

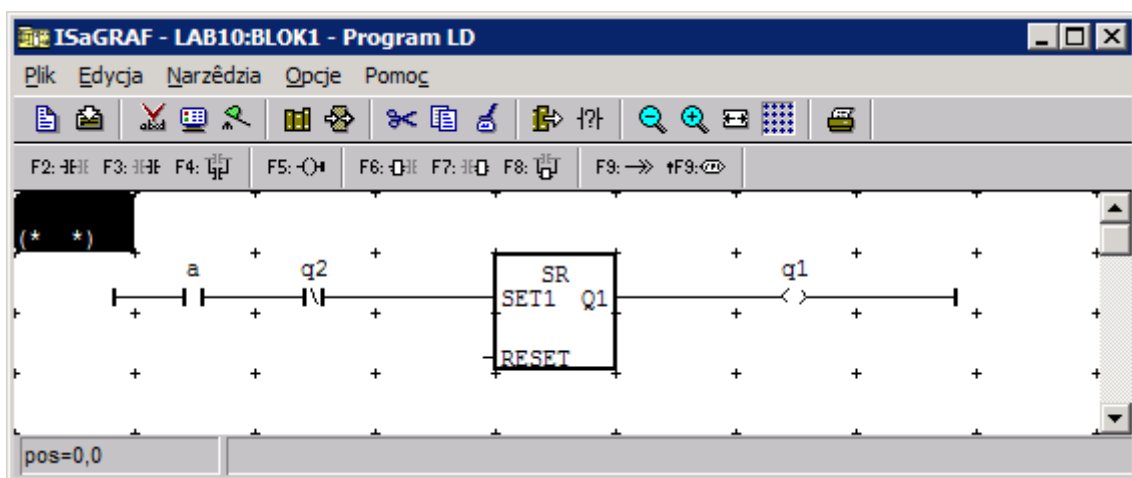
## 11. PRZYKŁAD WYKORZYSTANIA PRZERZUTNIKÓW W JĘZYKU LD

Prezentowany poniżej przykład realizuje program dla układu opisanego w punkcie 2. w Dodatku 2b. W materiale pominięte zostały czynności wykonywane podczas tworzenia nowego projektu (deklarowanie zmiennych, ustalanie konfiguracji) i przedstawiono wyłącznie sposób zapisu programu w języku LD (zapis programu w FBD nie sprawia problemu).

Należy zbudować układ o dwóch elementach pamięci, którego blok pamięci został skonstruowany przy pomocy przerzutników SR. Zgodnie z analizą przeprowadzoną w dodatku 2b wejścia poszczególnych przerzutników dane są zależnościami:

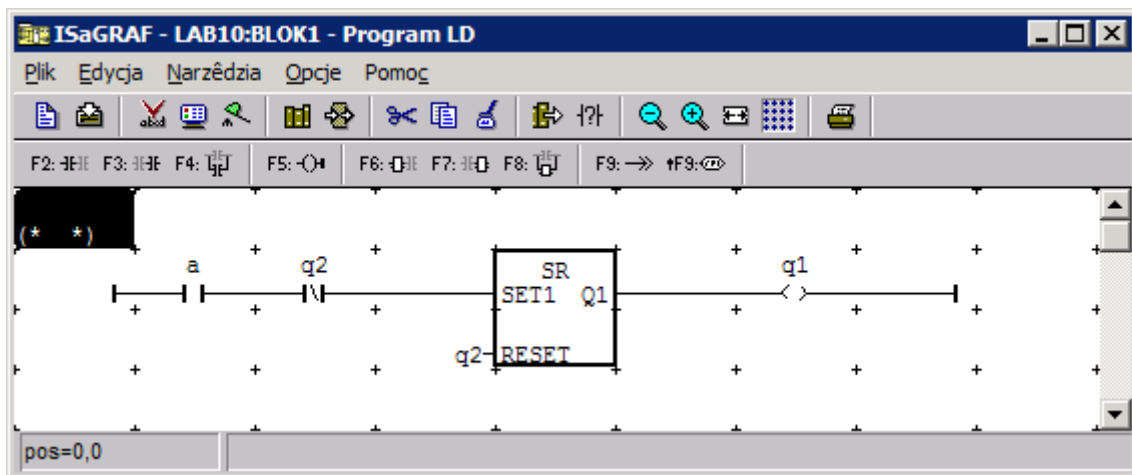
$$S_1 = a\bar{q}_2, \quad R_1 = q_2, \quad S_2 = \bar{a}\bar{b}q_1, \quad R_2 = a + b.$$

W celu realizacji programu należy utworzyć projekt i zdefiniować zmienne wejściowe  $a$  i  $b$  oraz zmienne wyjściowe  $q_1$  i  $q_2$ , a następnie skonfigurować karty wejścia i wyjścia. Realizacja pierwszego elementu pamięci wymaga podania na wejścia przerzutnika sygnałów  $S_1$  (tzn. koniunkcję  $a\bar{q}_2$ ) i  $R_1$  (w tym przypadku zmienna  $q_2$ ). Po utworzeniu programu LD należy dodać kolejno dwa połączone szeregowo styki oraz blok funkcyjny SR (F6: SR F7: SR). Po wprowadzeniu negacji i przypisaniu zmiennych program wygląda jak na rys.11.1.



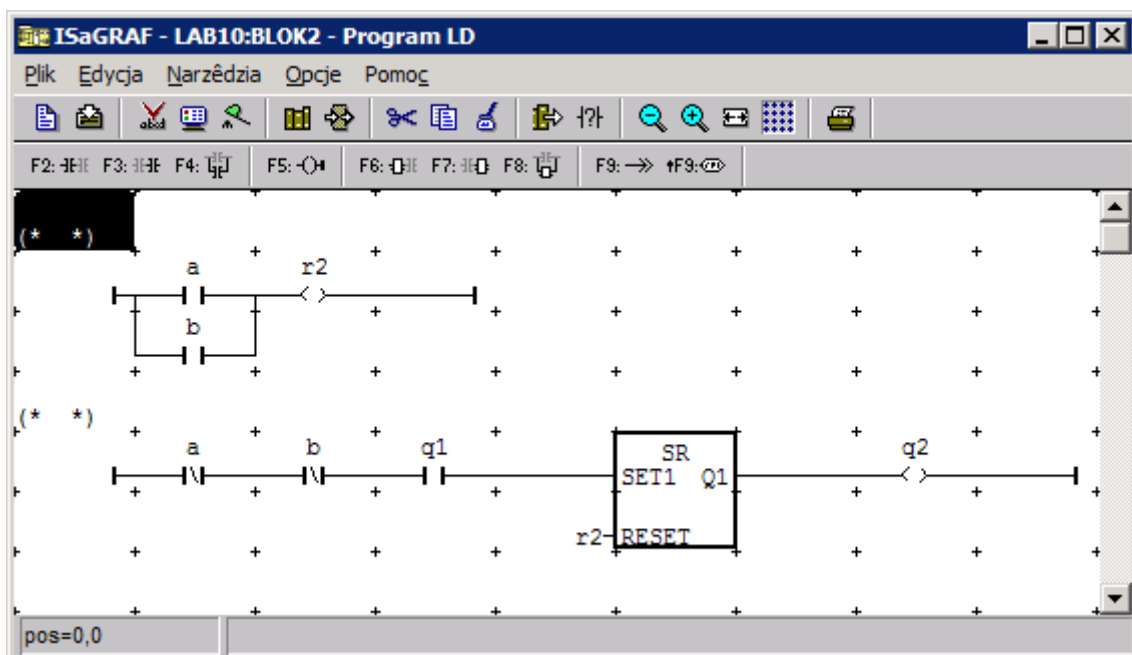
Rys.11.1. Przerzutnik z sygnałem  $S_1$

W celu podania sygnału na wejście zerujące należy dwukrotnie kliknąć na polu znajdującym się po lewej stronie wejścia opisanego jako RESET i wybrać zmienną  $q_2$ . Kompletny program realizujący pierwszy blok pamięci jest pokazany na rys. 11.2.



Rys.11.2. Pierwszy blok pamięci

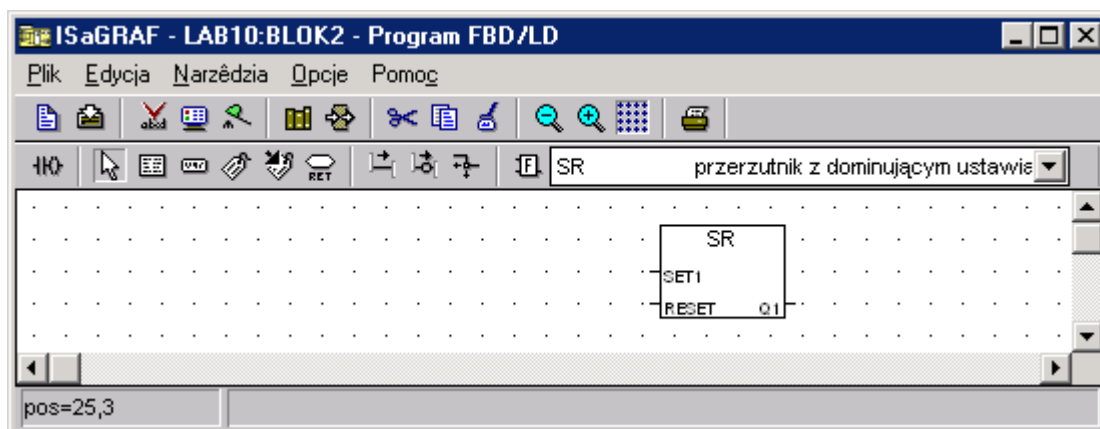
Realizacja drugiego bloku pamięci wymaga dodatkowych czynności – w edytorze LD nie można połączyć drugiego wejścia bloku funkcyjnego ze szczeblem drabinki (możliwe jest jedynie połączenie pojedynczej zmiennej jak pokazano powyżej). Układ taki można zbudować definiując pomocniczą zmienną wewnętrzną, której wartość zostanie obliczona zgodnie z wyrażeniem  $R_2$ , a następnie przypisując tą zmienną do wejścia zerującego przerzutnika. Kompletny program realizujący drugi blok pamięci został przedstawiony na rys.11.3.





Rys.11.3. Drugi blok pamięci

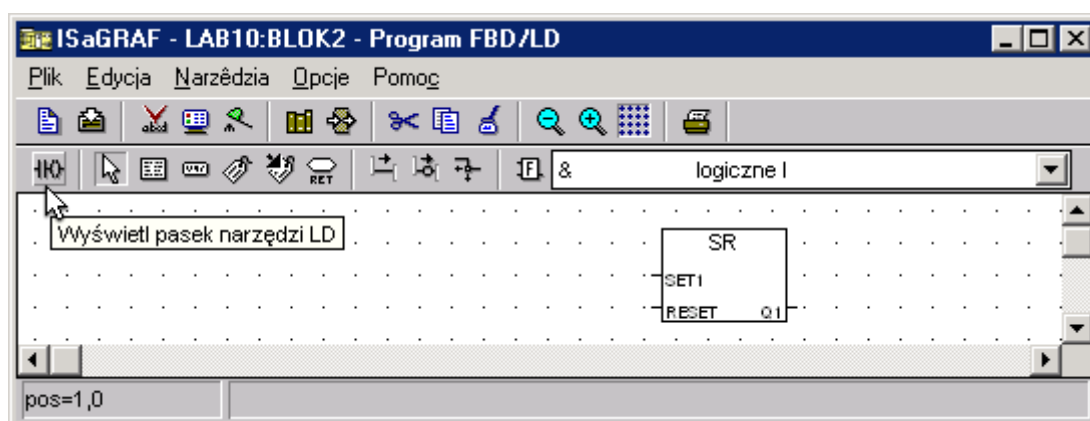
Inną metodą realizacji takiego układu w LD jest wykorzystanie edytora FBD, który pozwala na używanie elementów języka LD. W tym celu należy utworzyć nowy program w języku FBD i dodać do niego blok funkcyjny SR (rys. 11.4.).





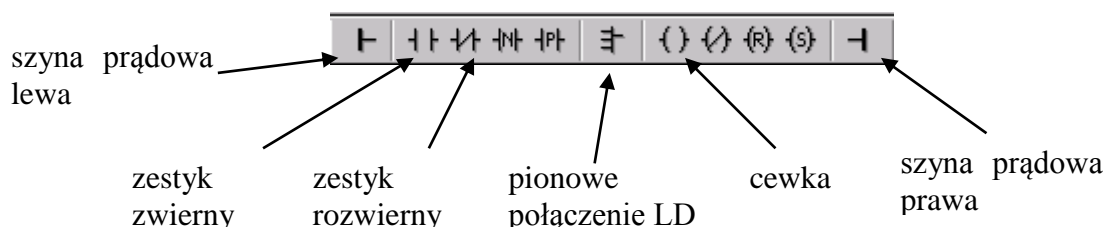
Rys.11.4. Edycja drugiego bloku pamięci

Elementy języka LD można pokazać wyświetlając pasek narzędzi LD. W tym celu należy kliknąć pierwszy przycisk znajdujący się na belce narzędziowej (). Po włączeniu paska LD przycisk ten zmienia swój wygląd () i pozwala na ponowne włączenie paska narzędzi FBD. Zmianę pasków narzędzi umożliwiają też opcje w menu: **Opcje|Pasek narzędziowy LD** i **Opcje|Pasek narzędziowy FBD**.



Rys.11.5. Włączanie paska narzędziowego LD

Po włączeniu paska LD dostępne stają się elementy, które umożliwiają skonstruowanie szczebla języka LD:

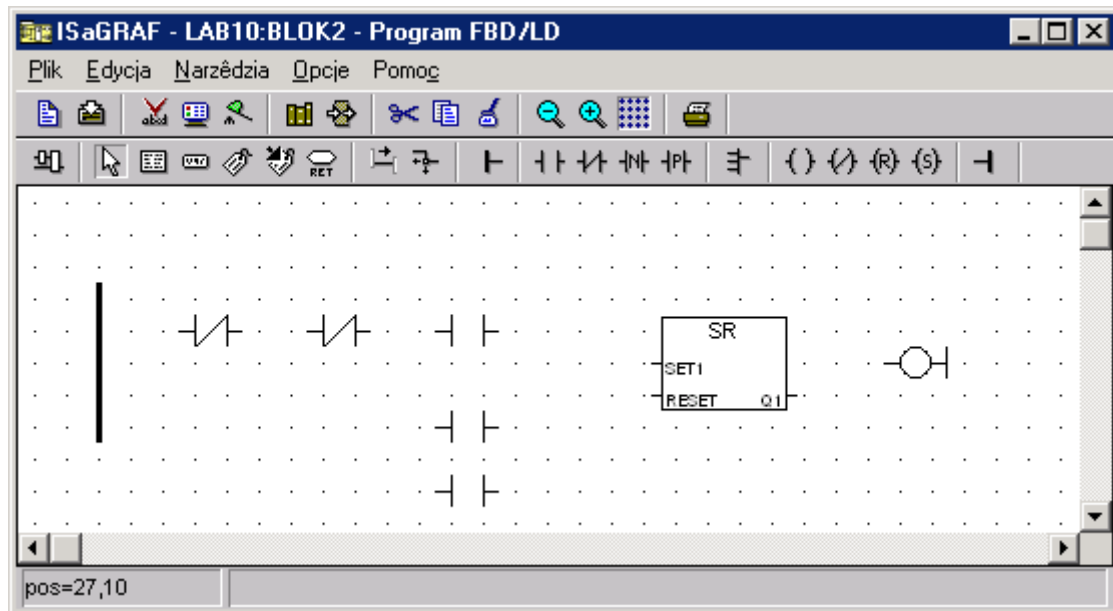


Rys.11.6. Pasek narzędziowy LD


Teraz można dodać już niezbędne elementy szczebla:

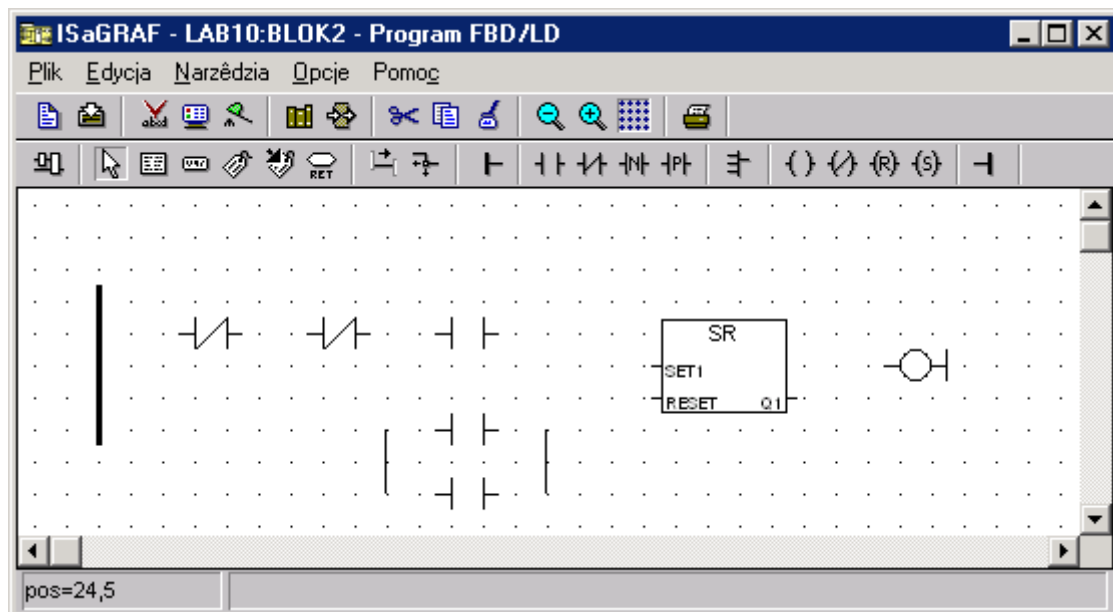


- szynę prądową lewą,
- dwa zestyki zwierne,
- trzy zestyki rozwierne,
- cewkę.




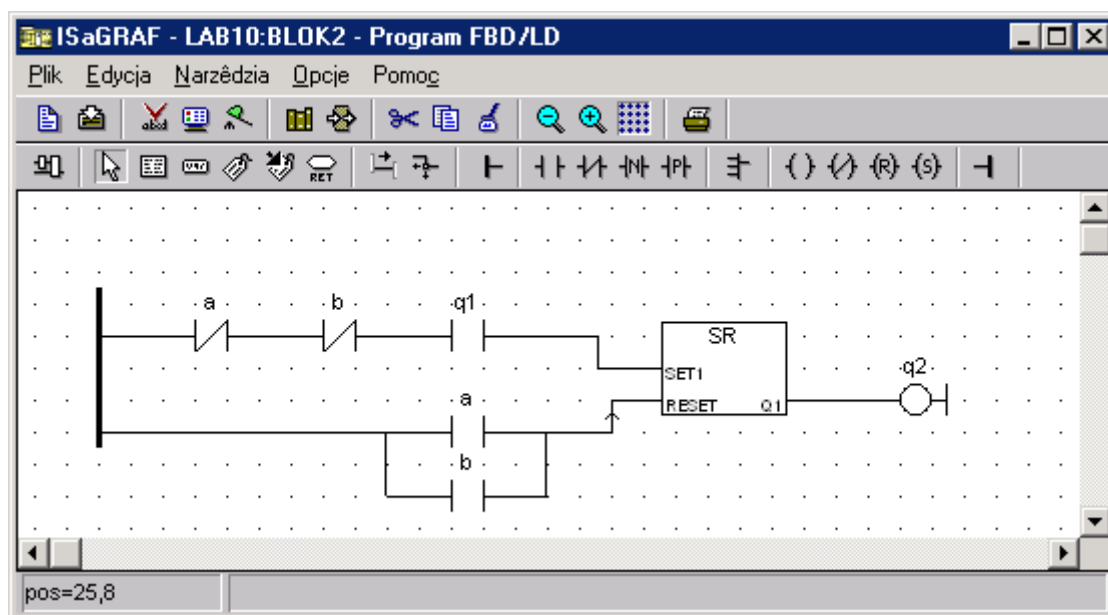
Rys.11.7. Edycja drugiego bloku pamięci cd.

Dwa dolne zestyki zwierne będą generowały sygnał zerujący dla przerzutnika, zgodnie z założeniami muszą być połączone w sposób równoległy. Do konstrukcji takiego połączenia wykorzystuje się w edytorze przycisk „pionowego połączenia LD”: . Stan edytora po dodaniu dwóch pionowych połączeń został przedstawiony na rys. 11.8.



Rys.11.8. Edycja drugiego bloku pamięci cd.

Do ukończenia programu niezbędne jest teraz połączenie wszystkich elementów szczebla (przycisk ) i przypisanie zmiennych do wszystkich zestyków oraz cewki (zmiennie przypisuje się w okienku, które jest wyświetlane po dwukrotnym kliknięciu na odpowiednim elemencie). Kompletny szczebel realizujący blok 2 został przedstawiony na rys. 11.9.



Rys.11.9. Drugi blok pamięci