

Übung Compilerbau SS 2014

Serie 1

Prof. Dr. F. Mayer-Lindenberg, Dipl.-Math. Martin Eckert

10. April 2014

Aufgabe 1)

Schreiben Sie eine C-Funktion, die eine Hexadezimale Zahl (0x...) übergeben bekommt und eine Binärzahl zurück liefert.

```
int hex_bin(const char *hex)
```

Aufgabe 2)

Modifizieren Sie Aufgabe 1) nach Belieben für andere Zahlbereiche (Binär-, Ternär-, Oktal-, Dezimal-, Duodezimal-, Hexadezimal- oder Römische Zahlen).

```
int rom_dec(const char *rom)
```

Aufgabe 3)

Bezeichner in der Sprache VHDL dürfen nur aus Buchstaben, Zahlen und Unterstrichen bestehen. Nicht erlaubt sind dabei Zahlen oder Unterstriche am Anfang des Bezeichners, sowie Unterstriche am Ende. Doppelte Unterstriche dürfen ebenfalls nicht vorkommen.

Schreiben Sie eine C-Funktion `isVHDLidentifier()`, die eine nullterminierte Zeichenkette übergeben bekommt und einen Integer zurück liefert. Dabei soll im Falle eines gültigen VHDL-Bezeichners, wie beschrieben, eine 1, sonst eine 0 zurückgegeben werden.

```
int isVHDLidentifier(const char *identifier)
```

Aufgabe 4)

Der CPU1C-Prozessor besitzt acht 18-bit Datenregister i0-i7. Der implementierte Befehlssatz umfasst unter anderem folgende Befehle:

Befehl	Assemblercode	Maschinencode
unbedingter Sprung an die Adresse aa..aa	jp a	00 00 aa aaaa aaaa aaaa
Unterprogrammaufruf an der Adresse aa..aa und Sichern der Rücksprungadresse	call a	01 00 aa aaaa aaaa aaaa (aa..aa < 11..11)
Rücksprung aus Unterprogramm an die vorher gesicherte Rücksprungadresse	ret	01 00 11 1111 1111 1111
lade die vorzeichenerweiterte Konstante ss..ssnnnnnnnnnn in das Register t	it = n	10 01 1s nnnn nttt 1nnn
bitweises AND	ir = it & is	11 01 10 1rrr 0ttt 1sss
Registerinhalt an Schnittstelle a ausgeben	io(a) = it	11 01 11 aaaa 0ttt 1101

Schreiben Sie ein C-Programm, welches eine Datei mit den oben angegebenen Assemblerbefehlen einliest und in den entsprechenden (binären :)) Maschinencode kompiliert. Konstanten und Adressen in den Instruktionen liegen im Hexadezimalformat (0x...) vor. Die einzelnen Befehle sind durch Semikolon getrennt. Der Maschinencode soll zur Kontrolle im **Binär- und Hexadezimalformat** auf dem Bildschirm ausgegeben werden.