

Digitale Systeme SS2019

Übungsblatt 5

Keine Abgabe

Aufgabe 1

(0 Punkte)

Die Verarbeitungspipeline einer CPU bestehe aus 5 Stufen: Befehl lesen (RI), Decodieren (DC), Operanden holen (RO), Ausführen (EX) und WriteBack (WB).

- (a) Wie lange dauert die Verarbeitung von 20 Befehlen, wenn jede Stufe 4 Takte benötigt?

- (b) Wie lange dauert die Verarbeitung von 25 Befehlen, wenn jede Stufe die folgende angegebene Zeit benötigt: RI - 3 Takte, DC - 2 Takte, RO - 3 Takte, EX - 1 Takte und WB - 3 Takte?

- (c) Ab welcher Anzahl abzuarbeitender Befehle ist der Prozessor nach (a) schneller als ein Prozessor ohne Pipeline, bei dem jeder Befehl in 6 Takten abgearbeitet wird?

Aufgabe 2

(0 Punkte)

Gegeben ist das folgende, in Assembler-Mnemonic, notierte Programm.

Das Format der 2-Operanden-Befehle sei `command op1,op2` mit `op2 = op2 op op1`

```
...  
1      SUB 2,R2  
2      BZ Label1  
3      ADD 5,R2  
4      MUL R4,R3  
5      SUB R4,R1  
6      MUL 2,R1  
7      ADD 5,R4  
8 Label1:  ADD 2,R2  
9      MOV R6,R5  
...
```

Das Programm wird von einem Prozessor ausgeführt, der mit einer Pipeline mit folgende Eigenschaften arbeitet:

- Pipeline zweiter Generation: jeder Befehl durchläuft alle Stufen, wobei einzelne je nach Befehl inaktiv bleiben (also wird auch der Sprungbefehl erst in der 5 Stufe abgeschlossen!)
- 5 Pipeline-Stufen:
 - Befehl holen
 - Befehl dekodieren
 - Register- und Speicheroperanden laden
 - Befehl ausführen
 - Register- und Speicheroperanden schreiben

Zu Beginn der Ausführung von Befehl 1 sei die Pipeline leer

- (a) Füllen Sie die untenstehende Tabelle aus, indem Sie die Belegung der Pipeline-Stufe durch Eintragen der jeweiligen Befehlsnummer andeuten (siehe Beispiele).

Welches Problem erkennen Sie?

Fetch	Decode	Read Ope- rands	Execute	Write Ope- rands
1				
2	1			

- (b) Nun wird für die Ausführung des Programms eine Pipeline der 3. Generation mit zweifach vorhandene Pipelinestufen angenommen.

Es sei jedoch keine automatische Berücksichtigung von Datenabhängigkeiten implementiert!

Wird das Programm fehlerfrei ausgeführt? Wie kann das Programm verändert werden, damit es wunschgemäß arbeitet? Hinweis: erzeugen Sie Verzögerungen indem Sie Befehle einfügen.

Fetch	Decode	Read Operands	Execute	Write Operands
1				
2				

Aufgabe 3

(0 Punkte)

Erläutern Sie kurz die folgenden, in Intel-Mnemonic notierten, x86-Assembler-Programmfragmente. Geben Sie außerdem an, welche Flags nach dem Ausführen der jeweils letzten Anweisung gesetzt sind (relevant sind SF, ZF, CF, OF und PF), was sie bedeuten und warum sie gesetzt wurden.

Hinweis: Verwenden Sie die Unterlagen aus dem Download-Bereich der Übungen.

(a)

```
mov eax, 0xffffffff
add eax, 2
```

(b)

```
mov eax, 0x7fffffff
add eax, 0x7fffffff
```

(c)

```
mov eax, 0
mov al, 8
mov ebx, eax
cmp eax, ebx
```

(d)

```
mov eax, 1  
mov ebx, 2  
sub eax, ebx
```

Aufgabe 4

(0 Punkte)

Schreiben Sie ein Intel-x86-Assemblerprogramm zur vorzeichenlosen, ganzzahligen Berechnung der Funktion $y = x^2 - 3 * x$.

Verwenden Sie je ein Unterprogramme zur Potenzbildung, Subtraktion und Multiplikation. Alle Werte können in 16-Bit dargestellt werden.

Die Unterprogramme unterscheiden sich in der Art der Parameterübergabe/-nahme.

Potenzierung:

Parameterübergabe in Registern:

Parameter und Ergebnis in Register *ax*

Multiplikation:

Parameterübergabe im Datenspeicher:

1. Parameter in Speicherzelle „xd“

2. Parameter in Speicherzelle „yd“

Ergebnis in Speicherzelle „product“

Subtraktion:

Parameterübergabe über den Stack

Ergebnis in Register *ax*

Die drei Unterprogramme für die Rechenoperationen dürfen die Register/den Hauptspeicher nur in angegebener Weise verändern.

Verwenden Sie zum Entwickeln das Tool „Jasmin“ (zu finden im Download-Bereich der Übungen).