



# Rechnerarchitektur – Praktische Übungen

## Übung 3: Leistungsmessung und Befehlssätze

### Aufgabe 3.1: Leistungsmessung (4 Punkte)

- a) Gegeben seien folgende Messwerte für Programme eines Benchmarks:

Programm	Referenzzeit	Gemessene Zeit
Programm A	1	10
Programm B	500	100

Berechnen Sie die Verhältnisse der Laufzeiten und anschließend das arithmetische sowie das geometrische Mittel der Einzelwerte. Vertauschen Sie anschließend Referenzzeit und gemessene Zeit. Was fällt dabei auf?

- b) In einem Server wartet die CPU 60% der Zeit auf Ein-/Ausgabe. Es wird eine zehn Mal schnellere CPU eingebaut. Wie groß ist die Gesamtbeschleunigung?
- c) Nach Einbau einer zehn Mal schnelleren CPU in einen Web-Server wartet die CPU 50% der Zeit auf I/O. Wie groß ist die Beschleunigung?
- d) Wie groß ist der IPC-Wert einer CPU, die ein Programm mit 50.000 Befehlen in 20.000 Taktzyklen (oops) ausführt?
- e) Ist es möglich, dass ein Programm auf einer CPU mit 1000MIPS in 6 Sekunden und von einer zweiten CPU mit 750MIPS in 4 Sekunden ausgeführt wird?
- f) Es werden zwei Prozessoren P1 und P2 miteinander verglichen. Dazu wird ein C-Programm mit entsprechenden Compilern für beide Prozessoren übersetzt. Eine Leistungsmessung ergibt folgende Daten:

Prozessor	CPI	Befehlsanzahl	Ausführungszeit
P1	5/4	2.000.000	2,0 ms
P2	4/5	1.500.000	1,5 ms

- Welchen Prozessor würden Sie wählen, um das Programm ausführen zu lassen (Begründung!)?
- Berechnen Sie den jeweiligen MIPS-Wert für P1 und P2.
- Berechnen Sie den jeweiligen Prozessortakt für P1 und P2.



### **Aufgabe 3.2: Spezialregister bei MMIX und OpenRISC 1200 (3 Punkte)**

- Welche der in der Vorlesung genannten Spezialregister gibt es bei MMIX? Vervollständigen Sie untenstehende Tabelle mit den entsprechenden Angaben zu Registerart und der entsprechenden Bezeichnung bei MMIX.
- Informieren Sie sich im WWW über die Register des OpenRISC-Prozessors (→ entsprechende Links finden Sie auf der Moodle-Seite zur Veranstaltung). Welche der in der Vorlesung genannten Spezialregister gibt es dort? Tragen Sie die passenden Bezeichnungen wiederum in untenstehender Tabelle ein.

Registerart	MMIX	OpenRISC 1200

### **Aufgabe 3.3 (praktisch): Leistungsmessung mit dem ORSoC Entwicklungsmodul**

Im Folgenden wollen wir die Leistung der OpenRISC 1200 CPU auf dem von uns im Praktikum verwendeten SoC mit Hilfe unterschiedlicher Benchmarks ermitteln und mit der Leistung eines Desktop-PCs vergleichen.

- Starten Sie Ihre VM und öffnen Sie eine Konsole.
- Verbinden Sie das Modul über das USB Kabel mit dem PC.
- Verbinden Sie das USB-Modul mit der virtuellen Maschine (Eine Anleitung zum Vorgehen finden Sie auf dem im ersten Praktikum ausgeteilten Hinweisblatt zur Verwendung der VM.)
- Wir werden zur Kommunikation mit dem Modul das minimalistische Kommunikationsprogramm „picocom“ verwenden. Informieren Sie sich durch Abruf der entsprechenden Manual-Page (Befehl: `man picocom`) über die Bedienung dieses Programmes.

- Öffnen Sie ein Terminal und verbinden Sie sich über die serielle Schnittstelle mit dem Modul, indem Sie folgende Befehle eingeben:

```
picocom -b 115200 /dev/ttyUSB2
```

Drücken Sie [ENTER] – wenn die Verbindung erfolgreich hergestellt wurde, sollten Sie nun den Prompt des ORPMon Programmes sehen.

- Informieren Sie sich (durch Eingabe von „help“ + [ENTER]) über die Bedienung des ORPMon Programmes.
- Führen Sie mit Hilfe von ORPMon den Dhrystone-Benchmark auf dem Modul mindestens fünf Mal aus und notieren Sie sich die Ergebnisse. Ermitteln Sie den Mittelwert und die Varianz. Schätzen Sie mit Hilfe des Mittelwerts ab, wie viele ORSoC Module sie benötigen würden, um die Rechenleistung eines aktuellen PCs zu erreichen (gemäß Dhrystone Benchmark).



- h) Führen Sie nun den Coremark Benchmark mehrfach in der Variante „VALIDATION RUN“ auf dem Modul aus und notieren Sie erneut Mittelwert und Varianz.
- i) Führen Sie nun den Coremark Benchmark mehrfach in der Variante „PERFORMANCE RUN“ auf dem Modul aus und notieren Sie erneut Mittelwert und Varianz. Schätzen Sie ab, wie viele ORSoC Module sie benötigen würden, um die Rechenleistung eines aktuellen PCs zu erreichen (gemäß Coremark Benchmark).
- j) Als letzter Benchmark soll nun der sogenannte BogoMips-Wert (abgeleitet von „bogos“: betrügerisch und MIPS) für das ORSoC Modul ermittelt werden. Dieser Wert wird beim Booten eines Linux-Systems vom Kernel gemessen. Daher soll nun zunächst auf dem OpenRISC Board ein Linux-Kernel gestartet werden.

- Ein Weg, um eigene Software auf das Entwicklungsmodul zu laden, ist, den GNU Debugger gdb dafür zu benutzen. Hierzu starten Sie zunächst einen sogenannten Debug-Proxy. Dies ist ein Programm, welches lokal auf Ihrem virtuellen Laborrechner ausgeführt wird und mit dem Entwicklungsmodul über ein USB JTAG Interface kommuniziert (JTAG steht für *Joint Test Action Group*, es handelt sich um eine standardisierte Schnittstelle zum Debuggen von integrierten Schaltungen). Dieses lokale Programm kommuniziert dann mit Hilfe eines lokalen Netzwerksockets mit dem GNU Debugger gdb, den man zur Steuerung und zum Debuggen von Software auf dem Entwicklungsmodul nutzen kann.

Entladen Sie zunächst das Treibermodul für die seriellen Schnittstellen zum Modul

```
sudo rmmod ftdi_sio
```

***Hinweis:** Sollte dieses mit dem Hinweis, dass das Modul noch in Benutzung ist, fehlschlagen, so beenden Sie alle Prozesse, die noch die seriellen Ports zum Entwicklungsmodul nutzen und versuchen Sie es erneut.*

Starten Sie den Debug-Proxy in einer eigenen Konsole durch Eingabe von:

```
sudo /opt/or_debug_proxy/bin/or_debug_proxy -r 55555
```

Der Proxy sollte sich melden und Ihnen mit dem Hinweis „Stalling OR1K CPU0“ mitteilen, dass er die CPU kontaktiert und angehalten hat.

- Wechseln Sie in das Verzeichnis mit den Linux-Quellen: `cd ~/soc-design/linux`
- Erzeugen Sie eine Standardkonfiguration für den OpenRISC Prozessor:

```
make ARCH=openrisc defconfig
```
- Kompilieren Sie den Linux-Kernel mit dieser Standard-Konfiguration:

```
make ARCH=openrisc
```
- Wenn dieser Übersetzungsvorgang vollständig durchgelaufen ist, finden Sie im Verzeichnis den ausführbaren Code des Linux-Kernels in der Datei `vmlinux`
- Bevor das erzeugte Binärfile auf das Modul geladen und ausgeführt wird, verbinden Sie sich wieder mit dem Programm picocom mit der seriellen Schnittstelle für die Ausgaben des Moduls. Da die eine über USB übertragene serielle Schnittstelle bereits durch den Debug-Proxy verwendet und nicht mehr sichtbar ist, laden wir jetzt den Treiber erneut und verbinden uns mit der seriellen, über USB angebundenen Schnittstelle `ttyUSB1`

```
sudo modprobe ftdi_sio  
picocom -b 115200 /dev/ttyUSB1
```



- Um den erstellten Kernel zu starten, muss dieser zunächst mit `gdb` auf das Entwicklungsboard übertragen werden. Starten Sie dazu den Debugger `gdb` und verbinden Sie sich mit dem Debug-Proxy über den lokalen Socket auf Port 55555, indem Sie folgende Kommandos eingeben:  

```
or32-elf-gdb  
(gdb) target remote :55555
```
- Als nächstes teilen wir dem Debugger mit, welches Programm wir debuggen/ausführen und laden die entsprechende Datei:  

```
(gdb) file vmlinux  
(gdb) load
```
- Jetzt setzen wir über den Debugger ein Spezialregister (Special Purpose Register, SPR) des Prozessor: Der neue Wert für den Befehlszähler (Next Program Counter, NPC) wird auf die Adresse des Startpunktes des Linux Kernels gesetzt und mit `c` (für continue) die Programmausführung fortgesetzt:  

```
(gdb) spr npc 0x100  
(gdb) c
```
- Jetzt sollten Sie innerhalb der Konsole, welche Sie zur Anzeige der Ausgaben des OpenRISC Moduls gestartet haben, mit dem Linux auf dem Board arbeiten können. Lesen Sie den `BogoMIPS`-Wert aus (Sie finden ihn unter `/proc/cpuinfo`), machen Sie zum Beleg einen Screenshot der Ausgabe und vergleichen Sie den `BogoMIPS`-Wert wiederum mit dem `BogoMIPS` Wert eines Desktop-PCs.

Vergleichsliste mit `BogoMIPS` Werten: <http://tldp.org/HOWTO/BogoMips/bogo-list.html>

Für die Abnahme des praktischen Teils dieses Aufgabenblattes benötigen Sie:

- Liste Ihrer Messwerte mit Angabe des Mittelwertes und der Varianz jeweils für `Dhrystone` und `Coremark`.
- Screenshot der Ausgabe des `BogoMIPS` Wertes des Boards.
- Angabe der jeweiligen Vergleiche mit einem Desktop-PC