Architektury systemów komputerowych

Lista 7

$x_7 = 6$ (minimum na bdb)

1. Rozważmy procesor jednocyklowy ze slajdu 16, wykład 7B. Załóżmy, że czasy działania bloków I-Mem, Add, Mux, ALU, Regs, D-mem i Control (jednostki sterującej) wynoszą odpowiednio 400ps, 100ps, 30ps, 120ps, 200ps, 350ps i 100ps., a ich ceny 1000, 30, 10, 100, 200, 2000 i 500 (pomijamy czasy działania i koszty układów Sign Extend, Shift Left oraz ALU control). Rozważmy teraz dwa usprawnienia procesora:

   - zastąpienie układów Add szybszymi (przyspieszenie o 20ps, wzrost ceny o 20)
   - użycie większego pliku rejestrów Regs (zmodyfikowany układ Regs działa o 100ps wolniej, cena wzrasta o 200, ale zmniejsza się liczba odwołań do pamięci, co zmniejsza całkowitą liczbę wykonywanych rozkazów o 5 procent).

   Na poniższe pytania odpowiedz rozważając warianty, w których obie modyfikacje wprowadzane są jednocześnie oraz każda z osobna.

   (a) Jaka jest długość cyklu zegarowego bez oraz z powyższymi modyfikacjami?

   (b) Jakie przyspieszenie uzyskuje się dzięki wprowadzonym modyfikacjom?

   (c) Porównaj stosunek ceny do efektywności procesora bez modyfikacji i z modyfikacjami.

2. Rozważmy następujące rozkazy: `lw $1, 40 ($6)` oraz `Label: bne $1, $2, Label`.

   (a) Jak zakodowane są powyższe rozkazy w kodzie maszynowym?

   (b) Ponownie rozważamy procesor ze slajdu 16, wykład 7B. Jakie wartości zostaną podane na wejścia *Read register 1, Read register 2 oraz Write register* podczas wykonywania powyższych rozkazów?

3. Które z rozkazów rozważanych na wykładzie 7B będą działały poprawnie na procesorze jednocyklowym ze slajdu 16, wykład 7B, jeśli pojawi się błąd ustawiający na stałe wartość jednego z sygnałów sterujących. Rozważamy następujące sytuacje:

   (a) `RegWrite=1` (`RegWrite` to sygnał zezwolenia na zapis rejestru)

   (b) `Branch=1` (`Branch` to informacja o tym, czy rozkaz jest skokiem warunkowym)

   (c) `MemRead=1` (`MemRead` to informacja o tym, że pamięć będzie odczytywana)

   (d) `MemWrite=1` (`MemWrite` to sygnał zezwolenia na zapis do pamięci)

   (e) `RegWrite=0`

   (f) `Branch=0`

   (g) `MemRead=0`

   (h) `MemWrite=0`

4. Zaprojektuj na poziomie bramek logicznych układ ALU Control ze slajdu 16, wykład 7B. Przyjmij, że sygnały odpowiednich rozkazów dla ALU są następujące: AND – 0000, OR – 0001, add – 0010, sub – 0110, set on less then – 0111, NOR – 1100. Zakłamy, że mamy listę rozkazów ze slajdu 2, wykład 7B. Wartości dwubitowego sygnału ALUop są następujące: 00 – wykryto rozkaz `lw`/`sw`, 01 – wykryto rozkaz `beq`, 10 – wykryto rozkaz arytmetyczno-logiczny/`slt`.

5. Rozważmy dodanie rozkazu `jr` (skocz pod adres zapisany w rejestrze o numerze podanym w argumencie rozkazu) do naszej implemntacji jednocyklowej. Opisz konieczne modyfikacje i rozszerzenia układu, dorysuj na rysunku potrzebne dodaktowe połączenia i układy.

6. Rozszerz procesor jednocyklowy o rozkaz `lui` (załaduj stałą 16-bitową do dwóch górnych bajtów wskazanego rejestru). Opisz konieczne modyfikacje i rozszerzenia oraz zaznacz je na rysunku.

7. Rozważmy dwie nowe instrukcje: `bezi (Rs), Label` oraz `swi Rd, Rs(Rt)`. Pierwsza z nich ma działać następująco: jeśli MEM[Rs]=0, to PC:=PC+Offs (przez MEM[RS] oznaczamy słowo pamięci pod adresem z rejestru `Rs`). Druga: Mem[Rs+Rt]:=Rd.

   (a) Podaj przykład kodu, w którym użycie nowych rozkazów jest użyteczne.

   (b) Przetłumacz nowe rozkazy na ciągi starych.

   (c) Zaproponuj maszynowe kodowanie nowych rozkazów.

8. Opisz zmiany w procesorze jednocyklowym, które pozwolą rozszerzyć listę rozkazów o `bezi`. (Jakie nowe układy i sygnały sterujące są potrzebne? Jak zmodyfikować stare?)

9. Opisz zmiany w procesorze jednocyklowym, które pozwolą rozszerzyć listę rozkazów o `swi`. (Jakie nowe układy i sygnały sterujące są potrzebne? Jak zmodyfikować stare? W szczególności: jak zmodyfikować plik rejestrów?)

10. Rozważmy instrukcję `bcmp`, która porównuje dwa bloki pamięci. Zakładamy, że adres pierwszego bloku podany jest w rejestrze `$t1`, drugiego w `$t2`, a długość bloków (w słowach, liczba naturalna) w `$t3`. Rozkaz zapiuje wynik w `$t1` – wynikiem jest adres pierwszej znalezionej różnicy lub zero, jeśli bloki są identyczne. Zakładamy również, że podczas wykonania rozkazu mogą być modyfikowane rejestry `$t1-$t5`.

    Napisz program w asemblerze MIPS emulujący opisany powyżej rozkaz. Zauważ, że dodanie rozkazu `bcmp` do naszego procesora jednocyklowego byłoby bardzo trudne.