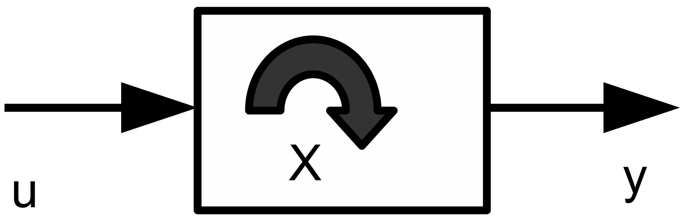
Układy kombinacyjne i sekwencyjne - przypomnienie

y = f(u)

Układ kombinacyjny



y = f(x,u)

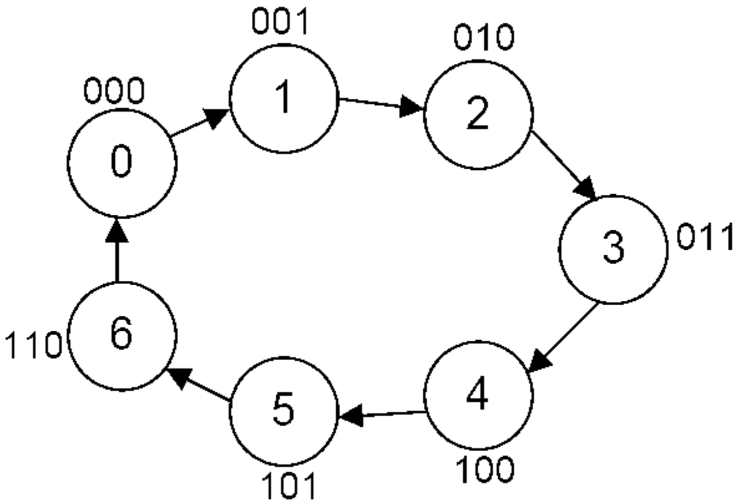
Układ sekwencyjny

Stan wewnętrzny

W *układach sekwencyjnych* wprowadza się pojęcie *stanu wewnętrznego*.

- *Stan wewnętrzny* "ogranicza" możliwość zmiany (ewolucji) układu, gdyż przyszła wartość stanu zależy od jej wartości obecnej i wartości wejść,
- Możliwe zachowania się układu sekwencyjnego modeluje się za pomocą **grafu skierowanego -** węzły modelują *stan* i związaną z nim *wartości wyjść*, strzałki określają zmianę stanu pod wpływem *wartości wejść*,
- Ze *stanem zewnętrznym* związana jest wartość wyjść (obserwowalna),
- *Stan wewnętrzny* może nie być "widoczny" na wyjściu (dlatego jest "wewnętrzny"),
- W *układach sekwencyjnych* stany wewnętrzne realizuje się za pomocą *przerzutników* i elementów kombinacyjnych.

Pojęcie stanu wewnętrznego - graf przejść



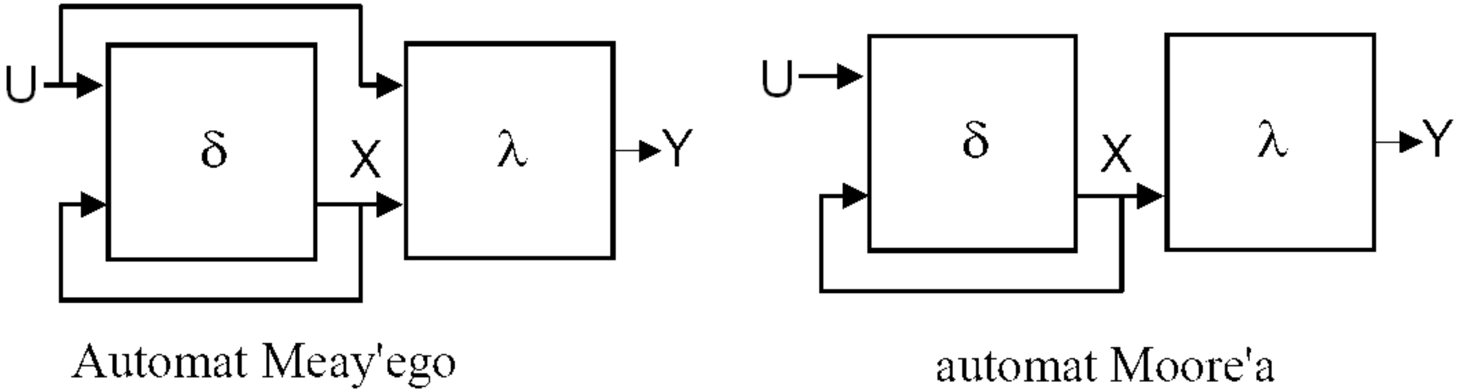
Przykładowy graf modelujący zachowanie się układu sekwencyjnego.

Kodowanie stanu

S_t	S_{t+1}	Y_t
S_0	S_1	0
S_1	S_2	1
S_2	S_3	2
S_3	S_4	3
S_4	S_5	4
S_5	S_6	5
S_6	S_0	6

kodowanie	S_t	S_{t+1}	Y_t
$S_0 \rightarrow 000$	000	001	000
$S_1 \rightarrow 001$	001	010	001
$S_2 \rightarrow 010$	010	011	010
$S_3 \rightarrow 011$	011	100	011
$S_4 \rightarrow 100$	100	101	100
$S_5 \rightarrow 101$	101	110	101
$S_6 \rightarrow 110$	110	000	110

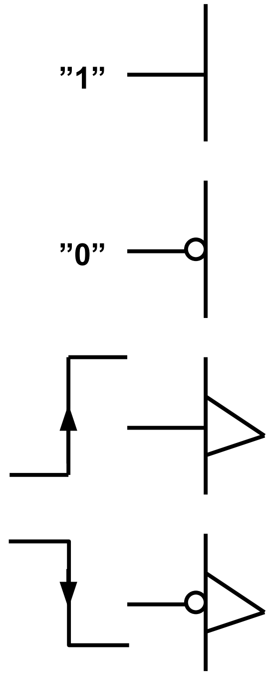
Automaty Mealy'ego i Moore'a



Układ (λ) realizujący *funkcję wyjść* jest układem kombinacyjnym a blok (δ) realizuje pamięć (*funkcję wzbudzeń*).

$$\begin{cases} x(k+1) = \delta(x(k), u(k)) & \text{równanie stanu: } \delta \text{ - \textit{funkcja wzbudzeń}} \\ y(k) = \lambda(x(k), u(k)) & \text{równanie wyjścia: } \lambda \text{ - \textit{funkcja wyjść}} \end{cases}$$

Zegar



- W układach sekwencyjnych istotna jest sekwencja stanów,
- W celu synchronizacji tych zmian wprowadza się dodatkowe wejście zwane *wejściem zegarowym* lub *zegarem*,
- W przeważającej liczbie przypadków zmiana stanu odbywa się wraz z dodatnim *zboczem zegara*.

Projektowanie układów sekwencyjnych

Etapy realizacji układów sekwencyjnych

Przy projektowaniu *układów sekwencyjnych* częściej wykorzystuje się *automaty Moore'a*. W dalszej części wykładu skupimy się tylko na tego typu układach. Wyróżnia się następujące etapy projektowania:

1. W oparciu o treść (cel) zadania tworzymy graf skierowany modelujący zachowanie się automatu. Węzły grafu modelują stany i związane z nim wartości wyjść. Zmiana stanu jest opisywana poprzez strzałkę, z którą związana jest wartość wejścia. Zmiana stanu musi uwzględniać wszystkie kombinacje wejść - w przeciwnym przypadku graf zachowywałby się w sposób niedeterministyczny.

2. Kodujemy stany, wykorzystując najmniejszą liczbę zmiennych,

3. "Rozbijamy" zakodowaną tabelę przejść stanów na pojedyncze przerzutniki, określając funkcję wzbudzeń dla każdego przerzutnika.

4. Dla każdego wyjścia określamy mapę Karnough'a mapującą stany (wyjścia przerzutników) na wyjścia układu.

Przykład

Zadanie:
Zaprojektuj i zrealizuj układ o dwóch wejściach i dwóch wyjściach, działający w następujący sposób:

1. Jeśli na wejściu pojawią się dwie jedynki układ powinien na przemian włączać oba wyjścia i wyłączać oba wyjścia dodatnim zboczem zegara C. Logiczna "1" na wyjściu oznacza, że wyjście jest włączone, w przeciwnym przypadku jest wyłączone.

2. Dwa zera na wejściu powodują wyłączenie obu wyjść.

3. Pozostałe kombinacje wejść powodują, że wyjścia zachowują się jak licznik $mod3$.

Określenie liczby stanów i grafu przejść

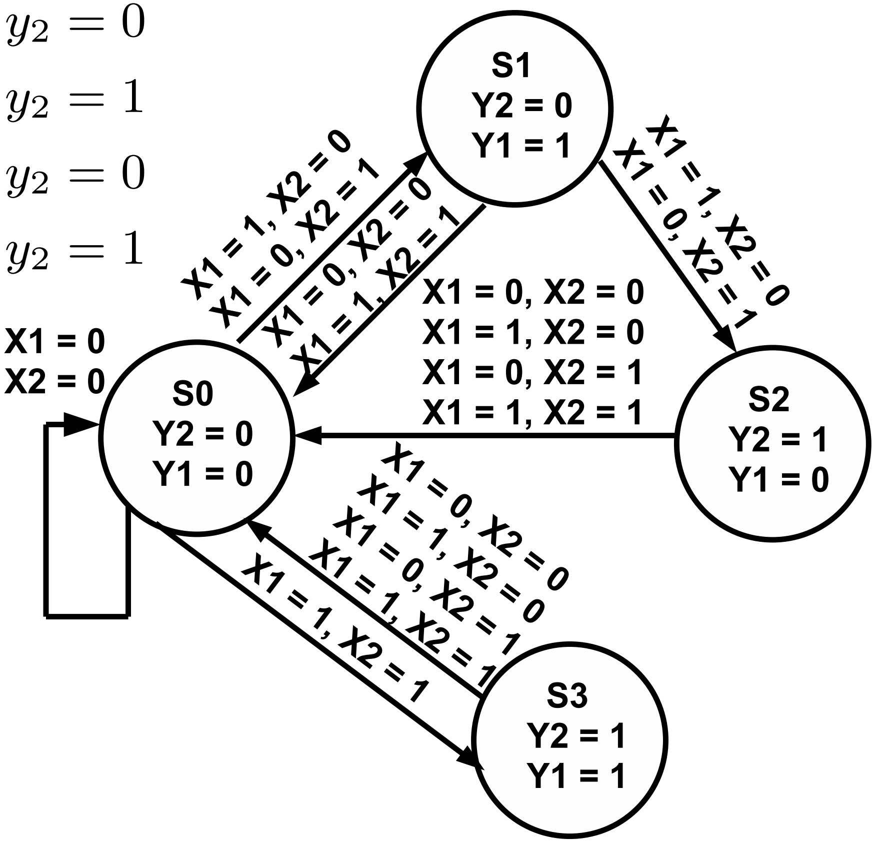
Ponieważ na wyjściach mogą się pojawić wszystkie możliwe kombinacje (4 różne) i z każdym wyjściem związany jest jeden *stan wewnętrzny* oznacza to, że minimalna liczba stanów wynosi 4:

Dla stanu S_0 wyjścia wynoszą $y_1 = 0$ i $y_2 = 0$

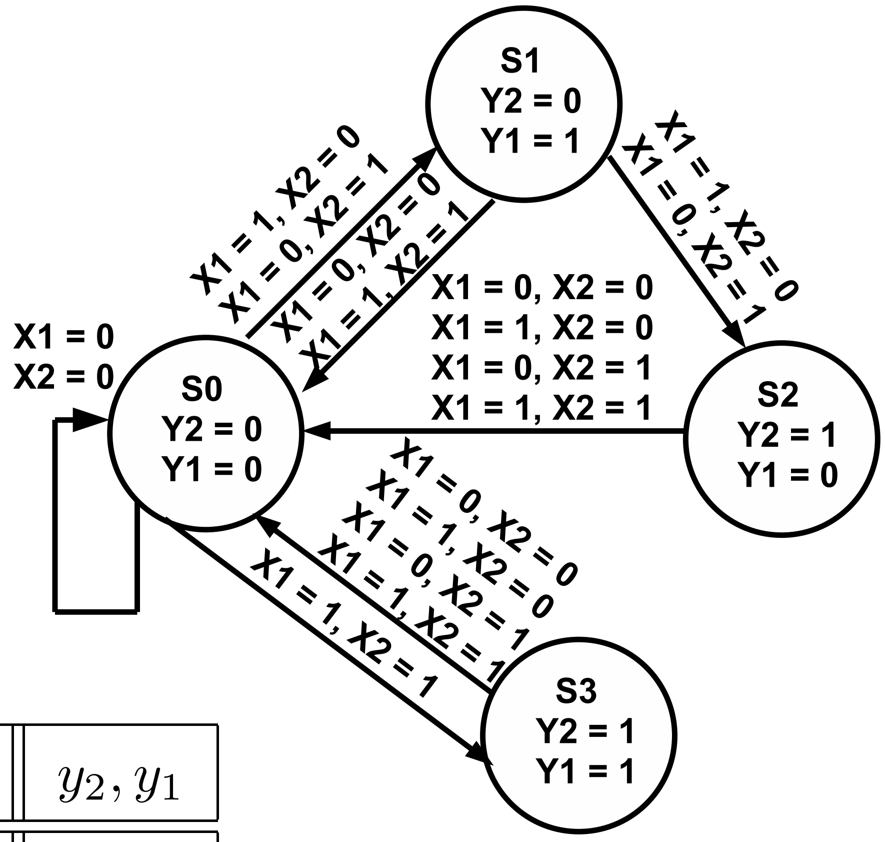
Dla stanu S_1 wyjścia wynoszą $y_1 = 0$ i $y_2 = 1$

Dla stanu S_2 wyjścia wynoszą $y_1 = 1$ i $y_2 = 0$

Dla stanu S_3 wyjścia wynoszą $y_1 = 1$ i $y_2 = 1$



Określenie tablicy przejść



S^{t+1} :

$S^t \backslash x_2, x_1$	00	01	11	10	y_2, y_1
S_0	S_0	S_1	S_3	S_1	00
S_1	S_0	S_2	S_0	S_2	01
S_2	S_0	S_0	S_0	S_0	10
S_3	S_0	S_0	S_0	S_0	11

Kodowanie tablicy przejść

S^{t+1} :

$S^t \backslash x_2, x_1$	00	01	11	10	y_2, y_1
S_0	S_0	S_1	S_3	S_1	00
S_1	S_0	S_2	S_0	S_2	01
S_2	S_0	S_0	S_0	S_0	10
S_3	S_0	S_0	S_0	S_0	11

Q_1^{t+1}, Q_0^{t+1} :

$\frac{x_2, x_1}{Q_1^t, Q_0^t}$	00	01	11	10	y_2, y_1
$S_0 \mapsto 00$	00	01	10	01	00
$S_1 \mapsto 01$	00	11	00	11	01
$S_2 \mapsto 11$	00	00	00	00	10
$S_3 \mapsto 10$	00	00	00	00	11

Kodowanie stanów dla przerzutnikach D

Q_1^{t+1}, Q_0^{t+1} :

$\frac{x_2,x_1}{Q_1^t,Q_0^t}$	00	01	11	10	y_2, y_1
$S_0 \mapsto 00$	00	01	10	01	00
$S_1 \mapsto 01$	00	11	00	11	01
$S_2 \mapsto 11$	00	00	00	00	10
$S_3 \mapsto 10$	00	00	00	00	11

D_1 :

$\frac{x_2,x_1}{Q_1^t,Q_0^t}$	00	01	11	10
00	0	0	1	0
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

D_0 :

$\frac{x_2,x_1}{Q_1^t,Q_0^t}$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

$D_1 = \overline{Q_1}Q_0\overline{x_2}x_1 + \overline{Q_1Q_0}x_2x_1 + \overline{Q_1}Q_0x_2\overline{x_1}$

$D_0 = \overline{Q_1}\,\overline{x_2}x_1 + \overline{Q_1}x_2\overline{x_1}$

Określenie funkcji wyjść

y_2, y_1 :

$\frac{x_2,x_1}{Q_1^t,Q_0^t}$	00	01	11	10	y_2, y_1
$S_0 \mapsto 00$	00	01	10	01	00
$S_1 \mapsto 01$	00	11	00	11	01
$S_2 \mapsto 11$	00	00	00	00	10
$S_3 \mapsto 10$	00	00	00	00	11

y_2 :

$Q_1 \backslash Q_0$	0	1
0	0	0
1	1	1

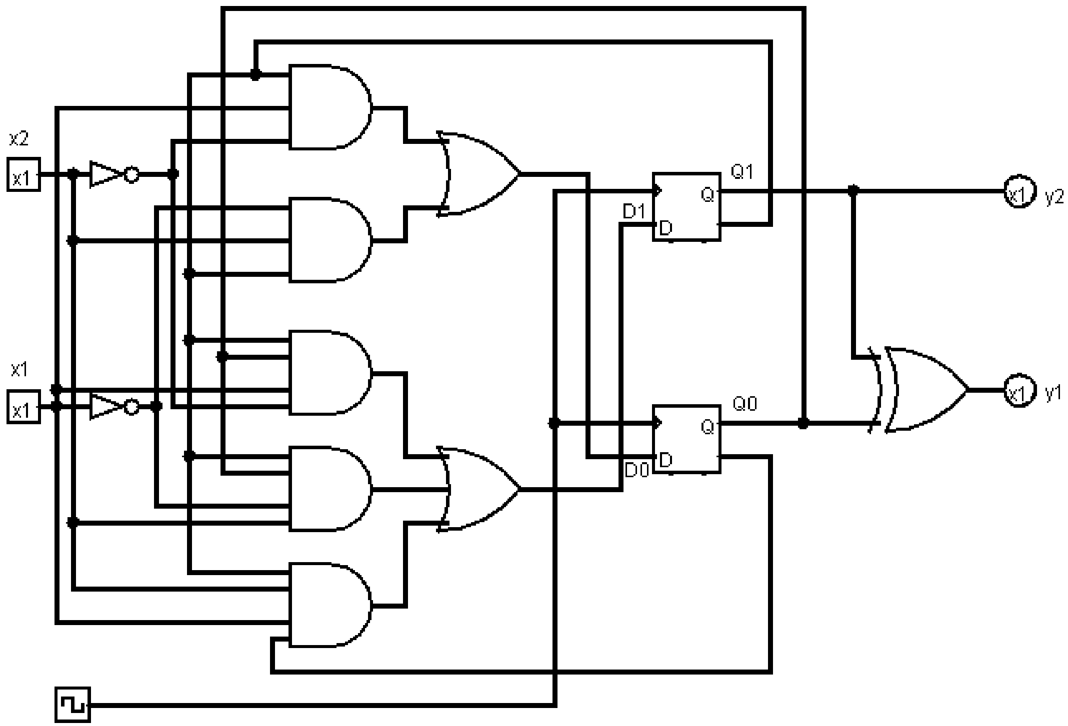
y_1 :

$Q_1 \backslash Q_0$	0	1
0	0	1
1	1	0

$y_2 = Q_1$

$y_1 = \overline{Q_1}Q_0 + Q_1\overline{Q_0} = Q_1 \otimes Q_0$

Realizacja układu



- funkcje wzbudzeń: $D_0 = \overline{Q_1}\,\overline{x_2}x_1 + \overline{Q_1}x_2\overline{x_1}$
  $D_1 = \overline{Q_1}Q_0\overline{x_2}x_1 + \overline{Q_1 Q_0}x_2x_1 + \overline{Q_1}Q_0x_2\overline{x_1}$
- funkcje wyjścia: $y_2 = Q_1$
  $y_1 = \overline{Q_1}Q_0 + Q_1\overline{Q_0} = Q_1 \otimes Q_0$

Kodowanie stanów

Przy kodowaniu stanów istotne jest to, aby osiągnąć możliwie prostą funkcję wzbudzeń. Niestety nie ma ogólnej metody, która prowadziłaby zawsze do minimalnej funkcji wzbudzeń. Możemy próbować to osiągnąć przestrzegając poniższych zasad (w/g priorytetu):

- **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, mającym ten sam stan bieżący, słowa kodowe różniące się tylko wartością jednego bitu,
- **Zasada 3:** - Należy przyporządkować stanom, dla których występują takie same wartości wyjściowe (przy tych samych sygnałach wejściowych) słowa kodowe różniące się wartością tylko jednego bitu.

Minimalizacja stanów - równoważność stanów

Dwa stany są *równoważne* (mogą być zastąpione jednym stanem) jeśli spełniają warunki:

1. Wartości sygnałów na wyjściach, związanych z dwoma stanami są takie same,
2. Odpowiadające im stany następne są takie same lub *równoważne*.

Minimalizacja stanów - przykład

S^{t+1} :

$S^t \backslash x_2, x_1$	00	01	11	10	Y - wyjście
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

- Nie ma stanów identycznych
- Ze względu wyjścia zgodne są stany: 3 i 7 oraz stany: 1, 2, 4, 5 i 6.
- Potencjalne pary zgodne: $\{1,2\}$, $\{1,4\}$, $\{1,5\}$, $\{1,6\}$, $\{2,4\}$, $\{2,5\}$, $\{2,6\}$, $\{3,7\}$, $\{4,5\}$, $\{4,6\}$, $\{5,6\}$,

Minimalizacja stanów - wybór stanów zgodnych

$S^t \backslash x_2, x_1$	00	01	11	10	Y - wyjście
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

potencjalne pary zgodne	Pary, które powinny być zgodne	Niezgodność	1 iteracja	2 iteracja
$\{1,2\}$	$\{1,2\},\{1,4\}$			
$\{1,4\}$	$\{1,2\}$			
$\{1,5\}$	$\{5,6\},\{3,7\},\{1,2\}$	niezgodne		bo $\{5,6\}$
$\{1,6\}$	$\{3,5\},\{1,2\},\{1,7\}$	niezgodne	bo $\{3,5\}$	
$\{2,4\}$	$\{1,2\},\{2,4\}$			
$\{2,5\}$	$\{5,6\},\{3,7\},\{1,4\}$	niezgodne		bo $\{5,6\}$
$\{2,6\}$	$\{3,5\},\{4,7\}$	niezgodne	bo $\{3,5\}$	
$\{3,7\}$	$\{3,7\},\{1,4\}$			
$\{4,5\}$	$\{5,6\},\{3,7\},\{1,2\}$	niezgodne		bo $\{5,6\}$
$\{4,6\}$	$\{3,5\},\{1,2\},\{2,7\}$	niezgodne	bo $\{3,5\},\{2,7\}$	
$\{5,6\}$	$\{3,6\},\{3,7\},\{1,7\}$	niezgodne	bo $\{3,6\},\{1,7\}$	

- A - stany $\{1,2,4\}$ B - stany $\{3,7\}$ C - stan $\{5\}$ D - stan $\{6\}$

Minimalizacja stanów - tablica uproszczona

- A - stany $\{1, 2, 4\}$ B - stany $\{3, 7\}$ C - stan $\{5\}$ D - stan $\{6\}$

$S^t \backslash x_2, x_1$	00	01	11	10	Y - wyjście
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

$S^t \backslash x_2, x_1$	00	01	11	10	Y - wyjście
A	C	B	A	A	0
B	B	A	A	C	1
C	D	B	A	A	0
D	B	B	A	B	0

Systemy nie w pełni określone

W systemach nie wpełni określonych możemy się zetknąć, gdy z treści zadania wynika, że

- nie pojawia się pewna kombinacja wejść - np. gdy w zadaniu ze slajdu 9, punkt 3 układ będzie liczył tylko przy kombinacji wejść $x_2x_1 = 10$ (tj. dla kombinacji wejść $x_2x_1 = 01$ zachowanie układu jest wówczas nieokreślone),
- Przy kodowaniu stanów zostają "stany nadmiarowe" np. licznik $mod3$ (wymaga 3 stanów, które można zrealizować na min. 2 przerzutnikach - 4 stany)

Przy minimalizacji stanów nieokreślonych możemy nadać im dowolne wartości - tak jak nam wygodnie.

Automaty Meal'ego - uwagi

- W automatach Meal'ego wyjście nie jest jednoznacznie związane ze stanem,
- W układach Meal'ego istnieje możliwość większej redukcji stanów kosztem bardziej rozbudowanej funkcji wyjścia,
- Przy tworzeniu grafu wartość wyjścia zależeć będzie od stany jak i od wejścia.

Zadania na ćwiczenia

1. Zaprojektuj i zrealizuj, używając przerzutników typu D, automat Moore'a liczący $mod3$ wstecz tj. generujący sekwencję 10 - 01 - 00 gdy wejście jest równe 1. Pojawienie się 0 na wejściu powinno powodować wygaszenie wyjść - oba wyjścia powinny mieć wartość 0.

2. Dla zadania z punktu 1. zmień kodowanie stanów - użyj kod stanu, który był pominięty w poprzednim punkcie. Zaprojektuj i zrealizuj układ.