Freie Universität Berlin

Übung 2

Rechnerarchitektur
Wintersemester 17/18
Prof. Dr.-Ing. Jochen Schiller

Ausgabe 27.10.2017
Abgabe 10.11.2017, 10:15 (s.t.)

Bitte beachten Sie die allgemeinen Hinweise auf Übungszettel 1

Aufgabe 1: Von-Neumann-Architektur

Beantworten Sie folgende Fragen zur Von-Neumann-Architektur:

1. Mit der Von-Neumann-Architektur wurde erstmalig ein Konzept für einen ”general-purpose-Computer” vorgestellt. Was ist darunter zu verstehen?
2. Nennen und beschreiben Sie die Hauptbestandteile eines Von-Neumanns-Rechners. Fertigen Sie dazu eine Skizze des Von-Neumann-Models an.
3. Was ist der signifikante Unterschied zwischen der Von-Neumann-Architektur und der Harvard-Architektur?
4. Welche Aufgabe hat das Steuerwerk (CU) in einem Von-Neumann-Rechner? Welche Aufgabe übernimmt welche Komponente der CU?
5. Welche Aufgabe hat das Rechenwerk (ALU) in einem Von-Neumann-Rechner? Welche Aufgabe übernimmt welche Komponente der ALU?
6. Die Von-Neumann-Architektur unterscheidet zwischen Daten- und Adressbus. Warum ist das sinnvoll?
7. Welche Zusammenhänge bestehen zwischen den Größen (in Bits) des MARs, des MBRs, dem Adressbus, dem Datenbus und einer Speicherzelle sowie der Gesamtzahl der Speicherzellen im Speicher?
8. Erläutern Sie das Konzept der universellen Programmierbarkeit.
9. Warum ist das Zwei-Phasen-Konzept für die Befehlsverarbeitung in der Von-Neumann-Architektur nötig?
10. Erläutern Sie den Von-Neumann-Flaschenhals.

Aufgabe 2: Collatz Conjecture

Sprünge

In dieser Aufgabe sollen Sie Programmflusskontrolle in Assembler erlernen. Machen Sie sich dazu mit Assembler-Sprüngen vertraut (Befehle: CMP, TEST, JMP, LOOP, Jxx). Wie hängen Sprünge mit den Flags im Register ’rflags’ zusammen? Wann wird welches Flag im Register ’rflags’ gesetzt? Was ist der Unterschied zwischen den Befehlen ’JG’ und ’JA’?

Übersetzen Sie folgenden C-artigen Pseudocode in NASM-artigen Pseudocode:

```
if(x < 10) {
    S();
} else if(x == 10) {
    T();
} else if(x > 10) {
    V();
} else {
    W();
}
```

Finden Sie für das folgende Fragment zwei verschiedene, aber semantisch äquivalente Übersetzungen:

```
while(x != 10) {
    x = S();
}
```

Collatz Conjecture

Implementieren Sie eine Funktion die, die Collatzfolge einer übergebenen Zahl bildet und die Länge der Folge zurück gibt. Pseudocode:

```
collatz(n) {
    k = 0;
    while(n>1) {
        if(isEven(n)) {
            n /= 2;
        } else {
            n *= 3;
            n++;
        }
        k++;
    }
    return k;
}
```

Ein geeigneter C-Wrapper wird ihnen gestellt. Die Funktion soll folgende Signatur haben:

```
uint64_t collatz(uint64_t n);
```

Hinweis: "isEven" kann als "n mod 2 = 0" dargestellt werden.

Zusatzaufgabe

Übersetzen Sie auch noch die folgenden Pseudocodefragmente:

```
do {
    x = S();
} while(x != 10);
```

```
for(i=0; i<10; i++) {
    S(i);
}
```

```
x = S();
switch(x) {
    case 1:
        x += 100;
    case 2:
        x -= 20;
    case 10:
        x *= 10;
    default:
        x = 100;
}
```