

Laboratorium Teorii Automatów	
Budowa automatu z użyciem sterownika PLC	
Grupa 4b (wtorek 17.15)	Sonia Wittek, Katarzyna Wątorska, Bartłomiej Mróz

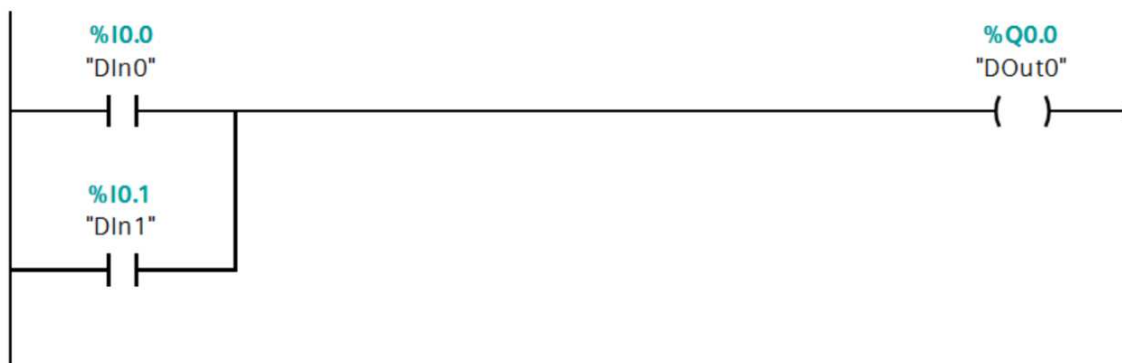
Wstęp teoretyczny

Celem laboratorium było zapoznanie się z technologią PLC oraz językiem programowania LAD oraz nabycie umiejętności programowania prostych automatów uruchamianych na sterownikach PLC. Podczas laboratorium używaliśmy sterownika PLC SIEMENS S7-1200 oraz środowiska Siemens TIA Portal.

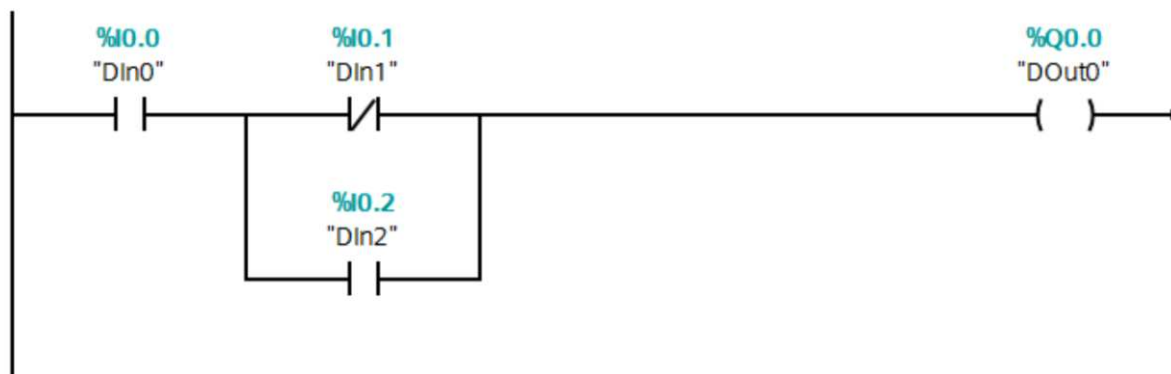
Dodatkowo rozpoznaliśmy jakie funkcje realizują programy zawarte w konspekcie.



Przedstawiono tu funkcję $DOut0 = DIn0 * DIn1$.



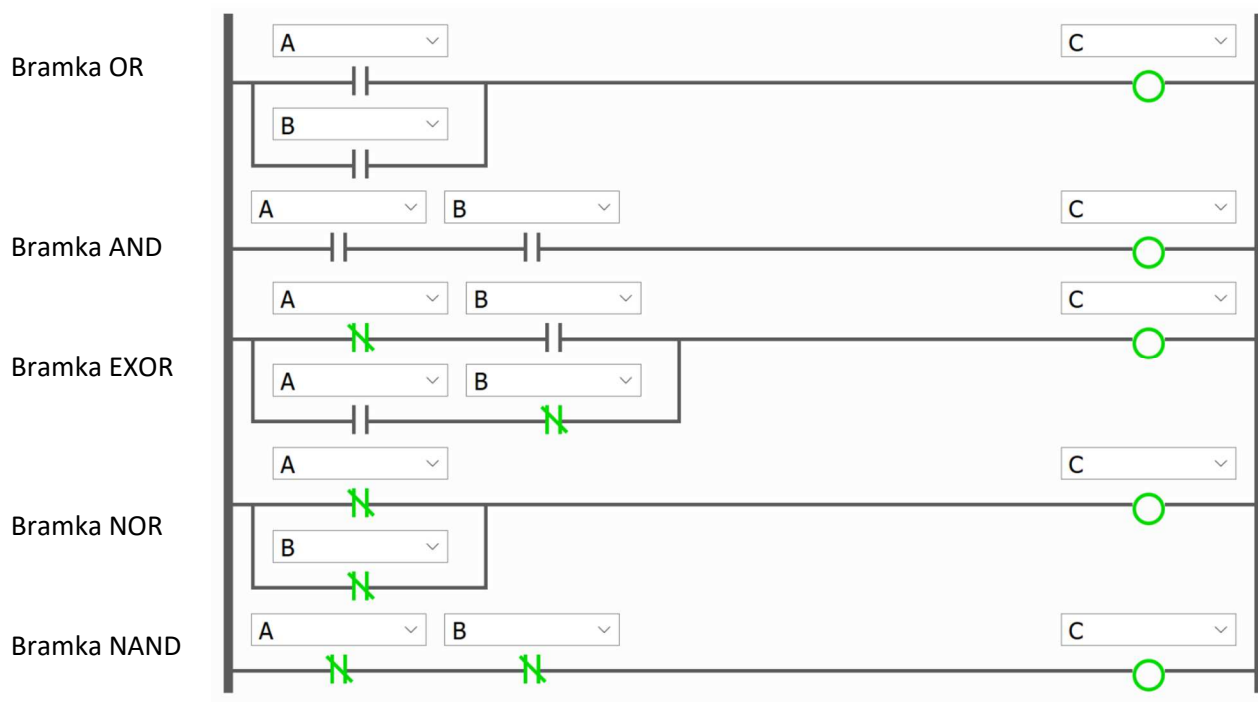
Przedstawiono tu funkcję $DOut0 = DIn0 + DIn1$.



Przedstawiono tu funkcję $DOut0 = DIn0 * (DIn1 + DIn2)$.

Odpowiedzi na pytania zawarte w konspekcie:

- Jakie jest znaczenie symboli styku normalnie otwartego i normalnie zamkniętego oraz cewki dla realizacji układów logicznych?
 - Styk normalnie otwarty realizuje funkcję AND dla wartości lewej strony przewodu i wartości przypisanej do niego zmiennej (tj. „przewodzi” jeśli jest zasilany i wartość zmiennej to 1).
 - Styk normalnie zamknięty realizuje funkcję NAND dla takich samych wartości (tj. „przewodzi” jeśli jest zasilany i wartość zmiennej to 0).
 - Cewka normalnie otwarta ustala wartość przypisaną do niej zmiennej na „1” logiczne jeśli dociera do niej sygnał „1”.
- W jaki sposób realizować proste funkcje logiczne typu AND, OR itp.?



- Co oznaczają symbole %I0.0÷%I0.2 i %Q0.0 przypisane do styków i cewek?
 - Symbole oznaczają adres zapisanej zmiennej oraz obszar pamięci. „I” oznacza wejście (np. położenie przełącznika), „Q” oznacza wyjście (np. zapalenie diody).

Przebieg laboratorium

Na początku zajęć musieliśmy zapoznać się z działaniem sterownika PLC firmy Siemens za pomocą projektu bazowego.

Zadanie 1&2

Pierwszym zadaniem było zrealizowanie, znanego z poprzednich zajęć przejazdu kolejowego, na sterowniku PLC. Funkcje realizujące przejazd:

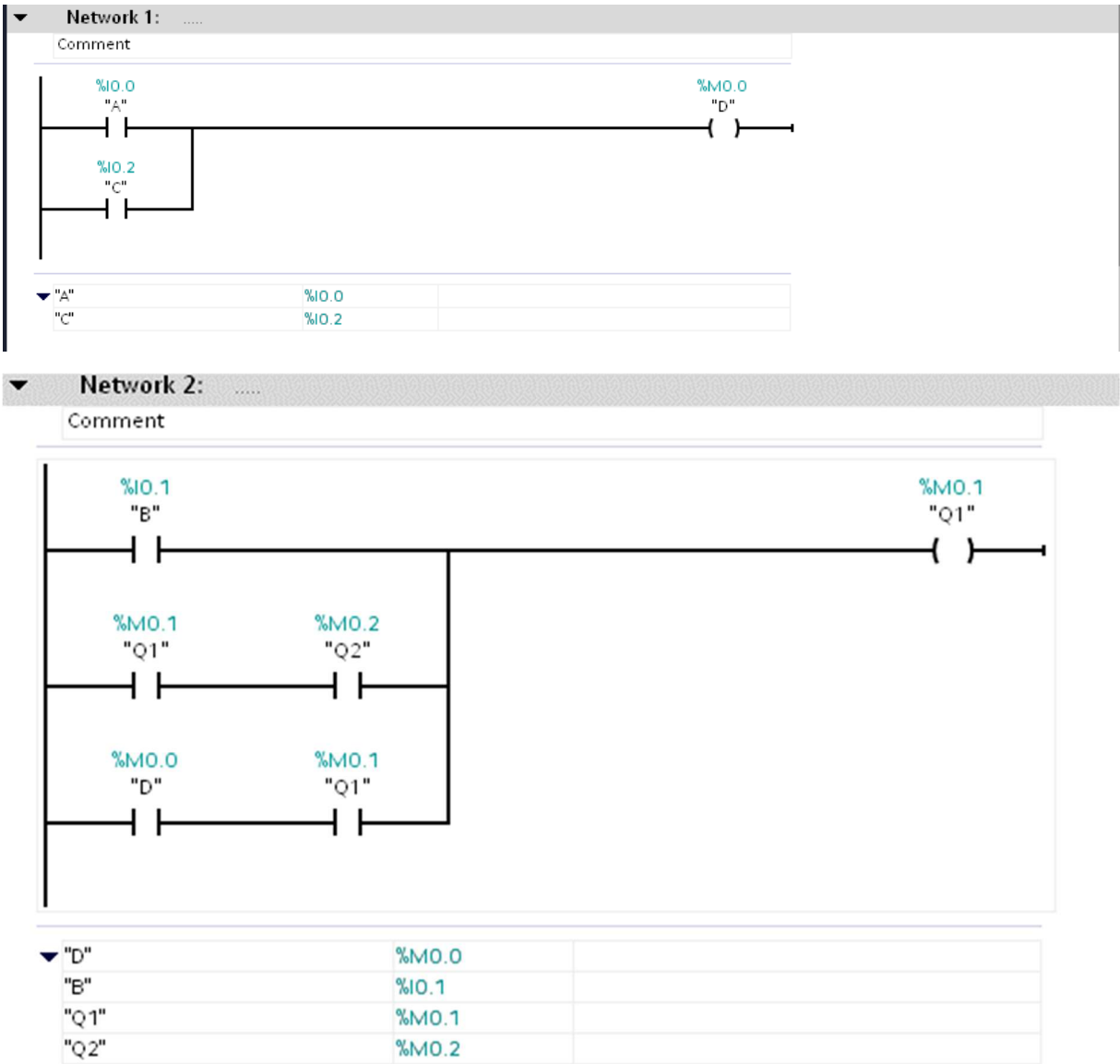
$$D = A \vee C$$

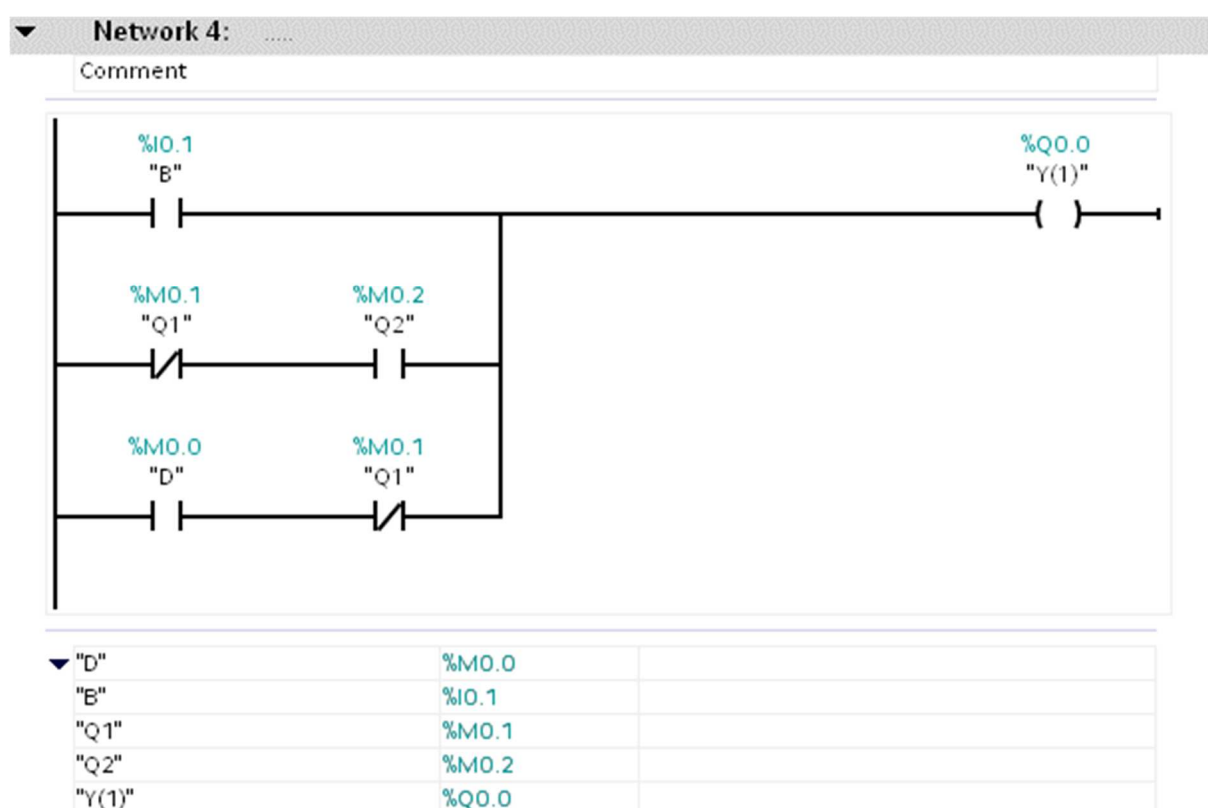
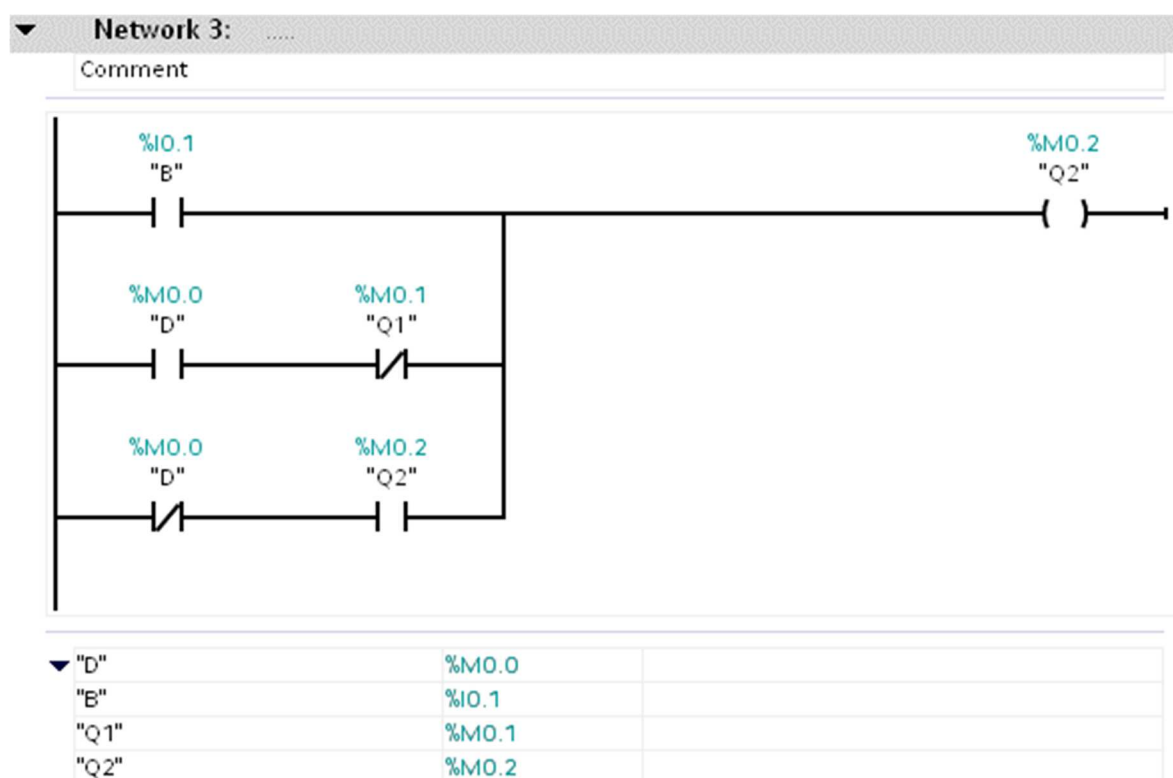
$$\dot{q}_1 = B \vee q_1 q_2 \vee D q_1$$

$$\dot{q}_2 = B \vee D \bar{q}_1 \vee \bar{D} q_2$$

$$Y = B \vee \bar{q}_1 q_2 \vee D \bar{q}_1$$

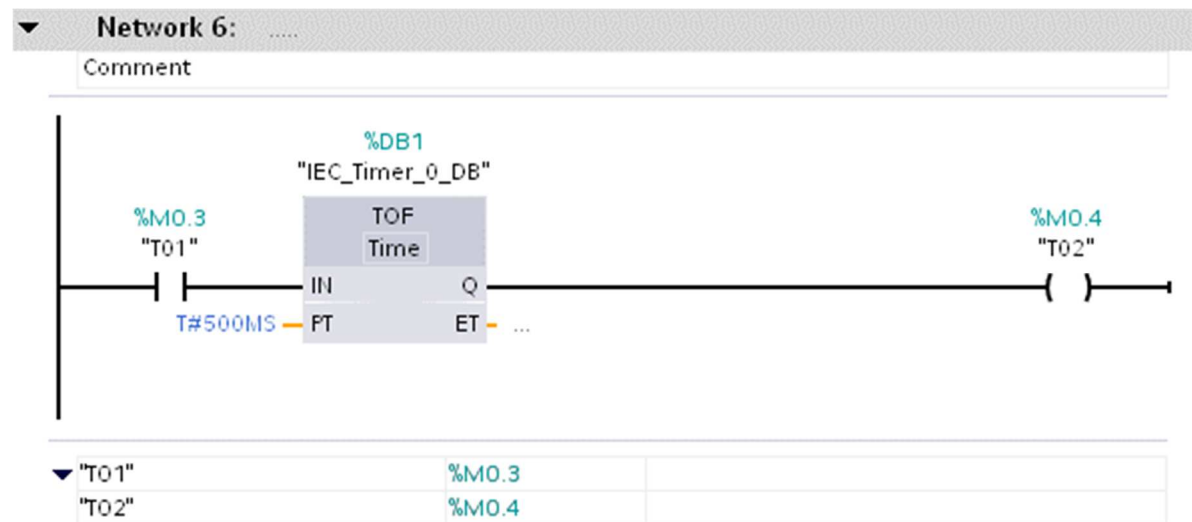
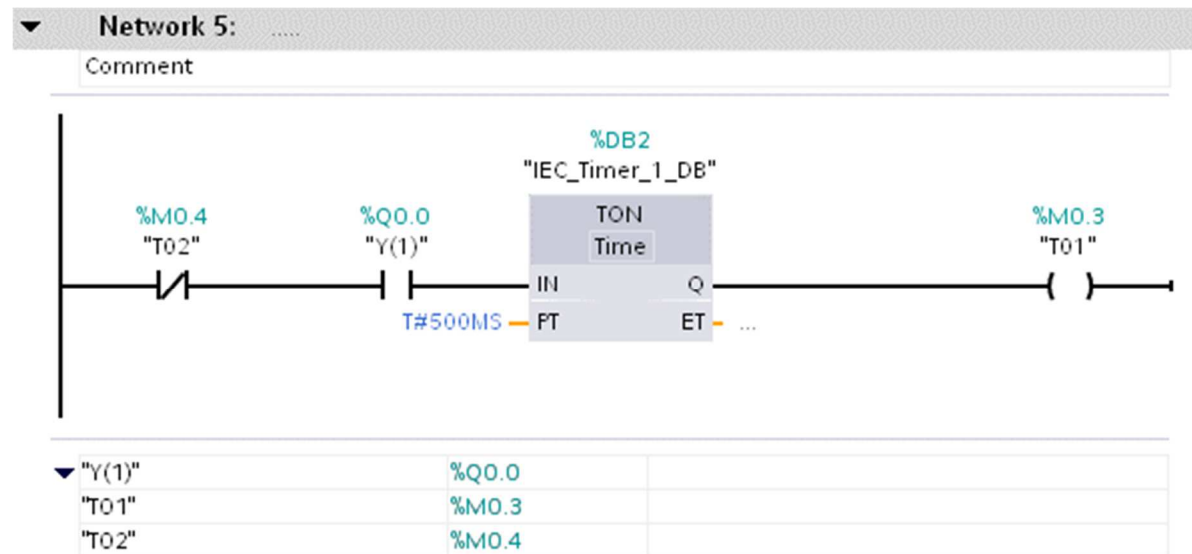
Realizacja w języku drabinkowym:

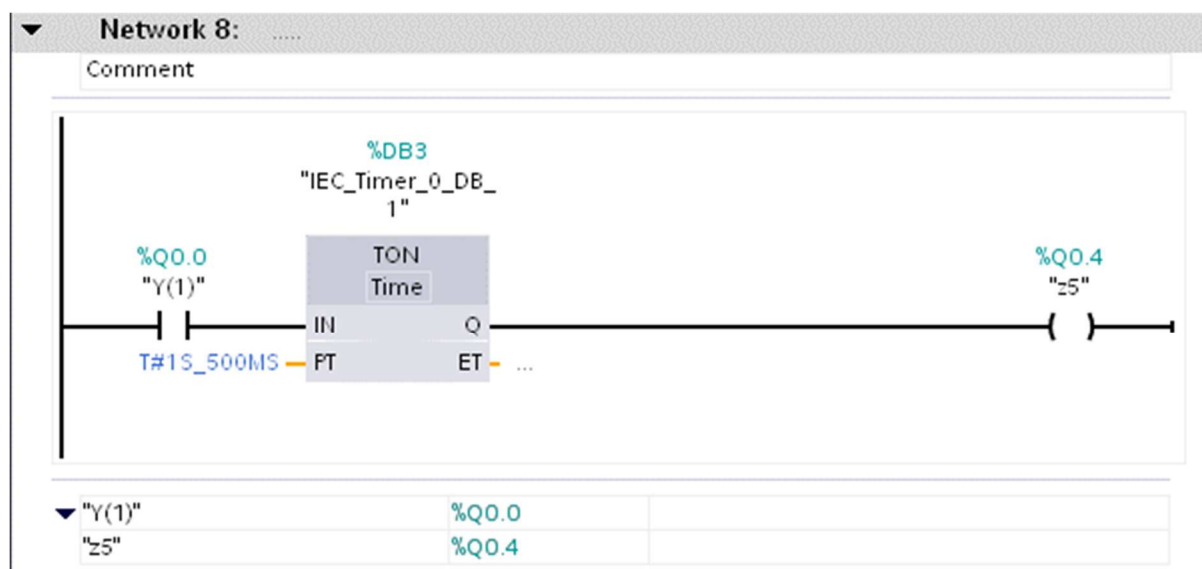




Kolejnym zadaniem było przeprojektowanie tego układu tak, aby cztery diody LED migwały z częstotliwością $f = 1\text{Hz}$ (po dwie z lewej i z prawej) oraz aby po zadany czasie zapalała się dioda oznaczająca opuszczenie zapór.

Aby to zrealizować były potrzebne timery TON oraz TOF:





Zadanie 3 – przejście dla pieszych

W tym zadaniu mieliśmy zaprojektować układ sterujący sygnalizacją świetlną na przejściu dla pieszych, według poniższego algorytmu:

1. Światła zielone dla samochodów i czerwone dla pieszych są zapalone.
2. Po wciśnięciu guzika przez pieszego oczekujemy 7s, następnie zapalają się żółte światła dla samochodów.
3. Żółte światła trwają 3s, po czym zapalają się czerwone dla samochodów i zielone dla pieszych.
4. Zielone dla pieszych trwa 5s. Pod koniec zapalają się żółte światła dla samochodów i czerwone dla pieszych.
5. Żółte światła trwają 3s, po czym zapalają się zielone światła dla samochodów.

Na podstawie tych założeń zaprojektowaliśmy tabele Carnaughy i napisaliśmy funkcje, jakie mają realizować wyjścia. Przyjęto dla nich oznaczenia:

Y_1 = zielone światło dla pieszych;

Y_2 = czerwone światło dla pieszych;

Y_3 = zielone światło dla samochodów;

Y_4 = żółte światło dla samochodów;

Y_5 = czerwone światło dla samochodów;

Projektowanie automatu:

STAN\A	0	1	Y_1	Y_2	Y_3	Y_4	Y_5
1	1	2	0	1	1	0	0
2	3	2	0	1	0	1	0
3	3	4	1	0	0	0	1
4	1	4	0	1	0	1	1

Tabela przejść po kodowaniu(1-00, 2-01, 3-11, 4-10) :

$Q_1Q_2 \backslash A$	0	1
00	00	01
01	11	01
11	11	10
10	00	10

Z tej tabeli można wyróżnić tabele przejść dla stanów Q_1 i Q_2 :

$Q_1Q_2 \backslash A$	0	1
00	0	0
01	1	0
11	1	1
10	0	1

$$\dot{Q}_1 = A Q_1 \vee \bar{A} Q_2$$

$Q_1Q_2 \backslash A$	0	1
00	0	1
01	1	1
11	1	0
10	0	0

$$\dot{Q}_2 = A \bar{Q}_1 \vee \bar{A} Q_2$$

Następnym krokiem było wyznaczenie funkcji dla poszczególnych wyjść Y:

$Q_1Q_2 \backslash A$	0	1
00	0	0
01	1	0
11	1	0
10	0	0

$$Y_1 = \bar{A} Q_2$$

$Q_1Q_2 \backslash A$	0	1
00	1	1
01	0	1
11	0	1
10	0	1

$$Y_2 = A \vee \bar{Q}_1 \bar{Q}_2$$

$Q_1Q_2 \backslash A$	0	1
00	1	0
01	0	0
11	0	0
10	1	0

$$Y_3 = \bar{A} \bar{Q}_2$$

$Q_1Q_2 \backslash A$	0	1
00	0	1
01	0	1
11	0	1
10	0	1

$$Y_4 = A$$

$Q_1 Q_2 \backslash A$	0	1
00	0	0
01	1	0
11	1	1
10	0	1

$$Y_5 = A Q_1 \vee \bar{A} \bar{Q}_2$$

Wszystkie funkcje automatu wyglądają następująco:

$$\dot{Q}_1 = A Q_1 \vee \bar{A} Q_2$$

$$\dot{Q}_2 = A \bar{Q}_1 \vee \bar{A} Q_2$$

$$Y_1 = \bar{A} Q_2$$

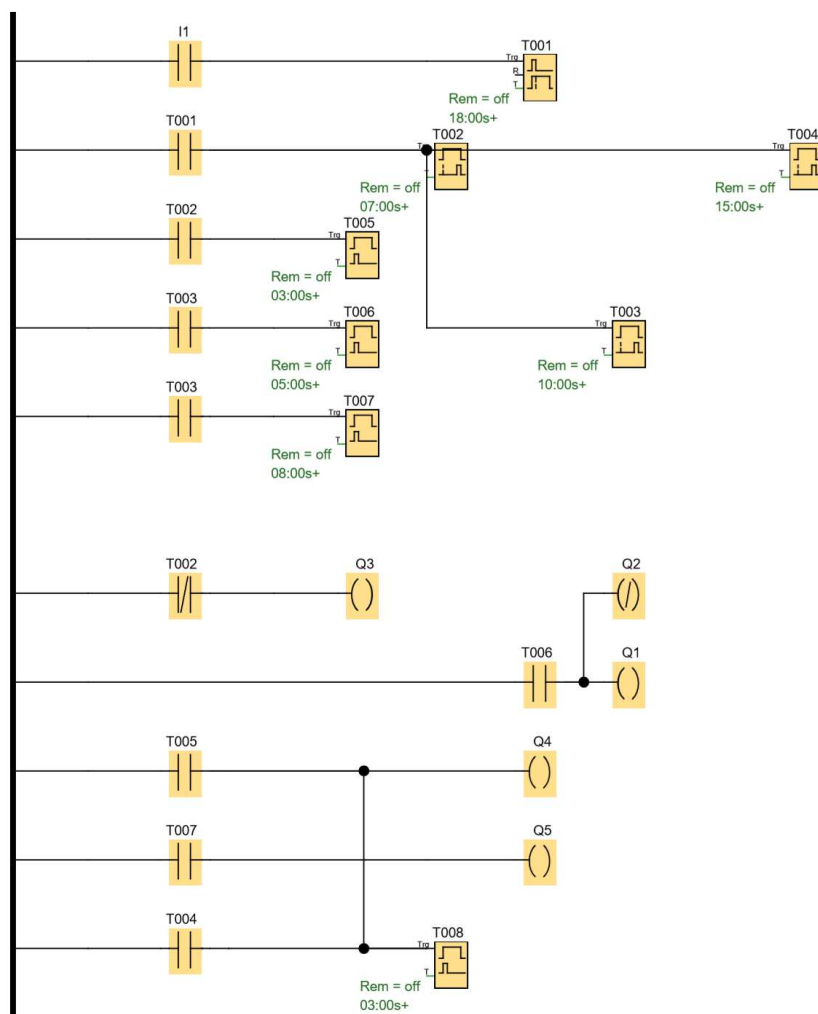
$$Y_2 = A \vee \bar{Q}_1 \bar{Q}_2$$

$$Y_3 = \bar{A} \bar{Q}_2$$

$$Y_4 = A$$

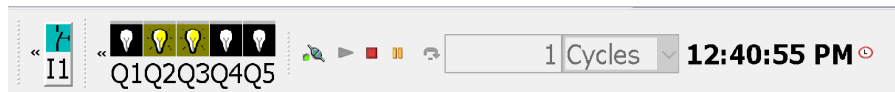
$$Y_5 = A Q_1 \vee \bar{A} \bar{Q}_2$$

Po nieudanych próbach realizacji powyższych funkcji na schematach drabinkowych, doszliśmy do wniosku, że łatwiej będzie zaprojektować ten układ nie bazując na wyprowadzonych wzorach, lecz „na logikę”. Nie zdążyliśmy zrobić tego na zajęciach, więc dokończyliśmy ćwiczenie w programie LOGO!Soft Comfort.

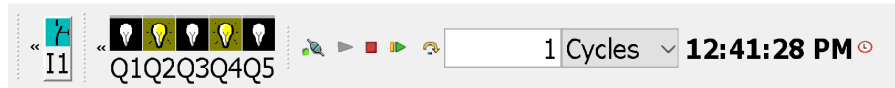


Przebieg symulacji:

- Przed wciśnięciem przycisku i po wciśnięciu przycisku przez 7 sekund



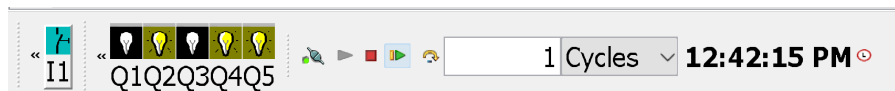
- Po 7 sekundach



- Po 10 sekundach



- Po 15 sekundach



- Po 18 sekundach cykl się kończy i sygnalizacja wraca do stanu początkowego



Podsumowanie

Podczas zajęć zapoznaliśmy się z podstawami programowania sterowników PLC oraz przeciwiczyliśmy tworzenie schematów w języku LAD. Poznaliśmy inne rozwiązanie do zadania o przejeździe kolejowym, które już wcześniej rozwiązaliśmy wykorzystując bramki NAND. Projektowanie układu sterującego światłami na przejściu dla pieszych również pokazało nam, że jeden problem może mieć wiele rozwiązań i nie zawsze to standardowe jest najlepsze i najlepsze.