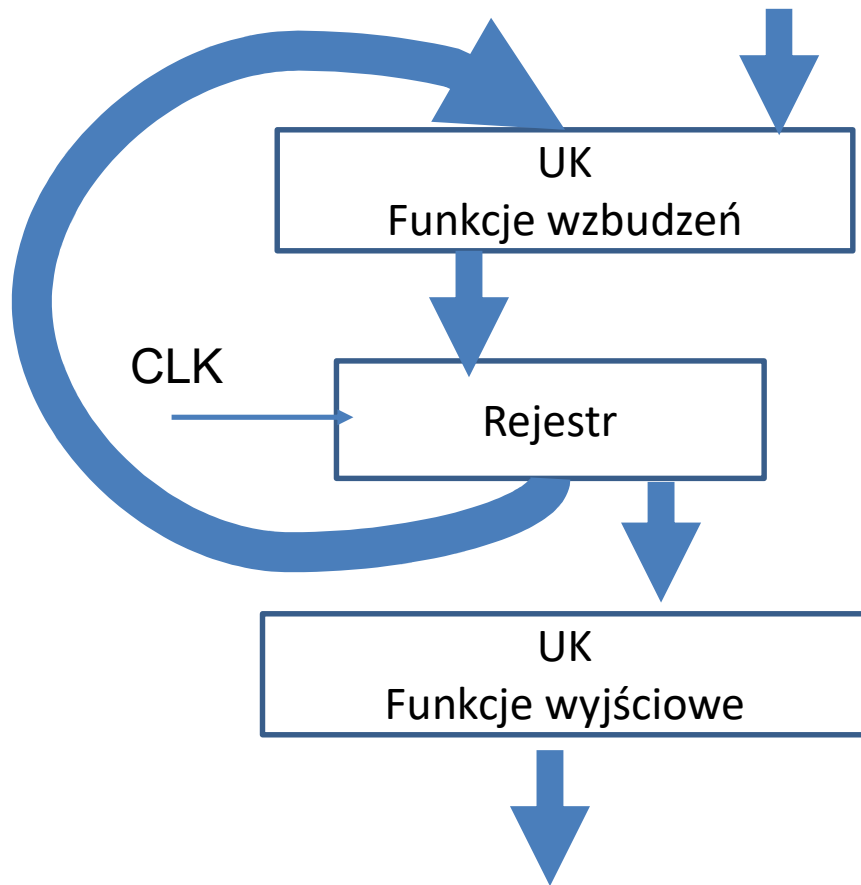


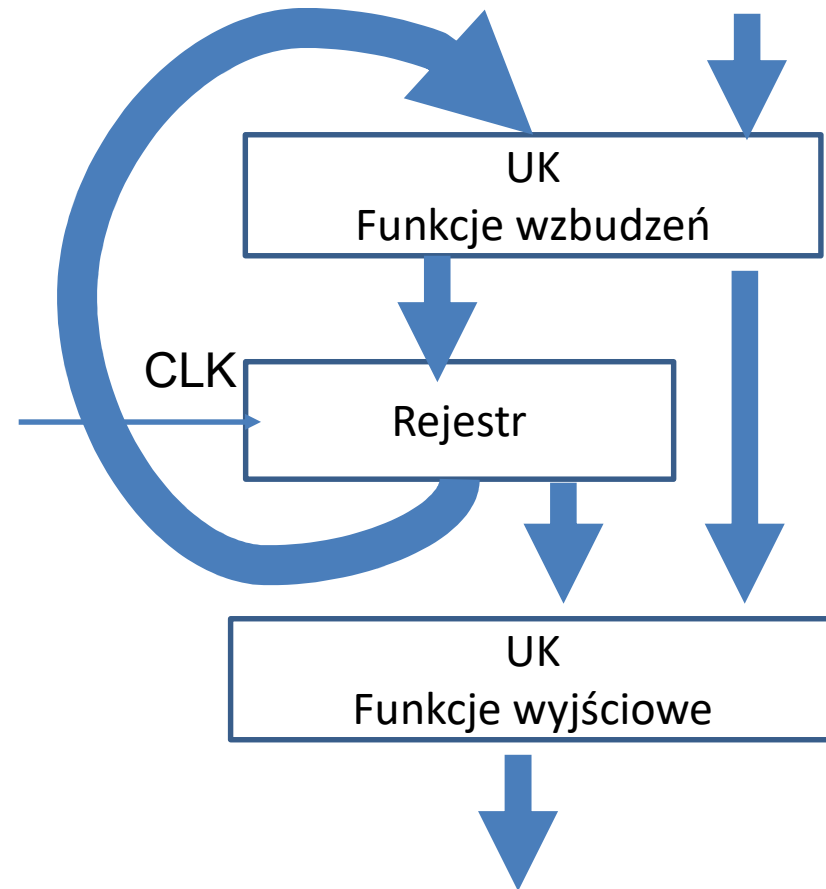
Projektowanie automatów synchronicznych

Rafał Walkowiak

Por. Układy cyfrowe Wilkinson r.5 s.133-156

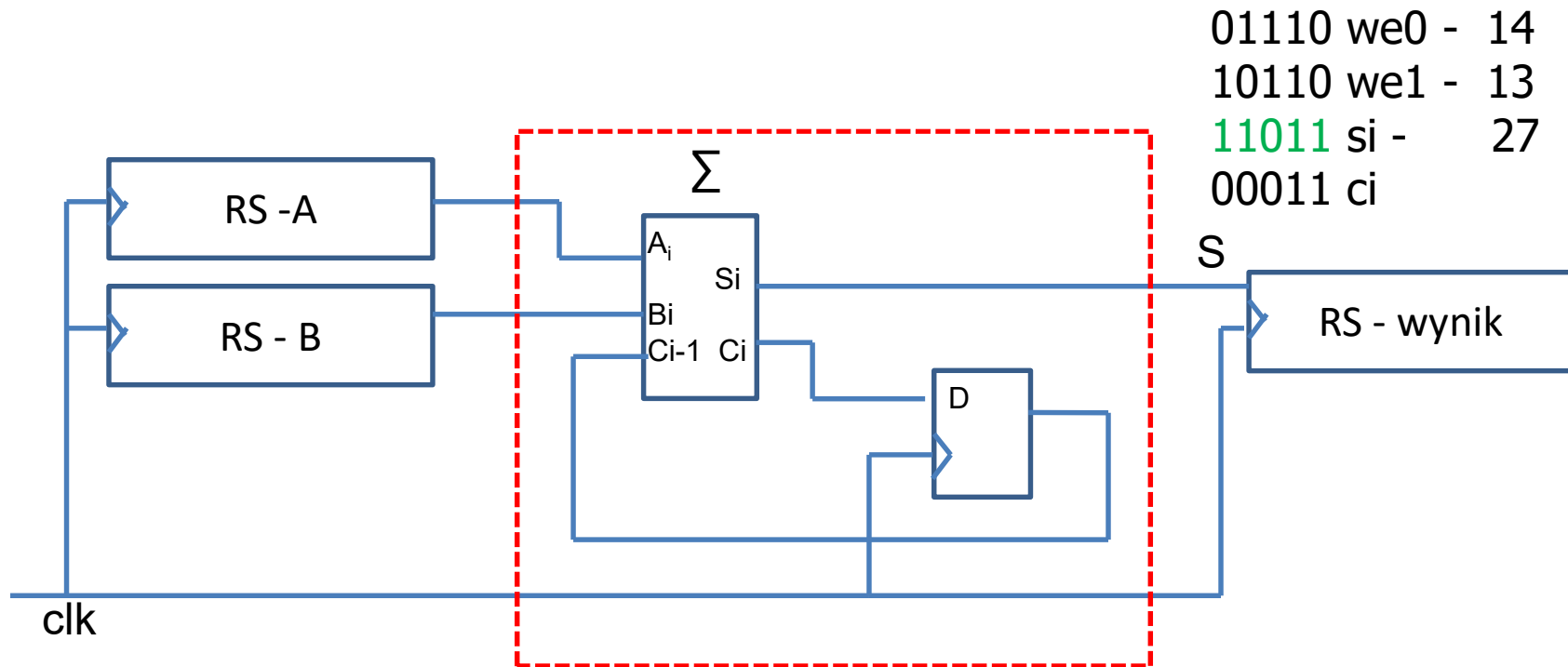


- Automat Moore'a



- Automat Mealy'ego

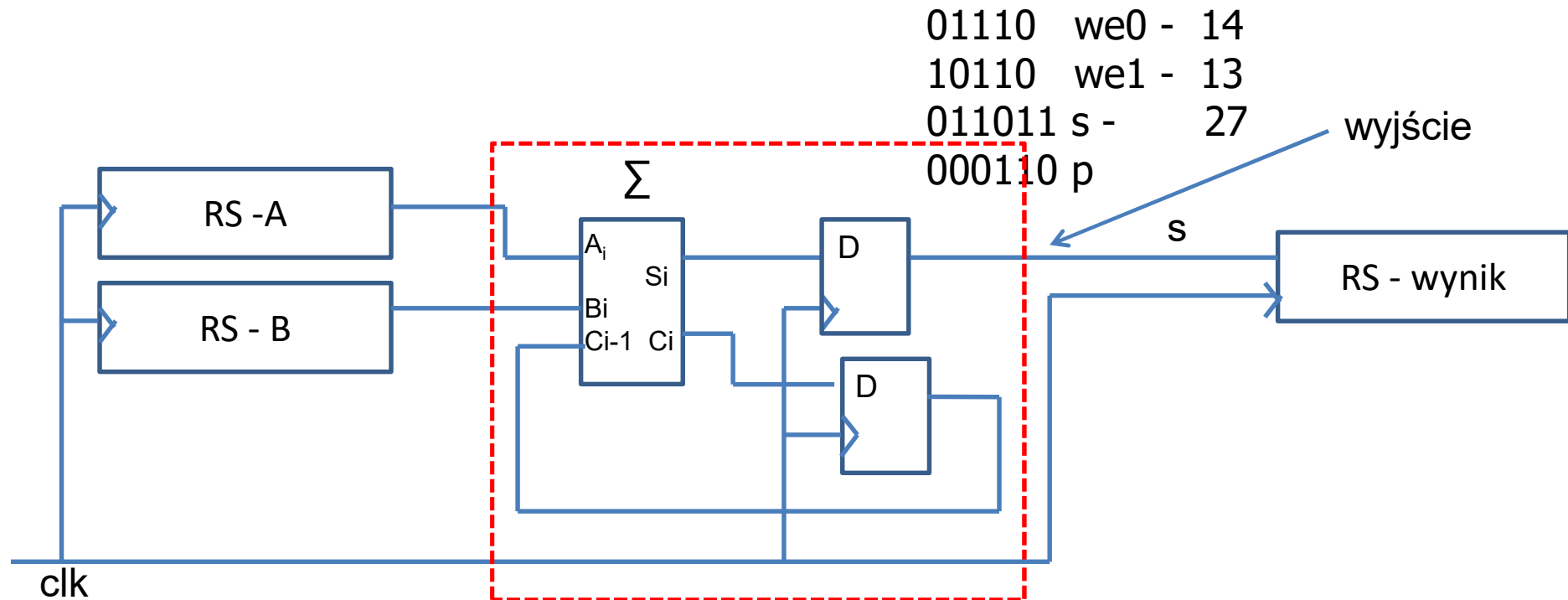
sumator szeregowy Mealy'ego



Sumator jednobitowy pełny:

- generuje wynik sumowania dla liczb dowolnego rozmiaru w czasie zależnym od rozmiaru liczb,
- liczby podawane są począwszy od najmłodszego bitu
- możliwe stany przejściowe na wyjściu S

sumator szeregowy automat Moore'a

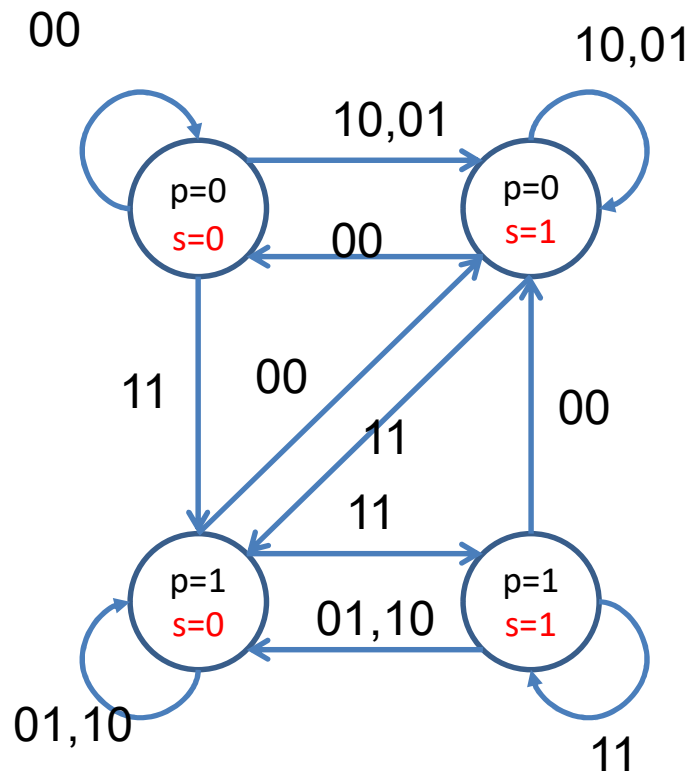


Sumator jednobitowy pełny:

- generuje wynik sumowania dla liczb dowolnego rozmiaru w czasie zależnym od rozmiaru liczb,
- liczby podawane są począwszy od najmłodszego bitu
- brak stanów przejściowych na wyjściu wyniku,
- wynik dostępny jeden cykl zegara później

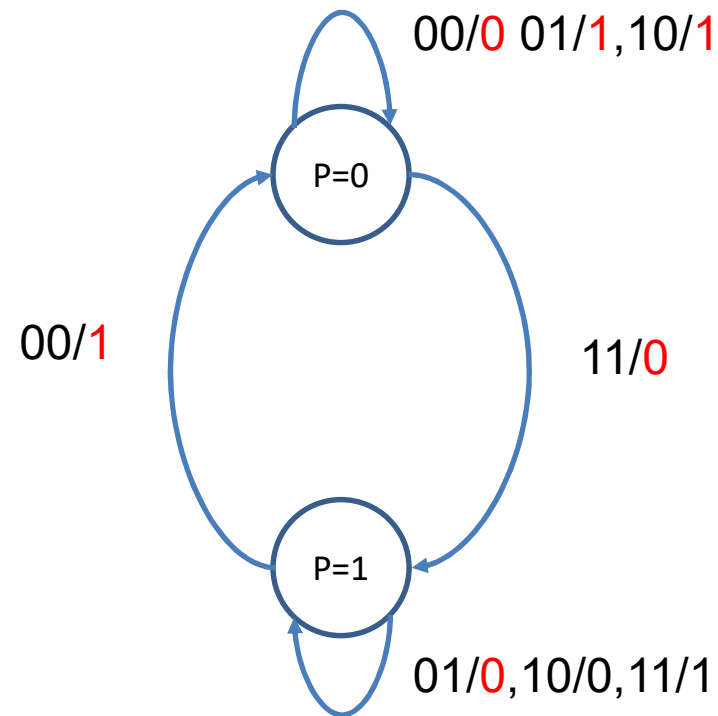
Sumator sekwencyjny - graf automatu

Model Moore'a



Stany oznaczone na grafie i określone przez wartość przeniesienia i sumy, ciąg kodujący stan 2 bitowy- wystarczy

Model Mealy'ego

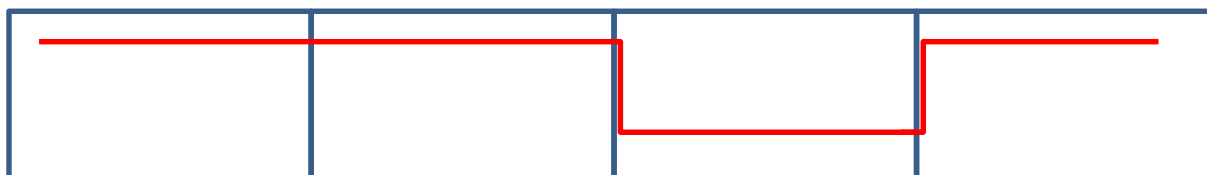


Stany oznaczone przez przeniesienie, przejścia oznaczone stanami **wejść/stanem wyjścia - sumy**, w obu stanach wyjście sumy może mieć różne wartości w zależności od stanu wejść.

Procedura projektowania automatu

- Opis działania
- Sporządzenie grafu przejść i tablicy przejść
- Redukcja stanów
- Kodowanie stanów (liczba i typ przerzutników)
- Tablica wzbudzeń (przejść) i wyjść
- Wyznaczenie funkcji wzbudzeń i funkcji wyjść
- Realizacja układu
- Testowanie poprawności działania

Wykrywanie sekwencji

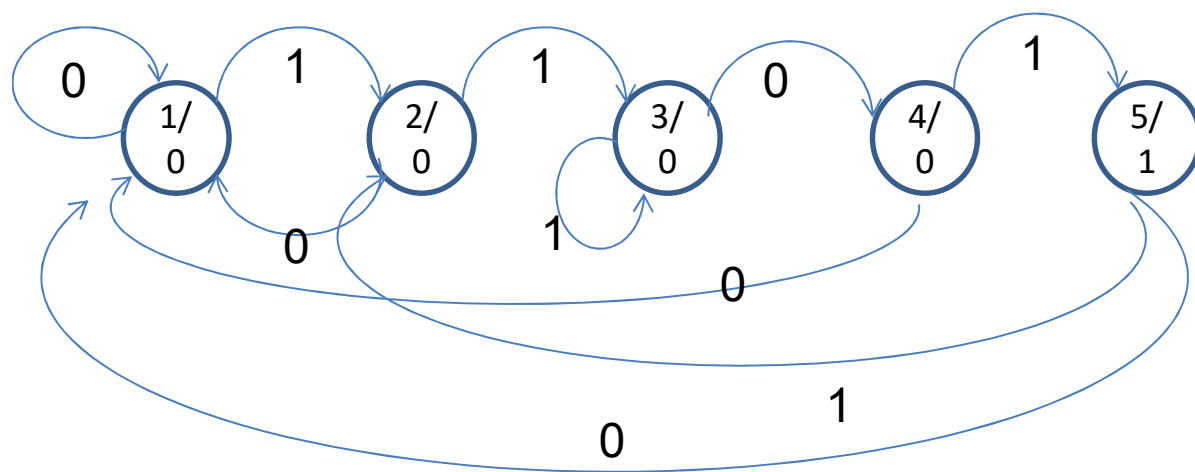


DANE
SEKWENCJA
1101

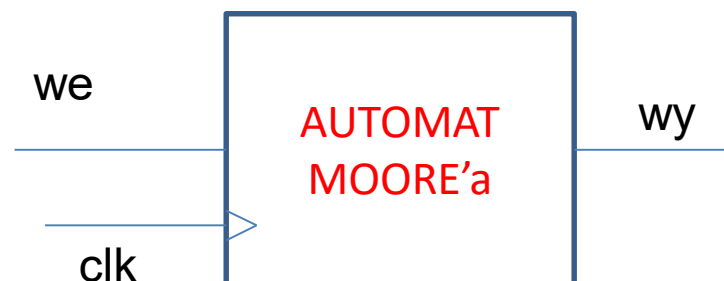


Dane analizowane
zboczem
narastajacym,
zbocze aktywne
przerzutnika

Automat Moore'a do wykrywania sekwencji bitów **1101**



	Stan	następny
Stan bieżący	We=0	We=1
1	1	2
2	1	3
3	4	3
4	1	5
5	1	2

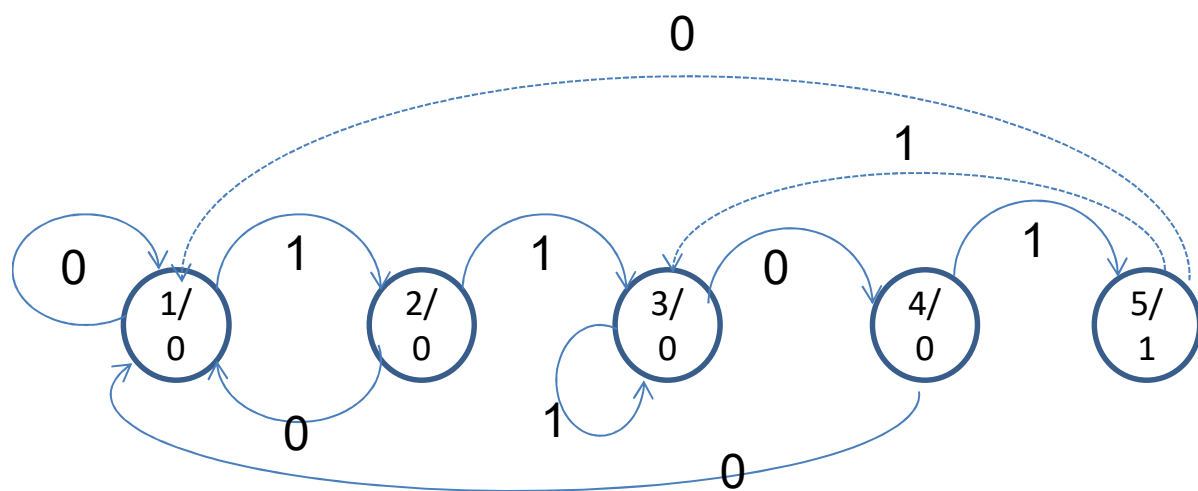


Graf stanów i tablica przejść

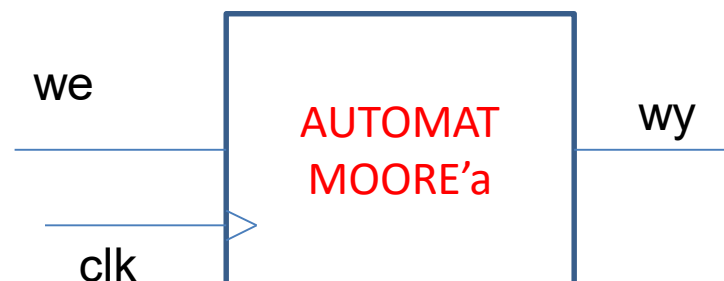
Stan 5 informuje o wykryciu szukanej sekwencji –

na wyjściu wartość 1 w tym stanie. Stan 5 oznacza, że poszukiwana sekwencja wystąpiła.

Automat Moore'a do wykrywania sekwencji bitów **1101**



	Stan	następny
Stan bieżący	We=0	We=1
1	1	2
2	1	3
3	4	3
4	1	5
5	1	2

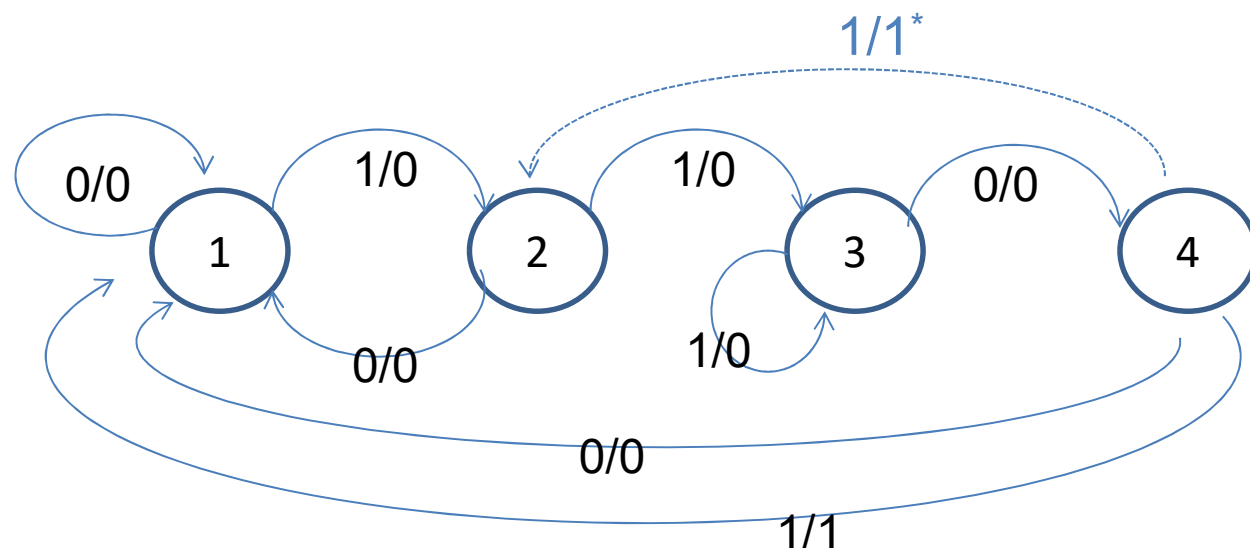


Graf stanów i tablica przejść

Wariant automatu:

Linie przerywane (zamiast ciągłych) ze stanu 5 - wykrywanie nakładających się ciągów
 Stan 5 informuje o wykryciu szukanej sekwencji – na wyjściu wartość 1 w tym stanie. Stan piąty oznacza, że poszukiwana sekwencja wystąpiła.

Automat Mealego'a do wykrywania sekwencji bitów 1101



	stan	następny
Stan bieżący	We=0	We=1
1	1/0	2/0
2	1/0	3/0
3	4/0	3/0
4	1/0	1/1

Sygnały wyjściowe są związane ze stanami **bieżącymi** i z wartościami sygnałów wejściowych. **Po wykryciu ciągu w stanie 4 na wyjściu pojawia się 1 i następuje przejście do stanu 1 (w stanie tym nie ma 1 na wyjściu).**

*Jeżeli dopuszczamy możliwość nakładania się sekwencji przejście z 4 następuje do stanu 2.

Liczba stanów 4 (o jeden mniej niż Moor'a, często mniej).

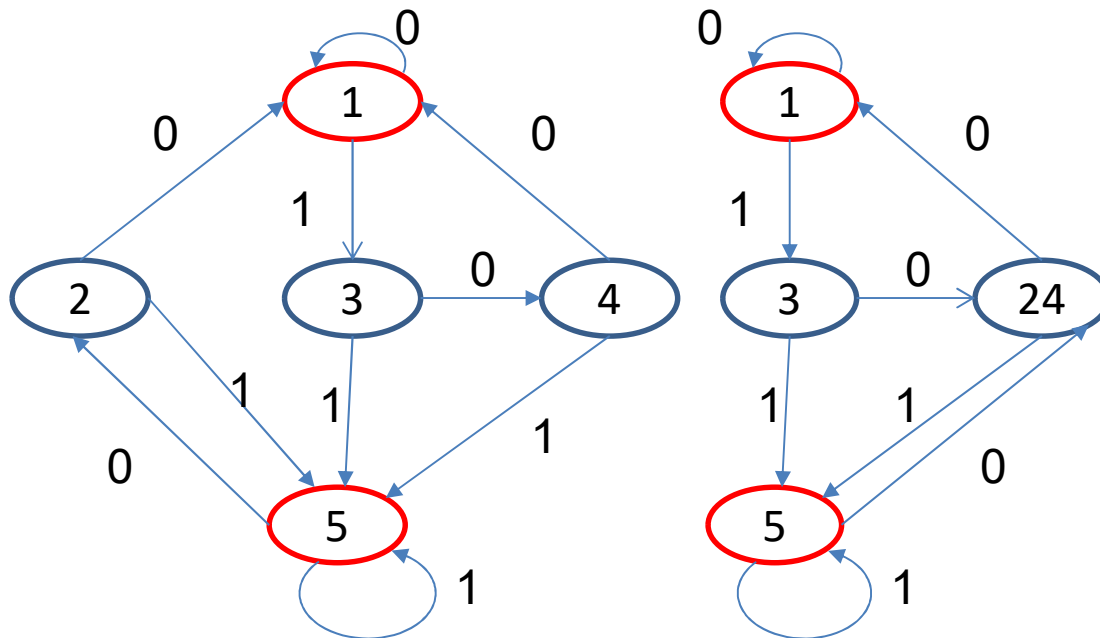
Na wyjściu w stanie 4 możliwe stany przejściowe przy zmianach wartości na wejściu.

Stan 4 oznacza że wystąpiła sekwencja 110 i wyjście jest zależne teraz od wejścia 0/1, przejście od stanu kolejnego zależne od wartości na wejściu przyjętej zboczem clk.

Redukcja stanów automatu

Dwa stany są równoważne (i mogą być zastąpione jednym stanem) jeżeli dla każdej kombinacji wejściowej:

- mają (dostarczają) taką samą wartość wyjścia oraz
- przenoszą automat do tego samego stanu lub stanów równoważnych



Wartości na wyjściach: kolor czerwony -1, kolor niebieski - 0

Wyznaczanie stanów równoważnych

1. W tablicy trójkątnej dla każdej pary stanów określ warunki ich równoważności – t.j. wymagane równoważności stanów wyjściowych.
2. Dla pary stanów różniących się wartością wyjścia zaznacz X – brak równoważności (automat Mealego – test dla każdej kombinacji wejść)
3. Dla każdego pola różnego od X zbadaj w tablicy możliwość równoważności – brak X w polu odpowiadającym parze stanów. Zaznacz (poprzez X) stany **nierównoważne** wynikające z braku równoważności stanów następnych.
4. Powtarzaj punkt 3 aż będzie on przynosił efekty.
5. Pozycje nieskreślone określają stany równoważne.

UWAGA:

równoważności jest **przechodnia** w przypadku pełnej określoności tablicy przejść (czyli jeśli - brak przejść do stanów dowolnych).

Jeśli są docelowe stany dowolne to dla określenia równoważności - konieczne sprawdzenie warunków pokrycia i zamknięcia dla wyznaczenia klasy stanów równoważnych (rozszerzenie pojęcia pary stanów) – omówione w projektowaniu automatów asynchronicznych.

Redukcja stanów automatu Moora

przykład

	stany		nastę pne		
Stan aktua lny	00	01	11	10	wyjscie
1	5	3	2	1	0
2	5	3	1	4	0
3	3	4	4	5	1
4	5	3	2	2	0
5	6	7	1	1	0
6	3	3	1	7	0
7	7	1	1	5	1

2	12 14					
3	X	x				
4	12	12 24	x			
5	56 37 12	56 37 14	x	56 37 12		
6	35 12 17	35 47	x	35 12 27	36 37 17	
7	x	x	37 14	x	x	x
	1	2	3	4	5	6

Redukcja stanów automatu Moora -przykład

2	12 14					
3	X	x				
4	12	12 24	x			
5	56 37 12	56 37 14	x	56 37 12		
6	35 12 17	35 47	x	35 12 27	36 37 17	
7	x	x	37 14	x	x	x
	1	2	3	4	5	6

Kolejność wykreślenia:
61,62,64(35), 56(36),51,52,54(56)

Stan akt	00	01	11	10	wyjsc ie
1	5	3	2	1	0
2	5	3	1	4	0
3	3	4	4	5	1
4	5	3	2	2	0
5	6	7	1	1	0
6	3	3	1	7	0
7	7	1	1	5	1

Stan akt	00	01	11	10	wyjsc ie
124	5	37	124	124	0
37	37	124	124	5	1
5	6	37	124	124	0
6	37	37	124	37	0

Automat Mealego wykrywania sekwencji (poprzedni slajd) – próba minimalizacji stanów

	23		
3	14 23	14	
4	x	x	x
	1	2	3

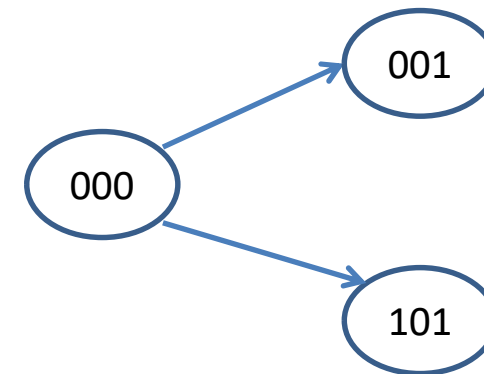
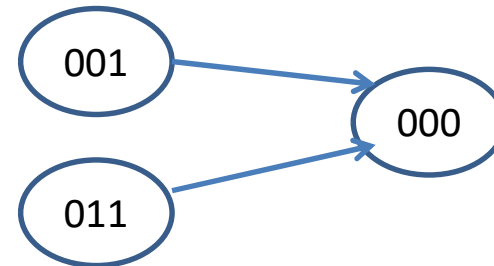
Stan bieżący y	We=0	We=1
1	1/0	2/0
2	1/0	3/0
3	4/0	3/0
4	1/0	1/1

Stany 1 i 4 nie mogą być równoważne gdyż mają różne wartości wyjść dla We=1

W konsekwencji nie równoważna jest są również para stanów 2 i 3.

Kodowanie stanów -zasady

- Zasada 1 – Należy przyporządkować stanom, które mają ten sam stan następny, słowa kodowe różniące się tylko wartością jednego bitu.
- Zasada 2- Należy przyporządkować stanom następnym, które mają ten sam stan bieżący, słowa kodowe różniące się tylko wartością jednego bitu.
- Stosowanie powyższych zasad ma na celu uzyskiwanie najprostszych funkcji wzbudzeń wymuszających odpowiednie stany następne przerzutników.



W sytuacji gdy każdy stan ma tylko jeden stan następny warto zastosować dla kolejnych stanów słowa kodowe różniące się na jednej pozycji np. wg kodu Greya. Kodowanie 1 z N – każdy stan reprezentowany przez wyróżniony stan jednego przerzutnika.

Programowany generator impulsów

Opis działania układu:

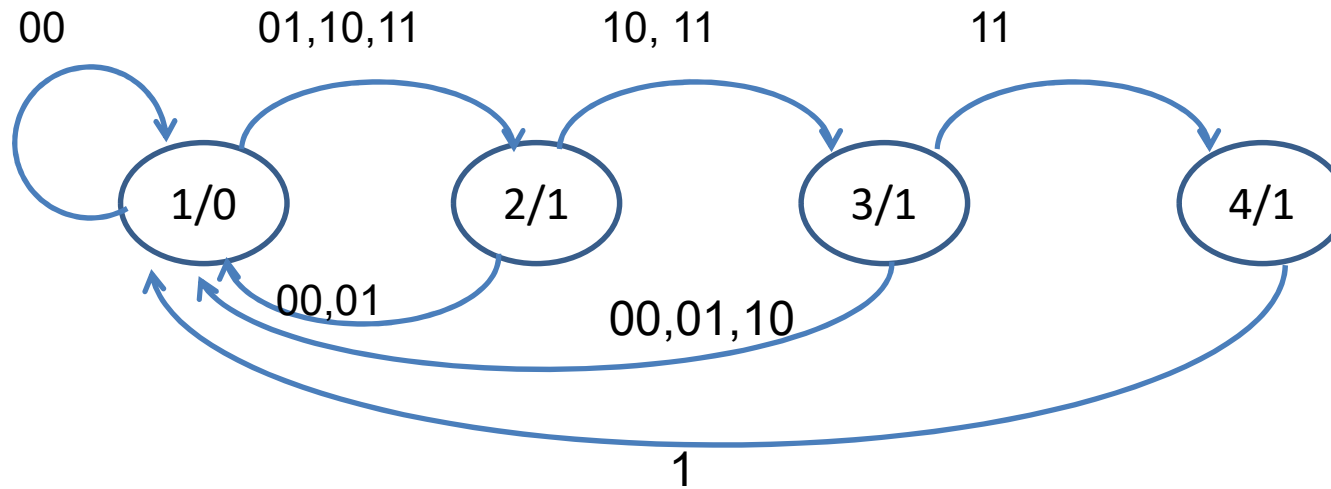
- Układ generuje na wyjściu impuls o długości trwania mierzonej w jednostkach równych wielokrotności zegara wejściowego synchronizującego pracę układu.
- Czas trwania jest kodowany na dwóch wejściach informacyjnych układu.
- Dla wejść równych zero impuls nie jest generowany

Sygnały :

- 2 wejścia kodujące czas trwania impulsu
- Wyjście

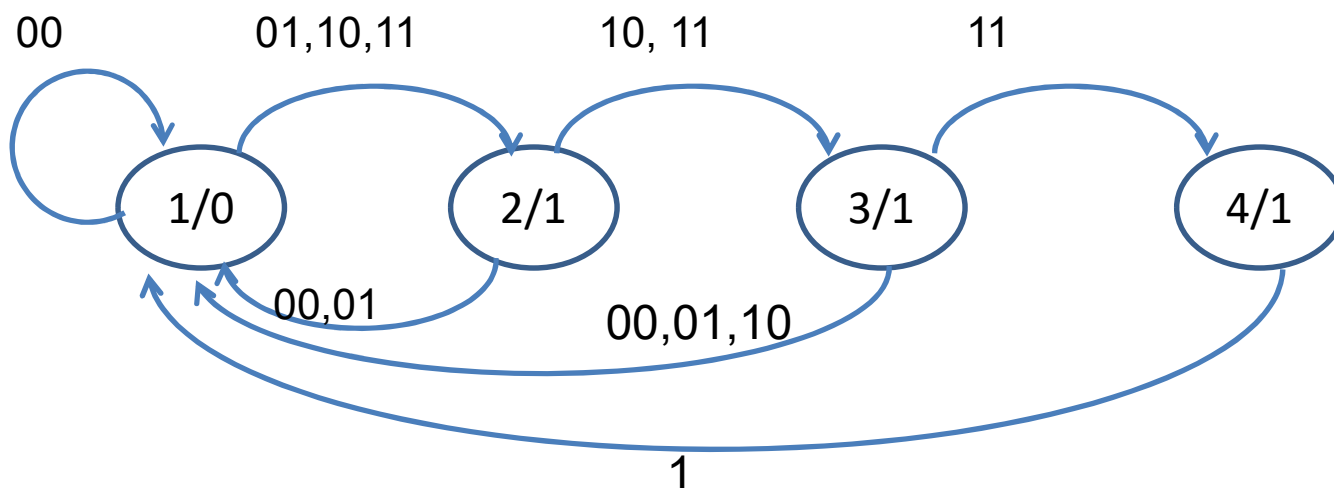


Graf stanów z równaniami określającymi przejścia



Generator impulsów z sterowaną wejściami długością trwania impulsu (długość kodowana binarnie). Oznaczenia na łukach muszą wyczerpywać wszystkie możliwe kombinacje wartości wejść dla jednoznacznego określenia przejścia.

Programowany generator impulsów



Stan bieżący	00	01	11	10	Wejścia ←
00	00	01	01	01	
01	00	00	11	11	
11	00	00	10	00	
10	00	00	00	00	

Graf stanów i tablica stanów generatora impulsów w postaci zakodowanej – kodowanie kolejnych stanów kodem Graya, sygnały stanu Q1, Q0, sygnały wejściowe IN1, IN0

Programowany generator impulsów

	wejścia			
Stan bieżący	00	01	11	10
00	00	01	01	01
01	00	00	11	11
11	00	00	10	00
10	00	00	00	00

Realizacja
przy użyciu
przerzutników
jk

Tabela zawiera kody stanów następnych

	wejścia	IN1, IN0		
Q1,Q0	00	01	11	10
00	0- 0-	0- 1-	0- 1-	0- 1-
01	0- -1	0- -1	1- -0	1- -0
11	-1 -1	-1 -1	-0 -1	-1 -1
10	-1 0-	-1 0-	-1 0-	-1 0-

Q	Q+	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

Tabela zawiera wymuszenia (wartości wejść J1,K1,J0,K0) określające bity stanów następnych przerzutnika, J1,K1,J0,K0 są funkcją Q1,Q0,IN1,IN0

	wejścia				
Q1,Q0	00	01	11	10	
00	0- 0-	0- 1-	0- 1-	0- 1-	
01	0- -1	0- -1	1- -0	1- -0	
11	-1 -1	-1 -1	-0 -1	-1 -1	
10	-1 0-	-1 0-	-1 0-	-1 0-	

Generator
impulsów
Realizacja
przy użyciu
przerzutników
jk

J0	we	01	11	10
stan	0	1	1	1
01	-	-	-	-
11	-	-	-	-
10	0	0	0	0

K0	we	01	11	10
stan	-	-	-	-
01	1	1	0	0
11	1	1	1	1
10	-	-	-	-

Q	Q+	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

J1	we	01	11	10
stan	0	0	0	0
01	0	0	1	1
11	-	-	-	-
10	-	-	-	-

K1	we	01	11	10
stan	-	-	-	-
01	-	-	-	-
11	1	1	0	1
10	1	1	1	1

$$J0 = we0Q1' + we1Q1'$$

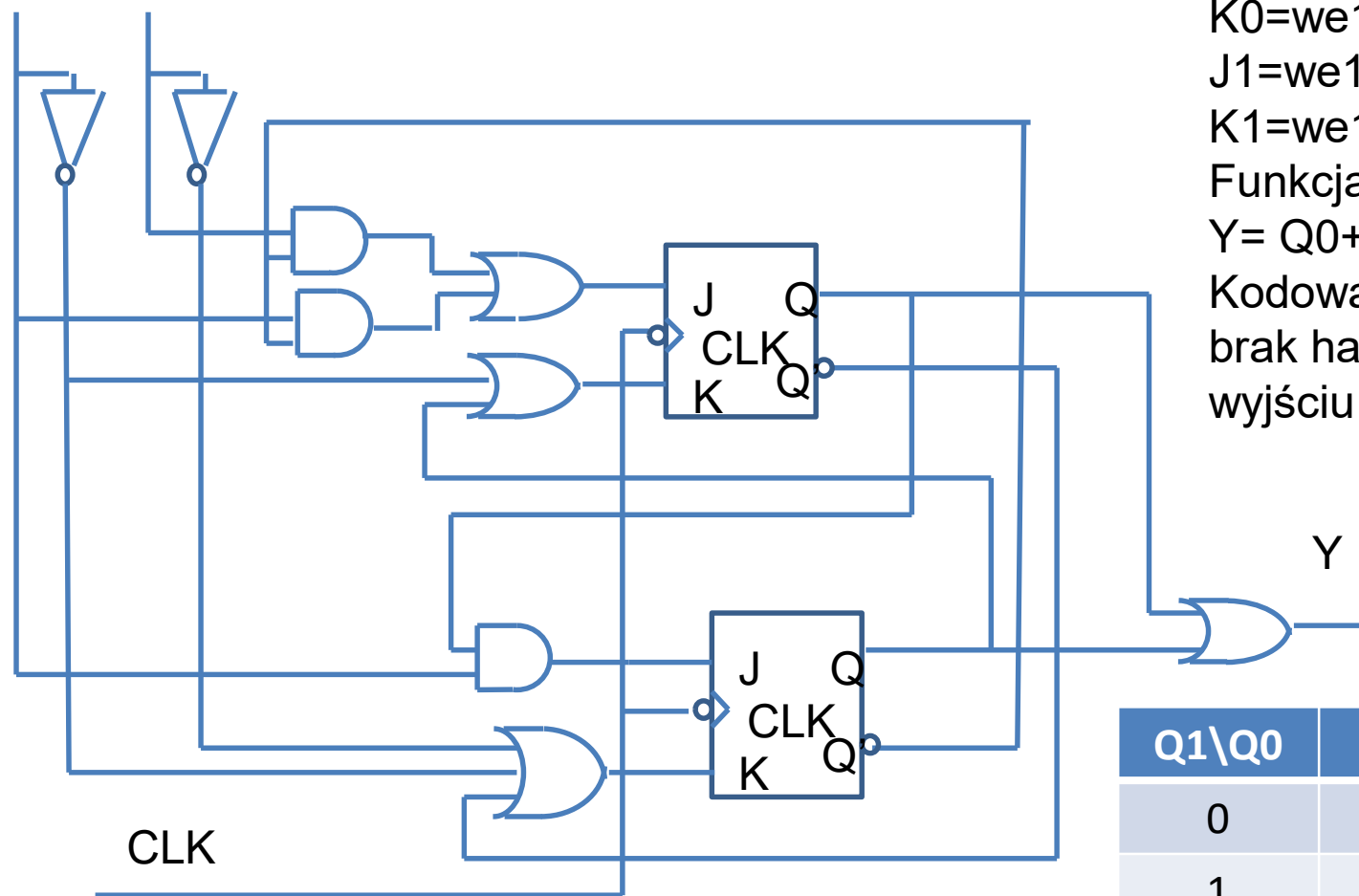
$$K0 = we1' + Q1$$

$$J1 = we1Q0$$

$$K1 = we1' + we0' + Q0'$$

Programowany generator impulsów implementacja układu na JK i bramkach

WE1 WE0



Funkcje wzbudzeń

$$J0 = we0 \cdot Q1' + we1 \cdot Q1'$$

$$K0 = we1' + Q1$$

$$J1 = we1 \cdot Q0$$

$$K1 = we1' + we0' + Q0'$$

Funkcja wyjścia

$$Y = Q0 + Q1$$

Kodowanie zapewnia
brak hazardu na
wyjściu

Q1\Q0	0	1
0	0	1
1	1	1

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



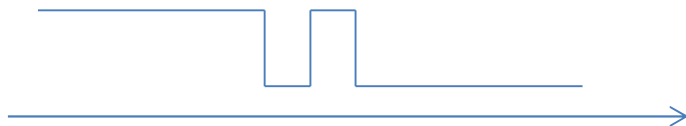
Hazard statyczny w „1”



Hazard statyczny w „0”

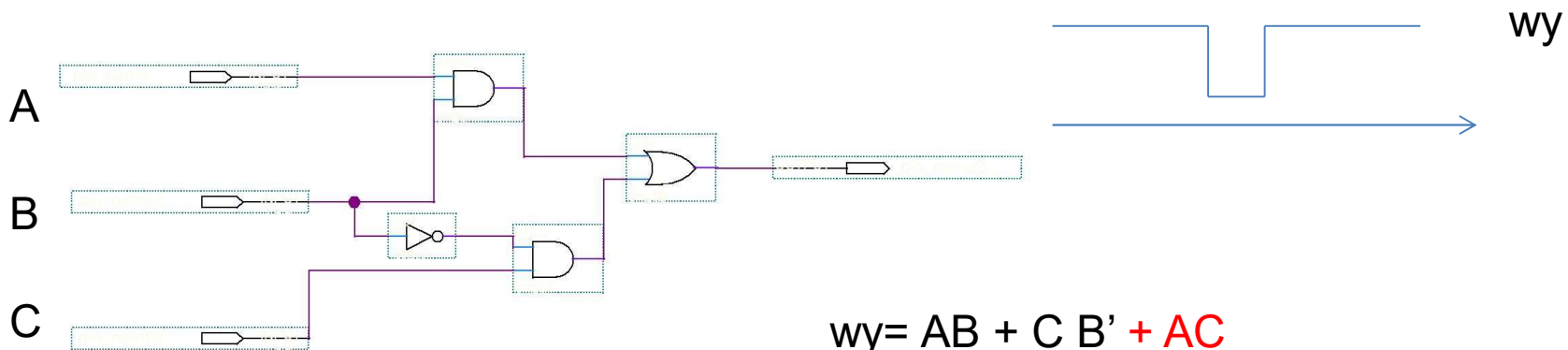


Hazard dynamiczny 0 ->1



Hazard dynamiczny 1 ->0

Hazard statyczny

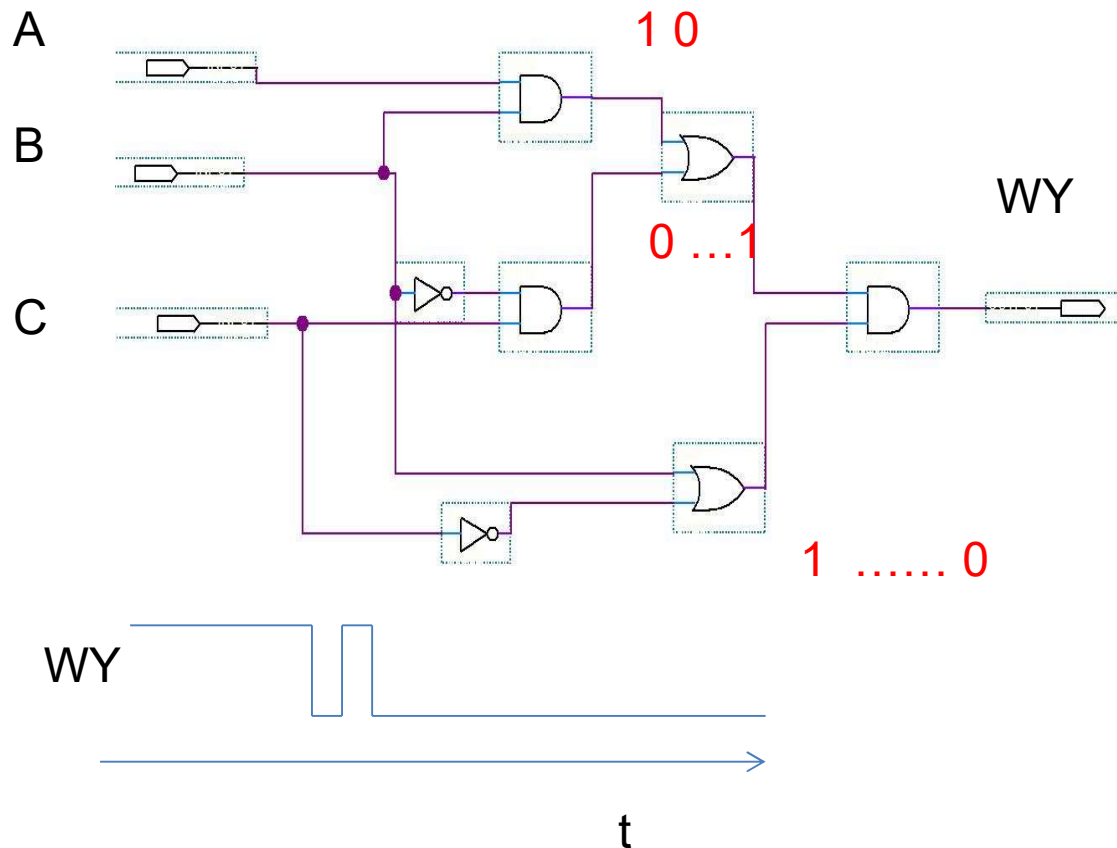


C\BA	00	01	11	10
0	0	0	1	0
1	1	1	1	0

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

W układzie powyżej zmiana wektora wejść (ABC) z 111 na 101 może spowodować przejściowo poziom niski na wyjściu. Wprowadzenie implikanta AC obejmującego zmiany sygnału generujące hazard pozwala na jego usunięcie.

Hazard dynamiczny



Realizacja funkcji $WY = AB$

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 poprzez zmianę struktury układu – przykład na poprzedniej stronie.