

Technika cyfrowa

Sprawozdanie z ćwiczenia nr 3

Rozmus Dariusz, Fabia Jakub, Smółka Tomasz, Czyż Bartosz, Zając Kacper

1. TREŚĆ ZADANIA

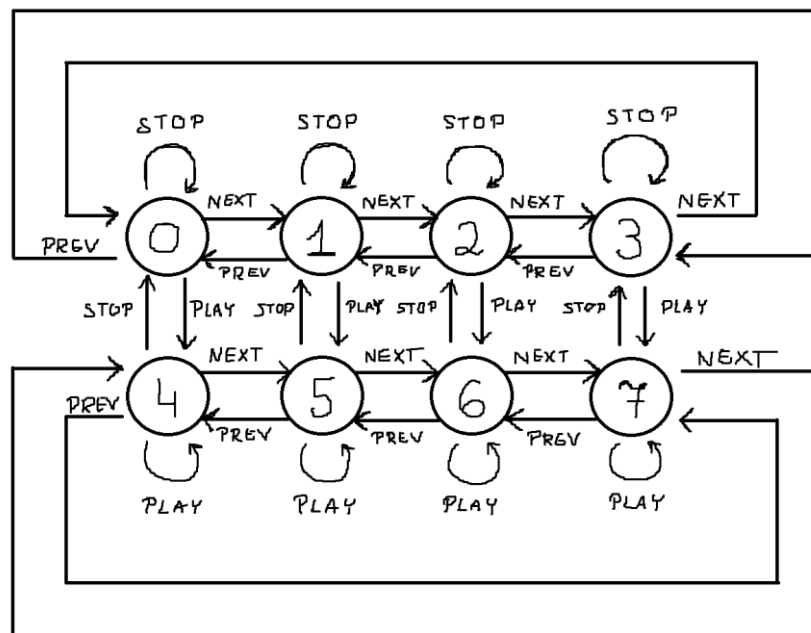
Zaprojektować automat mogący posłużyć do sterowania jakimś prostym odtwarzaczem plików muzycznych mp3.

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. IDEA ROZWIĄZANIA



Rysunek 1: Schemat automatu w postaci grafu

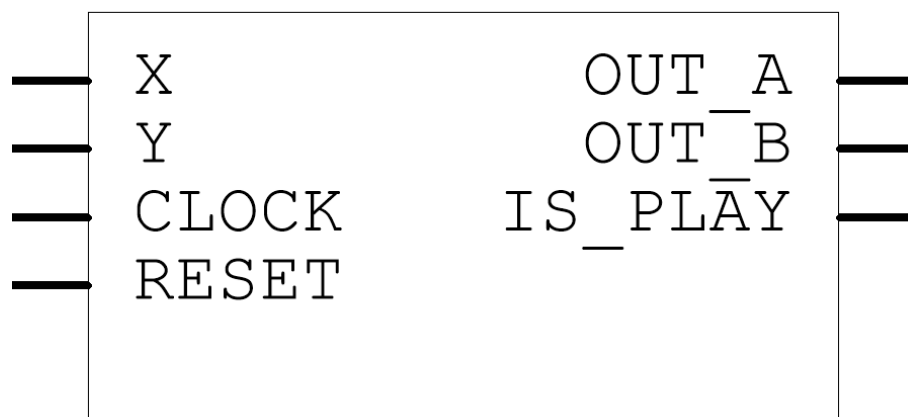
Układ ten będzie automatem nieskończonym, zupełnym. Wyjścia zależą będą tylko od stanu automatu, a zatem będzie to także automat Moore'a.

Każdy utwór w danej chwili może być włączony (PLAY) albo wyłączony (STOP), zatem w sumie nasz automat będzie miał 8 różnych stanów.

Układ będzie miał cztery różne wejścia – PLAY, STOP, PREV, NEXT. Można zatem postąpić się dwubitową liczbą binarną do ich reprezentacji:

- PLAY – 01
- STOP – 00
- PREV – 10
- NEXT – 11

SC1



MAIN

Rysunek 2: Schemat wejść i wyjść dla podukładu głównego

Wejścia podukładu:

- *X, Y* – bity dwucyfrowej liczby binarnej reprezentującej wciśnięty przycisk PLAY, STOP, NEXT albo PREV.
- *CLOCK* – zegar
- *RESET* – reset (na potrzeby podukładu testującego)

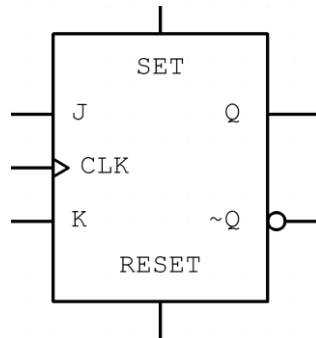
Wyjścia podukładu:

- *OUT_A, OUT_B* – bity dwucyfrowej liczby binarnej reprezentującej dany utwór
- *IS_PLAY* – bit oznaczający czy dany utwór się wykonuje ('1' – utwór gra, '0' – utwór nie gra)

3. PROJEKTOWANIE UKŁADU

3.1. Podukład główny - automat

Do realizacji wszystkich stanów wykorzystano trzy przerzutniki typu JK ($2^3 = 8$ stanów):



Rysunek 3: Przerzutnik typu JK - Flip Flop

Do sporządzenia tabelki Karnaugh wykorzystano wiedzę o przejściach między stanami automatu i tablicy prawdościowej przerzutnika JK:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Tabela 1: Tablica prawdościowa dla przerzutników typu JK

J	K	Q_n	Q_{n+1}
0	-	0	0
1	-	0	1
-	0	1	1
-	1	1	0

Tabela 2: Inna wersja tablica prawdościowej dla przerzutników typu JK

	WEJŚCIA			
STAN	STOP 00	PLAY 01	NEXT 10	PREV 11
0	0	4	1	3
1	1	5	2	0
2	2	6	3	1
3	3	7	0	2
4	0	4	5	7
5	1	5	6	4
6	2	6	7	5
7	3	7	4	6

Tabela 3: Tabela wejść automatu

STAN	WYJŚCIE (OUT_A, OUT_B, IS_PLAY)
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Tabela 4: Tabela wyjść automatu

XY C B A				J_A				K_A				J_B				K_B				J_C				K_C			
				00 STOP	01 PLAY	11 NEXT	10 PREV	00 STOP	01 PLAY	11 NEXT	10 PREV	00 STOP	01 PLAY	11 NEXT	10 PREV	00 STOP	01 PLAY	11 NEXT	10 PREV	00 STOP	01 PLAY	11 NEXT	10 PREV	00 STOP	01 PLAY	11 NEXT	10 PREV
(0)	0	0	0	0	0	1	1	-	-	-	-	0	0	0	1	-	-	-	-	0	1	0	0	-	-	-	-
(1)	0	0	1	-	-	-	-	0	0	1	1	0	0	1	0	-	-	-	-	0	1	0	0	-	-	-	-
(3)	0	1	1	-	-	-	-	0	0	1	1	-	-	-	-	0	0	1	0	0	1	0	0	-	-	-	-
(2)	0	1	0	0	-	1	1	-	1	-	-	-	-	-	-	0	0	0	1	0	1	0	0	-	-	-	-
(6)	1	1	0	0	0	1	1	-	-	-	-	-	-	-	-	0	0	0	1	-	-	-	-	1	0	0	0
(7)	1	1	1	-	-	-	-	0	0	1	1	-	-	-	-	0	0	1	0	-	-	-	-	1	0	0	0
(5)	1	0	1	-	-	-	-	0	0	1	1	0	0	1	0	-	-	-	-	-	-	-	-	1	0	0	0
(4)	1	0	0	0	0	1	1	-	-	-	-	0	0	0	1	-	-	-	-	-	-	-	-	1	0	0	0

Tabela 5: Sumaryczna tablica Karnaugh dla automatu

Gdzie:

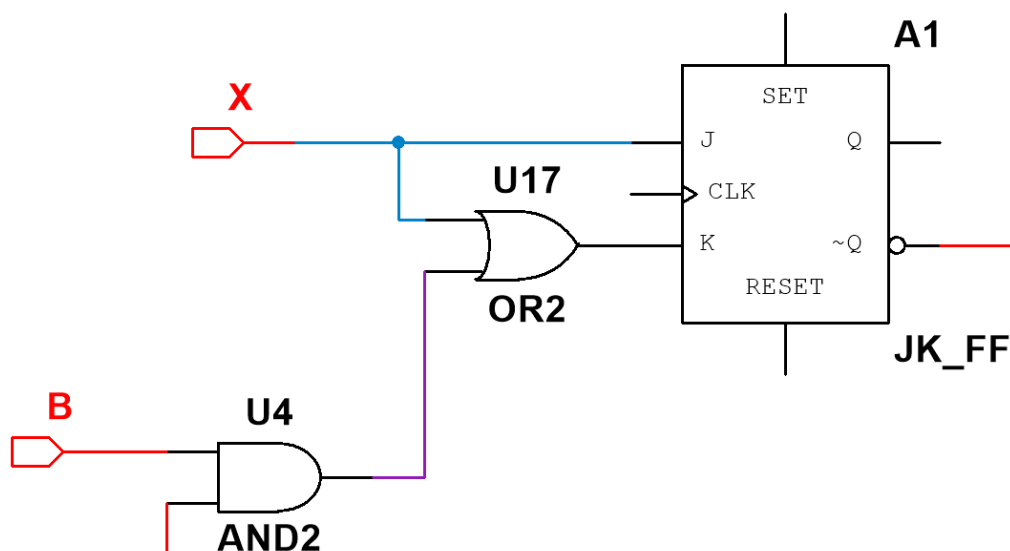
- A – odpowiada wyjściu OUT_A
- B – odpowiada wyjściu OUT_B
- C – odpowiada wyjściu IS_PLAY

XY				J_A				K_A			
				00	01	11	10	00	01	11	10
C	B	A		STOP	PLAY	NEXT	PREV	STOP	PLAY	NEXT	PREV
(0)	0	0	0	0	0	1	1	-	-	-	-
(1)	0	0	1	-	-	-	-	0	0	1	1
(3)	0	1	1	-	-	-	-	0	0	1	1
(2)	0	1	0	0	-	1	1	-	1	-	-
(6)	1	1	0	0	0	1	1	-	-	-	-
(7)	1	1	1	-	-	-	-	0	0	1	1
(5)	1	0	1	-	-	-	-	0	0	1	1
(4)	1	0	0	0	0	1	1	-	-	-	-

Tabela 6: Tablica Karnaugh dla pierwszego przerzutnika $J_A K_A$

$$J_A = X$$

$$K_A = X + B\bar{A}$$



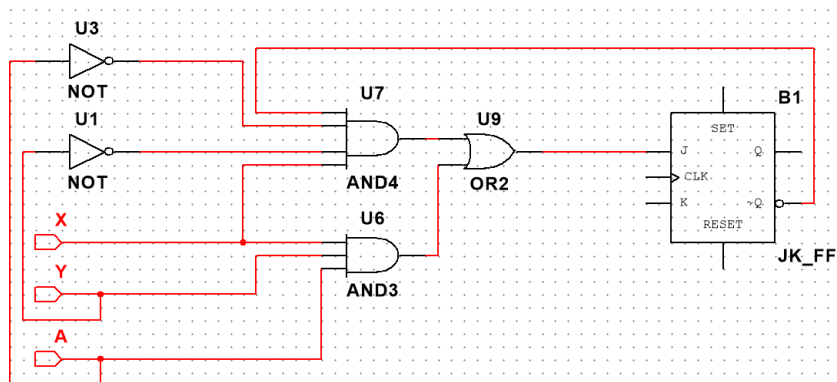
Rysunek 4: Układ wejść J, K przerzutnika A

XY			J_B				K_B			
			00	01	11	10	00	01	11	10
C	B	A	STOP	PLAY	NEXT	PREV	STOP	PLAY	NEXT	PREV
(0)	0	0	0	0	0	1	-	-	-	-
(1)	0	0	1	0	0	1	0	-	-	-
(3)	0	1	1	-	-	-	0	0	1	0
(2)	0	1	0	-	-	-	0	0	0	1
(6)	1	1	0	-	-	-	0	0	0	1
(7)	1	1	1	-	-	-	0	0	1	0
(5)	1	0	1	0	0	1	0	-	-	-
(4)	1	0	0	0	0	0	1	-	-	-

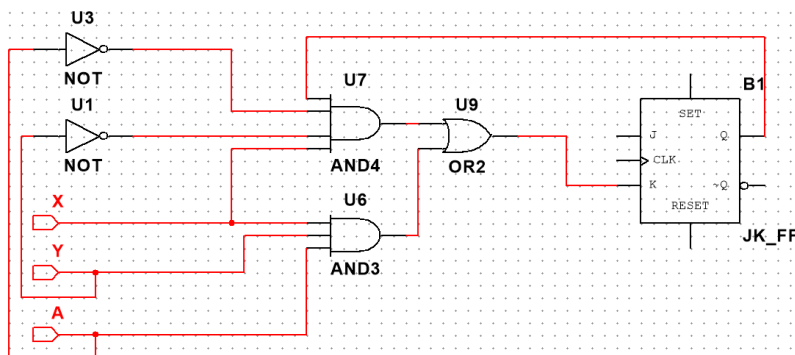
Tabela 7: Tablica Karnaugh dla drugiego przerzutnika $J_B K_B$

$$J_B = XYA + \bar{B}\bar{A}X\bar{Y}$$

$$K_B = XYA + B\bar{A}X\bar{Y}$$



Rysunek 5: Układ wejścia J przerzutnika B



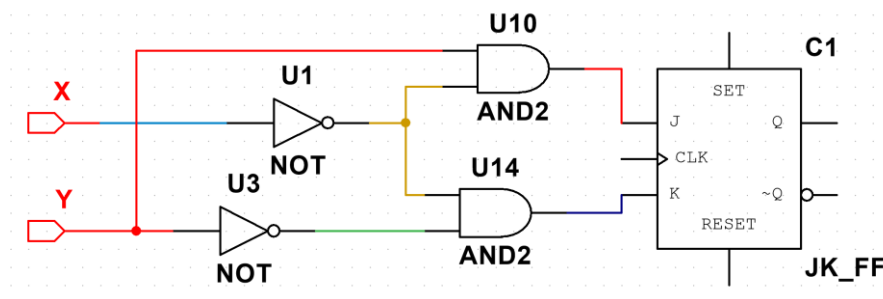
Rysunek 6: Układ wejścia K przerzutnika B

XY C B A				J_c				K_c			
				00 STOP	01 PLAY	11 NEXT	10 PREV	00 STOP	01 PLAY	11 NEXT	10 PREV
(0)	0	0	0	0	1	0	0	-	-	-	-
(1)	0	0	1	0	1	0	0	-	-	-	-
(3)	0	1	1	0	1	0	0	-	-	-	-
(2)	0	1	0	0	1	0	0	-	-	-	-
(6)	1	1	0	-	-	-	-	1	0	0	0
(7)	1	1	1	-	-	-	-	1	0	0	0
(5)	1	0	1	-	-	-	-	1	0	0	0
(4)	1	0	0	-	-	-	-	1	0	0	0

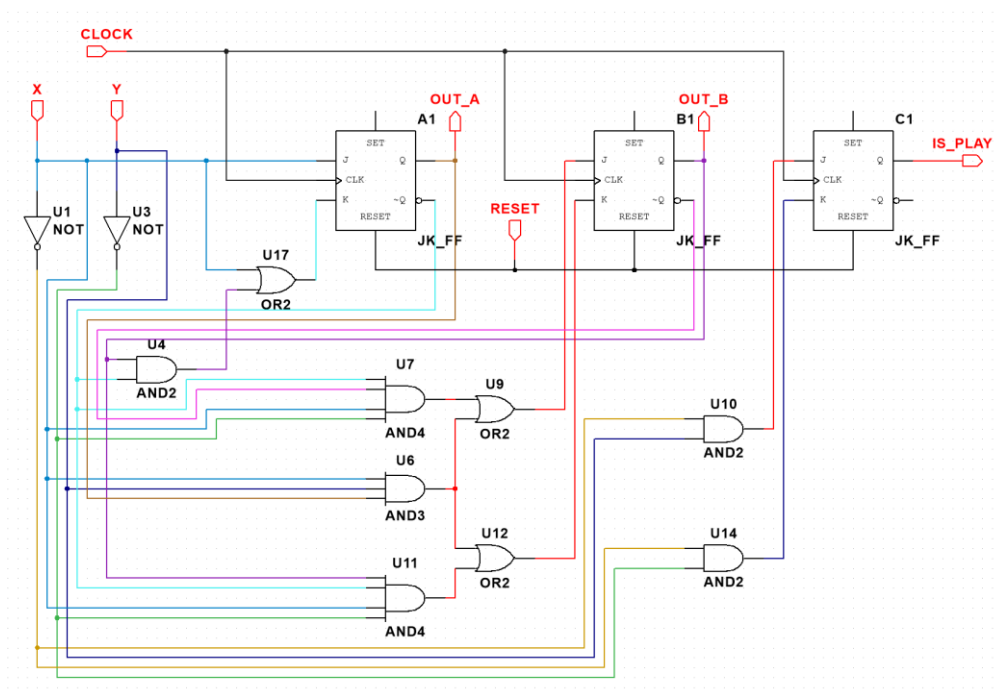
Tabela 8: Tablica Karnaugh dla trzeciego przerzutnika $J_c K_c$

$$J_c = \bar{X} Y$$

$$K_c = \bar{X} \bar{Y}$$

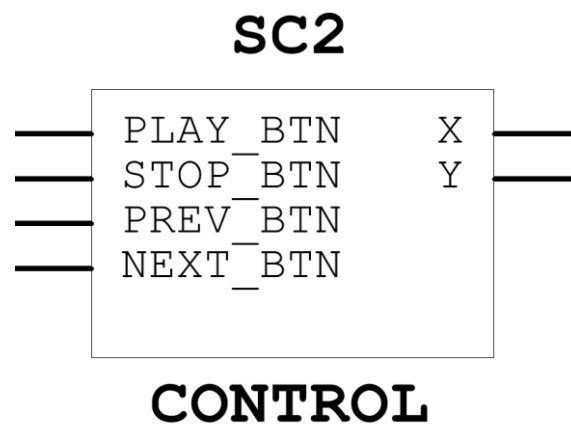


Rysunek 7: Układ wejść J, K przerzutnika C



Rysunek 8: Implementacja produktu głównego w programie Multisim

3.2. Podukład dekodujący i spamiętujący wciśnięty przycisk



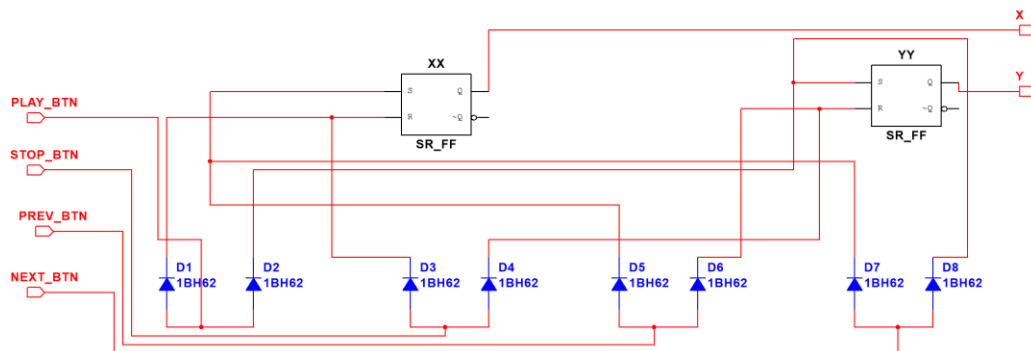
Rysunek 9: Schemat wejść i wyjść dla podukładu dekodującego wciśnięty przycisk

Wejścia podukładu:

- *PLAY_BTN* – przycisk PLAY służący do włączania danego utworu
- *STOP_BTN* – przycisk STOP służący do wyłączania danego utworu
- *PREV_BTN* – przycisk PREV służący do przechodzenia po poprzednich utworach
- *NEXT_BTN* – przycisk NEXT służący do przechodzenia po następnych utworach

Wyjścia podukładu:

- *X, Y* – cyfry otrzymanej liczby binarnej



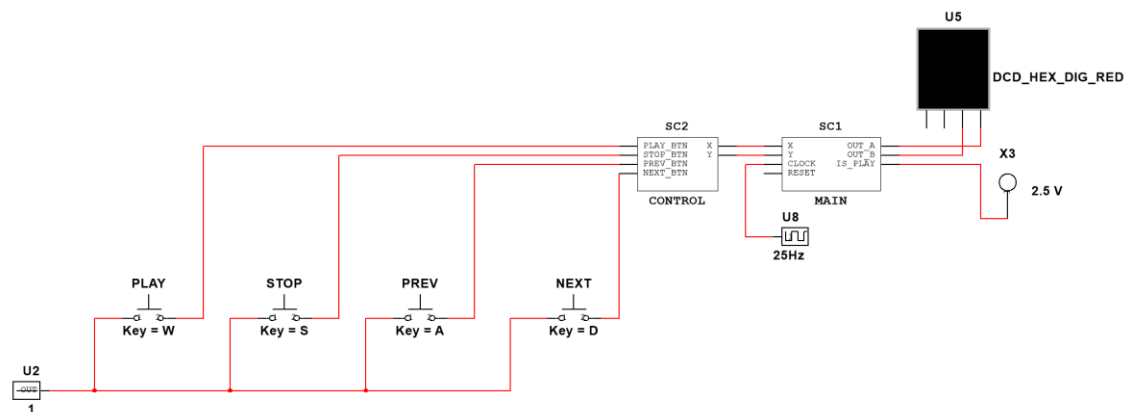
Rysunek 10: Schemat podukładu dekodującego w programie Multisim

Do implementacji tego podukładu zastosowano dwa przerzutniki typu SR – wyjścia *Q* każdego z nich reprezentują odpowiedni bit dwucyfrowej liczby binarnej. Przyjęto:

- *PLAY* – 01 (RESET na pierwszym przerzutniku i SET na drugim)
- *STOP* – 00 (RESET na pierwszym i drugim)
- *PREV* – 10 (SET na pierwszym przerzutniku i RESET na drugim)
- *NEXT* – 11 (SET na pierwszym i drugim)

W podukładzie wykorzystano również diody półprzewodnikowe.

3.3. Gotowy układ cyfrowy



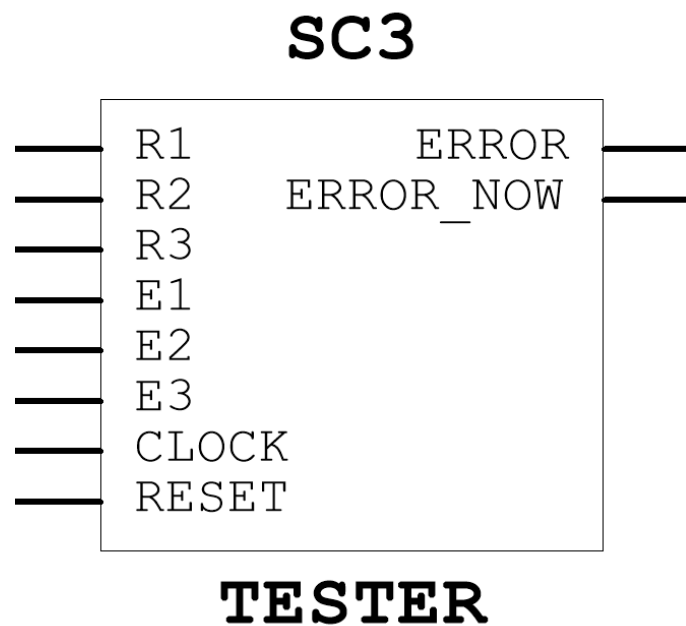
Rysunek 11: Gotowy układ cyfrowy

Numer utworu pokazywany jest na wyświetlaczu. Lampka obok wyświetlacza określa czy utwór jest włączony, czy wyłączony. Wejścia PLAY, STOP, PREV, NEXT zrealizowano przy pomocy kluczy.

4. PODUKŁAD TESTUJĄCY

4.1. Schemat podukładu testującego

Do implementacji podukładu testującego wykorzystano generator słów i analizator stanów logicznych.



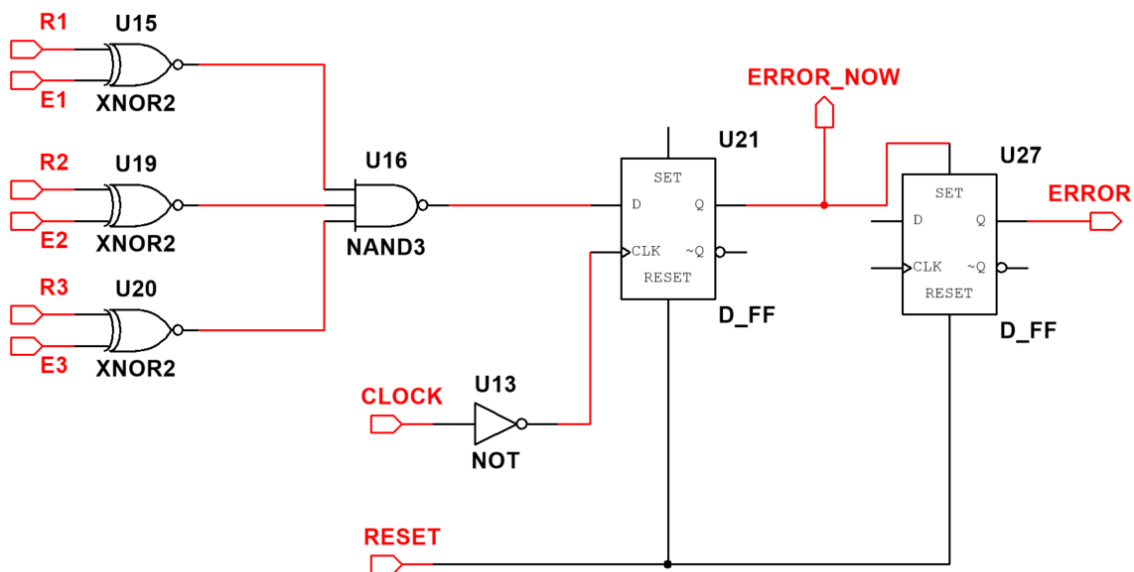
Rysunek 8: Schemat podukładu testującego

Wejścia podukładu:

- *R1, R2, R3* – sygnały generowane przez podukład główny
- *E1, E2, E3* – sygnały generowane przez generator słów
- *CLOCK* – zegar z generatora słów
- *RESET* – reset generowany na początku każdego cyklu (z generatora słów)

Wyjścia podukładu:

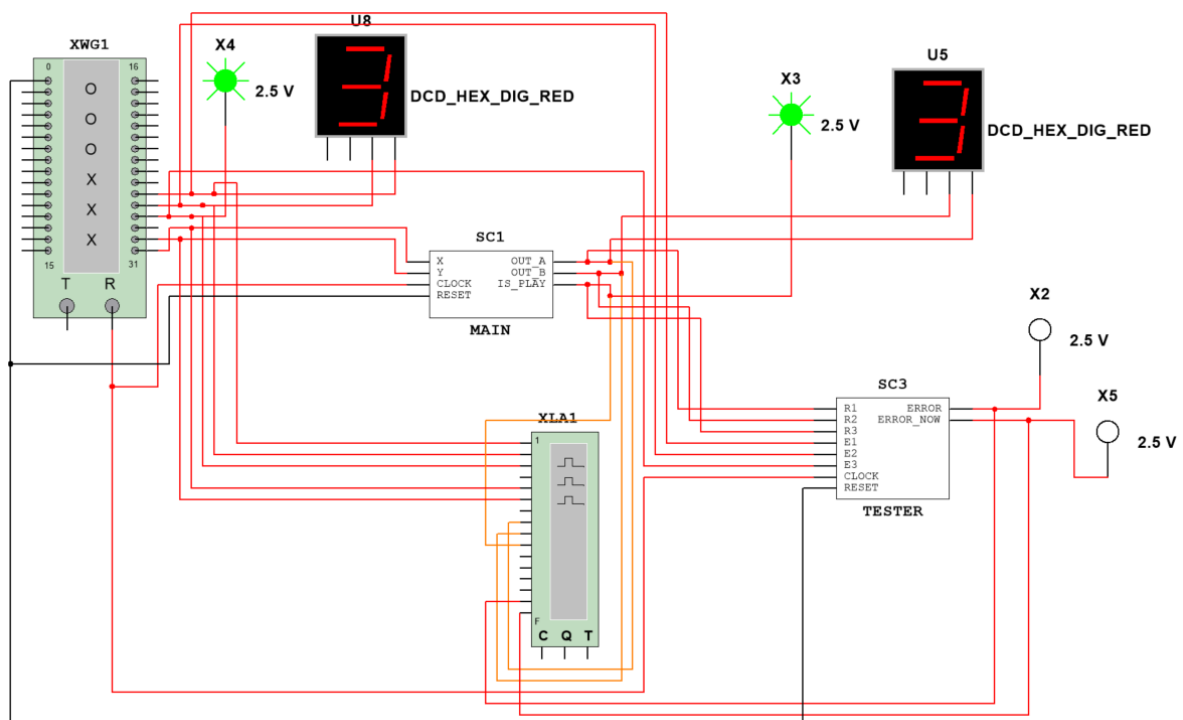
- *ERROR* – sygnał błędu
- *ERROR_NOW* – asynchroniczny sygnał ustalający moment w którym wystąpił błąd



Rysunek 9: Schemat podukładu testującego zaimplementowany w programie Multisim

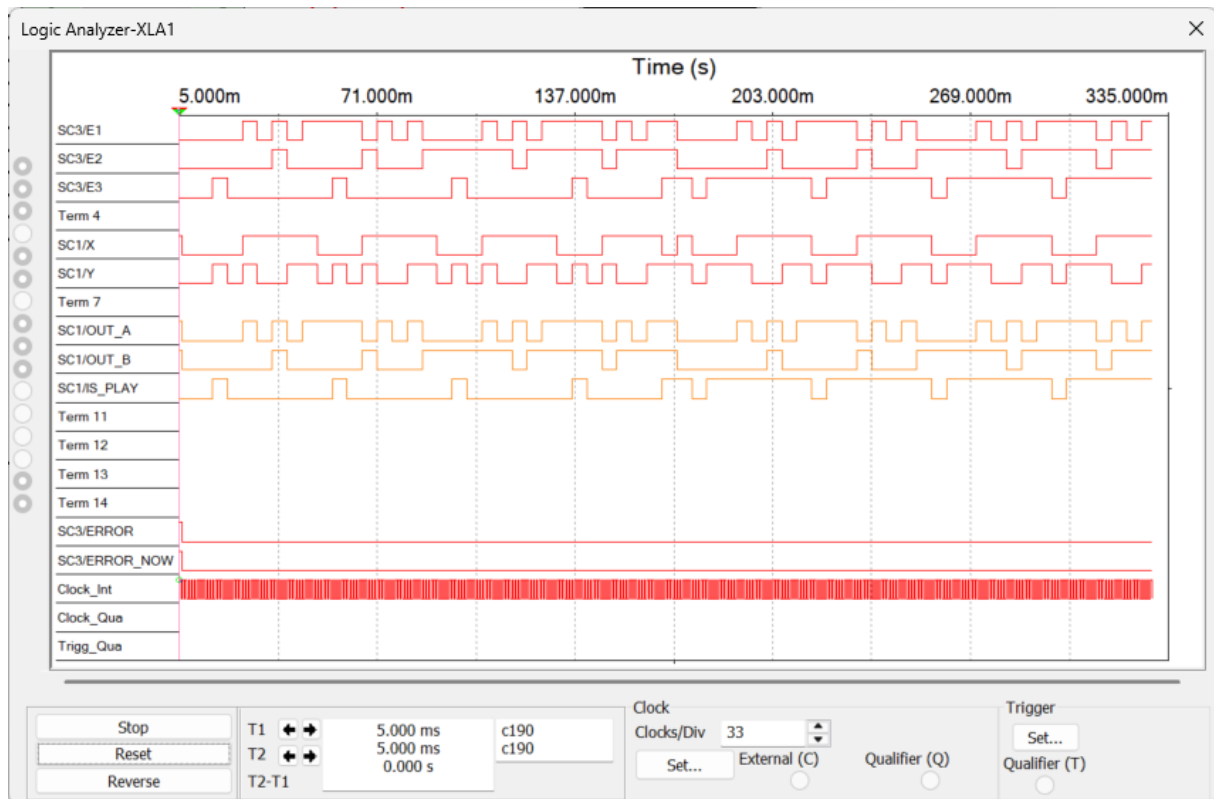
Wykorzystano dwa przerzutniki typu D, aby zapamiętać sygnał błędu do czasu zakończenia cyklu. Dodatkowo zastosowano bramkę NOT opóźniającą zegar o pół taktu, ponieważ sygnały z generatora słów i układu głównego nie są ze sobą zsynchronizowane przez sygnał reset.

4.2. Podpięcie układu testującego do zaprojektowanego układu cyfrowego



Rysunek 10: Układ cyfrowy z wpiętym podukładem testującym

4.4. Analizator logiczny



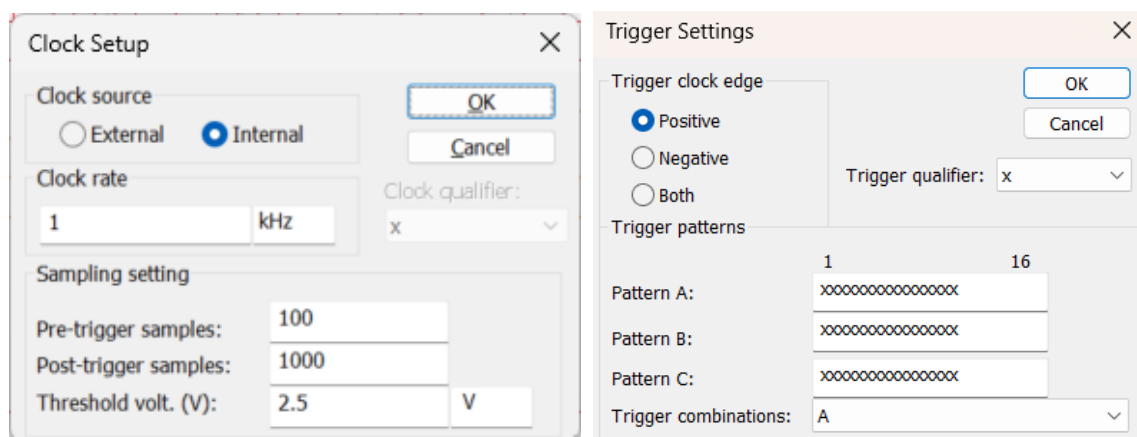
Rysunek 12: Pokazania analizatora logicznego dla działającego układu

Pierwsze 3 sygnały wyjścia układu głównego: OUT_A, OUT_B, IS_PLAY.

Kolejne 2 to symulowane przyciski (wejścia głównego) X, Y.

Kolejne 3 to wyjścia generatora słów: OUT_A, OUT_B, IS_PLAY.

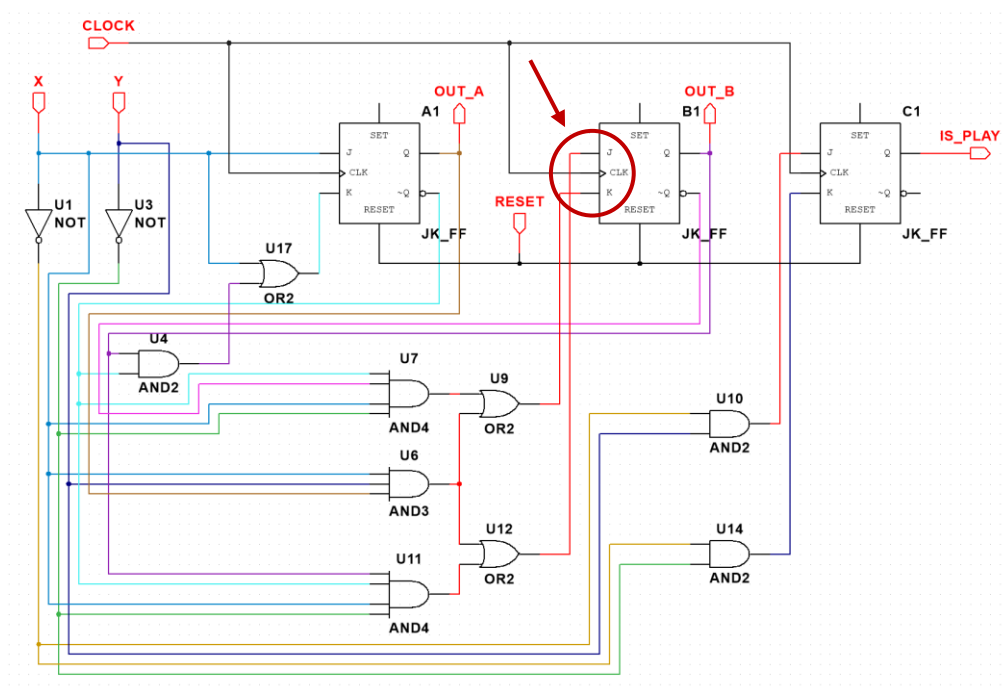
Ostatnie 3 to kolejno: ERROR, ERROR_NOW i CLOCK.



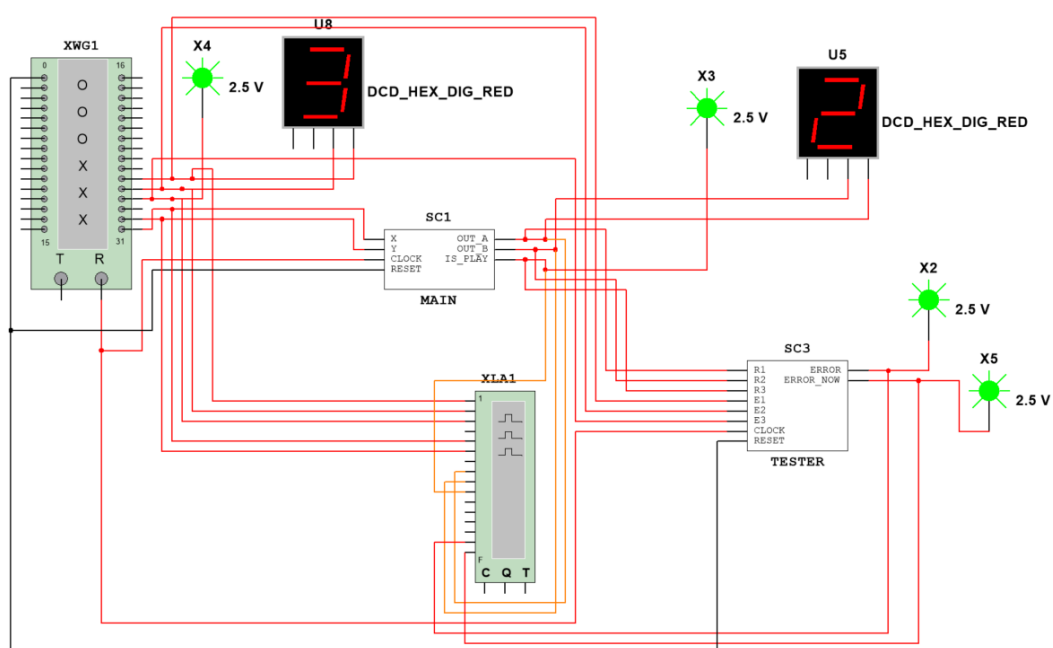
Rysunek 13: Ustawienia analizatora logicznego

4.5. Przeprowadzenie testu

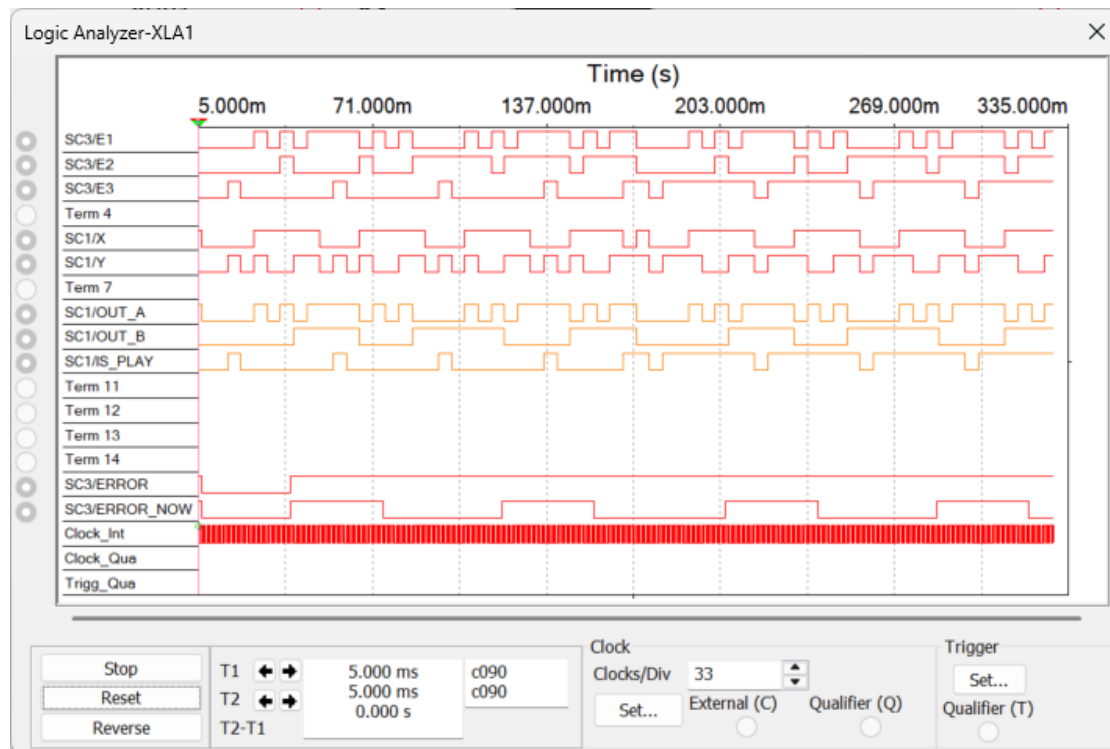
W poniższym przypadku w podukładzie głównym dla jednego przerzutnika podpięto na odwrót wejścia J i K przerzutnika B:



Rysunek 14: Błędny podukład główny



Rysunek 15: Działanie podukładu testującego na sфорsowanym układzie. Widać, że lampki podukładu testującego świecą się, a więc błąd został wykryty

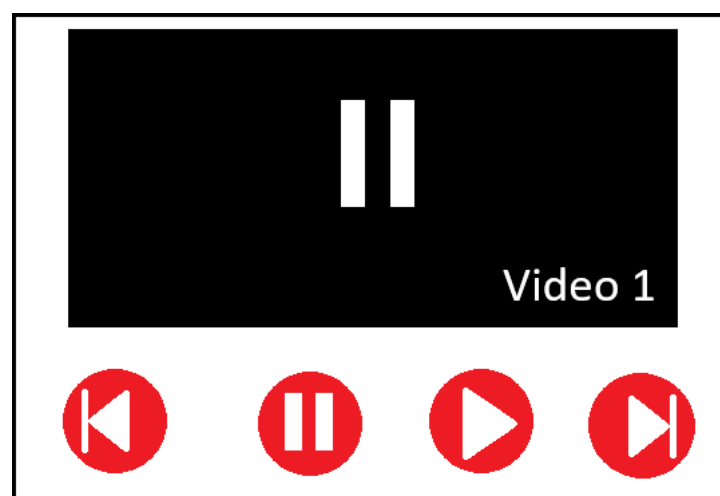


Rysunek 16: Pokazania błędu w analizatorze logicznym

5. WNIOSKI I ZASTOSOWANIA

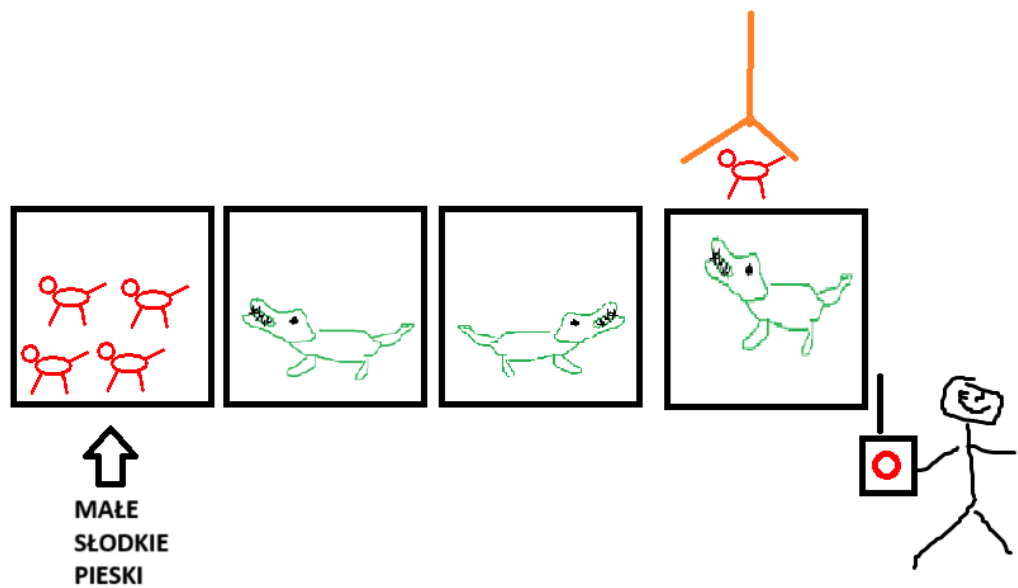
5.1 Zastosowania

- Taki automat mógłby posłużyć do obsługi odtwarzacza muzyki (z tylko 4 piosenkami).
- Można go też wykorzystać jako metoda zmiany plików wideo na interaktywnym kiosku np. w muzeum.



Rysunek 17: Wykorzystanie układu w interaktywnym kiosku

- UWAGA DRASTYCZNE: Oraz do karmienia dzikich zwierząt w klatkach!



Rysunek 18: Wykorzystanie układu do karmienia

5.2 Co można było zrobić lepiej / inaczej?

- Można było stworzyć dodatkową logikę, która sprawi, że po wciśnięciu przycisku NEXT lub PREV wciskany jest przycisk PLAY. W tym momencie po wciśnięciu jednego z tych przycisków, utwory są przewijane do momentu wciśnięcia PLAY lub STOP.