

CT1 Übungsaufgaben Log- Shift-Operationen und Casting

Aufgabe 1 – Logische Instruktionen

Geben Sie die Assemblerinstruktionen für die folgenden Fälle an.

- a) Invertieren des Inhaltes des Registers R1 (Bilden des Einerkomplementes)

- b) Verändern Sie den Inhalt des Registers R1 so, dass
 - die Bits 3..0 alle eins,
 - die Bits 7..4 alle null,
 - die Bits 17-16 alle invertiert und
 - alle übrigen Bits unverändert sind.

Aufgabe 2 – Multiplikation mit einer Konstanten

Schreiben Sie eine Codesequenz in Assembler welche den vorzeichenlosen Inhalt des Registers R7 mit der Dezimalzahl 43 multipliziert und im Register R7 abspeichert.

- a) Verwenden Sie die Multiplikationsinstruktion

- b) Ersetzen Sie die Multiplikationsinstruktion durch Shift- und Additionsbefehle

Aufgabe 3 – Reverse Engineering

Die folgenden Assemblersequenzen wurden durch den bcc Compiler aus einem C Programm erzeugt. Schreiben Sie einen möglichen C-Code, welcher diese Assemblersequenz erzeugt.

Die Speicherstellen im Assembler werden den ursprünglichen Variablen im C-Programm wie folgt zugeordnet:

ux	DCD	?	↔	uint32_t ux
uy	DCD	?	↔	uint32_t uy
uz	DCD	?	↔	uint32_t uz

	Assembler	C
Beispiel	<pre>LDR R0,=ux LDR R1,[R0] ADDS R1,#1 STR R1,[R0]</pre>	ux++;
a)	<pre>LDR R0,=ux LDR R1,[R0] LSLS R1,R1,#1 LDR R2,=uy LDR R3,[R2] ADDS R3,R1 LSLS R3,R3,#3 STR R3,[R2]</pre>	
b)	<pre>LDR R0,=ux LDR R1,[R0] LDR R2,=uy LDR R3,[R2] LDR R4,=uz LDR R5,[R4] LSRS R1,R1,#3 LSLS R3,R3,#4 ORRS R1,R3 MVNS R1,R1 ANDS R1,R5 STR R1,[R4]</pre>	

Aufgabe 4 – Explicit Casting in C

- a) Gegeben ist der folgende C Code:

```
uint8_t ux = 100;  
int8_t  sx = (int8_t)ux;
```

Als welche Dezimalzahl wird der Inhalt der Variable sx nach dem Cast interpretiert?

- b) Gegeben ist der folgende C Code:

```
int8_t  sx = -10;  
uint8_t ux = (uint_8)sx;
```

Als welche Dezimalzahl wird der Inhalt der Variable ux nach dem Cast interpretiert?