

CT1 Übungsaufgaben Arithmetische und Logische Instruktionen

Aufgabe 1 – Codieren von arithmetischen Ausdrücken

Für die folgenden Variablen wurde entsprechend Speicherplatz im Datenbereich reserviert:

	AREA	progData, DATA, READWRITE
Zahl1	DCD	0; (32 bit)
Zahl2	DCD	0; (32 bit)
Zahl3	DCW	0; (16 bit)
Zahl4	DCW	0; (16 bit)
Zahl5	DCB	0; (8 bit)
Zahl6	DCB	0; (8 bit)

Codieren Sie die folgenden Ausdrücke in Assembler.

a) $R0 = R1 + R2$

```
ADDS    R0,R1,R2
```

b) $R0 = R1 + R2 + R3$

```
ADDS    R0,R1,R2
ADDS    R0,R0,R3
```

c) $R8 = R8 + 1$

```
MOVS    R0,#1
ADD     R8,R8,R0
```

d) $R1 = R2 + 200$

```
; ADDS    R1,R2,#200 falsch, rd = rn
MOVS    R1,#200
ADD     R1,R1,R2
```

e) $R10 = R8 - R7$

```
MOV     R0,R8
SUBS    R0,R0,R7
MOV     R10,R0
```

f) $R8 = - R8$

```
MOV     R7, R8
RSBS    R7,#0
MOV     R8,R7
```

g) $Zahl1 = Zahl1 - 1$

```
LDR     R1,=Zahl1
LDR     R0,[R1]
SUBS    R0,#1
STR     R0,[r1]
```

h) $\text{Zahl1} = \text{Zahl1} + \text{Zahl2}$

```
LDR    R0, =Zahl1
LDR    R1, [R0]
LDR    R2, =Zahl2
LDR    R3, [R2]
ADDS   R1, R1, R3
STR    R1, [R0]
```

i) $\text{Zahl3} = \text{Zahl3} - \text{Zahl4}$ (unsigned)

```
LDR    R0, =Zahl3
LDRH   R1, [R0]
LDR    R2, =Zahl4
LDRH   R3, [R2]
SUBS   R1, R1, R3
STRH   R1, [R0]
```

j) $\text{Zahl1} = \text{R5} * \text{Zahl3}$ (unsigned)

```
LDR    R0, =Zahl1
LDR    R1, =Zahl3
LDRH   R2, [R1]
MULS   R2, R5, R2
STR    R2, [R0]
```

Aufgabe 2 – Multiword Addition/Subtraktion

Für die folgenden 64-bit Variablen wurde entsprechend Speicherplatz im Datenbereich reserviert:

	AREA	progData, DATA, READWRITE
Long1	DCD	0; low word
	DCD	0; high word
Long2	DCD	0; low word
	DCD	0; high word
Long3	DCD	0; low word
	DCD	0; high word

Implementieren Sie die folgenden 64-bit Operationen in Assembler.

a) $\text{Long3} = \text{Long1} + \text{Long2}$

```
LDR    R0, =Long1
LDM    R0!, {R1-R4}
ADDS   R1, R1, R3
ADCS   R2, R2, R4
LDR    R0, =Long3
STM    R0!, {R1, R2}
```

b) $\text{Long3} = \text{Long2} - \text{Long1}$

```
LDR    R0, =Long1
LDM    R0!, {R1-R4}
SUBS   R3, R3, R1
SBCS   R4, R4, R2
LDR    R0, =Long3
STM    R0!, {R3, R4}
```