Pojęcia podstawowe

Jeżeli stan wyjść układu cyfrowego zależy nie tylko od aktualnego stanu wejść, lecz także od poprzednich stanów wejść, jest to ***układ sekwencyjny***. Uwzględnienie wpływu stanów poprzednich jest możliwe tylko wówczas, gdy układ jest wyposażony w pamięć. Posiadanie pamięci jest podstawową cechą układu sekwencyjnego. Działanie układu sekwencyjnego można opisać przy pomocy:

- stanów wejść: $\mathbf{X} = \{X_0, X_1, ..., X_{N-1}\}$, $X = \{x_1, x_2, ..., x_n\}$
- stanów wyjść: $\mathbf{Y} = \{Y_0, Y_1, ..., Y_{M-1}\}$, $Y = \{y_1, y_2, ..., y_m\}$
- stanów pamięci (stanów wewnętrznych) $\mathbf{A} = \{A_0, A_1, ..., A_{K-1}\}$, $A = \{Q_1, Q_2, ..., Q_k\}$

Do właściwego funkcjonowania pamięci potrzeba, aby jej stan również zależał od stanu wejść i poprzedniego stanu pamięci. Wobec tego układ sekwencyjny można opisać funkcjami:

$$\delta: \mathbf{A} \times \mathbf{X} \rightarrow \mathbf{A} \qquad 1.1$$

$$\lambda: \mathbf{A} \times \mathbf{X} \rightarrow \mathbf{Y} \qquad 1.2$$

Funkcje δ i λ to odpowiednio funkcja przejść i funkcja wyjść.
Tak zdefiniowany model matematyczny układu sekwencyjnego jest nazywany ***automatem***. Uściślając, związki 1.1 i 1.2 opisują tzw. ***automat Mealy'ego***.

Warto zauważyć, że układ kombinacyjny jest szczególnym przypadkiem tak opisanego układu sekwencyjnego, gdyż dla $\mathbf{A} = 1$:

$$\lambda: \mathbf{X} \rightarrow \mathbf{Y}$$

Jeśli stan pamięci zależy od stanu wejść, to stan wyjść można uzależnić wprost tylko od stanu pamięci (zależność od stanu wejść będzie pośrednia)

$$\delta: \mathbf{A} \times \mathbf{X} \rightarrow \mathbf{A} \qquad 1.3$$

$$\lambda: \mathbf{A} \rightarrow \mathbf{Y} \qquad 1.4$$

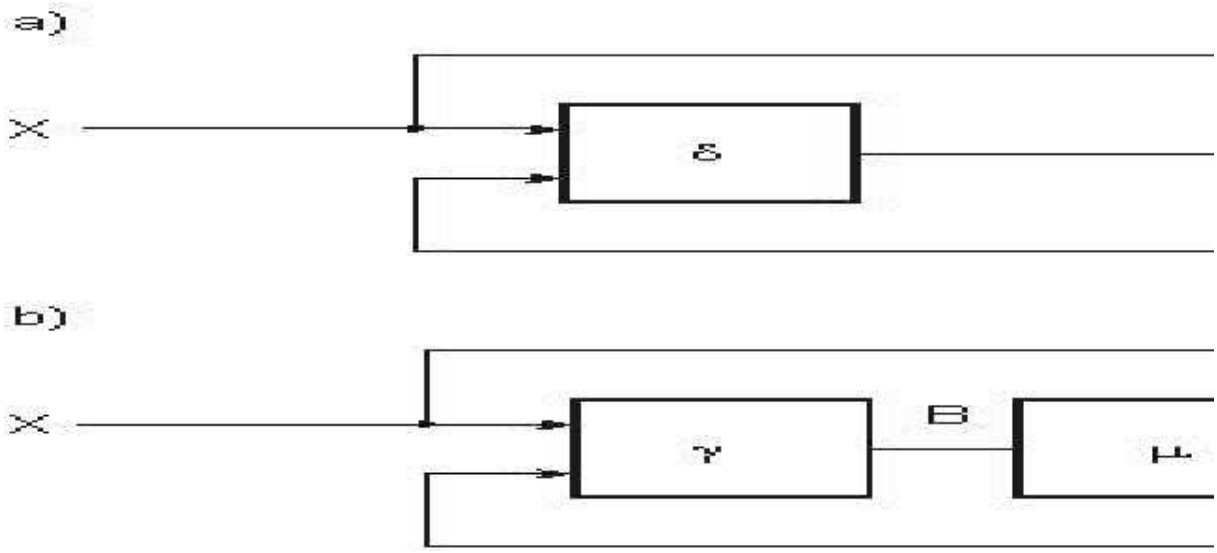
Tak zdefiniowany model matematyczny nazywamy ***automatem Moore'a***.

Podział automatów

W rozważanych poprzednio układach, zbiory **A, X, Y** są skończone i dlatego opisujący je model jest nazywany ***automatem skończonym***. Zadeklarowanie δ i λ jako funkcji określa ***automat deterministyczny***. Ogólna teoria automatów zajmuje się także ***automatami niedeterministycznymi***, w których parze stanów A, X może odpowiadać więcej niż jeden nowy stan A lub więcej niż jeden stan wyjść Y. W szczególnym przepadku ***automatów probabilistycznych*** operuje się prawdopodobieństwami uzyskania odpowiednich nowych stanów A lub stanów wyjść Y.
Automat nazywamy ***zupełnym***, jeśli jego funkcje przejść i wyjść są określone dla wszystkich par (A, X) ze zbioru $\mathbf{A} \times \mathbf{X}$.

Struktury automatów

Z zależności 1.1 i 1.2 wynika podstawowa struktura automatu (a) przedstawiona na poniższym rysunku:



Struktury układów sekwencyjnych.

Blok realizujący funkcję λ jest układem kombinacyjnym, natomiast zadaniem bloku opisanego funkcją δ jest zapamiętywanie potrzebnych informacji. Skoro wektor A określa stan pamięci, to w bloku opisanym funkcją δ można wydzielić część pamięciową μ (o sygnałach wyjściowych Q, tworzących wektor A) i część pomocniczą gamma przygotowującą sygnały dla bloku pamięci. Wprowadzenie funkcji pomocniczej umożliwi zastosowanie standardowych elementów pamięciowych, bowiem zadanie dostosowania pamięci do potrzeb funkcji δ przejmuje funkcja γ (b). Mamy więc:

$$A = \delta(A, X) = \mu(\gamma(A, X))$$

lub wprowadzając stan pośredni:

$$\gamma(A, X) = B, \qquad \mu(B) = A$$

gdzie:

B – **wektor wzbudzeń** elementów pamięci

$$B = (q_a, q_b, ..., q_z)$$

Warto zauważyć, że blok opisany funkcją γ jest układem kombinacyjnym, dlatego dla uproszczenia realizacji można go połączyć z blokiem λ (c), co prowadzi do wniosku:

> Układ **sekwencyjny** składa się z układu **kombinacyjnego** i elementów **pamięci**.

Problem czasu w automatach

W opisie układów kombinacyjnych pomijaliśmy dotychczas problem czasu realizacji funkcji, gdyż – poza oczywistym opóźnieniem wyników przetwarzania – nie miał on wpływu na samą funkcję. W stosunku do układów kombinacyjnych nadal możemy tak postępować.

$$Y = \lambda(A, X), B = \gamma(A, X) \text{ itd.}$$

Jednakże w układach odpowiedzialnych za pamiętanie stanów czas odgrywa bardzo ważną rolę, trzeba go uwzględnić:

$$\delta(A^t, X^t) = A^{t+\tau}$$

Wielkość τ symbolizuje zwłokę, wprowadzoną przez układ realizujący funkcję δ (w rzeczywistości każdy sygnał Q wektora A może mieć inny czas przejścia przez układ, ale dla ogólnych rozważań nie jest to istotne). Taki opis umożliwia odróżnienie stanu poprzedniego A^t od następnego $A^{t+\tau}$, co bywa też zapisywane w postaci:

$$\delta(A, X) = A'$$

Układy asynchroniczne i synchroniczne

Współczesne układy półprzewodnikowe mają czas opóźnienia rzędu nanosekund. Zatem jeśli dla jakiegoś stanu X_i jest $\delta(A_a, X_i) = A_b$, $\delta(A_b, X_i) = A_c$, $\delta(A_c, X_i) = A_d$, itd., to przejście układu od stanu A_a do A_d (lub dalej) uniemożliwiają określenie chwili zmiany, np. chwili pojawienia się stanu A_c dla zmiany stanu X_i w X_j. Takie lawinowe zmiany stanu pamięci w zupełnie przypadkowych chwilach są niepożądane i dlatego naturalne przejścia zachowuje się tylko w przypadkach, gdy:

$$(\forall X_i, A_j) \qquad \delta(A_j, X_i) = A_k \Rightarrow \delta(A_k, X_i) = A_k \qquad 1.5$$

Stan A_k jest ***stabilny*** przy X_i, a układ spełniający powyższy (1.5) warunek jest nazywany układem ***asynchronicznym***.

Jego cechą charakterystyczną jest to, że wszelka zmiana stanów A lub Y może w nim wystąpić jedynie pod wpływem zmiany stanu X.

Przykładem asynchronicznego układu sekwencyjnego może być układ sterowania dźwigu osobowego. Zmiana położenia klatki następuje tylko pod wpływem sygnałów z przycisków zewnętrznych lub wewnętrznych.
Zupełnie inaczej działa kalkulator elektroniczny. Równoczesne przetwarzanie wielocyfrowych liczb jest bardzo kosztowne i dlatego w kalkulatorach działania wykonuje się kolejno (sekwencyjnie) na poszczególnych cyfrach. Naciśnięcie przycisku dowolnej operacji wywołuje więc całą sekwencję działań, bez dalszego udziału sygnałów zewnętrznych. W celu uporządkowania tych działań i uniknięcia niekontrolowanej zmiany stanów wprowadza się w takich przypadkach specjalny sygnał taktujący, zwany sygnałem zegara. Jego zadaniem jest narzucanie rytmu zmian stanów wewnętrznych układu.
Obecność sygnałów taktujących jest niezbędna, zwłaszcza w urządzeniach, gdzie są przetwarzane szeregowo (bit po bicie) przekazywane informacje cyfrowe. Jej nośnikami są stany 0 i 1. Stąd odróżnienie ciągu 110000 od 10 jest możliwe tylko wówczas, gdy istnieje sposób wyróżniania poszczególnych chwil odpowiadających sygnałom binarnym. Układy, w

których zmiana stanu wewnętrznego jest synchronizowana sygnałami taktującymi naszą nazwę ***układów synchronicznych***. Impulsy taktujące w tych układach dzielą czas na odcinki zwane **taktami**, które można ponumerować, nadając przez to czasowi charakter ziarnisty (dyskretny).
Przyjmując t = 0, 1, 2, ... można układ synchroniczny opisać równaniami:

$$A^{t+1} = \delta(A^t, X^t), \qquad Y^t = \lambda(A^t, X^t)$$

lub

$$A^{t+1} = \mu(B^t), \qquad B^t = \gamma(A^t, X^t), \qquad Y^t = \lambda(A^t, X^t)$$

Zależności te dotyczą bardziej ogólnego przypadku – układów ***Mealy'ego***.
Pominięcie X^t w funkcjach δ i γ daje opisy układów ***Moore'a***.

Impulsy taktujące mają powodować zmianę stanu wewnętrznego tylko w ściśle określonych chwilach, są więc jak gdyby impulsami strobującymi, pobierającymi próbki sygnałów ze stanów A i X (lub B) i kierującymi je do elementów pamięci. Takie zadanie impulsów zegara najprościej można zrealizować przez bramkowanie (w elementach AND lub NAND) wszystkich sygnałów wzbudzeń sygnałem taktującym c. Jednak takie rozwiązanie daje złe wyniki.

W przypadku pierwszym zmiana stanu A następuje jeszcze w czasie trwania impulsu c. Nowy stan ma więc możliwość wygenerowania kolejnego stanu (jeszcze nowszego) aż do zaniku impulsu c. Jak łatwo zauważyć, jest analogia do przerzutnika typu D-Latch, w którym wejście zegarowe bramkowało sygnał informacyjny D. Więcej niż jedna zmiana stanu A w czasie trwania jednego taktu jest niezgodna z założeniami. Takie rozwiązanie należy odrzucić. Próbą walki z tą niedogodnością może okazać się skracanie impulsu c tak, aby nastąpiła tylko jedna zmiana stanu A. Jednak przy nieuniknionych technologicznych rozrzutach parametrów układu oraz skończonym pasmem przenoszenia sygnału takie rozwiązanie grozi błędami w działaniu elementów pamięci.

Proste ominięcie opisanych problemów polega na takim opóźnieniu reakcji elementów pamięciowych, aby zmiana stanu A nastąpiła już po zakończeniu impulsu c. Nowa zmiana może wówczas nastąpić dopiero po następnym impulsie zegarowym. Układ zadziała zgodnie z założeniami.

Sposoby opisu automatów

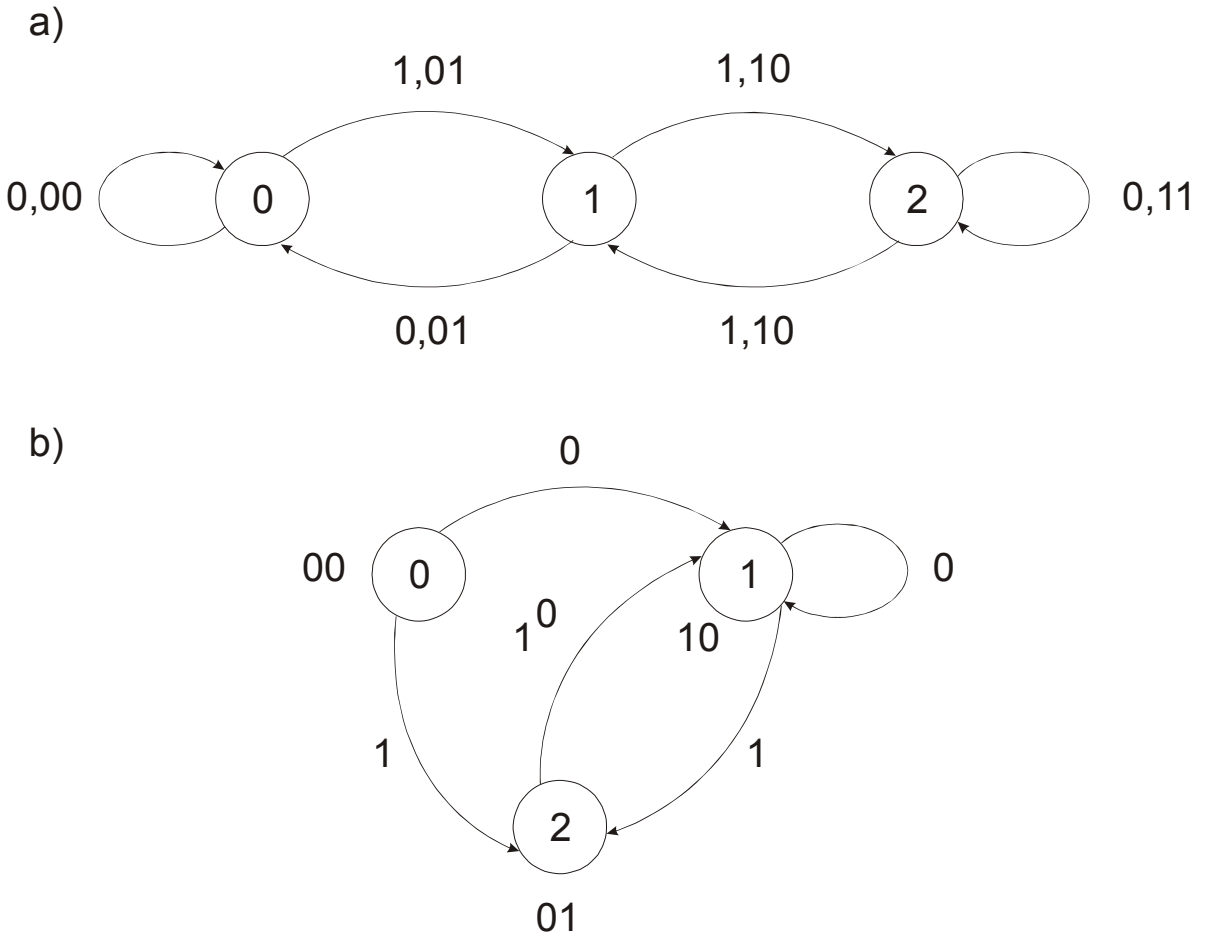
W układach kombinacyjnych podanie stanów wejść i odpowiadających im stanów wyjść w pełni opisywało funkcje przełączające, a tym samym zadania układu i sposób jego działania.
W układach sekwencyjnych podanie ciągów stanów wejść i odpowiadających im ciągów stanów wyjść stanowi tylko zewnętrzny opis układu, nie określający ani pamięci, ani funkcji przejść i wyjść. Dla uwzględnienia tych parametrów układu potrzebny jest inny opis, wiążący wszystkie elementy modelu automatu i wprowadzony już po określeniu niezbędnej pamięci układu.

Opis słowny powinien przyporządkowywać sygnałom **wejściowym** w odpowiednim porządku ich występowania, sygnały **wyjściowe**. W zależności od rodzaju przekształcenia i liczby sygnałów, może to być opis bardzo prosty lub bardzo rozbudowany. Na przykład wymaganie „zbudować licznik zliczający impulsy wejściowe w naturalnym kodzie binarnym od 0 do 15", określa układ prawie jednoznacznie. Można się spierać, czy mamy zliczać impulsy rozpoczęte i zakończone oraz w jaki dokładnie sposób zrealizować ten licznik (jakie przerzutniki, a może

gotowy układ 7493 lub inny), jednak opis licznika zmusza projektanta do zbudowania konkretnego układu. W innym zadaniu: „Zbudować kontroler portu drukarki" potrzeba wielu dodatkowych założeń. Opis słowny powinien podawać nie tylko relacje wejście-wyjście, lecz także zakres określoności tej relacji, tzn. dziedzinę X.

Wykresy czasowe są pewnym ukonkretnieniem *opisu słownego*, ponieważ opisują kolejność występowania stanów X i odpowiadających im stanów Y. Zazwyczaj oś czasu nie jest skalowana, choć czasem może stanowić podstawę do oceny czasu zwłaszcza w układach generujących impulsy o znanej długości. Na podstawie wykresów czasowych można odczytać czy jest to układ sekwencyjny. W układzie sekwencyjnym identycznym stanom X odpowiadają różne stany Y.

Grafy przejść i wyjść sporządza się przypisując stanom pamięci A wierzchołki grafu, a stanom wejść X – gałęzie grafu, tak aby stan X_k, zmieniający stan A_i w A_j, opisywał gałąź skierowaną od A_i do A_j. W ten sposób jest opisana funkcja przejść δ. Stany wyjść Y opisuje się dwojako. Ponieważ w układach Moore'a zależą one tylko od stanów A, więc przypisuje się je odpowiednim wierzchołkom grafu. W układach Mealy'ego dochodzi jeszcze zależność stanów Y od X i dlatego, jeśli $Y_a = \lambda(A_i, X_k)$ to Y_a wpisuje się obok gałęzi X_k wychodzącej z wierzchołka A_i. Kilka gałęzi skierowanych od A_i do A_j można zastąpić jedną, umieszczając obok siebie odpowiednie symbole X i ewentualnie Y.
Poniższy rysunek przedstawia omawiane grafy:



Grafy układów Mealy'ego (a) i Moore'a (b)

Stan wejść jest opisany jednym bitem: X = (x), a stan wyjść dwoma bitami: Y = (y_1, y_2), a zamiast A_i w wierzchołki wpisano po prostu i. W automacie Mealy'ego gałęzie opisuje się parami (X, Y).

Tablice przejść i wyjść są pełnym odpowiednikiem grafów, mniej obrazowym, ale znacznie wygodniejszym w przekształceniach. Dlatego osobom nie dość wprawnym, zaleca się wpierw stworzenie grafu danego automatu, a dopiero potem, na podstawie otrzymanego grafu stworzenie odpowiedniej tablicy przejść i wyjść. Tablica przejść opisuje funkcję δ, a więc każdej parze (A, X) przypisuje nowy stan A.
Poniższy rysunek przedstawia tablice przejść i wyjść odpowiadające grafom z poprzedniego rysunku.

a)

		X	
		0	1
A	0	0	1
	1	0	2
	2	2	1

b)

		X	
		0	1
A	0	00	01
	1	01	10
	2	11	10

c)

		X		Y
		0	1	
A	0	1	2	00
	1	1	2	10
	2	1	1	01

Tablica: przejść dla automatu Mealy'ego (a), wyjść dla automatu Mealy'ego (b), dla automatu Moore'a (c).

Minimalizacja liczby stanów wewnętrznych

Otrzymanie tablic przejść i wyjść jest ważnym etapem syntezy automatu, gdyż precyzuje wszystkie elementy jego opisu w modelu matematycznym. Dalsze postępowanie polega na takim przekształceniu tego opisu, aby realizacja automatu stała się możliwie prosta. Ponieważ – podobnie jak w układach kombinacyjnych – stopień złożoności układu sekwencyjnego zależy od rodzaju zastosowanych elementów, a na rozważanym poziomie opisu funkcje mają dopiero postać tabelaryczną. Jedyną możliwością uproszczenia automatu jest zmniejszenie jego stanów wewnętrznych. Można się spodziewać, że funkcje działające na mniejszych zbiorach będą prostsze i obsługa mniejszej liczby stanów będzie wymagała mniejszej liczby elementów.
Podstawowym zadaniem układu sekwencyjnego jest przetwarzanie w zadany sposób ciągów *p* stanów X w ciągi *q* stanów Y. Oznaczając pierwotny opis układu jako:

$$\boldsymbol{A}_P = (\mathbf{A}_P, \mathbf{X}, \mathbf{Y}, \delta_P, \lambda_P)$$

można problem minimalizacji liczby stanów uważać za poszukiwanie układu o opisie:

$$\boldsymbol{A}_M = (\mathbf{A}_M, \mathbf{X}, \mathbf{Y}, \delta_M, \lambda_M)$$

przy czym:

$\overline{\overline{A_M}} \le \overline{\overline{A_P}}$ i nie istnieje $\boldsymbol{A}_K$ taki, że $\overline{\overline{A_K}} \le \overline{\overline{A_M}}$ i automaty $\boldsymbol{A}_M$ i $\boldsymbol{A}_K$ reagują tak samo jak $\boldsymbol{A}_P$. Określenie „reagują tak samo" oznacza, że jeśli automat $\boldsymbol{A}_P$ na ciąg p_i odpowiada ciągiem q_i, to również automaty $\boldsymbol{A}_M$ i $\boldsymbol{A}_K$ odpowiadają identycznym ciągiem q_i wszędzie tam, gdzie q_i jest określony.

Automat $\boldsymbol{A}_M$, reagujący tak samo jak $\boldsymbol{A}_P$ i mający najmniejszy zbiór stanów pamięci, jest nazywany ***automatem minimalnym***.

Liczba stanów automatu może zostać zmniejszona (bez zmiany wyników działania), jeśli co najmniej dwa jego stany wewnętrzne można zastąpić jednym, tzn. gdy dwa wiersze jego tablicy przejść i wyjść można zastąpić jednym. Zatem warunki tego zastępowania są kluczem do rozwiązania problemu minimalizacji.

W najprostszym ujęciu problemu minimalizacji stanów wewnętrznych automatu można uzależnić od koniunkcji dwóch warunków:

1. $S_1 \Leftrightarrow S_2$ gdy $Y(S_1) = Y(S_2)$
2. $S_1 \Leftrightarrow S_2$ gdy $S_1 \to S_3 \wedge S_2 \to S_3$

Stany S_1 i S_2 można połączyć ze sobą gdy odpowiadające im stany wyjść są sobie równe oraz gdy zarówno stan S_1 jak i stan S_2 przechodzą w ten sam stan S_3. Możliwe jest warunkowe łączenie stanów:

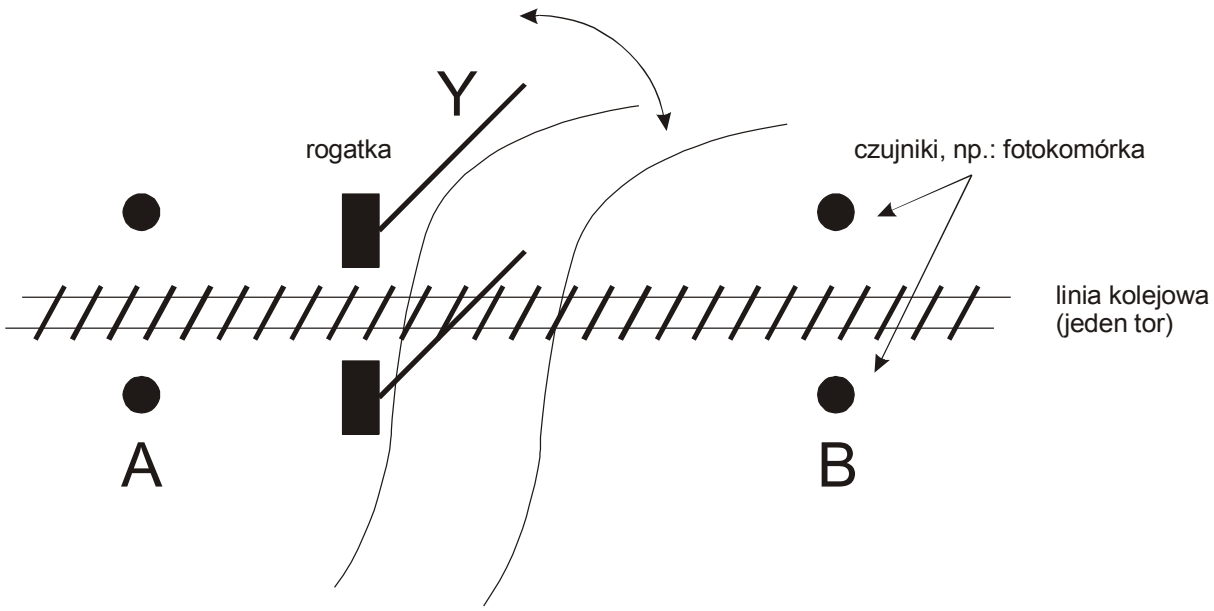
1. $S_1 \Leftrightarrow S_2$ gdy $Y(S_1) = Y(S_2)$
2. $S_1 \Leftrightarrow S_2$ gdy $S_1 \to S_3 \wedge S_2 \to S_4$ ale $S_3 \Leftrightarrow S_4$

Stany S_1 i S_2 można połączyć ze sobą gdy odpowiadające im stany wyjść są sobie równe oraz gdy stan S_1 przechodzi w inny stan niż S_2 ($S_1 \to S_3 \wedge S_2 \to S_4$) ale stany S_3 i S_4 można ze sobą połączyć.

Przykłady – przejazd kolejowy

Przedstawiony wcześniej opis teoretyczny przedstawimy teraz na konkretnym przykładzie przejazdu kolejowego. Naszym zadaniem będzie zaprojektowanie logiki (automatu) sterującej przejazdem kolejowym. Na początku należy przyjąć pewne założenia:

1. Przejazd dotyczy linii jednotorowej.
2. Opuszczenie rogatki powinno się odbyć z pewnym wyprzedzeniem.
3. Podniesienie rogatki powinno się odbyć z pewnym opóźnieniem.
4. Wykluczamy jednoczesny przejazd pociągu przez oba punkty kontroli.
5. Wykluczamy przejazd więcej niż jednego pociągu.
6. Pociąg w danej chwili może jechać tylko w jednym kierunku
7. Linia kolejowa jest dwukierunkowa.
8. Pociąg, który dotarł do punktu A lub B nie może się cofać.

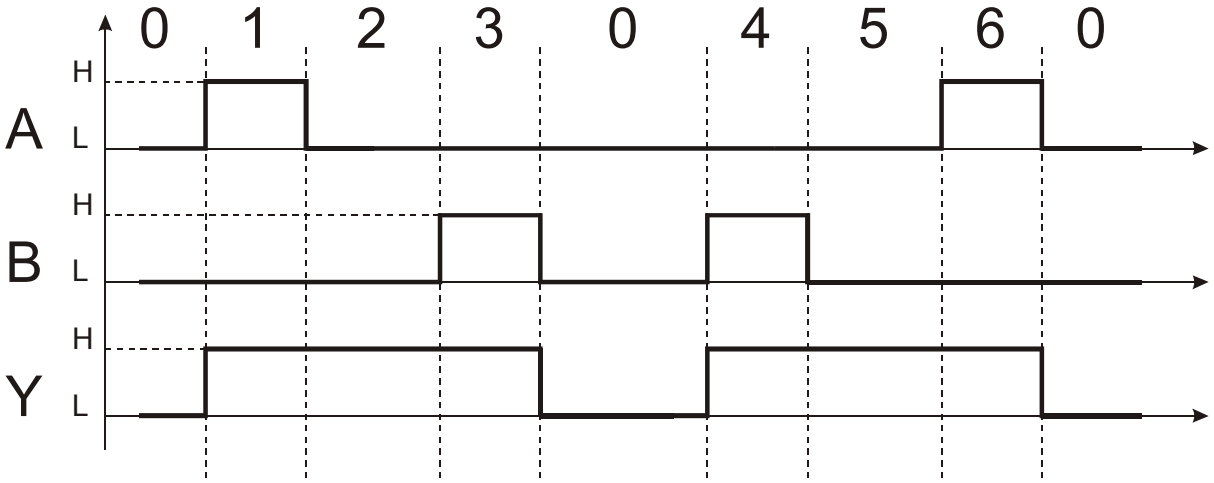


Powyższy rysunek przedstawia uproszczony schemat przejazdu kolejowego.

A, B – punkty kontrolne
Y – Rogatka

Punkty kontrolne A i B przyjmują wartość 1 podczas przejazdu pociągu (przez te punkty) lub wartość 0 gdy brak pociągu. Rogatka Y przyjmuje wartość 1 gdy jest opuszczona lub wartość 0 gdy jest podniesiona.

Na podstawie opisu słownego możemy przejść do narysowania przebiegów czasowych:



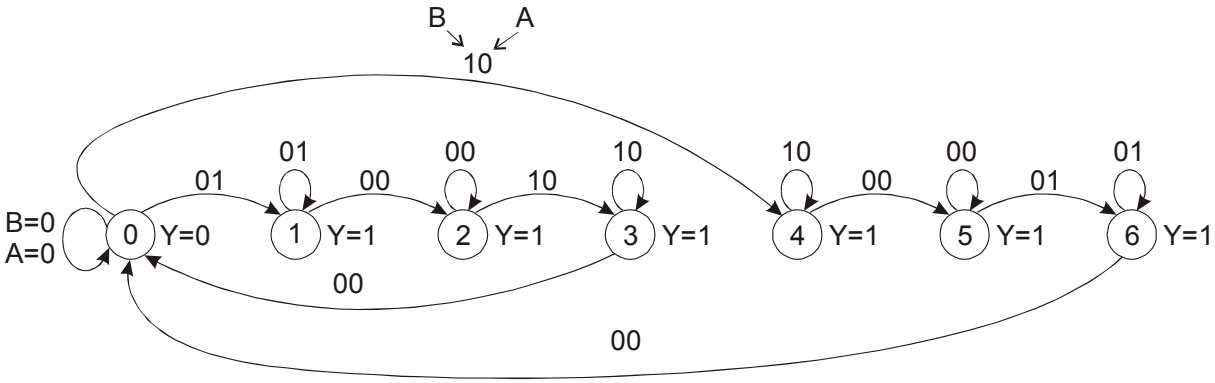
Przebiegi czasowe dla wejść (A, B) i wyjść (Y) przejazdu kolejowego.

Cyframi od 0 do 6 wstępnie ponumerowano stany w jakich może znajdować się układ. Jak się okaże, nie jest to liczba minimalna.

Stan 0 odpowiada takiej chwili czasowej, podczas której pociąg jeszcze nie dojechał do punktów A lub B. Zakładając, że pociąg jedzie od lewej, to najpierw dojedzie do punktu A. W tym momencie (zgodnie z założeniami) wartość wejścia A = H oraz rogatka zostaje

opuszczona (Y = 1). Pociąg opuszcza punkt A (A=0), ale jeszcze nie dojechał do punktu B. Taka sytuacja odpowiada na wykresie stanowi 2. Pociąg dojeżdża do punktu B. W tym momencie (zgodnie z założeniami) wartość wejścia B = H (stan 3). Gdy pociąg opuszcza punkt B (B = 0) rogatka powinna zostać podniesiona Y = 0.
Analogiczna sytuacja wystąpi gdy pociąg będzie jechać od prawej do lewej.

W celu dodatkowego zobrazowania działania automatu można narysować graf:



Graf automatu Moore'a dla przejazdu kolejowego.

Jak łatwo zauważyć powyższy graf jest odpowiada automatowi Moore'a. Dzieje się tak dlatego że stan następny **nie** zależy od stanu poprzedniego wyjścia Y. Wartości wpisane w wierzchołki grafu (kółka) odpowiadają numerom stanu, które nadaliśmy przy okazji omawiania przebiegów czasowych. Wartości bitów wejść B i A (właśnie w takiej kolejności!) wpisano przy gałęziach grafu (linie ze strzałkami, strzałka pokazuje kierunek przejścia). Aktualny stan wyjścia Y wpisano tuż obok węzłów.

Ponieważ graf nie jest zbyt dobrym medium do minimalizacji automatu, zatem przejdziemy teraz do sporządzenia tabeli przejść i wyjść.

S_n	BA 0 0	BA 0 1	BA 1 0	Y
0	0	1	4	0
1	2	1	-	1
2	2	-	3	1
3	0	-	3	1
4	5	-	4	1
5	5	6	-	1
6	0	6	-	1

Tablica przejść i wyjść dla automatu Moore'a opisująca działanie przejazdu kolejowego.

Przykładowo rozważmy pierwszy wiersz tabeli dla $S_n = 0$. Jeśli automat znajduje się w stanie 0 to po podaniu na jego wejścia B = 0 i A = 0 automat zmieni stan na 0 (czyli sam w siebie). Natomiast jeśli B = 1 i A = 0 to automat przejdzie do stanu 4. Stan wyjścia Y (czyli fakt opuszczenia lub podniesienia rogatki) dla stanu $S_n = 0$ wynosi Y = 0. Pozostałe linijki opisują zachowanie się automatu dla pozostałych stanów S_n oraz różnych sygnałów wejściowych BA.

Stan oznaczony przez „-„ to stan który nie wystąpi w trakcie działania automatu. Podczas minimalizacji liczby stanów można potraktować go jako dowolny tzn. (jak się okaże) można bezwarunkowo połączyć ze sobą stan 1 i 2.

Przystępujemy do minimalizacji stanów wewnętrznych automatu. Chodzi o to, aby max(S_n) było mniejsze od aktualnej liczby stanów, czyli max(S_n') < 7. Trzeba spróbować połączyć niektóre stany ze sobą. Na podstawie poprzednio stworzonej tabeli przejść i wyjść można znaleźć takie stany, które można ze sobą połączyć. Potrzebna będzie znajomość warunków, które muszą zostać spełnione, aby dwa stany mogły być zastąpione jednym.

1. $S_1 \Leftrightarrow S_2$ gdy $Y(S_1) = Y(S_2)$
2. $S_1 \Leftrightarrow S_2$ gdy $S_1 \to S_3 \wedge S_2 \to S_3$

lub:

1. $S_1 \Leftrightarrow S_2$ gdy $Y(S_1) = Y(S_2)$
2. $S_1 \Leftrightarrow S_2$ gdy $S_1 \to S_3 \wedge S_2 \to S_4$ ale $S_3 \Leftrightarrow S_4$

Poniższa tabela zawiera wszelkie możliwe sposoby łączenia różnych stanów z wyszczególnieniem tych, które można połączyć:

1	X	-----	-----	-----	-----	-----
2	X	**V**	-----	-----	-----	-----
3	X	(0,2)	(2,0)	-----	-----	-----
4	X	(2,5)	(2,5) (3,4)	(0,5) (3,4)	-----	-----
5	X	(5,2) (6,1)	(2,5)	(0,5)	**V**	-----
6	X	(2,0) (1,6)	(2,0)	**V**	(5,0)	(5,0)
S_n	0	1	2	3	4	5

1	X	-----	-----	-----	-----	-----
2	X	**V**	-----	-----	-----	-----
3	X	X	X	-----	-----	-----
4	X	(2,5)	X	X	-----	-----
5	X	X	**V**	X	**V**	-----
6	X	X	X	**V**	X	X
S_n	0	1	2	3	4	5

Zauważamy, że stan 2 i 5 można połączyć w jeden pod warunkiem, że stany 2 i 5 są połączone. Zatem stany 2 i 5 można połączyć. Efekt tego połączenia pokazuje sąsiednia tabela.
Stany (1,4) można podłączyć pod warunkiem że połączymy stany (2,5). Czyli automat o stanach 0, (1,4), 2, 5, (3,6) jest nieprawidłowy ponieważ nie są podłączone stany (2,5)!!
Połączyliśmy zatem stany 1 z 4 tworząc stan P_1, następnie 2 z 5 tworząc stan P_2. Stan 0 nie może być połączony z żadnym przez niespełnienie pierwszego warunku łączenia stanów. zatem stanowi 0 przypisujemy stan P_0. Okazuje się, że pozostałe stany 6 i 3 można ze sobą połączyć i otrzymujemy nowy stan P_3.

Tworzymy nową tabelę przejść i wyjść dla stanów P_0 – P_3.

P_n	BA 0 0	BA 0 1	BA 1 0	Y
P_0	P_0	P_1	P_1	0
P_1	P_2	P_1	P_1	1
P_2	P_2	P_3	P_3	1
P_3	P_0	P_3	P_3	1

Nowa tabela automatu Moore'a opisująca przejazd kolejowy (po zminimalizowaniu).

Znając tablicę przejść i wyjść automatu minimalnego opisującego nasz przejazd kolejowy, należy zbudować układ realizujący ten automat, tzn. układ o dwóch wejściach A i B oraz jednym wyjściu Y.
Ponieważ ostateczna wersja automatu bazuje na czterech stanach, do fizycznej jego realizacji wystarczą dwa przerzutniki.

$$L = \lceil \log_2(n+1) \rceil = \left\lceil \frac{\ln(n+1)}{\ln 2} \right\rceil$$

gdzie:

L – ilość przerzutników potrzebna do realizacji automatu
n+1 – ilość stanów przyjmowana przez automat
⌈ ⌉ - oznacza zaokrąglenie wyniku do najbliższej liczby całkowitej w kierunku $+\infty$.

Przypisujemy kolejnym stanom automatu (P_0 – P_3.) stany wyjść przerzutników:

	Q_1	Q_0	Y
P_0	0	0	0
P_1	0	1	1
P_2	1	1	1
P_3	1	0	1

Tabela przypisująca kolejnym stanom automatu (P_0 – P_3.) stany wyjść przerzutników Q_0 i Q_1.

D	Q_{n+1}
0	0
1	1

Tablica prawdy przerzutnika D

W następnej kolejności należy znaleźć logikę opisującą wejście (wejścia) przerzutników. Wybieramy np. przerzutnik typu D. Zależność stanu następnego od wejścia informacyjnego D opisuje powyższa tabela.

Tworzymy tabelę Karnaugh'a dla poszczególnych wejść przerzutników:

		BA			
		00	01	11	10
Q_1Q_0	00	00	01	- -	01
	01	11	01	- -	01
	11	11	10	- -	10
	10	00	10	- -	10

Tabela Karnaugh'a opisująca zależność wejść przerzutników D_1D_0 od wejść BA oraz wyjść tychże przerzutników (Q_1Q_0).
UWAGA!!!

Wyjścia przerzutników to nic innego jak stany wewnętrzne automatu $P_0 – P_3$.

Jak tworzyć tabelę opisującą logikę wejść D_1D_0?

Weźmy pod uwagę przypadek, w którym automat znajduje się w stanie P_0 czyli stan wyjść przerzutników wynosi odpowiednio $Q_1 = 0$ i $Q_0 = 0$. Załóżmy, że podajemy na wejścia B i A stany odpowiednio B = 0 i A = 0. Zgodnie z tabelą przejść i wyjść automat powinien zmienić stan w P_0, czyli sam w siebie. Odpowiada to stanom następnym wyjść przerzutników $Q_1 = 0$ i $Q_0 = 0$. Trzeba zadać sobie pytanie: Jakie stany logiczne podać na wejścia D_1 i D_0 przerzutników aby przy stanach poprzednich $Q_1 = 0$ i $Q_0 = 0$ otrzymać nowe stany wyjść: $Q_1 = 0$ i $Q_0 = 0$ (w tym konkretnym przypadku te same)? Zgodnie z zasadą działania przerzutnika D należy podać na wejście $D_1 = 0$ i $D_0 = 0$. Takie rozumowanie należy przeprowadzić dla pozostałych kombinacji stanów wejściowych BA i wyjściowych Q_1Q_0. (Pokażemy dalej, że realizacja tego automatu jest także możliwa na innych przerzutnikach np. RS. Tabele Karnaugh'a są trudniejsze do zapisania.)
Poprzednią tabelę można rozdzielić na dwie niezależne tabele. Jedną opisującą wejście D_0 przerzutnika pierwszego, oraz drugą tabelę opisującą wejście D_1 przerzutnika drugiego.

		BA			
		00	01	11	10
Q_1Q_0	00	0	0	-	0
	01	1	0	-	0
	11	1	1	-	1
	10	0	1	-	1

Tabela Karnaugh'a opisująca zależność wejścia D_1 od stanów BA oraz Q_1Q_0

		BA			
		00	01	11	10
Q_1Q_0	00	0	1	-	1
	01	1	1	-	1
	11	1	0	-	0
	10	0	0	-	0

Tabela Karnaugh'a opisująca zależność wejścia D_0 od stanów BA oraz Q_1Q_0

Po sklejeniu sąsiednich jedynek w dobrze znany sposób, otrzymujemy na podstawie pierwszej tabeli:

$$D_1 = \overline{B}\,\overline{A}Q_0 + AQ_1 + BQ_1$$

oraz na podstawie drugiej tabeli:

$$D_0 = \overline{B}\,\overline{A}Q_0 + A\overline{Q_1} + B\overline{Q_1}$$

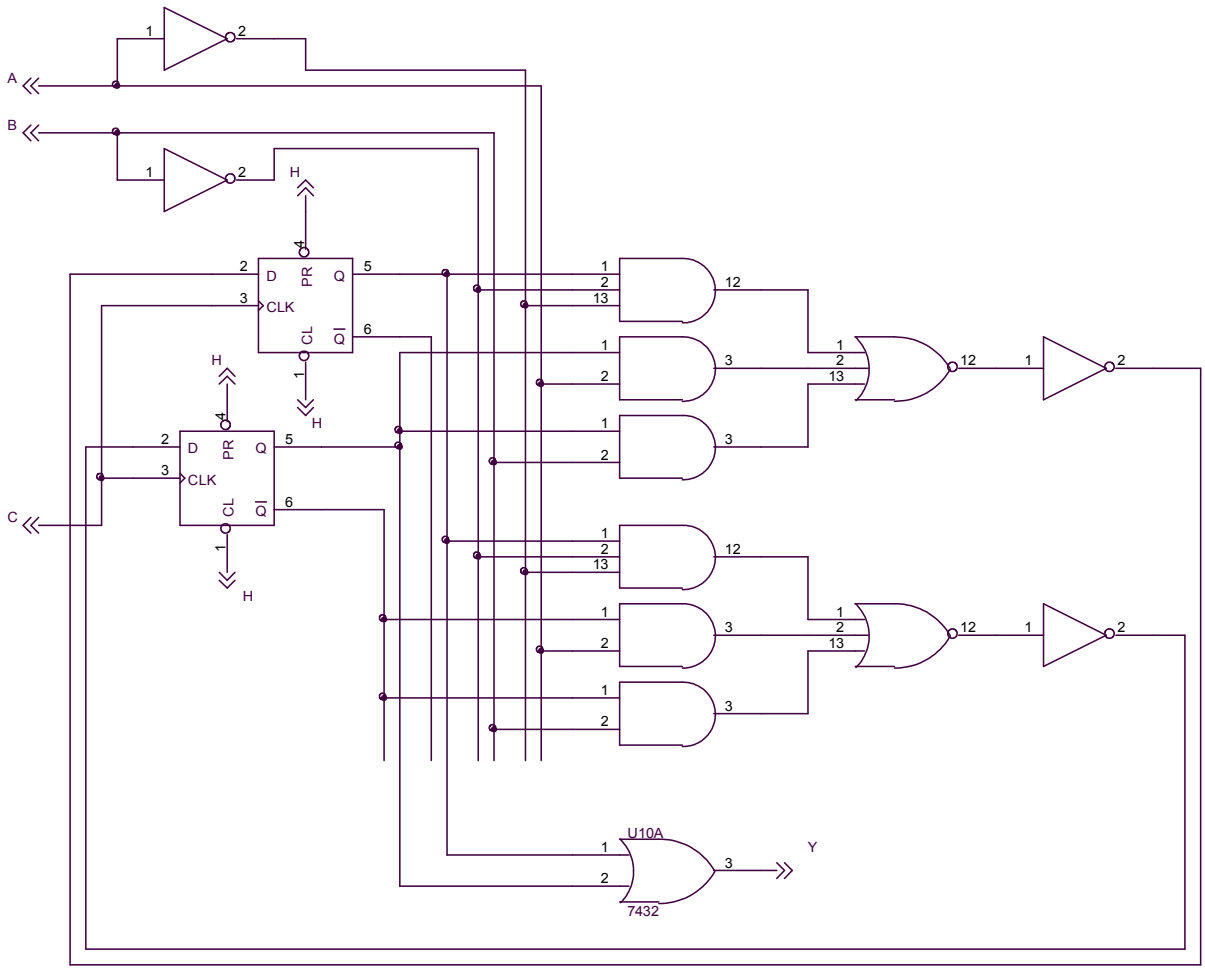
Ostatnim krokiem projektowania logiki opisującej przejazd kolejowy będzie znalezieniu funkcji, która przekształci stany wewnętrzne automatu $P_0 – P_3$ (czyli odpowiednie stany wyjść Q_1 i Q_0 przerzutników) w odpowiedni stan Y podnoszący lub opuszczający rogatkę. Można to zrobić tworząc tabelę Kornough'a opisującą zależność Y od stanów Q_1Q_0 na podstawie tabeli przypisującej kolejnym stanom automatu ($P_0 – P_3$.) stany wyjść przerzutników Q_0 i Q_1. Czyli:

		Q_1	
		0	1
Q_0	0	0	1
	1	1	1

Tabela Karnaugh'a opisująca zależność stanu wyjściowego Y od stanów wyjścia przerzutników Q_1 i Q_0.

Czyli:

$$Y = Q_1 + Q_0$$



Przykładowa realizacja przejazdu kolejowego.

Powyższy automat można także zrealizować na bazie przerzutnika RS. Trzeba tylko znaleźć logikę opisującą wejścia R_0S_0 pierwszego przerzutnika oraz wejścia R_1S_1 drugiego przerzutnika. Stąd prosty wniosek, że musimy znaleźć cztery funkcje, a nie jak w przypadku przerzutnika typu D – dwie funkcje.

Wielce przydatną tabelą będzie tablica prawdy przerzutnika RS:

R	S	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	-

Tablica prawdy dla przerzutnika RS

Na jej podstawie tworzymy tabelę opisującą zależność wejść R_1S_1 drugiego przerzutnika od stanów wejściowych BA oraz stanów poprzednich Q_1Q_0 w jakim były oba przerzutniki (czyli od stanu poprzedniego automatu)

		BA			
		00	01	11	10
Q_1Q_0	00	-0	-0	- -	-0
	01	01	-0	- -	-0
	11	0-	0-	- -	0-
	10	10	0-	- -	0-

Tabela Karnaugh'a opisująca zależność wejść R_1S_1 drugiego przerzutnika od wejść BA oraz poprzedniego stanu wyjść obu przerzutników.

		BA			
		00	01	11	10
Q_1Q_0	00	-0	01	- -	00
	01	0-	0-	- -	0-
	11	0-	10	- -	10
	10	-0	-0	- -	-0

Tabela Karnaugh'a opisująca zależność wejść R_0S_0 pierwszego przerzutnika od wejść BA oraz poprzedniego stanu wyjść obu przerzutników.

Sposób tworzenia tych tablic pozostaje taki sam jak w przypadku automatu budowanego na przerzutnikach typu D. Dla przypomnienia. Rozważmy lewą tabelę opisującą wejścia R_1S_1. W tabeli wpisujemy takie stany R_1S_1, które należy podać aby uzyskać żądaną zmianę stanu wyjściowego przerzutników na pobudzenie BA. Załóżmy, że automat był w stanie P_0 czyli zgodnie z zastosowanym kodowaniem stanów $Q_1 = 0$ i $Q_0 = 0$. Na wejścia BA podajemy odpowiednio B = 0 i A = 1. Zgodnie z tabelą przejść automat powinien przejść w stan P_1 co zgodnie z zastosowanym kodowaniem odpowiada stanom $Q_1 = 0$ i $Q_0 = 1$ na wyjściu przerzutników. Ponieważ wzięliśmy pod uwagę tylko lewą tabelę (czyli wejścia R_1S_1 drugiego przerzutnika), zatem interesuje nas tylko bit Q_1 (bit Q_0 będzie tyczył się prawej tabeli opisującej wejścia R_0S_0 pierwszego przerzutnika). Jakie stany trzeba podać na wejścia R_1S_1, aby pierwszy przerzutnik zmienił stan z 0 na 0. Zgodnie z zasadą działania przerzutnika RS należy podać odpowiednio $R_1 = 1$ $S_1 = 0$ (co powoduje ustawienie przerzutnika w stan $Q_1 = 0$) albo $R_1 = 0$ $S_1 = 0$ (co powoduje zachowanie stanu poprzedniego, czyli $Q_1 = 0$). Wniosek: w tym przypadku zmiana stanu automatu z P_0 na P_1 zależy od S_1 (trzeba podać $S_1 = 0$) ale nie zależy od R_1 – stan dowolny przy minimalizacji. Nieco inna sytuacja wystąpi podczas opisu stanu wejść pierwszego przerzutnika (tabela prawa). Jak już wspomniano, w tym przypadku interesuje nas bit Q_0. Zmiana stanu automatu z P_0 na P_1 jest równoważna ze zmianą bitu Q_0 z 0 na 1. Taki stan można uzyskać przy pomocy przerzutnika RS jedynie podając na jego wejścia $R_0 = 0$ oraz $S_0 = 1$. W analogiczny sposób wypełniamy obie tabele.
W celu bardziej przejrzystego opisu poszczególnych wejść obie tabele można rozbić tak, aby opisywały tylko jedno wejście danego przerzutnika. Bardziej zaawansowani projektanci z pewnością uznają to za stratę czasu.

		BA			
		00	01	11	10
Q_1Q_0	00	-	-	-	-
	01	0	-	-	-
	11	0	0	-	0
	10	1	0	-	0

Tablica Kornough'a opisująca zależność wejścia R_1 od wejść BA oraz poprzednich stanów obu przerzutników

$$R_1 = \overline{B}\,\overline{A}\,\overline{Q_0} + A\overline{Q_1} + B\overline{Q_1}$$

		BA			
		00	01	11	10
Q_1Q_0	00	0	0	-	0
	01	1	0	-	0
	11	-	-	-	-
	10	0	-	-	-

Tablica Kornough'a opisująca zależność wejścia S_1 od wejść BA oraz poprzednich stanów obu przerzutników

$$S_1 = \overline{B}\,\overline{A}Q_0 + AQ_1 + BQ_1$$

		BA			
		00	01	11	10
Q_1Q_0	00	-	0	-	0
	01	0	0	-	0
	11	0	1	-	1
	10	-	-	-	-

Tablica Kornough'a opisująca zależność wejścia R_0 od wejść BA oraz poprzednich stanów obu przerzutników

$$R_0 = \overline{B}\,\overline{A}Q_1 + AQ_1 + BQ_1$$

		BA			
		00	01	11	10
Q_1Q_0	00	0	1	-	0
	01	-	-	-	-
	11	-	0	-	0
	10	0	0	-	0

Tablica Kornough'a opisująca zależność wejścia S_0 od wejść BA oraz poprzednich stanów obu przerzutników

$$S_0 = \overline{B}\,\overline{A}\,\overline{Q_0} + A\overline{Q_1} + B\overline{Q_1}$$

Pomimo zmiany przerzutników z D na RS, sposób kodowani stanów automatu $P_0 - P_3$ przy pomocy stanów wyjściowych przerzutników Q_1Q_0 nie zmienił się. Zatem logika opisująca zależność wyjścia Y od Q_1Q_0 zostaje taka sama jak dla automatu zbudowanego na przerzutnikach D.

$$Y = Q_1 + Q_0$$

Oczywiście można spróbować zaprojektować ten sam automat na przerzutnikach JK. Metodologia postępowania będzie identyczna jak poprzednio.

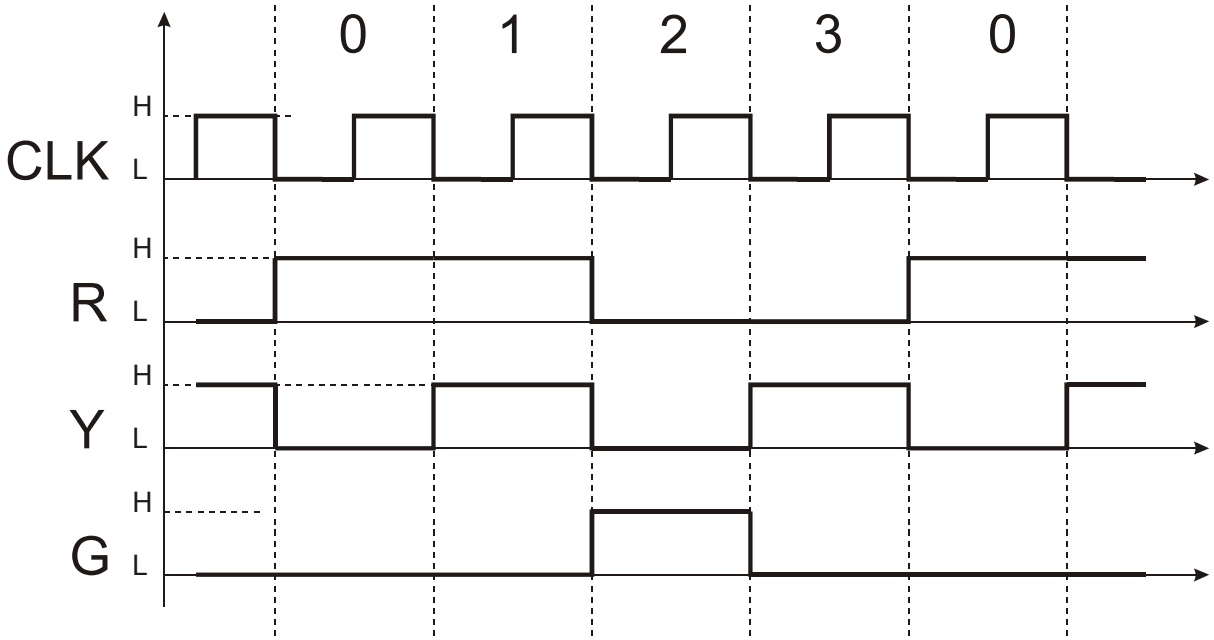
Przykład – sygnalizator drogowy

Zaprojektować sterownik do świateł ulicznych. Na wstępie należy przyjąć pewne założenia.

1. Rozważamy pojedynczy semafor świetlny.
2. Kolejność świateł liczona od góry: czerwone, żółte, zielone.
3. Cykl zmian: czerwone → czerwone i żółte → zielone → żółte → czerwone itd.

4. W projekcie należy uwzględnić, że pojedyncze światło zielone i czerwone świeci się znacznie dłużej niż żółte i żółte z czerwonym.

Przebiegi czasowe:



Rysunek przedstawiający przebiegi czasowe wyjść RYG automatu sterujących sygnalizacją świetlną

R – bit sterujący światłem czerwonym
Y – bit sterujący światłem żółtym
G – bit sterujący światłem zielonym

Przyjęto założenie, że wartość 1 (H) oznacza „świeci", a 0 (L) „nie świeci".
Przełączanie układu dla ujemnego zbocza to tylko założenie. Równie dobrze może to być zbocze narastająca. Wszystko zależy od typu przerzutników zastosowanych do budowy automaty.

UWAGA!!!

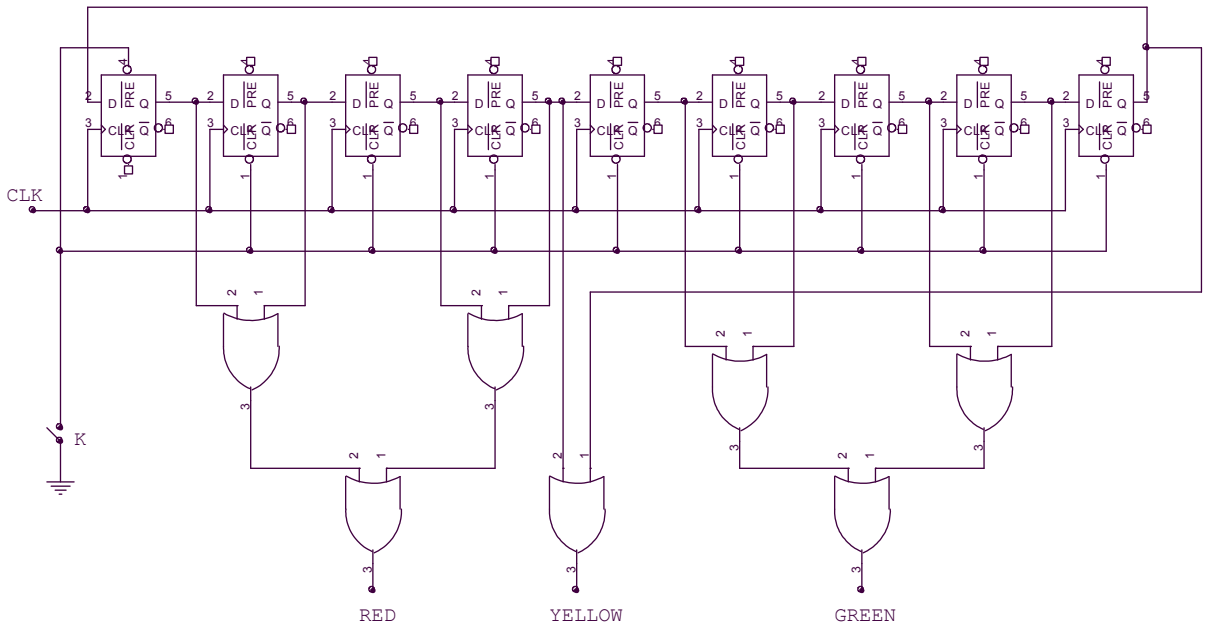
Oś czasu jest nieskalowana i nie stanowi żadnej podstawy do oceny jak długo trwa poszczególny stan. Nie należy sądzić, że stan 0 trwa tyle samo co stan 1. Przeczyłoby to założeniu 4).

STEROWANIA SYGNALIZACJĄ ŚWIETLNĄ – „ONE HOT"

Poniższa realizacja wykorzystuje rejestr pierścieniowy. Jest to tzw. rejestr z krążącą jedynką. Krótkotrwałe zamknięcie przełącznika K powoduje (asynchronicznie) ustawienie jedynki w pierwszym przerzutniku i wyzerowanie pozostałych. Jedynka ta jest przesuwana w rejestrze w takt impulsów zegarowych. Dzięki połączeniu wyjścia ostatniego przerzutnika z wejściem pierwszego proces ten odbywa się cyklicznie. Jeżeli więc do wejść bramki OR podłączymy wyjścia np. czterech przerzutników to na wyjściu OR-a będzie stan wysoki dopóty dopóki przesuwająca się jedynka nie opuści ostatniego z czterech przerzutników. Wyjście OR-a steruje zapalaniem światła. Sterowanie długością świecenia się danego światła może odbywać się poprzez zmianę częstotliwości taktowania lub poprzez przyporządkowanie danemu segmentowi innej liczby przerzutników. Jednak te sposoby sterowania są od siebie zależne. Gdyby zbudować taki układ w praktyce musiałby on zawierać dość dużą ilość

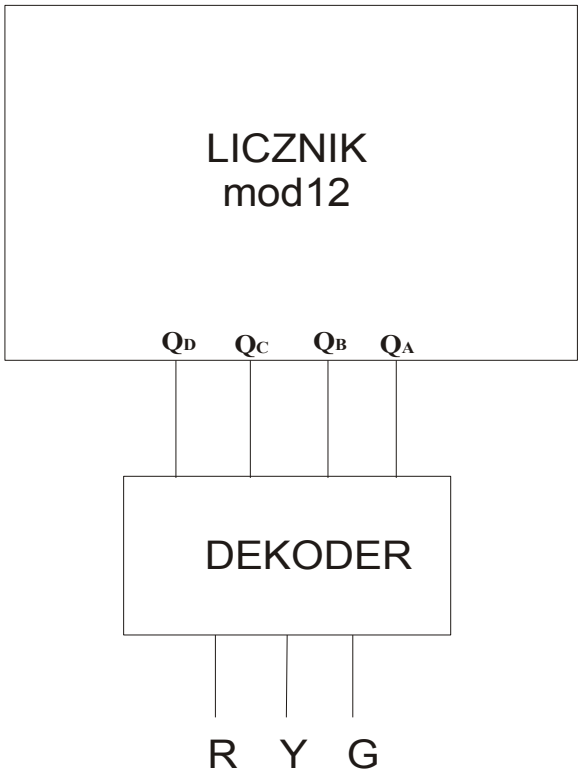
przerzutników (np. gdy chcemy aby czerwone świeciło się przez 40s żółte przez 4s, zielone przez 40s przy okresie taktowania 4s trzeba byłoby użyć 21 przerzutników). Niemniej realizacja poniższa jest funkcjonalnie najprostsza.

Schemat układu:



Rysunek przedstawiający układ typu „one-hot" realizujący funkcję sterownika sygnalizacji świetlnej.

STEROWNIK DO SYGNALIZACJI ŚWIETLNEJ Z UKŁADEM LICZNIKA I DEKODERA STANÓW



Rysunek przedstawiający schemat blokowy sterownika do sygnalizatora ulicznego zbudowanego na bazie dekodera stanów.

Licznik zlicza w kodzie binarnym od 0000 do 1011, następnie jest zerowany i operacja powtarza się. Poszczególnym światłom odpowiadają następujące stany licznika:

- czerwonemu: 0, 1, 7, 8, 9, 10, 11
- żółtemu: 1, 7
- zielonemu: 2, 3, 4, 5, 6

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	-	-	-	-
10	1	1	1	1

Tabela stanów dla światła czerwonego

Logika dla światła czerwonego po zoptymalizowaniu:

$$R = Q_D + \overline{Q}_C \overline{Q}_B$$

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1	0	0
01	0	0	1	0
11	-	-	-	-
10	0	0	0	0

Tabela stanów dla światła żółtego

Logika dla światła żółtego po zoptymalizowaniu:

$$Y = Q_A Q_B Q_C + Q_A \overline{Q}_B \overline{Q}_C \overline{Q}_D$$

Q_BQ_A / Q_DQ_C	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	-	-	-	-
10	0	0	0	0

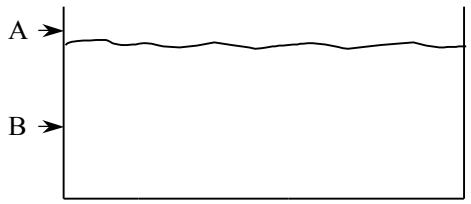
Tabela stanów dla światła zielonego

Logika dla światła zielonego po zoptymalizowaniu:

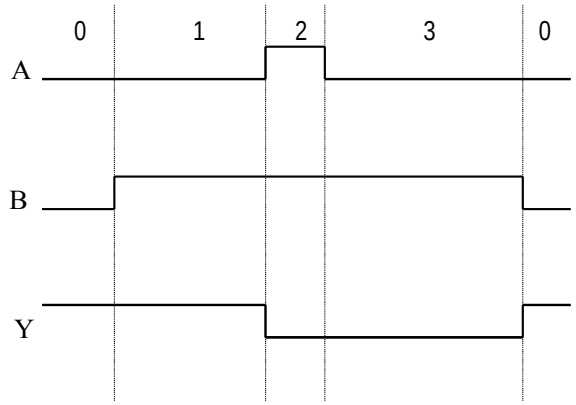
$$G = \overline{Q}_B Q_C + Q_B \overline{Q}_C \overline{Q}_D + \overline{Q}_A Q_B \overline{Q}_D$$

Przykład - sterowanie napełnianiem basenu

W basenie zamontowane są dwa czujniki A i B. Czujnik A sygnalizuje maksymalny poziom wody w basenie, a czujnik B minimalny. Jeżeli basen napełniamy „od zera" poziom wody znajduje się poniżej czujnika B wówczas stany wejściowe wymuszane przez czujniki są A = 0 i B = 0. Wyjście Y sterujące pompą jest wówczas w stanie wysokim. Pompa pompuje wodę do basenu. Po przekroczeniu poziomu czujnika B sygnał wejściowy B zmienia stan na wysoki, ale Y = 1 i pompa dalej pompuje. Stan ten trwa aż do momentu osiągnięcia poziomu maksymalnego. Czujnik A zmienia stan na wysoki, wyjście Y = 0, pompa przestaje pompować. Ubytek wody następuje z przyczyn naturalnych (parowanie, wycieki). Gdy poziom wody osiągnie minimum pompa włącza się i proces powtarza się.



Rysunek poglądowy przedstawiający umieszczenie czujników w basenie



Rysunek przedstawiający przebiegi czasowe w układzie.

Tabela dla automatu Moore'a

S_n	BA 00	BA 01	BA 11	BA 10	Y
0	0	-	-	1	1
1	-	-	2	1	1
2	-	-	2	3	0
3	0	-	-	3	0

Tabela przejść i wyjść dla automatu realizującego funkcję sterowania basenem.

0				
1	**V**			
2	X	X		
3	X	X	**V**	
S_n	0	1	2	3

Tabela do optymalizacji liczby stanów

Legenda:

V – można połączyć stany
X – połączenie stanów niemożliwe

Można więc połączyć stany 0 z 1 oraz 2 z 3.

Wprowadzamy nowe oznaczenia połączonych stanów:
$P_0 = (0, 1)$ i $P_1 = (2, 3)$

Do realizacji układu użyjemy przerzutnika synchronicznego D:

P_n	Q
P_0	0
P_1	1

Tabela przypisująca stany wyjścia Q przerzutnika odpowiednim stanom automatu P_n.
Q – wyjście przerzutnika D

P_n	BA 00	BA 01	BA 11	BA 10
P_0	P_0	-	P_1	P_0
P_1	P_0	-	P_1	P_1

Tabela przejść po zoptymalizowaniu

Układamy logikę dla przerzutnika D (na podstawie poprzedniej tabeli P_0 – 0, P_1 - 1)

Q	BA 00	BA 01	BA 11	BA 10
0	0	-	1	0
1	0	-	1	1

Tabela Karnaugha uzależniająca stan wejścia D przerzutnika od stanu wejść BA oraz stanu poprzedniego Q.

Stąd: D = A + BQ

Stan wyjścia dla automatu Moore'a zależy tylko od stanów przerzutników i w naszym przypadku łatwo wyprowadzić że:

$$Y = \overline{Q}$$

UWAGA!!!
Częstotliwość sygnału taktującego CLK powinna być znacznie większa niż częstotliwość zmian poziomów na wejściach A i B.