Jednokrotny wybór

pytania losowane

dosłownie powiedziałby używać notatek

30 pytań

45 minut

W13 – Testowanie układów cyfrowych

Henryk Maciejewski

Jarosław Sugier

Uszkodzenia i modele błędów

1. Modele błędów w układach cyfrowych

2. Generowanie wektorów testowych dla układów kombinacyjnych
   - Metody tablicowa
   - Metoda różnic boolowskich
   - Metoda pobudzenia ścieżek

B. Wilkinson, Układy cyfrowe, WKŁ.

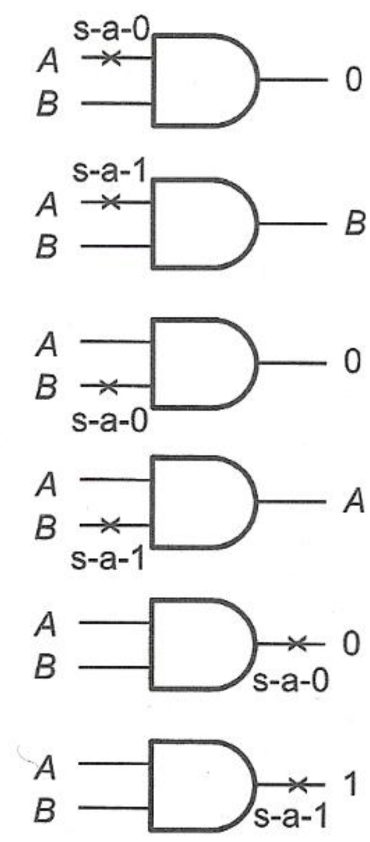
Modele błędów w układach cyfrowych

Rozważamy uszkodzenia prowadzące do <u>błędów logicznych</u> układu.

Zakładamy model: pojedynczy błąd „sklejenia”:

stuck-at-0, stuct-at-1

Lokalizacje błędów: każdy sygnał wejściowy, wewnętrzny, wyjściowy



Generowanie wektorów testowych - metoda tablicowa

Rozważamy układ:

$$z = F(x_1, x_2, x_3, x_4) = (x_1 + x_2 x_3)\overline{(x_2 x_3 + x_4)}$$

uszkodzenie α = s-a-0 na wejściu x_4

z = wartość funkcji bez uszkodzenia

z^{α} = wartość funkcji z uszkodzeniem α

Generowanie wektorów testowych - metoda tablicowa

Rozważamy układ:

$$z = F(x_1, x_2, x_3, x_4) = (x_1 + x_2 x_3)\overline{(x_2 x_3 + x_4)}$$

Uszkodzenie wykrywają testy:

$x^1 = (1{,}0{,}0{,}1) = x_1\,\overline{x_2}\,\overline{x_3}\,x_4$
$x^2 = (1{,}0{,}1{,}1) = x_1\,\overline{x_2}\,x_3\,x_4$
$x^3 = (1{,}1{,}0{,}1) = x_1\,x_2\,\overline{x_3}\,x_4$

x1	x2	x3	x4	z	z^α	xor(z, z^α)
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	1	0
1	0	0	1	0	1	**1**
1	0	1	0	1	1	0
1	0	1	1	0	1	**1**
1	1	0	0	1	1	0
1	1	0	1	0	1	**1**
1	1	1	0	0	0	0
1	1	1	1	0	0	0

Metoda różnic boolowskich (*boolean difference*)

Różnica boolowska funkcji $F(x_1, x_2, \dots, x_n)$ wzgl. zmiennej $x_j, 1 \leq j \leq n$:

$$\frac{\partial F}{\partial x_j} = F(x_1, x_2, \dots, 0, \dots, x_n) \oplus F(x_1, x_2, \dots, 1, \dots, x_n) = F_{\overline{x_j}} \oplus F_{x_j}$$

j-ta pozycja

- Funkcja n-1 zmiennych
- $\frac{\partial F}{\partial x_j} = 1$ → dla danego wektora wejściowego $(x_i, i \neq j)$ wartość *F* <u>zależy</u> od zmiennej x_j
- $\frac{\partial F}{\partial x_j} = 0$ → dla danego wektora wejściowego $(x_i, i \neq j)$ wartość *F* <u>nie zależy</u> od x_j

Metoda różnic boolowskich (*boolean difference*)

Poprzedni przykład: $z = F(x_1, x_2, x_3, x_4) = (x_1 + x_2x_3)\overline{(x_2x_3 + x_4)}$

$$F_{x_4} = (x_1 + x_2x_3)\overline{(x_2x_3 + 1)} = 0$$

$$F_{\overline{x_4}} = (x_1 + x_2x_3)\overline{(x_2x_3 + 0)} = (x_1 + x_2x_3)\overline{x_2x_3} = x_1\overline{x_2x_3}$$

$$\frac{\partial F}{\partial x_4} = F_{\overline{x_4}} \oplus F_{x_4} = x_1\overline{x_2x_3}$$

→ stąd wektory testowe, uwzględniając warunek, aby x_4 odróżniało się od uszkodzenia α

Wynik metody tablicowej:

$x^1 = (1,0,0,1) = x_1 \overline{x_2}\, \overline{x_3}\, x_4$

$x^2 = (1,0,1,1) = x_1 \overline{x_2}\, x_3\, x_4$

$x^3 = (1,1,0,1) = x_1\, x_2\, \overline{x_3}\, x_4$

$$\frac{\partial F}{\partial x_4} = F_{\overline{x_4}} \oplus F_{x_4} = x_1 \overline{x_2 x_3} = x_1(\overline{x_2} + \overline{x_3}) = x_1\overline{x_2}(x_3 + \overline{x_3}) + x_1\overline{x_3}(x_2 + \overline{x_2})$$
$$= x_1\overline{x_2}x_3 + x_1\overline{x_2}\,\overline{x_3} + x_1 x_2 \overline{x_3}$$

stąd wektory testowe (przy x_4 = 1, aby odróżniało się od uszkodzenia α)

$x^1 = x_1 \overline{x_2}\, \overline{x_3}\, x_4 = (1,0,0,1)$

$x^2 = x_1 \overline{x_2}\, x_3\, x_4 = (1,0,1,1)$

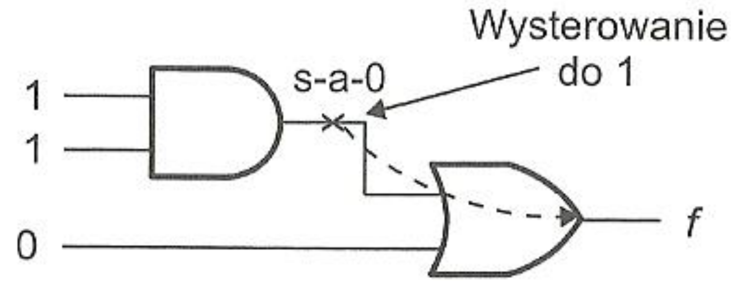
$x^3 = x_1\, x_2\, \overline{x_3}\, x_4 = (1,1,0,1)$

Wektory testowe dla błędów @ x_j uzyskuje się jako:

$x_j \frac{\partial F}{\partial x_j}$ dla s-a-0 $\qquad\qquad$ $\overline{x_j} \frac{\partial F}{\partial x_j}$ dla s-a-1

Metoda pobudzenia ścieżek (*path sensitization*)

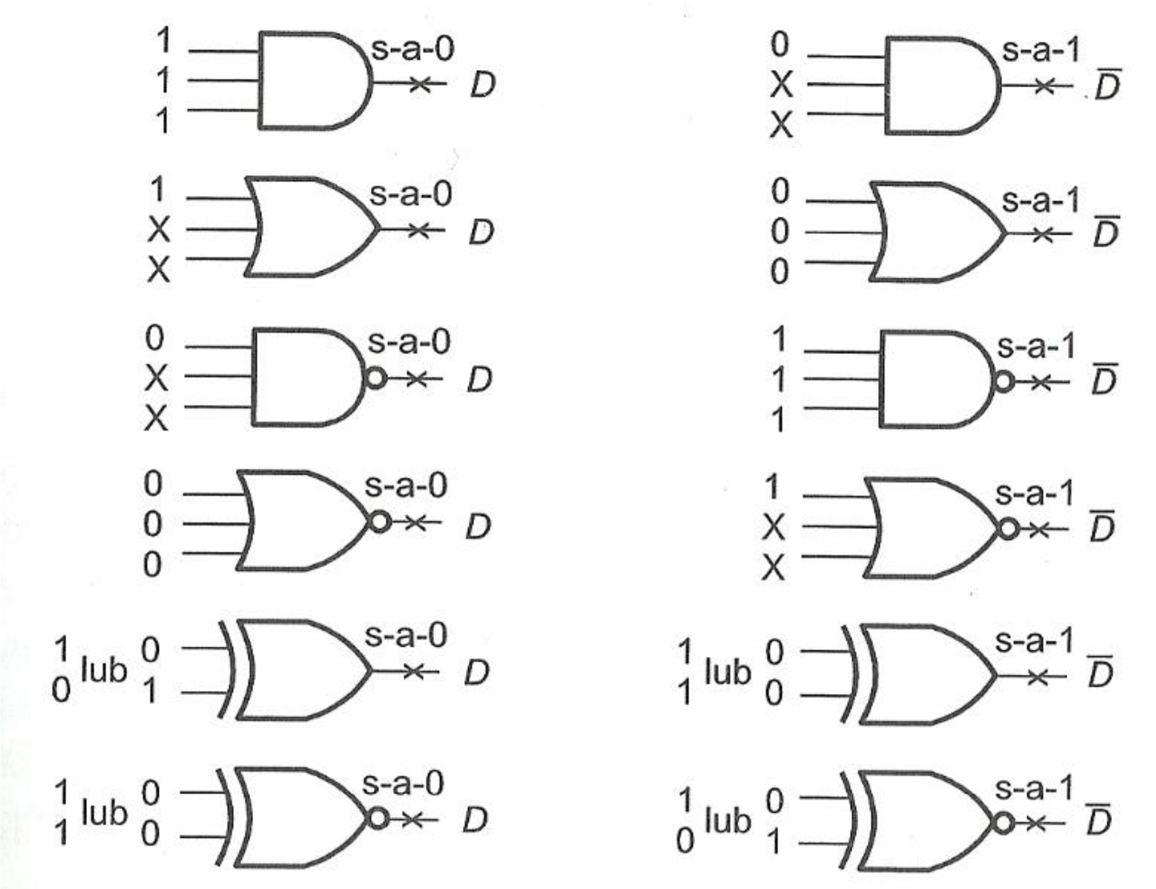
Test = *activate* + *propagate*



Activate: wymuszamy w danym punkcie układu wartość przeciwną do błędu badanego w tym punkcie
(np. wymuszamy 1, gdy badamy s-a-0)

Propagate: ustawiamy wartości sygnałów na pozostałych bramkach w układzie tak żeby spowodować propagację sygnału błędu (sygnału D / $\overline{D}$) do zewnętrznego wyjścia układu.

<u>Aktywowanie</u> podstawowych bramek

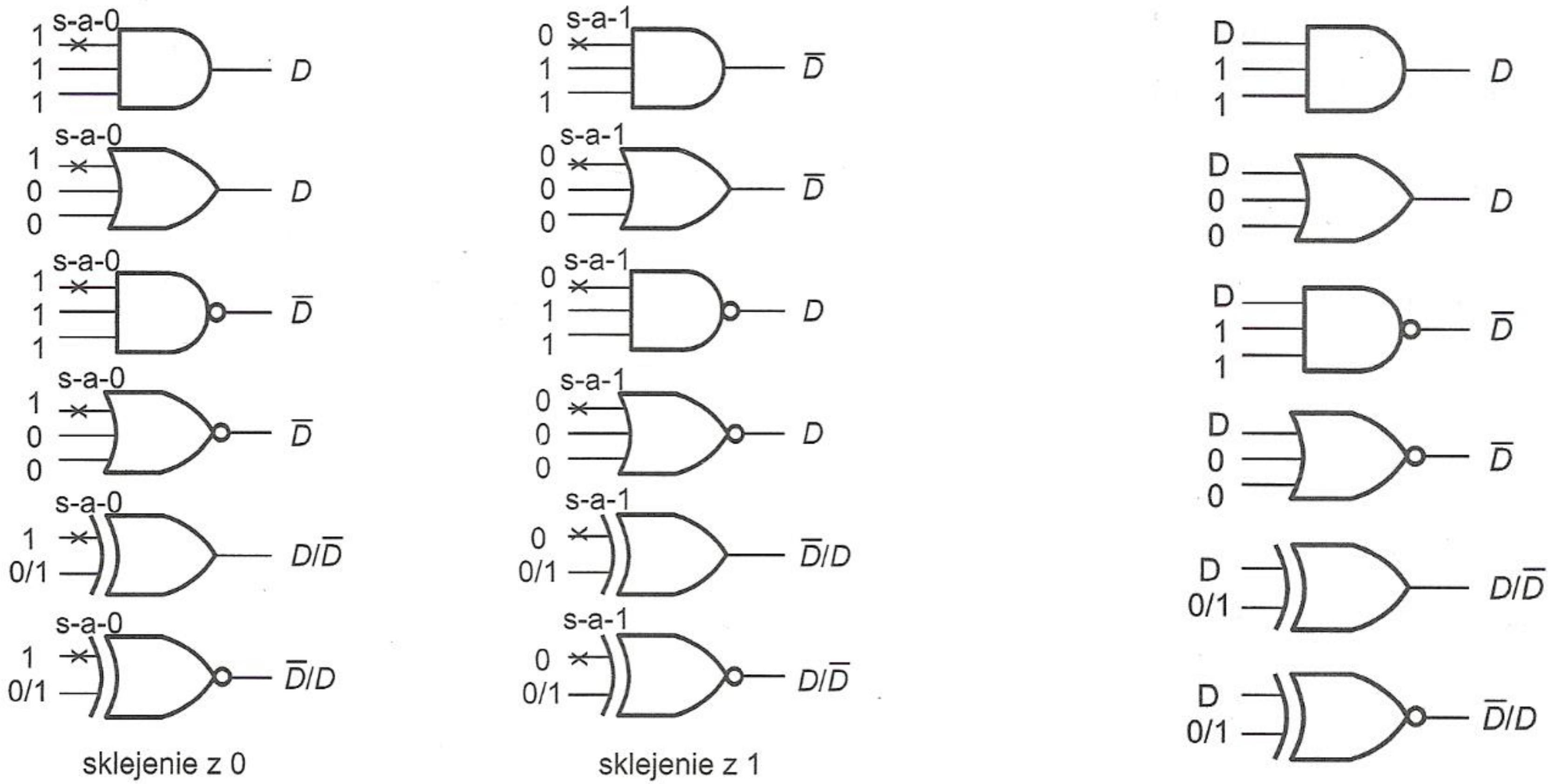


Symbol D:

D: sygnał przy niewystępowaniu błędu ma mieć wartość 1

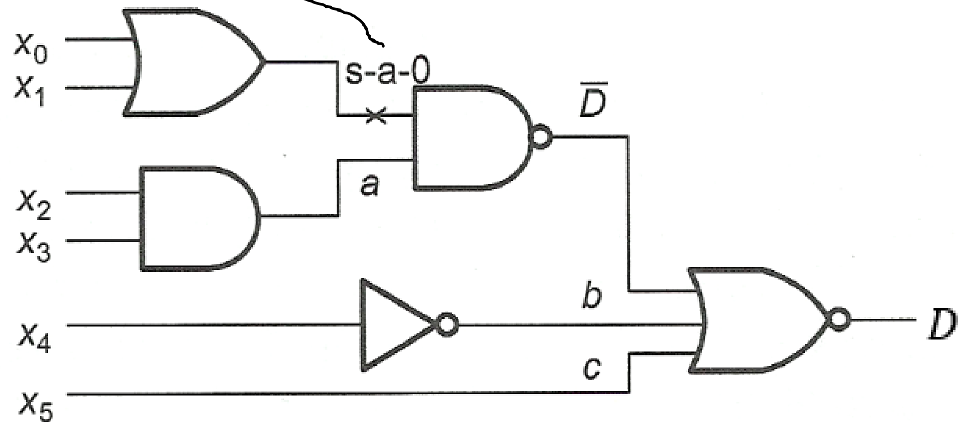
$\overline{D}$: sygnał przy niewystępowaniu błędu ma mieć wartość 0

<u>Propagowanie</u> sygnału D / $\overline{D}$ do wyjścia układu



Bramki XOR i XNOR zawsze propagują D (jako D albo $\overline{D}$)

Przykład



by było inne niż błąd

Activate

$$x_0 + x_1 = 1 \rightarrow \qquad x_0, x_1 \in \{01,10,11\}$$

Propagate

$$a = x_2 \cdot x_3 = 1 \qquad x_2 = x_3 = 1 = x_4$$

$$b = 0 \qquad \cancel{x_4 = 1}$$

$$c = 0 \qquad x_5 = 0$$

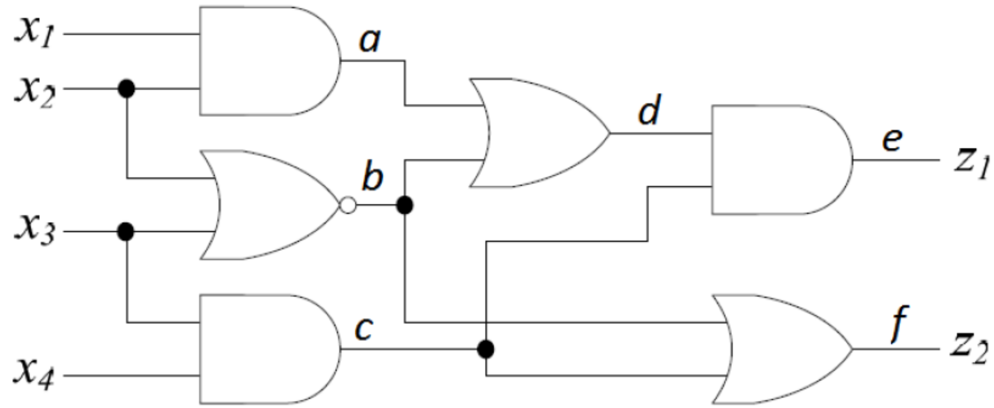
Stąd zbiór testów wykrywających uszkodzenie:

(0,1,1,1,1,0)

(1,0,1,1,1,0)

(1,1,1,1,1,0)

Przykład 2



s-a-0 @ b

Activate

$$b = \overline{x_2 + x_3} = 1 \rightarrow \qquad x_2 = x_3 = 0$$

Propagate – ścieżka do z_1

$$a = 0 = x_1 \cdot x_2 \qquad x_1x_2 \in \{00,01,10\}$$

$$c = 1 = x_3 \cdot x_4 \qquad x_3 = 1, x_4 = 1$$

Sprzeczność: $x_3 = 0$ i $x_3 = 1$ $\rightarrow$ błąd nie może być wykryty na wy z_1

Propagate – ścieżka do z_2

$$c = 0 = x_3 \cdot x_4 \qquad x_3x_4 \in \{00,01,10\}$$

Zbiór testów wykrywających uszkodzenie:

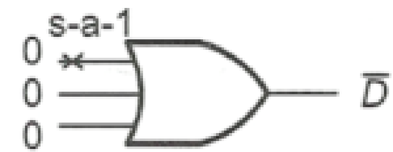
(X,0,0,X)

Błędy równoważne w bramce

Błędy równoważne – wykrywane przez te same wektory testowe (zbiory wektorów testowych wykrywających każdy z błędów równoważnych są równe)

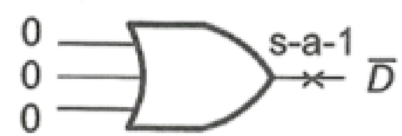
Przykład – bramka OR:

Błąd s-a-1 na każdym z wejściu – testuje wektor 000



Błędy s-a-1 na każdym z wejść - równoważne

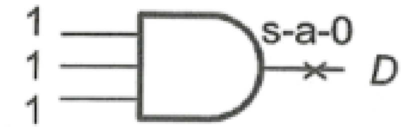
Wektor 000 testuje też błąd s-a-1 na wyjściu

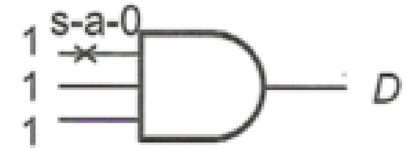


Błędy s-a-1 na każdym z wejść i na wyjściu – równoważne (nierozróżnialne)

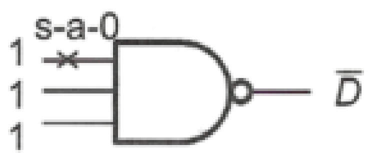
Przykład – bramka AND:

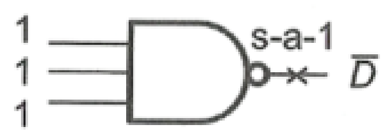
Błędy s-a-0 na każdym z wejść i na wyjściu – równoważne (nierozróżnialne)



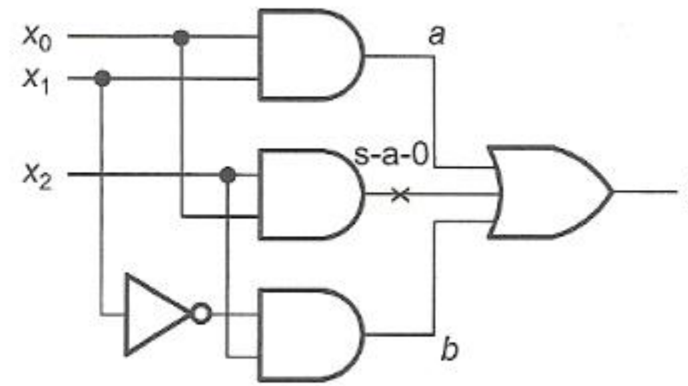


? Błędy równoważne dla bramki NAND i NOR ?





Błędy niewykrywalne



W tym przykładowym układzie błąd s-a-0 jest niewykrywalny (dlaczego?)

Przyczyną niewykrywalnych błędów (pojedynczych) jest nadmiarowość w układzie

Zminimalizowana postać układu – nie ma błędów niewykrywalnych:

$$f = x_0 x_1 + \overline{x_1}\, x_2$$

Realizacja układów nieuproszczonych czasami konieczna (eliminacja hazardów, korzystanie z określonego typu bramek – np. dwuwejściowych) → stosuj dodatkowe wyjścia - wyprowadzenia punktów wewnętrznych (dla testowalności)