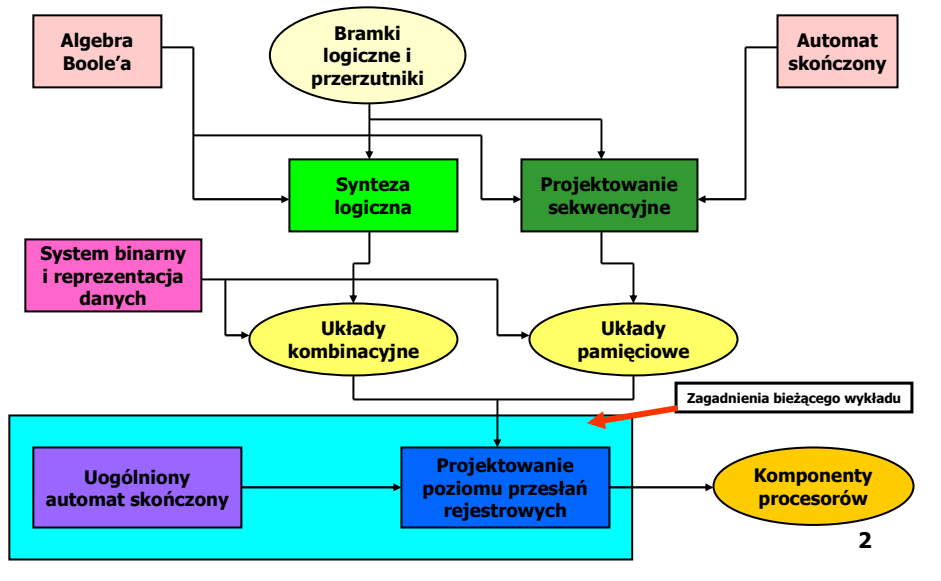


**Literatura**

1. D. Gajski, Principles of Digital Design, Prentice-Hall, 1997
  2. C. Zieliński, Podstawy projektowania układów cyfrowych, PWN, Warszawa 2003
  3. G. de Micheli, Synteza i optymalizacja układów cyfrowych, WNT, Warszawa 1998.
- Wykład oparty jest na [1], ale w bardzo przystępnej formie można znaleźć zagadnienia w [2].

**Projektowanie układów cyfrowych**



## Ścieżka danych

Wykorzystanie ścieżki danych:

- standardowe procesory
- implementacje ASIC

Ścieżka danych służy do tymczasowego przechowywania informacji w połączeniu z układami arytmetycznymi i logicznymi.

### Przykład

Operacja sumowania 100 liczb:  $y = \sum_{i=1}^{100} x_i$

```
suma = 0
loop:
  for i = 1 to 100
    suma = suma + xi
  end loop
```

3

## Ścieżka danych

Ciało pętli:

- 32-bitowa ścieżka danych,
- akumulator,
- ALU.

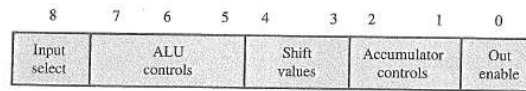
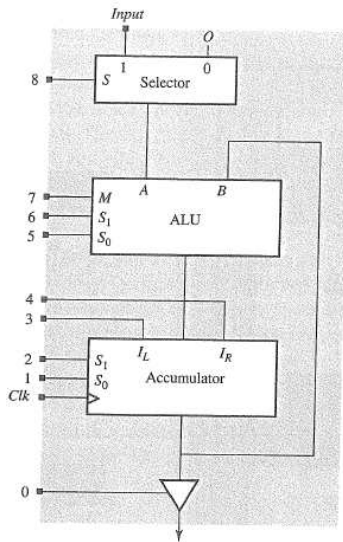
- ▶ Zmienna **suma** jest przechowywana w akumulatorze,
- ▶ W każdym taktie zegara nowa wartość **x<sub>i</sub>** jest dodawana do zmiennej **suma** w ALU,
- ▶ Nowa wartość **suma** jest ponownie przechowywana w akumulatorze

Cechy projektowania układów cyfrowych

- ✓ wartości zmiennych i stałych są przechowywane w rejestrach lub pamięciach,
- ✓ zmienne i stałe są pobierane z miejsc przechowywania przy narastającym zboczach sygnału zegarowego,
- ✓ w układach kombinacyjnych dokonywana jest transformacja (w czasie pomiędzy dwoma narastającymi zboczami sygnału zegarowego),
- ✓ wyniki są przechowywane w pamięciach lub rejestrach przy kolejnym zboczach zegara

4

## Technika cyfrowa



↑  
Słowo sterujące, określające wartości sygnałów sterujących selektora, ALU, akumulatora i bufora wyjściowego.

Słowo sterujące definiuje zachowanie ścieżki danych w każdym taktie zegara.  
Obliczenie sumy wymaga 102 taktów zegara:  
1 – zerowanie akumulatora  
100 – dodawanie  
1 – wyprowadzenie danych

← Schemat ścieżki danych

5

## Technika cyfrowa

### Przedstawiony przykład – proste operacje arytmetyczne

#### Projekty złożone:

- wiele tymczasowych zmiennych,
- rozbudowana ścieżka danych,
- bloku rejestrów zamiast pojedynczego akumulatora.

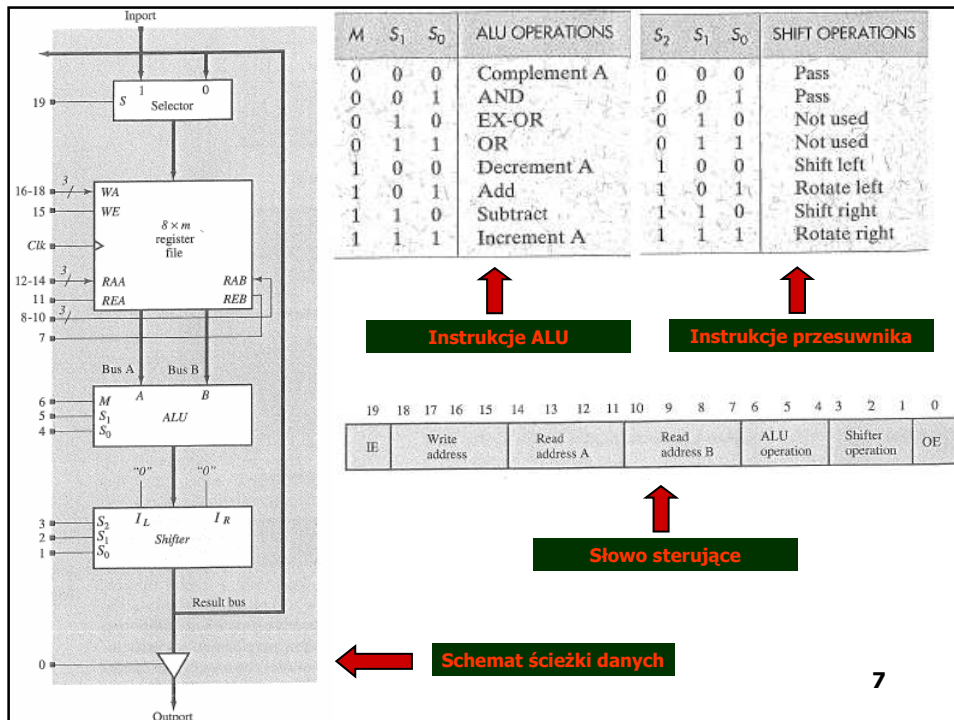
#### Przykład

##### Struktura sprzętowa

Układ cyfrowy zawierający ścieżkę danych z trójportowym blokiem rejestrów (8-rejestrowy), ALU, rejestr przesuwany i sterowniki linii (*driver*).

Ścieżka danych jest sterowana 20-bitowym słowem sterującym, wyznaczającym wszystkie operandy źródłowe i przeznaczenia oraz operacje w ścieżce danych.

6



7

## Technika cyfrowa

### Implementacja licznika „jedynek”

Układ cyfrowy zawierający ścieżkę danych z trójportowym blokiem rejestrów (8-rejestrowy), ALU, rejestr przesuwny i sterowniki linii (*driver*).

Ścieżka danych jest sterowana 20-bitowym słowem sterującym, wyznaczającym wszystkie operandy źródłowe i przeznaczenia oraz operacje w ramach ścieżki danych.

### Rozwiązanie

Słowo danych wprowadzane jest z zewnątrz na wejście ścieżki danych.

Zmienne wykorzystane do opisu algorytmu:

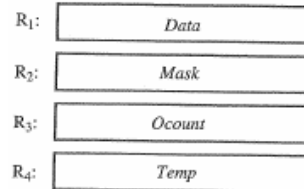
- **Data**,
- **Ocount**,
- **Mask**,
- **Temp**.

8

### Technika cyfrowa

```

1. Data := Inport
2. Ocount := 0
3. Mask := 1
while Data := 0 repeat
4. Temp := Data AND Mask
5. Ocount := Ocount + Temp
6. Data := Data >> 1 --przesuwanie
end while
7. Outport := Ocount
    
```



CONTROL WORDS	IE	WRITE ADDRESS	READ ADDRESS A	READ ADDRESS B	ALU OPERATION	SHIFTER OPERATION	OE
1	1	R <sub>1</sub>	X	X	X	X	0
2	0	R <sub>3</sub>	0	0	Add	Pass	0
3	0	R <sub>2</sub>	0	X	Increment	Pass	0
4	0	R <sub>4</sub>	R <sub>1</sub>	R <sub>2</sub>	AND	Pass	0
5	0	R <sub>3</sub>	R <sub>3</sub>	R <sub>4</sub>	Add	Pass	0
6	0	R <sub>1</sub>	R <sub>1</sub>	0	Add	Shift right	0
7	0	None	R <sub>3</sub>	0	Add	Pass	1

### Technika cyfrowa

#### Implementacja algorytmu:

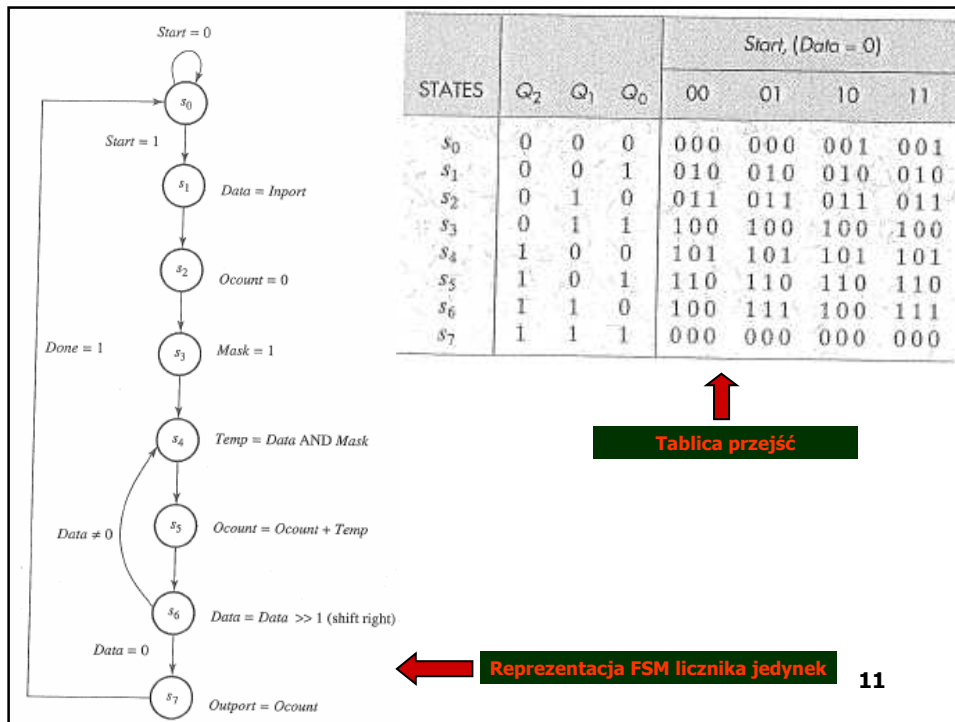
1. Przypisanie zmiennych do rejestrów
2. Wyznaczenie właściwego słowa sterującego odpowiadającego poszczególnym instrukcjom.

Sterowanie ścieżką danych wymaga, aby licznik jedynek stanowił odrębny moduł.

Reprezentacja licznika jedynek za pomocą automatu skończenie stanowego:

- układ w stanie  $s_0$  oczekuje na sygnał **Start=1**
- w stanie  $s_1 \dots s_7$  następuje wykonanie algorytmu
- w stanie  $s_7$  – wyprowadzenie wyniku, ustawienie **Done=1** i powrót do stanu  $s_0$ .

Implementacja ośmiu stanów wymaga trzech przerzutników typu D ( $Q_2, Q_1, Q_0$ )



**Technika cyfrowa**

Start (Data = 0)	Q <sub>2</sub> = 0				Q <sub>2</sub> = 1			
	Q <sub>1</sub> Q <sub>0</sub> 00	01	11	10	00	01	11	10
00	000	010	100	011	101	110	000	100
01	000	010	100	011	101	110	000	111
11	001	010	100	011	101	110	000	111
10	001	010	100	011	101	110	000	100

Q<sub>2</sub>(next), Q<sub>1</sub>(next), Q<sub>0</sub>(next)

**↑**  
**Siatka Karnaugh**

**12**

**Technika cyfrowa**

$$Q_2(next) = Q_2'Q_1Q_0 + Q_2Q_1' + Q_2Q_0'$$

$$Q_1(next) = Q_1'Q_0 + Q_2'Q_1Q_0' + (Data = 0)Q_1Q_0'$$

$$Q_0(next) = Q_2'Q_1Q_0' + Q_2Q_1'Q_0' + Start Q_1'Q_0' + (Data = 0)Q_2Q_0'$$

 **Funkcje przejść**

STATE	Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	IE	WRITE ADDRESS				READ ADDRESS A				READ ADDRESS B				ALU OPERATIONS			SHIFT OPERATIONS			OE
			WA <sub>2</sub>	WA <sub>1</sub>	WA <sub>0</sub>	WE	RAA <sub>2</sub>	RAA <sub>1</sub>	RAA <sub>0</sub>	REA	RAB <sub>2</sub>	RAB <sub>1</sub>	RAB <sub>0</sub>	REB	M	S <sub>1</sub>	S <sub>0</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	
s <sub>0</sub>	0 0 0	0	X	X	X	0	X	X	X	0	X	X	X	0	X	X	X	X	X	X	0
s <sub>1</sub>	0 0 1	1	0	0	1	1	X	X	X	0	X	X	X	0	X	X	X	X	X	X	0
s <sub>2</sub>	0 1 0	0	0	1	1	1	X	X	X	0	X	X	X	0	1	0	1	0	0	0	0
s <sub>3</sub>	0 1 1	0	0	1	0	1	X	X	X	0	X	X	X	0	1	1	1	0	0	0	0
s <sub>4</sub>	1 0 0	0	1	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0
s <sub>5</sub>	1 0 1	0	0	1	1	1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0
s <sub>6</sub>	1 1 0	0	0	0	1	1	0	0	1	1	X	X	X	0	1	0	1	1	1	0	0
s <sub>7</sub>	1 1 1	0	X	X	X	0	0	1	1	1	X	X	X	0	1	0	1	0	0	0	1

 **Tablica logiki wyjściowej**

**Technika cyfrowa**

$$IE = Q_2'Q_1'Q_0$$

$$WA_2 = Q_1'Q_0'$$

$$WA_1 = Q_2Q_0 + Q_2'Q_1$$

$$WA_0 = Q_1'Q_0 + Q_1Q_0'$$

$$WE = Q_2Q_1' + Q_2'Q_0 + Q_1Q_0'$$

$$RAA_2 = 0$$

$$RAA_1 = Q_0$$

$$RAA_0 = 1$$

$$REA = Q_1$$

$$RAB_2 = Q_0$$

$$RAB_1 = Q_0'$$

$$RAB_0 = 0$$

$$REB = Q_2Q_1'$$

$$M = Q_1 + Q_0$$

$$S_1 = Q_2'Q_0$$

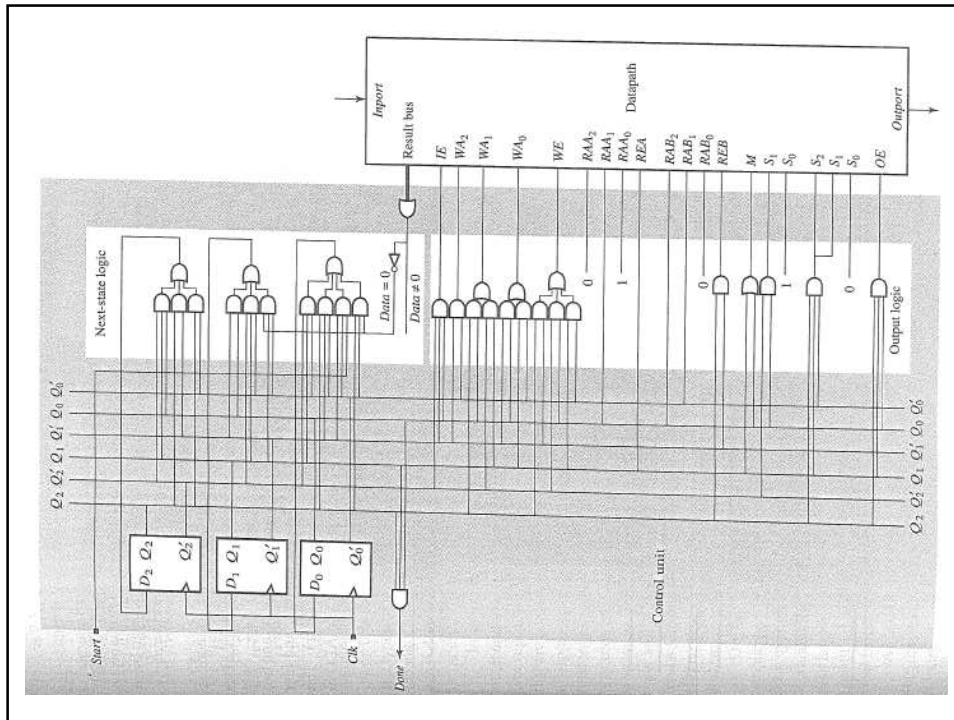
$$S_0 = 1$$

$$S_2 = S_1 = Q_2Q_1Q_0'$$

$$S_0 = 0$$

$$OE = Q_2Q_1Q_0$$

 **Funkcje wyjść**



## Technika cyfrowa

Układ ASIC lub mikroprocesor:

- co najmniej jeden układ sterujący
- ścieżka danych

Synteza RTL -> model FSM (ang. *Finite State Machine with Datapath*)

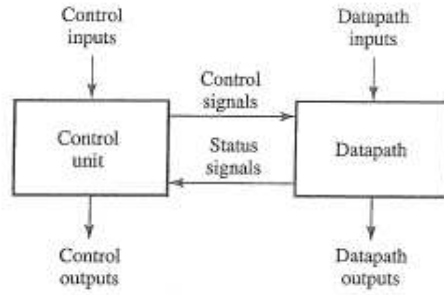
Specyfikacja projektu na poziomie RTL: schematy ASM (ang. *Algorithmic State Machine*)

Konwersja schematu ASM do implementacji z modułem sterującym i ścieżką danych

Optymalizacja implementacji ze względu na koszt i wydajność



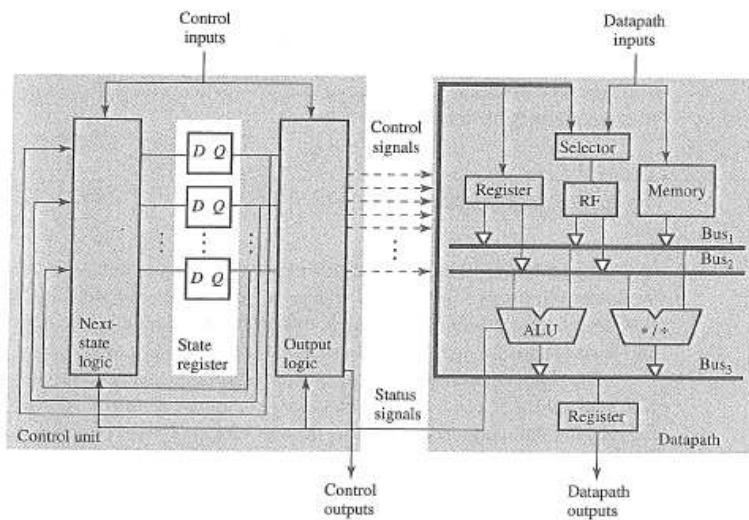
Technika cyfrowa



Schemat blokowy modelu projektu cyfrowego

17

Technika cyfrowa



Schemat blokowy na poziomie RTL

18

**Budowa ścieżki danych:**

- Moduły przechowywania informacji (rejstry, zbiory rejestrów, pamięci),
- Bloki kombinacyjne (ALU, multiplikatory, komparatory).

**Funkcje ścieżki danych:**

- Pobieranie argumenty z jednostki przechowującej dane i zwraca do niej rezultaty obliczeń w czasie pojedynczego stanu (odpowiadającemu zwykle jednemu taktowi zegara),
- Wybór operandu źródłowego, operacji oraz operandu przeznaczenia realizowany jest za pomocą jednostki sterującej poprzez ustawienie odpowiednich sygnałów sterujących ścieżki danych,
- Sygnalizacja przechowywania odpowiednich wartości bądź też spełnienie określonych relacji pomiędzy wartościami.

**Moduł sterujący:**

- sygnały wejściowe (zewnętrzne – sygnał *Start* i statusu – stan ścieżki danych, otrzymywany przez porównanie wartości wybranych zmiennych występujących w ścieżce danych np. *Data=0*),
- sygnały wyjściowe (zewnętrzne i sterujące ścieżki danych – wybór modułu).

**Specyfikacja funkcjonalności projektu cyfrowego na przykładzie licznika jedynek:**

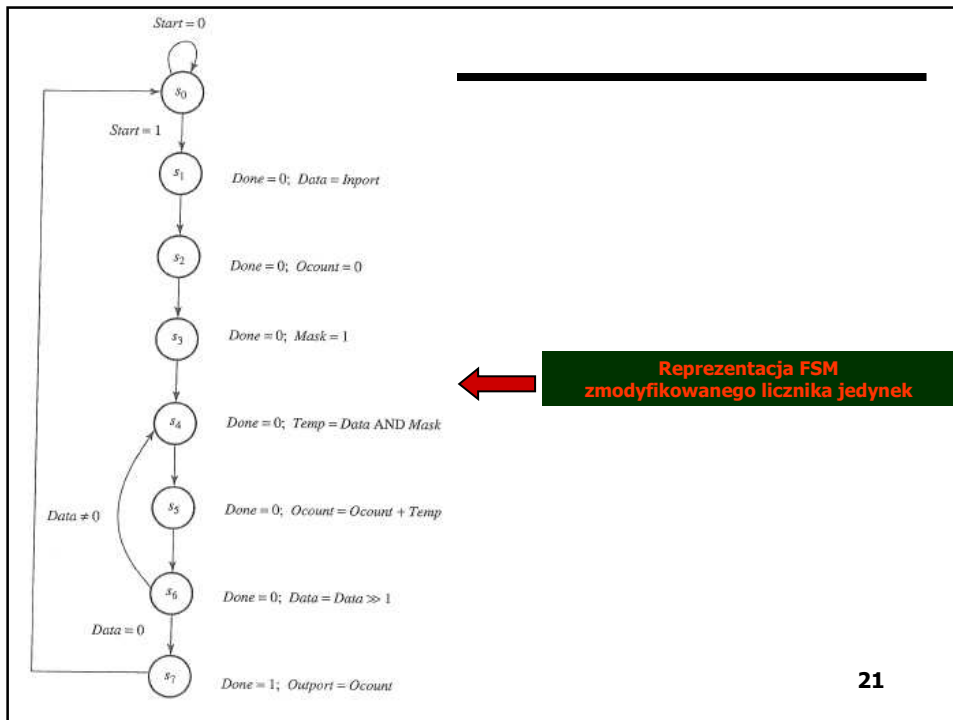
- automat FSM reprezentujący jednostkę sterującą oraz zbiór przypisań zmiennych reprezentujących przejścia w ścieżce danych.

Przypisanie zmiennych może być wykorzystane do obliczenia nowej wartości zmiennej.

W każdym stanie i dla każdego przypisania zmiennych powiązanego z tym stanem, ścieżka danych wyznacza wartość wyrażenia z prawej strony przypisania i przekazuje wynik w trakcie kolejnego narastającego zbocza sygnału zegarowego do rejestru, który przechowuje zmienną znajdującą się po lewej stronie przypisania.

Wyznaczenie wartości zmiennej odbywa się w pojedynczym stanie, ale nowa wartość zmiennej nie może zostać użyta wcześniej niż w następnym stanie.

**Model FSM zawierający instrukcję przypisania w każdym stanie, nazywany modelem FSM ze ścieżką danych (datapath) lub inaczej FSMD, służy do kompletnego opisu funkcjonalnego dowolnego projektu cyfrowego.**



**Technika cyfrowa**

**Formalna definicja FSMD rozszerza definicję FSM przez wprowadzenie zbioru zmiennych, wejść oraz wyjść ścieżki danych, uzupełniając tym samym zbiory stanów, wejść i wyjść FSM.**

**Dla FSM mamy piątkę:**

$$\langle S, I, O, f, h \rangle$$

- S** – zbiór stanów,
- I** – zbiór wejść,
- O** – zbiór wyjść,
- f** – funkcje definiujące stan następnny (przejść),
- h** – funkcje definiujące stan wyjść.

$$f: S \times I \rightarrow S$$

$$h: S \times I \rightarrow O$$

**Każdy symbol stanu, wejścia oraz wyjścia jest zdefiniowany jako iloczyn wektorowy zmiennych boolowskich.**

$$I = A_1 \times A_2 \times \dots \times A_k$$

$$S = Q_1 \times Q_2 \times \dots \times Q_m$$

$$O = Y_1 \times Y_2 \times \dots \times Y_n$$

**gdzie  $A_k$ :  $1 \leq k \leq k$ , jest sygnałem wejściowym,  $Q_m$ :  $1 \leq m \leq m$ , jest wyjściem przerzutnika, a  $Y_n$ :  $1 \leq n \leq n$ , jest sygnałem wyjściowym.**

**Rozszerzenie definicji FSM do FSMD poprzez dodanie zbioru zmiennych, wejść oraz wyjść ścieżki danych**

Zbiór zmiennych:

$$V = V_1 \times V_2 \times \dots \times V_q,$$

definiuje stan ścieżki danych poprzez wyspecyfikowanie wartości wszystkich zmiennych w każdym stanie.

Zbiór wejść FSMD można podzielić na zbiór wejść FSM  $I_C$  oraz zbiór wejść ścieżki danych  $I_D$ :

$$I = I_C \times I_D$$

gdzie:

$$I_D = B_1 \times B_2 \times \dots \times B_p$$

Zbiór wyjść FSMD można podzielić na zbiór wyjść FSM  $O_C$  oraz zbiór wyjść ścieżki danych  $O_D$ :

$$O = O_C \times O_D$$

gdzie:

$$O_D = Z_1 \times Z_2 \times \dots \times Z_r$$

**A, Q, Y oznaczają zmienne boolowskie  
B, V, Z oznaczają wektory boolowskie reprezentujące w konsekwencji wartości całkowite, zmiennoprzecinkowe oraz znakowe.**

Tablica stanów i wyjść FSMD dla licznika jedynek



PRESENT STATE	NEXT STATE {Start, Data = 0}				CONTROL OUTPUT Done	DATAPATH OUTPUT Output	DATAPATH VARIABLES			
	00	01	10	11			Data	Ocount	Temp	Mask
$s_0$	$s_0$	$s_0$	$s_1$	$s_1$	0	Z	X	X	X	X
$s_1$	$s_2$	$s_2$	$s_2$	$s_2$	0	Z	Inport	X	X	X
$s_2$	$s_3$	$s_3$	$s_3$	$s_3$	0	Z	Data	0	X	X
$s_3$	$s_4$	$s_4$	$s_4$	$s_4$	0	Z	Data	Ocount	X	1
$s_4$	$s_5$	$s_5$	$s_5$	$s_5$	0	Z	Data	Ocount	Data AND Mask	Mask
$s_5$	$s_6$	$s_6$	$s_6$	$s_6$	0	Z	Data	Ocount+Temp	X	Mask
$s_6$	$s_4$	$s_7$	$s_4$	$s_7$	0	Z	Data>>1	Ocount	X	Mask
$s_7$	$s_0$	$s_0$	$s_0$	$s_0$	1	Ocount	Data	Ocount	X	X

**Technika cyfrowa**

PRESENT STATE	NEXT STATE (Start, Data = 0)				CONTROL OUTPUT	DATAPATH OUTPUT	DATAPATH VARIABLES
	00	01	10	11			
$s_0$	$s_0$	$s_0$	$s_1$	$s_1$	0	Z	
$s_1$	$s_2$	$s_2$	$s_2$	$s_2$	0	Z	<i>Data = Inport</i>
$s_2$	$s_3$	$s_3$	$s_3$	$s_3$	0	Z	<i>Ocount = 0</i>
$s_3$	$s_4$	$s_4$	$s_4$	$s_4$	0	Z	<i>Mask = 1</i>
$s_4$	$s_5$	$s_5$	$s_5$	$s_5$	0	Z	<i>Temp = Data AND Mask</i>
$s_5$	$s_6$	$s_6$	$s_6$	$s_6$	0	Z	<i>Ocount = Ocount + Temp</i>
$s_6$	$s_4$	$s_7$	$s_4$	$s_7$	0	Z	<i>Data = Data &gt;&gt; 1</i>
$s_7$	$s_0$	$s_0$	$s_0$	$s_0$	1	<i>Ocount</i>	

↑  
Tablica stanów, wyjść i przypisań zmiennych

25

**Technika cyfrowa**

PRESENT STATE	NEXT STATE	CONTROL AND DATAPATH ACTIONS
	CONDITION, STATE	CONDITION, ACTIONS
$s_0$	[ <i>Start = 0, <math>s_0</math></i> <i>Start = 1, <math>s_1</math></i> ]	[ <i>Done = 0</i> <i>Output = Z</i> <i>Data = Inport</i> ]
$s_1$		
$s_2$	$s_3$	<i>Ocount = 0</i>
$s_3$	$s_4$	<i>Mask = 1</i>
$s_4$	$s_5$	<i>Temp = Data AND Mask</i>
$s_5$	$s_6$	<i>Ocount = Ocount + Temp</i>
$s_6$	[ <i>Data ≠ 0, <math>s_4</math></i> <i>Data = 0, <math>s_7</math></i> ]	<i>Data = Data &gt;&gt; 1</i>
$s_7$		
		[ <i>Done = 1</i> <i>Output = Ocount</i> ]

↑  
Tablica „State-Action”

26

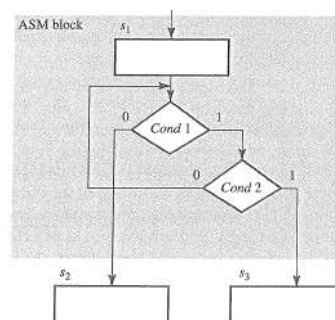
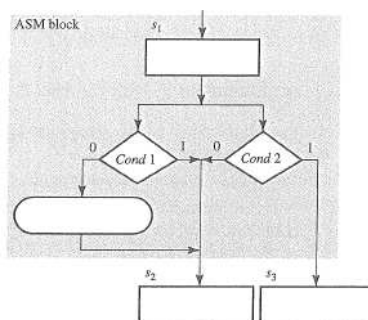
Wykresy ASM

Nazwa	Definicja	Przykład
<b>blok stanu</b>	State name ↓ State code ↓ Unconditional variable and output assignment	$S_3$ ↓ 011 ↓ $Temp = Temp \text{ AND } Mask$
<b>blok decyzyjny</b>	↓ Condition	↓ Data = 0
<b>blok warunkowy</b>	↓ Conditional variable assignment	↓ $Ocount = Ocount + 1$

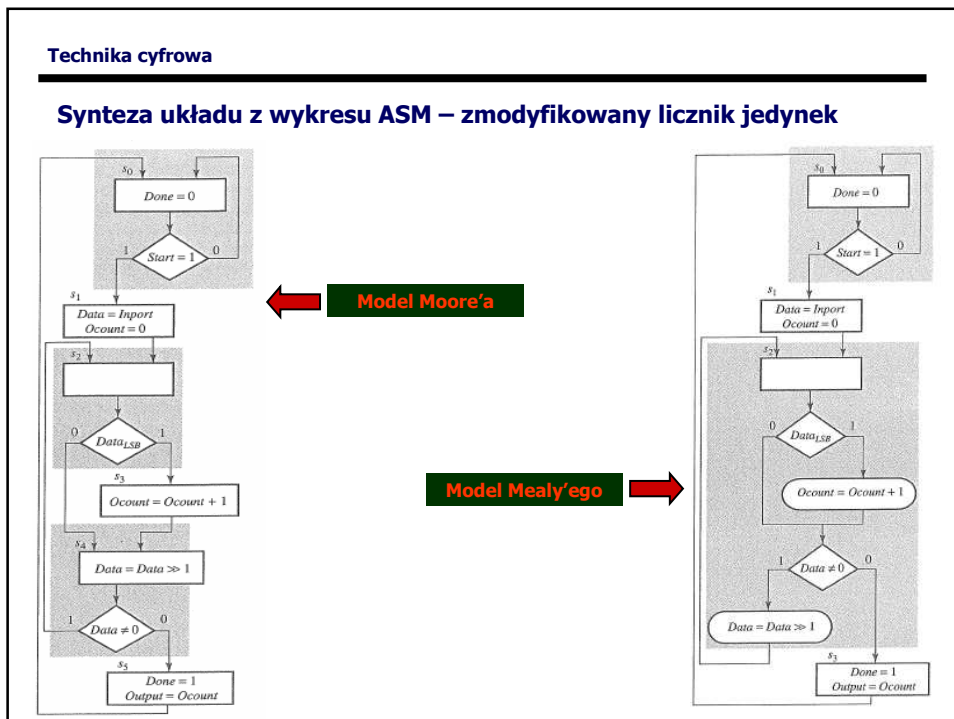
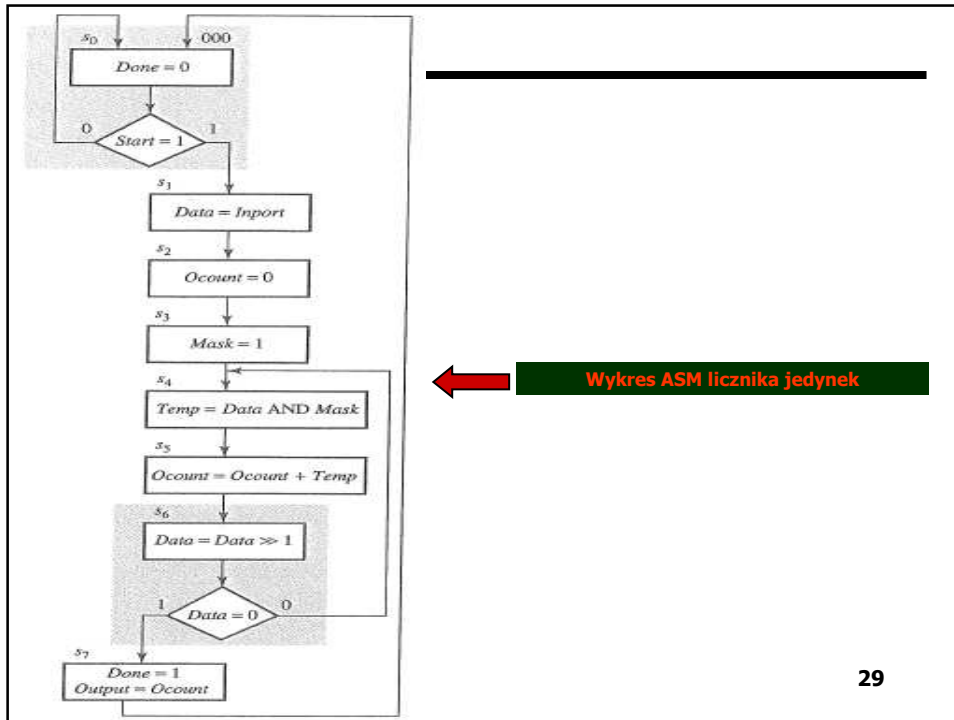
27

Zasady specyfikacji FSM za pomocą ASM

1. Dla każdego stanu i zestawu warunków stan następny musi być unikalny.
2. Każda ścieżka zdefiniowana przez sieć bloków warunkowych musi prowadzić do innego stanu.



28

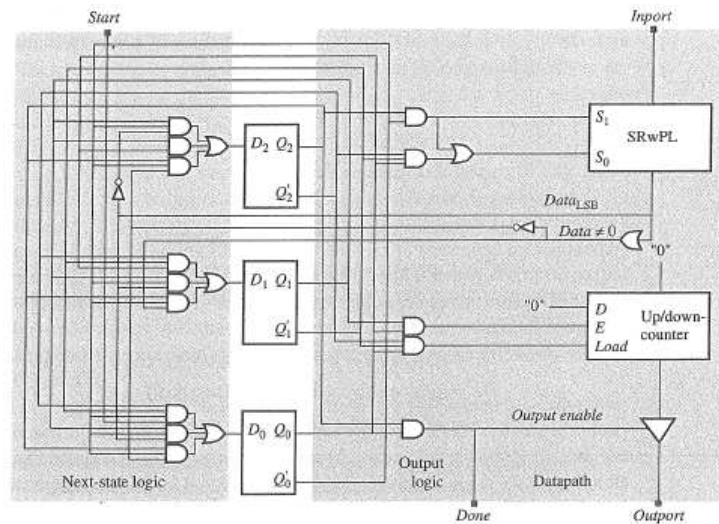


Synteza układu z wykresu ASM – zmodyfikowany licznik jedynek

PRESENT STATE			NEXT STATE		DATAPATH ACTIONS	
Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	NAME	CONDITION, STATE	CONDITION, OPERATIONS	
0	0	0	s <sub>0</sub>	[ Start = 0, s <sub>0</sub> ]	Done = 0	
0	0	1	s <sub>1</sub>	[ Start = 1, s <sub>1</sub> ]	[ Data = Inport ]	
0	1	0	s <sub>2</sub>	[ Data <sub>LSB</sub> = 1, s <sub>3</sub> ]	[ Ocount = 0 ]	
0	1	1	s <sub>3</sub>	[ Data <sub>LSB</sub> = 0, s <sub>4</sub> ]	Ocount = Ocount + 1	
1	0	0	s <sub>4</sub>	[ Data ≠ 0, s <sub>5</sub> ]	Data = Data >> 1	
1	0	1	s <sub>5</sub>	[ Data = 0, s <sub>0</sub> ]	[ Done = 1 ]	
					[ Output = Ocount ]	

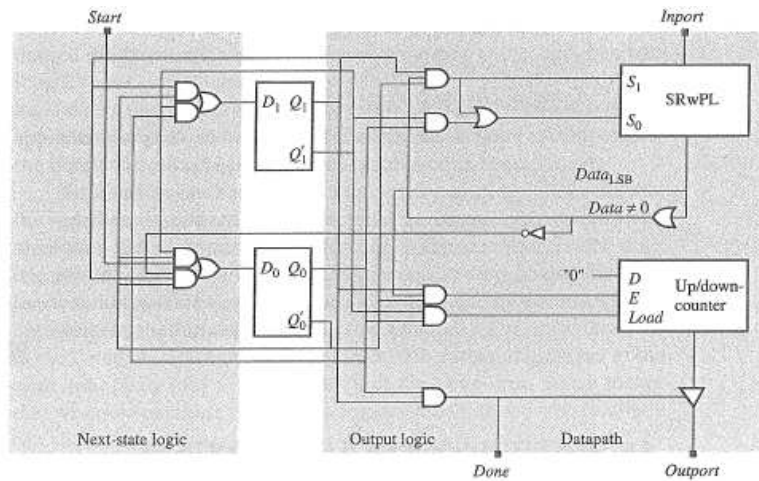
PRESENT STATE			NEXT STATE		DATAPATH ACTIONS	
Q <sub>1</sub>	Q <sub>0</sub>	NAME	CONDITION, STATE	CONDITION, OPERATIONS		
0	0	s <sub>0</sub>	[ Start = 0, s <sub>0</sub> ]	Done = 0		
0	1	s <sub>1</sub>	[ Start = 1, s <sub>1</sub> ]	[ Data = Inport ]		
1	0	s <sub>2</sub>	[ Data ≠ 0, s <sub>2</sub> ]	[ Ocount = 0 ]		
1	1	s <sub>3</sub>	[ Data = 0, s <sub>3</sub> ]	[ Data <sub>LSB</sub> = 1, Ocount = Ocount + 1 ]		
				[ Data ≠ 0, Data = Data >> 1 ]		
				[ Done = 1 ]		
				[ Output = Ocount ]		

Synteza układu z wykresu ASM – zmodyfikowany licznik jedynek





Synteza układu z wykresu ASM – zmodyfikowany licznik jedynek



33

Wybrane techniki optymalizacji ścieżki danych:

- współdzielenie rejestrów/pamięci (łączenie lub grupowanie zmiennych),
- współdzielenie modułów funkcjonalnych (łączenie operatorów),
- współdzielenie magistral (łączenie połączeń).

34