

# Übungsblatt 11

(10 Punkte)

Präsenzaufgaben zur Woche ab Montag, 23. Januar 2017

Für die Bearbeitung dieses Blatts wird die virtuelle Maschine **RTEMS-de** verwendet.

Da unsere Lizenz für eines der verwendeten Tools auf bestimmte Rechner beschränkt ist, müssen Sie sich aus dieser VM heraus per SSH zum Zielrechner verbinden. Es sollte dabei keine erneute Passwortheingabe notwendig sein. Zur Verbindung mit dem Zielrechner verwenden Sie bitte das Kommando **ssh -X ls12pc5.cs.tu-dortmund.de** in einem Terminalfenster.

## Vorbereitung

In Ihrem Homeverzeichnis auf dem Zielsystem sind bereits die Dateien entpackt, die für die Übung notwendig sind. Sie befinden sich im Verzeichnis `wcet` – bitte beachten Sie, dass Sie wegen der Verbindung auf einen anderen Rechner nicht einfach den Dateibrowser der VM verwenden können, sondern auf die Kommandozeile angewiesen sind!

### 11.1 Schritt 1: Einfache Analyse (5 Punkte)

Verwenden Sie den Befehl `cd ~/wcet/step1` um in das Verzeichnis für diese Teilaufgabe zu wechseln. In diesem Verzeichnis befindet sich eine Textdatei `README.txt`, die Sie mit dem Befehl `cat README.txt` anzeigen lassen können.

Compilieren Sie das Beispielprogramm mit dem Befehl **`tricore-gcc -o test.elf -g -T ../tc1796.lds test.c`** und schauen Sie sich zudem den Quellcode an (z.B. via `kate test.c`, `gedit test.c`, `gvim test.c` oder `cat test.c`).

#### Was macht das Programm?

Verwenden Sie nun den Befehl `a3tricore` um dem Tricore-Analyzer zu starten. Wählen Sie unter Configuration→Files die Datei `test.elf` als Executable aus und `a.ais` als "AIS file". Klicken Sie danach auf das Stiftsymbol in der "AIS file"-Zeile, um den AIS-Editor zu starten. **Wie werden die Loop Bounds mit dieser Datei annotiert?**

Wählen Sie den Punkt "Analyses→Create" aus der Navigationsbox links aus und klicken Sie auf die Option "aiT" (Safe WCET Analysis). In dem daraufhin geöffneten Bildschirm für die Analyseparameter muss nun noch `main` als Symbol für den Analyse-Start ausgewählt werden, bevor Sie die Analyse mit Klick auf den "Play"-Knopf (der rechte nur mit Dreieck und ohne Punkt) in der Symbolleiste starten können.

**Was ist die WCET des Programms?** Sie können zusätzliche Details aufrufen, indem Sie auf den Punkt "WCET contributions" in der Navigationsbox links klicken.

Schauen Sie sich zudem den Kontrollflussgraphen an (Analysis→Control-Flow Graph).

**Wie wird das Programm im Kontrollflussgraphen repräsentiert?** Beachten Sie insbesondere, wie Schleifen im CFG dargestellt werden.

Starten Sie die Analyse erneut, aber verwenden Sie diesmal die interaktive Analyse (Play-Symbol mit Punkt an der Spitze). Schauen Sie sich an, welche zusätzlichen Informationen in den Boxen dargestellt werden und öffnen Sie

einige der Boxen durch Doppelklick. **Was wird dort dargestellt?**

## 11.2 Schritt 2: Scratchpad-Allokation (5 Punkte)

Ein Scratchpad (SPM, ScratchPad Memory) ist ein besonders schneller, interner Speicher, der für die temporäre Zwischenspeicherung von Rechenergebnissen, Daten oder Programmteilen verwendet wird. Im Gegensatz zu einem Cache wird ein Scratchpad-Speicher vom Programmierer verwaltet und nicht automatisch von der Hardware.

**Welche Teile eines Programms sollten in ein SPM ausgelagert werden?**

**Was unterscheidet ein "normales" System von einem Echtzeitsystem wenn es um den Einsatz eines SPMs geht?**

Wechseln Sie in das Verzeichnis `step2` (`cd ~/wacet/step2`). Hier befindet sich eine weitere Kopie des Programms aus Aufgabe 1, in dem zwei Funktionen in das SPM verschoben wurden. Compilieren Sie es mit `tricore-gcc -o test.elf -g -T ../tc1796.lds test.c` und lassen Sie es von aiT analysieren.

**Wie hat die SPM-Nutzung die Laufzeit des Programms beeinflusst?**

**Wenn Sie eine weitere Funktion ins SPM legen könnten, welche würden Sie wählen? Warum?**