Sprawozdanie

**Zamek Szyfrowy**



Autor:

**Michał Czwórnóg**

Prowadzący:

**dr inż. Paweł Rajda**

**1. Założenia Projektowe**

Celem projektu było stworzenie zamku szyfrowego z pomocą oprogramowania Active-HDL używając do tego płytki NEXYS 4 DDR oraz klawiatury KYPD, która posiada 16 przycisków (przyjmujemy 8 cyfrową kombinację liczb od 0 do F).

**2. Opis funkcjonalny**

Po zaprogramowaniu płytki na ośmiu wyświetlaczach 7 segmentowych powinny być widoczne myślniki. Aby wprowadzić kod należy posługiwać się wspomnianą wyżej zewnętrzną klawiaturą. Po wprowadzeniu każdej liczby na kolejnych wyświetlaczach powinny się pojawiać wybrane liczby.

Gdy zostanie wprowadzone wszystkie 8 cyfr użytkownik zostaje poinformowany o poprawnośći wpisanego kod za pomocą diody RGB (na płytce oznaczonej jako LD 17). Jeżeli dioda świeci się na czerwono oznacza to że podana kombinacja jest błędna i po naciśnięciu jakiegokolwiek przycisku na klawiaturze zamek sie zresetuje i znowu będzie można wprowadzić kombinację. Jeżeli natomiast dioda zaświeci się na zielono będzie to oznaczało podanie prawidłowej kombinacji.

Po podaniu prawidłowej kombinacji klawiatura zostanie “zablokowana” i wciskanie na niej przycisków nie ma wpływu na działanie układu. Zostają natomiast odblokowane przyciski BTNL i BTND. Przycisk BTNL pozwala ponownie “zamknąć” zamek (zamek zostanie zresetowany). Wciśnięcie przycisku BTND spowoduje zaświecenie diody RGB na niebiesko i pozwoli zmienić użytkownikowi kod zamka.

Podczas zmiany kodu dioda RGB cały czas świeci się na niebiesko. Wprowadzanie nowej kombinacji odbywa się w taki sam sposób jak wprowadzanie kodu. Po wpisaniu 8 cyfr kombinacji dioda zmieni kolor na zielony co oznacza powrót do stanu po odblokowaniu zamka.

Kod zamka umożliwia również zresetowanie wpisywanej kombinacji (zarówno podczas odblokowywania jak i zmiany kodu) i służy do tego przycisk BTNR.

W projekcie zostały użyte następujące przyciski / przełączniki:

* przełączniki SW0 => służy do odblokowania zegara w układzie (1 - odblokowany, 0 - zablokowany)
* wyświetlacze 7-seg => wyswietlają wpisaną kombinację (służa wyłącznie wizualizacji)
* dioda RGB (LD 17) => informuje o stanie w którym znajduje się zamek
* przycisk BTNL => ponowne zablokowanie zamka po podaniu prawidłowej kombinacji
* przycisk BTND => przejście w stan ustawianie nowej kombinacji dla zamka (możliwe wyłącznie po podaniu dobrej kombinacji
* przycisk BTNR => reset wpisywanej kombinacji (pozwala zacząć wpisywać kombinację od nowa zarówno podczas odblokowywania jak i zmiany kodu)
* przycisk BTNU => zresetowanie układu do stanu zerowego
* zewnętrzna klawiatura KYPD16 (podpięta pod złącze JA) => pozwala na wpisywanie kodu

**3. Informacje dotyczące kluczowych elementów**

W projekcie zostały użyte 3 nietrywialne elementy:

1. Odczytywanie wciśniętego przycisku z KYPD16

Jako że klawiatura została zrealizowana w taki a nie inny sposób (patrz punkt 7)

by odczytać jaki przycisk został wciśnięty poprzez multipleksację trzeba wybrać kolumnę z której będziemy odczytywać czy został wciśnięty przycisk. Dopiero po pewnym odstępie czasu można odczytać z kolumny czy i który z przycisków został wciśnięty. Jeśli przycisk zostanie wykryty zostaje wygenerowany sygnał (Pressed) informujący o wciśnięciu przycisku. Sygnał jest zerowany gdy żaden przycisk nie jest wciśnięty.

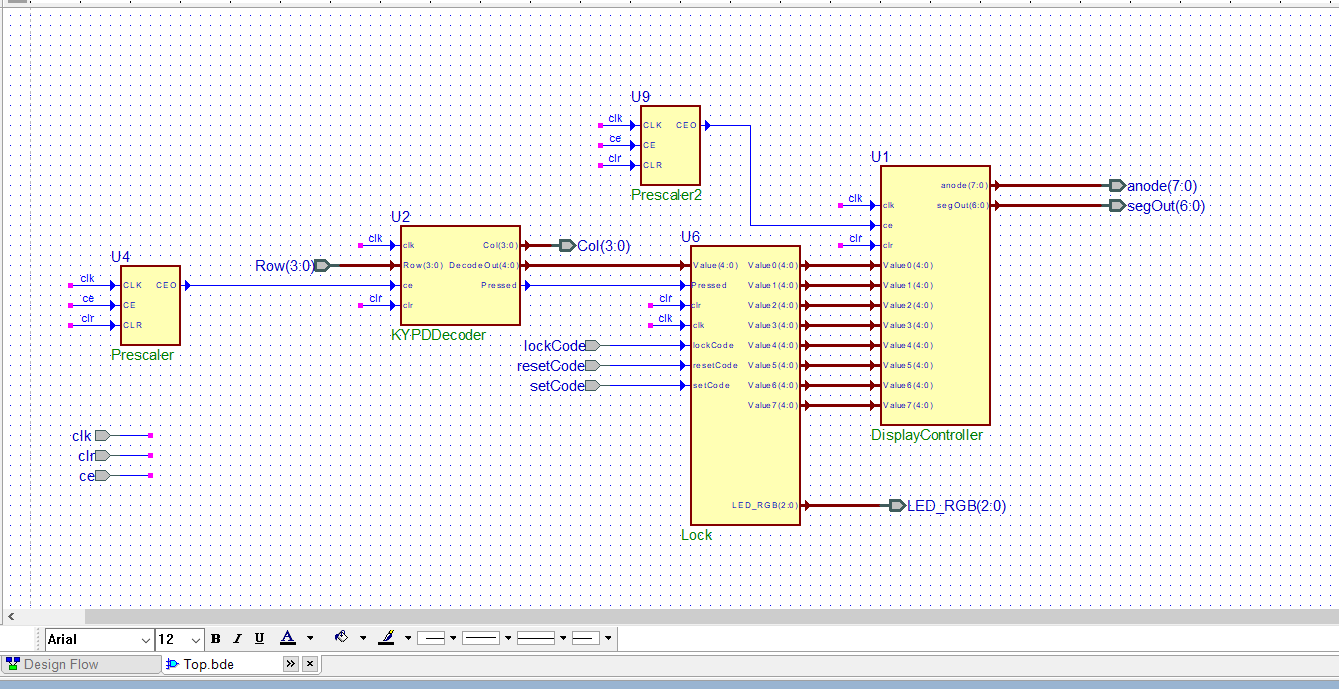
1. Zamek

Algorytm zamka został zrealizowany jako automat

1. Wyświetlanie 7-segmentowe.

Wyświetlanie na 8 wyświetlaczach zostało zrealizowane przez multipleksację.

**4. Schemat Blokowy**



Projekt został podzielony na 3 bloki:

* KYPDDecoder
* Lock
* DisplayController

KYPDDecoder odpowiada za odczytanie naciśniętego przycisku i podanie sygnału Pressed na wejście bloku Lock. W bloku Lock został zrealizowany automat zamka. Ostatni blok DisplayController odpowiada za wyświetlanie odpowiednich wartości (Value), które są podawane przez Lock, na poszczególnych wyświetlaczach 7-segmentowych.

Do projektu zostały także dodane 2 preskalery mające za zadanie obniżyć częstotliwość zegara.

**5. Opis Bloków (porty)**

|  |  |  |  |
| --- | --- | --- | --- |
| KYPDDecoder | | | |
| clk | in | 1 | zegar |
| clr | in | 1 | clear |
| ce | in | 1 | clock enable |
| Row | in | 4 | wartość otrzymana z klawiatury |
| Col | out | 4 | wartość wysłana z bloku do klawiatury |
| DecodeOut | out | 5 | numer przycisku wciśnięty na klawiaturze |
| Pressed | out | 1 | sygnał informujący o wciśnięciu przycisku |

|  |  |  |  |
| --- | --- | --- | --- |
| Lock | | | |
| Value | in | 5 | Ostatnio wciśnięty przycisk |
| Pressed | in | 1 | został wciśnięty przycisk |
| clr | in | 1 | clear |
| clk | in | 1 | zegar |
| lockCode | in | 1 | wartość przycisku BTNL |
| resetCode | in | 1 | wartość przycisku BTNR |
| setCode | in | 1 | wartość przycisku BTND |
| LED\_RGB | out | 3 | wartość wysłana na diode LD 17 |
| Value0 - Value7 | out | 5 | wartości wyświetlane na wyświetlaczach 7-seg |

|  |  |  |  |
| --- | --- | --- | --- |
| DisplayController | | | |
| Value0 - Value7 | out | 5 | wartości wyświetlane na wyświetlaczach 7-seg |
| Pressed | in | 1 | został wciśnięty przycisk |
| clr | in | 1 | clear |
| clk | in | 1 | zegar |
| ce | in | 1 | clock enable |
| anode | out | 8 | wybór wyświetlacza na którym będzie wyświetlona wartość segOut |
| segOut | out | 7 | wartość wyświetlana na wyświetlaczu 7-seg |

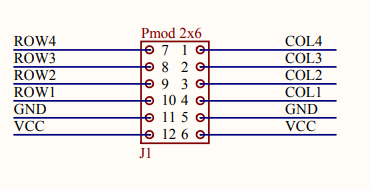
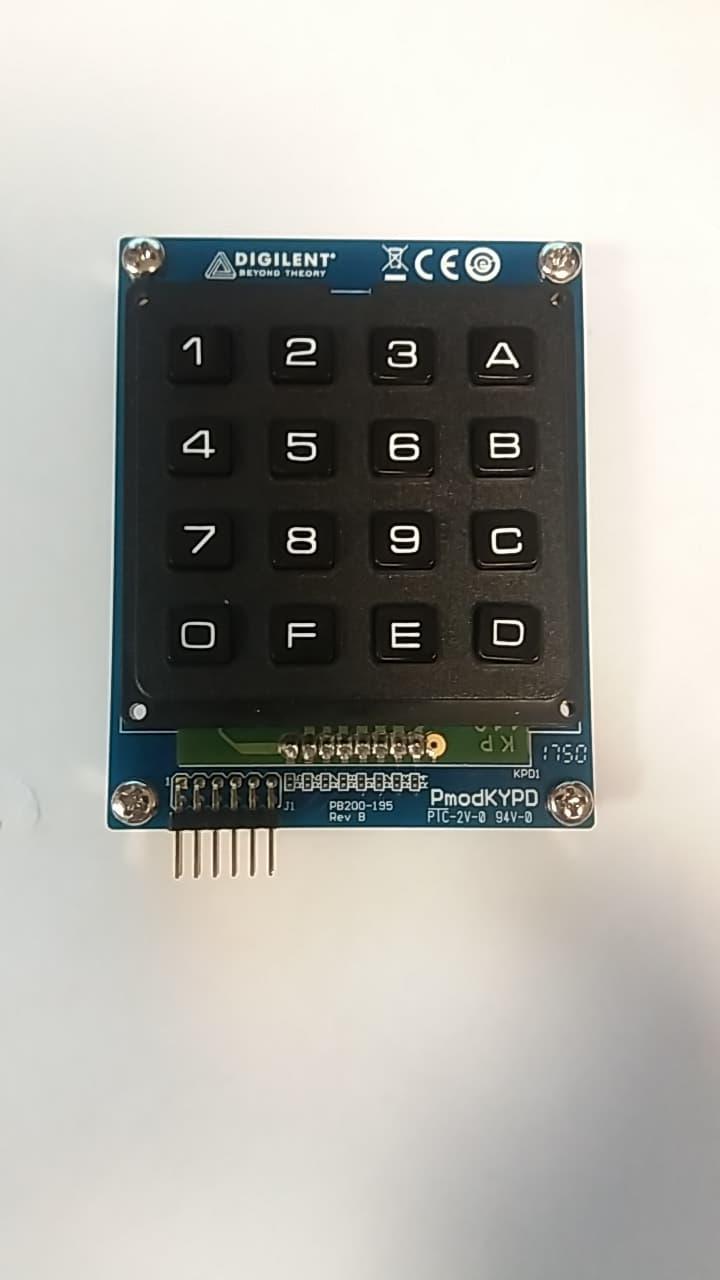
**6. Opis Algorytmu Bloku Lock**

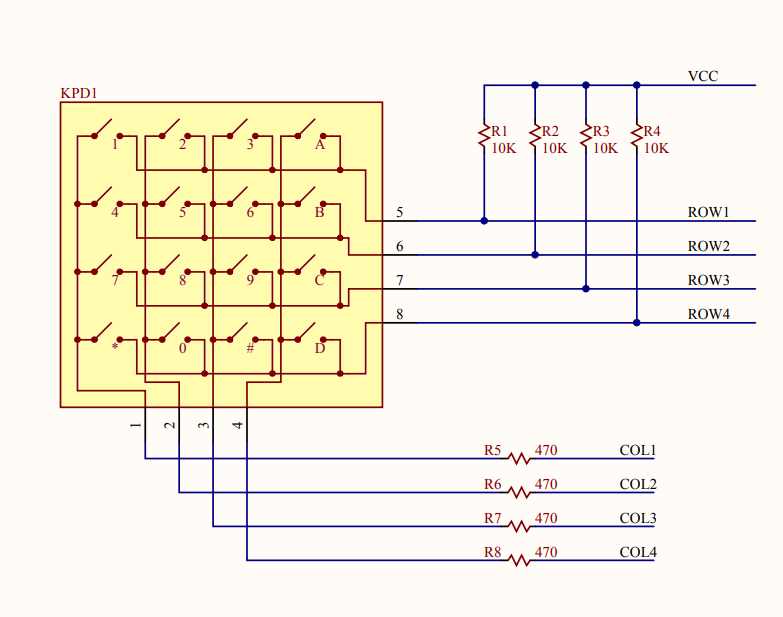
Block Lock został zaimplementowany jako automat, w którym stan:

* initialize => stan początkowy (resetuje automat nie resetuje kodu do tego służy sygnał clr) po jego wykonaniu automat przechodzi do stanu S1
* enterCode => zaimplementowany moduł zamka, po wpisaniu 8 cyfr przechodzi do stanu S2, po wciśnięciu przycisku BTNR następny stan to S0
* checkCode => odpowiada za wykrycie dobrego / złego kodu w zależność od tego przekierowywyje do stanu S3 lub S4
* badCode => został wprowadzony zły kod, po wciśnięciu przycisku z klawiatury przechodzi do S0
* goodCode => dobry kod, klawiatura jest “zablokowana” w zależności od przycisków BTN automat przechodzi do kolejnych stanów S0, S5
* initCodeChange => przygotowanie do wprowadzenia nowej kombinacji, przechodzi do stanu S6,
* enterNewCode => zaimplementowany moduł zmieniający kombinacje, po wpisaniu 8 cyfr przechodzi do stanu S7, po wciśnięciu przycisku BTNR następny stan to S5
* changeCod => ustawia nową kombinację, przechodzi do stanu S2

**7. Dodatkowe układy**

Klawiatura KYPD16





**8. Doświadczenia**

Podczas realizacji projektu można było natknąć się na wiele problemów wliczając w to instalację środowiska Active-HDL oraz Vivado. Najwięcej problemów przysporzyła zewnętrzna klawiatura którą trzeba obsługiwać w dość ciekawy sposób. Podczas testowania kodu pojawiły się problemy z wizualizacją (nie było widać co się wpisuje), co zaskutkowało zaimplementowanie bloku DisplayController odpowiedzialnego za poprawne wyświetlanie wartości na wyświetlaczach 7-segmentowych.

Podsumowując, projekt był ciekawy i pozwalał nauczyć się zagadnień związanych z multipleksacją sygnałów, obsługą zewnętrznej klawiatury, a także konstrukcją automatu.