《计算机组成与设计》
书后作业

姓名: 刘韬
学院: 竺可桢学院
专业: 人工智能
邮箱: 3220103422@zju.edu.cn

日期: 2024 年 4 月 4 日

chapter1

1.1

1. PC:personal computer , 个人计算机，是一种供个人使用的小型计算机。它的特点是体积小，价格相对便宜，操作简单，功能强大。
2. server: 服务器，是一种提供服务的计算机，通常用于存储数据、提供网络服务等。
3. embedded computer: 嵌入式计算机，是一种专用计算机，通常用于控制设备、嵌入到其他设备中。
4. supercomputer: 超级计算机，是一种计算能力极强的计算机，通常用于科学计算、模拟等。

1.2

- a. Performance via Pipelining
- b. Dependability via Redundancy
- c. Performance via Prediction
- d. Make the Common Case Fast
- e. HierarchyofMemories
- f. PerformanceviaParallelism
- g. DesignforMoore'sLaw
- h. UseAbstractiontoSimplifyDesign

1.4

- a. 1280 × 1024 pixels = 1,310,720 pixels => 1,310,720 × 3 = 3,932,160 bytes/ frame.
- b. 3,932,160 bytes×(8 bits/byte)/100e6 bits/second = 0.31 seconds

1.6

- CPI(P1) = $\frac{1\times1e5+2\times2e5+3\times5e5+3\times2e5}{1e6} = 2.6$
- CPI(P2) = $\frac{2\times1e5+2\times2e5+2\times5e5+2\times2e5}{1e6} = 2.0$
- clock cycles(P1) = $1e6 \times 2.6 = 2.6e6$
- clock cycles(P2) = $1e6 \times 2.0 = 2.0e6$

1.7

1. CPI = $\frac{T_{\text{exec}}\times \text{时钟速率}}{\text{Num}_{\text{instr}}}$
   - compiler A CPI = 1.1
   - compiler B CPI = 1.25
2. $\frac{f_B}{f_A} = \frac{T_{\text{execB}}}{T_{\text{execA}}} = 1.36$
3. $T_{\text{new}} = \frac{6.0e8\times1.1}{1e9} = 0.66s$
   - $\frac{T_A}{T_{\text{new}}} = 1.67$

◊ $\frac{T_B}{T_{\text{new}}} = 2.27$

1.14

1.14.1

Clock cycles = 5.12e8, $T_{\text{CPU}} = \frac{\text{Clock cycles}}{\text{clock rate}} = 0.256s$

假设要提升到 2 倍，那么 Clock cycles 变为 1/2，那么$\text{CPI}_{\text{improved}} = \frac{256-462}{50} < 0$这不可能

1.14.2

$\text{CPI}_{\text{improved}} = \frac{256-198}{80} = 0.725$

1.14.3

T_{CPU} (before improv.) = 0.256 s

T_{CPU} (after improv.) = 0.1712 s

chapter2

2.3

```
sub x30, x28, x29   # x30 = i - j
slli x30, x30, 3     # x30 = (i - j) * 8
add x30, x30, x10   # x30 = &A[i-j]
ld x30, 0(x30)      # x30 = A[i-j]
sd x30, 64(x11)     # B[8] = A[i-j]
```

2.4

```
B[g]= A[f] + A[f+1]
```

2.5

Address	little-endian	big-endian
0x00	0x12	0xab
0x01	0xef	0xcd
0x02	0xcd	0xef
0x03	0xab	0x12

2.11

2.11.1

128 + x6 > 2^{63} −1 即 x6 > 2^{63} - 129

或者 128 + x6 < -2^{63} 即 x6 < −128 - 2^{63}，这不成立

2.11.2

128 - x6 > 2^{63} −1 即 x6 < 129 - 2^{63}

或者 128 - x6 < -2^{63} 即 x6 > 128 + 2^{63}，这不成立

2.11.3

x6 - 128 > 2^{63} −1 即 x6 > 127 + 2^{63} 这不成立

或者 x6 - 128 < -2^{63} 即 x6 < 128 - 2^{63}

2.12

R-type: add x1, x1, x1

2.13

sd x5,32(x30)

S-type:0x025F3023

2.23

2.23.1

The UJ instruction format would be most appropriate

2.23.2

```
loop:
    addi x29, x29, -1
    bgt x29, x0, loop
    addi x29, x29, 1
```

2.24

2.24.1

20

2.24.2

```
acc = 0, i =10;
while(i != 0)
{
  i = i - 1;
  acc = acc + 2;
}
```

2.24.3

4*N + 1

2.24.4

```
acc = 0, i =10;
while(i >= 0)
{
  i = i - 1;
  acc = acc + 2;
}
```

2.25

```
addi x7, x0, 0
LoopI:
    bge x7, x5, LoopJ
    addi x30, x10, 0
    addi x29, x0, 0
LoopJ:
    bge x29, x6, EndJ
    add x31, x7, x29
    slli x32, x29, 2
    add x32, x32, x30
    sd x31, 0(x32)
    addi x29, x29, 1
    jal x0, LoopJ
EndJ:
    addi x7, x7, 1
    jal x0, LoopI
EndI:
```

2.26

13 条 risc-v 指令， $12 \times 10 + 3 = 123$

2.31

```
```

```
f:
    addi x2, x2, -16
    sd x1, 0(x2)
    add x5, x12, x13
    sd x5, 8(x2)
    jal x1, g
    ld x11, 8(x2)
    jal x1, g
    ld x1, 0(x2)
    addi x2, x2, 16
    jalr x0, x1
```

2.35

2.35.1

0x11

2.35.2

0x88

2.36

```
lui x10, 0x11223
addi x10, x10, 0x344
slli x10, x10, 32
lui x5, 0x55667
addi x5, x5, 0x788
add x10, x10, x5
```

2.40

2.40.1

$2 \times 0.7 + 6 \times 0.1 + 3 \times 0.2 = 2.6$

2.40.2

CPI = $2.6 \times 0.75 = 1.95$

$\frac{1.95 - 6 \times 0.1 - 3 \times 0.2}{0.7} = 1.07$

2.40.3

CPI = $2.6 \times 0.5 = 1.3$

$\frac{1.3 - 6 \times 0.1 - 3 \times 0.2}{0.7} = 0.14$

chapter3

3.1

5730

3.5

$4365 - 3412 = (753)_8$

3.8

63,不溢出

3.9

151 + 214 = 365, 溢出 饱和值为 255

3.13

iteration	step	Multiplier	Multiplicand	Product
0	Initial values	1100	0011 1110	0000 0000
1	1: 0 ⇒ No operation	1100	0011 1110	0000 0000
	2: Shift left Multiplicand	1100	0111 1100	0000 0000
	3: Shift right Multiplier	0110	0111 1100	0000 0000
2	1: 0 ⇒ No operation	0110	0111 1100	0000 0000
	2: Shift left Multiplicand	0110	1111 1000	0000 0000
	3: Shift right Multiplier	0011	1111 1000	0000 0000
3	1a: 1 ⇒ Prod = Prod + Mcand	0011	1111 1000	1111 1000
	2: Shift left Multiplicand	0011	0001 1111 0000	0111 1100
	3: Shift right Multiplier	0001	0001 1111 0000	0111 1100
4	1a: 1 ⇒ Prod = Prod + Mcand	0001	0001 1111 0000	0010 1111 0100
	2: Shift left Multiplicand	0001	0011 1110 0000	0010 1111 0100
	3: Shift right Multiplier	0000	0011 1110 0000	0010 1111 0100

3.19

iteration	step	divisor	remainder
0	Initial values	0001 0101	0000 0000 0100 1010
	Shift remainder left 1	0001 0101	0000 0000 1001 0100
1	1. remainder=remainder-divisor	0001 0101	1110 1010 1001 0100
	2b: remainder < 0 ⇒ + divisor, sll R, $R_0 = 0$	0001 0101	0000 0001 0010 1000
2	1. remainder=remainder-divisor	0001 0101	1110 1100 0010 1000
	2b: remainder < 0 ⇒ + divisor, sll R, $R_0 = 0$	0001 0101	0000 0010 0101 0000
3	1. remainder=remainder-divisor	0001 0101	1110 1101 0101 0000
	2b: remainder < 0 ⇒ + divisor, sll R, $R_0 = 0$	0001 0101	0000 0100 1010 0000
4	1. remainder=remainder-divisor	0001 0101	1110 1111 1010 0000
	2b: remainder < 0 ⇒ + divisor, sll R, $R_0 = 0$	0001 0101	0000 1001 0100 0000
5	1. remainder=remainder-divisor	0001 0101	1110 0100 0100 0000
	2b: remainder < 0 ⇒ + divisor, sll R, $R_0 = 0$	0001 0101	0001 0010 1000 0000
6	1. remainder=remainder-divisor	0001 0101	1111 1101 1000 0000
	2b: remainder < 0 ⇒ + divisor, sll R, $R_0 = 0$	0001 0101	0010 0101 0000 0000
7	1. remainder=remainder-divisor	0001 0101	0001 0000 0000 0000
	2a: remainder >0 ⇒ sll R, $R_0 = 1$	0001 0101	0010 0000 0000 0001
8	1. remainder=remainder-divisor	0001 0101	0000 1011 0000 0001
	2a: remainder >0 ⇒ sll R, $R_0 = 1$	0001 0101	0001 0110 0000 0011
	Shift left half of remainder right 1	quotient = 0000 0011	remainder = 0000 1011

3.20

都是 201326592，正数的补码是本身，等于无符号整数

3.21

0x0C000000 = 0000 1100 0000 0000 0000 0000 0000 0000 =0000110 00000 000 00000 000000000000

lb x0 0(x0)

3.22

0x0C000000 = 0000 1100 0000 0000 0000 0000 0000 0000

=0 00011000 00000000000000000000000

$=(-1)^0 \times (1+0) \times 2^{24-127} = 2^{-103}$

3.23

$63.25 = 111111.01 \times 2^0 = 1.1111101 \times 2^5$

sign = 0, exp = 127 + 5 = 132 = 10000100, frac = 11111010000000000000000

single float = 0100 0010 0111 1101 0000 0000 0000 0000 = 0x427D0000

3.24

$63.25 = 111111.01 \times 2^0 = 1.1111101 \times 2^5$

sign = 0,exp = 1023 + 5 = 1028 = 10000000100, frac = 1111101000000000000000000000000000000000000000000000

double float = 0100 0000 0100 1111 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 = 0x404FA00000000000

3.27

$-1.5625 \times 10^{-1} = -0.15625 \times 10^0 = -0.00101 \times 2^0 = -1.01 \times 2^{-3}$

sign = 1,exp = 15 - 3 = 12 = 01100, frac = 0100000000

half float = 1011 0001 0000 0000

◊ 半精度的表示范围，假设是规格数

- ▹ 最小
  - exp = 00001 actual exp = −14 frac = 0000000000
  - $\pm 1.0 \times 2^{-14} \approx \pm 6.1 \times 10^{-5}$
- ▹ 最大
  - exp = 11110 actual exp = 15 frac = 1111111111
  - $\pm 2.0 \times 2^{15} \approx \pm 65536$

3.29

$2.6125 \times 10^1 + 4.150390625 \times 10^{-1}$

$2.6125 \times 10^1 = 11010.001 \times 2^0 = 1.1010001 \times 2^4$

$4.150390625 \times 10^{-1} = 0.011010100111 = 1.11010100111 \times 2^{-2}$

然后对齐到小数点，把小的左移六位，相加得到

$1.1010100011 \times 2^4 = 11010.100011 \times 2^0 = 26.546875 = 2.6546875 \times 10^1$