

Systemy cyfrowe i komputerowe

Dokumentacja projektu “exe_unit_w1”

Karol Ambroziński
Nr. albumu: 318488

Spis treści

| | | |
|----------|---|----------|
| 1 | Wejścia, wyjścia, parametry i zakresy ich wartości | 2 |
| 1.1 | Wejścia | 2 |
| 1.2 | Wyjścia | 2 |
| 1.3 | Parametry | 2 |
| 2 | Realizowane funkcje i ich argumenty | 2 |
| 2.1 | Podmoduł <i>mod1</i> : | 2 |
| 2.2 | Podmoduł <i>mod2</i> : | 3 |
| 2.3 | Podmoduł <i>mod3</i> : | 3 |
| 2.4 | Podmoduł <i>mod4</i> : | 3 |
| 3 | Schemat blokowy struktury jednostki | 4 |
| 4 | Sygnaly zaimplementowanych flag i ich wartości | 4 |
| 5 | Przykład użycia modułu | 5 |
| 6 | Lista plików | 5 |
| 7 | Raport z syntezy logicznej | 5 |
| 8 | Raport z symulacji układów | 5 |

1 Wejścia, wyjścia, parametry i zakresy ich wartości

1.1 Wejścia

- *i_oper* - n-bitowe wejście określające wykonywaną operację,
- *i_argA* - m-bitowe wejście,
- *i_argB* - m-bitowe wejście,
- *i_clk* - 1 bitowe wejście zegarowe,
- *i_rsn* - 1 bitowe wejście resetu synchronicznego.

1.2 Wyjścia

- *o_status* - m-bitowe wyjście,
- *o_result* - 4 bitowe wyjście.

1.3 Parametry

- *m* - określa wielkość w bitach główne wejścia danych i wyjść,
- *n* - określa ilość operacji.

2 Realizowane funkcje i ich argumenty

Układ realizuje 4 operacje (4 podmoduły):

2.1 Podmoduł *mod1*:

Odejmowanie argumentów ($A - B$); jeśli operacja nie może zostać wykonana, jednostka zgłasza błąd, a wyjście jest niezdefiniowane.

Wejścia

- *i_argA* - m-bitowe wejście,
- *i_argB* - m-bitowe wejście,

Wyjścia

- *o_result* - m-bitowe wyjście,
- *o_status* - 4 bitowe wyjście.

2.2 Podmoduł *mod2*:

Porównanie argumentów ($A < B$); jeśli warunek jest spełniony to wynikiem jest liczba 1, w przeciwnym wypadku wynikiem jest 0.

Wejścia

- *i_argA* - m-bitowe wejście,
- *i_argB* - m-bitowe wejście,

Wyjścia

- *o_result* - m-bitowe wyjście,
- *o_status* - 4 bitowe wyjście.

2.3 Podmoduł *mod3*:

Ustawienie bitu w argumencie A na wartość 0; numer bitu jest określony w argumencie B; zgłoszenie błędu jeśli wartość B jest ujemna lub przekracza liczbę bitów argumentu A.

Wejścia

- *i_argA* - m-bitowe wejście,
- *i_argB* - m-bitowe wejście,

Wyjścia

- *o_result* - m-bitowe wyjście,
- *o_status* - 4 bitowe wyjście.

2.4 Podmoduł *mod4*:

Konwersja argumentu A z kodu ZNAK-MODUŁ na U2; jeśli konwersja nie może zostać wykonana - zgłaszany jest błąd a wynik jest nieokreślony.

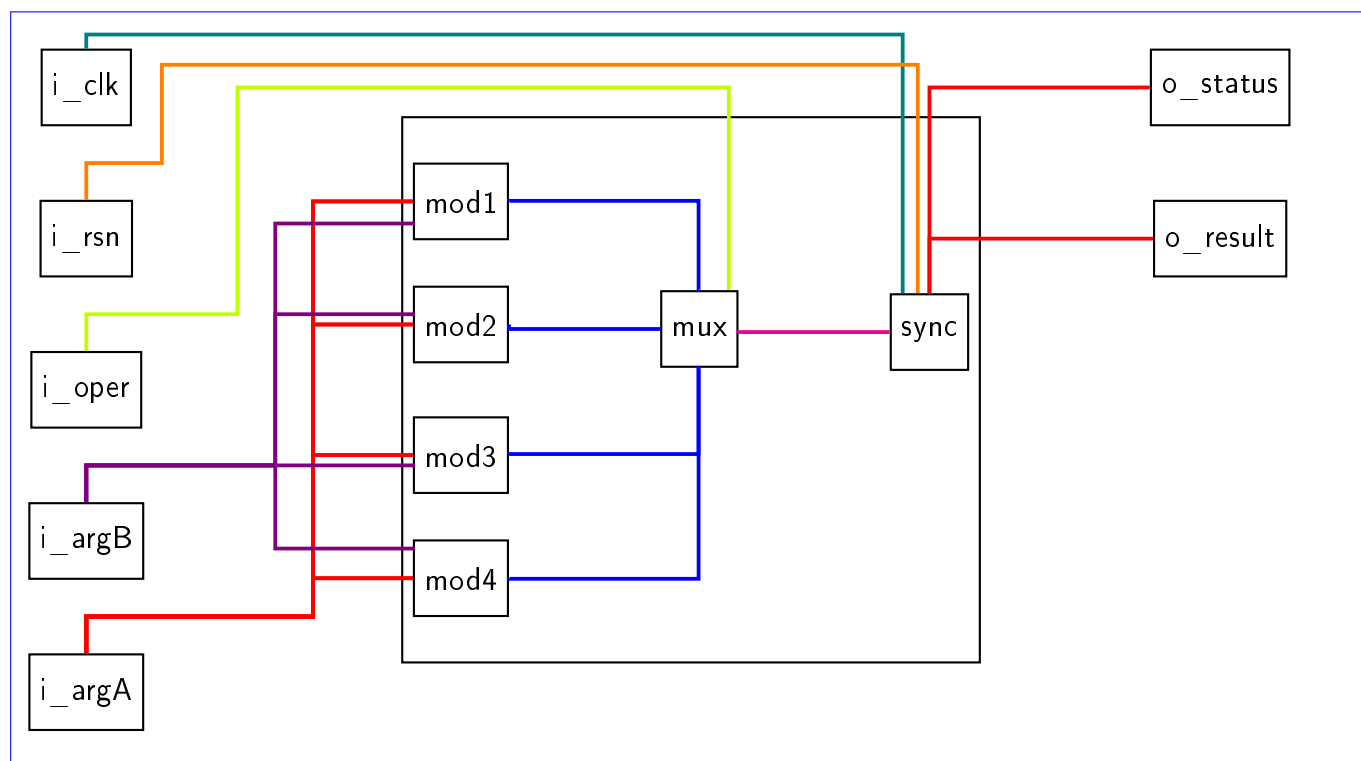
Wejścia

- *i_argA* - m-bitowe wejście,

Wyjścia

- *o_result* - m-bitowe wyjście,
- *o_status* - 4 bitowe wyjście.

3 Schemat blokowy struktury jednostki



Ilustracja nr. 1: Schemat blokowy `exe_unit_w1`

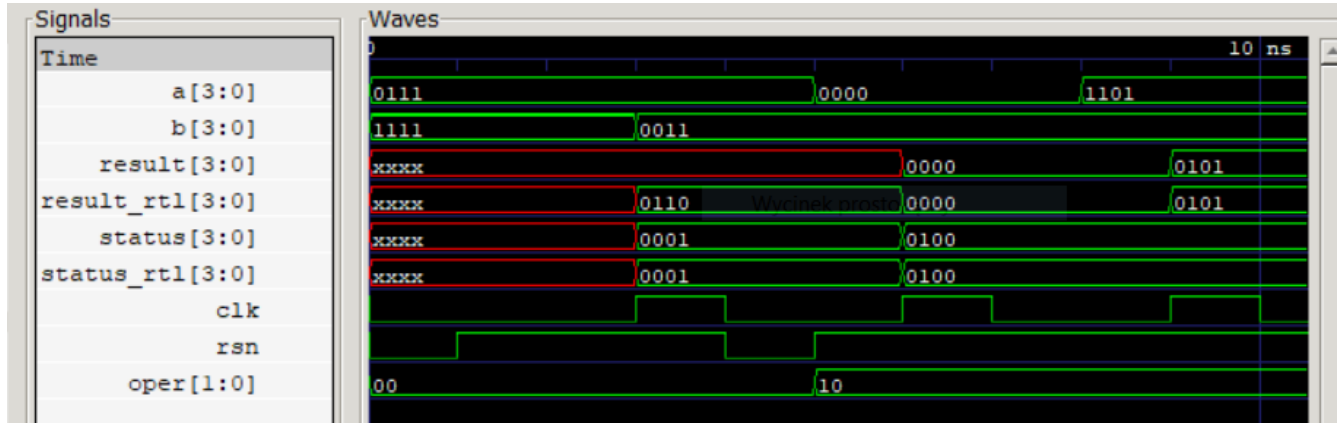
4 Sygnały zaimplementowanych flag i ich wartości

Zaimplementowane flagi `o_status`:

- **ERROR** - operacja nie została wykonana - `o_result` jest nieokreślone; pierwszy bit `o_status` od prawej jest równy 1,
- **NEG** - wynik jest liczbą ujemną; drugi bit `o_status` od prawej jest równy 1,
- **EVEN** - w wyniku jest parzysta liczba jedynek; trzeci bit `o_status` od prawej jest równy 1,
- **ONES** - wszystkie bity `o_result` ustawione; ostatni bit `o_status` od prawej jest równy 1,

Jeśli wynik jest nieokreślony (flaga **ERROR**) to pozostałe bity nie są ustawiane; warunki pozostałych flag nie są sprawdzane.

5 Przykład użycia modułu



Ilustracja nr. 2: Widok wykonania testbenchu RTL i oryginalnych plików w GTKWave

Na ilustracji nr. 2 przedstawiono wykonanie dwóch operacji: odejmowanie liczb A i B i zmiana bitu w argumencie A oznaczonego indeksem B. Przy pierwszej operacji na początku wynik jest niezdefiniowany i flaga błędu ustawiona na 1; przepełnienie. W kolejnej operacji w argumencie B został zmieniony na indeksie B: B równe jest 3, więc bit nr. 3 (liczony od zera) w A został zmieniony na 0.

6 Lista plików

- exe_unit_w1.sv - plik zawierający ALU,
- otherModules.sv - plik zawierający wszystkie podmoduły ALU,
- exe_unit_w1_rtl.sv - plik ALU po syntezie,
- synth.log - raport Yosysa po syntezie.

7 Raport z syntezy logicznej

```
$ gcc lab3.c -o wielomian.out wielomiany.c -lm -g
```

8 Raport z symulacji układów

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante

lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.