

# Systemy komputerowe

## Lista zadań nr 9

Na ćwiczenia 24 i 25 kwietnia 2024

**Zadanie 1.** Przypomnij zasadę działania bufora predykcji skoków (ang. *branch-target buffer*). W jaki sposób jest on używany w prostym 5-fazowym procesorze potokowym? Zdefiniuj 2-bitowy predyktor skoków, opisz jego zasadę działania i pokaż przykład programu, dla którego sprawdza się on lepiej, niż predyktor statyczny. Gdzie zapamiętać bity potrzebne do funkcjonowania predyktora 2-bitowego? W jaki sposób uogólnić ten predyktor do predyktora n-bitowego?

**Wskazówka:** Hennessy, John L., and David A. Patterson. Computer Architecture: A Quantitative Approach **6e**. 2019 (**HP6e**). s. C-23 – C-26.

**Zadanie 2.** Jaka jest motywacja dla predyktorów korelujących (ang. *correlating* lub *hybrid branch predictors*)? Pokaż przykład kodu, dla którego predyktor lokalny nie jest w stanie osiągnąć idealnej dokładności predykcji. Jaka ogólna struktura mają predyktory korelujące? Jak działa predyktor *gshare*?

**Wskazówka:** HP6e, s. 182 – 184.

**Zadanie 3.** Czym są predyktory turniejowe. W jaki sposób można połączyć globalne i lokalne predyktory w jeden mechanizm predykcji skoków?

**Wskazówka:** HP6e, s. 184 – 187.

**Zadanie 4.** Zdefiniuj strukturę predyktora TAGE i podaj jego zasadę działania.

**Wskazówka:** HP6e, s. 188 – 190.

**Zadanie 5.** Wykonaj zadanie 3.17 z HP6e (str. 277).

**Zadanie 6 (2pkt).** Wykonaj zadanie 3.18 z HP6e (str. 278).

**Zadanie 7 (2pkt).** Wykonaj zadanie 3.19 z HP6e (str. 278).

**Wskazówka:** Może się zdarzyć, że bufor predykcji skoków (ang. *branch-target buffer*) nie zawiera wpisu o instrukcji skoku, o którą chcemy go zapytać. W tym przypadku płacimy karę (ang. *buffer miss penalty*). HP6e, s. 228 – 234.