

Professor Dr.-Ing. Stefan Kowalewski  
Dipl.-Inform. Dominik Franke  
Dipl.-Inform. Sebastian Biallas

Aachen, 10. Dezember 2010  
SWS: V4/Ü2, ECTS: 7

## Einführung in die Technische Informatik

WS 2010/2011

Blatt 9: MMIX

Ihre Lösung zu den mit (\*) gekennzeichneten Übungen sollen Sie am **17.12.** in der Übung abgeben. Die Bearbeitung der Aufgaben in Lerngruppen ist sinnvoll. Bitte geben Sie nur eine Lösung pro Lerngruppe ab.

**Hinweis:** Zur Lösung der Aufgaben sollten Sie auch in die MMIX-Dokumentation schauen.  
<http://www-cs-faculty.stanford.edu/~uno/mmix-news.html>

### Aufgabe 1: (\*) Arithmetik

- a) Geben Sie den Inhalt des Registers \$0 nach Ausführung der folgenden Befehle an. Es gilt: \$1 = 11, \$2 = 3, \$3 = -8, \$4 = -19
- i) ADD \$0,\$3,\$4      ii) SADD \$0,\$1,\$2
  - iii) CMP \$0,\$1,\$3    iv) CMPL \$0,\$1,\$3
  - v) XOR \$0,\$3,\$4      vi) SL \$0,\$1,\$2
  - vii) SRU \$0,\$4,\$2    viii) SR \$0,\$4,\$2
- b) Wie unterscheiden sich die Befehle ADD und ADDU? Geben Sie ein Beispiel mit konkreten Registerinhalten vor und nach Ausführung der Befehle an, das den Unterschied zeigt.
- c) Mit Hilfe des Befehls SETL \$0,c kann Register \$0 auf eine Konstante  $0 \leq c < 2^{16}$  gesetzt werden. Wie können Sie Register \$0 mit einem Befehl auf -1 setzen, ohne etwas aus dem Speicher zu laden? (Es gibt min. drei Möglichkeiten)

### Aufgabe 2: (\*) Bitweise Operationen

- a) Wie können Sie das 5. Bit in Register \$0
- i) setzen,              ii) löschen,
  - iii) invertieren,      iv) bzw. testen?
- b) Es gelte nun Register  $0 \leq \$1 \leq 63$ . Wie können Sie das Bit, das durch Register \$1 vorgegeben wird, in Register \$0 setzen?
- c) Wie können Sie die Bits in einem Register nach links rotieren (das MSB wird in das LSB geschrieben, alle anderen eine Stelle weiter nach links)?

### Aufgabe 3: (★) Speicherzugriffe

- a) Geben Sie den Inhalt des Registers \$0 nach Ausführung der folgenden Befehle an. Gehen Sie davon aus, dass vor der Ausführung folgende Register- und Speicherbelegungen gelten: Register \$1 = 1000, Register \$2 = 5

$a$	999	1000	1001	1002	1003	1004	1005	1006	1007	1008
$M_1[a]$	$8f_{16}$	$90_{16}$	$91_{16}$	$92_{16}$	$93_{16}$	$94_{16}$	$95_{16}$	$96_{16}$	$97_{16}$	$98_{16}$

- i) LDB \$0,\$1,\$2      ii) LDBU \$0,\$1,\$2  
iii) LDT \$0,\$1,\$2      iv) LDO \$0,\$1,\$2
- b) Geben Sie für alle veränderten Speicherstellen ihren neuen Inhalt nach Ausführung der folgenden Befehle an. Es gilt dabei die Registerbelegungen:  
Register \$0 = 09f911029d74e35b<sub>16</sub>, Register \$1 = 1000, Register \$2 = 5

- i) STB \$0,\$1,\$2      ii) STBU \$0,\$1,\$2  
iii) STWU \$0,\$1,\$2      iv) STTU \$0,\$1,\$2

### Aufgabe 4: (★) Absolutbetrag

Geben Sie ein MMIX-Programm an, das den Absolutbetrag  $\text{abs}(\$1)$  von Register \$1 berechnet und in Register \$0 schreibt,

- a) unter Verwendung von Sprungbefehlen, und  
b) ohne Verwendung von Sprungbefehlen.

**Hinweis:** Sie können annehmen, dass für Register \$1 gilt  $-2^{63} < \$1 < 2^{63}$ .

### Aufgabe 5: (★) Hammingabstand

In dieser Aufgabe betrachten wir die MMIX-Register als Bitvektoren mit jeweils 64 Stellen. Für Bitvektoren  $x, y$  verstehen wir unter dem *Hammingabstand*  $h(x, y)$  die Anzahl der Stellen, an denen sich  $x$  und  $y$  unterscheiden. So gilt beispielsweise  $h(\underline{0111}_2, \underline{1110}_2) = 2$  und  $h(101010_2, 101010_2) = 0$ .

- a) Für einen Bitvektor  $x$  wird  $h(x, 0)$  auch als *Hamminggewicht* von  $x$  bezeichnet. Geben Sie einen MMIX-Befehl an, der das Hamminggewicht  $h(\$1, 0)$  für Register \$1 berechnet und in Register \$0 schreibt.
- b) Geben Sie nun ein MMIX-Programm aus max. zwei Befehlen an, welches den Hammingabstand  $h(\$1, \$2)$  aus Register \$1 und \$2 berechnet und in Register \$0 schreibt.

### Aufgabe 6: Programmieren

Schreiben Sie ein MMIX-Programm, das einen Rechner nach umgekehrter polnischer Notation implementiert.