



4xP

STL

Do pobrania z

GrzegorzCzekala.pl

V2023-1

Lista najczęściej popełnianych błędów:

1. Wielokrotne użycie tej samej cewki (!!!),
2. Nie załączony / zasilony sterownik (lub w błędzie SF),
3. Nowa funkcja nie wywołana w OB1,
4. Dalej nie działa? Przejdź do aplikacji „4xP App”.

Przydatne skróty klawiszowe: (część tylko TIA lub STEP7)

- Ctrl+R** – nowy network **Ctrl+L** – prześlij na CPU
Ctrl + Tab – przełączanie kart **Ctrl + I** - Zdalna stacyjka CPU
Ctrl + J - Wstaw symbol w STL **Ctrl + D** - Moduł Information
Ctrl + Q - Wyświetl symbole / adresy **Ctrl + 1 / 2 / 3** - zmiana LAD / STL / FBD

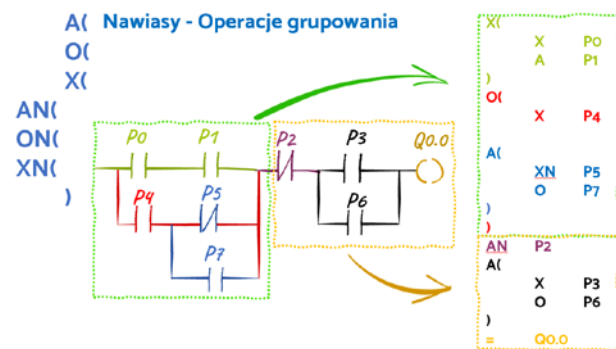
Domyślny Clock Memory (po aktywacji w HW Config)

Mo.7	Mo.6	Mo.5	Mo.4	Mo.3	Mo.2	Mo.1	Mo.0
0.5Hz	0.625Hz	1Hz	1.25Hz	2Hz	2.5Hz	5Hz	10Hz

Podstawowe instrukcje

- = Przypisanie stanu (cewka)
- A Iloczyn logiczny
- O Suma logiczna
- X XOR (nieparzystość)
- ...N Negacja pojedynczego bitu (np. AN "ON", XN I1.1)

Grupowanie



...(Rozkaz otwierający grupowanie np. A(AN(X(
) Rozkaz zamykający grupowanie

Przerzutniki

- R Warunkowe kasowanie bitu
- S Warunkowe setowanie bitu

Operacje na RLO i wywołania funkcji

- BEC Warunkowe zakończenie wykonywanego bloku
- BE / BEU Bezwarunkowe zakończenie wykonywanego bloku

Operacje na RLO (bezwarunkowe)

- NOT Negacja stanu RLO
- SET Ustawienie stanu RLO
- CLR Kasowanie stanu RLO
- SAVE Zapisanie RLO w bicie BR

// Warunkowe wywołanie Fc1

Wywołania funkcji

- CC Warunkowe wywołanie funkcji
- UC Bezwarunkowe wywołanie funkcji np. UC FC 1
- CALL Bezwarunkowe wywołanie funkcji z parametrami

X „ON”
 CC Fc1

Zamieszczone grafiki są mojego autorstwa - Grzegorza Czekają. Wykorzystuję je na swoich szkoleniach i w podręcznikach. W tym miejscu poprzez Fundację CALM edu udostępniam je Tobie za darmo ponieważ wierzę, że wartościowe treści edukacyjne należy promować i rozpowszechniać, tak aby jak najwięcej osób mogło się dzięki nim rozwijać dla naszego wspólnego dobra. Też tak uważasz? Chcesz podzielić się swoimi materiałami? Chcesz się przyłączyć do Fundacji? Daj nam znać.

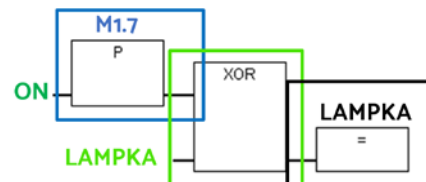


Chcesz poznać nowoczesne podejście do edukacji? Chcesz coś zmienić? Odwiedź stronę Fundacji CALM edu i odkryj nowy wymiar nauki. „Edukacja 4.0” i „Szkoła 4.0”.

Detekcja zbocza umożliwia zmianę sygnału trwającego wiele cykli CPU na pojedyncze impulsy. **Zbocze narastające (FP)** to zmiana stanu niskiego na wysoki (zwykle wciśnięcie przycisku). **Zbocze opadające (FN)** to zmiana z stanu wysokiego na niski (puszczenie przycisku, zanik sygnału).

// Opadające
 X „ON”
 FN M6.1
 S Q0.0

Przykład detekcji zbocza



X ON
 FP M1.7
 X Lampka
 = Lampka

Neurodydaktyka - czyli jak się skutecznie uczyć

Chcesz nauczyć się jak się uczyć, aby się nauczyć? Poszukaj informacji na stronie fundacji na temat **neurodydaktyki** – dziedziny nauki badającej m.in. w jaki sposób nasz mózg przyswaja nowe informacje i w jaki sposób tworzyć skuteczne materiały edukacyjne (takie jak 4xP ;) Chcesz poznać szkołę przyszłości? Znajdź „Szkoła 4.0” i „Edukacja 4.0”

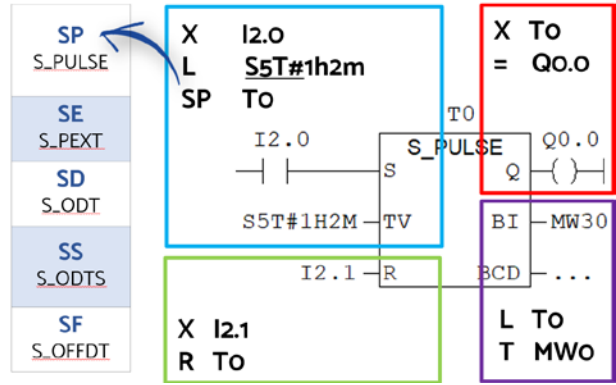
Arytmetyka, komparatory i konwersja

- TAK – zamienia miejscami zawartość ACCU 1 i ACCU2
- PUSH – kopiuje z ACCU1 do ACCU2 (w dwóch będzie to samo)
- POP – kopiuje z ACCU2 do ACCU1 (w dwóch będzie to samo)
- L ... – załadowanie do ACCU1 T ... –transfer (kopiowanie z ACCU1)
- ITD – Integer to Double, DTR – Double to REAL,
- RND, RND+, RND-, TRUNC – zaokrąglenie i ucinanie zmiennych REAL

MW 10 = MW20 + 30 = MW 100



Timery w STL



Wszystkie country oraz timery z powodu ograniczonego tutaj miejsca zostały zamieszczone na osobnej wersji 4xP „Country i Timery w TIA Portal / Step7”.

Arytmetyka

- „+” „+I” „+D” „+R” (ADD)
- „-I” „-D” „-R” (SUB)
- „*I” „*D” „*R” (MUL)
- „/I” „/D” „/R” (DIV)
- „MOD” reszta z dzielenia
- INC / DEC +1 / -1

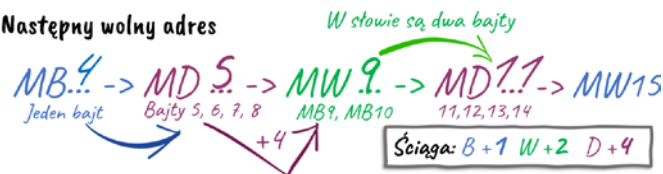
Komparatory

==I / D / R (EQ)	// Spr. większość
<I / D / R (NE)	L MD10
>I / D / R (GT)	L 45
<I / D / R (LE)	==D
>=I / D / D (GE)	= Lampka
<=I / D / R (LE)	

Znalazłeś błędy, literówki, masz pomysły jak coś zrobić lepiej? **Konieczn**ie daj nam znać w wiadomości lub komentarzu.

Podoba się? Oceń post i udostępnij go swoim znajomym.

Następny wolny adres

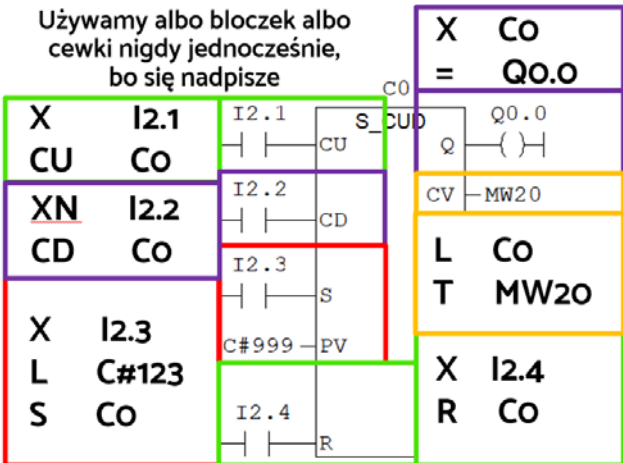


Rotacje i przesunięcia

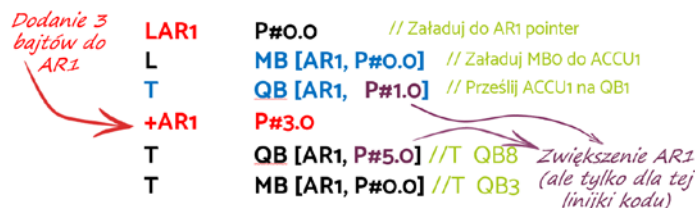
- RLD / RRD – rotacja 32 bitów w L (lewo) lub R (prawy)
- SLW / SLD – przesunięcie w lewo 16 / 32 bitów
- SRW / SRD – przesunięcie w prawo 16 / 32 bitów

Liczniki w STL

Używamy albo bloczek albo cewki nigdy jednocześnie, bo się nadpisze



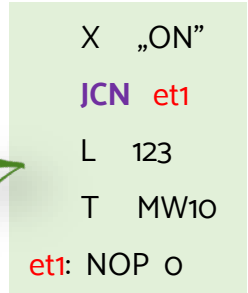
Odwwołanie do zmiennej o długości BYTE poprzez wskaźniki 32 bitowy



SKOKI podstawowe

- JU Skok bezwarunkowy
- JC Skok warunkowy RLO=1
- JCN Skok warunkowy RLO=0
- NOP o Pusty rozkaz

Gdy wcisnięty ON to ładuje 123 do MW10, a gdy nieaktywny przeskakuje do et1.



Bloki Danych w STL

(dostęp bez optymalizacji)

- OPN DB... / DI ... – otwarcie DB dla rejestrów DB1 / DB2
- Odwwołanie do zawartości DB o rozmiarach: (? – odpowiedni numer)
 - bit – DB?.DBX?.? np.: X DB10.DBX5.3
 - BYTE – DB?.DBB np.: L DB11.DBB8
 - WORD – DB?.DBW np.: L DB11.DBW2
 - DWORD – DB?.DBD (też dla REAL)

Przykłady zastosowania:

- OPN DB1 // otwarcie DB1
- L DBW2 // załaduj drugie słowo + 1 // dodaj 1
- T DBW4 // prześlij do word 4

Przekopiowanie zawartości

- OPN DB1 // otwórz DB1
- OPN DI2 // otwórz też DB2
- L DBW2 // pobierz słowo 2 z DB1
- T DIW0 // prześlij na słowo 0 z DB2

Struktura pętli LOOP



Operacje logiczne na słowach

- AW – AND na WORD, OW – OR na WORD, XOW – XOR na WORD
- INVI – inwersja WORD, INVD – inwersja DWORD
- AD – AND na DWORD, OD – OR na DWORD, XOD – XOR DWORD

Wskaźniki 32 bitowe

- LAR1 / LAR2 – kopiuje ACCU1 do AR1 / AR2
- TAR1 / TAR2 – kopiuje AR1 do ACCU1
- +AR1 / +AR2 – dodaje wartość ACCU1 do AR1 / AR2