

ZADANIE 1

Komparator szeregowy 2 liczb

Specyfikacja wymagań dla układu

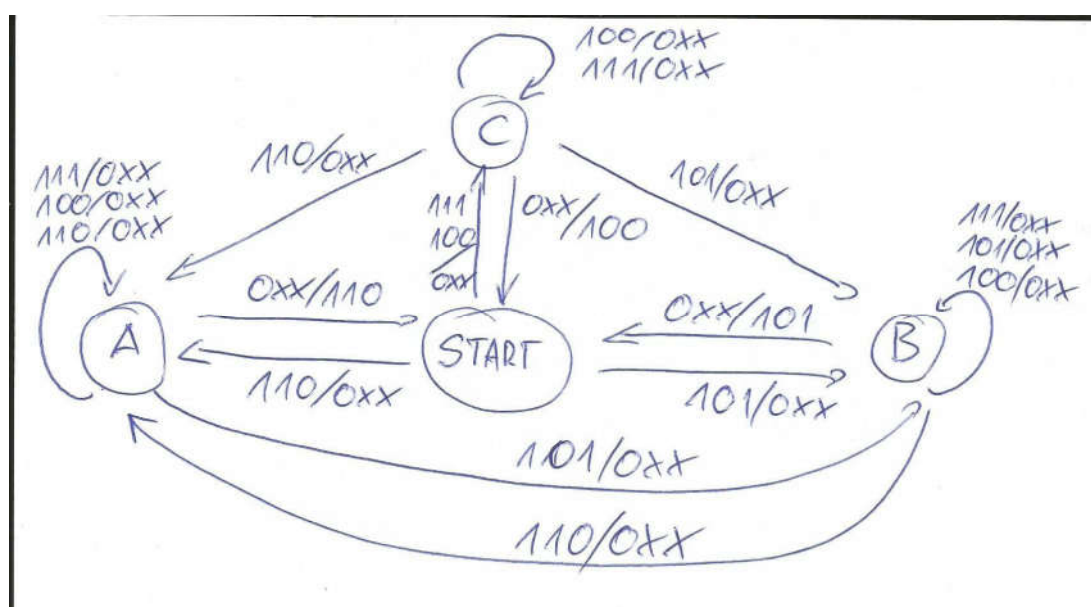
Bity podawane na wejściu od najmniej znaczącego bitu

Wejścia od najstarszego: liczba, bitA, bitB

Wyjścia od najstarszego: wynik, A>B, B>A

Liczba=1 oznacza 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; w grafie brakuje informacji o pozostaniu w stanie START gdy brakuje aktywności sygnału start.

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



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

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

$$B \gg A(Q1, Q0, \text{liczba}, \text{bita}, \text{bitB}) = \sum (24-27) + d(0-7, 12-15, 20-23, 28-31)$$

Zawartość ROM dla implementacji automatu „komparator szeregowy” typu Mealy’ego

Adresy=(Q1,Q0,liczba,bita,bitB)

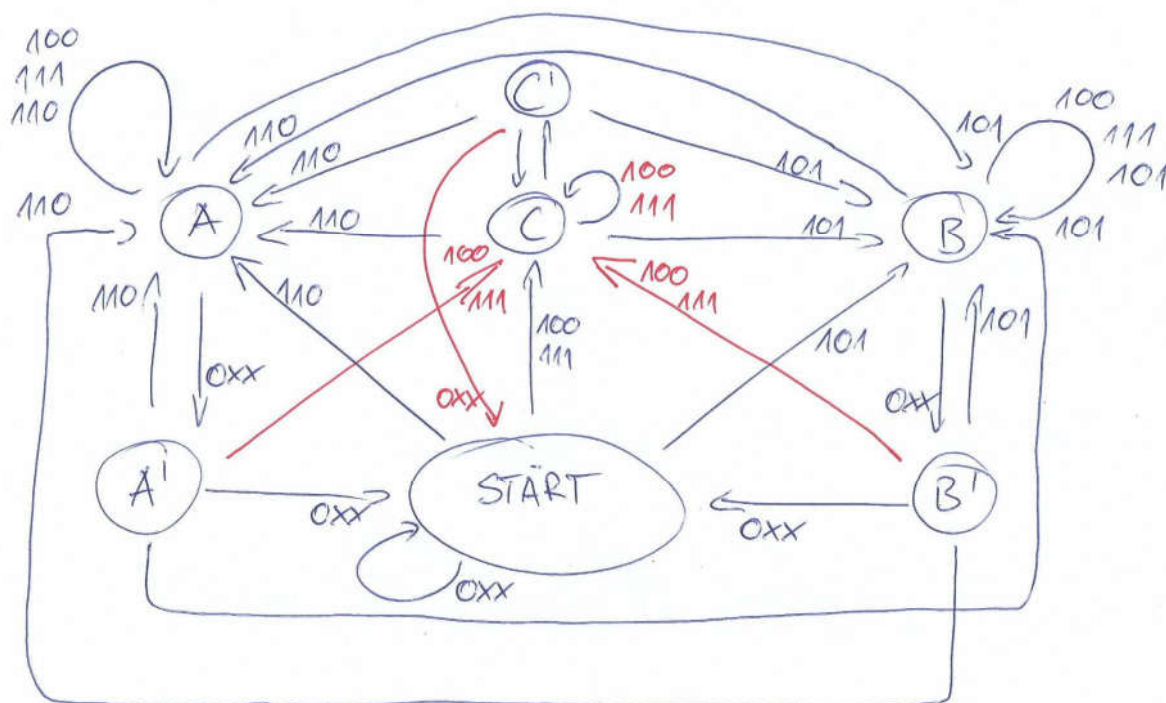
Dane=(D1,D0,Wynik,A>B,B>A)

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--

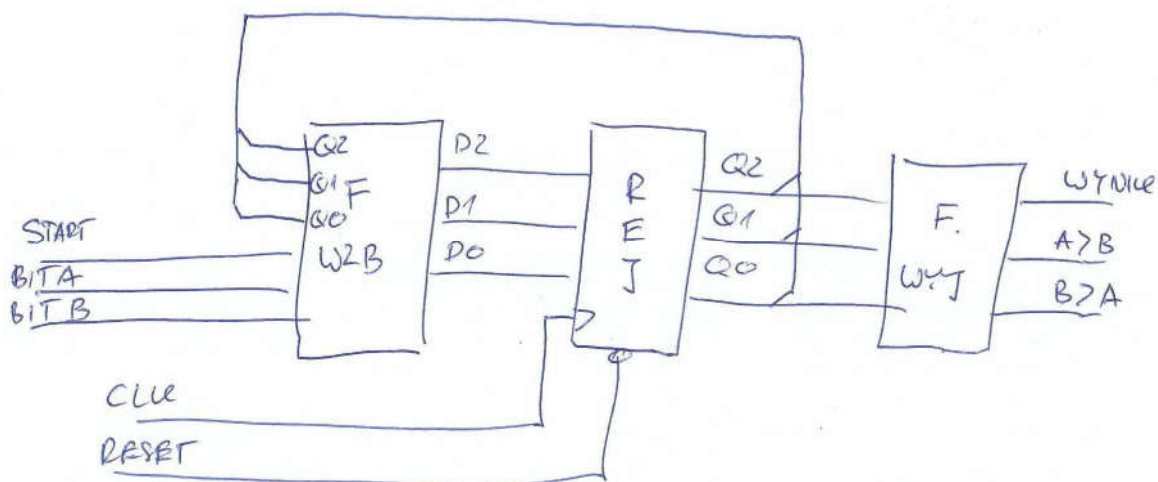
Tab.4 Zawartość pamięci dla implementacji automatu „komparator szeregowy” typu Mealy’ego, „-” oznacza wartość dowolną

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):

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

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

$D0(Q2, Q1, Q0, \text{liczba}, \text{bita}, \text{bitB}) = \sum(5-6, 12-19, 21-22, 24-31, 45-46, 53-54, 61-62) + d(32-39)$
 $\text{Wynik}(Q2, Q1, Q0) = \sum(5-7) + d(4)$
 $A > B(Q2, Q1, Q0) = \sum(6) + d(0-4)$
 $B > A(Q2, Q1, Q0) = \sum(7) + d(0-4)$

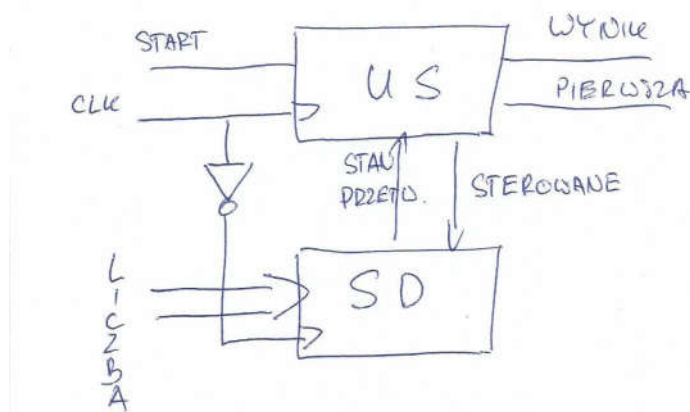
Zawartości ROM (częściowa) dla implementacji automatu „komparator szeregowy” typu Mealy’ego
 Adresy=(Q2Q1,Q0,liczba,bita,bitB)
 Dane=(D2,D1,D0,Wynik,A>B,B>A)

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

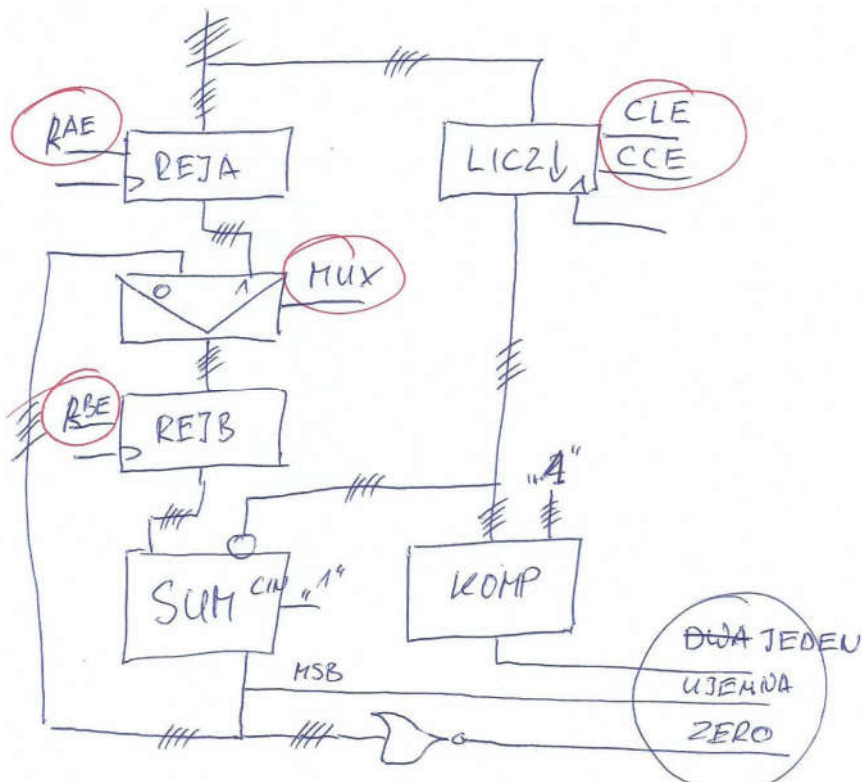
Tab.8 Zawartość pamięci dla implementacji automatu „komparator szeregowy” typu Moore’a, „–” oznacza wartość dowolną

Zadanie 2 TEST LICZBY PIERWSZEJ

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



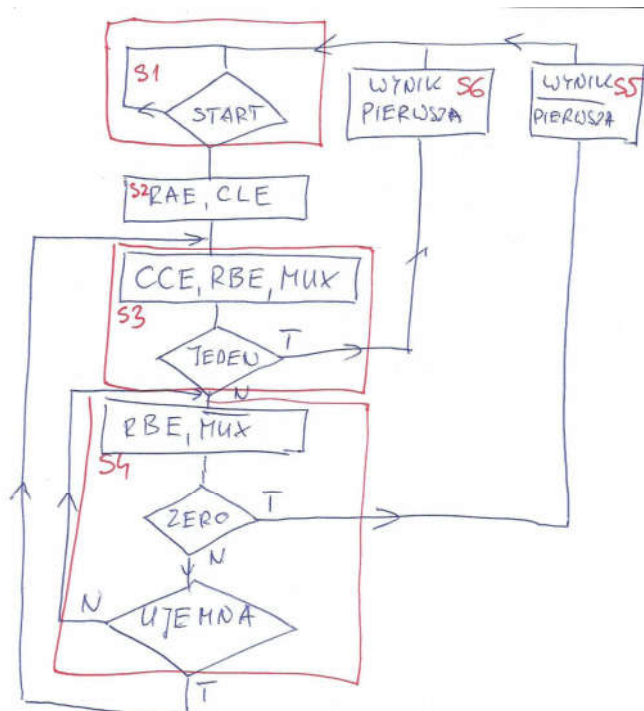
Rys.5 Schemat struktury układu do testu liczby pierwszej (układ sterowania i ścieżka danych)



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.

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

Sygnały stanu: JEDEK, UJEMNA, ZERO



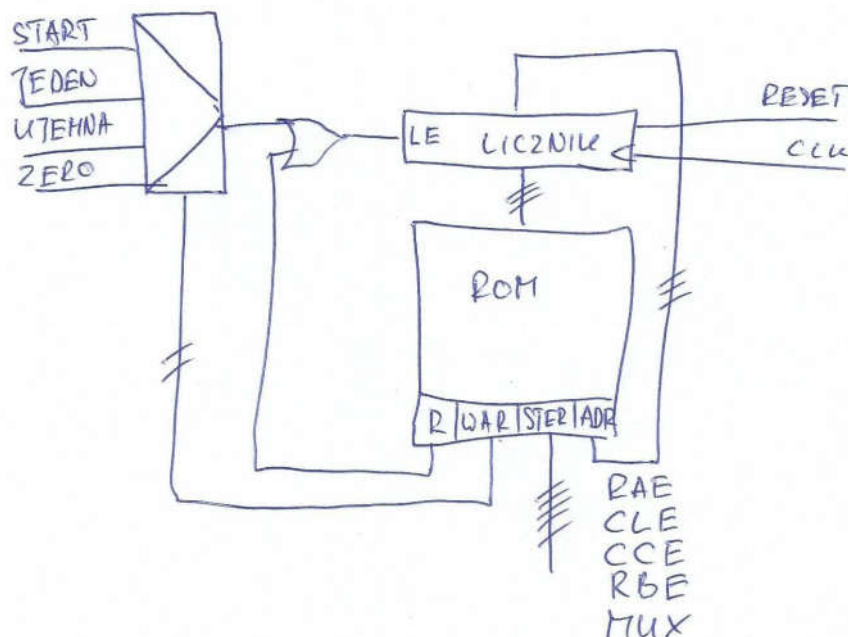
Rys. 7 Diagram ASM układ sterowania dla układu testowania liczby pierwszej, wyodrębniono 6 stanów niezbędnych do realizacji oddzielnych kroków sterowania układem wykonawczym i obsługi sygnałów, wejściowych i wyjściowych.

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.

Implementacja układu sterowania w oparciu o układ programowalny

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'.

Zadanie 3

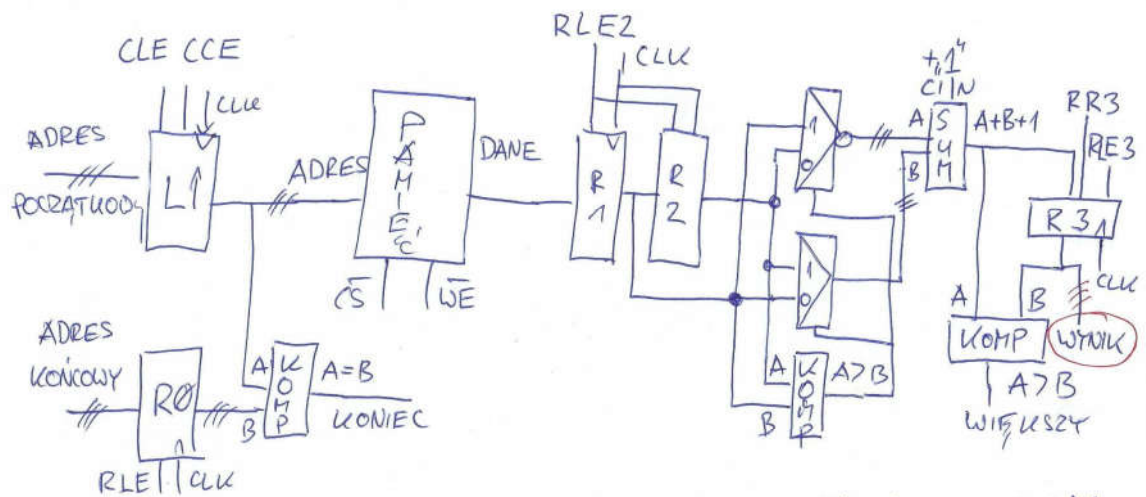
PRZYKŁADOWY TEST ZAWARTOŚCI PAMIĘCI

Opis problemu:

W pamięci od adresu podanego jako adres startowy do adresu podanego jako adres końcowy odczytywać liczby. Wyznaczyć maksymalną wartość spośród wartości bezwzględnych różnic kolejnych liczb, np. dla danych 5,7,3,0 wynik wynosi 4.

Należy określić interfejs z układem sterowania.

Uwaga: na wyjściu pamięci, rejestrów i sumatora znajdują się wielobitowe wektory sygnałów.



SYGNAŁY STERUJĄCE:

CLE - ZGODA NA ZAPIS LICZNIKA
 CCE - ZGODA NA ZLICZANIE LICZN.
 RLE - ZGODA NA ZAPIS R0
 RLE2 - ZGODA NA ZAPIS R1 i R2
 RLE3 - ZGODA NA ZAPIS R3
 RR3 - RESET REJESTRU 3

SYGNAŁY STANU PRZETWARZANIA:

KONIEC - OSIĄGNIĘTO ADRES KOŃCOWY
 WIĘKSZY - NOWA RÓŻNICA WIĘKSZA OD MAX

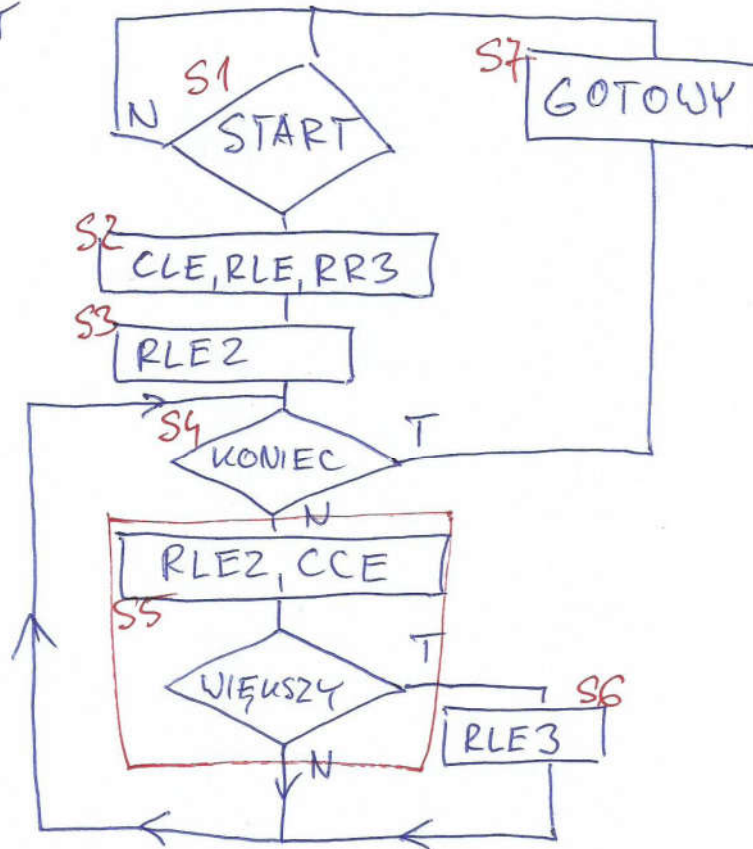
Rys 9. Schemat układu wykonawczego dla układu testu zawartości pamięci – zadanie 3.

SYGNAŁY WEJŚCIOWE:

- START
- ADRES POZĄTKU PAMIĘCI
- ADRES KONCA PAMIĘCI

SYGNAŁY WYJŚCIOWE

- WYNIK
- GOTOWY



S2 PRZYGOTOWANIE UKŁADU

S3 ŁADOWANIE PIERWSZEJ WARTOŚCI Z PAM

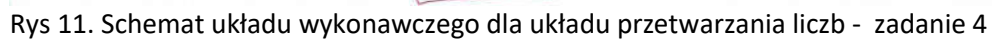
S4 SPRAWDZANIE KONCA ZAKRESU

S5 ŁADOWANIE KOLEJNEJ WARTOŚCI, TEST KONCA

S6 ŁADOWANIE MAX DOREJ. WYJ.

Rys 10. Diagram ASM układu sterującego dla testu zawartości pamięci-zadanie 3.

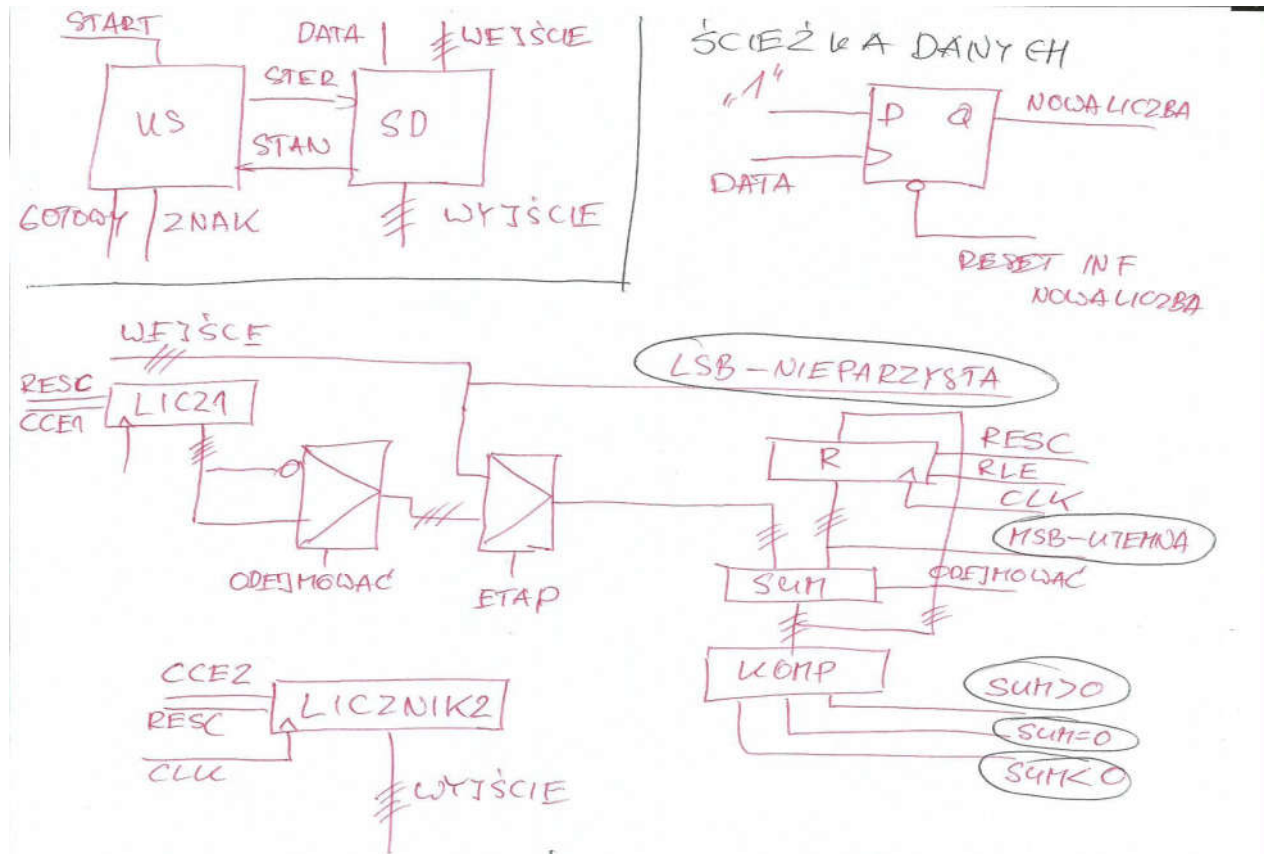
Opis problemu: Dane z wejścia układu należy przetwarzać w następujący sposób, liczbę większą od poprzedniej należy dodać do sumy, mniejszą odjąć; należy określić interfejs z układem sterowania, sygnalizacja nowej liczby zboczem sygnału DATA.



Zadanie 5

Opis problemu:

wyznaczenie wartości średniej (całkowitej) z ze zbioru liczb nieparzystych, należy określić interfejs z układem sterowania. Sygnalizacja nowej liczby zboczem sygnału DATA, układ otrzymuje sygnał końca zbioru liczb.



Rys 12. Schemat układu wykonawczego dla układu przetwarzania liczb - zadanie 5