

Podstawy Techniki Cyfrowej Repetytorium

Wróg publiczny nr 1

Wróg publiczny nr 2

anonim

anonim

anonim

22.01.2021

Spis treści

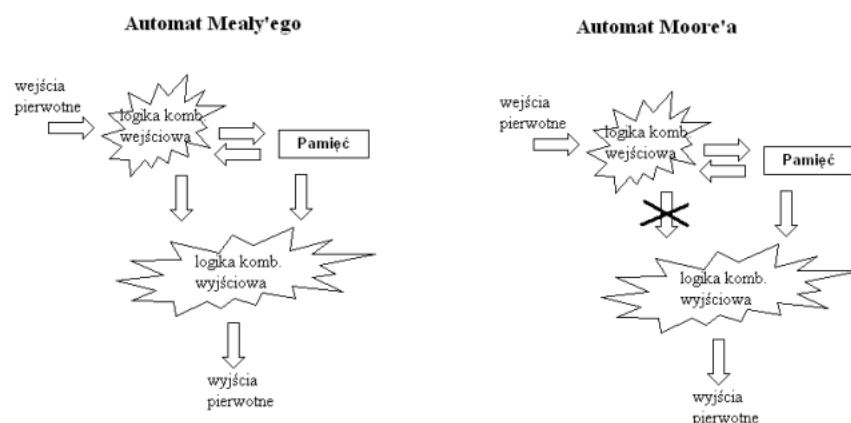
1	Wstęp	3
2	Automaty synchroniczne	3
2.1	Automat Mealy’ego	3
2.2	Automat Moore’a	4
2.3	Analiza i synteza grafów przejść dla automatów	5
2.3.1	Wykrywanie oraz analiza sekwencji bitów	5
2.3.2	Komparator szeregowy	6
2.3.3	Mnożenie, dodawanie, odejmowanie, sprawdzanie podzielności i podobne	6
2.4	Projektowanie automatów	7
2.4.1	Redukcja stanów automatów Mealy’ego	7
2.4.2	Redukcja stanów automatów Moore’a	8
2.4.3	Określenie wielkości i zawartości pamięci	8
2.5	Przerzutniki	8
2.5.1	Tabele przejść	8
2.5.2	Konwersja przerzutników	9
2.6	ROM	10
2.6.1	Implementacja automatu Moore’a w ROM	10
2.7	Implementacja automatu Mealy’ego w ROM	11
3	Synteza strukturalna układu cyfrowego na poziomie instrukcji przesłań między rejestrowych (RTL)	11
3.1	Bloki	11
3.2	Operacje	12
3.3	przesłania między-rejestrowych	12
3.3.1	RTL dla jednego rejestru	12
3.4	Zastosowanie multipleksera do realizacji prostych funkcji logicznych	13
4	Układ sterowania	13
4.1	Diagramy ASM	13
4.1.1	Specyfikacja układu	13
4.1.2	Przykład	14
4.1.3	Kod 1 z N	14
4.2	Mikrokod	15
4.2.1	Notacja	15
4.2.2	Semantyka mikrokodu (Moore’a)	15
4.2.3	Przykładowy mikrokod (Moore’a)	15

4.2.4	Przykład układu wykonującego mikrokod (Moore'a)	16
4.2.5	Semantyka mikrokodu (Mealy'ego)	16
4.3	Przykład układu wykonującego mikrokod (Mealy'ego)	16
5	Pamięci półprzewodnikowe	17
5.1	Pamięć statyczna SRAM	17
5.1.1	Własności	17
5.2	Pamięć dynamiczna DRAM	17
5.2.1	Własności	17
6	Przykładowe zadania	18
6.1	Zestaw 0	18
6.1.1	Zadanie 1	18
6.1.2	Przerzutniki D - wariant B	19
6.1.3	Zadanie 2	19
6.1.4	Zadanie 3	20
6.2	Zestaw 1	23
6.2.1	Zadanie 1	23
6.2.2	Zadanie 2	24
6.2.3	Zadanie 3	25
6.3	Zestaw 2	25
6.3.1	Zadanie 1	25
6.3.2	Zadanie 2	27
6.3.3	Zadanie 3	28
6.4	Zestaw 3	29
6.4.1	Zadanie 1	29
6.4.2	Zadanie 2	30
6.4.3	Zadanie 3	31

1 Wstęp

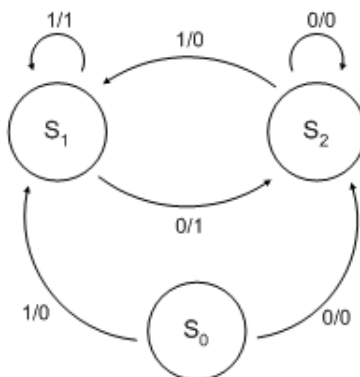
Poniższe 30 stron wiedzy jest wynikiem wielu godzin prawdziwej współpracy i konsultacji z doktorem Wal-kowiakiem. Materiały powstały w trakcie przygotowywania się do pierwszego terminu drugiego kolokwium zaliczeniowego z PTC, które odbyło się dnia 22.01.2021 roku. Praca nad materiałami zakończyła się tego samego dnia, a w pliku brakuje zagadnień dotyczących pamięci półprzewodnikowych statycznych RAM. Do-głębne zapoznanie się z całością dokumentu powinno być wystarczające do zaliczenia kolokwium w pierwszym terminie, niemniej jednak autorzy nie podejmują żadnej odpowiedzialności za możliwe ewentualne błędy za-równo w części teoretycznej jak i praktycznej. Powodzenia w nauce!

2 Automaty synchroniczne



Rysunek 2.1: Porównanie automatów Mealy'ego i Moore'a.

2.1 Automat Mealy'ego



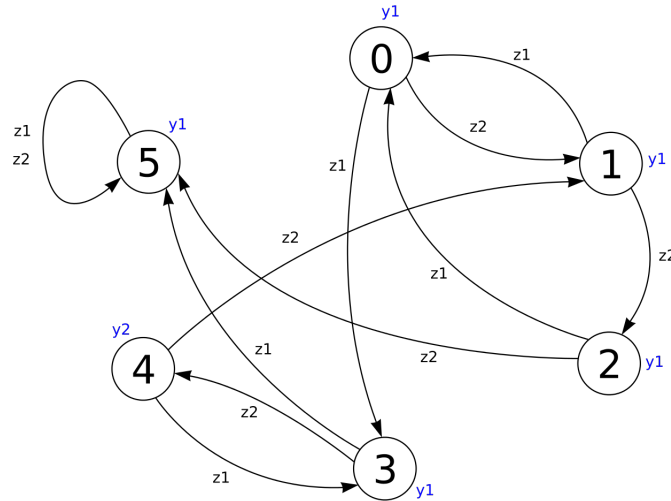
Rysunek 2.2: Przykład schematu automatu Mealy'ego.

AUTOMAT MEALY'EGO – automat, którego wyjście jest funkcją stanu wewnętrznego i sygnałów wejściowych. AUTOMAT MEALY'EGO reprezentowany jest przez uporządkowaną szóstkę:

- $Z = \{z_1, z_2, \dots, z_n\}$ - zbiór sygnałów wejściowych,
- $Q = \{q_1, q_2, \dots, q_n\}$ - zbiór stanów wewnętrznych,

- $Y = \{y_1, y_2, \dots, y_n\}$ - zbiór sygnałów wyjściowych,
- Φ - funkcja przejść $q(t+1) = \Phi[q(t), z(t)]$,
- Ψ - funkcja wyjść $y(t) = \Psi[q(t), z(t)]$, zależy od stanu, w którym znajduje się automat oraz od sygnału wejściowego,
- q_0 - stan początkowy, należy do zbioru Q .

2.2 Automat Moore'a



Rysunek 2.3: Przykład automatu Moore'a.

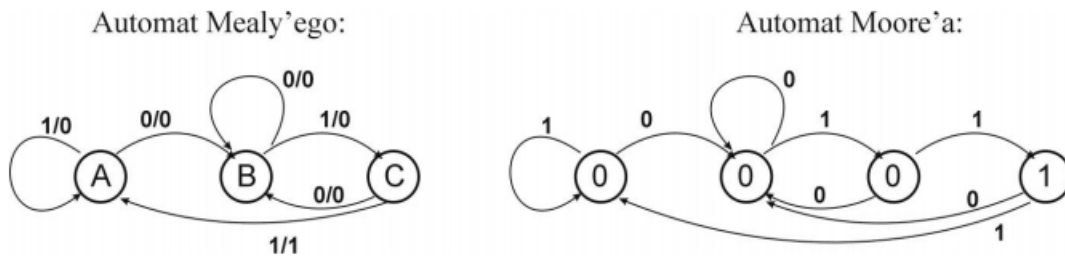
AUTOMAT MOORE'A – automat, którego wyjście jest funkcją wyłącznie stanu wewnętrznego. AUTOMAT MOORE'A reprezentowany jest przez uporządkowaną szóstkę:

- $Z = \{z_1, z_2, \dots, z_n\}$ - zbiór sygnałów wejściowych,
- $Q = \{q_1, q_2, \dots, q_n\}$ - zbiór stanów wewnętrznych,
- $Y = \{y_1, y_2, \dots, y_n\}$ - zbiór sygnałów wyjściowych,
- Φ - funkcja przejść $q(t+1) = \Phi[q(t), z(t)]$,
- Ψ - funkcja wyjść $y(t) = \Psi[q(t)]$, zależy od stanu, w którym znajduje się automat oraz od sygnału wejściowego,
- q_0 - stan początkowy, należy do zbioru Q .

Dla AUTOMAT MOORE'A poszczególnym stanom przypisano wartości wyjścia automatu. W ogólności liczba stanów AUTOMAT MOORE'A nie może być mniejsza niż dla realizacji według MEALY'EGO.

2.3 Analiza i synteza grafów przejść dla automatów

2.3.1 Wykrywanie oraz analiza sekwencji bitów



Rysunek 2.4: Przykładowe grafy automatów dla sekwencji 011, gdzie 1 oznacza odnalezienie sekwencji, a 0 niepowodzenie.

AUTOMAT MEALY'EGO

Przyjmijmy, że nasz ciąg cyfr wygląda następująco: *0010110*, natomiast na wyjściu zaobserwujemy następującą sekwencję zer oraz jedynek: *0000001*.

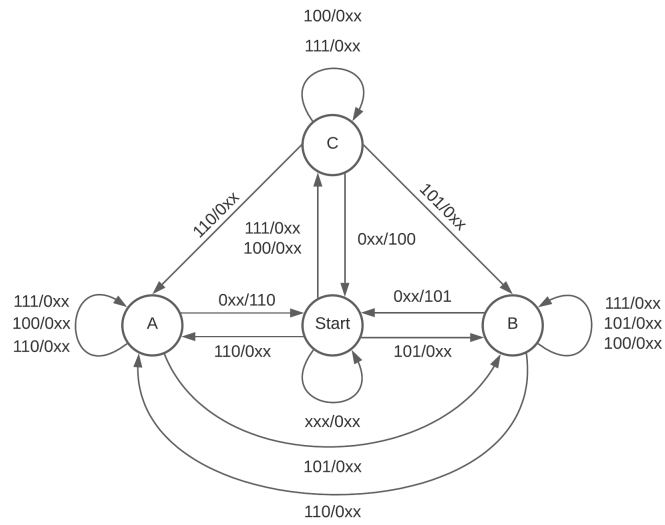
Zaczynamy w stanie **A** pierwsze zero przeniesie nas do stanu **B**, a na wyjściu zaobserwujemy 0. Kolejne zero podtrzyma nas w stanie wewnętrznym **B**. Jedyńska przeniesie nas do stanu **C**, a na wyjściu zaobserwujemy kolejne 0. Zero podane na wejściu w stanie **C** przeniesie nas z powrotem do stanu **B** (kolejne 0 na wyjściu). Następnie sekwencja dwóch jedynek przeniesie nas kolejno do stanów **C** i **A**, wystąpienie drugiej jedynki spowoduje wyświetlenie 1 na wyjściu.

AUTOMAT MOORE'A

Przyjmijmy, że nasz ciąg cyfr ma postać: *1011*, w takim razie na wyjściu otrzymamy ciąg: *0001*.

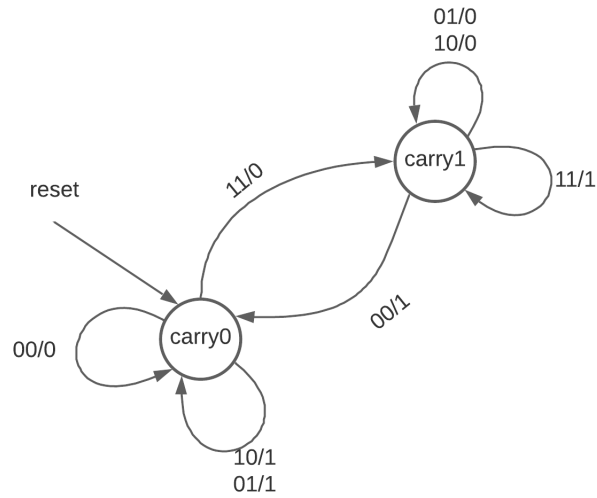
W tym przypadku zaczynamy od lewego stanu z 0 i dalej poruszamy się według strzałek. Za każdym razem gdy na wejściu pojawia się 1 wybieramy krawędź oznaczoną 1, która *wychodzi* z danego stanu, analogicznie dla 0. Po dotarciu do stanu wypisujemy jego wartość jako wyjście.

2.3.2 Komparator szeregowy

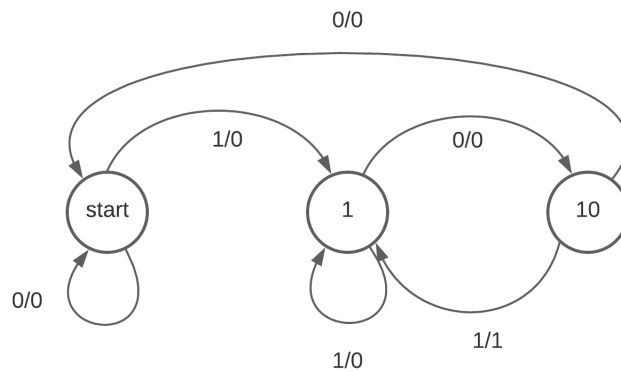


Rysunek 2.5: Komparator szeregowy dwóch liczb.

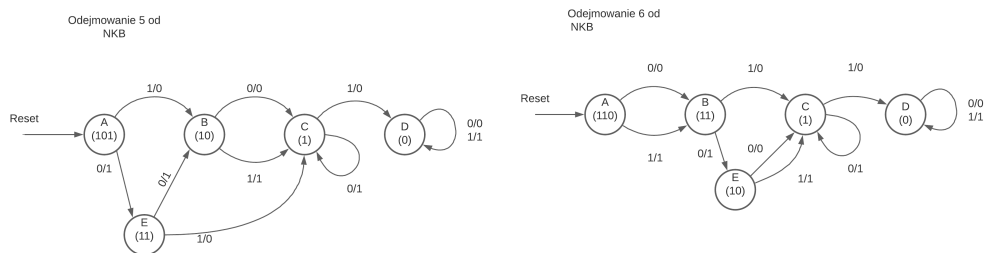
2.3.3 Mnożenie, dodawanie, odejmowanie, sprawdzanie podzielności i podobne



Rysunek 2.6: Sumator dwóch liczb pobierający po jednym bicie zaczynając najmłodszego (najmniej znaczącego) bitu.



Rysunek 2.7: Przeszukiwanie ciągu liczb w celu znalezienia liczby 5 (101).



Rysunek 2.8: Odejmnowanie 5 od NKB oraz odejmowanie 6 od NKB.

2.4 Projektowanie automatów

2.4.1 Redukcja stanów automatów Mealy'ego

1. Tworzymy tabelę pomocniczą stanów w taki sposób, żeby uzyskać każdą możliwą kombinację par stanów. Żeby pary się nie powtarzały, będziemy rozpatrywać tylko dolny trójkąt tabeli (patrz rysunek 9)

B				
C				
D				
E				
	A	B	C	D

Rysunek 2.9: Przykładowa tabela dla automatu o 5 stanach

2. Dla każdej pary stanów patrzymy czy wyjścia się pokrywają, jeżeli nie, w odpowiedniej komórce wpisujemy 'X', jeżeli tak, to możemy przejść do następnego kroku.
3. Sprawdzamy każdą parę przejść dla analizowanej pary

- (a) Jeżeli wszystkie przejścia są identyczne, to stany można bezwarunkowo połączyć, w odpowiedniej co zaznaczamy w odpowiedniej komórce np. wpisując "OK"
 - (b) Jeżeli jakieś przejścia się różnią, wpisujemy w odpowiedniej komórce równoważność tych stanów jako warunek równoważności stanów analizowanych.
4. Sprawdzamy każdy warunek równoważności stanów, jeżeli choć jeden z nich nie będzie spełniony (prowadzi do sprawdzenia komórki w której wcześniej wpisaliśmy 'X') to możemy przekreślić całą komórkę, od teraz traktujemy ją jako 'X'.
 5. Jeżeli pozostały jakiegokolwiek nieprzekreślone warunki, to znaczy, że dane stany możemy zapisać jako równoważne. Zapisujemy tabelę przejść dla zredukowanego automatu.

2.4.2 Redukcja stanów automatów Moore'a

Redukcja stanów automatu Moore'a przebiega analogicznie jak automatu Mealy'ego.

2.4.3 Określenie wielkości i zawartości pamięci

Wielkość pamięci to ilość bitów zapisanych w jednym rejestrze razy ilość rejestrów.

2.5 Przerzutniki

2.5.1 Tabele przejść

Q na Q+	J	K
0 na 0	0	X
0 na 1	1	X
1 na 0	X	1
1 na 1	X	0

Tablica 1: Tabela przejść dla przerzutnika JK.

Q na Q+	D
0 na 0	0
0 na 1	1
1 na 0	0
1 na 1	1

Tablica 2: Tabela przejść dla przerzutnika D.

Q na Q+	T
0 na 0	0
0 na 1	1
1 na 0	1
1 na 1	0

Tablica 3: Tabela przejść dla przerzutnika T.

2.5.2 Konwersja przerzutników

D	Q	Q+	J	K
0	0	0	0	x
1	0	1	1	x
0	1	0	x	1
1	1	1	x	0

Tablica 4: Konwersja przerzutnika JK na D (D z JK). Funkcje: $J = DQ'$ oraz $K = D'Q$.

D	Q	Q+	T
0	0	0	0
1	0	1	1
0	1	0	1
1	1	1	0

Tablica 5: Konwersja przerzutnika T na D (D z T). Funkcje: $T(D,Q) = \text{SUMA}(1,2)$ $T = D'Q + DQ'$.

J	K	Q	Q+	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Tablica 6: Konwersja przerzutnika D na JK (JK z D). Funkcje: $D(J,K,Q) = \text{SUMA}(1,4,5,6)$ $D = JQ' + K'Q$.

J	K	Q	Q+	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

Tablica 7: Konwersja przerzutnika T na JK (JK z T). Funkcje: $T(J,K,Q) = \text{SUMA}(3,4,6,7)$ $T = KQ + JQ'$.

T	Q	Q+	D
0	0	0	0
1	0	1	1
1	1	0	0
0	1	1	1

Tablica 8: Konwersja przerzutnika D na T (T z D). Funkcje: $D = TQ' + T'Q$.

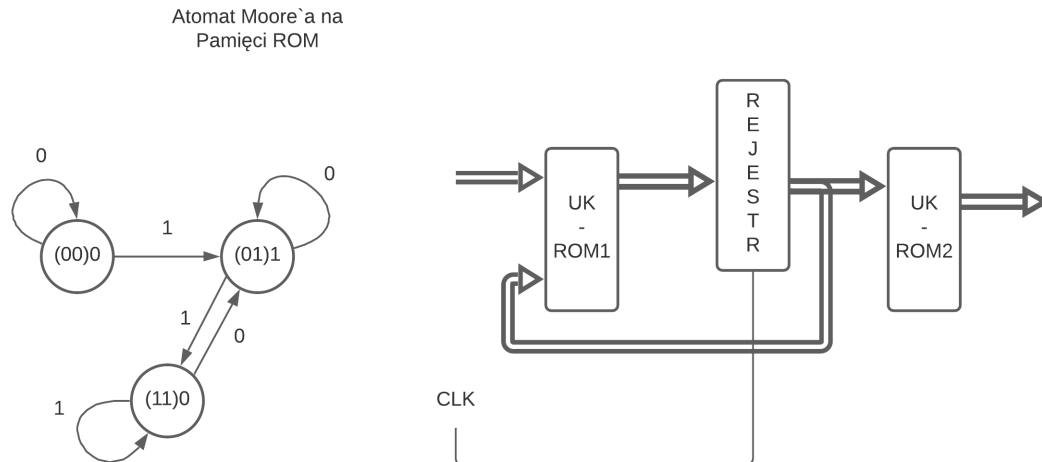
T	Q	Q+	J	K
0	0	0	0	x
1	0	1	1	x
1	1	0	x	1
0	1	1	x	0

Tablica 9: Konwersja przerzutnika JK na T (T z JK). Funkcje: $J = TQ'$ $K = TQ$.

2.6 ROM

2.6.1 Implementacja automatu Moore'a w ROM

Implementacja automatu Moore'a w tym przypadku polega na podaniu na odpowiednie adresy w pamięci ROM1 stanów na które chcemy przejść np. dla automatu pokazanego po lewej stronie w po wpisaniu na wejście 1 i w stanie 00 (adres 00 1) chcemy otrzymać stan 01. W ROM2, zaś wpisujemy odpowiednie wyjścia dla danych stanów np. 0 dla stanu (adresu) 00. Rejestr służy do ustabilizowania wyjścia - stan będzie MÓGŁ się zmienić tylko wtedy, gdy dostaniemy sygnał CLK. Skutkuje to tym że za rejestrem utrzymamy jest bieżący stan a przed rejestrem pojawia następny stan.



Rysunek 2.10: Przedstawienie automatu Moore'a na pamięci ROM.

Stan	Wejście		Wyjście
	Z=0	Z=1	
Q 00	00	01	0
01	01	11	0
11	01	11	0

Tablica 10: Tablica przejść automatu.

Adres		Dane
Stan	Wejście	Stan
00	0	00
00	1	01
01	0	01
01	1	11
10	0	DC
10	1	DC
11	0	01
11	1	11

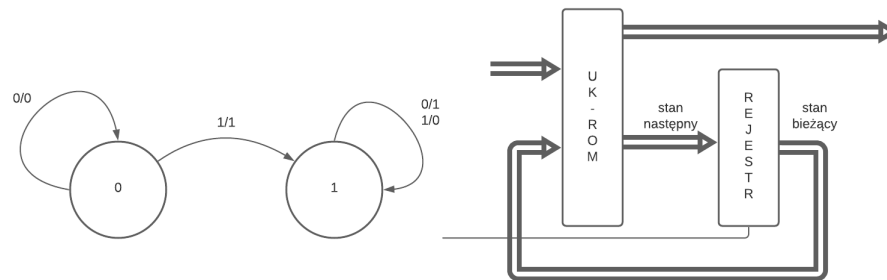
Tablica 11: Zawartość ROM1

Stan	Wyjście
00	0
01	1
10	DC
11	0

Tablica 12: Zawartość ROM2

2.7 Implementacja automatu Mealy'ego w ROM

Implementacja automatu Mealy'ego w ROM polega praktycznie na zakodowaniu w kolumnie 'ADRES' wszystkich kombinacji zmiennych STAN i WEJŚCIE, a w kolumnie DANE odpowiednich pól (z przecięć kolumn i wierszy) z tabeli przejść i wyjść.



Rysunek 2.11: Przedstawienie automatu Mealy'ego na pamięci ROM.

Stan Q	Przejście		Wyjście	
	We=0	We=1	Wy=0	Wy=1
0	0	1	0	1
1	1	1	1	0

Tablica 13: Tablica przejść automatu.

Adres		Dane
Stan, Wejście	Wyjście, Stan*	
0 0	0 0	
0 1	1 1	
1 0	1 1	
1 1	0 1	

Tablica 14: Zawartość ROM

3 Synteza strukturalna układu cyfrowego na poziomie instrukcji przesłań między rejestrowych (RTL)

3.1 Bloki

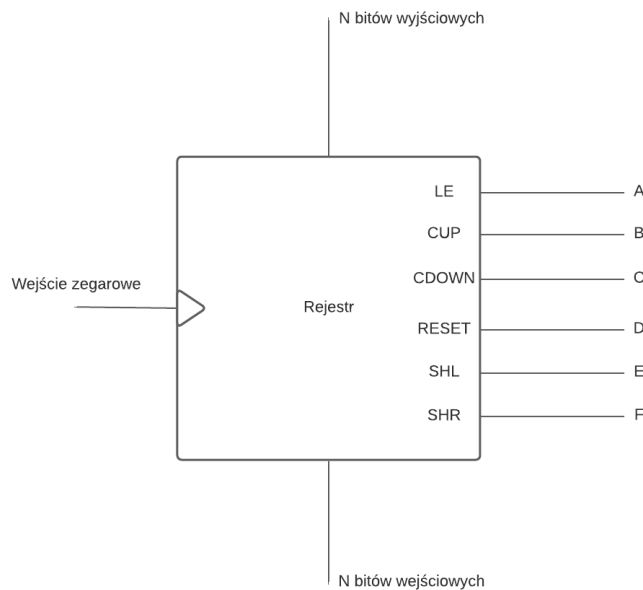
- Funkcjonalne: służące do przechowywania zmiennych
- Operacyjne: służące do wykonywania operacji
- Funkcjonalne: służące do przesyłania danych między rejestrami i blokami operacyjnym

3.2 Operacje

3.3 przesłania między-rejestrowych

- Mikrooperacje arytmetyczne
 - $R_0 \leftarrow R_1 + R_2$: Dodawanie
 - $R_0 \leftarrow R'_0$: Negacja (uzupełnienie do jeden)
 - $R_0 \leftarrow R'_0 + 1$: Uzupelnienie do 2
 - $R_0 \leftarrow R_1 + R'_2 + 1$: Odejmowanie
 - $R_0 \leftarrow R_0 + 1$: Inkrementacja
 - $R_0 \leftarrow R_0 - 1$: Dekrementacja
- Mikrooperacje logiczne - operacje bitowe
 - $R_0 \leftarrow R_0 \text{ OR } 1$: Ustawienie zawartości (wpis 1)
 - $R_0 \leftarrow R_0 \text{ AND } 0$: Zerowanie
 - $R_0 \leftarrow R_0 \text{ XOR } R_2$: negacja bitów rej R_0 sterowania zawartością rej R_1
- Mikrooperacje przesunięcia
 - $R_0 \leftarrow SR R_0$ - shift right
 - $R_0 \leftarrow SL R_0$ - shift left
 - Przesuniecie zawartości rejestru – dzielenie, mnożenie; rejestr przesuwny ???

3.3.1 RTL dla jednego rejestru

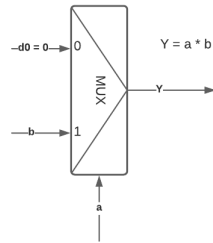


Rysunek 3.1: Piękny rejestr.

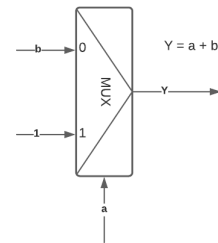
- $A : R_0 \leftarrow WE$: wpis wartości - sygnał LE: Pozwolenie na zapis,
- $B : R_0 \leftarrow R_0 + 1$: Dodawanie 1 - sygnał Count Up: CUP,

- $C : R_0 \leftarrow R_0 - 1$: Odejmowanie 1 - sygnał Count Down: CDOWN,
- $D : R_0 \leftarrow 0$: Zerowanie wartości - RESET,
- $E : R_0 \leftarrow R_0 * 2$: Mnożenie przez 2 - sygnał shift left: SHL (wpis najmłodszego bitu = 0),
- $F : R_0 \leftarrow R_1/2$: Dzielenie przez 2 - sygnał shift right: SHR (wpis najstarszego bitu = 0),
- **Realizacja operacji zboczem zegara poprzedzona ustaleniem wartości sygnałów sterujących**

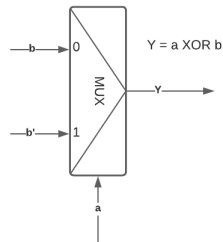
3.4 Zastosowanie multipleksera do realizacji prostych funkcji logicznych



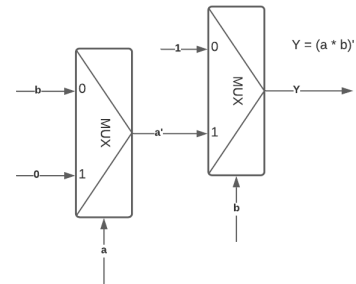
Rysunek 3.2: Logiczny AND.



Rysunek 3.3: Logiczny OR.



Rysunek 3.4: Logiczny XOR.



Rysunek 3.5: Logiczny NAND.

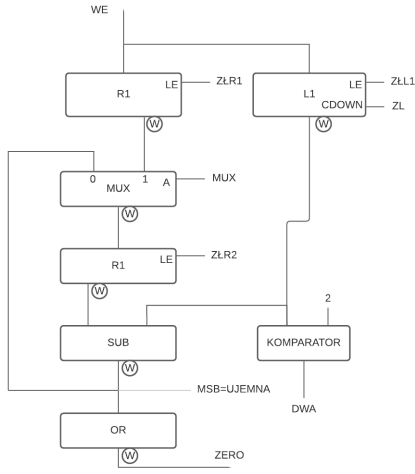
4 Układ sterowania

4.1 Diagramy ASM

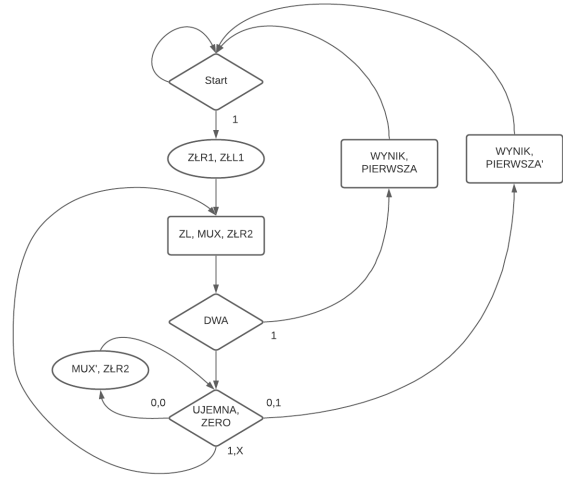
4.1.1 Specyfikacja układu

- Stanowią alternatywę opisu automatów.
- Pozwalają na reprezentacje cyfrowe układu sekwencyjnego w postaci sieci działań.
- Składa się z bloków ASM, Klatek:
 - Operacyjna: Prostokąt reprezentujący stan; nazwa stanu obok figury. Wewnątrz umieszcza się akcje przypisywania wartości do sygnałów, w momencie wejścia automatu do tego stanu. są odpowiadające wyjściom w Moore'rze
 - Decyzyjna: Deltoid, Sprawdza warunek, stan sygnału, w celu określenia ścieżki przejścia do następnego stanu. Możliwe jest łączenie wielu w jedną złożoną.
 - Warunkowe klatki wyjść: Owale, opisuje przypisanie do sygnałów. Umieszczone są one na ścieżkach wyjściowych ze stanu. są odpowiadające wyjściom w Mealy'm.

4.1.2 Przykład

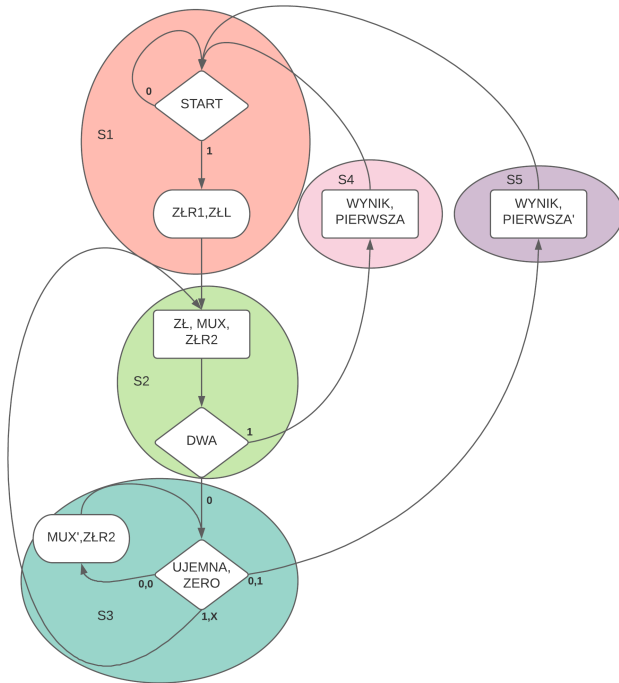


Rysunek 4.1: Blok sterujący.



Rysunek 4.2: Diagram ASM dla bloku sterującego.

4.1.3 Kod 1 z N



$$\begin{aligned}
 D_{S_1} &= S_1 * START' + S_4 + S_5 \\
 D_{S_2} &= S_1 * START + S_3 * UJEMNA \\
 D_{S_3} &= S_2 * DWA' + S_3 * UJEMNA' * ZERO' \\
 D_{S_4} &= S_2 * DWA \\
 D_{S_5} &= S_3 * UJEMNA' * ZERO
 \end{aligned}$$

$$\begin{aligned}
 ZLR1 &= S_1 * START \\
 ZLL &= S_1 * START \\
 ZL &= S_2 \\
 MUX &= S_2 \\
 ZLR2 &= S_2 + S_3 * UJEMNA' * ZERO' \\
 WYNIK &= S_4 + S_5 \\
 PIERWSZA &= S_4
 \end{aligned}$$

Rysunek 4.3: Przykładowy diagram AMS.

4.2 Mikrokod

4.2.1 Notacja

A_i - Adres obecnej instrukcji,
 Z - Przypisanie wektora sterującego,
 Z_{A_i} - Wektor sterujący dla danej instrukcji,
 x_c - warunek,
 A^+ - Przypisanie stanu następnego,
 A_j - Adres skoku,
 A_{i+1} - Adres następnej instrukcji .

Z_{x_c} - Wektor sterujący dla danego stanu przypisywany, jeżeli warunek x_c jest spełniony

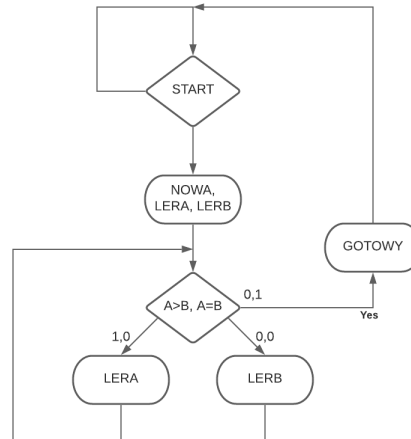
$Z_{x'_c}$ - Wektor sterujący dla danego stanu przypisywany, jeżeli warunek x_c nie jest spełniony

Warto zaznaczyć że w obu rodzajach kodu w else-ie ZAWSZE skaczemy do następnego adresu instrukcji (np. $A_0 \rightarrow A_1$)

4.2.2 Semantyka mikrokodu (Moore'a)

$$A_i : Z = Z_{A_i} \text{ if } x_c \text{ then } A^+ = A_j \text{ else } A^+ = A_{i+1}$$

4.2.3 Przykładowy mikrokod (Moore'a)

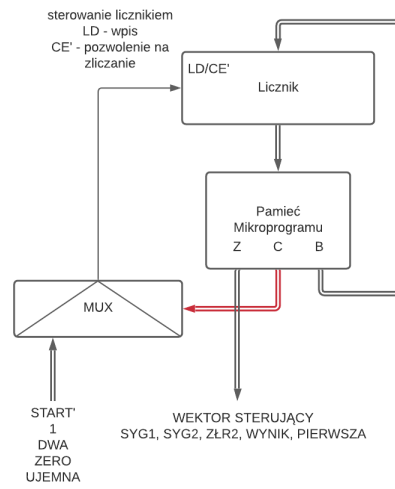


Rysunek 4.4: Diagram ASM dla układu sterowania NWD.

Dla tego grafu kod wygląda tak:

```
A0: Z=0 if START` A0 else A+=Ai+1
A1: Z=NOWA,LERA,LERB if 1 A+=A2 else A+=Ai+1
A2: Z=0 if A=B A+=Agotowy else A+=Ai+1 # Agotowy = A6
A3: Z=0 if A>B A+=ALera else A+=Ai+1 # Alera = A5
A4: Z=LerB if 1 A+=A2 else A+=Ai+1
A5: Z=LerA if 1 A+=A2 else A+=Ai+1
A6: Z=GOTOWY if 1 A+=A0 else A+=Ai+1
```

4.2.4 Przykład układu wykonującego mikrokod (Moore'a)

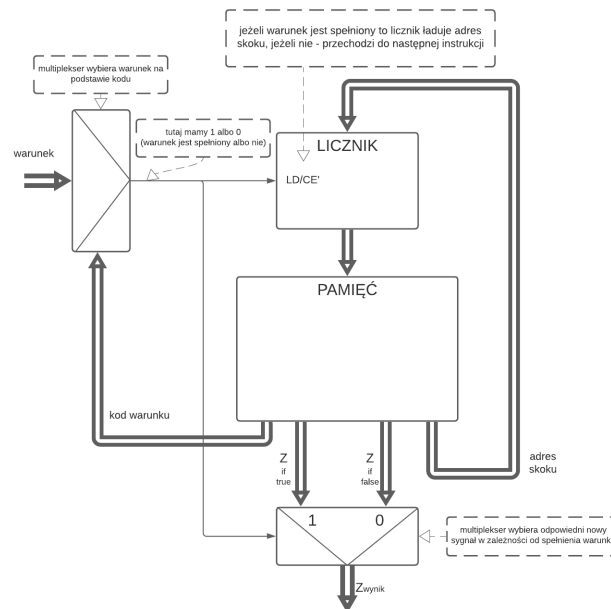


Rysunek 4.5: Struktura mikroprogramowalnego układu sterującego dla automatu typu Moore'a.

4.2.5 Semantyka mikrokodu (Mealy'ego)

$$A_i : \text{if } x_c \text{ then } Z = Z_{x_c}; A^+ = A_j \text{ else } Z = Z_{x_c}; A^+ = A_{i+1}$$

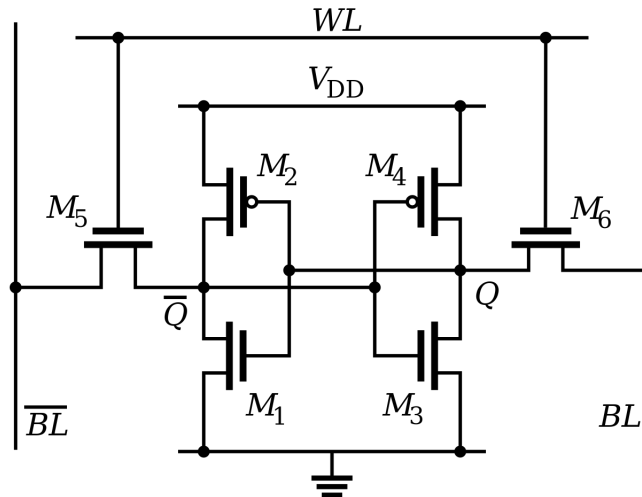
4.3 Przykład układu wykonującego mikrokod (Mealy'ego)



Rysunek 4.6: Struktura mikroprogramowalnego układu sterującego dla automatu typu Mealy'ego.

5 Pamięć półprzewodnikowe

5.1 Pamięć statyczna SRAM



Rysunek 5.1: Komórka pamięci **SRAM** złożona z 6 tranzystorów.

5.1.1 Własności

- przechowywanie informacji w układzie zbudowanym z tranzystorów, dwa inwertery zwarte naprzemiennie wej-wyj \rightarrow układ stabilny,
- cykl działa (odczyt, zapis) krótszy niż dla pamięci dynamicznych - **DRAM**,
- po dostępie gotowość natychmiastowa do realizacji kolejnych cykli dostępu,
- wykorzystywane (ze względu na koszt) głównie jako szybkie pamięci podręczne procesora.

5.2 Pamięć dynamiczna DRAM

5.2.1 Własności

- przechowywanie informacji w kondensatorze,
- odczyt niszczy zapisaną informację i konieczne jest jej ponowne zapisanie,
- przed odczytem należy zapisaną informację zregenerować - uzupełnić ładunek znikający ze względu na upływność kondensatora,
- cykliczna regeneracja ładunku - odświeżenie zawartości pamięci,
- dłuższe dla **DRAM** (nie w przypadku **SRAM**) cykle działania pamięci,
- powierzchnia krzemu dla **DRAM** równa ok $\frac{1}{4}$ powierzchni **SRAM** o tej samej pojemności \rightarrow ok $4 \times$ tańsza,
- wykorzystywane jako duże pamięci operacyjne systemu komputerowego.

6 Przykładowe zadania

6.1 Zestaw 0

6.1.1 Zadanie 1

Proszę zaproponować implementację układu sterowania opisanego diagramem ASM z rysunku 1.

Proszę wybrać implementację jako:

- A - układ mikroprogramowalny
- B - układ sekwencyjny zbudowany z rejestru (kodowanie stanów kodem 1z N) i bramek

W przypadku A należy:

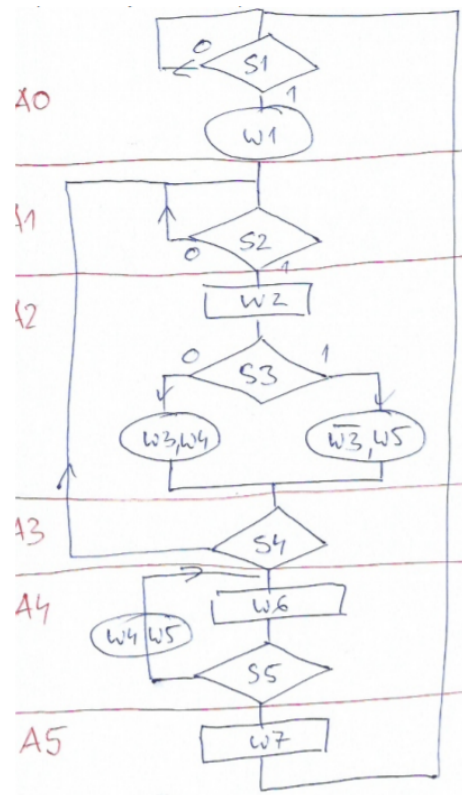
Zaproponować kod: instrukcje z proponowanymi wartościami pól: typ instrukcji, warunek, adres skoku , sterowanie.

- schematu układu mikroprogramowanego pozwalającego wykonać proponowany kod (uwzględnienie liczby rozkazów i wielkości słowa rozkazowego),

- oraz czytelnie w kodzie szesnastkowym zapisać postać binarną pierwszych 4 rozkazów.

W przypadku B należy:

Zapisać funkcje wzbudzeń przerzutników D oraz funkcje wyjść



Rysunek 6.1: Rysunek do zadania 1.

Mikrokod - wariant A

```
A0: if S1' then Z=0; A+=A0 else Z=W1; A1
A1: if S2' then Z=0; A+=A1 else Z=0; A2
A2: if S3 then Z=W2,W3,W5; A+=A3 else Z=W2,W3,W4; A3
A3: if S4' then Z=0; A+=A1 else Z=0; A4
```

A4: **if** S5' **then** Z=W4,W5,W6; A+=A4 **else** Z=W6; A5
A5: **if** 1 **then** Z=W7; A+=A0

Warunek	Kod warunku
S1'	000
1	001
S2'	010
S3	011
S4'	100
S5'	101

Tablica 15: Warunki przejść.

Znaczenie bitów wektora sterowania							
W1	W2	W3	W3'	W4	W5	W6	W7

Tablica 16: Znaczenie bitów wektora sterowania

Adres Pamięci	Bity słowa rozkazowego				
	Kod Warunku	Sterowanie	Adres skoku	Sterowanie	Słowo Pamięci
000	000	00000000	000	10000000	000080
001	010	00000000	001	00000000	100100
010	011	01010100	011	01101000	1AA368
011	100	00000000	001	00000000	200100
100	101	00001110	100	00000010	287402
101	001	00000001	000	00000000	080810

Tablica 17: Słowo rozkaz.

6.1.2 Przerzutniki D - wariant B

$$D_{A_0} = A_0 * S'_1 + A_5$$

$$D_{A_1} = A_0 * S_1 + A_1 * S'_2 + A_3 * S'_4$$

$$D_{A_2} = A_1 * S_2$$

$$D_{A_3} = A_2$$

$$D_{A_4} = A_3 * S_4 + A_4 * S'_5$$

$$D_{A_5} = A_4 * S_5$$

$$W_1 = A_0 * S_1$$

$$W_2 = A_2$$

$$W_3 = A_2 * S'_3$$

$$W'_3 = A_2 * S_3$$

$$W_4 = A_2 * S'_3 + A_4 * S'_5$$

$$W_5 = A_2 * S_3 + A_4 * S'_5$$

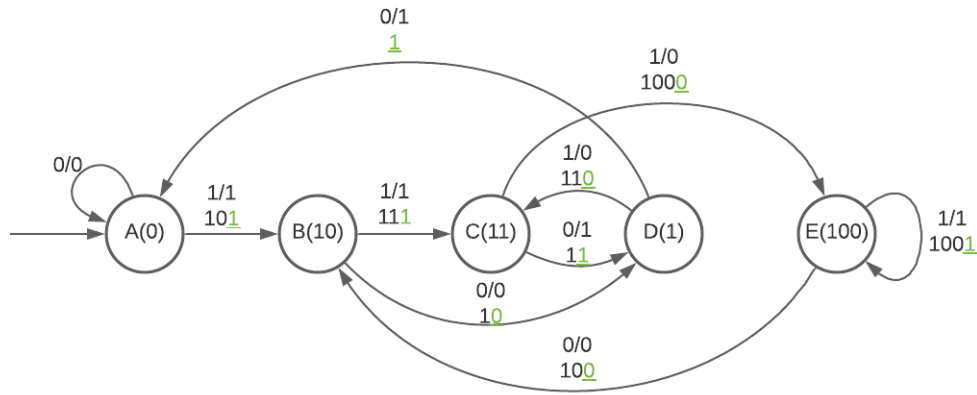
$$W_6 = A_4$$

$$W_7 = A_5$$

6.1.3 Zadanie 2

Proszę zaprojektować automat z jednym wejściem i jednym wyjściem. Na wejściu pojawia się szeregowo liczba w NKB od najmłodszego bitu. Na wyjściu ma się pojawić liczba 5 razy większa (automat typu Mealy'ego)
Zakres zadania:

- graf automatu o minimalnej liczbie stanów (5 stanów),
- ewentualna minimalizacja liczby stanów jeśli graf jest większy.



Rysunek 6.2: Automat typu Mealy'ego mnożący podaną liczbę razy 5.

6.1.4 Zadanie 3

Projekt układu cyfrowego na poziomie przesłań międzyrejestrowych.

Układ cyfrowy posiada następujące wejścia:

1. Start – sygnał uruchomienia testowany w stanie początkowym układu sterowania
2. Nowa.liczba – sygnał informujący o nowej liczbie na wejściu
3. Liczba – sygnały tworzące liczbę zapisaną w kodzie binarnym uzupełnieniowym

Układ cyfrowy posiada następujące wyjścia:

4. GOTOWY - Sygnał gotowości wyniku przetwarzania
5. LICZBA POBRANA – sygnał informujący o przeanalizowaniu liczby z wejścia
6. WYNIK1 – sygnały tworzące liczbę zapisaną w kodzie binarnym uzupełnieniowym będącą wynikiem obliczeń

CZĘŚĆ A

Proszę określić liczbę rejestrów (również liczników) używanych do przechowywania informacji oraz znaczenie przechowywanej informacji.

Proszę określić wszystkie operacje przesłań między rejestrów realizowane na poszczególnych rejestrach.

Proszę zaproponować schemat układu wykonawczego z czytelnym określeniem typu każdego elementu na schemacie. Należy użyć: rejestry, multipleksery, liczniki, jeden komparator i jeden sumator.

Proszę opisać znaczenie wszystkich sygnałów sterujących oraz sygnałów stanu układu.

CZĘŚĆ B

Proszę zaprojektować diagram ASM (DASM)układu sterującego zaproponowaną ścieżką danych.

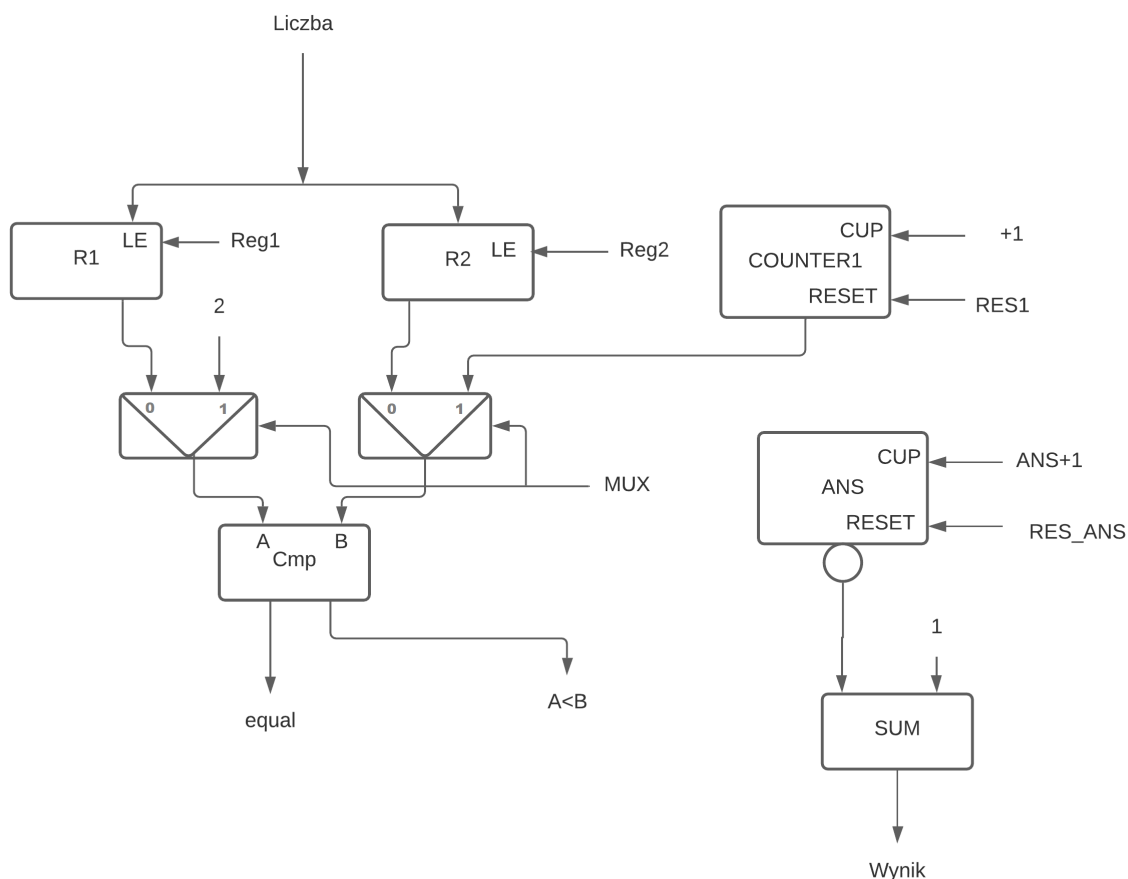
Proszę w diagramie czytelnie wyodrębnić stany układu sekwencyjnego oraz określić kształtem typ występujących w stanie bloków składowych.

Proszę czytelnie opisać znaczenie występujących w DASM sygnałów oraz wartości sygnałów, które decydują o przejściu do kolejnego stanu.

ZAKRES PRACY UKŁADU CYFROWEGO:

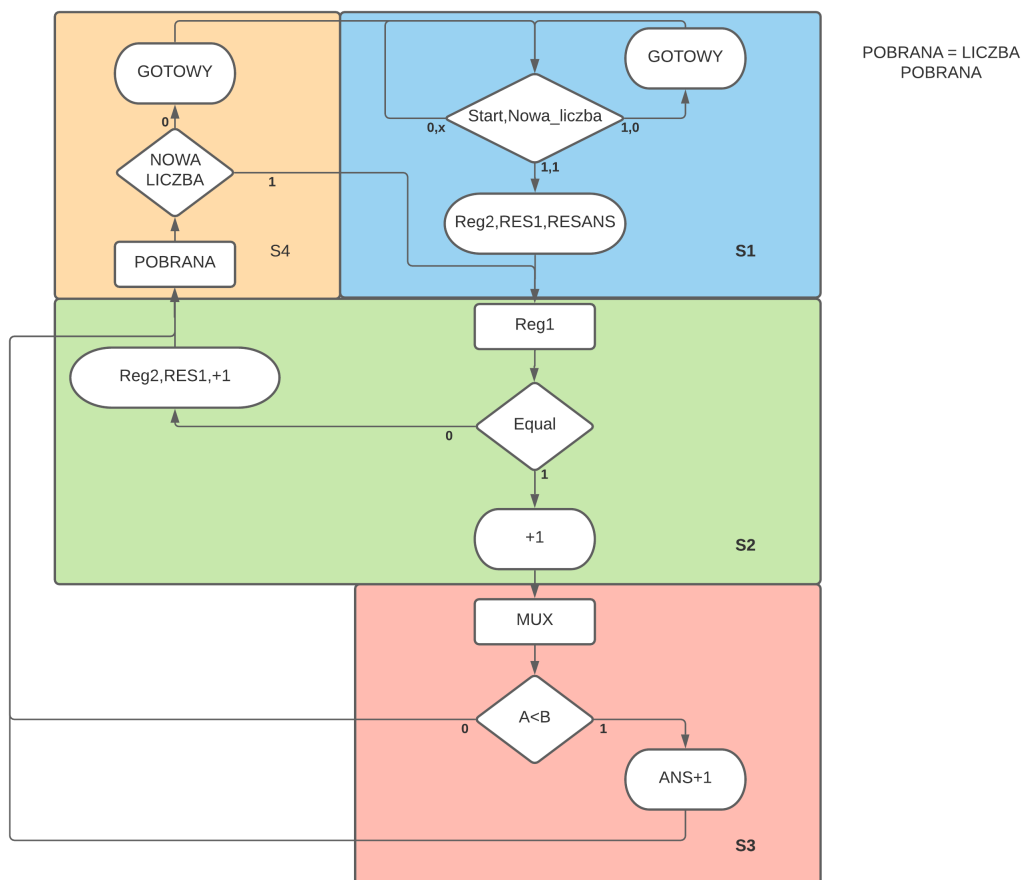
Układ analizuje N liczb w kodzie uzupełnieniowym (dodatnie i ujemne) z wejścia. Na wyjściu dostarcza informację ile razy w ciągu wejściowym pojawiły się trójki jednakowych liczb bezpośrednio po sobie. W ciągu 1,2,2,2,2,3,3,3,1,2,1 trójki pojawiają się 3 razy: 2,2,2; 2,2,2; oraz 3,3,3. Układ nie sprawdza przepełnienia zakresu obliczanych liczb.

CZĘŚĆ A:



Rysunek 6.3: Schemat układu wykonawczego.

CZĘŚĆ B:



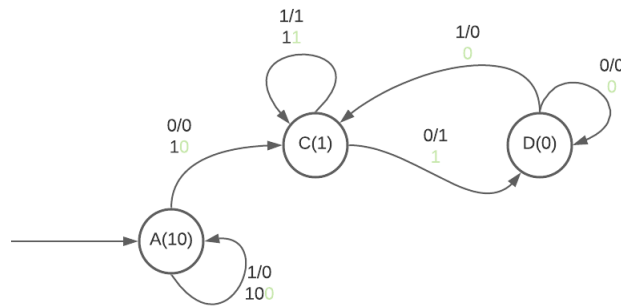
Rysunek 6.4: Diagram ASM do zadania 3.

6.2 Zestaw 1

6.2.1 Zadanie 1

Dla przygotowanego wcześniej graf i tablicy przejść automatu z pierwszej części zadań:

- Proszę zaprojektować automat Mealego realizujący operację dodawania do liczby (podawanej w NKB szeregowo na wejście od najmłodszego bitu automatu) liczby o 2 większej od podawanej liczby. UWAGA: AUTOMAT MA JEDNO WEJŚCIE INFORMACYJNE, NA KTÓRYM POJAWIA SIĘ BADANA LICZBA OD NAJMŁODSZEGO BITU. Automat ma jedno wejście resetujące go do stanu początkowego, wejście pozwala wystartować pracę automatu od stanu początkowego. wykonać dwie implementacje w postaci:
 - Przerzutników i bramek – stany proszę zakodować w NKB, proszę określić postać funkcji wzbudzeń i funkcji wyjść – BEZ MINIMALIZACJI.
 - Rejestru i pamięci ROM – należy narysować schemat oraz określić rozmiar i zawartość pamięci ROM.



Rysunek 6.5: Automat Mealy'ego $2n+2$ w NKB.

Stan	Wejście		Wyjście	
	0	1	0	1
A (00)	C (01)	A (00)	0	0
C (01)	D (11)	C (01)	1	1
D (11)	D (11)	C (01)	0	0

Tablica 18: Tabela przejść automatu.

Adres		Dane	
Stan	Wejście	Wyjście	Stan*
00	0	0	01
00	1	0	00
01	0	1	11
01	1	1	01
11	0	0	11
11	1	0	01
10	0	DC	DC
10	1	DC	DC

Tablica 19: ROM = 24bity pamięci.

Funkcje przejść

$$D_a = \Sigma(2, 6) + d(4, 5)$$

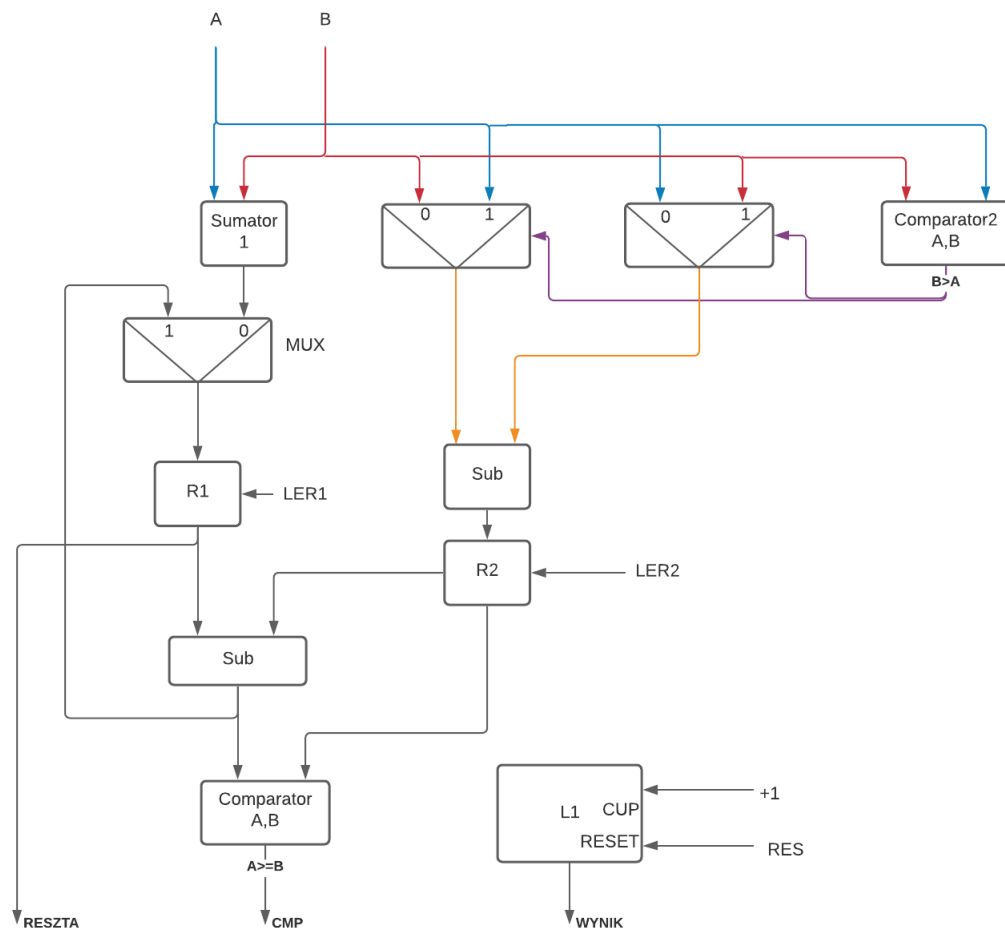
$$D_b = \Sigma(0, 2, 3, 6, 7) + d(4, 5)$$

$$W(Q_a, Q_b) = \Sigma(2, 3) + d(4, 5)$$

6.2.2 Zadanie 2

Ścieżka przetwarzania danych dla projektu realizowanego na poziomie przesłań między rejestrowych

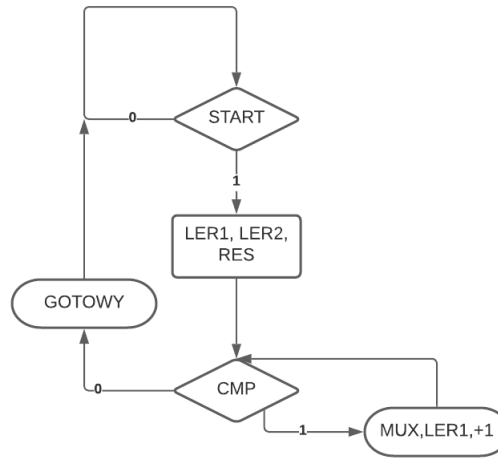
- Opis działania układu:
Za każdym razem, gdy układ zostanie włączony pobierze z wejść dwie wielobitowe liczby A i B dodatnie (zapisane w NKB). Proszę zaprojektować część wykonawczą układu, który podzieli sumę liczb A i B przez dodatnią różnicę tych liczb i na wyjściu poda całkowity wynik dzielenia oraz resztę z dzielenia. Układ może zostać zbudowany z podstawowych układów sekwencyjnych i kombinacyjnych takich jak: bramki, komparatory, sumatory, multipleksery, liczniki i rejestry.
- Zadania do wykonania:
 1. Proszę narysować schemat z określeniem: układów składowych, sposobu połączenia układów i sposobu sterowania układami.
 2. Proszę określić znaczenie sygnałów stanu oraz sygnałów sterujących pracą układu.
 3. Proszę krótko słowami w punktach opisać przebieg przetwarzania.



Rysunek 6.6: Schemat blokowy.

6.2.3 Zadanie 3

- **Diagram ASM** dla układu sterującego ścieżką przetwarzania danych zaprojektowaną w rozwiązaniu zadania 2. Ustalenie zakresu i liczby stanów układu.
- Wyznaczenie funkcji wzbudzeń przerzutników i funkcji wyjść na podstawie diagramu ASM przy założeniu, że stany są kodowane kodem jeden z N.
- Wyznaczenie kodu dla implementacji układu sterowania w oparciu o automat mikroprogramowalny. Do wykorzystania są typowe mikroinstrukcje układu mikroprogramowalnego posiadające pola: sterowania, warunku, adresu i typu rozkazu. Narysować schemat układu mikroprogramowalnego umożliwiającego wykonanie przygotowanego kodu.



Rysunek 6.7: Diagram ASM.

$$D_{S_1} = S_1 * START' + S_2 * (A \geq B)'$$

$$D_{S_2} = S_1 * START + S_2 * (A \geq B)$$

$$LER_1 = S_1 * START + S_2 * (A \geq B)$$

$$LER_2 = S_1 * START$$

$$RES = S_1 * START$$

```

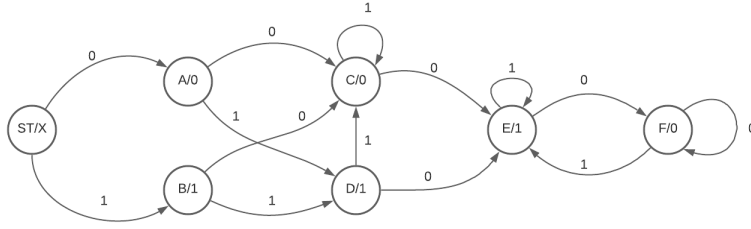
A0: Z=0 if START' A+=A0 else A+=Ai+1
A1: Z=LER1, LER2, RES if 1 A+=A2 else A+=Ai+1
A2: Z=0 if A>=B' A+=A0 else A+=Ai+1
A3: Z=LER1 if 1 A+=A2 else A+=Ai+1

```

6.3 Zestaw 2

6.3.1 Zadanie 1

- Automat Moore +4 w NKB. Wejście od najmłodszego bitu automatu. Automat ma jedno wejście resetujące go do stanu początkowego, wejście pozwala wystartować pracę automatu od stanu początkowego.



Rysunek 6.8: Automat Moore’a NKB+4.

- Implementacja przerzutników i bramek – stany proszę zakodować w NKB, proszę określić postać funkcji wzbudzeń i funkcji wyjść – BEZ MINIMALIZACJI.

kodowanie		0	1	OUT
000	START	001	011	NIE MA
001	A	010	110	0
011	B	010	110	1
010	C	111	010	0
110	D	111	010	1
111	E	101	111	1
101	F	101	111	0
100	G	DC	DC	DC

Tablica 20: Przejścia i wyjścia

$$D_1 = \Sigma(3, 7, 4, 12, 14, 10, 11, 15)$$

$$D_2 = \Sigma(1, 2, 3, 6, 7, 12, 13, 15, 11)$$

$$D_3 = \Sigma(0, 1, 4, 12, 5, 13, 15, 11)$$

$$OUT = \Sigma(3, 6, 7) + d(0, 4)$$

- Implementacja rejestru i pamięci ROM – należy narysować schemat oraz określić rozmiar i zawartość pamięci ROM.

Adres		Dane
Stan	Wejscie	Stan
000	0	001
001	0	010
011	0	010
010	0	111
110	0	111
111	0	101
101	0	101
100	0	DC
000	1	011
001	1	110
011	1	110
010	1	010
110	1	010
111	1	111
101	1	111
100	1	DC

Tablica 21: ROM 1

ROM2	
Stan	Wyjście
000	NIE MA
001	0
011	1
010	0
110	1
111	1
101	0
100	DC

Tablica 22: ROM 2

6.3.2 Zadanie 2

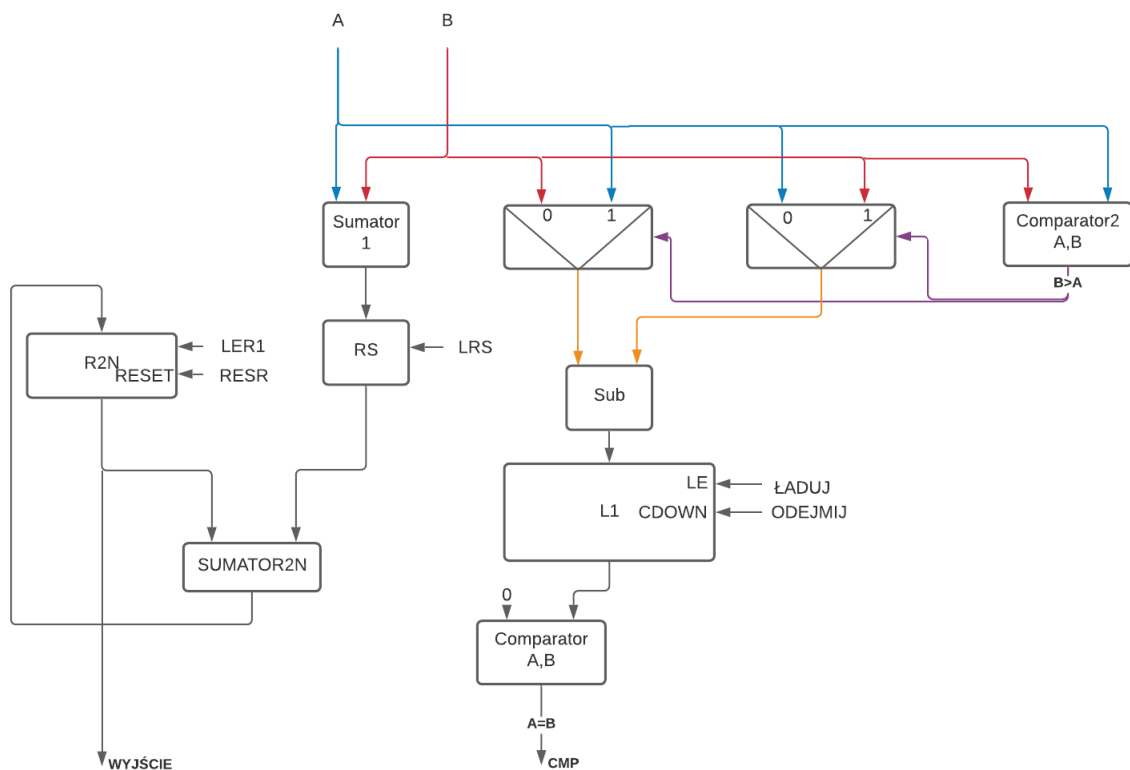
Ścieżka przetwarzania danych dla projektu realizowanego na poziomie przesłań między rejestrowych

Opis działania układu:

Za każdym razem gdy układ zostanie włączony pobierze z wejść dwie wielobitowe liczby A i B dodatnie (zapisane w NKB). Proszę zaprojektować część wykonawczą układu, który pomnoży sumę liczb A i B przez dodatnią różnicę tych liczb. Układ może być zbudowany z podstawowych układów sekwencyjnych i kombinacyjnych takich jak: bramki, komparatory, sumatory, multipleksery, liczniki i rejestry.

Zadania do wykonania:

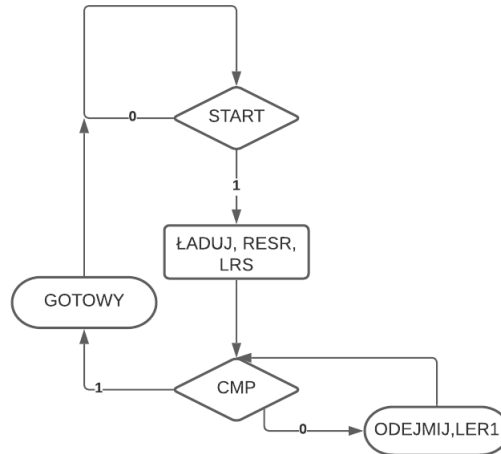
- Proszę narysować schemat z określeniem: układów składowych, sposobu połączenia układów i sposobu sterowania układami,
- Proszę określić znaczenie **sygnałów stanu oraz sygnałów sterujących** pracą układu.
- Proszę krótko słowami **w punktach opisać przebieg przetwarzania**.



Rysunek 6.9: Schemat blokowy.

6.3.3 Zadanie 3

- **Diagrm ASM** dla układu sterującego ścieżką przetwarzania danych zaprojektowaną w rozwiązaniu zadania 2. Ustalenie zakresu i liczby stanów układu.
- Wyznaczenie funkcji wzbudzeń przerzutników i funkcji wyjść na podstawie diagramu ASM przy założeniu, że stany są kodowane kodem jeden z N.
- Wyznaczenie kodu dla implementacji układu sterowania w oparciu o automat mikroprogramowalny. Do wykorzystania są typowe mikroinstrukcje układu mikroprogramowalnego posiadające pola: sterowania, warunku, adresu i typu rozkazu. Narysować schemat układu mikroprogramowalnego umożliwiającego wykonanie przygotowanego kodu.



Rysunek 6.10: Diagram ASM.

$$D_{S_1} = S_2 * CMP + S_1 * START'$$

$$D_{S_2} = S_2 * CMP' + S_1 * START$$

$$ADUJ = S_1 * START$$

$$RESR = S_1 * START$$

$$LRS = S_1 * START$$

$$ODEJMIJ = S_2 * CMP'$$

$$LER_1 = S_2 * CMP'$$

$$GOTOWY = S_2 * CMP$$

Mealy

A0: **if** START' **then** Z = 0; A+=A0 **else** Z=ŁADUJ,RESR,LRS; A+=A1
 A1: **if** CMP' **then** Z = ODEJMIJ,LER1; A+=A1 **else** Z=GOTOWY; A+=A0

Moore

A0: Z = 0 **if** START' **then** A+=A0 **else** A+=Ai+1
 A1: Z = ŁADUJ,RESR,LRS **if** 1 **then** A+=A2 **else** A+=Ai+1
 A2: Z = 0 **if** CMP' **then** A+=A4 **else** A+=Ai+1
 A3: Z = GOTOWY **if** 1 **then** A+=A0 **else** A+=i+1
 A4: Z = ODEJMIJ,LER1 **if** 1 **then** A+=A2 **else** A+=Ai+1

6.4 Zestaw 3

6.4.1 Zadanie 1

Dla przygotowanego wcześniej graf i tablicy przejść automatu z pierwszej części zadań:

- Proszę zaprojektować automat Moore'a realizujący operację dodawania do liczby (podawanej w NKB szeregowo na wejście od najmłodszego bitu automatu) liczby o 1 większej od podawanej liczby. UWAGA: AUTOMAT MA JEDNO WEJŚCIE INFORMACYJNE, NA KTÓRYM POJAWIA SIĘ BADANA LICZBA OD NAJMŁODSZEGO BITU. Automat ma jedno wejście resetujące go do stanu początkowego, wejście pozwala wystartować pracę automatu od stanu początkowego

Wykonać dwie implementacje w postaci:

- przerzutników i bramek – stany proszę zakodować w NKB, proszę określić postać funkcji wzbudzeń i funkcji wyjść – BEZ MINIMALIZACJI.
- rejestru i pamięci ROM – należy narysować schemat oraz określić rozmiar i zawartość pamięci ROM.

	Wejście		
Stan	0	1	Wyjście
A(000)	010	001	X
B(001)	010	001	1
C(010)	011	100	1
D(011)	011	100	0
E(100)	010	001	0

Tablica 23: Tablica przejść.

Adres		Dane
Stan	Wejście	Stan++
000	0	010
001	0	010
010	0	011
011	0	011
100	0	010
000	1	001
001	1	001
010	1	100
011	1	100
100	1	001

Tablica 24: ROM 1.

Stan	Wyjście
000	X
001	1
010	1
011	0
100	0

Tablica 25: ROM 2.

6.4.2 Zadanie 2

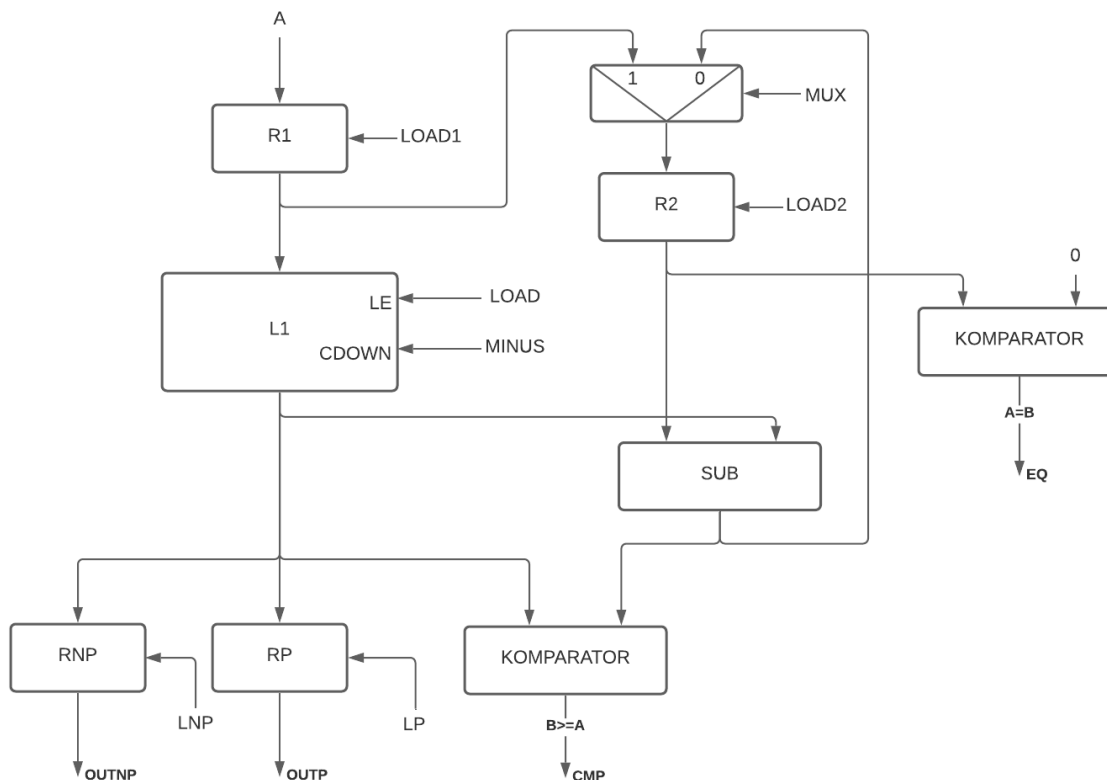
Ścieżka przetwarzania danych dla projektu realizowanego na poziomie przesłań między rejestrowych
Opis działania układu:

Za każdym razem gdy układ zostanie włączony pobierze z wejść dwie wielobitowe liczby A i B dodatnie (zapisane w NKB). Proszę zaprojektować część wykonawczą układu, który pomnoży sumę liczb A i B przez dodatnią różnicę tych liczb. Układ może być zbudowany z podstawowych układów sekwencyjnych i kombinacyjnych takich jak: bramki, komparatory, sumatory, multipleksery, liczniki i rejestry.

Zadania do wykonania:

- Proszę narysować schemat z określeniem: układów składowych, sposobu połączenia układów i sposobu sterowania układami.
- Proszę określić **znaczenie sygnałów stanu oraz sygnałów sterujących** pracą układu.
- Proszę krótko słowami **w punktach opisać przebieg przetwarzania**.

Rysunek 6.12: Diagram ASM (Nie chciało nam się już robić).



Rysunek 6.11: Schemat blokowy.

6.4.3 Zadanie 3

- **Diagram ASM** dla układu sterującego ścieżką przetwarzania danych zaprojektowaną w rozwiązaniu zadania 2. Ustalenie zakresu i liczby stanów układu.
- Wyznaczenie funkcji wzbudzeń przerzutników i funkcji wyjść na podstawie diagramu ASM przy założeniu, że stany są kodowane kodem jeden z N.
- Wyznaczenie kodu dla implementacji układu sterowania w oparciu o automat mikroprogramowalny. Do wykorzystania są typowe mikroinstrukcje układu mikroprogramowalnego posiadające pola: sterowania, warunku, adresu i typu rozkazu. Narysować schemat układu mikroprogramowalnego umożliwiającego wykonanie przygotowanego kodu.

Literatura

- [1] Graf automatu. <https://www.youtube.com/watch?v=Ni5rIqIznV0>
- [2] Automaty. <http://cygnus.et.put.poznan.pl/mkrasic/Lab7.pdf>

masz trudniej u Walkowiaka?
no trudno "życie jest niesprawiedliwe"

kiedy ktoś nie wyśle gotowego
skryptu rozwiązującego zadanie z egzaminu



Rysunek 6.13: meme.