UNIVERSITÄT SALZBURG

Rechnerarchitektur

SS 2017

Übungszettel 6

18. Ermitteln Sie Art und Anzahl der Stalls aus dem gegebenen Codesegment über ein Taktzyklendiagramm für folgende Fälle:

```
ADDUI   R1, R1, #4
LW      R2, 0(R1)
ADDUI   R8, R0, #2
MULTU   R5, R2, R2
SLLI    R8, R8, #3
XOR     R5, R5, R8
SW      0(R9), R5
SGEUI   R7, R8, #1000
BEQZ    R7, elsewhere
```

   (a) Die DLX Pipeline arbeitet ohne Takthalbierung und ohne Forwarding.

   (b) Die DLX Pipeline arbeitet mit Forwarding (EX→EX, MEM→EX) aber ohne Takthalbierung.

   (c) Versuchen Sie die Anzahl der Stalls für Fall a) durch Codeoptimierung (Umordnen der Befehle) zu minimieren.

19. Folgendes Programm zum Kopieren eines Speicherbereichs läuft durch eine einfache DLX Pipeline (ohne Verbesserungen).

```
      .data
len:  .word 100          ;length of data array
a:    .space 100         ;source
b:    .space 100         ;destination
      .text
main: addi r1, r0, a     ;source address
      addi r2, r0, b     ;destination address
      addi r3, r0, len   ;address of length
      lw   r3, 0(r3)     ;load length
loop: lb   r4, 0(r1)
      addi r1, r1, #1    ;increment source pointer
      sb   0(r2), r4
      addi r2, r2, #1    ;increment destination pointer
      subi r3, r3, #1    ;decrement counter
      bnez r3, loop
end:  trap 0
```

   (a) Geben Sie Art und Anzahl der Stalls an.

   (b) Optimieren Sie den Code so, dass die Anzahl der Stalls minimiert wird.

20. Schreiben Sie ein DLX-Unterprogramm, das die Fakultät einer ganzen Zahl $n \geq 0$ berechnet. Die Zahl n wird in `R1` übergeben und das Ergebnis $n!$ in `R2` zurückgegeben.

   (a) Schreiben Sie das Unterprogramm in möglichst kompakter Form.

   (b) Durch eine Verbesserung in der DLX Pipeline steht eine Sprungadresse nicht mehr nach MEM, sondern schon nach ID zur Verfügung, sodass ein Jump/Branch nur mehr einen Stall erzeugt. Eine weitere Verbesserung ist ein *Delay Slot*. Hier führt die Pipeline den Befehl nach dem Jump/Branch **immer** aus und führt eine eventuelle Verzweigung erst nach dem Befehl im delay slot aus. Dies bedeutet aber auch, dass im delay slot immer ein korrekter Befehl stehen muss.

   Schreiben Sie Ihr Programm so um, dass es auf einer Pipeline mit delay slot korrekt läuft und die Rechenzeit (im Vergleich zu a) verkürzt.