Computer Architecture and system programming

Kristoffer Klokker

2022

Contents

1 Exercises

1.1 Benchmark

For the 40MHz processor which performed the instructions

Instruciton type	Instruction count	Cycles per instruction
Integer artihemtic	41,000	1
Data transfer	28,000	2
Floating point	25,000	2
Control transfer	6,000	2

1.1.1 Find the average CPI

$$\frac{1 \cdot 41000 + 2 \cdot 28000 + 2 \cdot 25000 + 2 \cdot 6000}{100000} = 1.59$$

CPI is the average cycles pr instruction. Therefore 4.5

1.1.2 Execution time

$$\begin{aligned}
CPI &= 1.59 \\
I_c &= 100000 \\
\tau &= \frac{1}{f} = \frac{1}{40000000Hz} \\
T &= I_c \cdot CPI \cdot \tau \\
&= 1.59 \cdot 100000 \cdot \frac{1}{40000000Hz} \\
T &= 0.003975s
\end{aligned}$$

1.1.3 MIPS

$$MIPS = \frac{f}{CPI \cdot 10^6}$$
$$CPI = 1.59$$
$$f = 40000000Hz$$
$$MIPS = \frac{40000000Hz}{1.59 \cdot 10^6}$$
$$MIPS = 25.16\frac{1}{s}$$

1.2 Explain how a negative number is represnted in the following representation

- Sign-magnitude - The left most bit must be 1 which result in the rest being interpretated as negative
- Twos compliment - The left most bit is 1 to subtract the maximum value from the rest
- Biased - Bias is most usually half the range therefore a negative value is simply less half the maximum value

1.3 Represent the following in 8 bit twos compliment and sing magnitude

- 64 - 00100000
- -28 - twos 11100100 - sign 10011100

1.4 Convert from twos compliment to decimal

- 1100110 : -26
- 1011101 : -35

1.5 Show the calculations in 8 bit twos compliment

1.5.1 6+12

$$6 = 00000110$$
$$12 = 00001100$$
$$00000110 + 00001100 = 00010010$$

1.5.2 -6+12

$$-6 = 11111010$$
$$12 = 00001100$$
$$11111010 + 00001100 = 00000110$$

Overflow is ignored

1.5.3 6-12

$$6 = 00000110$$
$$-12 = 11110100$$
$$00000110 + 11110100 = 11111010$$

1.5.4 -6-12

$$-6 = 11111010$$
$$-12 = 11110100$$
$$11111010 + 11110100 = 11101110$$

Overflow is ignored

1.6 Fill out the table for the most twos compliment addition

Input			Output		
x_{n-1}	y_{n-1}	c_{n-2}	z_{n-1}	c_{n-1}	v
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	1	1	1

Here x_{n-1} and y_{n-1} is the most signifact bits of the two addends.
c is the carry bit and z_{n-1} is the results most significatn bit.
v is a bit singnaling overflow.
If can be seen in row 2 and the last row that:
Overflow occurs if and only if the carry into the addition of the MSBs isdifferent from the carry out of that addition

1.7 Convert 23 and 29 to 6 bit twos compliment and multiply using Booths algorithm

$$23 = 010111$$
$$29 = 011101$$
$$A = 0$$
$$Q_{-1} = 0$$
$$M = 010111$$
$$Q = 011101$$
$$count = 5$$

$$Q_0, Q_{-1} = 10$$
$$A = A - M = 101001$$
$$shift\ A = 101001\ Q = 011101\ Q_{-1} = 0$$
$$A = 110100$$
$$Q = 101110$$
$$Q_{-1} = 1$$
$$count = 4$$

$$Q_0, Q_{-1} = 01$$
$$A = A + M = 110100 + 010111 = 001011$$
$$shift$$
$$A = 000101$$
$$Q = 110111$$
$$Q_{-1} = 0$$
$$count = 3$$

$$Q_0, Q_{-1} = 10$$
$$A = A - M = 000101 - 011001$$
$$A = 000101 + 101001 = 101110$$
$$shift\ A = 101110\ Q = 110111\ Q_{-1} = 0$$
$$A = 110111$$
$$Q = 011011$$
$$Q_{-1} = 1$$
$$count = 3$$

$$
\begin{aligned}
Q_0, Q_{-1} &= 11 \\
shift\ A = 110111\ Q = 011011\ Q_{-1} &= 1 \\
A &= 111011 \\
Q &= 101101 \\
Q_{-1} &= 1 \\
count &= 2
\end{aligned}
$$

$$
\begin{aligned}
Q_0, Q_{-1} &= 11 \\
shift\ A = 111011\ Q = 101101\ Q_{-1} &= 1 \\
A &= 111101 \\
Q &= 110110 \\
Q_{-1} &= 1 \\
count &= 1
\end{aligned}
$$

$$
\begin{aligned}
Q_0, Q_{-1} &= 01 \\
A = A + M &= 111101 + 011001 \\
A &= 010100 \\
shift\ A = 010100\ Q = 110110\ Q_{-1} &= 1 \\
A &= 001010 \\
Q &= 011011 \\
Q_{-1} &= 0 \\
count &= 0
\end{aligned}
$$

$$
010111 \times 011101 = AQ = 001010011011
$$

2 Exercises

2.1 Convert from IEE 754 floating point to decimal

2.1.1 1 1000 0010 0010 0000 0000 0000 0000 000

$$
\begin{aligned}
&= (-1)^1 \cdot 2^{(10000010)_2 - 127} \cdot (1.00100000000000000000000)_2 \\
&= -1 \cdot 2^{130-127} \cdot 1.125 \\
&= -9.000
\end{aligned}
$$

2.1.2 0 0111 1110 0000 1100 1100 1100 1100 110

$$\begin{aligned} &= (-1)^0 \cdot 2^{(01111110)_2 - 127} \cdot (1.00001100110011001100110)_2 \\ &= 1 \cdot 2^{126-127} \cdot 1.04999995231628417969 \\ &\approx 0.5249999760 \end{aligned}$$

2.1.3 0 1000 0000 1100 1100 1100 1100 1100 110

$$\begin{aligned} &= (-1)^0 \cdot 2^{(10000000)_2 - 127} \cdot (1.11001100110011001100110)_2 \\ &= 1 \cdot 2^{128-127} \cdot 1.79999995231628417969 \\ &\approx 3.599999904 \end{aligned}$$

2.2 Convert from decimal to IEEE 754 floating point

2.2.1 -720

$$\begin{aligned} 720 = 1011010000_2 \\ 1.011010000_2 \cdot 2^{9_{10}+127} \\ 1.011010000_2 \cdot 2^{136} \\ 10001000_2 = 117_{10} \\ 110001000011010000 \end{aligned}$$

2.2.2 0.645

$$\begin{aligned} 0.645 = 0.10100101000111101011110_2 \\ 1.0100101000111101011110_2 \cdot 2^{-1_{10}+127} \\ 1.0100101000111101011110_2 \cdot 2^{126} \\ 01111110_2 = 126_{10} \\ 00111111001001010001111010111 0 \end{aligned}$$

2.3 Which numbers can be exactly represented in IEE 754

- 17.0 - representable inside the range
- -1 - representable inside the range

- $\frac{7}{16}$ - representable inside range since equal 0.4375
- $\frac{1}{3}$ - not representable due to infinite
- π - not representable due to infinite
- $5.4321 \cdot 10^6$ - representable inside range
- $6.022 \cdot 10^{23}$ - representable inside range

2.4 Let C and D denote two number in IEEE 754 single-precision floating point format

$$C = 01000010101010100000000000000000$$
$$D = 11000010011111100000000000000000$$

2.4.1 What are the decimal values of C and D

$$\begin{aligned} C = (-1)^0 \cdot 2^{(10000101)_2 - 127} \cdot (1.01010100000000000000000)_2 \\ C = 1 \cdot 2^{133-127} \cdot 1.328125 \\ C = 85.000000 \end{aligned}$$

$$\begin{aligned} D = (-1)^1 \cdot 2^{(10000100)_2 - 127} \cdot (1.11111100000000000000000)_2 \\ D = -1 \cdot 2^{132-127} \cdot 1.984375 \\ D = -63.500000 \end{aligned}$$

2.4.2 Make the addition of floating points

$$
\begin{aligned}
X &= 01000001010101010000000000000000 \\
Y &= 11000001001111110000000000000000 \\
X_e &= 6 \\
X_b &= 1.01010100 \\
Y_e &= 5 \\
Y_b &= 0.11111100 \\
Y &= 1.11111100 \cdot 2^5 \\
Y &= 0.11111110 \cdot 2^6 \\
S = x_b - Y_b &= 0.01010110 \cdot 2^6 \\
S &= 1.010110 \cdot 2^4 \\
S_e = 4 + 127 = 131 &= 10000011_2 \\
S &= 01000001101011000000000000000000
\end{aligned}
$$

2.5 Create a truth table for the following algebra expression

A	B	C	$(A + \neg B + C)$	$(\neg A + B + \neg C)$	$(A + \neg B + C)(\neg A + B + \neg C)$
1	0	0	1	1	1
1	1	0	1	1	1
0	0	0	1	1	1
0	1	0	0	1	0
1	0	1	1	0	0
1	1	1	1	1	1
0	0	1	1	1	1
0	1	1	1	1	1

3 Exercises

3.1 Consider these two programs

```
for (i=1; i<n; i++) {
   z[i]=x[i]-y[i]
   z[i]=z[i]*z[i]
}
```

```
for (i=1; i<n; i++) {
   z[i]=x[i]-y[i]
}
for (i=1; i<n; i++) {
   z[i]=z[i]*z[i]
}
```

The function of the programs are finding the elemental difference between list x and y and squaring it.
This is done most efficient by the first program since the list can stay in cache unlike the second program which loads the list x, y and z and then loads z again.

3.2 Consider the cache with an access time of 5 ns and a hit ratio of $H = 0.9$. The memory access time alone is 100ns.

3.2.1 What is the average access time for this system?

$0.9 \cdot 5ns + 0.1 \cdot (100ns + 5ns) = 15ns$

3.2.2 Suppose the cache access time is increased to 6 ns. What is the minimum hit raio needed in order to not increase the average access time?

$x \cdot 6ns + (1 - x) \cdot (100ns + 6ns) = 15ns$
$x = 0.91$

3.2.3 Suppose the cache access time is instead increased to 10 ns. What is the minimum hit ratio needed in order to not increase the average access time?

$x \cdot 10ns + (1 - x) \cdot (100ns + 10ns) = 15ns$
$x = 0.95$

3.3 What is the average time in the following system

9ns cache
80ns main memory
8ms from disk to main memory
cache miss rate 9%
main memory mis rate 30%
$0.91 \cdot 9ns + 0.09 \cdot ((0.7 \cdot 80ns + 0.3 \cdot (80ns + 8ms)) + 9ns) = 0.216ms$

3.4 A cache has a line size of 64 bytes. To determine which byte within a cache line an address points to, how many bits are in the Offset field?

$\log_2(64) = 6$

3.5 A two-way set-associative cache in a word addressable machine consist of 128 cache lines divided into several sets. The main memory contain 8 K (8192) block of size 256 words. Show and explain the format of main memory addresses

cache size: 128 lines
MM size: $8192 \cdot 256 = 2097152$ words
block size: 256 words
block size $= 2^x = 256 \rightarrow x =$ offset $= 8$
MM size $= 2^z = 2097152 \rightarrow z =$ physical address bits $= 21$
Since the set contains 2 lines since it is a two-way set
Number of lines $= 128 = 2^y \rightarrow y = 7$
Therefore set number is line number /2 $2^7/2 = 2^6$ making the set number equal to 6.
Tag size = physical address bits - offset - set number = 21 - 8 - 6 = 7

3.6 Calculate the cache

En 32-bit maskine har en cache med 32 indgange (”cache lines”), hver på 16 bytesdata. En adresse er på 32 bit og adresserer de enkelte bytes.Cachen er organiseretsom en 2-vejs cache (2-way set associative).
Line numbers: 32 lines $= 2^5$
MM address: 32 bit system $= 2^32$
Block size: 16 byte $= 2^4$
offset = log(blocksize = 4 bit
set number bits = index = log(line numbers)/2=4 bit
tag = log(MM) - offset - index = 24

3.7 **Using the same cache system from last exercise check if the following gives a hit or miss**

indeks	V	T	D	V	T	D
0	1	2	x	1	AA	x
1	1	29FF	x	1	7600	x
2	0	F5	x	1	4	x
3	1	39E0	x	1	1210	x
4	1	2221	x	0	443B	x
5	1	60	x	0	1BBB	x
6	0	60	x	0	61	x
7	1	1210	x	0	61	x
8	0	60	x	0	61	x
9	0	76	x	1	61	x
10	0	76F	x	1	D59	x
11	0	0	x	0	0	x
12	1	1210	x	1	7701	x
13	1	1210	x	0	222A	x
14	1	1210	x	0	223C	x
15	1	1210	x	1	100A	x

Tilfælde	Læs/skriv	Antal byte	Adresse
A:	læs	4	00076FA4
B:	skriv	4	00121070
C:	læs	2	000D59C6
D:	læs	4	00000428
E:	læs	2	00021080

A index 10 no longer valid hit
B index 7 still valid hit
C index 12 miss
D index 2 still valid
E index 8 miss