



Politechnika Łódzka

Instytut Fizyki

Laboratorium elektroniki

Ćwiczenie E53

Realizacja asynchronicznych układów
sekwencyjnych z bramek NAND

Spis treści:

1. Cel ćwiczenia.....	3
2. Zagrożenia	3
3. Wprowadzenie teoretyczne.....	3
3.1. Opis asynchronicznego układu sekwencyjnego	3
3.2. Metoda syntezy i minimalizacji asynchronicznego układu sekwencyjnego	5
4. Dostępna aparatura	14
4.1. Moduł zadajnika stanów logicznych	14
4.2. Moduł bramek logicznych.....	15
4.3. Moduł testera stanów logicznych	17
4.4. Zasilacz.....	17
5. Przebieg doświadczenia.....	18
5.1. Kolejność czynności	18
5.2. Propozycje zadań realizowanych przez asynchroniczny układ sekwencyjny	19
6. Wskazówki do raportu.....	21
7. Literatura	22

Przed zapoznaniem się z instrukcją i przystąpieniem do wykonywania ćwiczenia należy opanować następujący materiał teoretyczny:

1. Projektowanie i minimalizacja kombinacyjnych układów logicznych w zakresie objętym ćwiczeniami E51 i E52 [1-9].
2. Podział układów przełączających na układy kombinacyjne i sekwencyjne. Podział układów sekwencyjnych na synchroniczne i asynchroniczne [1-8].
3. Struktura automatów asynchronicznych Moore'a i Mealy'ego. Stan wewnętrzny automatu, opis przejść automatu przy użyciu tablicy, grafu i funkcji przejść, kodowanie tablicy przejść, funkcja i tablica wyjść [1-7].
4. Procedura syntezy i minimalizacji automatów asynchronicznych Moore'a i Mealy'ego [1-8].
5. Zjawiska wyścigów i hazardu w asynchronicznych układach sekwencyjnych [1-6].

1. Cel ćwiczenia

Celem ćwiczenia jest zaprojektowanie, wykonanie i przetestowanie asynchronicznego układu sekwencyjnego złożonego z bramek NAND, który realizuje postawione zadanie w możliwie najprostszy sposób. Dodatkowym celem jest zapoznanie się ze zjawiskiem hazardu w układach logicznych oraz z oceną ryzyka dla przejść zachodzących w automacie.

2. Zagrożenia

Rodzaj	Brak	Małe	Średnie	Duże
zagrożenie elektryczne		+		
zagrożenie optyczne	+			
zagrożenie mechaniczne (w tym akustyczne, hałas)	+			
zagrożenie polem elektro-magnetycznym (poza widmem optycznym)	+			
zagrożenie biologiczne	+			
zagrożenie radioaktywne (jonizujące)	+			
zagrożenie chemiczne	+			
zagrożenie termiczne (w tym wybuch i pożar)	+			

Przewody z wtykami bananowymi są przeznaczone wyłącznie do użytku w obwodach niskiego napięcia – nie wolno podłączać ich do gniazda sieci zasilającej 230 V.

3. Wprowadzenie teoretyczne

3.1. Opis asynchronicznego układu sekwencyjnego

Układy cyfrowe mogą posiadać dowolną liczbę wejść. W celu uproszczenia zapisu wprowadzimy wektor stanów wejściowych X obejmujący wszystkie binarne sygnały wejściowe x_1, x_2, \dots, x_n

$$\mathbf{X} = (x_1, x_2, \dots, x_n). \quad (1)$$

Analogicznie wprowadzimy wektor stanów wyjściowych \mathbf{Y} złożony ze wszystkich binarnych sygnałów wyjściowych y_1, y_2, \dots, y_m

$$\mathbf{Y} = (y_1, y_2, \dots, y_m). \quad (2)$$

W układach kombinacyjnych stan wyjścia układu w dowolnej chwili t jest jednoznacznie określony przez bieżący stan jego wejść

$$\mathbf{Y}^t = f(\mathbf{X}^t). \quad (3)$$

W układach sekwencyjnych stan wyjścia zależy nie tylko od bieżącego stanu wejść \mathbf{X}^t układu, lecz także od wcześniejszych stanów wejść

$$\mathbf{Y}^t = f(\mathbf{X}^t, \mathbf{X}^{t-1}, \mathbf{X}^{t-2}, \dots). \quad (4)$$

Do opisu stanu pamięci układu sekwencyjnego wprowadzimy pojęcie wektora stanu wewnętrznego \mathbf{Q} złożonego ze wszystkich binarnych stanów linii sprzężenia zwrotnego q_1, q_2, \dots, q_k

$$\mathbf{Q} = (q_1, q_2, \dots, q_k). \quad (5)$$

Pamięć układu sekwencyjnego realizuje się w postaci układu kombinacyjnego objętego sprzężeniem zwrotnym, przy czym część wejść układu kombinacyjnego jest połączonych bezpośrednio z jego wyjściami, a część stanowi wejścia układu sekwencyjnego (Rys. 1).

Funkcja przejść δ , realizowana przez układ kombinacyjny, określa kolejny stan wewnętrzny \mathbf{Q}^{t+1} na podstawie poprzedzającego stanu wewnętrznego \mathbf{Q}^t i wejściowego \mathbf{X}^t

$$\mathbf{Q}^{t+1} = \delta(\mathbf{Q}^t, \mathbf{X}^t). \quad (6)$$

Nowy stan wewnętrzny automatu jest **stanem stabilnym** gdy $\mathbf{Q}^{t+1} = \mathbf{Q}^t$. W wyniku zmiany na wejściach \mathbf{X} może się także pojawić jeden lub więcej **stanów niestabilnych**, tzn. takich dla których $\mathbf{Q}^{t+1} \neq \mathbf{Q}^t$. Jeżeli zmiany sygnałów wejściowych zachodzą dostatecznie wolno, to po stanach niestabilnych powinien zawsze następować stan stabilny.

Drugi układ kombinacyjny realizuje **funkcję wyjść** λ , której wynik określa stan wyjść automatu. W automacie Moore'a (rys. 1.a) argumentem funkcji wyjść jest tylko stan wewnętrzny

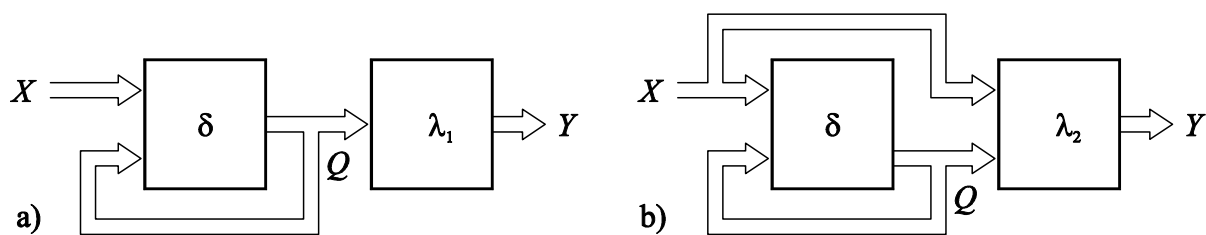
$$\mathbf{Y}^t = \lambda_1(\mathbf{Q}^t), \quad (7)$$

natomiast w automacie Mealy'ego (rys. 1.b) jest to zarówno stan wyjściowy jak i stan wewnętrzny

$$\mathbf{Y}^t = \lambda_2(\mathbf{X}^t, \mathbf{Q}^t). \quad (8)$$

W przypadku szczególnie prostych automatów stan wyjściowy \mathbf{Y} może być tożsamy ze stanem wewnętrznym \mathbf{Q} .

Układy asynchroniczne nie posiadają zegara, z którym można by powiązać kolejne takty pracy 1, 2, ..., $t-2$, $t-1$, t . Zmiany stanu wewnętrznego i wyjściowego następują wyłącznie pod wpływem zmian stanów wejściowych. W tym ćwiczeniu rozważymy tylko sytuację, w której ustalone stany wejściowe trwają dostatecznie długo, by układ zdążył ustalić swój stan wewnętrzny przed następną zmianą na wejściach. Kolejne takty pracy 1, 2, ..., $t-2$, $t-1$, t możemy zatem powiązać z kolejnymi przełączeniami stanów wejść.



Rys. 1. Asynchroniczne automaty cyfrowe: a) automat Moore'a; b) automat Mealy'ego.

3.2. Metoda syntezy i minimalizacji asynchronicznego układu sekwencyjnego

Projektowanie asynchronicznego układu sekwencyjnego z bramek logicznych przebiega zazwyczaj według procedury złożonej z następujących etapów:

1. Analiza treści zadania i dodatkowe **założenia projektowe**.
2. Sporządzenie pierwotnej **listy stanów wewnętrznych** automatu.
3. Przedstawienie przejść między stanami automatu w formie **grafu przejść**.
4. Zapisanie **pierwotnej tablicy przejść i wyjść** dla stanów automatu Moore'a.
5. Wyszukanie w pierwotnej tablicy przejść **stanów równoważnych** i połączenie ich w pojedyncze stany.
6. Wyszukanie i połączenie **stanów zgodnych**. Decyzja o połączeniu takich stanów oznacza przejście od automatu o strukturze Moore'a do automatu Mealy'ego.
7. Jeżeli liczba stanów wewnętrznych została zredukowana, sporządzenie **zredukowanej tablicy przejść** oraz grafu dla stanów zredukowanych.
8. **Kodowanie stanów wewnętrznych** poprzez binarne stany wewnętrzne q_1, q_2, q_3, \dots .
9. Zapisanie **zakodowanych tablic przejść** dla każdego z binarnych stanów wewnętrznych q_1, q_2, q_3, \dots .
10. **Synteza funkcji przejść** δ kolejno dla poszczególnych binarnych stanów wewnętrznych q_1, q_2, q_3, \dots . Na tym etapie należy rozważyć i wyeliminować **ryzyko** wynikające z możliwości wystąpienia **hazardu** w układzie kombinacyjnym realizującym funkcję przejść.
11. **Synteza funkcji wyjść** λ . W zależności od przeznaczenia automatu trzeba podjąć decyzję o dopuszczeniu albo eliminacji **hazardu** w układzie wyjściowym.
12. Sporządzenie **schematu** układu zrealizowanego przy użyciu dostępnych elementów logicznych.

W przypadku niektórych projektów wybrane etapy mogą zostać pominięte, np. funkcja wyjść może być funkcją tożsamościową i w takim przypadku stany wewnętrzne q_i będą jednocześnie stanami wyjściowymi y_i .

Rozważmy kolejne etapy projektowania automatu na przykładzie układu, którego zadaniem jest sygnalizowanie stanem 1 na pojedynczym binarnym wyjściu Y wzrostu temperatury mierzonej przez termometr kontaktowy względem poprzedniego stanu termometru. Termometr wyposażony jest w trzy kontakty A, B i C , które podczas wzrostu temperatury przechodzą ze stanu 0 na 1 w wymienionej kolejności, tzn. możliwe kombinacje sygnałów wyliczone w kolejności od najniższej do najwyższej temperatury to: $ABC = 000, 100, 110$ oraz 111 . Kasowanie pamięci wzrostu temperatury następuje przez podanie stanu 1 na wejście R (reset) automatu, a także gdy zostanie zarejestrowany spadek temperatury.

Zauważamy, że słowny opis postawionego problemu nie daje podstaw do podjęcia całkowicie jednoznacznych decyzji i musimy przyjąć **dotatkowe założenia projektowe**. W rozważanym projekcie założymy:

1. Nie rozważamy zachowania układu w przypadku awarii termometru kontaktowego.
2. Układ bezpośrednio po włączeniu zasilania może wejść w dowolny stan stabilny i stan ten może nie mieć sensu dla aktualnej kombinacji sygnałów z termometru kontaktowego. Założymy, że układ można wyprowadzić z każdego błędnego stanu początkowego przez podanie 1 na wejście resetujące R .
3. Zmiany sygnałów A , B i C z termometru kontaktowego zachodzą wolno w porównaniu do czasu dochodzenia automatu do stanu stabilnego. W konsekwencji nie są możliwe przejścia postrzegane przez automat jako jednoczesne na wejściach automatu A , B lub C . Analogicznie założymy, że nie jest możliwy impuls na wejściu R trwający krócej niż czas dojścia automatu do stanu stabilnego.
4. Przełączenia stanów na wejściu R nie są w żaden sposób skorelowane w czasie ze zmianami na wejściach ABC . Przejścia jednoczesne na tych liniach nie mogą doprowadzić do wprowadzenia automatu w błędny stan stabilny.

Następnym etapem jest sporządzenie **listy stanów wewnętrznych** projektowanego automatu. Początkowo rozważymy następujące stany:

1. Żaden kontakt w termometrze nie jest załączony, tzn. bieżąca temperatura ma najniższą rozpoznawalną wartość $t < t_A$, gdzie symbol t_i oznacza temperaturę załączenia i -tego kontaktu w termometrze. W takiej sytuacji zarejestrowanie wzrostu temperatury nie jest możliwe ($Y = 0$).
2. Bieżąca temperatura $t_A \leq t < t_B$ i zarejestrowano wzrost temperatury ($Y = 1$).
3. Bieżąca temperatura $t_A \leq t < t_B$ i nie zarejestrowano wzrostu temperatury. Poprzednia temperatura była wyższa, nastąpił reset albo wejście $R = 1$ i reset wciąż trwa, przy czym nie widać potrzeby opisywania tych trzech sytuacji jako oddzielnych stanów wewnętrznych ($Y = 0$).
4. Bieżąca temperatura $t_B \leq t < t_C$ i zarejestrowano wzrost temperatury ($Y = 1$).
5. Bieżąca temperatura $t_B \leq t < t_C$ i nie zarejestrowano wzrostu temperatury. Poprzednia temperatura była wyższa, nastąpił reset albo wejście $R = 1$ i reset wciąż trwa ($Y = 0$).
6. Bieżąca temperatura $t_C \leq t$ i zarejestrowano wzrost temperatury ($Y = 1$).
7. Bieżąca temperatura $t_C \leq t$ i nie zarejestrowano wzrostu temperatury. Ponieważ nie ma możliwości zarejestrowania spadku od jeszcze wyższej temperatury, musiał nastąpić reset albo reset wciąż trwa ($Y = 0$).

W rozważanym zadaniu wykonanie **grafu przejść** dla siedmiu stanów pierwotnych przy znacznej liczbie dopuszczonych przejść będzie zadaniem pracochłonnym a osiągnięty efekt może być mało czytelny. Odłożmy sporządzenie grafu do czasu gdy zostaną określone stany wewnętrzne po redukcji.

Kolejnym krokiem jest przygotowanie **pierwotnej tablicy przejść** między stanami wewnętrznymi układu zapisanej razem z tablicą wyjść (Tabela 1). Zazwyczaj tablicę przejść wykonuje się w formie dwuwymiarowej tablicy, w której kolejne wiersze odpowiadają kolejnym wewnętrznym stanom ustalonym, natomiast kolejne kolumny odpowiadają poszczególnym kombinacjom stanów wejściowych uporządkowanych zgodnie z **kodem Graya**. Przejście pomiędzy dwoma stanami stabilnymi automatu odpowiada następującej sekwencji przejść w tabeli: zmiana stanu wektora wejściowego $RABC$ wyzwala przejście poziome wzdłuż wiersza tabeli; następnie realizowane jest przejście wzdłuż kolumny do wiersza tabeli wskazanego w komórce rozpoczynającej to przejście.

stan	RABC															wyjście Y	
	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001		1000
1	①	-	-	-	-	-	-	2	3	-	7	5	-	-	-	①	0
2	1	-	-	-	4	-	-	②	3	-	7	5	-	-	-	1	1
3	1	-	-	-	4	-	-	③	③	-	7	5	-	-	-	1	0
4	-	-	-	-	④	6	-	3	3	-	7	5	-	-	-	1	1
5	-	-	-	-	⑤	6	-	3	3	-	7	⑤	-	-	-	1	0
6	-	-	-	-	5	⑥	-	-	3	-	7	5	-	-	-	1	1
7	-	-	-	-	5	⑦	-	-	3	-	⑦	5	-	-	-	1	0

Tabela 1. Pierwotna tablica przejść i wyjść automatu Moore'a. Kreskami (-) oznaczono stany nieokreślone. Kółka wokół numerów stanów oznaczają stany stabilne. Numery stanów niestabilnych wskazują docelowy numer linii dla przejścia wzdłuż kolumny.

Zauważmy, że dodatkowe założenia projektowe mają istotny wpływ na postać przyjętej Tabeli 1. Omówimy ten wpływ kolejno dla każdego założenia.

1. Wszystkie kolumny w Tabeli 1 odpowiadające przypadkom niekolejnego załączenia kontaktów w termometrze wypełniono stanami nieokreślonymi (-).
2. Założenie 2 uwzględniono w Tabeli 1 przez wypełnienie wszystkich komórek w kolumnach $RABC = 1100, 1111, 1110$ i 1000 stanami przejściowymi prowadzącymi do stanów stabilnych po przejściach wzdłuż kolumn.
3. Założenie 3 pozwala m.in. na zapisanie (-) w komórce należącej do linii 1 i kolumny 0110 w Tabeli 1. Gdyby możliwe było przejście np. od $RABC = 0000$ bezpośrednio do 0110, to w tej komórce należałoby zapisać stan niestabilny 4 prowadzący dalej do stanu stabilnego 4 poprzez przejście wzdłuż kolumny $RABC = 0110$. Znalezienie pozostałych sytuacji, w których możliwe było zapisanie (-) na podstawie założenia 3 pozostawimy czytelnikom.
4. Dopuszczenie albo odrzucenie możliwości jednoczesnego przejścia na linii R i jednej z linii ABC nie wpłynie na postać Tabeli 1 jeżeli wcześniej zostały uwzględnione założenia 1, 2 i 3. Gdyby założenie 4 zostało rozpatrzone jako pierwsze, to prowadziłoby do konieczności zamiany (-) na określone stany niestabilne np. w niektórych komórkach kolumn $RABC = 1100, 1111, 1110$ i 1000 .

Dwa stany są **stanami równoważnymi** jeżeli w odpowiadających im wierszach pierwotnej tablicy przejść i wyjść:

- stany stabilne znajdują się w jednakowych kolumnach,
- stany wyjść w obu wierszach są jednakowe,
- przy dowolnej zmianie stanu wejść układ przechodzi z rozpatrywanych stanów do stanów jednakowych lub równoważnych,
- w miejsce (-) możemy wpisać cokolwiek jeżeli umożliwi to uznanie stanów za równoważne.

Analiza numerów stanów wewnętrznych zapisanych wewnątrz Tabeli 1 może sugerować istnienie stanów równoważnych, np. 1 i 2, jednakże niezgodność poziomów logicznych wyjścia $Y = 0$ oraz $Y = 1$ przesądza o nierównoważności stanów 1 i 2. Ponadto wiersze 1 i 2 nie są równoważne pod względem kolumn zawierających stany stabilne. Analiza całej Tabeli 1 prowadzi do wniosku, że nie istnieją żadne stany równoważne.

Dwa stany są **stanami zgodnymi w strukturze Mealy’ego** jeżeli w odpowiadających im wierszach pierwotnej tablicy przejść:

- stany z jednakowej kolumny mają takie same numery,
- w miejsce (–) możemy wpisać cokolwiek jeżeli umożliwi to uznanie stanów za zgodne,
- zgodność wyjść nie jest wymagana.

W Tabeli 1 możemy dostrzec liczne pary stanów zgodnych. W szczególności zgodne są stany 1 i 2 oraz 1 i 7, jednakże nie można połączyć tych trzech stanów, bo stany 2 i 7 nie są zgodne w przypadku gdy $RABC = 0110$. Niezależnie od wybranego wariantu łączenia stanów uzyskamy zawsze możliwość trzech połączeń obejmujących po dwa stany pierwotne. Liczba stanów po redukcji wyniesie $7 - 3 = 4$ i do ich reprezentowania potrzebujemy dwóch zmiennych binarnych q_1 i q_2 .

Decydując się na połączenie stanów zgodnych odpowiadających różnym poziomom logicznym wyjścia trzeba pamiętać, że wiąże się to ze zmianą struktury automatu Moore’a w automat Mealy’ego (rys. 1). Zazwyczaj taka zmiana jest korzystna, gdy jednocześnie można zredukować liczbę binarnych linii sprzężenia zwrotnego q_i , jednakże w szczególnych przypadkach funkcja wyjść może okazać się bardziej złożona dla automatu Mealy’ego. Jeżeli liczba dopuszczonych stanów nie będzie równa dokładnie 2^k , gdzie k jest liczbą naturalną, to należy także rozważyć wyprowadzanie automatu z niepożądanych nadmiarowych stanów stabilnych, które układ może przyjąć samorzutnie po włączeniu. W rozważanym zadaniu mamy przesłanki do decyzji o wyborze automatu Mealy’ego. W dalszych etapach projektu będziemy posługiwać się stanami wewnętrznymi po redukcji, które zostały zdefiniowane w Tabeli 2.

stany pierwotne	stany po redukcji
1, 2	1'
3, 4	2'
5, 6	3'
7	4'

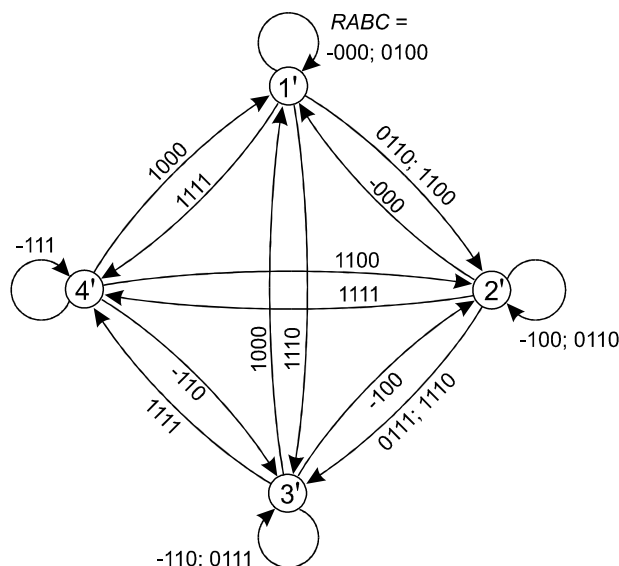
Tabela 2. Redukcja stanów automatu przez łączenie stanów zgodnych.

Wykorzystując dane z Tabel 1 i 2 zapisujemy **zredukowaną tablicę przejść** automatu, którą przedstawiono w Tabeli 3. Jeżeli w danej kolumnie w wierszach odpowiadających łączonym stanom zapisano stan stabilny oraz niestabilny, to w wierszu zredukowanej tabeli będzie stan stabilny.

		RABC															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
stan	1'	Ⓚ	–	–	–	2'	–	–	Ⓚ	2'	–	4'	3'	–	–	–	Ⓚ
	2'	1'	–	–	–	Ⓚ	3'	–	Ⓚ	Ⓚ	–	4'	3'	–	–	–	1'
	3'	–	–	–	–	Ⓚ	Ⓚ	–	2'	2'	–	4'	Ⓚ	–	–	–	1'
	4'	–	–	–	–	3'	Ⓚ	–	–	2'	–	Ⓚ	3'	–	–	–	1'

Tabela 3. Tablica przejść automatu Mealy’ego po redukcji stanów. Kreskami (–) oznaczono stany nieokreślone. Kółka wokół numerów stanów oznaczają stany stabilne.

Redukcja stanów umożliwiła sporządzenie **grafu przejść** o czytelnej formie przedstawionej na rys. 2. Analogiczny graf dla stanów pierwotnych byłby o wiele bardziej skomplikowany.



Rys. 2. Graf przejść automatu Mealy’ego. Wartości podane przy strzałkach oznaczają kombinacje stanów wejść RABC, które wywołują przejście. Kreską (–) oznaczono przypadki w których przejścia zachodzą niezależnie od stanu wejścia R.

Następnym krokiem jest **kodowanie stanów automatu**, polegające na przyporządkowaniu numerom stanów zredukowanych reprezentacji na binarnych liniach sprzężenia zwrotnego q_1, q_2, \dots, q_k . Wybór sposobu kodowania ma krytyczne znaczenie dla podatności układu na zjawiska wyścigów i hazardu występujące w układzie kombinacyjnym realizującym funkcję przejść. Przykładowo przejście na liniach q_2q_1 od stanu 00 do 11 w rzeczywistych układach może przebiegać bezpośrednio albo poprzez stany pośrednie 01 lub 10 w zależności od struktury układu i różnic w czasach propagacji sygnału przez poszczególne elementy logiczne. Pojawienie się stanu pośredniego nie może prowadzić do skierowania układu w niewłaściwy stan stabilny. Kodowanie zaproponowane w Tabeli 4 jest szczególnie korzystne, gdyż ogranicza do koniecznego minimum przejścia związane z przełączeniami więcej niż jednej binarnej linii q_i .

stany przed zakodowaniem	stany po zakodowaniu	
	q_2	q_1
1'	0	0
2'	0	1
3'	1	1
4'	1	0

Tabela 4. Kodowanie stanów automatu.

Jeżeli w Tabeli 3 zastąpimy numery stanów ich binarnymi kodami z Tabeli 4, to otrzymamy **zakodowaną tablicę przejść** (nazywaną także tablicą wzbudzeń). Tablicę przejść rozetniemy na dwie części przedstawione w Tabelach 5.a i 5.b. Podział tabeli wynika z niezależnie przeprowadzonej **syntezy funkcji przejść** $q_1^{t+1} = \delta_1(\mathbf{Q}^t, \mathbf{X}^t)$ oraz $q_2^{t+1} = \delta_2(\mathbf{Q}^t, \mathbf{X}^t)$ dla poszczególnych linii sprzężenia zwrotnego.

(a)

		RABC															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
q_2q_1	00	0	-	-	-	1	-	-	0	1	-	0	1	-	-	-	0
	01	0	-	-	-	1	1	-	1	1	-	0	1	-	-	-	0
	11	-	-	-	-	1	1	-	1	1	-	0	1	-	-	-	0
	10	-	-	-	-	1	0	-	-	1	-	0	1	-	-	-	0

(b)

		RABC															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
q_2q_1	00	0	-	-	-	0	-	-	0	0	-	1	1	-	-	-	0
	01	0	-	-	-	0	1	-	0	0	-	1	1	-	-	-	0
	11	-	-	-	-	1	1	-	0	0	-	1	1	-	-	-	0
	10	-	-	-	-	1	1	-	-	0	-	1	1	-	-	-	0

Tabela 5. Zakodowane tablice przejść automatu Mealy’ego zapisane w formie tablic Karnaugh dla poszczególnych binarnych stanów wewnętrznych: (a) stan wewnętrzny q_1 , (b) stan wewnętrzny q_2 . Przedstawiono syntezę funkcji przejść przez łącznie jedynek. Komórki tabel zawierają kolejne stany wewnętrzne automatu zależne od stanów poprzedzających związanych z wierszami tabel oraz stanów wejściowych związanych z kolumnami.

Pętla oznaczona kolorem czerwonym w Tabeli 5.a byłaby nadmiarowa w przypadku układu kombinacyjnego, w którym występowanie zjawiska wyścigów i hazardu nie ma znaczenia. W rozważanym asynchronicznym układzie sekwencyjnym uwzględnienie tej pętli ma jednak zasadnicze znaczenie dla stabilności pracy automatu. Rozważmy najpierw syntezę funkcji, która jest równoważnikiem trzech pętli z Tabeli 5.a bez uwzględnienia pętli czerwonej (wykorzystanie tablicy Karnaugh do syntezy funkcji logicznej zostało omówione w instrukcjach do ćwiczeń E51 i E52)

$$q_1^{t+1} = \overline{BC} + q_1 \overline{RA} + RAB. \tag{9a}$$

W szczególnym przypadku dla $A = 1, B = 0, C = 0, q_1 = 1$ otrzymujemy

$$q_1^{t+1} = \overline{R} \cdot 1 + R \cdot 1 = \overline{\overline{\overline{R} \cdot 1}}. \tag{10a}$$

Przy przejściu $1 \rightarrow 0$ na wejściu R wartość wyrażenia prostego R ulegnie zmianie wcześniej niż wyrażenia zanegowanego \overline{R} i przez chwilę będą współistniały wartości $R = 0$ oraz $\overline{R} = 0$. Taka sytuacja doprowadzi do **hazardu statycznego** na linii q_1^{t+1} , na której dojdzie do przejść stanów $1 \rightarrow 0 \rightarrow 1$. Zamiana sumy logicznej na iloczyn przy wykorzystaniu praw De Morgana nie wpłynie na warunek wystąpienia hazardu.

W rozważanym projekcie istnieje ryzyko, że automat trwający np. w stanie stabilnym 2' przy $RABC = 1100$ nie pozostanie w tym stanie podczas przejścia $1 \rightarrow 0$ na wejściu R , co byłoby niezgodne z grafem przejść na rys. 2. Niezamierzone wyzerowanie linii q_1 może wywołać przejście do stanu 1' i pozostanie w nim, gdyż według Tabeli 3 jest to stan stabilny dla $RABC = 0100$. Eliminację hazardu osiągniemy przez otoczenie stanów zagrożonych przejściem hazardowym dodatkową wspólną pętlą oznaczoną na czerwono w Tabeli 5.a.

Dodanie kolejnej pętli obejmującej jedynki odpowiada wprowadzeniu nowego składnika do wzoru (9a)

$$q_1^{t+1} = \overline{BC} + q_1 \overline{RA} + R \overline{AB} + q_1 \overline{AB}. \quad (9b)$$

W rozważonym wcześniej szczególnym przypadku $A = 1$, $B = 0$, $C = 0$, $q_1 = 1$ wzór (9b) uprości się do postaci

$$q_1^{t+1} = \overline{R} \cdot 1 + R \cdot 1 + 1 = \overline{\overline{\overline{\overline{\overline{R}}}} \cdot 1} \cdot \overline{\overline{\overline{\overline{\overline{R}}}} \cdot 1} \cdot 0 = 1. \quad (10b)$$

Nietrudno zauważyć, że zmiany na wejściu R tym razem nie mogą doprowadzić do hazardu.

Funkcję daną wzorem (9b) możemy przekształcić do wygodniejszej postaci

$$q_1^{t+1} = \overline{BC} + q_1 \overline{RA} + (R + q_1) \overline{AB} = \overline{BC} + q_1 \overline{RA} + \overline{\overline{\overline{\overline{\overline{R}}}} \cdot \overline{\overline{\overline{\overline{\overline{q_1}}}}} \cdot \overline{\overline{\overline{\overline{\overline{AB}}}}} = \overline{\overline{\overline{\overline{\overline{\overline{BC}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{q_1}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{RA}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{R}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{q_1}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{AB}}}}}}}. \quad (11)$$

Realizacja funkcji w postaci danej wzorem (11) wymaga jedynie dwu- i trzywejściowych bramek NAND dostępnych na stanowisku laboratoryjnym w pracowni. Przekształcenia wykonywane przy użyciu tożsamości algebry Boole'a zazwyczaj nie wnoszą ryzyka destabilizacji automatu dodatkowym hazardem, jednakże istnieją wyjątkowe sytuacje.

Wyrażenia typu $a \cdot \overline{a}$ oraz $a + \overline{a}$ możemy bezpiecznie zastąpić przez 0 (dla iloczynu) oraz 1 (dla sumy), natomiast odwrotne przekształcenie jest hazardogenne.

Zauważmy, że w rozważanym automacie możliwe są jeszcze inne przejścia zagrożone hazardem statycznym na linii q_1 , jednakże likwidowanie pozostałych przypadków doprowadziłoby jedynie do zbędnej komplikacji układu. Przykładowo hazardem zagrożone jest przejście ze stanu stabilnego 2' do 3' następujące po zmianie na wejściach $RABC = 1100 \rightarrow 1110$, jednakże chwilowe przełączenie do $q_1 = 0$ doprowadzi wg. Tabel 3 i 4 jedynie do zmiany trasy przejścia, natomiast dalsze podążanie układu do stanu stabilnego 3' wzdłuż kolumny $RABC = 1110$ nie jest zagrożone.

Bezpośrednio na podstawie pętli oznaczonych na tablicy 5.b otrzymujemy

$$q_2^{t+1} = C + q_2 B + RB = \overline{\overline{\overline{\overline{\overline{C}}}}} \cdot \overline{\overline{\overline{\overline{\overline{q_2}}}}} \cdot \overline{\overline{\overline{\overline{\overline{RB}}}}}. \quad (12)$$

Wzór (12) nie wnosi żadnego ryzyka przejść hazardowych, bo wszystkie dozwolone przejścia między komórkami zawierającymi „1” w Tabeli 5.b odbywają się zawsze w obrębie którejś pętli. Zauważmy, że przejścia wzdłuż wiersza $q_2 q_1 = 01$ bezpośrednio pomiędzy komórkami leżącymi w kolumnach 0111 oraz 1110 mogłyby być zagrożone hazardem, ale nie są one dozwolone w ramach przyjętych założeń projektowych. Przejścia jednoczesne na wejściu R oraz jednym z wejść ABC są wprowadzone dozwolone, jednakże w Tabeli 3 widać, że obie wymienione komórki są związane ze stanami niestabilnymi. Opisane przejście musiałoby zatem nastąpić bardzo szybko po poprzednim przejściu zanim automat osiągnąłby stan stabilny, co jest wykluczone przez założenie projektowe nr 3.

RABC

(a)

q_2q_1	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
00	0	-	-	-	1	-	-	0	1	-	0	1	-	-	-	0
01	0	-	-	-	1	1	-	1	1	-	0	1	-	-	-	0
11	-	-	-	-	1	1	-	1	1	-	0	1	-	-	-	0
10	-	-	-	-	1	0	-	-	1	-	0	1	-	-	-	0

q_1^{t+1}

RABC

(b)

q_2q_1	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
00	0	-	-	-	0	-	-	0	0	-	1	1	-	-	-	0
01	0	-	-	-	0	1	-	0	0	-	1	1	-	-	-	0
11	-	-	-	-	1	1	-	0	0	-	1	1	-	-	-	0
10	-	-	-	-	1	1	-	-	0	-	1	1	-	-	-	0

q_2^{t+1}

Tabela 6. Zakodowane tablice przejść automatu Mealy’ego zapisane w formie tablic Karnaugh dla poszczególnych binarnych stanów wewnętrznych: (a) stan wewnętrzny q_1 , (b) stan wewnętrzny q_2 . Przedstawiono syntezę funkcji przejść przez łącznie zer.

Alternatywnie syntezę i minimalizację funkcji przejść δ_1 i δ_2 możemy wykonać przez łączenie zer na tablicy przejść. Na podstawie pętli w Tabeli 6.a otrzymujemy

$$q_1^{t+1} = A(q_1 + R + B)(q_1 + \bar{C})(\bar{R} + \bar{C}). \quad (13)$$

Hazard występuje jedynie w przypadkach jednoczesnych zmian na wejściu R i którejś z linii A , B albo C . Likwidacja tego hazardu nie jest konieczna, gdyż jego obecność może jedynie wpłynąć na trasy przejść, nie zaś na docelowe stany stabilne. Przekształćmy funkcję (13) do wygodniejszej postaci wykorzystując prawo rozdzielności dodawania względem mnożenia (które nie ma odpowiednika dla działań w dziedzinie liczb rzeczywistych) do połączenia wyrazów zawierających \bar{C} . Następnie wykorzystamy prawa De Morgana w celu dostosowania wyrażenia do bezpośredniej realizacji przy użyciu samych bramek NAND

$$q_1^{t+1} = A(q_1 + R + B)(\bar{C} + q_1 \bar{R}) = A \cdot \overline{\overline{q_1} \bar{R} B \cdot C} \cdot \overline{\overline{q_1} \bar{R} B \cdot C} \cdot q_1 \bar{R}. \quad (14)$$

Postać funkcji danej wzorem (14) jest nieco prostsza niż we wzorze (11), dlatego przyjmiemy ją do realizacji. Analogicznie na podstawie Tabeli 6.b otrzymujemy

$$q_2^{t+1} = A \cdot B \cdot (q_2 + R + \bar{A} + C) = A \cdot B \cdot \left(\overline{\overline{q_2} \bar{R} \bar{C} + \bar{A}} \right) = A \cdot B \cdot \left(\overline{\overline{\overline{\overline{q_2} \bar{R} \bar{C} + \bar{A}}}} \right). \quad (15)$$

Do realizacji praktycznej wybierzemy jednak prostszą postać funkcji daną wzorem (12).

		RABC															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
q_2q_1																	
00	0	-	-	-	-	1	-	-	1	0	-	-	-	-	-	-	0
01	0	-	-	-	-	1	1	-	0	0	-	-	0	-	-	-	0
11	-	-	-	-	-	0	1	-	0	0	-	0	0	-	-	-	0
10	-	-	-	-	-	0	0	-	-	0	-	0	0	-	-	-	-

Tabela 7. Tablica wyjścia automatu Mealy’ego. Syntezę funkcji wyjść wykonano przez łączenie jedynek. Stany stabilne oznaczono kółkami.

Kolejnym etapem jest synteza **funkcji wyjść** na podstawie **tablicy wyjść**. Wypełnianie tej tablicy rozpoczynamy od zapisania komórek odpowiadających stabilnym stanom wewnętrznym w Tabeli 3. Ponieważ stany 1', 2', 3' i 4' nie są powiązane jednoznacznie ze stanem wyjścia Y, konieczne jest rozważenie które stany 1, 2, ..., 7 z Tabeli 1 zostały zastąpione w komórkach Tabeli 3 przez połączone stany zgodne. Sygnał wyjściowy Y dla stanów niestabilnych określamy następująco:

- Jeżeli sygnały Y odpowiadające początkowemu i końcowemu stanowi stabilnemu są jednakowe, to taki sam sygnał przypisujemy pośredniemu stanowi niestabilnemu.
- Jeżeli sygnały Y odpowiadające początkowemu i końcowemu stanowi stabilnemu są różne, to pośredniemu stanowi niestabilnemu przypisujemy sygnał dowolny (-).
- Niektóre komórki mogą uczestniczyć w przejściach od różnych stabilnych stanów początkowych. Jeżeli różne przejścia prowadzą do odmiennych wniosków, to wybieramy stan zgodny z docelowym stanem stabilnym (zamiast „-”).

W przypadku gdy układ podłączony do wyjścia jest niewrażliwy na krótkie nadmiarowe impulsy (np. żarówka), wszystkie komórki Tabeli 7 odpowiadające stanom niestabilnym można wypełnić „-”.

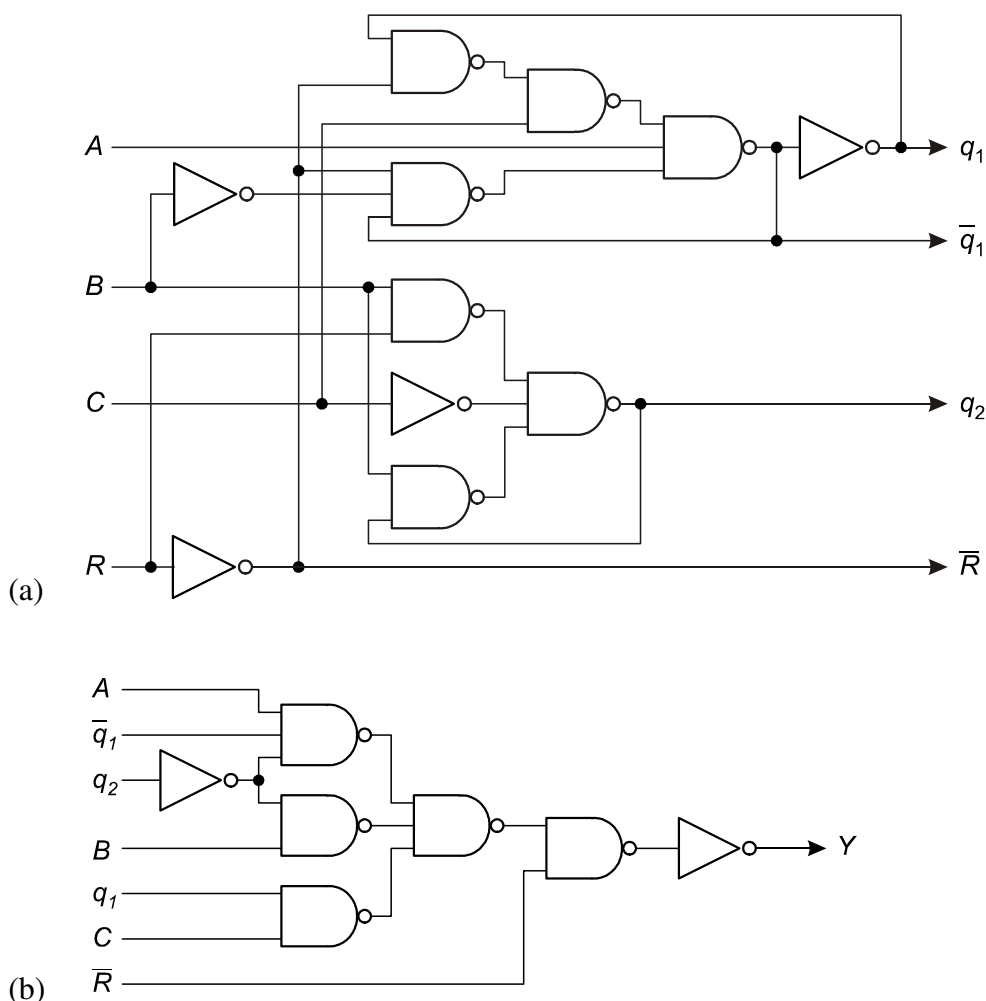
Można wykazać, że pokrycie pętlami wszystkich zer w Tabeli 7 przy uwzględnieniu eliminacji hazardu statycznego wymaga większej liczby pętli niż w przypadku łączenia jedynek. Bezpośrednio z pętli zaznaczonych w Tabeli 7 otrzymujemy funkcję wyjść

$$Y = \overline{q_2} \overline{R} B + q_1 \overline{R} C + \overline{q_2} \overline{q_1} \overline{R} A, \tag{16}$$

w której nie ma zagrożenia hazardem statycznym, bo wszystkie dozwolone przejścia między jedynekami odbywają się zawsze w ramach którejś pojedynczej pętli. Czteroargumentowy iloczyn logiczny nie jest możliwy do bezpośredniej realizacji przy użyciu bramek dostępnych w pracowni. Przekształćmy funkcję opisaną wzorem (16) do wygodniejszej postaci możliwej do bezpośredniej realizacji przy użyciu dostępnych bramek NAND

$$Y = \overline{R} \left(\overline{q_2} B + q_1 C + \overline{q_2} \overline{q_1} A \right) = \overline{R} \left(\overline{\overline{\overline{q_2} B} \overline{q_1} C \overline{q_2} \overline{q_1} A} \right). \tag{17}$$

We wzorach (12), (14) i (17) nie widać wspólnych iloczynów, natomiast negacje $\overline{q_1}$ i \overline{R} powtarzają się w funkcjach przejść i funkcji wyjścia i nie ma potrzeby ich powielania. Schemat układu zrealizowanego na bramkach NAND i NOT, który odpowiada wymienionym wzorom, przedstawiono na rys. 3.

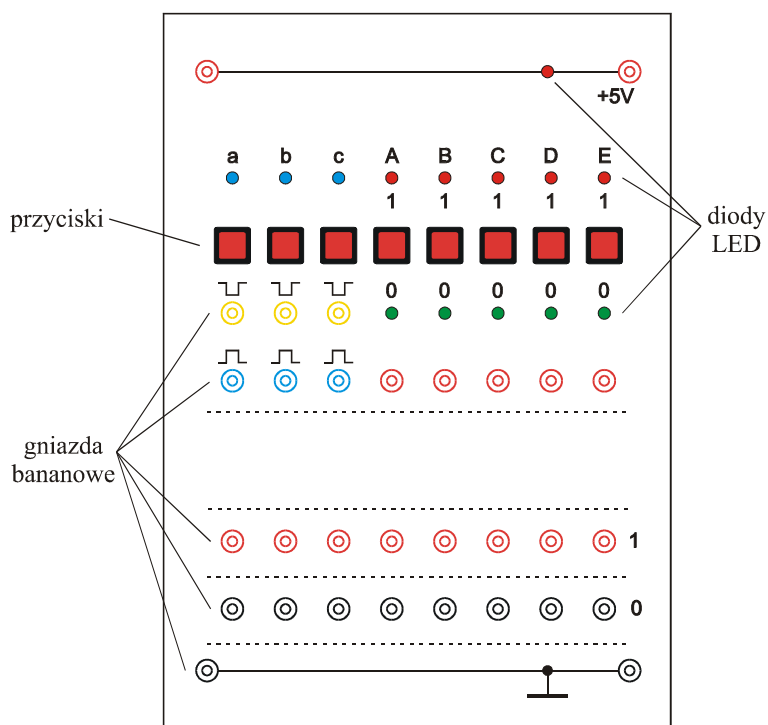


Rys. 3. Schemat układu automatu Mealy'ego. (a) układ realizujący funkcję przejść, (b) układ realizujący funkcję wyjść.

4. Dostępna aparatura

4.1. Moduł zadajnika stanów logicznych

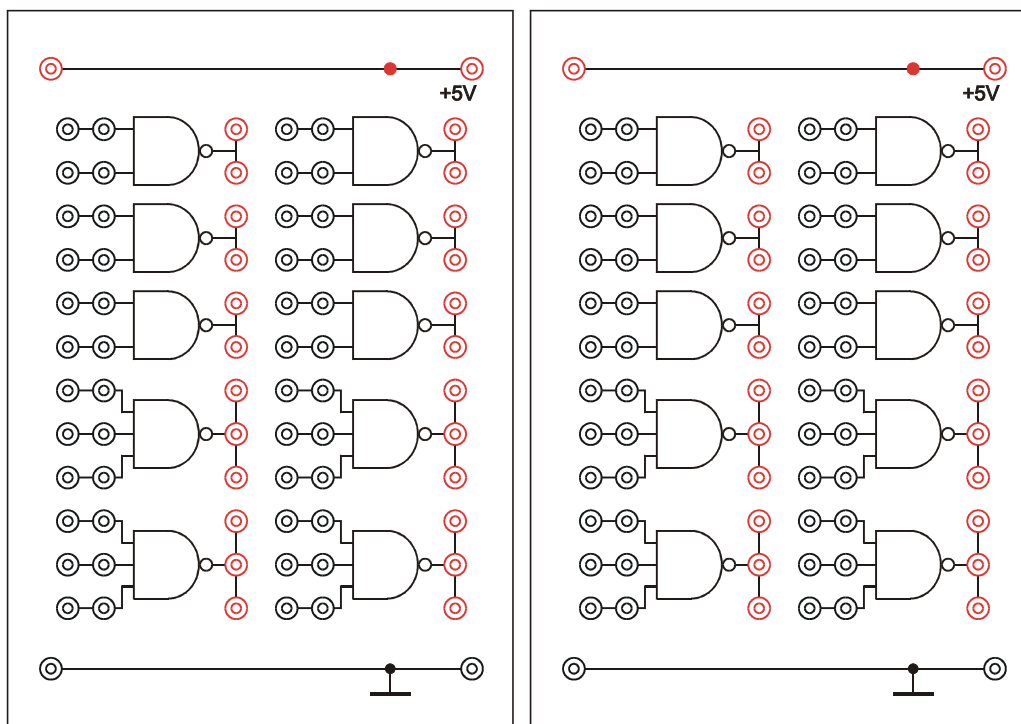
Moduł zadajnika stanów logicznych składa się z pięciu przełączników umożliwiających wybór stanu logicznego 0 albo 1 w czerwonych gniazdach umieszczonych pod przełącznikami (rys. 4). Ponadto moduł zawiera trzy generatory pojedynczego impulsu, przy czym na osobnych gniazdach dostępny jest zarówno impuls stanu 1 (gniazda niebieskie) jak i impuls stanu 0 (gniazda żółte). W przypadku gdy konieczne jest niezależne zadawanie stanów na więcej niż pięciu wejściach układu kombinacyjnego, najrzadziej przełączane wejścia należy podłączyć do gniazd o ustalonym stanie 0 albo 1 umieszczonych w dolnej części urządzenia. Wykorzystanie gniazd o stanie ustalonym wprowadza jednak ryzyko szybkich wielokrotnych przełączeń stanów logicznych podczas wsuwania wtyczek do gniazd.



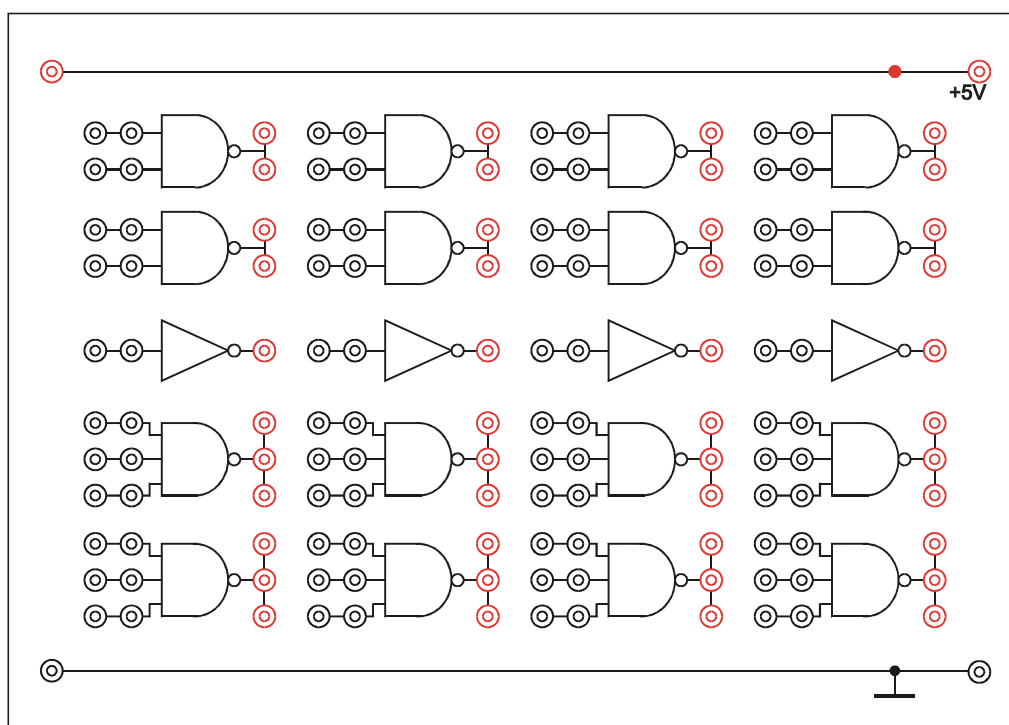
Rys. 4. Panel czołowy modułu zadajnika stanów logicznych.

4.2. Moduł bramek logicznych

Moduły bramek logicznych zawierają 12 dwuwejściowych bramek NAND oraz 8 trzywejściowych bramek NAND (rys. 5a), zaś potrzebne bramki NOT realizuje się przy użyciu zworek zwierających wejścia bramek NAND. Alternatywnie na stanowisku może znajdować się moduł złożony z 4 bramek NOT, 8 dwuwejściowych bramek NAND oraz 8 trzywejściowych bramek NAND (rys. 5b). Wszystkie pozostałe funkcje logiczne należy realizować przez połączenia dostępnych bramek NAND i NOT.



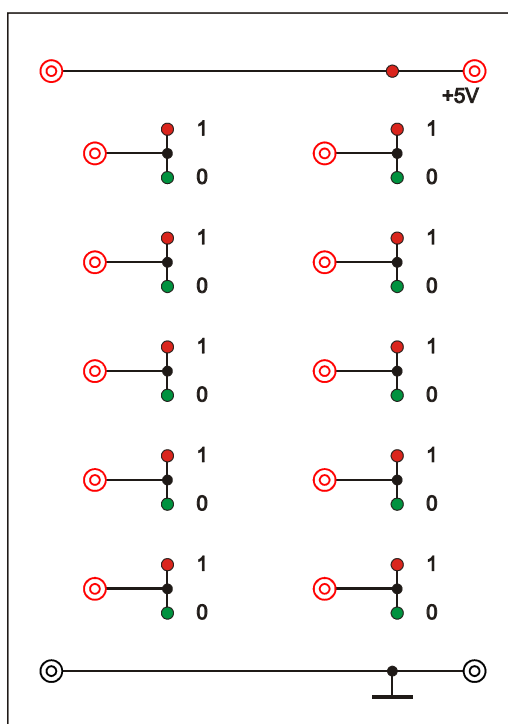
Rys. 5a. Panele czołowe zestawu dwóch modułów bramek logicznych NAND.



Rys. 5b. Panel czołowy modułu bramek logicznych NAND i NOT.

4.3. Moduł testera stanów logicznych

Moduł testera stanów logicznych zawiera 10 niezależnych testerów (rys. 6). Każdy tester zaopatrzony jest w jedno wejście pomiarowe oraz diody czerwoną i zieloną, których zapalenie symbolizuje stan logiczny odpowiednio 1 i 0. Jednoczesne świecenie diody czerwonej i zielonej oznacza wzbudzenie oscylacji w badanym układzie. Trwałe oscylacje nie powinny występować w poprawnie połączonym układzie automatu asynchronicznego, zatem ich obecność wskazuje na błąd w sieci połączeń. W przypadku gdy na wejściu pomiarowym występuje stan wysokiej rezystancji lub doprowadzone napięcie względem masy nie odpowiada żadnemu stanowi logicznemu obie diody są zgaszone. Wystąpienie takiego stanu na wejściu pomiarowym podłączonym do wyjścia bramki logicznej lub modułu zadawania stanów wskazuje na złamanie przewodu, brak kontaktu, awarię przyrządów lub brak zasilania.



Rys. 6. Panel czółowy modułu testera stanów logicznych.

4.4. Zasilacz

Zasilanie wszystkich modułów doświadczalnych opisanych powyżej zrealizowano przy użyciu jednego zasilacza laboratoryjnego SIGLENT SPD3303D. Spośród trzech kanałów tego zasilacza w tym ćwiczeniu wykorzystywany jest tylko kanał o ustalonym napięciu +5V prądu stałego. Alternatywnie na stanowisku może znajdować się zasilacz dogniazdkowy +5V z przewodem zakończonym wtyczkami bananowymi.

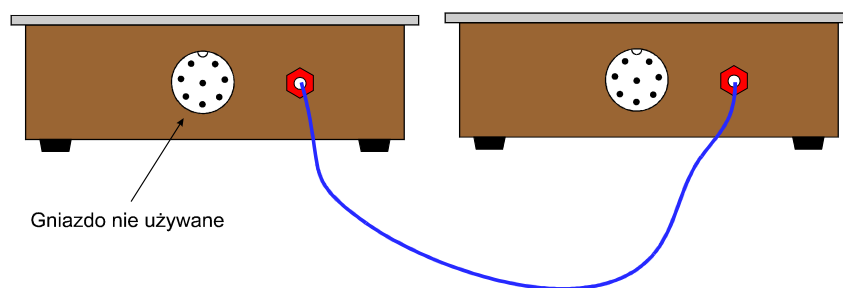
5. Przebieg doświadczenia

Pierwszym krokiem powinno być staranne przygotowanie projektu automatu obejmującego: założenia projektowe, listę stanów pierwotnych, graf przejść, tablicę kodowania stanów, tablice przejść dla stanów pierwotnych, zredukowanych oraz zakodowanych, tablice wyjść, syntezę funkcji przejść i wyjść oraz przewidywany schemat układu. **Ze względu na znaczny nakład pracy niezbędnej na tym etapie zalecane jest przygotowanie brudnopisu projektu przed zajęciami laboratoryjnymi.** Dysponując projektem można przystąpić do połączenia zaprojektowanego układu wykorzystując dostępne bramki logiczne.

Odpowiedź układu bada się testerem stanów logicznych zarówno na liniach stanów wewnętrznych q_1, q_2, \dots, q_k jak i na jego wyjściach y_1, y_2, \dots, y_m . Badania automatu prowadzi się dla wszystkich dozwolonych w projekcie przejść a wyniki zapisuje się w doświadczalnych tablicach przejść i wyjść. Powtarzanie tych samych przejść ma na celu przetestowanie podatności układu na hazard. Wyznaczone doświadczalnie tablice przejść i wyjść porównuje się z tablicami teoretycznymi.

5.1. Kolejność czynności

1. W porozumieniu z prowadzącym zajęcia wybrać zadania do realizacji. Przykładowe zadania zebrano w następnym rozdziale. Wykonawcy ćwiczeń mogą zaproponować własne zadania pod warunkiem, że przedstawiony projekt automatu nie będzie oczywistym plagiatem projektu dostępnego w źródłach publicznych lub opracowaniach innych zespołów, które powinny pracować samodzielnie.
2. Przygotować pełny projekt automatu obejmujący listę stanów pierwotnych, graf przejść, tablicę kodowania stanów, tablice przejść dla stanów pierwotnych, zredukowanych oraz zakodowanych, tablice wyjść, syntezę funkcji przejść i wyjść oraz przewidywany schemat układu.
UWAGA: próba zgadnięcia układu połączeń bez projektu zazwyczaj kończy się niepowodzeniem z wyjątkiem szczególnie prostych zadań.
3. Połączyć z wyłączonym zasilaczem wszystkie niezbędne moduły. W tym celu w skrajnym prawym albo lewym module połączyć przewodami linię masy (na dole modułu) i zasilania +5V (na górze) z wyjściem zasilacza. Zasilanie pozostałych modułów realizuje się poprzez połączenie analogicznych linii w sąsiednich modułach.
UWAGA:
 - a) wszystkie moduły powinny być zasilane z wyjścia zasilacza zapewniającego stałe napięcie +5V (gniazda z prawej strony zasilacza). Nie używać wyjść umożliwiających płynną zmianę napięcia,
 - b) nie wolno łączyć zasilania +5V z wyjściami bramek lub wyjściami zadajnika stanów logicznych.**Nieprzestrzeżenie powyższych zaleceń grozi uszkodzeniem urządzeń.**
4. Jeżeli moduły z elementami logicznymi wyposażone są w dodatkowe gniazda bananowe umieszczone od dołu na blaszanych obudowach, to zalecane jest ich połączenie zgodnie z rys. 7. Połączenie to umożliwi synchronizację pracy modułów.



Rys. 7. Zalecane połączenie między modułami przy realizacji układów sekwencyjnych.

5. Po uzyskaniu zezwolenia włączyć zasilacz i sprawdzić czy świecą się czerwone diody umieszczone na linii +5V. Świecenie diody o innym kolorze oznacza awarię wewnętrzną modułu z elementami logicznymi. W module testera stanów logicznych wszystkie diody symbolizujące stan 0 lub 1 powinny być zgaszone przy rozwartych wejściach testerów.
6. Połączyć zaprojektowany układ, do wejść układu doprowadzić sygnały z modułu zadawania stanów, zaś do testera stanów logicznych doprowadzić zarówno sygnały wyjściowe automatu y_1, y_2, \dots, y_m jak i wewnętrzne q_1, q_2, \dots, q_k . Zmiany w układzie połączeń logicznych mogą być bezpiecznie wykonywane przy włączonym zasilaniu. Układy elektroniczne są zabezpieczone przez przypadkowymi błędnymi połączeniami obejmującymi kilka wyjść bramek logicznych lub wyjść modułu zadawania stanów logicznych.
7. Zadając różne kombinacje sygnałów wejściowych prześledzić kolejno wszystkie przejścia dopuszczone w projekcie układu. Wskazania testera dla wszystkich linii wyjściowych i wewnętrznych automatu nanosić na tablice wyjść oraz przejść dla stanów zredukowanych.
8. Sprawdzić zgodność tablic doświadczalnych wykonanych w punkcie 7 z tablicami przewidzianymi w projekcie. W przypadku stwierdzenia rozbieżności podjąć próbę ich wyjaśnienia i usunięcia przez sprawdzenie luzów w połączeniach realizowanych przy użyciu przewodów i zworek, sprawdzenie poprawności połączeń wykonanych w punkcie 6 oraz ponowne przeanalizowanie projektu. Po dokonaniu poprawek sporządzić ponownie tablice wyjść i przejść automatu. Jeśli próby usunięcia rozbieżności nie powiodą się zawiadomić obsługę pracowni.
9. W celu przetestowania podatności automatu na przejścia zagrożone hazardem zalecane jest powtarzanie badań z punktu 7 w zakresie uwzględniającym dostępny czas zajęć laboratoryjnych.
10. Zapisać podsumowanie wyników badań automatu. Jeżeli układ był poprawiany w porównaniu z pierwszą wersją projektu, to brudnopis powinien zawierać aktualizację zmienionych składników projektu.
11. W obecności opiekuna dydaktycznego wykonać skróconą prezentację działania układu dla niektórych przejść. Przedstawić do zatwierdzenia brudnopis z projektem automatu i wynikami badań.
12. Wyłączyć zasilanie.
13. Rozłączyć połączenia i uprzątnąć stanowisko.

5.2. Propozycje zadań realizowanych przez asynchroniczny układ sekwencyjny

Liczba gwiazdek w nawiasach opisuje stopień trudności zadania. Realizacja zadań o zbyt małej łącznej liczbie gwiazdek powoduje ograniczenie oceny za kompletny i poprawny raport (zasady oceniania podano na końcu rozdziału 6).

1. (*) Zaprojektować i połączyć układ sterowania pojedynczym grzejnikiem na podstawie sygnałów z termometru kontaktowego wyposażonego w dwa kontakty A i B , które przy wzrastającej temperaturze powinny zmieniać swój stan z 0 na 1 w wymienionej kolejności. Oznaczmy przez t_A i t_B temperatury przełączeń kontaktów odpowiednio A i B . W celu zmniejszenia częstotliwości przełączeń grzejnika układ sterowania powinien wykazywać histerezę, tzn. jego działanie powinno być następujące:
 - $t < t_A$ – grzejnik włączony (wyjście układu $Y = 1$),
 - $t_A \leq t < t_B$ – kontynuacja poprzedniego stanu, tzn. jeżeli przed ostatnią zmianą na wejściach było $t < t_A$ to grzejnik pozostaje włączony, a jeżeli przed ostatnią zmianą było $t_B \leq t$ albo termometr wyszedł właśnie ze stanu awaryjnego (opisanego poniżej) to grzejnik pozostaje wyłączony, natomiast jeżeli układ został właśnie włączony to stan wyjścia Y może być dowolny.
 - $t_B \leq t$ – grzejnik wyłączony (wyjście układu $Y = 0$). Przypadek załączenia kontaktu B w termometrze bez załączenia kontaktu A należy uznać za stan awaryjny i konieczne wyłączyć grzejnik.
2. (*) Zaprojektować i połączyć układ przerzutnika o trzech wejściach x_1, x_2, x_3 i trzech wyjściach y_1, y_2, y_3 . Podanie stanu 0 na i -te wejście przy pozostałych wejściach utrzymywanych w stanie 1 powinno spowodować ustawienie i -tego wyjścia w stan 1 i wyzerowanie pozostałych wyjść. Stan wyjść nie powinien ulec zmianie po podaniu 1 na wszystkie wejścia. Do optymalizacji układu wykorzystać możliwość dowolnego zachowania układu w przypadku utrzymywania stanu 0 lub przejść jednoczesnych na dwóch lub trzech wejściach.
3. (***) Zaprojektować i połączyć układ sekwencyjny o trzech wejściach połączonych z przyciskami P (Prawo), L (Lewo) i S (Stop). Układ ten steruje poprzez dwa wyjścia Y_P i Y_L pracą silnika prądu stałego o dwóch kierunkach obrotów, przy czym stan 1 na wyjściu Y_P powoduje obracanie silnika w prawo, zaś 1 na wyjściu Y_L powoduje obroty w lewo. Jednoczesne załączenie stanów 1 na obu wyjściach Y_P i Y_L nie może się zdarzyć w żadnym przypadku. Przyciśnięcie przycisku powoduje podanie stanu 1 na odpowiednie wejście układu. Po zwolnieniu przycisku wejście powraca natychmiast do stanu 0, jednakże załączone obroty silnika powinny być podtrzymywane do czasu wciśnięcia przycisku Stop. Wejście S musi mieć wyższy priorytet niż wejścia P i L , tzn. przytrzymywanie przycisku S blokuje reakcję na wciskanie przycisków P i L . Do optymalizacji układu wykorzystać założenie, że jednoczesne wciśnięcie przycisków P i L a także przyciśnięcie jednego z nich podczas przytrzymywania drugiego może prowadzić do dowolnej reakcji układu (z wyłączeniem jednoczesnego załączenia stanów 1 na obu wyjściach). **Uwaga:** w module zadawania stanów logicznych nie są dostępne przyciski o właściwościach opisanych w zdaniu. Przyciśnięcie oraz puszczenie przycisku należy zasymulować przez dwa kolejne przyciśnięcia dostępnego przełącznika.
4. (***) Zaprojektować od postaw i połączyć układ dwójki liczącej, tzn. układ, który przełącza stan swojego wyjścia Y na przeciwny po wykryciu przejścia $0 \rightarrow 1$ na jego wejściu X . W praktyce układ taki zazwyczaj realizuje się przy wykorzystaniu standardowych przerzutników D, T lub JK-MS, jednakże w tym zadaniu należy stwierdzić czy ograniczenie funkcji układu do dwójki liczącej daje możliwość dokonania uproszczeń w porównaniu do realizacji kompletnych przerzutników z bramek logicznych?
5. (***) Zaprojektować i połączyć układ przerzutnika o dwóch wejściach x_1, x_2 i dwóch wyjściach Q i \bar{Q} . Przerzutnik ten powinien działać analogicznie do standardowego przerzutnika RS, ale zapisywanie stanu przerzutnika powinno odbywać się podczas przejść $0 \rightarrow 1$ na jego wejściach a nie poprzez utrzymywanie stanu 0 (alternatywnie 1 dla układu z bramek NOR) jak w przerzutniku RS. W przypadku przejść jednoczesnych na obu wejściach dopuścić nieokreślone zachowanie układu.

6. (***) Zaprojektować i połączyć układ o dwóch wejściach A i B , który na wyjściu Y wskazuje gdzie zaszła ostatnia zmiana stanu wejściowego. Układ powinien rozpoznawać zarówno przejścia $0 \rightarrow 1$ jak i przejścia $1 \rightarrow 0$. Jeżeli ostatnia zmiana nastąpiła na wejściu A , to wyjście powinno przyjąć stan $Y = 1$, jeżeli ostatnia zmiana nastąpiła na wejściu B , to $Y = 0$. Do optymalizacji układu wykorzystać założenie, że wykrycie jednoczesnych przejść na obu wejściach może prowadzić do dowolnej reakcji układu.
7. (***) Zaprojektować i połączyć układ dzielący częstotliwość sygnału wejściowego przez 3. Jeżeli częstotliwość na wejściu jest ustalona i współczynnik wypełnienia sygnału wynosi 50%, to współczynnik wypełnienia sygnału na wyjściu także powinien być równy 50%.
8. (***) Zaprojektować i połączyć układ, który na podstawie dwubitowej liczby binarnej steruje linijką złożoną z trzech diod świecących, w taki sposób, że liczba załączonych kolejnych diod odpowiada zapamiętanej przez układ maksymalnej wartości liczby binarnej na wejściu układu. Ponadto układ posiada wejście resetujące pamięć, przy czym reset może nastąpić wyłącznie po przejściu $0 \rightarrow 1$ i nie jest przedłużany na czas podtrzymywania stanu ustalonego 1. Reset powinien spowodować zapamiętanie bieżącej wartości liczby binarnej na wejściu jako największej zarejestrowanej dotąd wartości.
9. (***) Zaprojektować i połączyć układ, który sygnalizuje stanem 1 na pojedynczym binarnym wyjściu Y wzrost temperatury mierzonej przez termometr kontaktowy względem stanu termometru zapamiętanego podczas resetu automatu. Termometr wyposażony jest w trzy kontakty A , B i C , które podczas wzrostu temperatury przechodzą ze stanu 0 na 1 w wymienionej kolejności, tzn. możliwe kombinacje sygnałów wyliczone w kolejności od najniższej do najwyższej temperatury to: $ABC = 000, 100, 110$ oraz 111 . Kasowanie pamięci wzrostu temperatury następuje wyłącznie przez podanie stanu 1 na wejście R (reset) automatu, natomiast nie może zdarzyć się na skutek spadku temperatury po wcześniejszym zarejestrowaniu wzrostu. **Uwaga:** warunek resetowania opisanego automatu odróżnia go od automatu zaprojektowanego w tej instrukcji w rozdziale „Metoda syntezy i minimalizacji asynchronicznego układu sekwencyjnego”.
10. (***) Do wejść A i B układu sekwencyjnego przyłączono wyjścia dwóch komparatorów analogowych pracujących jako detektory znaku napięcia (tzn. wejście „-” każdego komparatora jest zwarte z masą, a do wejścia „+” przyłączony jest sygnał, dla którego trzeba określić chwilowy znak napięcia). Do wejść komparatorów doprowadzono dwa okresowe przebiegi analogowe o takim samym kształcie i okresie ale przesunięte w fazie. Zaprojektować i połączyć automat, który wykrywa i podaje na swoim wyjściu Y znak przesunięcia fazowego sygnału z wejścia A względem wejścia B . Rozważyć możliwie najprostszą realizację układu. Podczas poprawnej eksploatacji układu nie są możliwe dwa kolejne przejścia na wybranym wejściu bez przejścia na drugim wejściu. Naruszenie tego założenia może prowadzić do dowolnej reakcji układu.

6. Wskazówki do raportu

Raport powinien zawierać:

1. Stronę tytułową (wg wzoru).
2. Sformułowanie celu ćwiczenia.
3. Wykaz użytej aparatury. Dla modułu/modułów bramek logicznych podać także pełną specyfikację dostępnego zestawu bramek logicznych (typy bramek, liczba wejść, liczba dostępnych bramek).
4. Treść zadania, które powinien realizować zbudowany automat.

5. Kompletny projekt asynchronicznego układu sekwencyjnego wykonującego postawione zadanie. **UWAGA:** Metoda projektowania układu nie jest narzucona, jednakże w każdym przypadku należy przedstawić kolejne etapy projektowania. Raporty, w których nie wykazano zrozumienia jakiegokolwiek metody projektowania będą zwracane do uzupełnienia niezależnie od wyników badania zrealizowanego układu.
6. Schemat połączeń automatu z bramek logicznych.
7. Doświadczalne tablice przejść i wyjść zbadanego układu.
8. Dyskusję uzyskanych wyników i wnioski. Przedstawić rezultat sprawdzenia zgodności zbudowanego automatu z tablicami przejść i wyjść przewidzianymi w projekcie. W przypadku wystąpienia rozbieżności opisać środki podjęte w celu ich usunięcia, znalezione błędy i uzyskany ostatecznie rezultat. Czy z perspektywy czasu zauważono możliwość dalszej minimalizacji układu? Czy treść zadania wymagała przyjęcia dodatkowych założeń projektowych i pierwotna postać założeń zostałaby utrzymana podczas ponownej realizacji układu? Dodatkowe wnioski dotyczące organizacji zajęć, przyrządów i instrukcji do ćwiczenia mogą być pomocne w usprawnieniu ćwiczenia wykonywanego przez studentów w kolejnym roku akademickim.

W raporcie ocenie podlegać będzie obecność i poprawność wszystkich wymienionych powyżej składników, czytelność prezentacji wyników w postaci list, tabel, grafów, wzorów i schematów wraz z opisami oraz jakość sformułowanych wniosków. Wstęp teoretyczny nie jest wymagany i w przypadku jego zamieszczenia w raporcie nie wpłynie na ocenę.

Ponadto ocena za poprawny i kompletny raport zależy od łącznej liczby gwiazdek opisujących trudność zadań, których wykonanie udokumentowano w zatwierdzonym brudnopisie. Jeżeli prowadzący zajęcia nie poda inaczej obowiązuje następująca tabela:

Suma gwiazdek za wykonane zadania	Maksymalna ocena ze sprawozdania w skali 0...5 pkt.
0	nie zaliczone
1	3 pkt.
2	4 pkt.
≥ 3	5 pkt.

7. Literatura

- [1] P. Misiurewicz, *Układy automatyki cyfrowej*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1984.
- [2] H. Kamionka-Mikuła, H. Małysiak, B. Pochopień, *Synteza i analiza układów cyfrowych*, Wydawnictwo Pracowni Komputerowej Jacka Skamierskiego, Gliwice 2006.
- [3] A. Skorupski, *Podstawy Techniki Cyfrowej*, WKiŁ, Warszawa 2004.
- [4] J. Kalisz, *Podstawy elektroniki cyfrowej*, WKiŁ, Warszawa 2002.
- [5] W. Traczyk, *Układy cyfrowe. Podstawy teoretyczne i metody syntezy*, WNT, Warszawa 1986.
- [6] C. Zieliński, *Podstawy projektowania układów cyfrowych*, PWN, Warszawa 2003.
- [7] A. Barczak, J. Florek, T. Sydoruk, *Elektroniczne Techniki Cyfrowe*, VIZJA PRESS&IT Sp. z o.o., Warszawa 2006.
- [8] U. Tietze, Ch. Schenk, *Układy półprzewodnikowe*, WNT, Warszawa 2009.
- [9] W. Głocki, *Układy cyfrowe*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 2008.