

INSTITUT FÜR INFORMATIK
Lehrstuhl für Rechnerarchitektur (Informatik 3)
Universität Erlangen-Nürnberg
Martensstr. 3, 91058 Erlangen

14.2.2003

Klausur
 zu

"Organisation und Technologie von Rechensystemen 4"

.....

Matrikelnummer Geb.-Datum Vorname Name

- Es sind keine elektronischen Hilfsmittel erlaubt.
- Legen Sie den Ausweis (mit Lichtbild!) griffbereit auf den Platz.
- Dieses Aufgabenheft umfaßt 10 Seiten. Überprüfen Sie die Vollständigkeit.
- Gesondert beigelegte Blätter werden nicht bewertet!
- Schreiben Sie deutlich! Unleserliches wird nicht bewertet!
- Es darf nicht mit der Farbe rot geschrieben werden!
- Bekanntgabe der Ergebnisse: Aushang in den Semesterferien
- Einsichtnahme: Siehe Aushang
- Bei Bestehen wird der Schein direkt an das Prüfungsamt geschickt!

Durch meine Unterschrift bestätige ich

- den Empfang der vollständigen Klausurunterlagen
- die Kenntnisnahme der obigen Informationen.

Erlangen, den 14.2.2003

(Unterschrift)

Ich bin damit einverstanden, daß mein Prüfungsergebnis der Klausur zu OTRS 4 unter Angabe der Matrikel-Nummer veröffentlicht wird.

Erlangen, den 14.2.2003

(Unterschrift)

Aufgabe	1	2	3	4	5	6	Summe
max. Punktzahl	6	7	6	7	17	17	60
erreichte Punktzahl							

Aufgabe 1: Speicher**(6 Punkte)**

Gegeben sei die folgende Datenstruktur eines 32-Bit-Rechners:

```
struct data {
    char name[14];
    int punkte;
    int note;
};
```

Ein Datensatz gemäß dieser Struktur sei mit korrektem Alignment ab Adresse 1240 (dezimal) gespeichert.

Das folgende ist ein Speicherauszug des Speichers von Adresse 1232 (dezimal) bis 1280 (dezimal). Die Werte der Bytes dieses Bereiches werden im 16er-System angegeben:

```
1216: ...
1232: 00 00 00 40 00 00 00 01 4D 55 45 4C 4C 45 52 00
1248: 00 00 00 00 00 00 6A 12 00 00 00 32 00 00 00 02
1264: 4D 45 49 45 52 00 00 00 00 00 00 00 00 00 00 00
1280: ...
```

Wie heisst die/der Student(in) (name), deren/dessen Daten in diesem Bereich abgespeichert sind? Wieviele Punkte hat sie/er in der Klausur erreicht (punkte) und welche Note gab es dafür (note)? In welchem Format (Big-Endian, Little-Endian) speichert der Rechner vermutlich die Daten? Begründen Sie Ihre Antworten!

Hinweis: der Name ist im ASCII-Format abgespeichert ('A' = 0x41, 'B' = 0x42, ..., 'E' = 0x45, ..., 'I' = 0x49, ..., 'L' = 0x4C, ..., 'R' = 0x52, ..., 'U' = 0x55, ... 'Z' = 0x5A).

Aufgabe 2: Adressierungsarten**(7 Punkte)**

Ein RISC-Prozessor besitze 32 General-Purpose-Register ($\%r0$ bis $\%r31$) und einen 32-Bit-Daten- und Adressbus. Er unterstützt nur 4 verschiedene Adressierungsarten (s.u.). Wieviele Bits sind zur Kodierung eines Operanden bei folgenden Adressierungsarten jeweils mindestens erforderlich? Begründen Sie alle Ihre Antworten!

2a) Adressierungsart „Register“ (1 Punkt)

2b) Adressierungsart „Register indirekt“ (1 Punkt)

2c) Adressierungsart „Register indirekt mit Displacement“
(Wert des Displacements liegt zwischen -128 und +127) (1 Punkt)

2d) Adressierungsart „Immediate Operand“
(Wert des Operanden liegt zwischen 0 und $2^{16} - 1$) (1 Punkt)

Schreiben Sie ein Stück Assembler-Code, das den 32-Bit-Wert von der Adresse 0x12345678 des Speichers in das Register $\%r0$ liest! Verwenden Sie nur oben stehenden Adressierungsarten! (3 Punkte)

Aufgabe 3: Arithmetik**(6 Punkte)**

- 3a) Auf einem einfachen 8-Bit-Mikro-Controller (ohne Multiplikationsbefehl) soll der folgende Ausdruck implementiert werden (x und y seien je 8-Bit-Werte ohne Vorzeichen.):

$$y = x * 15;$$

- i) Schreiben Sie entsprechenden Assembler-Code für eine schnelle Multiplikation! (3 Punkte)

- ii) Für welche Werte von x kommen bei der Berechnung korrekte Wert von y heraus? Begründen Sie Ihre Antwort! (3 Punkte)

Aufgabe 4: Kontrollstrukturen**(7 Punkte)**

Gegeben sei folgender Assembler-Code:

```
    pushl %ebx
    pushl %ecx
    movl $1234, %eax
    movl $2345, %ebx
    movl $0, %ecx
    jmp cont
loop:
    movl (%eax), %edx
    movl %edx, (%ebx)
    addl $4, %eax
    addl $4, %ebx
    addl $1, %ecx
cont:
    cmp $10, %ecx
    jl loop
    popl %ecx
    popl %ebx
    ret
```

Die Ausführung jeder Instruktion (einschließlich Holen der vollständigen Instruktion) dauert mindestens zwei Takte. Jeder Speicherzugriff des Befehls verlängert die Ausführung um einen weiteren Takt. Der Prozessor läuft mit einer Taktrate von 10 MHz. Wie lange braucht er mindestens, um obiges Code-Stück zu durchlaufen? Begründen Sie Ihre Antwort!

Aufgabe 5: Unterprogramme**(17 Punkte)**

Gegeben sei folgende Hochsprachen-Unterfunktion (sie berechnet den größten gemeinsamen Teiler zweier Zahlen):

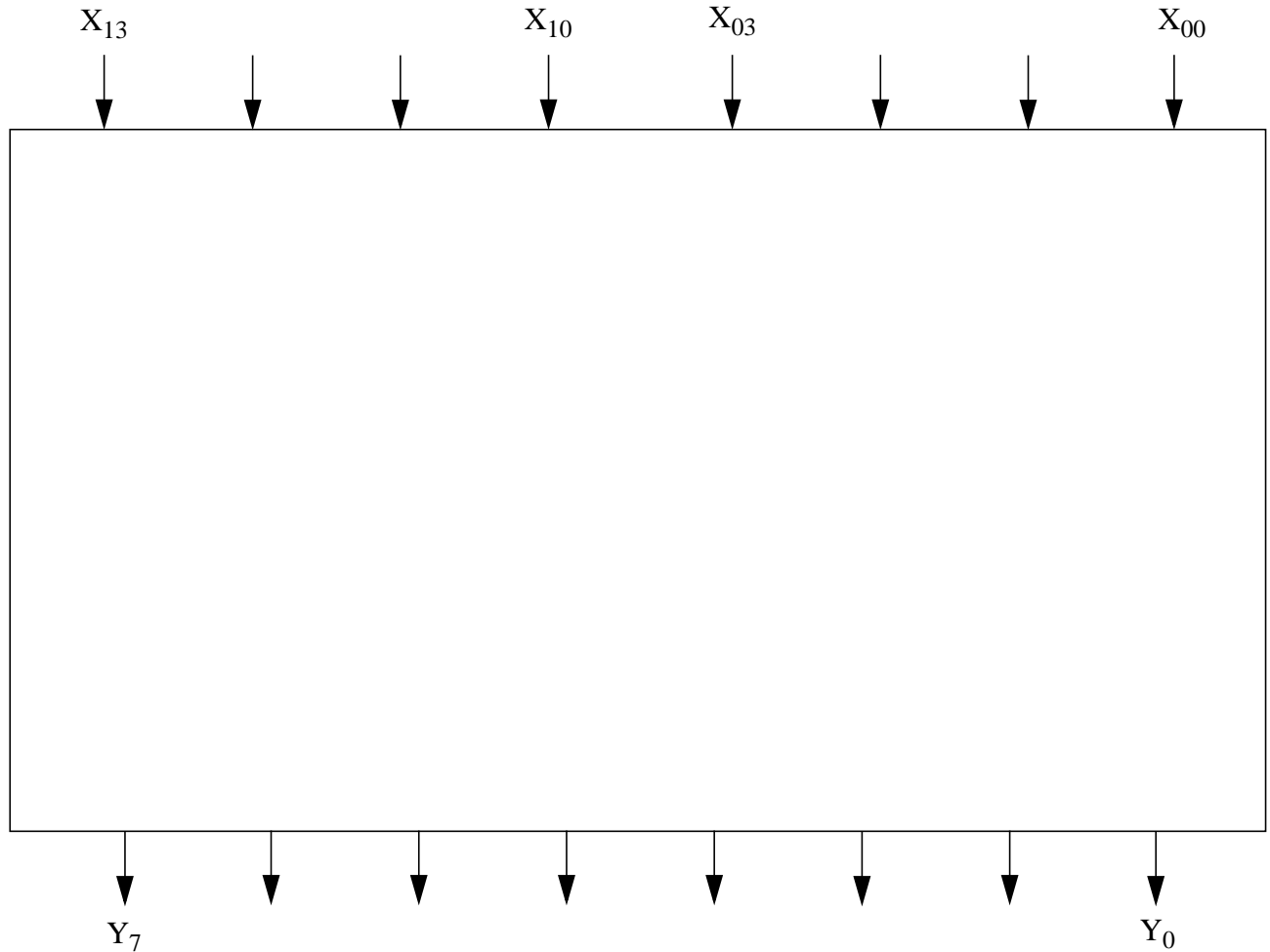
```
int ggt(int a, int b)
{
    if (a < b) {
        return ggt(a, b - a);
    } else if (b < a) {
        return ggt(b, a - b);
    } else {
        return a;
    }
}
```

Übersetzen Sie obige Hochsprachenfunktion in Assembler-Code! Kommentieren Sie die einzelnen Assembler-Instruktionen!

(Die Parameter sollen auf dem Stack, der Rückgabewert in einem Register übergeben werden.)

Aufgabe 6: Hardware**(17 Punkte)**

Ein Uhrenbaustein liefere in zwei 4-Bit-Nibbles X_1 , X_0 die BCD-Kodierung des Sekunden-Anteils (00...59). Geben Sie ein möglichst einfaches Schaltnetz aus elementaren Voll- und/oder Halb-Addierern an, das in ein Byte Y die Binärkodierung des Sekunden-Wertes ausgibt!



Zusätzlicher Platz

Zusätzlicher Platz

Zusätzlicher Platz