

Sterowniki PLC

Wprowadzenie – nazewnictwo

- Sterownik PLC (Programmable Logic Controller)
- Sterownik SPS (Speicherprogrammierbare Steuerung)
- Programowalny sterownik logiczny (sterownik programowalny)
- Sterownik binarny

Historia

- Sterowniki PLC wprowadzono na linie produkcyjne na początku lat 70 XX wieku w celu zastąpienia sztywno połączonych układów setek lub więcej przekaźników w szafach sterowniczych jednym urządzeniem swobodnie programowalnym (przekaźnik programowalny)

Sterowniki PLC

Dlaczego sterowniki PLC stały się tak popularne?

- Podobieństwo schematów drabinkowych używanych w oprogramowaniu PLC do stosowanych schematów stykowo – przekaźnikowych
- Łatwość i szybkość programowania sterowników bez znajomości specjalistycznej wiedzy na temat systemów komputerowych
- Możliwość swobodnej zmiany programu sterującego bez konieczności modyfikacji struktury sprzętowej sterownika i sieci połączeń między urządzeniami
- Oferowany bogaty zestaw modułów wejść/wyjść cyfrowych, analogowych, dedykowanych np. do sterowania napędami, modułów komunikacyjnych

Zadania sterowników PLC

Zadania realizowane przez sterowniki

- Sterowanie maszynami lub procesami zgodnie z zapisanym programem
- Realizacja funkcji diagnostycznych sterowanych urządzeń i procesów (również autodiagnostyka)
- Kontrola układów zabezpieczeń (specjalne sterowniki bezpieczeństwa np. Wieland)
- Komunikacja z systemami zarządzania produkcją i urządzeniami typu HMI (panele operatorskie, monitory itp.) umożliwiająca na bieżąco prezentację i archiwizację stanu procesów produkcyjnych (współpraca z systemami SCADA)

3

Definicja sterownika PLC

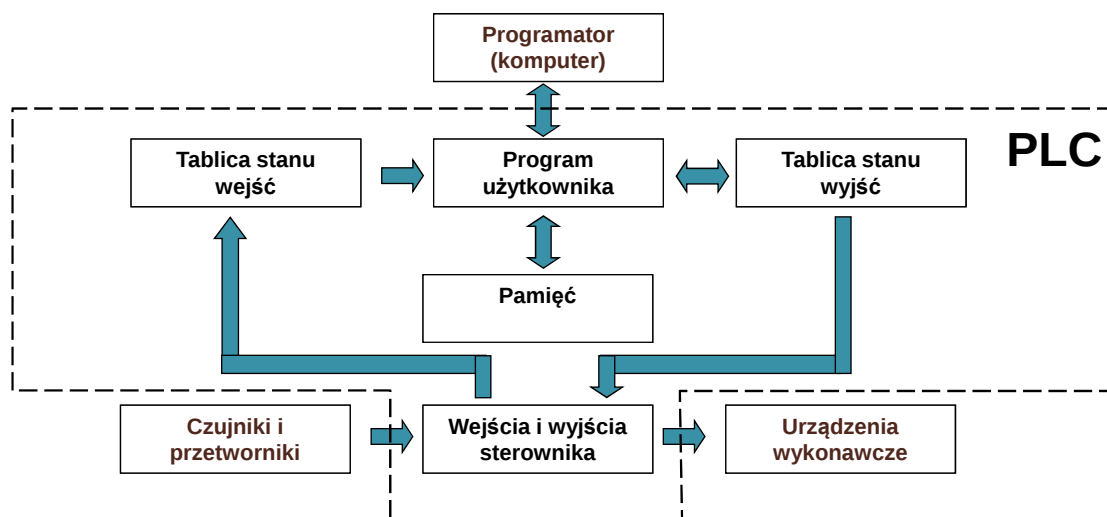
- Sterownik PLC to uniwersalne urządzenie, które pobiera informacje z przełączników, czujników, modułów komunikacyjnych itp. i zgodnie ze swoim programem generuje sygnały wyjściowe, które służą do sterowania maszyną lub procesem



4

Definicja sterownika PLC

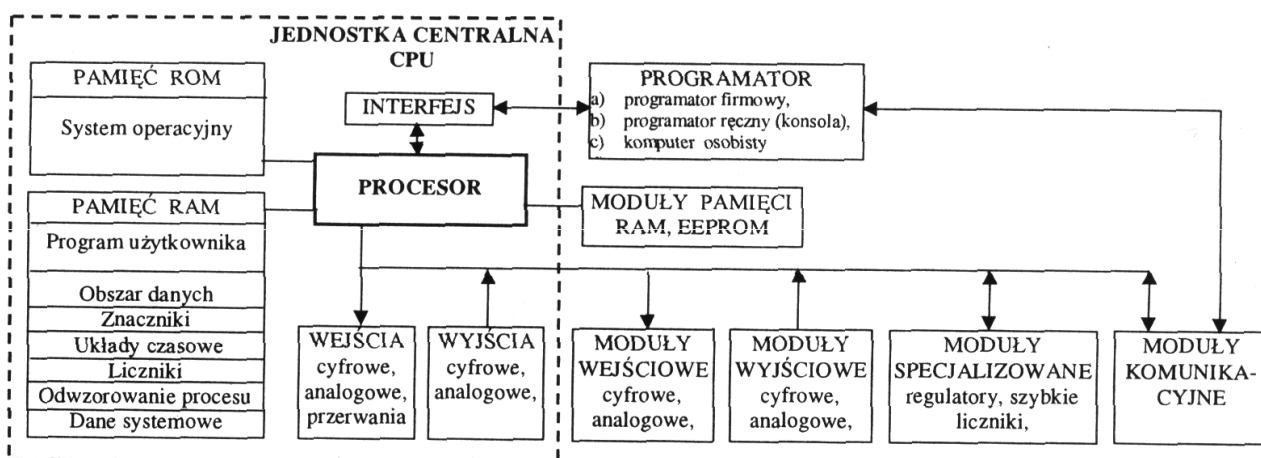
- Sterownik PLC jest układem komputerowym (systemem mikroprocesorowym jedno lub wieloprocesorowym) wyposażonym w rozbudowany układ wejść/wyjść, którego języki programowania umożliwiają proste i szybkie zapisanie programu sterującego



5

Definicja sterownika PLC

- Schemat blokowy sterownika PLC



6

Zasada działania sterownika PLC

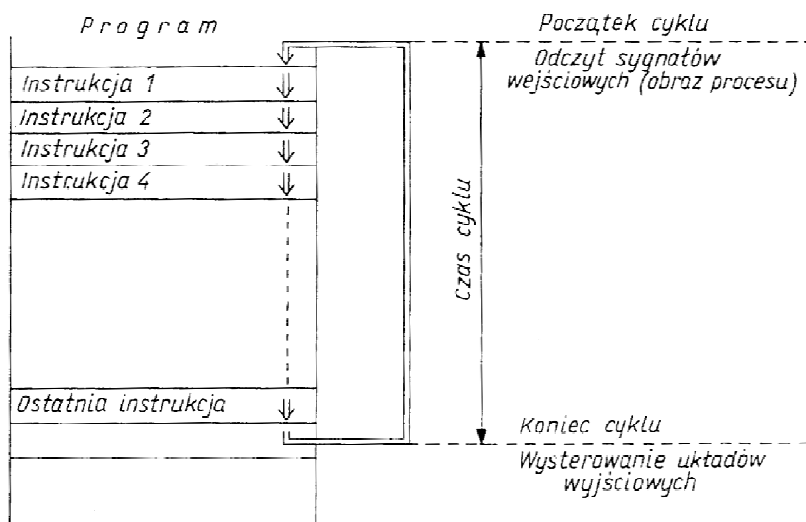
- Podstawową zasadą pracy sterowników jest praca cykliczna (cykliczny obieg pamięci programu), w której sterownik wykonuje kolejno po sobie pojedyncze rozkazy programu w takiej kolejności, w jakiej są one zapisane w programie.
- Pojedynczy cykl pracy sterownika obejmuje:
 1. Autodiagnostykę
 2. Odczyt stanu wejść do tablicy stanu wejść
 3. Wykonanie programu użytkownika
 4. Realizację zadań komunikacyjnych
 5. Ustawienie wyjść – przepisanie z tablicy stanu wyjść na wyjścia fizyczne

7

Zasada działania sterownika PLC

Cykl pracy sterownika

- Skan sterownika to jednokrotne wykonanie wszystkich cyklicznie powtarzanych operacji (inicjalizacja cyklu, diagnostyka, czytanie wejść, wykonanie programu, ustawienie wyjść, komunikacja)



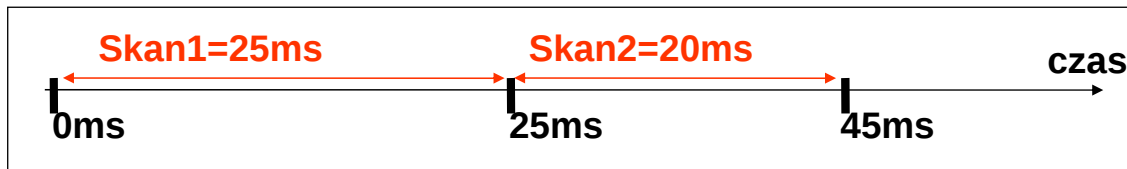
Czas cyklu sterownika, czas skanu (Scan Time): czas potrzebny procesorowi do odczytania stanu wszystkich wejść, realizacji programu oraz ustawienia wyjść zgodnie z realizowanym programem. „Skan” sterownika powtarzany jest cyklicznie gdy sterownik pracuje w trybie RUN.

8

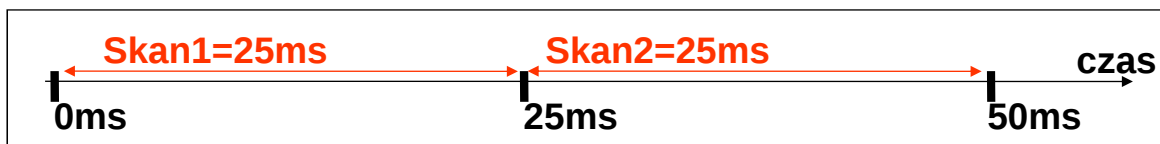
Zasada działania sterownika PLC

Typy cykli (skanów) sterownika

- Normal Sweep: sterownik wykonuje operacje kolejno po sobie, bez wprowadzania żadnych dodatkowych opóźnień



- Constant Sweep: sterownik zaczyna wykonywanie ciągu operacji, wchodzących w skład cyklu pracy sterownika, co ściśle określony czas. Z reguły sterownik w takim przypadku „czeka” przez odpowiedni czas, tak aby zacząć kolejny cykl po upływie ściśle określonego czasu od momentu rozpoczęcia poprzedniego cyklu.



9

Zasada działania sterownika PLC

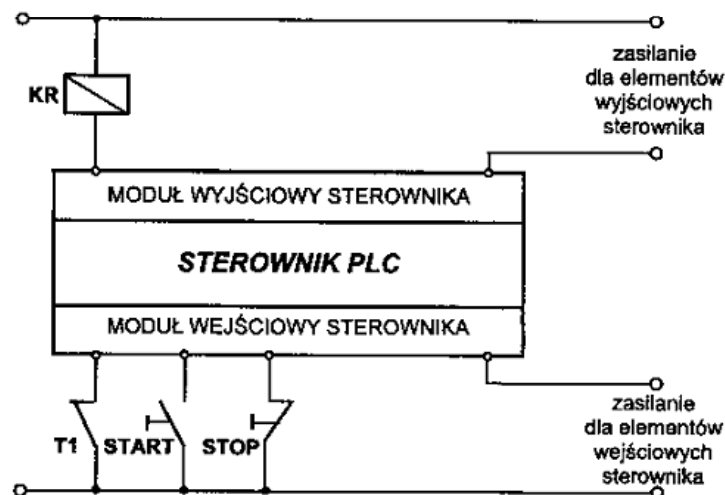
Działanie sterownika - uwagi

- Wykonanie programu trwa pewien czas zależny od parametrów i długości programu (kilka do kilkuset milisekund). W najgorszym przypadku opóźnienie wnoszone przez sterownik może osiągnąć do 2 cykli sterownika.
- Stany wejść są dla całego przebiegu programu takie same (nie ma niejednoznaczności polegającej na tym, że w pewnej części programu sygnał ma wartość 1 a w innej 0).
- Jeżeli sygnał wejściowy trwa krócej niż jeden cykl, to nie ma pewności, czy zostanie przez sterownik zauważony (sygnały o częstotliwości kilku Hz mogą sprawiać problemy). W celu umożliwienia szybkiej reakcji na zmianę stanu na wejściu stosuje się przerwania alarmowe.

Układy wejściowe i wyjściowe sterownika PLC

- Wejścia (Inputs): elementy (część sterownika) pośredniczące w przekazywaniu informacji od czujników, przetworników, itp. do jednostki centralnej w sterowniku
- Wyjścia (Outputs): elementy (część sterownika) pośredniczące w przekazywaniu informacji z jednostki centralnej do urządzeń wykonawczych (silników, zaworów, lampek alarmowych, itp.)

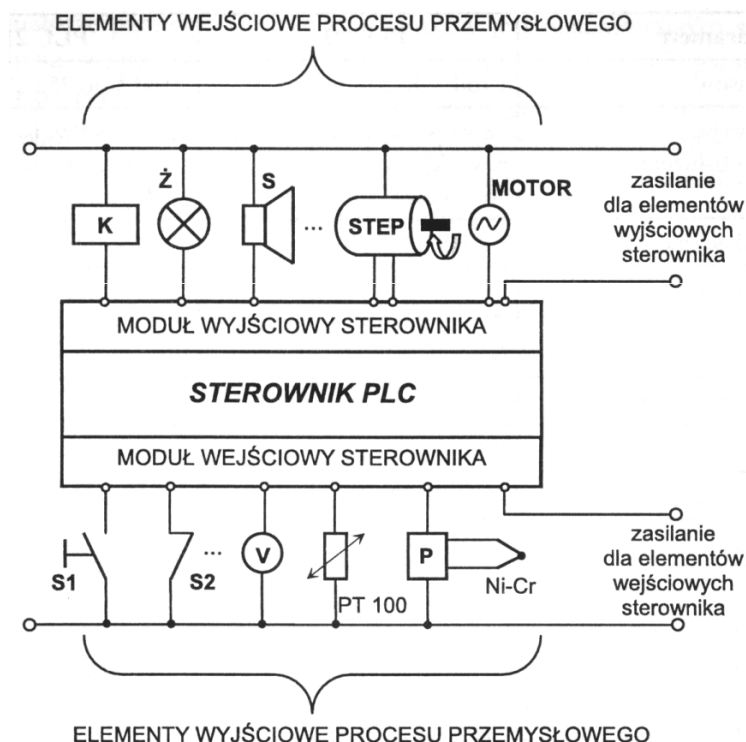
Przykład układu sterowania silnikiem



11

Układy wejściowe i wyjściowe sterownika PLC

Rodzaje sygnałów w układach wejść/wyjść

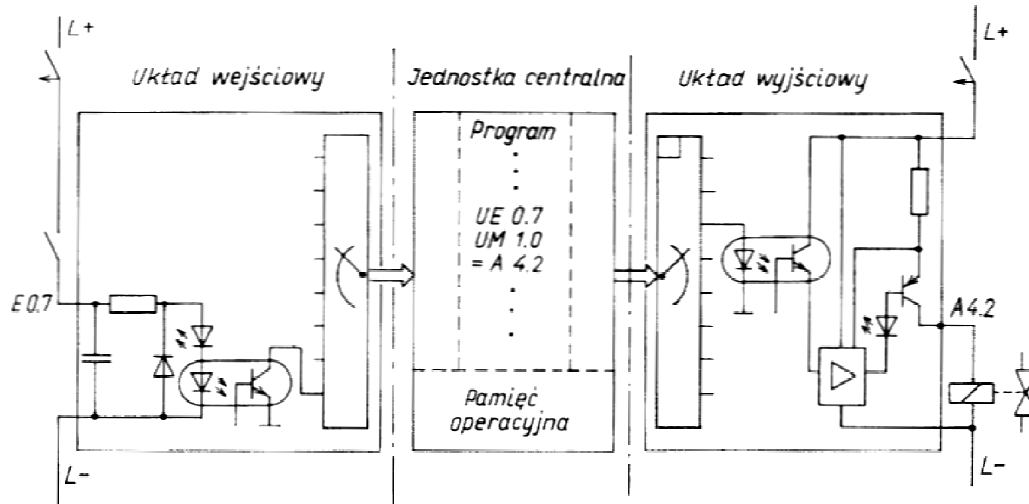


12

Układy wejściowe i wyjściowe sterownika PLC

- Sygnał dwustanowy (Discrete Signal, Binary Signal): sygnał przyjmujący dwa stany: 0 lub 1. Najczęściej doprowadzeniu napięcia 24VDC do wejścia odpowiada stan logiczny 1 a napięciu 0VDC odpowiada stan logiczny 0. Specjalne wejścia HSC (wejścia szybkich liczników).

Schemat ideowy dwustanowego układu wejściowego i wyjściowego



13

Układy wejściowe i wyjściowe sterownika PLC

- Sygnał analogowy (Analog Signal): sygnał przyjmujący wartości ciągłe z pewnego przedziału. W automatyce z reguły występują sygnały w standardach:
 - (0...20)mA
 - (4...20)mA
 - (0...10)V
 - (-10 ... +10)V
 - oprócz tego wykorzystuje się sygnały typu Pt100, tensometry, termopary

W przypadku układów wejściowych sygnał analogowy jest przetwarzany na reprezentację cyfrową w sterowniku – konwersja w przetwornikach A/C

Dla układów wyjściowych wartość cyfrowa przetwarzana jest na postać analogową w przetwornikach C/A

14

Pamięć sterownika PLC

- W pamięci sterownika przechowywane są tymczasowo lub trwale programy sterujące, ustawienia konfiguracyjne oraz dane.
- Typ pamięci w **sensie sprzętowym**: rodzaj pamięci do przechowywania programu sterującego, konfiguracji sprzętowej czy danych. Wyróżnia się pamięć ulotną (RAM), zazwyczaj podtrzymywaną bateryjnie oraz pamięć nieulotną (ROM, EPROM EEPROM lub FLASH).
- Typ pamięci w sterowniku w **sensie programistycznym** związany z typami zmiennych używanych w programie sterującym (typy związane z wejściami, wyjściami, znacznikami, stałymi, danymi, licznikami).
- Korzystanie z pamięci wiąże się z jej **adresowaniem**. Adres to wartość numeryczna która w połączeniu z typem zmiennej jednoznacznie określa miejsce w pamięci sterownika.

15

Języki programowania sterowników PLC

Firmy produkujące sterowniki programowalne dostarczają zwykle wraz z nimi środowisko programistyczne, pozwalające pisać aplikacje w jednym lub kilku językach programowania. Języki te są zwykle mniej lub bardziej dokładną implementacją zaleceń normy IEC 61131-3, która definiuje następujące warianty:

- LD (Ladder Diagram) logika drabinkowa - język programowania podobny do klasycznego zapisu na schematach stykowo-przełącznikowych;
- FBD (Function Block Diagram) - diagram bloków funkcyjnych, sekwencja linii zawierających bloki funkcji logicznych;
- ST (Structured Text) tekst strukturalny - język zbliżony do Pascala/C;
- IL (Instruction List) lista instrukcji - język assemblera dla sterowników;
- SFC (Sequential Function Chart) sekwencyjny ciąg bloków - sekwencja bloków programowych z ustalonymi warunkami przejścia.

16

Język drabinkowy LD

Podstawowe symbole używane w języku drabinkowym

- Styk przekaźnika normalnie otwarty (Normally Open Contact): symbol używany w programie, który „przewodzi” sygnał, gdy zmienna z nim związana ma wartość 1.

-| |-

- Styk przekaźnika normalnie zamknięty (Normally Closed Contact): symbol używany w programie, który „przewodzi” sygnał, gdy zmienna z nim związana ma wartość 0.

-|/ |-

17

Język drabinkowy LD

- Przekaznik normalnie otwarty (Normally Open Coil): symbol używany w programie, który ustawia na 1 wartość zmiennej z nim związanej, gdy dopływa do niego sygnał.

-()-

- Przekaznik normalnie zamknięty (Normally Close Coil): symbol używany w programie, który ustawia na 0 wartość zmiennej z nim związanej, gdy dopływa do niego sygnał.

-(/)-

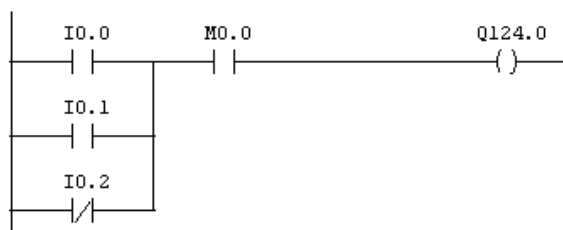
18

Organizacja programu w języku drabinkowym

- Program jest wykonywany przez sterownik od lewej do prawej i z góry na dół (za wyjątkiem przypadków kiedy użyte są instrukcje związane ze zmianą wykonywanej sekwencji w programie - skoki).
- Szczegół programu (przykład poniżej) jest to połączenie lewej i prawej linii pionowej w drabince, symbolizujące przepływ prądu w układach przekaźnikowych, zawierające styki, cewki, bloki funkcyjne, itp.

Network 1: Pokaz

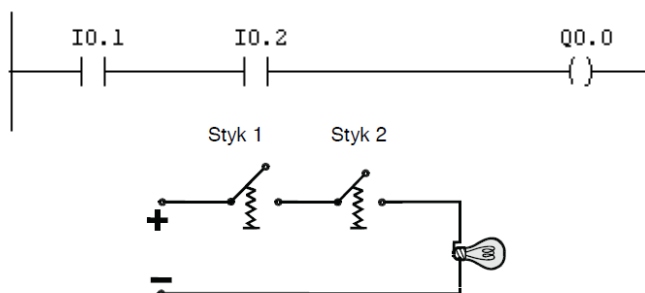
Przykład



19

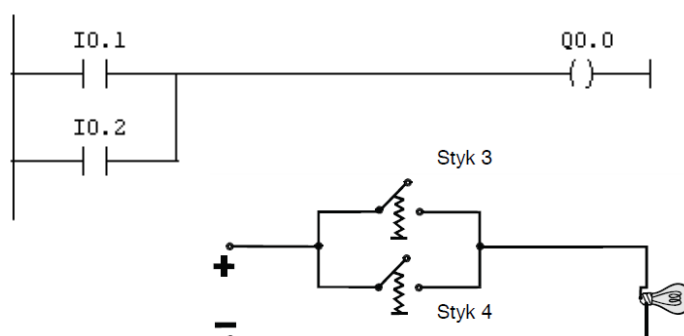
Układy kombinacyjne w języku drabinkowym

- Iloczyn logiczny (funkcja AND)



I0.1	I0.2	Q0.0
0	0	0
0	1	0
1	0	0
1	1	1

- Suma logiczna (funkcja OR)

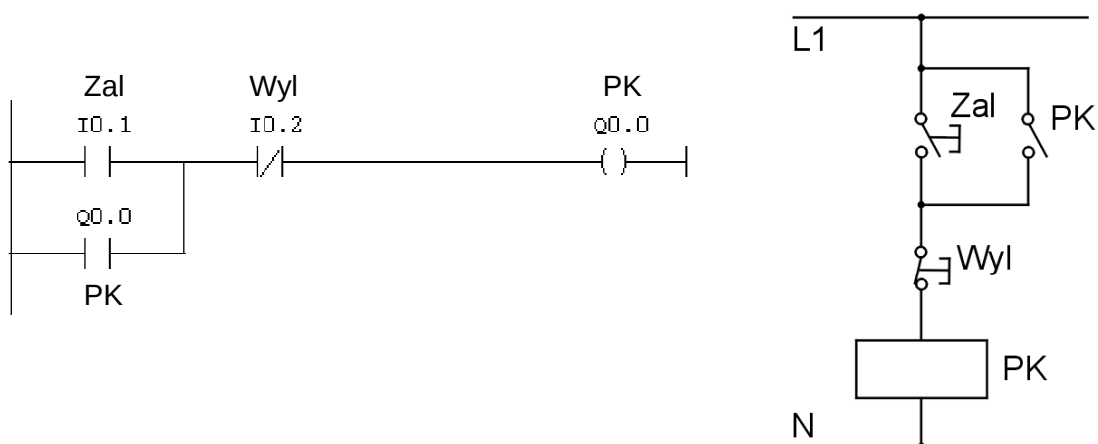


I0.1	I0.2	Q0.0
0	0	0
0	1	1
1	0	1
1	1	1

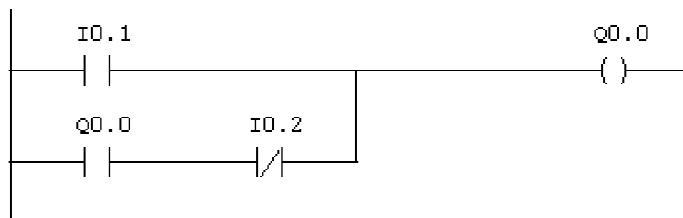
20

Funkcje pamięciowe w języku drabinkowym

- Samopodtrzymanie w układzie z uprzywilejowaniem wyłączenia



- Samopodtrzymanie w układzie z uprzywilejowaniem załączenia



21

Programowanie sterowników Siemens S7

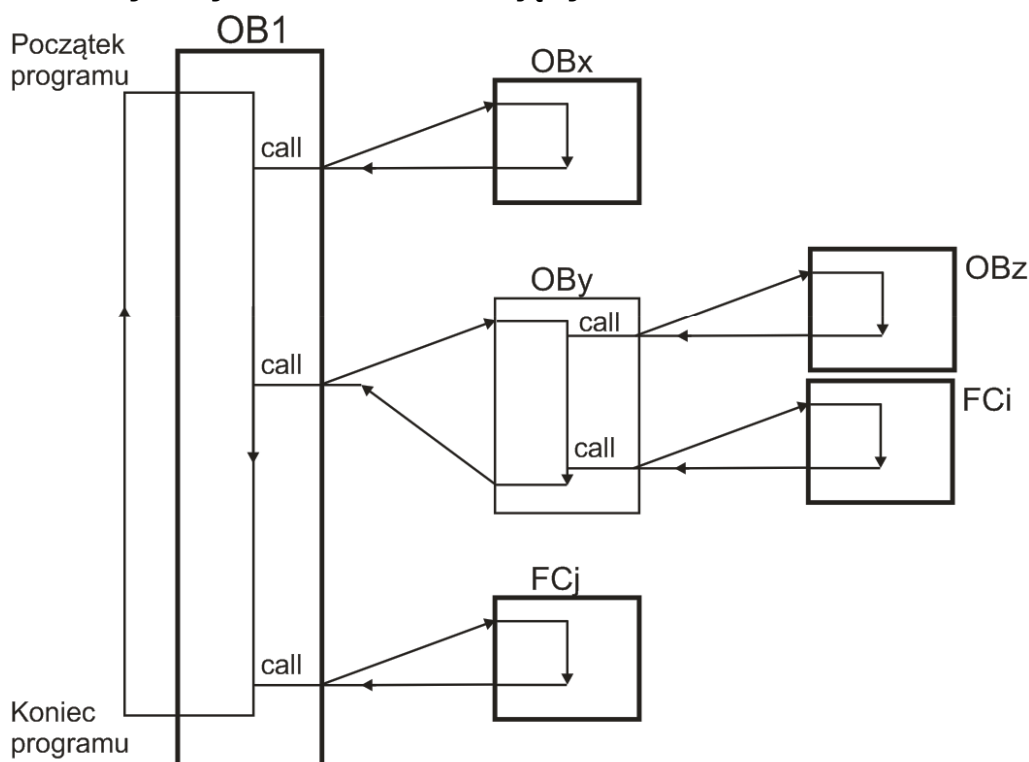
Bloki programowe w języku STEP 7

- Bloki organizacyjne (OB) – służą do zarządzania programem użytkownika i są wywoływane przez system operacyjny sterownika. **Głównym blokiem programu wywoływanym cyklicznie jest blok OB1.** Pozostałe bloki są wywoływane w takiej kolejności, w jakiej zostały umieszczone w bloku OB1.
- Bloki funkcyjne (FB, FC) – zasadniczym przeznaczeniem jest realizacja złożonych lub powtarzalnych funkcji z możliwością parametryzacji, tzn. wywoływania ich w różnych punktach programu i wykonywaniu z różnymi argumentami (parametrami bloku).
- Bloki danych (DB, UDT, VAT) – służą do przechowywania danych wykorzystywanych w programie użytkownika, mogą być wykorzystane do parametryzacji wywołania bloków funkcyjnych.

22

Programowanie sterowników Siemens S7

Schemat wywoływania bloków w języku STEP 7



23

Programowanie sterowników Siemens S7

Typy argumentów (zmiennych) w języku STEP 7

- Wejścia I – zapamiętane na początku cyklu sygnały wejściowe,
- Wyjścia Q – wyliczone wartości sygnałów wyjściowych,
- Znaczniki M – przeznaczone do zapamiętania wyników działań logicznych,
- Dane D – elementy bloków danych,
- Timery T – przeznaczone do realizacji funkcji czasowych,
- Peryferia PI, PQ – te same sygnały co wejścia I i wyjścia Q, ale przesyłane w komunikacji bezpośredniej w momencie wykonania instrukcji bez czekania na koniec cyklu programowego,
- Liczniki C – umożliwiają realizację funkcji związanych z liczeniem impulsów.

24

Programowanie sterowników Siemens S7

Adresowanie w języku STEP 7

- Adresowanie wejść/wyjść/znaczników (zmiennych logicznych)

I 10.7

Q 1 .0

M 2 .5



Numer bitu w bajcie

Numer bajtu w adresie

Argument (określa typ zmiennej)

Zmienne logiczne mogą też być adresowane bajtowo np. IB 10 oraz za pomocą słowa np. IW 10

- Adresowanie danych (elementów bloków danych) odbywa się za pomocą słów, np. DW 5 lub można adresować poszczególne bity w słowie D 5.14
- Adresy liczników, timerów obraz bloków są liczbami całkowitymi podawanymi obok ich symboli, np. OB10, FB15

25

Programowanie sterowników Siemens S7

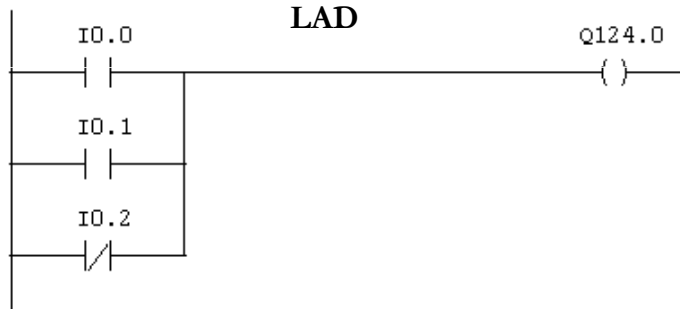
Sposoby zapisu programów w języku STEP 7

- Lista instrukcji (STL – Statement List)
- Schemat drabinkowy (LAD – Ladder Logic)
- Schemat funkcjonalny (FBD – Function Block Diagram)

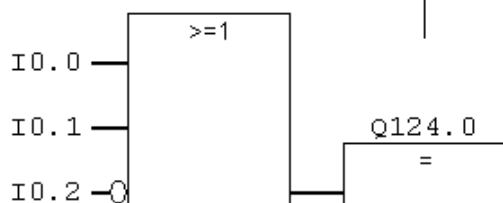
STL

```
O      I      0.0
O      I      0.1
ON     I      0.2
=      Q      124.0
```

LAD



FBD

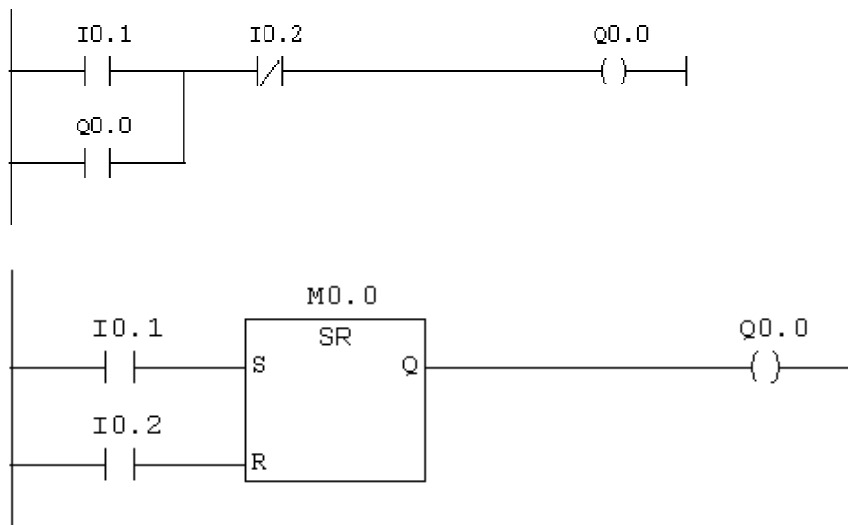


26

Programowanie sterowników Siemens S7

Funkcje pamięciowe w języku STEP 7

- Przerzutniki RS i SR



Funkcję ustawiania i kasowania w STEP7 pełnią bloki -(S)- oraz -(R)-

27

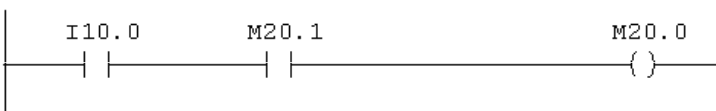
Programowanie sterowników Siemens S7

Wykrywanie zbocza sygnału

- Zbocze narastające

Network 3: Wykrywanie zbocza narastającego

Znacznik M20.0 jest znacznikiem zbocza na wejściu I10.0, jeśli M20.0 ma wartość 1 to wykryto zbocze narastające (tylko przez jeden cykl programu)



Network 4: Wykrywanie zbocza narastającego cd

Comment:



Wykrycie zbocza narastającego w STEP 7 realizuje blok -(P)-

28

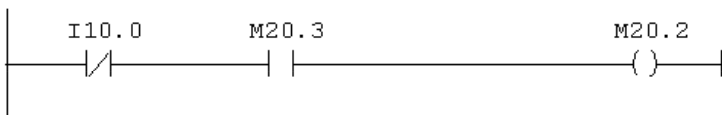
Programowanie sterowników Siemens S7

Wykrywanie zbocza sygnału

- Zbocze opadające

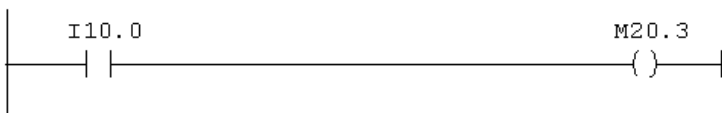
Network 5: Wykrywanie zbocza opadającego

Znacznik M20.2 jest znacznikiem zbocza na wejściu I10.0, jeśli M20.2 ma wartość 1 to wykryto zbocze opadające (tylko przez jeden cykl programu)



Network 6: Title:

Comment:



Wykrycie zbocza opadającego w STEP 7 realizuje blok -(N)-

29

Programowanie sterowników Siemens S7

Funkcje czasowe

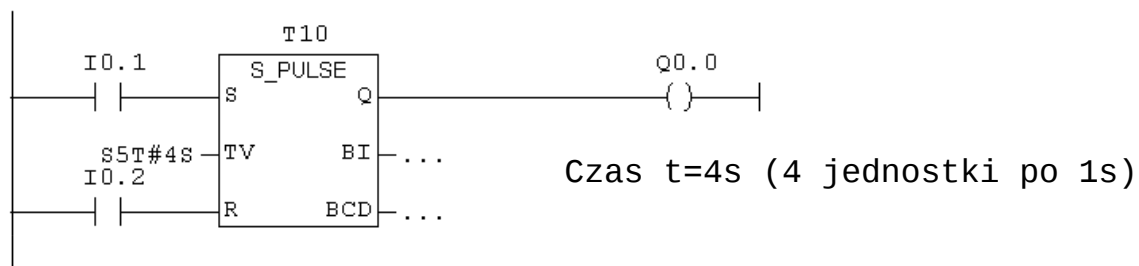
- Impuls – funkcja umożliwiająca odmierzenie zadanego czasu od momentu wyzwolenia
- Przedłużony impuls – funkcja odliczająca zadany czas od momentu wyzwolenia z możliwością przedłużenia odmierzania czasu, gdy pojawi się kolejny impuls na wejściu ustawiającym
- Opóźnione załączenie – funkcja umożliwiająca załączenie po zadanym czasie
- Opóźnione załączenie z pamięcią – funkcja umożliwiająca załączenie po zadanym czasie bez konieczności utrzymywania sygnału na wejściu ustawiającym
- Opóźnione wyłączenie – funkcja umożliwiająca wyłączenie po zadanym czasie

30

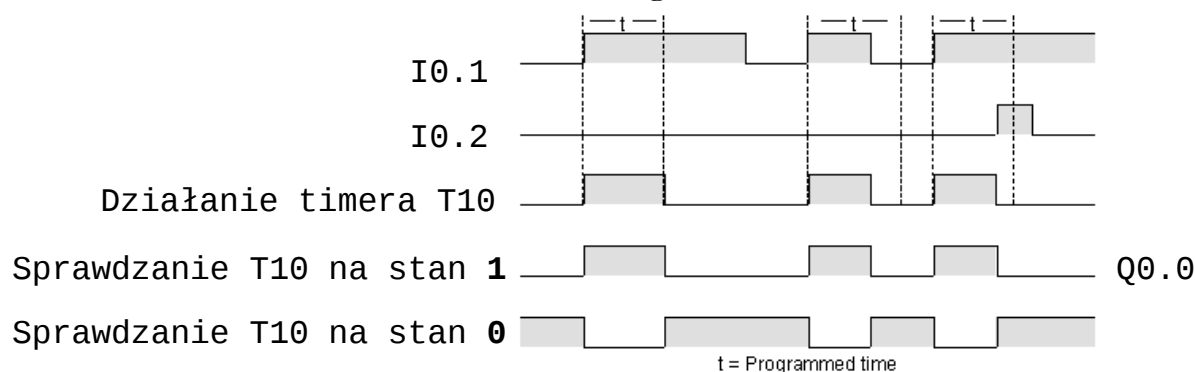
Programowanie sterowników Siemens S7

Funkcje czasowe

- Funkcja impuls S_PULSE



Przebiegi czasowe

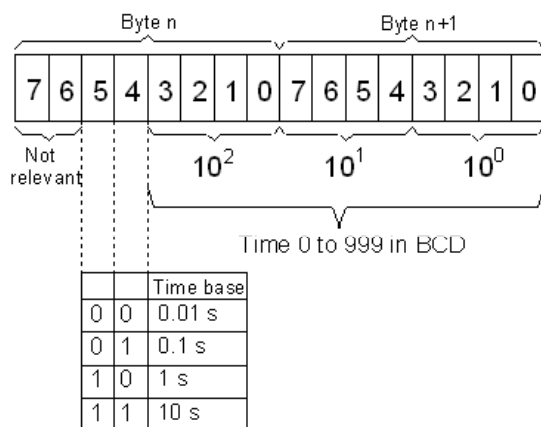


31

Programowanie sterowników Siemens S7

Format czasu S5TIME

- Typ danych 16 bitowy



Format:

- H M S
- lub
- M S MS

Wartość

maksymalna:

S5T#2H46M30S

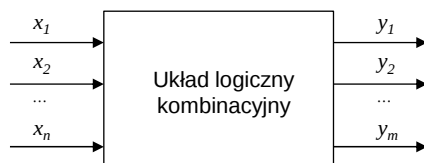
- Ograniczenia formatu czasu w blokach typu Timer związane z rozdzielczością
 - 0,01s zakres od 10ms do 9s 990ms
 - 0,1s zakres od 100ms do 1m 39s 900ms
 - 1s zakres od 1s do 16m 39s
 - 10s zakres od 10s do 2h 46m 30s

32

Układy logiczne

Funkcje logiczne (boolowskie)

- Funkcja logiczna f może przyjmować wartości 1 lub 0 (lub być nieokreślona) zależnie od stanu wektora zmiennych wejściowych (argumentów o wartościach 0 lub 1)



Dla układu kombinacyjnego związek między sygnałami wejściowymi a wyjściowymi opisany jest funkcjami logicznymi $f_1, f_2, \dots, f_m$

$$y_1 = f_1(x_1, x_2, \dots, x_n)$$

gdzie

$$y_2 = f_2(x_1, x_2, \dots, x_n)$$

$$x = (x_1, x_2, \dots, x_n)$$

...

jest zbiorem wartości zmiennych wejściowych w danej chwili (0 lub 1), nazywany stanem wejść

$$y_m = f_m(x_1, x_2, \dots, x_n)$$

- Funkcje logiczne jednej zmiennej

x	0	1	Nazwa funkcji	Równanie
$f_0(x)$	0	0	Stała zerowa	$y = 0$
$f_1(x)$	0	1	Powtórzenie	$y = x$
$f_2(x)$	1	0	Negacja	$y = \bar{x}$
$f_3(x)$	1	1	Stała jedynkowa	$y = 1$

33

Układy logiczne

- Wybrane funkcje logiczne dwóch zmiennych

x_1 x_2	0 0	0 1	1 0	1 1	Nazwa funkcji	Równanie
$f_1(x)$	0	0	0	1	Koniunkcja (iloczyn logiczny), AND	$y = x_1 x_2$
$f_6(x)$	0	1	1	0	Nierównoważność, EX-OR	$y = x_1 \bar{x}_2 + \bar{x}_1 x_2$
$f_7(x)$	0	1	1	1	Alternatywa (suma logiczna), OR	$y = x_1 + x_2$
$f_8(x)$	1	0	0	0	Negacja alternatywy, NOR	$y = \overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2$
$f_{14}(x)$	1	1	1	0	Negacja koniunkcji, NAND	$y = \overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$

Elementy algebry Boole'a

- Kolejność wykonywania działań (gdy wyrażenie nie zawiera nawiasów):

- 1) negacja,
- 2) koniunkcja,
- 3) alternatywa.

- Prawa przemienności i łączności

$$x_1 + x_2 = x_2 + x_1$$

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3 = x_1 + x_2 + x_3$$

$$x_1 x_2 = x_2 x_1$$

$$x_1 (x_2 x_3) = (x_1 x_2) x_3 = x_1 x_2 x_3$$

- Prawa rozdzielności

$$x_1 (x_2 + x_3) = x_1 x_2 + x_1 x_3$$

$$x_1 + x_2 x_3 = (x_1 + x_2)(x_1 + x_3)$$

34

Układy logiczne

- Prawa de Morgana:

$$x_1 + x_2 = \bar{x}_1 \bar{x}_2$$

$$\overline{x_1 + x_2 + x_3 + \dots} = \bar{x}_1 \bar{x}_2 \bar{x}_3 \dots$$

$$\overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$$

$$\overline{x_1 x_2 x_3 \dots} = \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \dots$$

- Prawo podwójnego zaprzeczenia wynikające z definicji negacji $\bar{\bar{x}} = x$

- Zależności wynikające z definicji funkcji alternatywy i koniunkcji:

$$x + 0 = x$$

$$x \cdot 0 = 0$$

$$x + 1 = 1$$

$$x \cdot 1 = x$$

$$x + x = x$$

$$x \cdot x = x$$

$$x + \bar{x} = 1$$

$$x \cdot \bar{x} = 0$$

- Operacje wykonywane na stałych zero i jeden:

$$0 + 0 = 0$$

$$0 \cdot 0 = 0$$

$$1 + 0 = 1$$

$$1 \cdot 0 = 0$$

$$1 + 1 = 1$$

$$1 \cdot 1 = 1$$

$$\bar{0} = 1$$

$$\bar{1} = 0$$

- Prawa pochłaniania

$$x_1 + x_1 x_2 = x_1$$

$$x_1(x_1 + x_2) = x_1$$

$$x_1 + x_1 \bar{x}_2 = x_1$$

$$x_1(x_1 + \bar{x}_2) = x_1$$

- Reguły sklejania

$$(x_1 + x_2)(x_1 + \bar{x}_2) = x_1$$

$$x_1 x_2 + x_1 \bar{x}_2 = x_1$$

$$x_1 + \bar{x}_1 x_2 = x_1 + x_2$$

$$\bar{x}_1 + x_1 x_2 = \bar{x}_1 + x_2$$

35

Układy logiczne

Funkcje logiczne w postaci kanonicznej (normalnej)

- Składniki jedynki i czynniki zera dla trzech zmiennych

i	x ₁	x ₂	x ₃	Składniki jedynki K _i	Czynniki zera D _i
0	0	0	0	K ₀ = $\bar{x}_1 \bar{x}_2 \bar{x}_3$	D ₀ = $x_1 + x_2 + x_3$
1	0	0	1	K ₁ = $\bar{x}_1 \bar{x}_2 x_3$	D ₁ = $x_1 + x_2 + \bar{x}_3$
2	0	1	0	K ₂ = $\bar{x}_1 x_2 \bar{x}_3$	D ₂ = $x_1 + \bar{x}_2 + x_3$
3	0	1	1	K ₃ = $\bar{x}_1 x_2 x_3$	D ₃ = $x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	K ₄ = $x_1 \bar{x}_2 \bar{x}_3$	D ₄ = $\bar{x}_1 + x_2 + x_3$
5	1	0	1	K ₅ = $x_1 \bar{x}_2 x_3$	D ₅ = $\bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	K ₆ = $x_1 x_2 \bar{x}_3$	D ₆ = $\bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	K ₇ = $x_1 x_2 x_3$	D ₇ = $\bar{x}_1 + \bar{x}_2 + \bar{x}_3$

- Kanoniczna postać alternatywna (rozkład względem składników jedności)

$$f(x_1, x_2, \dots, x_n) = \sum_{i=0}^{2^n-1} f_i K_i$$

- Kanoniczna postać koniunkcyjna (rozkład względem czynników zera)

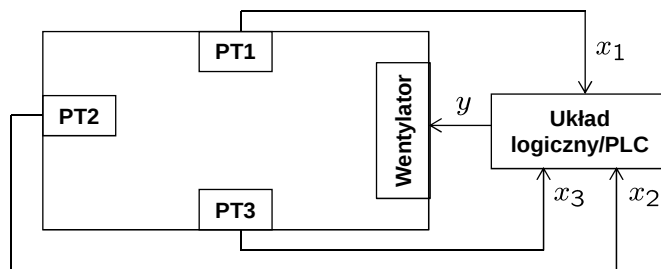
$$f(x_1, x_2, \dots, x_n) = \prod_{i=0}^{2^n-1} (f_i + D_i)$$

36

Układy logiczne

Przykład syntezy układu kombinacyjnego

- Opis słowny układu sterowania pracą wentylatora
Temperatura mierzona jest w trzech punktach hali produkcyjnej. Wentylator ma być uruchamiany, jeżeli temperatura w hali przekroczy 25°C co najmniej w dwóch punktach pomiarowych.



Dla temperatury poniżej 25°C, sygnały wejściowe $x_{1..3}$ przyjmują wartość 0, dla temperatury powyżej przyjmują 1. Wentylator jest wyłączony, gdy sygnał wyjściowy $y=0$, załączony dla $y=1$.

- Tablica prawdy

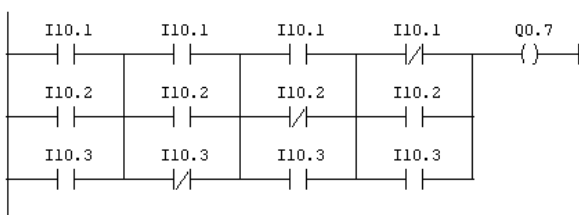
i	0	1	2	3	4	5	6	7
x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
y	0	0	0	1	0	1	1	1

37

Układy logiczne

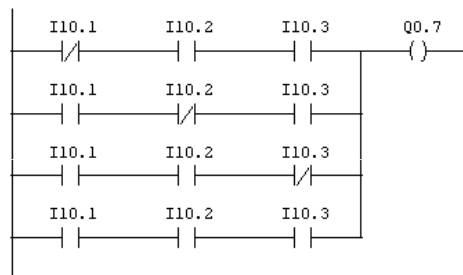
- Równanie funkcji logicznej w postaci kanonicznej koniunkcyjnej (względem czynników zera)

$$y = (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + x_2 + x_3)$$



- Równanie funkcji logicznej w postaci kanonicznej alternatywnej (względem składników jedności)

$$y = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3$$



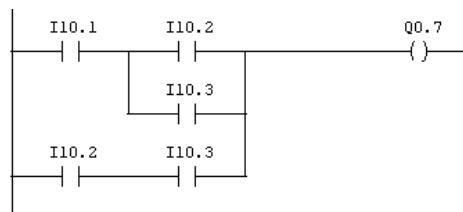
- Minimalizacja funkcji w postaci alternatywnej

$$y = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3 + x_1 x_2 x_3 + x_1 x_2 x_3 + x_1 x_2 x_3 + x_1 x_2 x_3$$

$$y = x_2 x_3 (\bar{x}_1 + x_1) + x_1 x_3 (\bar{x}_2 + x_2) + x_1 x_2 (\bar{x}_3 + x_3)$$

$$y = x_2 x_3 + x_1 x_3 + x_1 x_2$$

$$y = x_1 (x_2 + x_3) + x_2 x_3$$



38

Układy logiczne

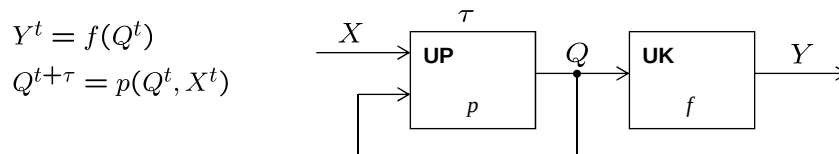
Synteza układów sekwencyjnych

- W układach sekwencyjnych sygnały wyjściowe są uzależnione nie tylko od kombinacji stanu wejść w danej chwili, lecz również od stanów poprzednich. Do opisu układów sekwencyjnych stosuje się automat Mealy'ego lub Moore'a.
- Automat Moore'a

Stan wejść X – zbiór wartości występujących w danej chwili sygnałów wejściowych

Stan wyjść Y – zbiór wartości występujących w danej chwili sygnałów wyjściowych

Stan wewnętrzny Q – zbiór wartości występujących w danej chwili sygnałów wyjściowych elementów pamięciowych (stan wewnętrzny automatu)



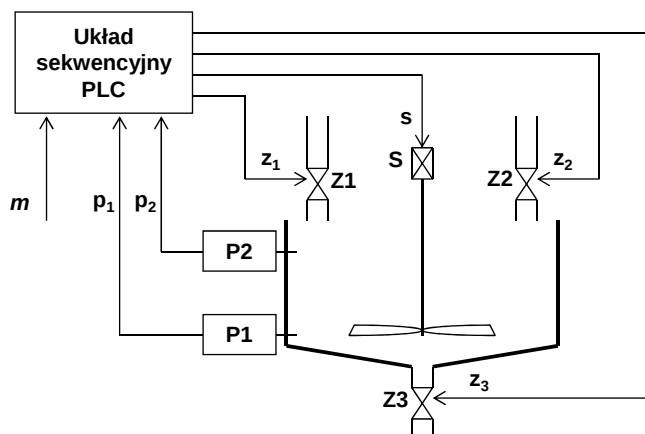
- Sposoby formułowania zadania układu sekwencyjnego:
 - opis słowny,
 - wykres czasowy lub odpowiadające mu ciągi zerojedynkowe (cyklogramy),
 - tablice przejść i wyjść,
 - graf stanów automatu.

39

Układy logiczne

Przykład syntezy układu sekwencyjnego

- Opis słowny układu mieszania dwóch substancji
W zbiorniku mieszane są dwie substancje w stosunku 1:1. Na początku zbiornik jest pusty, silnik mieszadła zatrzymany a zawory zamknięte. Przycisk m służy do uruchomienia całego cyklu napełniania, mieszania i opróżniania zbiornika. Po uruchomieniu cyklu zbiornik jest napełniany przez zawory Z1 i Z2. Po osiągnięciu poziomu P1 należy uruchomić mieszadło. Po osiągnięciu poziomu P2 dopływ obu substancji jest przerywany a rozpoczyna się wypuszczanie mieszanki. Gdy zostanie osiągnięty poziom P1 kończy się cykl, a rozpoczęcie kolejnego zależy od stanu przycisku m .



Stany automatu:

- 0 - Stop
- 1 - Napełnianie bez mieszania
- 2 - Napełnianie z mieszaniem
- 3 - Wypuszczenie z mieszaniem
- 4 - Wypuszczenie z mieszaniem c.d.

40

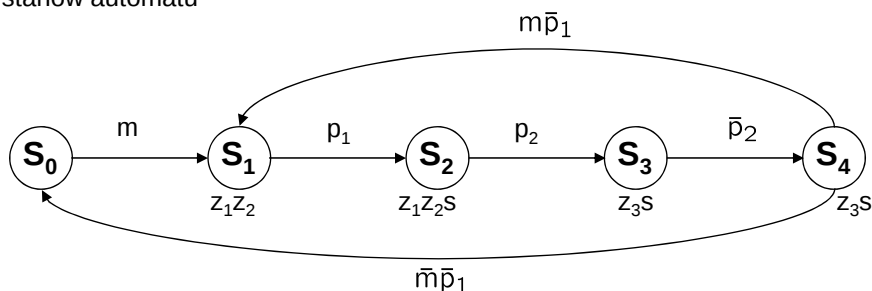
Układy logiczne

- Tablica przejść i wyjść

m p ₂ p ₁ Q	Tablica przejść								Tablica wyjść			
	000	001	010	011	100	101	110	111	z ₁	z ₂	z ₃	s
0	0	-	-	-	1	-	-	-	0	0	0	0
1	1	2	-	-	1	2	-	-	1	1	0	0
2	-	2	-	3	-	2	-	3	1	1	0	1
3	-	4	-	3	-	4	-	3	0	0	1	1
4	0	4	-	-	1	4	-	-	0	0	1	1

- Pogrubione stany oznaczają stany stabilne
- Szare pola w tablicy przejść są to kombinacje niedozwolone dla czujników poziomu, pola oznaczone kreską określają niedozwolone przejścia ze względu na działanie układu

- Graf stanów automatu



41

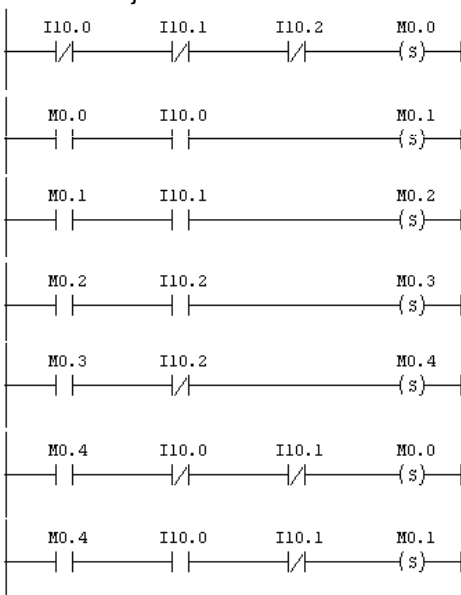
Układy logiczne

- Program w języku drabinkowym bez minimalizacji tablicy przejść i wyjść
 - Część związana z aktualizacją stanu
 - Część porządkująca stany, kontrola stanów niedozwolonych
 - Część ustawiająca wyjścia

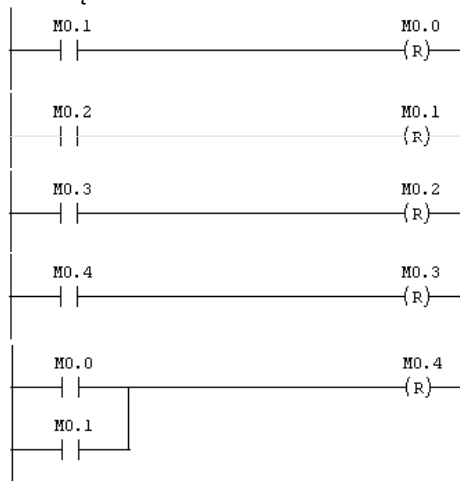
Oznaczenia:

m - I10.0, p₁ - I10.1, p₂ - I10.2; Q0:Q4 - M0.0:M0.4; z₁:z₃ - Q0.1:Q0.3, s - Q0.4

Aktualizacja stanu



Porządkowanie stanów



42

Układy logiczne

Ustawienie wyjść

