

COD WEEK5

1. T1

1.1 (1)

题目 1. 计算题：假设能保存5位有效数位并且指数的表示范围与IEEE 754标准单精度浮点数相同

1.将1.062510和-0.687510用十进制科学计数法表示后计算浮点加法

$$1.0625 \times 10^0 + (-6.875 \times 10^{-1})$$

$$= 0.3750 \times 10^0$$

$$= 3.7500 \times 10^{-1}$$

1.2 (2)

2.将1.062510和-0.687510用二进制科学计数法表示后计算浮点加法

$$1.0625 = 1.0001 \times 2^0$$

$$-0.6875 = (-1.0110 \times 2^{-1})$$

$$\text{相加得: } 0.0110 \times 2^0 = 1.1000 \times 2^{-2}$$

2. T2

题目 2. 假设用来实现处理器数据通路的各功能模块延迟如下所示：

I-Mem / D-Mem	Register File	Mux	ALU	Adder	Single gate	Register Read	Register Setup	Sign extend	Control
250ps	150ps	25ps	200ps	150ps	5ps	30ps	20ps	50ps	50ps

其中，寄存器读延迟指的是，时钟上升沿到寄存器输出端稳定输出新值所需的时间。该延迟仅针对PC寄存器。寄存器建立时间指的是，寄存器的输入数据稳定到时钟上升沿所需的时间。该数值针对PC寄存器和寄存器堆。

1. R型指令的延迟是多少？
2. ld指令的延迟是多少？
3. sd指令的延迟是多少？
4. beq指令的延迟是多少？
5. I型指令(不考虑存储器)的延迟是多少？
6. 该CPU的最小时钟周期是多少？

- R: $30\text{ps}+250\text{ps}+150\text{ps}+25\text{ps}+200\text{ps}+25\text{ps}+20\text{ps}=700\text{ps}$
- ld: $30\text{ps}+250\text{ps}+150\text{ps}+200\text{ps}+250\text{ps}+25\text{ps}+20\text{ps}=925\text{ps}$
- sd: $30\text{ps}+250\text{ps}+150\text{ps}+200\text{ps}+250\text{ps}=880\text{ps}$
- beq: $30\text{ps}+250\text{ps}+150\text{ps}+25\text{ps}+200\text{ps}+5\text{ps}+25\text{ps}=685\text{ps}$
- I: $700\text{ps}-25\text{ps}=675\text{ps}$
- 925s

3. 实验题

假设从t0所保存的地址开始的八个字节存储了一个IEEE 754标准的双精度浮点数，使用RV32I指令读取该双精度浮点数，将其转换为IEEE 754标准的单精度浮点数后 存储在t1所指向的地址中。（尾数部分可以直接截断，不用考虑指数部分的溢出，双精度浮点数以小端序存储）

```
1  lw t2, 0(t0)      # 从t0所指向的地址读取双精度浮点数的低4字节
2  lw t3, 4(t0)      # 从t0所指向的地址读取双精度浮点数的高4字节
3
4
```

```
5  srli t4, t3, 20      # 将双精度的高4字节右移20位，得到单精度浮点数的指数部分
6  andi t5, t4, 0x7ff  # 从单精度的指数部分中提取11位指数位
7  addi t5, t5, -1023  # 减去双精度的偏置值
8  addi t5, t5, 127    # 加上单精度的偏置值，得到单精度浮点数的指数部分
9  slli x11, t5, 23    # 将指数部分左移23位，得到单精度指数部分
10
11 srli x12, t3, 31     # 获取双精度符号位
12 slli x12, x12, 31    # 得到单精度符号位
13 or t5, t5, x12      # 符号位和指数部分合并
14
15 slli t3, t3, 12      # 将双精度的高4字节左移12位，得到双精度的尾数部分的前20位
16 srli t3, t3, 9       # 将尾数部分的前20位右移9位，得到单精度的尾数部分的前20位
17 srli t2, t2, 29      # 将双精度的低4字节右移29位，得到后三位
18 add t3, t3, t2       # 后三位与前20位合并，得到尾数部分
19
20 or t4, x11, t3       # 符号位和指数部分和尾数部分合并，得到单精度浮点数
21
22 sw t4, 0(t1)        # 将单精度存储到t1所指向的地址中
```