veritas iustitia libertas
Freie Universität Berlin

# Rechnerorganisation
# SS 2004
# Übungsblatt Nr. 2

**Prof. Dr.-Ing. Robert Tolksdorf**
**Freie Universität Berlin**

***Ausgabe am 06.05.2005 — Abgabe spätestens 20.05.2005, 12:00 Uhr***

**Bitte bei der Abgabe nicht vergessen:**
alle Namen und Matrikelnummern der Gruppe,
Nummer der Übung/Teilaufgabe und
Datum auf die Abgabe schreiben.

Darauf achten, dass die Lösungen beim richtigen Tutor/der richtigen Tutorin abgegeben werden.
**Zu spät abgegebene Lösungen werden nicht mehr berücksichtigt!**

## 1. Aufgabe: ISHR (8 Punkte)

Die Instruktion `ISHR` (Arithmetic Shift Right Integer) gibt es in der JVM, nicht aber in der vereinfachten IJVM nach Tanenbaum. Dieser neue Befehl nutzt die beiden oberen Werte des Stapels und ersetzt beide durch einen einzelnen Wert – das Ergebnis. Das zweite Wort von oben im Stapel ist der zu verschiebende Operand. Sein Inhalt wird um einen Wert zwischen 0 und einschließlich 31 nach rechts verschoben, je nach dem Wert der fünf am wenigsten signifikanten Bits des obersten Stapelworts (die übrigen 27 Bits des obersten Worts werden ignoriert). Das Vorzeichenbit wird um die Bitzahl, die der Verschiebungszahl entspricht, nach rechts repliziert. Der Opcode für `ISHR` lautet 122 (0x7A).

1. Welche arithmetische Operation entspricht der Verschiebung nach rechts?
2. Erweitern Sie den Mikrocode um diese Instruktion als Teil von IJVM.

## 2. Aufgabe: MIC, MAL & Co. (8 Punkte)

Nennen Sie den Micro Assembler Language (MAL) Befehl für die folgende hexadezimale Mikroinstruktion : 0x`948048027` (1 Punkt)

Was macht dieses Micro Assembler Language Codefragment ? (4 Punkte)

```
test1 MAR  = SP = SP - 1; rd
test2 H    = TOS
test3 OPC  = MDR OR H
test4 H    = MDR AND H
test5 H    = NOT H
test6 TOS  = MDR = OPC AND H; wr; goto Main1
```

Was ist die hexadezimale Mikroinstruktion für den folgenden Micro Assembler Language (MAL) Befehl, wenn L1 an Adresse 111010011 und L2 an 011010011 liegt? (3 Punkte)

```
Z = TOS ; if (Z) goto L1 ; else goto L2
```

## 3. Aufgabe: Vorbereitung auf Assembler (4 Punkte)

Die Assemblerprogrammierung in RO soll mit Hilfe von MS Visual Studio .NET geschehen. Diese Entwicklungsumgebung erlaubt nicht nur das Programmieren in C++, C# etc. – sondern eben auch Assembler unter Nutzung der IA-32-Befehle. Das Programm ist auf allen Pool-Rechnern vorhanden. Machen Sie sich jetzt schon mit

der Oberfläche, dem Compiler, Debugger etc. etwas vertraut. Gehen Sie dazu am Besten in die Hilfe, schauen Sie sich Beispiele an und vollziehen Sie diese nach. Natürlich können Sie auch mit anderen Systemen Ihrer Wahl IA-32-Programmierung durchführen.
Einfaches Loslegen in Richtung Assembler: File – New – Project, dann Visual C++ projects/Win 32 projects anklicken, Name eingeben und auf OK. Danach (und das ist am Anfang sinnvoll) via Application Settings „Console Application" wählen. Damit erhalten wir eine ganz einfache Anwendung, ohne irgendwelchen Fenster-Zusatz – einfach ein Programm für eine Kommandozeile. Finish anklicken.
In dem von Ihnen gewählten Projektverzeichnis finden Sie eine Datei mit dem Projektnamen und der Endung cpp, diese können Sie nun z.B. wie folgt erweitern (das ist nur ein Beispiel, je nach dem von Ihnen gewählten Namen sieht das natürlich etwas anders aus).

```
#include "stdafx.h"
#include <stdio.h>

int power2(int num, int power);

int _tmain(int argc, _TCHAR* argv[])
{
      printf("Hello World\n");
      printf("3 times 2 to the power of 5 is %d\n", power2(3, 5));
      while (getchar()==EOF);
      return 0;
}

int power2(int num, int power)
{
      __asm
      {
            mov eax, num;
            mov ecx, power;
            shl eax, cl;
      }
}
```

} Das ist der Teil, der uns interessiert.

Dies ist nur ein primitives Programm, zeigt aber, wie wir in RO Assembler programmieren wollen: eingebettet in C-Funktionen! Dazu müssen Sie kein C können, dieses Vorgehen macht aber das Leben leichter, indem beliebig Variablen verwendet werden können (hier *num* und *power*), der Debugger zum Einsatz kommen kann etc.

1. Was steht in EAX nach Ausführung von SHL?
2. Welche Bedeutung haben die Bezeichnungen EAX, CL oder ECX?