

UKŁADY CYFROWE LABORATORIUM – informacje pomocnicze

Informacje:

- a) Strona czołowa sprawozdania studenckiego z zajęć laboratoryjnych
- b) Układy cyfrowe i elementy elektroniczne dostępne na każdym stanowisku laboratoryjnym
- c) Przykładowe rozwiązania wybranych zadań laboratoryjnych

a) Strona czołowa sprawozdania studenckiego z zajęć laboratoryjnych

Politechnika Gdańska	Wydział ETI	KSA - KSDiR
UKŁADY CYFROWE – LABORATORIUM		
CYKL ĆWICZEŃ GRUPOWYCH	ĆWICZENIE NR ...	
Tytuł ćwiczenia:		
Skład grupy ćwiczeniowej:		Rok studiów:
1. Imię i Nazwisko nr indeksu	Semestr:
2. Imię i Nazwisko nr indeksu	Grupa dziekańska:
3. Imię i Nazwisko nr indeksu	Specjalność:
Sprawdzenie stanu przewodów z wtyczkami bananowymi: - przewody koloru: - sztuk - przewody koloru: - sztuk		
POTWIERDZENIE WYKONANIA ZADAŃ: 1. Zadanie 1: 2. Zadanie 2: 3. Zadanie 3:		DATA WYKONANIA ĆWICZENIA: OCENA:

b) Układy cyfrowe i elementy elektroniczne dostępne na każdym stanowisku laboratoryjnym

ĆWICZENIE 2 - BADANIE BRAMEK LOGICZNYCH

Na stanowisku dydaktycznym dostępne są:

- układ 7400 - bramki NAND 2 wejściowe (4 szt.),
- układ 7437 - bramki NAND 2 wejściowe (4 szt.),
- układ 74HCT132 - bramki NAND z histerezą 2 wejściowe (4 szt.),
- układ 7406 - bramki NOT z otwartym kolektorem (6 szt.),
- układ 75451 - bramki NAND z wyjściem analogowym 2 wejściowe (2 szt.).

Pozostałe układy:

- układ 7404,
- układ 74LS32,
- układ 74LS86,
- układ 74LS125,
- układ 74HCT02,
- układ 74HCT04,
- układ 74HC08,
- układ 74HC4050,
- układ MC14504B,
- układ 4009B,
- układ 4071B,
- układ 4011B,
- układ 4049B.

Inne dostępne elementy:

- diody świecące (LED): czerwona, żółta, zielona, czerwono-zielona,
- diody,
- tranzystory,
- transoptory,
- przekaźniki,
- oporniki:

$R_1 = 150\Omega$, $R_2 = 510\Omega$, $R_3 = 2.2k\Omega$, $R_4 = 4.7k\Omega$,
 $R_5 = 12k\Omega$, $R_6 = 30k\Omega$, $R_7 = 150\Omega$, $R_8 = 510\Omega$,
 $R_9 = 1k\Omega$, $R_{10} = 4.7k\Omega$, $R_{11} = 10k\Omega$, $R_{12} = 22k\Omega$,
 $R_{13} = 100\Omega$, $R_{14} = 200\Omega$, $R_{15} = 390\Omega$, $R_{16} = 510\Omega$,
 $R_{17} = 750\Omega$, $R_{18} = 1k\Omega$, $R_{19} = 1k\Omega$, $R_{20} = 1.5k\Omega$,
 $R_{21} = 3.3k\Omega$, $R_{22} = 4.7k\Omega$, $R_{23} = 10k\Omega$, $R_{24} = 12k\Omega$,
 $R_z = 12\Omega$.

Na stanowisku dostępny jest oscyloskop 2-kanałowy.

Łączenie układów odbywa się za pomocą przewodów bananowych.

ĆWICZENIE 3 - UKŁADY ITERACYJNE

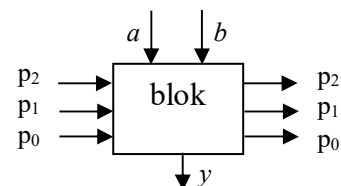
Na stanowisku dydaktycznym dostępne są:

- multiplexery mpx 16/1 (4 szt.) - do realizacji logiki pojedynczego bloku iteracyjnego,
- multiplexer mpx 8/1 (1 szt.) - do realizacji logiki dekodera wyjściowego (na końcu kaskady).

Blok może mieć 1 lub 2 wejścia (dane binarne zadawane z klawiszy).

Blok może mieć 1, 2 lub 3 przeniesienia.

Blok może mieć 1 indywidualne wyjście.



Wszystkie multiplexery mpx 16/1 mają wspólne adresowanie (D C B A):
(D = A₃ , C = A₂ , B = A₁ , A = A₀).

Adresowanie multiplexera mpx 8/1 realizującego funkcję dekodera wyjściowego (C B A):
(C = p₂ , B = p₁ , A = p₀).

Za pomocą multiplexerów realizowany jest jeden blok iteracyjny.

Testując zaprojektowany układ obserwujemy pracę kaskady 8 identycznych bloków.

Stałe logiczne (0 i 1) zadawane są z przełączników.

Łączenie układów odbywa się za pomocą mini-przewodów.

ĆWICZENIE 4 - UKŁADY CZASOWE

Na stanowisku dydaktycznym dostępne są:

- układy monostabilne 74121 (2 szt., obydwa z możliwością regulacji stałej czasowej T),
- układy monostabilne 74123 (2 szt., jeden z możliwością regulacji stałej czasowej T),
- układ czasowy 555,
- układ czasowy 4047,
- licznik synchroniczny 74193,
- multiplexer mpx 8/1 (74151),
- przerzutniki synchroniczne D (2 szt.),
- bramki NAND 3 wejściowe (3 szt.),
- bramki NAND z histerezą 2 wejściowe (3 szt.),
- bramki NOT (3 szt.),
- diody (2 szt.),
- przełączniki (8 szt.) do zadawania stałych logicznych (0 i 1).

Na stanowisku dostępny jest oscyloskop umożliwiający obserwację 4 przebiegów TTL (na kanale A) i przebieg analogowy (na kanale B).

Łączenie układów odbywa się za pomocą przewodów bananowych.

ĆWICZENIE 5 - UKŁADY SYNCHRONICZNE

Na stanowisku dydaktycznym dostępne są:

- przerzutniki synchroniczne JK (5 szt.),
- bramki NAND 2 wejściowe (9 szt.),
- bramki NAND 3 wejściowe (8 szt.),
- bramki NAND 4 wejściowe (6 szt.),
- bramki XOR 2 wejściowe (4 szt.),
- bramki NOT (10 szt.),
- bramki AND-OR-INVERT 2 we. + 2 we. (6 szt.),
- bramki AND-OR-INVERT 3 we. + 3 we. (6 szt.),
- rejestry przesuwne z wpisem równoległym (dane we/wy),
- przełączniki (8 szt.) do zadawania stałych logicznych (0 i 1).

Wejścia zegarowe przerzutników są podłączone do wbudowanego układu taktującego (użytkownik nie ma dostępu do wejść zegarowych przerzutników).

Łączenie układów odbywa się za pomocą mini-przewodów.

ĆWICZENIE 6 - LICZNIKI SCALONE

Na stanowisku dydaktycznym dostępne są:

- licznik asynchroniczny 7490,
- licznik asynchroniczny 7492,
- licznik asynchroniczny 7493,
- licznik synchroniczny 74160,
- licznik synchroniczny 74190 (2 szt.),
- licznik synchroniczny 74192,
- licznik synchroniczny 74193,
- układy 7447 dekodujące BCD na kod 7-segm. z wyświetlaczami (2 szt.),
- bramki NAND 3 wejściowe (2 szt.),
- bramki NAND 2 wejściowe (2 szt.),
- bramki XOR 2 wejściowe (2 szt.),
- bramki NOT (3 szt.),
- przerzutnik asynchroniczny SR,
- przełączniki (6 szt.) do zadawania stałych logicznych (0 i 1).

Łączenie układów odbywa się za pomocą mini-przewodów.

Uwaga:

W zestawie laboratoryjnym wyprowadzenia wszystkich liczników scalonych nie posiadają oznaczeń symbolicznych, ale są odpowiednio ponumerowane (zgodnie z numeracją podaną na kartach katalogowych odpowiednich układów). Dlatego na każdym przygotowanym schemacie muszą się pojawić numery wszystkich wyprowadzeń (w przeciwnym razie uruchomienie układu liczącego nie będzie możliwe).

ĆWICZENIE 7 - REJESTRY SCALONE

Na stanowisku dydaktycznym dostępne są:

- rejestry przesuwne z szeregowym wejściem i równoległym wyjściem 74164 (2 szt.),
- rejestry przesuwne z równoległym wejściem i szeregowym wyjściem 74165 (2 szt.),
- rejestry przesuwne z równoległym wejściem i równoległym wyjściem 74198 (2 szt.),
- sumator iteracyjny 1-bitowy,
- sumator 4-bitowy: układ 7483,
- przerzutniki synchroniczne D (4 szt.): układ 74175,
- bramki NAND 2 wejściowe (4 szt.),
- bramki XOR 2 wejściowe (8 szt.),
- bramki NOT (4 szt.),
- przełączniki (8 szt.) do zadawania stałych logicznych (0 i 1).

Łączenie układów odbywa się za pomocą przewodów bananowych.

ĆWICZENIE 8 - UKŁADY ASYNCHRONICZNE

Na stanowisku dydaktycznym dostępne są:

- bramki NAND 4 wejściowe (4 szt.),
- bramki NAND 3 wejściowe (6 szt.),
- bramki NAND 2 wejściowe (12 szt.),
- bramki NOT (6 szt.),
- przerzutniki asynchroniczne SR (4 szt.),
- rejestry (3 szt.) do wprowadzania sekwencji binarnych (sygnały wejściowe).

Na stanowisku dostępny jest oscyloskop umożliwiający obserwację 4 przebiegów TTL.

Łączenie układów odbywa się za pomocą przewodów bananowych.

ĆWICZENIE 9 - STEROWANIE SZYNĄ DANYCH

Ćwiczenie ilustruje sposób przesyłania informacji pomiędzy kilkoma układami. Szyna danych, jak też wszystkie nadajniki i odbiorniki są układami 4-bitowymi. W zestawie wykorzystuje się układ arytmetyczno-logiczny (ALU) 74181. W ćwiczeniu nie wykonuje się jakichkolwiek połączeń za pomocą przewodów (zadanie polega na napisaniu prostego programu).

Nadajnikiem może być:

- rejestr A (RA),
- rejestr B (RB),
- rejestr C (RC),
- układ arytmetyczno-logiczny (ALU),
- klawiatura (4 klawisze) do wprowadzania 4-bitowych danych wejściowych (DANE),
- klawiatura do zadawania instrukcji (INSTRUKCJA).

Odbiornikiem może być:

- rejestr A (RA),
- rejestr B (RB),
- rejestr C (RC),
- rejestr pomocniczy (R1),
- rejestr pomocniczy (R2),
- rejestr wyjściowy (Rwy),
- rejestr instrukcji (RI).

Aby wysłać daną z wybranego nadajnika (ozn. symbolem T) do wybranego odbiornika (ozn. symbolem C) należy na odpowiednie wejścia T i C podać sygnał logiczny 1.

Linia programu (mikrocycł) ma postać:

I ₂	I ₁	I ₀	C ₁	C ₀	C _T	B _T	A _T	C _C	B _C	A _C	W ₁	W ₂
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

I₂ I₁ I₀ - oznacza numer instrukcji (od 000 do maksymalnie 111)

C₁ C₀ - oznacza numer mikrocycłu (00, 01, 10, 11)

Kodowanie wyboru nadajnika:

C_T B_T A_T = 0 0 0 (INSTRUKCJA)

C_T B_T A_T = 0 0 1 (RA)

C_T B_T A_T = 0 1 0 (RB)

C_T B_T A_T = 0 1 1 (RC)

C_T B_T A_T = 1 0 0 (ALU)

C_T B_T A_T = 1 0 1 (DANE)

Kodowanie wyboru odbiornika:

C_C B_C A_C = 0 0 0 (RI)

C_C B_C A_C = 0 0 1 (RA)

C_C B_C A_C = 0 1 0 (RB)

C_C B_C A_C = 0 1 1 (RC)

C_C B_C A_C = 1 0 0 (R1)

C_C B_C A_C = 1 0 1 (R2)

C_C B_C A_C = 1 1 0 (Rwy)

Jeśli dana instrukcja programu wykonuje operację arytmetyczno-logiczną (układ 74181), to w kolumnach W_1 i W_2 linii programu należy wpisać właściwe sygnały $S_3 S_2 S_1 S_0$ (zgodnie z tabelą ALU).

I_2	I_1	I_0	C_1	C_0	C_T	B_T	A_T	C_C	B_C	A_C	W_1	W_2
x	x	x	0	0							M	S_3
			0	1							S_2	S_1
			1	0							x	S_0
			1	1							x	x

Układ arytmetyczno-logiczny 74181 (ALU) realizuje funkcje podane w tabeli (dane A i B są 4-bitowe).

S_3	S_2	S_1	S_0	M = 0 funkcje arytmetyczne	M = 1 funkcje logiczne
0	0	0	0	$Y = 0$	$Y = \overline{R_1}$
0	0	0	1	$Y = R_1 + R_2$	$Y = \overline{R_1} \wedge \overline{R_2}$
0	0	1	0	$Y = R_1 \times R_2 - 1$	$Y = \overline{R_1} \vee R_2$
0	0	1	1	$Y = R_1 \div R_2$	$Y = 1$
0	1	0	0	$Y = R_1^2 - R_2^2$	$Y = R_1 \oplus R_2$
0	1	0	1	$Y = R_1 \times 2$	$Y = \overline{R_1} \Leftrightarrow \overline{R_2}$
0	1	1	0	$Y = (R_1 + R_2) / 2$	$Y = R_1 \Rightarrow R_2$
0	1	1	1	$Y = R_1 \times R_2$	$Y = R_1 \vee R_2$
1	0	0	0	$Y = R_1 \% R_2$	$Y = \overline{R_1} \Rightarrow \overline{R_2}$
1	0	0	1	$Y = R_1 + 2$	$Y = \overline{\overline{R_1} \wedge R_2}$
1	0	1	0	$Y = (R_1 \times R_2) - (R_1 + R_2)$	$Y = R_1 \wedge R_2$
1	0	1	1	$Y = R_1 - R_2$	$Y = \overline{R_1} \oplus \overline{R_2}$
1	1	0	0	$Y = (R_1 - R_2)^2$	$Y = \overline{R_2}$
1	1	0	1	$Y = R_1$	$Y = \overline{R_1} \vee \overline{R_2}$
1	1	1	0	$Y = R_1^2$	$Y = R_1 \Leftrightarrow R_2$
1	1	1	1	$Y = R_1 - 1$	$Y = \overline{R_1} \oplus \overline{R_2}$

Aby wykonać wybrane działanie należy wpisać liczby do rejestrów R1 i R2.

W przypadku wykonywania operacji dodawania arytmetycznego układ nie udostępnia bitu przeniesienia.

W dalszej części podany jest przykładowy program.

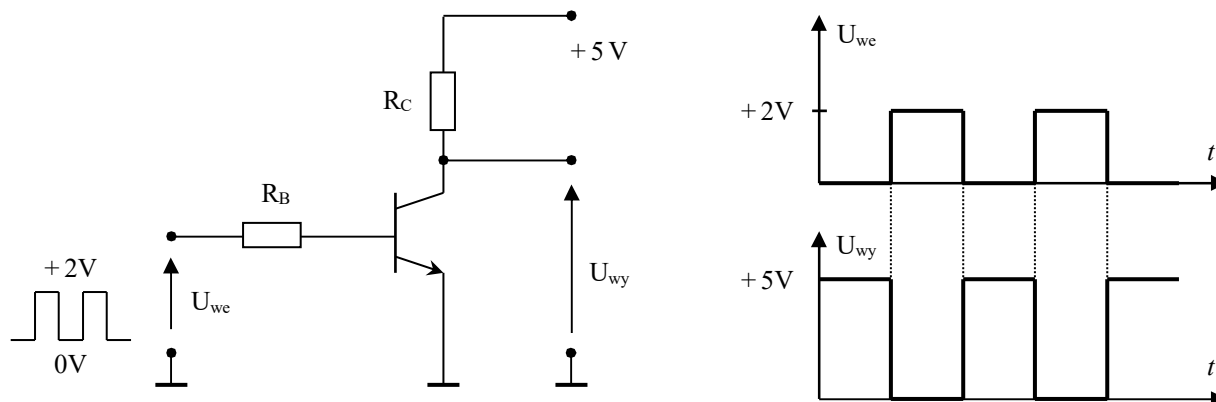
c) Przykładowe rozwiązania wybranych zadań laboratoryjnych

Ćwiczenie 2: Badanie bramek logicznych

Zadanie:

Zbudować układ zamieniający przebieg prostokątny (0 – 2V) na przebieg prostokątny (0 – 5V).

Zmianę amplitudy lub przesuwanie poziomu sygnału uzyskuje się stosując klucze tranzystorowe.



Zakładając, że napięcie wejściowe zmienia się skokowo w zakresie $0\text{ V} \div 2\text{ V}$ przy napięciu zasilania wynoszącym $U_{CC} = +5\text{ V}$, można dobrać wartości rezystorów następująco:

$$R_B = (U_{we} - U_{BE}) / I_B = (2\text{ V} - 0,7\text{ V}) / 1\text{ mA} = 1,3\text{ k}\Omega \quad , \quad R_C = 1\text{ k}\Omega \div 4,7\text{ k}\Omega .$$

Wówczas dla $U_{we} = 2\text{ V}$ tranzystor nasycy się poprzez wymuszenie prądu bazy $I_B = 1\text{ mA}$. Wartość rezystora R_C natomiast jest typowa, gdy sygnał wyjściowy ma być podany dalej na wejście TTL (logiczne „1” w układach TTL to napięcie $+5\text{ V}$ podłączone do rezystora $1\text{ k}\Omega \div 4,7\text{ k}\Omega$).

(UWAGA! W SPRAWOZDANIU KAŻDY SCHEMAT MUSI BYĆ NARYSOWANY RĘCZNIE)

Ćwiczenie 3: Układy iteracyjne

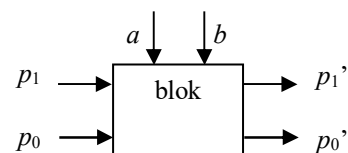
Zadanie:

Zbudować iteracyjny komparator liczb binarnych poczynając od bitów młodszych.

Tabela komparatora liczb binarnych (od bitów młodszych)

p_1	p_0	a	b	p_1'	p_0'
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	x	x
1	1	0	1	x	x
1	1	1	0	x	x
1	1	1	1	x	x

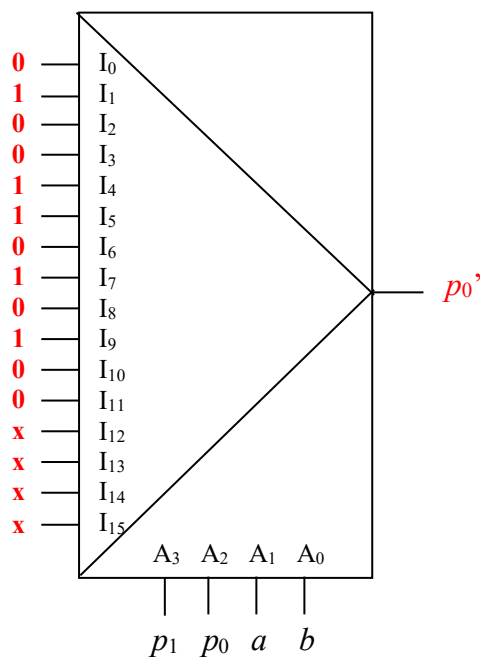
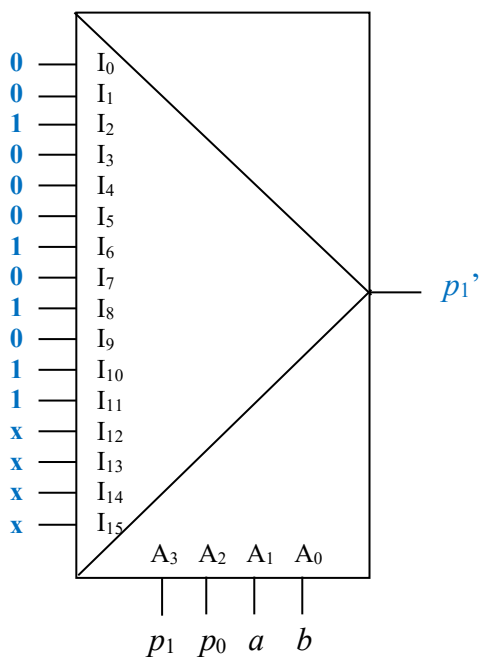
Przeniesienia – kodowanie			
p_1	p_0		
0	0	$a = b$ (równe)	
0	1	$a < b$ (a mniejsze)	
1	0	$a > b$ (a większe)	
1	1	x (nie używane)	



$$p_1' = f(p_1, p_0, a, b) = \Sigma(2, 6, 8, 10, 11, (12, 13, 14, 15))$$

$$p_0' = g(p_1, p_0, a, b) = \Sigma(1, 4, 5, 7, 9, (12, 13, 14, 15))$$

Realizacja na dwóch mpx 16/1: (przeniesienia początkowe: $p_1 p_0 = 0 0$)



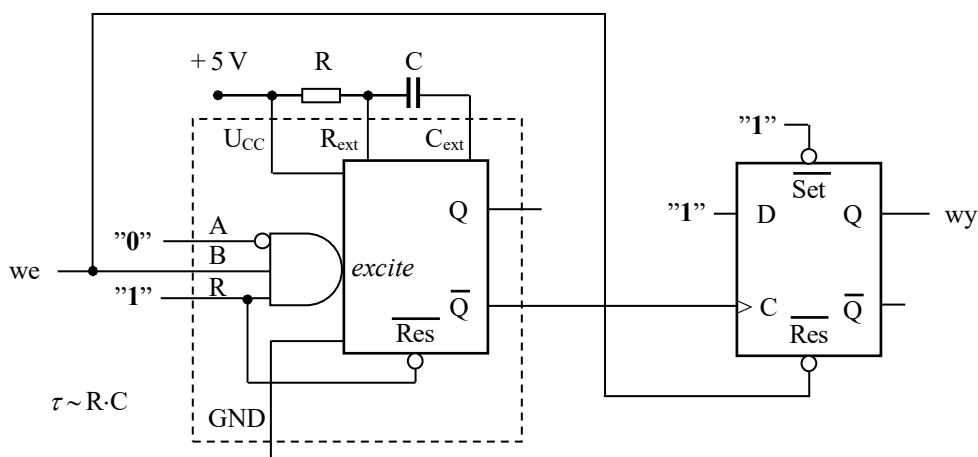
(UWAGA! W SPRAWOZDANIU KAŻDY SCHEMAT MUSI BYĆ NARYSOWANY RĘCZNIE)

Ćwiczenie 4: Układy czasowe

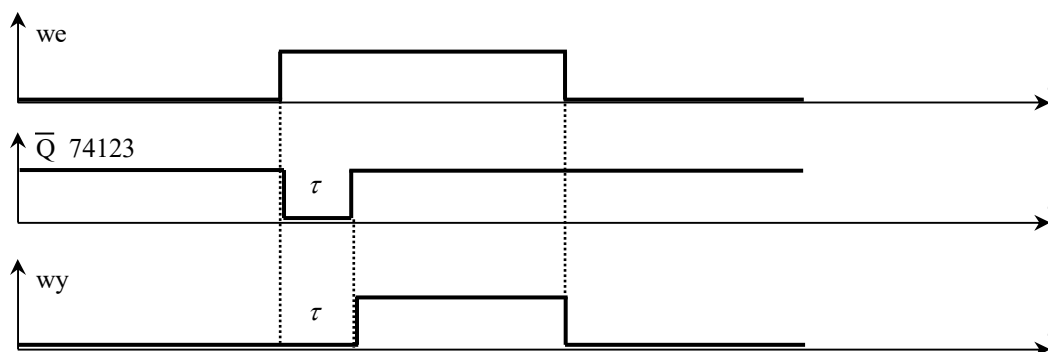
Zadanie:

Zbudować układ opóźniający narastające zbocze impulsu wejściowego o czas τ (regulowany).

Układ przepuszcza impuls wejściowy (we) na wyjście (wy) opóźniając pojawienie się jego narastającego zbocza o regulowany czas τ ($\tau \sim R \cdot C$). Narastające zbocze impulsu wejściowego powoduje uruchomienie układu monostabilnego 74123 odmierzającego impuls o czasie τ . Jednocześnie stan wysoki na wejściu odblokowuje przerzutnik D ustawiając stan "1" na wejściu kasującym (Res) przerzutnika. Zakończenie impulsu odmierzanego przez układ monostabilny powoduje, że na wyjściu negowanym układu 74123 pojawia się zbocze narastające. Zbocze to przepisuje wartość "1" z wejścia D przerzutnika na wyjście Q przerzutnika (wpis synchroniczny). W momencie zakończenia impulsu wejściowego niski stan wejścia (podłączonego do wejścia kasującego Res przerzutnika) zmienia wyjście przerzutnika na "0". Impuls wyjściowy obserwowany na końcówce Q przerzutnika kończy się zatem w chwili zakończenia impulsu wejściowego, ale jego narastające zbocze jest opóźnione o czas τ (odmierzany przez układ 74123) w relacji do impulsu wejściowego.



Działanie układu ilustruje przebieg czasowy.



(UWAGA! W SPRAWOZDANIU KAŻDY SCHEMAT MUSI BYĆ NARYSOWANY RĘCZNIE)

Ćwiczenie 5: Układy synchroniczne

Zadanie:

Zbudować układ synchroniczny konwersji szeregowej kodu binarnego na kod Graya.

Zamianę kodu binarnego na kod Graya opisuje wzór:

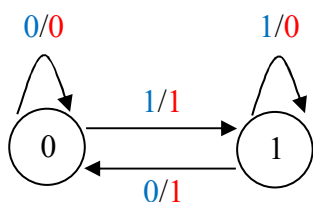
$$\begin{cases} g_n = b_n & \text{przepisać najstarszy bit} \\ g_i = b_{i+1} \oplus b_i & \text{dla } i = N-1 \dots 0 \end{cases}$$

Konwersja realizowana jest od bitów starszych do bitów młodszych.

Stan układu to zapamiętany poprzedni bit binarny ($q_i = b_{i+1}$), a wejściem jest bieżący bit binarny (b_i).

Wyjście układu (bit w kodzie Graya) wynika ze wzoru: $g_i = q_i \oplus b_i$.

Graf układu oraz tabele przejść stanów (klasyczna i Karnaugh) mają postać:



q	b	q'	g
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

$q \backslash b$	0	1
0	0/0	1/1
1	0/1	1/0

Sterowanie przerzutnikiem JK dla bitu q

Tabela wyjścia g

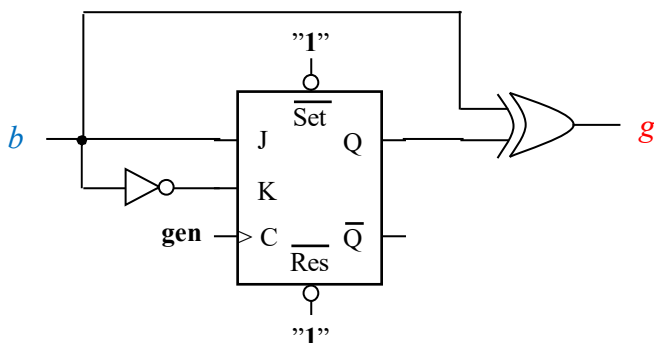
Zasada działania przerzutnika JK

$q \backslash b$	0	1
0	0/x	1/x
1	x/1	x/0

$q \backslash b$	0	1
0	0	1
1	1	0

Q	Q'	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Wyniki: $J = b$, $K = \bar{b}$, $g = \bar{q} \cdot b + q \cdot \bar{b} = q \oplus b$



Stan początkowy przerzutnika: $q = 0$.

Liczbę binarną należy wpisać do rejestru i wyjście rejestru podłączyć do wejścia b układu.

Wynik g konwersji szeregowej wpisywać do drugiego rejestru.

W zestawie wejścia zegarowe wszystkich układów są podłączone do wspólnego sygnału taktującego (gen).

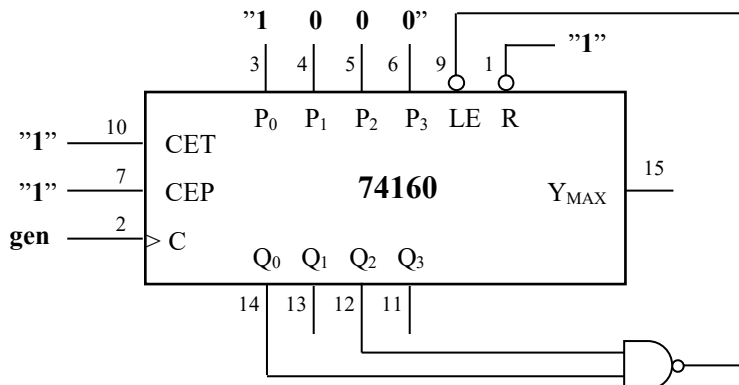
(UWAGA! W SPRAWOZDANIU KAŻDY SCHEMAT MUSI BYĆ NARYSOWANY RĘCZNIE)

Ćwiczenie 6: Liczniki scalone

Zadanie:

Zmodyfikować licznik 74160, tak aby liczyć: 1 ... 5.

Sygnal synchronicznego wpisu (LE) ładuje wejścia $P_3 P_2 P_1 P_0 = "0001"$ na wyjścia $Q_3 Q_2 Q_1 Q_0$ licznika.



W podanym rozwiązaniu wykrywany kod binarny "0101" ($Q = 5$) jest ostatnim stanem należącym do grafu licznika (liczenie 1 ... 5). Ponieważ wejście ładujące ma charakter synchroniczny, podanie sygnału aktywnego ($LE = 0$) nie wywołuje początkowo żadnego efektu. Dopiero pojawienie się najbliższego narastającego zbocza sygnału taktującego (tzn. zbocza wyznaczającego koniec stanu $Q = 5$) powoduje załadowanie wejść równoległych "0001" ($P = 1$) na wyjście.

Uwaga:

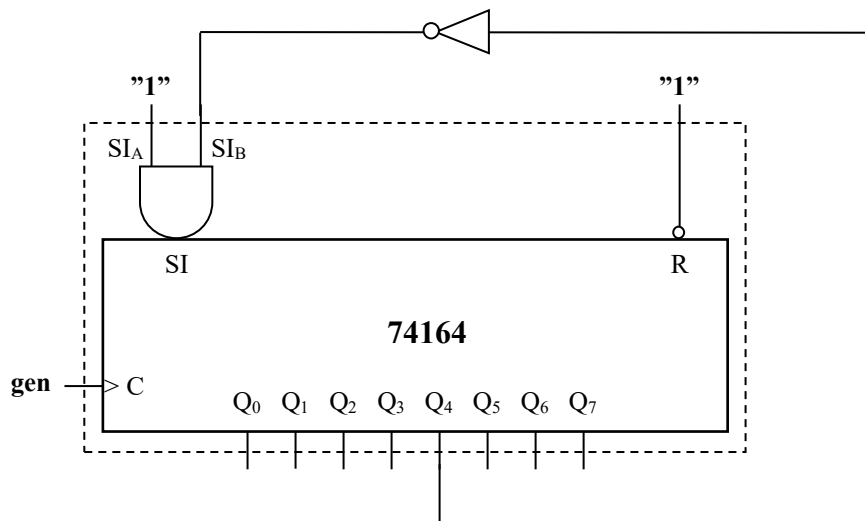
W zestawie laboratoryjnym wyprowadzenia wszystkich liczników scalonych nie posiadają oznaczeń symbolicznych, ale są odpowiednio ponumerowane (zgodnie z numeracją z kart katalogowych odpowiednich układów). Dlatego na każdym schemacie wykorzystywanym w laboratorium muszą się pojawić numery wszystkich wyprowadzeń. Brak takich oznaczeń uniemożliwia uruchomienie zaprojektowanego licznika.

(UWAGA! W SPRAWOZDANIU KAŻDY SCHEMAT MUSI BYĆ NARYSOWANY RĘCZNIE)

Ćwiczenie 7: Rejestry scalone

Zadanie:

Zbudować licznik Johnsona (4 bity, 8 stanów).



Zadanie było rozwiązywane na zajęciach projektowych.

Wynik liczenia obserwowany jest na wyjściach Q₀ Q₁ Q₂ Q₃ rejestru.

Sekwencja liczenia: 0000 → 1000 → 1100 → 1110 → 1111 → 0111 → 0011 → 0001 → 0000 itd.

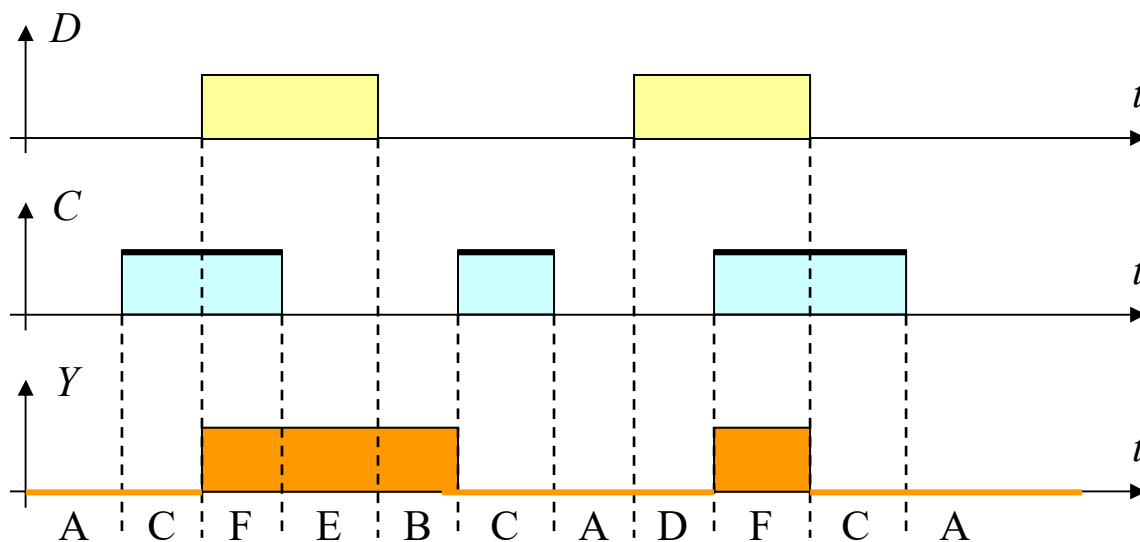
(UWAGA! W SPRAWOZDANIU KAŻDY SCHEMAT MUSI BYĆ NARYSOWANY RĘCZNIE)

Ćwiczenie 8: Układy asynchroniczne

Zadanie:

Zbudować synchroniczny przerzutnik typu D z atrzask.

Dla $C=1$ dana D przechodzi na wyjście Y , a dla $C=0$ wyjście Y nie zmienia się (zatrask).

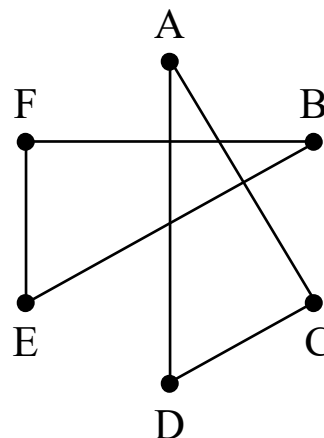


$D \ C$		00	01	11	10	Y
Q	A	A	C	-	D	0
	B	B	C	-	E	1
	C	A	C	F	-	0
	D	A	-	F	D	0
	E	B	-	F	E	1
	F	-	C	F	E	1

D	C	Y	stan Q
0	0	0	= A
0	0	1	= B
0	1	0	= C
0	1	1	niemożliwe
1	0	0	= D
1	0	1	= E
1	1	0	niemożliwe
1	1	1	= F

Minimalizacja liczby stanów:

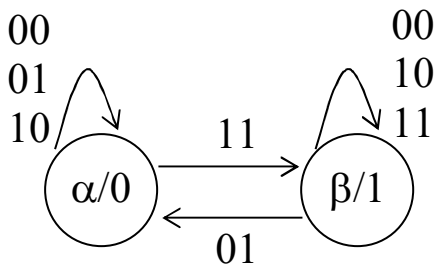
	A	B	C	D	E
B	×				
C	○	×			
D	○	×	○		
E	×	○	×	×	
F	×	○	×	×	○



Pary stanów zgodnych: (AC, AE, AG, BF, CG, DF, DH, FH)

Rodzina minimalna (skreślenia nie zachodzą): { ACD , BEF }.

Graf minimalny Moore'a: ($\alpha = \text{ACD}$, $\beta = \text{BEF}$), $\alpha = 0$, $\beta = 1$.



$$q' = D \cdot C + q \cdot \bar{C} + q \cdot D, \quad Y = q$$

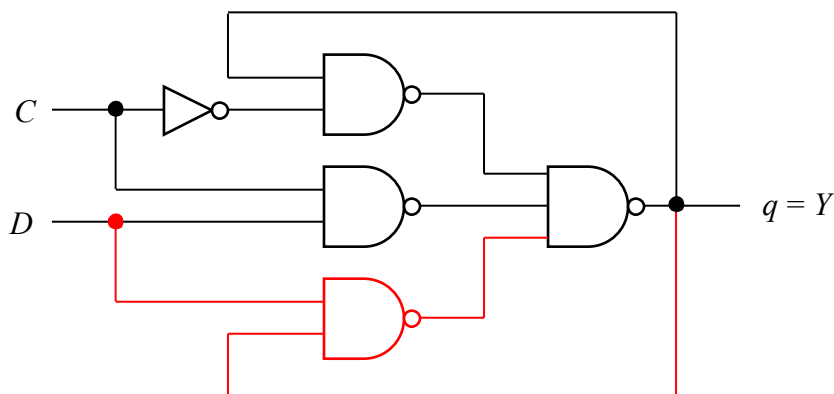
$D C$	0 0	0 1	1 1	1 0
q				
0	0	0	1	0
1	1	0	1	1

sterowanie S R dla q : $\bar{S} = \bar{D} + \bar{C}$, $\bar{R} = D + \bar{C}$.

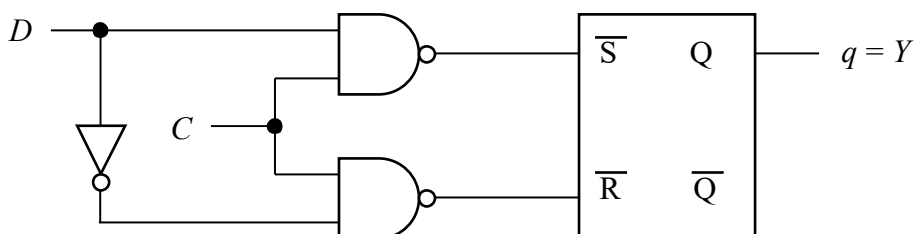
$D C$	0 0	0 1	1 1	1 0
q				
0	1 x	1 x	0 1	1 x
1	x 1	1 0	x 1	x 1

Q	Q'	\bar{S}	\bar{R}
0	0	1	x
0	1	0	1
1	0	1	0
1	1	x	1

Rozwiązanie 1: Sieć sprzężeniowa NAND (bramka oznaczona kolorem czerwonym usuwa hazard).



Rozwiązanie 2: Sieć z przerzutnikiem S R.



(UWAGA! W SPRAWOZDANIU KAŻDY SCHEMAT MUSI BYĆ NARYSOWANY RĘCZNIE)

Ćwiczenie 9: Sterowanie szyną danych

Zadanie:

Napisać i zakodować program (może być maksymalnie 8 instrukcji):

- wczytać dwie liczby 4-bitowe A i B (do rejestrów RA i RB),
- wyzerować rejestr Rwy (wykorzystując operację XOR),
- wykonać dodawanie arytmetyczne (A plus B) i wynik wysłać do rejestru RC.

I ₂	I ₁	I ₀	C ₁	C ₀	C _T	B _T	A _T	C _C	B _C	A _C	W ₁	W ₂
a)	0	0	0	0	0	0	0	0	0	0	x	x
			0	1	1	0	1	0	1	0	x	x
			1	0	0	1	0	0	0	1	x	x
			1	1	1	0	1	0	1	0	x	x
b)	0	1	0	0	0	0	0	0	0	0	1	0
			0	1	0	0	1	1	0	0	1	0
			1	0	0	0	1	1	0	1	x	0
			1	1	1	0	0	1	1	0	x	x
c)	0	1	0	0	0	0	0	0	0	0	0	0
			0	1	0	0	1	1	0	0	0	0
			1	0	0	1	0	1	0	1	x	1
			1	1	1	0	0	0	1	1	x	x

Bit y C_T B_T A_T C_C B_C A_C W₁ W₂ (oznaczone kolorem czerwonym) kolejnych linii programu oraz numer instrukcji (I₂ I₁ I₀) wprowadzane są z klawiatury, natomiast numer mikrocyklu (C₁ C₀) zmienia się automatycznie i jest widoczny na diodach świecących.

Uwaga:

Każda instrukcja jest realizowana w 4 mikrocyklach. Pierwszy mikrocykl (tj. dla C₁ C₀ = 0 0) każdej instrukcji ma zawsze postać (C_T B_T A_T C_C B_C A_C = 0 0 0 0 0 0).