

Podstawy techniki cyfrowej

Układy asynchroniczne

Opracował:

R.Walkowiak

Styczeń 2016

Charakterystyka układów asynchronicznych

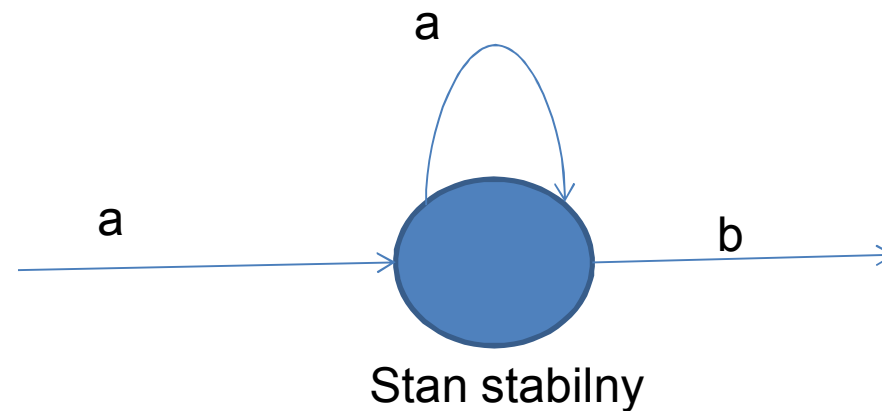
- Brak wejścia: zegarowego, synchronizującego.
- Natychmiastowa - niesynchronizowana reakcja układu potencjalnie na każdą zmianę w wektorze sygnałów wejściowych.
- Nowy stan układu ustala się po opóźnieniu wynikającym z czasów propagacji elementów zastosowanych do realizacji układu.
- Stan przejściowy, niestabilny – to taki, że dla danego wektora wejściowego układ przechodzi tylko tymczasowo (czasy propagacji) do tego stanu.
- Stan stabilny – to taki, że dla określonego wektora wejściowego układ wchodzi i pozostaje w tym stanie.

Stan niestabilny

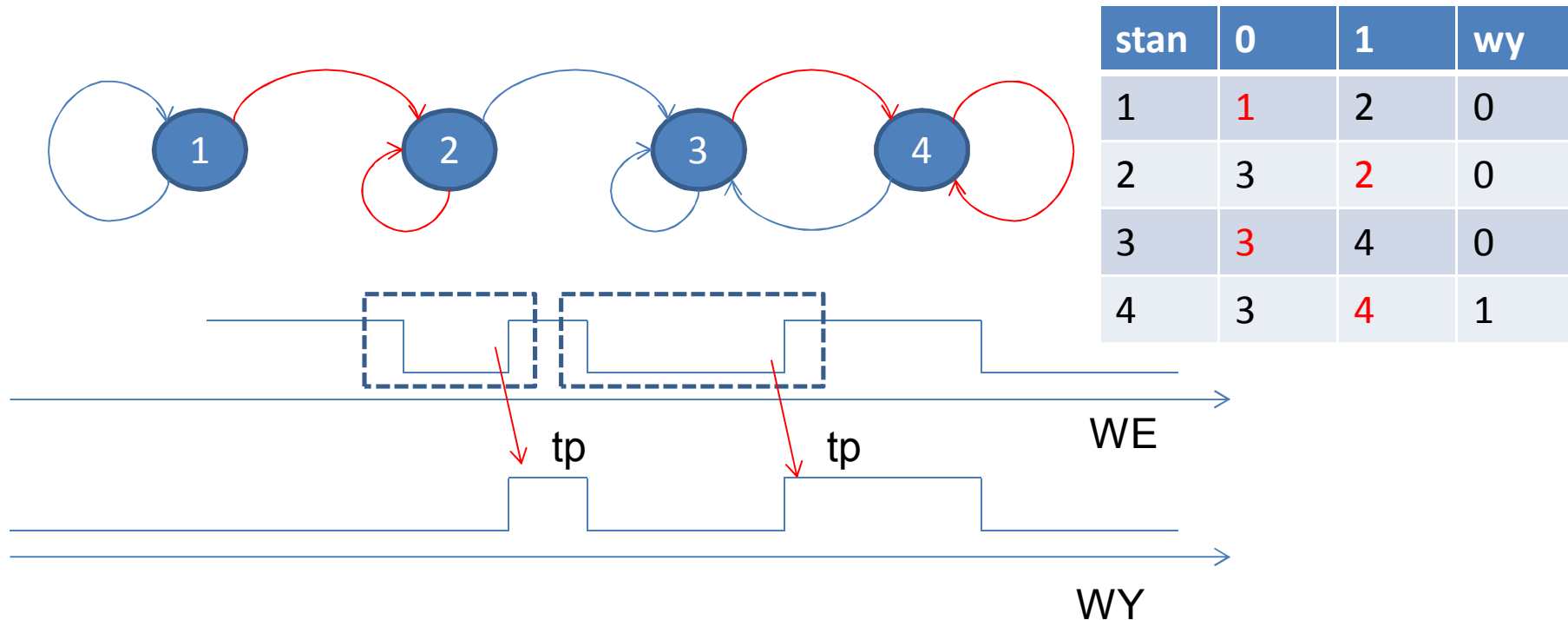
- Stan przejściowy, niestabilny – dla danego wektora wejściowego układ przechodzi tylko tymczasowo (na czas propagacji układów użytych do realizacji układu) do tego stanu a następnie z niego wychodzi



- Stan stabilny – dla określonego wektora wejściowego układ wchodzi i pozostaje w tym stanie.



Przykład automatu asynchronicznego wykrywającego sekwencję bitów 101

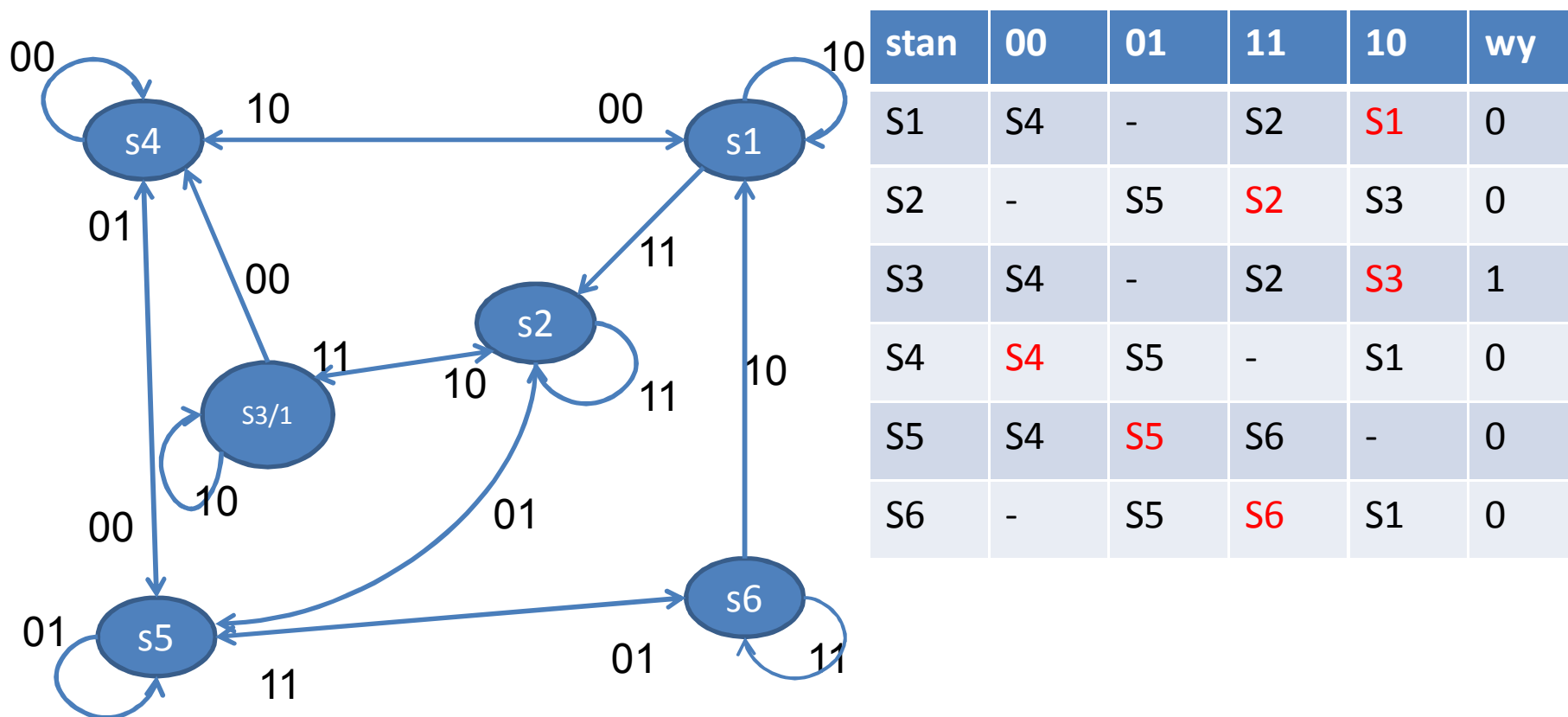


Automat asynchroniczny na schemacie posiada tylko stany stabilne. Przejścia pod wpływem jedynek na wejściu oznaczono kolorem czerwonym w grafie. Po czasie propagacji elementów układu następuje przejście do kolejnego stanu (ewentualna zmiana na wyjściu). Przejścia po czasie propagacji tp.

Układy asynchroniczne

Przykład automatu asynchronicznego

Automat ma 2 wejścia i jedno wyjście, wykrywa sekwencję par bitów 10, 11, 10.
Należy założyć, że jednoczesna zmiana sygnałów wejściowych jest niemożliwa.



Do każdego stanu przechodzimy kombinacją wejść, dla której nowy stan jest stabilny. Ciągi bitowe przy łukach oznaczają warunek wyjścia.

Minimalizacja stanów

stan	00	01	11	10	wy
S1	S4	-	S2	S1	0
S2	-	S5	S2	S3	0
S3	S4	-	S2	S3	1
S4	S4	S5	-	S1	0
S5	S4	S5	S6	-	0
S6	-	S5	S6	S1	0

s2	1,3				
S3	nie	nie			
S4	tak	1,3	Nie		
S5	2,6	2,6	Nie	Tak	
S6	2,6	1,3	nie	Tak	Tak
	s1	S2	s3	s4	s5

„-” oznacza stan dowolny, „tak” oznacza spełniony warunek zgodności

- **Pary zgodne** to takie, że z każdego ze stanów pary następują przejścia do tego samego stanu, lub stanów równoważnych (para stanów: **dowolny** i określony są sobie równoważne).
- **Pary zgodne** stanów to: {S1,S4}, {S4,S5}, {S4,S6}, {S5,S6}.
- Maksymalne klasy stanów zgodnych powstają przez połączenie par i klas stanów zgodnych bez uwzględnienia stanów następnych: {S4,S5,S6}, {S1,S4}.
- Klasa {S4,S5,S6} powstaje z par zgodnych: {S4,S5}, {S4,S6} i {S5,S6}.
- Dalej: szukamy minimalnego zbioru klas zgodnych spełniającego warunek pokrycia i zamknięcia (por. str. następna).

Minimalizacja stanów

- Szukamy minimalnego zbioru klas zgodnych spełniającego warunek pokrycia i zamknięcia.
- **Warunek pokrycia** zbioru klasy zgodnych oznacza pogrupowanie w ramach klas zgodnych **wszystkich stanów pierwotnych** (w grafie przed optymalizacją)
- **Warunek zamknięcia** oznacza przejścia (pod wpływem jednakowych słów wektora wejścia) do tych samych lub równoważnych stanów następnych. Przejścia jednakowe dotyczą wszystkich stanów klasy.
- Uwaga: W wyniku łączenia par w klasy warunek zgodności obowiązujący dla par mógł przestać być spełniony dla klasy (niespełniony warunek zamknięcia) -np. dla klasy stanów złożonych ze stanów pochodzących z par stanów równoważnych: jedna para stanów przejście (S5, dowolny) oraz druga para stanów (S6, dowolny) (przejścia pod wpływem jednakowego słowa wejściowego) warunek zamknięcia jest spełniony dopiero w przypadku równoważności stanów 5 i 6.

stan	00	01	11	10	wy
S1	S4	-	S2	S1	0
S2	-	S5	S2	S3	0
S3	S4	-	S2	S3	1
S4	S4	S5	-	S1	0
S5	S4	S5	S6	-	0
S6	-	S5	S6	S1	0

s2	1,3				
S3	nie	nie			
S4	tak	1,3	Nie		
S5	2,6	2,6	Nie	Tak	
S6	2,6	1,3	nie	Tak	Tak
	s1	S2	s3	s4	s5

Klasy stanów zgodnych {S4,S5,S6}, {S1,S4}.

- Klasa {S4,S5,S6} powstaje z par zgodnych: {S4,S5}, {S4,S6} i {S5,S6}.
- Klasa {S4,S5,S6} spełnia warunek zamknięcia – pod wpływem różnych wektorów wejściowych następuje przejście do stanu S1 lub przejście do stanu {S4,S5,S6}.

Warunek pokrycia i zamknięcia spełniają następujące wariantowe zbiory klas zgodnych (oba zbiory klas są równoliczne)

{S4,S5,S6}, S1,S2,S3

{S1,S4}, {S5,S6}, S2, S3

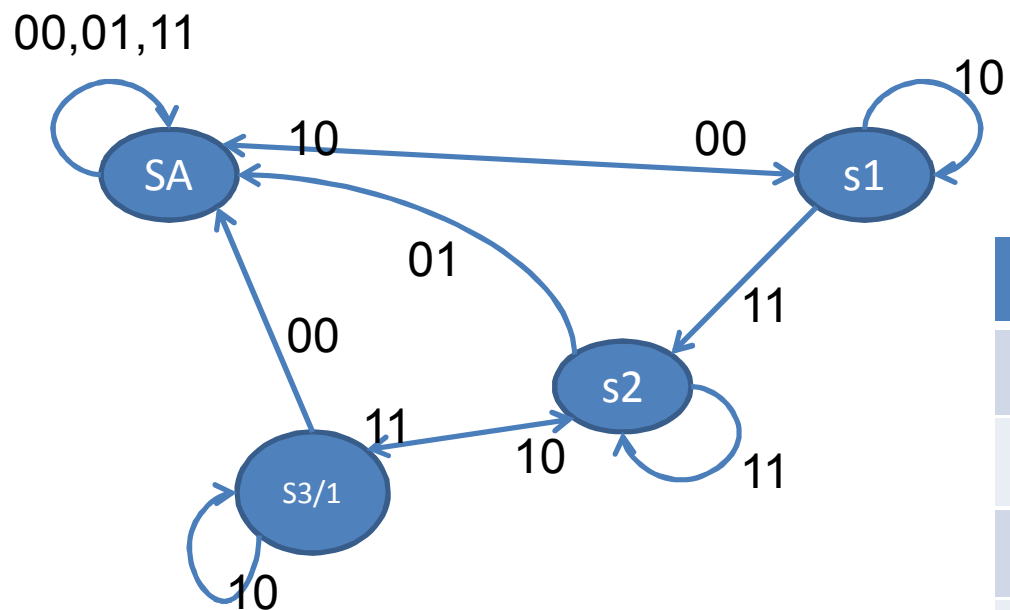
Stosując pierwszy zbiór klas zgodnych otrzymamy następującą minimalną tabelę przejść i wyjść automatu – następna strona.

Tablica przejść i wyjść przed i po minimalizacji stanów

stan	00	01	11	10	wy
S1	S4	-	S2	S1	0
S2	-	S5	S2	S3	0
S3	S4	-	S2	S3	1
S4	S4	S5	-	S1	0
S5	S4	S5	S6	-	0
S6	-	S5	S6	S1	0

stan	00	01	11	10	wy
S1	SA	-	S2	S1	0
S2	-	SA	S2	S3	0
S3	SA	-	S2	S3	1
SA	SA	SA	SA	S1	0

Klasę stanów {4,5,6} oznaczono jako SA



stan	00	01	11	10	wy
S1	SA	-	S2	S1	0
S2	-	SA	S2	S3	0
S3	SA	-	S2	S3	1
SA	SA	SA	SA	S1	0

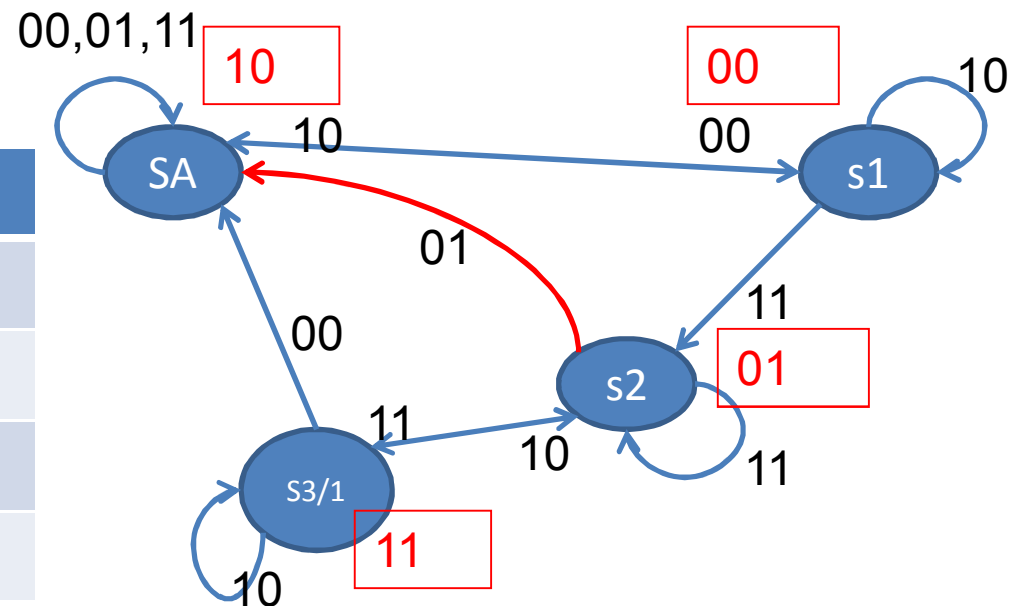
Automat wykrywa sekwencję par bitów 10, 11,10

Kodowanie stanów - trudne

stan	00	01	11	10	wy
S1	SA	-	S2	S1	0
S2	-	SA	S2	S3	0
S3	SA	-	S2	S3	1
SA	SA	SA	SA	S1	0

Należy unikać różnic na wielu bitach w ramach wektorów kodujących stany pomiędzy którymi następuje przejście.

stan	00	01	11	10	wy
00	10	-	01	00	0
01	-	10	01	11	0
11	10	-	01	11	1
10	10	10	10	00	0



Kodowanie stanów

Dla przyjętego kodowania kolejne stany automatu charakteryzują się różnicą na jednym bicie (ich sekwencji kodującej) oprócz sytuacji przejścia ze stanu 01 pod wpływem wejść 01. Wtedy następuje przejście z 01 do 10. W

zależności od prędkości zmian na wyjściach możliwe są potencjalnie dwa warianty zmian na wyjściu (zmiany wyjść nie muszą wystąpić równocześnie)

- 01 -> 00

- 01 -> 11

- Po przejściu do stanu 00 powinno nastąpić przejście do stanu 10 lecz w tablicy przejść takiej sytuacji jeszcze nie uwzględniono – trzeba je wprowadzić. Analogiczna sytuacja dotyczy stanu 11.

Stany 00 i stany 11 staną się stanami niestabilnymi przy przejściu 01->10 :
możliwe kolejności przejść: 01->11->10 lub 01->00->10

stan	00	01	11	10	wy
00	10	10	01	00	0
01	-	10	01	11	0
11	10	10	01	11	1
10	10	10	10	00	0

Zielone strzałki oznaczają przejścia do stanów pośrednich, a niebieskie do stanu stabilnego. Efektem modyfikacji jest tablica Karno na stronie kolejnej.

stan	00	01	11	10	wy
00	10	10	01	00	0
01	-	10	01	11	0
11	10	10	01	11	1
10	10	10	10	00	0

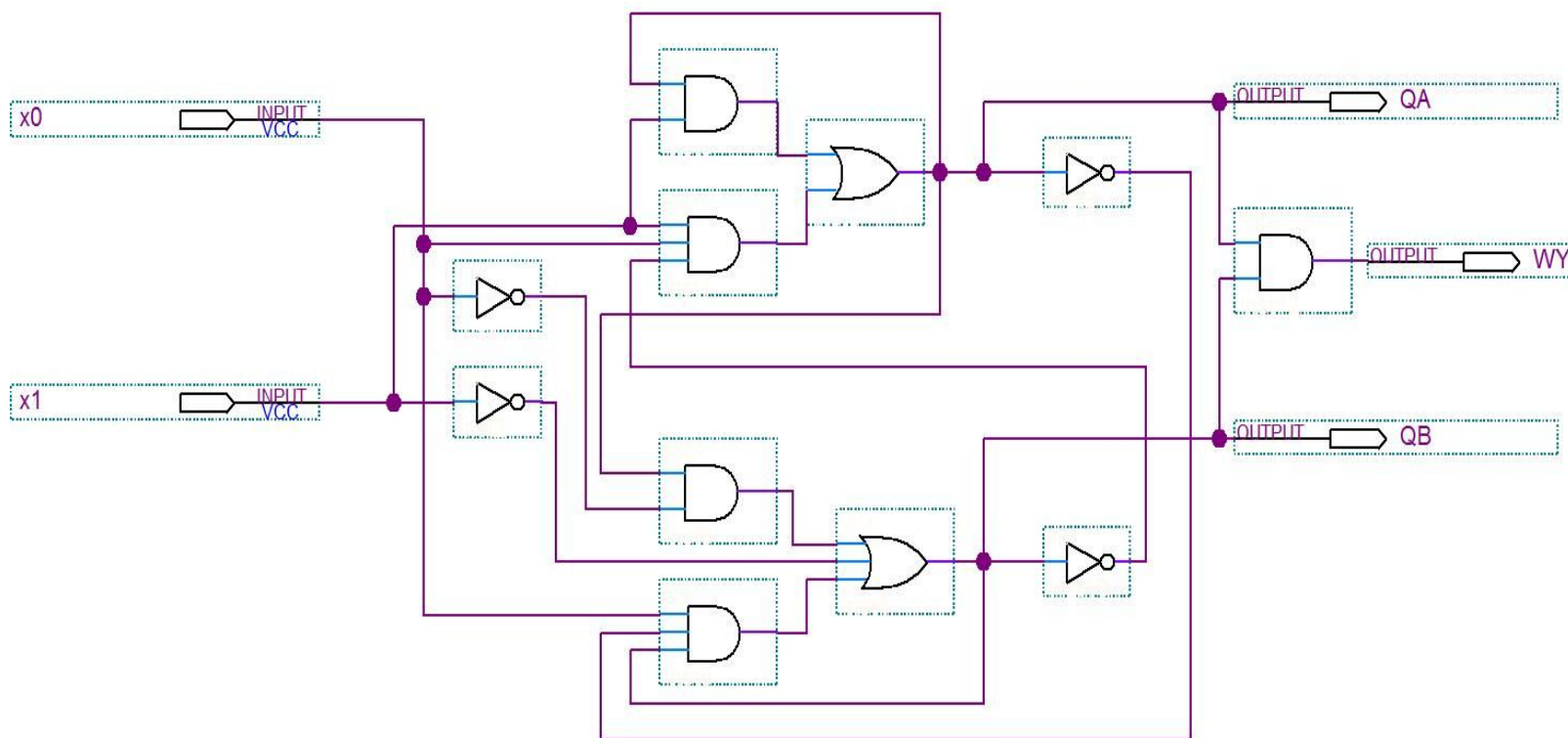
Obserwowana sytuacja, w której układ osiąga stan stabilny niezależnie od kolejności w jakiej następuje zmiana zmiennych stany nosi nazwę **gonitwy niekrytycznej**: przejścia 01->00 i 01->11 spowodują ostatecznie przejście do stanu 10 (wyścig niekrytyczny).

Q1Q0\ x1x0	00	01	11	10
00	1	1	0	0
01	-	1	0	1
11	1	1	0	1
10	1	1	1	0

Optymalizacja dla Q1
 $Q1_{next} = x1' + Q0x0' + Q1Q0'x0$

Q1Q0\ x1x0	00	01	11	10
00	0	0	1	0
01	-	0	1	1
11	0	0	1	1
10	0	0	0	0

Optymalizacja dla Q0
 $Q0_{next} = x1x0Q1' + Q0x1$

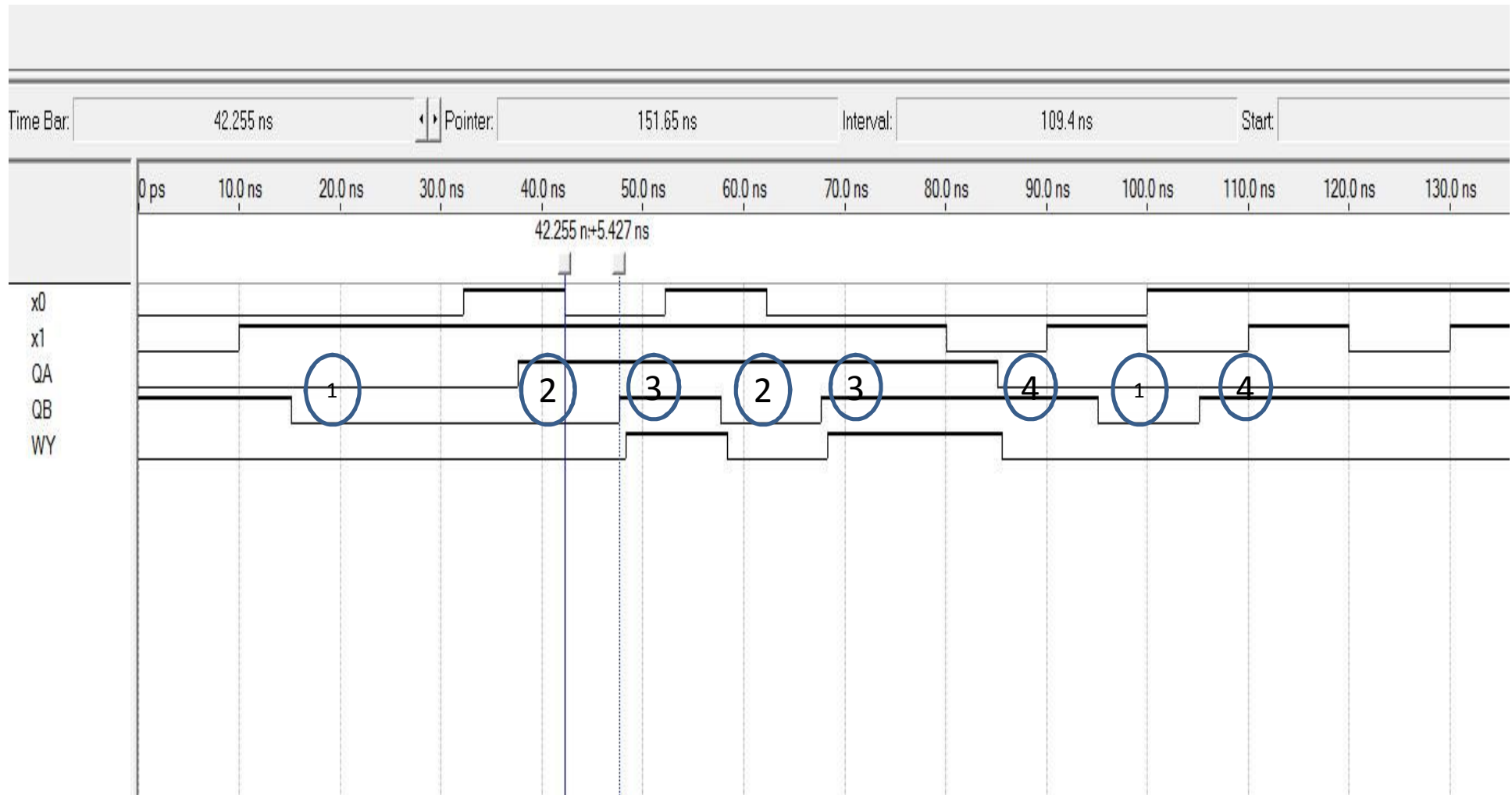


stan	00	01	11	10	wy
S1	SA	-	S2	S1	0
S2	-	SA	S2	S3	0
S3	SA	-	S2	S3	1
SA	SA	SA	SA	S1	0

stan	00	01	11	10	wy
00	10	-	01	00	0
01	-	10	01	11	0
11	10	-	01	11	1
10	10	10	10	00	0

Przykład automatu asynchronicznego

Automat ma 2 wejścia i jedno wyjście, wykrywa sekwencję par bitów 10, 11, 10.

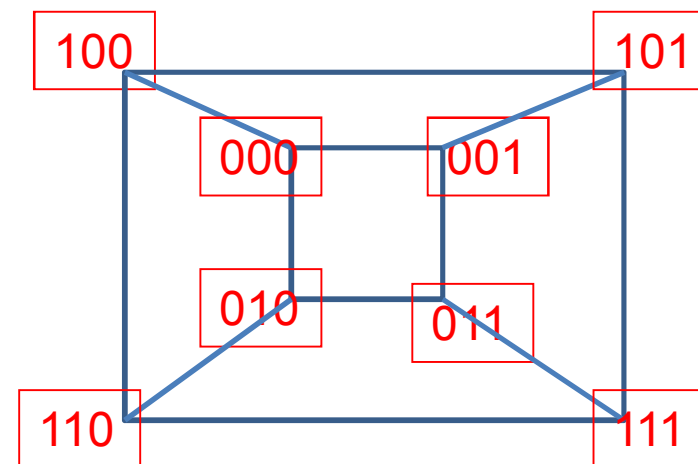
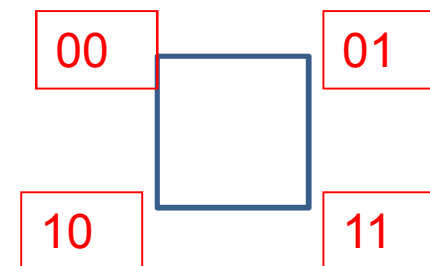
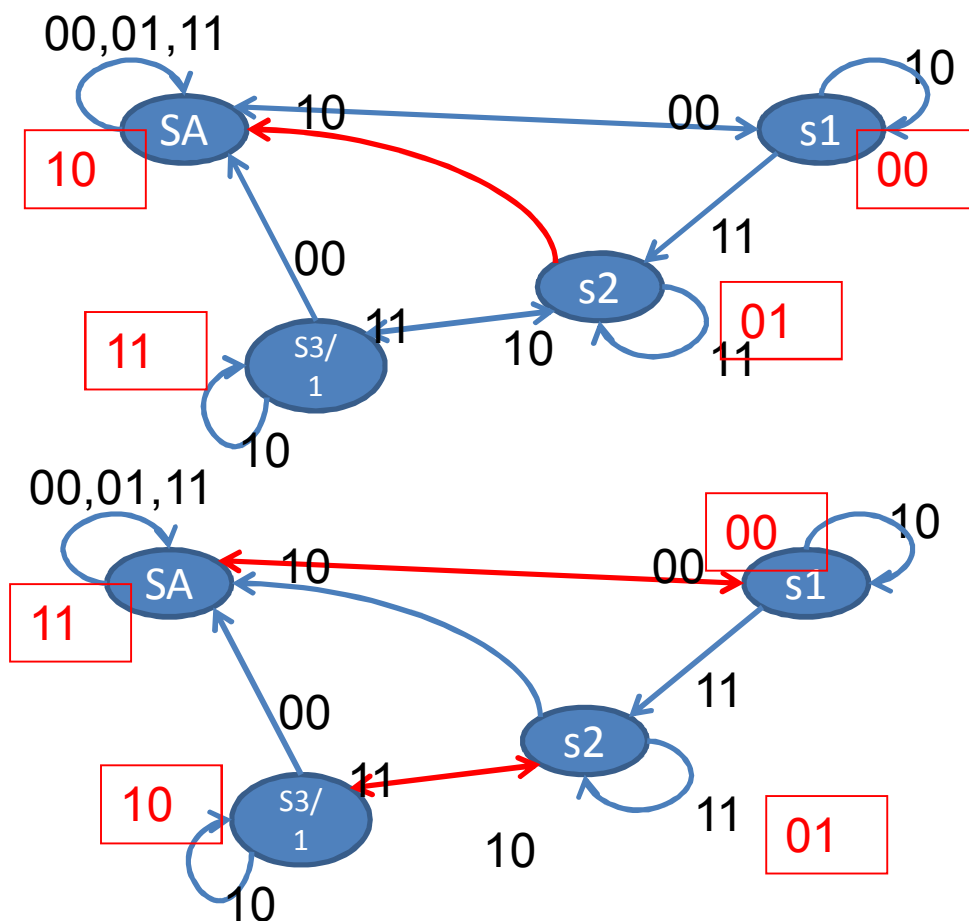


Inne kodowanie stanów automatu

stan	00	01	11	10	wy
S1	SA	-	S2	S1	0
S2	-	SA	S2	S3	0
S3	SA	-	S2	S3	1
SA	SA	SA	SA	S1	0

stan	00	01	11	10	wy
00	11	-	01	00	0
01	- 11	11	01	10	0
10	11	-	01	10	1
11	11	11	11	00	0

Dla zaznaczonego na żółto przejścia występuje gonitwa (wyścig) krytyczna – brak możliwości przejścia do stanu docelowego ze względu na stan przejściowy stabilny: 10->11 zamiast przejścia 10->01.
Konieczna zmiana kodowania (rozpięcie stanów automatu na kwadracie lub sześciacie kodowym) i dodatkowo rozszerzenie przestrzeni stanów – umożliwiające bezpieczne przejście między stanami poprzez stany pośrednie. Przejście 10->01 można zrealizować wprowadzając stany pośrednie wynikające z poszerzenia słowa kodowego o jeden bit: zamiast proponowanego przejścia 10-01 (niemożliwego) przejście przez nowe stany pośrednie (niestabilne) 010->110->111->101->001



Czerwonym kolorem oznaczono te przejścia pomiędzy stanami, które **mogą** wprowadzić gonitwę **krytyczną**. Przejścia te nie wpisują się w kwadrat kodowy. Kwadrat i sześcián kodowy zawierają krawędzie pomiędzy wierzchołkami różniącymi się na jednym bicie etykiety. Przejścia pomiędzy takimi wierzchołkami nie powodują gonitwy.

Automat wykrywający sekwencję 00,01,11,10

Tablica przejść

stan	wejścia	wejścia	wejścia	wejścia
	00	01	11	10
A	B	A	A	A
B	B	C	-	A
C	B	C	D	-
D	-	A	D	E
E/1	B	-	A	E

B	X			
C	X	TAK		
D	X	X	X	
E	X	X	X	X
	A	B	C	D

Tablica
minimalizacji
stanów

Zminimalizowana tablica przejść i zakodowana

Likwidacja gonitwy dla przejścia 10 do 01 i gonitwa krytyczna dla 11 do 00

stan	wejścia	wejścia	wejścia	wejścia
	00	01	11	10
A	BC	A	A	A
BC	BC	BC	D	A
D	-	A	D	E
E/1	B	-	A	E

stan	wejścia	wejścia	wejścia	wejścia
	00	01	11	10
00	01	00	00	00
01	01	01	11	00
11	- (01)	00	11	10
10/1	01	-	00	10

Automat wykrywający sekwencję
00,01,11,10

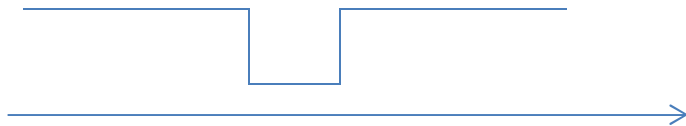
Rozszerzona (dla usunięcia gonitwy krytycznej) tablica przejść

stan	wejścia	wejścia	wejścia	wejścia
	00	01	11	10
000	001	000	000	000
001	001	001	011	000
011	001	111	011	010
010	001	000	000	010
111	110	110	110	110
110	100	100	100	100
100	000	000	000	000
101	-	-	-	-

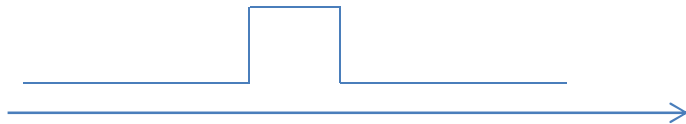
Hazard

- Krótkie **zakłócenie** impulsowe na wyjściu układu pojawiające się przejściowo na skutek zmian na wejściu.
- Hazard funkcjonalny - w odpowiedzi na zmiany **kilku** sygnałów wejściowych propagujące różnymi drogami do wyjścia (trudny do eliminacji).
- Hazard logiczny – w odpowiedzi na zmianę **jednego** sygnału propagującą różnymi drogami do wyjścia (możliwy do eliminacji za pomocą dodatkowych elementów układu).

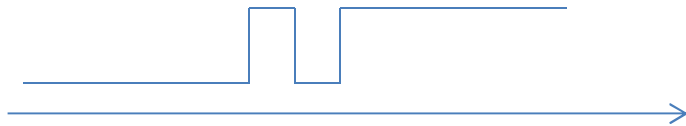
Postacie hazardu



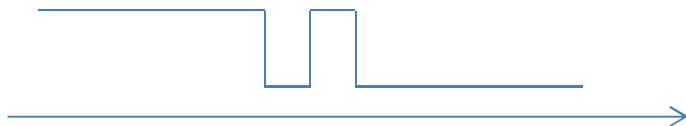
Hazard statyczny w „1”



Hazard statyczny w „0”



Hazard dynamiczny 0 ->1

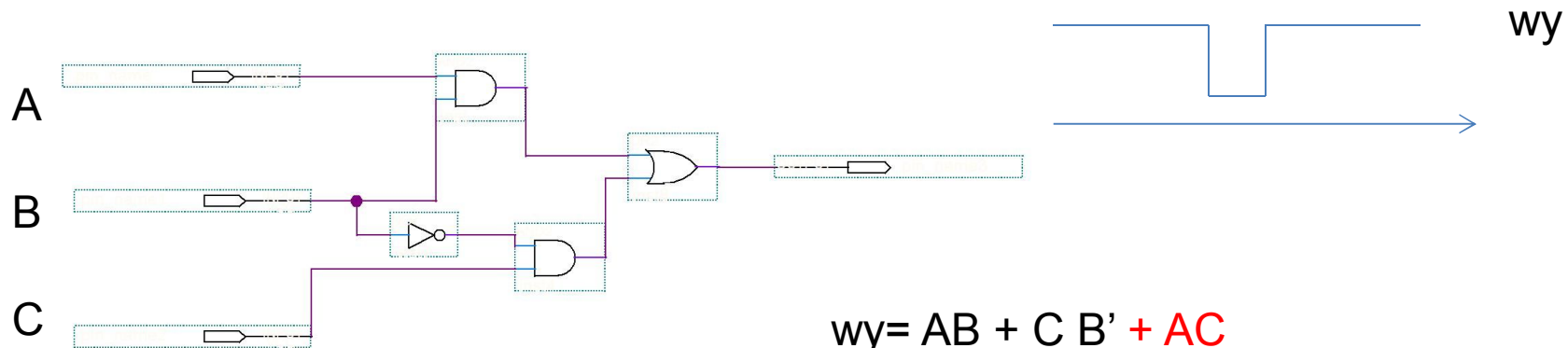


Hazard dynamiczny 1 ->0

Hazard

- Hazard funkcjonalny - w odpowiedzi na zmiany **kilku** sygnałów wejściowych propagujące różnymi drogami do wyjścia (trudny do eliminacji).
- Hazard logiczny – w odpowiedzi na zmianę **jednego** sygnału propagującą różnymi drogami do wyjścia (możliwy do eliminacji za pomocą dodatkowych elementów układu).

Hazard statyczny



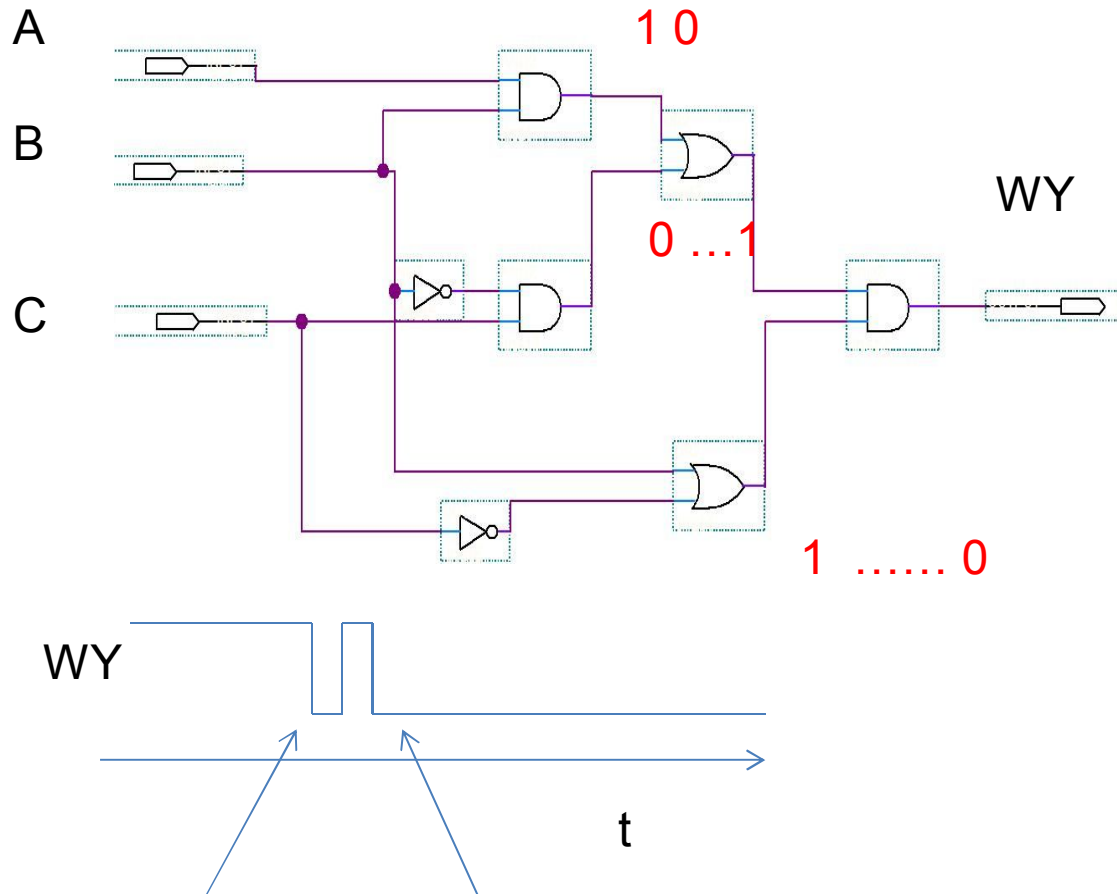
C\BA	00	01	11	10
0	0	0	1	0
1	1	1	1	0

$$wy = AB + C B' + AC$$

W układzie powyżej zmiana wektora wejść (ABC) z 111 na 101 może spowodować przejściowo poziom niski na wyjściu.

- Gdy bramka górna zadziała szybciej niż połączone szeregowo bramki dolne.
- Wprowadzenie do układu bramki wynikającej z zastosowania dodatkowego implikanta AC obejmującego generujące hazard zmiany sygnału pozwala na jego usunięcie.

Hazard dynamiczny



Pierwsze, ostateczne przejścia

W układzie zmiana wektora wejść (ABC) z 111 na 101 może spowodować - po pierwszym, a przed ostatecznym przejściem wyjścia do poziomu niskiego - przejściowo poziom wysoki.

Jest to możliwe, gdy dolna bramka OR będzie powolna w generacji ostatecznego wyniku, a wcześniej pojawi się przejściowo 0 na górnej bramce OR spowodowane przełączeniem stanu wejścia B. Hazard ten można również usunąć (strona poprzednia).