

Warszawa, dnia 6 grudnia 2022 r.

Systemy Cyfrowe i Komputerowe, Projekt Indywidualny 1 – wariant 48.

Spis Treści:

1. Założenia projektu;
2. Realizacja poszczególnych modułów wewnętrznych, opisy realizacji testbenchów i raporty z syntez logicznych;
  - 2.1. Moduł dodawania,
  - 2.2. Moduł porównywania,
  - 2.3. Moduł ustawiania bitu,
  - 2.4. Moduł konwersji,
3. Realizacja głównego modułu ALU oraz testbencha ALU;
4. Graficzny model realizacji ALU;

Ad 1.

Realizowany przeze mnie projekt ma na celu zrealizowanie jednostki arytmetycznej ALU złożonej z czterech wewnętrznych modułów na dwóch danych argumentach wejściowych, w moim przypadku sygnały wejściowe i sygnał wyjściowy przyjmują wartości maksymalnie 32-bitowe. Listy wejść i wyjść poszczególnych modułów oraz jednostki arytmetyczno-logicznej znajdują się w adnotacjach poniżej.

Ad 2.1

Moduł dodawania – dodawanie argumentu A i B.

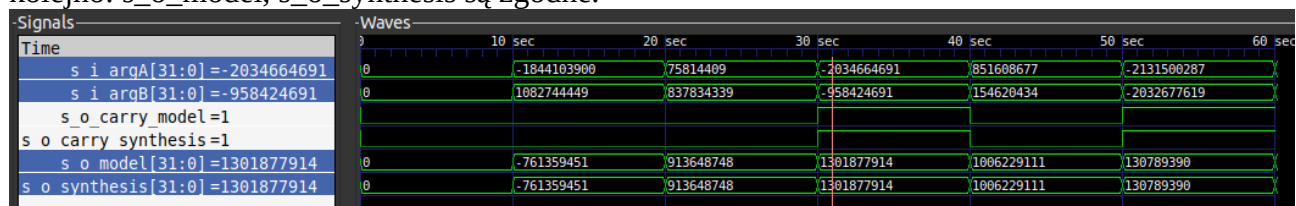
Wejścia i Wyjścia:

- i\_argA – 32-bitowe wejście argumentu A;
- i\_argB – 32-bitowe wejście argumentu B;
- o\_result – 32-bitowy wynik operacji;
- o\_carry – bit przepełnienia, równy 1, gdy wynik operacji przekracza zakres 32 bitów.

Parametry wewnętrzne:

- WIDTH – ma przypisaną wartość 32 i jest parametrem wielkości danych wejściowych;

Synteza modułu przebiegła bez błędów, w testbenchu została sprawdzona zgodność działania modułów przed syntezą (modelu) i po syntezie (plik z końcówką \_rtl). Na grafice poniżej znajduje się zrzut ekranu pokazujący działanie testbencha, wyniki działań modułów przed i po syntezie, kolejno: s\_o\_model, s\_o\_synthesis są zgodne.



Ad 2.2

Moduł porównywania – porównywanie argumentu A i B – operacja sprawdzania, czy  $A \leq B$  – jeśli warunek jest spełniony, wówczas wynikiem jest 1 w przeciwnym wypadku jest to 0;

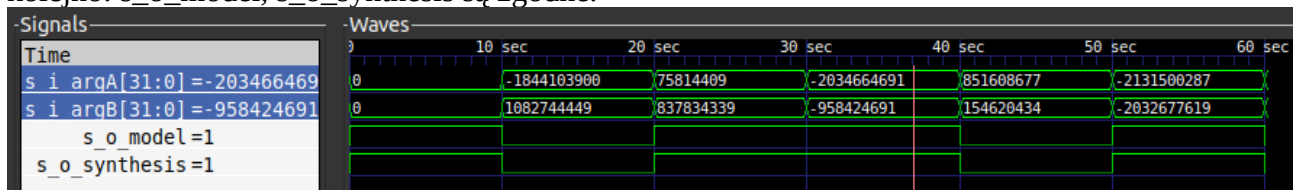
Wejścia i Wyjścia:

- i\_argA – 32-bitowe wejście argumentu A;
- i\_argB – 32-bitowe wejście argumentu B;
- o\_result – 1-bitowy wynik operacji;

Parametry wewnętrzne:

- WIDTH – ma przypisaną wartość 32 i jest parametrem wielkości danych wejściowych;

Synteza modułu przebiegła bez błędów, w testbenchu została sprawdzona zgodność działania modułów przed syntezą (modelu) i po syntezie (plik z końcówką \_rtl). Na grafice poniżej znajduje się zrzut ekranu pokazujący działanie testbenchu, wyniki działań modułów przed i po syntezie, kolejno: s\_o\_model, s\_o\_synthesis są zgodne.



#### Ad 2.3

Moduł ustawiania – ustawienie bitu w argumencie A na wartość 1; numer bitu jest określony w argumencie B; zgłoszenie błędu jeśli wartość w B jest ujemna lub przekracza liczbę bitów w A.

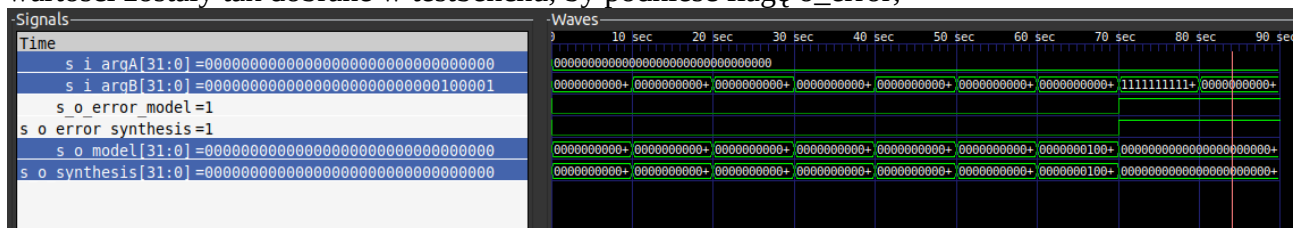
Wejścia i Wyjścia:

- i\_argA – 32-bitowe wejście argumentu A;
- i\_argB – 32-bitowe wejście argumentu B;
- o\_result – 32-bitowy wynik operacji;
- o\_error – bit błędu, równy 1, gdy wartość podana w argumencie B jest większa od 31 lub mniejsza od 0.

Parametry wewnętrzne:

- BITS – ma przypisaną wartość 32 i jest parametrem wielkości danych wejściowych i wyniku;
- tymczasowy\_rejestr – zmienna lokalna przechowująca wynik operacji, na której możliwe będzie dokonanie operacji wielobitowej alternatywy.

Synteza modułu przebiegła bez błędów, w testbenchu została sprawdzona zgodność działania modułów przed syntezą (modelu) i po syntezie (plik z końcówką \_rtl). Na grafice poniżej znajduje się zrzut ekranu pokazujący działanie testbenchu, wyniki działań modułów przed i po syntezie, kolejno: s\_o\_model, s\_o\_synthesis, s\_o\_error\_model, s\_o\_error\_synthesis są zgodne. Ostatnie dwie wartości zostały tak dobrane w testbenchu, by podnieść flagę o\_error;



#### Ad 2.4

Moduł konwersji - konwersja liczby z kodu U2 na kod ZNAK-MODUŁ; jeśli konwersja nie może zostać poprawnie wykonana to ma zostać zgłoszony błąd a wynik konwersji jest nieokreślony.

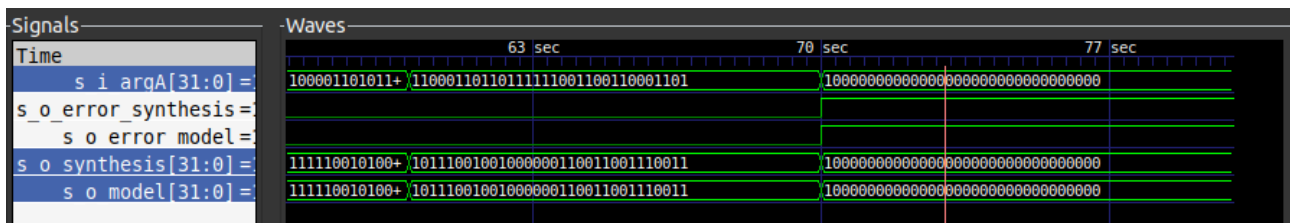
Wejścia i Wyjścia:

- i\_argA – 32-bitowe wejście argumentu A;
- o\_result – 32-bitowy wynik operacji;
- o\_error – bit błędu, równy 1, gdy konwersja nie może zostać poprawnie wykonana;

Parametry wewnętrzne:

- WIDTH – ma przypisaną wartość 32 i jest parametrem wielkości danych wejściowych;

Synteza modułu przebiegła bez błędów, w testbenchu została sprawdzona zgodność działania modułów przed syntezą (modelu) i po syntezie (plik z końcówką \_rtl). Na grafice poniżej znajduje się zrzut ekranu pokazujący działanie testbenchu, wyniki działań modułów przed i po syntezie, kolejno: s\_o\_model, s\_o\_synthesis, s\_o\_error\_model, s\_o\_error\_synthesis są zgodne. Ostatnia wartość i\_argA została tak dobrana, by została podniesiona flaga o\_error.



Ad 3.

Moduł exe\_unit\_w48 – jednostka arytmetyczno-logiczna realizująca wszystkie operacje opisane powyżej na 32-bitowych danych wejściowych.

Wejścia i Wyjścia:

Rodzaj wykonywanej operacji jest określony przez n-bitowe wejście sterujące i\_oper.

Dane m-bitowe, na których wykonywane są operacje są pobierane z dwóch wejść

i\_argA i i\_argB. Wynik jest podawany na m-bitowe wyjście o\_result.

- i\_oper - n-bitowe wejście sterujące (kombinacyjne) określające rodzaj operacji do wykonania na argumentach A i B:

Lista kodów i przypisanych do nich operacji:

- 1'b00 – dodawanie
- 1'b01 - porównywanie
- 1'b10 – ustawianie
- 1'b11 – konwersja
- i\_argA - m-bitowe wejście argumentu A
- i\_argB - m-bitowe wejście argumentu B
- i\_clk - wejście zegara (aktywne zbocze narastające)
- i\_rsn - wejście resetu synchronicznego (wartość logiczna 0 ustawia wszystkie rejestry modułu exe\_unit\_w48 na wartość 0)
- o\_result - wyjście synchroniczne (z rejestru) wyniku operacji, zmiana wyjścia następuje na zboczu narastającym zegara i\_clk
- o\_carry – bit przepełnienia, równy 1, gdy wynik operacji przekracza zakres 32 bitów
- o\_error\_ust – bit błędu, równy 1, gdy wybrana została operacja ustawiania i wartość podana w argumencie B jest większa od 31 lub mniejsza od 0
- o\_error\_konw – bit błędu, równy 1, gdy konwersja nie może zostać poprawnie wykonana;

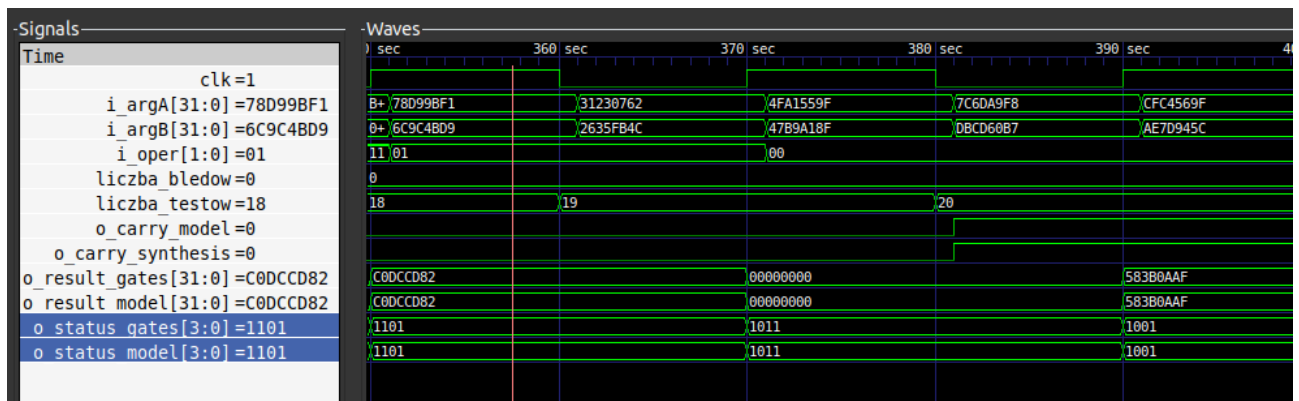
Jednostka exe\_unit\_w48 ma dodatkowe synchroniczne (z rejestru) wyjście

statusu o\_status (zmiana wyjścia następuje na zboczu narastającym zegara i\_clk),

którego kolejno bity określają znaczniki:

- ERROR - operacja nie została wykonana; wartość o\_result jest nieokreślona – występuje tylko dla modułów konwersji i ustawiania, gdy flaga ta jest równa 1, wtedy wynik jest nieokreślony, w przeciwnym wypadku flaga jest równa 0;
- ZERO - wszystkie bity wyniku są ustawione na 0 – flaga jest równa 1, gdy wynik składa się z samych zer, w przeciwnym wypadku flaga jest równa 0;
- NEG - wynik jest liczbą ujemną – flaga jest równa 1, gdy najstarszy bit jest równy 1, w przeciwnym wypadku flaga jest równa 0;
- EVEN - w wyniku jest parzysta liczba jedynek – flaga jest równa 1, gdy po dokonaniu operacji XOR na parach bitów wyniku, wynik tej operacji pozostanie 0, w przeciwnym wypadku, flaga jest równa 0;

Synteza modułu przebiegła bez błędów, w testbenchu została sprawdzona zgodność działania modułów przed syntezą (modelu) i po syntezie (plik z końcówką \_rtl). Na grafice poniżej znajduje się zrzut ekranu pokazujący działanie testbenchu, wyniki działań modułów przed i po syntezie są zgodne



#### Ad. 4 Graficzny model realizacji ALU

