

# Konstruowanie automatów sekwencyjnych w środowisku Matlab Simulink

**Streszczenie.** W artykule omówiono algorytmy, pozwalające wyznaczać funkcje logiczne automatów sekwencyjnych. Algorytmy zostały zaimplementowane w programie Matlab. Zaimplementowany algorytm został wykorzystany do wyznaczenia funkcji logicznych przykładowego automatu. Zostały wyznaczone stany wewnętrzne oraz funkcje logiczne automatu. Wyznaczone funkcje logiczne zostały wykorzystane do symulacji pracy automatu w programie Fluidsim.

**Abstract.** The article discusses the algorithms that allow, to find, logic functions for finite state machines. The algorithms were implemented in MATLAB program. The implemented algorithm was used to determine the internal states and logic functions for finite state machine. Determined logic has been used to simulate of automata in FluidSIM program. (Constructing logics of finite-state machine in the Matlab Simulink environment).

**Słowa kluczowe:** automat sekwencyjny, funkcje logiczne, Matlab S-simulink, sterownik PLC

**Keywords:** finite state machine, logic function, matlab simulink, PLC

doi:10.12915/pe.2014.01.43

## Wstęp

Procesory są stosowane w coraz większej ilości urządzeń zarówno w przemyśle jak w urządzeniach domowego użytku. W systemach produkcyjnych wykorzystuje się powszechnie układy oparte na procesorach. W systemach sterowania stosowane są sterowniki PLC. Same procesory są połączeniem specjalizowanych elektronicznych układów kombinacyjnych i sekwencyjnych.

Przy projektowaniu systemów sterowania np. sekwencyjnych układów pneumatycznych, które są stosowane w zautomatyzowanych liniach produkcyjnych mamy do dyspozycji zaawansowane sterowniki PLC oparte na procesorach. Pomimo tak zaawansowanych narzędzi przy projektowaniu systemu sterowania tj. programowaniu układów z procesorami takich jak sterowniki PLC konieczna jest znajomość metod, które zostały wykorzystane do projektowania samych procesorów. Szczególnie istotne są metody, które pozwalają projektować logikę układów sekwencyjnych.

Do programowania sterowników PLC powszechnie stosowane są metody intuicyjne. Niestety w przypadku zaawansowanych automatów z wieloma wejściami i wejściami zaprojektowane w ten sposób funkcje są często zawodne.

W artykule został omówiony algorytm, który pozwala wyznaczyć funkcje logiczne automatu sekwencyjnego pozwalające niezawodnie programować sterowniki PLC.

Założono, że automat steruje procesem produkcyjnym, w którym sygnały wyjściowe są związane z sygnałami wejściowymi przez podany cyklogram pracy. Założono, że automat ma jeden cykl pracy. Na cyklogramie w jednym przedziale czasu pracuje tylko jeden element wykonawczy np.: siłownik oraz nie ma założeń o dodatkowych cyklach pracy. Praca siłowników jest zsynchronizowana. Pewne modele synchronizacji zostały przedstawione w pracy [4].

Algorytm wyznaczania funkcji logicznych został zaimplementowany w środowisku Matlab. W artykule omówiono działanie algorytm na przykładzie systemu z dwoma siłownikami.

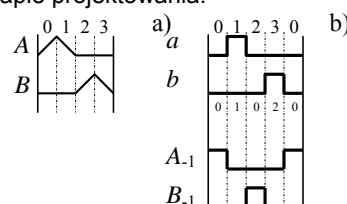
## Przykład konstrukcji automatu sekwencyjnego

W rozdziale zostało omówione działanie algorytm dodawania pamięci na przykładzie automatu sterującego dwoma siłownikami.

Sygnały wyjściowe automatu sterują elementami wykonawczymi takimi jak siłowniki pneumatyczne, silniki itp.

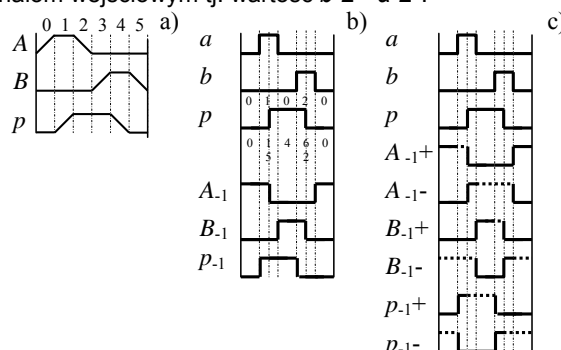
Poprzez ich załączanie i wyłączanie. Sygnały wejściowe pochodzą od elementów krańcowych siłowników itp.

W przypadku układu sterującego zespołem siłowników podany jest cyklogram (Rys.1a) reprezentujący pracę dwóch siłowników A, B. Dla takiego układu sygnały wejściowe i wyjściowe automatu są ze sobą związane (Rys.1b) poprzez cyklogram. Dwa sygnały wejściowe z czujników krańcowych siłownika zastąpiono jednym sygnałem na etapie projektowania.



Rys.1. a) Cyklogram pracy dwóch siłowników A, B, b) Sygnały wejściowe i wyjściowe sterownika układu

Omówiona metoda opiera się na regułach pozwalających dodawać sygnały do wykresu sygnałów wejściowych. Wykresy sygnałów wejściowych i wyjściowych pod dodaniem pamięci  $p$  oraz sygnału załączania pamięci  $p-1$  przedstawia (Rys.2b,c). Na wykresie sygnałów (Rys.2b) pod sygnałami wejściowymi  $a, b$  zaznaczono liczby są to wartości dziesiętne liczby binarnej  $(b, a)$  odpowiadającej sygnałom wejściowym tj. wartość  $b \cdot 2^1 + a \cdot 2^0$ .



Rys.2. a) cyklogram pracy, b) wykres sygnałów wejściowych i wyjściowych

W wyniku dodania sygnału  $p$  powstały dwa dodatkowe przedziały pracy automatu. Pod sygnałami  $a, b, p$  zaznaczono liczby są to wartości dziesiętne liczby binarnej  $(p, b, a)$  odpowiadającej sygnałom wejściowym i stanom tj.

wartości  $p \cdot 2^2 + b \cdot 2^1 + a \cdot 2^0$ . Wartości te są różne w każdym przedziale pracy automatu. Sygnał załączający dodaną pamięć tj. sygnał  $p_{-1}$  jest traktowany jest jako sygnał wyjściowy ze sterownika i zmienia wartość na brzegu przedziału wcześniejszego niż dodany sygnał  $p$ . Na rysunku (Rys.2b) sygnały  $A_{-1}$ ,  $B_{-1}$ ,  $p_{-1}$  dla automatu z przerzutnikami SR na wejściu oraz (Rys.2c) sygnały  $A_{-1}^+$ ,  $A_{-1}^-$ ,  $B_{-1}^+$ ,  $B_{-1}^-$ ,  $p_{-1}^+$ ,  $p_{-1}^-$ , dla automatu z przerzutnikami SR na wyjściu.

Po do daniu sygnałów pamięci na podstawie wykresu sygnałów można utworzyć minimalną tablicę przejść i wyjść automatu Mealy'ego Tabela 1. W tablicy w nawiasach zaznaczono liczby odpowiadające przedziałom czasu pracy automatu.

Tabela 1. Minimalna tablica przejść sygnał  $p_{-1}$  i wyjść  $A_{-1}$ ,  $B_{-1}$  automatu Mealy'ego

Q	Wejście a, b			
	00	01	11	10
0	(0)0	(2)0	-	(1)1
1	(4)1	(6)0	-	(5)1

Wejście a, b				
00	01	11	10	
(0)10	(2)00	-	(1)10	
(4)01	(6)01	-	(5)00	

### Reguły dodawania stanów

W programie napisanym w środowisku matlab zostały zastosowane następujące reguły dodawania stanów:

1) Przeglądane są indeksy odpowiadające sygnałom wejściowym, dodanym stanom, jeśli indeks się powtórzy dodawana jest pamięć i załączana w przedziale poprzedzającym pierwsze powtórzenie indeksu. Przedział ten dzielony jest na dwa przedziały.

2) Po załączeniu pamięci przeglądane są kolejne indeksy i szukany jest ostatni przedział, dla którego, wystąpiło powtórzenie danego indeksu.

a) Pamięć jest wyłączana w przedziale następnym po ostatnim powtórzeniu indeksu.

b) Jeśli indeks powtarza się tylko raz pamięć wyłączana jest w przedziale, w którym oraz wszystkich następnym nie ma indeksu, który wystąpił przed załączeniem pamięci. Jeśli takiego przedziału nie ma pamięć wyłączana jest w przedziale ostatnim gdzie jest indeks 0.

Został napisany skrypt w matlab do wyznaczania minimalnych tablic automatu Mealy'ego za pomocą podanych reguł.

Dane do programu podawane są w postaci tablic reprezentujących wartości logiczne sygnałów wejściowych tablica A, tabele sygnałów wyjściowych tabela B.

Tabela A powinna mieć wartości zerowe wszystkich sygnałów w pierwszym i ostatnim przedziale czasu pracy automatu.

### Funkcje logiczne automatu

Funkcje logiczne opisujące sygnały wejściowe od wyjściowych można wyznaczyć za pomocą tablic minimalizacji Karnaugh. Tablice te tworzone są na podstawie tabel sygnałów po dodaniu pamięci. Przebieg i ilość sygnałów wyjściowych zależy od tego np.: czy na wyjściach automatu są, czy też nie przerzutniki np. SR.

Jeśli na wyjściach oraz jako pamięć zostaną zastosowane przerzutniki SR upraszcza to funkcje logiczne, ponieważ w sygnałach wyjściowych dopuszcza się sygnały dowolne. Konieczne jest wtedy dwukrotne zwiększenie ilości funkcji logicznych  $A_{-1}^+$ ,  $A_{-1}^-$ ,  $B_{-1}^+$ ,  $B_{-1}^-$ ,  $p_{-1}^+$ ,  $p_{-1}^-$ , ponieważ każdy przerzutnik SR ma dwa sygnały sterujące.

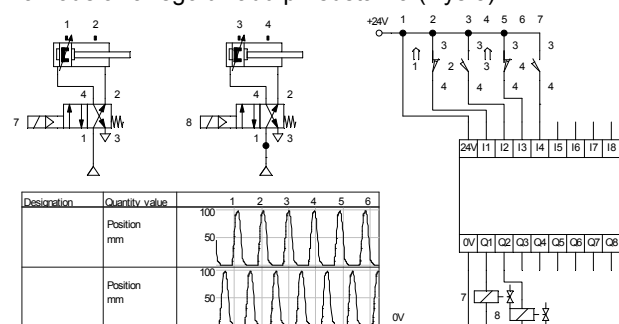
Wyznaczone funkcje logiczne podano w równaniach 1 dla automatu, dla którego przerzutnik SR jest na wejściu, sygnały  $A_{-1}$ ,  $B_{-1}$ ,  $p_{-1}$  oraz równania (2) dla automatu, w którym przerzutnik SR jest na wyjściu sygnały  $A_{-1}^+$ ,  $A_{-1}^-$ ,  $B_{-1}^+$ ,  $B_{-1}^-$ ,  $p_{-1}^+$ ,  $p_{-1}^-$ .

$$(1) \quad A_{-1} = \overline{b}p_1, \quad B_{-1} = ap_1, \quad p_{-1} = a \vee bp_1$$

$$(2) \quad \begin{aligned} A_{-1}^+ &= \overline{b}p_1, & B_{-1}^+ &= ap, & p_{-1}^+ &= a \\ A_{-1}^- &= p, & B_{-1}^- &= \overline{p_1}, & p_{-1}^- &= b \end{aligned}$$

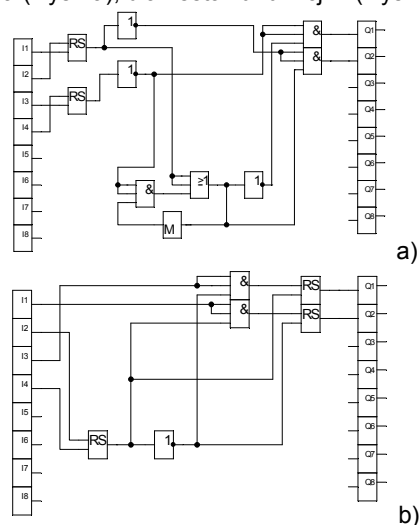
Przy budowaniu automatu sterującego układem siłowników, zastosowano uproszczenie polegające na zastąpieniu dwóch sygnałów wejściowych z czujników krańcowych siłownika jednym sygnałem. Dzięki temu upraszcza się etap projektowania (Rys.1), (Rys.2). Warunek ten nie jest spełniony, dlatego zastosowanie zestawu funkcji  $A_{-1}$ ,  $B_{-1}$ ,  $p_{-1}$  jest możliwe jedynie po dodaniu przerzutników SR na wejściu sterownika.

Przykład zastosowania wyznaczonych funkcji logicznych przedstawiono w programie fluidsims. Schemat zamodelowanego układu przedstawia (Rys.3).



Rys.3. Schemat układu sterowania dwoma siłownikami ze sterownikiem PLC fluidsims.

Schemat logiki sterownika dla zestawu funkcji 1, przedstawia (Rys.4a), dla zestawu funkcji 2 (Rys.4b).



Rys.4. Funkcje logiczne w programie fluidsims a) przerzutniki SR na wejściach, b) przerzutniki SR na wyjściach

Pisząc program za pomocą schematu drabinkowego na sterownik PLC są dostępne bloki przerzutnika SR na wyjściu  $-(S)-$ ,  $-(R)-$  oraz wyjście bez przerzutnika  $-()-$ . W przypadku sterownika PLC możliwe jest wykorzystanie obu zestawów wyznaczonych funkcji logicznych równania 1, 2.

### Algorytm minimalizacji funkcji logicznych

Algorytm wyznaczania funkcji logicznych został zaimplementowany w skrypcie programu matlab. Danymi wejściowymi skryptu są tabele sygnałów wejściowych oraz sygnałów wyjściowych po dodaniu pamięci. Algorytm wyznaczania funkcji logicznych oparty jest metodzie Quine'a McCluskeya [2]. Inne metody optymalizacji zostały przedstawione w pracy [3]. Schemat działania programu pokazano na przykładzie sygnału  $A_{-1}$ .

Wyznaczana jest jednowymiarowa kolumnowa macierz minimalizacji Karnaugh tabela 2 dla danego sygnału wyjściowego. Z tabeli odrzucane są wiersze, dla których w macierzy są zera w kolumnie Y reprezentującej wartość sygnału wyjściowego.

Tabela 2. Tabela początkowa sygnału A<sub>1</sub>+

Wejście			Y
p <sub>1</sub>	b	a	
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	∅
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	∅

W wyniku odrzucenia wierszy otrzymuje się macierz drugą tabela 3.

Tabela 3, 4, 5. Tabele minimalizacji funkcji logicznych sygnału A<sub>1</sub>+

Wejście			Y
p <sub>1</sub>	b	a	
0	0	0	1
0	0	1	1
0	1	1	∅
1	1	1	∅

Wejście			Y
p <sub>1</sub>	b	a	
0	0	∅	1
0	∅	1	1
∅	1	1	∅

Wejście			Y
p <sub>1</sub>	b	a	
0	0	∅	1
0	∅	1	1

W macierzy drugiej szukane są wiersze różniące się na jednej pozycji sygnału wejściowego oraz takie, które na pozycjach rozróżniających wiersze mają wartości (1, 0). W rozpatrywanym przypadku są to wiersze 1-2, 2-3 oraz 3-4. Wiersze są łączone i przepisywane do nowej tablicy. W pozycji, w której wiersze się różnią wstawiany jest symbol sygnału dowolnego ∅. Wiersze, które nie zostały połączone są przepisywane. W pole odpowiadające sygnałowi wyjściowemu Y wpisywana jest wartość 1 albo ∅. Wartość 1, jeśli dla łączonych wierszy, przynajmniej jeden miał wartość 1, wartość ∅, jeśli dla żadnego z łączonych wierszy nie było wartości 1 w polu sygnału wyjściowego. W wyniku otrzymuje się tabele 4. Łączenie wierszy prowadzi się do momentu, w którym nie jest możliwe połączenie żadnych dwóch wierszy. Z powstałej w ten sposób macierzy tablica 4 odrzuca się wiersze, dla których w polach sygnału wyjściowego są wartości dowolne w wyniku uzyskuje się macierz tabela 5. Wiersze macierzy 5 odpowiadają elementom sumy funkcji logicznej  $A_{-1+} = b\overline{p_1} \vee ap_1$ .

Uzyskana funkcja dla sygnału A<sub>1</sub> nie jest minimalna. Dlatego w kolejnej części algorytmu stosowana jest procedura usuwania z powstałej tablicy końcowej niepotrzebnych składowych. W algorytmie szukane są składowe funkcji logicznej, które można odrzucić. Składowe odrzucane są pojedynczo w krokach. W każdym kroku odrzucania składowych tworzone są nowe macierze OZ oraz Z.

Przykład działania algorytmu odrzucania składowych funkcji logicznych podano w tabelach 6, 7, 8. Tabela 6 to przepisana tablica 3. Tablica 7 zawierają informacje, jakie pola tabeli 6 pokrywa dana składowa funkcji logicznej. Tabela 8 zawiera informacje, które składowe funkcji logicznej można odrzucić.

W tabeli 7 wstawiane są jedynki, jeśli dana składowa funkcji logicznej wiersz tablicy 5 pokrywa pola tablicy 3. Tabela 8 tworzona jest na podstawie tablic 6, 7. W tabeli 8

wstawiane są jedynki, jeśli dane pole tabeli minimalizacji 6 jest pokryte tylko przez tą funkcję oraz na wyjściu jest sygnał 1. Finalnie składowe można odrzucić, jeśli kolumna macierzy 8 zawiera same zera.

Tabela. 6, 7, 8. tablica 23, tabela informacji o polach pokrytych przez składowe funkcji logicznej, tabela informacji, która składowa do usunięcia.

Wejście			Y
p <sub>1</sub>	b	a	
0	0	0	1
0	0	1	1
0	1	1	∅
1	1	1	∅

Wiersz Tab. 5	
1	2
1	0
1	1
0	1
0	0

Wiersz Tab. 5	
1	2
1	0
0	0
0	0
0	0

Został napisany skrypt, który za pomocą omówionego algorytmu wyznacza zestawy 1 oraz 2 funkcji logicznych dla dowolnego automatu o jednym cyklu pracy. Należy zaznaczyć, że problem wyznaczania funkcji logicznych jest to problem, który ma złożoność wykładniczą 2<sup>N</sup>, gdzie N to rozmiar problemu zarówno ze względu na pamięci oraz ilość operacji arytmetycznych (czas działania).

### Podsumowanie i Wnioski

Celem pracy, której dotyczy artykuł było stworzenie oprogramowania w matlab wspomagające programowanie sterowników PLC. W artykule zostały przedstawione algorytmy pozwalające wyznaczać funkcje logiczne dla automatu z jednym cyklem pracy.

Zostały omówione zaproponowane reguły pozwalające dodawać sygnały pamięci do wykresów sygnałów automatu.

Omówiono zaimplementowany algorytm wyznaczania funkcji logicznych metoda Quine'a McCluskeya z usuwaniem nadmiarowych składowych.

W wyniku zastosowania omówionych algorytmów powstało oprogramowanie pozwalające wyznaczać tablice przejść i wyjść minimalnego automatu Mealy'ego oraz funkcje logiczne automatu konieczne do poprawnego projektowania sterowników PLC.

Ze względu na złożoność obliczeniową wyznaczenie funkcji logicznych jest możliwe dla automatów o niewielkiej ilości wejść.

W dalszej pracy autor zamierza sformułować i zaimplementować algorytm pozwalające budować w podobny sposób na tj: na podstawie cyklogramu automat dla dowolnej liczby cykli pracy. W przypadku automatu o kilku cyklach pracy konieczna jest modyfikacja algorytmu pozwalającego dodawać stany do wykresu sygnałów.

### LITERATURA

- [1] Węsierski Ł., Podstawy logiki i wnioskowania. *Oficyna Wydawnicza Politechniki Rzeszowskiej*, 2004
- [2] Żelazny M., Podstawy automatyki. *PWN* 1976
- [3] Szostek R., Zastosowanie teorii kolejek do modelowania procesów zachodzących w urządzeniach liczących. *Elektrotechnika i Elektronika*, t. 18, z. 3, Kraków 1999, 81-88
- [4] Szostek R., Modelowanie systemów współbieżnych za pomocą sieci kolejkowych, *Zarządzanie i Marketing*, Rzeszów 2007, z. 11, nr 245, pp. 153-165
- [5] Traczyk W., Układy cyfrowe automatyki. *WNT*, Warszawa 1974
- [6] Majewski W., Układy logiczne. *WNT*, Warszawa 1974

**Autorzy:** dr inż. Karol Szostek, Politechnika Rzeszowska, Katedra Termodynamiki i Mechaniki Płynów, ul. Powstańców Warszawy 8, 35-959 Rzeszów, E-mail: kszostek@prz.edu.pl.