



Rechnerarchitektur – Praktische Übungen

Übung 6: Gleitkommazahlen, Abhängigkeiten

Aufgabe 6.1: Gleitkommazahlen nach IEEE 754-2008 mit 64 Bit (4 Punkte)

Welches Bitmuster besitzen die folgenden Zahlen in IEEE 754-Darstellung? Geben Sie die hexadezimale Darstellung an (16 Stellen).

- a) 4,0
- b) 18,0
- c) -18,4

Welche Dezimalzahlen repräsentieren folgende Bitmuster?

- d) #4028 8000 0000 0000
- e) #C029 4000 0000 0000

Aufgabe 6.2 (Bonus): Genauigkeit von Fließkommazahlen nach IEEE 754-2008 (2 Punkte)

Schreiben Sie ein kurzes Programm (Java, C/C++, Excel, oder ...), welches die Genauigkeit von Fließkommazahlen auf dem von Ihnen benutzten Rechnersystem ermittelt. Beantworten Sie damit folgende Fragen:

- a) Was ist die kleinste von 0 verschiedene positive Zahl, die dargestellt werden kann? Durch welches Bitmuster wird sie dargestellt?
- b) Was ist die betragsmäßig kleinste Zahl, die zur Zahl 1 addiert werden kann, so dass ein von 1 verschiedener Wert entsteht?

Bestimmen Sie diese Genauigkeiten für Fließkommazahlen in doppelter Genauigkeit.

Hinweis: Die Lösung dieser praktischen Aufgabe ist freiwillig. Eine Abnahme ist nicht erforderlich – kann aber auf Wunsch durchgeführt werden.

Aufgabe 6.3: Abhängigkeiten (3 Punkte)

Geben Sie die Abhängigkeiten zwischen den folgenden MMIX-Befehlen in einem Abhängigkeitsgraphen an (Knoten sind die Befehle bzw. Zeilennummern; Pfeile existieren dann, wenn es eine Abhängigkeit gibt; transitive Pfeile brauchen nicht gezeichnet zu werden; keine RAR-Abhängigkeiten):

```

1      PREFIX      :CRC:
2 GenP  GREG        #0000000104C11DB2
3 degree IS        32
4 x      IS         $0
5 crc    IS         $1
6 b      IS         $2
7 msb    IS         $3
8 k      IS         $4
9 counter IS        $5
10
11 :CRC    SET       k, 7
12 2H      SRU       x, counter, k
13         AND       x, x, 1
14         SLU       crc, crc, 1
15         SLU       b, x, degree
16         XOR       crc, crc, b
17         SRU       msb, crc, degree
18         BZ        msb, weiter
19         XOR       crc, GenP, crc
20 weiter SUB       k, k, 1
21         BNN       k, 2B
22         POP       1, 0

```

Aufgabe 6.4 (praktisch): RAW Hazards auf OpenRISC (2 Punkte)

Ergänzen Sie den orpmon um einen neuen Befehl, welcher die Auswirkungen von RAW-Hazards auf unserer OpenRISC CPU überprüft. Hierzu messen Sie zunächst die Anzahl an Takten, welche für zwei arithmetische Instruktionen benötigt werden, ohne dass ein RAW-Hazard vorliegt. Danach ändern Sie die arithmetischen Instruktionen so ab, dass ein RAW-Hazard vorliegt und messen erneut die Takte. Geben Sie die ermittelten Messwerte auf der Konsole aus.

Beantworten Sie mit Hilfe der ermittelten Messergebnisse die Frage, ob die verwendete OpenRISC CPU die Technik des Result Forwarding unterstützt.

Für die Abnahme dieser praktischen Aufgabe brauchen Sie:

- das lauffähige, von Ihnen modifizierte orpmon-Programm,
- den zugehörigen, übersetzbaren Quellcode,
- die Messwerte für die Variante mit und ohne RAW-Hazard
- Ihre Antwort auf die Frage zur Unterstützung von Result Forwarding auf der OpenRISC CPU