

Architektury systemów komputerowych

Egzamin

3.II.2011

Rozwiązanie każdego zadania proszę umieścić na osobnej kartce. Do zdobycia jest 100 punktów. Przewidywana skala ocen: 0-41 ndst, 42-51 - dst, 52-61 +dst, 62-71 db, 72-81 +db, 82- bdb. Czas: 120 min.

1. **[16 = 5 + 5 + 6]** Rozważmy funkcję boolowską f , czterech zmiennych, zwracającą 1, dokładnie wtedy, gdy na wejściu jest zakodowana binarnie jedna z liczb 0, 2, 3, 6, 7, 11, 15:

$$f = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c\bar{d} + \bar{a}b\bar{c}d + \bar{a}bcd + a\bar{b}c\bar{d} + a\bar{b}cd + abcd.$$

- (a) Zminimalizuj f do postaci sumy iloczynów, używając metody siatek Karnaugh'a.
 - (b) Zminimalizuj f do postaci iloczynu sum.
 - (c) Zbuduj układ realizujący funkcję f , używając tylko bramek NAND (2-, 3- lub 4-wejściowych).
2. **[15 = 5 + 5 + 5]**

- (a) Jakie liczby reprezentowane są w ośmiobitowej reprezentacji uzupełnień do 2 jako

11101010 oraz 00101011 ?

- (b) Jakie liczby reprezentowane są w formacie pojedynczej precyzji IEEE 754 jako

0011 1111 1110 0000 0000 0000 0000 0000 oraz 1100 0000 1000 0000 0000 0000 0000 0000 ?

Przypominam, że wykładnik ma w tym standardzie 8, a mantysa 23 bity. Wykładnik jest pamiętany z przesunięciem o 127; mantysa jest znormalizowana, a ukryta jedynka jest przed przecinkiem.

- (c) Jaki będzie wynik mnożenia liczb z punktu (b)? Wynik przedstaw również w formacie pojedynczej precyzji IEEE 754.

3. **[16 = 7 + 9]**

- (a) Przedstaw realizację przerzutnika S-R za pomocą bramek NOR, w wersji sterowanej poziomem sygnału zegarowego. Przypominam, że przerzutnik S-R podtrzymuje swój stan przy układzie wejść $S = R = 0$, ustawia się na 1, gdy $S = 1, R = 0$ oraz ustawia się na 0, gdy $S = 0, R = 1$. Układ wejść $S = R = 1$ jest zabroniony.
- (b) Na poniższym wykresie przedstawione są zmiany sygnału zegarowego oraz wejść przerzutnika S-R. Narysuj wykres zmian wyjścia przerzutnika w wersji prostej, bez sygnału zegarowego (Q), w wersji sterowanej poziomem sygnału zegarowego (Q') oraz w wersji sterowanej zboczem opadającym sygnału zegarowego, (Q''). Początkowo, przerzutnik jest w stanie 0.

4. **[16 = 8 + 8]** Rozważmy następujące przypisanie w języku wysokiego poziomu:

$$d := (b+3) * c + a * (b+c)$$

- (a) Zapisz program w asemblerze MIPS, realizujący powyższe przypisanie. Możesz użyć pseudoinstrukcji `mul rd, rs, rt`. Zakładamy, że zmienne a, b, c, d są przechowywane odpowiednio w rejestrach `$s0, $s1, $s2, $s3`. Rejestry te (oczywiście oprócz `$s3`) nie mogą być zmienione przez program. Program może jednak używać rejestrów pomocniczych `$t0 - $t3` (i modyfikować ich początkową zawartość).
- (b) Rozważmy teraz maszynę dysponującą czterema rozkazami dwuargumentowymi: `add, addi, mul, move`.
- Rozkaz `add rd, rs` dodaje rejestry `rd` i `rs`, a wynik zapisuje w `rd`.
 - Rozkaz `mul rd, rs` mnoży rejestr `rd` przez `rs`, a wynik zapisuje w `rd`.
 - Rozkaz `addi rd, const` dodaje stałą `const` do rejestru `rd`.
 - Rozkaz `move rd, rs` kopiuje zawartość rejestru `rs` do rejestru `rd`.

Napisz program wyliczający wartość rozważanego wyrażenia przy użyciu omówionych rozkazów. Założenia o rejestrach są takie jak w podpunkcie (a).

5. **[15 = 5 + 5 + 5]** W tym zadaniu oczekuję krótkich, maksymalnie kilkudziesięciu odpowiedzi.

- (a) System pamięci cache jest efektywny, gdyż wykonywane programy zazwyczaj spełniają zasadę lokalności czasowej (*temporal locality*) i miejscowej (*spatial locality*) odwołań. Wyjaśnij na czym polegają obydwie zasady. Podaj naturalne przykłady lokalności czasowej i miejscowej.
- (b) Wymień co najmniej dwie korzyści płynące ze stosowania mechanizmu pamięci wirtualnej.
- (c) Na czym polega mechanizm DMA?

6. **[22 = 9 + 3 + 3 + 3 + 4]** W tym zadaniu projektować będziemy procesor realizujący listę rozkazów z zadania 4(b). Zakładamy, że procesor będzie dysponował plikiem ośmiu, 16-bitowych rejestrów ogólnego przeznaczenia. Mamy tylko pamięć rozkazów (programy operują jedynie na rejestrach; pamięć jest tylko do odczytu – zakładamy, że dostajemy ją razem z programem), adresowanie jest bajtowe, a adresy mają długość 16 bitów. Każdy rozkaz jest długości jednego bajtu. Wszystkie dane są 16-bitowe. Kodowanie rozkazów jest następujące:

- dwa najbardziej znaczące bity to kod operacji: `add` – 00, `addi` – 01, `mul` – 10, `move` – 11,
- kolejne trzy bity to numer rejestru `rd`,
- trzy najmniej znaczące bity to numer rejestru `rs` lub (trzybitowa) stała w reprezentacji uzupełnień do 2.

Do dyspozycji mamy następujące bloki (patrz również rysunek):

- układ pamięci **MEM**: jedno szesnastobitowe wejście adresowe *Addr*, jedno ośmiobitowe wyjście *MOut*; czas działania: 200 ps,
- plik rejestrów **REG**: dwa trzybitowe wejścia *RReg* (nr rejestru do odczytu), *WReg* (nr rejestru do odczytu/zapisu), dwa 16-bitowe wyjścia *ROut1* – odczytany rejestr *RReg*, *ROut2* – odczytany rejestr *WReg*, wejście zezwolenia na zapis *Enb*; czas odczytu: 120 ps, czas zapisu: 120 ps,
- jednostka arytmetyczno-logiczna **ALU**: dwa szesnastobitowe wejścia danych *Data1*, *Data2*, jedno szesnastobitowe wyjście *AOut*, jedno dwubitowe wejście operacji *Op* (00 – dodaj, 11 – pomnóż, 01 – odejmij); czas działania: 220 ps.
- Szesnastobitowy rejestr licznika rozkazów, **PC**, z wejściem zezwolenia na zapis *WEnb*

Możesz użyć też dodatkowych układów realizujących pewne proste czynności (np. dodawanie stałej do PC) – wyspecyfikuj ich działanie. Przydadzą się też na pewno multipleksery, może coś jeszcze... W podpunktach dotyczących wyliczania czasu działania możesz pominąć czas działania wszystkich dodatkowych układów.

- Przedstaw schematyczny rysunek procesora w wersji jednocyklowej. Zaznacz dokładnie, gdzie kierowane są poszczególne bity. Opisz precyzyjnie działanie jednostki sterującej (jakie sygnały sterujące są potrzebne, jak są generowane).
- Jaka powinna być optymalna długość cyklu zegarowego dla wersji jednocyklowej?
- Ile etapów potoku powinien mieć prosty procesor potokowy? Jakie to etapy? Jaka powinna być optymalna długość cyklu zegarowego?
- Z jakiego rodzaju hazardami możemy mieć do czynienia w naszym procesorze potokowym? Podaj przykład kodu powodującego hazardy.
- Opisz prosty i efektywny sposób eliminowania hazardów z poprzedniego podpunktu (nie musi być to tak dokładny opis jak w podpunkcie (a) – wystarczy krótkie objaśnienie ogólnej idei).