Materiały pomocnicze do ćwiczeń z podstaw techniki cyfrowej (przygotował R.Walkowiak)
ZADANIE 1

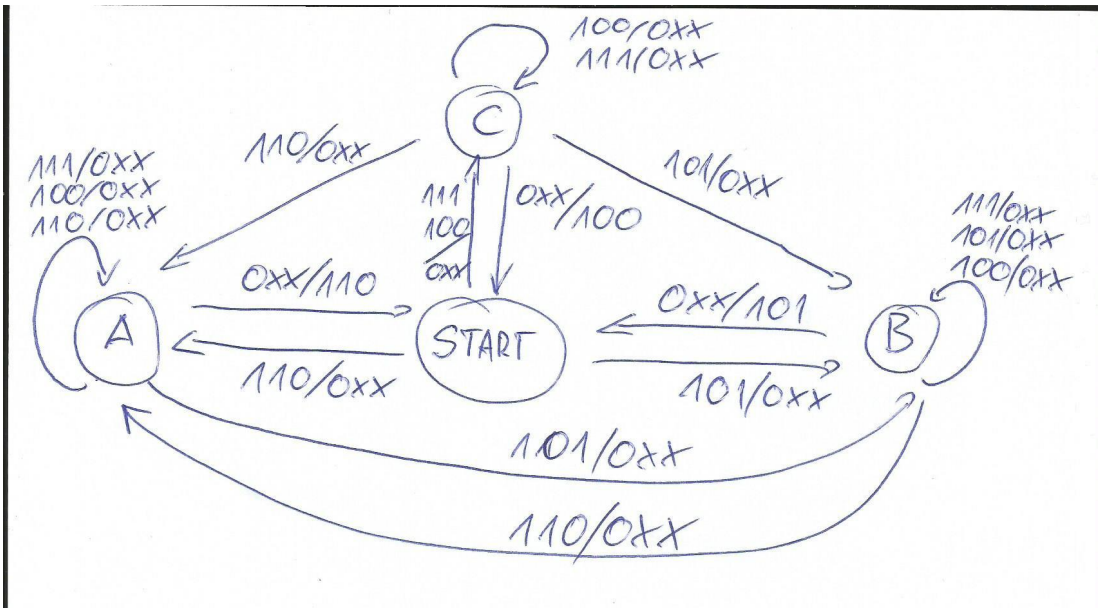
Komparator szeregowy 2 liczb

Specyfikacja wymagań dla układu

Bity podawane na wejściu od najmniej znaczącego bitu
Wejścia od najstarszego: liczba, bitA, bitB
Wyjścia od najstarszego: wynik, A>B,B>A

Liczba =1 ozancza podawanie kolejnych bitów próbkowanych zgodnie z taktem sygnału zegarowego, Liczba=0 oznacza, że ostatni bit został podany i należy wygenerować wynik (dostępny w czasie trwania jednego stanu), a następnie przejść do analizy kolejnej liczby.

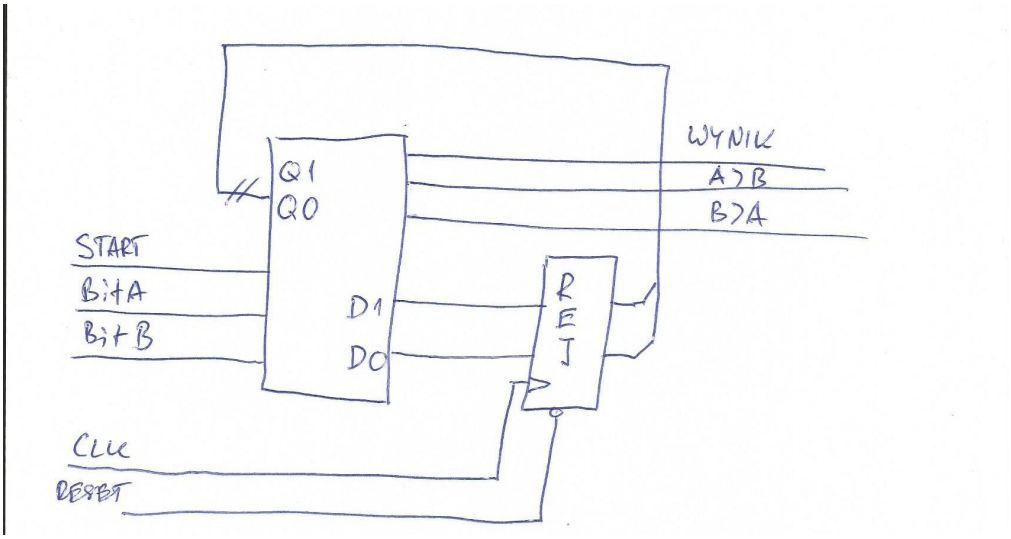
Automat typu Mealy'ego



Rys.1 Graf stanów automatu „komparator szeregowy" typu Mealy'ego

Stan/wejścia	000	001	010	011	100	101	110	111
start	start	start	start	start	C	B	A	C
A	start	start	start	start	A	B	A	A
B	start	start	start	start	B	B	A	B
C	start	start	start	start	C	B	A	C
Stan/wyjścia								
Start	0--	0--	0--	0--	0--	0--	0--	0--
A	110	110	110	110	0--	0--	0--	0--
B	101	101	101	101	0--	0--	0--	0--
C	100	100	100	100	0--	0--	0--	0--

Tab.1 Tablica przejść i wyjść automatu „komparator szeregowy" typu Mealy'ego



Rys.2 Realizacja automatu „komparator szeregowy” typu Mealy’ego

Stan	Kod (Q1,Q0)
Start	00
A	01
B	11
C	10

Tab. 2 Kodowanie stanów „komparator szeregowy” typu Mealy’ego

Stan/wejścia	000	001	010	011	100	101	110	111
00	00	00	00	00	10	11	01	10
01	00	00	00	00	01	11	01	01
11	00	00	00	00	11	11	01	11
10	00	00	00	00	10	11	01	10
Stan/wyjścia								
00	0--	0--	0--	0--	0--	0--	0--	0--
01	110	110	110	110	0--	0--	0--	0--
11	101	101	101	101	0--	0--	0--	0--
10	100	100	100	100	0--	0--	0--	0--

Tab.3 Kodowana tablica przejść i wyjść automatu „komparator szeregowy” typu Mealy’ego

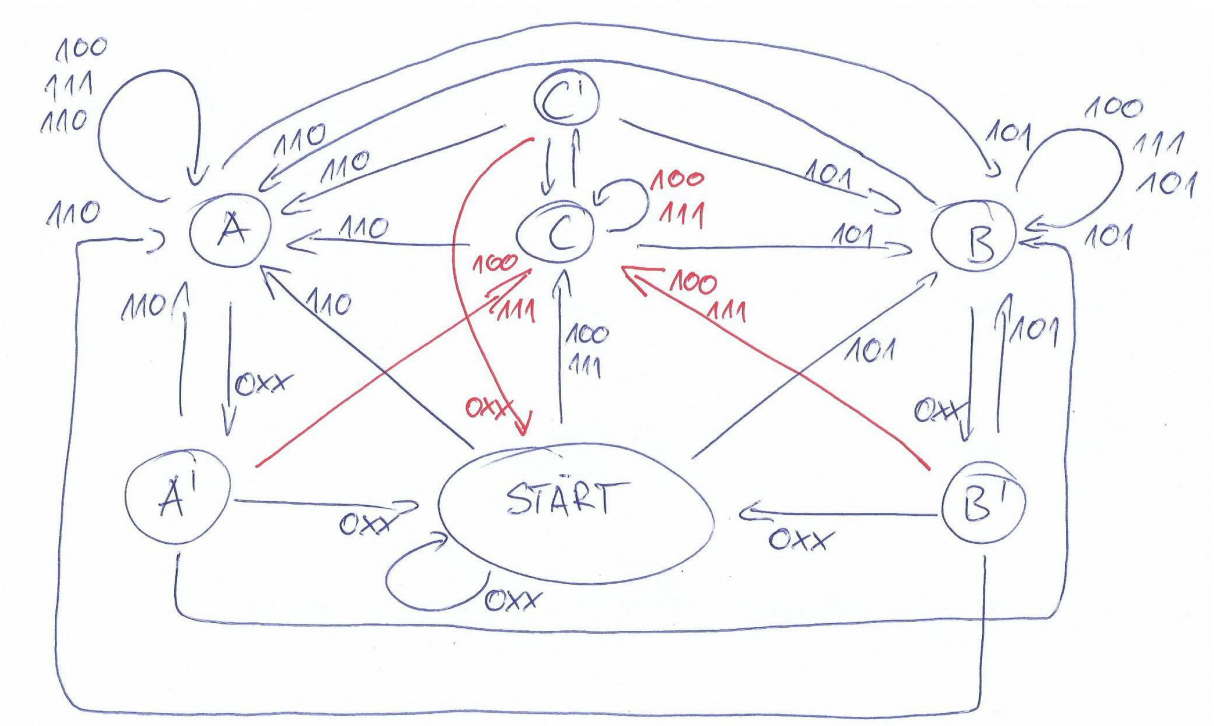
Funkcje Wzbudzeń i wyjść (wejścia podano w kolejności od najwyższej wagi):
D1(Q1,Q0,liczba,bita,bitB)=∑(4,5,7,13,20,21,23,28,29,31)
D0(Q1,Q0,liczba,bita,bitB)=∑(5,6,12-15,21,22,28-31)
Wynik(Q1,Q0,liczba,bita,bitB)=∑(8-11,24-27,16-19)
A>B(Q1,Q0,liczba,bita,bitB)=∑(8-11)+d(0-7,12-15,20-23,28-31)
B>A(Q1,Q0,liczba,bita,bitB)=∑(24-27)+d(0-7,12-15,20-23,28-31)

Zawartość ROM dla implementacji automatu „komparator szeregowy” typu Mealy’ego
Adresy=(Q1,Q0,liczba,bita,bitB)
Dane=(D1,D0,Wynik,A>B,B>A)

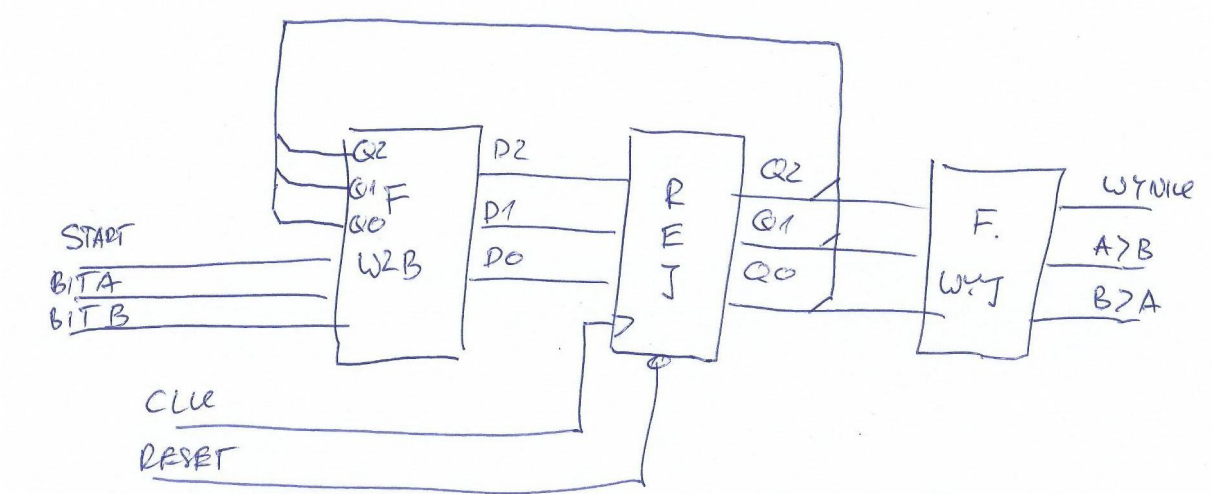
Adres	Dane
0-3	000--
4	100--
5	110--
6	010--
7	100--
8-11	00110
12	010--
13	110--
14	010--
15	010--
16-19	00100
20	100--
21	110--
22	010--
23	100--
24-27	00101
28	110--
29	110--
30	010--
31	110--

Tab.4 Zawartość pamięci dla implementacji automatu „komparator szeregowy” typu Mealy’ego,” –„ oznacza wartość dowolną

Automat Moore’a
Komparator szeregowy 2 liczb (ta sama specyfikacja)



Rys.3 Graf stanów automatu „komparator szeregowy” typu Moore’a, X oznacza stan dowolny wejścia



Rys4. Realizacja automatu „komparator szeregowy” typu Moora

Stan/wejście	000	001	010	011	100	101	110	111	WYJŚCIA
Start	Start	Start	Start	Start	C	B	A	C	0--
A	A’	A’	A’	A’	A	B	A	A	0--
B	B’	B’	B’	B’	B	B	A	B	0--
C	C’	C’	C’	C’	C	B	A	C	0--
A’	Start	Start	Start	Start	C	B	A	C	110
B’	Start	Start	Start	Start	C	B	A	C	101
C’	Start	Start	Start	Start	C	B	A	C	100

Tab.5 Kodowana tablica przejść i wyjść automatu „komparator szeregowy” typu Moore’a

Stan	Kod (Q2,Q1,Q0)
Start	000
A	001
B	011
C	010
A’	110
B’	111
C’	101

Tab. 6 Kodowanie stanów „komparator szeregowy” typu Moore’a

Stan/wejście	000	001	010	011	100	101	110	111	WYJŚCIA
000	000	000	000	000	010	011	001	010	0--
001	110	110	110	110	001	011	001	001	0--
011	111	111	111	111	011	011	001	011	0--
010	101	101	101	101	010	011	001	010	0--
110	000	000	000	000	010	011	001	010	110
111	000	000	000	000	010	011	001	010	101
101	000	000	000	000	010	011	001	010	100

Tab.7 Zakodowana tablica przejść i wyjść automatu „komparator szeregowy” typu Moore’a

Funkcje Wzbudzeń i Wyjść automatu „komparator szeregowy” typu Moore’a (wejścia podano w kolejności od najwyższej wagi):

D2(Q2,Q1,Q0,liczba,bita,bitB)=∑(8-11,16-19,24-27)+d(32-39)

D1(Q2,Q1,Q0,liczba,bita,bitB)=∑(5,7,8-13,21,23,24-29,31,45,47,53,57,61,63) +d(32-39)

D0(Q2,Q1,Q0,liczba,bita,bitB)=∑(5-6,12-19,21-22,24-31,45-46,53-54,61-62)+d(32-39)

Wynik(Q2,Q1,Q0)=∑(5-7)+d(4)
A>B(Q2,Q1,Q0)=∑(6)+d(0-4)
B>A(Q2,Q1,Q0)=∑(7)+d(0-4)

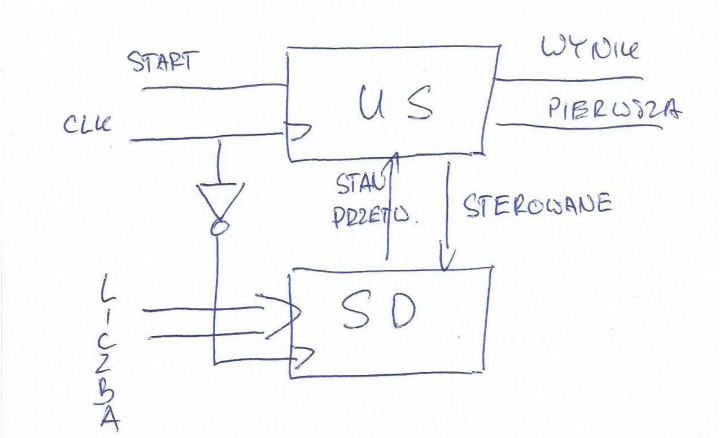
Zawartości ROM (częściowa) dla implementacji automatu „komparator szeregowy" typu Mealy'ego
Adresy=(Q2Q1,Q0,liczba,bita,bitB)
Dane=(D2,D1,D0,Wynik,A>B,B>A)

Adres	Dane
0000--	0000--
000100	0100--
000101	0110--
000110	0010--
000111	0100--
…	…
1100--	000110
110100	010110
110101	011110
110110	001110
110111	010110
1110--	000101
111100	010101
111101	011101
111110	001101
111111	010101

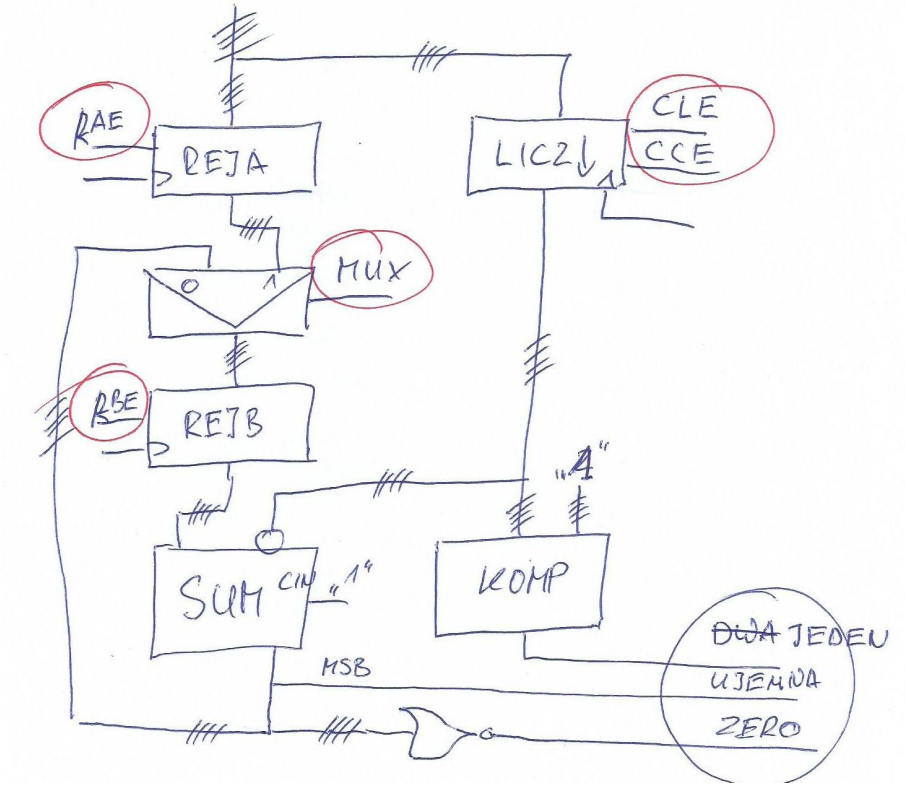
Tab.8 Zawartość pamięci dla implementacji automatu „komparator szeregowy" typu Moore'a, " –„ oznacza wartość dowolną

TEST LICZBY PIERWSZEJ

Układ cyfrowy testujący liczbę, podaje poprawnie wynik dla liczb>=2, po zbadaniu liczby na wyjściu pojawi się sygnałY WYNIK=1 i Pierwsza=1 gdy liczba jest pierwsza lub WYNIK=1 i Pierwsza=0 gdy liczba jest złożona.



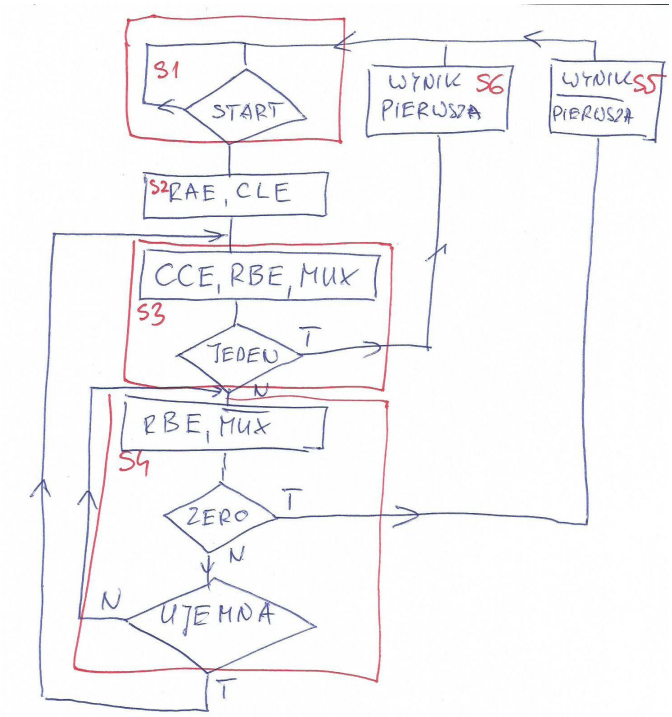
Rys.5 Schemat struktury układu do testu liczby pierwszej (układ sterowania i ścieżka danych)



Rys. 6 Schemat układu operacyjnego do testu liczby pierwszej (układ sterowania i ścieżka danych), w czerwonych okręgach sygnały sterujące, w niebieskim okręgu - stan przetwarzania, wejście danych z góry.

Syganły sterujące: RAE,CLE,CCE,RBE,MUX

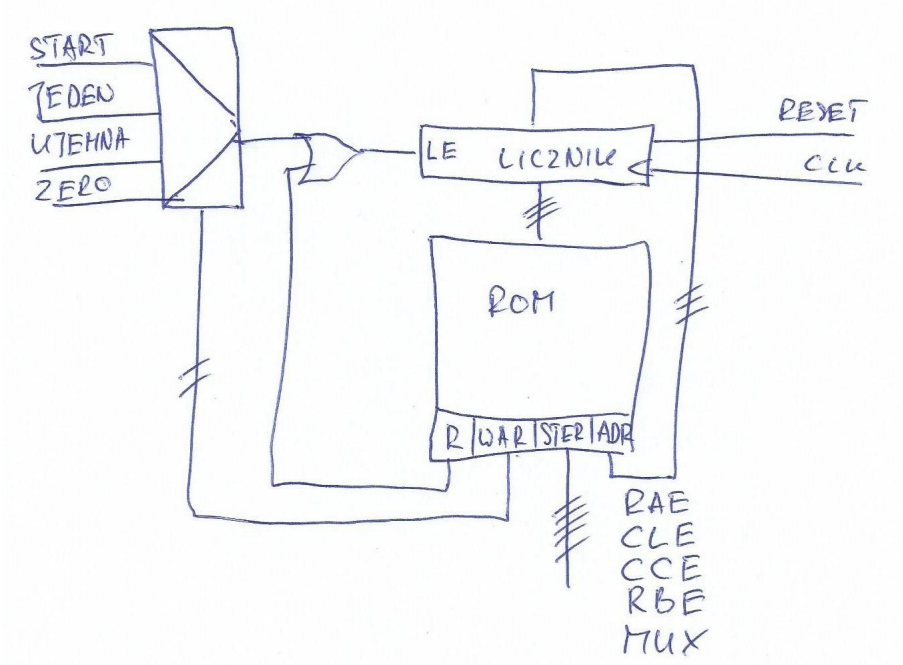
Syganały stanu: JEDEN, UJEMNA, ZERO



Rys. 7 Diagram ASM układ sterowania dla układu testowania liczby pierwszej, wyodrębniono 6 stanów niezbędnych do realizacji oddzielnych kroków sterowania układem wykonawczym i obsługi sygnałów, wejściowych i wyjściowych.

Stan S2 odpowiada za pobranie danych z wejścia układu.
Stan S3 odpowiada za wygenerowanie nowego dzielnika, przywrócenie w rejestrze B dzielnej i test zakresu dzielników.
Stan S4 odpowiada za:

- zapisanie wyniku odejmowania jako etapu dzielenia,
- test zakończenia dzielenia z resztą (UJEMNA) lub bez reszty (ZERO).



Rys.8 Układ mikroprogramowalny dla układu sterowania testowania liczby pierwszej, pole sterowanie zawiera sygnały sterujące i wyjściowe, pole R zawiera kod typu rozkazu.

Układ sterowania wykonany w oparciu o układ mikroprogramowalny implementuje dwa rozkazy:

- Skoku bezwarunkowego R=1 i
- Skoku warunkowego lub przejścia do następnego rozkazu R=0.

Program realizacji układu sterującego składa się z kolejnych rozkazów (z parametrami) umieszczonych w Tab9

Adres pamięci	stan	R	War.	Nr War.	Stan Nast.	Dane pamięci, znaczenie, kolejnych pól R,warunek, sterowanie, wynik,pierwsza, adres,
000	S1	0	Start'	0	S1	0, 00,00000,0,0,000
001	S2	1		--	S3	1,--,1100-,0,0,010
010	S3	0	jedynka	1	S6	0,01,00111,0,0,110
011	S4	0	ujemna	2	S3	0,10,00010,0,0,010
100	S4A	0	Zero'	3	S4	0,11,00000,0,0,011
101	S5	1		--	S1	1,--,00000,1,0,000
110	S6	1		--	S1	1,--,00000,1,1,000

Tab.9 Program układu mikroprogramowalnego dla „testowania liczby pierwszej"

Ze względu na możliwości układu testowania w stanie tylko jednego warunku, stan S4 jest realizowany za pomocą dwóch kolejnych rozkazów: testowania kolejno sygnału UJEMNA, a następnie sygnału ZERO'.

Analiza parametrów czasowych układu „testowania liczby pierwszej „

Zakładamy sterowanie US i SD przeciwnymi fazami zegara pokazane na Rys. 5 i wypełnienie sygnału zegarowego równe ½.

Czas wyprzedzenia układu sterowania dla sygnałów stanu przetwarzania Tsus

Tsus= Tsp+Tpmux_d+Tb (czas wyprzedzenia przerzutnika, czas propagacji multipleksera dla danych, czas propagacji bramki)

Czas propagacji układu sterowania: Tpus

Tpus=Tpp+Taa (czas propagacji przerzutnika i czas dostępu od adresu dla pamięci)

Dla poprawnego uwzględnienia gotowych warunków w multiplekserze konieczne jest uwzględnienie propagacji od adresu multipleksera (wyznaczonego przez układ sterujący):

Tclk > Tpus+Tpmux_a +Tpb+Tsp

Analiza zależności czasowych dla układu wykonawczego:

Stan 2 Zapis danych wejściowych poprawny gdy spełnione będą wymagania czasu wyprzedzenia rejestru Tsrs i licznika Tscs względem zegara dla sygnałów sterujących:

1/2Tclk>Tpus+Tsrs

1/2Tclk>Tpus+Tscs

Stan 3 Zapis danych wejściowych i zliczanie licznika poprawne, gdy spełnione będą wymagania czasu wyprzedzenia rejestru Tsrs i licznika Tscs względem zegara dla sygnałów sterujących:

1/2Tclk>Tpus+Tsrs

1/2Tclk>Tpus+Tscs

Stan 3 Zapis danych do rejestru poprawny gdy dane będą dostępne na wejściu rejestru przed czasem wyprzedzenia danych rejestru Tsr

1/2Tclk>Tpus+Tpmux+Tsr (czas propagacji multipleksera)

Stan 3 Próbkowanie sygnału JEDNYKA poprawne gdy będzie on dostępny dla układu sterowania: czas propagacji licznika (Tpc), czas propagacji komparatora (Tpk):

1/2Tclk>Tpc+Tpk+Tsus

Stan 4 Zapis danych do rejestru poprawny gdy dane będą dostępne na wejściu rejestru przed czasem wyprzedzenia danych rejestru Tsr

1/2Tclk>Tpus+Tpmux+Tsr (czas propagacji multipleksera)

Stan 4 Zapis danych wejściowych do rejestru poprawne, gdy spełnione będą wymagania czasu wyprzedzenia rejestru Tsrs względem zegara dla sygnałów sterujących:

1/2Tclk>Tpus+Tsrs

Zakładając, że w układzie wykonawczym (realizacja mikroprogramowalna) testowany jest wpierw sygnał UJEMNA jego wyznaczenie będzie wąskim gardłem dla zależności czasowych przy realizacji tego stanu. Próbkowanie sygnału UJEMNA poprawne gdy będzie on dostępny dla układu sterowania: czas propagacji rejestru (Tpr), czas propagacji sumatora (Tps):

1/2Tclk>Tpr+Tps+Tsus

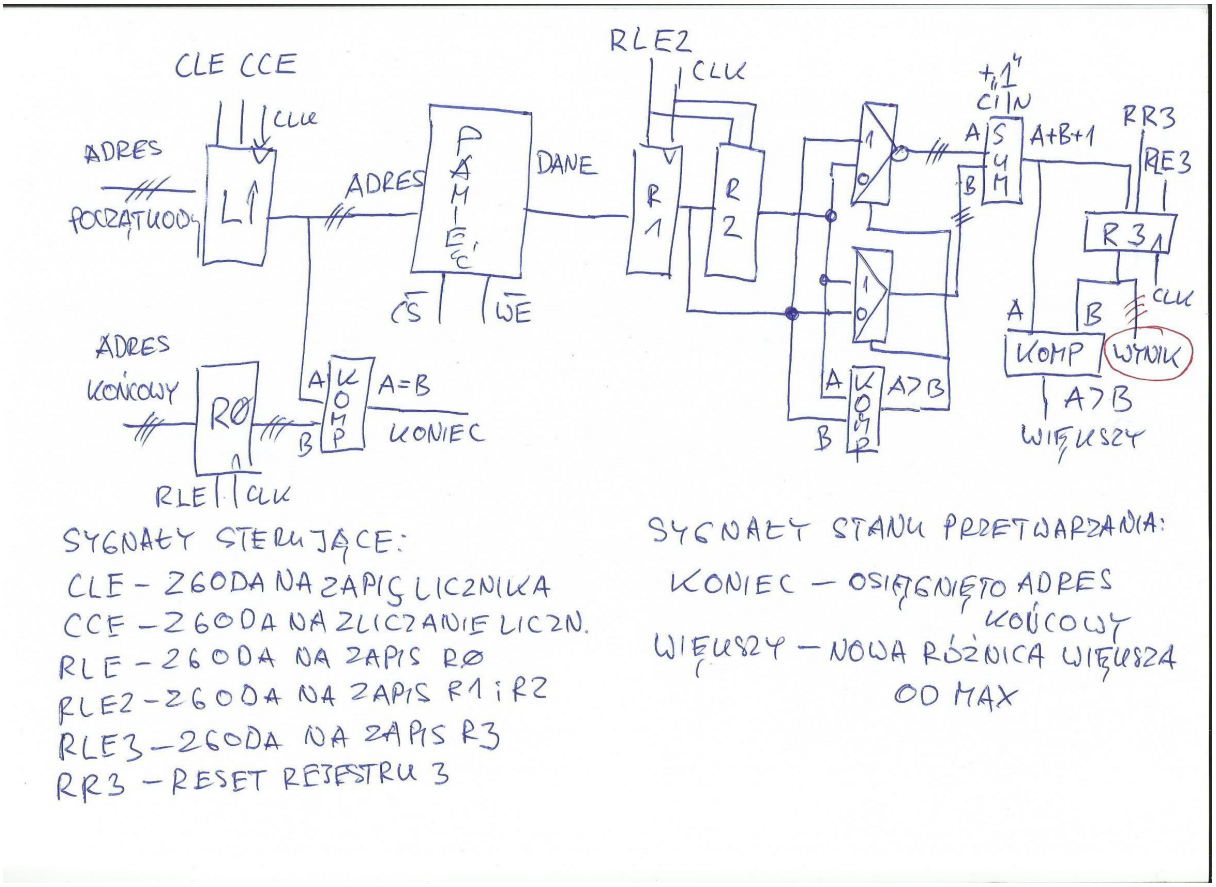
Testowanie sygnału ZERO realizowane w kolejnym takcie nie wprowadza dodatkowych wymagań.

PODSUMOWANIE

Powyższe warunki stanowią wymagania na prędkość taktowania układów wykonawczego i sterującego.

TEST ZAWARTOŚCI PAMIĘCI

Proszę znaleźć w pamięci od adresu podanego jako adres startowy do adresu podanego jako adres końcowy maksymalną wartość spośród wartości bezwzględnych różnic kolejnych liczb zapisanych na kolejnych adresach, np. dla danych 5,7,3,0 wynik wynosi 4.



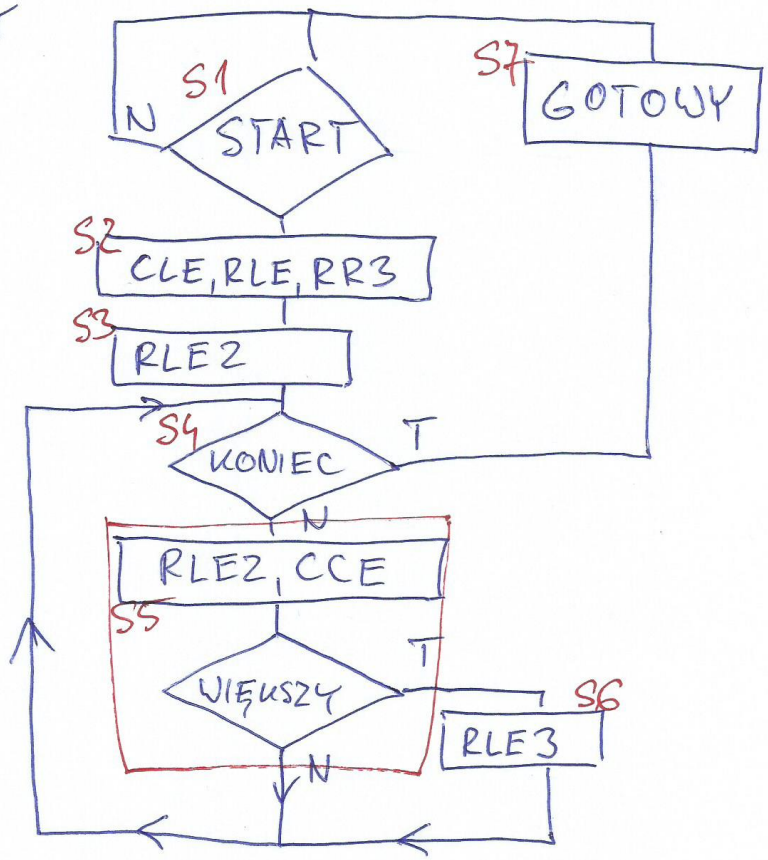
Rys 9. Schemat układu wykonawczego dla układu przeszukiwania pamięci.

SYGNAŁY WEJŚCIOWE:

- START
- ADRES POCZĄTKU PAMIĘCI
- ADRES KOŃCA PAMIĘCI

SYGNAŁY WYJŚCIOWE

- WYNIK
- GOTOWY



S2 PRZYGOTOWANIE UKŁADU
S3 ŁADOWANIE PIERWSZEJ WARTOŚCI Z PAM
S4 SPRAWDZANIE KOŃCA ZAKRESU
S5 ŁADOWANIE KOLEJNEJ WARTOŚCI, TEST KOŃCA
S6 ŁADOWANIE MAX DOREJ. WYJ.

Rys 10. Diagram ASM układu sterującego dla układu przeszukiwania pamięci.