Logika układów cyfrowych - laboratorium

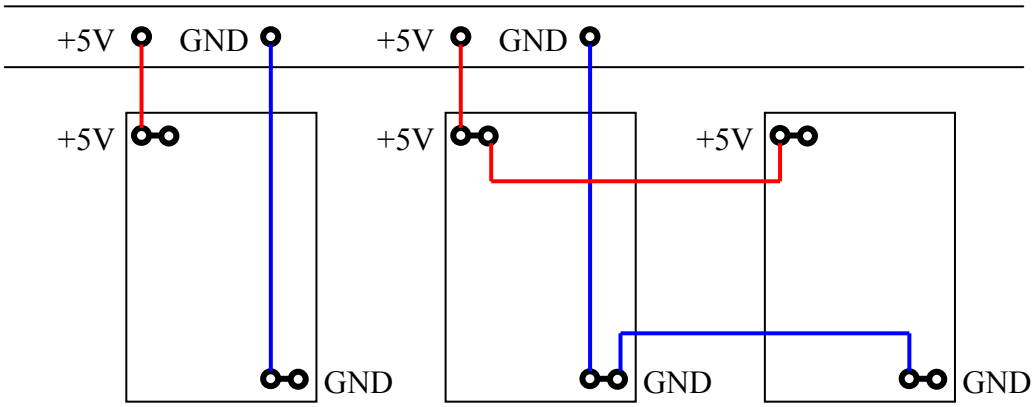
Podane poniżej wskazówki mogą pomóc w uzyskaniu dobrych wyników w czasie pracy w laboratorium. Powstały one na podstawie obserwacji najczęściej występujących problemów i popełnianych błędów.

Wskazówki dotyczą zarówno różnych niuansów związanych ze sprzętem używanym w laboratorium jak też podejścia do projektowania i uruchamiania układów. Większość uwag dotyczy typowych problemów, z którymi będzie można spotkać się już od początkowych zajęć (najpierw układy kombinacyjne, potem sekwencyjne). Wyjątkiem są rozważania dotyczące uruchamiania automatu asynchronicznego, które będą przydatne dopiero przy realizacji tego konkretnego ćwiczenia.

1. Uwagi na temat sprzętu

Zasilanie modułów

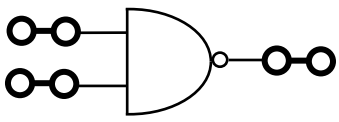
Wymienne moduły w zestawach UNILOG muszą być oczywiście zasilane. Piny modułów oznaczone napisami +5 V i GND powinny być podłączone do tak samo oznaczonych pinów na środkowej listwie (bezpośrednio lub przez zdublowane piny innego modułu jak pokazano na poniższym rysunku):



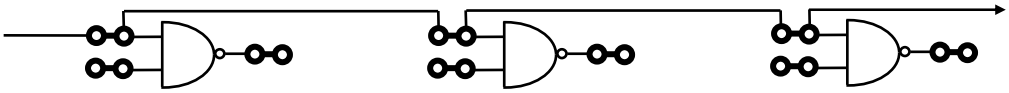
Warto przyjąć zasadę podłączania zasilania na początku łączenia (bo i tak przecież nie można tego pominąć). W przeciwnym razie, zwłaszcza przy dużej liczbie połączeń sygnałowych, można łatwo przeoczyć brak zasilania modułu. Wykrycie takiej sytuacji w gotowym układzie utrudnia fakt, że moduł bez zasilania (lub z zasilaniem niekompletnym, na przykład podłączone tylko +5 V lub GND) nie zachowuje się wcale jak „izolator”. Może być wtedy obserwowane dziwne zachowanie bramek w związku z pasożytniczym zasilaniem od wejść lub wyjść.

Łączenie bramek

Wejścia i wyjścia bramek i przerzutników w wymiennych modułach (podobnie jak piny zasilania) są zdublowane (dwa połączone piny). Zilustrowane jest to poniżej:



Najczęściej wykorzystywane jest to w sytuacji, gdy jeden sygnał sterujący musi być podłączony do wejść wielu bramek jak na poniższym rysunku:



Taki sposób łączenia pozwala doprowadzić jeden sygnał do praktycznie nieograniczonej liczby wejść. Alternatywnym sposobem łączenia jest zastosowanie modułu rozgałęźnika, ale jest to raczej ostateczność, gdyż wymaga większej liczby przewodów połączeniowych, co niepotrzebnie komplikuje układ.

Sprawdzanie sprzętu

Moduły i przewody połączeniowe nie zawsze są sprawne. Aby zwiększyć szansę uruchomienia układu, warto sprawdzić je przed użyciem. Jednak pełne testy nie zawsze są możliwe ze względu na ograniczony czas na wykonanie ćwiczenia. Poza tym nawet sprawdzony przewód może ulec uszkodzeniu już w trakcie łączenia, może też wystąpić problem niepewnego kontaktu wtyku i gniazda.

Jednak w przypadku niektórych układów wstępne testy są dość proste i warto je wykonać. Na przykład przerzutniki JK można sprawdzić podłączając najpierw tylko zasilanie, sygnał zegarowy i wyjście Q do obserwacji stanu (wejścia JK mogą pozostać wiszące). Połączenia te będą i tak potrzebne, niezależnie od tego, jaki będzie układ docelowy. Znając działanie przerzutnika JK i obserwując zmiany jego stanu przy podawaniu impulsów zegarowych łatwo wykryć uszkodzony przerzutnik. Nie jest to oczywiście pełny test, ale z praktyki wynika, że rzadko zdarza się sytuacja, że przerzutnik, który przeszedł pomyślnie taki ograniczony test okazuje się mimo to niesprawny (problem musiałby dotyczyć wejść J lub K).

Wiszące wejście bramki

W układach TTL (a takie używane są w laboratorium) pozostawienie wiszącego wejścia jest równoważne podaniu na to wejście jedynki logicznej. Można to wykorzystać praktycznie na przykład wtedy, gdy potrzebna jest 3-wejściowa bramka NAND a pod ręką jest akurat bramka NAND 4-wejściowa. Wynika to wprost z algebry Boole'a:

$$\overline{a\,b\,c\,1} = \overline{a\,b\,c}$$

Ten sam efekt można oczywiście uzyskać podłączając nieużywane wejścia do jedynki logicznej (pin oznaczony symbolem H na listwie środkowej zestawu UNILOG) lub zwierając je z którymś z używanych wejść, ale wymaga to użycia dodatkowych przewodów (zbędna komplikacja układu połączeń). Podłączanie nieużywanych wejść jest co prawda zalecane w praktycznych zastosowaniach (większa odporność na zakłócenia), ale w testach laboratoryjnych można się tym nie przejmować.

Zasada interpretacji wiszących wejść jest jednoznaczna (jedynka logiczna), ale nie może być stosowana mechanicznie. Nie w każdym przypadku nieużywane wejścia można pozostawić wiszące, tak jak w pokazanym przykładzie dla bramki NAND. Na przykład dla bramki NOR mamy:

$$\overline{a + b + c + 1} = 0 \neq \overline{a + b + c}$$

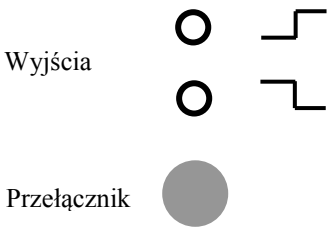
Przełączniki stabilne i niestabilne

Wejścia układu mogą być sterowane z przełączników dwóch typów: stabilnych i niestabilnych (oznaczonych napisem PULSE). Czasem wybór między nimi wynika tylko z wygody użytkowania. Na przykład, jeśli w układzie potrzebny jest sygnał typu RESET (kasowanie układu krótkim impulsem), to lepiej użyć przycisku niestabilnego (nie ma ryzyka, że zapominając wyłączyć przycisk pozostawimy układ w niezamierzonym stanie reset).

Jest bardzo istotna różnica między tymi dwoma typami przełączników. Sygnał wyjściowy z przełączników stabilnych odzwierciedla bezpośrednio stan styków przełącznika. Oznacza to, że przy zmianie stanu przełącznika mogą pojawiać się impulsy wynikające z drgań styków. Impulsy te mają czasy rzędu milisekund, co przy statycznych obserwacjach nie ma znaczenia. Jeśli jednak chcemy sterować wejściem, dla którego istotne są zbocza impulsu (w typowym przypadku jest to wejście zegarowe przerzutnika albo wejścia automatu asynchronicznego), to musimy skorzystać z przełączników niestabilnych. Wyposażone są one w odpowiednie układy logiczne eliminujące efekt drgań zestyków (wciśnięcie i zwolnienie przełączników daje zawsze sygnał o „czystych" zboczach).

UWAGA

W przeciwieństwie do pinów zasilania oraz wejść i wyjść bramek dwa piny przy przełącznikach (zarówno stabilnych jak i niestabilnych) NIE są zwarte (pin dolny daje negację sygnału wyjściowego przełącznika – zero logiczne przy wciśnięciu):



2. Przygotowanie do zajęć

Samodzielna praca

W czasie zajęć laboratoryjnych łączone i uruchamiane są układy, które trzeba wcześniej zaprojektować (wykonać syntezę) na podstawie podanych założeń. Schemat logiczny układu jest tylko efektem końcowym. Prace projektowe należy wykonać samodzielnie, nawet jeśli czasem zdarza się, że tematem ćwiczenia jest układ standardowy, dla którego można znaleźć gotowy schemat. Sensem ćwiczenia jest przecież nauczenie się samodzielnego projektowania (i uruchamiania) dowolnych układów a nie wyszukiwania gotowych rozwiązań.

Ważne jest, aby w czasie zajęć mieć przy sobie kompletny projekt, zawierający wszystkie kroki syntezy. Jest to potrzebne nie tylko do uwiarygodnienia autorstwa projektu, ale przede wszystkim jest niezbędne w fazie uruchamiania układu, gdy może okazać się, że projekt zawiera błędy. Aby je odnaleźć i usunąć trzeba prześledzić poszczególne kroki syntezy układu (zostanie to wyjaśnione dokładniej poniżej).

Przy pracy w grupach dwuosobowych zwykle następuje podział zadań. Trzeba jednak pamiętać, że obie osoby powinny znać szczegóły projektu. Argument typu: „nie wiem, o co tu chodzi, to kolega przygotowywał" jest raczej niepoważny.

Czytelność projektu

Często zdarza się, że przyczyną niepowodzenia nie są wcale braki merytoryczne, lecz niedbałość przy wykonywaniu projektu. Przy nieczytelnym zapisie równań logicznych łatwo popełnić błąd. W najgorszym przypadku, gdy taki błąd zostanie popełniony już w początkowej fazie projektu, może to oznaczać konieczność ponownego wykonania całej pracy. Trudno będzie również sprawnie połączyć i uruchomić układ na podstawie nieczytelnego schematu logicznego.

Weryfikacja projektu

Błędy w projekcie mogą się zdarzyć zawsze, nawet przy zachowaniu największej staranności. Dlatego warto przygotowywać projekt wcześniej (nie dopiero kilka godzin przed zajęciami) tak, aby pozostał czas na jego sprawdzenie. Jeśli (przy pracy w grupach dwuosobowych) jedna osoba wykonuje syntezę układu, to dobrym pomysłem jest sprawdzenie go przez drugą osobę. Do weryfikacji projektu można też wykorzystać ogólnie dostępne symulatory układów logicznych (choć oczywiście nie jest to obowiązkowe).

Odpowiedni schemat

Każdy układ logiczny można zrealizować przy użyciu bramek i przerzutników różnych typów. W niektórych przypadkach typ bramek będzie narzucony w założeniach projektu, najczęściej jednak będzie to pozostawione do wyboru. Warto oczywiście zoptymalizować projekt tak, aby uzyskać jak najmniejszą liczbę bramek (taki układ łatwiej uruchomić).

Trzeba jednak koniecznie pamiętać, że nie wszystkie typy bramek są dostępne w laboratorium. Na przykład brak w ogóle bramek AND i OR (dostępne są NAND, NOR i ExOR) oraz przerzutników T (dostępne są D i JK). Bramki NOR i ExOR są tylko dwuwejściowe, bramki NAND mają warianty z różną liczbą wejść. Oczywiście każdy schemat można przekształcić tak, aby zawierał tylko bramki zadanego typu, ale wymaga to czasu. Dlatego trzeba jak najszybciej zapoznać się z zestawem bramek dostępnych w laboratorium i zawsze to uwzględniać już w fazie projektowania układu. Ewentualne przekształcanie schematu dopiero w trakcie zajęć powinno być sytuacją wyjątkową.

Istnieją dwa sposoby tworzenia schematu na podstawie funkcji logicznej przy wykorzystaniu bramek logicznych zadanego typu. Pierwszy polega użyciu schematów zastępczych bramek przy przekształcaniu schematu. Jest to oczywiście sposób poprawny, ale dość pracochłonny, dlatego polecane jest raczej podejście, polegające na przekształceniach funkcji logicznej (wykorzystuje się w nim głównie neutralność podwójnej negacji i prawa de Morgana). Ilustruje to poniższy przykład:

$$f = a\,b + c\,d = \overline{\overline{a\,b + c\,d}} = \overline{\overline{ab}\;\overline{cd}}$$

Postać wyjściowa funkcji wymagałaby dwóch dwuwejściowych bramek AND (niedostępne) oraz jednej dwuwejściowej bramki OR (również niedostępna). W końcowej postaci funkcji widać natomiast wyraźnie trzy dwuwejściowe bramki NAND (dostępne).

Trzeba pamiętać też o konwencji logicznej wejść asynchronicznych SET i RESET przerzutników. Dla przerzutników dostępnych w laboratorium aktywny jest stan niski. Dzięki temu nieużywane wejścia mogą pozostać wiszące.

3. Uruchamianie układów

Uruchamianie układu w czasie zajęć jest końcowym etapem projektu. Mogłoby się wydawać, że jest to czynność banalna, rutynowa i pozbawiona większego sensu. Trzeba jednak pamiętać, że każdy układ logiczny, nawet zaprojektowany przy użyciu doskonałych narzędzi i wykonany w najlepszej technologii może nie działać poprawnie. Wówczas autor projektu musi umieć znaleźć przyczynę problemu, w sposób metodyczny, wykorzystując wiedzę o działaniu układów logicznych. Można się tego nauczyć tylko poprzez praktykę. Na szczęście (dla efektu dydaktycznego) układy w laboratorium po połączeniu nie zawsze od razu działają. Przyczyny mogą być różne:

- błędy w projekcie,
- błędy w połączeniach,
- niesprawne bramki lub przewody.

Pierwsza z tych przyczyn może być w dużym stopniu ograniczona przez dobrze przygotowany projekt (praktycznie w 100 % jeśli układ został sprawdzony przy użyciu symulatora logicznego). Dwie pozostałe można zminimalizować przez staranne łączenie i sprawdzanie bramek i przewodów. Trzeba jednak pamiętać, że czas przeznaczony na uruchomienie układu jest ograniczony i nie zawsze takie pełne sprawdzenie jest możliwe. Poza tym, nawet przy największej staranności usterki mogą się jednak pojawić i trzeba koniecznie zarezerwować czas na ewentualne ich usuwanie.

Jeśli połączony układ nie działa, to właściwie dopiero wtedy można w pełni wykorzystać wiedzę o działaniu układów logicznych. Nie należy w żadnym wypadku wpadać w panikę, trzeba próbować metodycznie znaleźć przyczynę problemu. Najgorszy z możliwych sposób „naprawy" problemu to rozłączenie układu i połączenie od nowa. Pomijając fakt, że taki sposób (a właściwie rozpaczliwa próba) jest zwykle nieskuteczny, to jako całkowicie kompromitujący nie powinien być nigdy stosowany.

Diagnozowanie problemu

Przy metodycznej diagnostyce układu trzeba najpierw ustalić, czy nieprawidłowe zachowanie układu wynika z błędów w projekcie czy też z błędnych połączeń lub problemów technicznych ze sprzętem. Pokazane zostanie to na przykładach, uwzględniających specyfikę układów kombinacyjnych i sekwencyjnych.

Układy kombinacyjne

Poprawność działania układu kombinacyjnego sprawdzamy podając wszystkie możliwe kombinacje sygnałów wejściowych i obserwując, czy stany logiczne wejść są zgodne z założonymi (według tabeli prawdy). Jeśli z projektu wynika, że dla pewnych kombinacji wejść stan wyjść jest nieistotny, to sprawdzanie tych kombinacji można oczywiście pominąć.

Każda niezgodność oznacza niepoprawne działanie układu. Liczba niezgodności nie odzwierciedla skali problemów w układzie. Nie można więc powiedzieć, że układ, w którym tylko jedna kombinacja daje niepoprawny stan wyjścia (lub wyjść) jest bliższy celu niż ten, w którym, tylko jedna kombinacja daje poprawny wynik.

Przykład

Tabela prawdy

a	*b*	*c*	*d*	*f*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

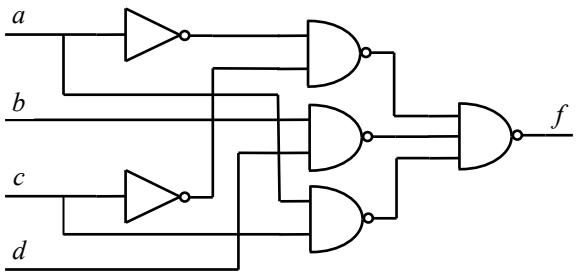
Siatka Karnaugh'a

cd \ *ab*	00	01	11	10
00	1	1	0	0
01	1	1	1	0
11	0	1	1	1
10	0	0	1	1

Funkcja logiczna

$$f = \bar{a}\,\bar{c} + b\,d + a\,c = \overline{\overline{a\,c}\;\;\overline{bd}\;\;\overline{ac}}$$

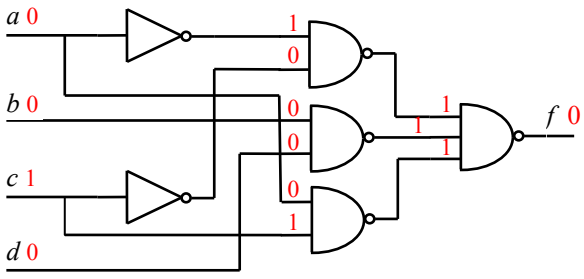
Schemat



Testując układ podajemy na wejścia *abcd* kolejne kombinacje stanów i sprawdzamy stan wyjścia według tabeli prawdy. Załóżmy, że dwie pierwsze kombinacje dają poprawny wynik, natomiast dla kombinacji *abcd* = 0010 wyjście ma poziom logiczny 1 zamiast 0. W tej sytuacji sprawdzanie dalszych kombinacji nie ma sensu i jest stratą czasu. Może się bowiem okazać, że błędy dla dalszych kombinacji mają to samo źródło. Trzeba znaleźć i usunąć przyczynę błędu dla pierwszej kombinacji wejść dającej błędne wyjście.

W pierwszym kroku trzeba ustalić, co wynika ze schematu, aby stwierdzić, czy jest on poprawny, to znaczy czy ze schematu wynika spodziewany stan wyjścia.

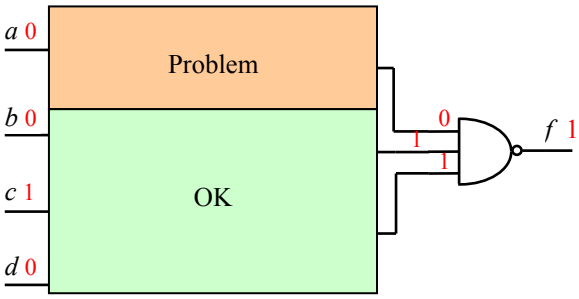
I. Poprawny projekt i schemat, błąd w układzie



Analiza poziomów logicznych na schemacie pokazuje, że spodziewany stan wyjścia to 0. Tak więc problem dla kombinacji *abcd* = 0010 nie wynika z błędu w projekcie, lecz z problemu połączonym układzie (błędne połączenie, uszkodzony przewód, brak kontaktu złącza, uszkodzona lub nieprawidłowo zasilana bramka). Ze schematu wynika, że końcowa bramka NAND powinna mieć na wszystkich trzech wejściach stan 1. Należy sprawdzić, jaki jest faktyczny stan wejść.

Jeśli stan logiczny wszystkich trzech wejść bramki NAND jest wysoki, oznacza to problem z bramką. Trzeba przede wszystkim sprawdzić doprowadzenie zasilania, ewentualnie sprawdzić działanie wymieniając bramkę (może być ona uszkodzona).

Najczęściej jednak okazuje się, że przynajmniej jedno z wejść ma niepoprawny stan 0. Jeśli jest to tylko jedno z wejść, jak w pokazanym poniżej przykładzie, to wystarczy skoncentrować się na fragmencie układu sterującym tym wejściem. Poszukiwanie problemu w pozostałych fragmentach jest niepotrzebne.



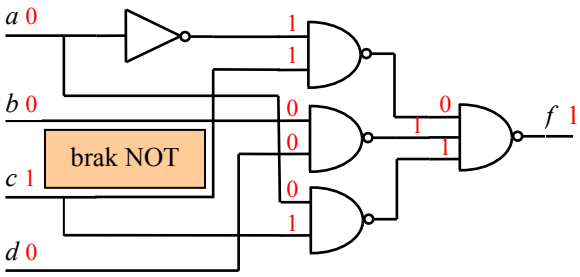
Może się oczywiście okazać (w najgorszym przypadku), że wszystkie wejścia mają niepoprawny stan. Jest to dość rzadka sytuacja a sposób postępowania jest podobny jak dla jednego wejścia. Sprawdzanie należy zacząć od fragmentu układu sterującego dowolnym z tych wejść. Po usunięciu błędu (uzyskaniu poprawnego stanu testowanego wejścia) trzeba ponownie sprawdzić również stan pozostałych wejść. Może się bowiem okazać, że po usunięciu problemu poprawił się również stan na pozostałych wejściach. Wynika to z faktu, że zwykle fragmenty układu sterujące różnymi wejściami bramki nie są od siebie niezależne. Jeśli problem występuje nadal, trzeba powtórzyć takie samo postępowanie dla pozostałych fragmentów układu.

Konsekwencją powiązań w układzie jest również to, że usunięcie jednego błędu może spowodować pojawienie się nowych. Jeśli (dla powyższego przykładu) mimo usunięcia problemu z górnym wejściem bramki NAND stan jej wyjścia jest nadal niepoprawny, to trzeba jeszcze raz sprawdzić, czy na pozostałych wejściach są nadal logiczne jedynki.

Po zawężeniu obszaru poszukiwań sposób postępowania jest analogiczny. Dla pokazanego schematu, zawierającego tylko 6 bramek, efekt metodycznego poszukiwania nie jest może zbyt spektakularny, chociaż już tutaj dzięki jednemu prostemu testowi możemy zawęzić obszar poszukiwania problemu do jednej lub najwyżej dwóch bramek. Jednak nawet przy układach zawierających kilkanaście lub kilkadziesiąt bramek często wystarczy tylko kilka kroków, aby uzyskać podobny efekt.

II. Poprawna funkcja logiczna, błąd w schemacie

Załóżmy, że podobny jak wyżej problem występuje dla kombinacji wejść *abcd* = 0010 (spodziewany stan wyjścia 0, obserwowany w układzie 1). Analiza poziomów logicznych na schemacie pokazana jest poniżej:



W tym przypadku okazuje się, że obserwowany (niepoprawny) stan wysoki wyjścia wynika ze schematu. Oznacza to, że schemat nie odzwierciedla wyjściowej tabeli prawdy.

Aby ustalić na którym etapie został popełniony błąd trzeba sprawdzić, co dla testowanej kombinacji wejść wynika z funkcji logicznej. Otrzymujemy:

$$f = \overline{\overline{a}\,c}\;\;\overline{bd}\;\;\overline{ac} = \overline{\overline{0}\,1}\;\;\overline{00}\;\;\overline{01} = \overline{1\;1\;1} = \overline{1} = 0$$

Widać, że z funkcji wynika spodziewany stan wyjścia równy 0. Oznacza to, że schemat nie jest zgodny z funkcją logiczną. Porównując funkcję ze schematem łatwo zauważyć, że niepoprawny stan wejść (dwie jedynki) ma górna dwuwejściowa bramka NAND. Sprawdzenie połączeń pozwala stwierdzić, że błąd polega na pominięciu negacji.

III. Niepoprawna funkcja logiczna

Jeśli okazałoby się, że wartość wynikająca z funkcji logicznej jest zgodna ze schematem, na przykład:

$$\mathrm{f} = \overline{\overline{a}\,\boldsymbol{c}}\;\;\overline{bd}\;\;\overline{ac} = \overline{\overline{0}\,\mathbf{1}}\;\;\overline{00}\;\;\overline{01} = \overline{0\;1\;1} = \overline{0} = 1$$

to dochodzimy do wniosku, że funkcja logiczna jest niepoprawna. Warto wtedy sprawdzić postać funkcji wynikającą wprost z minimalizacji (zwykle nie jest ona taka sama jak postać końcowa). W naszym przykładzie byłoby to:

$$\mathrm{f} = \bar{a}\,\bar{c} + b\,d + a\,c = \bar{0}\,\bar{1} + 0\,0 + 0\,1 = 1\,0 + 0\,0 + 0\,1 = 0 + 0 + 0 = 0$$

Wartość funkcji jest tutaj inna niż w postaci końcowej. Oznacza to błąd popełniony przy przekształceniach funkcji (w pokazanym przykładzie błąd polega na pominięciu negacji przy zmiennej *c*).

Gdyby jednak okazało się, że również funkcja uzyskana wprost z minimalizacji daje niepoprawną wartość, na przykład:

$$\mathrm{f} = \bar{a}\,\boldsymbol{c} + b\,d + a\,c = \bar{0}\,1 + 0\,0 + 0\,1 = 1\,1 + 0\,0 + 0\,1 = 1 + 0 + 0 = 1$$

to trzeba sprawdzić, co dla testowanej kombinacji wejść pokazuje siatka Karnaugh'a:

cd \ *ab*	00	01	11	10
00	1	1	0	0
01	1	1	1	0
11	0	1	1	1
10	0	0	1	1

W tym przypadku widać, że zawartość odpowiedniej komórki siatki Karnaugh'a jest

niezgodna z wartością funkcji, co pokazuje, że popełniony został błąd przy minimalizacji (pominiecie negacji przy zmiennej *c*).

IV. Siatka Karnaugh'a niezgodna z tabelą prawdy

Jeśli zawartość siatki Karnaugh'a jest zgodna z wartością funkcji, to pozostaje tylko możliwość, że siatka Karnaugh'a nie jest zgodna z tabelą prawdy. Błąd taki jest całkiem realny, zwłaszcza przy braku doświadczenia, w związku z charakterystycznym dla siatki opisem wierszy i kolumn kodem Graya.

V. Niepoprawna tabela prawdy

Może się w końcu okazać, że błąd jest już w tabeli prawdy. Ten dość rzadki przypadek nie jest oczywiście możliwy, gdy sprawdzamy działanie układu właśnie na podstawie tabeli prawdy. Może się to jednak zdarzyć wówczas, gdy testowanie układu wykonywane jest bezpośrednio na podstawie opisu słownego problemu (na przykład dla układu czterowejściowego spełniającego funkcję komparatora dwóch liczb dwubitowych).

Układy sekwencyjne

Przykład

Licznik synchroniczny o sekwencji stanów 0 – 3 – 1 – 0 – ...., przerzutniki JK.

t		t+1		J_1	K_1	J_0	K_0
Q_1	Q_0	Q_1	Q_0				
0	0	1	1	1	-	1	-
0	1	0	0	0	-	-	1
1	0	-	-	-	-	-	-
1	1	0	1	-	1	-	0

J_1

Q_1 \ Q_0	0	1
0	1	0
1	-	-

$J_1 = \overline{Q_0}$

K_1

Q_1 \ Q_0	0	1
0	-	-
1	-	1

$K_1 = 1$

J_0

Q_1 \ Q_0	0	1
0	1	-
1	-	-

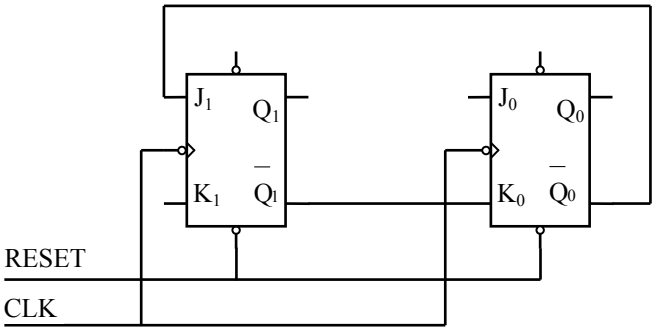
$J_0 = 1$

K_0

Q_1 \ Q_0	0	1
0	-	1
1	-	0

$K_0 = \overline{Q_1}$

Schemat

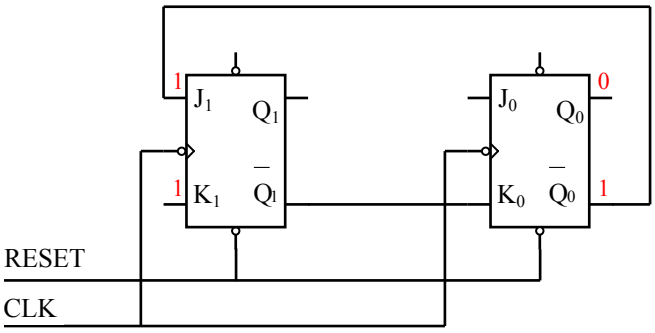


Testowanie układu rozpoczynamy od sprowadzenia układu do stanu początkowego 0 ($Q_1Q_0 = 00$) przy pomocy impulsu na linii RESET. Następnie sprawdzamy czy po kolejnych impulsach zegarowych następuje przejście do odpowiednich stanów, zgodnie z zadaną sekwencją. Testowanie przerywamy przy napotkaniu pierwszej niezgodności.

Załóżmy, że taka niezgodność nastąpiła już po pierwszym sygnale zegarowym: układ powinien przejść do stanu 3 ($Q_1Q_0 = 11$) a przechodzi do stanu 2 ($Q_1Q_0 = 01$). W pierwszej kolejności musimy ustalić, w którym przerzutniku zmiana stanu jest nieprawidłowa. W naszym przypadku niepoprawnie zachowuje się przerzutnik Q_1.

Podobnie jak dla układów kombinacyjnych najpierw musimy ustalić, czy obserwowane zachowanie wynika ze schematu, aby rozstrzygnąć, czy jest on poprawny.

I. Poprawny projekt i schemat, błąd w układzie



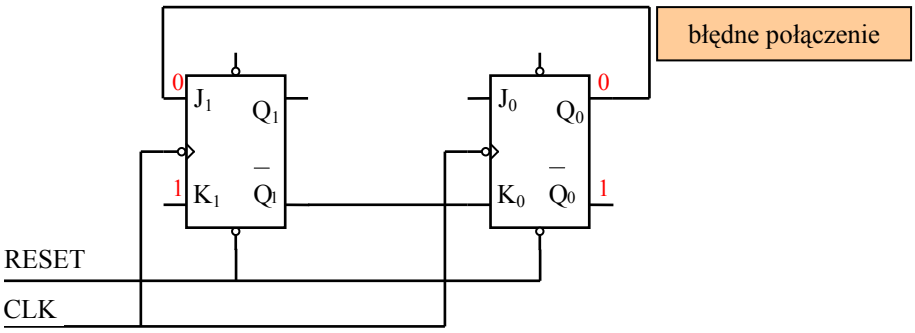
Ze schematu wynika, że wejścia J_1 i K_1 powinny być w stanie 1, spodziewana jest więc zmiana stanu 0 → 1 a więc zachowanie poprawne. W tej sytuacji musimy sprawdzić, jaki jest faktyczny stan wejść J_1 i K_1 w stanie poprzedzającym niewłaściwe przejście, dlatego musimy sprowadzić układ do stanu 0 ($Q_1Q_0 = 00$), czyli stanu bezpośrednio poprzedzającego błędne przejście.

Wiszące wejście K_1 jest w stanie 1, pozostaje więc sprawdzenie stanu wejścia J_1. Jeśli wejście to jest również w stanie 1, oznacza to nieprawidłowe zachowanie przerzutnika. Należy sprawdzić, czy przerzutnik jest prawidłowo zasilany, ewentualnie wymienić przerzutnik na inny.

W przypadku, gdy na wejściu J_1 jest stan 0 (czyli niezgodny ze schematem) należy ustalić przyczynę. Niezgodność może wynikać z błędnego połączenia (na przykład J_1 pomyłkowo podłączone do Q_0 zamiast jego negacji) lub problemu z przerzutnikiem Q_0. Wprawdzie przerzutnik ten prawidłowo zmienia stan, ale możliwe jest takie uszkodzenie, przy którym wyjście $\overline{Q_0}$ nie będzie negacją wyjścia Q_0.

II. Poprawna funkcja logiczna, błąd w schemacie

Załóżmy, że podobnie jak powyżej występuje problem z przejściem ze stanu 0 (zamiast do stanu 3 ($Q_1Q_0 = 11$) układ przechodzi do stanu 2 ($Q_1Q_0 = 01$)). Analiza poziomów logicznych na schemacie pokazana jest poniżej:



W tym przypadku ze schematu wynika, że wejście J_1 jest w stanie 0. Obserwowany (niepoprawny) brak ustawienia przerzutnika wynika więc ze schematu. Oznacza to, że schemat nie odzwierciedla zadanej sekwencji stanów.

Aby ustalić gdzie został popełniony błąd trzeba sprawdzić funkcję dla wejścia J_1:

$$J_1 = \overline{Q_0} = \overline{0} = 1$$

Widać, że z funkcji wynika spodziewany stan J_1 równy 1, co oznacza, że schemat nie jest zgodny z funkcją. Łatwo zauważyć, że na schemacie wejście J_1 zostało pomyłkowo podłączone do Q_0 zamiast do $\overline{Q_0}$ (jak wynika z funkcji).

Pozostałe przypadki są podobne jak dla układów kombinacyjnych:

- niepoprawna funkcja logiczna dla wejścia przerzutnika (błąd w minimalizacji),
- niepoprawna zawartość siatki Karnaugh'a (błędnie określony stan wejścia dla zadanej zmiany stanu przerzutnika, pomyłka przy wpisywaniu),
- błąd w określeniu sekwencji stanów.

Oczywiście niepoprawna zmiana stanu może wystąpić dla więcej niż jednego przerzutnika. Należy wówczas przeprowadzić podobną analizę dla kolejnych przerzutników.

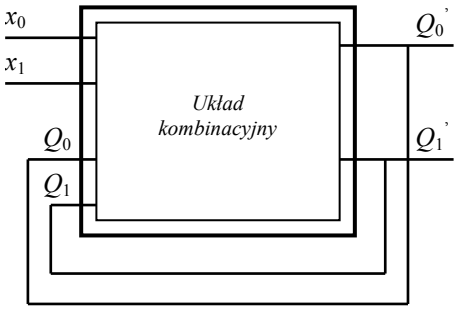
W przedstawionym przykładzie funkcje logiczne dla wejść przerzutników są bardzo proste i nie wymagają użycia żadnych bramek (wykorzystane są zanegowane wyjścia przerzutników). W bardziej złożonym układzie wejścia przerzutników mogą być sterowane z wyjścia pewnego układu złożonego z bramek. Sposób postępowania jest wówczas podobny do omówionego wcześniej dla układów kombinacyjnych.

Automat asynchroniczny

Uruchamianie automatu asynchronicznego często sprawia problemy ze względu na sprzężenie zwrotne, utrudniające testowanie układu w przypadku błędnego działania. Poniżej pokazane zostanie podejście, dzięki któremu można łatwo znaleźć i usunąć problemy w układzie.

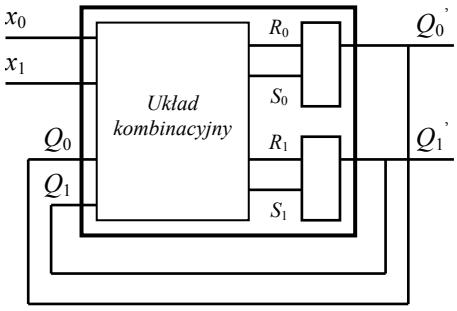
Przykład

Załóżmy, że zrealizowany został automat asynchroniczny o dwóch wejściach x_0 i x_1 i liczbie stanów nie większej niż 4 (stąd dwa elementy pamięciowe Q_0 i Q_1). Dwa możliwe warianty realizacji pokazane są poniżej:



Wersja ze sprzężeniem zwrotnym

$Q_0' = f(x_0, x_1, Q_0, Q_1)$
$Q_1' = f(x_0, x_1, Q_0, Q_1)$



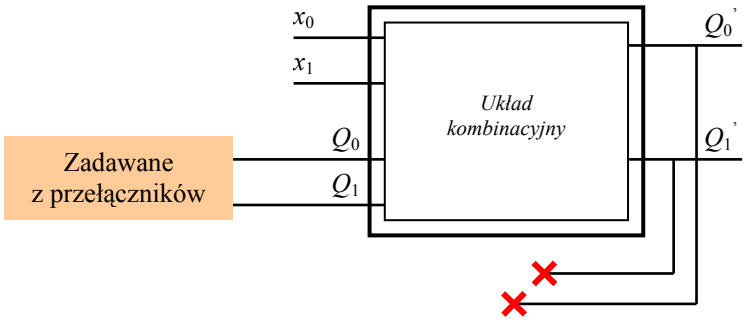
Wersja z przerzutnikami RS

$R_0 = f(x_0, x_1, Q_0, Q_1)$
$S_0 = f(x_0, x_1, Q_0, Q_1)$
$R_1 = f(x_0, x_1, Q_0, Q_1)$
$S_1 = f(x_0, x_1, Q_0, Q_1)$

Zależnie od wariantu syntezy w wyniku minimalizacji otrzymujemy dwie lub cztery zwykłe funkcje kombinacyjne. W wariancie pierwszym sprzężenie zwrotne jest wyraźnie widoczne, wartość funkcji zależy „od samej siebie". Różne oznaczenia sygnałów, na przykład Q_0 i Q_0' służą tylko do rozróżnienia miejsca ich doprowadzenia (wejście lub wyjście), ale w rzeczywistości jest to ten sam sygnał. W wariancie z przerzutnikami sprzężenie zwrotne, chociaż ukryte, również występuje. Ponieważ przerzutniki RS są asynchroniczne, to zmiany sygnałów Q_0' i Q_1', poprzez układ kombinacyjny, wpływają na wejścia przerzutników i mogą powodować kolejne zmiany stanów Q_0' i Q_1'.

Konsekwencją obecności sprzężenia zwrotnego i braku sygnału zegarowego (układ asynchroniczny) jest to, że przy ewentualnych błędach w syntezie lub przy łączeniu funkcji kombinacyjnych układ może zacząć „żyć własnym życiem". Efekty zewnętrzne mogą być różne, na przykład całkowity brak reakcji na zmianę sygnałów wejściowych x_0 i x_1, nieprawidłowa sekwencja stanów lub w skrajnym przypadku ciągłe generowanie zapętlonej sekwencji stanów, bez osiągania stanu stabilnego.

Jest dość proste wyjście z tej pozornie beznadziejnej sytuacji. Aby przeprowadzić testowanie, należy rozewrzeć pętlę sprzężenia zwrotnego i sygnały Q_0 i Q_1 zadawać przy pomocy przełączników, co pokazano na poniższym rysunku:



Dzięki rozwarciu pętli sprzężenia zwrotnego otrzymujemy zwykły układ kombinacyjny. Możemy wtedy zadać dowolną z 16 kombinacji sygnałów wejściowych x_1 x_0 Q_1 Q_0 i sprawdzić, czy wartości funkcji Q_0' i Q_1' (które stały się teraz w pełni „sterowalne") są zgodne z ich tabelami prawdy. Dla wariantu z przerzutnikami RS postępowanie jest podobne z tym, że trzeba sprawdzić wartości funkcji dla wejść przerzutników, czyli R_0, S_0, R_1 i S_1.

Co jest potrzebne aby znaleźć problem w układzie

Z przedstawionych wyżej sposobów poszukiwania problemu w przypadku, gdy połączony układ działa nieprawidłowo wynika kilka ważnych uwag:

- jeśli połączony układ nie działa (z rożnych przyczyn - zależnych lub nie od samego projektu), to trzeba ustalić źródło problemu. Jak już wspomniano, nie wolno rozłączać takiego układu i łączyć ponownie (z nadzieją, że tym razem się uda),
- znalezienie ewentualnego błędu w projekcie jest możliwe tylko wtedy, gdy dysponujemy projektem kompletnym, czyli zawierającym wszystkie kroki syntezy układu (a nie tylko końcowym efektem syntezy, czyli schematem). Tylko wtedy można szybko ustalić miejsce popełnienia błędu (siatka Karnaugh'a, minimalizacja, przekształcenia funkcji logicznej, przejście z funkcji do schematu),
- schemat powinien być czytelny i zgodny z tym, co zostało połączone. W przypadku, gdy konieczne są poprawki lub modyfikacje (na przykład konwersje typów bramek) należy na bieżąco aktualizować schemat,
- trzeba zapewnić możliwość szybkiej i sprawnej identyfikacji na schemacie bramek użytych w połączonym układzie. Bez spełnienia tego warunku metodyczne poszukiwanie błędu w układzie jest praktycznie niemożliwe,
- aby sprawnie przeprowadzać testy trzeba oczywiście wiedzieć jak działają bramki logiczne i przerzutniki.

4. Sprawozdanie

Sprawozdanie powinno być dokumentacją wykonanego i uruchomionego projektu. W przypadku wątpliwości, co powinno zawierać sprawozdanie najlepiej wyobrazić sobie, czego oczekiwałby ktoś, chcący prześledzić tok rozumowania autora a następnie połączyć i uruchomić zaprojektowany układ. Większość sprawozdania to po prostu projekt, którym powinno się dysponować już w czasie wykonywania ćwiczenia.

Zawartość sprawozdania

- dane organizacyjne (autorzy, temat ćwiczenia, data wykonania),
- zadania do wykonania, szczegółowe założenia projektu,
- poszczególne kroki projektu (na przykład dla syntezy układu kombinacyjnego: tabela prawdy, minimalizacja, ewentualne przekształcenia funkcji logicznych, schemat układu),
- uwagi i wnioski (co zostało wykonane a co nie, ewentualne poprawki i optymalizacje, propozycje lepszych rozwiązań itp.).

Uwagi szczegółowe

- sprawozdanie może być wykonane odręcznie (musi być czytelne) lub na komputerze. Elementy, które wymagają pracochłonnej edycji (na przykład negacje zmiennych czy obszary w siatkach Karnaugh'a) można uzupełnić ręcznie, już na wydruku,
- nie należy stosować nadmiernych skrótów (na przykład przy przekształceniach wyrażeń logicznych) tak, aby możliwe było prześledzenie wszystkich etapów wyprowadzeń,

- trzeba pamiętać o konsekwentnych oznaczeniach sygnałów logicznych (sygnał logiczny nie może mieć innej nazwy na początku i na końcu projektu),
- trzeba koniecznie pamiętać o zaznaczaniu obszarów minimalizacji w siatkach Karnaugh'a, bez tego są one bezwartościowe,
- wejścia, wyjścia i przerzutniki na schemacie powinny być oznaczone, zgodnie z wcześniejszymi krokami projektu,
- schemat powinien zawierać układ, który został połączony (lub mógłby być połączony) w laboratorium. Nie może być to schemat równoważny logicznie, którego nie da się wprost połączyć i zweryfikować w laboratorium (bo na przykład zawiera on bramki AND i OR niedostępne w używanych modułach logicznych).

Ważna uwaga

Jeśli któreś z zadań nie zostało połączone i/lub uruchomione, nie oznacza to, że może być pominięte w sprawozdaniu. Przeciwnie, należy wtedy wyjątkowo dokładnie sprawdzić projekt, aby nie było w nim ewidentnych błędów. Z uwag i wniosków powinno jasno wynikać, które układy zostały uruchomione i zweryfikowane a które nie.