

Quizlet: <https://quizlet.com/770121640/flashcards>

1. **Multiplexer** jest układem przetwarzającym równoległy kod dwójkowy na kod szeregowy.

Jest to przełącznik sterowany kodowo za pomocą dekodera adresu. Układ ten pozwala na zrealizowanie tzw. multipleksowego systemu transmisji danych.

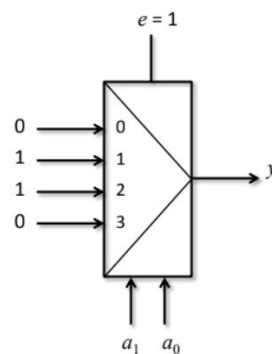
( $x_0, x_1 \dots x_n$  - wejścia informacyjne)

( $a_0, a_1 \dots a_m$  - wejścia adresowe)

posiada:

- m wejść adresowych
- $n = 2^m$  wejść informacyjnych
- jedno wyjście informacyjne
- e (enable) - wejście zezwolenia

### Multiplexer (MUX)



Multiplexer jako **przełącznik** realizuje funkcję  $y = a_1 a_0 d + a_1 a_0 \bar{d}$ .

Stan na wyjściu  $y$  będzie taki, jak stan wejścia  $d$  wskazywanego (wybranego) stanem na wejściach adresowych  $a_k$ .

Multiplexery łączy się kaskadowo w celu eliminacji tworzenia multiplexerów o dużej liczbie wejść adresowych. (dlaczego?)

2. **Demultiplexer**, układem przetwarzającym kod szeregowy na równoległy. (chyba dwójkowy)

Jest to przełącznik sterowany kodowo za pomocą dekodera adresu. Układ ten pozwala na zrealizowanie tzw. multipleksowego systemu transmisji danych.

( $y_0, y_1 \dots y_n$  - wyjścia informacyjne)

( $a_0, a_1 \dots a_m$  - wejścia adresowe)

posiada:

- m wejść adresowych
- jedno wejście informacyjne
- $n = 2^m$  wyjść informacyjnych
- e (enable) - wejście zezwolenia

3. **Kody refleksyjne** - Możliwość powstawania błędów niejednoczesnej zmiany na pozycjach kodu jest wyeliminowana w kodach, w których nie więcej niż jeden bit zmienia swoją wartość przy przejściu między kolejnymi zakodowanymi wartościami.

Przykładem kodu refleksyjnego jest kod Graya.

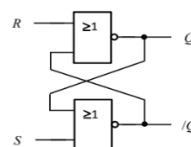
4. **Enkoder** - kod wejściowy to „1 z n”, wyjściowy dowolny dwójkowy. głównie do wprowadzania danych w postaci liczb dziesiętnych (kod wyjściowy to najczęściej 8421)
5. **Dekoder** -wejściowy dowolny dwójkowy, kod wyjściowy to „1 z n”.
6. **Transkoder** -zmienia z innego niż „1 z n” na inny niż „1 z n.” np z NKB na kod Gray'a
7. **Stan stabilny** - stan jednoznacznie określony - jak w układach kombinacyjnych
8. **Stan astabilny** - wystąpienie sprzeczności  $Q = \neg Q$
9. **Stan bistabilny** - stan 0 lub 1 zależny od stanu poprzedniego (stanu zapamiętanego, wewnętrznego)  
Występowanie STANU BISTABILNEGO charakteryzuje układ sekwencyjny. Jeżeli w układzie występuje STAN BISTABILNY, to układ nie jest już kombinacyjny.

## 10. Asynchroniczny przerzutnik RS

### Asynchroniczny przerzutnik RS

W przypadku przerzutnika RS wszystkie kombinacje stanów początkowych  $Q$  i  $\neg Q$  oraz pobudzeń  $R$ ,  $S$  (badane tzw. lawinę zmian stanów), prowadzą do osiągnięcia stanów stabilnych bądź bistabilnych, bez niebezpieczeństwa osiągnięcia stanów astabilnych.

$S$	$R$	$Q$	$\neg Q$	
0	0	$Q$	$\neg Q$	STAN BISTABILNY
0	1	0	1	STANY STABILNE
1	0	1	0	
1	1	0	0	

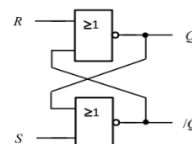


### Asynchroniczny przerzutnik RS

Elementarny i w pełni funkcjonalny układ sekwencyjny zdolny do pamiętania jednego bitu informacji

$S$	$R$	$Q$	$\neg Q$
0	0	$Q$	$\neg Q$
0	1	0	1
1	0	1	0
1	1	0	0

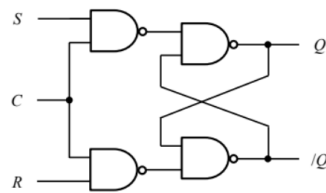
„pamiętaj” stan  
reset (ustaw  $Q = 0$ )  
set (ustaw  $Q = 1$ )  
stan zabroniony – pobudzenie sprzeczne



## 11. Synchroniczny przerzutnik RS

Synchroniczny przerzutnik RS

$/S$	$/R$	$Q_{t+1}$
0	0	$Q_t$
0	1	0
1	0	1
1	1	x



Dla  $C = 0$  zmiany sygnałów  $R$  i  $S$  nie mają wpływu na stan  $Q$ .

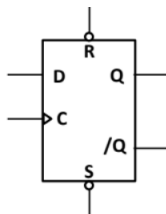
Dla  $C = 1$  zmiany zachodzą zgodnie z powyższą tabelą.

Zmiana sygnału  $C$  z 1 na 0 powoduje zatrzaśnięcie stanu wyjścia – układ typu *latch*.

## 12. Przerzutnik D

Przerzutnik D

$D$	$Q_t$	$Q_{t+1}$
0	0	0
0	1	0
1	0	1
1	1	1



Wyjście  $Q$  przyjmuje wartość  $D$ .

Przerzutnik posiada dwa stany.

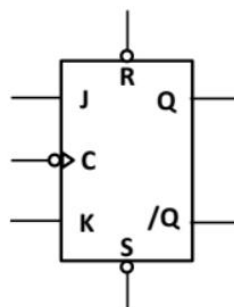
Zmiana stanu następuje (tu:) ze zboczem narastającym sygnału  $C$ .

Przerzutnik posiada asynchroniczne wejścia: zerujące ( $Q = 0$ ), RESET i ustawiające ( $Q = 1$ ), SET.

## 13. Przerzutnik JK

Przerzutnik JK

$J$	$K$	$Q_{t+1}$
0	0	$Q_t$
0	1	0
1	0	1
1	1	$/Q_t$



$$Q_{t+1} = J/Q_t + /K Q_t$$

Przerzutnik posiada dwa stany.

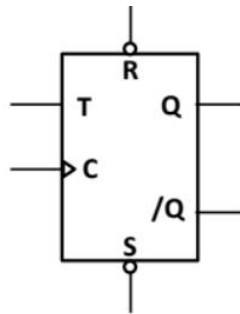
Zmiana stanu następuje (tu:) ze zboczem opadającym sygnału  $C$ .

Przerzutnik posiada asynchroniczne wejścia: zerujące ( $Q = 0$ ), RESET i ustawiające ( $Q = 1$ ), SET.

#### 14. Przerzutnik T (Trigger)

##### Przerzutnik T (Trigger)

$D$	$Q_t$	$Q_{t+1}$
0	0	0
0	1	1
1	0	1
1	1	0



$$Q_{t+1} = J/Q_t + /K Q_t$$

Przerzutnik posiada dwa stany.

Zmiana stanu następuje (tu:) ze zboczem narastającym sygnału  $C$ .

Przerzutnik posiada asynchroniczne wejścia: zerujące ( $Q = 0$ ), RESET i ustawiające ( $Q = 1$ ), SET.

#### 15. Formalny opis układów sekwencyjnych

$X = \langle x_0, x_1, \dots, x_{n-1} \rangle$  wektor  $n$  binarnych sygnałów WE

$Y = \langle y_0, y_1, \dots, y_{m-1} \rangle$  wektor  $m$  binarnych sygnałów WY

$X = \{ X_1, X_2, \dots, X_N \}$  zbiór stanów (słów) WE ( $N^{2n}$ ) - alfabet wejściowy

$Y = \{ Y_1, Y_2, \dots, Y_M \}$  zbiór stanów (słów) WY ( $M^{2m}$ ) - alfabet wyjściowy

$Q = \{ Q_1, Q_2, \dots, Q_N \}$  zbiór stanów wewnętrznych - alfabet wewnętrzny

Stan wewnętrzny  $Q$  reprezentowany jest wektorem bitowym złożonym z sygnałów binarnych  $Q_i$

symbolizujących stan elementarnych układów pamięci (przerzutników).

#### 16. Automat skończony

Automat skończony składa się ze skończonego zbioru stanów i zbioru przejść (ruchów/działań/operatorów), ze stanu do stanu zachodzących przy różnych symbolach wejściowych wybranych z pewnego alfabetu  $E$  ( $\leftarrow$  znak sumy).

Dla każdego symbolu wejściowego istnieje dokładnie jedno przejście odpowiadające temu symbolowi. Przejście to może prowadzić do tego samego stanu.

Wyróżniony jest stan początkowy, oznaczony  $q_0$ .

Z każdym automatem skończonym związany jest diagram przejść, będący grafem skierowanym, w którym wierzchołki odpowiadają stanom automatu, natomiast krawędzie odpowiadają przejściom pomiędzy stanami, jeśli takie przejście istnieje.

Stany automatu oznaczamy symbolami  $q_i$ , etykiety krawędzi symbolami  $z_k$ .

Automat skończony akceptuje łańcuch  $s$ , jeżeli ciąg przejść odpowiadający symbolom łańcucha  $s$  prowadzi od stanu początkowego  $q_0$  do dowolnego stanu końcowego.

Automat skończony przedstawiamy formalnie jako piątkę uporządkowaną

$(Q, \Sigma, \delta, q_0, F)$ , gdzie:

$Q$  – skończony zbiór stanów,

$\Sigma$  – skończony alfabet wejściowy,

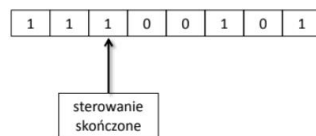
$\delta$  – funkcja przejść odwzorowująca  $Q \times \Sigma$  w  $Q$ ,

$q_0$  – stan początkowy,

$F$  – zbiór stanów końcowych  $F \subseteq Q$ .

Automat skończony można wyobrazić sobie jako układ skończonej liczby stanów

(sterowanie skończone), który znajduje się w pewnym stanie należącym do  $Q$  i czyta ciąg symboli z  $\Sigma$  zapisany na taśmie.



W automacie skończonym sygnał wyjściowy przyjmuje wartość zero bądź jeden (akceptuje/nie akceptuje). Nie zawsze wystarczająco. Stąd rozwiązania z wyborem sygnału wyjściowego z pewnego alfabetu.

wyjście związane ze stanem - Automat Moore'a

wyjście związane z przejściem - Automat Mealy'ego

## 17. Automat Moore'a

Stany wewnętrzne zależą od stanów WE. Zatem,

(znak sigmy):  $Q \times X \rightarrow Q$  funkcja przejść oraz

(znak lambdy):  $Q \rightarrow Y$  funkcja wyjść

## 18. Automat Mealy'ego

Automat Mealy'ego <sup>(1)</sup>

Automat Mealy'ego to (formalnie) szóstka uporządkowana

$(Q, \Sigma, \delta, \lambda, q_0, \Lambda)$ , gdzie:

$Q$  – skończony zbiór stanów,

$\Sigma$  – skończony alfabet wejściowy,

$\Lambda$  – skończony alfabet wyjściowy,

$\lambda$  – odwzorowanie  $Q \times \Sigma$  w  $\Lambda$ ,

$\delta$  – funkcja przejść odwzorowująca  $Q \times \Sigma$  w  $Q$ ,

$q_0$  – stan początkowy.

Automat Mealy'ego <sup>(2)</sup>

$\lambda$  – odwzorowanie  $Q \times \Sigma$  w  $\Lambda$  oznacza, że  $\delta(q_0, z)$  podaje wejście związane z przejściem

ze stanu  $q$  przy wejściu  $z$ .

Wyjściem  $M$  odpowiadającym wejściu  $z_1, z_2, \dots, z_n, n \geq 0$ , jest  $\lambda(q_0, z_1)\lambda(q_1, z_2)\dots\lambda(q_{n-1}, z_n)$ ,

gdzie  $q_0, q_1, \dots, q_n$  jest ciągiem stanów, takim że  $\delta(q_{i-1}, z_i) = q_i$ , dla  $1 \leq i \leq n$

Powyższy ciąg ma długość  $n$ , inaczej niż w automacie Moore'a ( $n + 1$ )

## 19. Automat Moore'a vs Automat Mealy'ego

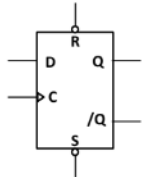
Dla każdego automatu Moore'a istnieje równoważny mu (w znaczeniu odpowiedzi), automat Mealy'go i na odwrót.

Zaletą automatu Moore'a jest odseparowanie wyjść od zmian sygnałów - być może przypadkowych.

## 20. Grafy przerzutników

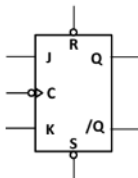
Przerzutnik D

D	$Q_t$	$Q_{t+1}$
0	0	0
0	1	0
1	0	1
1	1	1



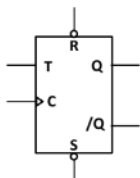
Przerzutnik JK

J	K	$Q_{t+1}$
0	0	$Q_t$
0	1	0
1	0	1
1	1	$\neg Q_t$



Przerzutnik T (Trigger)

D	$Q_t$	$Q_{t+1}$
0	0	0
0	1	1
1	0	1
1	1	0



## 21. Etapy projektowania układu kombinacyjnego

1. Opis działania układu.
2. Ustalenie sygnałów WE i WY (jeśli nie podano wprost w opisie).
3. Określenie funkcji przełączającej [tabela prawdy i minimalizacja].
4. Znalezienie minimalnej postaci funkcji (minimalnej zgodnie z celem minimalizacji).
5. Stworzenie schematu układu.
6. Ocena kosztu realizacji, ewentualnie korekta układu.

## 22. Metody opisu układu

1. Opis słowny
2. Tablica prawdy
3. Zbiór jedynek / zer funkcji  $F$  ( $F^1 / F^0$ )
4. Funkcji boolowska - postać kanoniczna iloczynu / sumy bądź postać zminimalizowana
5. Schemat układu

## 23. Metody minimalizacji wyrażeń logicznych

1. Przekształcenia w oparciu o prawa algebry Boole'a
2. Metoda siatek Karnaugh'a
3. Metoda Quine'a - McCluskeya
4. Metoda Kazakowa
5. Metoda Tablic Niezgodności
6. Metoda ESPRESSO

## 24. Metoda Quina'a – McCluskeya

1. Funkcja  $F$  opisujemy przy pomocy dwóch zbiorów:  $F^1$  i  $F^0$ .

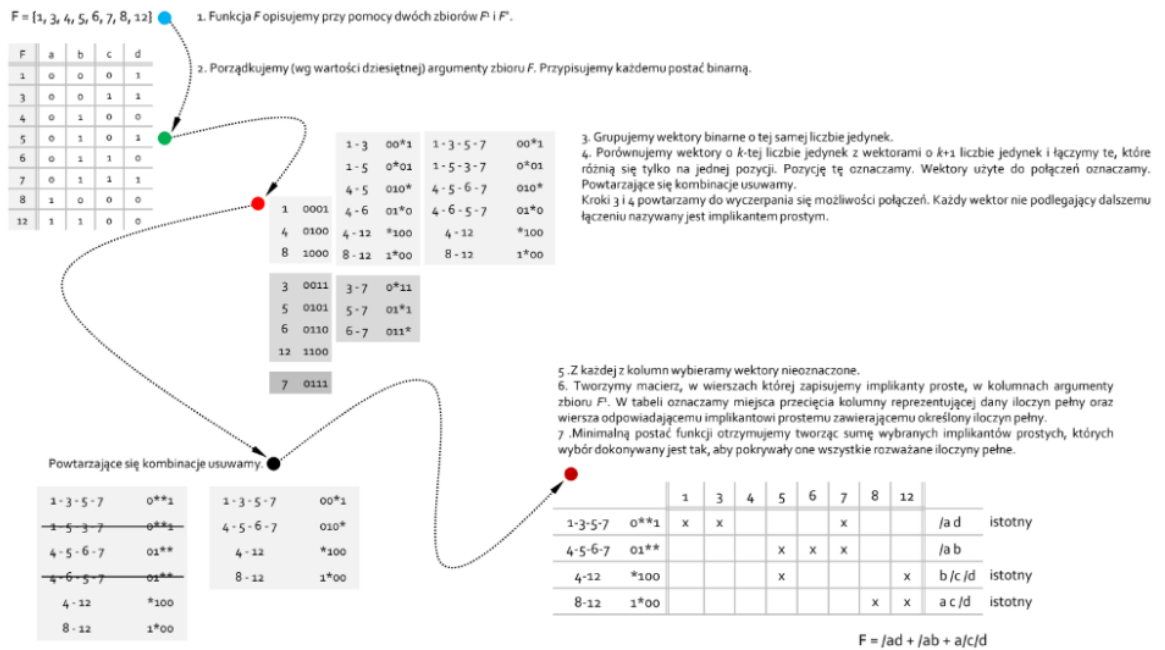
$F^1$  to zbiór argumentów, dla których  $F$  przyjmuje wartość 1.

$F^0$  to zbiór argumentów, dla których  $F$  przyjmuje wartość nieokreśloną. Łączymy oba zbiory w zbiór  $F$ .

2. Porządkujemy (wg wartości dziesiętnej) argumenty zbioru  $F$ . Przypisujemy każdemu postać binarną.
3. Grupujemy wektory binarne o tej samej liczbie jedynek.
4. Porównujemy wektory o  $k$ -tej liczbie jedynek z wektorami o  $k+1$  liczbie jedynek i łączymy te, które różnią się tylko na jednej pozycji. Pozycję tę oznaczamy. Wektory użyte do połączeń oznaczamy. Powtarzające się kombinacje usuwamy.

Kroki 3 i 4 powtarzamy do wyczerpania się możliwości połączeń. Każdy wektor nie podlegający dalszemu łączeniu nazywany jest implikantem prostym.

5. Wybieramy wektory nieoznaczone (implikanty proste).
6. Tworzymy macierz, w wierszach której zapisujemy implikanty proste, w kolumnach argumenty zbioru  $F$ . W tabeli oznaczamy miejsca przecięcia kolumny reprezentującej dany iloczyn pełny oraz wiersza odpowiadającemu implikantowi prostemu zawierającemu określony iloczyn pełny.
7. Minimalną postać funkcji otrzymujemy tworząc sumę wybranych implikantów prostych, których wybór dokonywany jest tak, aby pokrywały one wszystkie rozważane iloczyny pełne.



## 25. Metoda Karnaugh'a (raczej wiadomo)

## 26. Metoda Karnaugh vs metoda Quine - McCluskey'a

- Obie metody dają identyczne wyniki (?) i mają identyczny przebieg (?)
- Metoda Karnaugh'a ma czytelną formę graficzną
- Metoda QMC też ma czytelną formę, tyle że nie-graficzną
- W obu wyszukuje się zależności relaksujących elementy wyrażenia boolowskiego
- Metoda Karnaugh'a jest łatwa (?) do sześciu zmiennych
- Metoda QMC - nie ma ograniczeń (?)
- Funkcje wielu zmiennych redukuje jednak się metodami heurystycznymi, np. metodą ESPRESSO.

## 27. Złożoność minimalizacji

Problem minimalizacji jest problemem NP-trudnym, którego czas realizacji rośnie wykładniczo.

Górne ograniczenie dla  $n$  zmiennych to  $3^n/n$ .

## 28. Automat skończony

Automat skończony jest modelem matematycznym systemu dyskretnego, działającego w dyskretnych chwilach czasu. Jego działanie jest określone na skończonych zbiorach sygnałów wejściowych, stanów wewnętrznych i sygnałów wyjściowych.

Automat skończony jest przetwornikiem ciągu symboli wejściowych na ciąg symboli wyjściowych.

Wystąpienie określonych symboli na wyjściu automatu zależne jest do zestawu symboli na wejściu oraz od stanu wewnętrznego automatu.

Stan wewnętrzny związany jest z istnieniem pamięci.

Pamięć automatu jest tym większa, im więcej ma on stanów wewnętrznych.



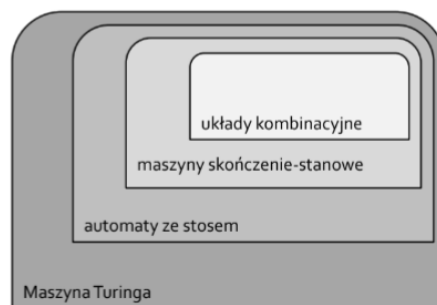
## 29. Teoria automatów

Teoria automatów (*ang. automata theory*) rozważa modele automatów i maszyn abstrakcyjnych oraz zajmuje się rozwiązywaniem problemów obliczeniowych z ich użyciem.

Teoria automatów jest ściśle związana z językami formalnymi.

Automat jest skończoną reprezentacją języka formalnego – który może być nieskończony.

Najbardziej ogólnym modelem automatu jest Maszyna Turinga.



## 30. Oznaczenia w automatach

Podstawowymi modelami automatów rozważanymi przez nas są: automaty Moore'a i Mealy'ego.

$A = \langle \Sigma, \Delta, Q, \delta, \lambda, q_0 \rangle$ , gdzie:

$\Sigma$  – niepusty zbiór symboli wejściowych

$\Delta$  – niepusty zbiór symboli wyjściowych

$Q$  – niepusty zbiór symboli stanów wewnętrznych

$\delta: Q \times \Sigma \rightarrow Q$  – funkcja przejść (przejścia)

$\lambda: Q \rightarrow \Delta$  – funkcja wyjść (wyjścia)

$\lambda: Q \times \Sigma \rightarrow \Delta$  – funkcja wyjść (wyjścia)

$q_0$  – stan początkowy,  $q_0 \in Q$

Przyjmując  $t$  jako chwilę bieżącą oraz  $t+1$  jako chwilę następną:

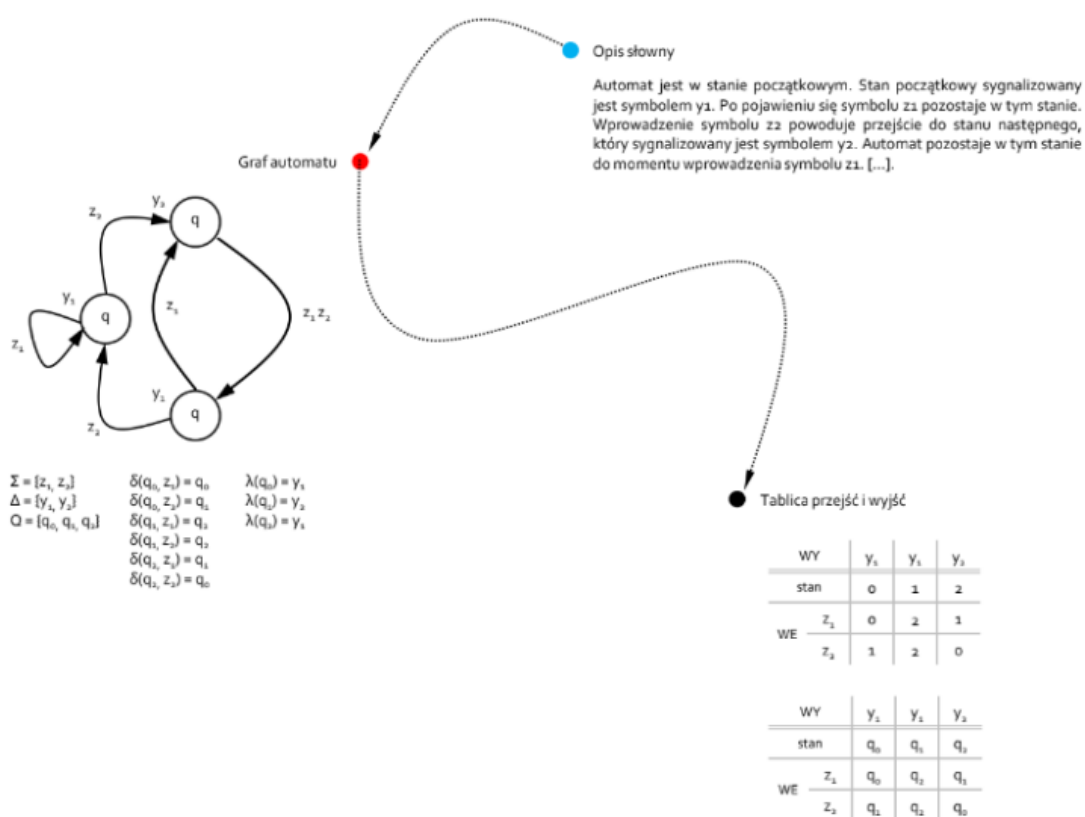
$$q_{t+1} = \delta(q_t, z_t), \quad z_t = \lambda(q_t)$$

### 31. Synteza abstrakcyjna

Synteza abstrakcyjna automatu to określenie opisu formalnego automatu, na podstawie którego można zbudować tabele przejść i wyjść automatu. Synteza sprowadza się do przejścia od algorytmu działania do grafu przejść automatu.

Etapy syntezy:

- opis działania automatu - algorytm słowny,
- przedstawienie algorytmu słownego w postaci wyrażeń regularnych bądź grafu automatu,
- określenie tablicy przejść i wyjść,
- określenie grafu przejść, jeśli znane jest wyrażenie regularne, a nie graf przejść.

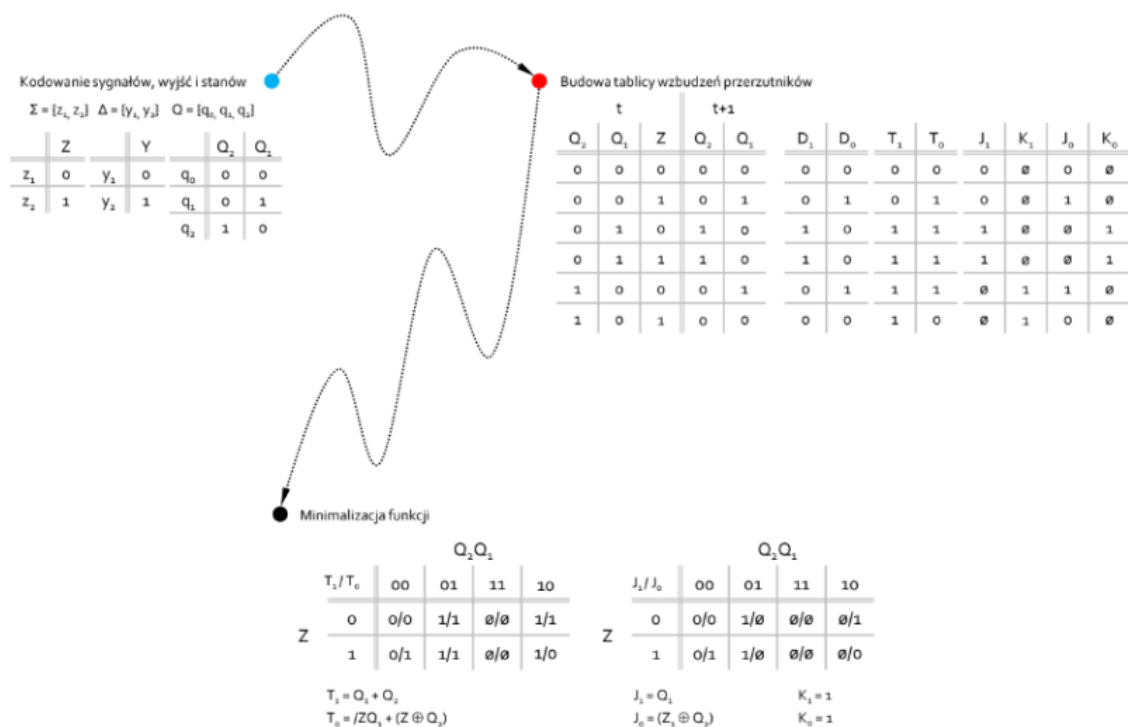


### 32. Synteza strukturalna

Synteza strukturalna automatu to stworzenie schematu układu elektronicznego odzwierciedlającego działanie automatu.

Etapy syntezy:

- kodowanie sygnałów, wejść i wyjść,
- budowa tablicy wzbudzeń przerzutników,
- określenie funkcji wzbudzeń przerzutników,
- określenie funkcji wyjścia,
- stworzenie schematu logicznego układu.



33. **Alfabet** - skończony, niepusty zbiór symboli - A.

34. **Słowo nad A** to dowolny skończony ciąg elementów zbioru A

35. **Język nad alfabetem A** to dowolny podzbiór zbioru  $A^*$   
 . ( $A^*$  - zbiór wszystkich słów nad A)

36. **Synteza strukturalna**

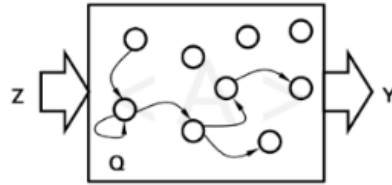
#### Definicja 4

Wyrażeniem regularnym nad alfabetem  $\Sigma$  nazywamy ciąg znaków składający się z symboli:  $\emptyset, \epsilon, +, *, \cdot, (, )$  ( oraz symboli  $a_i$  alfabetu  $\Sigma$  w następującej postaci:

- $\emptyset, \epsilon$  są wyrażeniami regularnymi,
- wszystkie symbole  $a_i \in \Sigma$  są wyrażeniami regularnymi,
- jeśli  $e_1, e_2$  są wyrażeniami regularnymi, to są nimi również:
  - $e_1^*$  (domknięcie Kleene'ego)
  - $e_1 e_2$  (konkatenacja)
  - $e_1 + e_2$  (suma)
  - $(e_1)$  (grupowanie)
- wszystkie wyrażenia regularne są postaci opisanej w punktach 1 – 3.

Każde wyrażenie regularne definiuje pewien język formalny.

Każdy język definiowany przez wyrażenie regularne jest regularny.



Z – alfabet wejściowy, Q – zbiór stanów wewnętrznych, Y – alfabet wyjściowy

Automat <A> akceptuje słowa należące do języka regularnego.

Język regularny jako zbiór słów jest reprezentowany przez wyrażenie regularne.

Alfabet

$$W = \{w_1, w_2, \dots, w_i, \dots, w_n\}$$

Słowa nad alfabetem W

$$w_1 w_2 w_3 w_1 \quad w_4 w_9 w_1 \quad w_1 w_3 w_3 w_3 w_5 \quad \dots$$

Zbiór wszystkich możliwych słów jest zbiorem nieskończonym  $W^*$ .

$$W^* = \{w_1 w_2 w_3 w_1, w_4 w_9 w_1, w_1 w_3 w_3 w_3 w_5, \dots\}$$

Na zbiorze  $W^*$  można określić rodzinę zbiorów  $S^*$ .

$$S^* = \{S_1, S_2, \dots, S_i, \dots, S_n\}$$

Na słowach  $S_i \in S^*$  można wykonywać określone operacje: sumy, konkatencji oraz iteracji.

$$\begin{array}{l} S_1 = z_1 z_1 z_2 + z_2 z_1 \quad | y_2 \\ S_2 = z_1 z_2 + z_2 z_1 \quad | y_2 \\ \hline S_3 = (S_1 + S_2) \quad | y_0 = \epsilon \end{array}$$

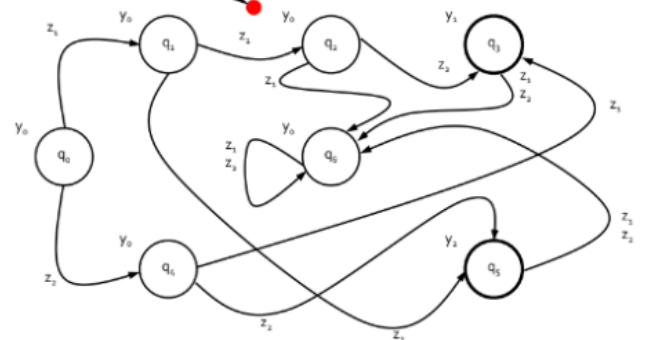
$$S_2 = | z_1 | z_1 | + | z_2 | z_1 |$$

$$0 \quad 1 \quad 6 \quad 0 \quad 4 \quad 7$$

$$S_1 = | z_1 | z_1 | z_2 | + | z_2 | z_1 |$$

$$0 \quad 1 \quad 2 \quad 3 \quad 0 \quad 4 \quad 5$$

WY	stan	Y <sub>0</sub>	Y <sub>0</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>0</sub>	Y <sub>2</sub>	Y <sub>0</sub>
		0	1	2	3	4	5	6
WE	z <sub>1</sub>	1	2	*	*	3	*	*
	z <sub>2</sub>	4	5	3	*	5	*	*



37. **Sygnal cyfrowy** to sygnał przyjmujący wartości ze skończonego (dyskretnego, przeliczalnego) zbioru. Sygnał o dwuelementowym zbiorze wartości to sygnał binarny.

38. **Jakie są podstawowe parametry w technologii TTL?** (zakresy napięć dla "0" i "1", czas propagacji)

Logiczne "0": **0-0.8V**

Logiczne "1": **2.4-5V**

Czas propagacji: **1.5-3ns**

39. **Czy wektor bitowy jest jednoznaczny?**

Nie, znaczenie wektora bitowego jest umowne, znaczenie jest zależne od potrzeb projektu

40. **Jakie są dwie podstawowe operacje w algebrze Boole'a?**

Operacja "+" czyli alternatywa/suma logiczna

Operacja "{znak mnożenia}" (nie używamy symbolu "\*" gdyż oznacza iterację) czyli koniunkcja/iloczyn logiczny

41. **Jakie są elementy neutralne w algebrze Boole'a?**

0 jest elementem neutralnym sumy a 1 elementem neutralnym iloczynu

42. **Przedstaw 4 aksjomaty Algebry Boole'a**

(A) przemienność

$$a + b = b + a \quad a \cdot b = b \cdot a$$

(B) rozdzielność

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$a + (b \cdot c) = (a + b) \cdot (a + c) !!!$$

(C) elementy neutralne działań

$$a + 0 = a \quad a \cdot 1 = a$$

(D) element przeciwny

$$\forall a \in P \exists /a \in P$$

$$a + /a = 1$$

$$a \cdot /a = 0$$

W b błąd – powinno być  $a + (bc) = (a+b)(a+c)$

43. **Podaj Prawo de Morgana**

$$/(ab) = /a + /b \quad /(a+b) = /a \cdot /b$$

44. **Czym jest funkcja boolowska (n argumentowa)?**

Funkcja boolowska to odwzorowanie  $f: X_n \rightarrow Y$  lub  $f: X_n \rightarrow \{0,1\}$ , jest równoważna układowi kombinacyjnemu o n wejściach i jednym wyjściu

45. **Podaj zbiory jedynek i zer dla funkcji alternatywy**

zbiór zer binarny- $f_0=[00]$ ;

zbiór jedynek binarny- $f_1=[01,10,11]$ ;

zbiór zer- $f_0\{0\}$ ;

zbiór jedynek- $f_1=\{1,2,3\}$

46. **Podaj kanoniczną postać sumy funkcji boolowskiej**

$f(x_1, x_2, \dots, x_n) = x_1 x_2 \dots x_n f(1, 1, \dots, 1) + x_1 x_2 \dots /x_n f(1, 1, \dots, 0) + \dots + x_1 /x_2 \dots /x_n f(0, 0, \dots, 0)$

47. **Podaj kanoniczną postać iloczynu funkcji boolowskiej**

$f(x_1, x_2, \dots, x_n) = [x_1 + x_2 + \dots + x_n + f(0, 0, \dots, 0)][x_1 + x_2 + \dots + /x_n + f(0, 0, \dots, 1)] \dots [/x_1 + /x_2 + \dots + /x_n + f(1, 1, \dots, 1)]$

48. **Czym jest zbiór funkcjonalnie pełny?**

Zbiór funkcjonalnie pełny to zbiór funkcji boolowskich za pomocą których, wpółłączeniu ze stałymi 0 i 1, jesteśmy w stanie przedstawić każdą inną funkcję boolowską. Aby wykazać że zbiór jest funkcjonalnie pełny wystarczy za jego pomocą przedstawić trzy funkcje {NOT, AND, OR}

49. **Czym może być minimum środków przy realizacji układu?**

- minimalna liczba bramek (w ogóle),
- minimalna liczba bramek określonego typu,
- wykorzystanie bramek określonego typu,
- eliminacja bramek wielowejściowych,
- minimalizacja liczby użytych układów scalonych,
- użycie układów o określonej strukturze wewnętrznej

50. **Czym są miejsca podstawowe i przedpodstawowe?**

Miejsca podstawowe to miejsca, na lewo od których znajdują się litera oraz miejsce początkowa. Miejsce przedpodstawowe to miejsce po którego prawej stronie znajduje się litera

51. **Jak formalnie definiujemy automat?**

Jako piątkę uporządkowaną

$$M = \langle \Sigma, F, Q, \delta, q_0 \rangle$$

$\Sigma$  - alfabet wejściowy

$Q$  - zbiór stanów wewnętrznych

$\delta$  - funkcja przejść

$F$  - zbiór stanów końcowych

$q_0$  - stan początkowy

52. **Kiedy język jest językiem regularnym(formalnym)?**

Wtedy gdy jest akceptowany przez pewien automat skończony

53. **Jak formalnie definiujemy niedeterministyczny automat skończony (NFA)?**

Podobnie jak automat

$$N = \langle \Sigma, F, Q, \delta, q_0 \rangle$$

$\Sigma$  - alfabet wejściowy (skończony)

$Q$  - zbiór stanów wewnętrznych (skończony)

$\delta$  - funkcja przejść  $Q \times \Sigma_{\epsilon} \rightarrow P(Q) = \{R \mid R \subseteq Q\}$

$F$  - zbiór stanów końcowych (akceptujących)

$q_0$  - stan początkowy

54. **Kiedy R jest wyrażeniem regularnym?**

Gdy R to:

1. a z pewnego alfabetu E;
2. język zawierający jedynie słowo puste;
3. język niezawierający słów;
4. Suma dwóch innych wyrażeń regularnych;
5. iloczyn dwóch innych wyrażeń regularnych;
6. Iteracja innego wyrażenia regularnego  $\{(R)^*\}$