Sterownik odtwarzacza mp3

Szymon Borusiewicz, Jakub Zając, Dawid Szłapa, Radosław Szepielak

Maj 2025

1 Temat

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

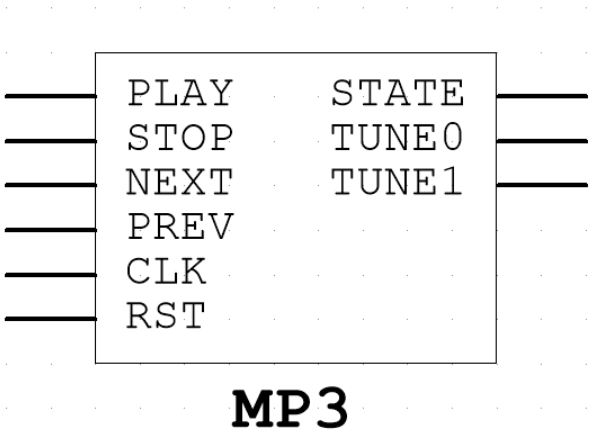
Układ powinien mieć następujące przyciski oraz odpowiadające im sygnały i wskaźniki:

- STOP
- PLAY
- NEXT
- PREVIOUS

oraz powinien posiadać dwubitowe wyjście binarne określające numer utworu.

2 Czarna skrzynka

Układ, który chcemy uzyskać będzie prezentował się następująco:



Rysunek 1: Oczekiwany układ

gdzie wejściami będą:

- **PLAY** – wznawianie odtwarzanie utworu,
- **STOP** – zatrzymanie odtwarzanie utworu,
- **NEXT** – przełączenie utworu na kolejny,
- **PREV** – przełączenie utworu na poprzedni,
- **CLK** – wejście zegara,
- **RST** – reset,

a wyjściami:

- **STATE** – aktualny stan odtwarzania
  (0 – utwór wznowiony, 1 – utwór zatrzymany),
- **TUNE0** – młodszy bit określający piosenkę,
- **TUNE1** – starszy bit określający piosenkę.

3 Automat

Pożądany przez nas automat pozwala na zmianę stanu odtwarzania utworu oraz na zmianę utworu na poprzedni lub następny.

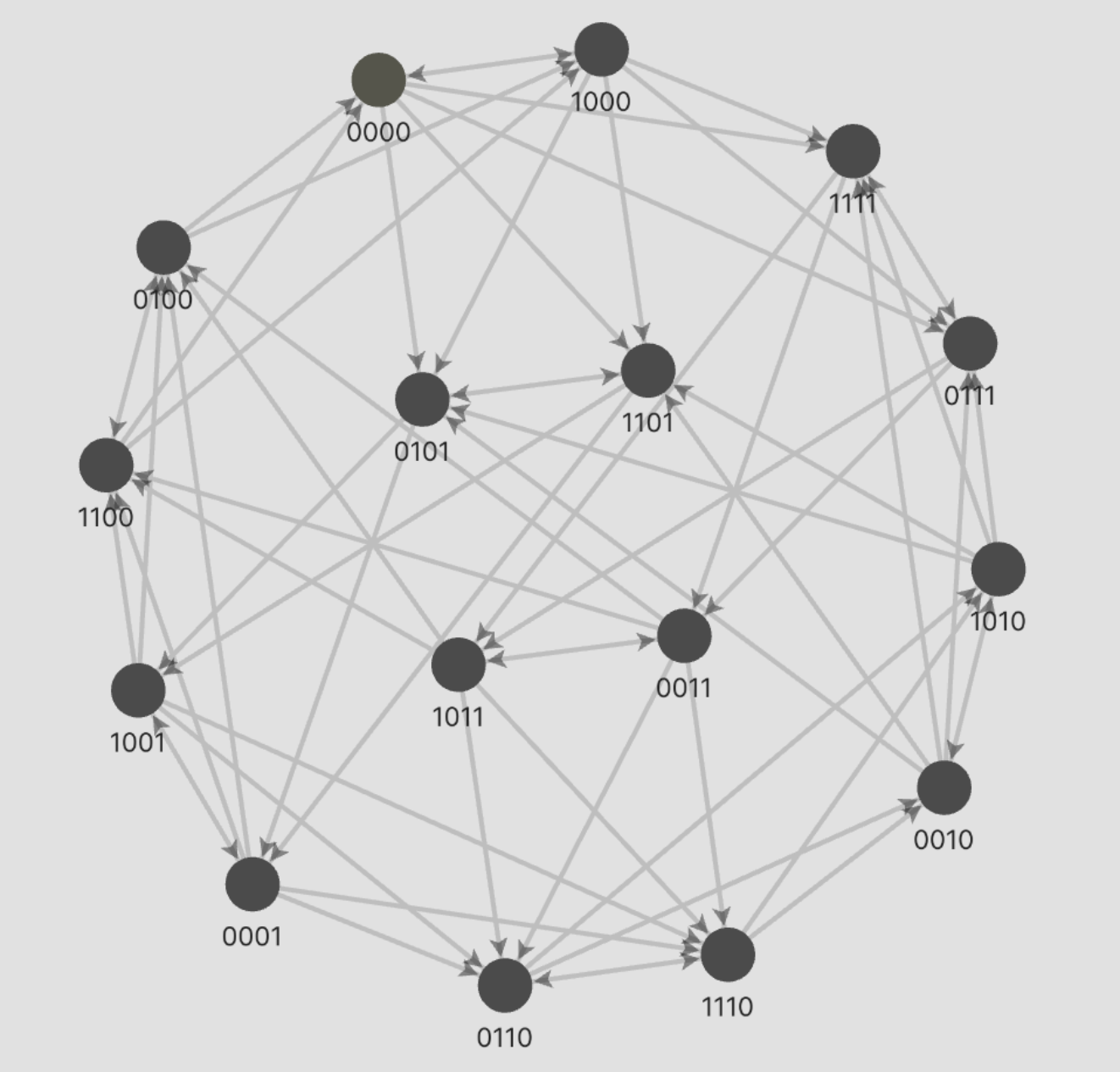
W celu uniknięcia wielokrotnej aktywacji PREV i NEXT wprowadzamy bit blokujący, który odblokowuje się dopiero przy nie wysyłaniu sygnału z PREV i NEXT.

Ponadto zakładamy możliwość jednoczesnej zmiany stanu odtwarzania i utworu.

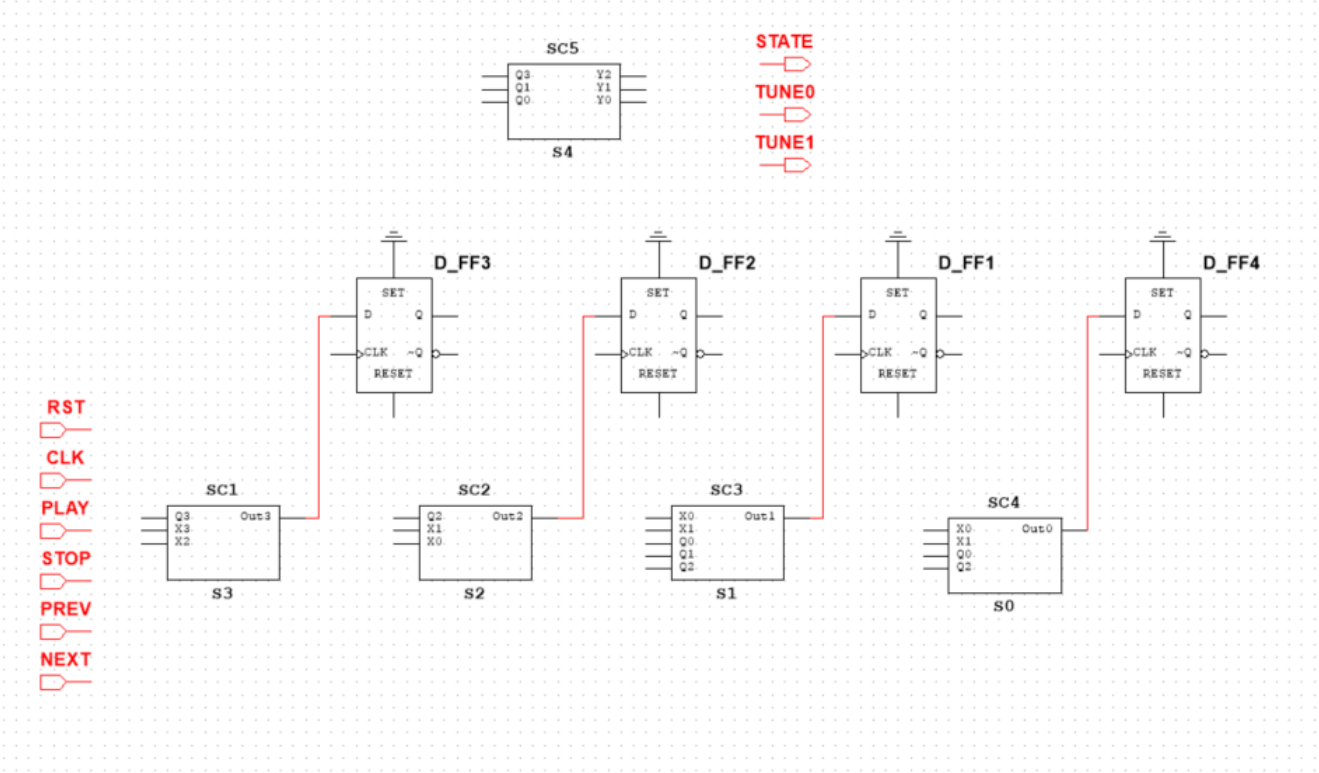
Stąd:

- Q_3 – bit odpowiedzialny za aktualny stan odtwarzania (0 – utwór wznowiony, 1 – utwór zatrzymany),
- Q_2 – bit odpowiedzialny za blokadę wielokrotnego działania PREV i NEXT,
- Q_1 – młodszy bit określający piosenkę,
- Q_0 – starszy bit określający piosenkę

3.a Projekt



Rysunek 2: Graf przedstawiający automat



Rysunek 3: Szkielet automatu

4 Tabela prawdy sterownika

Reprezentuje ona wyjścia układu w zależności od stanu układu. Możemy zauważyć, że wyjście układu jest takie samo jak jego stan, z pominięciem bitu blokującego.

Y w tabeli prawdy odpowiadają sygnałom wyjściowym układu:

- Y_2 – State,
- Y_1 – Tune0,
- Y_0 – Tune1

Q				Y		
Q_3	Q_2	Q_1	Q_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	1

0	1	1	0	0	1	0
0	1	1	1	0	1	1
1	0	0	0	1	0	0
1	0	0	1	1	0	1
1	0	1	0	1	1	0
1	0	1	1	1	1	1
1	1	0	0	1	0	0
1	1	0	1	1	0	1
1	1	1	0	1	1	0
1	1	1	1	1	1	1

5 Tabela prawdy automatu

Poprzez X oznaczymy sygnały wejściowe układu:

- X_3 – PLAY
- X_2 – STOP
- X_1 – NEXT
- X_0 – PREV

Q				X				Q_+			
Q_3	Q_2	Q_1	Q_0	X_3	X_2	X_1	X_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	1	1
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	1	1	0	1	1	0	1
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	1	1	0	1	0	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	0	0	1	0	1
0	0	0	0	1	0	1	0	0	1	0	1
0	0	0	0	1	0	1	1	0	0	0	0
0	0	0	0	1	0	0	1	0	1	1	1
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0	1	0	0
0	0	0	1	0	0	1	1	0	0	0	1
0	0	0	1	0	0	1	0	0	1	1	0
0	0	0	1	0	1	1	0	1	1	1	0
0	0	0	1	0	1	1	1	1	0	0	1
0	0	0	1	0	1	0	1	1	1	0	0
0	0	0	1	0	1	0	0	1	0	0	1
0	0	0	1	1	1	0	0	0	0	0	1
0	0	0	1	1	1	0	1	0	1	0	0
0	0	0	1	1	1	1	1	0	0	0	1
0	0	0	1	1	1	1	0	0	1	1	0
0	0	0	1	1	0	1	0	0	1	1	0
0	0	0	1	1	0	1	1	0	0	0	1
0	0	0	1	1	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0	1	1
0	0	1	1	0	0	0	1	0	1	1	0
0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	1	0	0
0	0	1	1	0	1	1	0	1	1	0	0
0	0	1	1	0	1	0	1	1	1	1	0
0	0	1	1	0	1	0	0	1	0	1	1
0	0	1	1	1	1	0	0	0	0	1	1
0	0	1	1	1	1	0	1	0	1	1	0
0	0	1	1	1	1	1	1	0	0	1	1
0	0	1	1	1	1	1	0	0	1	0	0
0	0	1	1	1	0	1	0	0	1	0	0
0	0	1	1	1	0	1	1	0	0	1	1
0	0	1	1	1	0	0	1	0	1	1	0
0	0	1	1	1	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	1	0	1	0	1
0	0	1	0	0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	0	0	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1	0	1	0
0	0	1	0	0	1	0	1	1	1	0	1
0	0	1	0	0	1	0	0	1	0	1	0
0	0	1	0	1	1	0	0	0	0	1	0
0	0	1	0	1	1	0	1	0	1	0	1
0	0	1	0	1	1	1	1	0	0	1	0
0	0	1	0	1	1	1	0	0	1	1	1
0	0	1	0	1	0	1	0	0	1	1	1
0	0	1	0	1	0	1	1	0	0	1	0
0	0	1	0	1	0	0	1	0	1	0	1
0	0	1	0	1	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0	1	1	0
0	1	1	0	0	0	1	1	0	1	1	0
0	1	1	0	0	0	1	0	0	1	1	0
0	1	1	0	0	1	1	0	1	1	1	0
0	1	1	0	0	1	1	1	1	1	1	0
0	1	1	0	0	1	0	1	1	1	1	0
0	1	1	0	0	1	0	0	1	0	1	0
0	1	1	0	1	1	0	0	0	0	1	0
0	1	1	0	1	1	0	1	0	1	1	0
0	1	1	0	1	1	1	1	0	1	1	0

0	1	1	0	1	1	1	0	0	1	1	0
0	1	1	0	1	0	1	0	0	1	1	0
0	1	1	0	1	0	1	1	0	1	1	0
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	0	1	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0	1	1
0	1	1	1	0	0	0	1	0	1	1	1
0	1	1	1	0	0	1	1	0	1	1	1
0	1	1	1	0	0	1	0	0	1	1	1
0	1	1	1	0	1	1	0	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1
0	1	1	1	0	1	0	1	1	1	1	1
0	1	1	1	0	1	0	0	1	0	1	1
0	1	1	1	1	1	0	0	0	0	1	1
0	1	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	0	0	1	1	1
0	1	1	1	1	0	1	0	0	1	1	1
0	1	1	1	1	0	1	1	0	1	1	1
0	1	1	1	1	0	0	1	0	1	1	1
0	1	1	1	1	0	0	0	0	0	1	1
0	1	0	1	0	0	0	0	0	0	0	1
0	1	0	1	0	0	0	1	0	1	0	1
0	1	0	1	0	0	1	1	0	1	0	1
0	1	0	1	0	0	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	0	1
0	1	0	1	0	1	1	1	1	1	0	1
0	1	0	1	0	1	0	1	1	1	0	1
0	1	0	1	0	1	0	0	1	0	0	1
0	1	0	1	1	1	0	0	0	0	0	1
0	1	0	1	1	1	0	1	0	1	0	1
0	1	0	1	1	0	1	0	0	1	0	1
0	1	0	1	1	0	1	1	0	1	0	1
0	1	0	1	1	0	0	1	0	1	0	1
0	1	0	1	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0

0	1	0	0	0	0	0	1	0	1	0	0
0	1	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	1	0	0
0	1	0	0	0	1	1	0	1	1	0	0
0	1	0	0	0	1	1	1	1	1	0	0
0	1	0	0	0	1	0	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	0	0
0	1	0	0	1	1	0	0	0	0	0	0
0	1	0	0	1	1	0	1	0	1	0	0
0	1	0	0	1	1	1	1	0	1	0	0
0	1	0	0	1	1	1	0	0	1	0	0
0	1	0	0	1	0	1	0	0	1	0	0
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	0	1	0	0	1	0	1	0	0
0	1	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	1	1	0	0
1	1	0	0	0	0	1	1	1	1	0	0
1	1	0	0	0	0	1	0	1	1	0	0
1	1	0	0	0	1	1	0	1	1	0	0
1	1	0	0	0	1	1	1	1	1	0	0
1	1	0	0	0	1	0	1	1	1	0	0
1	1	0	0	0	1	0	0	1	0	0	0
1	1	0	0	1	1	0	0	1	0	0	0
1	1	0	0	1	1	0	1	1	1	0	0
1	1	0	0	1	1	1	1	1	1	0	0
1	1	0	0	1	1	1	0	1	1	0	0
1	1	0	0	1	0	1	0	0	1	0	0
1	1	0	0	1	0	1	1	0	1	0	0
1	1	0	0	1	0	0	1	0	1	0	0
1	1	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	1	0	0	1
1	1	0	1	0	0	0	1	1	1	0	1
1	1	0	1	0	0	1	1	1	1	0	1
1	1	0	1	0	0	1	0	1	1	0	1
1	1	0	1	0	1	1	0	1	1	0	1
1	1	0	1	0	1	1	1	1	1	0	1
1	1	0	1	0	1	0	1	1	1	0	1

1	1	0	1	0	1	0	0	1	0	0	1
1	1	0	1	1	1	0	0	1	0	0	1
1	1	0	1	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	0	1
1	1	0	1	1	1	1	0	1	1	0	1
1	1	0	1	1	0	1	0	0	1	0	1
1	1	0	1	1	0	1	1	0	1	0	1
1	1	0	1	1	0	0	1	0	1	0	1
1	1	0	1	1	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	1	0	1	1
1	1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	0	1	1	1	1
1	1	1	1	0	1	1	0	1	1	1	1
1	1	1	1	0	1	0	1	1	1	1	1
1	1	1	1	0	1	0	0	1	0	1	1
1	1	1	1	1	1	0	0	1	0	1	1
1	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	0	1	0	0	1	1	1
1	1	1	1	1	0	1	1	0	1	1	1
1	1	1	1	1	0	0	1	0	1	1	1
1	1	1	1	1	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	1	0	1	0
1	1	1	0	0	0	0	1	1	1	1	0
1	1	1	0	0	0	0	1	1	1	1	0
1	1	1	0	0	0	1	1	1	1	1	0
1	1	1	0	0	0	1	0	1	1	1	0
1	1	1	0	0	1	1	0	1	1	1	0
1	1	1	0	0	1	1	1	1	1	1	0
1	1	1	0	0	1	0	1	1	1	1	0
1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	0	1	1	0	0	1	0	1	0
1	1	1	0	1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	0
1	1	1	0	1	1	1	0	1	1	1	0
1	1	1	0	1	0	1	0	0	1	1	0

1	1	1	0	1	0	1	1	0	1	1	0
1	1	1	0	1	0	0	1	0	1	1	0
1	1	1	0	1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1	1	1	0	1
1	0	1	0	0	0	1	1	1	0	1	0
1	0	1	0	0	0	1	0	1	1	1	1
1	0	1	0	0	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	0
1	0	1	0	0	1	0	1	1	1	0	1
1	0	1	0	0	1	0	0	1	0	1	0
1	0	1	0	1	1	0	0	1	0	1	0
1	0	1	0	1	1	0	1	1	1	0	1
1	0	1	0	1	1	1	1	1	0	1	0
1	0	1	0	1	1	1	0	1	1	1	1
1	0	1	0	1	0	1	0	0	1	1	1
1	0	1	0	1	0	0	0	0	0	1	0
1	0	1	1	0	0	0	0	1	0	1	1
1	0	1	1	0	0	0	1	1	1	1	0
1	0	1	1	0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0	1	1	0	0
1	0	1	1	0	1	1	0	1	1	0	0
1	0	1	1	0	1	1	1	1	0	1	1
1	0	1	1	0	1	0	1	1	1	1	0
1	0	1	1	1	1	0	0	1	0	1	1
1	0	1	1	1	1	0	1	1	1	1	0
1	0	1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	0	1	1	0	0
1	0	1	1	1	0	1	0	0	1	0	0
1	0	1	1	1	0	1	1	0	0	1	1
1	0	1	1	1	0	0	1	0	1	1	0
1	0	1	1	1	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1	1	1	0	0
1	0	0	1	0	0	1	1	1	0	0	1

1	0	0	1	0	0	1	0	1	1	1	0
1	0	0	1	0	1	1	0	1	1	1	0
1	0	0	1	0	1	1	1	1	0	0	1
1	0	0	1	0	1	0	1	1	1	0	0
1	0	0	1	0	1	0	0	1	0	0	1
1	0	0	1	1	1	0	0	1	0	0	1
1	0	0	1	1	1	0	1	1	1	0	0
1	0	0	1	1	1	1	1	1	0	0	1
1	0	0	1	1	1	1	0	1	1	1	0
1	0	0	1	1	0	1	0	0	1	1	0
1	0	0	1	1	0	1	1	0	0	0	1
1	0	0	1	1	0	0	1	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	1	1	1	1
1	0	0	0	0	0	1	1	1	0	0	0
1	0	0	0	0	0	1	0	1	1	0	1
1	0	0	0	0	1	1	0	1	1	0	1
1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	0	0	1	0	0	1	0	0	0
1	0	0	0	1	1	0	0	1	0	0	0
1	0	0	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	0	0
1	0	0	0	1	1	1	0	1	1	0	1
1	0	0	0	1	0	1	0	0	1	0	1
1	0	0	0	1	0	1	1	0	0	0	0
1	0	0	0	1	0	0	1	0	1	1	1
1	0	0	0	1	0	0	0	0	0	0	0

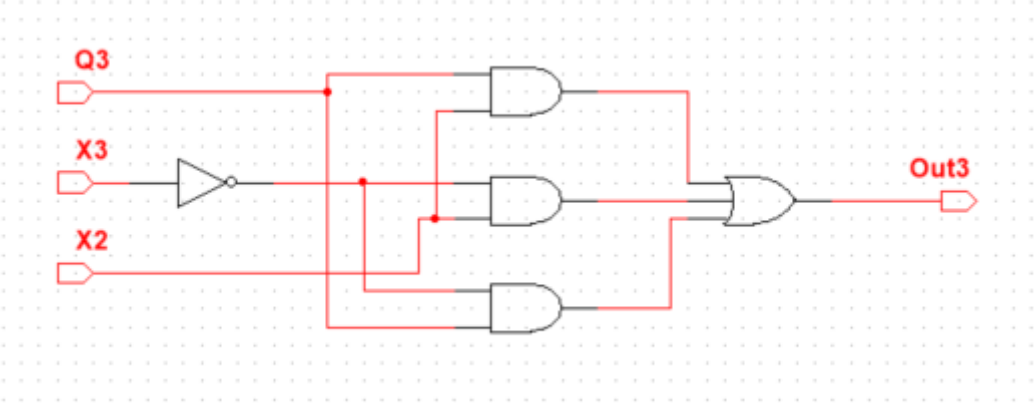
6 Tablice Karnaugh & Wyprowadzenia wzorów

W celu wyznaczenia uproszczonych wzorów pochodzących z tablic Karnaugh skorzystamy ze strony `https://www.charlie-coleman.com/experiments/kmap/`, która wykorzystuje metodę Petricka do minimalizacji funkcji logicznych.

6.a Stan odtwarzania (D3)

D3		Q_3,Q_2,Q_1,Q_0															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
X_3,X_2,X_1,X_0	0000	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0001	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0011	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0010	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0110	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0111	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0101	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0100	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1100	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	1101	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	1111	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	1110	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	1010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$D_3 = \overline{Q_3}X_3 + \overline{Q_3}Q_2 + Q_2X_3$$

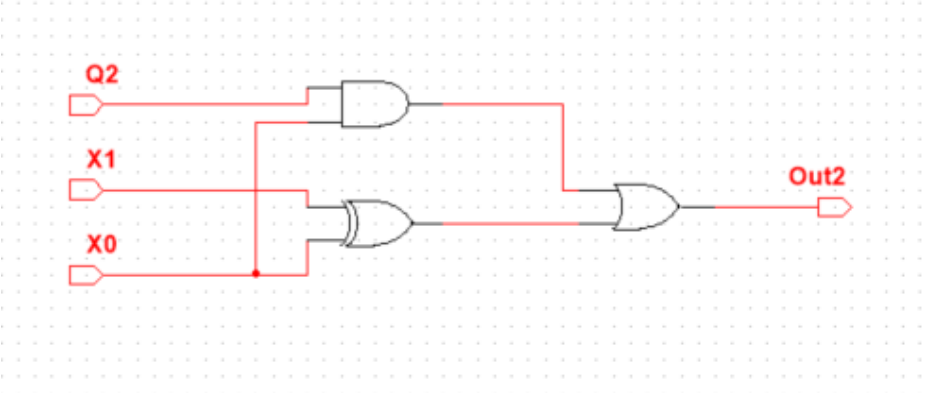


Rysunek 4: Podukład S3

6.b Bit blokujący (D2)

D2		Q_3,Q_2,Q_1,Q_0															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
X_3,X_2,X_1,X_0	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0001	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0011	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
	0010	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0110	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0111	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
	0101	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1101	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1111	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
	1110	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1010	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1011	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
	1001	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$D_2 = \overline{X_1}X_0 + X_1\overline{X_0} + X_0Q_2$$
$$= X_1 \oplus X_0 + X_0Q_2$$

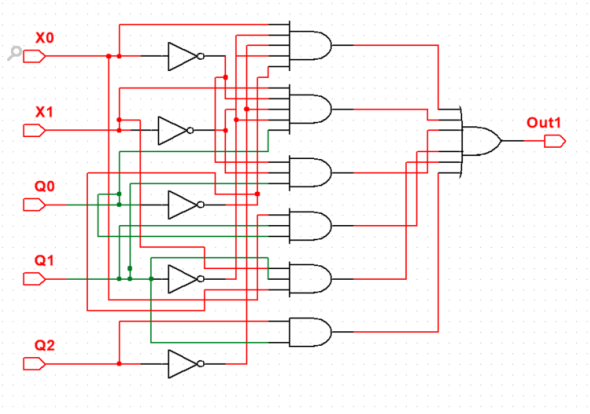


Rysunek 5: Podukład S2

6.c Starszy bit określający nr piosenkę (D1)

TUNE1		Q_3,Q_2,Q_1,Q_0															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
X_3,X_2,X_1,X_0	0000	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	0001	1	0	1	0	1	1	0	0	0	0	1	1	0	1	0	1
	0011	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	0010	0	1	0	1	1	1	0	0	0	0	1	1	1	0	1	0
	0110	0	1	0	1	1	1	0	0	0	0	1	1	1	0	1	0
	0111	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	0101	1	0	1	0	1	1	0	0	0	0	1	1	0	1	0	1
	0100	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	1100	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	1101	1	0	1	0	1	1	0	0	0	0	1	1	0	1	0	1
	1111	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	1110	0	1	0	1	1	1	0	0	0	0	1	1	1	0	1	0
	1010	0	1	0	1	1	1	0	0	0	0	1	1	1	0	1	0
	1011	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	1001	1	0	1	0	1	1	0	0	0	0	1	1	0	1	0	1
	1000	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0

$$D_1 = \overline{X_1}X_0\overline{Q_2Q_1},\overline{Q_0} + Q_2Q_1 + X_1\overline{X_0Q_2Q_1}Q_0 + \overline{X_1X_0}Q_1 + X_0Q_1Q_0 + X_1Q_1\overline{Q_0}$$

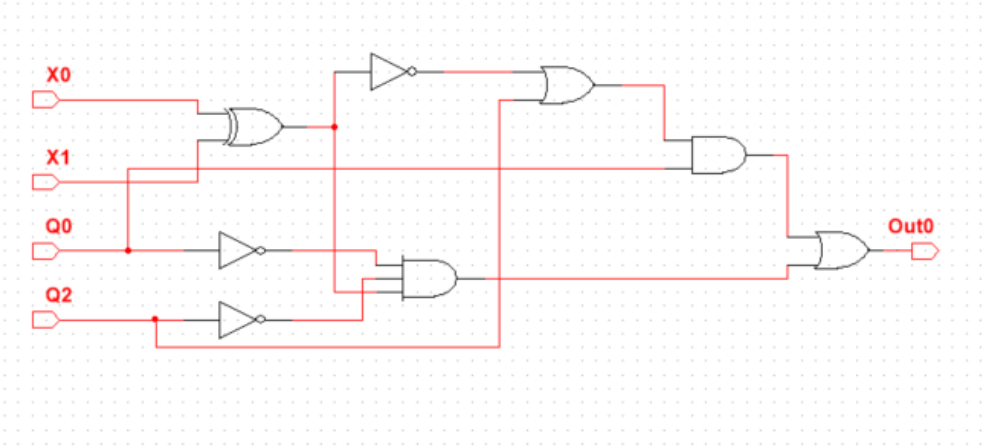


Rysunek 6: Podukład S1

6.d Młodszy bit określający nr piosenki (D0)

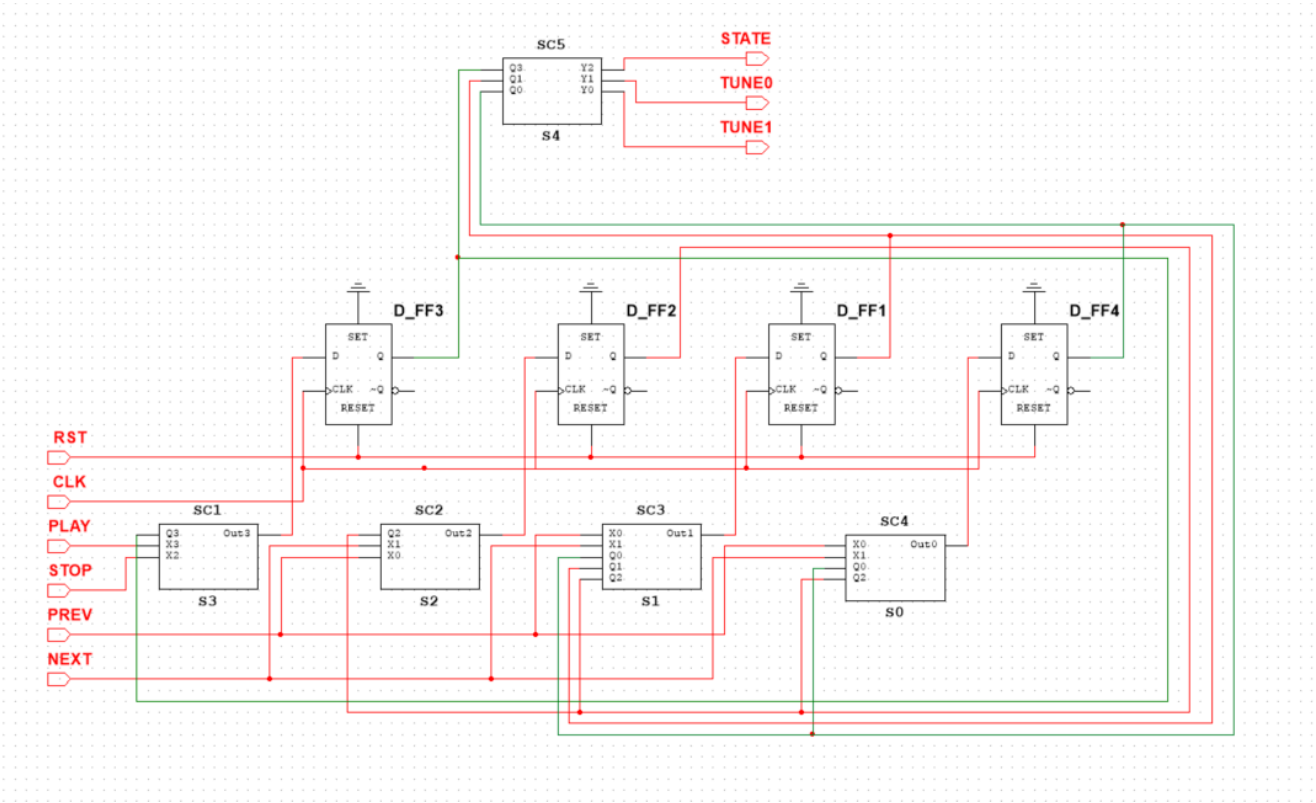
TUNE0		Q_3,Q_2,Q_1,Q_0															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
X_3,X_2,X_1,X_0	0000	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	0001	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	0011	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	0010	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	0110	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	0111	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	0101	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	0100	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1100	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1101	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	1111	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1110	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	1010	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	1011	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1001	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
	1000	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

$$
\begin{aligned}
D_0 &= \overline{X_1X_0}Q_0 + \overline{X_1}X_0\overline{Q_2Q_0} + Q_2Q_0 + X_1\overline{X_0}\,\overline{Q_2Q_0} + X_1X_0Q_0 \\
&= \overline{Q_2Q_0}\left(\overline{X_1}X_0 + X_1\overline{X_0}\right) + Q_0\left(\overline{X_1X_0} + X_1X_0 + Q_2\right) \\
&= \overline{Q_2Q_0}(X_0 \oplus X_1) + Q_0\left(\overline{X_1X_0} + \overline{X_0}X_0 + X_1\overline{X_1} + X_1X_0 + Q_2\right) \\
&= \overline{Q_2Q_0}(X_0 \oplus X_1) + Q_0\left(\left(X_1 + \overline{X_0}\right)\left(\overline{X_1} + X_0\right) + Q_2\right) \\
&= \overline{Q_2Q_0}(X_0 \oplus X_1) + Q_0\left(\overline{\overline{X_1X_0}} \cdot \overline{X_1\overline{X_0}} + Q_2\right) \\
&= \overline{Q_2Q_0}(X_0 \oplus X_1) + Q_0\left(\overline{\overline{X_1}X_0 + X_1\overline{X_0}} + Q_2\right) \\
&= \overline{Q_2Q_0}(X_0 \oplus X_1) + Q_0\left(\overline{X_0 \oplus X_1} + Q_2\right)
\end{aligned}
$$



Rysunek 7: Podukład S0

7 Implementacja układu w Multisimie

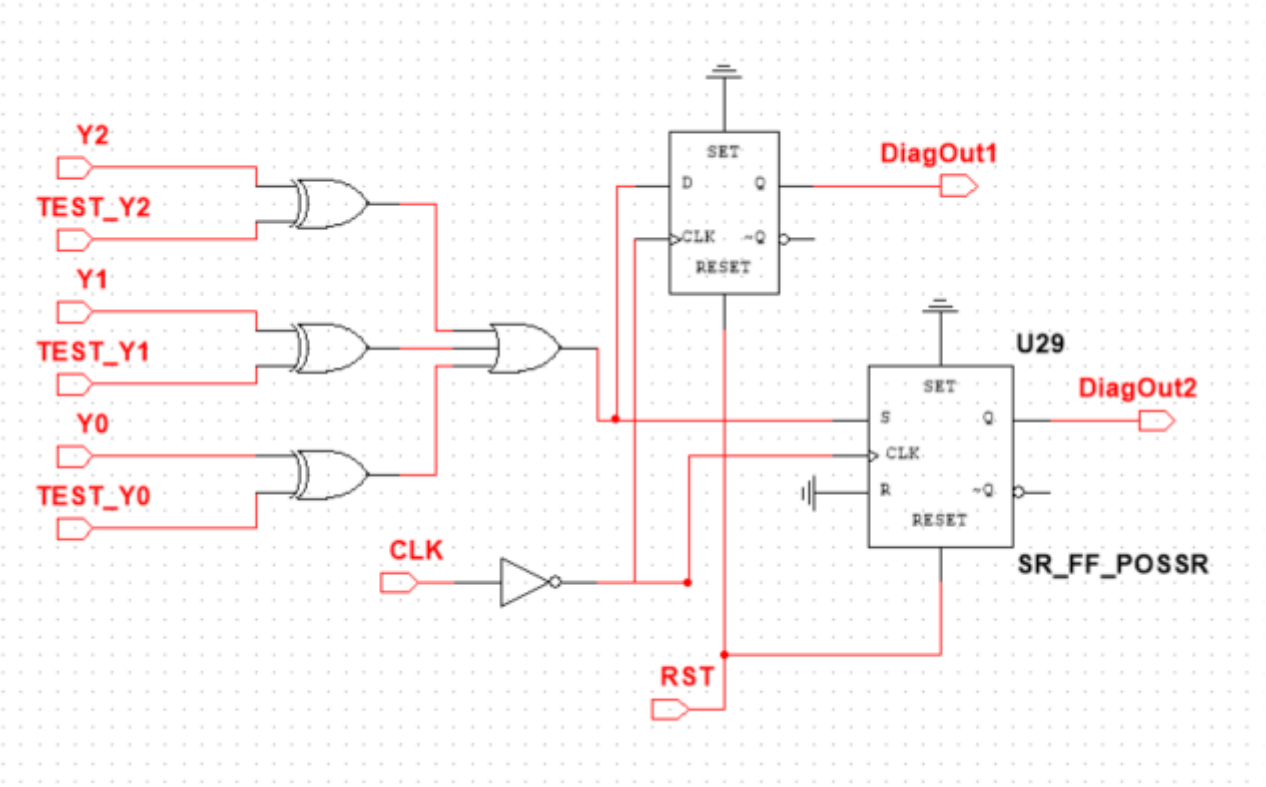


Rysunek 8: Cały układ

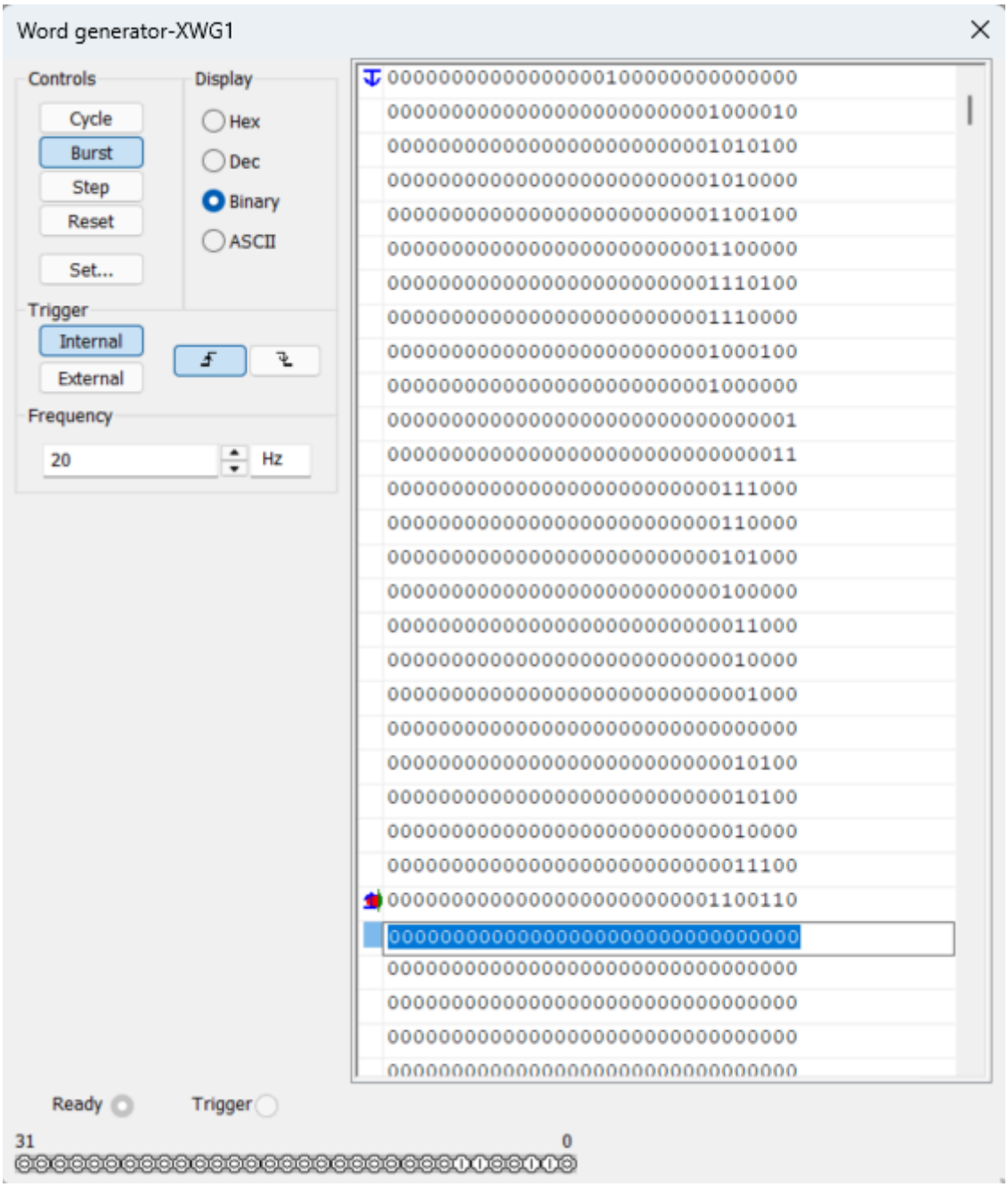
8 Testowanie układu

Testowanie układu sprowadza się do porównania sygnałów wyjściowych rzeczywistych do oczekiwanych. Testy, które zastosowaliśmy sprawdzają:

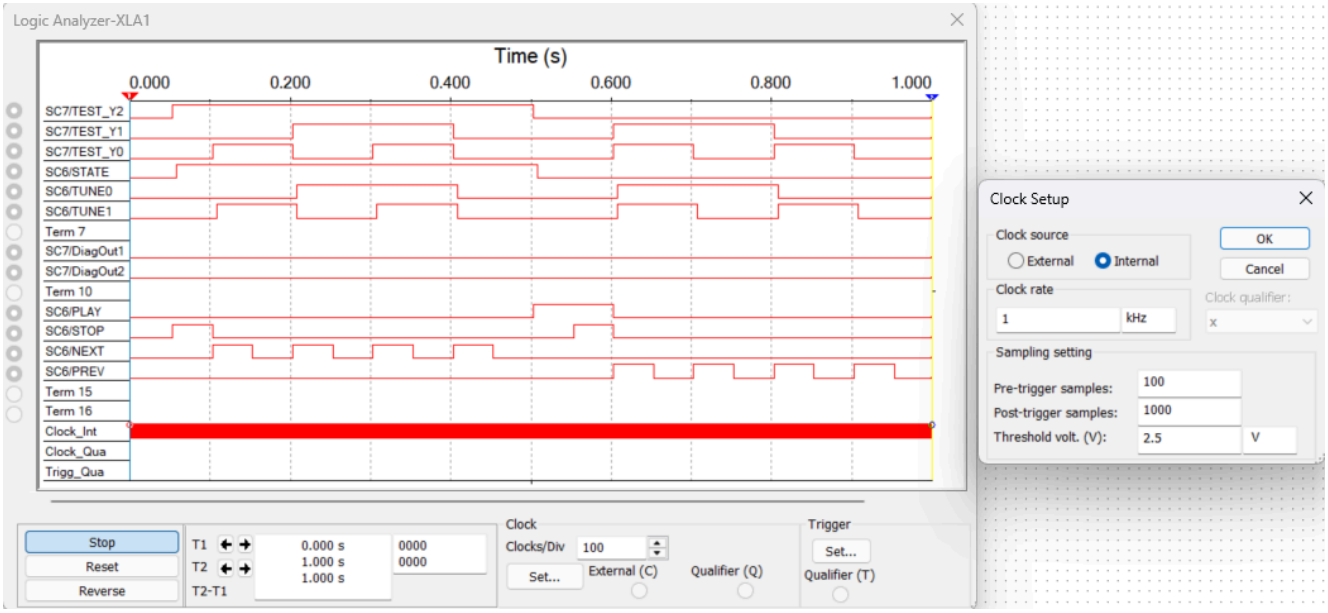
- czy istnieje cykl przy użyciu NEXT,
- czy istnieje cykl przy użyciu PREV,
- czy utwór można wstrzymać, wznowić,
- czy układ poprawnie się resetuje,
- czy poprawna jest reakcja na naciśnięcie na raz PLAY i STOP
- czy poprawna jest reakcja na naciśnięcie na raz PREV i NEXT
- czy bit blokujący działa jak powinien,



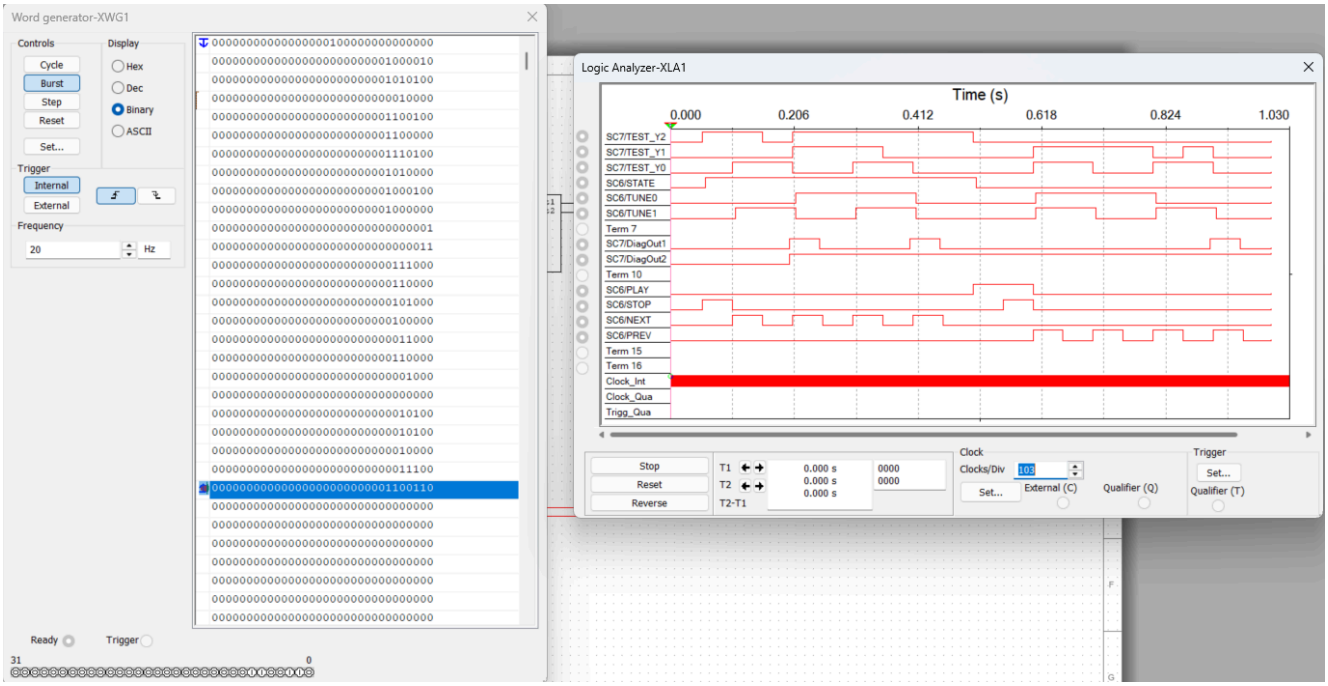
Rysunek 9: Układ testujący



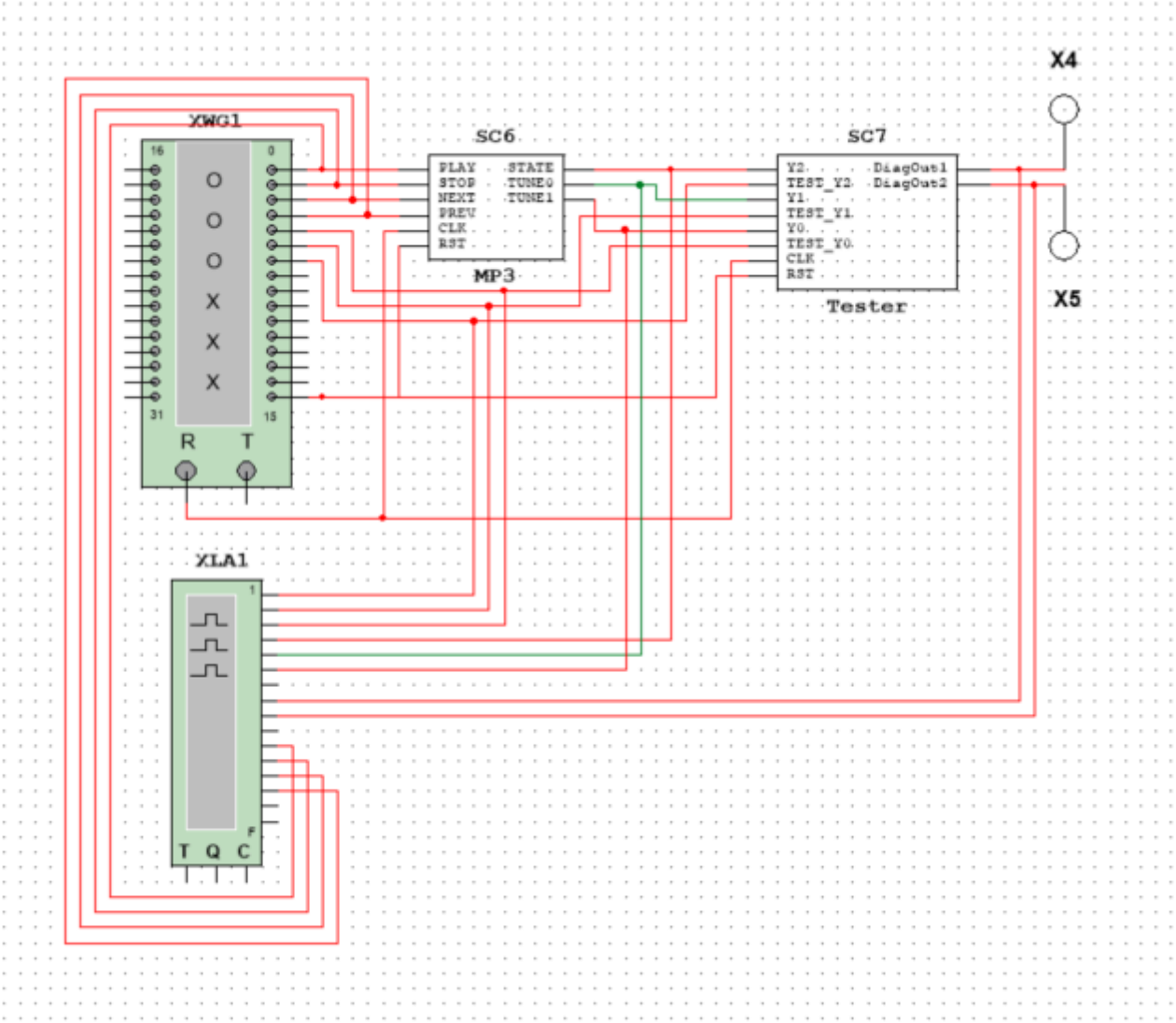
Rysunek 10: Konfiguracja generatora słów



Rysunek 11: Przedstawiający poprawność układu

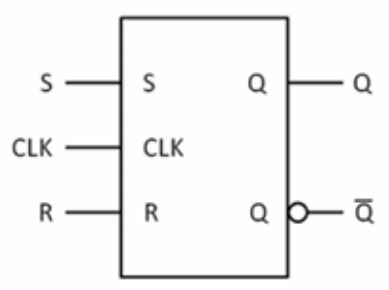


Rysunek 12: Przykład układu uszkodzonego



Rysunek 13: Implementacja całego układu wraz z testerem, generatorem słów i analizatorem logicznym

Sygnał błędu jeśli błąd wystąpił w jakimkolwiek teście rejestrowany jest przez przerzutnik synchroniczny RS (dodatkowe wejście CLK) i zapalana jest dioda X5.

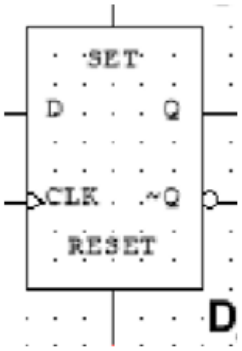


Rysunek 16: Przerzutnik RS

CLK	S	R	Q_{n+1}	$-Q_{n+1}$
↑	0	0	Q_n	$-Q_n$
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	X	X
X	X	X	Q_n	$-Q_n$

Tabela 6: Tabela prawdy przerzutnika RS

Sygnał błędu jeśli błąd wystąpił w danym teście rejestrowany jest przez przerzutnik synchroniczny D i zapalana jest dioda X4.



Rysunek 17: Przerzutnik D

CLK	D	SET	RESET	Q_{n+1}
↑	0	0	0	0
↑	1	0	0	1
X	X	0	0	Q_n
X	X	1	0	1
X	X	0	1	0
X	X	1	1	X

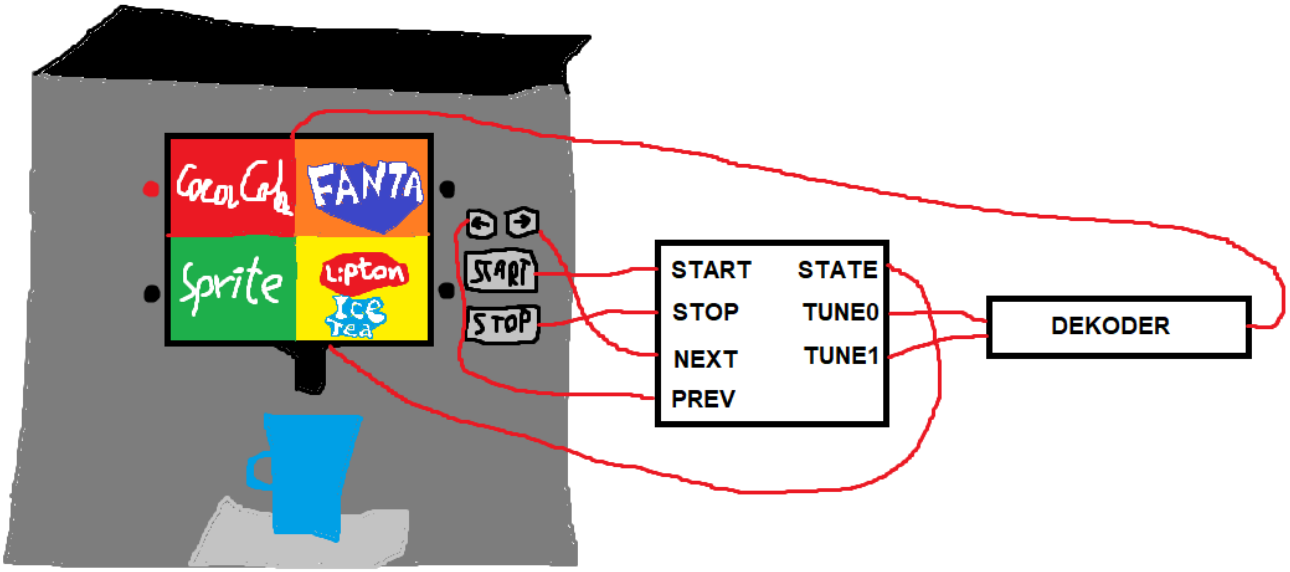
Tabela 7: Tabela prawdy dla przerzutnika D

9 Zastosowania

Podstawowym zastosowaniem zaprojektowanego przez nas urządzenia jest sterowanie odtwarzaczem muzycznym mp3. Przyciski STOP i PLAY odpowiadają za wstrzymywanie i wznawianie odtwarzania muzyki. Przy pomocy przycisków PREVIOUS i NEXT możemy zmieniać piosenki. W prezentowanym przez nas układzie, korzystając z 2 bitów wyjścia możemy wybierać odtwarzanie pomiędzy co najwyżej 4 piosenkami. W przypadku, gdybyśmy chcieli mieć możliwość wyboru między większą liczbą piosenek, należałoby dodać więcej przerzutników, które zapisywałyby numer piosenki i obliczyć nowe funkcje przejścia.

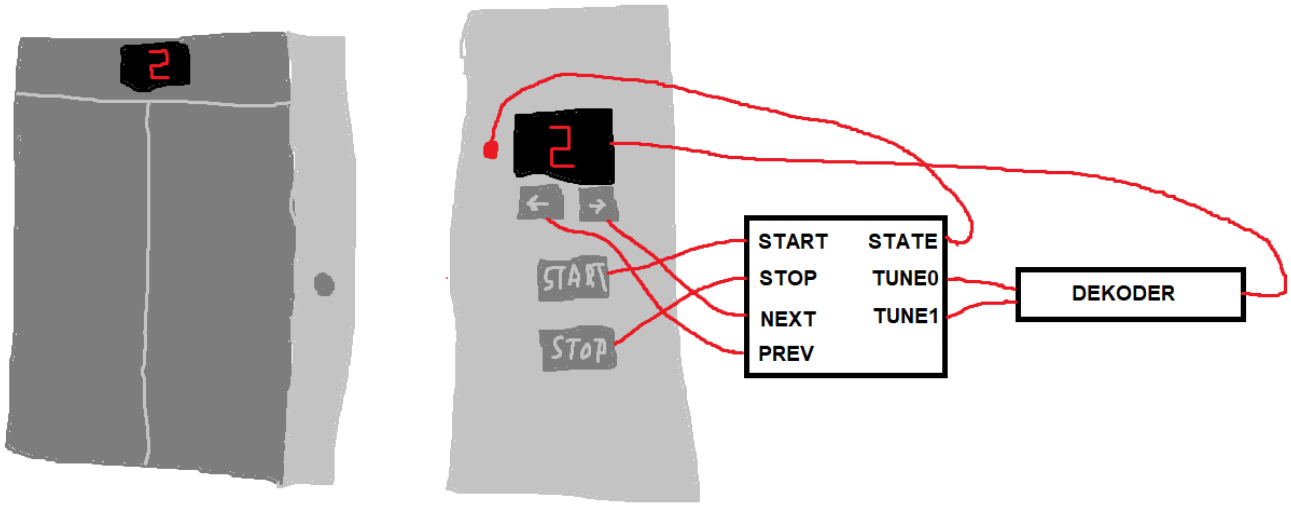
9.a Dozownik do napojów

Jednym z alternatywnych zastosowań układu, może być użycie go do obsługi dozownika do napojów. Przyciski START i STOP odpowiadają za wznawianie i wstrzymywanie polewania napoju. Diody obok typu napoju symbolizują aktualnie wybrany napój, który jest kodowany za pomocą 2 bitów i dekodowany za pomocą dekodera.



9.b Winda

Innym z alternatywnych zastosowań układu jest wykorzystanie go do obsługi przycisków w windzie. Stany repre


START
STOP
NEXT
PREV
STATE
TUNE0
TUNE1
DEKODER
START
STOP

10 Wnioski

Przerzutniki typu D wykorzystane w głównej części układu są zupełnie wystarczające, gdyż potrzebujemy zapamiętać tylko jeden stan.

Od pewnego rozmiaru tablic Karnaugh, zaznaczanie jak największych obszarów staje się bardzo problematyczne, warto skorzystać z narzędzi służących do szukania funkcji minimalizujących, które pozwalają zaoszczędzić dużo czasu.

Ważne spostrzeżenia, które zaobserwowaliśmy:
- funkcje przejścia dla przerzutników D1, D0 nie zależą od wejść PLAY, STOP
- funkcja przejścia dla przerzutnika D2 zależy jedynie od jego obecnego stanu i wejść NEXT, PREV
- funkcja przejścia dla przerzutnika D3 zależy jedynie od jego obencego stanu i wejść PLAY, STOP

10.a Co można zrobić inaczej?

Uważamy, że można by ewentualnie zmienić przejścia ze stanów, które powodują przejście do następnej lub poprzedniej piosenki, a mianowicie nie zapamiętywać czy piosenka, z której wyszliśmy była odtwarzana czy zatrzymana, tylko zawsze włączać odtwarzanie - tak jak to wygląda choćby na platformie YouTube czy Spotify.

Można by również wykorzystać innego typu przerzutniki zamiast przerzutników typu D, choćby przerzutniki typu JK (do wejścia J podać sygnał D, do wejścia K sygnał $\overline{D}$).

Do produkcji takich układów lepiej nie tworzyć podukładów S0, S1, S2 oraz S3 i w ten sposób na pewno nie pogorszyć, a jedynie dać szansę na zmniejszenie kosztu produkcji, poprzez nie powielanie tych samych fragmentów implementacji funkcji logicznych na bramkach logicznych.