6 . Hausübung

Gruppe: - Ulf Gebhardt (rbg: hu56nifa)
- Michael Scholz (rbg: mi48azih)

Aufgabe 1: Vergleichen zweier Architekturen:

MFP:
fp mul: 6 cpi
fp add: 3 cpi
fp div: 22 cpi
int op: 2 cpi

MNFP:
fp mul: 60 cpi
fp add: 40 cpi
fp div: 100 cpi
int op: 2 cpi

a)

MIPS-Rate = Millionen Instruktionen Pro sekunde
CPI = Cycles Per Instruction
Taktzyklen pro Sekunde = 1,5*10^9 (für beide Maschinen)

(Alle Angaben auf Programm P bezogen)

CPI für eine durchschnittliche Instruktion:
MFP: 0.12*6 + 0.13*3 + 0.07*22 + 0.68*2 = **4,01** cpi
MNFP: 0.12*60 + 0.13*40 + 0.07*100 + 0.68*2 = **20,76** cpi

Zeit(sec) für eine durchschnittliche Instruktion:
MFP: 4,01/(1,5*10^9) =~ **2,673 * 10^-9** sec/inst
MNFP: 20.76/(1,5*10^9) =~ **1.384*10^-8** sec/inst

MIPS-Rate:
MFP: 1 / (2,673 * 10^-9) = 374064837,9 inst/sec = **374**,0648379 MIPS
MNFP: 1 / (1.384*10^-8) = 72254335.26 Inst/sec = **72**.25433526 MIPS

b)

(Alle Angaben auf Programm P bezogen)

MFP: 350 Mio Inst		**MNFP: 3633 Mio Inst**
238 000 000 Int op	**->**	**238 000 000 Int op**
45 500 000 fp add	**->**	**910 000 000 fp add als int op**
42 000 000 fp mul	**->**	**1 260 000 000 fp mul als int op**
24 500 000 fp div	**->**	**1 225 000 000 fp div als int op**
----------		**----------**
350 000 000		**3 633 000 000**

c)

MFP: 350 / MIPS(MFP)* -> **0.9357 sec**
MNFP: 3 633 / MIPS(MNFP)* -> **50.2807 sec**

*(siehe a)

Aufgabe 2: Caches:

a)

Cache	m	C	B	E	S	t	s	b
1.	32	2048	8	1	256	21	8	3
2.	32	2048	32	1	64	21	6	5

b)

Speicherabbild:

a0	0x10010000
a1	0x10010004
a2	0x10010008
a3	0x1001000C
a4	0x10010010
a5	0x10010014
a6	0x10010018
a7	0x1001001C
b0	0x10010020
b1	0x10010024
b2	0x10010028
b3	0x1001002C
b4	0x10010030
b5	0x10010034
b6	0x10010038
b7	0x1001003C

Speicherzugriffsreihenfolge:

a0	0x10010000
b0	0x10010020
a1	0x10010004
b1	0x10010024
a2	0x10010008
b2	0x10010028
a3	0x1001000C
b3	0x1001002C
a4	0x10010010
b4	0x10010030
a5	0x10010014
b5	0x10010034
a6	0x10010018
b6	0x10010038
a7	0x1001001C
b7	0x1001003C

c)

Da wir jedes Datum (a0-b7) jedes mal nur ein einziges mal brauchen, bewirkt der Cache nichts. Es kommt zu keiner Performanceverbesserung. Das ist auch bei k != 3 der Fall.