

Architektura Systemów Komputerowych

Laboratorium 6

Przykład prostego arytmometru

mgr inż. Leszek Ciopiński

Wstęp:

Arytmometr – co to jest?

Arytmometr jest podstawowym układem części operacyjnej procesora. Jest to układ ALU razem z plikiem rejestrów na których może wykonywać operacje.

LATCH:

Układ ten jest rejestrem typu D o charakterystyce wyjść podanej poniżej:

Inputs (wejścia)		Output (wyjście)
ENA	D	Q
L	X	Q ₀ *
H	L	L
H	H	H

* Q₀ = Stan Q zanim ENA przyjęło wartość L.

L – stan niski

H – stan wysoki

X – wartość nieznana (przy wyjściach) lub nieistotna (przy wejściach)

Zadania:

1. Wykorzystując układy LATCH utwórz ośmiobitowy rejestr z ośmiobitowymi szynami wejścia i wyjścia oraz jednobitowym wejściem aktywującym zapis. Następnie wykonaj symulację układu tak, aby wykazać prawidłowe jego działanie. (3 punktów)
2. Utwórz komponent z pliku z zadania 1. Następnie na jego podstawie w nowym pliku zaprojektuj plik czterech rejestrów. Cechy układu:
 - jedno ośmiobitowe wejście danych
 - jedno 2-bitowe wejście adresowe określające miejsce przeznaczenia danych
 - jedno wejście uaktywniające zapis
 - dwa wyjścia ośmiobitowe zwracające wartości zapisane w danych rejestrach
 - dwa 2-bitowe wejścia adresowe określające niezależnie z których rejestrów mają zostać wystawione wartości na obydwie wyjścia danych

Podpowiedź:

Do zbudowania układu wybierającego rejestr do zapisu można użyć bramek AND3 oraz bramek NOT w taki sposób, aby na wejściach bramek AND3 były podawane wartości: wartość wejścia uaktywniającego zapis i wartości z wejścia adresowego – w zależności od numeru rejestru część z tych danych przed wejściem do bramki AND musi zostać poddana negacji.

Do budowy multipleksera dla danych wyjściowych można wielokrotnie użyć multipleksera szyny BUSMUX, który omawiany był na poprzednim laboratorium.

Wykonaj symulację zadania pokazującą poprawność jego wykonania. (5 punktów)

3. Utwórz symbol komponentu ALU napisanego w języku VHDL. W tym celu z menu File | New wybierz opcję VHDL File lub Text Editor i do nowego pliku wklej następujący kod:

```
library IEEE;                                -- dołączenie standardowej
    biblioteki IEEE
use IEEE.STD_LOGIC_1164.all;                 -- dołączenie pakietu
    'STD_LOGIC_1164' z biblioteki IEEE
use IEEE.STD_LOGIC_UNSIGNED.all;            -- dołączenie
    biblioteki operacji arytmetycznych na liczbach bez znaku
    -- definiującego popularne stany
    logiczne
entity alu is                                 -- opis interfejsu ALU
    port(
        BB   : in   STD_LOGIC_VECTOR(7 downto 0);
        BC   : in   STD_LOGIC_VECTOR(7 downto 0);
        BA   : out  STD_LOGIC_VECTOR(7 downto 0);
        Salu : in   integer range 0 to 7
    );
end alu;

architecture alu_arch of alu is              -- opis wnętrza ALU (w tym
    przypadku
begin                                         -- jest to opis strukturalny)

with Salu select
    BA <=
        BB + BC when 0,
        BB - BC when 1,
        BB(3 downto 0) * BC(3 downto 0) when 2,
        BB when 3,
        BB or BC when 4,
```

```
BB and BC when 5,  
not BB when 6,  
"00000000" when others;
```

```
end alu_arch;
```

Następnie zapisz plik jako alu.vhd. UWAGA! Zapisanie pliku pod inną nazwą zakończy się błędem kompilacji! Następnie proszę wybrać plik jako główny element projektu i wykonać kompilację. W przypadku prawidłowego jej ukończenia proszę utworzyć komponent.

(3 punkty)

4. Utwórz schemat arytmometru, w którym połączysz utworzony plik rejestru i układ ALU. Układ zapisz i skompiluj. Wykonaj test tego układu w taki sposób, aby wykazać poprawność jego implementacji.*(4 punkty)*