

SYNTEZA METODĄ GRAFPOL PROCEDUR WSPÓLBIEŻNYCH W KTÓRYCH REALIZACJA KROKÓW JEST SYGNALIZOWANA WARUNKAMI LOGICZNYMI I CZASOWYMI

Łukasz DWORZAK*, Tadeusz MIKULCZYŃSKI*

*Instytut Technologii Maszyn i Automatykacji, Politechnika Wroclawska, ul. Ignacego Łukasiewicza 5, 50-371 Wrocław

lukasz.dworzak@pwr.wroc.pl, tadeusz.mikulczynski@pwr.wroc.pl

Streszczenie: W pracy zaprezentowano zastosowanie metody Grafpol do syntezy współbieżnych algorytmów sterowania w których realizacja kroków jest sygnalizowana warunkami logicznymi i czasowymi. Opracowane zasady stanowią uproszczenie metody transformacji sieci (MTS) programowania procesów produkcyjnych w zakresie realizacji pamięci. Dzięki temu czas syntezy równania schematowego, w porównaniu z metodą MTS, uległ znacznemu skróceniu. Projektowane równanie schematowe stanowi podstawę do zapisu programu użytkowego, sterownika PLC, za pomocą dowolnego języka zdefiniowanego w normie PN-EN 61131-3.

1. WPROWADZENIE

W dobie kryzysu ekonomicznego oraz coraz krótszego cyklu życia produktów jedną z cech przedsiębiorstw produkcyjnych pozwalającą na uzyskanie przewagi konkurencyjnej jest zdolność szybkiego i elastycznego przeobrażania produkcji. Nerozerwanie wiąże się to z koniecznością przeprogramowania układów sterowania maszyn produkcyjnych. To z kolei wymusza konieczność poszukiwania szybkich, niezawodnych i prostych, w codziennym stosowaniu, metod umożliwiających opracowywanie algorytmów sterowania dla nowych procesów technologicznych. W pracy przedstawiono zasady stosowania metody Grafpol do modelowania i programowania sterowników PLC, która umożliwi syntezę równania schematowego także w odniesieniu do algorytmów sterowania procedurami współbieżnymi w których realizacja kroków jest sygnalizowana warunkami logicznymi i czasowymi.

2. SYNTEZA METODĄ GRAFPOL RÓWNANIA SCHEMATOWEGO

Równanie schematowe stanowiące sumę funkcji wszystkich zmiennych wyjść (Y) oraz elementarnych ko-

mórek pamięci (M), jest podstawą do zapisu programu użytkowego sterownika PLC za pomocą dowolnego języka zdefiniowanego w normie PN-EN 61131-3. Można je wyznaczyć za pomocą, opracowanej w Laboratorium Podstaw Automatykacji Politechniki Wrocławskiej metody Grafpol (Mikulczyński, 2006). Procedura syntezy metodą Grafpol równania schematowego składa się z pięciu etapów elementarnych (Rys. 1).

2.1. Synteza równania schematowego procedur współbieżnych

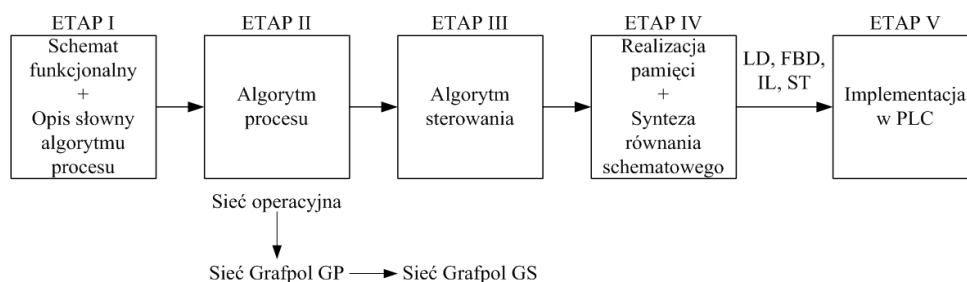
Syntezę równania schematowego współbieżnych algorytmów sterowania określają następujące zasady:

Zasada 1

Każdą procedurę sekwencyjną rozpatruje się niezależnie od pozostałych.

Zasada 2

Zgodnie z zasadami syntezy algorytmu sterowania metodą Grafpol (Dworzak i inni, 2009) dla poszczególnych procedur sekwencyjnych wyznacza się warunki zapisu i kasowania poszczególnych elementarnych komórek pamięci oraz określa funkcje zmiennych wyjściowych.



Rys. 1. Etapy modelowania wg metody Grafpol

Zasada 3

Postać tranzycji t_0^* , określającej współbieżne rozpoczęcie realizacji procedur sekwencyjnych opisuje zależność:

$$t_0^* = t_{1,0}^* \cdot t_{2,0}^* \cdot \dots \cdot t_{k,0}^* \quad (1)$$

gdzie: $t_{j,0}^*$ – zerowa tranzycja algorytmu sterowania procedury współbieżnej, $t_{k,n}^*$ – zerowa tranzycja algorytmu sterowania k-tej procedury sekwencyjnej.

Zasada 4

Zależność określająca kasowanie wszystkich użytych, w procedurach sekwencyjnych, elementarnych komórek pamięci, określa iloczyn:

$$\sum_{i=1}^k M_{i,1...L}(R) = t_{1,n}^* \cdot t_{2,n}^* \cdot \dots \cdot t_{k,n}^* \quad (2)$$

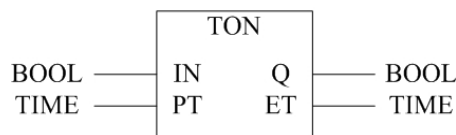
gdzie: L – liczba elementarnych komórek pamięci j -tej procedury sekwencyjnej, $t_{k,n}^*$ – ostatnia tranzycja k -tej procedury sekwencyjnej, w której uwzględniono pamięć.

2.2. Synteza równania schematowego sekwencyjnych algorytmów sterowania, w których realizacja kroków jest sygnalizowana warunkami czasowymi

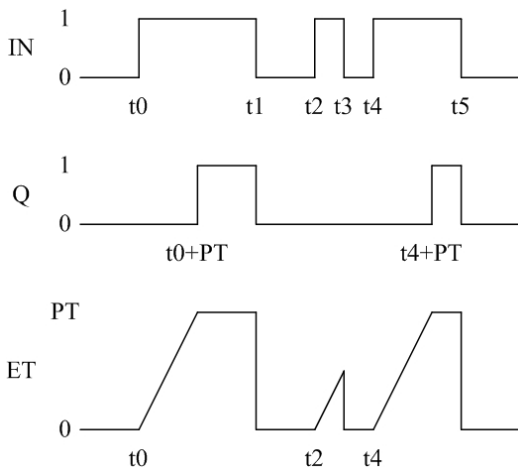
W odniesieniu do procedur w których realizację etapów elementarnych określają warunki czasowe, podczas syntezy równania schematowego obowiązują następujące zasady:

Zasada 1

W krokach czasowych stosowany jest licznik czasu typu TON (norma, 2004), którego blok oraz diagram funkcjonalny przedstawiono na Rys. 2 i 3.



Rys. 2. Schemat bloku licznika czasu TON



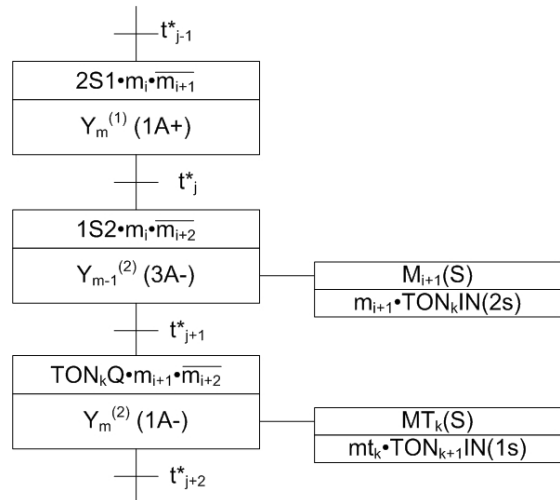
Rys. 3. Diagram funkcjonalny licznika czasu TON

Zasada 2

Warunki które reprezentuje tranzycja t_j poprzedzająca krok, w którym następuje inicjacja k -tego licznika czasu zapisują tzw. skojarzoną z licznikiem czasu komórkę pamięci inicjującą licznik czasu (Rys. 4).

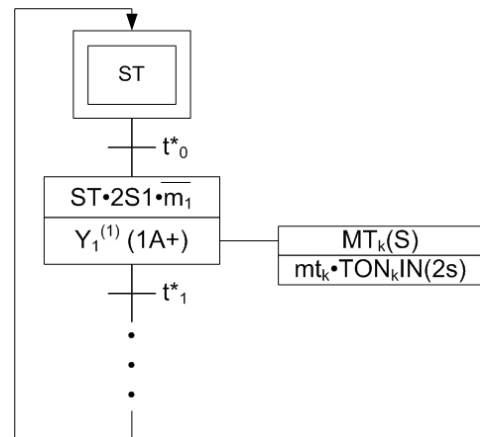
Jeśli z zasad realizacji pamięci dla procedur sekwencyjnych (Dworzak i inni, 2009) wynika, że w tranzycji t_j poprzedzającej krok, w którym następuje inicjacja licznika czasu:

- następuje zapis komórki pamięci, wówczas sygnał tej komórki pamięci inicjuje licznik,
- nie następuje zapis komórki pamięci wówczas należy ją zapisać przez warunki tej tranzycji, a jej sygnał zastosować do inicjacji licznika czasu. Taką komórkę pamięci oznaczamy MT_k , a jej sygnał mt_k .

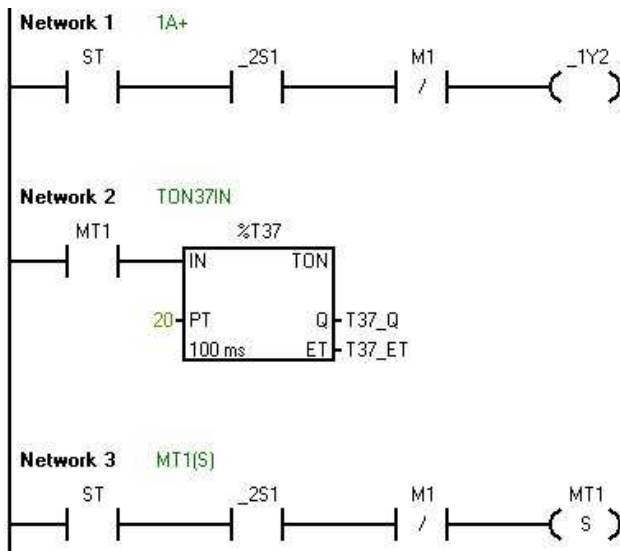


Rys. 4. Ilustracja zasady 2

UWAGA: Jeśli w pierwszym kroku algorytmu sterowania zapisywana jest komórka pamięci (MT_k) inicjująca licznik, wówczas istotne jest by zapis tej komórki pamięci w trakcie implementacji w sterowniku PLC następował po zależności, w której inicjowany jest licznik czasu, ale przed zależnością określającą zapis komórki pamięci $M_i(S)$ (Rys. 5 i 6).



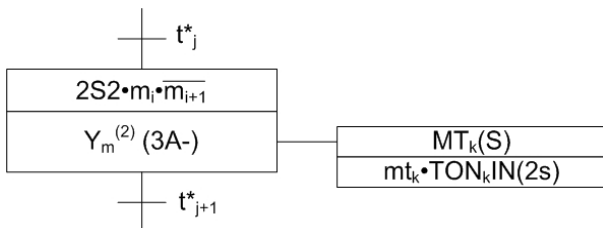
Rys. 5. Ilustracja uwagi zasady 2



Rys. 6. Ilustracja uwagi zasady 2

Zasada 3

W tranzycji występującej po kroku w których nastąpiła inicjacja k -tego licznika czasu (Rys. 7) stosujemy sygnał wyjściowy tego licznika (3).



Rys. 7. Ilustracja zasady 3

$$t_{j+1} = TON_k Q \dots \quad (3)$$

gdzie: $TON_k Q$ – sygnał wyjściowy k -tego licznika czasu.

Zasada 4

Jeśli w ostatniej tranzycji (t_n^*) występuje sygnał k -tego licznika czasu, wówczas w tranzycji t_0 należy zastosować zanegowany sygnał wyjściowy k -tego licznika czasu (4).

$$t_n = TON_k Q \dots \quad (4)$$

$$t_0 = ST \cdot \overline{TON_k Q} \dots$$

gdzie: ST – sygnał wyjściowy elementu niezależnego od procesu (np. przycisk).

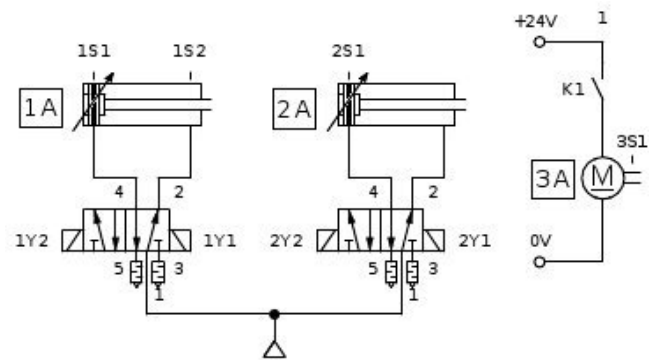
Zasada 5

Kasowanie wszystkich zastosowanych w algorytmie sterowania komórek pamięci następuje, gdy spełniony jest warunek ostatniej tranzycji (t_n^*).

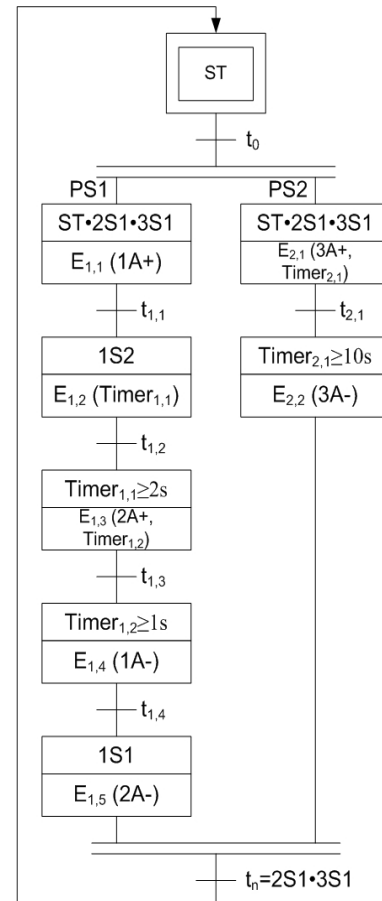
Użycie powyższych zasad zilustrowano na przykładzie.

Przykład

Na Rysunku 8 przedstawiono schemat funkcjonalny dwóch napędów pneumatycznych (1A, 2A) oraz silnika elektrycznego – 3A. Ich algorytm pracy stanowią dwie procedury sekwencyjne realizowane współbieżnie. Procedura PS1 składa się z pięciu etapów elementarnych E1.1-E1.5, natomiast procedura PS2 z dwóch etapów elementarnych E2.1-E2.2.



Rys. 8. Schemat funkcjonalny napędów



Rys. 9. Algorytm przykładowego procesu zapisany za pomocą sieci Grafpol GP

Procedura PS1

ETAP E1.1: *wysuw tłoczyska siłownika 1A*

Realizacja: 1A+ (1Y2)

Sygnalizacja: 1S2

ETAP E1.2: *oczekiwanie*

Realizacja: Timer_{1,1}

Sygnalizacja: TON_{1,1}Q

ETAP E1.3: *wysuw tłoczyska siłownika 2A po zadanym

czasie 2s przez czas 1s*

Realizacja: 2A+ (2Y2), Timer_{1,2}

Sygnalizacja: TON_{1,2}Q

ETAP E1.4:*wsuw tłoczyska siłownika 1A*

Realizacja: 1A- (1Y1)

Sygnalizacja: 1S1

ETAP E1.5:*wsuw tłoczyska siłownika 2A*

Realizacja: 2A- (2Y1)

Sygnalizacja: 2S1

Procedura PS2

ETAP E2.1:*praca silnika 3A przez zadany czas*

Realizacja: 3A+ (K1), Timer_{2,1}

Sygnalizacja: TON_{2,1}Q

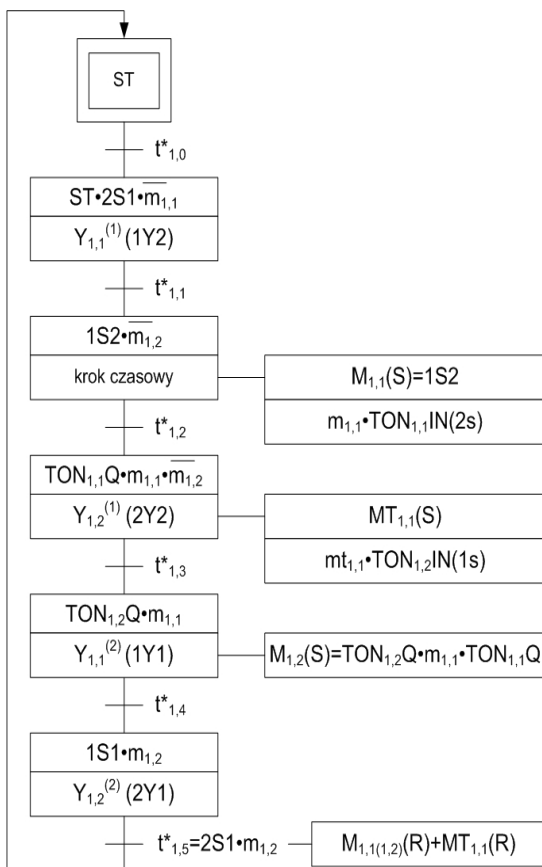
ETAP E2.2:*zatrzymanie silnika 3A*

Realizacja: 3A- ($\overline{K1}$)

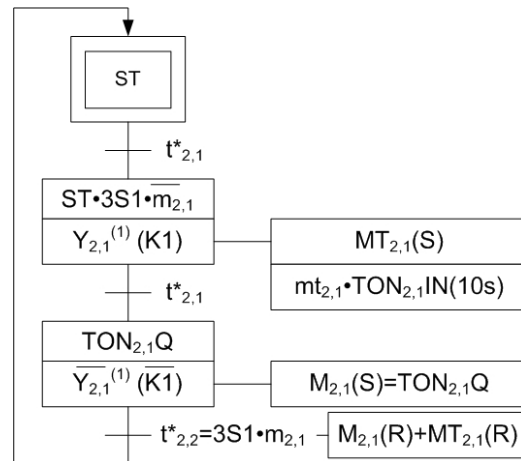
Sygnalizacja: 3S1

Procedury te stanowią reprezentację algorytmu procesu przedstawionego w postaci sieci Grafpol GP na Rys. 9.

Postępując zgodnie z zasadą 1 i 2 syntezy równania schematowego współbieżnych algorytmów sterowania, zasadami syntezy równania schematowego sekwencyjnych algorytmów sterowania w których realizacja kroków jest sygnalizowana warunkami czasowymi oraz zasadami realizacji pamięci (Dworzaki inni, 2009), otrzymujemy sieć Grafpol GS przedstawioną na Rys. 10 i 11. Na jej podstawie możliwe jest wyznaczenie równań schematowych dla PS1 (5) i PS2 (6).



Rys. 10. Algorytmy sterowania PS1 zapisany w postaci sieci Grafpol GS



Rys. 11. Algorytmy sterowania PS2 zapisany w postaci sieci Grafpol GS

$$F_{PS1}(Y,M) = \sum \left\{ \begin{array}{l} ST \cdot 2S1 \cdot \overline{m_{1,1}} \cdot Y_{1,1}^{(1)}, \\ TON_{1,1}Q \cdot m_{1,1} \cdot \overline{m_{1,2}} \cdot Y_{1,2}^{(1)}, \\ TON_{1,2}Q \cdot m_{1,1} \cdot Y_{1,1}^{(2)}, \\ 1S1 \cdot m_{1,2} \cdot Y_{1,2}^{(2)}, \\ m_{1,1} \cdot TON_{1,1}IN(2s), \\ m_{1,1} \cdot TON_{1,2}IN(1s), \end{array} \right. \quad (5)$$

$$F_{PS2}(Y,M) = \sum \left\{ \begin{array}{l} TON_{1,1}Q \cdot m_{1,1} \cdot \overline{m_{1,2}} \cdot MT_{1,1}(S) \\ 1S2 \cdot M_{1,1}(S) \\ TON_{1,2}Q \cdot m_{1,1} \cdot TON_{1,1}Q \cdot M_{1,2}(S) \\ 2S1 \cdot m_{1,2} \cdot [M_{1,1(1,2)}(R) + MT_{1,1}(R)] \\ ST \cdot 3S1 \cdot \overline{m_{2,1}} \cdot Y_{2,1}^{(1)}(S), \\ TON_{2,1}Q \cdot Y_{2,1}^{(1)}(R), \\ mt_{2,1} \cdot TON_{2,1}IN(10s), \\ St \cdot 3S1 \cdot \overline{m_{2,1}} \cdot MT_{2,1}(S) \\ TON_{2,1}Q \cdot M_{2,1}(S) \\ 3S1 \cdot m_{2,1} \cdot [M_{2,1}(R) + MT_{2,1}(R)] \end{array} \right. \quad (6)$$

Równania te poddajemy modyfikacjom, zgodnie z zasadą 3 i 4 syntezy równania schematowego współbieżnych algorytmów sterowania, otrzymując w rezultacie równanie schematowe (7) opisujące układ zależności sterujących urządzeniem zgodnie z założonym algorytmem procesu. Równanie to stanowi podstawę do implementacji w sterowniku za pomocą dowolnego języka programowania opisanego w normie PN-EN 61131-3.

LITERATURA

1. **Mikulczyński T.** (2006), *Automatyzacja procesów produkcyjnych*, WNT, Warszawa.
2. **Dworzak Ł., Ciskowski S., Mikulczyński T., Bogdan M.** (2009), Synteza metodą Grafpol sekwencyjnych algorytmów sterowania, *Pomiary, Automatyka, Robotyka*. 2009, R. 13, nr 2, s. 686-694.
3. Norma PN-EN 61131-3 - *Sterowniki programowalne - Część 3: Języki programowania*
4. **Mikulczyński T., Samsonowicz Z.** (1997), *Automatyzacja dyskretnych procesów produkcyjnych: metody modelowania procesów dyskretnych i programowania PLC*, WNT, Warszawa.
5. **Banaszak Z., Drzazga A., Kuś J.** (1993), *Metody interakcyjnego modelowania i programowania procesów dyskretnych*, Wydawnictwo Politechniki Wrocławskiej, Wrocław.
6. **Kasprzyk J.** (2007), *Programowanie sterowników przemysłowych*, WNT, Warszawa.

$$F_{PW}(Y, M) = \sum \left\{ \begin{array}{l} ST \cdot 2S1 \cdot \overline{m_{1,1}} \cdot 3S1 \cdot \overline{m_{2,1}} \cdot [Y_{1,1}^{(1)} + Y_{2,1}^{(1)}(S)] \\ TON_{1,1} \cdot Q \cdot m_{1,1} \cdot \overline{m_{1,2}} \cdot Y_{1,2}^{(1)} \\ TON_{1,2} \cdot Q \cdot m_{1,1} \cdot Y_{1,1}^{(2)} \\ 1S1 \cdot m_{1,2} \cdot Y_{1,2}^{(2)} \\ TON_{2,1} \cdot Q \cdot Y_{2,1}^{(1)}(R) \\ \\ m_{1,1} \cdot TON_{1,1} \cdot IN(2s) \\ m_{1,1} \cdot TON_{1,2} \cdot IN(1s) \\ m_{2,1} \cdot TON_{2,1} \cdot IN(10s) \\ \\ TON_{1,1} \cdot Q \cdot m_{1,1} \cdot \overline{m_{1,2}} \cdot MT_{1,1}(S) \\ St \cdot 3S1 \cdot \overline{m_{2,1}} \cdot MT_{2,1}(S) \\ 1S2 \cdot M_{1,1}(S) \\ TON_{1,2} \cdot Q \cdot m_{1,1} \cdot TON_{1,1} \cdot Q \cdot M_{1,2}(S) \\ TON_{2,1} \cdot Q \cdot M_{2,1}(S) \\ 2S1 \cdot m_{1,2} \cdot 3S1 \cdot m_{2,1} \cdot [M_{1,1(1,2)(2,1)}(R) + MT_{1,1(2,1)}(R)] \end{array} \right. \quad (7)$$

3. PODSUMOWANIE

W pracy przedstawiono sposób szybkiego i łatwego wyznaczania metodą Grafpol postaci równania schematowego procedur współbieżnych w których realizacja kroków sygnalizowana jest warunkami logicznymi i czasowymi. Dzięki opracowanym zasadom czas syntezy równania schematowego dla tego typu procedur uległ znacznemu skróceniu w porównaniu z metodą MTS (Mikulczyński i inni, 1997). Do zalet metody zaliczyć należy także konieczność użycia mniejsze ilości komórek pamięci w porównaniu z metodą Grafset (Banaszak i inni, 1993), czy SFC (Kasprzyk, 2007).

SYNTHESIS GRAFPOL METHOD
OF CONCURRENT PROCEDURES
IN WHICH STEPS EXECUTION
IS INDICATED BY LOGIC AND TIME CONDITIONS

Abstract: The paper presents a method for the synthesis of concurrent control algorithms in which control steps execution is indicated by logic and time terms. The developed principles simplify the network transformation method (MTS) of programming production processes in the scope of the memory realisation. Thanks to this, time for synthesising the schematic equation can be significantly reduced in comparison to the MTS method. The designed schematic equation makes a ground for writing an application program of a PLC using any language defined in standard PN- EN 61131-3:2004-Programmable controllers-Part 3: Programming languages.