

# Systemy komputerowe

Lista zadań nr 7

Na ćwiczenia 21. i 27. marca 2022

wersja 1

Każde zadanie warte jest 1 punkt.

**Zadanie 1.** Przypomnij definicje hazardu sterowania, strukturalnego, RAW, WAR, WAW. Dla każdego z procesorów a) jednocyklowego b) potokowego z pojedynczym ALU c) potokowego z wieloma jednostkami wykonawczymi podaj wszystkie rodzaje hazardów, jakie mogą w nim wystąpić podczas wykonywania programu. Pokaż fragmenty kodu generujące odpowiednie hazardy, lub uzasadnij, że nie mogą one zachodzić.

**Zadanie 2.** Przypomnij algorytm przetwarzania potokowego przy użyciu *scoreboardu*<sup>1</sup>. W jaki sposób ten algorytm radzi sobie z hazardami RAW, WAR i WAW?

**Zadanie 3.** Przypomnij algorytm przetwarzania potokowego przy użyciu *reorder buffera*. W jaki sposób ten algorytm radzi sobie z hazardami RAW, WAR i WAW? Porównaj jego wydajność z algorytmem używającym *scoreboardu* dla procesora o takich samych parametrach (takie same fazy przetwarzania, liczba i budowa jednostek wykonawczych).

**Zadanie 4.** Dany jest program

```
r3 = r1 * r2
r5 = r4 + r3
r6 = r4 + r1
r7 = r8 * r9
r4 = r3 + r7
r10 = r5 * r6
```

Podaj diagram wykonania i wylicz liczbę cykli zużytych przez ten program uruchomiony na następujących procesorach

---

<sup>1</sup> We wszystkich zadaniach dotyczących tej techniki przetwarzania potokowego załóż, że bufor pomiędzy fazami pobrania (ang. *fetch*) a zlecenia (ang. *issue*) ma rozmiar jednej instrukcji.

- a) procesorze potokowym z pojedynczym ALU. To ALU samo jest w pełni potokowe i działa w 6 cyklach. Każda instrukcja wykonuje się więc w fazach<sup>2</sup> F D E1 E2 E3 E4 E5 E6 WB.
- b) procesorze potokowym z w pełni potokowymi niezależnymi jednostkami (po jednej sztuce) dla mnożenia (6 cykli) i dodawania (4 cykle). Instrukcja mnożenia wykonuje się więc w fazach F D E1 E2 E3 E4 E5 E6 WB, a dodawania w fazach F D E1 E2 E3 E4 WB. Procesor używa *scoreboardu*.
- c) procesorze potokowym jak wyżej, ale używającym *reorder buffera*.

Jedynie procesor z punktu c) może używać *forwardingu* danych. Możesz zignorować ewentualne hazardy strukturalne, ale zaobserwuj fazy procesora, w którym mogą one wystąpić.

**Zadanie 5.** Powtórz powyższe zadanie dla programu

```

r3 = r1 * r2
r5 = r4 + r3
r7 = r2 + r6
r10 = r8 + r9
r11 = r7 * r10
r5 = r5 + r11

```

W punktach b) i c) załóż tym razem, że jednostki wykonawcze dla dodawania i mnożenia nie są potokowe, tzn. ich czas działania wynosi, jak poprzednio 4 i 6 cykli, ale można zlecić wykonanie kolejnej operacji danego typu, dopiero po zakończeniu poprzedniej.

**Zadanie 6.** Przypomnij algorytm Tomasulo przetwarzania potokowego. W szczególności scharakteryzuj używane w nim struktury danych. W jaki sposób ten algorytm radzi sobie z hazardami RAW, WAR i WAW? Do czego służy technika przemianowywania rejestrów (ang. *register renaming*). Porównaj jego wydajność z algorytmem używającym *reorder buffera* dla procesora o takich samych parametrach.

**Zadanie 7.** TBD

**Zadanie 8.** TBD

**Zadanie 9.** TBD

---

<sup>2</sup> Dla uproszczenia nie ma fazy MEM.