

Rechnerarchitektur

SS 2020

Übungszettel 6

19. Geben Sie Taktzyklendiagramme, die Anzahl der Stalls und die CPI-Raten für folgendes Codestück (in den Varianten (a) und (b)) an.

```

ADDI    R1, R0, #4
LW      R3, 0(R2)
MULT    R3, R3, R3
SUB     R4, R1, R5
AND     R5, R5, R8
SW      0(R2), R5
OR      R4, R3, R8

```

- (a) Die DLX Pipeline arbeitet mit Takthalbierung in den Stufen IF, ID, MEM und WB.
 (b) Versuchen Sie, die Anzahl der Stalls für Fall a) durch Codeoptimierung (Umordnen der Befehle) zu minimieren.
 (c) Berechnen Sie den *Speedup* von (b) in Relation zu (a).
20. 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.
21. 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.