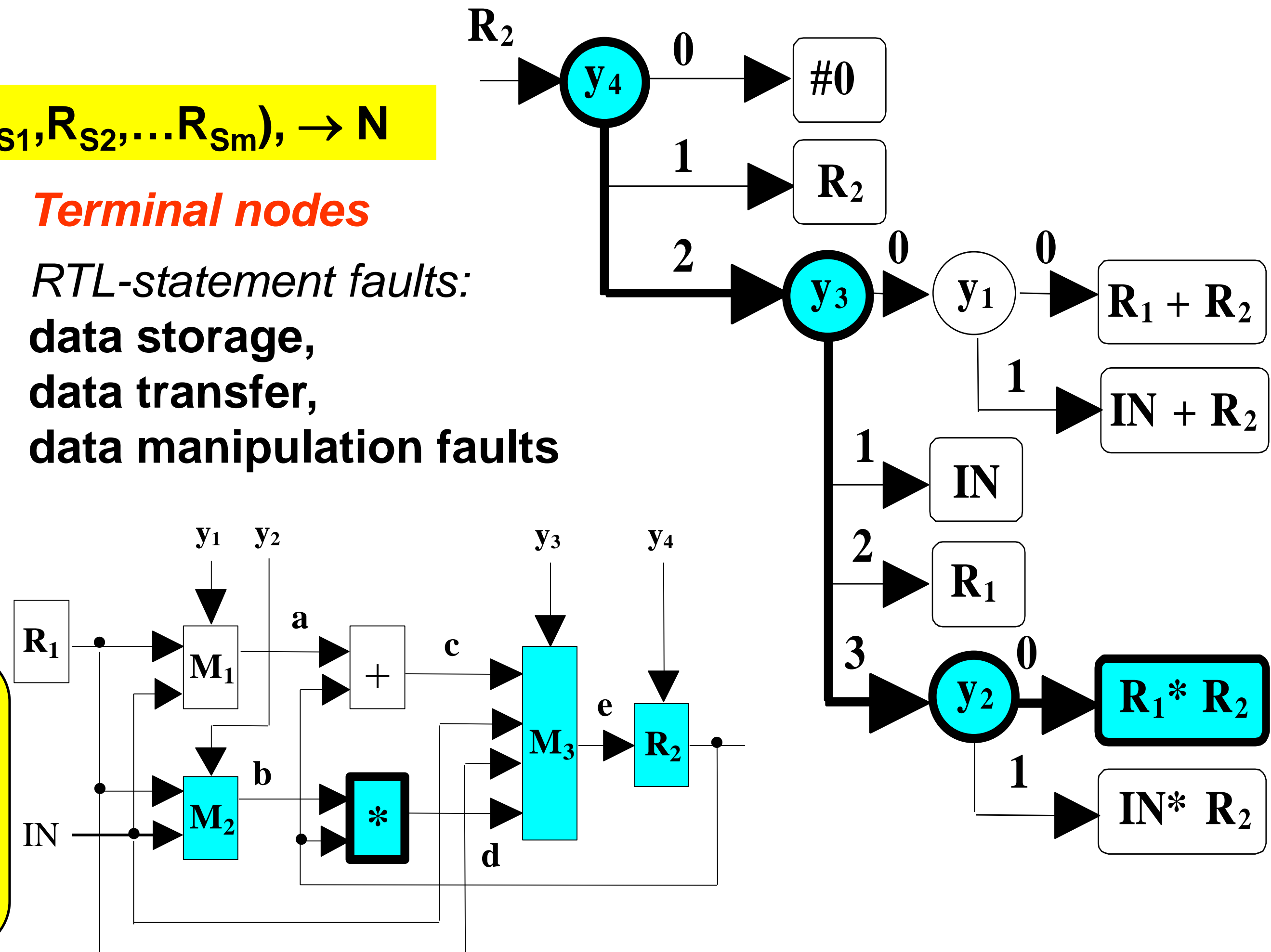
Nonterminal nodes

RTL-statement faults:
label,
timing condition,
logical condition,
register decoding,
operation decoding,
control faults

Terminal nodes

RTL-statement faults:
data storage,
data transfer,
data manipulation faults

Testing concept on the DD-model (uniform for all nodes):
1) Exhaustive testing
2) Optimization



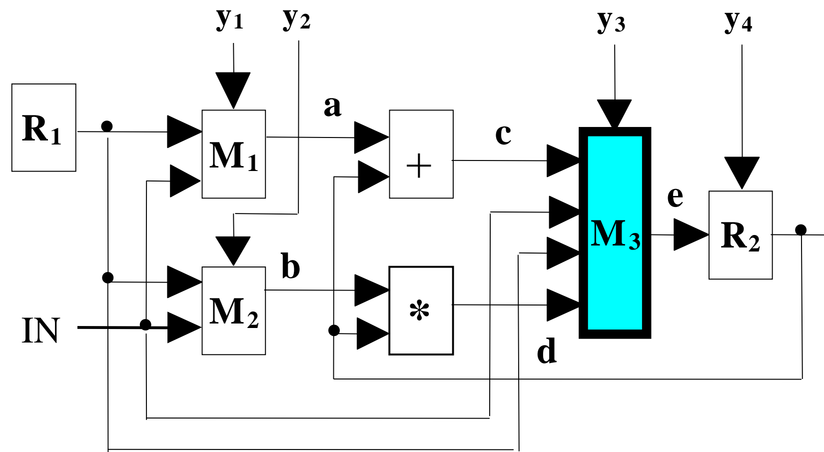
Test Generation for Digital Systems

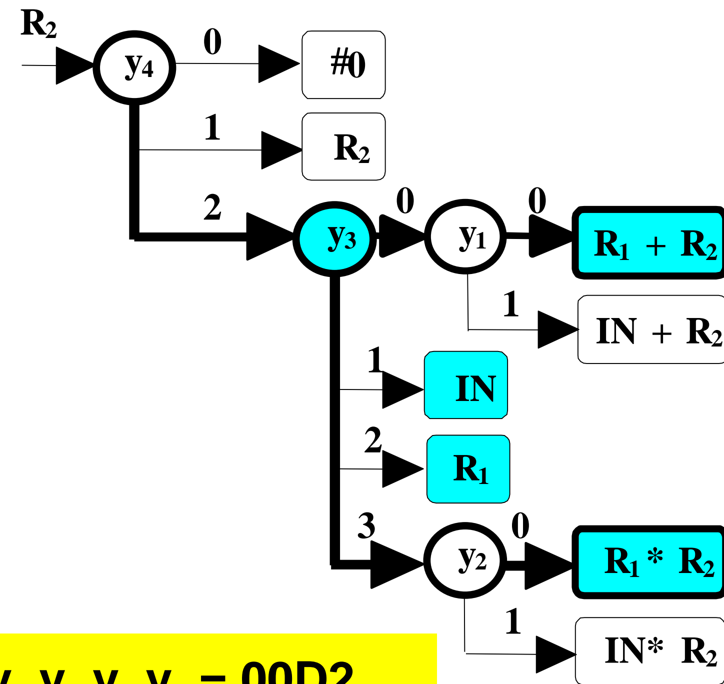
High-level test generation with DDs: Conformity test

Multiple paths activation in a single DD
Control function y_3 is tested

Decision Diagram

Data path





Test program:

Control: *For* **D = 0,1,2,3:** $y_1\, y_2\, y_3\, y_4 = 00D2$

Data: *Solution of* $R_1 + R_2 \neq IN \neq R_1 \neq R_1 * R_2$

Test Program Synthesis with HLDDs

Test algorithm:

Control: *For* **D** = **0,1,2,3:** $y_1 y_2 y_3 y_4$ = **00D2**
Data: *Solution of* $R_1 + R_2 \neq IN \neq R_1 \neq R_1 * R_2$

Comment:

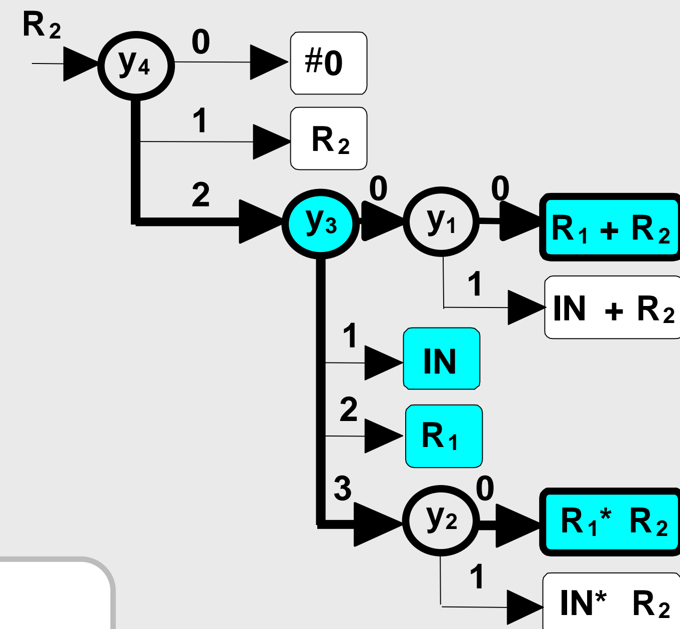
The data rule is simplified

Test program:

For D = 0,1,2,3
Begin
Load R1 = **IN1**
Load R2 = **IN2**
} *Initialization*
Apply
IN = **IN3**
$y_1 y_2 y_3 y_4$ = 00**D**2
} *Test*
Read **R2**
} *Observation*
End

Data: *(IN1+IN2) ≠ IN3 ≠ IN1 ≠ (IN1*IN2)*

Decision Diagram



Advantages:
- Straightforward synthesis procedure
- Compactness of the cycle-based test program

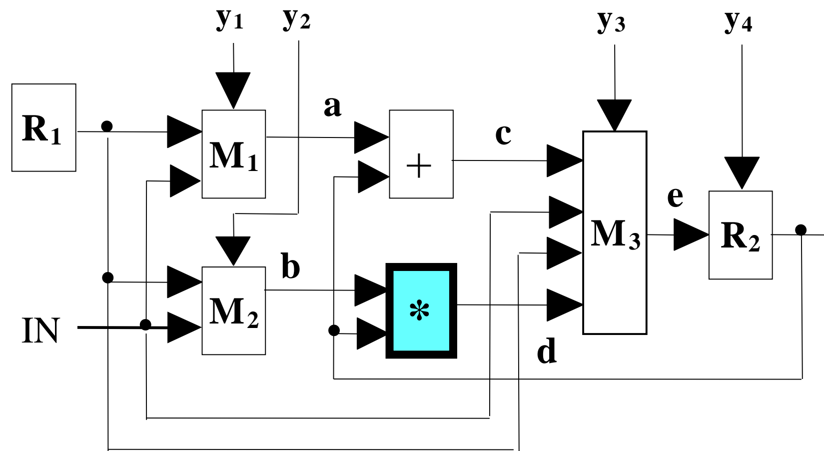
Test Generation for Digital Systems

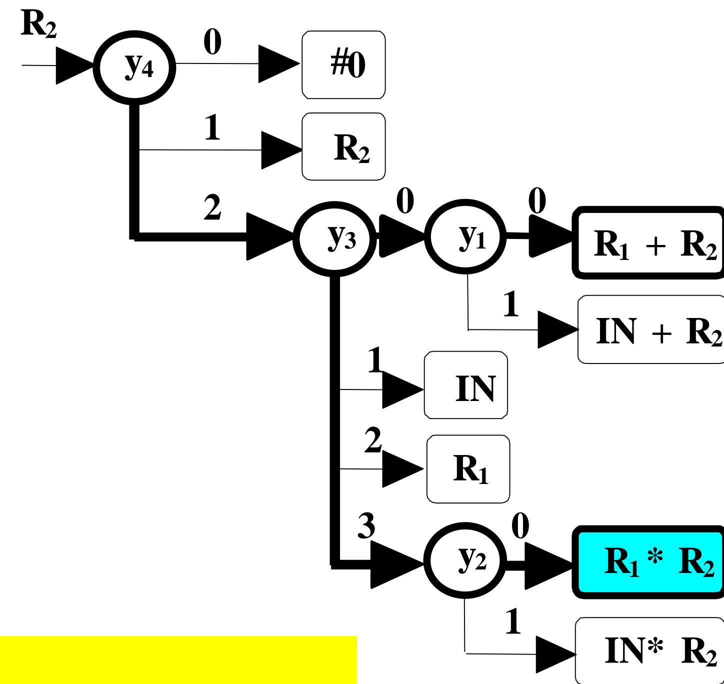
High-level test generation with DDs: Scanning test

Single path activation in a single DD
Data function $R_1 * R_2$ is tested

Decision Diagram

Data path





Test program:

Control: $y_1 y_2 y_3 y_4 = 0032$

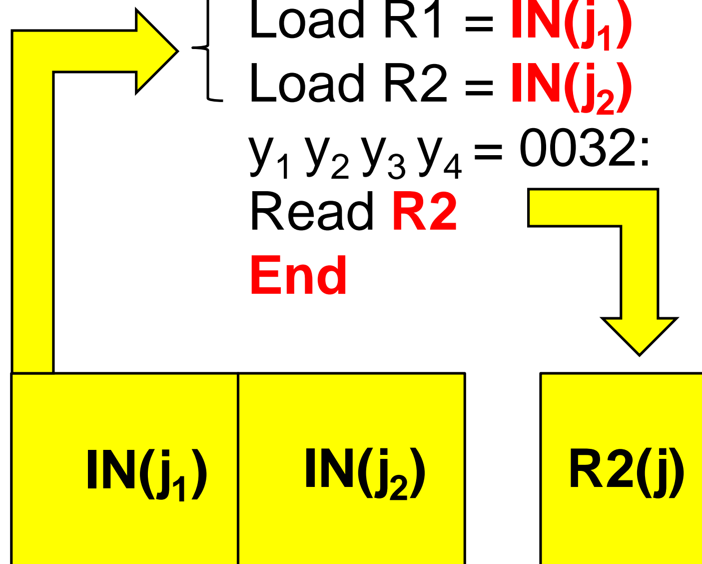
Data: *For all specified pairs of* (R_1, R_2)

Test Generation for Digital Systems

High-level test generation with DDs: Scanning test

Test program:

For j=1,n
Begin
Load R1 = **IN(j_1)**
Load R2 = **IN(j_2)**
$y_1\, y_2\, y_3\, y_4$ = 0032:
Read **R2**
End

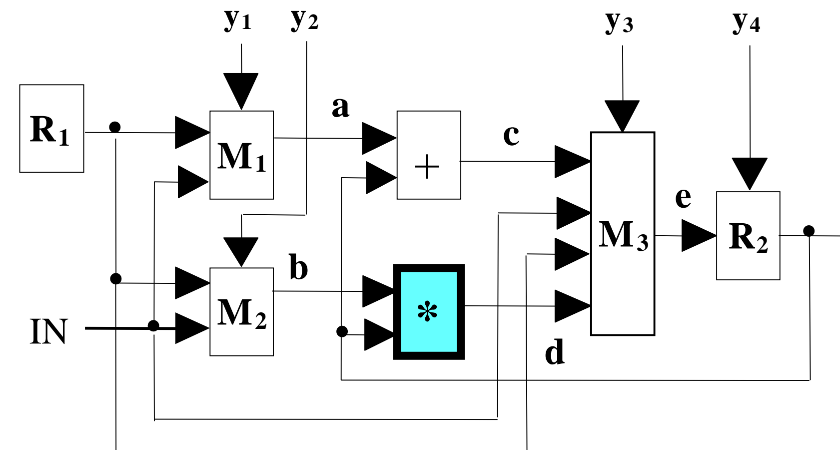


Test data

Test results

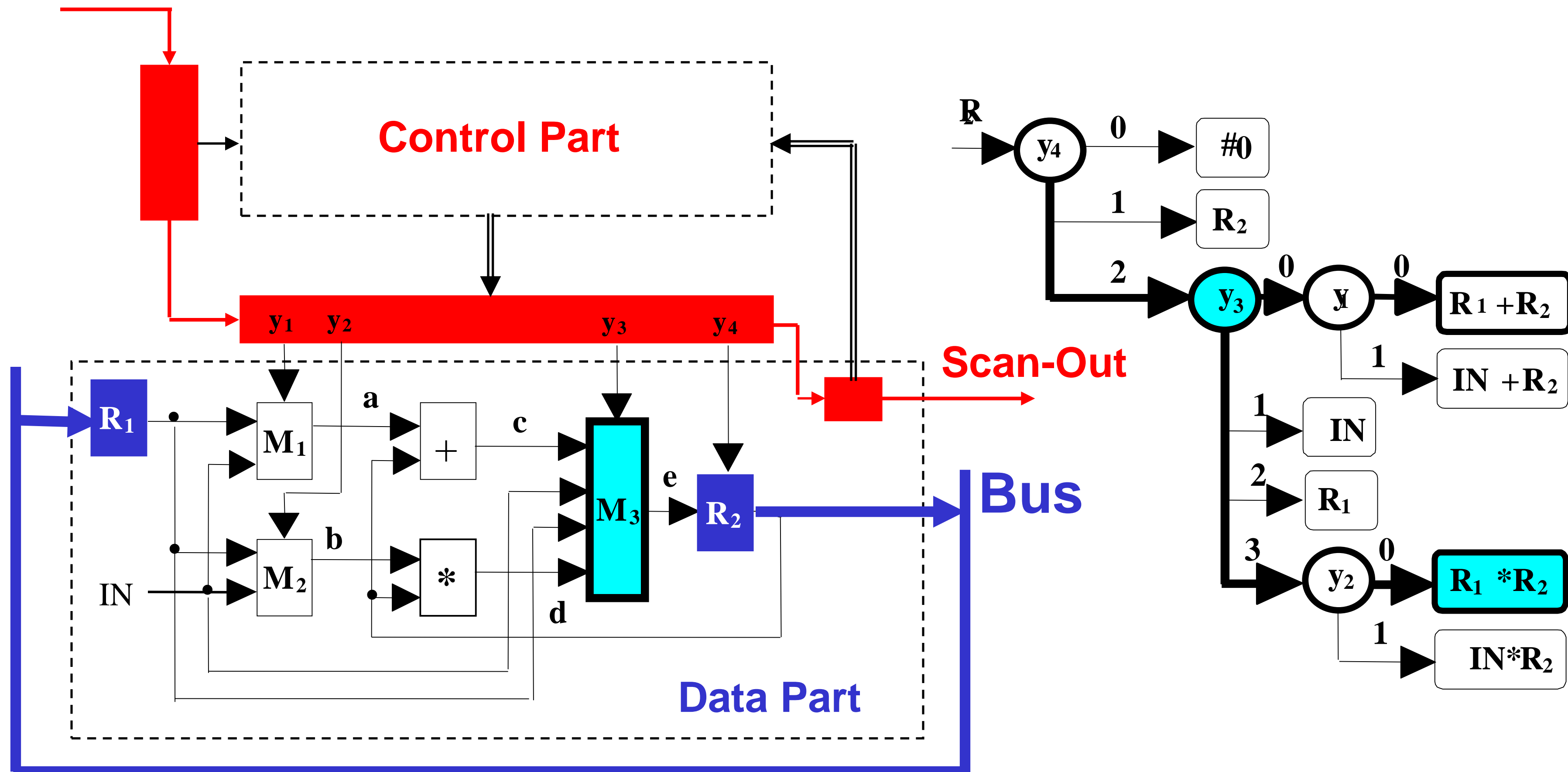
Test template:

Control: $y_1\, y_2\, y_3\, y_4$ = 0032
Data: *For all specified pairs of* (R_1, R_2)



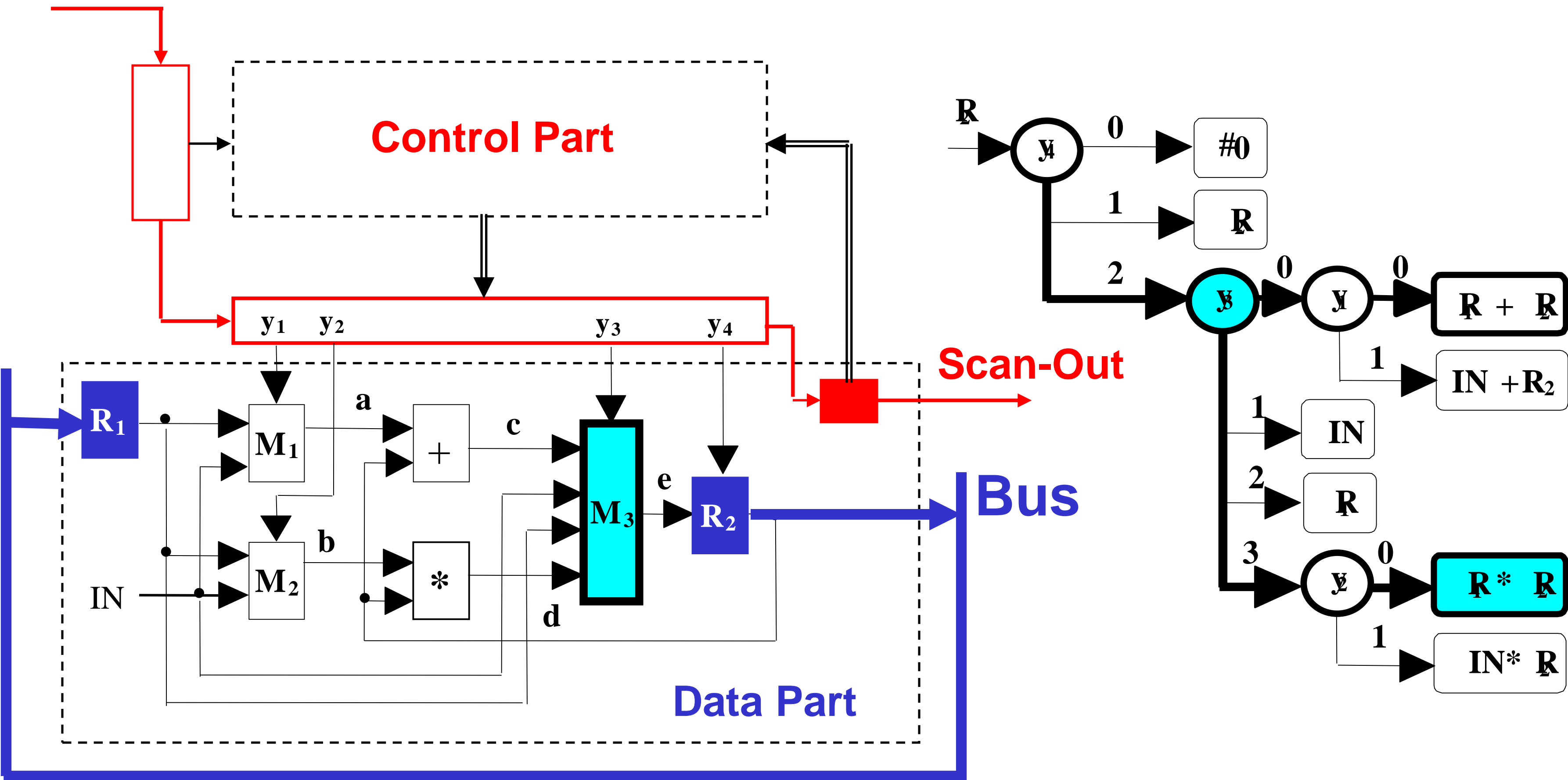
Scan-Path for Making Systems Transparent

Hierarhical test generation with Scan-Path:



Testing with Minimal DFT for Scan-Path

Hierarhical test generation with Scan-Path:

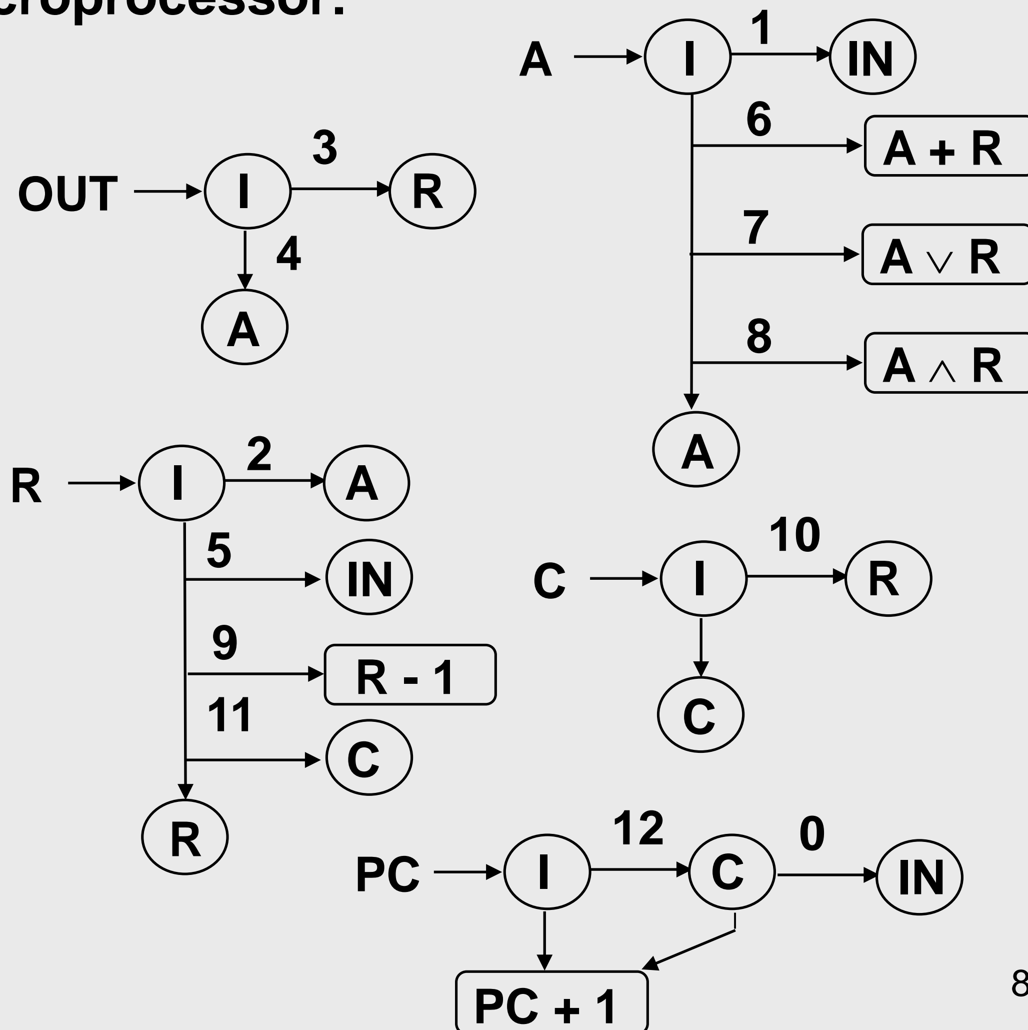


Microprocessor Modeling with HLDDs

HLDD-model of a microprocessor:

Instruction set:

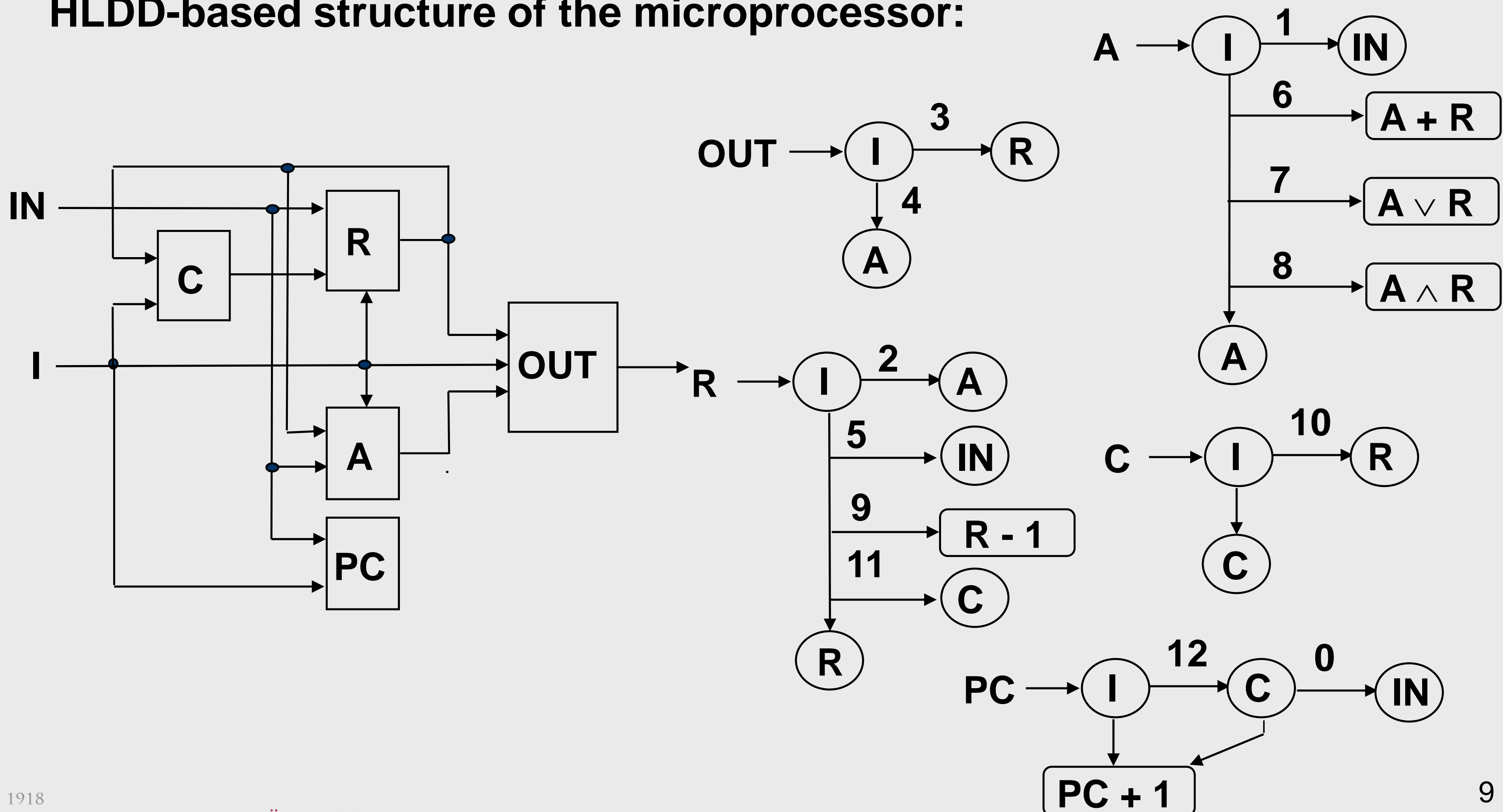
I_1:	MVI A,M	A ← IN
I_2:	MOV R,A	R ← A
I_3:	MOV M,R	OUT ← R
I_4:	MOV M,A	OUT ← A
I_5:	MOV R,M	R ← IN
I_6:	ADD R	A ← A + R
I_7:	ORA R	A ← A ∨ R
I_8:	ANA R	A ← A ∧ R
I_9:	SUB R	R ← R - 1
I_{10}:	MOV C,R	C ← R
I_{11}:	CMA R,C	R ← C
I_{12}:	JMP PC, C	IF C=0 THEN PC = IN



TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Microprocessor Modeling with HLDDs

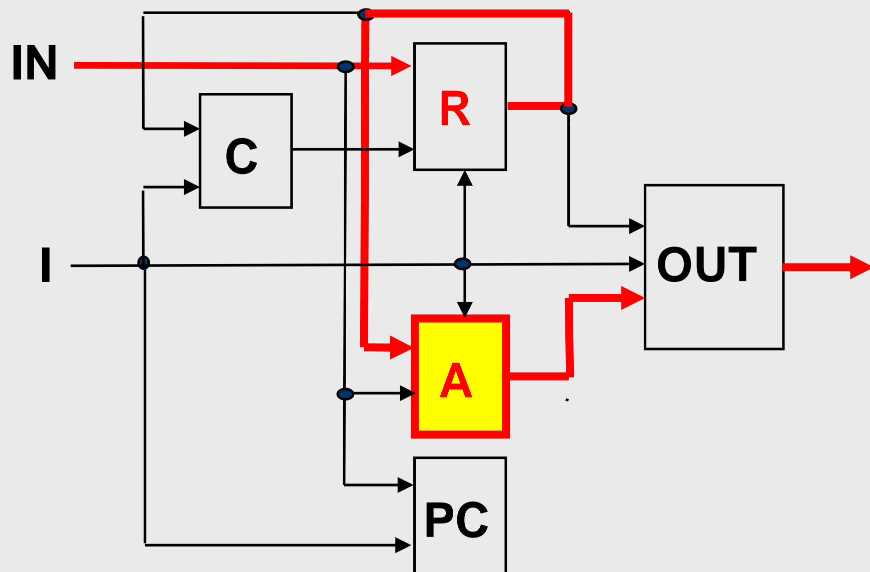
HLDD-based structure of the microprocessor:



1918
TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Simulation of Microprocessors with HLDDs

HLDD-based structure of the microprocessor:



OUT → I; 3 → R; 4 → A

R → I; 2 → A; 5 → IN; 9 → R - 1; 11 → C; R

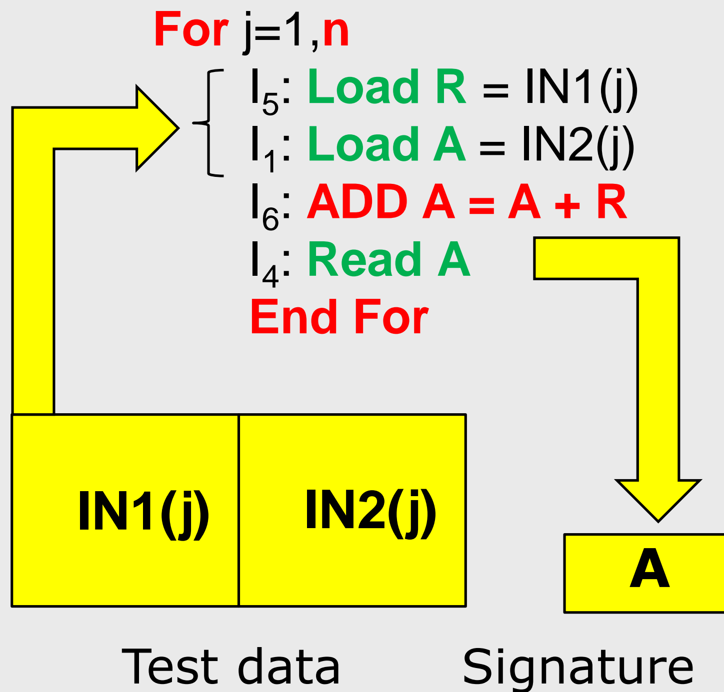
A → I; 1 → IN; 6 → A + R; 7 → A ∨ R; 8 → A ∧ R; A

C → I; 10 → R; C

PC → I; 12 → C; 0 → IN; PC + 1

Test Generation for Microprocessors

Scanning test program:



OUT → I, 3 → R, 4 → A

R → I, 2 → A, 5 → IN, 9 → R - 1, 11 → C, → R

A → I, 1 → IN, 6 → A + R, 7 → A ∨ R, 8 → A ∧ R, → A

C → I, 10 → R, → C

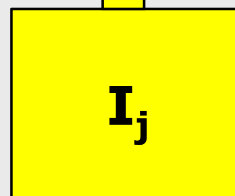
PC → I, 12 → C, 0 → IN, → PC + 1

1918
TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

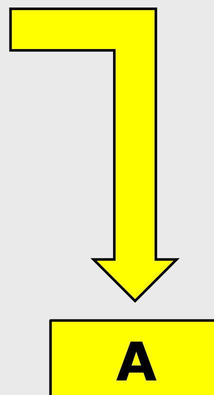
Test Generation for Microprocessors

Conformity test program:

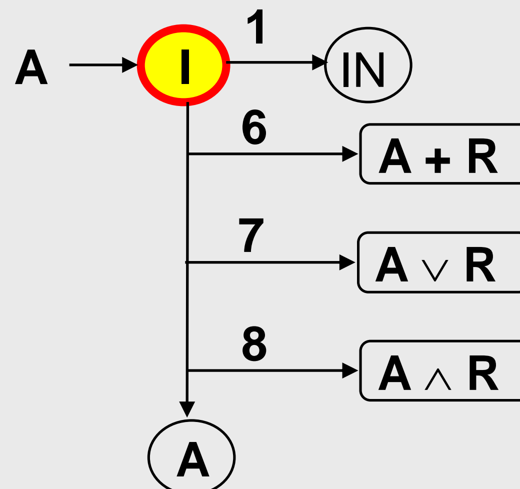
For D=1,n
I_5: **Load R** = IN(1)
I_1: **Load A** = IN(2)
I_D: **D**
I_4: **Read A**
End For



Test data



Signature



Algorithm:

Control: *For* **D = 0,1,….12: I_D**
Data: *Solution of* **IN ≠ (A + R) ≠ (A ∨ R) ≠ (A ∧ R)**

Side effects:

1) Special type of test compaction:
   - DD model
   - Test program template
   - ATPG
2) When testing all the functions of A with the same LOAD and READ conditions, the probability of fault masking will reduce
3) The faults of type "added erroneous actions" are as well easily tested

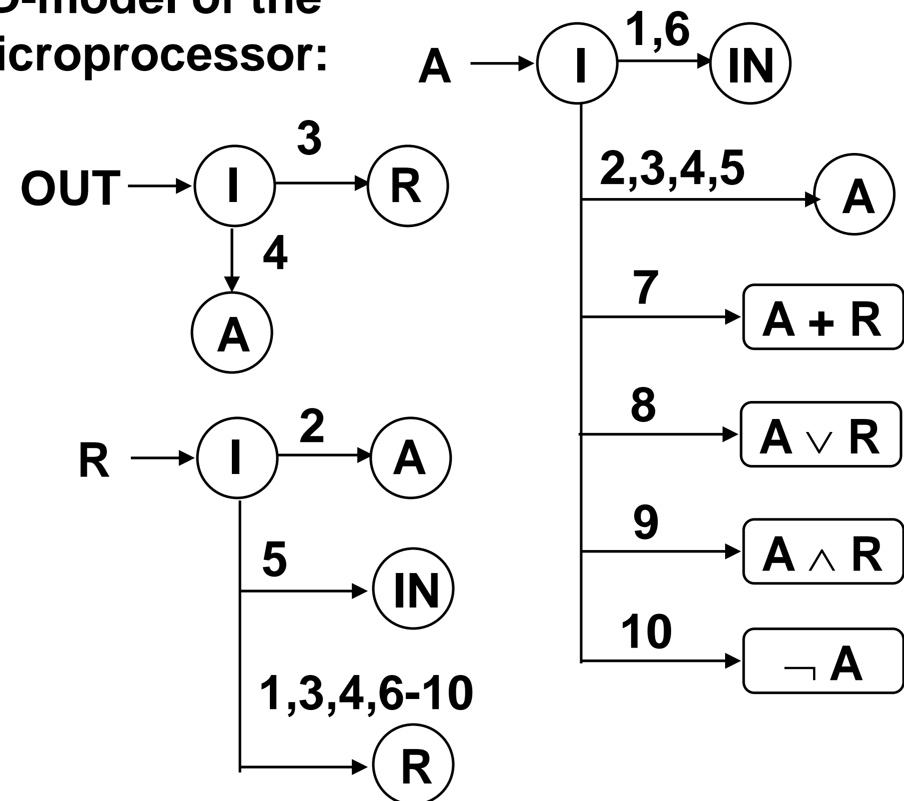
Test Data Generation for Microprocessors

Test program generation for a microprocessor (example):

Instruction set:

I_1:	MVI A,D	$A \leftarrow IN$
I_2:	MOV R,A	$R \leftarrow A$
I_3:	MOV M,R	$OUT \leftarrow R$
I_4:	MOV M,A	$OUT \leftarrow IA$
I_5:	MOV R,M	$R \leftarrow IN$
I_6:	MOV A,M	$A \leftarrow IN$
I_7:	ADD R	$A \leftarrow A + R$
I_8:	ORA R	$A \leftarrow A \vee R$
I_9:	ANA R	$A \leftarrow A \wedge R$
I_{10}:	CMA A,D	$A \leftarrow \neg A$

DD-model of the microprocessor:



Test Data Generation for Microrocessors

Test data generation:

Test: D

Find IN, A, R by solving the inequality:

$IN \neq A \neq (A+R) \neq (A \vee R) \neq (A \wedge R) \neq \neg A$

Data	IN		110
	A		101
	R		110
Functions	I_1, I_6	IN	110
	I_2, I_3 I_4, I_5	A	101
	I_7	A + R	1011
	I_8	$A \vee R$	111
	I_9	$A \wedge R$	00
	I_{10}	$\neg A$	010

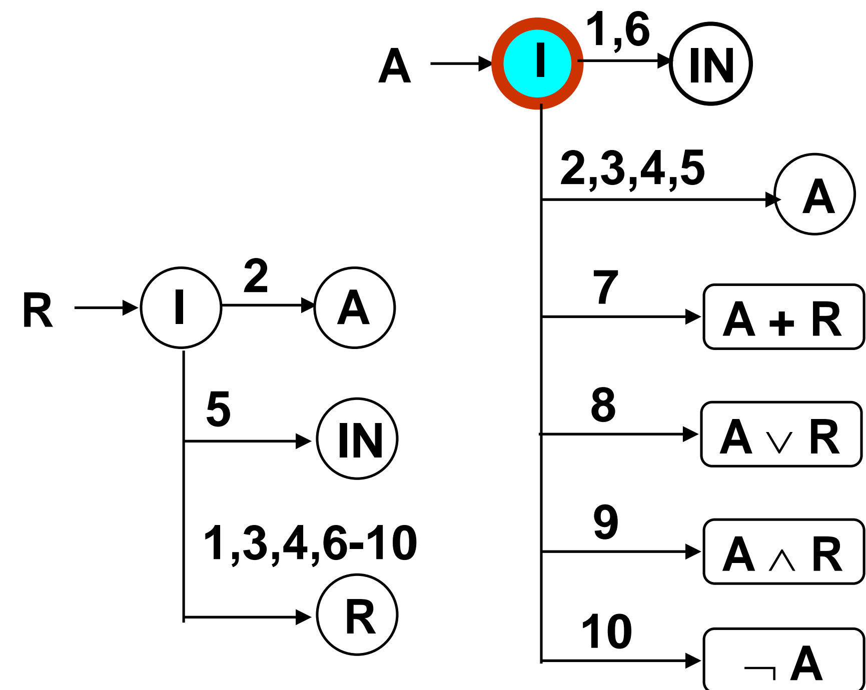
Data IN,A,R are generated so that the values of all functions were different

Response: RRR

Conformity test program for I in A:

Instruction sequence $T = I_5 I_1 D I_4$

for all $D \in \{I_1 - I_{10}\}$ **at given A,R,IN**



Test Data Generation for Microrocessors

Data generation for conformity test program:

Test: D

Data	IN		110
	A		101
	R		110
Functions	I_1, I_6	IN	110
	I_2, I_3 I_4, I_5	A	101
	I_7	A + R	1011
	I_8	$A \vee R$	111
	I_9	$A \wedge R$	00
	I_{10}	$\neg A$	010

Response: RRR

Data IN,A,R are generated so that the values of all functions were different

Conformity test program for decoder:

Instruction sequence $T = I_5 I_1 D I_4$

for all $D \in \{I_1 - I_{10}\}$ at given A,R,IN

Final test program:

Step	I	Mode	IN	A	R
1	I_5	Load R	110		110
2	I_1	Load A	101	101	110
3	I_D	Test	110	101	110
4	I_4	Observe		RRR	

© Raimund Ubar

Summary of Test Generation

- ✓ Low-level methods
  - Pseudo-exaustive methods, gate-level methods
  - Test generation with BDDs
  - Problems: fault redundancy, single vs multiple paths
- ✓ Multiple fault problem
  - Topological view on fault diagnosis
  - Fault masking, test pairs, test groups
- ✓ High-level test generation
  - Hierarchical approach
  - Test generation for RTL and microprocessors