

**AKADEMIA GÓRNICZO-HUTNICZA**

Dokumentacja do projektu

**Gra - Jednoręki bandyta**

z przedmiotu

**Projektowanie Systemów Cyfrowych**

Elektronika III rok

*Dominik Górny*

prowadzący: dr inż. Paweł Rajda

24.01.2022r

1. **Temat**

Tematem projektu jest gra - “jednoręki bandyta”, w której celem jest wylosowanie trzech takich samych znaków/liczb.

1. **Założenia projektowe**

Trzy obracające się z różną prędkością bębny są symulowane przez trzy liczniki. Dla uzyskania efektu losowości oraz widocznego losowania każdy z nich jest sterowany własnym preskalerem, dzielącym częstotliwość zegara (100 MHz) przez kolejne liczby(5000011,5200000,5740000). Naciśnięcie przycisku losowania uruchamia preskalery oraz liczniki, a puszczenie go zatrzymuje losowanie, pozwalając na odczytanie wyniku. Wylosowane wartości są prezentowane na 7SEG.

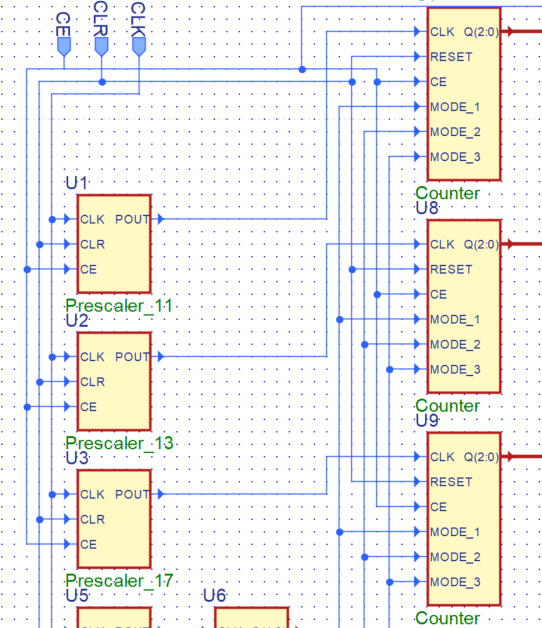
1. **Opis funkcjonalny**

Początkowo po włączeniu programu wyświetlają się trzy zera na wyświetlaczach siedmiosegmentowych.W celu uruchomienia losowania należy nacisnąć przycisk *BTNC(N17)* i należy go trzymać do momentu, w którym chcemy zakończyć losowanie. Po wylosowaniu odpowiednio 3 lub 2 takich samych wartości włącza się dioda czerwona(dla 3) lub zielona(dla 2). Po naciśnięciu przycisku *BTND(P18)* następuje reset układu oznaczający powrót do początkowej konfiguracji. Przełączenie *switcha (J15)* powoduje zmianę wyświetlanych cyfr na odpowiednio stworzone wcześniej znaki. Przełączanie kolejnych switchów powoduje zmianę prawdopodobieństwa wygranej.

1. **Schemat blokowy**

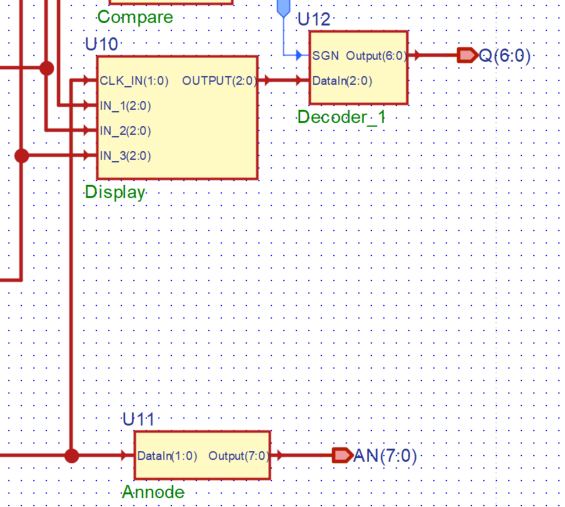
Projekt można podzielić na 3 segmenty:

1. Część odpowiedzialna za obsługę 3 liczników, które są obsługiwane przeskalowanym sygnałem zegarowym o częstotliwości 100MHz. Każdy *Prescaler* jest ustawiony na inną wartość w celu uzyskania losowości układu. Wartość zliczania liczników jest modyfikowana za pomocą wejść MODE\_1, MODE\_2, MODE\_3(odpowiednio liczy do 4,3,2).



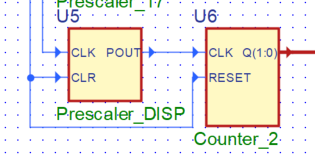
*Rys.1 Liczniki taktowane przeskalowanym sygnałem zegarowym*

1. Segment odpowiedzialny za dekodowanie liczby na wyświetlacz 7-segmentowy oraz wyświetlanie jej odpowiednim miejscu. Dane *IN\_1, IN\_2, IN\_3* są wysyłane w kolejności do *Decodera\_1* co jeden takt sygnału *CLK\_IN*, gdzie później są dekodowane i wysyłane na wyjście. Element *Annode* uruchamia odpowiedni wyświetlacz(1-3) co takt sygnału *CLK\_IN.*



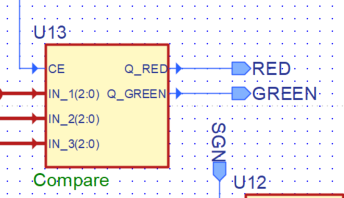
*Rys.2 Multiplekser z dekoderem do wyboru i wyświetlenia danej*

1. Część odpowiedzialna za ustawienie odpowiedniej częstotliwości odświeżania wyświetlaczy 7-segmentowych oraz sygnału taktującego dla wspomnianego wcześniej multipleksera *Display*.

**

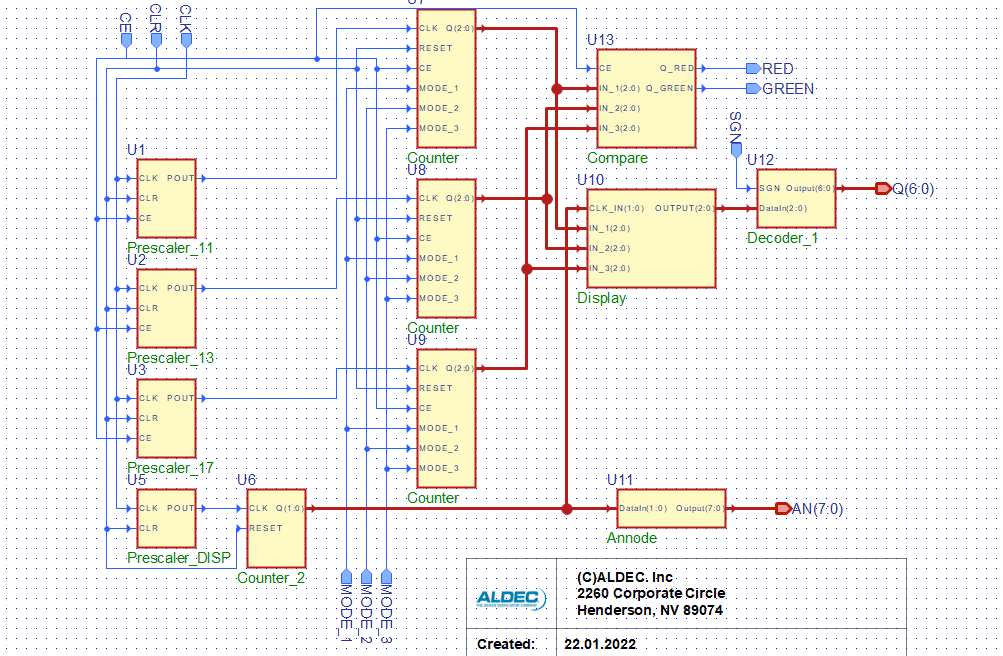
*Rys.3 Ustalenie częstotliwości odświeżania wyświetlaczy 7-seg*

1. Moduł odpowiedzialny za porównywanie wartości oraz obsługę diody RGD.



*Rys.4 Obsługa diody RGB*

1. Schemat blokowy



*Rys.4 Schemat blokowy całego układu*

**5. Opis działania oraz poszczególnych modułów**

Sygnał z wbudowanego zegara o częstotliwości 100MHz jest kierowany do 3 różnych preskalerów o następujących dzielnikach 5000011, 5200000, 5740234. Następnie sygnał trafia do liczników zliczających do wartości równej 5. Dane z liczników trafiają do multipleksera, gdzie wysyłane są do dekodera przy liczeniu *Countera\_2 (IN\_1 dla “00”, IN\_2 dla “01”, IN\_3 dla “10”). Prescaler\_DISP* jest ustawiony w taki sposób, aby częstotliwość odświeżania wyświetlaczy 7-seg była równa 500Hz (podzielone przez 200000). Do naszej gry wystarczające są 3 wyświetlacze, dlatego włączanie ich odbywa się również za pomocą *Countera\_2* zliczającego do 3.

Projektskłada się z następujących modułów:

1. *Prescaler służący do dzielenia częstotliwości zegara*

| *Port* | *Wejście/Wyjście* | *Szerokość [bit]* | *Funkcja* |
| --- | --- | --- | --- |
| *CLK* | *Wejście* | *1* | *Zegar taktujący* |
| *CLR* | *Wejście* | *1* | *Reset* |
| *CE* | *Wejście* | *1* | *Clock Enable* |
| *POUT* | *Wyjście* | *1* | *Podzielony sygnał zegarowy* |

1. *Counter - licznik zliczający od 0 do 5,4,3,2.*

| *Port* | *Wejście/Wyjście* | *Szerokość [bit]* | *Funkcja* |
| --- | --- | --- | --- |
| *CLK* | *Wejście* | *1* | *Zegar taktujący* |
| *CLR* | *Wejście* | *1* | *Reset* |
| *CE* | *Wejście* | *1* | *Clock Enable* |
| *Q* | *Wyjście* | *3* | *Dana/liczba wysyłana do multipleksera* |
| *MODE\_1,MODE\_2, MODE\_3* | *Wejście* | *1* | *Ustawienie wartości zliczania* |

1. *Display - multiplekser odpowiedzialny za wybór odpowiedniej danej do wyświetlania*

| *Port* | *Wejście/Wyjście* | *Szerokość [bit]* | *Funkcja* |
| --- | --- | --- | --- |
| *CLK\_IN* | *Wejście* | *2* | *Podzielony sygnał taktujący* |
| *IN\_1* | *Wejście* | *3* | *Dane z pierwszego licznika* |
| *IN\_2* | *Wejście* | *3* | *Dane z drugiego licznika* |
| *IN3* | *Wejście* | *3* | *Dane z trzeciego licznika* |
| *OUTPUT* | *Wyjście* | *3* | *Wybrana liczba z powyższych zostaje wysłana do dekodera* |

1. *Decoder - dekoduje liczbę na kod do wyświetlacza 7-seg*

| *Port* | *Wejście/Wyjście* | *Szerokość [bit]* | *Funkcja* |
| --- | --- | --- | --- |
| *SGN* | *Wejście* | *1* | *Sygnał odpowiadający za wyświetlanie liczb(SGN=0)/*  *znaków*  *(SGN=1)* |
| *DataIn* | *Wejście* | *3* | *Dane wejściowe* |
| *Output* | *Wejście* | *7* | *Dane wyjściowe do 7-seg* |

1. *Annode - włącza odpowiednie wyświetlacze o określonej częstotliwości odświeżania*

| *Port* | *Wejście/Wyjście* | *Szerokość [bit]* | *Funkcja* |
| --- | --- | --- | --- |
| *DataIn* | *Wejście* | *2* | *Podzielony sygnał taktujący* |
| *Output* | *Wyjście* | *8* | *Ustawia “0” na danym wyświetlaczu co odpowiada jego włączeniu* |

*f) Compare – porównanie wartości liczników przy nieaktywnym CE i włączanie diody czerwonej przy 3 takich samych wartościach i zielonej przy 2.*

| *Port* | *Wejście/Wyjście* | *Szerokość [bit]* | *Funkcja* |
| --- | --- | --- | --- |
| *IN\_1,IN\_2, IN\_3* | *Wejście* | *3* | *Porównywane dane* |
| *CE* | *Wejście* | *1* | *Clock Enable* |
| *Q\_RED* | *Wyjście* | *1* | *Sygnał obsługujący diodę czerwoną* |
| *Q\_GREEN* | *Wyjście* | *1* | *Sygnał obsługujący diodę zieloną* |

***6. Podsumowanie***

Podczas pracy nad projektem zdobyłem doświadczenie z obsługi wyświetlaczy 7-segmentowych na platformie Nexys DDR4. Ponadto wykonanie większego projektu zwiększa umiejętność weryfikacji poprawności działania układu za pomocą TestBenchu. Dzięki temu można się wystrzec błędu na wczesnym etapie projektu.

Bibliografia:

-Nexys DDR4 Reference Manual [[1]](https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual)