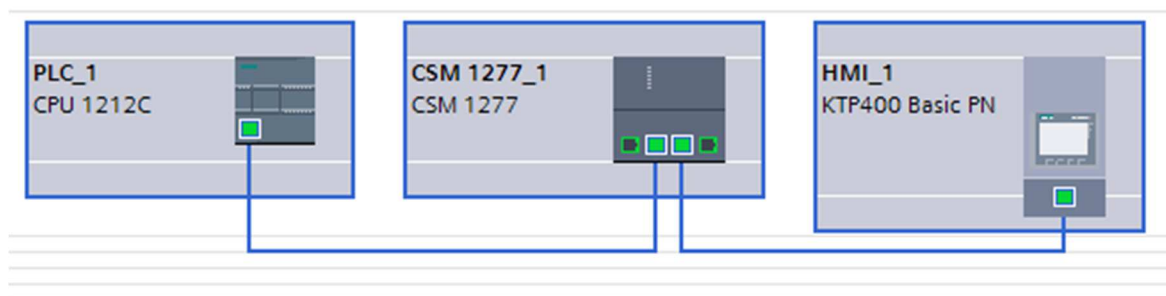


Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie Laboratorium Aparatury Automatykacji			
Numer i temat ćwiczenia: Ćwiczenie 7. SIEMENS S7 1200 – funkcje definiowane przez użytkownika – język drabinkowy (LD)			
Grupa ćwiczeniowa: Wtorek 17:00-19:15, Zespół: 3			
Lp.	Imię i nazwisko	Ocena	Podpis
1.	Katarzyna Wątorska		
2.	Sonia Wittek		
3.	Karolina Świerczek		
Data wykonania ćwiczenia: 07.05.2019			

1. Schemat i opis konfiguracji systemu:

Celem ćwiczenia było przeprowadzenie w środowisku TIA PORTAL podstawowej konfiguracji sterownika PLC SIEMENS S7 1200 z panelem operatorskim, zapoznanie się z funkcjami definiowanymi przez użytkownika, jak również językiem LD, oraz uruchomienie na panelu prostego programu sterowania logicznego, z wykorzystaniem aplikacji SCADA, realizującego sumator dwóch liczb 2-bitowych bez znaku.

Konfigurację rozpoczęliśmy od dodania do utworzonego w TIA PORTAL projektu sprzętu znajdującego się na stanowisku laboratoryjnym, tj. jednostki centralnej CPU 1212 C, switcha sieciowego CSM 1277 oraz panelu operatorskiego KTP400 Basic PN. Następnie elementy te odpowiednio, zgodnie z instrukcją skonfigurowaliśmy i połączyliśmy, czego efekt przedstawia poniższy schemat:



Rys 1. Połączone elementy sterownika PLC.

2. Przebieg ćwiczenia:

Operacja, którą miał realizować nasz program, przedstawiała się następująco: $y = a + b$, gdzie a i b to liczby wejściowe, a y to wynik ich dodania. Każda z tych liczb może być zapisana tak, jak w poniższej tabeli.

Tabela 1. Zapis w kodzie binarnym liczb używanych na panelu operatorskim.

Liczba	Zapis w kodzie binarnym
a	(x2 x1)
b	(x4 x3)
y	(y3 y2 y1)

Funkcję logiczną realizującą sumator zapisałyśmy z użyciem tabeli Karnaugh, w wyniku czego (po minimalizacji) otrzymaliśmy następujące wzory:

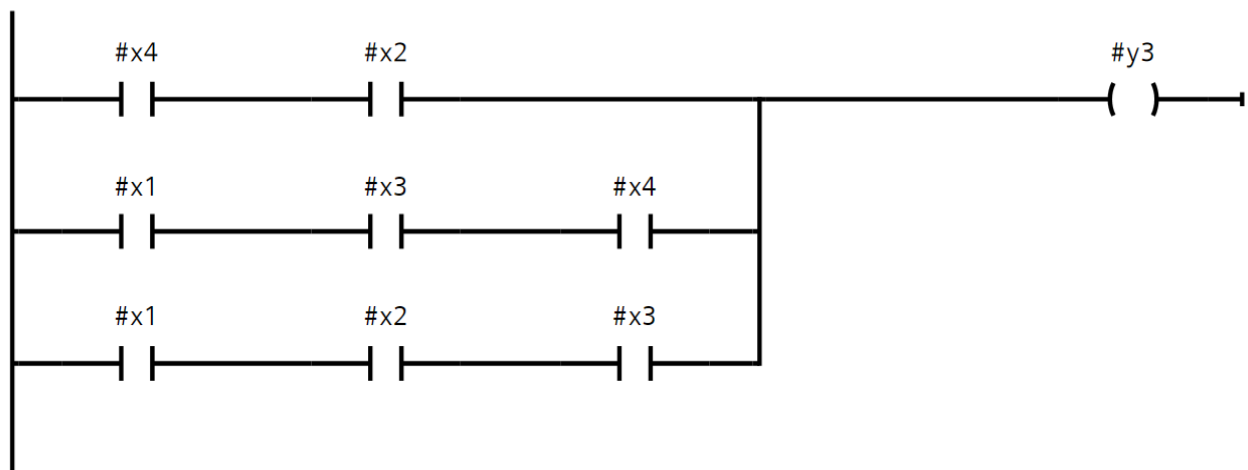
$$y_3 = x_2 \cdot x_4 + x_4 \cdot x_3 \cdot x_1 + x_2 \cdot x_1 \cdot x_3$$

$$y_2 = \overline{x_4} \cdot \overline{x_3} \cdot x_2 + x_2 \cdot \overline{x_1} \cdot \overline{x_4} + \overline{x_4} \cdot x_3 \cdot \overline{x_2} \cdot x_1 + x_4 \cdot \overline{x_3} \cdot \overline{x_2} + \overline{x_2} \cdot \overline{x_1} \cdot x_4 + x_4 \cdot x_3 \cdot$$

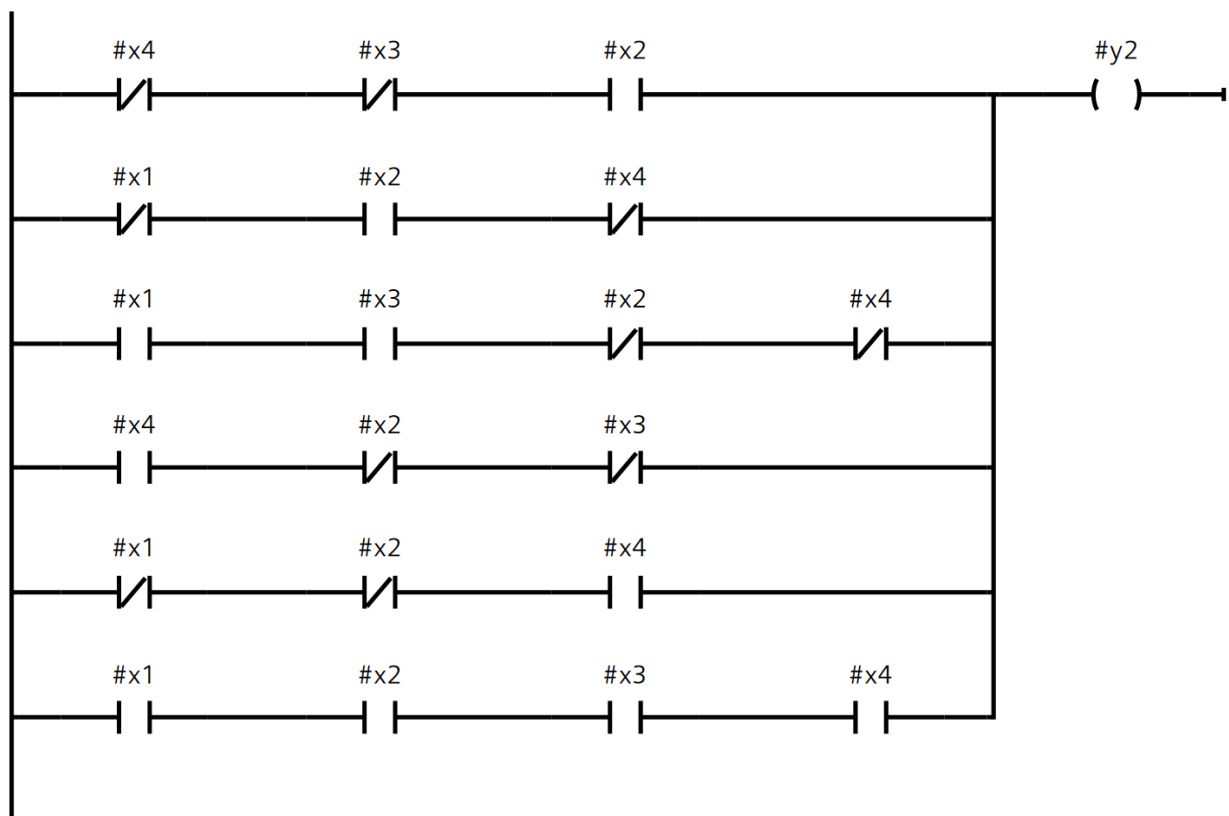
$$x_2 \cdot x_1$$

$$y_1 = x_3 \cdot \overline{x_1} + \overline{x_3} \cdot x_1$$

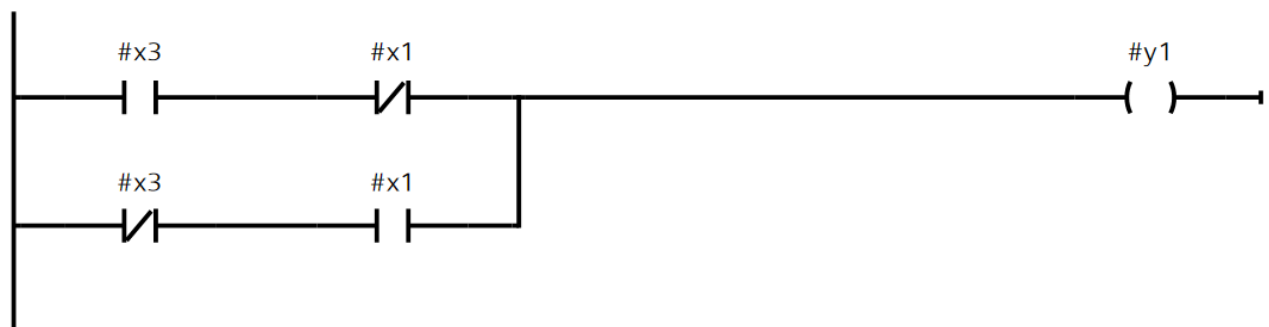
Następnie wstawiliśmy do projektu w TIA PORTAL trzy funkcje wyliczające bity wyniku y zgodnie z powyższymi wzorami: funkcję FC3 zwracającą bit y3, funkcję FC2 zwracającą bit y2 i funkcję FC1 zwracającą bit y1, oraz funkcję FC4 wywołującą pozostałe funkcje i wystawiającą wynik na wyjście. Dla wszystkich funkcji zdefiniowaliśmy zmienne wejściowe, wyjściowe i wewnętrzne. Dla funkcji FC3-FC1 zapisałyśmy algorytmy ich działania z użyciem języka drabinkowego, natomiast dla funkcji FC4 zrealizowaliśmy, również z użyciem języka drabinkowego, bezwarunkowe wywołanie pozostałych funkcji w odpowiedniej kolejności. W bloku organizacyjnym OB1 zrealizowaliśmy wywołanie funkcji FC4 i przypisałyśmy do poszczególnych wejść i wyjść tej funkcji zmienne globalne, przedstawione w tabeli 2. Efekty zaprezentowaliśmy poniżej.



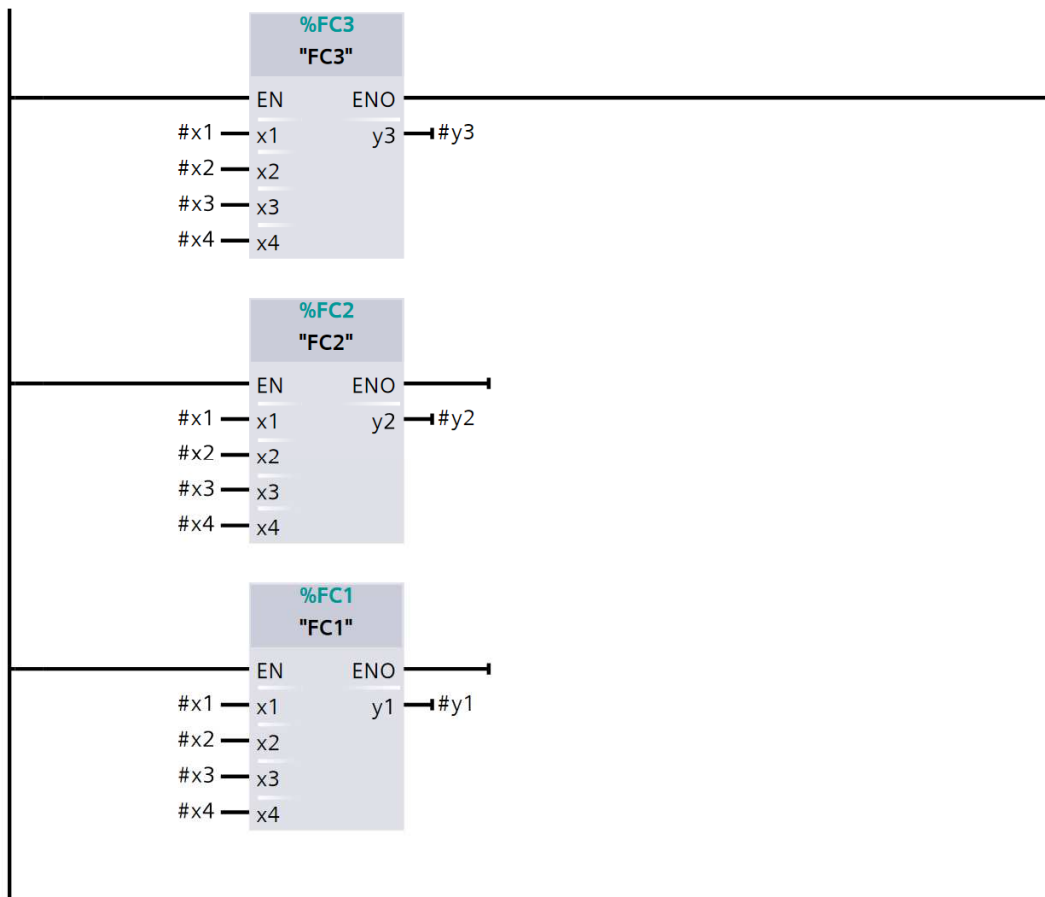
Rys 2. Funkcja FC3 w języku drabinkowym.



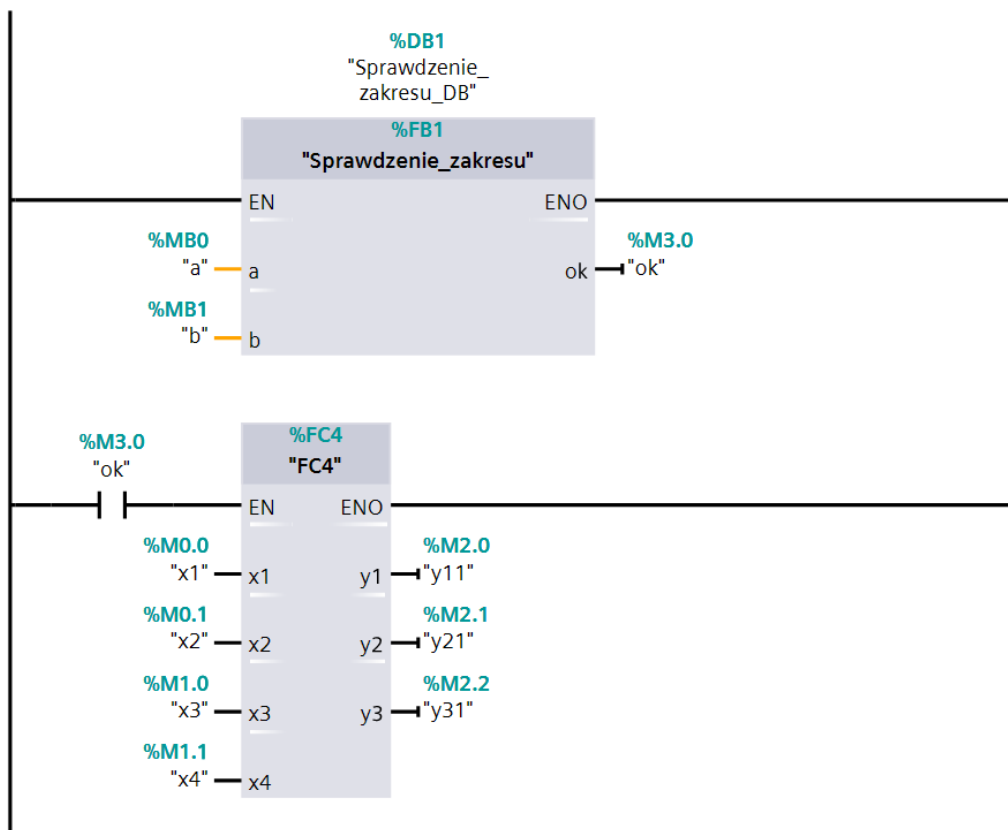
Rys 3. Funkcja FC2 w języku drabinkowym.



Rys 4. Funkcja FC1w języku drabinkowym.



Rys 5. Funkcja FC4 w języku drabinkowym.













Rys 6. Blok organizacyjny OB1.

Tabela 2. Zmienne globalne odpowiadające zmiennym użytym w funkcji FC4.

Zmienna użyta w funkcji FC4	Zmienna globalna
x4	x4
x3	x3
x2	x2
x1	x1
y3	y31
y2	y21
y1	y11

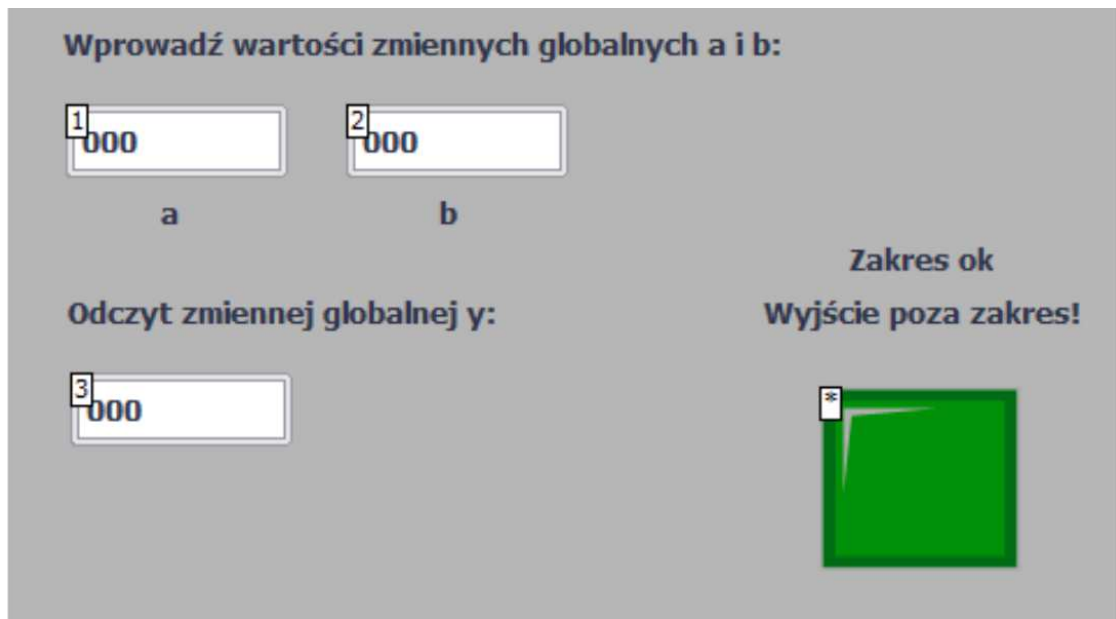
Kolejnym krokiem było zdefiniowanie zmiennych PLC, których użyliśmy do komunikacji z HMI – do wprowadzania wejść oraz odczytu wyników działania programu. Efekt tych działań jest widoczny na rysunku 7.

PLC tags						
	Name	Data type	Address	Retain	Visible in HMI	Accessible from HMI
	a	Byte	%MB0	False	True	True
	b	Byte	%MB1	False	True	True
	y	Byte	%MB2	False	True	True
	x4	Bool	%M1.1	False	True	True
	x3	Bool	%M1.0	False	True	True
	x2	Bool	%M0.1	False	True	True
	x1	Bool	%M0.0	False	True	True
	y31	Bool	%M2.2	False	True	True
	y21	Bool	%M2.1	False	True	True
	y11	Bool	%M2.0	False	True	True

Rys 7. Zmienne zdefiniowane w zakładce PLC Tags.

Następnie zdefiniowaliśmy możliwość wprowadzania wartości zmiennych globalnych a i b oraz odczytu zmiennej globalnej y na panelu operatorskim.

Po skończeniu i sprawdzeniu poprawności działania podstawowej części ćwiczenia, przystąpiliśmy do realizacji dodatkowej jego części – dla zmiennych wprowadzanych z panelu zdefiniowaliśmy sprawdzanie poprawności zakresu: poprawny zakres dla każdej zmiennej globalnej a i b wynosił od 0 do 3. W tym celu napisaliśmy odpowiedni program w języku SCL oraz zmodyfikowaliśmy wygląd panelu operatorskiego, dodając sygnalizator, informujący o poprawności wprowadzonych zmiennych poprzez świecenie na zielono.



Rys 8. Panel operatorski.

Na koniec załadowaliśmy zbudowany program na sprzęt i sprawdziliśmy działanie programu, wpisując wartości zmiennych wejściowych i odczytując wynik na panelu operatorskim.

3. Wnioski:

Ćwiczenie wykonaliśmy w całości, włączając w to część dodatkową. Nasz program działał poprawnie, jak również panel operatorski. Dzięki temu ćwiczeniu rozwinęliśmy nasze umiejętności w zakresie podstawowych zasad programowania PLC w języku drabinkowym i miałyśmy możliwość przetestowania ich na rzeczywistym sprzęcie. Ponadto, przypomnieliśmy sobie najważniejsze zasady realizacji funkcji logicznych, tabeli Karnaugh'a oraz minimalizacji. Nowym aspektem zrealizowanego ćwiczenia było napisanie prostego programu w języku SCL, który w szybki i efektywny sposób pozwolił nam na wykonanie dodatkowej części ćwiczenia.