Logika cyfrowa

Lista zadań nr 13

Termin: 7 czerwca 2020

1. Rozważmy następującą instrukcję:

   `add rd, rs1, rs2`

   a) Jakie sygnały sterujące wygeneruje **jednocyklowa** ścieżka sterowania dla tej instrukcji?

   b) Które elementy ścieżki danych wykonują użyteczną pracę dla tej instrukcji?

   c) Które elementy ścieżki danych produkują wynik, który nie jest używany?

2. Rozważmy program, w którym poszczególne rodzaje instrukcji występują z następującą częstością:

OP	typ I (poza LOAD)	LOAD	STORE	BRANCH	JAL	typ U
24%	20%	25%	10%	11%	2%	8%

   a) Jaki odsetek instrukcji używa pamięci danych?

   b) Jaki odsetek instrukcji używa pamięci instrukcji?

   c) Jaki odsetek instrukcji używa ALU?

   d) Jaki odsetek instrukcji używa generatora stałych?

   e) Co robi generator stałych w cyklach, w których nie jest potrzebny?

3. Częstą usterką pojawiającą się przy produkcji układów krzemowych są przerwane połączenia pomiędzy elementami. W efekcie elementy podłączone do takiego uszkodzonego połączenia typowo rejestrują ciągle wartość 0.

   W przypadku wystąpienia takiej usterki, które instrukcje nie będą działać poprawnie, jeśli uszkodzonym połączeniem jest:

   a) `alu_bsel`

   b) `rd_we`

   c) `alu_zero`

4. Załóżmy, że wymienione poniżej elementy mają następujące opóźnienia:

pamięć kodu i danych	plik rejestrów	multiplekser	ALU	sumator	odczyt PC	czas ustalania rejestru	generator stałych	ścieżka sterowania
250 ps	150 ps	25 ps	200 ps	150 ps	30 ps	20 ps	50 ps	50 ps

   „Odczyt PC” to czas od zbocza narastającego zegara do pojawienia się nowej wartości wskaźnika instrukcji. Czas ustalania dotyczy zarówno wskaźnika instrukcji, jak i pliku rejestrów. Czasy podane dla elementów kombinacyjnych (np. ALU) oznaczają ścieżkę krytyczną.

   a) Jakie jest opóźnienie instrukcji OP? (To znaczy, jak długi musi być okres zegara, aby instrukcje OP wykonywały się prawidłowo.)

   b) Jakie jest opóźnienie instrukcji LOAD?

   c) Jakie jest opóźnienie instrukcji STORE?

   d) Jakie jest opóźnienie instrukcji BRANCH?

5. Rozważmy możliwość dodania do RISC V następującej instrukcji:

   `lwiw rd, rs1, rs2`

   Nazwa instrukcji rozwija się do *load with increment, word*. Jej działanie ma polegać na załadowaniu do rejestru `rd` wartości słowa 32-bitowego z pamięci danych spod adresu `rs1` + `rs2`.

   a) Jakie nowe elementy ścieżki danych musimy dodać (jeśli trzeba)?
   b) Jakie istniejące elementy ścieżki danych musimy zmodyfikować (jeśli trzeba)?
   c) Jakie nowe połączenia musimy dodać w ścieżce danych (jeśli trzeba)?
   d) Jakie nowe sygnały sterujące należy dodać (jeśli trzeba)?

6. Rozważmy możliwość dodania do RISC V następującej instrukcji:

   `swap rs1, rs2`

   Działanie instrukcji ma polegać na zamianie miejscami wartości rejestrów `rs1` i `rs2`.

   Odpowiedz na pytania z podpunktów zadania 5.

7. Rozważmy możliwość dodania do RISC V następującej instrukcji:

   `lwinc rd, imm(rs1)`

   Nazwa instrukcji rozwija się do *load word, increment*. Jej działanie ma polegać na załadowaniu do rejestru `rd` wartości słowa 32-bitowego z pamięci danych spod adresu `rs1` + `imm` (tak jak zwykłe `lw`) i zwiększeniu wartości `rs1` o 4. Innymi słowy, instrukcja ma działać tak jak następujący ciąg instrukcji:

   ```
   lw rd, imm(rs)
   addi rs, rs, 4
   ```

   Odpowiedz na pytania z podpunktów zadania 5.