

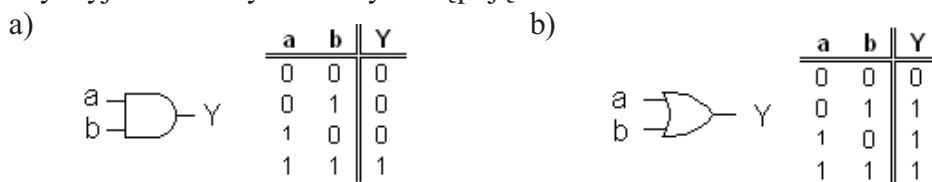
4. MATERIAŁ NAUCZANIA

4.1 Podstawowe układy cyfrowe – rodzaje, parametry, zastosowanie.

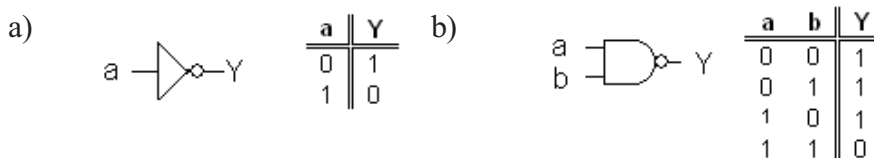
4.1.1 Materiał nauczania

Bramki

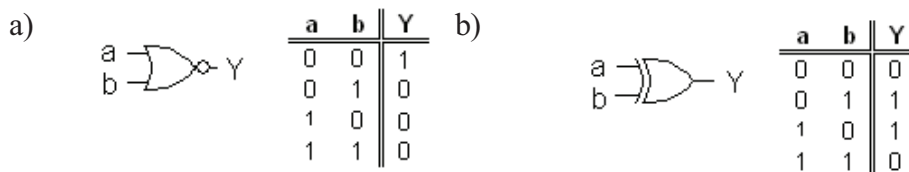
Bramką (funkctorem) nazywamy podstawowy układ kombinacyjny realizujący funkcję logiczną jednej lub kilku zmiennych. Do opisu działania bramek logicznych stosuje się tablice prawdy, zawierające zbiór wszystkich kombinacji sygnałów wejściowych oraz odpowiadające im sygnały wyjściowe. Wyróżniamy następujące bramki:



Rys. 1. Bramka AND (a) i bramka OR (b) [1, s.34]



Rys. 2. Bramka NOT (a) i bramka NAND (b) [1, s.34 – 35]



Rys. 3. Bramka NOR (a) i bramka ExOR (b) [1, s.35 – 36]

Przykładowe układy scalone zawierające bramki to: 7400 (4 dwuwejściowe bramki NAND), 7410 (3 trójwejściowe NAND), 7432 (4 dwuwejściowe bramki OR).

Każda bramka posiada szereg podstawowych parametrów determinujących warunki jej działania. Do parametrów tych zaliczamy:

- Czas propagacji t_p [ns] sygnału przez bramkę, czyli czas odpowiedzi układu na wejściowy sygnał sterujący. Jest on podstawowym czynnikiem wpływającym na szybkość działania układów cyfrowych zbudowanych z danej klasy układów scalonych. Czas propagacji definiuje się jako odstęp czasu między zboczem impulsu wejściowego (dodatnim lub ujemnym, zależnie od przyjętych założeń) a zboczem na wyjściu bramki, będącym jego odpowiedzią. Pomiaru tego dokonuje się przy określonym poziomie napięcia, najczęściej przy tzw. progu logicznym układu, tzn. wartości napięcia przy której następuje przełączenie układu z niskiego do wysokiego stanu logicznego bądź odwrotnie. Czasy propagacji przy przełączaniu się bramki ze stanu niskiego do wysokiego (t_{PLH}) oraz ze stanu wysokiego do niskiego (t_{PHL}) są różne, dlatego czas propagacji t_p jest ich średnią arytmetyczną. Produkowane obecnie układy mają czas propagacji wynoszący przeważnie od kilku do kilkudziesięciu nanosekund.

- Straty mocy P_S [mW] – parametr ten określa się w sposób uproszczony, jako wartość iloczynu napięcia zasilania układu i średniego prądu pobieranego przez układ ze źródła zasilania. Wartość P_S zależy od obciążenia oraz częstotliwości przełączania. Im wyższa częstotliwość przełączania lub do układu podpięte jest większe obciążenie tym większa moc jest pobierana ze źródła zasilania.
- Maksymalna częstotliwość pracy (przełączania)– określa maksymalną szybkość zmian na wejściu, na które układ jest w stanie prawidłowo zareagować. Wyraża się ją w MHz.
- Margines zakłóceń – określa dopuszczalne amplitudy sygnału zakłócającego, które nie powodują jeszcze nieprawidłowej pracy układu, czyli innymi słowy określa on odporność układu na zakłócenia. Jest on określony dla obydwu stanów logicznych na wejściu układu scalonego:

$$M_{Lmin} = U_{IL\ max} - U_{OL\ max} [V]$$

$$M_{Hmin} = U_{OH\ min} - U_{IH\ min} [V]$$

$U_{OL\ max}$ – maksymalne napięcie wyjściowe bramki w stanie niskim,

$U_{OH\ min}$ – minimalne napięcie wyjściowe w stanie wysokim,

$U_{IL\ max}$ – maksymalne napięcie wejściowe w stanie niskim,

$U_{IH\ min}$ – minimalne napięcie wejściowe w stanie wysokim,

Dla układów TTL typowe wartości marginesów zakłóceń wynoszą odpowiednio:

$$M_{Lmin} = 0,8V - 0,4V = 0,4V$$

$$M_{Hmin} = 2,4V - 2V = 0,4V$$

- Obciążalność określa maksymalną liczbę innych elementów tego samego typu, które mogą być z określonego wyjścia prawidłowoysterowane. Jako definicję obciążalności przyjmuje się odpowiednio w stanie niskim i wysokim:

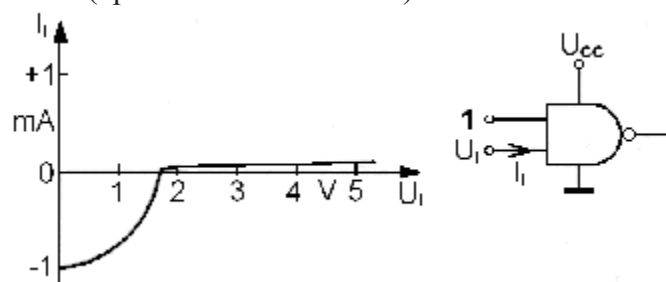
$$N_L = \frac{I_{OL\ min}}{I_{IL\ max}} = \frac{16\ mA}{1,6\ \mu A} = 10\ \text{bramek}$$

Tak samo definiuje się obciążalność wyjścia w stanie wysokim:

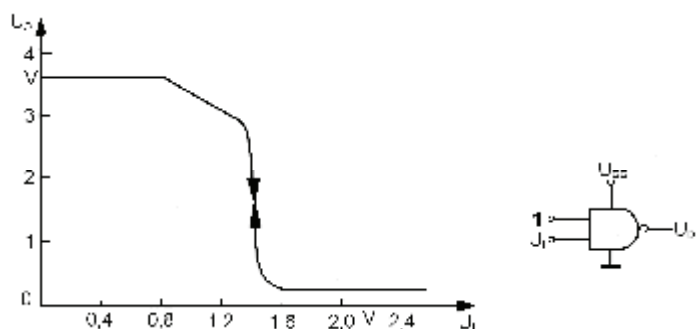
$$N_H = \frac{I_{OH\ min}}{I_{IH\ max}} = \frac{0,8\ mA}{40\ \mu A} = 20\ \text{bramek}$$

Pamiętając o tym, że na wyjściu bramki może być „0” jak i „1” jako obciążalność zatem przyjmujemy mniejszą z tych wartości.

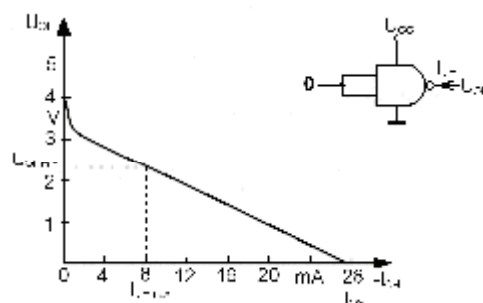
- Charakterystyki bramek (np. bramki NAND TTL):



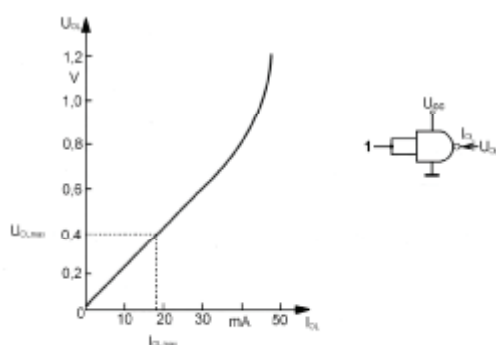
Rys. 4. Charakterystyka wejściowa bramki NAND (TTL) [1, s.86]



Rys. 5. Charakterystyka przejściowa bramki NAND (TTL) [1, s.84]



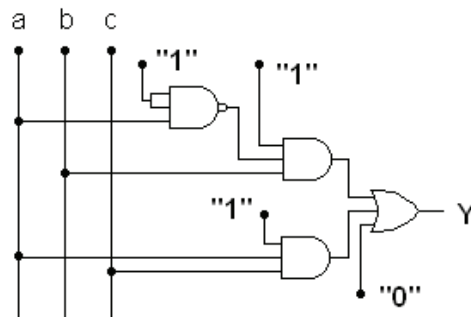
Rys. 6. Charakterystyka wyjściowa bramki NAND (TTL) w stanie wysokim na wyjściu [1, s.88]



Rys. 7. Charakterystyka wyjściowa bramki NAND (TTL) w stanie niskim na wyjściu [1, s.87]

Często zdarza się, że aby zrealizować funkcję logiczną na bramkach wystarczą np. dwa wejścia, podczas gdy mamy do dyspozycji bramki 3-wejściowe. Powstaje wówczas problem, co zrobić z niewykorzystanym wejściem. Najlepszym rozwiązaniem dla obu rodzin technologicznych: CMOS i TTL jest dołączenie ich do stanu wysokiego lub niskiego, zależnie od realizowanej funkcji. Istnieje jeszcze jedna możliwość, mianowicie zwarcie niewykorzystanego wejścia z innym. W obydwu przypadkach należy dokładnie przeanalizować układ i zwrócić uwagę, czy nie zmieni to wartości funkcji logicznej. W rodzinie TTL niewykorzystywane wejścia są w stanie wysokim (logiczna „1” – tzw. „wisząca jedynka”). W technologii CMOS pozostawienie niewykorzystanego wejścia nie podłączonego może prowadzić do uszkodzenia układu cyfrowego.

Załóżmy, że chcemy zrealizować funkcję logiczną $y = \bar{a} \cdot b + a \cdot c$ przy pomocy bramek AND, OR i NAND. Rozwiązanie tego problemu przedstawiono na rys. 8.



Rys. 8. Układ realizujący funkcję logiczną

Trójwejściowa bramka NAND posłużyła do stworzenia negacji sygnału „a”. Bramka NAND wykonuje operację mnożenia z negacją, tak więc, aby nie zmienić wyniku operacji, na niewykorzystane wejścia musimy podać logiczną „1”. W przypadku podania logicznego „0”, niezależnie od pozostałych sygnałów na wyjściu bramki otrzymamy „1”.

Podobnie się ma sytuacja z dwoma 3-wejściowymi bramkami AND, w których wykorzystujemy tylko dwa wejścia. Aby nie zmienić wartości funkcji na wyjściu bramki mnożącej, musimy podać na niewykorzystane wejścia logiczne „1”. Dokładnie odwrotnie ma się sytuacja z bramką OR. Aby nie zmienić wartości na wyjściu tej bramki, należy na pozostałe wolne wejścia podać logiczne „0”. Niezależnie od rodzaju bramki, niewykorzystane wejścia możemy zewrzeć do innego wejścia, co też nie zmieni wartości funkcji na wyjściu bramki.

Układy kombinacyjne.

Układem kombinacyjnym nazywamy układ logiczny, dla którego w dowolnej chwili czasu stan na wyjściach układu zależy tylko od aktualnego stanu na jego wejściach, a nie zależy od stanów poprzednich.

Zaprojektujemy układ kombinacyjny sterujący ruchem wagonika transportującego piasek z punktu A do punktu B. W punkcie A następuje załadunek wagonika, następnie materiał transportowany jest do punktu B, gdzie dokonywany jest rozładunek. Załadunek i rozładunek odbywają się automatycznie. Po rozładowaniu wagonik wraca do punktu A, gdzie oczekuje na ponowne załadowanie. Położenie wagonika wykrywane jest przez czujniki A i B (czujniki wystawiają sygnał „1”, gdy wagonik znajduje się w ich pobliżu). W wagoniku zainstalowany jest również czujnik ciężaru C (równy „0” – wagonik pusty, „1” – wagonik załadowany).

Projektowany układ ma za zadanie sterować ruchem wagonika za pomocą 2 sygnałów L i P wg poniższego opisu:

Tabela 1. Opis sterowania wagonikiem

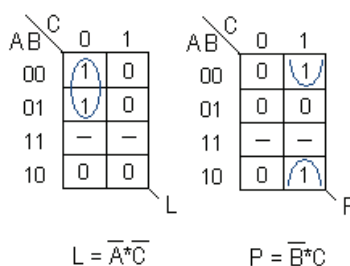
L	P	Działanie
0	0	Wagonik zatrzymany
0	1	ruch w prawo
1	0	ruch w lewo
1	1	stan zabroniony

Rozwiązanie zadania rozpoczynamy od przeanalizowania jego treści i uzupełnienia tablicy prawdy. W tablicy znajdują się wszystkie kombinacje sygnałów wejściowych, z opisem sytuacji, której dotyczą. W przedstawionej poniżej tabeli pominięto kombinacje zmiennych wejściowych, które nie wystąpią (np. wagonik nie może być jednocześnie w położeniu A i B).

Tabela 2. Tablica prawdy

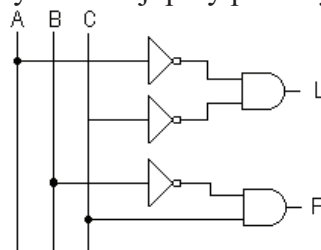
A	B	C	L	P	Opis
1	0	0	0	0	Postój, oczekiwanie na załadunek
1	0	1	0	1	Wagonik załadowany, ruch w prawo
0	0	1	0	1	Ruch w prawo
0	1	1	0	0	Postój, oczekiwanie na rozładunek
0	1	0	1	0	Wagonik rozładowany, ruch w lewo
0	0	0	1	0	Ruch w lewo

Następnym etapem jest przeniesienie tablicy prawdy do siatek Karnaugh'a, zakreslenie i wyznaczenie funkcji opisujących sygnały L i P.



Rys. 9. Siatki Karnaugh'a

Przystępujemy do zrealizowania zapisanych funkcji przy pomocy bramek:



Rys. 10. Przykładowa realizacja funkcji sterowania wagonikiem

Lokalizacja uszkodzeń

Uszkodzeniem układu cyfrowego nazywamy defekt powodujący nieprawidłowe działanie tego układu. W wyniku uszkodzenia na wyjściu układu cyfrowego pojawia się niewłaściwa informacja, czyli błąd. Przyczyną uszkodzenia może być np.: zwarcie lub rozwarcie ścieżek czy też przebicie elementów. Patrząc od strony logicznej możliwe są następujące rodzaje uszkodzeń: sklejenie (czyli zwarcie) z jakimś sygnałem lub przerwa.

Tak więc dla każdego elementu cyfrowego istnieją następujące możliwości wystąpienia uszkodzenia:

- sklejenie wejścia z „0”,
- sklejenie wejścia z „1”,
- przerwa na wejściu,
- sklejenie wyjścia z „0”,
- sklejenie wyjścia z „1”,
- przerwa na wyjściu.

Wykrycie uszkodzenia, czyli zauważenie, że układ przy zadanym stanie wejść działa w sposób różny od założonego nie jest trudną operacją. Dużo więcej problemów sprawia lokalizacja oraz identyfikacja uszkodzenia.

Najprostszym narzędziem do badania stanów logicznych jest sonda logiczna, bardzo przydatna przy testowaniu poprawności działania układu. Jest to połączona przez rezystor do masy dioda LED. Budowę takiej sondy przedstawia poniższy rysunek:



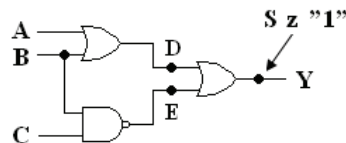
Rys. 11. Sonda logiczna

Jeżeli sonda logiczna (punkt A) dotyka punktu, w którym występuje stan logiczny „1”, wówczas przez diodę LED płynie prąd, o wartości ustalonej przez rezystor R, powodując świecenie diody. W przypadku dotknięcia punktu będącego w stanie „0” dioda nie świeci.

Sprawdzenie poprawności działania zbudowanego układu cyfrowego polega na tym, że na wejścia podaje się pewien wektor wymuszeń sprawdzając jednocześnie sondą logiczną czy reakcje na wyjściach są poprawne (zgodne z realizowaną przez układ funkcją). Jeżeli zostanie wykryty błąd, to znaczy, że w danym układzie powstało uszkodzenie, które trzeba zlokalizować i usunąć. Można do tego celu użyć tzw. metody „śledzenia wstecz”.

Metoda ta polega na sprawdzaniu stanów logicznych występujących na wejściach i wyjściach poszczególnych elementów. Sprawdzanie to jest realizowane wstecz, tzn. od wyjścia, na którym wykryto błąd, poprzez kolejne elementy logiczne wzdłuż linii propagacji błędu, aż do natrafienia na poprawny stan logiczny. Znajdując w ten sposób granicę poprawnego działania układu, można przeanalizować i wyeliminować powstałe uszkodzenia. Aby wykryć określone uszkodzenie trzeba odpowiednio dobrać sygnały na wejściu układu, ponieważ może się zdarzyć, że układ jest uszkodzony, ale stan na jego wyjściu jest zgodny z oczekiwanym. Sytuacja taka może zaistnieć w momencie, gdy na jednym wejściu bramki OR wystąpiło zwarcie do „1”, podczas gdy na drugim jest stan logiczny „1”.

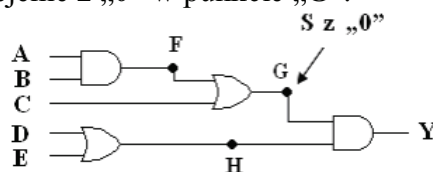
Dlatego też, aby wykryć określony rodzaj uszkodzenia trzeba zadać takie stany na wejściach układu, aby w omawianym punkcie uzyskać stan przeciwny. Na przykład chcąc sprawdzić czy w punkcie A nie ma sklejenia z „0”, trzeba wymusić w nim stan logiczny „1”. Czasami może się zdarzyć, że struktura układu uniemożliwi wykrycie pewnego rodzaju uszkodzenia.



Rys. 12. Układ, w którym nie jest możliwe wykrycie uszkodzenia typu sklejenie wyjścia z „1”

Aby wykryć na wyjściu Y uszkodzenie typu sklejenie z „1” (S z „1”) trzeba wymusić w tym punkcie stan przeciwny, czyli logiczne „0”. To oznacza, że w punktach D i E też musi być stan niski. Żeby w punkcie D otrzymać stan „0”, to konieczne jest wymuszenie następującego stanu wejść: $A = 0$ i $B = 0$. Aby otrzymać stan „0” w punkcie E, konieczne jest wymuszenie na wejściach następującego stanu wejść: $B = 1$ oraz $C = 1$. Wejście B nie może jednocześnie znajdować się w stanie niskim i wysokim, stąd też wniosek, że w powyższym układzie nie istnieje taki wektor wejściowy, którego podanie umożliwi wykrycie uszkodzenia typu sklejenie z jedynką w punkcie Y.

W układzie przedstawionym na rysunku poniżej należy podać wektor wejść, który umożliwi wykrycie uszkodzenia typu sklejenie z „0” w punkcie „G”.



Rys. 13. Układ, w którym należy wykryć uszkodzenie

Chcąc znaleźć uszkodzenie typu sklejanie z „0” trzeba wymusić w pożądanym miejscu stan przeciwny, czyli w tym przypadku logiczną „1”. Aby w punkcie G otrzymać logiczną „1” musi być spełniony przynajmniej jeden z dwóch warunków: w punkcie F powinien być stan wysoki, co jest równoznaczne z tym, że $A = 1$ i $B = 1$ lub wejście C powinno być w stanie wysokim. Uwzględniając również fakt, że stan wejść D i E w żaden sposób nie wpływa na wartość funkcji w punkcie G można wyznaczyć następujące wektory, które wykryją zadane uszkodzenie:

A	B	C	D	E
x	x	1	x	x
1	1	x	x	x

Rys. 14. Wektory pozwalające wykryć uszkodzenie (x oznacza stan nieokreślony – „0” lub „1”)

4.1.2 Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Czy potrafisz narysować siatki Karnaugh’a dla bramek AND, NOT, OR, NAND, ExOR?
2. Jakie parametry charakteryzują bramki?
3. Czy potrafisz naszkicować charakterystykę wejściową, przejściową i wyjściową bramki NAND?
4. Czy potrafisz zrealizować prostą funkcję logiczną przy pomocy bramek?
5. Czy potrafisz zaprojektować prosty układ kombinacyjny?
6. Czy potrafisz zlokalizować uszkodzenia w układzie cyfrowym?

4.1.3 Ćwiczenia

Ćwiczenie 1

Sprawdź poprawność działania bramek.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) zaproponować i narysować schematy do sprawdzenia poprawności działania następujących bramek: AND, OR, NOT, NAND, NOR oraz ExOR,
- 2) połączyć układ zgodnie ze sporządzonym schematem,
- 3) sprawdzić działanie wymienionych wyżej bramek uzupełniając w trakcie pomiarów tabelę prawdy dla poszczególnych bramek,
- 4) dokonać oceny poprawności wykonanego ćwiczenia,
- 5) sformułować i przedstawić wnioski.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca bramki,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych,
- arkusze białego papieru A4.

Ćwiczenie 2

Zrealizuj funkcję logiczną $f(x, y, z) = x\bar{y} + yz$ przy pomocy bramek wykonanych w technologii TTL lub CMOS.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) zaproponować, w jaki sposób można by zrealizować funkcję $f(x, y, z) = x\bar{y} + yz$ dysponując układami 7420,
- 2) zaproponować, co zrobić z niewykorzystanymi wejściami bramek,
- 3) połączyć układ wg sporządzonego schematu,
- 4) sprawdzić, czy zaproponowany układ realizuje założoną funkcję,
- 5) dokonać oceny poprawności wykonanego ćwiczenia,
- 6) sformułować i przedstawić wnioski.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca bramki,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych,
- arkusze białego papieru A4.

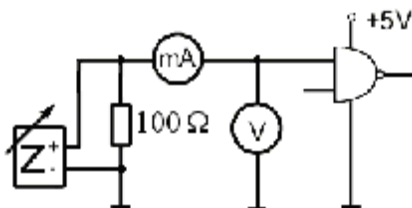
Ćwiczenie 3

Zbadać charakterystyki bramki NAND.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) połączyć układ pomiarowy wg poniższego schematu pomiarowego,



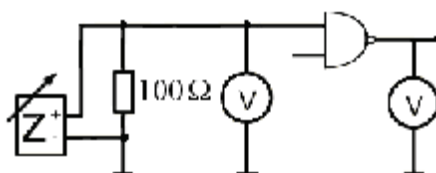
Schemat pomiarowy do badania charakterystyki wejściowej bramki NAND

- 2) wykonać pomiary „punkt po punkcie” dla około 10 – 15 wartości napięcia ($0V \leq U_{we} \leq 5V$), tak aby zaobserwować zależność $I_{we} = f(U_{we})$, wyniki zapisać w tabeli

Tabela pomiarowa – charakterystyka wejściowa

Lp.	I_{we} [A]	U_{we} [V]
1.		
...		
15.		

- 3) wykreślić charakterystykę wejściową bramki na podstawie wyników pomiarów,
- 4) połączyć układ pomiarowy wg poniższego schematu pomiarowego,



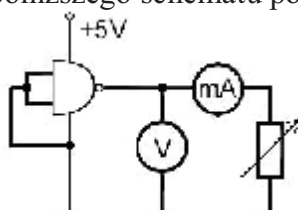
Schemat pomiarowy do badania charakterystyki przejściowej bramki NAND

- 5) wykonać pomiary „punkt po punkcie” dla około 10 – 15 wartości napięcia ($0V \leq U_{we} \leq 5V$), tak, aby zaobserwować zależność $U_{wy} = f(U_{we})$, wyniki zapisać w tabeli,

Tabela pomiarowa – charakterystyka przejściowa

Lp.	U_{we} [V]	U_{wy} [V]
1.		
...		
15.		

- 6) wykreślić charakterystykę przejściową bramki na podstawie wyników pomiarów,
7) połączyć układ pomiarowy wg poniższego schematu pomiarowego,



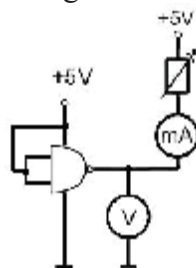
Schemat pomiarowy do badania charakterystyki wyjściowej bramki NAND w stanie wysokim

- 8) wykonać pomiary „punkt po punkcie” dla około 10 – 15 wartości prądu (max 30mA), tak aby zaobserwować zależność $U_{OH} = f(I_{OH})$, wyniki zapisać w tabeli,

Tabela pomiarowa – charakterystyka wyjściowa bramki NAND w stanie wysokim na wyjściu

Lp.	I_{wy} [A]	U_{wy} [V]
1.		
...		
15.		

- 9) wykreślić charakterystykę wyjściową bramki na podstawie wyników pomiarów,
10) połączyć układ pomiarowy wg poniższego schematu pomiarowego,



Schemat pomiarowy do badania charakterystyki wyjściowej bramki NAND w stanie niskim

- 11) wykonać pomiary „punkt po punkcie” dla około 10 – 15 wartości prądu (max 30mA), tak aby zaobserwować zależność $U_{OL} = f(I_{OL})$, wyniki zapisać w tabeli,

Tabela pomiarowa – charakterystyka wyjściowa bramki NAND w stanie niskim na wyjściu

Lp.	I_{wy} [A]	U_{wy} [V]
1.		
...		
15.		

- 12) wykreślić charakterystykę wyjściową bramki na podstawie wyników pomiarów,
13) sformułować i przedstawić wnioski.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

- Wyposażenie stanowiska pracy:
- zadajniki i wskaźniki stanów logicznych, makieta zawierająca bramki,
 - zasilacz laboratoryjny, mierniki uniwersalne, rezystor nastawny,
 - dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 4

Projektowanie układów kombinacyjnych.

Zaprojektować układ kombinacyjny sterujący pracą dźwigu – suwnicy pracującego wg następującego algorytmu:

Dźwig przenosi bloczki betonowe z punktu A do punktu B. Zaczeplenie bloczka (załadunek) oraz odzecieplenie bloczka (rozładunek) wykonywane są przez pracowników, co sygnalizowane jest odpowiednim stanem czujnika C ($C=1$, bloczek zaczepiony, $C=0$ bloczek odzecieplony). Obecność dźwigu w okolicach punktu A sygnalizowana jest pojawieniem się na wyjściu czujnika A stanu „1”, podobnie jak z czujnikiem B, który sygnalizuje pojawienie się dźwigu w punkcie B. Sygnały sterujące pracą dźwigu to L i P (odpowiedni ruch suwnicy w lewo i w prawo).

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinienś:

- 1) zaproponować tablicę prawdy opisującą działanie dźwigu,
- 2) przenieść zawartość tablicy prawdy do siatek Karnaugh'a, zakreślić i spisać funkcję logiczną,
- 3) zrealizować funkcję przy pomocy dostępnych układów zawierających bramki,
- 4) sprawdzić działanie zaprojektowanego układu kombinacyjnego,
- 5) sformułować i przedstawić wnioski.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych, makieta zawierająca bramki, przewody połączeniowe, arkusze białego papieru A4.

4.1.4 Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) narysować funktory podstawowych bramek i zapisać funkcje realizowane przez te bramki?	<input type="checkbox"/>	<input type="checkbox"/>
2) wyjaśnić wpływ podstawowych parametrów na działanie bramki?	<input type="checkbox"/>	<input type="checkbox"/>
3) wykreślić na podstawie wyników pomiarów charakterystykę wejściową, przejściową i wyjściową bramki NAND?	<input type="checkbox"/>	<input type="checkbox"/>
4) narysować i zmontować schemat pomiarowy do badania charakterystyki wejściowej, przejściowej, wyjściowej bramki NAND?	<input type="checkbox"/>	<input type="checkbox"/>
5) określić na podstawie wyników pomiarów czy badana bramka działa prawidłowo?	<input type="checkbox"/>	<input type="checkbox"/>
6) zaprojektować i zmontować prosty układ kombinacyjny sterujący pracą wybranego urządzenia?	<input type="checkbox"/>	<input type="checkbox"/>
7) wykryć i zlokalizować uszkodzenie w układzie cyfrowym?	<input type="checkbox"/>	<input type="checkbox"/>

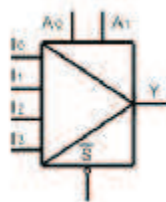
4.2 Układy komutacyjne. Przetworniki kodów.

4.2.1 Materiał nauczania

Multiplekser

Multiplekser jest to układ kombinacyjny zbudowany z n wejść adresowych, 2^n wejść informacyjnych, oraz jednego wyjścia. Układ ten służy do łączenia (w sensie przepływu informacji) jednego z wejść informacyjnych z wyjściem. Wyboru, które wejście ma być połączone z wyjściem dokonuje się poprzez podanie adresu odpowiedniego wejścia na wejścia adresowe.

Poniżej przedstawiono symbol przykładowego multipleksera typu '153, składającego się z czterech wejść informacyjnych ($I_0 - I_3$), dwóch wejść adresowych (A_0 i A_1), wejścia strobującego (\bar{S}) oraz jednego wyjścia (Y).

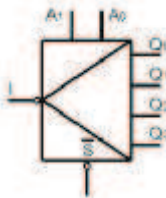


Rys. 15. Symbol graficzny multipleksera o dwóch wejściach adresowych

Aktualnie produkowane są układy multiplekserów o 1, 2, 3 i 4 wejściach adresowych. Mogą one mieć wyjścia dwustanowe (stan wysoki i stan niski) lub trójstanowe (dodatkowo stan wysokiej impedancji). Większość multiplekserów (o wyjściu dwustanowym) ma dodatkowe wejście sterujące zwane wejściem strobującym. W podanym powyżej przykładzie wejście to jest aktywne w stanie niskim, co oznacza, że podanie logicznego „0” na to wejście uaktywnia multiplekser, a podanie „1” powoduje, że na wyjściu niezależnie od stanu wejść informacyjnych będzie stan niski. Przykładem układu scalonego zawierającego multiplekser 1 z 8 jest układ '151, natomiast 1 z 16 – '150.

Demultiplekser.

Demultiplekser jest to układ kombinacyjny składający się z jednego wejścia informacyjnego, 2^n wyjść oraz n wejść adresowych. Układ ten służy do łączenia (w sensie przepływu informacji) wejścia informacyjnego z jednym z wyjść. Poniżej przedstawiono przykład demultipleksera typu '155, składającego się z wejścia informacyjnego (I), dwóch wejść adresowych (A_0 i A_1), wejścia strobującego (\bar{S}) oraz czterech wyjść ($Q_3 - Q_0$). Układ 74155 zawiera dwa takie demultipleksery.



Rys. 16. Symbol graficzny demultipleksera o dwóch wejściach adresowych

Przykłady zastosowań multiplekserów i demultiplekserów:

- Multipleksowany system przesyłania danych – umożliwiający przesyłanie słów wielobitowych jedną linią danych. System taki składa się z multipleksera i demultipleksera, które pełnią rolę przetworników zamieniających format słów z postaci równoległej na szeregową i odwrotnie. Warunkiem koniecznym poprawnego działania systemu jest zapewnienie takich samych stanów na wejściach adresowych multipleksera i demultipleksera.

- Realizacja układów kombinacyjnych – przykłady realizacji funkcji logicznych przy pomocy multiplexera i demultiplexera zostały przedstawione w zadaniach poniżej.

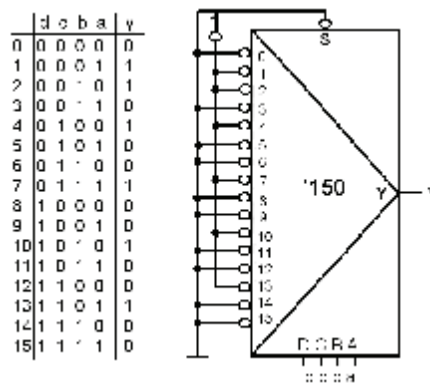
Zadanie 1.

Zaprojektować przy użyciu multiplexera układ kombinacyjny opisany następującą funkcją logiczną:

$$Y = f(d,c,b,a) = \sum(1,2,4,7,10,13)$$

Rozwiązanie:

Podana funkcja jest opisana przy pomocy 4 zmiennych, więc do zrealizowania układu można wykorzystać multiplexer z 2^4 wejściami informacyjnymi, wybór wejścia informacyjnego sterowany będzie wtedy 4 wejściami adresowymi, do których należy podłączyć zmienne a, b, c, d. Wówczas w zależności od aktualnego stanu tych zmiennych, wejście o numerze odpowiadającym numerowi stanu będzie połączone z wyjściem. Aby rozwiązać zadanie należy zgodnie z podaną funkcją logiczną uzupełnić tablicę prawdy, a następnie zgodnie z tą tablicą połączyć układ. Jeżeli w i – tym wierszu tablicy funkcja ma wartość 1 to należy i – te wejście multiplexera połączyć ze źródłem poziomu logicznego „1”, natomiast jeżeli funkcja w i – tym wierszu tablicy funkcja ma wartość 0 to należy i – te wejście multiplexera połączyć ze źródłem poziomu logicznego „0”.



Rys. 17. Tablica prawdy oraz schemat logiczny układu realizującego funkcję $Y = \sum(1,2,4,7,10,13)$ [1, s. 207]

Zadanie 2.

Zaprojektować przy użyciu demultiplexera układ kombinacyjny trzywyjściowy opisany następującymi funkcjami:

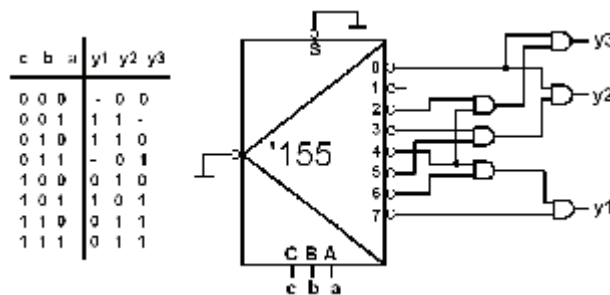
$$y_1 = f_1(c,b,a) = \prod(4,6,7(0,3))$$

$$y_2 = f_2(c,b,a) = \sum(1,2,4,6,7)$$

$$y_3 = f_3(c,b,a) = \prod(0,2,4(1))$$

Rozwiązanie:

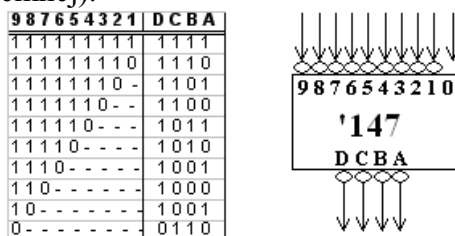
Projektowany układ kombinacyjny jest opisany funkcjami zależnymi od trzech zmiennych: a, b i c, użyjemy więc demultiplexera o ośmiu wyjściach. Aby rozwiązać zadanie należy zgodnie z podanymi funkcjami logicznymi uzupełnić tablicę prawdy, a następnie według tej tablicy połączyć układ. Do realizacji funkcji wybrano zera, ponieważ ich liczba w tablicy prawdy jest mniejsza niż liczba 1. Funkcja y_1 powinna mieć wartość zero dla stanu 4, 6 i 7. Z zasady działania demultiplexera wynika, że wszystkie wyjścia oprócz aktualnie wybranego mają wartość 1. Tak więc aby zrealizować funkcję y_1 wyjścia 4, 6 i 7 należy połączyć z wejściami trójwejściowej bramki AND (w przedstawionym rozwiązaniu zamiast bramki trójwejściowej zastosowano dwie dwuwejściowe). Podobnie realizuje się pozostałe dwie funkcje y_2 i y_3 .



Rys. 18. Tablica prawdy oraz schemat logiczny realizowanej funkcji

Koder

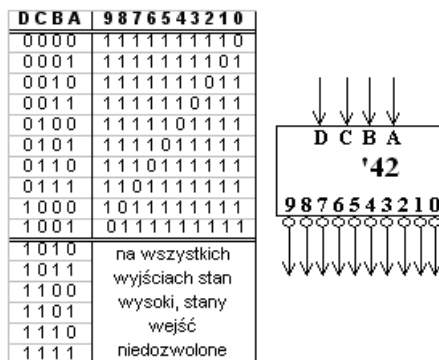
Koderami nazywamy pewną klasę konwerterów kodów służących do zmiany kodu „1 z n” na kod wewnętrzny urządzenia. Przykładem koder jest układ scalony typu ‘147. Zamienia on kod „1 z n” na kod naturalny BCD przedstawiony w logice ujemnej. Normalna praca tego koder to wyróżnienie jednego z wejść (wyróżnione wejście w stanie niskim). Na wyjściu pojawia się słowo czterobitowe w kodzie naturalnym BCD, odpowiadające numerowi wejścia wyróżnionego (w logice ujemnej).



Rys. 19. Tablica działania i symbol graficzny koder ‘147 [1, s. 217]

Dekoder

Dekoderem nazywamy konwerter, którego zadaniem jest zamiana kodu wewnętrznego urządzenia na kod „1 z n”. Przykładem dekodera jest układ scalony typu ‘42. Zamienia on kod naturalny BCD na kod „1 z n”. Normalna praca tego dekodera to wyróżnienie wyjścia (wyróżnione wyjście w stanie niskim), którego numer odpowiada wartości binarnej podanej na jego wejścia.

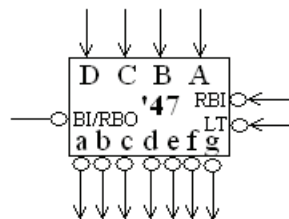


Rys. 20. Tablica działania i symbol graficzny dekodera ‘42 [1, s. 219]

Transkoder

Transkoderami (translatorami kodów) nazywamy takie konwertery kodów, które przetwarzają informację z dowolnego kodu A na kod B, przy czym żaden z kodów nie jest kodem „1 z n”. Przykładem transkodera jest układ typu ‘47, który zamienia kod BCD na kod wskaźnika siedmiosegmentowego. Przeznaczony jest on do sterowania wskaźnika 7 – segmentowego, ze wspólną anodą. Na rysunku przedstawiono tablicę działania i symbol graficzny takiego transkodera.

LT	RBI	DCBA	BI/RBO	a b c d e f g	liczba dziesiętna lub realizowana funkcja
1	1	0000	1	0 0 0 0 0 0 1	0
1	-	0001	1	1 0 0 1 1 1 1	1
1	-	0010	1	0 0 1 0 0 1 0	2
1	-	0011	1	0 0 0 0 1 1 0	3
1	-	0100	1	1 0 0 1 1 0 0	4
1	-	0101	1	0 1 0 0 1 0 0	5
1	-	0110	1	1 1 0 0 0 0 0	6
1	-	0111	1	0 0 0 1 1 1 1	7
1	-	1000	1	0 0 0 0 0 0 0	8
1	-	1001	1	0 0 0 1 1 0 0	9
1	-	1010	1	1 1 1 0 0 1 0	(10)
1	-	1011	1	1 1 0 0 1 1 0	(11)
1	-	1100	1	1 0 1 1 1 0 0	(12)
1	-	1101	1	0 1 1 0 1 0 0	(13)
1	-	1110	1	1 1 1 0 0 0 0	(14)
1	-	1111	1	1 1 1 1 1 1 1	(15)
-	-	----	0	1 1 1 1 1 1 1	wygaszenie wskaźnika
1	0	0000	0	1 1 1 1 1 1 1	wygaszenie zera
0	-	----	1	0 0 0 0 0 0 0	test wskaźnika



Rys. 21. Tablica działania i symbol graficzny transkodera '47 [1, s. 221]

Układ transkodera '47 posiada wejścia sterujące LT i RBI oraz wyprowadzenie oznaczone BI/RBO, które może być wykorzystane zarówno jako wejście jak i wyjście. Wejście LT (Lamp Test) jest aktywne w stanie niskim, pozwala kontrolować świecenie wszystkich segmentów wskaźnika. Wejście RBI (Ripple Blanking Input) służy do wygaszania wskaźnika, jeśli wskazuje on 0 (wygaszanie zer nieznaczących). Wyjście RBO (Ripple Blanking Output) jest ustawiane w stan niski, jeżeli na wskaźniku zostało wygaszone nieznaczące zero. Może ono być stosowane jako wejście BI (Blanking Input). Stan niski podany na to wejście powoduje wygaszenie wskaźnika niezależnie od innych sygnałów wejściowych. Może być ono wykorzystywane do regulowania jasności świecenia wskaźnika.

4.2.2 Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Wyjaśnij zasadę działania multiplexera i demultiplexera.
2. Zaprojektuj prosty układ kombinacyjny z wykorzystaniem multiplexera i demultiplexera.
3. Wyjaśnij różnicę między koderem, dekoderym i transkoderym.
4. Wyjaśnij zasadę działania kodera typu '147.
5. Wyjaśnij zasadę działania dekodera typu '42.
6. Wyjaśnij zasadę działania transkodera typu '47.

4.2.3 Ćwiczenia

Ćwiczenie 1

Zbadać działanie multiplexera.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych multiplexerów,
- 2) zaproponować schematy pomiarowe oraz tabele do zapisywania wyników pomiarów,
- 3) podawać na wejścia multiplexera, za pomocą zadajnika stanów, następujące stany logiczne:

Nr wejścia	Wartość
0	0
1	1
2	0
3	1

- 4) podać stan wysoki na wejście strobujące,
- 5) podawać na wejścia adresowe kolejno adresy: 00, 01, 10, 11 odczytując, za każdym razem stan logiczny na wyjściu, wyniki zapisywać w tabeli,
- 6) podać stan niski na wejście strobujące,
- 7) podawać na wejścia adresowe kolejno adresy: 00, 01, 10, 11, odczytując za każdym razem stan logiczny na wyjściu, wyniki zapisywać w tabeli,
- 8) sprawdzić na podstawie otrzymanych pomiarów, czy multiplekser działa poprawnie,
- 9) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca multipleksery, przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 2

Zbadać działanie demultipleksera.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych demultipleksarów,
- 2) zaproponować schematy pomiarowe oraz tabele do zapisywania wyników pomiarów,
- 3) podać stan niski na wejście demultipleksera,
- 4) podać stan wysoki na wejście strobujące,
- 5) podawać na wejścia adresowe kolejno adresy: 00, 01, 10, 11, odczytując za każdym razem stany logiczne na wyjściach, wyniki zapisywać w tabeli,
- 6) podać stan niski na wejście strobujące,
- 7) podawać na wejścia adresowe kolejno adresy: 00, 01, 10, 11, odczytując za każdym razem stany logiczne na wyjściach, wyniki zapisywać w tabeli,
- 8) podać stan wysoki na wejście demultipleksera,
- 9) powtórzyć czynności wymienione w punktach 4 – 7,
- 10) sprawdzić na podstawie otrzymanych pomiarów, czy demultiplekser działa poprawnie,
- 11) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca demultipleksery,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 3

Zbadać linię transmisyjną.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych multiplekserów i demultiplekserów,
- 2) zaproponować połączenie multipleksera i demultipleksera tak, aby powstała linia transmisyjna, umożliwiająca przesyłanie słów ośmiobitowych za pomocą minimalnej ilości przewodów,
- 3) narysować schemat pomiarowy, zaproponować tabele do zapisywania wyników pomiarów,
- 4) połączyć układ według schematu,
- 5) sprawdzić działanie zaproponowanego układu,
- 6) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca multipleksery i demultipleksery,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 4

Zbadać działanie dekodera „1 z 10”.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych dekoderek,
- 2) zaproponować schemat pomiarowy oraz tabele do zapisywania wyników pomiarów,
- 3) połączyć układ według schematu,
- 4) zbadać działanie dekodera, wyniki pomiarów zapisywać w tabeli,
- 5) dokonać oceny poprawności wykonanego ćwiczenia,

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca dekodery,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 5

Zbadać działanie transkodera typu '47.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych transkoderów,
- 2) zaproponować schemat pomiarowy oraz tabele do zapisywania wyników pomiarów,

- 3) połączyć układ według schematu,
- 4) zbadać działanie transkodera, wyniki pomiarów zapisywać w tabeli,
- 5) dokonać oceny poprawności wykonanego ćwiczenia,

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca transkodery, wskaźniki 7 – segmentowe,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 6

Realizacja układu kombinacyjnego.

Zaprojektować układ kombinacyjny realizujący następującą funkcję logiczną:

$$Y = f(d,c,b,a) = \prod(1,2,4,7,10,13)$$

przy użyciu multiplexera o trzech wejściach adresowych.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) narysować i wypełnić tablicę prawdy,
- 2) zaproponować układ realizujący powyższą funkcję,
- 3) zaproponować sposób sprawdzenia poprawności działania zaprojektowanego układu kombinacyjnego (schemat pomiarowy oraz tabele do zapisywania wyników pomiarów),
- 4) połączyć układ według schematu,
- 6) sprawdzić działanie zaprojektowanego układu, wyniki pomiarów zapisywać w tabeli,
- 7) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca multiplexery i demultiplexery, przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

4.2.4 Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) zbadać poprawność działania multipleksera i demultipleksera?	<input type="checkbox"/>	<input type="checkbox"/>
2) zaprojektować prosty układ kombinacyjny z wykorzystaniem multipleksera i demultipleksera?	<input type="checkbox"/>	<input type="checkbox"/>
3) wyjaśnić różnicę między koderem, dekerem i transkoderem?	<input type="checkbox"/>	<input type="checkbox"/>
4) zbadać poprawność działania kodera typu '147?	<input type="checkbox"/>	<input type="checkbox"/>
5) zbadać poprawność działania dekodera typu '42?	<input type="checkbox"/>	<input type="checkbox"/>
6) zbadać poprawność działania transkodera typu '47?	<input type="checkbox"/>	<input type="checkbox"/>
7) zbadać poprawność działania multipleksera i demultipleksera?	<input type="checkbox"/>	<input type="checkbox"/>
8) zrealizować prostą funkcję logiczną przy pomocy multipleksera lub demultipleksera?	<input type="checkbox"/>	<input type="checkbox"/>
9) zastosować transkoder do sterowania wyświetlaczem siedmiosegmentowym?	<input type="checkbox"/>	<input type="checkbox"/>

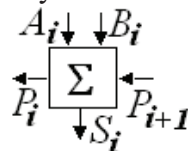
4.3 Układy arytmetyczne.

4.3.1 Materiał nauczania

Sumator

Sumatorem nazywamy podstawowy układ arytmetyczny, służący do wykonywania operacji dodawania. Układ sumatora n-bitowego zbudowany jest z połączonych ze sobą n elementarnych sumatorów.

Na rysunku przedstawiono symbol graficzny sumatora elementarnego (jednobitowego):



Rys. 22. Symbol graficzny sumatora elementarnego

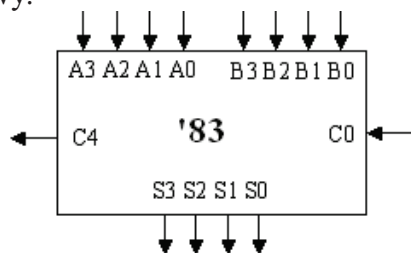
A_i, B_i – dodawane bity liczb na i -tej pozycji,

P_i – przeniesienie do i -tej pozycji powstałe przy dodawaniu bitów pozycji $(i-1)$,

P_{i+1} – przeniesienie do starszej pozycji $(i+1)$ powstałe przy dodawaniu liczb na i -tej pozycji,

S_i – wynik arytmetycznego dodawania bitów liczb dla pozycji i -tej.

W praktyce stosuje się układy sumujące dwie liczby n -bitowe, przy czym n jest wielokrotnością liczby cztery. Do układów takich zaliczamy m.in. monolityczny układ scalony typu '83. Jest to sumator 4-bitowy.



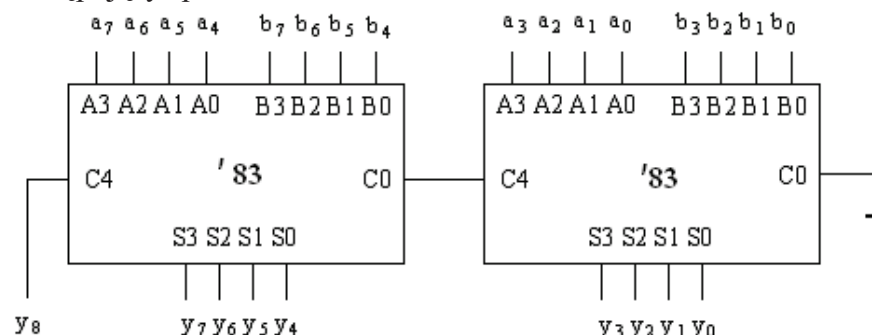
Rys. 23. Symbol graficzny układu '83 [1, s. 230]

Zawiera on 4 sumatory elementarne. Działanie sumatora można opisać następująco:

$$\begin{array}{r}
 A3 \ A2 \ A1 \ A0 \\
 B3 \ B2 \ B1 \ B0 \\
 + \qquad \qquad C0 \\
 \hline
 C4 \ S3 \ S2 \ S1 \ S0
 \end{array}$$

Rys. 24. Działanie sumatora typu '83 [1, s. 230]

Wykorzystując wejście przeniesienia $C0$ i wyjście przeniesienia $C4$ można rozbudować układ np. do układu o długości słowa równej 8 bitów, łącząc kaskadowo układy '83. Na rysunku poniżej przedstawiono połączenie 2 układów typu '83 realizujące dodawanie dwóch słów 8 bitowych w następujący sposób:



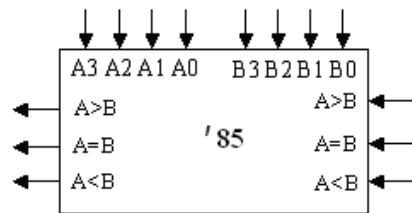
Rys. 25. Sumator 8-bitowy zrealizowany przy pomocy dwóch układów typu '83 [1, s. 230]

Powyższy układ dodaje do siebie dwie liczby 8 – bitowe A i B ($A = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$, $B = b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$). Jako wynik otrzymujemy 9 bitów (dziewiątym bitem y_8 jest bit przeniesienia C_4) – liczbę $Y = y_8 y_7 y_6 y_5 y_4 y_3 y_2 y_1 y_0$.

Komparator

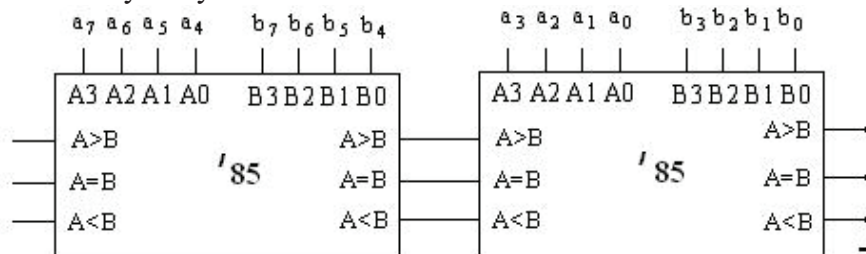
Komparator jest to układ arytmetyczny służący do porównywania liczb. Porównanie wartości bezwzględnych liczb w kodzie dwójkowym może się odbywać w układzie porównującym poszczególne bity liczb, przy uwzględnieniu przeniesień z poprzednich pozycji, które już zostały ze sobą porównane. Wynikiem porównania liczb A i B może być jedna z trzech relacji: $A > B$, $A = B$ i $A < B$. Czasami można spotkać się z komparatorem umożliwiającym odróżnienie jedynie relacji $A \neq B$ i $A = B$ lub np. $A \geq B$ i $A < B$; nie jest on jednak w pełni funkcjonalny.

Układ '85 jest komparatorem 4-bitowym, tzn. układem porównującym liczby o długości czterech bitów.



Rys. 26. Symbol graficzny układu '85 [1, s. 233]

Układ ten ma osiem wejść danych (cztery bity liczby A i cztery bity liczby B). Wynik porównania liczb A i B jest wskazywany przez poziom wysoki na którymś z trzech wyjść (np. jeśli poziom logicznej „1” pojawi się na wyjściu $A=B$ oznacza to, że liczby A i B są sobie równe). Wejścia oznaczone $A > B$, $A = B$ i $A < B$ służą do łączenia układów kaskadowo. Przykładowo, aby uzyskać komparator 8-bitowy należy połączyć ze sobą dwa układy '85 w sposób przedstawiony na rysunku.

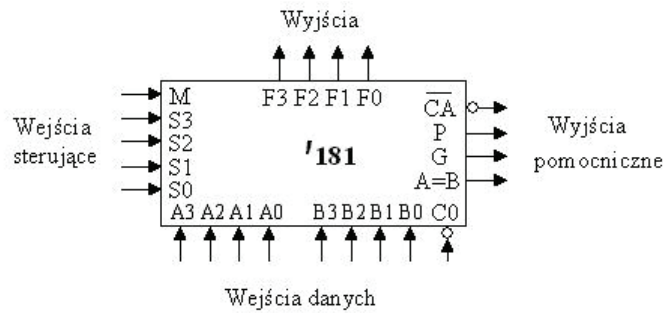


Rys. 27. Komparator 8 – bitowy zrealizowany przy pomocy dwóch układów typu '85

Wykorzystując wejścia na których przekazywany jest wynik porównania na poprzedniej pozycji można stworzyć za pomocą układu '85 komparator n bitowy, przy czym $n = k \cdot 4$ ($k=1,2,3\dots$).

Jednostka arytmetyczno – logiczna (ALU)

Jednostka arytmetyczno – logiczna jest blokiem funkcjonalnym, przeznaczonym do wykonywania operacji logicznych i arytmetycznych. Układ scalony typu '181 wykonuje te działania na słowach czterobitowych.



Rys. 28. Symbol graficzny układu '181 [1, 233]

Na wejścia danych $A_0 \div A_3$, $B_0 \div B_3$ podawane są czterobitowe liczby, na których przeprowadzane są operacje. Zanegowane wejście C_0 jest wejściem przeniesienia wykorzystywanym w przypadku kaskadowego łączenia układów. Wynik operacji pojawia się na wyjściach $F_0 \div F_3$, wyjście \overline{CA} (zanegowane) jest to bit przeniesienia generowanego podczas wykonywania operacji arytmetycznych. Wyjścia P i G to wyjścia przeniesień (propagowanego i generowanego) wykorzystywane w przypadku konieczności przyspieszenia działania układu zbudowanego z kilku jednostek ALU i zastosowania układu '182 (generator przeniesień jednoczesnych). Wyjście $A=B$ jest wyjściem komparatora, na którym ustawiana jest logiczna „1” jeśli wszystkie bity wyniku są równe 1. Za pomocą wejścia M wybiera się, czy układ ma realizować funkcje logiczne ($M=1$) czy mieszane ($M=0$). Wejścia sterujące $S_0 \div S_3$ są to wejścia wyboru funkcji:

Tabela 3 Funkcje realizowane przez układ '181

S3S2S1S0	Funkcje logiczne (M=1)	Funkcje arytmetyczne i logiczne (M=0)
0000	$F = \overline{A}$	$F = A + C_0$
0001	$F = \overline{A + B}$	$F = (A \cup B) + C_0$
0010	$F = \overline{AB}$	$F = (A \cup \overline{B}) + C_0$
0011	0	$F = C_0 - 1$
0100	$F = \overline{AB}$	$F = A + A\overline{B} + C_0$
0101	$F = \overline{B}$	$F = (A \cup B) + A\overline{B} + C_0$
0110	$F = A \oplus B$	$F = A - B - (1 - C_0)$
0111	$F = A\overline{B}$	$F = A\overline{B} - (1 - C_0)$
1000	$F = \overline{A} + B$	$F = A + AB + C_0$
1001	$F = \overline{A} \oplus \overline{B}$	$F = A + B + C_0$
1010	$F = B$	$F = (A \cup \overline{B}) + AB + C_0$
1011	$F = AB$	$F = AB - (1 - C_0)$
1100	1	$F = 2A + C_0$
1101	$F = A + \overline{B}$	$F = (A \cup B) + A + C_0$
1110	$F = A + B$	$F = (A \cup \overline{B}) + A + C_0$
1111	$F = A$	$F = A - (1 - C_0)$

4.3.2 Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Czy potrafisz wyjaśnić zasadę działania sumatora?
2. Czy potrafisz wyjaśnić zasadę działania komparatora?
3. Czy potrafisz wyjaśnić zasadę działania jednostki arytmetyczno – logicznej?
4. Czy potrafisz zaprojektować układ sumatora 12 – bitowego wykorzystując układ '83?
5. Czy potrafisz zaprojektować układ komparatora 12 – bitowego wykorzystując układ '85?
6. Czy potrafisz zaprojektować układ, który będzie wykonywał określoną operację arytmetyczną lub logiczną?

4.3.3 Ćwiczenia

Ćwiczenie 1

Zbadać działanie sumatora scalonego.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych sumatorów scalonych,
- 2) zaproponować schemat pomiarowy oraz tabele do zapisywania wyników pomiarów,
- 3) połączyć układ według schematu,
- 4) zbadać działanie sumatora zmieniając stany wejść układu, wyniki pomiarów zapisywać w tabeli, wykonać przynajmniej 15 pomiarów,
- 5) zbadać wpływ wartości podanej na wejście przeniesienia na wynik sumowania,
- 6) przeanalizować wyniki i sformułować wnioski,
- 7) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca sumatory scalone,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 2

Zaprojektować układ sumatora 12 – bitowego z wykorzystaniem układów '83.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych sumatorów scalonych,
- 2) zaproponować sposób połączenia dostępnych sumatorów, aby otrzymać układ realizujący dodawanie dwóch liczb 12 – bitowych,
- 3) zaproponować schemat pomiarowy do sprawdzenia poprawności działania zaprojektowanego sumatora oraz tabele do zapisywania wyników pomiarów,
- 4) połączyć układ według schematu,

- 5) zbadać działanie sumatora zmieniając stany wejść układu, wyniki pomiarów zapisywać w tabeli, wykonać przynajmniej 15 pomiarów,
- 6) przeanalizować wyniki i sformułować wnioski,
- 7) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca sumatory scalone,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 3

Zbadać działanie komparatora scalonego.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych komparatorów scalonych,
- 2) zaproponować schemat pomiarowy oraz tabele do zapisywania wyników pomiarów,
- 3) połączyć układ według schematu,
- 4) zbadać działanie komparatora zmieniając stany wejść układu, wyniki pomiarów zapisywać w tabeli, wykonać przynajmniej 15 pomiarów,
- 5) zbadać wpływ wartości podanych na wejścia $A=B$, $A>B$, $A<B$ na wynik porównania,
- 6) przeanalizować wyniki i sformułować wnioski,
- 7) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca komparatory scalone,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 4

Zbadać działanie jednostki arytmetyczno – logicznej.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną jednostki arytmetyczno – logicznej,
- 2) zaproponować sposób połączenia układu jednostki arytmetyczno – logicznej do realizacji funkcji wskazanej przez nauczyciela,
- 3) zaproponować schemat pomiarowy oraz tabele do zapisywania wyników pomiarów,
- 4) połączyć układ według schematu,
- 5) zbadać działanie ALU zmieniając stany wejść układu, wyniki pomiarów zapisywać w tabeli, wykonać przynajmniej 5 pomiarów,

- 6) powtórzyć punkty 2 – 5 dla 2 funkcji logicznych i 2 funkcji mieszanych,
- 7) przeanalizować wyniki i sformułować wnioski,
- 8) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca jednostkę arytmetyczno - logiczną,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

4.3.4 Sprawdzian postępów

Czy potrafisz	Tak	Nie
1) wyjaśnić zasadę działania sumatora?	<input type="checkbox"/>	<input type="checkbox"/>
2) wyjaśnić zasadę działania komparatora?	<input type="checkbox"/>	<input type="checkbox"/>
3) wyjaśnić zasadę działania jednostki arytmetyczno – logicznej?	<input type="checkbox"/>	<input type="checkbox"/>
4) zaprojektować i zmontować układ sumatora 12 – bitowego wykorzystując układ 7483?	<input type="checkbox"/>	<input type="checkbox"/>
5) zaprojektować i zmontować układ komparatora 12 – bitowego wykorzystując układ 7485?	<input type="checkbox"/>	<input type="checkbox"/>
6) zaprojektować i zmontować układ, który będzie wykonywał określoną operację arytmetyczną lub logiczną?	<input type="checkbox"/>	<input type="checkbox"/>
7) zbadać poprawność działania sumatora?	<input type="checkbox"/>	<input type="checkbox"/>
8) zbadać poprawność działania komparatora?	<input type="checkbox"/>	<input type="checkbox"/>
9) zbadać poprawność działania ALU?	<input type="checkbox"/>	<input type="checkbox"/>
10) zastosować poznane elementy realizujące funkcje arytmetyczne w projektowanych układach cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>

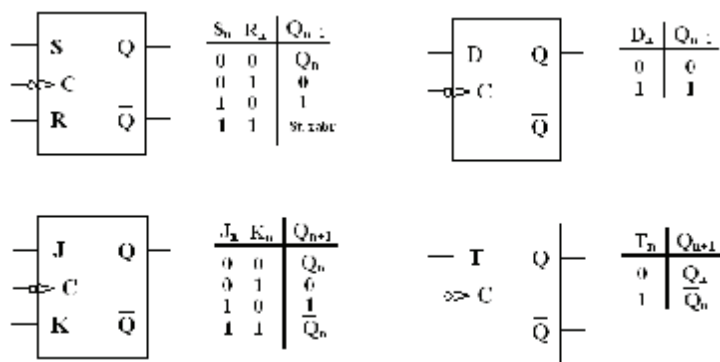
4.4 Układy pamiętające. Liczniki. Układy programowalne

4.4.1 Materiał nauczania

Przerzutniki

Przerzutnik (z ang. flip flop) jest elementarnym układem zapamiętującym zmianę stanu logicznego wejścia. Stan zapamiętany sygnalizowany jest zmianą wyjścia przerzutnika. Przerzutniki posiadają zwykle dwa wejścia informacyjne, dwa wejścia sterujące (asynchroniczne – set i reset), wejście zegarowe i dwa wyjścia komplementarne (stan tych wyjść jest zawsze przeciwny). Wyróżniamy przerzutniki asynchroniczne (zmiana stanu wyjścia przerzutnika spowodowana jest bezpośrednio zmianą któregoś z jego wejść danych) i synchroniczne. Wejścia synchroniczne przerzutnika używane są do wywołania zmiany wartości zmiennych wyjściowych, ale zmiana ta następuje w takt impulsu synchronizującego, zwanego również impulsem zegarowym (ang. clock) lub taktującym. Brak impulsu taktującego oznacza, że przerzutnik nie będzie reagował na zmiany wartości zmiennych informacyjnych synchronicznych przerzutnika.

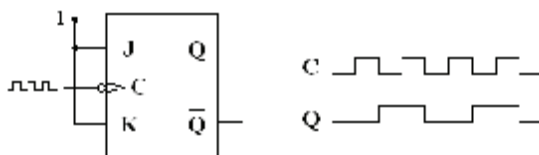
Na rysunku przedstawiono symbole graficzne i tabele przejść opisujące działanie przerzutników synchronicznych.



Rys. 29. Symbole graficzne i tablice przejść przerzutników synchronicznych [1, s. 122 – 123]

Analizując tablice przejść przerzutnika JK i T łatwo zauważyć, że przerzutnik JK oraz T są podobne w działaniu, tzn. jeżeli stan wejść przerzutnika JK jest taki sam (na obydwu wejściach logiczne „0” lub „1”), to przerzutnik ten zachowuje się jak przerzutnik T.

Przeanalizujmy działanie przerzutnika JK z tak połączonymi wejściami:

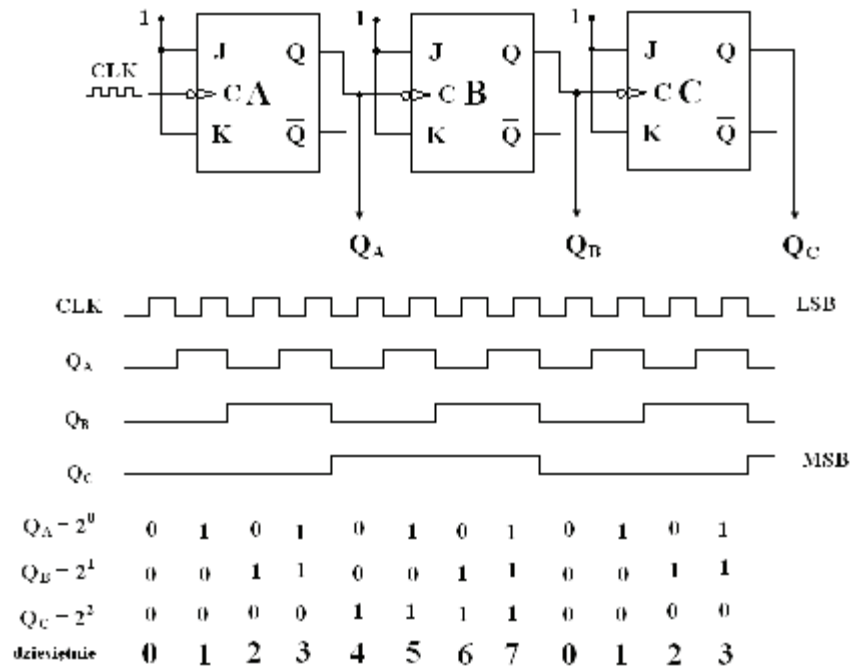


Rys. 30. Przerzutnik T zbudowany w oparciu o przerzutnik JK [1, s. 126]

Jak wynika z tabeli przejść dla przerzutnika JK, jeżeli jego wejścia informacyjne są zwarte i podane jest na nie logiczne „0”, wówczas stan wyjścia przerzutnika się nie zmienia. Jeżeli natomiast na zwarte wejścia J i K zostanie podana logiczna „1” wówczas wyjście przerzutnika zmienia stan na przeciwny w takt impulsów zegarowych. Powstaje w ten sposób, tzw. „dwójka licząca”, czyli licznik mod 2 (ma dwa stany, w których na przemian się znajduje). Układ ten znajduje również zastosowanie jako dzielnik częstotliwości przez dwa.

Przy kaskadowym połączeniu tego typu układów można utworzyć licznik mod N (dzielnik częstotliwości przez N), gdzie N jest dodatnią potęgą liczby 2. Projektując licznik mod 8 (czyli

innymi słowy dzielnik częstotliwości przez 8) potrzebne będą 3 dwójki liczące połączone jak na rysunku poniżej.



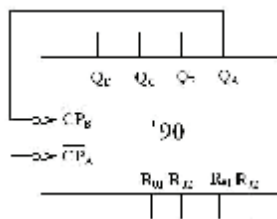
Rys. 31. Licznik mod 8 zrealizowany przy użyciu przerzutników JK

Pierwszy przerzutnik dzieli częstotliwość zegara taktującego przez dwa. Ponieważ wejście taktujące następnego przerzutnika połączone jest z wyjściem pierwszego przerzutnika, wyjście Q_B zmienia się z czterokrotnie mniejszą częstotliwością niż wartość sygnału zegarowego pierwszego przerzutnika. Łącząc wejście zegarowe kolejnego przerzutnika z wyjściem Q_B uzyskamy podział częstotliwości jeszcze przez dwa, czyli licznik mod 8.

Liczniki.

Licznik jest to cyfrowy układ sekwencyjny, służący do zliczania i pamiętania liczby impulsów podawanych na jego wejście zliczające. Liczniki dostępne są w różnych wersjach w postaci gotowych układów scalonych. Różnią się one pojemnością (bardzo popularne są liczniki 4 – bitowe), sposobem taktowania (asynchroniczne – każdy stopień licznika taktowany jest sygnałem z poprzedniego stopnia oraz synchroniczne – stan wszystkich przerzutników zmienia się równocześnie), kierunkiem (liczniki zliczające w dół, w górę i liczniki rewersyjne) i sposobem zliczania (BCD – dziesiętne, binarne – szesnastkowe, mod n – umożliwiają dzielenie częstotliwości wejściowej przez liczbę n podawaną jako słowo wejściowe).

Bardzo popularnym układem jest licznik typu '90, który jest licznikiem dziesiętnym, asynchronicznym. Zawiera on cztery przerzutniki, z których pierwszy stanowi licznik mod 2 (z wejściem zegarowym $\overline{CP_A}$ i wyjściem Q_A), natomiast trzy pozostałe stanowią licznik mod 5 (z wejściem zegarowym $\overline{CP_B}$ i wyjściami $Q_B Q_C Q_D$). Tak więc w zależności od tego, do którego wejścia zegarowego podłączony zostanie sygnał zegarowy oraz z których wyjść będzie odczytywany stan licznika, licznik pracuje w jednym w dwóch wymienionych wyżej trybów. Jeżeli połączymy wyjścia Q_A z wejściem $\overline{CP_B}$ otrzymamy licznik mod 10. Omówiony sposób połączenia został przedstawiony na rysunku:



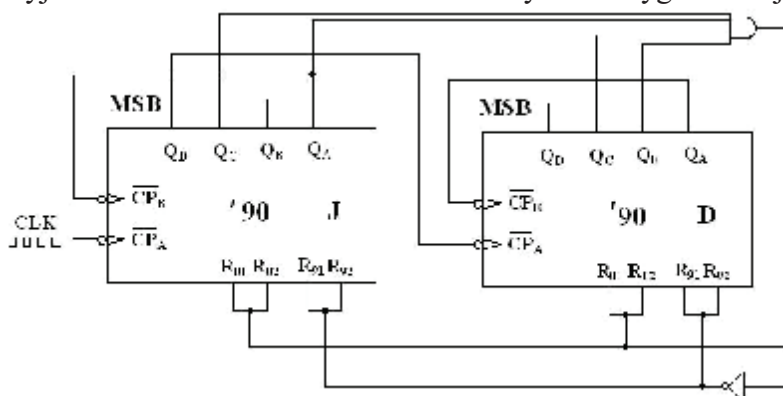
Rys. 32. Licznik '90 [1, s. 241]

Wejścia R_{01} i R_{02} służą do zerowania licznika, natomiast wejścia R_{91} i R_{92} do ustawiania go w stan 1001. Chociaż istnieje cały szereg różnych liczników np. mod 2, mod 5, mod 6 itp. często można się spotkać z koniecznością zaprojektowania licznika innego niż standardowo dostępne.

Wówczas nie pozostaje projektantowi takiego układu cyfrowego nic innego jak zbudować licznik z liczników o mniejszej pojemności łącząc je kaskadowo. Niestety w rzeczywistości otrzymujemy najczęściej licznik o pojemności większej niż jest potrzebna, dlatego bardzo ważnym elementem jest umiejętne skrócenie cyklu pracy licznika. Polega ono na wykryciu (zdekodowaniu) stanu, do którego licznik nie powinien już doliczyć i podaniu sygnału resetu licznika, zaczynając a ten sposób nowy cykl.

Przykładowo, aby zaprojektować licznik mod 65 należy:

- rozszerzyć pojemność licznika łącząc kaskadowo dwa liczniki mod 10 – otrzymuje się w ten sposób licznik mod 100,
- skrócić cykl liczenia ze 100 do 65 – w tym celu należy wykryć moment pojawienia się liczby 65 na wyjściu licznika i równocześnie z nim wystawić sygnał zerujący układ.



Rys. 33. Licznik mod 65

Przedstawiony na rysunku licznik mod 65 składa się z dwóch połączonych kaskadowo liczników mod 10, z których pierwszy z lewej zlicza jednostki (J), natomiast drugi dziesiątki (D). Wyjścia licznika 7490 zliczającego jednostki o wagach 2^0 i 2^2 podane są na bramkę AND wraz z wyjściami o wagach 2^1 oraz 2^2 licznika zliczającego dziesiątki. W ten sposób w momencie, gdy licznik przechodzi w stan 65 na wyjściu 4-wejściowej bramki AND pojawia się stan „1”, który zeruje cały licznik. W podobny sposób można zaprojektować licznik o dowolnej pojemności.

Bardzo podobne w działaniu do licznika '90 są układy:

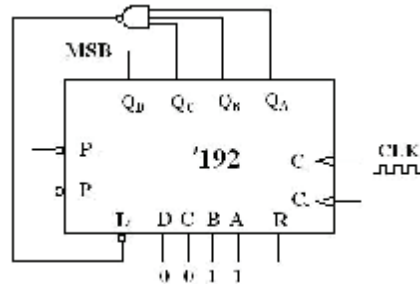
- '92 – który składa się z 4 przerzutników, z których pierwszy stanowi licznik mod 2, a pozostałe mod 6, czyli przy odpowiednim połączeniu można otrzymać licznik mod 12,
- '93 – który składa się z 4 przerzutników, z których pierwszy pracuje jako licznik mod 2 a pozostałe jako mod 8, czyli przy odpowiednim połączeniu można otrzymać licznik mod 16.

Przedstawione powyżej liczniki mają pewną wadę, mianowicie zawsze zaczynają liczyć od stanu 0000. Nie ma możliwości ustawienia wartości od której licznik powinien zacząć liczyć.

Takie możliwości dają liczniki synchroniczne np. '192 (dziesiętny) i '193 (binarny). Liczniki te mają 4 wyjścia (Q_D, Q_C, Q_B, Q_A), cztery wejścia służące do określenia wartości, od jakiej licznik ma zacząć zliczanie (D, C, B, A), wejście ustawiające L (aktywne w stanie niskim), wejście zerujące R (aktywne w stanie wysokim), wejścia zegarowe (C_+ - zliczanie w górę, C_- - zliczanie w dół) oraz wyjścia przeniesienia P_+ i P_- .

Na rysunku przedstawiono zastosowanie licznika '192 do zliczania od stanu 3 do stanu 6.

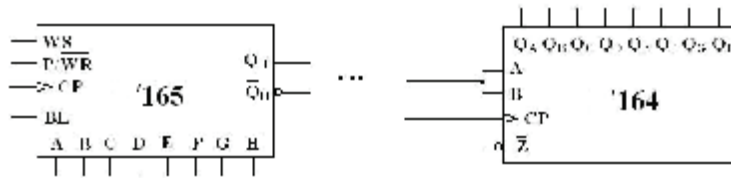
Aby wykonać to zadanie należy zdekodować stan, który już nie powinien się pojawić (czyli 7) i w momencie jego wykrycia podać sygnał ustawiający licznik w stan początkowy (czyli 3). Stanem początkowym jest liczba 0011 podana na wejścia DCBA, tak więc zamiast na wejście zerowania sygnał „przepełnienia licznika” podajemy na asynchroniczne wejście L, co powoduje wpisanie wartości ustawionej na wejściach DCBA licznika.



Rys. 34. Licznik zliczający od 3 do 6

Rejestry

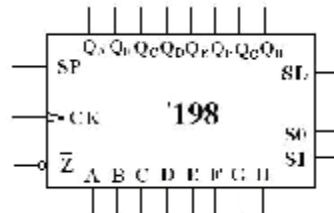
Rejestr nazywamy układ zbudowany z przerzutników, służący do przechowywania informacji. Liczba bitów informacji, jaka może być przechowywana w rejestrze, jest nazywana długością rejestru i odpowiada zawsze liczbie przerzutników, z których jest zbudowany rejestr. Informacja może być wpisywana do rejestru i odczytywana zarówno szeregowo, jak i równoległe – stanowi to kryterium podziału rejestrów na szeregowo-szeregowo, szeregowo-równoległe, równoległo-szeregowo i równoległo-równoległe. Rejestr mający wyłącznie możliwość równoległego wprowadzania i pobierania informacji nazywa się w skrócie rejestrzem równoległym. Pozostałe rodzaje rejestrów są nazywane rejestrmi przesuwającymi ze względu na to, że informacja w nich zawarta jest przesuwana bit po bicie synchronicznie z impulsami taktującymi. Rejestr przesuwający, którego wyjście jest połączone z wejściem, nazywa się licznikiem (rejestrem) pierścieniowym. Rejestry scalone budowane jako 4-, 5-, 6-, 8-, 9-, 16-bitowe należą do układów o średnim stopniu scalenia. Układy rejestrów mogą być wykorzystywane jako pamięci buforowe, układy przesyłania informacji lub do budowy liczników pierścieniowych, dzielników częstotliwości itp. Na rysunku poniżej przedstawiono symbole graficzne dwóch rejestrów scalonych. Rejestr '165 jest rejestrzem 8-bitowym o równoległym wejściu (A, B, C, D, E, F, G, H) i szeregowym wyjściu (dwa komplementarne wyjścia Q_H i \overline{Q}_H). Pozostałe wyprowadzenia układu to: wejście zegarowe CP, WS – wejście szeregowo danych, BL – wejście blokujące sygnał zegarowy (aktywne w stanie wysokim), P/\overline{WR} - wejście sterujące (stan wysoki powoduje przesuwanie danych z wejścia szeregowo synchronicznie z sygnałem zegarowym, a stan niski asynchroniczne wpisanie informacji z wejść równoległych do rejestru). Układ '164 jest natomiast rejestrzem o szeregowym wejściu (z każdym taktem zegara jest wprowadzana informacja będąca iloczynem logicznym sygnałów A i B) i równoległym wyjściu (bity $Q_A, Q_B, Q_C, Q_D, Q_E, Q_F, Q_G, Q_H$). Wejście \overline{Z} - wejście zerujące (zerowanie asynchroniczne), aktywne poziomem niskim, natomiast wyprowadzenie CP jest wejściem zegarowym.



Rys. 35. Symbole graficzne rejestrów scalonych '165 i '164 [1, s. 256 – 257]

Przy pomocy układów '164 oraz '165 można zbudować tor transmisyjny, działający podobnie jak tor zbudowany przy użyciu multipleksera i demultipleksera. 8-bitowa liczba wprowadzana jest równoległe do rejestru nadajnika, następnie z każdym taktom zegara przesuwana jest o 1 bit w prawo. W ten sposób podczas 8 kolejnych taktów na wyjściu Q_H , które jest połączone ze zwartymi wejściami A i B rejestru odbiornika pojawia się całe 8 bitowe słowo, które można odczytać z wyjść równoległych rejestru '164. Zaletą tego rozwiązania jest mała ilość linii wykorzystanych do transmisji. Trzeba natomiast pamiętać o tym, że aby wysłać informację o rozmiarze większym od 8 bitów trzeba po stronie nadajnika sterować wpisywaniem danych do rejestru (P/\overline{WR}).

Przykładem uniwersalnego rejestru rewersyjnego jest układ '198. Ma on możliwość równoległego oraz szeregowo wprowadzania i wyprowadzania informacji.



Rys. 36. Symbol graficzny rejestru '198 [1, s. 258]

Posiada on dwa wejścia szeregowo (SP – wykorzystywane przy przesuwaniu w prawo i SL – przy przesuwaniu w lewo), wejścia równoległe (A, B, C, D, E, F, G, H), wejście zegarowe CK i wejście zerujące \overline{Z} , wyjścia równoległe ($Q_A, Q_B, Q_C, Q_D, Q_E, Q_F, Q_G, Q_H$) oraz wejścia sterujące S0 i S1.

Tabela 4. Działanie układu '198 w zależności od wartości sygnałów sterujących [1, 259]

Z	CK	S0	S1	Funkcja
0	-	-	-	zerowanie rejestru
1	-	0	0	blokada pracy rejestru
1	┐	0	1	przesuwanie w prawo z wpisaniem do przerzutnika A stanu wejścia SP
1	┐	1	0	przesuwanie w lewo z wpisaniem do przerzutnika D stanu wejścia SL
1	┐	1	1	wprowadzanie informacji z wejść równoległych, synchronizowane zegarem

Ze względu na duże możliwości układ ten jest bardzo często używany w układach cyfrowych, posiada wiele zastosowań. Przykładowo może służyć jako prosty układ mnożący bądź dzielący liczbę przez dwa. Aby wykonać takie działanie należy daną liczbę podać na wejścia informacyjne, ustawiając tryb na wejściach S1 i S0 zgodnie z powyższą tabelką, wczytujemy liczbę. Pamiętając o tym, że aby przemnożyć liczbę przez dwa należy ją przesunąć o 1 bit w lewo, natomiast aby podzielić - w prawo, ustawiamy odpowiedni kierunek przesuwania rejestru za pomocą wejść S1 i S0. Po następnym taktie zegara na wyjściach rejestru '198 znajdzie się wynik operacji. Niestety pewną wadą układu jest brak przeniesienia i w sytuacji, gdy mnożymy liczbę 8 bitową tracimy informację o najstarszej pozycji. Tak więc w pełni funkcjonalny układ mnożący zbudowany na tym rejestrze potrafi mnożyć przez dwa liczbę o maksymalnej długości 7 bitów.

Pamięci.

Pamięci są układami służącymi do przechowywania informacji w postaci ciągów słów binarnych. Są one zbudowane z rejestrów (komórek), w których zapamiętywane jest słowo 1-, 4-, 8-bitowe. Każdej komórce przypisany jest adres, czyli numer zapisany w kodzie dwójkowym, oznaczający położenie informacji. Jeśli chcemy odczytać (lub zapisać) daną informację, trzeba na wejścia adresowe pamięci podać numer komórki, z której chcemy informację odczytać (lub do której chcemy zapisać), a także na wejścia sterujące, odpowiednie sygnały. Do zaadresowania $N = 2^n$ komórek pamięci potrzebnych jest n wejść adresowych (A_0, \dots, A_n). Liczba bitów w komórce pamięci świadczy o organizacji pamięci. Przez organizację pamięci należy rozumieć sposób dostępu do informacji. Od organizacji pamięci zależy ilość wejść / wyjść danych (D_0, \dots, D_m). W pamięciach o organizacji słownej wyprowadzenia danych są dwukierunkowe, tzn. są zarówno wejściami jak i wyjściami w zależności od stanu wejść sterujących. Wyjścia w pamięciach są zawsze trójstanowe, co umożliwia łączenie ze sobą wyjść różnych modułów w celu zwiększenia pojemności pamięci. Oprócz wejść informacyjnych każda pamięć ma również wejścia sterujące:

- CS (Chip Select) lub CE (Chip Enable) – wejścia uaktywniające pamięć, pozwalają ją „wyłączyć”, ustawiając wyjścia w stan wysokiej impedancji i zmniejszając pobór prądu,
- WE (Write Enable) lub WR (WRite) – wejście zezwalające na zapis,
- OE (Output Enable) lub RD (ReaD) – wejście zezwalające na odczyt,
- ALE (Adres Latch Enable), RAS (Row Address Select), CAS (Column Address Select) – wejścia strobuujące adresów.

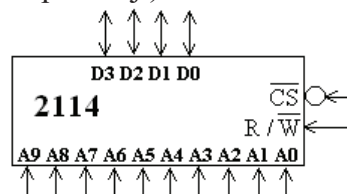
Podstawowe parametry pamięci:

- Pojemność pamięci – oznacza ilość informacji, jaką można w niej przechowywać. Pojemność pamięci określa się podając liczbę słów i długość słowa. Przykładowo pojemność pamięci może wynosić 512Kx64, co oznacza, że jest to 2^{19} słów 64-bitowych.
- Czas dostępu t_A (ang. Access time) – jest to czas jaki upływa od wystąpienia nowego adresu do pojawienia się na wyjściach układu zawartości komórki o podanym adresie.
- Czas cyklu t_{CY} (ang. Cycle time) – jest to najkrótszy odstęp czasu, jaki musi upłynąć pomiędzy dwoma żądaniami dostępu do pamięci.

Pamięci półprzewodnikowe dzieli się na :

- pamięci odczyt – zapis, zwane też pamięciami o dostępie bezpośrednim (Random Access Memory) ,
- pamięci stałe (Read Only Memory – „tylko do odczytu”) – pamięci, której zawartość w czasie normalnej eksploatacji jest niezmienna, tzn. raz zapisana informacja jest trwale przechowywana i może być wielokrotnie odczytywana.

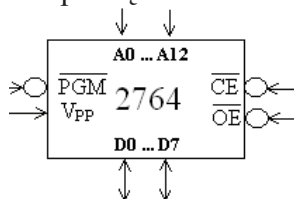
Przykładem pamięci RAM jest układ 2114 (produkowany przez firmę INTEL i NTE). Jest to pamięć statyczna o pojemności 4Kb. Układ ma 10 wejść adresowych ($A_0 \dots A_9$) umożliwiających zaadresowanie 1024 słów 4 – bitowych (4 linie danych $D_0 \dots D_3$). Wejście R/\overline{W} pozwala wybrać tryb pracy pamięci (odczyt lub zapis), wejście \overline{CS} umożliwia odłączenie pamięci (ustawienie wejść w stan wysokiej impedancji).



Rys. 37. Symbol graficzny pamięci 2114 [1, s. 270]

Przykładem pamięci EEPROM jest układ 27C64. Jest to pamięć stała, kasowana elektrycznie o pojemności 64 Kb (8x8K). Układ posiada 13 wejść adresowych ($A_0 \dots A_{12}$), 8 linii danych

(D₀...D₇), wejście \overline{CE} i \overline{OE} wyboru trybu pracy, oraz dodatkowo wejście \overline{PGM} (Program Enable) i V_{PP} służące do programowania pamięci.

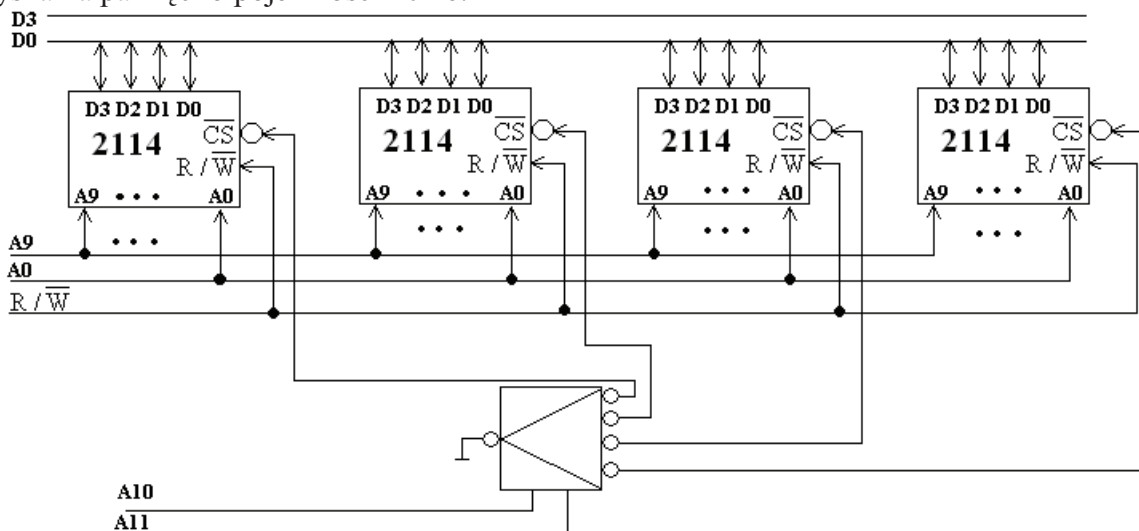


Rys. 38. Symbol graficzny pamięci 28C64 [dokumentacja techniczna firmy Microchip]

Tabela 5. Działanie układu 27C64 w zależności od wartości sygnałów sterujących [dokumentacja techniczna firmy Microchip]

Funkcja	\overline{CE}	\overline{OE}	\overline{PGM}	V _{PP}	A ₉	O ₀ ...O ₇
odczyt	L	L	H	5V	-	dane do odczytu
zapis	L	H	L	H	-	dane do zapisu
weryfikacja	L	L	H	H	-	dane do odczytu
standby	H	-	-	5V	-	wysoka impedancja
odłączone wyjście	L	H	H	5V	-	wysoka impedancja
identyfikacja	L	L	H	5V	H	kod identyfikacyjny

Podczas eksploatacji układów pamięci spotyka się dwa problemy techniczne: zwiększenie pojemności pamięci przez zwiększenie długości pamiętanego słowa oraz zwiększenie pojemności pamięci bez zmiany liczby bitów zapamiętywanych w komórce pamięci. Aby zwiększyć długość słowa należy połączyć ze sobą odpowiednie wejścia adresowe i sterujące kolejnych bloków pamięci. Należy jedynie pamiętać, aby nie przeciążyć układów sterujących wejściami tych bloków. Natomiast aby zwiększyć pojemność pamięci przez zwiększenie liczby pamiętanych słów należy oprócz odpowiedniej liczby bloków pamięci wykorzystać również dekodery (demultiplekser), którego wejścia adresowe będą sterowane przez najstarsze bity adresu komórki pamięci. Sygnał z wyjścia dekodera będzie sterował wejściami \overline{CS} kolejnych bloków pamięci. Na rysunku poniżej przedstawiono sposób połączenia 4 bloków pamięci 2114 w celu uzyskania pamięci o pojemności 16Kb.



Rys. 39. Powiększenie pojemności pamięci przez zwiększenie liczby pamiętanych słów [1, s. 273]

Układy programowalne

Układ programowalny to taki układ, którego fizyczna struktura wewnętrzna może być zmieniana pod wpływem różnych czynników. W wyniku zmian struktury zmieniają się właściwości elektryczne tego układu, a co za tym idzie, także realizowane funkcje. Istnieją różne rodzaje struktur programowalnych, m. in. układy ASIC (z ang. Application Specific

Integrated Circuits). Są to układy umożliwiające realizację całego, nawet bardzo złożonego systemu cyfrowego. Układ taki jest programowany na etapie produkcji, dlatego też koszt zaprojektowania takiego układu jest na tyle wysoki, że wykorzystywanie ich do budowy układów cyfrowych opłacalne jest dopiero przy produkcji na dużą skalę. Innym przykładem układów programowalnych są programowalne struktury logiczne PLD (z ang. Programmable Logic Devices). Są to cyfrowe układy scalone, których właściwości funkcjonalne mogą być ustalane (programowane) przez użytkownika. Stanowią więc pewien kompromis pomiędzy standardowymi rozwiązaniami kombinacyjnymi, a wysoko specjalizowanymi układami ASIC. Są to niezbyt złożone, dwupoziomowe układy kombinacyjne ułożone w formie matrycy bramki AND i OR.

Mogą być programowane jednokrotnie lub wielokrotnie za pomocą odpowiednich narzędzi programowych.

Wśród tego typu układów wyróżnia się układy:

- PAL (z ang. Programmable Array Logic) z programowalną matrycą bramek AND,
- PLA (z ang. Programmable Logic Array) z programowalnymi matrycami bramek AND i OR,
- PLE (z ang. Programmable Logic Element) z programowalną matrycą bramek OR.

Wyjścia tego typu układów często zaopatrzone są w przerzutniki.

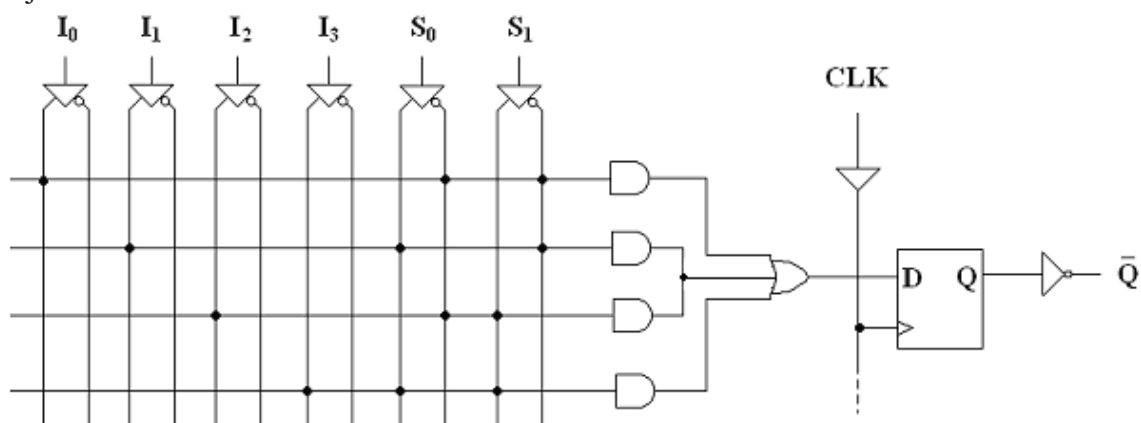
Proste projekty można zrealizować np. przy pomocy układu PAL, rysując docelowy schemat za pomocą symboli elementów wewnętrznych układu PAL, określając położenie bezpieczników, które muszą zostać przepalane. Przepalanie bezpieczników następuje przy użyciu programatora układów PAL.

Na przykład aby zrealizować 4-wejściowy multiplexer za pomocą struktury programowalnej, należy najpierw zapisać równanie logiczne multiplexerowej części układu (tzn. równanie układu doprowadzonego do wejścia D przerzutnika na wyjściu układu):

$$Q_D = I_0 \overline{S_1} \overline{S_0} + I_1 \overline{S_1} S_0 + I_2 S_1 \overline{S_0} + I_3 S_1 S_2$$

Rejestrowy układ PAL umożliwia zatrzaśnięcie otrzymanego wyniku.

Tak więc aby zrealizować powyższą funkcję należy zaznaczyć które bezpieczniki mają zostać przepalane. Powstanie wówczas układ połączony w sposób umożliwiający realizację zadanej funkcji. Poniższy rysunek przedstawia schemat połączeń potrzebnych do realizacji zadanej funkcji.



Rys. 40. Realizacja 4 – wejściowego multiplexera przy użyciu struktury PAL[3, s.70]

4.4.2 Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Narysuj symbole i tablice przejść poznanych przerzutników.
2. Narysuj układ licznika z wykorzystaniem przerzutników.
3. Wyjaśnij zasadę działania poznanych układów liczników.
4. Wyjaśnij zasadę działania poznanych rejestrów monolitycznych.
5. Wymień zastosowania rejestrów w układach cyfrowych.
6. Wyjaśnij zasadę działania pamięci półprzewodnikowych.
7. Wyjaśnij zasadę zwiększania pojemności pamięci.
8. Wyjaśnij zasadę działania układów PAL.

4.4.3 Ćwiczenia

Ćwiczenie 1

Zastosowanie przerzutników w układach cyfrowych – zaprojektować licznik 3 – bitowy z wykorzystaniem przerzutników T.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych przerzutników scalonych,
- 2) zaproponować schemat układu licznika oraz schemat pomiarowy do sprawdzenia poprawności działania licznika,
- 3) zaproponować tabele, w których będą zapisywane wyniki pomiarów,
- 4) połączyć układ według schematu,
- 5) zbadać działanie licznika sterując wejściem zegarowym układu, wyniki pomiarów zapisywać w tabeli, wykonać przynajmniej 18 pomiarów,
- 6) narysować przebiegi czasowe w liczniku na podstawie otrzymanych wyników,
- 7) przeanalizować wyniki i sformułować wnioski,
- 8) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca przerzutniki scalone,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 2

Zaprojektować układ zliczający od 15 do 43.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych liczników scalonych,

- 2) zaproponować sposób połączenia dostępnych liczników, aby otrzymać układ zliczający mod100,
- 3) zaproponować sposób skrócenia licznika tak, aby realizował założenia zadania,
- 4) zaproponować schemat pomiarowy do sprawdzenia poprawności działania zaprojektowanego układu oraz tabelę do zapisywania wyników pomiarów,
- 5) połączyć układ według schematu,
- 6) zbadać działanie licznika sterując wejściem zegarowym układu, wyniki pomiarów zapisywać w tabeli, wykonać przynajmniej 30 pomiarów,
- 7) przeanalizować wyniki i sformułować wnioski,
- 8) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca liczniki scalone,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 3

Zastosowanie rejestrów w układach cyfrowych – zaprojektować układ mnożący liczbę 6 – bitową przez 4.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną dostępnych rejestrów scalonych,
- 2) zaproponować schemat układu realizującego operacje mnożenia liczby przez 4 oraz schemat pomiarowy do sprawdzenia działania układu
- 3) zaproponować tabelę do zapisywania wyników pomiarów,
- 4) połączyć układ według schematu,
- 5) zbadać działanie układu, wyniki pomiarów zapisywać w tabeli, wykonać pomiary dla co najmniej trzech operacji mnożenia (trzy różne liczby 6 – bitowe),
- 6) przeanalizować wyniki i sformułować wnioski,
- 7) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca rejestry scalone,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 4

Zbadać działanie pamięci RAM.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną pamięci RAM,
- 2) zaproponować schemat pomiarowy do zbadania działania pamięci RAM,
- 3) zaproponować tabele do zapisywania wyników pomiarów,
- 4) połączyć układ według schematu
- 5) zapisać dziesięć komórek pamięci dowolnymi danymi,
- 6) zapisać w tabelach wartości zapisanych słów oraz adresy komórek pamięci, do których zapisano dane,
- 7) odczytać zawartość zapisanych wcześniej komórek,
- 8) wyłączyć i ponownie włączyć napięcie zasilające układ,
- 9) odczytać ponownie zawartość komórek o adresach zapisanych w tabeli,
- 10) przeanalizować wyniki i sformułować wnioski,
- 11) dokonać oceny poprawności wykonanego ćwiczenia.

Uwaga: Zanim zostanie przyłączone napięcie, połączony układ pomiarowy musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- zadajniki i wskaźniki stanów logicznych,
- makieta zawierająca pamięci półprzewodnikowe,
- przewody połączeniowe,
- dokumentacja techniczna badanych układów scalonych, arkusze białego papieru A4.

Ćwiczenie 5

Zaprojektować układ realizujący funkcję $y = \overline{(a + \overline{b})(cd + a)}$ przy pomocy układu PAL .

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z dokumentacją techniczną układu programowalnego PAL,
- 2) zaproponować sposób realizacji funkcji y przy pomocy układu programowalnego PAL,
- 3) narysować strukturę układu z zaznaczeniem bezpieczników, które nie zostaną przepalone,
- 4) dokonać oceny poprawności wykonanego ćwiczenia.

Wyposażenie stanowiska pracy:

- dokumentacja techniczna układów PAL,
- arkusze białego papieru A4.

4.4.4 Sprawdzian postępów

Czy potrafisz	Tak	Nie
1) narysować symbole i tablice przejść poznanych przerzutników?	<input type="checkbox"/>	<input type="checkbox"/>
2) zaprojektować i zmontować układ licznika z wykorzystaniem przerzutników?	<input type="checkbox"/>	<input type="checkbox"/>
3) zaprojektować i zmontować układ licznika o dowolnej długości cyklu?	<input type="checkbox"/>	<input type="checkbox"/>
4) wyjaśnić zasadę działania poznanych rejestrów monolitycznych?	<input type="checkbox"/>	<input type="checkbox"/>
5) zastosować rejestry w układach cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
6) zastosować układ pamięci półprzewodnikowej w układach cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
7) rozbudować układ pamięci zwiększając jej pojemność?	<input type="checkbox"/>	<input type="checkbox"/>
8) zrealizować funkcję logiczną przy pomocy układu PAL	<input type="checkbox"/>	<input type="checkbox"/>
9) rozpoznać na symbole graficzne i oznaczenia stosowane na cyfrowych układach scalonych?	<input type="checkbox"/>	<input type="checkbox"/>
10) zaprojektować prosty układ cyfrowy z wykorzystaniem poznanych elementów?	<input type="checkbox"/>	<input type="checkbox"/>

6. LITERATURA

1. Głocki W.: Układy cyfrowe. WSiP, Warszawa 2002
2. Głocki W.: Grabowski L.: Pracownia podstaw techniki cyfrowej, WSiP, Warszawa 1998
3. Hill W.: Horowitz P.: Sztuka elektroniki 1 i 2. WKŁ, Warszawa 2001
4. Marusak A.: Urządzenia elektroniczne. WSiP, Warszawa 2000
5. Pasierbiński J., Rusek M.: Elementy i układy elektroniczne w pytaniach i odpowiedziach. WNT, Warszawa 1999
6. Pióro B., Pióro M.: Podstawy elektroniki. WSiP, Warszawa 2002
7. Schenk Ch., Tietze U.: Układy półprzewodnikowe. WNT, Warszawa 1996
8. Zioło K. (red): Laboratorium elektroniki II. Podstawowe układy analogowe, impulsowe i cyfrowe. Wydawnictwo Politechniki Śląskiej, Gliwice 2000