

## 1. Sterowanie procesami sekwencyjnymi

### 1.1. Układy logiczne i ich rodzaje

Układ logiczny (często nazywany również dyskretnym) jest to układ w którym zarówno sygnały wejściowe jak i wyjściowe są zmiennymi binarnymi (przyjmują tylko dwie wartości: 0 i 1). Układy logiczne dzielą się na kombinacyjne i sekwencyjne.

W układach kombinacyjnych aktualny stan ich wyjść zależy jedynie od aktualnego stanu wejść.

W układzie kombinacyjnym wyróżnia się:

- skończony zbiór sygnałów wejściowych  $X = \{ X_1, X_2, \dots, X_n \}$  ;
- skończony zbiór sygnałów wyjściowych  $Y = \{ Y_1, Y_2, \dots, Y_m \}$  .

Działanie układu kombinacyjnego opisywane jest przez funkcję  $\lambda$  wyjść układu

$$Y^t = \lambda(X^t) ,$$

gdzie indeks górny t odnosi się do pewnej, tej samej chwili czasu t.

W układach sekwencyjnych aktualny stan ich wyjść zależy nie tylko od aktualnego stanu wejść ale również od poprzednich stanów wejść. Zależność między sygnałami wyjściowymi i sygnałami wejściowymi nie jest więc jednoznaczna. Tym samym sygnałom wejściowym mogą odpowiadać różne sygnały wyjściowe, zależnie od stanu w jakim układ znajdował się poprzednio. Zależność aktualnego stanu układu sekwencyjnego od jego stanu w chwili poprzedniej jest realizowana za pomocą elementów pamięci. Stan elementów pamięci jest nazywany stanem wewnętrznym układu.

W układzie sekwencyjnym wyróżnia się więc:

- skończony zbiór sygnałów wejściowych  $X = \{ X_1, X_2, \dots, X_n \}$  ;
- skończony zbiór sygnałów wyjściowych  $Y = \{ Y_1, Y_2, \dots, Y_m \}$  ;
- skończony zbiór stanów wewnętrznych  $Q = \{ Q_1, Q_2, \dots, Q_k \}$  .

Działanie układu sekwencyjnego opisywane jest przez funkcję  $\delta$  przejść układu

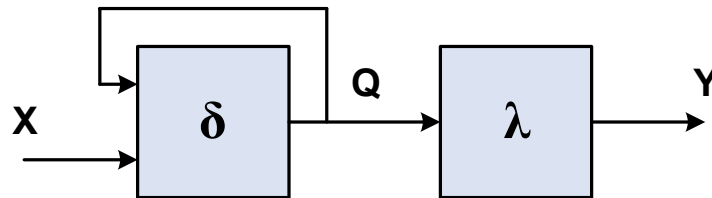
$$Q^t = \delta(Q^{t-1}, X^t) ,$$

gdzie indeks t-1 odnosi się do poprzedniej chwili czasu w której nastąpiła zmiana stanu wewnętrznego układu, oraz, w zależności czy jest to układ sekwencyjny Moore'a czy Mealy'ego, przez odpowiednią funkcję  $\lambda$  wyjść układu:

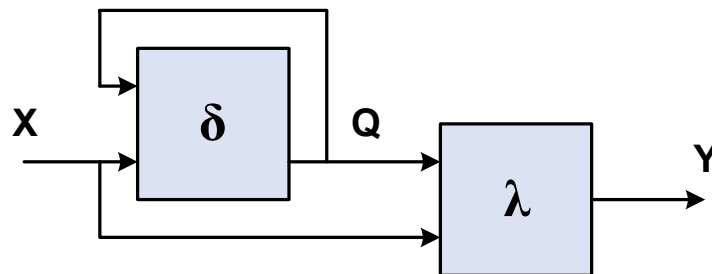
$$Y^t = \lambda(Q^t) \quad \text{dla układu Moore'ego ,}$$

lub  $Y^t = \lambda(Q^t, X^t)$  dla układu Mealy'ego .

Schematy blokowe układów sekwencyjnych Moore'a i Mealy'ego składające się z układu pamięci opisanego funkcją przejść  $\delta$  i układu kombinacyjnego opisanego funkcją wyjść  $\lambda$  przedstawiono na rysunkach 1.1 i 1.2.



Rys.1.1. Schemat blokowy układu sekwencyjnego Moore'a



Rys.1.2. Schemat blokowy układu sekwencyjnego Mealy'ego

Wszystkie układy sekwencyjne dzielą się także na asynchroniczne i synchroniczne.

Układy sekwencyjne asynchroniczne to takie układy sekwencyjne w których zmiany stanów wewnętrznych układu mogą występować w dowolnych chwilach czasu określonych przez zmiany sygnałów wejściowych.

Układy sekwencyjne synchroniczne to takie układy sekwencyjne w których zmiany stanów wewnętrznych mogą występować tylko w ściśle określonych, dyskretnych chwilach czasu wyznaczonych zmianami dodatkowego sygnału taktującego.

W przemyśle zdecydowana większość układów sterowania pojedynczych urządzeń, linii produkcyjnych i procesów technologicznych może być zakwalifikowana jako układy sterowania sekwencyjnego asynchronicznego [20]. Z tego względu dalsza część tego rozdziału poświęcona jest sposobom opisu, modelowania i programowania układów sterowania procesów sekwencyjnych.

## 1.2. Metody opisu i realizacji algorytmów sterowania sekwencyjnego

Do opisu dyskretnych procesów sekwencyjnych najczęściej wykorzystuje się metody graficzne przedstawienia algorytmów sterowania sekwencyjnego. Cechą charakterystyczną tych metod jest to, że wymagają one wcześniejszego zamodelowania działania urządzenia lub realizacji procesu technologicznego w procesie sterowania. Poprzez modelowanie rozumiane jest modelowanie działania układu, a nie budowa modelu komputerowego programowanego systemu sekwencyjnego.

Zgodnie z normą międzynarodową IEC 61131-3 [31] dotyczącą programowania sterowników jednym z języków programowania jest język graficzny SFC (*ang. Sequential Function Chart*). Jest to język używający kroków (etapów) i przejść (warunków przejść) w celu przedstawienia algorytmu programu sterowania procesem sekwencyjnym. Język graficzny SFC został oparty na języku GRAFCET (*fr. Graphe de Commande Etape – Transition*) [7,32] opisanym we francuskiej normie NF-C-3-190 z 1978 roku. W 1988 roku organizacja IEC (*ang. International Electrical Commission*) adaptowała metodę GRAFCET jako międzynarodowy standardowy język do programowania procesów sekwencyjnych; oznaczono go symbolem IEC 848 – „Preparation of function charts for control systems”. Norma ta została następnie powtórzona w normie IEC 61131-3 właśnie jako język SFC.

Język GRAFCET został opisany w podrozdziale 1.3, a w podrozdziale 1.4 zamieszczono i omówiono przykład wykorzystania tego języka do przedstawienia algorytmu sterowania prasą hydrauliczną będącą jednym z urządzeń linii produkcyjnej.

Metoda GRAFCET posiada wiele elementów wspólnych, stosowanych do opisu sterowania procesami sekwencyjnymi, z niemiecką normą DIN 40719. Główna różnica pomiędzy metodą GRAFCET a metodą DIN 40719 polega na tym, że w normie niemieckiej warunki przejścia nie są przyporządkowane do konkretnego połączenia pomiędzy etapami, lecz do każdego z etapów. Warunki przejścia, zwane w normie DIN 40719 warunkami początkowymi, stanowią zatem integralną część każdego z etapów.

Należy zauważyć, że w oprogramowaniach narzędziowych różnych producentów sterowników programowalnych języki graficzne do programowania procesów sekwencyjnych (bazujące na językach SFC i GRAFCET) mogą mieć różne nazwy, jak np.: GRAFTEC dla sterowników firmy SAIA, MEGRAF dla sterowników firmy Mitsubishi, czy GRAPH 5/II dla sterowników firmy Siemens.

Przykładami innych metod opisu dyskretnych procesów sekwencyjnych i sterowania nimi są też: sieć operacyjna ZPT (zautomatyzowanych procesów technologicznych) [17] i sieć logiczna GRAFPOL [18].

Do programowania sterowania procesami sekwencyjnymi niekoniecznie trzeba wykorzystywać graficzne języki programowania takie jak SFC i GRAFCET. Niekiedy wystarczające jest wykorzystanie możliwości jakie dają trzy podstawowe języki programowania sterowników, czyli: lista instrukcji (*ang. Instruction List - IL*), schemat drabinkowy (*ang. Ladder Diagram – LD*) i schemat bloków funkcji (*ang. Function Block Diagram – FBD*). A możliwości te to np. wykorzystanie instrukcji S i R (Set i Reset), wykorzystanie bloków funkcyjnych rejestrów przesuwanych (dla zmiennych binarnych) jako elementów przyporządkowujących i pamiętających. Niekiedy można też wykorzystać specjalne bloki funkcyjne sterowania sekwencyjnego jak np. SK w sterownikach firmy Moeller Electric czy SB w sterownikach firmy Siemens.

W podrozdziale 1.5 pokazano, w jaki sposób proces sterowania sekwencyjnego zamodelowany za pomocą grafu skierowanego może być oprogramowany wykorzystując do tego celu instrukcje SET i RESET.

### 1.3. Język GRAFCET i jego charakterystyka

Język GRAFCET jest jednym z języków pozwalających na opisanie systemu odnoszącego się do urządzenia lub procesu technologicznego pracującego sekwencyjnie tj. krok po kroku. Może być stosowany w przemysłowych systemach sterowania niezależnie od wykorzystywanej technologii: elektryczna, elektromechaniczna, pneumatyczna itp. , gdyż norma francuska opisuje tylko funkcjonalne przedstawienie systemu sterowania nie opisując programowania.

Celem języka GRAFCET jest standaryzacja funkcjonalna i graficzna przedstawienia systemu sterowania. W swoich zastosowaniach diagram GRAFCET wykorzystuje ograniczoną liczbę prostych symboli i kieruje się zbiorem określonych reguł. Na rysunku 1.3 przedstawiono elementy języka GRAFCET z których tworzony jest diagram GRAFCET.

Diagram GRAFCET składa się z:

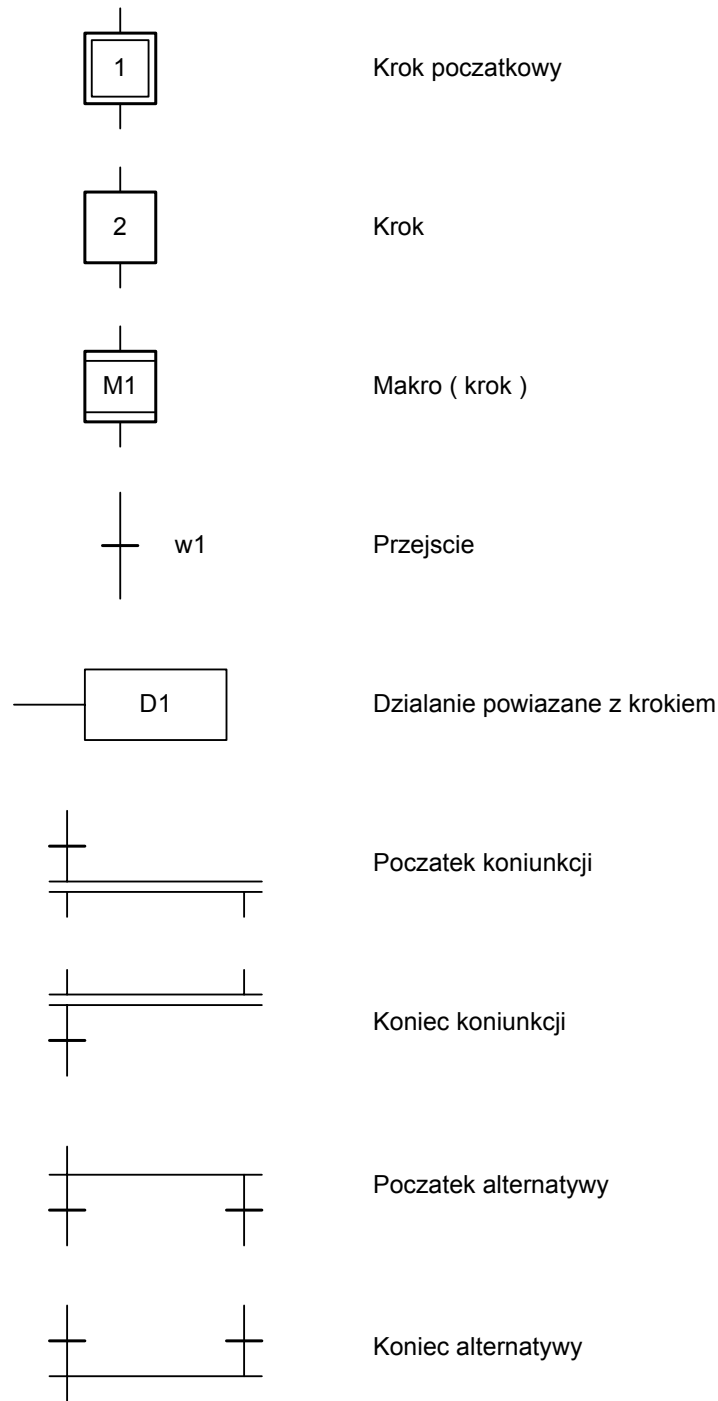
- kroków - które zawierają działania;
- przejść - które zawierają warunki;
- połączeń pomiędzy krokami i przejściami .

Przejście jest realizowane tylko wówczas, gdy:

- krok bezpośrednio przed nim jest aktywny;
- warunek przejścia jest spełniony (prawdziwy).

Realizacja przejścia powoduje dezaktywację kroku (kroków) który wcześniej był aktywny i aktywację kroku (kroków) do którego prowadzi przejście którego warunek przejścia został

spełniony. Następnstwo kroków i przejść które muszą występować naprzemiennie tworzy sekwencję diagramu.

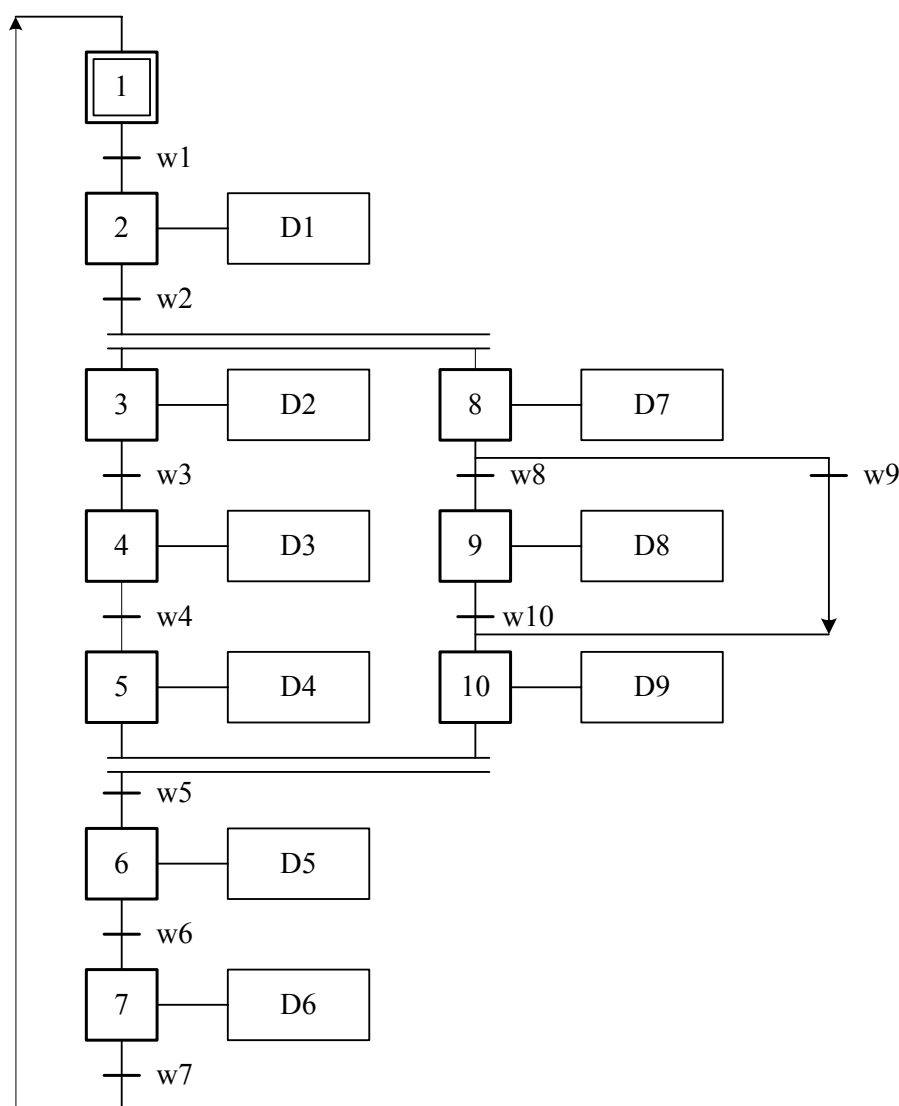


Rys.1.3. Symbole języka GRAFCET

Jeśli wykonanie przejścia prowadzi do uaktywnienia kilku sekwencji mówi się o równoczesnym lub równoległym rozgałęzieniu. Po równoczesnym uruchomieniu sekwencje te są wykonywane równolegle, jedna niezależnie od drugiej.

Jeśli wykonanie kroku prowadzi do rozgałęzienia na kilka przejść mówi się o alternatywnym rozgałęzieniu. Tylko jedna gałąź jest wykonywana, ta mianowicie, dla której warunek początkowego przejścia został spełniony jako pierwszy.

Powyższe, skrótowo przedstawione reguły tworzenia diagramu GRAFCET i reguły ewolucyjne określające przechodzenie w diagramie pomiędzy krokami związane z ich aktywacją i dezaktywacją można zilustrować na przykładzie diagramu GRAFCET przedstawionego na rysunku 1.4.



Rys.1.4. Przykładowy diagram języka GRAFCET (oznaczenia: 1÷10 – kroki, w1÷w10 – przejścia, D1÷D9 – działania powiązane z krokami)

W tym diagramie jest w sumie 10 kroków (w tym jeden krok początkowy oznaczony numerem 1) oraz 10 przejść zawierających warunki przejść oznaczone na rysunku 1.4 od w1

do w10. Z założenia połączenia pomiędzy krokami przebiegają z góry do dołu diagramu; w przypadkach nieoczywistych lub innych połączeniach, kierunek połączenia może być oznaczony strzałką. Na rysunku 1.4 dwa takie połączenia oznaczono strzałkami.

Z krokami, za wyjątkiem kroku początkowego który jest zazwyczaj krokiem spoczynkowym, są powiązane określone działania na rysunku oznaczone od D1 do D9. Działania te są wykonywane wówczas, gdy krok z którym są powiązane jest aktywny, i przestają być wykonywane wówczas, gdy krok przestaje być aktywny.

Możliwość aktywacji kroków zależy od przejść je poprzedzających i aktualnej aktywności kroków w diagramie. Przykładowo, jeśli aktywny jest krok 1 i spełniony jest warunek przejścia w1 to nastąpi przejście do kroku 2, czyli dezaktywacja kroku 1 i aktywacja kroku 2. Po spełnieniu warunku przejścia realizacja przejścia jest obligatoryjna i natychmiastowa.

W diagramie warunki przejść mogą się powtarzać, ale nie mogą być identyczne dla dwóch kolejnych przejść, gdyż prowadziłoby to do pominięcia zawartego pomiędzy nimi kroku. Przykładowo, gdyby warunki w3 i w4 były identyczne, to krok 4 byłby zawsze pomijany i nigdy nie byłyby realizowane działania D3 powiązane z tym krokiem; taka sytuacja byłaby błędna i nie może mieć miejsca.

W diagramie oprócz połączeń prostych (typu: krok – przejście – krok –przejście itd.) są również połączenia wielokrotne.

Gdy aktywny jest krok 2 i spełniony jest warunek w2 znajdującego się za nim przejścia, to nastąpi dezaktywacja kroku 2 i równoczesna aktywacja kroków 3 i 8 (synchronizacja sterowania) co na diagramie pokazane jest równoległymi liniami po warunku w2. Wyjście z tej sekwencji równoległej nastąpi wówczas, gdy będą aktywne kroki 5 i 10 oraz spełniony będzie warunek przejścia w5. Nastąpi wówczas dezaktywacja kroków 5 i 10 oraz aktywacja kroku 6.

Gdy aktywny jest krok 8, to kolejnym krokiem aktywnym może stać się albo krok 9 albo krok 10. Gdy spełniony będzie warunek przejścia w8 to nastąpi dezaktywacja kroku 8 i aktywacja kroku 9. Natomiast, gdy spełniony będzie warunek przejścia w9, to nastąpi dezaktywacja kroku 8 i aktywacja kroku 10 (krok 9 i działania z nim powiązane są omijane). Ponieważ jest to wybór jednego kroku z dwóch możliwych to, aby sterowanie było określone, warunek przejścia w9 powinien być negacją warunku w8.

Jak widać z powyższego, w diagramie GRAFCET w danej chwili może być aktywny więcej niż jeden krok. W przykładzie z rysunku 1.4 aktywne mogą być równocześnie dwa kroki (jeden spośród kroków 3, 4, 5 i jeden spośród kroków 8, 9, 10).

W diagramie GRAFCET może być także więcej niż jeden krok początkowy, ale taki

przypadek nie został na rysunku 1.4 pokazany.

Kilka kroków i przejść może być połączonych tworząc tzw. makro krok. Takie połączone kroki i przejścia powinny być funkcjonalnie powiązane, tworząc w sumie złożone zadanie (funkcję) sterowania. Makro krok zawsze rozpoczyna się i kończy krokiem, i dzięki temu może być umieszczony w diagramie GRAFCET tak samo jak zwykły pojedynczy krok z poprzedzającym go przejściem i z następującym po nim przejściem. Dzięki wykorzystaniu makro kroków można osiągnąć większą przejrzystość diagramu. Diagram taki odzwierciedla ogólną strukturę sterowania, podczas gdy szczegóły ukryte są w makro krokach (możliwe jest zagnieżdżanie – tj. umieszczanie makro kroków w makro krokach).

Język GRAFCET steruje tylko sekwencją sterowań. W celu opisanie działań zawartych w krokach i warunków przejść należy użyć innego języka programowania np.: języka drabinkowego (LD) lub listy instrukcji (IL).

Organizacja sekcji programu w języku GRAFCET jest ściśle określona. Program (lub fragment programu) zapisany w języku GRAFCET można podzielić na trzy części:

- przetwarzanie wstępne;
- przetwarzanie sekwencyjne;
- przetwarzanie końcowe.

Przetwarzanie wstępne umożliwia przetwarzanie wszystkich zdarzeń związanych z zarządzaniem systemem w razie zaniku zasilania i podczas kolejnych inicjalizacji, oraz z kasowaniem lub wstępnym zadawaniem parametrów diagramu GRAFCET.

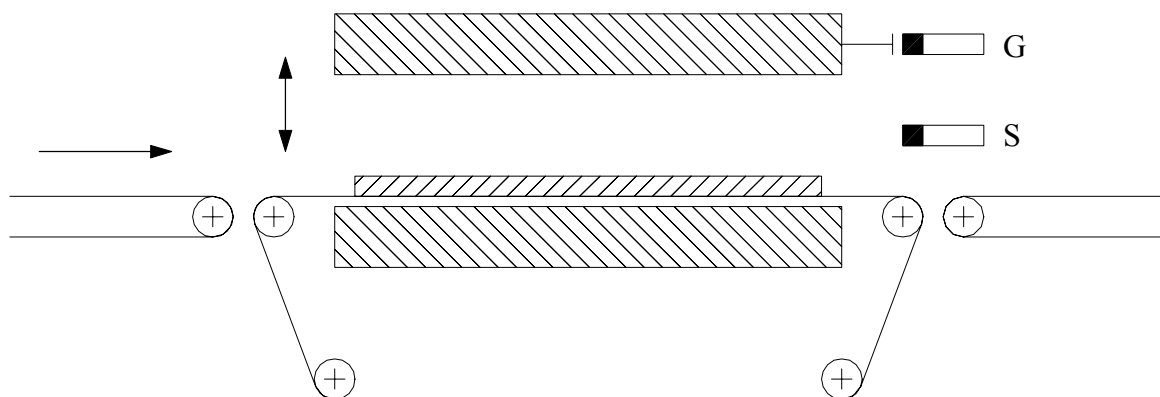
Przetwarzanie sekwencyjne jest główną częścią sekcji GRAFCET i służy do przetwarzania struktury sekwencyjnej programu sterowania urządzeniem lub procesem technologicznym. W czasie przetwarzania sekwencyjnego analizowane są warunki przejść i warunki powodujące dezaktywację lub uaktywnienie kroków, w dalszej kolejności aktualizowany jest stan diagramu GRAFCET zgodnie z aktualnie spełnionymi warunkami, a następnie wykonywane są akcje przyporządkowane aktywnym krokom.

Przetwarzanie końcowe obejmuje wykonanie akcji związanych bezpośrednio z aktualizacją stanów wyjść sterownika przypisanych do akcji dla poszczególnych kroków, aktualizację stanów wyjść nie związanych z przetwarzaniem sekwencyjnym programu oraz monitorowanie wykonywania diagramu GRAFCET poprzez kontrolę czasu aktywności wybranych kroków. W przetwarzaniu końcowym podczas aktualizowania stanów wyjść uwzględniane są również wszystkie zaprogramowane zabezpieczenia.



#### 1.4. Przykład wykorzystania języka GRAFCET

Jako przykład rozpatrywane jest sterowanie fragmentem linii technologicznej z prasą hydrauliczną [24] przedstawioną na rysunku 1.5. Należy skonstruować diagram GRAFCET opisujący działanie prasy hydraulicznej w trybie pracy automatycznej linii technologicznej.

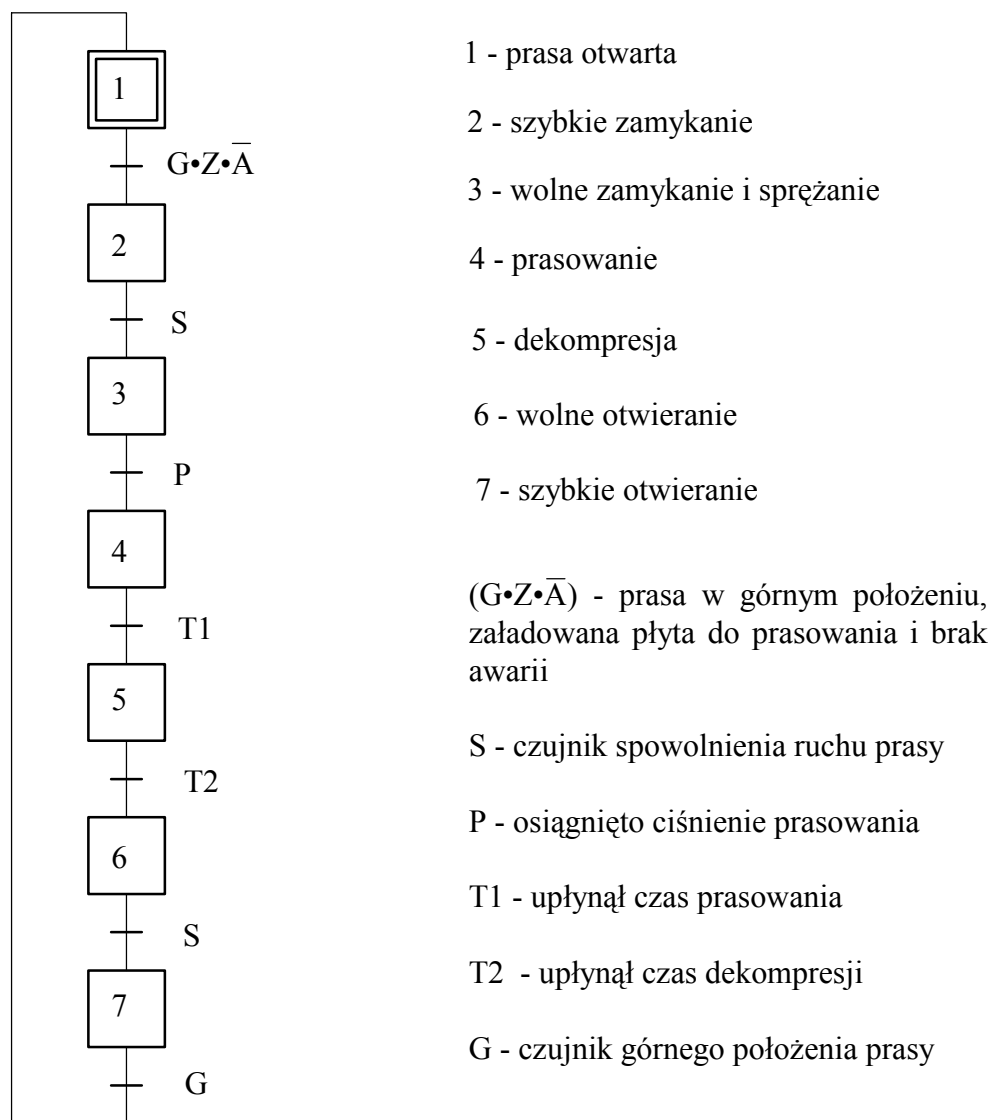


Rys.1.5. Fragment linii technologicznej z prasą hydrauliczną.

Prasa jest otwarta gdy jest sygnał z czujnika indukcyjnego G , tj. gdy ruchoma płyta prasy jest w górnym położeniu. Sygnał z czujnika indukcyjnego S określa pośrednie położenie górnej płyty prasy. Wartość ciśnienia jest odczytywana z analogowego czujnika ciśnienia z przetwornikiem pomiarowym. Pozostałe (nieistotne z punktu widzenia tworzonego diagramu) czujniki pomiarowe linii zostały pominięte.

Skonstruowany diagram GRAFCET przedstawiono na rysunku 1.6. Warunkiem wykonania cyklu prasowania w trybie pracy automatycznej jest górne położenie prasy, obecność płyty do prasowania i brak awarii linii. Cykl prasowania przedstawia się następująco: prasa otwarta (krok początkowy) - szybkie zamykanie prasy (do sygnału z czujnika S) - wolne zamykanie i sprężanie (do osiągnięcia ciśnienia prasowania P) - prasowanie (T1 - zadany czas prasowania) - dekompresja (T2 - zadany czas dekompresji) - wolne podnoszenie prasy (do sygnału z czujnika S) - szybkie podnoszenie (do sygnału z czujnika G) - prasa otwarta (oczekiwanie na wyładunek sprasowanej i załadunek nowej płyty). Jest to więc prosty diagram GRAFCET dla sterowanego urządzenia uwzględniający tylko normalne (poprawne) stany pracy prasy.

Rozpatrywany jest tylko proces prasowania. Nie rozpatruje się procesu załadunku elementów (płyt) do prasowania i procesu rozładunku elementów sprasowanych.



Rys.1.6. Diagram GRAFCET cyklu prasowania w trybie pracy automatycznej

Do przedstawienia procesu rozładunku / załadunku musiałby być stworzony oddzielny diagram GRAFCET. Załadunek i rozładunek mogą odbywać się równocześnie lub też kolejno: rozładunek – załadunek. Z punktu widzenia wydajności linii najbardziej pożądanym jest równoczesny rozładunek i załadunek elementów. W przypadku, gdy odbyło się prasowanie i brak jest lub nie jest gotowy do załadunku nowy element do prasowania odbędzie się tylko rozładunek bez załadunku. Następnie, gdy jest już gotowy element do prasowania, a przenośnik w prasie jest pusty, odbywa się tylko załadunek. Przy opracowywaniu diagramu należałoby pamiętać o konieczności dokonywania rozładunku i załadunku przy takich samych prędkościach (ustalonych) przenośników przed prasą, w prasie i za prasą, oraz o załączaniu przenośnika przed prasą z opóźnieniem w stosunku do przenośnika w prasie. Opóźnienie to wynika z konieczności centrycznego załadunku elementu

do prasowania pod płytę prasy i jest określone na podstawie różnicy w długościach przenośników w prasie i przed prasą.

Sygnalami dzięki którym procesy prasowania i załadunku / rozładunku są realizowane naprzemiennie są: sygnał Z – obecności załadowanego elementu do prasowania w diagramie GRAFCET opisującym prasowanie i analogiczny sygnał (element sprasowany) który byłby w warunku przejścia po kroku początkowym w diagramie GRAFCET opisującym załadunek / rozładunek. Dzięki tym sygnałom możliwe byłoby połączenie tych dwóch diagramów w jeden diagram GRAFCET dla procesu z dwoma krokami początkowymi: dla prasy i dla przenośników.

W diagramie przedstawionym na rysunku 1.6 nie pokazano działań (operacji) powiązanych z krokami. Należy jednakże mieć zawsze na uwadze, że z każdym krokiem – za wyjątkiem kroku początkowego który odpowiada stanowi spoczynkowemu – są powiązane określone działania. Dla przedstawionego przykładu będą to odpowiednie sekwencje załączeń elektrozaworów w układzie hydraulicznym, dzięki którym realizowany jest cykl pracy prasy. Przykładowe sekwencje załączeń elektrozaworów dla prasy hydraulicznej z czterema elektrozaworami [23] przedstawiono w tabeli 1.1.

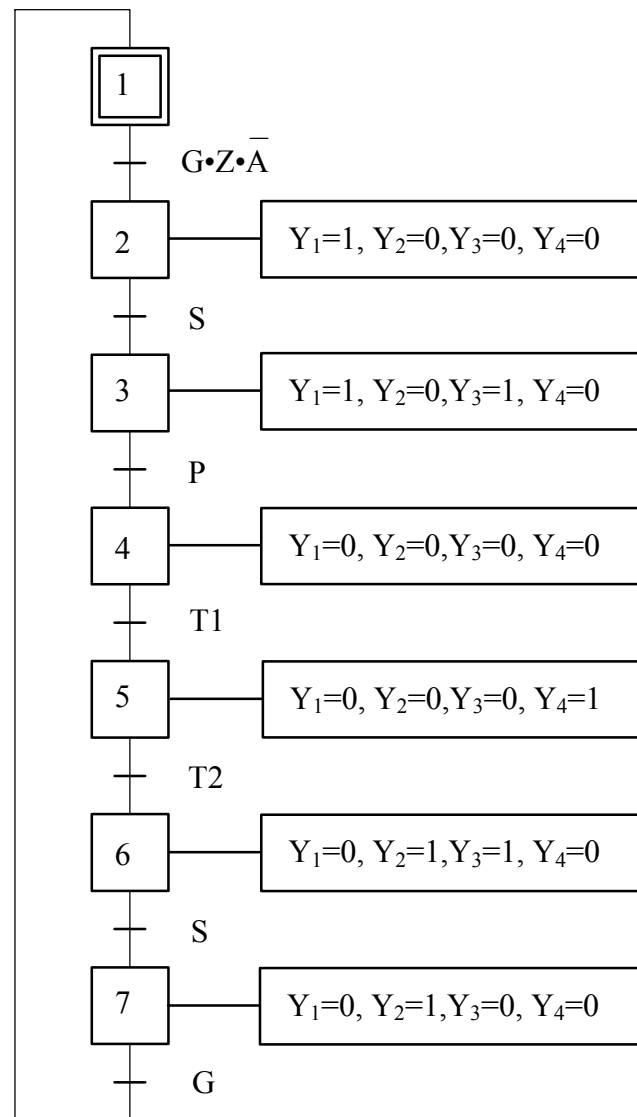
Tabela 1.1. Sterowanie elektrozaworami ( $Y_1, Y_2, Y_3, Y_4$ )

<b>Etap</b>	<b>Sterowanie</b>	<b><math>Y_1</math></b>	<b><math>Y_2</math></b>	<b><math>Y_3</math></b>	<b><math>Y_4</math></b>
Prasa otwarta		0	0	0	0
Szybkie zamykanie		1	0	0	0
Wolne zamykanie i sprężanie		1	0	1	0
Prasowanie		0	0	0	0
Dekompresja		0	0	0	1
Wolne otwieranie		0	1	1	0
Szybkie otwieranie		0	1	0	0

Informacja przedstawiona w formie tabeli może być również przedstawiona w diagramie GRAFCET. Na rysunku 1.7 przedstawiono diagram GRAFCET z rysunku 1.6, ale uzupełniony o działania powiązane z krokami diagramu.

Często spotykanym wymaganiem dla czynności prasowania w procesach technologicznych jest utrzymywanie z dużą dokładnością stałej wartości ciśnienia prasowania. Ponieważ podczas prasowania ciśnienie może wolno się obniżać, co jest związane z nieidealnością układu hydraulicznego, w przypadku długich czasów prasowania niezbędne jest wprowadzenie funkcji dobijania ciśnienia i funkcji dalszego prasowania. Dobijanie ciśnienia oznacza, że jeśli ciśnienie zmaleje poniżej dopuszczalnej, granicznej wartości (określonej zazwyczaj w % wartości zadanej), to musi być zrealizowana funkcja sterowania powodująca ponowny wzrost ciśnienia do wartości zadanej. Następnie prasowanie

jest kontynuowane.

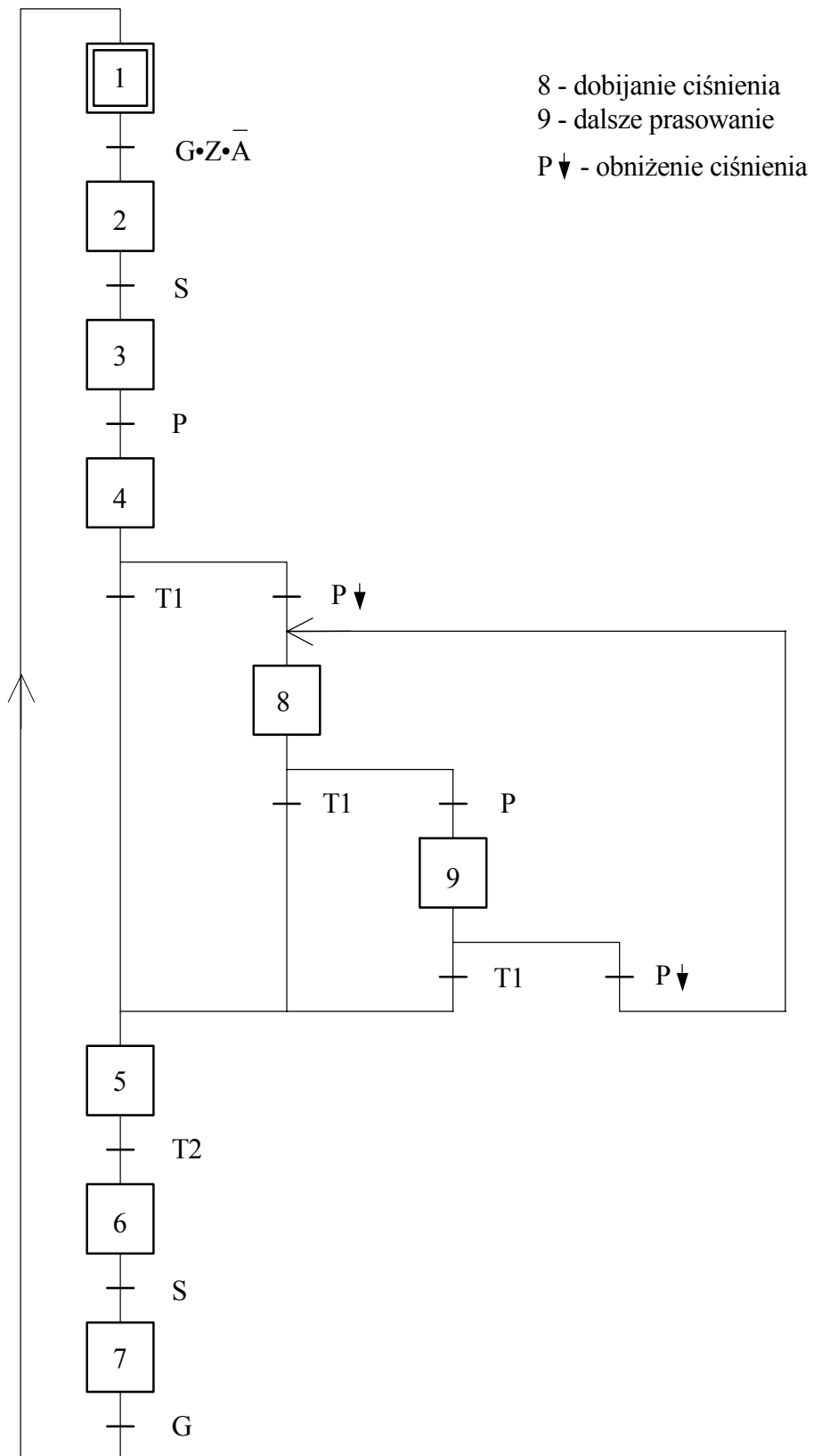


Rys.1.7. Diagram GRAFCET cyklu prasowania w trybie pracy automatycznej z uwzględnieniem działań powiązanych z krokami

Diagram GRAFCET cyklu prasowania w trybie pracy automatycznej uzupełniony o funkcje dobijania ciśnienia i dalszego prasowania przedstawiono na rys.1.8. Z punktu widzenia przedstawienia algorytmu sterowania najważniejsze są kroki, połączenia pomiędzy krokami i przejścia (warunki przejść) na tych połączeniach, a mniej istotne są działania. Dlatego więc na tym i kolejnym rysunku, aby nie pogarszać czytelności rysunków, pominięto graficzne przedstawienie działań (operacji) powiązanych z krokami diagramu.

Warunkiem przejścia do kroku dobijania ciśnienia jest obniżenie ciśnienia poniżej wartości dopuszczalnej ( $P \downarrow$ ). Czas prasowania  $T1$  jest liczony łącznie dla kroków: prasowania, dobijania ciśnienia i dalszego prasowania. Sekwencja dobijania ciśnienia i dalszego

prasowania może być wykonywana wielokrotnie.



Rys.1.8. Diagram GRAFCET cyklu prasowania w trybie pracy automatycznej uzupełniony o funkcję dobijania ciśnienia i funkcję dalszego prasowania

Podobnie jak w przypadku funkcji dodatkowych (rozpatrzona powyżej funkcja dobijania ciśnienia) diagram GRAFCET może być uzupełniony również o kroki odpowiadające awaryjnym stanom pracy.

Rozpatrzone zostaną dwa przypadki:

- uszkodzenie czujnika spowolnienia S ;
- nie nastąpił wzrost ciśnienia prasowania do wartości zadanej w założonym czasie.

Uszkodzenie czujnika spowolnienia ruchu prasy jest wykrywane wówczas, gdy:

- podczas opuszczania prasy brak będzie sygnału z czujnika S przed osiągnięciem zadanej wartości ciśnienia prasowania P ;
- podczas podnoszenia prasy brak będzie sygnału z czujnika S przed osiągnięciem górnego położenia prasy (sygnał z czujnika G).

Przyczyną, że w założonym czasie nie nastąpił wzrost ciśnienia do wartości zadanej (T3 - maksymalny czas narastania ciśnienia) może być niesprawność układu hydraulicznego podczas cyklu prasowania, spowodowana np. wyciekami oleju z układu hydraulicznego, uszkodzeniem zaworu realizującego sprężanie itp. Charakterystyczne dla tego typu uszkodzeń jest to, że program sterowania nie może stwierdzić ich zaistnienia bezpośrednio na podstawie sygnałów wejściowych, tylko w sposób pośredni.

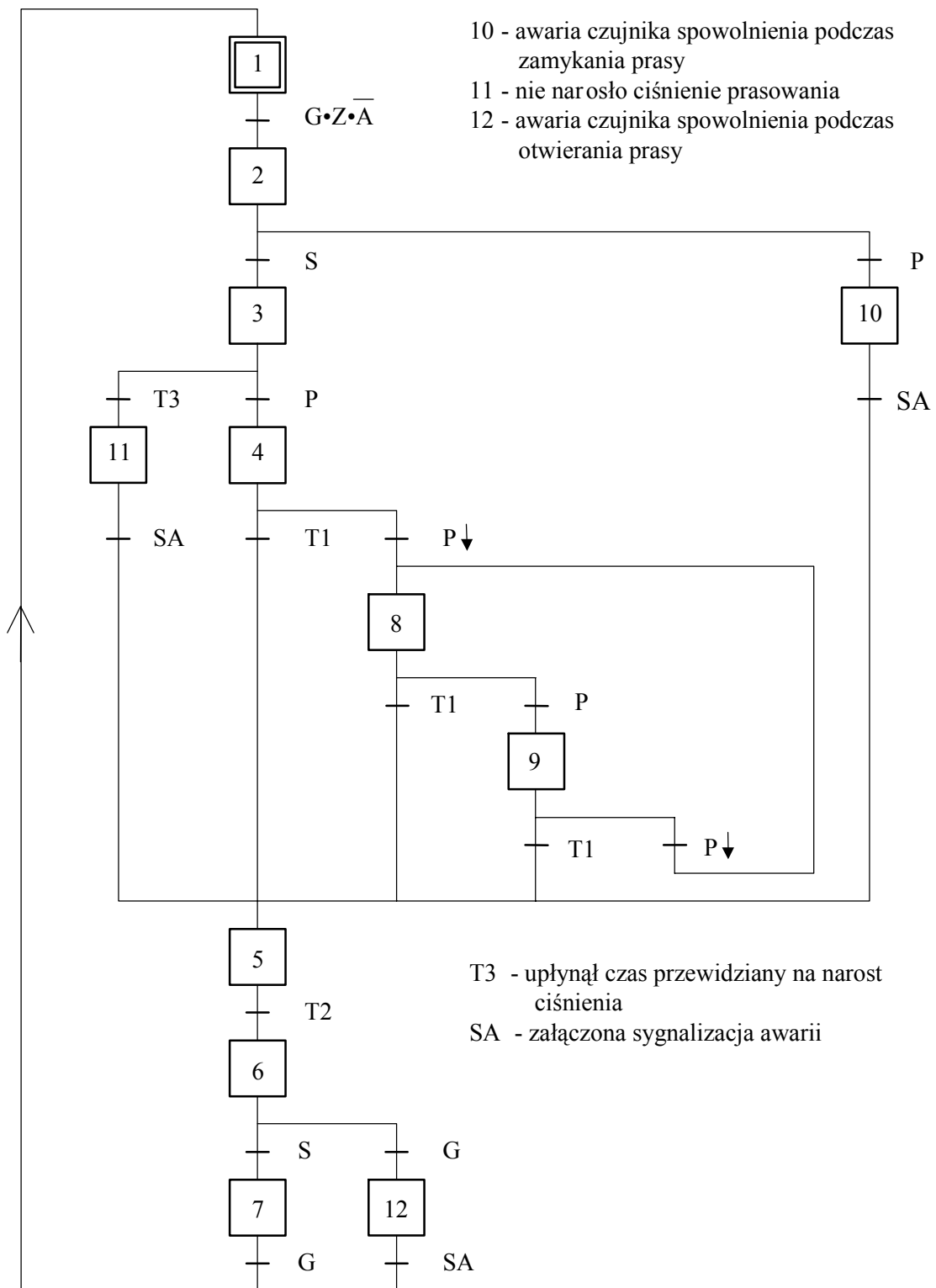
Diagram GRAFCET, analogiczny do przedstawionego na rysunku 1.8, uzupełniony trzema krokami odpowiadającymi rozpatrywanym awaryjnym stanom pracy przedstawiono na rysunku 1.9. Ponieważ krokom dla tych trzech awaryjnych stanów pracy nie są przyporządkowane działania (za wyjątkiem sygnalizacji awarii) możliwe jest umieszczenie w diagramie po tych krokach a przed dekompresją warunków przejść które będą spełnione od razu po zasygnalizowaniu awarii (SA).

Rozpatrzone trzy stany awaryjne nie obejmują wszystkich możliwych stanów awaryjnych które mogą wystąpić w trybie pracy automatycznej prasy w linii; zostały one wybrane i przedstawione tylko dla zilustrowania przyjętego sposobu tworzenia algorytmu.

Przedstawiony sposób opisu algorytmów w języku GRAFCET może być dużym ułatwieniem podczas weryfikacji opracowywanych algorytmów. Dzięki takiemu opisowi algorytmu sterowania istnieje również możliwość szybkiego sprawdzenia jakie uszkodzenia (czy wszystkie istotne i najczęściej występujące) zostały uwzględnione na etapach projektowania systemu i pisania programu sterowania. Analiza samego programu sterowania nie dałaby odpowiedzi tak szybko.

Do niewątpliwych zalet języka GRAFCET należy więc duża przejrzystość i uporządkowanie algorytmów sterowania oraz możliwość przedstawienia za pomocą diagramu

GRAF CET całego, złożonego zadania automatyzacji (jak i wydzielonych, poszczególnych zadań sterowania), co może być ułatwieniem przy zespołowej pracy nad oprogramowaniem.



Rys.1.9. Diagram GRAFCET z rysunku 1.8 uzupełniony trzema krokami odpowiadającymi rozpatrywanym awaryjnym stanom pracy cyklu prasowania w trybie pracy automatycznej

### 1.5. Modelowanie procesów sekwencyjnych za pomocą grafów skierowanych

W tym podrozdziale opisana jest metoda wykorzystująca do opisu procesów sekwencyjnych grafy skierowane, oraz sposób jej realizacji w programie sterownika programowalnego [23]. Do najważniejszych zalet tej metody (zbliżonej do metody GRAFCET) należą:

- prostota metody wynikająca z intuicyjnego sposobu tworzenia algorytmu sterowania procesem sekwencyjnym;
- możliwość uwzględnienia w opisie algorytmu sterowania sekwencyjnego stanów normalnej pracy jak i stanów awaryjnych;
- możliwość programowej realizacji algorytmu sterowania sekwencyjnego za pomocą prostych instrukcji SET i RESET.

Stanowi normalnej pracy odpowiada etap podstawowy, a stanowi awaryjnemu etap zabezpieczeń. Wszystkie etapy są wierzchołkami grafu skierowanego. Etapy podstawowe są to etapy które przy poprawnym działaniu programu wystarczają do prawidłowego (poprawnego z punktu widzenia założeń czyli zgodnego z wymaganiami użytkownika) sterowania urządzeniem lub procesem technologicznym. Etapy zabezpieczeń (inaczej nazywane etapami awaryjnymi) są to etapy, które realizują funkcje bezpieczeństwa w przypadku wystąpienia defektu.

Algorytmy sterowania w tej metodzie są przedstawiane za pomocą grafów skierowanych. Każdy graf skierowany algorytmu sterowania składa się (jest złożeniem) dwóch grafów skierowanych - jednego dla etapów podstawowych i drugiego obejmującego wszystkie etapy zabezpieczeń. Pełny opis algorytmu sterowania, oprócz zbudowania grafów skierowanych, wymaga jeszcze utworzenia tablic etapów i warunków przejść pomiędzy etapami.

Tak więc metoda ta jest opisywana przez następujące zbiory:

- $E = \{ E1, E2, \dots, En \}$  skończony zbiór etapów podstawowych, gdzie  $n$  - liczba etapów podstawowych;
- $A = \{ A1, A2, \dots, Ak \}$  skończony zbiór etapów zabezpieczeń, gdzie  $k$  - liczba etapów zabezpieczeń (awaryjnych);
- $X = \{ X1, X2, \dots, Xp \}$  skończony zbiór połączeń skierowanych (gałęzi grafu) wskazujących kierunki przejścia z jednego etapu do drugiego, gdzie  $p$  - liczba połączeń skierowanych;
- $W = \{ W1, W2, \dots, Wm \}$  skończony zbiór warunków przejść pomiędzy etapami,



gdzie  $m = p$  (warunki przejścia są przyporządkowane do połączeń skierowanych) lub  $m = n + k$  (gdy warunki przejścia są tzw. warunkami brzegowymi i nie są powiązane z etapami).

Dowolna liczba etapów może zostać połączona tworząc w ten sposób fragment programu modelujący złożoną funkcję realizowaną przez program, lub też może zostać połączona tworząc podprogram przywoływany z programu głównego.

Tworzony jest wówczas zbiór dodatkowy:

- $M = \{ M_1, M_2, \dots, M_i \}$  skończony zbiór funkcji złożonych (podprogramów),  
gdzie  $i$  - liczba funkcji złożonych,  $i < n$ .

Do funkcji złożonej może prowadzić tylko jedno połączenie skierowane (wejściowe) ze związanym z nim warunkiem przejścia, oraz wychodzić tylko jedno połączenie skierowane (wyjściowe) ze związanym z nim warunkiem przejścia. Dzięki temu funkcja złożona może być uwzględniana w grafie skierowanym modelującym proces sekwencyjny tak samo jak pojedynczy etap.

Funkcje złożone (podprogramy) nie posiadają warunków przejść nawet wówczas, gdy są to tzw. warunki brzegowe powiązane z etapami. Warunki te posiadają etapy z których składa się funkcja złożona, w szczególności istotne są warunki przejścia dla pierwszego i ostatniego etapu składającego się na funkcję złożoną.

Do grafu pierwszego przynależą tylko etapy podstawowe i odpowiednie warunki przejść pomiędzy nimi. Graf ten zawiera te etapy, które reprezentują wypełnienie wszystkich zadań postawionych systemowi gdy nie występują defekty. Tak więc pierwszy graf przedstawia poprawne działanie układu.

Graf drugi zawiera zarówno etapy podstawowe jak i etapy awaryjne. Graf ten reprezentuje reakcje programowanego systemu sterowania na wystąpienie defektu w systemie.

Etapy podstawowe spełniają w nim dwie funkcje:

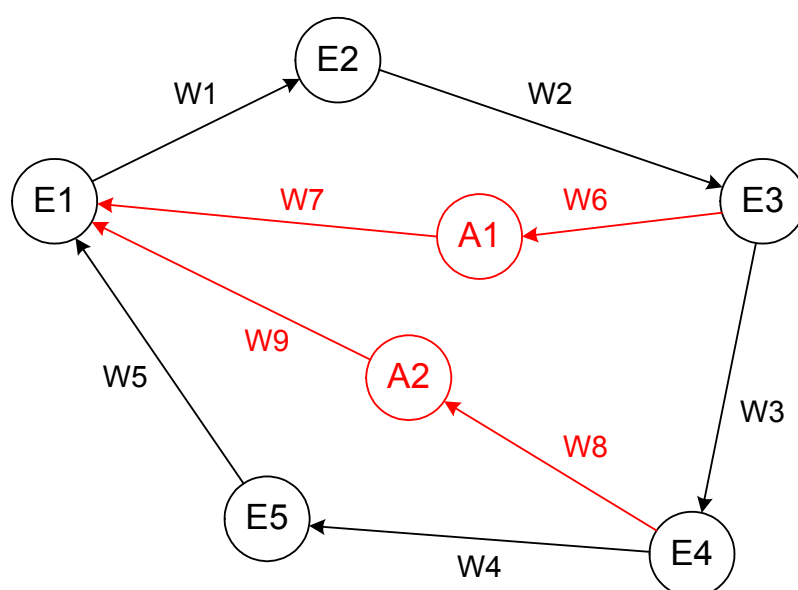
- przedstawiają etapy ochraniające; z etapu tego, po wykryciu defektu, układ przechodzi do etapu awaryjnego;
- przedstawiają etapy do których układ przechodzi z etapów awaryjnych.

Złożenie obu tak zbudowanych grafów skierowanych daje w wyniku graf opisujący działanie programowanego sekwencyjnego układu sterowania dla wszystkich przewidzianych przez projektanta zdarzeń. Takie przedstawienie programu czyni jego strukturę bardziej przejrzystą i upraszcza jego analizę.

Podstawowe reguły tworzenia grafu skierowanego modelującego proces sekwencyjny są następujące:

- jeden z etapów powinien być wyróżniony jako etap początkowy;
- etapy podstawowe tworzą obwód (drogę zamkniętą z etapem początkowym);
- z etapu podstawowego można przejść do kolejnego etapu podstawowego lub do etapu zabezpieczeń;
- z etapu zabezpieczeń można przejść tylko do etapu podstawowego;
- w grafie co najmniej jeden etap jest etapem aktywnym;
- przejście jest realizowane wówczas, gdy aktywny jest etap je poprzedzający i spełniony jest warunek przejścia;
- realizacja przejścia jest obligatoryjna i natychmiastowa;
- realizacja przejścia jest związana z dezaktywacją etapu poprzedzającego i aktywacją etapu następnego;
- z etapem powiązane są działania realizowane wówczas, gdy etap jest aktywny;
- synchronizacja sterowania (koniunkcja) jak i wybór wariantu sterowania (alternatywa) są realizowane poprzez odpowiednie zdefiniowanie warunków przejść.

Dla przykładu, na rysunku 1.10 pokazano prosty diagram skierowany modelujący proces sekwencyjny. Diagram posiada etap początkowy oznaczony jako E1, cztery inne etapy podstawowe (E2, E3, E4, E5), dwa etapy zabezpieczeń (A1 i A2) oraz dziewięć połączeń skierowanych zawierających odpowiednie warunki przejść (W1÷W9).

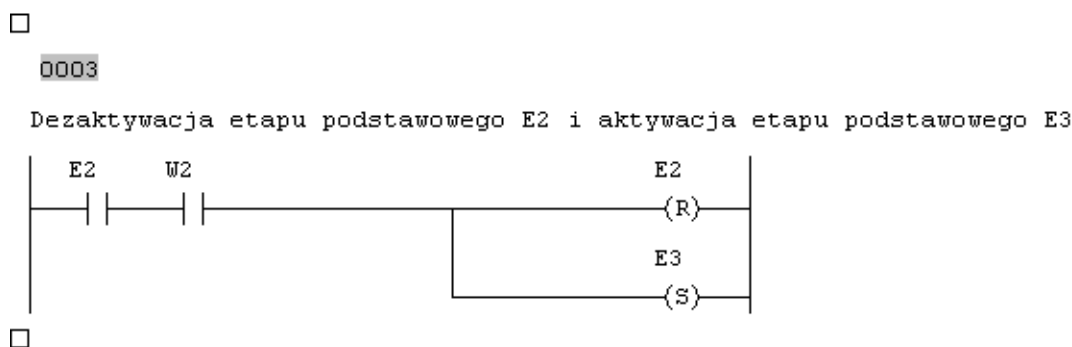


Rys.1.10. Przykładowy graf algorytmu sterowania z etapami podstawowymi i z etapami zabezpieczeń

Przedstawiony na rysunku graf jest złożeniem dwóch grafów – pierwszego składającego się z etapów podstawowych (E1÷E5) i drugiego składającego się z dwóch etapów zabezpieczeń (A1 i A2) oraz trzech etapów podstawowych (E1, E3 i E4).

Ponieważ reguła budowy grafu skierowanego mówi o tym, że co najmniej jeden etap powinien być etapem aktywnym, to w chwili pierwszego uruchomienia programu jedna zmienna odpowiadająca etapowi (zazwyczaj etapowi początkowemu) powinna przyjąć wartość logiczną TRUE (poprzez zadeklarowanie wartości początkowej lub poprzez iloczyn negacji zmiennych dla wszystkich etapów). Kolejne uruchomienia programu już tego nie wymagają, pod warunkiem oczywiście, że zmienne odpowiadające etapom zadeklarowano jako pamiętane.

Na rysunku 1.11 przedstawiono przykładową realizację programową przejścia pomiędzy etapami podstawowymi.



Rys.1.11. Przykład przejścia w grafie z rysunku 1.10 pomiędzy etapami podstawowymi

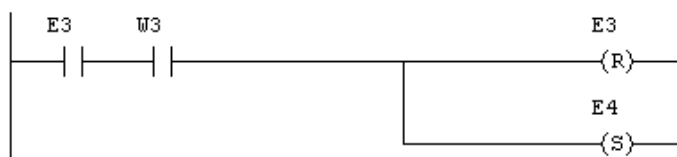
Na rysunku 1.12 przedstawiono natomiast realizację programową przejść pomiędzy etapem podstawowym E3 a etapem podstawowym E4 oraz pomiędzy etapem podstawowym E3 a etapem zabezpieczeń A1.

W podobny sposób wygląda realizacja programowa w przypadku wyboru jednego wariantu spośród kilku wariantów sterowania (realizacji procesu sekwencyjnego). Na rysunku 1.13 przedstawiono przykładowy fragment grafu z wyborem (alternatywą) wariantu sterowania (w tym przypadku jeden spośród trzech). Realizowany będzie ten wariant sterowania, dla którego warunek przejścia będzie spełniony jako pierwszy. Sprawdzenie odbywa się w ustalonej kolejności przejść (warunków przejść) czyli najpierw W15, następnie W16 i na końcu W17. Aby sterowanie było określone, jeden z warunków (najczęściej ostatni) powinien być dopełnieniem logicznym wszystkich poprzednich, czyli W17 powinien być negacją sumy logicznej W15 i W16. Dla poszczególnych wariantów sterowania graf może zawierać również etapy zabezpieczeń.

□

0004

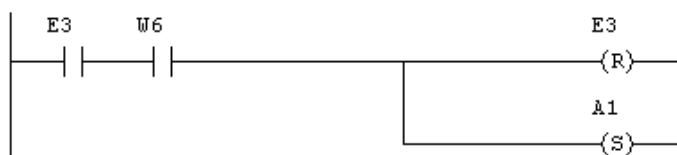
Dezaktywacja etapu podstawowego E3 i aktywacja etapu podstawowego E4



□

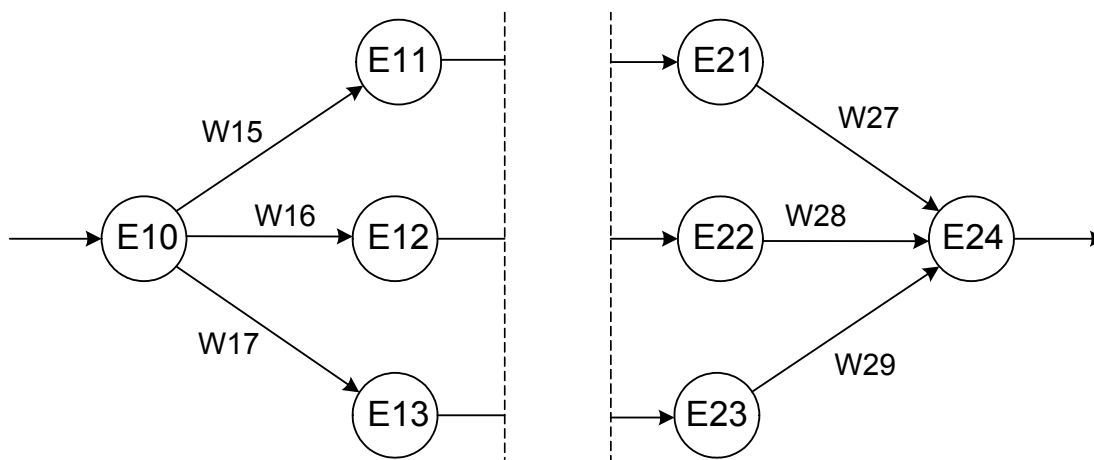
0005

Dezaktywacja etapu podstawowego E3 i aktywacja etapu awaryjnego A1



□

Rys.1.12. Przykłady możliwości przejść w grafie z rysunku 1.10 pomiędzy etapami podstawowymi oraz pomiędzy etapem podstawowym a etapem awaryjnym

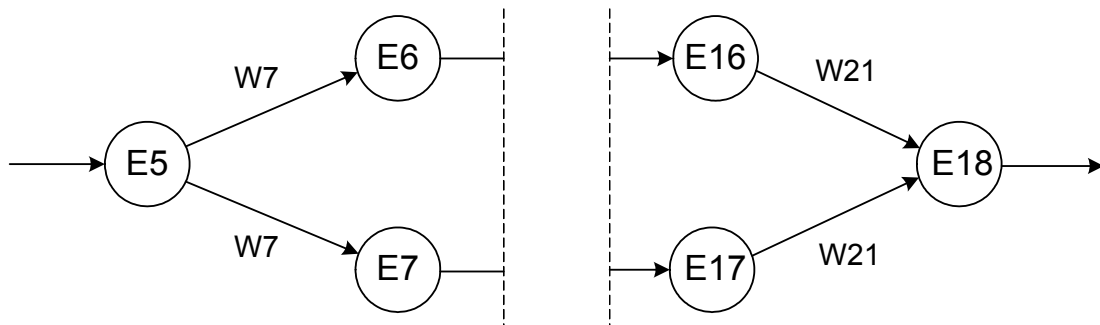


Rys.1.13. Przykładowy fragment grafu dla wyboru (alternatywy) sterowania

W przypadku synchronizacji (koniunkcji) sterowania realizacja programowa musi uwzględniać wszystkie etapy które muszą stać się aktywne równocześnie, jak również wszystkie etapy które równocześnie muszą przestać być aktywne.

Na rysunku 1.14 przedstawiono przykładowy fragment grafu z synchronizacją (koniunkcją) sterowania (w tym przypadku dla dwóch etapów: E6 i E7). Synchronizacja jest realizowana dzięki zdefiniowaniu takich samych warunków przejść pomiędzy etapami E5 i E6 oraz

między E5 i E7. Realizację programową rozpoczęcia synchronizacji sterowań przedstawiono na rysunku 1.15.

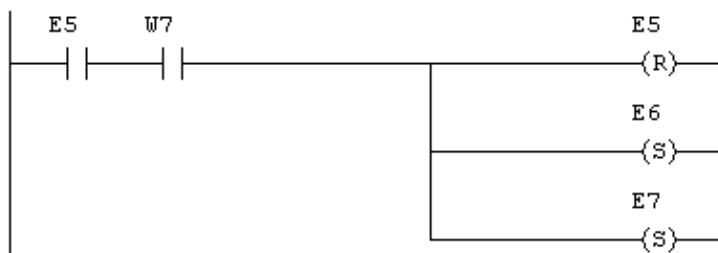


Rys.1.14. Przykładowy fragment grafu dla synchronizacji (koniunkcji) sterowania

□

0008

Dezaktywacja etapu E5 i aktywacja etapów E6 i E7 w galeziach równoległych ( synchronizacja )



□

Rys.1.15. Przykład przejścia w grafie z rysunku 1.14 do etapów w gałęziach równoległych z synchronizacją sterowania

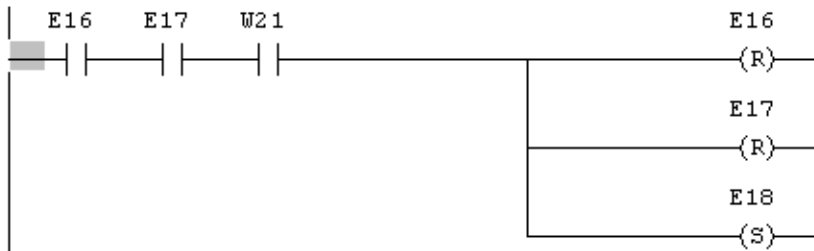
Realizację programową zakończenia synchronizacji sterowań przedstawiono na rysunku 1.16. Zakończenie synchronizacji może być zrealizowane wówczas, gdy wszystkie etapy kończące gałęzi równoległe staną się aktywne. Dla rozpatrywanego przykładu są to etapy E16 i E17. Przy zakończeniu synchronizacji również musi być zdefiniowany dla każdej gałęzi sterowania ten sam warunek przejścia. Tak więc aktywacja etapu E18 nastąpi tylko wówczas, gdy aktywne są etapy E16 i E17 oraz spełniony jest warunek przejścia W21.

Z każdym z etapów są powiązane odpowiednie działania, które wykonywane są tylko wówczas, gdy dany etap jest aktywny, czyli gdy zmienna dla danego etapu określająca jego aktywność ma wartość logiczną TRUE.

□

0009

Dezaktywacja etapów E16 i E17 w gałęziach równoległych ( koniec synchronizacji ) i aktywacja etapu E18



□

Rys.1.16. Przykład przejścia w grafie z rysunku 1.14 z gałęzi równoległych z synchronizacją sterowania do jednego wspólnego etapu podstawowego

Przedstawione w tym podrozdziale modelowanie procesu sekwencyjnego z wykorzystaniem grafu skierowanego oraz jego realizacja programowa mogą być przydatne do stworzenia algorytmu sterowania i programu sterowania cyklami jazdy silnika liniowego na bieżni liniowej omawianego w rozdziale 4.