



Lernziele

- Wiederholung: Pipelinekonflikte und ihre Behebungsmaßnahmen
- Speicherhierarchie
- Cache-Speicher
 - Funktionsweise: Lese- und Schreibzugriffe
 - Aufbau eines Cache-Speichers (Folie 13-37)

Aufgabe 1

Das folgende Programmstück soll in der DLX-Pipeline abgearbeitet werden.

```
S1:          add  $t1, $zero, $zero
S2:          lw   $t3, 0x1500($zero)
S3:    loop:  lw   $t4, 0x5000($t1)
S4:          add  $t5, $t4, $t3
S5:          sw   $t5, 0x400($t1)
S6:          addi $t1, $t1, 4
S7:          subi $t2, $t1, 0x400
S8:          bnez $t2, loop
S9:    end:   srli $t1, $t1, 2
S10:         sw   $t1, 0x2000($zero)
```

1. Die einzige Methode zur Behebung von Pipelinekonflikten sei das Einfügen von NOP-Befehlen (*No Operation*) in den Befehlsstrom.
Ergänzen Sie das Programmstück durch das Einfügen von möglichst wenigen NOP-Befehlen, so dass alle Pipelinekonflikte behoben werden.
2. Wie viele NOP-Befehle sind noch notwendig, wenn *result-* und *load-forwarding*-Techniken implementiert sind?
3. Nehmen Sie an, dass die Befehlsdekodierung, Berechnung der Sprungzieladresse und Rückschreibung des Befehlszählers (PC) bei bedingten Sprungbefehlen in der ID/RF-Stufe erfolgt¹. Wie viele NOP-Befehle sind jetzt noch notwendig, wenn

¹U. a. durch Ressourcenreplizierung, ... usw.

- (a) *Result*- und *load-forwarding*-Techniken implementiert sind?
- (b) *Result*- und *load-forwarding*-Techniken nicht implementiert sind?