Sławomir Kulesza

Technika cyfrowa
Projektowanie automatów asynchronicznych

Wykład dla studentów III roku Informatyki

Wersja 3.0, 03/01/2013

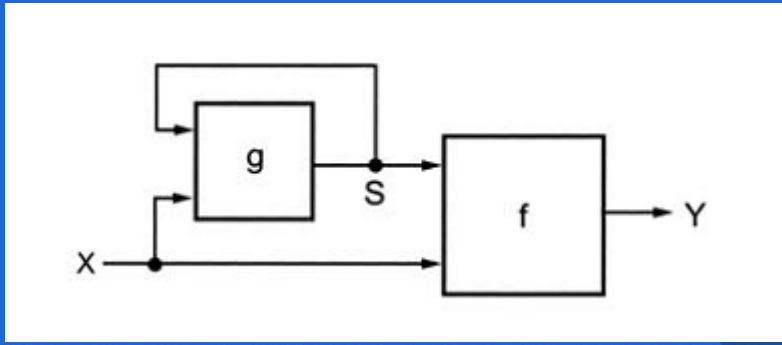
Automaty skończone

Automat skończony (Finite State Machine – FSM) – układ, który może w danej chwili przebywać w jednym ze skończonej liczby logicznie odmiennych stanów wewnętrznych.

Przykłady: liczniki, rejestry, detektory sekwencji, automaty wrzutowe itp.

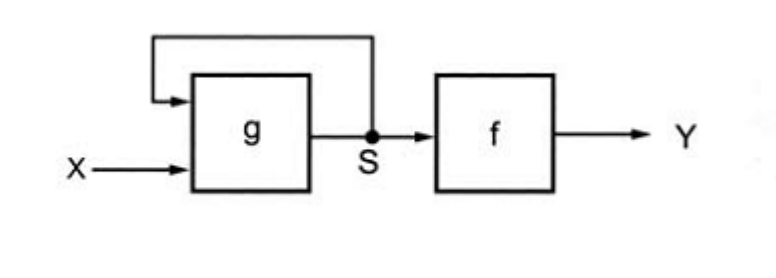
Ogólna struktura FSM

Automat Mealy'ego

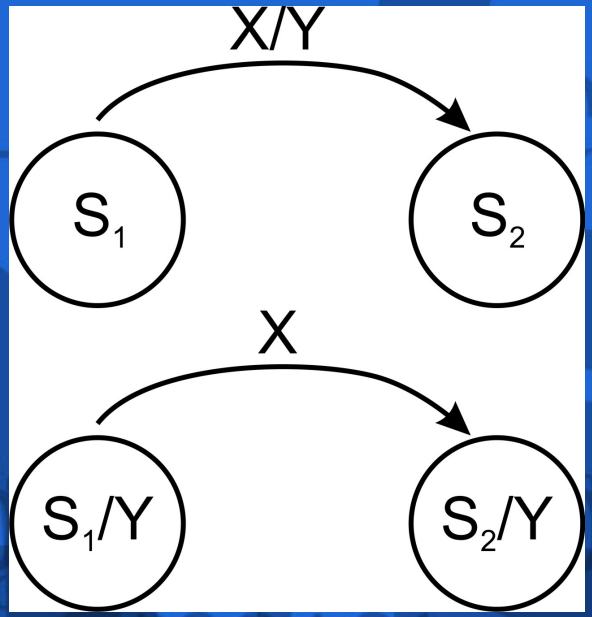


Funkcja wyjść: $Y^t = f(S^t, X^t)$

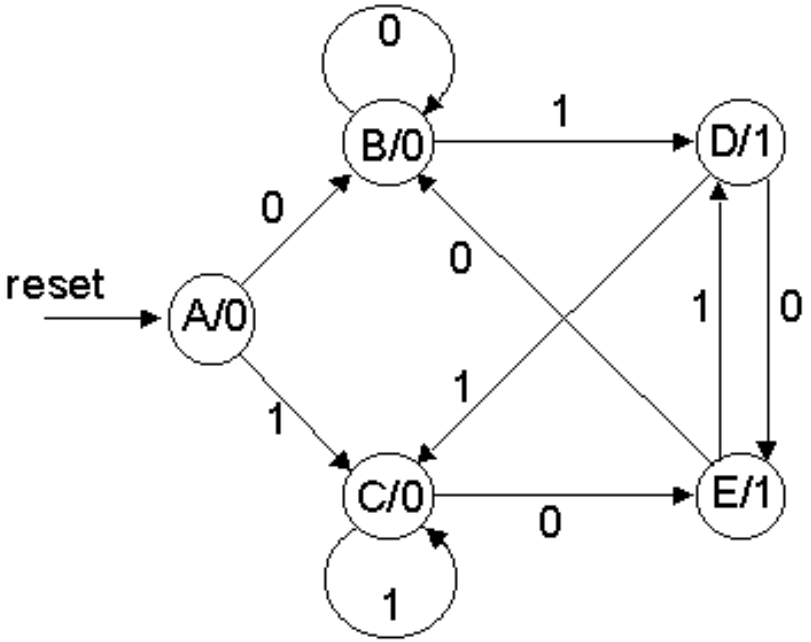
Automat Moore'a



Funkcja wyjść: $Y^t = f(S^t)$

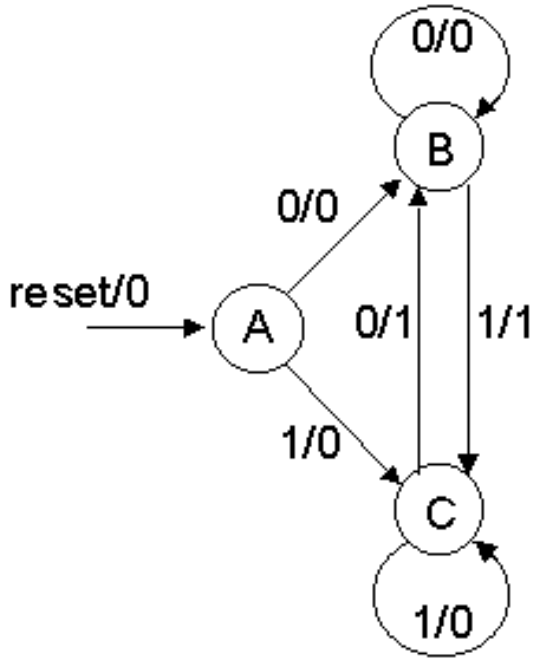


Wyjścia w modelu Moore'a



reset	input	current state	next state	output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	D	1
0	0	C	E	1
0	1	C	C	0
0	0	D	E	1
0	1	D	C	0
0	0	E	B	0
0	1	E	D	1

Wyjścia w modelu Mealy'ego



reset	input	current state	next state	output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	C	1
0	0	C	B	1
0	1	C	C	0

Asynchroniczne FSM

W asynchronicznych FSM zmiany następują w dowolnej chwili czasu. Szybkość działania takich układów zależy od wydajności źródeł napięcia i szybkości elementów przełączających. Bloki logiczne komunikują się metodą handshaking – dodatkowe komunikaty żądania i potwierdzenia dostępu.

Automaty asynchroniczne można traktować jak synchroniczne z nieskończenie wysoką częstotliwością taktowania.

Asynchroniczne FSM mogą być rozwiązaniem przyszłościowym – np. modele mieszane: podukłady synchroniczne w asynchronicznej ramie układowej.

Kiedy asynchroniczne FSM?

Układy synchroniczne zmieniają swój stan w takt sygnału zegarowego, przy czym zmiana napięcia wejściowego musi dokonywać się odpowiednio daleko od impulsu zegara (problem t_{hold} i t_{setup})

Powody użycia asynchronicznych FSM:
- wejścia układu zmieniają się w dowolnych chwilach,
- nie można pominąć opóźnień wprowadzanych przez ścieżki sygnałowe,
- układ działa zbyt wolno,
- moc pobierana na synchronizację układów jest zbyt duża.

Asynchroniczne FSM

Układy asynchroniczne:

- działanie nie wymaga synchronizacji zegarem
- stany mogą zmieniać się równocześnie ze zmianami wejścia
- przerzutniki nie muszą działać równocześnie

Podstawowy tryb pracy asynchronicznych FSM:

- wejścia mogą się zmieniać tylko, gdy układ jest w stanie stabilnym (tj. przy ustalonych sygnałach wewnętrznych)
- ważny jest poziom sygnału wejściowego, a nie jego zbocze czy czas trwania
- wejścia nie zmieniają się równocześnie (w danej chwili zmiana tylko jednego bitu)
- układy asynchroniczne można opisywać w postaci Moore'a lub Mealy'ego

Stany wewnętrzne i zupełne

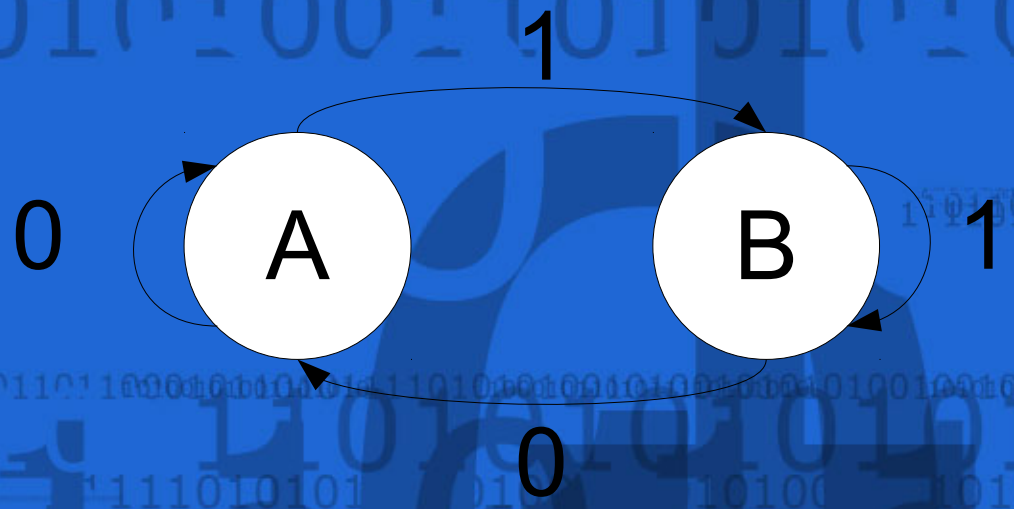
Stan wewnętrzny FSM przechowywany jest w elementach pamięciowych (przerzutnikach).

Stan zupełny FSM powstaje z połączenia stanu wewnętrznego i stanu wejść.

Stany stabilne FSM

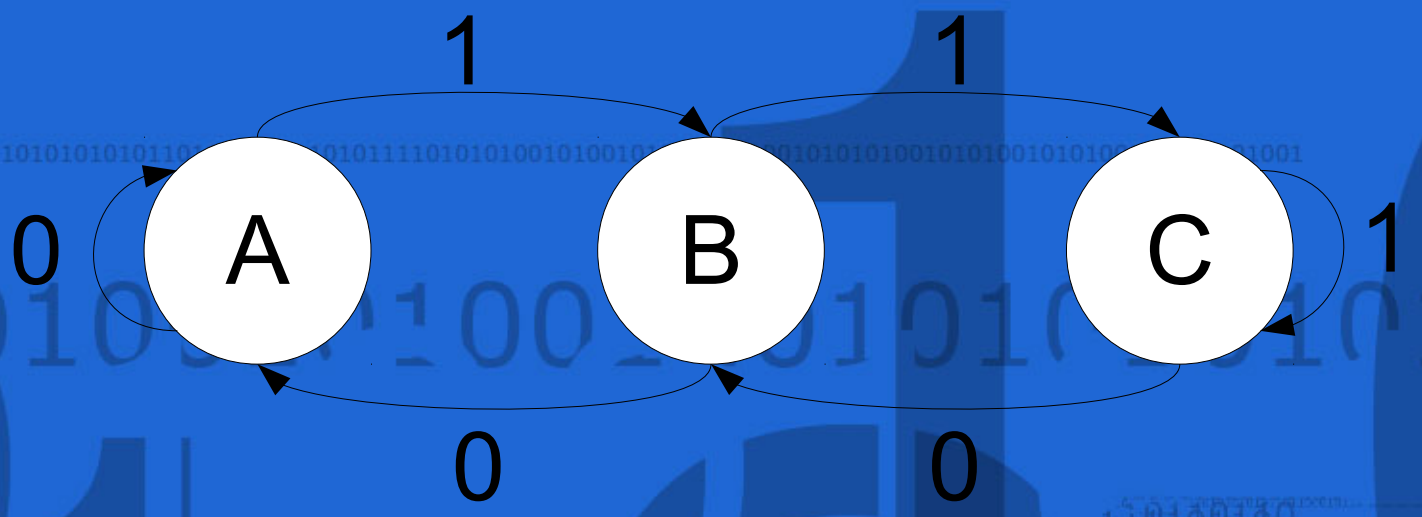
Stan stabilny FSM to stan, który nie zmienia się w czasie przy niezmieniającym się sygnale wejściowym → dotyczy tylko układów asynchronicznych!!!

Układ asynchroniczny może działać jedynie w obrębie **stanów stabilnych**.



Stany niestabilne FSM

Stan niestabilny FSM to stan, który zmienia się w czasie przy niezmieniającym się sygnale wejściowym.



Z uwagi na współistnienie stanów stabilnych i niestabilnych, w automatach asynchronicznych mówi się o przepływach, a nie przejściach.

Automat asynchroniczny
Etapy projektu

1. Budowa pierwotnego grafu przepływów lub tablicy przepływów
2. Eliminacja stanów nadmiarowych i budowa zredukowanej tablicy przepływów
3. Konwersja to postaci Mealy'ego
4. Kodowanie stanów
5. Budowa tablicy przejść-wyjść
6. Budowa tablicy wzbudzeń
7. Schemat układu

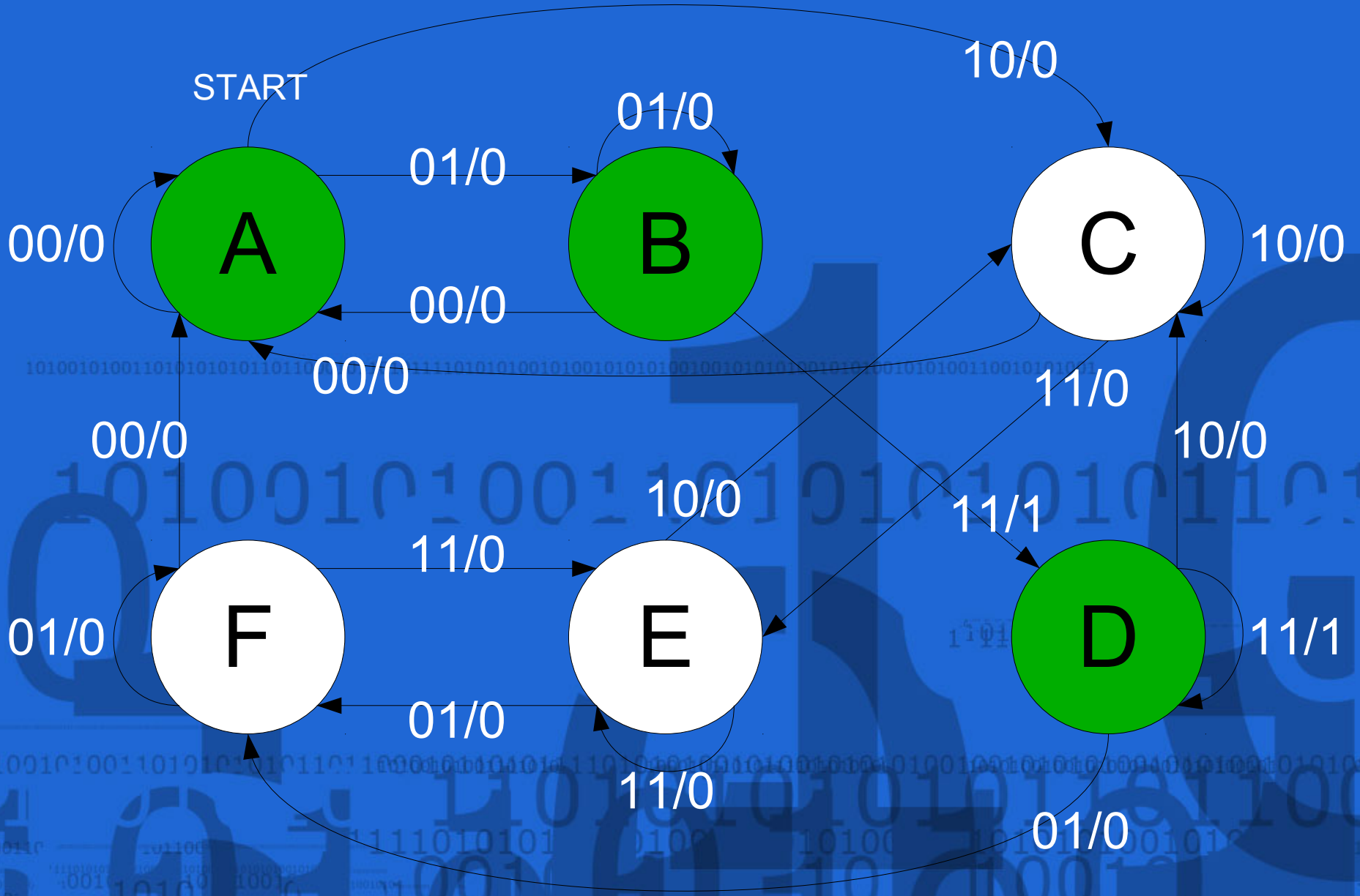
Asynchroniczny detektor sekwencji $00 \rightarrow 01 \rightarrow 11$

2-wejściowy układ asynchroniczny, którego wyjście zmienia się z 0 na 1 jeśli na wejściu pojawi się ww. sekwencja sygnałów.

Poprawne działanie układu wymaga wykorzystania 6 stanów stabilnych.

Ponieważ układ jest asynchroniczny, przepływy dokonują się tylko przy zmianie jednego z bitów wejścia (niemożliwa jednoczesna zmiana 2 i więcej bitów słowa wejściowego).

Graf przepływów 00 → 01 → 11



Pierwotna tablica przepływów

Dopuszczalny tylko jeden stan stabilny w każdym wierszu.
Każda zmiana wejścia musi powodować zmianę stanu wewnętrznego do kolejnego stanu stabilnego.

X1X2	00	01	11	10	Y
State 1(reset)	(1)	2	-	3	0
State 2(00→01)	1	(2)	4	-	0
State 3(00→10)	1	-	5	(3)	0
State 4(s2→11)	-	6	(4)	3	1
State 5(s3→11)	-	6	(5)	3	0
State 6(s3→11)	1	(6)	5	-	0

Usuwanie stanów nadmiarowych

Dwa stabilne stany zupełne są równoważne, gdy jednocześnie:

1. Ich wejścia są identyczne,
2. Ich wyjścia są identyczne,
3. Ich stany następne są identyczne dla tych samych wejść

Usuwanie stanów nadmiarowych
(inna tablica przepływów)

1. Wybierz wiersze zawierające stany stabilne w tych samych kolumnach

2. Porównaj wyjścia: zgodne? → OK,

3. Porównaj stany kolejne: zgodne? → OK

	00	01	11	10	Y1	Y2	
1	(1)	7	-	4	1	1	
2	(2)	5	-	4	0	1	
3	-	7	(3)	11	1	0	
4	2	-	3	(4)	0	0	
5	6	(5)	9	-	1	1	
6	(6)	7	-	11	0	1	
7	1	(7)	14	-	1	0	
~~8~~	~~(8)~~	~~12~~	~~-~~	~~4~~	~~0~~	~~1~~	→ 2
9	-	7	(9)	13	0	1	
~~10~~	~~-~~	~~7~~	~~(10)~~	~~4~~	~~1~~	~~0~~	→ 3
~~11~~	~~8~~	~~-~~	~~10~~	~~(11)~~	~~0~~	~~0~~	→ 4
~~12~~	~~6~~	~~(12)~~	~~9~~	~~-~~	~~1~~	~~1~~	→ 5
13	8	-	14	(13)	1	1	
14	-	12	(14)	11	0	0	

Uproszczona tablica przepływów

	00	01	11	10	Y1	Y2
1	(1)	7	-	4	1	1
2	(2)	5	-	4	0	1
3	-	7	(3)	11	1	0
4	2	-	3	(4)	0	0
5	6	(5)	9	-	1	1
6	(6)	7	-	11	0	1
7	1	(7)	14	-	1	0
9	-	7	(9)	13	0	1
13	8	-	14	(13)	1	1
14	-	12	(14)	11	0	0

Konwersja do postaci Mealy'ego

Inputs: X1X2	00	01	11	10	Ouput: Y 00	01	11	10
State 1(reset)	(1)	2	-	3	0	-	-	-
State 2(00→01)	1	(2)	4	-	-	0	-	-
State 3(00→10)	1	-	5	(3)	-	-	-	0
State 4(S2→11)	-	6	(4)	3	-	-	1	-
State 5(S3→11)	-	6	(5)	3	-	-	0	-
State 6(S3→11)	1	(6)	5	-	-	0	-	-

Kodowanie stanów wewnętrznych jest kluczowe dla asynchronicznego FSM

1. Kodowanie wpływa na szybkość działania i wielkość (liczbę elementów) FSM

2. Wykorzystanie słowa wyjściowego jako części kodu może pomóc zmniejszyć blok wyjściowy

3. Problem inicjalizacji: samostartujący FSM vs. asynchronicznie ustawiany FSM

Kodowanie stanów automatu asynchronicznego

1. Kodowanie proste (porządkowe),
2. Kodowanie w kodzie Graya
3. Kodowanie losowe
4. Kodowanie z pojedynczym gorącym bitem

Kodowanie stanów automatu asynchronicznego

Dla m-stanów wewnętrznych FSM i n-bitów kodu ($m \geq n \geq \log_2 m$) liczba możliwych kodów wynosi:

$$N_K = \frac{2^n!}{(2^n - m)!}$$

Ex.: n = 3, m = 4, N_K = 1680

Kodowanie z pojedynczym gorącym bitem

1. n-stanów kodowanych przy pomocy n-przerzutników

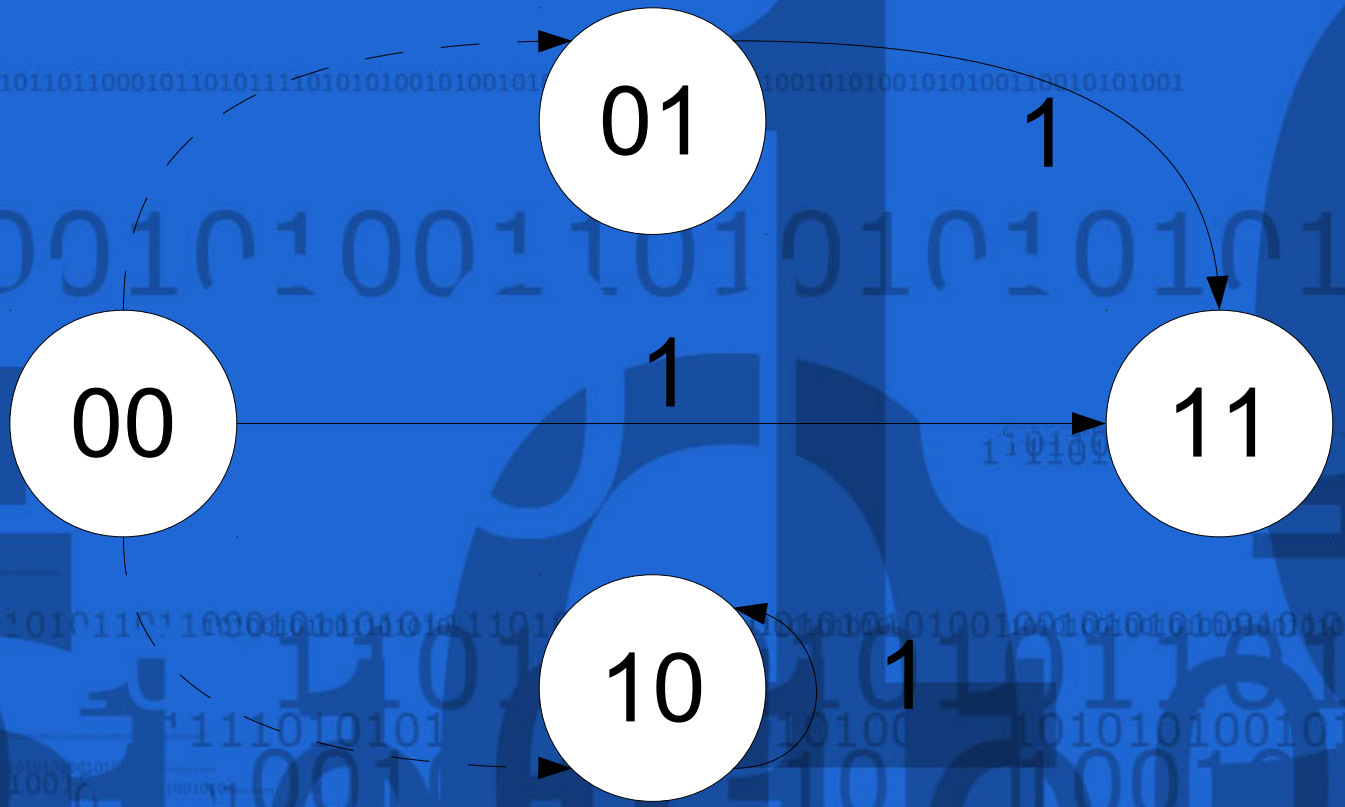
2. Łatwa kontrola poprawności działania bloku przerzutników

3. Skomplikowany diagram stanów → złożony układ

4. Rozbicie układu na 2 podukłady daje (n+m)-przerzutników zamiast (n*m).

Gonitwy w asynchronicznych FSM

Gonitwa to zmiana ścieżki działania automatu asynchronicznego na skutek niejednoczesnego przełączania bitów stanów wewnętrznych (opóźnienia układowe, opóźnienia ścieżek sygnałowych itp.) przy zmianie sygnału wejściowego.

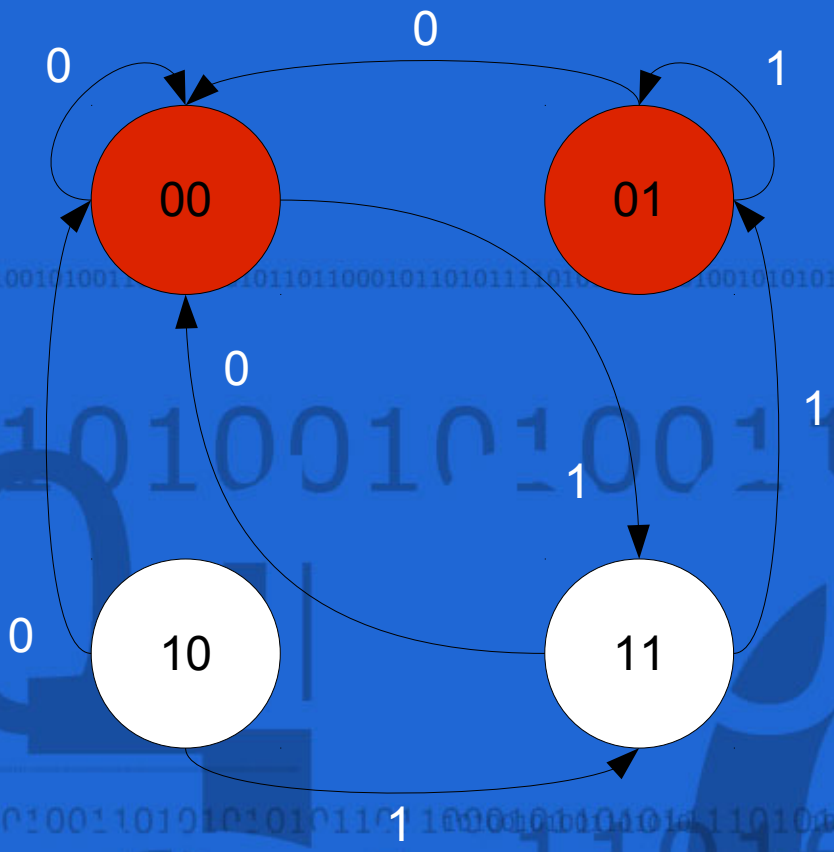


Gonitwy – zagrożenia

Występowanie gonitw nie jest własnością automatu asynchronicznego, ile wynika ze sposobu zakodowania jego stanów wewnętrznych. Gonitwy nie występują zatem w tablicach przepływów (stany opisane abstrakcyjnymi symbolami), a dopiero ew. w tablicach przejść.

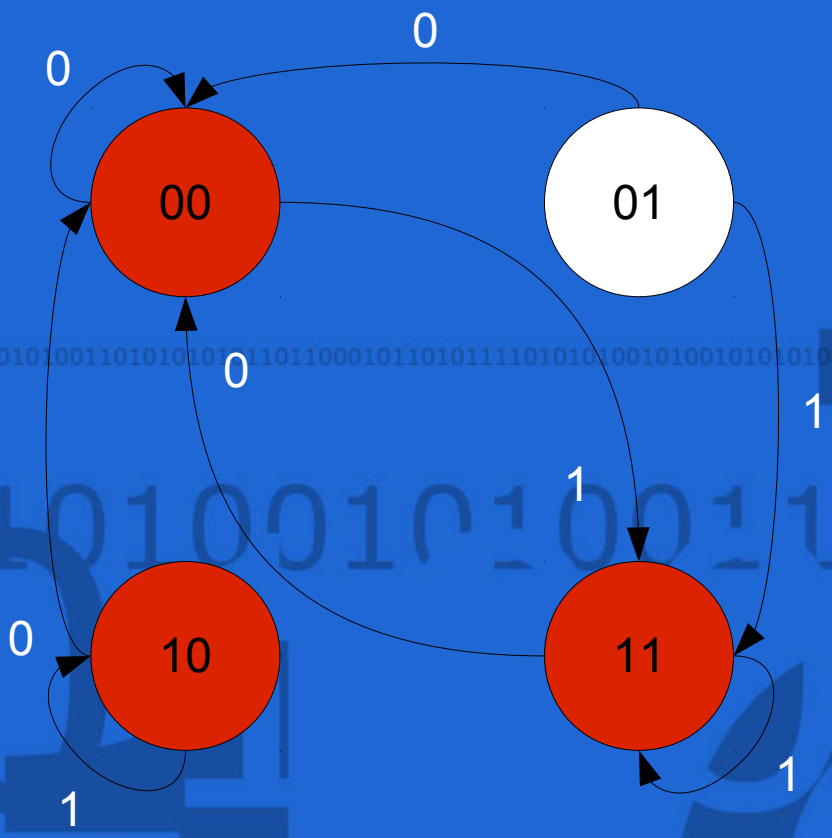
Rolą projektanta jest wyeliminowanie wyścigów, aby zapewnić kontrolowane i oczekiwane działanie układu.

Gonitwy niekrytyczne



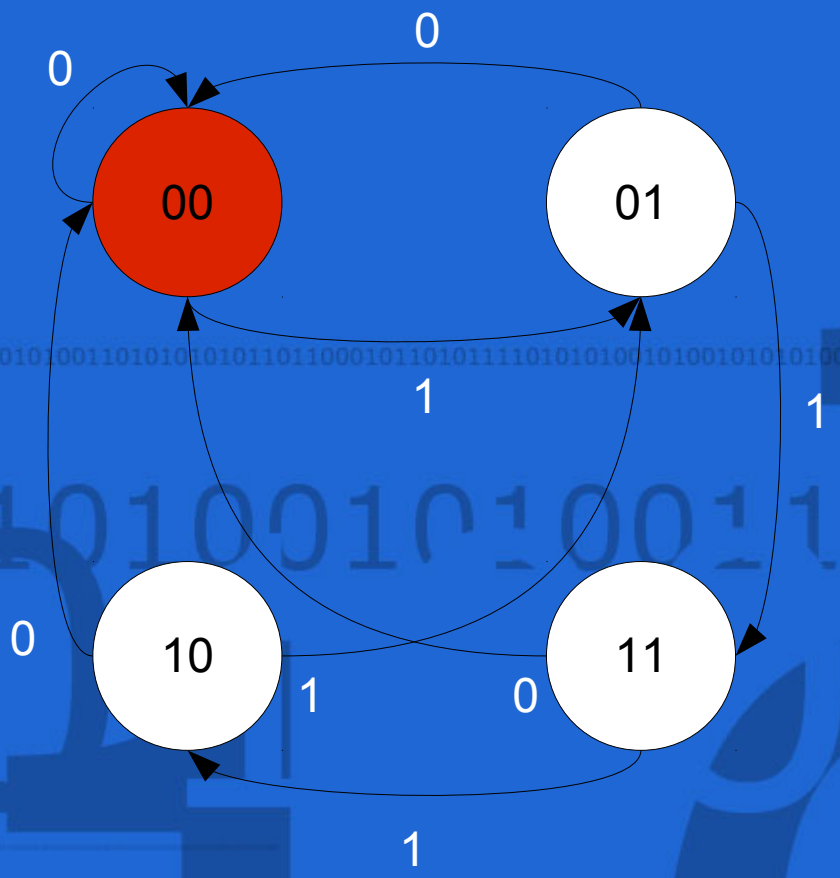
		x	
	y1y2	0	1
curr state	00	00	11
	01	00	01
	11	00	01
	10	00	11

Gonitwy krytyczne



		x	
	y1y2	0	1
curr state	00	00	11
	01	00	11
	11	00	11
	10	00	10

Oscylatory



		x	
	y1y2	0	1
curr state	00	00	01
	01	00	11
	11	00	10
	10	00	01

Bezgonitwowe kodowanie stanów

Aby wyeliminować gonitwy należy tak zakodować stany automatu, aby przy zmianie wejścia następowała zmiana tylko na jednym bicie stanu wewnętrznego, np.:

$$00 \rightarrow 01 \rightarrow 11$$
$$00 \rightarrow 10 \rightarrow 11$$

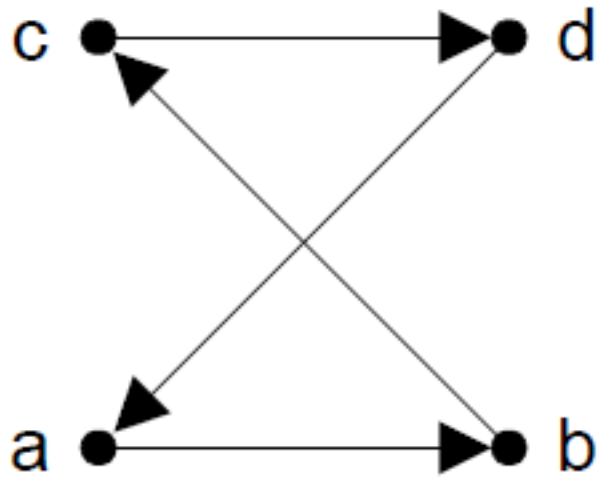
Metody BgKS – przykład

Wyjściowa tablica przepływów

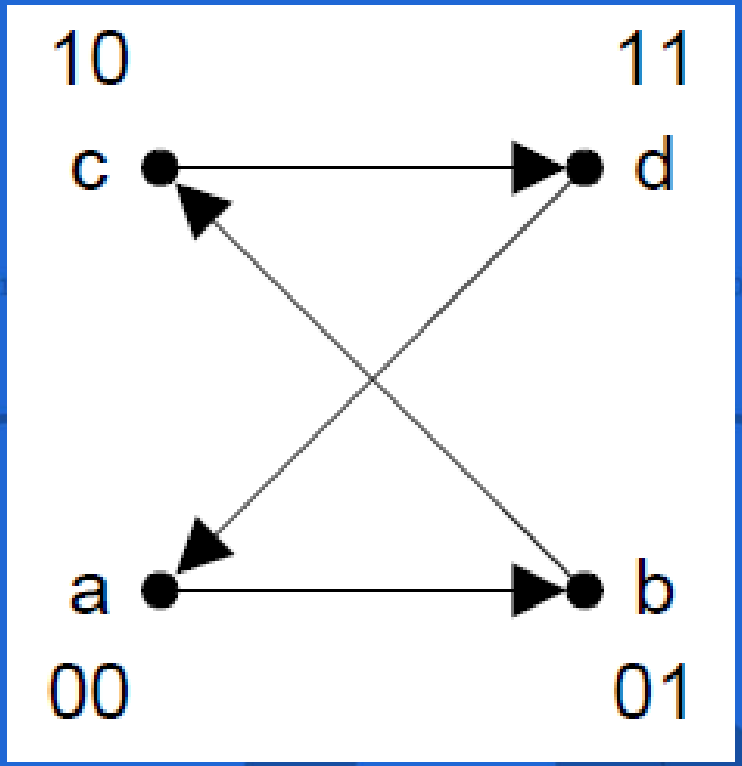
curr state	next state	
	x=0	x=1
a	(a)	b
b	c	(b)
c	(c)	d
d	a	(d)

Diagram stanów

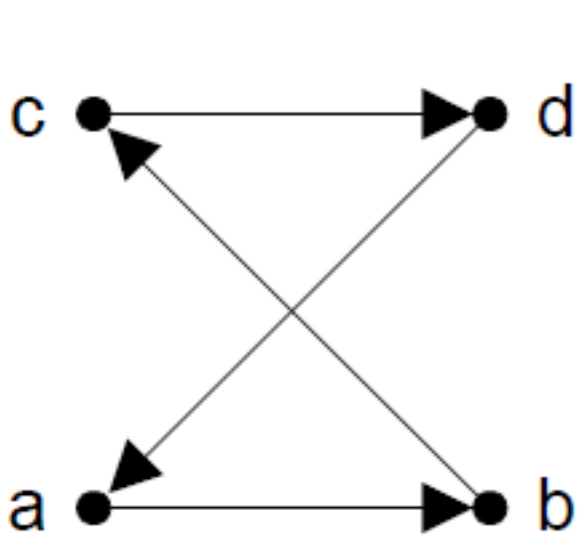
curr state	next state	
	x=0	x=1
a	(a)	b
b	c	(b)
c	(c)	d
d	a	(d)

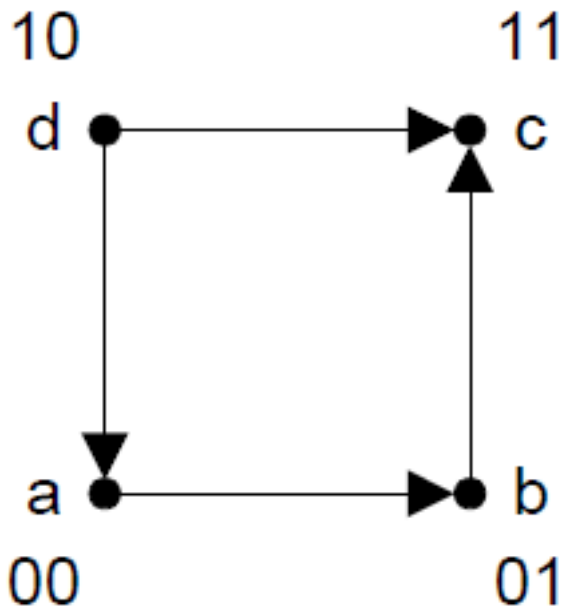


Złe przypisanie stanów



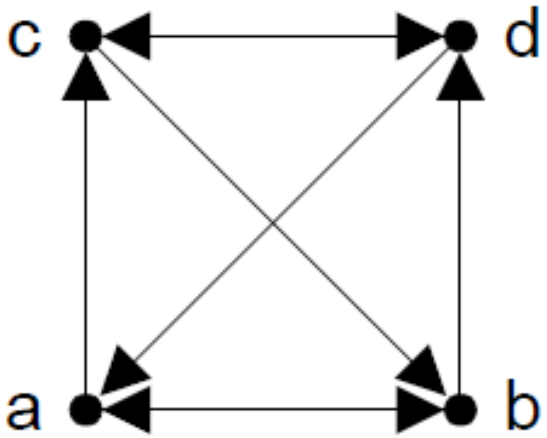
Poprawne przypisanie stanów





BgKS – dodawanie stanów niestabilnych

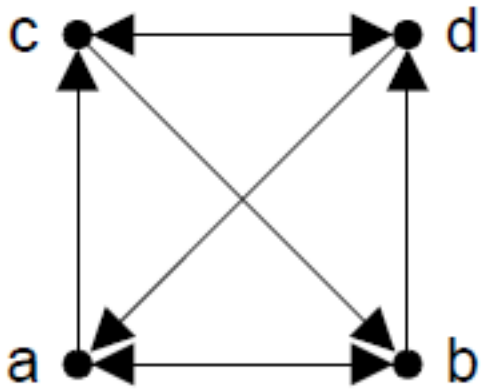
curr state	next state			
	x1x2=00	x1x2=01	x1x2=10	x1x2=11
a	(a)	(a)	c	b
b	a	(b)	d	(b)
c	(c)	b	(c)	d
d	c	a	(d)	(d)

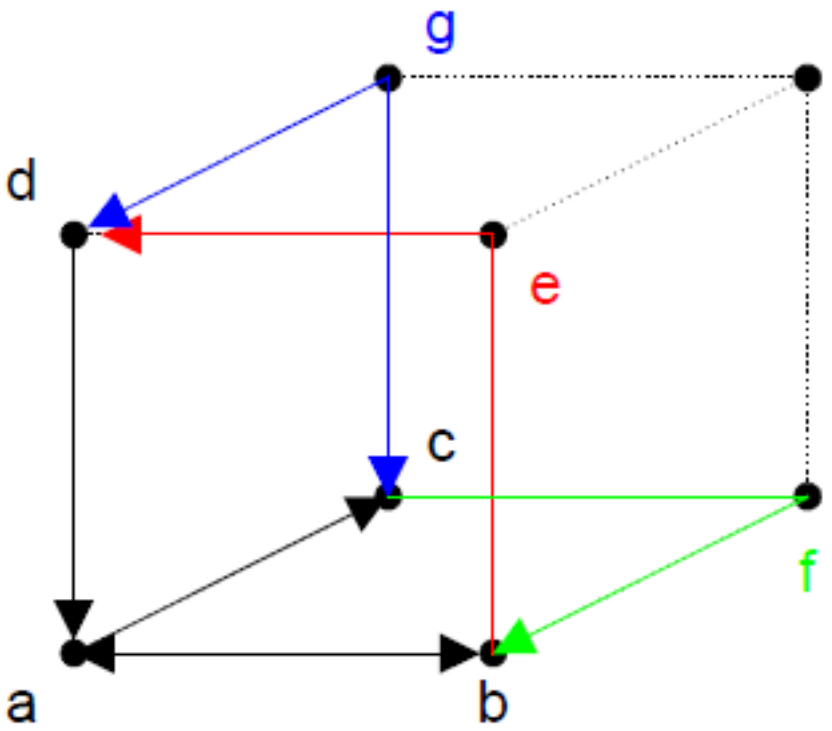


Nie istnieje dobre przypisanie stanów!

Rozwiązanie:
Wprowadzenie stanów **niestabilnych**

BgKS – dodawanie stanów niestabilnych





BgKS – dodawanie stanów niestabilnych

curr state	next state			
	x1x2=00	x1x2=01	x1x2=10	x1x2=11
a	(a)	(a)	c	b
b	a	(b)	d	(b)
c	(c)	b	(c)	d
d	c	a	(d)	(d)

curr state	next state			
	x1x2=00	x1x2=01	x1x2=10	x1x2=11
a	(a)	(a)	c	b
b	a	(b)	e	(b)
c	(c)	b	(c)	g
d	g	a	(d)	(d)
e	-	-	d	-
f	-	b	-	-
g	c	-	-	d

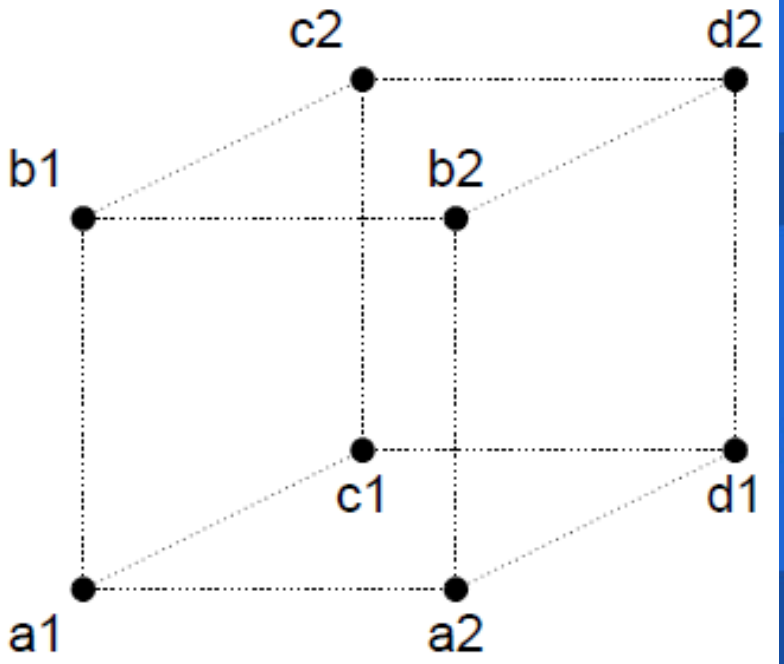
BgKS – dodawanie stanów równoważnych

curr state	next state				output
	x1x2=00	x1x2=01	x1x2=10	x1x2=11	
a	(a)	(a)	c	b	0
b	a	(b)	d	(b)	1
c	(c)	b	(c)	d	0
d	c	a	(d)	(d)	1

curr state	next state				output
	x1x2=00	x1x2=01	x1x2=10	x1x2=11	
a1	(a1)	(a1)	c1	b1	0
a2	(a2)	(a2)	a1	b2	0
b1	a1	(b1)	b2	(b1)	1
b2	a2	(b2)	d2	(b2)	1
c1	(c1)	c2	(c1)	d1	0
c2	(c2)	b1	(c2)	d2	0
d1	c1	d2	(d1)	(d1)	1
d2	c2	d1	(d2)	(d2)	1

BgKS – dodawanie stanów równoważnych

curr state	next state				output
	x1x2=00	x1x2=01	x1x2=10	x1x2=11	
a1	(a1)	(a1)	c1	b1	0
a2	(a2)	(a2)	a1	b2	0
b1	a1	(b1)	b2	(b1)	1
b2	a2	(b2)	d2	(b2)	1
c1	(c1)	c2	(c1)	d1	0
c2	(c2)	b1	(c2)	d2	0
d1	c1	d2	(d1)	(d1)	1
d2	c2	d1	(d2)	(d2)	1



BgKS – kodowanie z gorącym bitem

	curr state	next state			
		x1x2=00	x1x2=01	x1x2=10	x1x2=11
0001	a	(a)	(a)	c	b
0010	b	a	(b)	d	(b)
0100	c	(c)	b	(c)	d
1000	d	c	a	(d)	(d)

BgKS – kodowanie z gorącym bitem

	curr state	next state			
		x1x2=00	x1x2=01	x1x2=10	x1x2=11
0001	a	(a)	(a)	e	f
0010	b	f	(b)	g	(b)
0100	c	(c)	h	(c)	i
1000	d	c	j	(d)	(d)
0101	e	-	-	c	-
0011	f	a	-	-	b
1010	g	-	-	d	-
0110	h	-	b	-	-
1100	i	c	-	-	d
1001	i	-	a	-	-

	curr state	next state			
		x1x2=00	x1x2=01	x1x2=10	x1x2=11
0001	a	(a)	(a)	c	b
0010	b	a	(b)	d	(b)
0100	c	(c)	b	(c)	d
1000	d	c	a	(d)	(d)

- Tematy do opracowania:
- Minimalizacja liczby stanow automatu
- Tablica trojkatna
- Pokrycie minimalne