

CT Praktikum: Unterprogramme und Parameterübergabe

1 Einleitung

In diesem Praktikum schreiben und testen Sie ein Unterprogramm, welches den grössten Wert in einer übergebenen Tabelle von Integer-Werten (signed) herausucht.

2 Lernziele

- Sie können ein Unterprogramm erstellen, dieses aufrufen und entsprechende Parameter in Registern übergeben.
- Sie beherrschen das Sichern und Wiederherstellen von Registerwerten innerhalb eines Unterprogrammes.
- Sie festigen Ihr Wissen im Umgang mit Tabellen und den damit verbundenen Adressierungsarten.

3 Aufgaben

3.1 Aufgabe 1 – Unterprogramm search_max

Schreiben Sie in Assembler ein Unterprogramm search_max, welches in einer Tabelle mit **32-Bit Integer-Werten (signed)** den grössten Wert sucht und in Register R0 zurückgibt.

Vorgehen

Im abgegebenen Projekt „searchmax“ befinden sich im Ordner „app“ zwei Dateien. Implementieren Sie das Unterprogramm search_max in der Datei „search_max.s“, indem Sie das vorbereitete Gerüst erweitern.

In der Datei „main.s“ befindet sich eine Test-Bench, mit der Sie Ihre Implementation von search_max testen können. Auch die Testreihen, mit denen Ihre Implementation getestet werden, sind dort abgelegt.

Schnittstellen

Die Adresse der Tabelle wird im Register R0 übergeben, die Anzahl der Tabellenelemente im Register R1. Die Rückgabe des Maximalwertes soll im Register R0 erfolgen. Falls die Länge der Tabelle 0 ist, soll der Wert 0x80000000 in R0 zurückgegeben werden.

Denken Sie daran, dass Sie am Anfang des Unterprogrammes die Register R4-R7 auf den Stack sichern, wenn Sie diese verändern möchten. Am Ende stellen Sie die gespeicherten Register wieder her.

Die Register R0-R3 müssen gemäss ARM Procedure Call Standard nicht erhalten bleiben.

Test

Um die mitgelieferte Test-Bench zu verwenden, assemblieren Sie die Datei „main.s“ und starten Sie diese auf dem Zielsystem. Mit dem Taster „T0“ können Sie den Test anstossen. Das Resultat wird Ihnen auf dem LCD angezeigt.

3.2 Aufgabe 2 – I/O-Funktionen in Assembler

Implementieren Sie in Assembler eine Eingabe- und Ausgabe-Funktion, welche aus einem C-Programm heraus aufgerufen werden kann. Die Funktionen sollen in C folgende Deklaration aufweisen:

```
void out_word(uint32_t out_address, uint32_t out_value);
uint32_t in_word(uint32_t in_address);
```

Die Funktion `out_word()` soll den Word-Parameter `out_value` an die Adresse `out_address` schreiben.

Die Funktion `in_word()` soll ein Word von der Adresse `in_address` einlesen und zurückgeben. Halten Sie sich dabei an den ARM Procedure Call Standard.

Vorgehen

Erstellen Sie ein neues C-Projekt gemäss der Anleitung auf dem CT-Wiki („Create a C Project“).

Erstellen Sie im Projekt eine Assemblerdatei und implementieren Sie dort die beiden Funktionen `out_word()` und `in_word()`. In den Folien „CT_ParameterPassing“ auf Seite 23 finden Sie ein Beispiel, wie Sie ein Assembler Unterprogramm aus C heraus aufrufen können («Calling assembly subroutines from C»).

Erstellen sie ein C-File, das den Einsprungs-Punkt `main()` enthält. Lesen Sie dort mithilfe der beiden Assemblerfunktionen den Zustand der DIP-Switches ein und geben Sie den eingelesenen Wert auf den LED-Balken aus.

4 Bewertung

Die lauffähigen Programme müssen präsentiert werden. Die einzelnen Studierenden müssen die Lösung/Quellcode verstanden haben und erklären können.

Kapitel / Aufgabe	Bewertungskriterien	Gewichtung
3.1	Das Programm erfüllt die in Aufgabe 3.1 geforderte Funktionalität.	2/4
3.2	Das Programm erfüllt die in Aufgabe 3.2 geforderte Funktionalität.	2/4