

Diagnostika číslicových zařízení

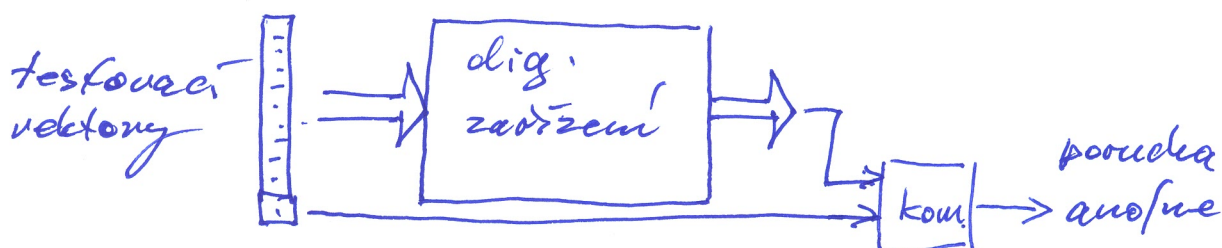
61

- diagnostika kombinovaných obvodů
- diagnostika sekvenčních obvodů
- D-algoritmus
- diagnostika paměťových obvodů
- návrh pro snadnou diagnostiku

porucha --- fyzikální jev, neschopnost
zařízení vykonávat správnou
funkci

chyba --- rozdíl mezi předpokládanou
(správnou) a skutečnou (pozorovanou)
hodnotou

testy --- lokalizační ... určit místo poruchy
detekční ... určit porucha ano/ne



test množina testovacích vektorů

délka testu ... počet test. vektorů

triviální test zkusit všechny možné
kombinace na vstupu

diagnostické pokrytí ...

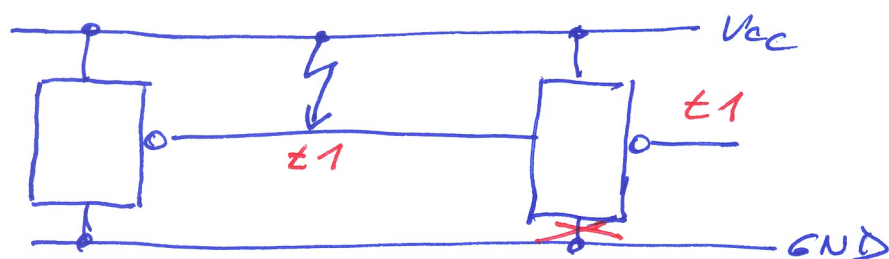
$$\text{diag. pokrytí} = \frac{\text{počet poruch odhalených testem}}{\text{počet všech možných poruch}}$$

úplný test ... diag. pokrytí 100%

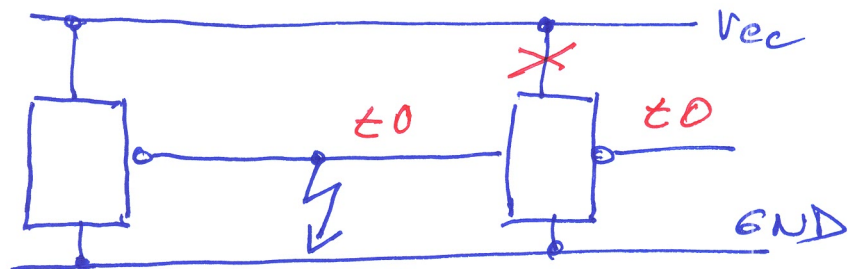
minimální test ... test s minimální možnou
délkou a pokrytím 100%

funkční testy ... známe jen funkci zařízení
strukturální testy ... známe i strukturu tj.
schéma zařízení

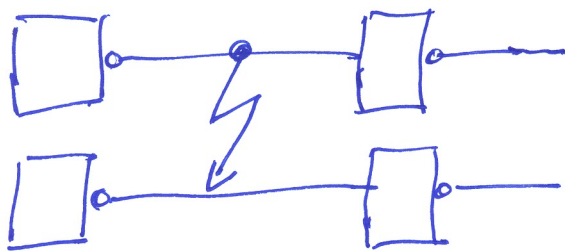
logický model poruchy:



trvalá 1 ... t_1



trvalá 0 ... t_0



nelze modelovat t_0 nebo t_1

a) funkční testy

$$y = f(x_1, x_2, \dots, x_n)$$

logická derivace $\frac{df}{dx_i}$

$$\frac{df}{dx_i} = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \oplus f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

např. $f(x): y = x_1 \cdot \overline{x_2} + x_1 \cdot x_3 + x_3 \cdot x_4$

$$\frac{df}{dx_1} :$$

$\overline{x_2}$
 $\overline{x_3}$
 $\overline{x_4}$

x_2
 x_3
 x_4

$$\frac{df}{dx_1} = \overline{x_2} \cdot \overline{x_3} + \overline{x_2} \cdot \overline{x_4}$$

Protestování nastavíme na x_2, x_3, x_4 takové hodnoty, aby

$\frac{df}{dx_1} = 1$; potom je log. obvod schopen přenášet změny ze vstupu x_1 na výstup.

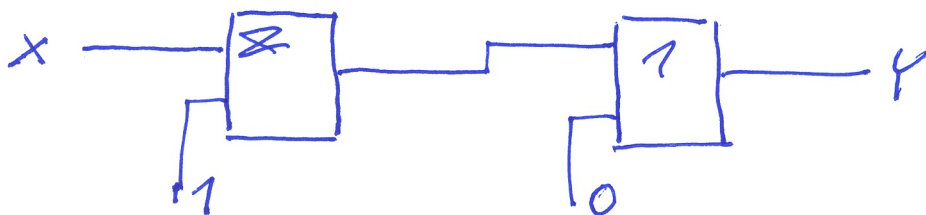
Postup testování:

- určíme derivaci $\frac{df}{dx_i}$
- vypočítáme hodnoty x_i , $x \neq x_i$,
pro které $\frac{df}{dx_i} = 1$
- na vstup x_i připojíme 1 a 0 a porovnáme očekávanou a skutečnou hodnotu na výstupu.

b) strukturální testy

cesta... posloupnost ~~sig~~ log. členů
a spojů

citlivá cesta... je schopná přenášet
změny ze začátku na konec

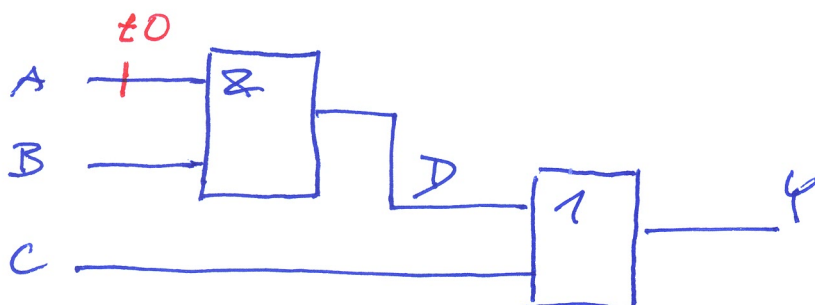


Metoda citlivé cesty:

- sestavíme citlivou cestu z místa předpokládané poruchy na výstup
- do testovaného bodu připojíme opačnou hodnotu, než je typ poruchy (0 pro t1, 1 pro t0).

- porovnáme předpokládanou a skutečnou hodnotu na výstupu
- udělíme další body a poručky, odhalíme tímto testem.

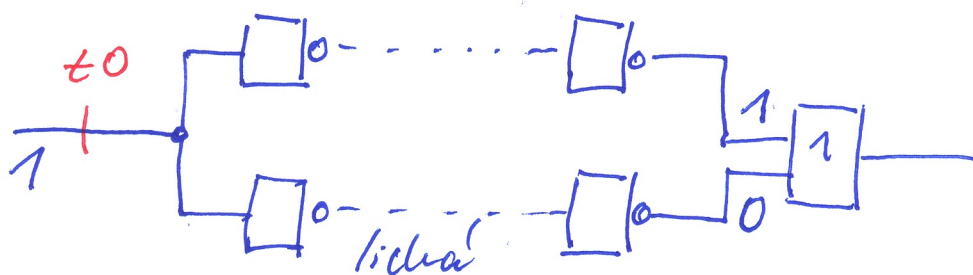
Průklad :



A	B	C	Y	A	B	C	D	Y
1	1	0	1	t0	t0		t0	t0
0	0	1	1			t0		t0
0	1	0	0	t1		t1	t1	t1
1	0	0	0		t1	t1	t1	t1

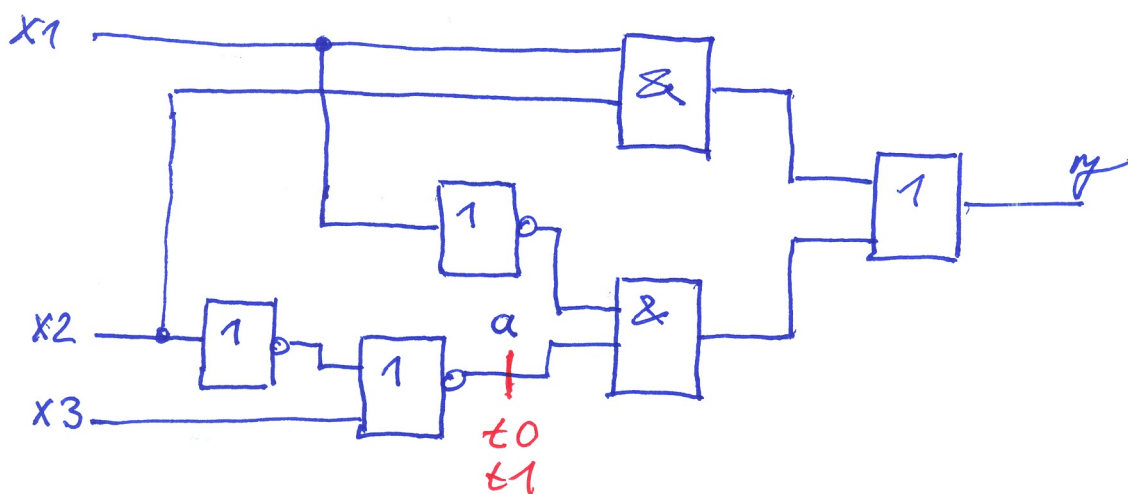
Věta: test kombinačního obvodu je úplný, pokud testuje primární vstupy a body za rozvětvením signálu.

Problém rekurentní cesty:
sada



je-li v každé větvi jiná parita
negací, musíme jednu z cest
zablokovat.

Použití logické derivace k sestavení
testu:



$$a = \overline{x_2 + x_3} = x_2 \cdot \overline{x_3}$$

$$y = a \cdot \overline{x_1} + x_1 \cdot x_2$$

$$\frac{dy}{da} : \begin{array}{c|cccc} & \overline{x_1} & x_1 & \overline{x_2} & x_2 \\ \hline a & 1 & 1 & 1 & 1 \end{array} \Rightarrow \begin{array}{c|cccc} & \overline{x_1} & x_1 & \overline{x_2} & x_2 \\ \hline & 1 & 1 & 0 & 0 \end{array}$$

$$\frac{dy}{da} = \overline{x_1}$$

Pro test na to v bodě a:

$$\begin{aligned} \frac{dy}{da} = \bar{x}_1 = 1 \\ a = x_2 \cdot \bar{x}_3 = 1 \end{aligned} \quad \left\{ \begin{aligned} \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 &= 1 \\ x_1 &= 0 \\ x_2 &= 1 \\ x_3 &= 0 \end{aligned} \right.$$

Pro test na t1 v bodě a:

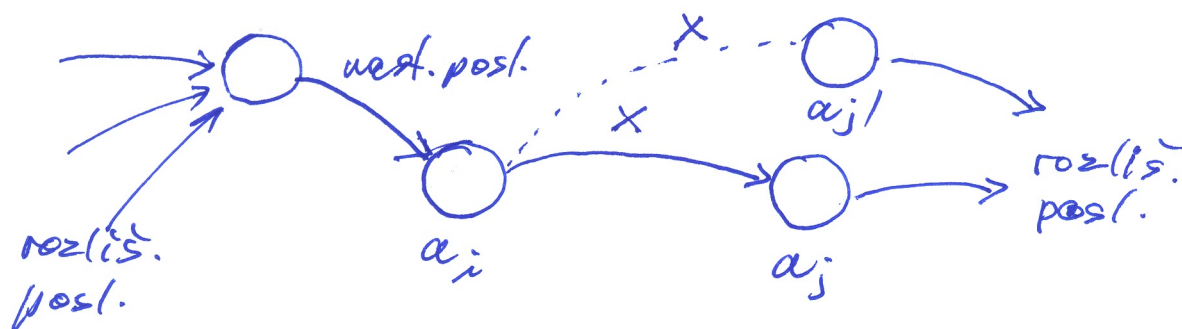
$$\begin{aligned} \frac{dy}{da} = \bar{x}_1 = 1 \\ a = x_2 \cdot \bar{x}_3 = 0 \end{aligned} \quad \left\{ \begin{aligned} \bar{x}_1 \cdot \overline{x_2 \cdot x_3} &= 1 \\ \bar{x}_1 \cdot (\bar{x}_2 + x_3) &= 1 \\ \bar{x}_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_3 &= 1 \end{aligned} \right.$$

$$x_1 = 0 ; x_2 = 0 ; x_3 = x$$

$$x_1 = 0 ; x_3 = 1 ; x_2 = x$$

Testování sekvencí obvodů

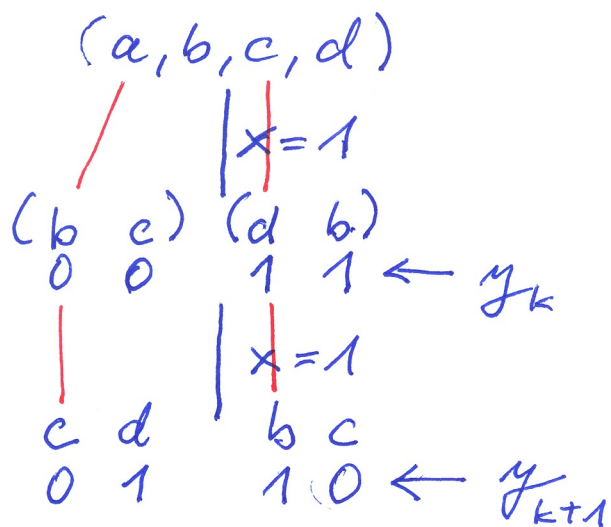
a) funkční testy



ověřují se předchozí podle tabulky přechodů.

1. Aplikací rozliš. posloupnosti se sekv. obvod uvede do známého stavu
2. Aplikací uast. posl. se sekv. obvod uvede do stavu a_i
3. Aplikuje se třetí testu
4. Použitím rozliš. posloupnosti se ověří dosažení stavu a_j

	x_0	x_1
a	a, 0	b, 0
b	a, 0	c, 0
c	c, 0	d, 1
d	d, 1	b, 1

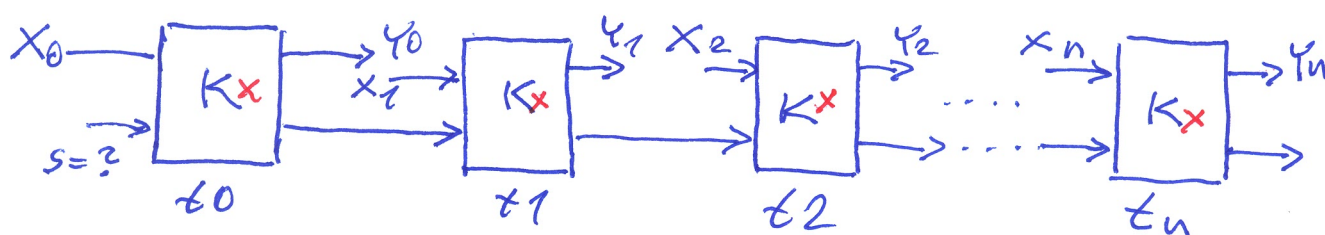
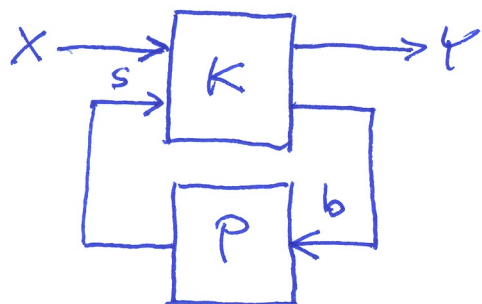


y_k	y_{k+1}	vých. stav
0	0	a
0	1	b
1	1	c
1	0	d

b) Strukturni testy

71

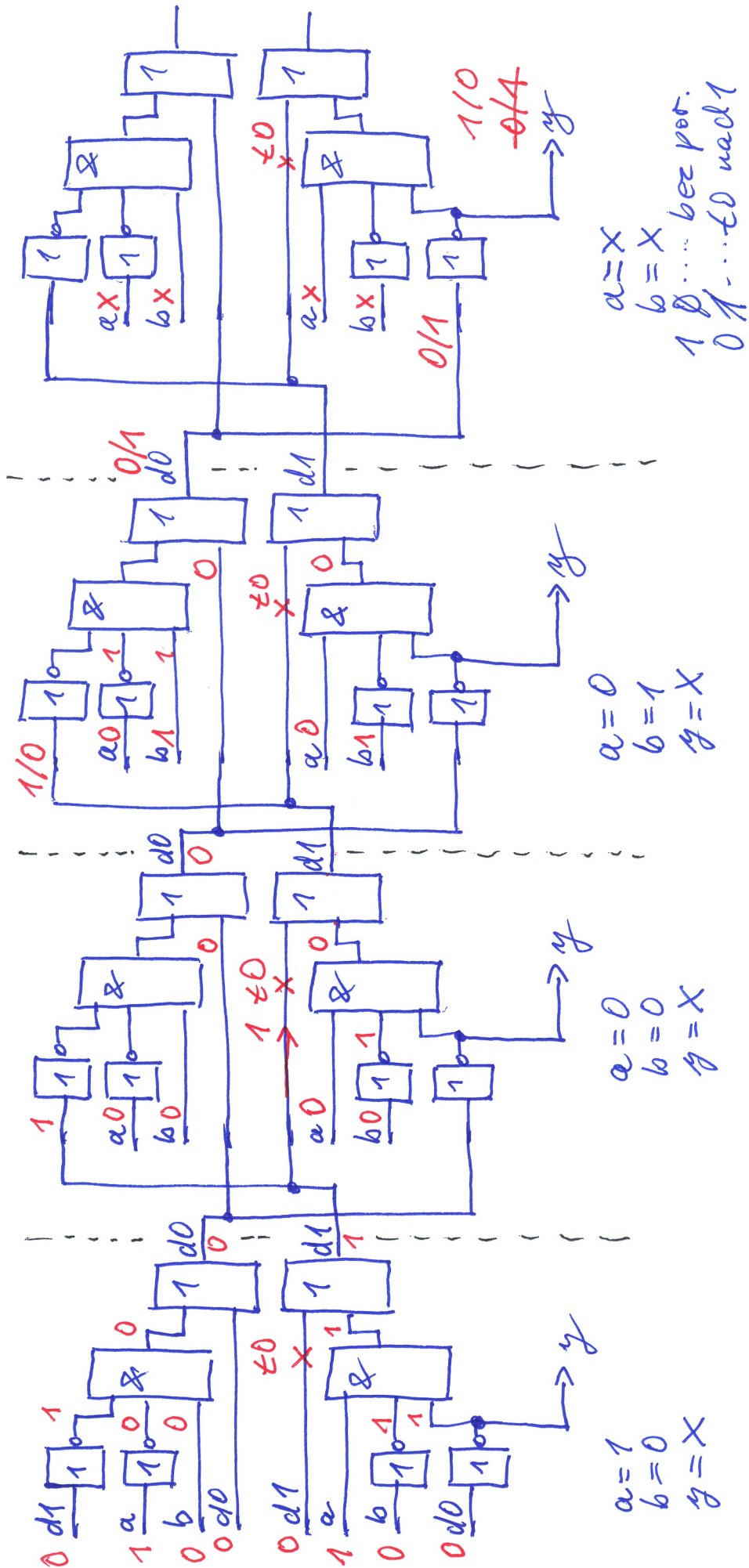
Počívaj se formálnu vzájomnú spätosť
vzťahy ... vytvorí se iteratívny obvod



Príklad: sekvenčný obvod pre učenie!
 $a \geq b$

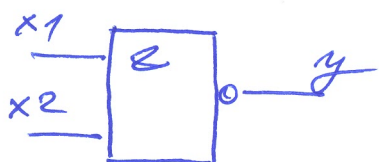
a	0	0	0	1	1	0	0	-x-...
b	0	0	0	1	1	0	1	-x-...
y	1	1	1	1	1	1	0	0-...

$$\begin{aligned} s_0 &= \overline{s_1} \cdot \overline{a} \cdot b + s_0 \\ s_1 &= \overline{s_0} \cdot a \cdot \overline{b} + s_1 \\ y &= s_0 \end{aligned}$$



D-algoritmus

1. Singulárny' pokyť log. čenŕ



$$y = f(X)$$

$$g(X, y) \dots g = 1, \text{ je-li } y = f(X)$$

$y=1$	1	1	0	1
$y=0$	0	0	1	0
y				

 \Rightarrow

singulárny' pokyť
log. čenŕ:

x_1	x_2	y	
1	1	0	s1
0	X	1	s2
X	0	1	s3

2. Přenosová D-krychle log. čenŕ

Přímik řádek sing. pokyť:

$$1 \cap 0 = D \quad 0 \cap 1 = D'$$

$$1 \cap X = X \cap 1 = 1 \cap 1 = 1$$

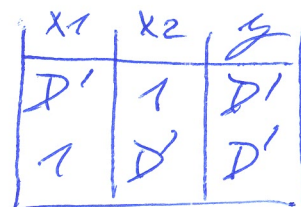
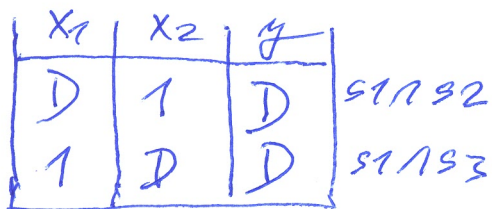
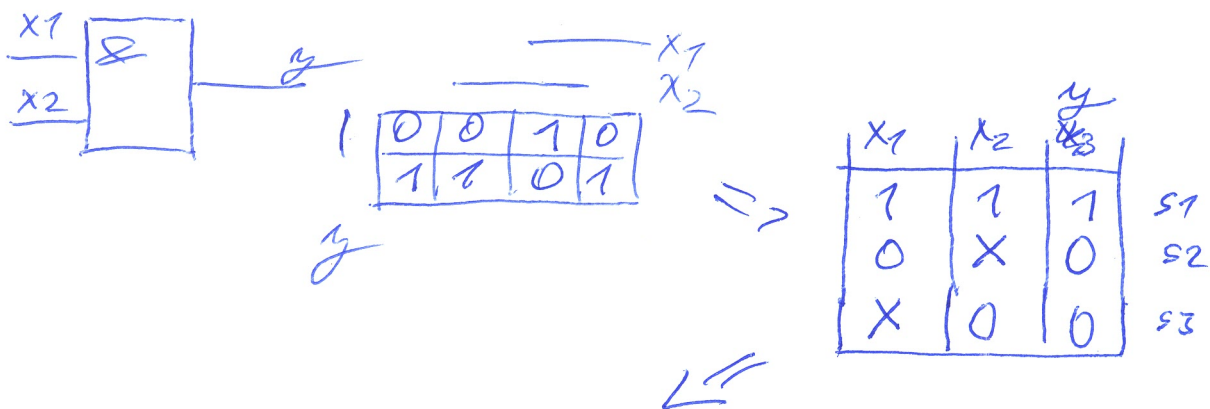
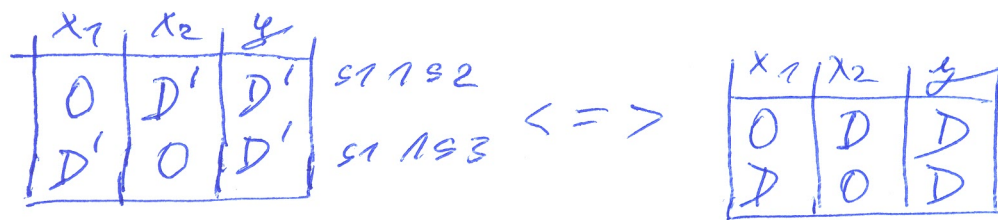
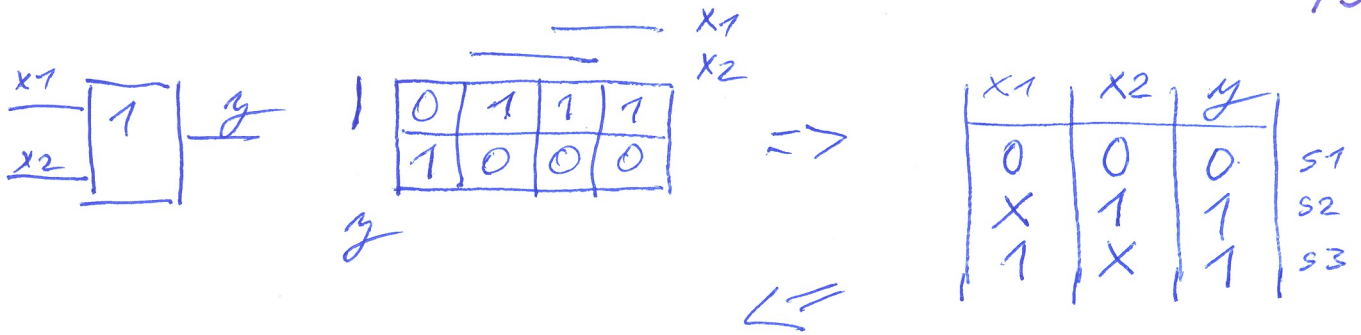
$$0 \cap X = X \cap 0 = 0 \cap 0 = 0$$

$$X \cap X = X$$

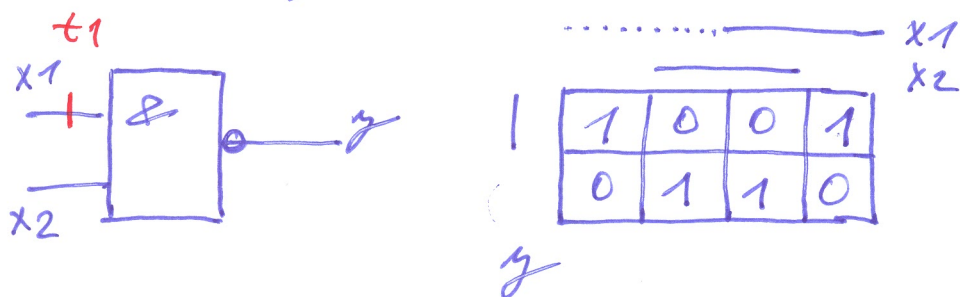
x_1	x_2	y	
D	1	D'	s1 s2
1	D	D'	s1 s3

 \Leftrightarrow

x_1	x_2	y
D'	1	D
1	D'	D



3. D-kychle logického s poruchou



sing. pokytky NAND s t_1 na x_1 :

x_1	x_2	y	
X	0	1	p1
X	1	0	p2

přenosová D-kychle čtem s poruchou:
 přečme řádek sing. p. čtem bez poruchy
 a řádek čtem s poruchou,
 navíc platí $1 \wedge 0 = 0 \wedge 1 \dots$ prázdný
 je-li to v x_1 nebo x_2

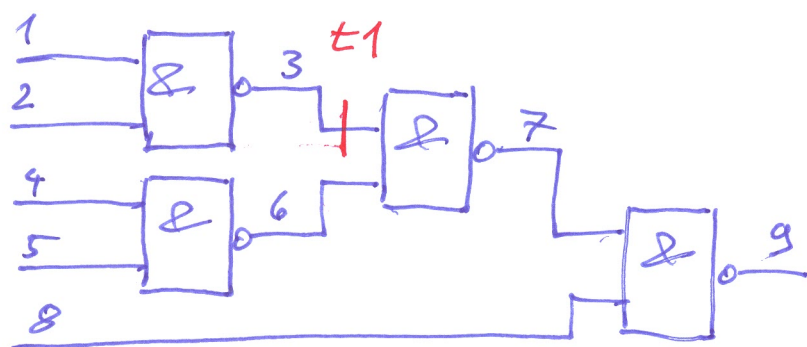
x_1	x_2	y	
1	0	1	s1 \wedge p1
0	1	0	s2 \wedge p2
X	0	1	s3 \wedge p2



takže D-kychle čtem NAND s
 poruchou t_1 na x_1 :

x_1	x_2	y
0	1	D

0 \rightarrow 1



Karta sing. polozky zapojení:

1	2	3	4	5	6	7	8	9
1	1	0						
0	X	1						
X	0	1						
			1	1	0			
			0	X	1			
			X	0	1			
		1			1	0		
		0			X	1		
		X			0	1		
						1	1	0
						0	X	1
						X	0	1

x1	x2	y
1	1	0
0	X	1
X	0	1

NAND

Přenosová D-ky chle zapojení:

1	2	3	4	5	6	7	8	9
D	1	D'						
1	D	D'						
			D	?	D'			
			1	D	D'			
		D			1	D'		
		1			D	D'		
						D	1	D'
						1	D	D'

x1	x2	y
D	1	D'
1	D	D'

1	2	3	4	5	6	7	8	9
D	1	D'						
1	D	D'						
			D	1	D'			
			1	D	D'			
		0			1	D		
						D	1	D'
						1	D	D'

x1	x2	z
0	1	D

D-logické zapojení s čtením s poruchou

4. Hledání cesty z místa poruchy na výstup:

Průniky řádek tabulky, přičemž platí:

	0	1	x	D	D'
0	0	0	0	n	n
1	0	1	1	n	n
x	0	1	x	D	D'
D	n	n	D	n	n
D'	n	n	D'	n	n

n ... jsou-li v průniku jen n a ne λ ,
pak $D \cap D = D$, $D' \cap D' = D'$

λ ... jsou-li v průniku jen členy λ
a ~~řádny~~ n , provede se negace
druhého čtení průniku a potom
se provede průnik

$n + \lambda$... jsou-li v průniku členy n i
 λ současně, je průnik prázdný.

1	2	3	4	5	6	7	8	9	
		0			1	D			p1
						D	1	D'	p2
						1	D	D'	p3
		0			1	M	1	D'	$p1 \wedge p2$
		0			1	D	1	D'	↓
		0			1	~	1	D'	$p1 \wedge p3$

→ prázdny' prámek

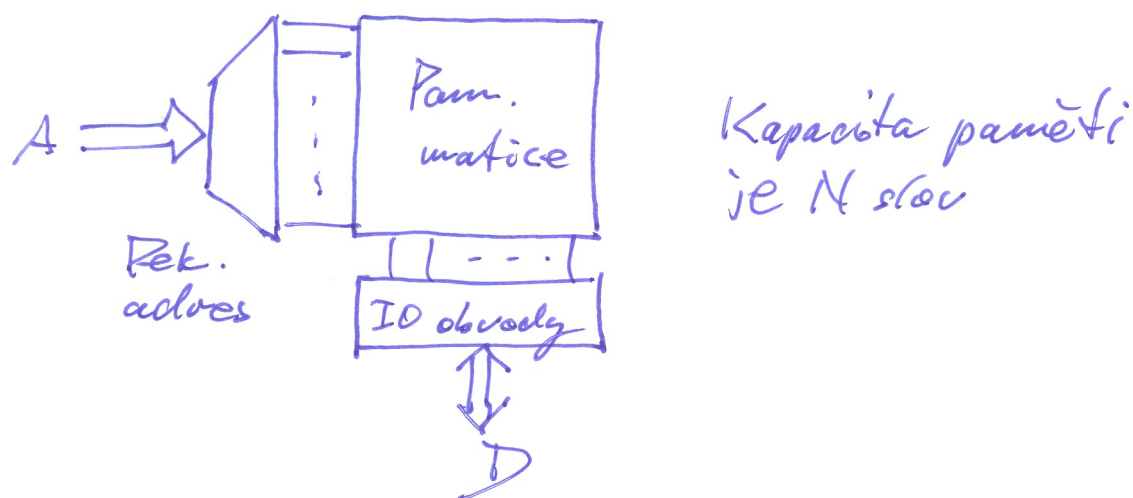
5. Zpetné hľadanie hodnot na primárných vstupoch podľa výstupu mapy sing. pokroky:

1	2	3	4	5	6	7	8	9
		0			1	D	1	D'
		↓	x	0	↓	D	1	D'
<u>1</u>	<u>1</u>	<u>0</u>	<u>X</u>	<u>0</u>	1	D	<u>1</u>	<u>D'</u>

→ testovací vektor

→ projev chyby na výstupu
 0... bez poruchy
 1... s poruchou

D-algorithmus je výpočetně náročný.
Existují různá vylepšení s menší
náročností.

a) Marching patterns (MARCPAT)

1. Do celé matice se zapíše pořadí 0
2. Pro každé slovo:
 - zapíše se 1
 - čte se 1
3. Pro každé slovo:
 - zapíše se 0
 - čte se 0

Test má složitost $O(N)$

b) Galloping patterns (GALPAT)

1. Do celé paměti se zapíše pořadí 0
2. Pro každé slovo:
 - zapíše 1
 - čte se slova $i, i+1, i, i+2, \dots$
 $\dots i+N$

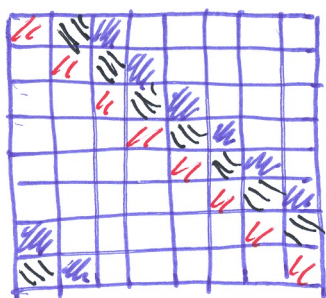
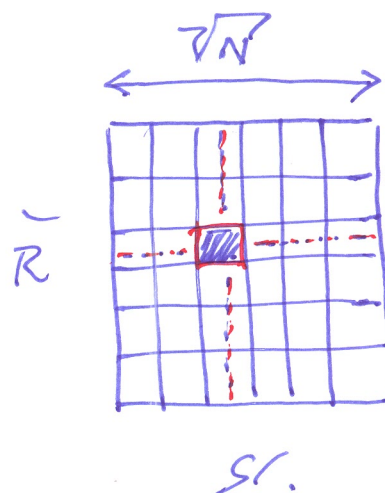
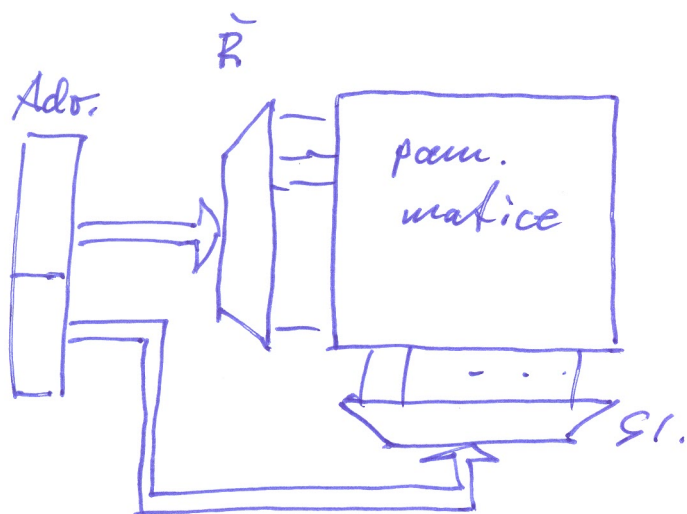
Složitost testu GALPAT je $O(N^2)$

Např. 1M pamět; je-li 1 krok testu řádově 10^{-7} sec :

$$(10^6)^2 \cdot 10^{-7} = 10^5 \text{ [sec]}$$

c) Galloping diagonal

Vyžaduje znalost uspořádané pam.
matice



Počet diagonál je \sqrt{N}^2

Postup:

1. Zapišete se poradí 0 do celé pam.
2. Pro každou diagonálu:
 - zapišete se diagonálu 1
 - testuje se celá pam. matice
3. Opakujte se pro opačné log. hodnoty

Složitost testu je $O(N^{\frac{3}{2}})$

Např. je-li test proveden za 100 ns
(10^{-7} sec).

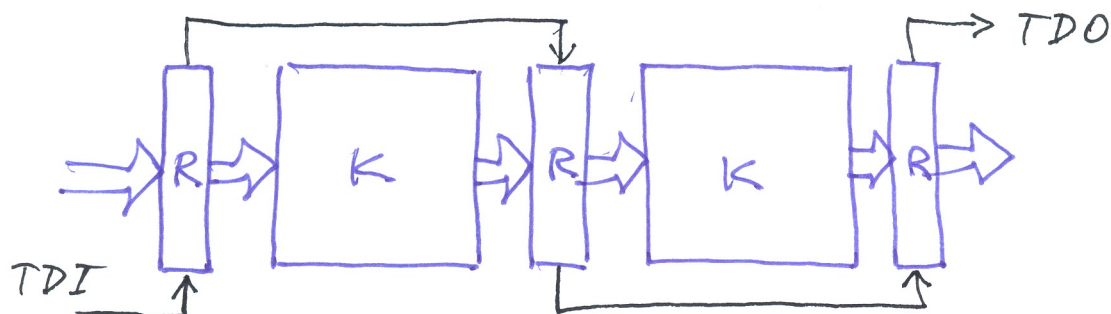
a kap. paměti 1M slov,

je trvání testu =

$$10^6 \cdot 10^3 \cdot 10^{-7} = 10^2 \text{ sec.}$$

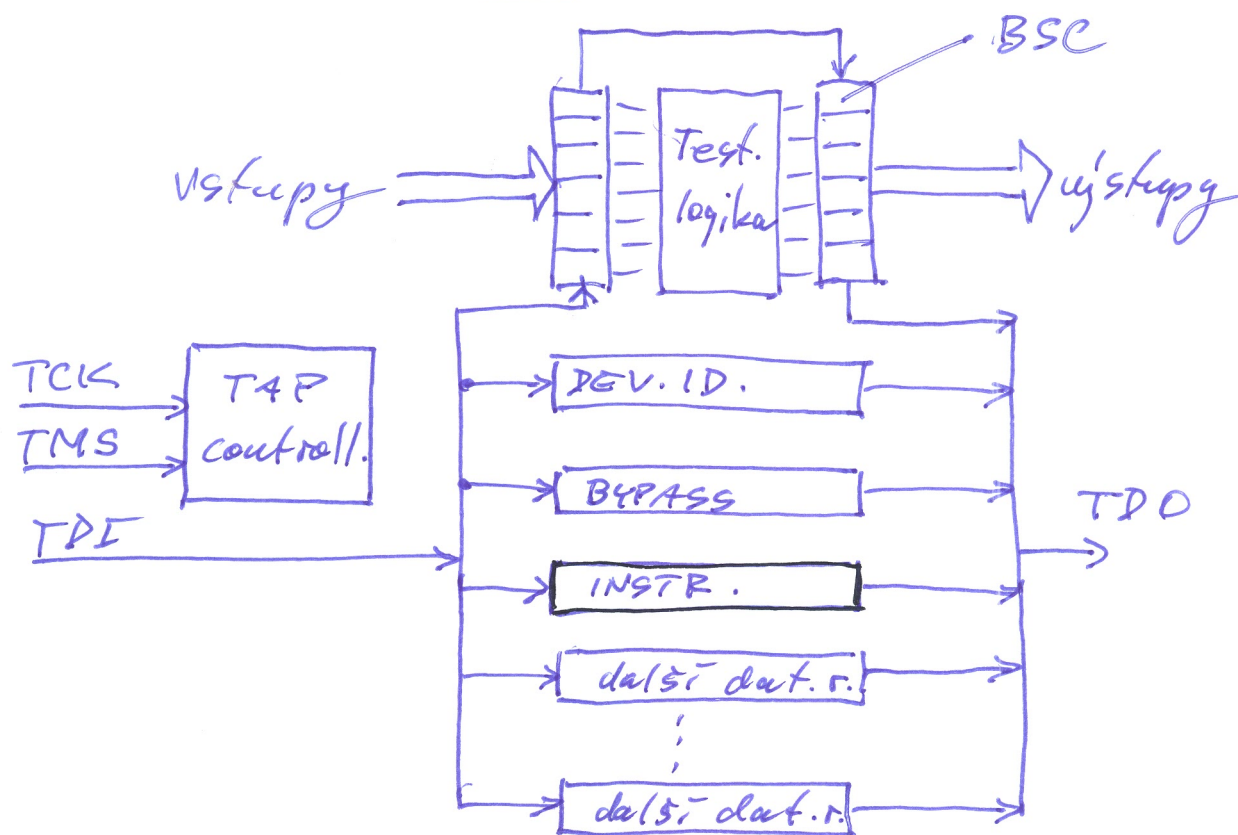
Na'voh pro snadenou diag nostiku

Architektura RTL

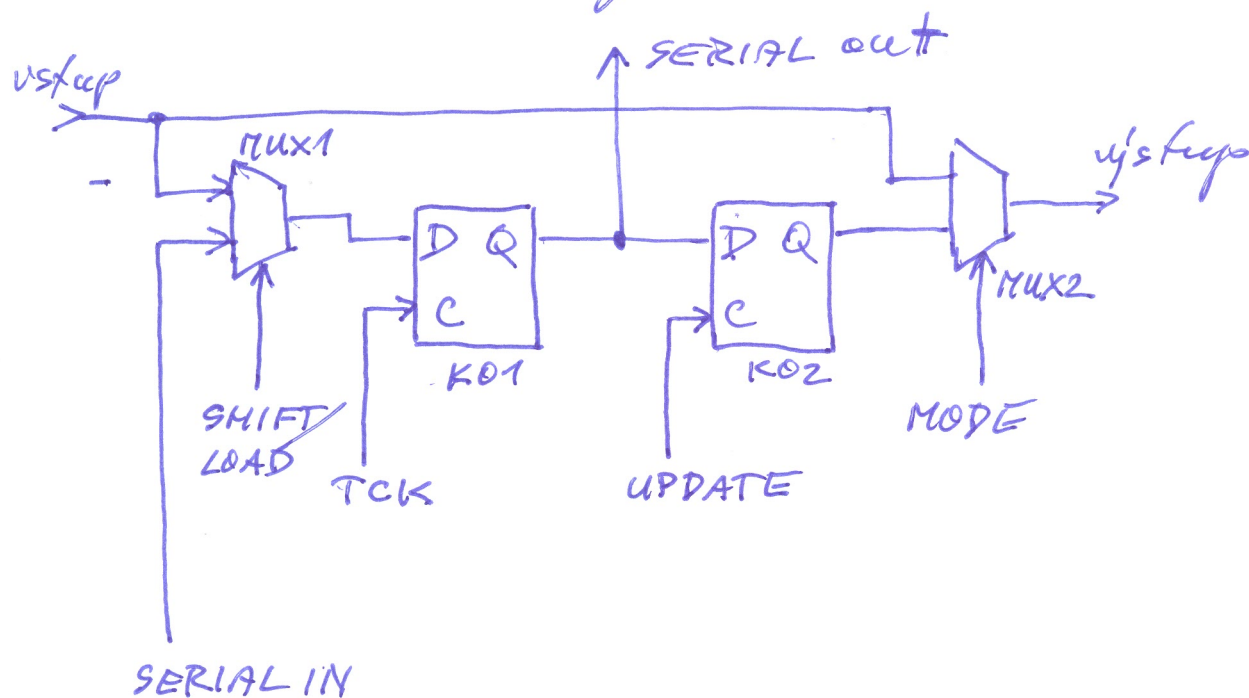


Rozkman' JTAG

IEEE 1149



Buňka Boundary Scan Cell :



Základní operace s BSC :

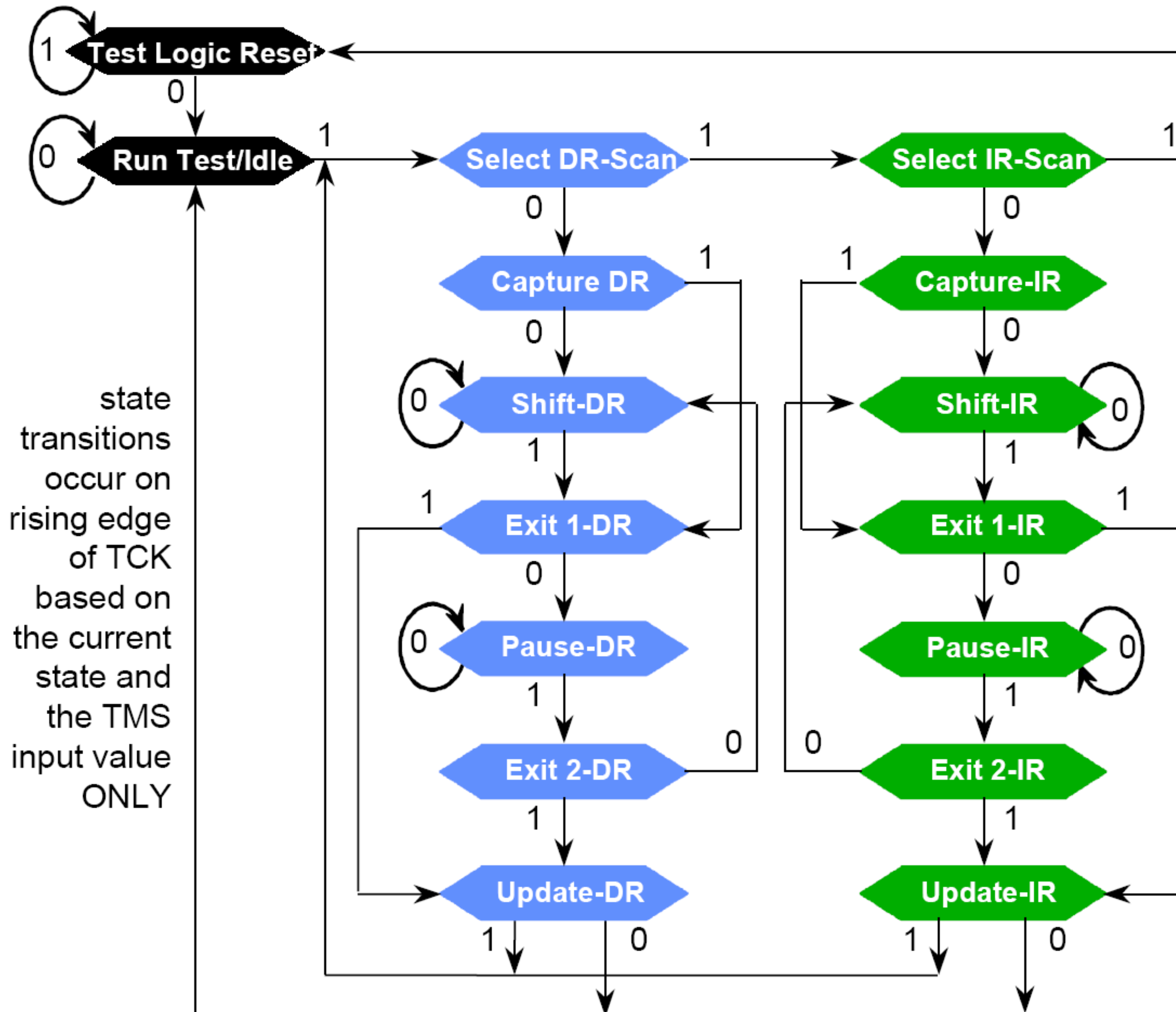
- přečtení hodnot na vstupech (CAPTURE)
- posun klop. obvodů (SHIFT)
- přepis posuv. registrů (K01) do pam. části (K02) -- (UPDATE)

Přepnutí výstupního multiplexoru instrukcí zapsaných v instr. registru lze přepnout MUX2.

Další využití rozhraní JTAG :

- programování paměti Flash v MCU
- programování FPGA
- přístup k laděním vybavení MCU

The Test Access Port Controller



■ 16-state TAP provides 4 major operations:

- RESET
- RUN-TEST
- SCAN-DR
- SCAN-IR

■ Scans consist of 3 primary steps:

- CAPTURE
- SHIFT
- UPDATE

