



Implementacja automatu współbieżnego z wykorzystaniem mikroprogramowanych układów sterujących

Obecnie przy projektowaniu systemów cyfrowych bierze się pod uwagę takie kryteria, jak efektywne wykorzystanie zasobów sprzętowych oraz niski pobór energii. Jest to szczególnie ważne w przemyśle, gdzie kładzie się szczególny nacisk na zmniejszenie kosztów produkcji masowej oraz przy projektowaniu urządzeń przenośnych.

W praktyce często są wykorzystywane automaty współbieżne, dla których jest konieczne opracowanie efektywnych metod syntezy. Najczęściej stosowaną metodą syntezy układu cyfrowego jest dekompozycja automatu współbieżnego na sekwencyjne automaty składowe.

W artykule zaprezentowano metodę implementacji automatu współbieżnego z wykorzystaniem mikroprogramowanych układów sterujących (*Compositional Microprogram Control Unit – CMCU*) opisanych powiązanymi sieciami działań (*Flow-Chart of Algorithm – FCA*). Dla implementacji i optymalizacji tych automatów zaproponowano następującą ścieżkę projektową:

- utworzenie opisu działania układu cyfrowego za pomocą diagramu **SFC** – (*Sequential Function Chart*),
- konwersja diagramu SFC na interpretowaną sieć Petriego
- pozycja automatu współbieżnego opisanego siecią Petriego na sekwencyjne automaty składowe,
- przekształcenie powstałych automatów składowych na sieci działań,
- implementacja układu cyfrowego opisanego powiązanymi sieciami działań z wykorzystaniem mikroprogramowanych układów sterujących.

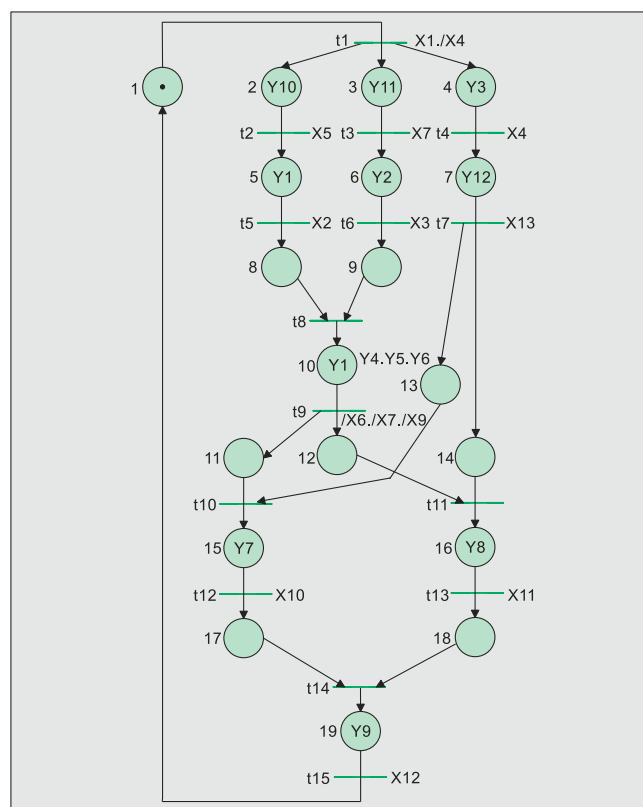
Metoda opisu działania układu cyfrowego za pomocą diagramu SFC oraz sposób konwersji SFC na sieć Petriego została przedstawiona w [2, 6].

DEKOMPOZYCJA AUTOMATU WSPÓLBIEŻNEGO

Podczas projektowania może zaistnieć sytuacja, gdy rozmiar układu wykracza poza ramy narzucone rozmiarami sterownika. Dokonuje się wtedy jego podziału na szereg współpracujących ze sobą podukładów. Dekompozycja układu ułatwia proces projektowania, a jej wyniki mogą być wykorzystane do efektywnego kodowania lokalnych stanów wewnętrznych układu sterującego. Metoda dekompozycji sieci Petriego na podsieci typu automatowego zostanie przedstawiona na przykładzie procesu produkcji napojów [1]. W artykule zostaną przedstawione tylko wyniki dekompozycji, szczegóły można znaleźć w pracy [2, 3].

System sterowania produkcją napojów rozpoczyna działanie po naciśnięciu przycisku startowego x1. Wówczas rozpoczyna się napełnianie zbiorników 1 i 2 oraz dostarczanie pojemników do napojów. Składniki są przekazywane do zbiornika

1 przez zawór y10, do zbiornika 2 przez zawór y11, aż do momentu ich napełnienia. Napełnienie zbiornika 1 jest wskazywane stanem czujnika x5=1, zbiornika 2 – stanem czujnika x7=1. Dostarczanie pojemników odbywa się za pomocą wózka pod wpływem sygnału y12 i kończy się sygnałem x13. Gdy zbiornik 1 lub 2 jest napełniany, następuje przygotowywanie składników, zakończone odpowiednimi sygnałami x2=1 (zbiornik 1), x3=1 (zbiornik 2) i x4=1 (pojemniki). Przygotowane składniki ulegają mieszaniu w pojemniku 3, do którego są wpuszczane zaworami y5 i y6. Zawory te zostają zamknięte po całkowitym opróżnieniu zbiorników 1, 2 i wymieszaniu składników napoju, co jest sygnalizowane odpowiednio czujnikami x6=0, x8=0, x9=0. Gdy pojemniki są gotowe, następuje niezależne od siebie napełnianie i zamykanie pojemników 5 i 6, zasygnalizowane odpowiednio sygnałami x10 i x11. Obydwa pojemniki równocześnie są kierowane do pasteryzacji. Po zakończeniu operacji (x12=1) układ jest gotowy do pracy.



■ Rys. 1. Sieć Petriego opisująca działanie urządzenia do produkcji napojów

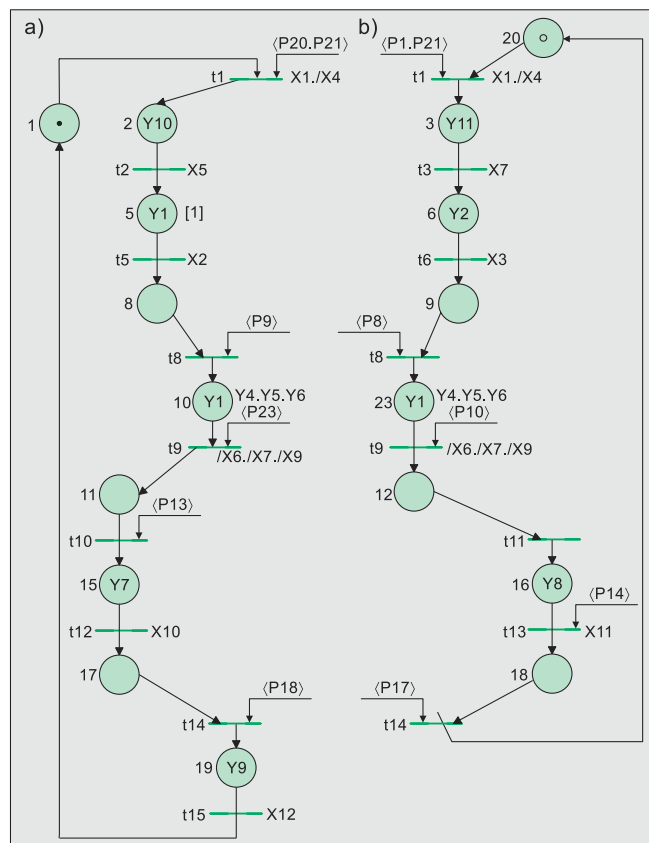
Na rys. 1 został przedstawiony model w postaci sieci Petriego, opisujący proces produkcji napojów. Sieć Petriego składa się z dziewiętnastu kroków opisanych w tabeli 1. Opis warunków, które muszą być spełnione, w celu realizacji tranzycji, oraz opis wyjść zamieszczono w tabeli 2.

* Uniwersytet Zielonogórski, Instytut Informatyki i Elektroniki,
e-mail: M.kolopienczyk@iie.uz.zgora.pl

■ Tabela. 1. Opis stanów lokalnych sterownika

Stan lokalny	Opis
p1	Stan początkowy
p2	Napełnianie zbiornika 1
p3	Napełnianie zbiornika 2
p4	Załadowanie pojemników
p5	Przygotowanie składnika 1
p6	Przygotowanie składnika 2
p7	Wózek jedzie w lewo
p10	Otwarcie zaworów zbiorników; mieszanie składników
p15	Napełnienie pojemnika 1
p16	Napełnienie pojemnika 2
p19	Wózek jedzie w prawo
p8, p9, p11, p12, p13, p14, p17, p18	Oczekiwanie

Szczegółowy opis poszczególnych etapów dekompozycji sieci Petriego na sekwencyjne automaty składowe można znaleźć w pracy [1]. W wyniku dekompozycji automatu współbieżnego, opisanego siecią Petriego (rys. 1), otrzymano cztery sekwencyjne automaty składowe (rys. 2, rys. 3).

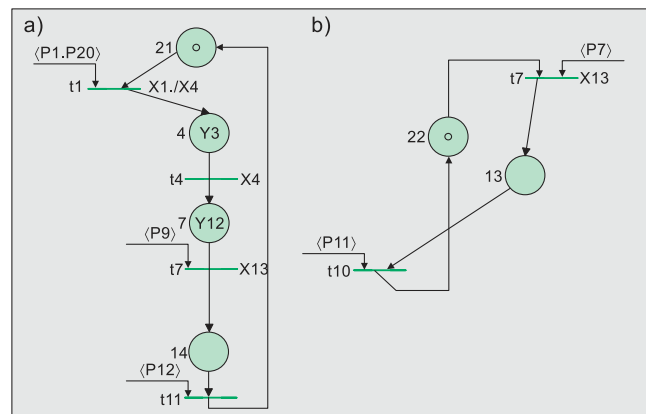


■ Rys. 2. Podsieci typu automatowego opisujące: a) zbiornik 1, b) zbiornik 2

Pierwsza podsieć (rys. 2a) opisuje działanie automatu realizującego zadanie napełnienia zbiornika 1, mieszania składników w zbiorniku 3, napełnienia pojemnika 1 na wózku oraz odjazd wózka spod maszyny. Druga podsieć (rys. 2b) opisuje działanie automatu pracującego analogicznie, tylko dla zbiornika 2. Trzecia podsieć (rys. 3a) opisuje załadowanie pojemników na wózek i jego podjazd pod maszynę. Aby automaty reprezentowane przez podsieci (rys. 2a i rys. 3a) mogły ze so-

■ Tabela. 2. Opis wejść i wyjść sterownika

Sygnały wejściowe	Sygnały wyjściowe
x1 – start	y1 – przygotuj składnik 1
x2 – składnik 1 gotowy	y2 – przygotuj składnik 2
x3 – składnik 2 gotowy	y3 – załaduj pojemniki
x4 – pojemniki na wózku	y4 – mieszaj
x5 – górny poziom w zbiorniku 1 (x5=1)	y5 – otwórz zawór spustowy składnika 1
x6 – dolny poziom w zbiorniku 1 (x6=0)	y6 – otwórz zawór spustowy składnika 2
x7 – górny poziom w zbiorniku 2 (x7=1)	y7 – napełnij pojemnik 1
x8 – dolny poziom w zbiorniku 2 (x8=0)	y8 – napełnij pojemnik 2
x9 – koniec mieszania (x9=0)	y9 – wózek jedzie w prawo
x10 – pojemnik 1 napełniony	y10 – napełnij zbiornik 1
x11 – pojemnik 2 napełniony	y11 – napełnij zbiornik 2
x12 – wózek w prawym skrajnym położeniu	y12 – wózek jedzie w lewo
x13 – wózek w lewym skrajnym położeniu	



■ Rys. 3. Podsieci typu automatowego: a) dla wózka, b) automat komunikacyjny

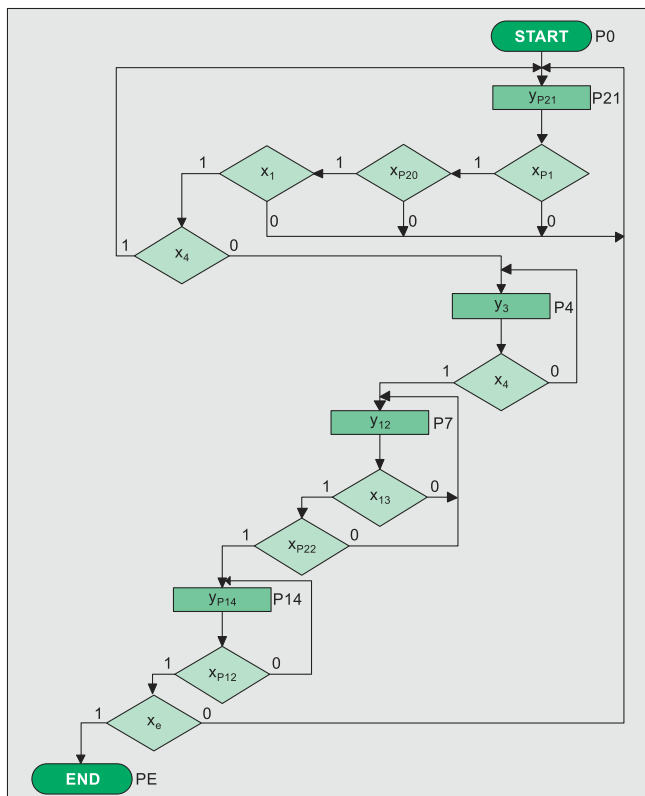
bą współdziałać, konieczne było utworzenie automatu komunikacyjnego (rys. 3b).

IMPLEMENTACJA

Otrzymane w wyniku dekompozycji sieci Petriego automaty składowe przekształcono na odpowiadające im sieci działań [3, 4]. Każdy z tych automatów jest implementowany jako mikroprogramowany układ sterujący [3, 5]: CMCU I – automat realizujący napełnianie zbiornika 1; CMCU II – automat realizujący napełnianie zbiornika 2; CMCU III – automat realizujący działanie wózka; CMCU IV – automat komunikacyjny. Dla potrzeb synchronizacji do układu dodano dodatkowe sygnały. Na rysunku 4 przedstawiono sieć działań dla podsieci typu automatowego z rysunku 3a.

Na rys. 5 przedstawiono strukturę układu sterującego, implementującego powiązane sieci działań, powstałe w wyniku dekompozycji sieci Petriego z rys. 1. Układ składa się z czterech mikroprogramowanych układów sterujących. Dla potrzeb synchronizacji poszczególnych sieci, do układu dodano dodatkowe sygnały wewnętrzne (x_{px} , y_{px}).

W tabeli 3. zostały przedstawione rozmiary pamięci mikro-



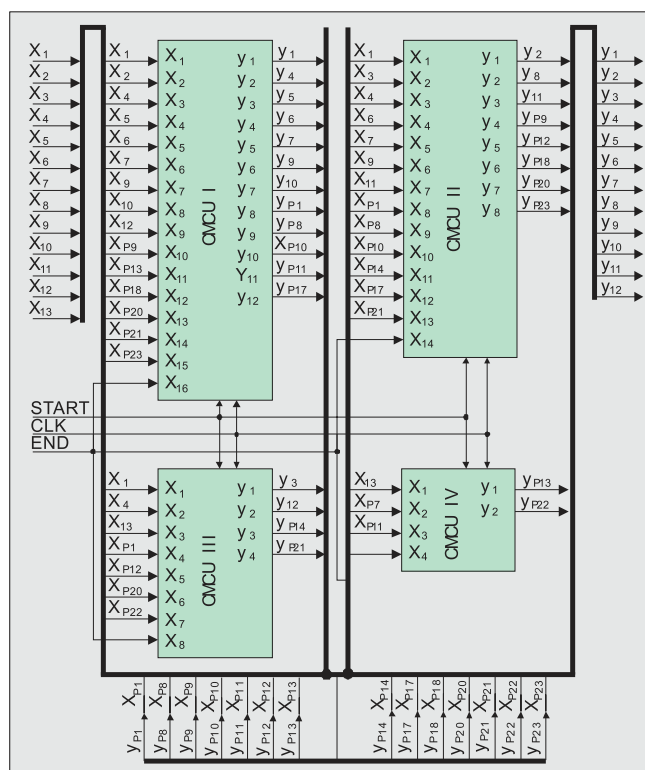
■ Rys. 4. Sieć działań

programowanych układów sterujących wygenerowanych przez program *fca2cmcu* [3] dla poszczególnych sieci działań uzyskanych w wyniku dekompozycji automatu współbieżnego opisanego siecią Petriego.

■ Tabela 3. Rozmiar pamięci [bity]

Układ	Sieć testowa	Rozmiar pamięci CMCU [bity]
CMCU I	fca_1	224
CMCU II	fca_2	160
CMCU III	fca_2	48
CMCU IV	fca_3	16

W artykule przedstawiono sposób implementacji układu cyfrowego z wykorzystaniem mikroprogramowanych układów sterujących opisanych powiązanymi sieciami działań. Zarówno diagramy SFC, jak i sieci Petriego umożliwiają w sposób jawny opisywanie procesów współbieżnych, jak również sposobów koordynacji tych procesów. Zastosowanie dekompozycji sieci Petriego na sekwencyjne automaty składowe ułatwia proces projektowania oraz syntezę układu cyfrowego, przedstawionego w postaci tej sieci. Zastosowanie mikroprogramowanych układów



■ Rys. 5. Struktura układu implementującego powiązane sieci działań

sterujących zapewnia zmniejszenie rozmiaru pamięci projektowanego układu cyfrowego oraz zmniejszenie zużycia zasobów sprzętowych.

LITERATURA

- [1] Adamski M.: *Projektowanie układów cyfrowych systematyczną metodą strukturalną*, Seria Monografie nr 49, Wydawnictwo Wyższej szkoły Inżynierskiej w Zielonej Górze, 1990
- [2] Adamski M., Chodań M.: *Modelowanie układów sterowania dyskretnego z wykorzystaniem sieci SFC*, Wydawnictwo Politechniki Zielonogórskiej, Zielona Góra 2000
- [3] Kołopieńczyk M.: *Zastosowanie konwertera adresów do zmniejszenia rozmiaru pamięci mikroprogramowanego układu sterującego ze współdzieleniem kodów*, rozprawa doktorska, Uniwersytet Zielonogórski, Zielona Góra 2007
- [4] Łuba T.: *Synteza układów logicznych*. Warszawa, WSIZ
- [5] Titarenko L., Kołopieńczyk M.: *Optymalizacja mikroprogramowanego układu sterującego przez zastosowanie współdzielenia kodów*, Pomiary Automatyka Kontrola 2006, nr 6, wyd.
- [6] Węgrzyn A.: *Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego*, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra 2003

Artykuł recenzowany