

Architektury Systemów Komputerowych

Pracownia 2: "Komponenty mikroprocesora"

Pod następującym [adresem](#) znajdują się slajdy po polsku do wykładu o Verilog HDL.

UWAGA! Do rozwiązywania zadań należy użyć dostarczonego środowiska Quartus Altera 13.0 SP1. Dodatkowo można oddawać: zadanie 1 (P0), zadania 5-10 (P1), ale wyłącznie w środowisku Quartus. Należy się upewnić, że synteza (*RTL Viewer*) daje układ o małym stopniu złożoności. Do rozwiązań należy dostarczyć testy wykazujące poprawność implementacji!

Przy tworzeniu projektu w środowisku *Quartus* należy wybierać model układu EP4C115F29C7N (układ zastosowany w zestawie *terasIC DE2-115*). Należy pamiętać o ustawieniu formatu *SystemVerilog* dla *ModelSim* (*Assignments* → *Settings* → *EDA Tools Settings*).

Zadanie 1

Napisz moduł pamięci ROM o 2^N słowach 32-bitowych. Będziemy w niej przechowywać instrukcje procesora. Wejście $a_{N-1:0}$ będzie wybierało instrukcję, której bity będą się pojawiały na wyjściu $rd_{31:0}$ na następnym zboczach zegara clk . Zawartość pamięci wczytaj z pliku binarnego.

Zadanie 2 [2pkt]

Posiłkując się przykładem 5.7 z książki napisz moduł asynchronicznej pamięci RAM¹ o 2^N słowach 32-bitowych. Ponieważ będziemy chcieli przyłączać pamięć do szyny procesora wraz z innymi urządzeniami będą potrzebne dodatkowe sygnały kontrolne. Twój moduł ma mieć szynę adresową $a_{N-1:0}$, wejście aktywacji układu CE (ang. *chip enable*), odczytu OE (ang. *output enable*) i zapisu WE (ang. *write enable*) oraz dwukierunkową szynę danych $io_{31:0}$ (nominalnie w stanie wysokiej impedancji). Operacje przebiegają następująco (o ile CE jest podniesione):

- **odczyt:** podajemy adres, podnosimy sygnał OE , na szynie $io_{31:0}$ pojawiają się dane,
- **zapis:** podajemy adres, podajemy dane na szynę $io_{31:0}$, podnosimy sygnał WE .

Zadanie 3

Zaimplementuj moduł licznika instrukcji (ang. *program counter*), który ma być aktywowany rosnącym zboczem zegara clk . Gdy wejście set jest aktywne², licznik ma ustalać swoją wartość na podstawie 30-bitowego wejścia $newPC_{29:0}$, w przeciwnym wypadku ma zwiększać wartość o 4. Adres instrukcji ma pojawiać się na wyjściu $PC_{29:0}$. Zauważ, że do adresu instrukcji będziemy doklejać 2 zera na najmniej znaczących pozycjach, gdyż wszystkie instrukcje procesora będą 32-bitowej długości.

Zadanie 4

Na podstawie przykładu 5.6 z książki napisz rejestr przesuwany do odbierania przez łącze szeregowe bajtów z bitem parzystości. Twój układ ma mieć wejścia s_{in} , $reset$, clk oraz wyjścia $q_{7:0}$, $ready$ i $error$. Po wczytaniu 9-bitów układ ma zgłosić gotowość poprzez zapalenie sygnału $ready$, a na wyjściu ma się pojawić pierwsze 8 bitów wczytanego słowa. Jeśli liczba bitów w całym 9-bitowym słowie jest parzysta to $error$ ma być wyzerowany. Póki układ nie zostanie zresetowany nie będzie wczytywał kolejnych słów.

¹ <http://www.issi.com/WWW/pdf/61WV102416ALL.pdf>

² W ten sposób będziemy obsługiwać skoki – warunkowe, bezwarunkowe i do podprocedur.

Krystian Bactawski