Systemy wbudowane'18

Lista po-zaliczeniowa nr 4

Terminowe oddanie poprzednich 6 list (oznaczonych „na zaliczenie") gwarantuje Ci ocenę 3.0 z laboratoriów. Chcąc ją podnieść, musisz samodzielnie wykonać dalsze zadania. Będzie ich 4, łączna liczba punktów za ich zrobienie to (4+6+8+10=28). Ostateczna ocena z laboratoriów będzie wyznaczona zgodnie z tabelą:

punkty	< 4	4-8	9-15	16-22	23 –
ocena	3.0	3.5	4.0	4.5	5.0

Ocenę 5.5 prowadzący przyznaje w wyjątkowych wypadkach.

Zadania będą pojawiać się do 22 kwietnia i oddawać je można we wskazanych terminach. Ich poziom trudności będzie rósł, będą też miały odrębną tematykę.

Zadanie 4. **10 punktów. Termin oddania: do końca semestru.** *Wspomnień czar... AKiSO.*

Napisz kod implementujący działanie procesora MARIE z podstawowym zestawem instrukcji (http://samples.jbpub.com/9781449600068/00068_CH04_Null3e.pdf, tabela 4.2), zmodyfikowanym tak, że słowa są 9 bitowe, 4 starsze bity stanowią opkod a 5 pozostałych – adres.

System powinien składać się z następujących elementów:

Pamięć RAM – przechowuje kod programu i dane. Powinna być możliwość wczytania programu z pliku do pamięci (optymalnie), lub wpisania programu na stałe do kodu VHDL (słabsze rozwiązanie).

PC – licznik instrukcji. Przy starcie ustawiany na 0, przechowuje adres kolejnej instrukcję do pobrania z pamięci, jest zwiększany po każdym pobraniu, lub w przypadku instrukcji `skipcond` lub `jump`.

Rejestry – przechowują wyniki operacji, dane pobierane/wysyłane z/do RAM oraz rejestr INOUT do komunikacji ze światem zewnętrznym.

Kontroler – pobiera z RAM instrukcję wskazywaną przez PC. Jest to główny element całego układu. Z pobranego słowa instrukcji wyodrębnia opkod i adres. Na podstawie opkodu ustawia linie sterujące pozostałymi komponentami oraz adresuje pamięć w celu pobrania/zapisania operandu/wyniku.

ALU – układ zdolny wykonywać podstawowe operacje arytmetyczne `ADD, SUBT` na dwóch operandach pochodzących z akumulatora i rejestru danych, umieszcza wynik w akumulatorze.

Poniższe uwagi mogą pomóc w zaplanowaniu poszczególnych komponentów:

RAM koncepcyjnie to pamięć podobna do ROM z listy 4 z tym, że musi umożliwiać również zapis. Potrzebny jest zatem dodatkowy sygnał, mówiący czy pod podany na wejściu adres należy zapisać daną, czy ją stamtąd odczytać.

PC jest licznikiem (lista 3) z tym, że nie będzie taktowany systemowym zegarem, tylko impulsami pochodzącymi z kontrolera

Kontroler jest układem realizującym maszynę stanową (lista 5), w najwyższym stopniu abstrakcji wykonującą przejścia między stanami FETCH (pobierz instrukcję), DECODE (sprawdź, co to za instrukcja), EXECUTE (wysteruj sygnały dla ALU, rejestrów, pamięci tak, aby dana instrukcja została wykonana), STORE (zapisz wynik do RAM). Oczywiście np. wykonanie instrukcji `JUMP` nie ma fazy STORE.

ALU ponieważ wykonuje tylko dwie operacje jest dość prostym układem kombinacyjnym. Z drugiej strony pamiętaj, że wysłanie kodu operacji do ALU (dodaj/odejmij) musi być połączone z wysterowaniem akumulatora tak, by zapisał wynik, który dostanie z wyjścia ALU – to zadanie kontrolera.

Magistrala łączy wszystkie komponenty, zatem aby uniknąć problemów ze sterowaniem jej z wielu źródeł musisz rozważyć trzywartościową logikę (lista 8).

Ogólny diagram rozważanej architektury znajdziesz w wyżej linkowanym dokumencie (w szczególności zob. obrazek 4.9). Staraj się pisać kod tak, żeby „zawsze było coś działającego do pokazania". Oddając swój kod zaproponuj program, który Twój procesor może wykonać (zapisany w pliku tekstowym, lub wpisany na stałe do pamiędzi ROM/RAM).

Powodzenia!