

Systemy cyfrowe i komputerowe

-

Projekt indywidualny - sync_arith_unit_4

Adam Szajgin s319821

Politechnika Warszawska

9 grudnia 2023

1. Opis modułu

1. Moduł 'sync_arith_unit_4' reprezentuje symulacyjną jednostkę arytmetyczno-logiczną (ALU), synchronizowaną z sygnałem zegarowym. Jednostka zdolna jest do wykonania czterech różnych operacji na dwóch wektorach wejściowych A i B. Zaimplementowane ALU działa w kodzie U2, natomiast zmiany zachodzące w układzie następują na zboczu narastającym zegara.

2. Porty układu:

- i_op – n-bitowe wejście określające kod operacji,
- i_arg_A – m-bitowe wejście wektora binarnego A,
- i_arg_B – m-bitowe wejście wektora binarnego B,
- i_clk – wejście sygnału zegarowego,
- i_reset – wejście resetu synchronicznego wyzwalanego stanem niskim,
- o_result – wyjście synchroniczne zwracające wynik operacji na wektorach wejściowych,
- o_status – 4-bitowe wyjście synchroniczne dostarczające informacje o statusie wyniku operacji.

bity o_status odpowiadają:

- **bit ERROR** - sygnalizacja o tym, iż wynik został określony niepoprawnie,
- **bit EVEN_1** - sygnalizuje parzystą liczbę jedynek w wyniku, Bit ten ma być ustawiany na 0 zawsze, gdy jest sygnalizowany błąd operacji,
- **bit ONES** - sygnalizuje, że wszystkie bity wyniku ustawione są na 1, Bit ten ma być ustawiony na 0 zawsze, gdy jest sygnalizowany błąd,
- **bit OVERFLOW** - bit ten sygnalizuje, że nastąpiło przepełnienie i wynik operacji wykracza poza szerokość wektora wejściowego

3. Parametry układu

By zachować elastyczność i uniwersalność kodu, do realizacji układu użyte zostały parametry:

- N = 2, do określenia wielkości wejścia określającego kod operacji
- M = 4, do określenia wielkości argumentów A oraz B

4. Uwagi do opisu modułu

2. Lista realizowanych operacji

Realizowane operacje:

Na cztery operacje wykonywalne przez jednostkę 'sync_arith_unit_4' składają się:

- $A - 2 * B$

Odejmowanie dwukrotności liczby B od A

- $A < B$

Sprawdzenie czy liczba A jest mniejsza od liczby B. Gdy warunek jest spełniony, układ ma wystawić na wyjściu liczbę większą od zera, w przeciwnym wypadku ma to być liczba 0.

- $(A + B)[B] = 0$

Wynikiem operacji ma być liczba będąca sumą A i B z bitem o index B ustawionym na wartość 0. Jeżeli liczba B jest mniejsza od zera lub większa od szerokości wektora A, układ ma zgłaszać błąd.

- $U2(A) \Rightarrow ZM(A)$

Zamiana liczby A zapisanej w kodzie U2 na zapis w kodzie ZM. Jeżeli nie można dokonać poprawnej konwersji, należy zgłosić błąd, a wyjście układu powinno pozostać nieokreślone.

3. Schemat blokowy realizowanego modułu

4. Specyfikacja kodów operacji oraz flag statusu

5. Synteza logiczna

6. Przeprowadzone symulacje - testy

1. Operacja pierwsza $A - 2 * B$

2. Operacja druga $A < B$

W ramach testów tej operacji użyłem poniższych zestawień:

- $A = 3, B = 5$

Oczekiwany wynik: 1

Oczekiwany status: 0000

- $A = 7, B = 4$

Oczekiwany wynik: 0

Oczekiwany status: 0100

- $A = -4, B = 3$

Oczekiwany wynik: 1

Oczekiwany status: 0000

- $A = -3, B = -3$

Oczekiwany wynik: 0

Oczekiwany status: 0100

- A = 4, B = -5

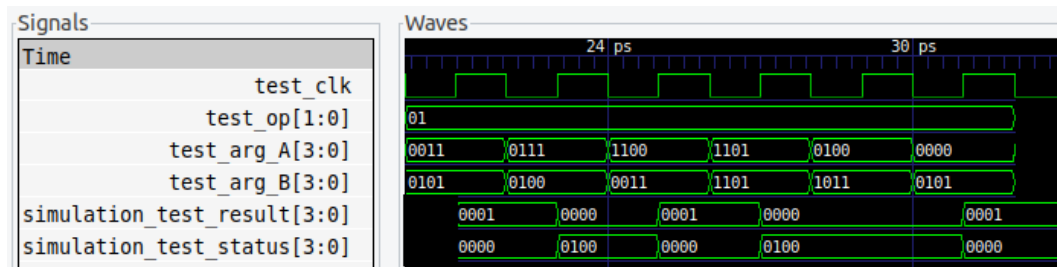
Oczekiwany wynik: 0

Oczekiwany status: 0100

- A = 0, B = 5

Oczekiwany wynik: 1

Oczekiwany status: 0000



Rys. 1. Wynik symulacji operacji nr. 2

Jak widać, dla każdej z testowanych operacji, wynik pokrywa się z oczekiwanym. Zarówno o_result jak i o_status.

3. Operacja trzecia $(A + B)[B] = 0$

4. Operacja czwarta $U2(A) \Rightarrow ZM(A)$

- A = 3 (U2: 0011)

Oczekiwany wynik: 0011

Oczekiwany status: 0100

- A = 7 (U2: 0111)

Oczekiwany wynik: 0111

Oczekiwany status: 0000

- A = 0 (U2: 0000)

Oczekiwany wynik: 0000

Oczekiwany status: 0100

- A = -5 (U2: 1011)

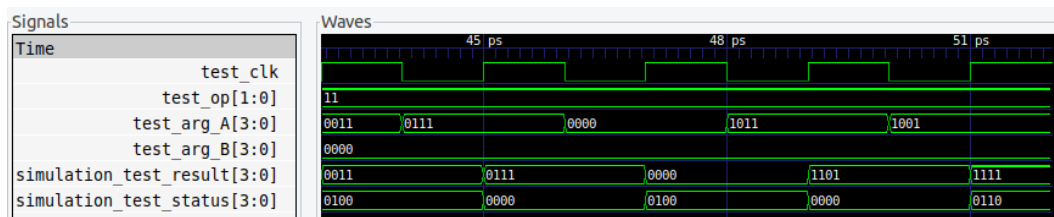
Oczekiwany wynik: 1101

Oczekiwany status: 0000

- A = -7 (U2: 1001)

Oczekiwany wynik: 1111

Oczekiwany status: 0110



Rys. 2. Wynik symulacji operacji nr. 4

Jak widać, dla każdej z testowanych operacji, wynik pokrywa się z oczekiwanym. Zarówno o_result jak i o_status.