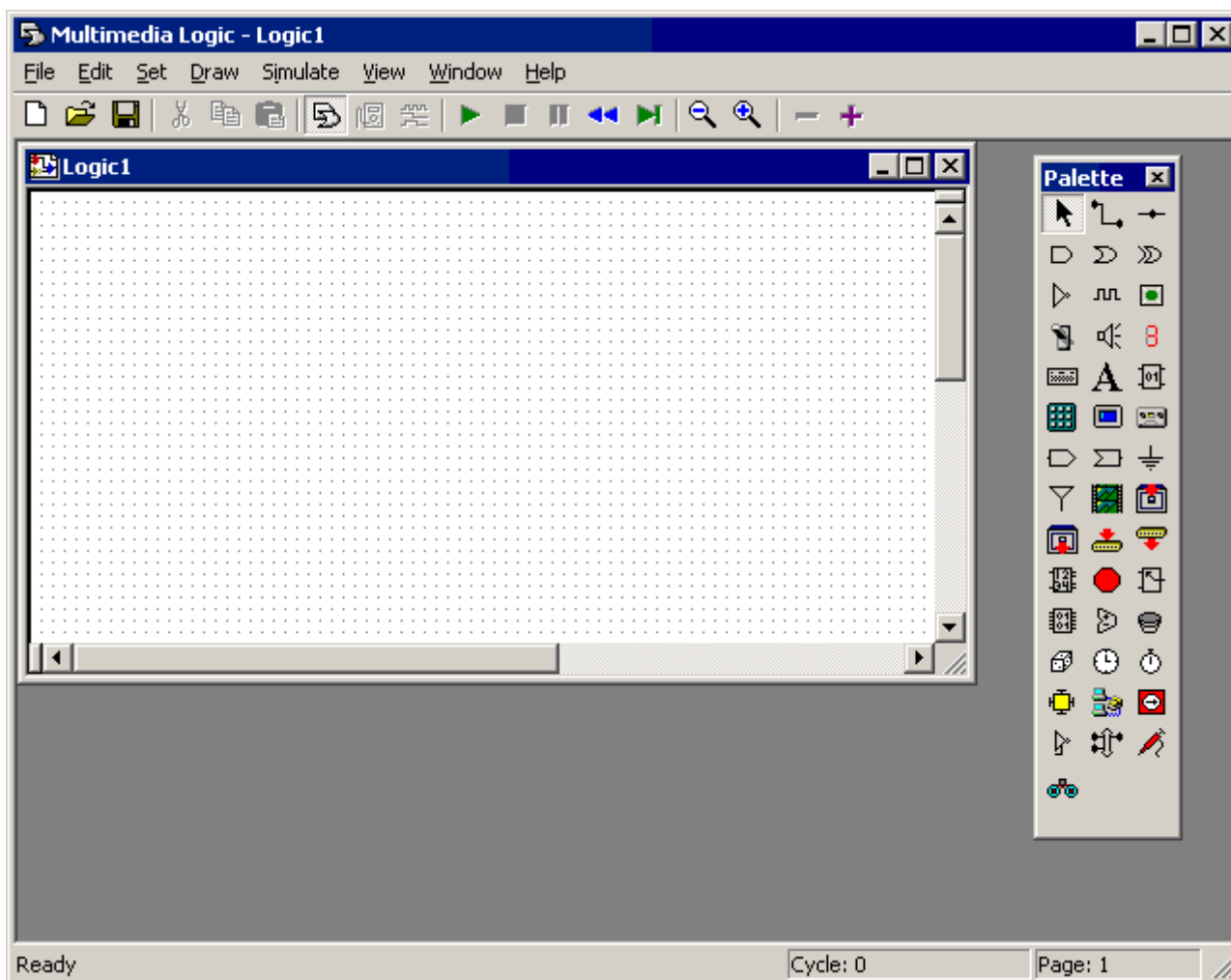


Multimedia Logic


1. Środowisko


Program MultiMedia Logic jest bezpłatnym środowiskiem (licencja GNU General Public License) firmy Softronics Inc wspomagającym projektowanie układów cyfrowych. Wersję instalacyjną programu można pobrać ze strony: <http://www.softronix.com/logic.html>.

Po uruchomieniu program startuje z czystym projektem układu o domyślnej nazwie „Logic1”. Wygląd środowiska przedstawia poniższy rysunek.



W górnej części okna programu znajdują się: *menu* z listą dostępnych funkcji oraz *pasek narzędzi* zawierający najczęściej wykonywane operacje.

Wewnątrz okna aplikacji widoczne jest okno z projektem układu „Logic1”. W programie może być jednocześnie otwartych kilka projektów – nazwa aktywnego projektu pojawia się w pasku tytułu programu (na rys.: *Multimedia Logic – Logic1*). Tworzenie nowych projektów, otwieranie istniejących i zapisywanie zmodyfikowanych projektów umożliwiają przyciski  lub odpowiednie opcje w menu **File** (**New**, **Open**, **Save**, **Save As**).

Właściwe przygotowanie projektu układu ułatwia *paleta narzędzi* zawierająca zestaw niezbędnych podzespołów (widoczna na rys. po prawej). Widoczność palety zależy od stanu przycisku  paska narzędzi.


2. Przygotowanie projektu

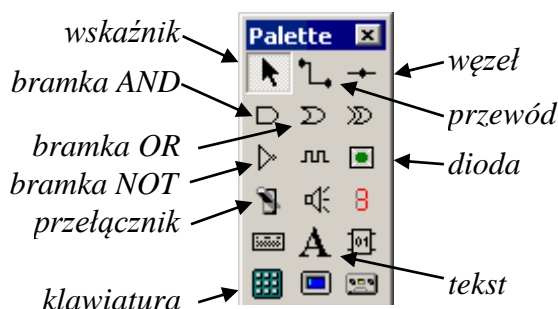
Okno projektu nowego układu zawiera pustą kartkę na którą należy nanieść właściwy schemat układu. Na kartce widoczna jest siatka, która może ułatwiać wyrównywanie elementów schematu. Parametry siatki i rozmiary kartki można zmienić po wybraniu opcji: **View|Grid Settings**, a przyciąganie elementów schematu do siatki włącza się i wyłącza po wybraniu polecenia **View|Snap to Grid** (domyślnie przyciąganie jest wyłączone).

W programie MultiMedia Logic elementy składające się na projekt układu dzielone są na trzy kategorie:

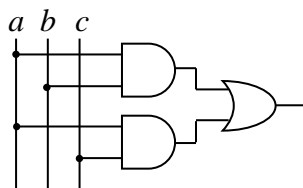
- **urządzenia wejściowe** (*Input Devices*) – urządzenia, które generują sygnały wejściowe dla projektowanego układu; w materiale zostaną omówione: *przełącznik* (*Switch Device*) oraz *klawiatura* (*Keypad Device*),
- **urządzenia wyjściowe** (*Output Devices*) – urządzenia, które przedstawią stany sygnałów wyjściowych układu; przykładowym elementem, który może przedstawić wizualnie stan swojego sygnału wejściowego jest *dioda* (*LED Device*) – jeżeli sygnał wejściowy ma wartość 0 dioda jest zgaszona, jeżeli sygnał ma wartość 1 dioda jest zapalona, jeśli sygnał ma wartość nieokreśloną dioda jest pomalowana w czerwono czarne paski,
- **urządzenia sterujące** (*Control Devices*) – urządzenia, które przetwarzają sygnały; omówione zostaną bramki logiczne *AND*, *OR*, *NOT* (*AND Gate*, *OR Gate* i *Inverter Gate*).

Odpowiednie łączenie elementów w programie umożliwiają narzędzia: *przewód* (*Wiring Tool*) oraz *węzeł* (*Node Tool*). Niezbędnym narzędziem programu jest także *wskaźnik* (*Selector Tool*), który pozwala na wybór elementu w celu jego usunięcia lub modyfikacji.

Elementy projektu tzn. urządzenia oraz narzędzia można wybierać posługując się opcją menu **Draw** lub z *palety narzędzi* (o ile jest ona widoczna – widoczność zależy od stanu przycisku ). Fragment *palety narzędzi* z elementami omawianymi w materiale został przedstawiony na poniższym rysunku.



Tworzenie projektu zostanie omówione na przykładzie urządzenia sterującego zwrotnicą, którego schematy logiczne zostały opracowane w punkcie 6. Na początku zrealizowany zostanie schemat zbudowany z elementów AND i OR.




Układ ma trzy wejścia: a , b , i c . Sygnały na tych wejściach mogą przyjmować wartości „0” i „1”. Do realizacji pojedynczego wejścia w programie zostanie wykorzystany tzw. **przełącznik**. Przełącznik w czasie symulacji układu może wysyłać sygnał „0” lub „1” w zależności od tego czy użytkownik go włączył czy wyłączył (jeśli typ przełącznika zostanie ustawiony na „chwilowy” to po puszczeniu klawisza myszy przełącznik wraca do poprzedniego stanu).

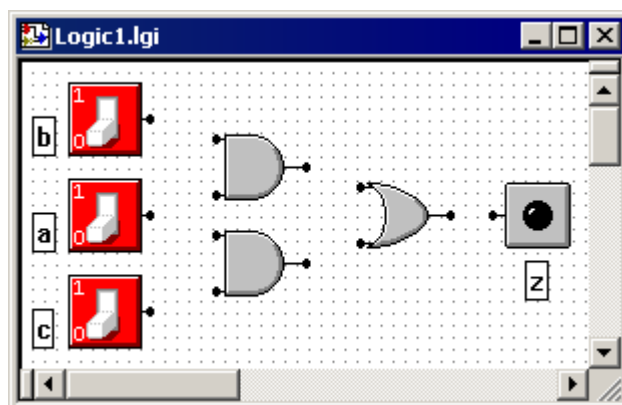
Urządzenie zbudowane jest z dwóch **bramek AND** i jednej **bramki OR**. Do przedstawienia stanu wyjścia układu wystarczy pojedyncza **dioda**.

Ostatecznie więc, na kartkę w oknie projektu należy przenieść trzy przełączniki, dwie bramki AND, jedną bramkę OR i jedną diodę. W tym celu należy wybrać właściwy element używając odpowiedniej opcji menu lub przycisku na palecie narzędzi i kliknąć lewym przyciskiem myszy w wybranym miejscu na kartce (po wyborze elementu, podczas ruchu myszą nad powierzchnią kartki kursor ma postać „+”).

Uwaga! Program pozostaje w trybie dodawania wybranego elementu dopóki nie zostanie wybrany inny element lub narzędzie *wskaźnika*.

Uwaga! Aby usunąć zbędny element należy:

- wybrać narzędzie *wskaźnik* korzystając z menu lub paska narzędzi (),
- kliknąć lewym przyciskiem myszy na elemencie,
- nacisnąć klawisz DEL.




Na powyższym rysunku pokazany został pierwszy etap projektu układu z naniesionymi wszystkimi niezbędnymi elementami. Dla zwiększenia czytelności schematu, przełącznik generujący sygnał „ a ” został umieszczony pomiędzy pozostałymi przełącznikami.



W kolejnym kroku należy uzupełnić brakujące połączenia. Każde połączenie rozpoczyna się i kończy węzłem. Wejścia i wyjścia wszystkich elementów zakończone są węzłami (na schemacie węzły zaznaczane są czarnymi kropkami). Użytkownik może wprowadzać również dodatkowe węzły (narzędzie *Node Tool*) w celu uzyskania bardziej czytelnych schematów.

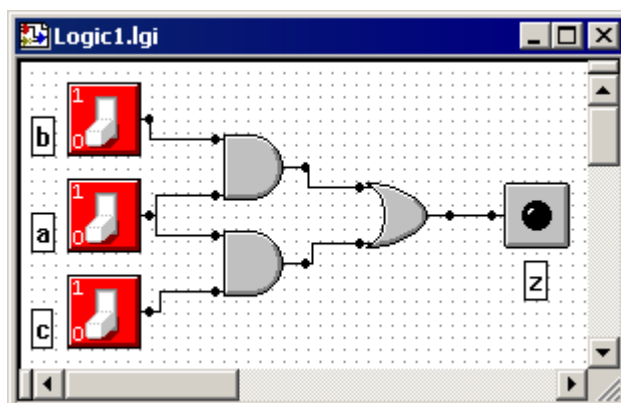
Do łączenia elementów w programie wykorzystywane są tzw. *przewody*. Aby dodać połączenie należy:

- wybrać narzędzie *przewód* korzystając z menu lub paska narzędzi (),
- kliknąć lewym przyciskiem myszy na początkowym węźle połączenia,
- nie zwalniając przycisku przesunąć kursor myszy do węzła końcowego,
- zwolnić przycisk myszy.

Uwaga! Połączenie znika jeżeli użytkownik nieprecyzyjnie wskaże węzeł końcowy.

Uwaga! Z wyjścia elementu można wyprowadzić wiele połączeń, do wejścia można doprowadzić tylko jedno połączenie.

Uwaga! Aby usunąć zbędne połączenie po wybraniu *wskaźnika* należy kliknąć lewym przyciskiem myszy na jednym z węzłów połączenia i nacisnąć klawisz DEL.



3. Symulacja działania układu

Zachowanie układu można zbadać po uruchomieniu symulacji. Polecenia związane z pracą w trybie symulacji układu zostały zgrupowane w menu **Simulate**. Dodatkowo część poleceń została umieszczona na pasku narzędzi.



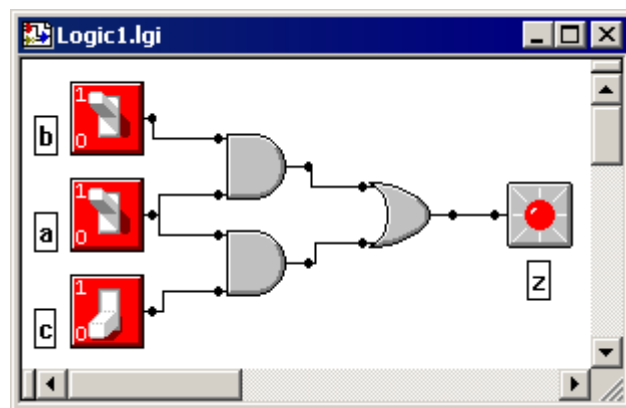
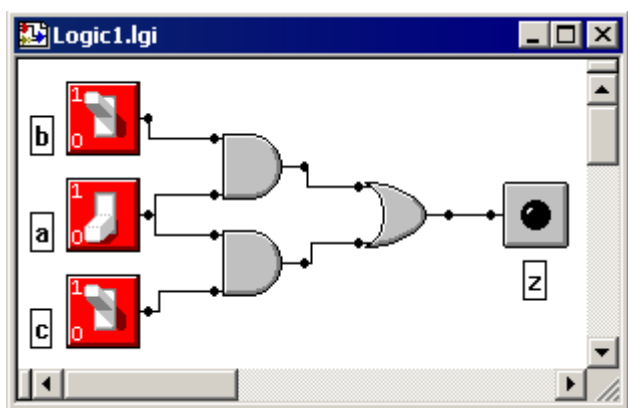
Pierwszy z przycisków widocznego na powyższym rysunku fragmentu paska, uruchamia symulację (polecenie można również wybrać z menu **Simulate|Run**). Po jej uruchomieniu znika *paleta narzędzi*, kartka na której narysowany jest schemat układu traci siatkę, a odpowiednie przyciski paska narzędzi tracą i zyskują aktywność:



W trybie symulacji można testować działanie układu badając wpływ sygnałów wejściowych na reakcję układu. Symulację można zakończyć klikając na drugim z przycisków widocznego powyżej paska lub

zatrzymać klikając na trzecim przycisku (można również wybrać z menu **Simulate** odpowiednio opcje: **Stop** lub **Pause**).

Zaprojektowane w punkcie poprzednim urządzenie powinno generować sygnał sterujący dla zwrotnicy. Jeżeli co najmniej dwie cechy detalu mają prawidłowe wartości i jedną z nich jest cecha a to na wyjściu układu powinien pojawić się sygnał „1”, w przeciwnym przypadku na wyjściu powinno pojawić się „0”. Na poniższych rysunkach pokazane zostały dwie spośród ośmiu możliwych kombinacji wartości cech (a, b, c) przesuwanego detalu. Rysunek po lewej stronie pokazuje reakcję układu na detal o cechach $a = 0, b = 1, c = 1$ – sygnał wyjściowy układu zgodnie z założeniami ma wartość „0” (dioda zgaszona). Na rysunku po prawej widoczna jest reakcja na detal o cechach $a = 1, b = 1, c = 0$ – sygnał wyjściowy ma wartość „1” (dioda zapalona).



Dwie prawidłowe reakcje nie oznaczają poprawnego funkcjonowania układu w każdej sytuacji. Słuszność zaproponowanego rozwiązania potwierdziłyby dopiero testy dla wszystkich możliwych typów detali (patrz tabela). Zgodnie z oznaczeniami typów detali umieszczonymi w tabeli, na powyższych rysunkach widoczne są testy dla detalu 3. (rysunek po lewej) i detalu 6. (rysunek po prawej).

<i>detal</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>z</i>
0.	0	0	0	0
1.	0	0	1	0
2.	0	1	0	0
3.	0	1	1	0
4.	1	0	0	0
5.	1	0	1	1
6.	1	1	0	1
7.	1	1	1	1

4. Przygotowanie projektu cd.

W trakcie pracy z projektem czasem niezbędna jest modyfikacja własności wykorzystywanych elementów. Każdy z dostępnych w programie podzespołów posiada zestaw specyficznych dla swojego typu własności. Własności te można przeglądać lub modyfikować w specjalizowanych okienkach, które wywołuje się klikając dwukrotnie lewym przyciskiem myszy na elemencie lub korzystając z opcji **Edit|Properties** (po wcześniejszym wskazaniu elementu).

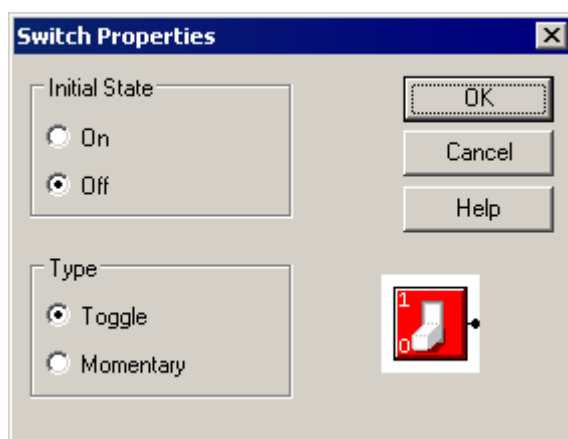
Poniżej zostaną omówione własności wybranych elementów przedstawionych w punkcie 2.

4.1. Własności przełącznika

Przełącznik zamyka (pozycja „1” przełącznika) lub otwiera (pozycja „0” przełącznika) obwód w którym został umieszczony. Sygnał na wyjściu przełącznika odpowiada stanowi tego odvodu: „1” jest przepływ prądu, „0” nie ma przepływu.

Początkowy stan przełącznika domyślnie jest ustawiany na „Off” – obwód jest przerwany. Stan ten można zmienić modyfikując własność „Initial State”.

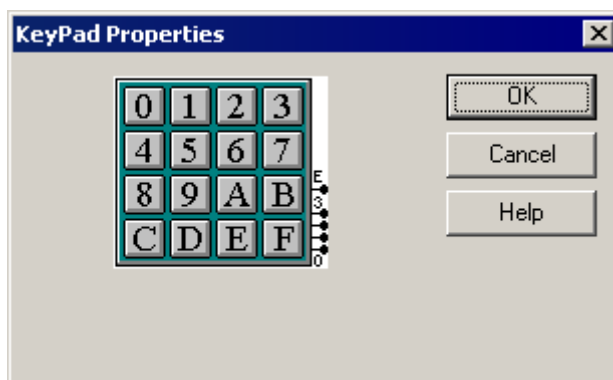
Domyślnie przełącznik powoduje stałe zamknięcie lub otwarcie obwodu. Zachowanie to można zmienić modyfikując typ przełącznika. Zamiast domyślnego przełącznika typu „Toggle” można wykorzystać przełącznik „Momentary”. Przełącznik taki powoduje zamknięcie lub otwarcie obwodu (w zależności od stanu początkowego) tylko w czasie gdy jest on naciskany.



4.2. Własności klawiatury

Klawiatura jest urządzeniem wejściowym, które może generować na swoich czterech wyjściach ponumerowanych od „0” do „3” 16 różnych kombinacji sygnałów „0” i „1”. Sygnały te odpowiadają dwójkowej reprezentacji cyfry szesnastkowej umieszczonej na klawiszu.

Klawiatura posiada również dodatkowe wyjście „E” – na którym pojawia się sygnał „1” jeżeli jeden z klawiszy jest aktualnie wciśnięty i „0” w przeciwnym przypadku. Sygnały odpowiadające kodowi ostatnio wciśniętego klawisza są wysyłane na wyjścia „0”-„3” niezależnie od tego czy klawisz ten jest jeszcze wciśnięty czy już nie.



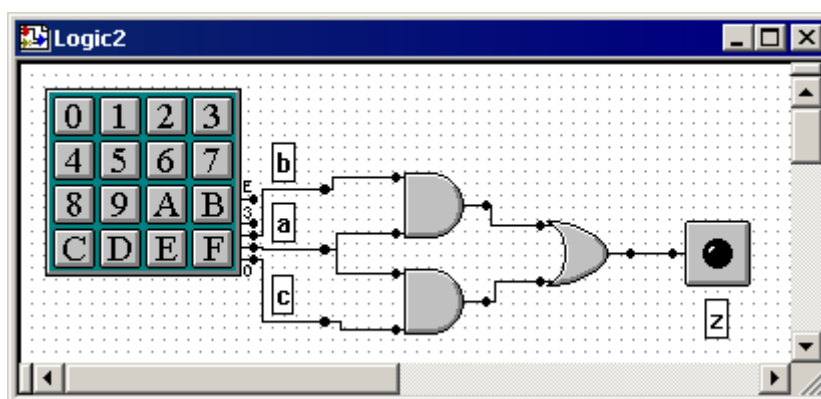
W poniższych tabelach przedstawiony został stan wyjść urządzenia w zależności od naciśniętego klawisza.

klawisz	wyjście			
	3	2	1	0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1

klawisz	wyjście			
	3	2	1	0
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Wykorzystanie klawiatury jako źródła sygnałów wejściowych może stanowić wygodną alternatywę dla zestawu przełączników w przypadku układów wielowijściowych.

Na poniższym rysunku przedstawiony został układ zaprojektowany w punkcie 2. w którym zestaw trzech przełączników został zastąpiony klawiaturą.



Wykorzystane zostały tylko trzy wyjścia klawiatury – wszystkie możliwe kombinacje sygnałów na tych wyjściach można otrzymać klikając w trakcie symulacji na klawiszach „0”-„7” lub „8”-„F” (patrz tabela po lewej). W tabeli po prawej przypomniane zostały oznaczenia typów detali wprowadzone w punkcie 3. Test dla detalu 3. wymagałby więc klawisza „5” lub „D” a dla detalu 6. klawisza „6” lub „E”.

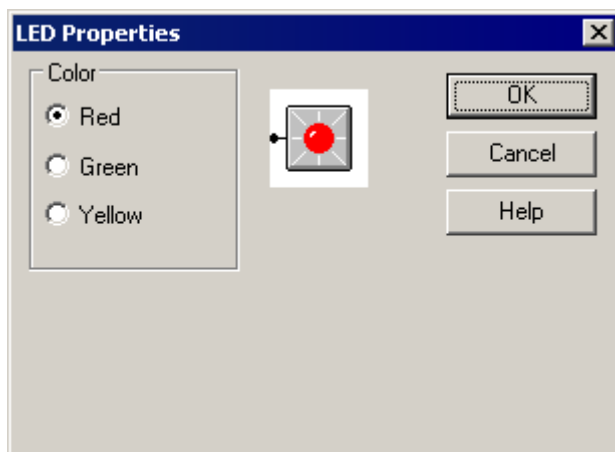


klawisz	wyjście		
	2 b	1 a	0 c
0 lub 8	0	0	0
1 lub 9	0	0	1
2 lub A	0	1	0
3 lub B	0	1	1
4 lub C	1	0	0
5 lub D	1	0	1
6 lub E	1	1	0
7 lub F	1	1	1

detal	a	b	c	z
0.	0	0	0	0
1.	0	0	1	0
2.	0	1	0	0
3.	0	1	1	0
4.	1	0	0	0
5.	1	0	1	1
6.	1	1	0	1
7.	1	1	1	1

4.3. Własności diody

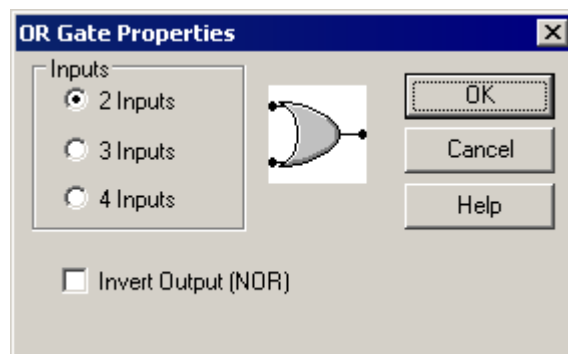
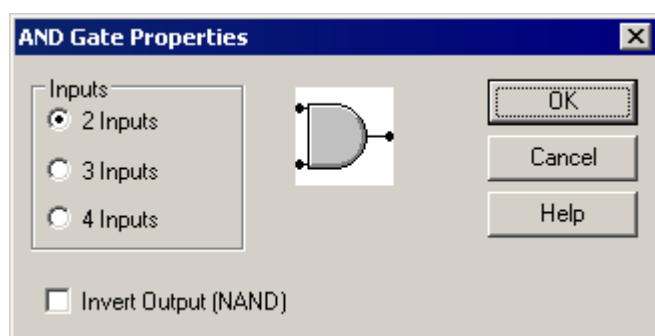
Jedyną własnością diody podlegającą modyfikacji jest jej kolor. Domyślnie dioda jest czerwona, w programie można używać także diody zielone i żółte.



4.4. Własności bramek AND i OR

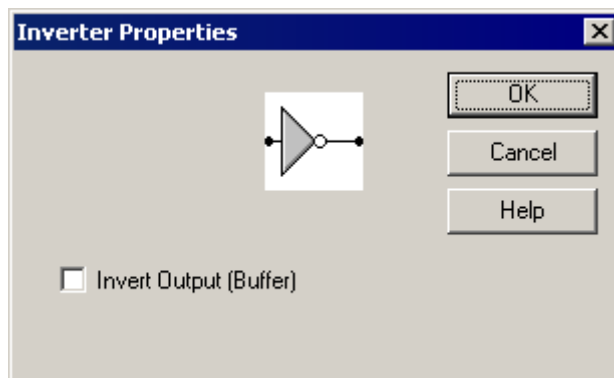
Bramki AND i OR mają domyślnie po dwa wejścia, liczbę wejść można jednak zmienić – w programie można również wykorzystywać bramki o trzech i czterech wejściach.

Okienko właściwości obydwu bramek posiada również możliwość zanegowania wartości sygnału wyjściowego bramki, tym samym negując wyjście bramki AND otrzymuje się bramkę NAND a negując wyjście bramki OR otrzymuje się bramkę NOR.



4.5. Własności bramki NOT

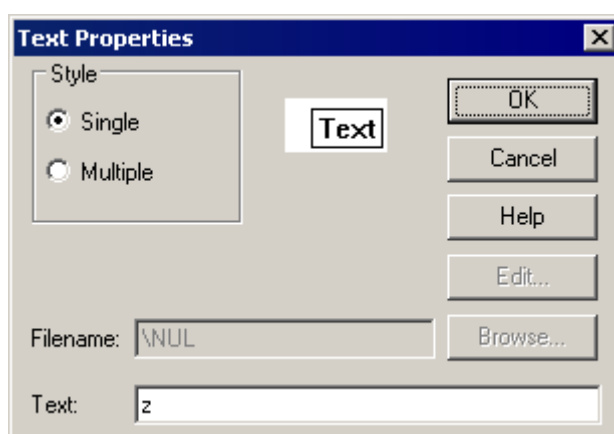
Bramka NOT może również pracować jako tzw. *bufor* (otrzymuje się go negując wyjście bramki NOT). *Bufor* nie modyfikuje wartości sygnału (sygnał na jego wyjściu jest równy sygnałowi wejściowemu), wprowadza jedynie opóźnienie równe czasowi, który jest potrzebny żeby sygnał z wejścia bramki przesłać na jej wyjście.



4.6. Własności elementu Text

Text może być elementem statycznym pozwalającym na umieszczanie opisów wybranych elementów projektu. Takimi opisami były oznaczenia sygnałów wejściowych i wyjściowych („a”, „b”, „c”, „z”) projektowanego w materiale urządzenia. *Text* może być także urządzeniem wyjściowym wewnątrz którego wyświetlane są wartości wybranych sygnałów. Sposób funkcjonowania *Textu* zależy od jego stylu.

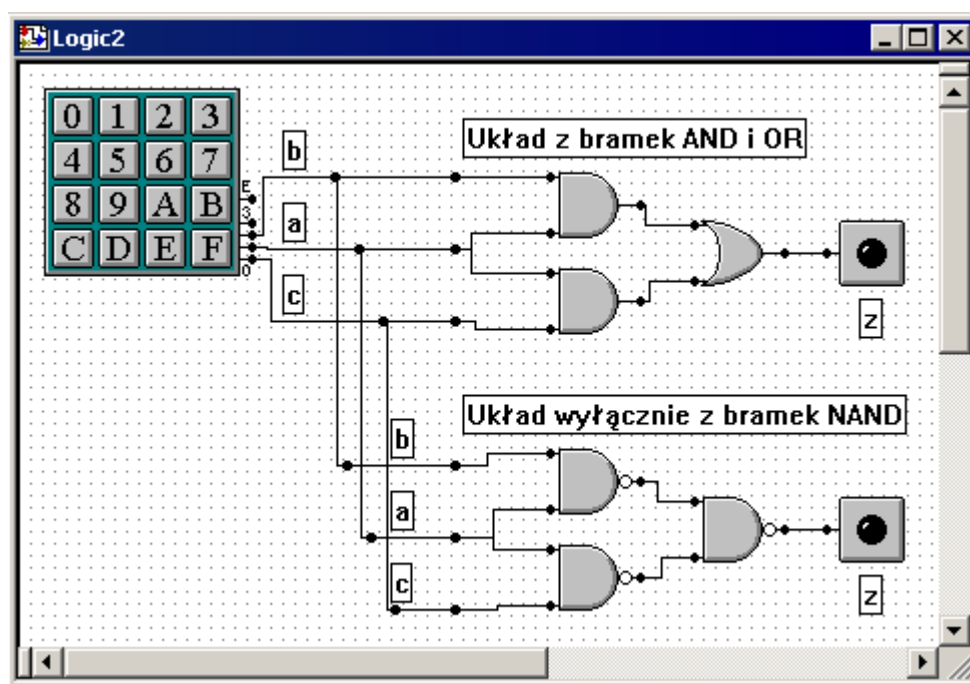
Domyślny styl „Single” oznacza element opisowy – właściwy opis należy wprowadzić w polu „Text” okienka właściwości. Zmiana stylu na „Multiple” powoduje, że *Text* staje się urządzeniem wyjściowym, w takim przypadku należy określić sygnały na podstawie których element ten będzie wyświetlał teksty oraz należy wskazać plik z tekstami do wyświetlania.



5. Rozbudowa przykładowego projektu

W punkcie 6. zaproponowane zostały dwa sposoby rozwiązania problemu sortowania detali. Układ odpowiadający schematowi urządzenia zbudowanego z bramek AND i OR został przygotowany w punkcie 2. (wejścia zadawane przełącznikami) i zmodyfikowany w punkcie 4. (wejścia zadawane klawiaturą).

Na poniższym rysunku przedstawiony został układ zbudowany z dwóch podukładów. Pierwszy z nich odpowiada układowi zbudowanemu pierwotnie, drugi to układ zbudowany z bramek NAND. Dzięki podłączeniu obydwu układów do tego samego źródła sygnałów wejściowych, można łatwo sprawdzić, że obydwa układy działają w ten sam sposób (dla zwiększenia czytelności schematu zostały wprowadzone dodatkowe węzły).



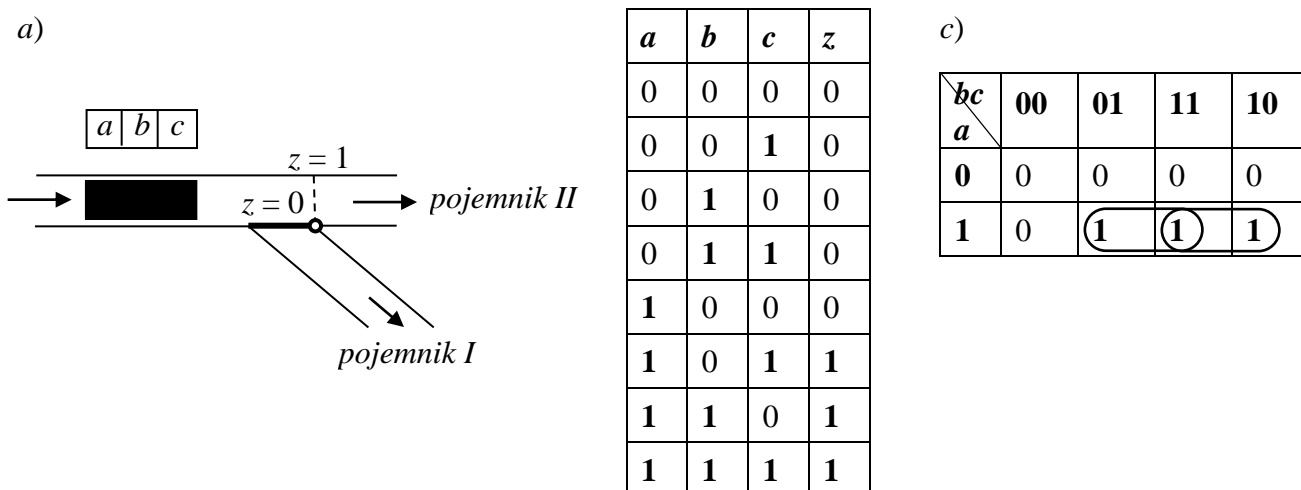
6. Przykład

Należy zaprojektować urządzenie sterujące w taki sposób aby generowało odpowiedni sygnał sterujący dla zwrotnicy urządzenia sortującego. Zwrotnica w stanie załączonym ($z = 1$) powoduje, że detal trafia do pojemnika I, w stanie nie załączonym ($z = 0$) kieruje detal do pojemnika II. Przed przesunięciem detalu do odpowiedniego pojemnika, badane są przy pomocy odpowiednich czujników trzy cechy (a , b , c) każdego z detali. Każdy z czujników sygnalizuje zbadaną przez siebie cechę jedną z dwóch wartości (1 – wartość prawidłowa, 0 – wartość nieprawidłowa). Detal powinien trafić do pojemnika I jeżeli co najmniej dwie cechy mają prawidłowe wartości i jedną z nich jest cecha a , w przeciwnym razie powinien być przesunięty do pojemnika II.

Układ należy zaprojektować z użyciem bramek: a) AND, OR, NOT, b) wyłącznie bramek NAND.

b)





Rys. 1. Ilustracja zadania a) schemat urządzenia sortującego, b) tablica prawdy i c) tablica Karnaugh urządzenia sterującego zwrotnicą.

Z tablicy Karnaugh można odczytać *minimalną normalną postać dysjunkcyjną* funkcji logicznej urządzenia:

$$z = ac + ab . \tag{1}$$

Na podstawie wyrażenia (1) można zbudować układ zbudowany z bramek AND i OR – układ taki został przedstawiony na rys. 2.a.

Konstrukcja układu z bramek NAND wymaga odpowiedniego przekształcenia wyrażenia (1):

$$z = ac + ab = \overline{\overline{ac}} + \overline{\overline{ab}} = \overline{\overline{ac} \cdot \overline{ab}} \tag{2}$$

Na rys. 2b. przedstawiony został schemat układu odpowiadający wyrażeniu (2).



Rys. 2. Schemat logiczny urządzenia sterującego zrealizowany na podstawie zależności a) 1, b) 2.