Wrocław
University
of Science
and Technology

HR EXCELLENCE IN RESEARCH

Weryfikacja systemów cyfrowych

Projekt

Temat: Jednostka arytmetyczno-logiczna

Przygotował:
Łukasz Sztuka 243168
10.05.2021 r.

Budowa testowanego układu

Rodzaj złącza	Nazwa	Rodzaj sygnału	Opis
Wejście	u_a	std_logic_vector(7 downto 0)	Dane wejściowe, słowo a
	u_b	std_logic_vector(7 downto 0)	Dane wejściowe, słowo b
	clk	std_logic	Wejście zegara
	op	std_logic_vector(3 downto 0)	Wybór operacji
	nreset	std_logic	Reset
Wyjście	borrow	std_logic	Wskaźnik pomocniczy (odejmowanie)
	u_result	std_logic_vector(15 downto 0)	Wyjście, wynik operacji

Wybór operacji

Wartość na wejściu op	Typ operacji
0000	And
0001	Or
0010	Xor
0011	Not
0100	Porównanie a=b
0101	Porównanie a>b
0110	Porównanie a<b
1000	Dodawanie
1001	Odejmowanie
1010	Dzielenie
1011	Mnożenie
inne	Zerowanie słowa wyniku

Funkcje dodatkowe

```
23  function  divide (
24      a : std_logic_vector(7 downto 0);
25      b : std_logic_vector(7 downto 0))
26      return std_logic_vector is
27
28      variable a1 : std_logic_vector(7 downto 0) :=a;
29      variable b1 : std_logic_vector(7 downto 0) :=b;
30      variable p1 : std_logic_vector(8 downto 0):= (others => '0');
31      variable i : integer:=0;
32      begin
33          for i in 0 to 7 loop
34              p1(7 downto 1) := p1(6 downto 0);
35              p1(0) := a1(7);
36              a1(7 downto 1) := a1(6 downto 0);
37                  p1 := p1-b1;
38              if(p1(8) ='1') then --gdy p1<b1 czyli wynik p1-b1 ujemny
39                  a1(0) :='0';
40                  p1 := p1+b1;
41              else
42                  a1(0) :='1';
43              end if;
44          end loop;
45      return a1;
46  end divide;
```

```
49  function  mul (
50      a : std_logic_vector(7 downto 0);
51      b : std_logic_vector(7 downto 0))
52      return std_logic_vector is
53
54      variable a1 : std_logic_vector(7 downto 0) :=a;
55      variable b1 : std_logic_vector(7 downto 0) :=b;
56      variable p1 : std_logic_vector(15 downto 0):= (others => '0');
57      variable i : integer:=0;
58      begin
59          while b1>"0000000" loop
60              p1 := p1+ a1;
61              b1 := b1 - "00000001";
62          end loop;
63      return p1;
64  end mul;
```

Sygnały pomocnicze

Typ	Nazwa	Rodzaj sygnału	Opis
signal	result_temp	std_logic_vector(7 downto 0)	Wynik operacji
	result_temp_reg	std_logic_vector(7 downto 0)	Wynik operacji na potrzeby funkcji dodatkowych
	a	std_logic_vector(7 downto 0)	Parametr funkcji przyjmujący słowo a
	b	std_logic_vector(7 downto 0)	Parametr funkcji przyjmujący słowo b
variable	a1	std_logic_vector(7 downto 0)	Wartość słowa a, wewnątrz funkcji dodatkowych
	b1	std_logic_vector(7 downto 0)	Wartość słowa b, wewnątrz funkcji dodatkowych
	p1	std_logic_vector(15 downto 0)/ std_logic_vector(8 downto 0)	Wartość wyniku, wewnątrz funkcji dodatkowych
	i	integer	Liczba iteracji pętli

Plan testów

1. Układ do poprawnego działania wymaga podłączenia prostokątnego sygnału zegarowego.
2. Następnie należy wybrać rodzaj operacji którą będzie wykonywać układ, dokonujemy tego przy użyciu czterobitowego wejścia „op”.
3. Jednocześnie należy podać wartości słów na których będziemy przeprowadzać operację na ośmiobitowe wejścia u_a, u_b.
4. Obserwuję wynik na szesnastobitowym wyjściu u_result i sprawdzam poprawność operacji.
5. Kroki od 2 do 4 należy powtórzyć dla każdej możliwej operacji arytmetycznej oraz dla różnych wartości słów wejściowych.
6. Na koniec należy sprawdzić czy podanie sygnału na wejście „reset” przebywa pracę.