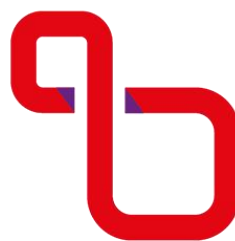


Wydział Automatyki, Elektroniki i Informatyki
Kierunek Teleinformatyka



Politechnika
Śląska



UCZELNIA
BADAWCZA
INICJATYWA DOSKONAŁOŚCI

Systemy Procesorowe w Układach
Reprogramowalnych

Projekt budowy prostego mikroprocesora

Autorzy:

Adam Skóbel

Piotr Kaleta

1. Cel projektu

Celem projektu jest nauka praktycznych aspektów procesu projektowania układów, realizowanych w strukturach programowalnych oraz zdobycie umiejętności korzystania z języków opisu sprzętu i posługiwania się narzędziami wspomagającymi proces projektowania.

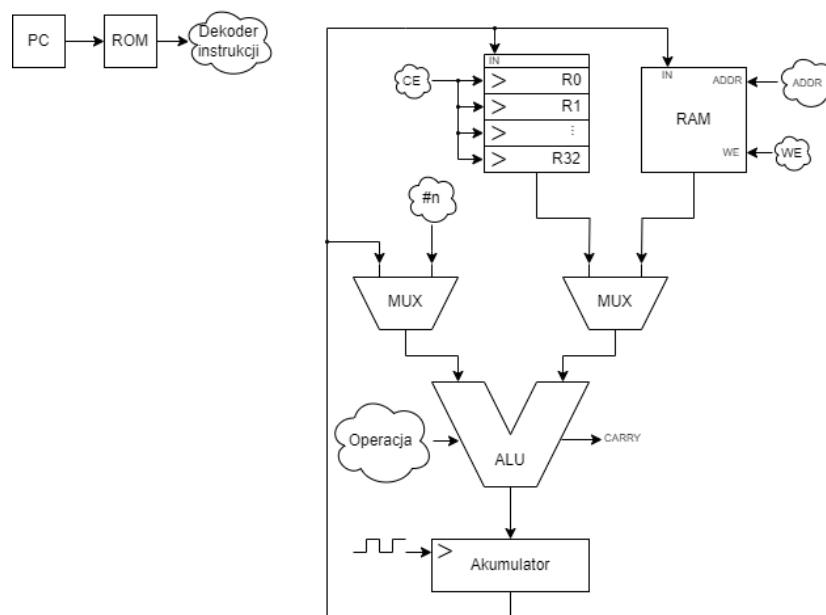
2. Cel projektu i założenia ogólne

Celem projektu jest stworzenie za pomocą środowiska Vivado mikroprocesora 16 bitowego, używając do tego języka opisu sprzętu Verilog. Projekt powinien składać się z takich elementów jak:

- pamięć ROM 64kB
- pamięć RAM 64kB,
- blok rejestrowy 1x32 (16-bit),
- blok ALU,
- licznik programu (PC),
- dekodek instrukcji,
- akumulator.

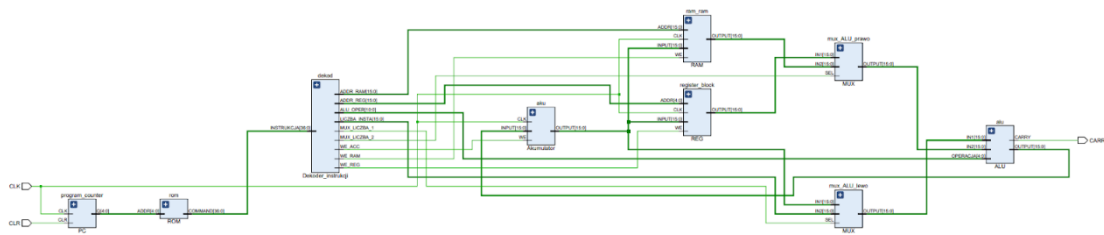
3. Opis projektu

Poniżej przedstawiono ogólny schemat koncepcyjny projektu.



Rysunek 1: Koncepcyjny schemat blokowy

Docelową realizację projektu przedstawiono poniżej schemat elektryczny projektu, który wygenerowany został z użyciem narzędzi programu Vivado.



Rysunek 2: Schemat elektryczny

Sygnałami wchodzącymi na wejście układu jest sygnał zegarowy CLK oraz sygnał zerowania licznika programu CLR. Sygnałem wyjściowym jest wyjście CARRY z modułu ALU.

Operacje zapisane są w module pamięci ROM, a wybierane są za pomocą licznika rozkazów PC. Kody operacji przesyłane są do modułu dekodera operacji, który rozdziela przychodzące kody operacji i rozsyła je do odpowiednich podzespołów, które odpowiednio interpretują dane przychodzące. Na umieszczonym wcześniej schemacie blokowym zaobserwować można, że dane pochodzące z modułu dekodera są zwizualizowane za pomocą symbolu chmury.

Jednostka arytmetyczno-logiczna ALU przyjmuje operację do wykonania i po zrealizowaniu operacji na zmiennych wejściowych, przekazuje wynik do akumulatora, z którego dane mogą zostać przekazane w 3 inne miejsca docelowe: do modułu ALU, do pamięci RAM oraz do modułu rejestrów. Operacje umożliwiają także czerpanie danych z wymienionych przed chwilą modułów, dodając do tego zmienną bezpośrednią, która także można wystać do jednostki ALU.

Jednostka arytmetyczno-logiczna pozwala na wykonywanie operacji na wprowadzanych do modułu zmiennych A oraz B (IN1 oraz IN2). Jest to m.in.:

- Dodawanie
- Odejmowanie
- Inkrementacja zmiennej A
- Inkrementacja zmiennej B
- Dekrementacja zmiennej A
- Dekrementacja zmiennej B
- Negacja zmiennej A
- Negacja zmiennej B
- Operacja porównania zmiennych

4. Wyniki

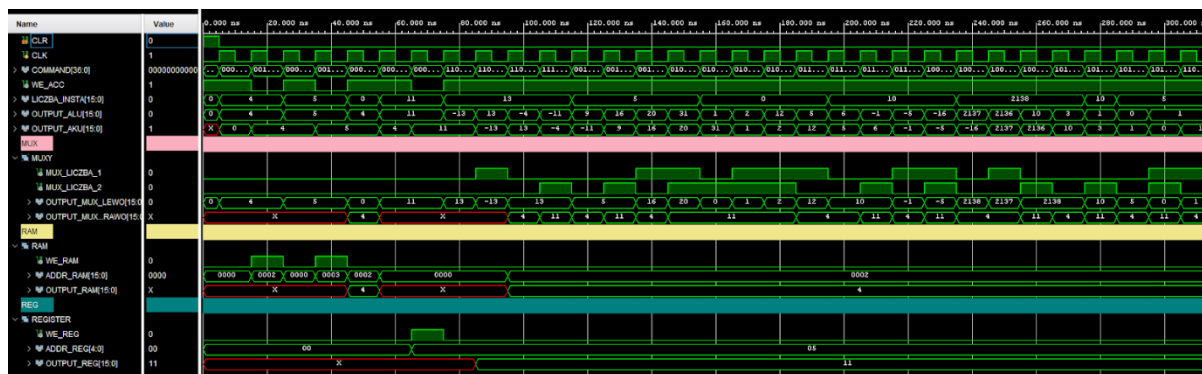
Testowaniu podlegał zestaw instrukcji zapisany w pamięci ROM układu. Został on przedstawiony poniżej wraz z krótkim opisem.

```

00000 - MOV A, INSTA - z insta do akumulatora
00001 - MOV A, RAM - z RAMU do akumulatora
00010 - MOV A, REG - z REJESTRU do akumulatora
00011 - MOV REG, A - z akumulatora do REJESTRU
00100 - MOV RAM, A - z akumulatora do RAMU
00101 - ADD RAM, INSTA
00110 - ADD REG, INSTA
00111 - ADD RAM, ACC
01000 - ADD REG, ACC
01001 - INC INSTA
01010 - INC ACC
01011 - INC REG
01100 - INC RAM
01101 - SUB INSTA, RAM
01110 - SUB INSTA, REG
01111 - SUB ACC, RAM
10000 - SUB ACC, REG
10001 - DEC INSTA
10010 - DEC ACC
10011 - DEC REG
10100 - DEC RAM
10101 - CMP INSTA, REG
10110 - CMP INSTA, RAM
10111 - CMP ACC, REG
11000 - CMP ACC, RAM
11001 - NOT INSTA
11010 - NOT ACC
11011 - NOT RAM
11100 - NOT REG

```

Poniżej przedstawiono wynik przeprowadzonej symulacji.



Rysunek 3: Symulacja programu Vivado

Jak widać, operacje przedstawione wcześniej wykonane zostały po kolei bez widocznych błędów.

5. Wnioski

Dzięki zajęciom projektowym dowiedzieliśmy w jaki sposób należy projektować układy FPGA. Udało nam się odświeżyć wiedzę z zakresu projektowania modułów oraz testowania zaprojektowanej części układu/modułu.

Zdobyto także wiedzę z zakresu wczesnego planowania projektu. Nieodzownym elementem projektowania okazało się tworzenie metodą krokową poszczególnych modułów, a następnie poddawania ich testom/symulacjom. Na końcu wystarczyło cierpliwie łączyć moduły oraz poddawać je dalszym testom.

Podczas projektowania natrafiono na problem z modułem RAM oraz modułem akumulatora. Okazało się, że w przypadku modułu akumulatora niefortunnie użyto wyzwalań powodujące opóźnienie w pracy układu o jeden cykl zegarowy. W przypadku modułu pamięci RAM, źle zdefiniowano tablicę rejestrową, przez co zapisywane dane miały niepełny wymiar. Kolejnym błędem było także użycie wysokiej impedancji na wyjściach modułów w przypadku ich nie używania. Powodowało to poważne błędy w funkcjonowaniu całego układu, które zdiagnozowano, używając narzędzi symulacyjnych. Wszystkie te błędy rozwiązano z pomocą konsultacji prowadzącego.