

Ćwiczenie 3.

Emil Siatka, Marek Swakoń, Patryk Górski, Piotr Pich

4 czerwca 2025

Spis treści

1	Treść ćwiczenia	1
2	Zarys rozwiązania	1
3	Funkcje stanu oraz wyjścia	2
4	Minimalizacja funkcji stanu	14
4.1	<i>ST</i>	15
4.2	<i>CPR</i>	16
4.3	<i>CN</i>	17
4.4	<i>R₀</i>	18
4.5	<i>R₁</i>	20
5	Schemat układu	22
6	Stanowisko testujące	23
7	Podsumowanie oraz wnioski	27

1 Treść ćwiczenia

Proszę zaprojektować automat mogący posłużyć do sterowania jakimś prostym odtwarzaczem plików muzycznych mp3.

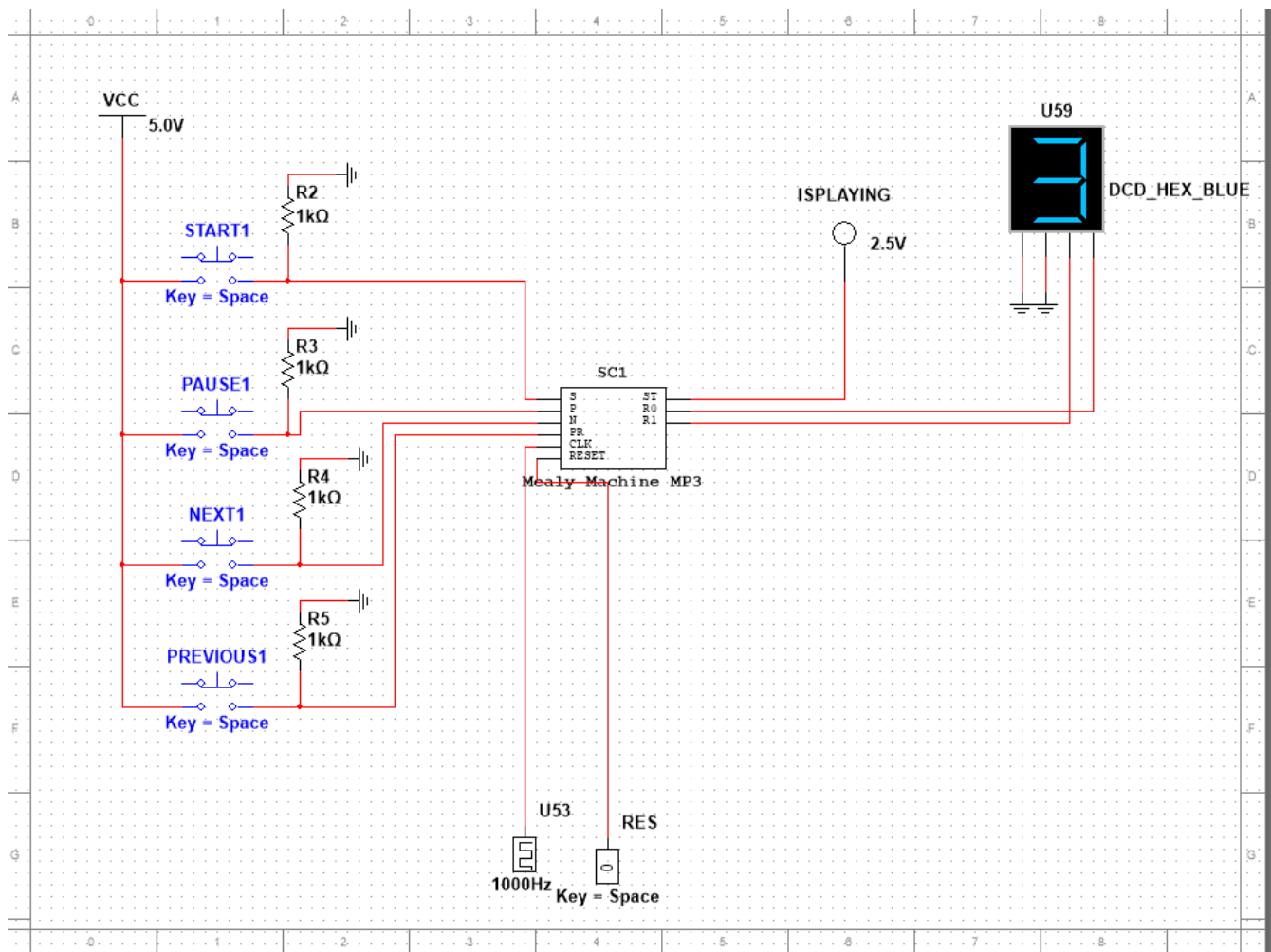
Układ powinien mieć następujące przyciski oraz odpowiadające im sygnały i wskaźniki:

- STOP
- PLAY
- NEXT
- PREVIOUS

oraz powinien posiadać dwubitowe wyjście binarne określające numer utworu.

2 Zarys rozwiązania

Projektowany układ zawierać będzie 4 wejścia dla każdego z 4 powyższych przycisków (odp. (*P*)ause, (*S*)tart, (*N*)ext, (*PR*)evious) oraz wejścia dla sygnału zegarowego oraz resetującego. Na wyjściu natomiast, oprócz dwubitowego numeru utworu (*R₀* oraz *R₁*) otrzymujemy również informację, czy obecnie utwór jest odtwarzany, czy zatrzymany (*ST*).



Rysunek 1: Interfejs projektowanego układu

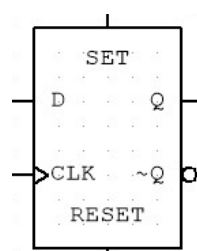
Zakładamy, że zmiana utworu następować ma po zwolnieniu odpowiedniego przycisku i skutkować ma rozpoczęciem odtwarzania następnej lub poprzedniej ścieżki.

3 Funkcje stanu oraz wyjścia

Powyższy układ zaprojektować można jako automat Mealy'ego $\langle Z, Q, Y, \Phi, \Psi, q_0 \rangle$ oparty na przerzutnikach typu D:

Tabela 1: Tabela przejść przerzutnika typu D

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

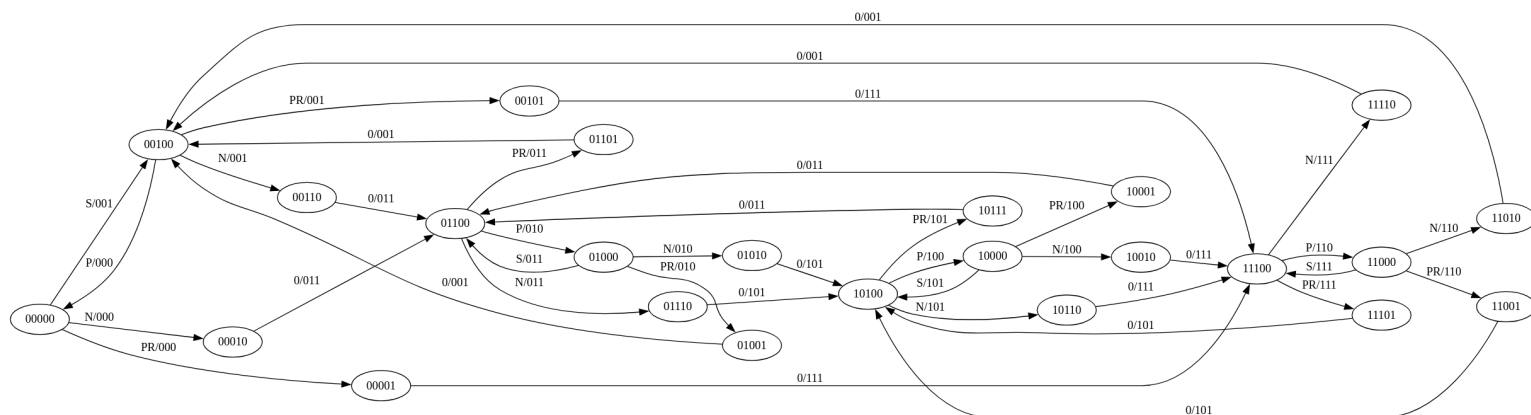


Rysunek 2: Przerzutnik typu D

Gdzie:

- $Z = \{P, S, N, PR, 0\}$, gdzie 0 oznacza stan niski na wszystkich wejściach
- $Y = 2^{\text{Out}}$, gdzie $\text{Out} = \{R_0, R_1, ST\}$
- $Q = 2^{\text{State}}$, gdzie $\text{State} = \text{Out} \cup \{CN, CPR\}$
- $q_0 = 00000$

CN oraz CPR oznaczają komórki stanu przechowujące informację o wysokim stanie wejść N oraz PR w poprzednim takcie zegara, aby projektowany automat mógł poprawnie zareagować na zwolnienie przycisków. Funkcje przejść i wyjść Φ oraz Ψ opisuje natomiast diagram:



Rysunek 3: Diagram projektowanego automatu

Lub też, dokładniej, biorąc pod uwagę przypadki brzegowe, tabela prawdy:

Tabela 2: Tabela prawdy dla funkcji Φ oraz Ψ

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0	0	1
0	0	0	0	0	0	1	1	0	0	0	0	1	0
0	0	0	0	0	0	1	1	1	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	0	0	0	1	1	0
0	0	0	0	0	1	0	1	1	0	0	1	1	0
0	0	0	0	0	1	1	0	0	0	0	1	0	0
0	0	0	0	0	1	1	0	1	0	0	1	0	1
0	0	0	0	0	1	1	1	0	0	0	1	1	0
0	0	0	0	0	1	1	1	1	0	0	1	1	0
0	0	0	0	1	0	0	0	0	1	1	1	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	0	0	1	0	0	1	1	0	0	0	1	1
0	0	0	0	1	0	1	0	0	1	1	1	0	0

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
0	0	0	0	1	0	1	0	1	0	0	0	0	1
0	0	0	0	1	0	1	1	0	0	0	0	1	1
0	0	0	0	1	0	1	1	1	0	0	0	1	1
0	0	0	0	1	1	0	0	0	1	1	1	0	0
0	0	0	0	1	1	0	0	1	0	0	1	0	1
0	0	0	0	1	1	0	1	0	0	0	1	1	1
0	0	0	0	1	1	0	1	1	0	0	1	1	1
0	0	0	0	1	1	1	0	0	1	1	1	0	0
0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1	0	0	0	1	1	1
0	0	0	0	1	1	1	1	1	0	0	1	1	1
0	0	0	1	0	0	0	0	0	0	1	1	0	0
0	0	0	1	0	0	0	0	1	0	1	1	0	0
0	0	0	1	0	0	0	1	0	0	0	0	1	0
0	0	0	1	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	1	1	0	0	0	0	1	0
0	0	0	1	0	0	1	1	1	0	0	0	1	1
0	0	0	1	0	1	0	0	0	0	1	1	0	0
0	0	0	1	0	1	0	0	1	0	1	1	0	0
0	0	0	1	0	1	0	1	0	0	0	1	1	0
0	0	0	1	0	1	1	0	0	0	0	1	1	0
0	0	0	1	0	1	1	1	1	0	0	1	1	1
0	0	0	1	0	1	1	1	0	0	1	1	0	0
0	0	0	1	0	1	1	1	1	0	0	1	1	0
0	0	0	1	1	0	0	0	0	0	1	1	0	1
0	0	0	1	1	0	0	0	1	0	1	1	0	1
0	0	0	1	1	0	0	1	0	1	1	1	0	1
0	0	0	1	1	0	0	1	1	0	0	1	1	0
0	0	0	1	1	0	1	0	0	0	1	1	0	1
0	0	0	1	1	0	1	1	0	1	1	1	1	0
0	0	0	1	1	0	1	1	1	0	0	0	1	1
0	0	0	1	1	1	0	0	0	0	1	1	0	1
0	0	0	1	1	1	0	0	1	0	1	1	0	1
0	0	0	1	1	1	1	0	0	0	0	1	0	1
0	0	0	1	1	1	1	1	0	1	1	1	1	0
0	0	0	1	1	1	1	1	1	0	0	1	1	1
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	0	1	0	1
0	0	1	0	0	0	0	1	1	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1	0	0	0	0	1	0
0	0	1	0	0	0	1	1	1	0	0	0	1	0
0	0	1	0	0	0	1	1	1	0	0	0	1	0

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
0	0	1	0	0	1	0	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	1	0	0	1	0	1
0	0	1	0	0	1	0	1	0	0	0	1	1	0
0	0	1	0	0	1	0	1	1	0	0	1	1	0
0	0	1	0	0	1	1	0	0	0	0	1	0	0
0	0	1	0	0	1	1	0	1	0	0	1	0	1
0	0	1	0	0	1	1	1	0	0	0	1	1	0
0	0	1	0	0	1	1	1	1	0	0	1	1	0
0	0	1	0	1	0	0	0	0	1	1	1	0	0
0	0	1	0	1	0	0	0	1	0	0	1	0	1
0	0	1	0	1	0	0	1	0	0	0	1	1	1
0	0	1	0	1	0	0	1	1	0	0	1	1	1
0	0	1	0	1	0	1	0	0	1	1	1	0	0
0	0	1	0	1	0	1	0	1	0	0	0	0	1
0	0	1	0	1	0	1	1	0	0	0	0	1	1
0	0	1	0	1	0	1	1	1	0	0	0	1	1
0	0	1	0	1	1	0	0	0	1	1	1	0	0
0	0	1	0	1	1	0	0	1	0	0	1	0	1
0	0	1	0	1	1	0	1	0	0	0	1	1	1
0	0	1	0	1	1	1	0	0	1	1	1	0	0
0	0	1	0	1	1	1	1	1	0	0	1	1	1
0	0	1	1	0	0	0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	0	1	0	1	1	0	0
0	0	1	1	0	0	0	1	0	0	0	1	1	0
0	0	1	1	0	0	0	1	1	0	0	1	1	1
0	0	1	1	0	0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	1	0	0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	1	1	1	0	0	1	1	1
0	0	1	1	0	1	1	0	0	0	1	1	0	0
0	0	1	1	0	1	1	0	1	0	1	1	0	0
0	0	1	1	0	1	1	1	0	0	0	1	1	0
0	0	1	1	0	1	1	1	1	0	0	1	1	1
0	0	1	1	1	0	0	0	0	0	1	1	0	1
0	0	1	1	1	0	0	0	1	0	1	1	0	1
0	0	1	1	1	0	0	1	0	1	1	1	1	0
0	0	1	1	1	0	0	1	1	0	0	1	1	1
0	0	1	1	1	0	0	1	1	0	0	1	1	1
0	0	1	1	1	0	1	0	0	0	1	1	0	1
0	0	1	1	1	0	1	1	0	1	1	1	1	0
0	0	1	1	1	0	1	1	1	0	0	0	1	1
0	0	1	1	1	1	0	0	0	0	1	1	0	1
0	0	1	1	1	1	0	0	1	0	1	1	1	

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

[illegible]

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
0	1	0	1	0	1	1	1	0	0	1	1	1	0
0	1	0	1	0	1	1	1	1	0	1	1	1	1
0	1	0	1	1	0	0	0	0	1	0	1	0	1
0	1	0	1	1	0	0	0	1	1	0	1	0	1
0	1	0	1	1	0	0	1	0	0	0	1	1	0
0	1	0	1	1	0	0	1	1	0	1	0	1	1
0	1	0	1	1	0	1	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0	1	1	0	1	0	1
0	1	0	1	1	0	1	1	0	0	0	1	1	0
0	1	0	1	1	0	1	1	1	0	1	0	1	1
0	1	0	1	1	1	0	0	0	1	0	1	0	1
0	1	0	1	1	1	0	0	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	0	0	1	1	0
0	1	0	1	1	1	0	1	1	0	1	1	1	0
0	1	0	1	1	1	1	1	0	0	0	1	1	0
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	0	0	0	0	0	0	1	1	0	0
0	1	1	0	0	0	0	0	1	0	1	1	0	1
0	1	1	0	0	0	0	1	0	0	1	1	1	0
0	1	1	0	0	0	0	1	1	0	1	1	1	0
0	1	1	0	0	0	1	0	0	0	1	0	0	0
0	1	1	0	0	0	1	1	0	0	1	0	1	0
0	1	1	0	0	0	1	1	1	0	1	1	0	0
0	1	1	0	0	1	0	0	0	0	1	1	0	0
0	1	1	0	0	1	0	0	1	0	1	1	0	1
0	1	1	0	0	1	1	1	0	0	1	1	1	0
0	1	1	0	0	1	1	1	1	0	1	1	1	0
0	1	1	0	0	1	1	0	1	0	1	1	0	1
0	1	1	0	0	1	1	1	0	0	1	1	1	0
0	1	1	0	0	1	1	1	1	0	1	1	1	0
0	1	1	0	0	1	1	0	0	0	1	1	0	0
0	1	1	0	0	1	1	0	1	0	1	1	0	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	0	0	1	1	0	0	0	0	1	0	0
0	1	1	0	0	1	1	1	0	0	1	1	1	0
0	1	1	0	0	1	1	1	1	0	1	1	1	0
0	1	1	0	0	1	1	0	1	0	1	1	0	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	0	0	1	1	0	0	0	0	1	0	0
0	1	1	0	0	1	1	1	0	0	1	1	1	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	0	0	1	1	1	1	0	1	1	1	

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
0	1	1	1	0	0	0	0	1	1	0	1	0	0
0	1	1	1	0	0	0	1	0	0	1	1	1	0
0	1	1	1	0	0	0	1	1	0	1	1	1	1
0	1	1	1	0	0	1	0	0	1	0	1	0	0
0	1	1	1	0	0	1	0	1	1	0	1	0	0
0	1	1	1	0	0	1	1	0	0	1	0	1	0
0	1	1	1	0	0	1	1	1	0	1	0	1	1
0	1	1	1	0	1	0	0	0	1	0	1	0	0
0	1	1	1	0	1	0	0	1	1	0	1	0	0
0	1	1	1	0	1	0	1	0	0	1	1	1	0
0	1	1	1	0	1	0	1	1	0	1	1	1	1
0	1	1	1	0	1	1	0	0	1	0	1	0	0
0	1	1	1	0	1	1	0	1	1	0	1	0	0
0	1	1	1	0	1	1	1	0	0	1	1	1	0
0	1	1	1	0	1	1	1	1	0	1	1	1	1
0	1	1	1	1	0	0	0	0	1	0	1	0	1
0	1	1	1	1	0	0	0	1	1	0	1	0	1
0	1	1	1	1	0	0	1	0	0	0	1	1	0
0	1	1	1	1	0	0	1	1	0	1	1	1	1
0	1	1	1	1	0	1	0	0	1	0	1	0	1
0	1	1	1	1	0	1	0	1	1	0	1	0	1
0	1	1	1	1	0	1	1	0	0	0	1	1	0
0	1	1	1	1	0	1	1	1	0	1	0	1	1
0	1	1	1	1	1	0	0	0	1	0	1	0	1
0	1	1	1	1	1	0	1	0	0	0	1	1	0
0	1	1	1	1	1	0	1	1	0	1	1	1	1
0	1	1	1	1	1	1	0	0	1	0	1	0	1
0	1	1	1	1	1	1	0	1	1	0	1	0	1
0	1	1	1	1	1	1	1	0	0	0	1	1	0
0	1	1	1	1	1	1	1	1	0	1	1	1	1
1	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	1	1	0	0	0	1
1	0	0	0	0	0	0	1	0	1	0	0	1	0
1	0	0	0	0	0	0	1	1	1	0	0	1	0
1	0	0	0	0	0	1	0	0	1	0	0	0	0
1	0	0	0	0	0	1	0	1	1	0	0	0	1
1	0	0	0	0	0	1	1	0	1	0	0	1	0
1	0	0	0	0	0	1	1	1	1	0	0	1	0
1	0	0	0	0	1	0	0	0	1	0	1	0	0
1	0	0	0	0	1	0	0	1	1	0	1	0	1
1	0	0	0	0	1	0	1	1	1	0	1	0	0
1	0	0	0	0	1	1	0	1	1	0	1	0	1
1	0	0	0	0	1	1	1	0	1	0	1	1	0
1	0	0	0	0	1	1	1	1	1	0	1	1	0
1	0	0	0	1	0	0	0	0	0	1	1	0	0
1	0	0	0	1	0	0	0	1	1	0	0	0	1
1	0	0	0	1	0	0	1	0	1	0	0	1	1
1	0	0	0	1	0	0	1	1	1	0	0	1	1
1	0	0	0	1	0	0	1	1	1	0	0	1	1

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
1	0	0	0	1	0	1	0	0	0	1	1	0	0
1	0	0	0	1	0	1	0	1	1	0	0	0	1
1	0	0	0	1	0	1	1	0	1	0	0	1	1
1	0	0	0	1	0	1	1	1	1	0	0	1	1
1	0	0	0	1	1	0	0	0	0	1	1	0	0
1	0	0	0	1	1	0	0	1	1	0	1	0	1
1	0	0	0	1	1	0	1	0	1	0	1	1	1
1	0	0	0	1	1	0	1	1	1	0	1	1	1
1	0	0	0	1	1	1	0	0	0	1	1	0	0
1	0	0	0	1	1	1	0	1	1	0	1	0	1
1	0	0	0	1	1	1	1	0	1	0	1	1	1
1	0	0	1	0	0	0	0	0	1	1	1	0	0
1	0	0	1	0	0	0	0	1	1	0	0	1	0
1	0	0	1	0	0	0	1	1	1	0	0	1	1
1	0	0	1	0	0	1	0	0	1	1	1	0	0
1	0	0	1	0	0	1	0	1	1	1	1	0	0
1	0	0	1	0	0	1	1	0	1	0	0	1	0
1	0	0	1	0	0	1	1	1	1	0	0	1	1
1	0	0	1	0	1	0	0	0	1	1	1	0	0
1	0	0	1	0	1	0	0	1	1	0	1	0	0
1	0	0	1	0	1	0	1	0	1	1	1	0	0
1	0	0	1	0	1	1	0	0	1	1	1	0	0
1	0	0	1	0	1	1	1	0	1	0	1	1	0
1	0	0	1	0	1	0	0	0	1	1	1	0	0
1	0	0	1	0	1	0	1	0	1	0	1	1	0
1	0	0	1	0	1	0	1	1	1	0	0	1	1
1	0	0	1	0	1	1	0	0	1	1	1	0	0
1	0	0	1	0	1	1	1	0	1	0	1	1	0
1	0	0	1	0	1	1	1	1	1	0	0	1	1
1	0	0	1	1	0	0	0	0	1	1	1	0	1
1	0	0	1	1	0	0	0	1	1	1	1	0	1
1	0	0	1	1	0	0	1	0	0	1	1	1	0
1	0	0	1	1	0	0	1	1	1	0	0	1	1
1	0	0	1	1	0	1	0	0	1	1	1	0	1
1	0	0	1	1	0	1	1	0	0	1	1	1	0
1	0	0	1	1	0	1	1	1	1	0	0	1	1
1	0	0	1	1	1	0	0	0	1	1	1	0	1
1	0	0	1	1	1	0	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	0	0	0	0	1	0	1	0	0
1	0	1	0	0	0	0	0	1	1	0	1	0	1
1	0	1	0	0	0	0	1	0	1	0	1	1	0
1	0	1	0	0	0	1	0	0	1	0	0	0	

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
1	0	1	0	0	0	1	1	1	1	0	0	1	0
1	0	1	0	0	1	0	0	0	1	0	1	0	0
1	0	1	0	0	1	0	0	1	1	0	1	0	1
1	0	1	0	0	1	0	1	0	1	0	1	1	0
1	0	1	0	0	1	0	1	1	1	0	1	1	0
1	0	1	0	0	1	1	0	0	1	0	1	0	0
1	0	1	0	0	1	1	0	1	1	0	1	0	1
1	0	1	0	0	1	1	1	0	1	0	1	1	0
1	0	1	0	0	1	1	1	1	1	0	1	1	0
1	0	1	0	1	0	0	0	0	0	1	1	0	0
1	0	1	0	1	0	0	0	1	1	0	1	0	1
1	0	1	0	1	0	0	1	0	1	0	1	1	1
1	0	1	0	1	0	0	1	1	1	0	1	1	1
1	0	1	0	1	0	1	0	0	0	1	1	0	0
1	0	1	0	1	0	1	0	1	1	0	0	1	1
1	0	1	0	1	0	1	1	1	1	0	0	1	1
1	0	1	0	1	1	0	0	0	0	1	1	0	0
1	0	1	0	1	1	0	0	1	1	0	1	0	1
1	0	1	0	1	1	0	1	0	1	0	1	1	1
1	0	1	0	1	1	0	1	1	1	0	1	1	1
1	0	1	0	1	1	1	0	0	0	1	1	0	0
1	0	1	0	1	0	0	0	1	1	1	1	0	0
1	0	1	0	1	0	0	1	0	1	0	1	1	0
1	0	1	0	1	0	0	1	1	1	0	1	1	1
1	0	1	0	1	1	1	0	0	0	1	1	0	0
1	0	1	0	1	1	1	0	1	1	0	1	0	1
1	0	1	0	1	1	1	1	1	1	0	1	1	1
1	0	1	1	0	0	0	0	0	1	1	1	0	0
1	0	1	1	0	0	0	0	1	1	1	1	0	0
1	0	1	1	0	0	0	1	0	1	0	1	1	0
1	0	1	1	0	0	0	1	1	1	0	1	1	1
1	0	1	1	0	0	1	0	0	1	1	1	0	0
1	0	1	1	0	0	1	0	1	1	0	1	1	0
1	0	1	1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	0	1	1	0	0	1	1	1	0	0
1	0	1	1	0	1	1	0	1	1	1	1	0	0
1	0	1	1	0	1	1	1	0	1	0	1	1	0
1	0	1	1	0	1	1	1	1	1	0	1	1	1
1	0	1	1	0	0	0	0	0	1	1	1	0	0
1	0	1	1	0	0	0	0	1	1	1	1	0	1
1	0	1	1	0	0	0	1	0	0	1	1	1	0
1	0	1	1	0	0	0	1	1	1	0	1	1	1
1	0	1	1	0	0	1	0	0	1	1	1	0	0
1	0	1	1	0	0	1	0	1	1	1	1	0	

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
1	0	1	1	1	1	0	1	0	0	1	1	1	0
1	0	1	1	1	1	0	1	1	1	0	1	1	1
1	0	1	1	1	1	1	0	0	1	1	1	0	1
1	0	1	1	1	1	1	0	1	1	1	1	0	1
1	0	1	1	1	1	1	1	0	0	1	1	1	0
1	0	1	1	1	1	1	1	1	1	0	1	1	1
1	1	0	0	0	0	0	0	0	1	1	0	0	0
1	1	0	0	0	0	0	0	1	1	1	0	0	1
1	1	0	0	0	0	0	1	0	1	1	0	1	0
1	1	0	0	0	0	0	1	1	1	1	0	1	0
1	1	0	0	0	0	1	0	0	1	1	0	0	0
1	1	0	0	0	0	1	0	1	1	1	0	0	1
1	1	0	0	0	1	0	0	0	1	1	1	0	1
1	1	0	0	0	1	0	1	0	1	1	1	1	0
1	1	0	0	0	1	1	0	0	1	1	1	0	0
1	1	0	0	0	1	1	1	1	1	1	1	1	0
1	1	0	0	1	0	0	0	0	1	0	1	0	0
1	1	0	0	1	0	0	0	1	1	1	0	0	1
1	1	0	0	1	0	0	1	0	1	1	0	1	1
1	1	0	0	1	0	0	1	1	1	1	0	1	1
1	1	0	0	1	0	1	0	0	1	0	1	0	0
1	1	0	0	1	0	1	0	1	1	1	1	1	1
1	1	0	0	1	0	1	1	1	1	1	0	1	1
1	1	0	0	1	1	0	0	0	1	0	1	0	0
1	1	0	0	1	1	0	0	1	1	1	1	0	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	0	0	0	0	1	0	0
1	1	0	1	0	0	0	0	1	0	0	1	0	0
1	1	0	1	0	0	0	1	0	1	1	0	1	0
1	1	0	1	0	0	0	1	1	1	1	0	1	1
1	1	0	1	0	0	1	0	0	0	0	1	0	0
1	1	0	1	0	0	1	0	1	0	0	1	0	0
1	1	0	1	0	0	1	1	0	1	1	0	1	0
1	1	0	1	0	0	1	1	1	1	1	0	1	1
1	1	0	1	0	1	0	0	0	0	0	1	0	0
1	1	0	1	0	1	0	0	1	0	0	1	0	0
1	1	0	1	0	1	0	1	0	1	1	1	1	0
1	1	0	1	0	1	0	1	1	1	1	1	1	1
1	1	0	1	0	1	1	0	0	0	0	1	0	0
1	1	0	1	0	1	0	1	0	1	1	1	1	0
1	1	0	1	0	1	0	1	1	1	1	1	1	1
1	1	0	1	0	1	1	0	0	0	0	1	0	0

Ciąg dalszy na następnej stronie

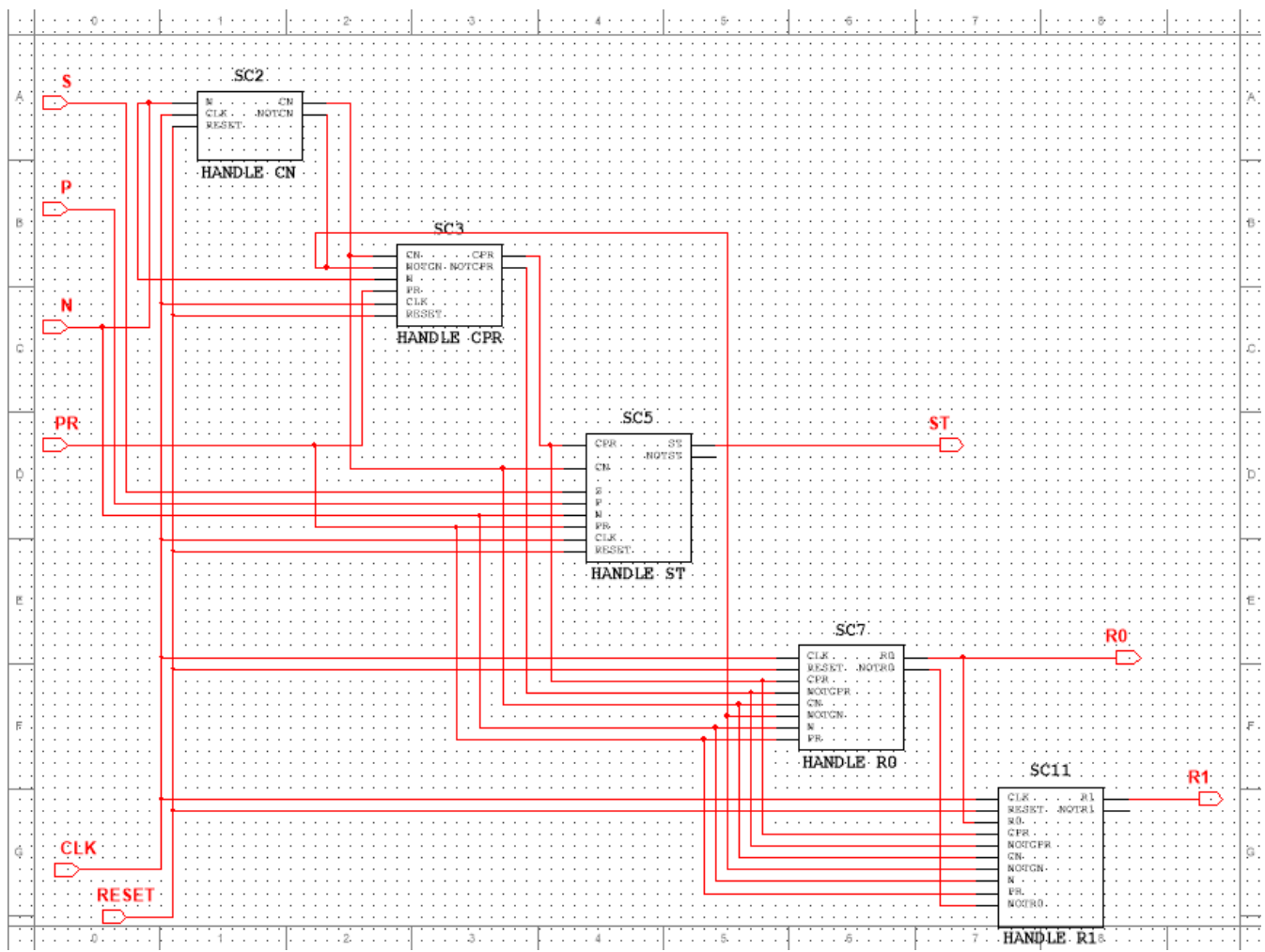
Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
1	1	0	1	0	1	1	0	1	0	0	1	0	0
1	1	0	1	0	1	1	1	0	1	1	1	1	0
1	1	0	1	0	1	1	1	1	1	1	1	1	1
1	1	0	1	1	0	0	0	0	0	0	1	0	1
1	1	0	1	1	0	0	0	1	0	0	1	0	1
1	1	0	1	1	0	0	1	0	1	0	1	1	0
1	1	0	1	1	0	0	1	1	1	1	0	1	1
1	1	0	1	1	0	1	0	0	0	0	1	0	1
1	1	0	1	1	0	1	0	1	0	0	1	0	1
1	1	0	1	1	0	1	1	0	1	0	1	1	0
1	1	0	1	1	0	1	1	1	1	1	0	1	1
1	1	0	1	1	1	0	0	0	0	0	1	0	1
1	1	0	1	1	1	0	0	1	0	0	1	0	1
1	1	0	1	1	1	1	0	1	0	0	1	0	1
1	1	0	1	1	1	1	1	0	1	0	1	1	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	1	1	1	0	0
1	1	1	0	0	0	0	0	1	1	1	1	0	1
1	1	1	0	0	0	0	1	0	1	1	1	1	0
1	1	1	0	0	0	0	1	1	1	1	1	1	0
1	1	1	0	0	0	1	0	0	1	1	1	0	0
1	1	1	0	0	0	1	0	1	1	1	0	1	0
1	1	1	0	0	0	1	1	1	1	1	0	1	0
1	1	1	0	0	1	0	0	0	1	1	1	0	0
1	1	1	0	0	1	0	1	1	1	1	1	1	0
1	1	1	0	0	1	1	1	1	1	1	1	1	0
1	1	1	0	1	0	0	0	0	1	0	1	0	0
1	1	1	0	1	0	0	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	0	1	1	1	1	1
1	1	1	0	1	0	1	0	0	1	0	1	0	0
1	1	1	0	1	0	1	0	1	1	1	0	0	1
1	1	1	0	1	0	1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	1	1	1	1	1	1	0
1	1	1	0	1	0	0	0	0	1	0	1	0	0
1	1	1	0	1	0	0	0	1	1	1	1	1	1
1	1	1	0	1	0	0	1	1	1	1	1	1	1
1	1	1	0	1	0	1	0	0	1	0	1	0	0
1	1	1	0	1	0	1	0	1	1	1	0	0	1
1	1	1	0	1	0	1	1	0	1	1	0	1	1
1	1	1	0	1	0	1	1	1	1	1	0	1	1
1	1	1	0	1	0	0	0	0	1	0	1	0	0
1	1	1	0	1	0	0	0	1	1	1	1	1	1
1	1	1	0	1	1	1	0	0	1	0	1	0	0
1	1	1	0	1	1	1	0	1	1	1	1	0	

Ciąg dalszy na następnej stronie

Ciąg dalszy poprzedniej strony

$R_1^{(n)}$	$R_0^{(n)}$	$ST^{(n)}$	$CN^{(n)}$	$CPR^{(n)}$	S	P	N	PR	$R_1^{(n+1)}$	$R_0^{(n+1)}$	$ST^{(n+1)}$	$CN^{(n+1)}$	$CPR^{(n+1)}$
1	1	1	1	0	0	0	0	0	0	0	1	0	0
1	1	1	1	0	0	0	0	1	0	0	1	0	0
1	1	1	1	0	0	0	1	0	1	1	1	1	0
1	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	1	1	0	0	1	0	0	0	0	1	0	0
1	1	1	1	0	0	1	0	1	0	0	1	0	0
1	1	1	1	0	0	1	1	0	1	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	1	1
1	1	1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	1	0	1	0	0	1	0	0	1	0	0
1	1	1	1	0	1	0	1	0	1	1	1	1	0
1	1	1	1	0	1	0	1	1	1	1	1	1	1
1	1	1	1	0	1	1	0	0	0	0	1	0	0
1	1	1	1	0	1	1	0	1	0	0	1	0	0
1	1	1	1	0	1	1	1	0	1	1	1	1	0
1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	1	0	1
1	1	1	1	1	0	0	0	1	0	0	1	0	1
1	1	1	1	1	0	0	1	0	1	0	1	1	0
1	1	1	1	1	0	0	1	1	1	1	1	1	1
1	1	1	1	1	0	1	0	0	0	0	1	0	1
1	1	1	1	1	0	1	0	1	0	0	1	0	1
1	1	1	1	1	0	1	1	0	1	0	1	1	0
1	1	1	1	1	0	1	1	1	1	1	0	1	1
1	1	1	1	1	1	0	0	0	0	0	1	0	1
1	1	1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	1	1	1	0	0	0	0	1	0	1
1	1	1	1	1	1	1	0	1	0	0	1	1	0
1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	0	0	0	1	0	1
1	1	1	1	1	1	1	0	1	0	0	1	0	1
1	1	1	1	1	1	1	1	0	1	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1



Rysunek 4: Szkielet wnętrza projektowanego układu

4 Minimalizacja funkcji stanu

Poniższe zminimalizowane wyprowadzenia otrzymaliśmy z wykorzystaniem biblioteki LogicMin w języku Python wykorzystującą wewnętrznie metodę Quine’a-McCluskeya, będącą, pod kątem teoretycznym, równoważną metodzie tablic Karnaugh:

```
import pandas
import logicmin
from collections import defaultdict

conflict_check = defaultdict(set)

df = pandas.read_csv("./truth_table3.csv", sep = ';')

t = logicmin.TT(9, 5)

mp3_input = df.iloc[:, 0:9].to_numpy().tolist()
mp3_output = df.iloc[:, 9:14].to_numpy().tolist()

mi = ["".join(str(el) for el in element) for element in mp3_input]
mo = ["".join(str(el) for el in element) for element in mp3_output]

for i in range(len(mi)):
    t.add(mi[i], mo[i])
```

```

for i in range(len(mi)):
    conflict_check[mi[i]].add(mo[i])

for inp, out_set in conflict_check.items():
    if len(out_set) > 1:
        print(f" Konflikt dla wejścia {inp}: {out_set}")

sols = t.solve()

print(sols.printN(xnames= ['R1','R0','ST','CN','CPR','S','P','N','PR'],
                        ynames = ["R1'", "R0'", "ST'", "CN'", "CPR'"]))

```

Wszystkie otrzymane wyniki przed implementacją w programie Multisim zaimplementowaliśmy i przetestowaliśmy programistycznie w języku Python:

```

def NOT(x): return 0 if x else 1
def AND(*args): return int(all(args))
def OR(*args): return int(any(args))

def simulate_fsm_from_string(input_string):
    ...

```

4.1 ST

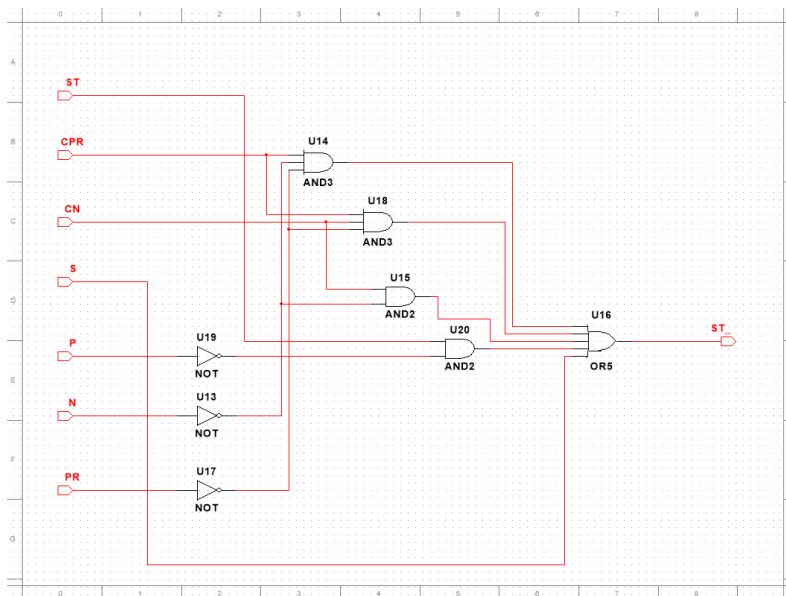
Kod symulujący funkcję stanu ST :

$$\begin{aligned}
 ST^{(n+1)} = & (CPR)^{(n)} \overline{N} \overline{(PR)} \\
 & + (CN)^{(n)} (CPR)^{(n)} \overline{(PR)} \\
 & + (CN)^{(n)} \overline{N} \\
 & + (ST)^{(n)} \overline{P} \\
 & + S
 \end{aligned} \tag{1}$$

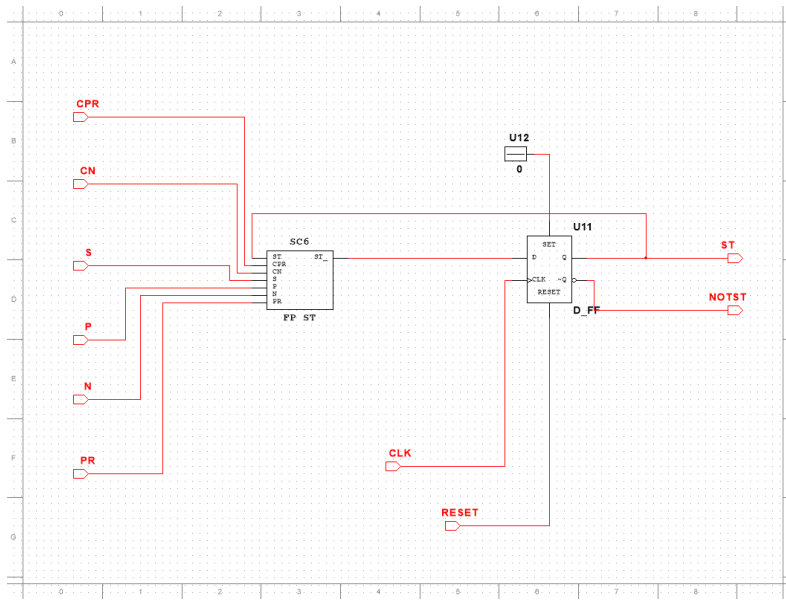
```

ST_ = OR(
    AND(CPR, NOT(N), NOT(PR)),
    AND(CN, CPR, NOT(PR)),
    AND(CN, NOT(N)),
    AND(ST, NOT(P)),
    S
)

```



Rysunek 5: Podukład FP ST



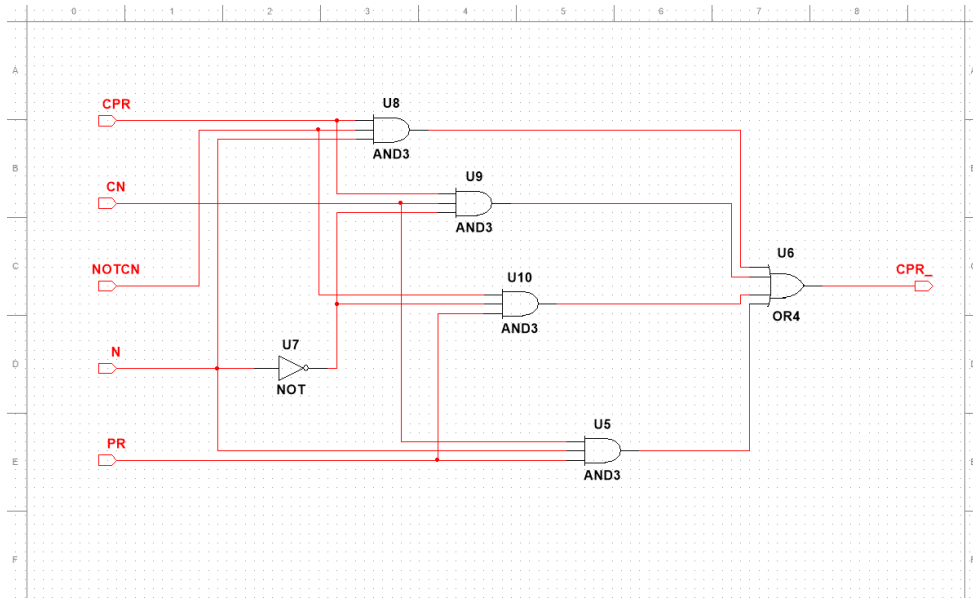
Rysunek 6: Podukład HANDLE ST

4.2 CPR

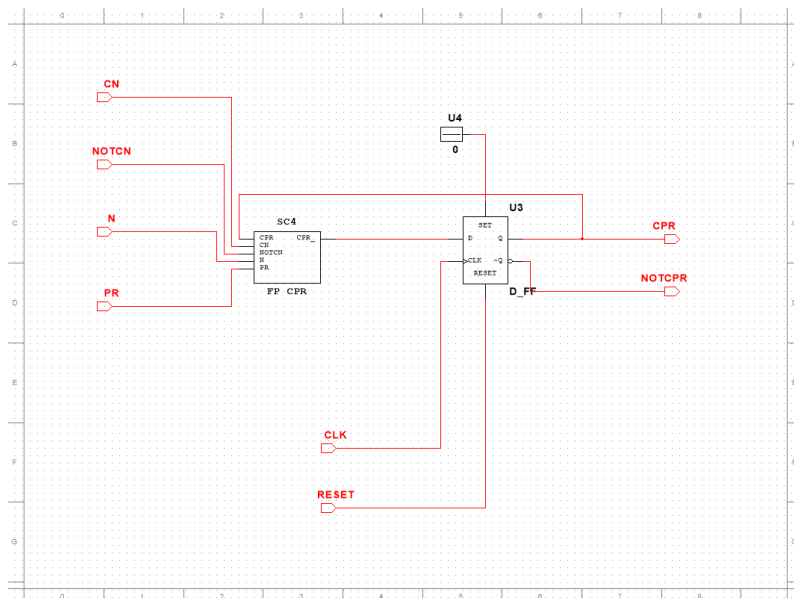
Kod symulujący funkcję stanu CPR

$$\begin{aligned}
 CPR^{(n+1)} = & \overline{(CN)^{(n)}} \overline{N}(PR) \\
 & + \overline{(CN)^{(n)}} N(CPR)^{(n)} \\
 & + (CN)^{(n)} N(PR) \\
 & + (CN)^{(n)} \overline{N}(CPR)^{(n)}
 \end{aligned} \quad (2)$$

$$\begin{aligned}
 CPR_ = & \text{OR}(\\
 & \text{AND}(\text{NOT}(\text{CN}), \text{NOT}(\text{N}), \text{PR}), \\
 & \text{AND}(\text{NOT}(\text{CN}), \text{CPR}, \text{N}), \\
 & \text{AND}(\text{CN}, \text{N}, \text{PR}), \\
 & \text{AND}(\text{CN}, \text{CPR}, \text{NOT}(\text{N})) \\
 &)
 \end{aligned}$$



Rysunek 7: Podukład FP CPR

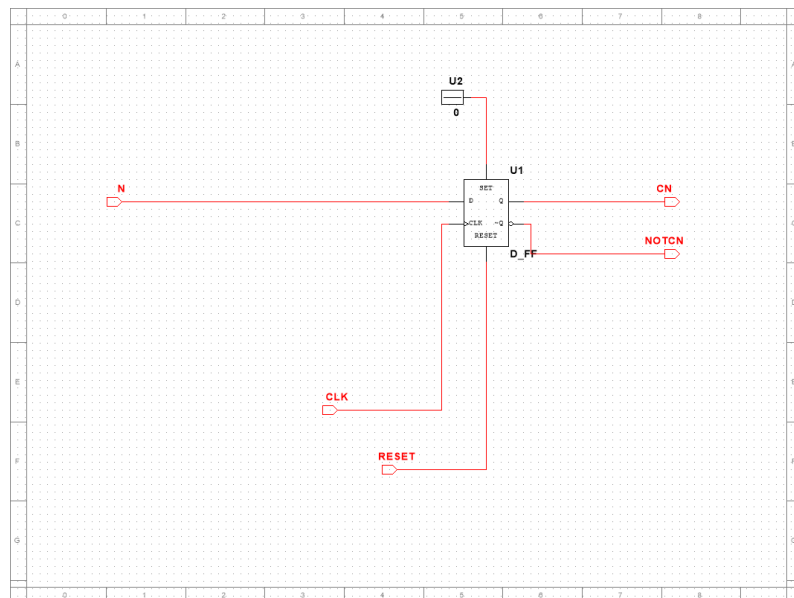


Rysunek 8: Podukład HANDLE CPR

4.3 CN

$$CN^{(n+1)} = N \quad (3)$$

Kod symulujący funkcję stanu CN : $CN_ = N$



Rysunek 9: Podukład HANDLE CN

4.4 R_0

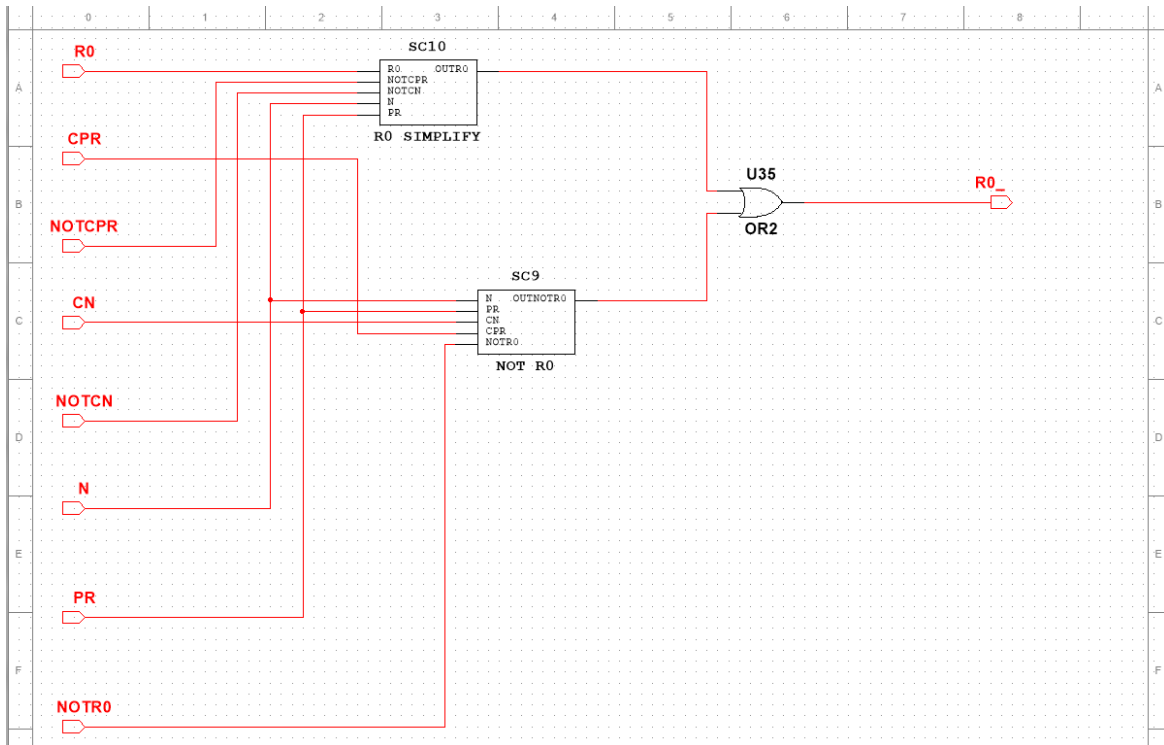
Kod symulujący funkcję stanu R_0

$$\begin{aligned}
 R_0^{(n+1)} = & \overline{R_0^{(n)}} \overline{N} \overline{(PR)} (CPR)^{(n)} \\
 & + \overline{R_0^{(n)}} \overline{(PR)} (CN)^{(n)} (CPR)^{(n)} \\
 & + \overline{R_0^{(n)}} \overline{N} (CN)^{(n)} \\
 & + R_0^{(n)} \overline{(CN)^{(n)}} \overline{(CPR)^{(n)}} \\
 & + R_0^{(n)} \overline{(CN)^{(n)}} (PR) \\
 & + R_0^{(n)} \overline{(CN)^{(n)}} N \\
 & + R_0^{(n)} \overline{(CPR)^{(n)}} N \\
 & + R_0^{(n)} N (PR)
 \end{aligned} \quad (4)$$

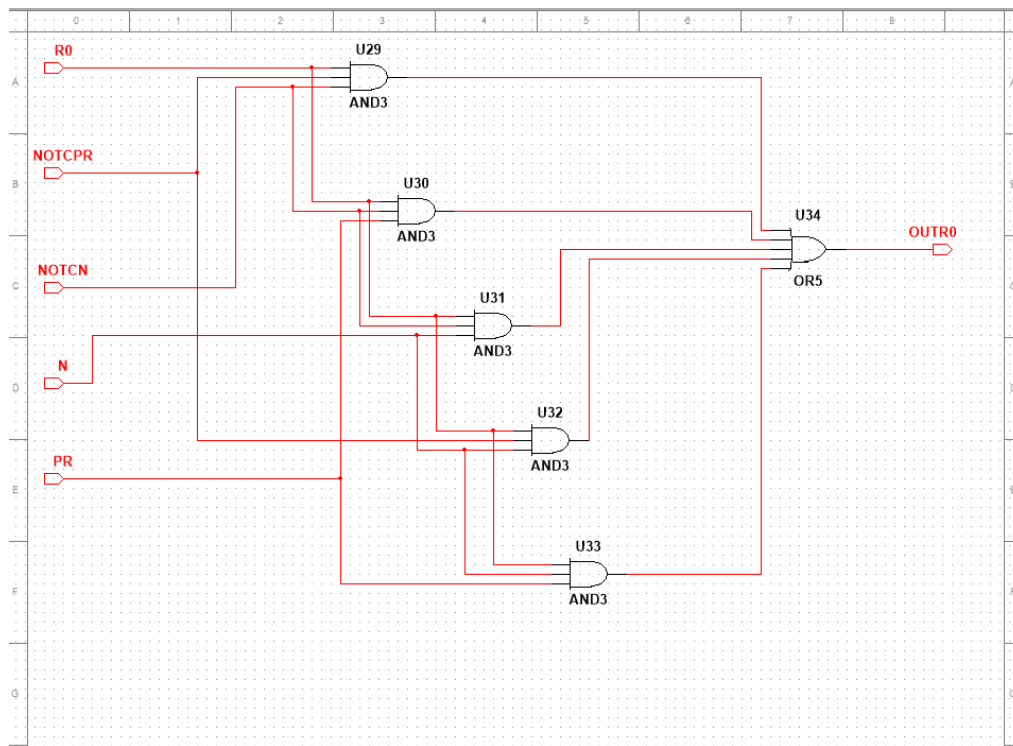
```

RO_ = OR(
    AND(NOT(RO), CPR, NOT(N), NOT(PR)),
    AND(NOT(RO), CN, CPR, NOT(PR)),
    AND(NOT(RO), CN, NOT(N)),
    AND(RO, NOT(CN), NOT(CPR)),
    AND(RO, NOT(CN), PR),
    AND(RO, NOT(CN), N),
    AND(RO, NOT(CPR), N),
    AND(RO, N, PR)
)

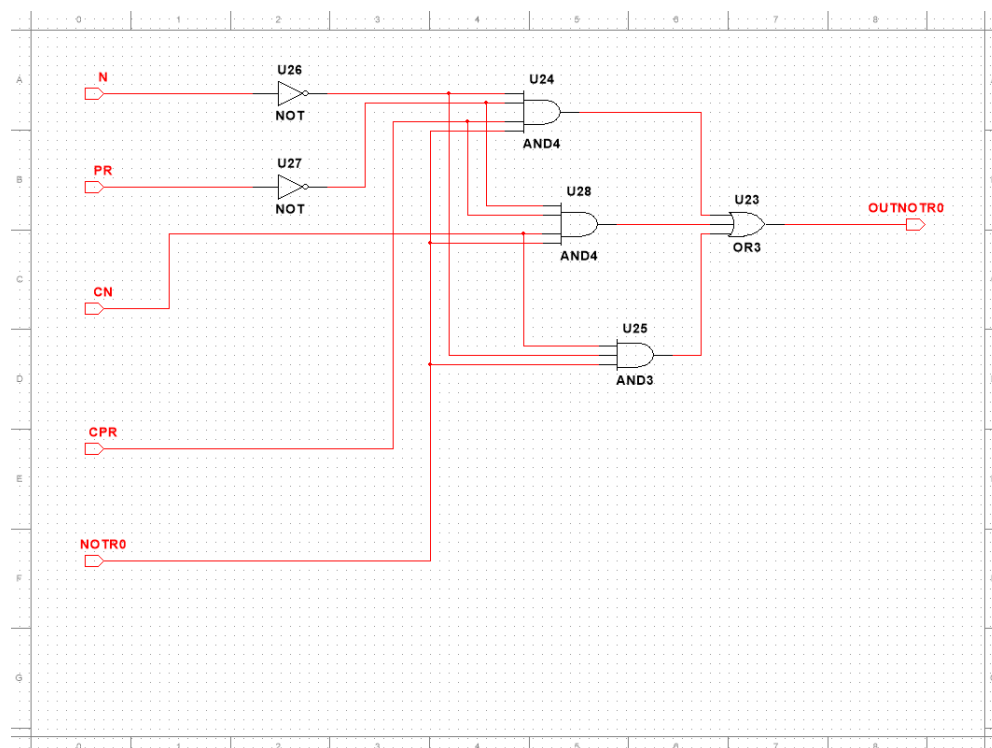
```



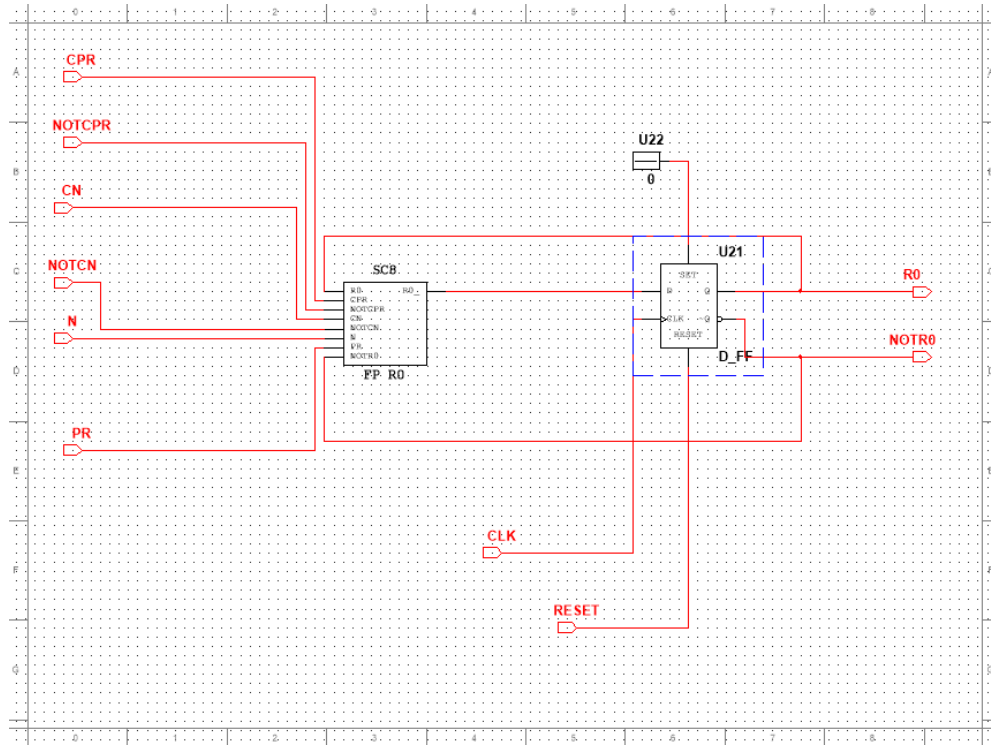
Rysunek 10: Podukład FP R0



Rysunek 11: Podukład R0 SIMPLIFY



Rysunek 12: Podukład NOT R0



Rysunek 13: Podukład HANDLE R0

4.5 R_1

Kod symulujący funkcję stanu R_1

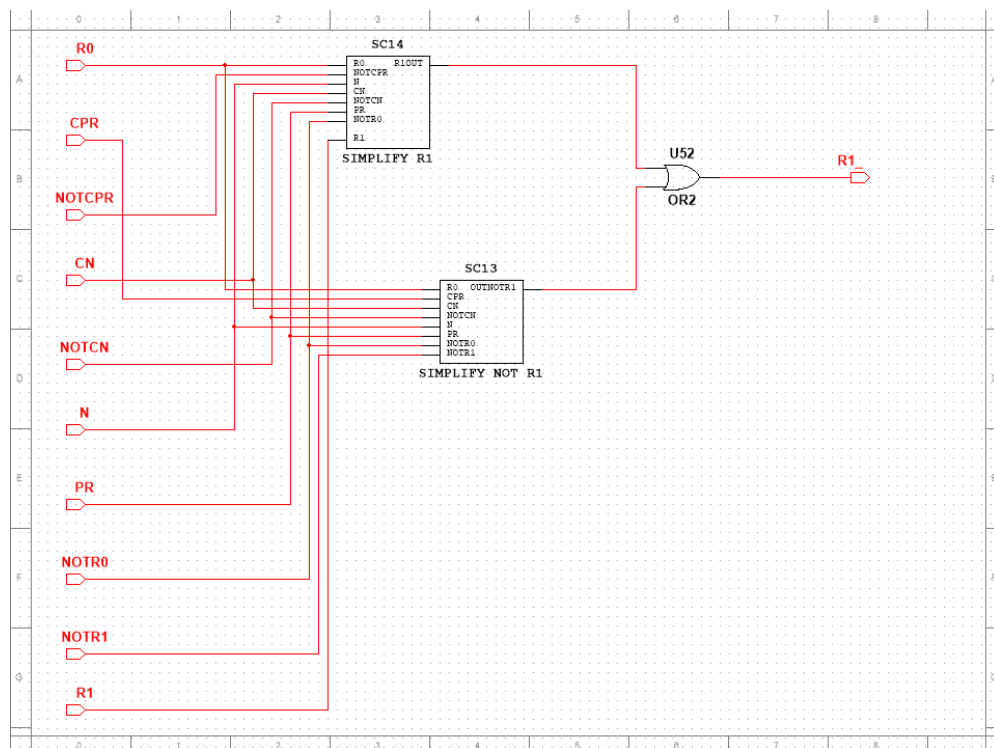
$$\begin{aligned}
 R_1^{(n+1)} = & \overline{R_1^{(n)}} \overline{R_0^{(n)}} \overline{(CN)^{(n)}} (CPR)^{(n)} \overline{N} \overline{(PR)} \\
 & + \overline{R_1^{(n)}} \overline{R_0^{(n)}} (CN)^{(n)} (CPR)^{(n)} \overline{N} \overline{(PR)} \\
 & + \overline{R_1^{(n)}} \overline{R_0^{(n)}} (CN)^{(n)} \overline{N} \\
 & + \overline{R_0^{(n)}} R_1^{(n)} (CN)^{(n)} \overline{N} \\
 & + \overline{R_0^{(n)}} R_1^{(n)} \overline{(CPR)^{(n)}} \\
 & + \overline{R_0^{(n)}} R_1^{(n)} (PR) \\
 & + \overline{(CN)^{(n)}} R_1^{(n)} N \\
 & + R_1^{(n)} R_0^{(n)} \overline{(CN)^{(n)}} \\
 & + R_1^{(n)} R_0^{(n)} N
 \end{aligned}$$

(5)

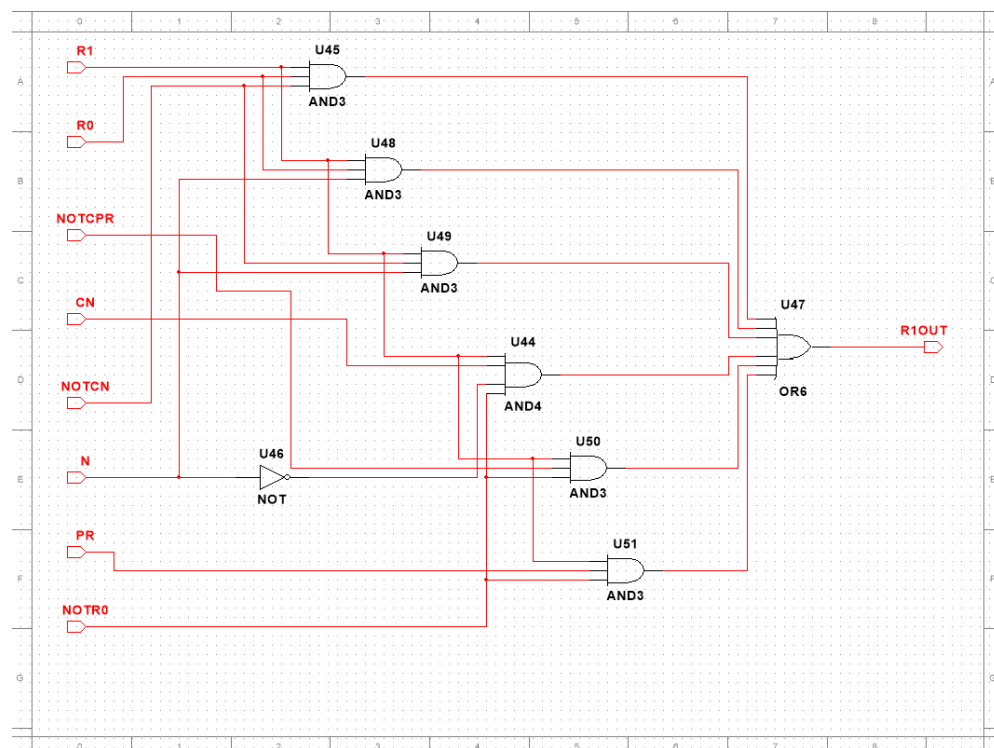
```

R1_ = OR(
    AND(NOT(R1), NOT(R0), NOT(CN), CPR, NOT(N),
        NOT(PR)),
    AND(NOT(R1), NOT(R0), CN, CPR, N, NOT(PR)),
    AND(NOT(R1), R0, CN, NOT(N)),
    AND(R1, NOT(R0), CN, NOT(N)),
    AND(R1, NOT(R0), NOT(CPR)),
    AND(R1, NOT(R0), PR),
    AND(R1, NOT(CN), N),
    AND(R1, R0, NOT(CN)),
    AND(R1, R0, N)
)

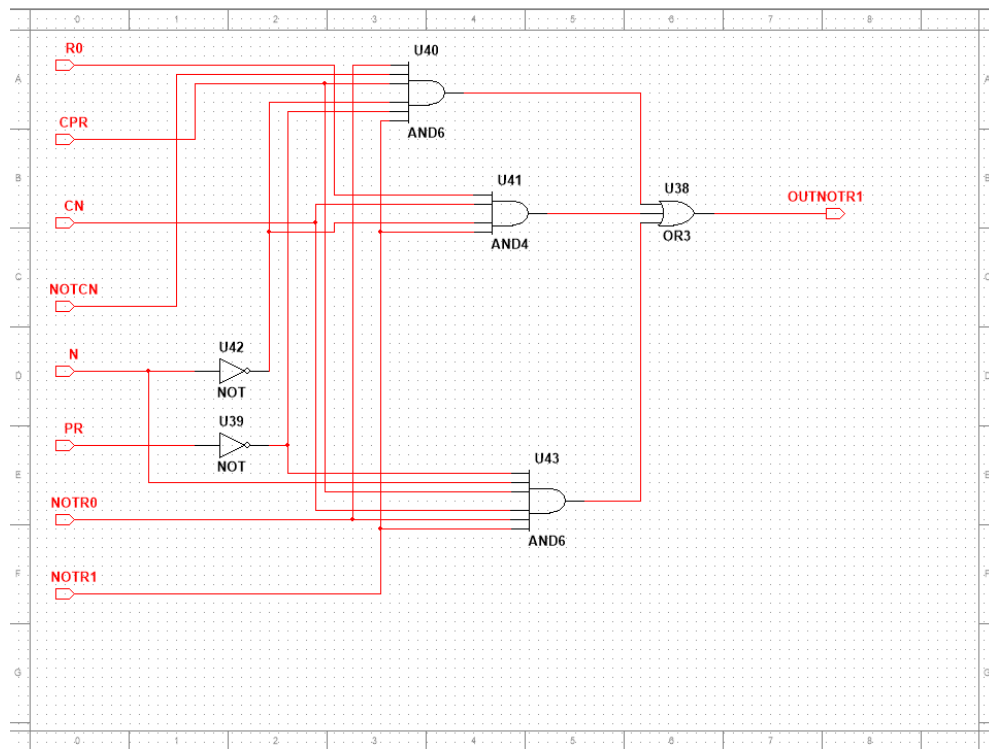
```



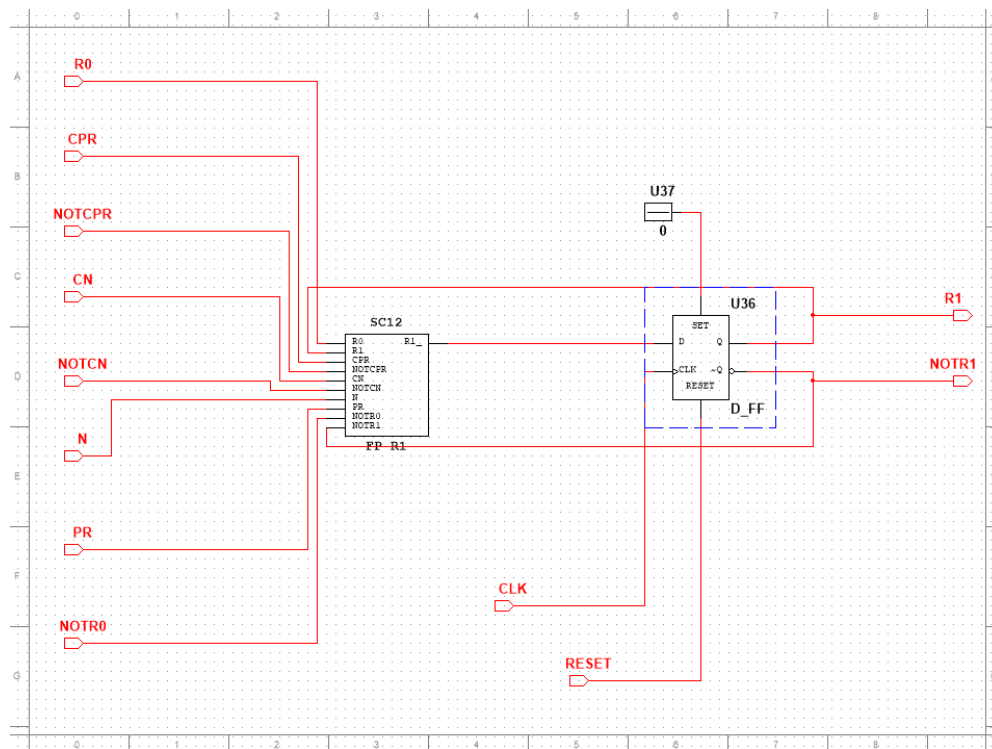
Rysunek 14: Podukład FP R1



Rysunek 15: Podukład SIMPLIFY R1



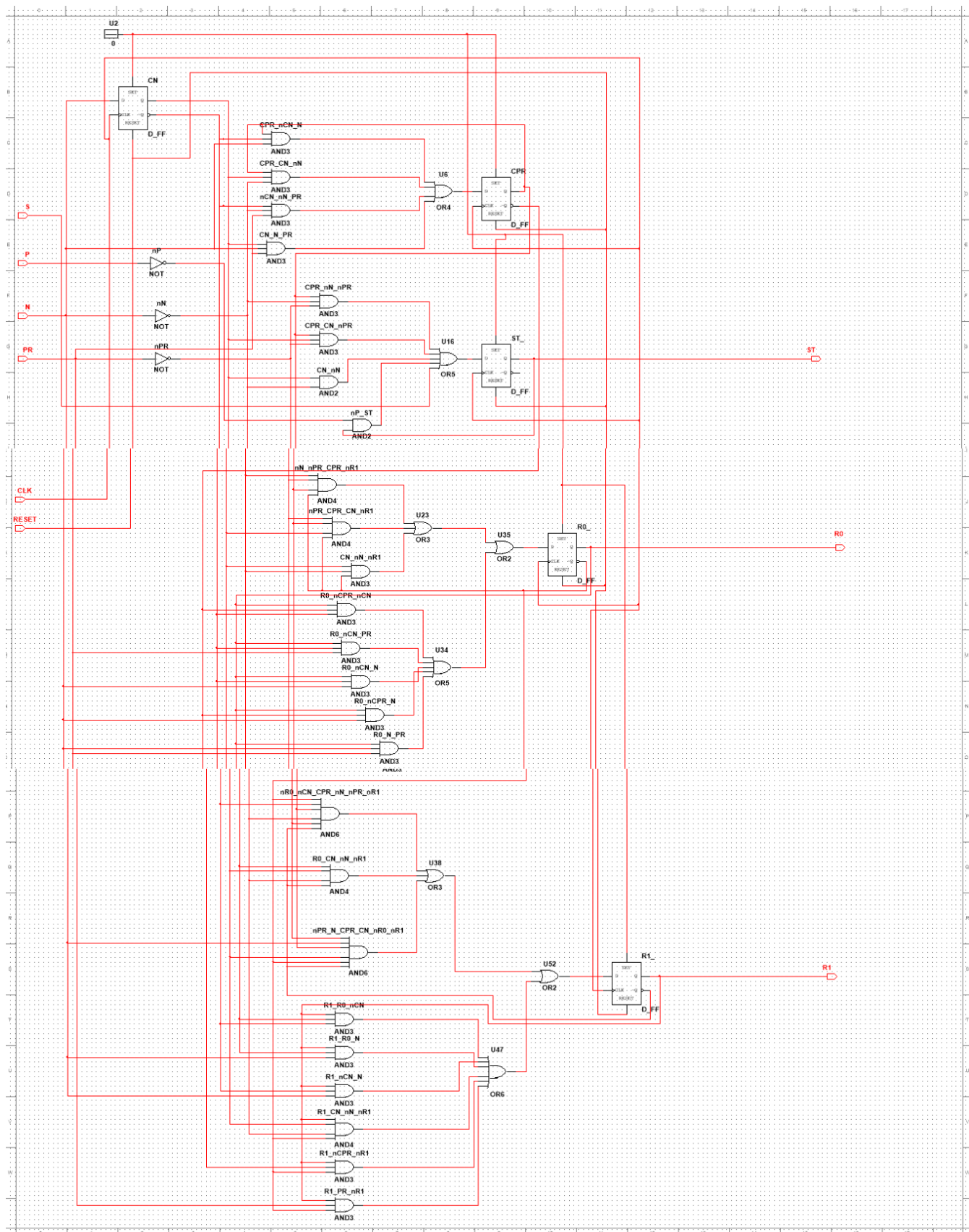
Rysunek 16: Podukład SIMPLIFY NOT R1



Rysunek 17: Podukład HANDLE R1

5 Schemat układu

Po umieszczeniu wszystkich bramek i przerzutników w jednym podukładzie i usunięciu powtarzających się bramek otrzymaliśmy nasz finalny układ:

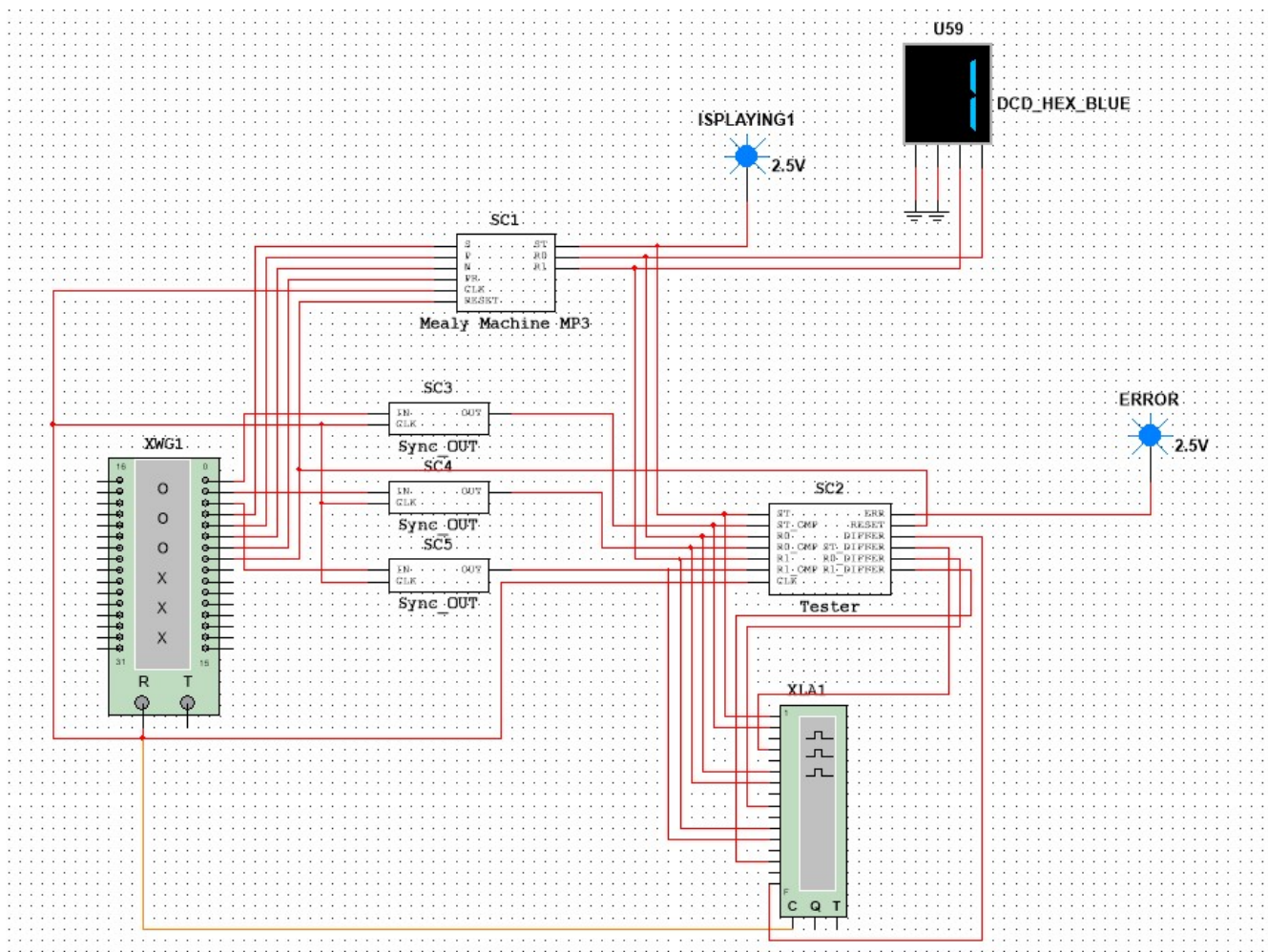


Rysunek 18: Zaprojektowany automat w programie Multisim

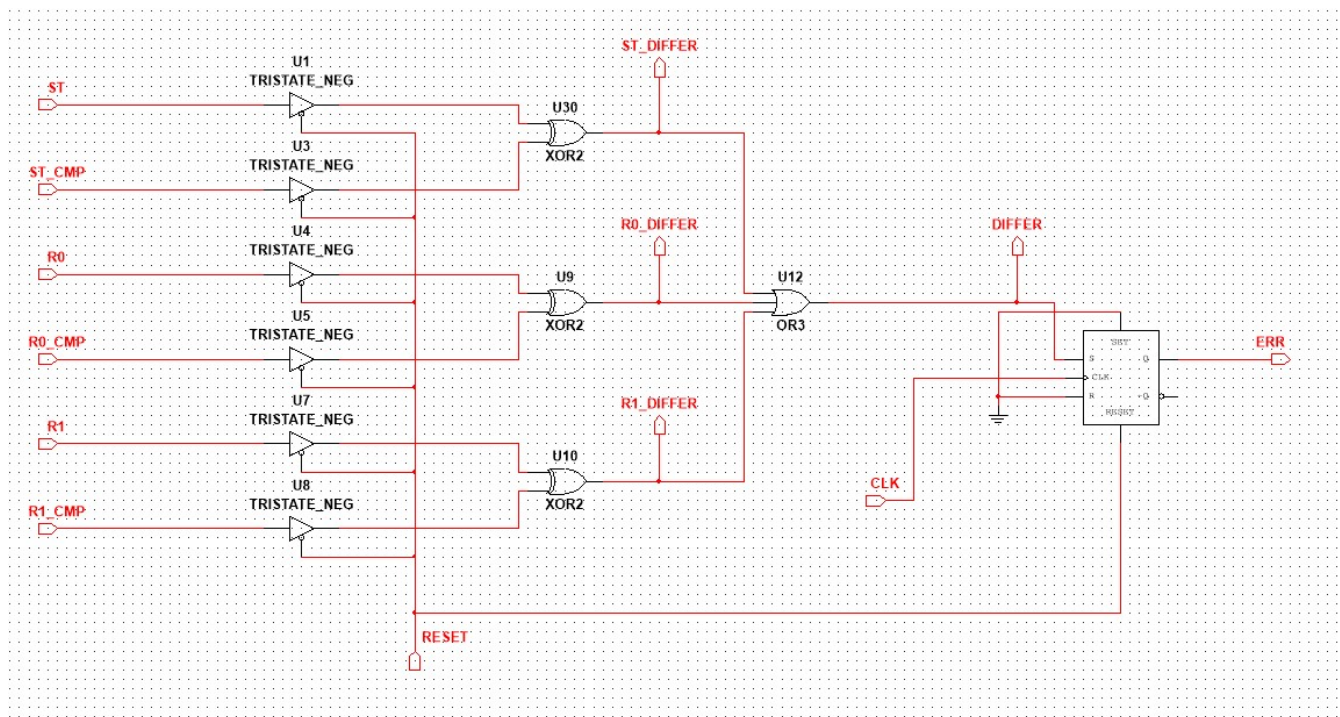
6 Stanowisko testujące

Układ testujący, podobnie jak w przypadku ćw. 2. oparty jest o podukład **Tester** obliczający różnice między wartościami oczekiwanymi wyjścia, a wyjściem testowanego układu, oraz analizator stanów logicznych (**XLA1**) umożliwiający

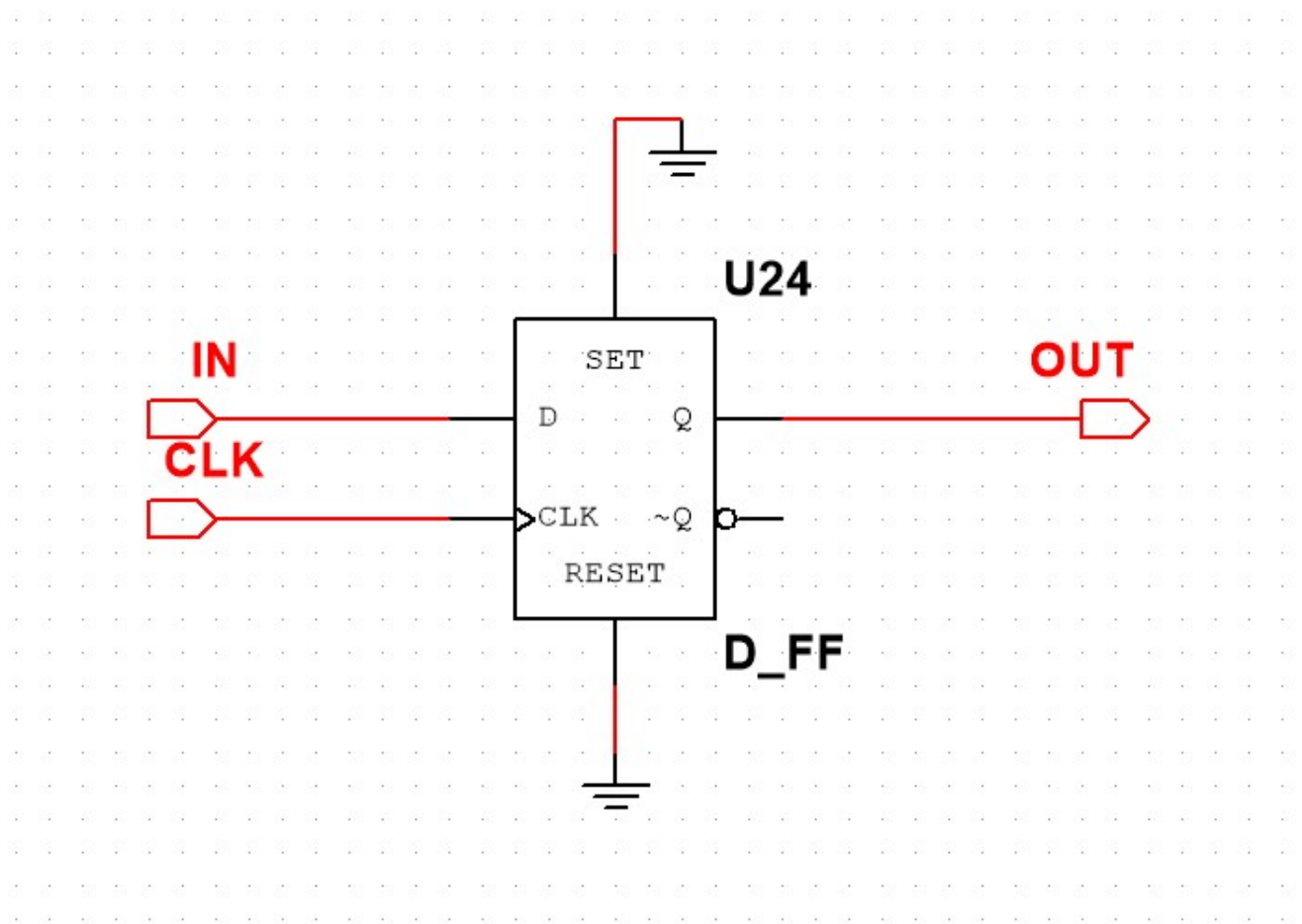
obserwowanie i zapisywanie błędów w czasie rzeczywistym. Generowane przez generator słów (XWG1) przypadki testowe, zsynchronizowane są z wyjściem testowanego układu za pomocą przerzutnika typu D (podukłady Sync_OUT).



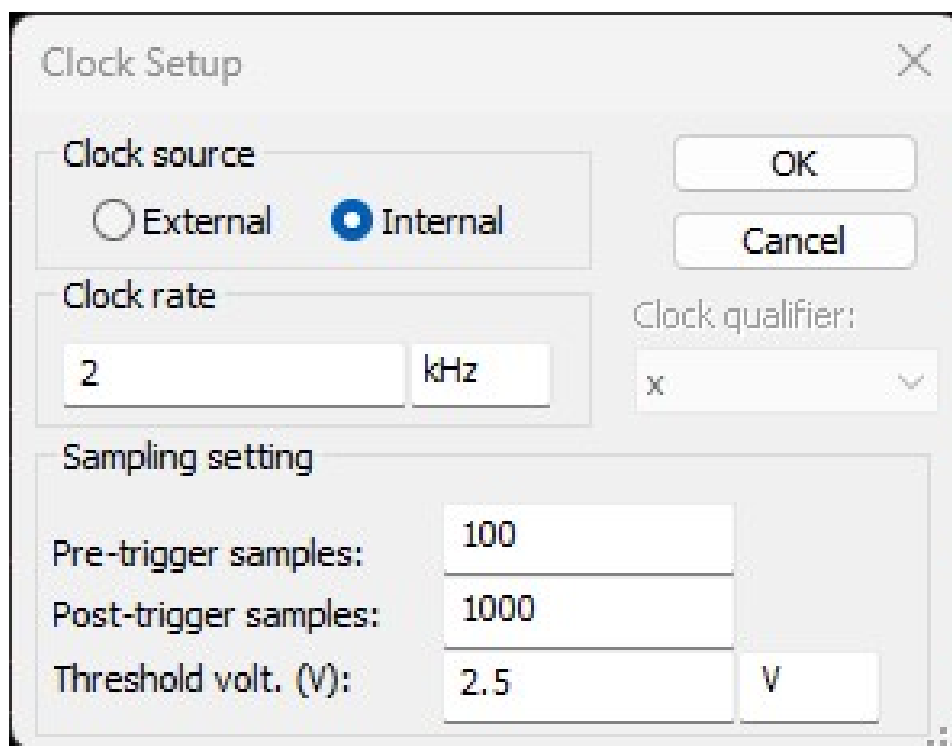
Rysunek 19: Stanowisko testujące



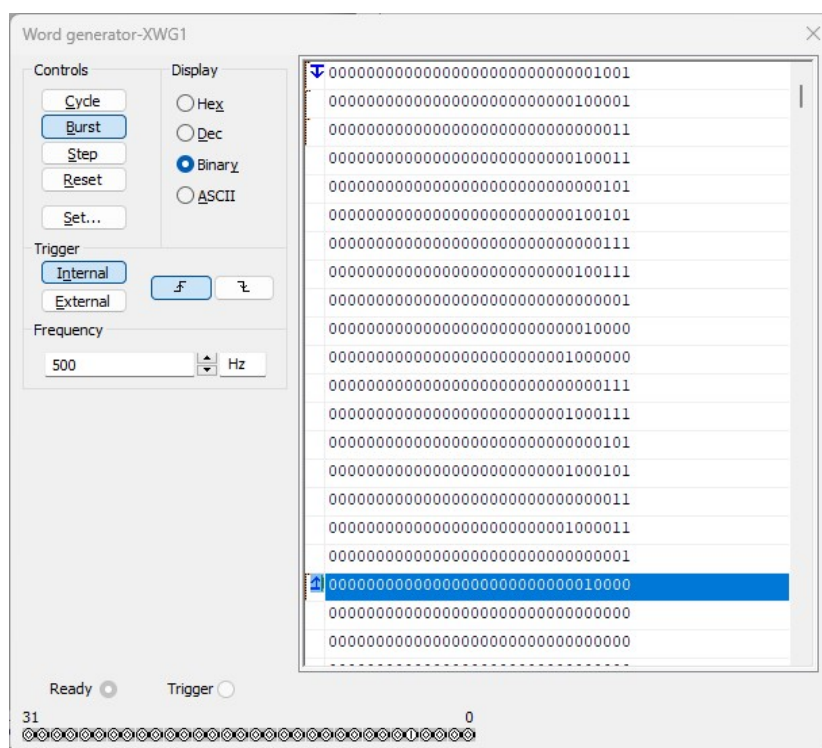
Rysunek 20: Podukład Tester



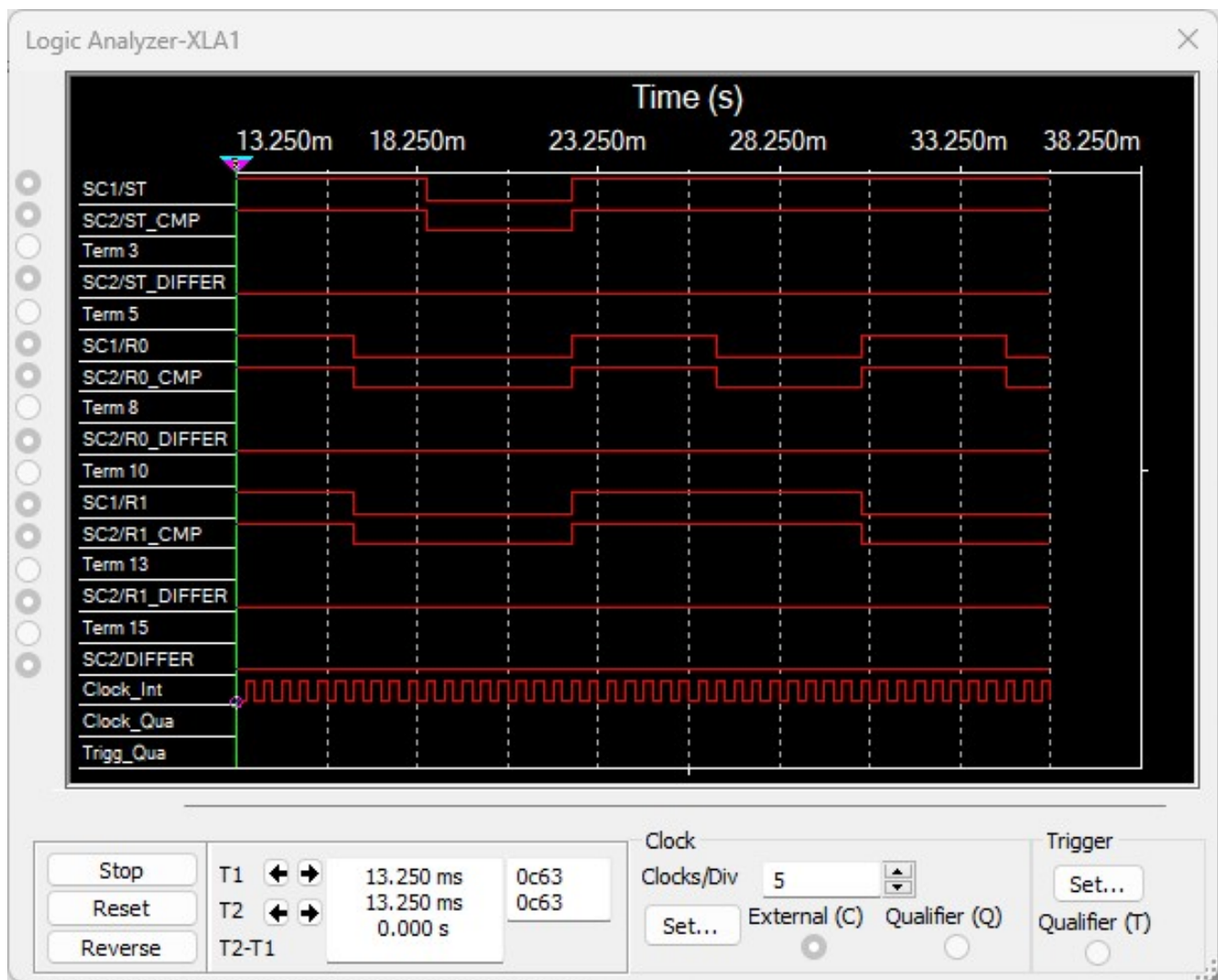
Rysunek 21: Podukład Sync_OUT



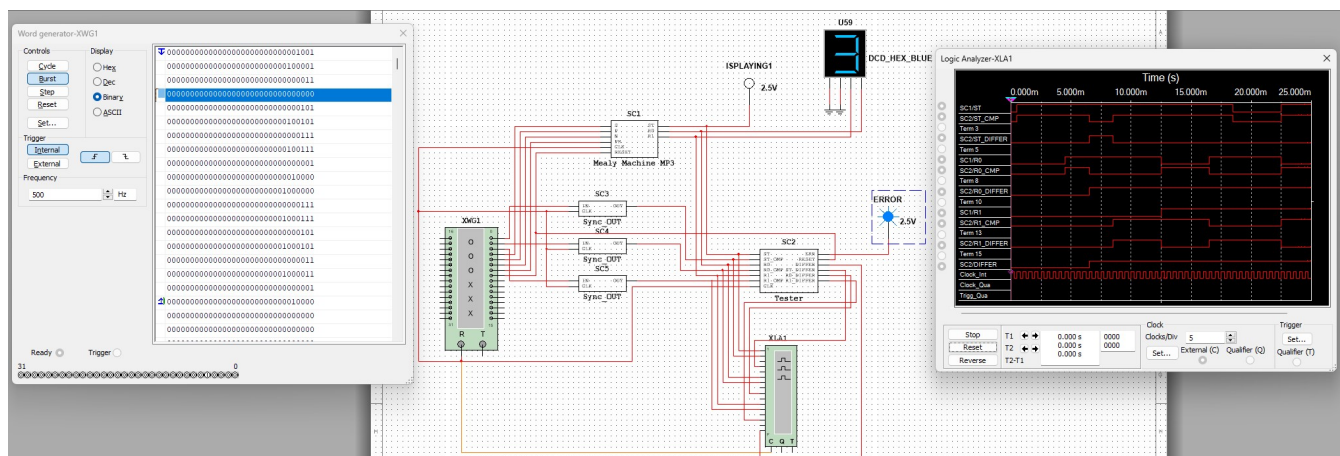
Rysunek 22: Ustawienia analizatora XLA1



Rysunek 23: Ustawienia generatora słów XWG1



Rysunek 24: Odczyt analizatora przy poprawnym działaniu układu



Rysunek 25: Odczyt analizatora przy niepoprawnym działaniu układu

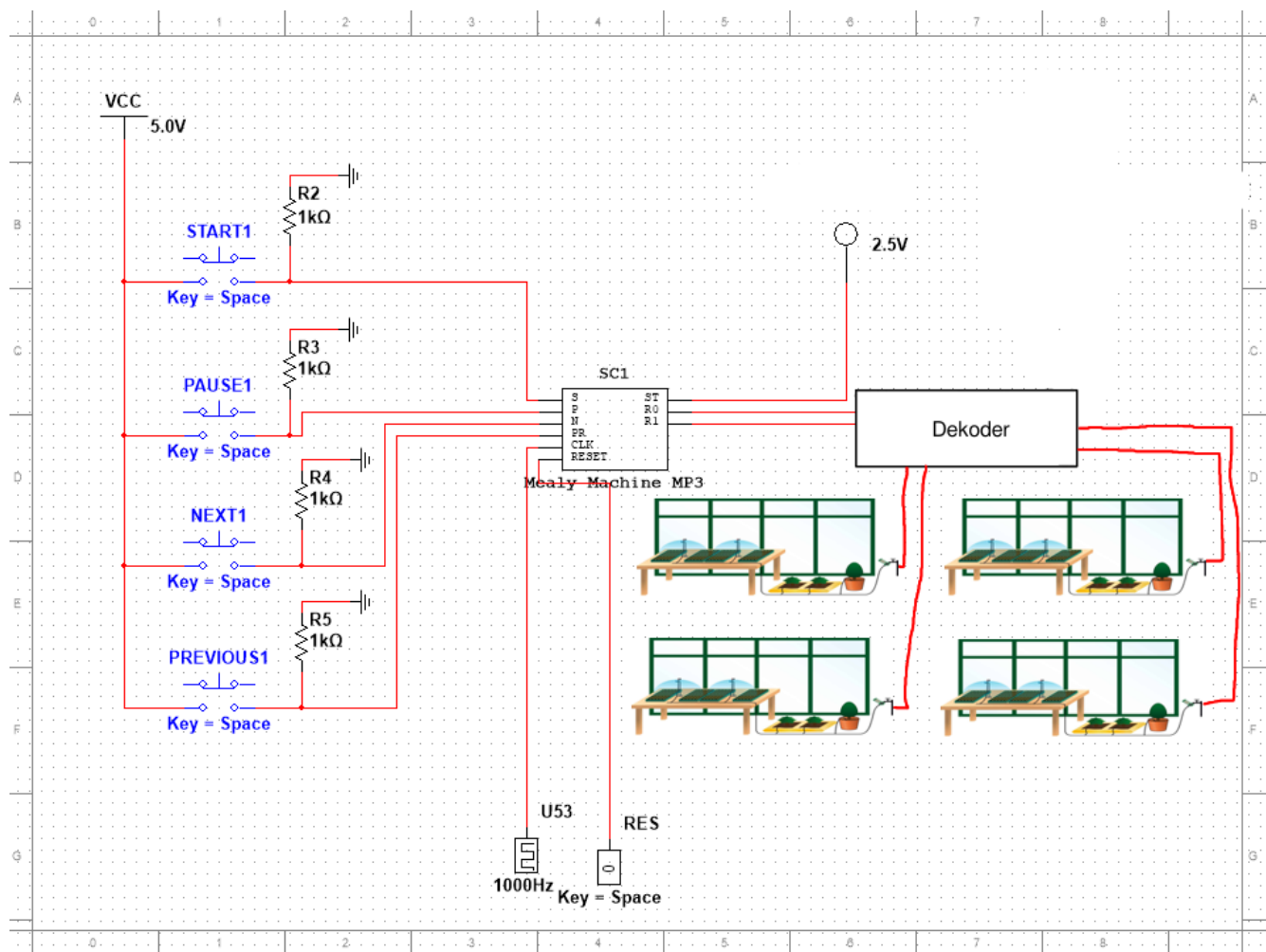
7 Podsumowanie oraz wnioski

Ćwiczenie to było najbardziej skomplikowane z dotychczasowych ze względu na dużą liczbę kroków i nieoczywistych decyzji jakie należało podjąć w każdym z nich. Sam automat zaprojektować można było na kilka różnych sposobów. Kluczowym wyborem był tutaj sam rodzaj automatu. Podjęliśmy decyzję, aby oprzeć naszą implementację o automat

Mealy'ego (zamiast automatu Moore'a) ze względu na, w naszym mniemaniu, prostsze rozumowanie na temat poprawności działania projektowanego układu, co mogło jednak skutkować bardziej skomplikowanym finalnym projektem. W końcu, automatyczna minimalizacja, z wykorzystaniem nieheurystycznej metody Quine'a-McCluskeya, znacznie przyspieszyła pracę nad finalną implementacją i upewniła nas o optymalności naszego rozwiązania ale, równocześnie, pozbawiła nas intuicji i głębszego zrozumienia funkcji stanu oraz wyjścia, jakie dałaby ręczna minimalizacja metodą tablic Karnaugh.

Zaprojektowany automat może znaleźć wiele nieoczywistych zastosowań poza odtwarzaczem MP3, może być częścią niemalże dowolnego układu, którego istotnym elementem jest kontrola i aktywacja przez użytkownika różnych trybów działania m.in.:

- urządzenia treningowe i rehabilitacyjne
- systemy wywołań w biurach lub warsztatach
- kontrolery systemów w przemyśle



Rysunek 26: Przykładowe zastosowanie układu - system nawadniania ogrodu szklarniowego