3.41 $-\frac{1}{4} = -2^{-2} = (-0.01)_2$

标准化绝对值，有 $(0.01)_2 = 1.0 \times 2^{-2} = 1.0 \times 2^{125-127}$

符号 = 1，指数 = 125 = 01111101，

小数 = 000 0000 0000 0000 0000 0000

故最终形式为

1	0111 1101	000 0000 0000 0000 0000 0000

3.44 $\frac{1}{3} = 0.333333\cdots$，由于 $(3)_{10} = (0011)_2$

故24位BCD编码为 0011 0011 0011 0011 0011 0011

这不是一个精确的表达，因为我们截断了太多小数

3.47 设计指令：load_128/store_128 与128位寄存器读取

mul_64, a, b, c　b与c寄存器的最低每16位相乘(共4组16位数)，结果依次存入a中.

sum_r, a, b　b中以16位为单位(8个16位)，计算8个16位数的和，放入a的最低位

rshift/lshift_16　a右移/左移16位

将f数组存入f寄存器的f[63:0]，将sig-in存入(每8个的存)

执行：
```
mul_64   temp_mul , sig_in . f
sum_r    temp , temp_mul
lshift_16   temp
rshift_16   sig-in
```
上述代码执行8次，可以算出 sig_out 的8个输出，再重新算 sig-in，循环LOOP，直到128位 sig_out算出

4.2 (1) 它执行ALU，所以可以使用ALU单元；

可以使用PC和指令存储器单元，数据存储器，寄存器单元，三个多路复用器单元

(2) 不需要添加功能单元

(3) 不需要新的信号，只需令 RegWrite=1, MemRead=1, MemWrite=0, ALU mux=0, REG mux=1, Branch=0 即可

4.4 (1) 有两条路径：① PC → ADD → PC ， 延时为 70 ps

② PC → I-Mem ， 延时为 200 ps

故时钟周期为 200 ps

(2) 路径如下： PC → I-mem → Sign → extend → Shift-left-2

→ Add → Mux → PC

故延时为 200+15+10+70+20 = 315 ps

(3) 路径： PC → I-mem → Regs → Mux → ALU → Mux → PC

因此延时为 200+90+20+90+20 = 420 ps

(4) 所有分支和跳转指令使用该单元

(5) 对无条件分支和跳转指令，该单元位于关键路径上

(6) add 指令不使用该单元，因此延迟变化对 add 执行时间无影响

beq 是条件分支指令，对比 (2) 和 (3) 结果可知

延时可以增加 420−315 = 105 ps ，而不会表现更差.

4.7. opcode 是头 6 位：101011

因此 opcode = $1\times2^5+1\times2^3+1\times2+1=43$ ，也是一个 sw 指令

rs 是接下来 5 位： 00011 = 3 ， rt 是之后 5 位： 00010 = 2

剩下是地址： 0000 0000 0001 0100 = 20

(1) 符号拓展单元输出: 0000 0000 0000 0000 0000 0000 0001 0100

左移两位单元输出: 0000 0000 0000 0000 0000 0000 0101 0100

(2) ALU 控制单元的输入是 0010

(3) 由于无跳转/分支指令，故执行后 PC 值是执行前的 4 倍

路径： PC → 第一个 Add unit → top-right Mux → PC

(4) 由于无 write register ，故 mux 输出给 write register 的值未定义

Mux 输出给 ALU 的地址值: 0x00000014

Mux 输出给 Data Memory 的值未定义

两个 top-right Muxes 输出为 PC + 4

(5) 第一个加法器输入为 PC 和 4

第二个加法器输入为： ① 第一个加法器的输出，即 PC+4

② 左移 2 位单元的输出，为 0x00000050

对于 ALU 第一个输入是 rs，因此第一个输出为 r3 = -3

用(4)，第二个输入为 0x00000014

(6) 由于 Write register 和 write data 未定义，read register 1 是 rs=3

read register 2 是 rt=2，contral signal 为 0

4.8 (1) lw 指令需要5个阶段

时间为 250+350+150+300+200 = 1250 ps

因此非流水线处理器时钟周期为 1250 ps

对于流水线寄存器，时钟周期取决于最长延时，为 350 ps

(2) lw 指令在非流水线处理器延迟为 1250 ps，同(1)，

在流水线处理器时延迟为 5×350 = 1750 ps

(3) 由于划分 ID 之外的级不会改变延迟，故我们划分 ID，

故每级 350÷2 = 175 ps

故时钟周期为 300 ps.

(4) 数据存储器只被 sw 和 lw 指令使用，

因此，利用率为 20% + 15% = 35%

(5) 写寄存器端口只被 alu 和 lw 指令使用

故利用率：45% + 20% = 65%

(6) 多周期指令时间等于 350 ps.

故时钟周期比 $\frac{t_{单周期}}{t_{多周期}} = \frac{1250}{350} = 3.57$

总执行时间比 $\frac{t_{多}}{t_{单}} = \frac{\text{时钟周期}_{多} + 3.2 \times \text{时钟周期}_{多}}{\text{时钟周期}_{单}} = 4.2$