

# Systemy komputerowe

Lista zadań nr 6

Na ćwiczenia 3 i 4 kwietnia 2024

wersja wstępna

Każde zadanie warte jest 1 punkt.

**Zadanie 1.** Pokaż ścieżkę przepływu danych w prostym jednocyklowym procesorze<sup>1</sup> wykonującym instrukcję  $x = *(y + \text{imm})$ , gdzie  $x$  i  $y$  są rejestrami, a  $\text{imm}$  stałą.

**Zadanie 2.** Powtórz powyższe zadanie dla instrukcji  $* (x + \text{imm}) = y$ .

**Zadanie 3.** Powtórz powyższe zadanie dla instrukcji  $x = y \text{ binop } z$ , gdzie  $\text{binop}$  jest binarnym operatorem arytmetycznym.

**Zadanie 4.** Powtórz powyższe zadanie dla instrukcji  $\text{if } x \text{ relop } y \text{ goto } L$ , gdzie  $\text{relop}$  jest binarnym operatorem relacyjnym a  $L$  adresem w pamięci. Instrukcja ta wykonuje skok warunkowy do adresu  $\text{PC} + L$ , gdzie  $\text{PC}$  jest bieżącą wartością licznika rozkazów.

**Zadanie 5.** Pokaż, w jaki sposób rozszerzyć prosty jednocyklowy procesor o możliwość wykonywania dwóch nowych typów instrukcji:

- a)  $x = y \text{ binop } \text{imm}$ , gdzie  $x$  i  $y$  są rejestrami a  $\text{imm}$  stałą, oraz
- b)  $\text{goto } L$ , gdzie  $L$  jest adresem w pamięci. Instrukcja wykona skok do adresu  $L$ .

Pokaż ścieżki przepływu danych dla obydwu tych instrukcji.

**Wskazówka:** Procesor ten widoczny jest na slajdzie 27. Nazwiemy go kompletnym jednocyklowym procesorem.

---

<sup>1</sup> w wariantcie widocznym na slajdzie 17

**Zadanie 6.** Załóżmy, że komponenty kompletnego jednocyklowego procesora mają następujące czasy działania (opóźnienia, ang. *latency*).

Inst./Data. memory	Register file	Mux	ALU	Adder (+)	PC	Sign Extend	Control Unit
250ps	150ps	25ps	200ps	150ps	50ps	50ps	50ps

Wylicz opóźnienia dla instrukcji:

- a)  $x = *(y + \text{imm})$
- b)  $*(x + \text{imm}) = y$
- c)  $x = y \text{ binop } z$
- d) `if x relop y goto L`
- e)  $x = y \text{ binop } \text{imm}$
- f) `goto L`

Jaka jest minimalna długość cyklu dla tego procesora?

**Zadanie 7.** Załóżmy, że umiemy zbudować kompletny jednocyklowy procesor w taki sposób, że długość cyklu może być różna dla różnych instrukcji. Jakie przyspieszenie osiągnie ten procesor w stosunku do procesora z zadania 6. przy założeniu, że wykonujemy program złożony z instrukcji w następujących proporcjach:

$x = *(y + \text{imm})$	$*(x + \text{imm}) = y$	<code>if x relop y goto L</code>	$x = y \text{ binop } z$
25%	11%	12%	52%

**Zadanie 8.** Chcemy rozszerzyć kompletny jednocyklowy procesor o możliwość wykonania instrukcji  $x = *(y + z)$ , gdzie  $x, y, z$  są rejestrami. Czy to rozszerzenie wymaga dodania nowych komponentów do procesora lub modyfikację istniejących? Przedstaw schemat zmodyfikowanego procesora i ścieżkę przepływu danych dla tej instrukcji.

**Zadanie 9.** Powtórz powyższe zadanie dla instrukcji `swap x y`, zamieniającej miejscami zawartości dwóch rejestrów.

**Zadanie 10.** Powtórz powyższe zadanie dla instrukcji  $*x = y + \text{imm}$ , gdzie  $x$  i  $y$  są rejestrami a  $\text{imm}$  stałą.

**Zadanie 11.** Każdą instrukcję postaci  $x = *(y + \text{imm})$  można zastąpić parą instrukcji  $t = y + \text{imm}; x = *t$ , gdzie  $t$  jest rejestrem pomocniczym. Podobnie można zastąpić instrukcję  $*(x + \text{imm}) = y$ , otrzymując program w którym wszystkie instrukcje dostępu do pamięci danych są postaci  $x = *y$  lub  $*x = y$ . Dzięki temu, wykonanie żadnej instrukcji nie wymaga użycia jednocześnie ALU i pamięci danych. Takie uproszczenie formatu instrukcji umożliwi skrócenie długości cyklu procesora, ale równocześnie sprawi, że wykonywanie programy będą musiały być przekompilowane do dłuższego kodu trójkowego. Jaka będzie długość cyklu w tym procesorze (czasy działania komponentów są jak w zad. 6.)? Czy program o charakterystyce podanej w zad. 7. wykona się szybciej czy wolniej na zmodyfikowanym procesorze (załóż, że wszystkie instrukcje dostępu do pamięci danych używają niezerowej stałej  $\text{imm}$ )? Który z procesorów, kompletny jednocyklowy, czy jego modyfikację uznasz ogólnie za lepszy projekt?