

Kalkulator 2-działaniowy

Dawid Ilba, Elektronika 3 rok
Wydział Informatyki Elektroniki i Telekomunikacji
Akademia Górniczo-Hutnicza
Opiekun : Dr inż. Paweł Rajda
Projektowanie Systemów Cyfrowych

Kraków 2020

Spis Treści

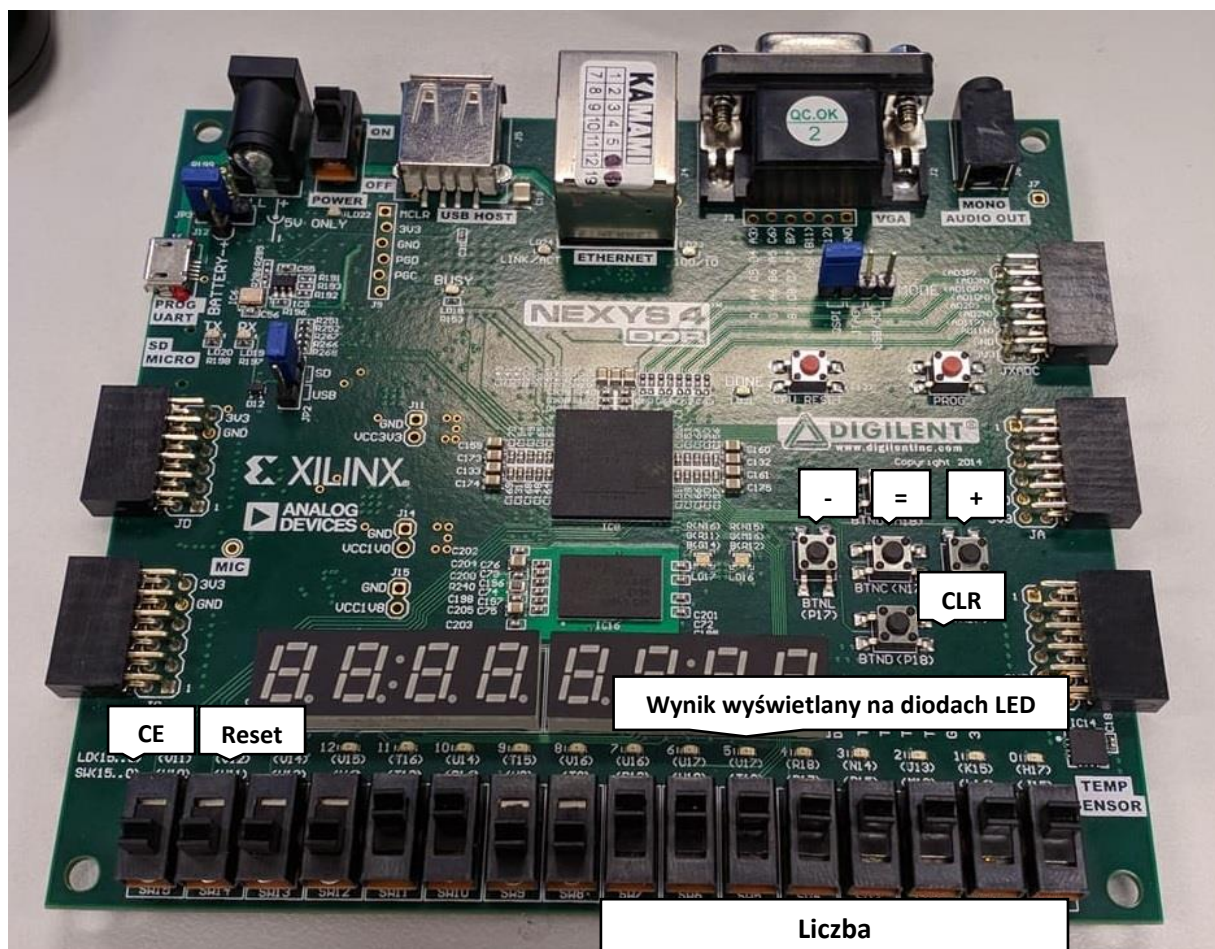
<i>1. Założenia projektowe</i>	<i>3</i>
<i>2. Opis funkcjonalny</i>	<i>3</i>
<i>3. Schemat blokowy projektu</i>	<i>4</i>
<i>4. Graf stanów i realizacja</i>	<i>5</i>
<i>5. Uwagi i wnioski</i>	<i>5</i>

1. Założenia projektowe

Celem projektu było stworzenie układu realizującego funkcję dodawania i odejmowania dwóch 8-bitowych liczb binarnych. Liczby wprowadzane są za pomocą przełączników, a działania (+, -, =, clear) za pomocą przycisków. Wynik wyświetlany jest na diodach LED. Projekt napisany został w programie Aldec Active HDL Student Version, a zaimplementowany na płycie Nexys 4 za pomocą Vivado Lab 2016.4.

2. Opis funkcjonalny

Do projektu wykorzystywana została tylko płytką Nexys 4 bez dodatkowych modułów.

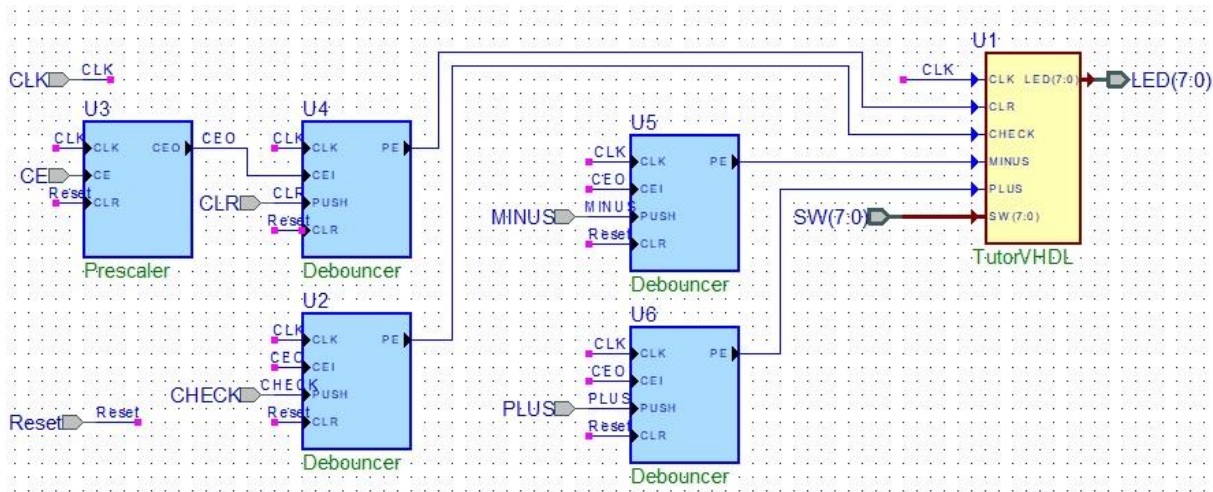


Rys 1. Opis przycisków, przełączników, i diod na płycie

Po uruchomieniu płytki należy ustawić przełącznik SW15 - CE (Clock Enable) na pozycję 1. Następnie przy pomocy przełączników (SW7 – SW0) „wpisujemy” pierwszą liczbę. Kolejno wybieramy znak działania (BNTR (+) żeby dodać liczbę, BNTL(-) aby odjąć).

Po kliknięciu przycisku wprowadzamy kolejną liczbę za pomocą przełączników. Teraz zostało już tylko przycisnąć przycisk BTNC (=) i wynik pojawi się na diodach LED (LED7 - LED0). Wynik będzie widoczny do momentu przyciśnięcia BTND (CLR). Po jego wciśnięciu diody zostaną zgaszone, a my możemy ponownie dodawać lub odejmować. Przełącznik SW14 (Reset) odpowiedzialny jest za zresetowanie układu.

3. Schemat blokowy projektu



Rys 2. Schemat blokowy

Na schemacie blokowym pierwsze co możemy dostrzec to fakt, iż wykorzystane zostały debouncer'y, aż do 4 przycisków. Debouncer to moduł odsprężania czasowego eliminujący niekorzystne zjawisko drgań styków.

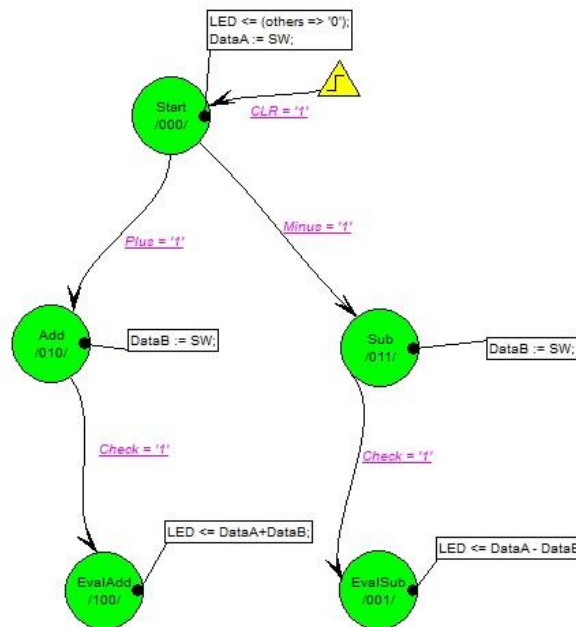
Po szczegółowej analizie programu (poniżej zamieszczę graf stanów – rys 3) okazało się, iż nie wszystkie są potrzebne, przez to że w poszczególnym stanie nie można trafić „źle”. Jedyny deboucer, który powinien być wykorzystany to ten dla przycisku CLR. Dla pozostałych przycisków nie są konieczne, choć do programów wymagających większej precyzji są niezbędne.

W projekcie został wykorzystany zewnętrzny oscylator o częstotliwości 100MHz wraz z odpowiednio skonstruowanym modułem preskalera (dzielnikiem częstotliwości).

Moduł projektowy – opis wejść i wyjść :

Nazwa portu	Kierunek	Szerokość	Funkcja
CLK	In	STD_LOGIC	Sygnał zegara
CLR	In	STD_LOGIC	Przycisk „Clear”
CHECK	In	STD_LOGIC	Przycisk „=”
MINUS	In	STD_LOGIC	Przycisk „-”
PLUS	In	STD_LOGIC	Przycisk „+”
SW	In	STD_LOGIC_VECTOR (7 downto 0)	Liczba 8 bitowa
LED	Out	STD_LOGIC_VECTOR(7 downto 0)	Wynik 8 bitowy

4. Graf stanów i realizacja



Rys 3. Graf stanów

Pisanie projektu rozpocząłem od narysowania powyższego grafu, który składa się z 5 stanów. W stanie pierwszym – Start, usuwamy wartość sygnału wyjściowego LED, a zmiennej przypisujemy wartość z przełącznika. Następnie po przyciśnięciu przycisku „+” lub „-” przechodzimy do kolejnych stanów, w których drugiej zmiennej przypisujemy wartość z przełącznika. Następnie po kliknięciu przycisku „=” przechodzimy do stanu przypisującego wyjściu wartość dodawania lub odejmowania. Niestety po wygenerowaniu kodu przez Active HDL miałem problemy z testbenchem. Postanowiłem stworzyć nowy projekt wykorzystując wygenerowany kod z grafu, który odpowiednio edytowałem. Jak się okazało całość funkcjonowała poprawnie. Następnie zaprojektowałem schemat blokowy, dodatkowo wykorzystując moduły debouncera i preskalera. Kolejno zrobiłem syntezę i implementację, która przebiegła bezproblemowo. Program zaimplementowałem na płytce i sprawdziłem jego działanie. Układ działa poprawnie.

5. Uwagi i wnioski

Pisząc projekt w Aldec’u Active HDL pracowało mi się znacznie przyjemniej niż w programie Vivado HDL, w którym programowałem na 3 semestrze studiów. Mimo, iż wygenerowany kod z grafów nie chciał zadziałać, wystarczyła lekka edycja kodu i układ funkcjonował tak jak zakładałem.