

Übung 6

Datentransfer-Befehle ARM V7M

Ziele dieser Übung:

Sie sollen in der Lage sein,

- 1) Ein Assembler-Projekt in der STM32CubeIDE zu erstellen
- 2) Die Datentransfer-Befehle eines Cortex-Mx anzuwenden

Die Übungen werden mit der STM32CubeIDE und dem Leguan-Board gelöst.

Aufgabe 6.1 Generieren eines Assembler-Projektes

Erzeugen Sie in der Entwicklungsumgebung STM32CubeIDE ein neues Assembler-Projekt. Importieren Sie dazu am besten das Assembler-Template auf Moodle (ganz am Schluss unter "Weiterführende Unterlagen"). Eine Anleitung dazu finden Sie ebenfalls auf Moodle.

Debuggen Sie dieses Projekt auf dem Leguan-Board. Setzen Sie auch Breakpoints und prüfen Sie die Werte der Register im Register-Fenster.

Aufgabe 6.2 Datentransfer-Befehle

Verwenden Sie für den Test Ihres Codes den Debugger und das Leguan-Board. Beachten Sie bei der Programmausführung die Registerinhalte, die Statusbits und das Memory.

Aufgabe 6.2a) MOV-Instruktion

Führen Sie die folgenden MOV-Instruktionen aus:

- Laden des Registers r1 mit dem Wert #1
- Laden des Registers r0 mit r1 mit anpassen der Statusbits, wie ändern diese?
- Löschen des Registers r2
- Laden des Registers r0 mit r2 mit anpassen der Statusbits, wie ändern diese?
- Laden des Registers r0 mit r1, welches um 3 Bit nach links verschoben wurde (LSL)
- Laden des Registers r0 mit r1, welches um 1 Bit nach rechts verschoben wurde (LSR)
- Laden des Registers r0 mit r1, welches um 1 Bit nach rechts rotiert wurde (ROR)

Aufgabe 6.2b) LDR und STR Instruktion, indirekte Adressierung

- Register r5 mit 0x20001000 laden (Adresse des Speichers für diese Übung)
- Register r0 mit dem Inhalt des Speichers auf Adresse 0x20001000 laden
- #0x55 im Speicher auf Adresse 0x20001000 ablegen

Aufgabe 6.2c) STM Instruktionen

- Register r5 mit 0x20001000 laden (Basis-Adresse des Speichers für diese Übung)
- r0 = 0x11, r1 = 0x22 und r2 = 0x33

Speichern Sie die Register r0 bis r2 jeweils ab Adresse 0x20001000 wie folgt:

- increment after

- decrement before

Um das Verhalten zu verifizieren ist es sinnvoll, wenn Sie jeweils nach einer STM-Instruktion r5 wieder mit der Basisadresse 0x20001000 laden und den Memory-Bereich in der Entwicklungsumgebung von Hand löschen.

Aufgabe 6.2d) PUSH Instruktion

- Auf welche Adresse zeigt der Stackpointer aktuell?
- PUSH von r5 auf den Stack, auf welche Adresse zeigt der Stackpointer?
- PUSH von r0 bis r2 auf den Stack, auf welche Adresse zeigt der Stackpointer?

Aufgabe 6.2e) POP Instruktion

- Löschen der Register r0 bis r2 und r5
- POP von Register r0 bis r2 vom Stack, auf welche Adresse zeigt der Stackpointer? Welche Werte nehmen r0 bis r2 an?
- POP von Register r5, auf welche Adresse zeigt der Stackpointer?