



Rechnerarchitektur: Übungssatz 14

Aufgabe 14.1

Gegeben sei ein RISC-Prozessor mit Harvardarchitektur, 16 Universalregistern und einer vierstufigen Pipeline:

1. IF instruction fetch
2. ID/OF instruction decode, operand fetch
3. EX/LS execute, load/store
4. WB write back

Der Prozessor verfügt über kein weiteres Bypassing (keine Forwarding-Register).

Auf diesem Prozessor soll nun folgendes Maschinenprogramm abgearbeitet werden. Die Besonderheiten der Befehlsabarbeitung sind dabei noch nicht berücksichtigt:

```
...
1.  M1:  SUB R3, R1, R2  (R3 <- R1 - R2)
2.      ADD R4, R6, R3  (R4 <- R6 + R3)
3.      SUB R5, R1, #1  (R5 <- R1 - 1)
4.      ADD R6, R1, R5  (R6 <- R1 + R5)
5.      LDI R6, #2      (R6 <- 2)
...
```

- (a) Eine wieviel-Adressmaschine liegt vor?
- (b) Geben Sie das Pipeline-Taktschema des unmodifizierten Programmes an!
- (c) Identifizieren Sie die darin auftretenden Pipeline-Konflikte!
- (d) Wie können diese Konflikte gelöst werden, wenn der Prozessor diese nicht selbständig erkennt und behebt?
- (e) Welche Ansätze in Hardware entsprechen den Lösungen aus d)?
- (f) Durch welche Hardwaremaßnahme kann die Zahl potentiell von einer Datenabhängigkeit betroffener Befehle reduziert werden?
- (g) Welche Konfliktarten neben Datenhazards gibt es noch bei der Befehlsausführung in Pipelines? Wie können diese in Hardware vermieden oder entschärft werden?