指令码

|  |  |  |
| --- | --- | --- |
| 指令（4位） | 激活函数 | 说明 |
| 0000 |  | 一阶段，基本算子支持 取反（浮点乘） （浮点加）浮点除 e指数 |
| 0001 |  | 一阶段，支持正负值分开操作 浮点常数乘法 |
| 0010 |  | 二阶段，支持归0 |
| 0011 |  | 二阶段，基于基础算子，基本激活函数功能添加 |
| 0100 |  | 三阶段，浮点除法（双操作数）硬件优化 |
| 0101 |  | 三阶段，浮点除法（双操作数）硬件优化 |
| 0110 |  | 四阶段，支持ln对数运算 |
| 0111 |  | 浮点乘法（双操作数）目前的最终目标 |
| …… | …… | …… |

注释：

1. 目前没有想到解决办法的函数
2. 需要考虑电路级别优化 尤其是 单双操作数的 除法和乘法
3. 电路中的寄存器如何优化，所有算子单独寄存器？所有算子复用寄存器？

操作码

|  |  |  |  |
| --- | --- | --- | --- |
| 操作码 | 操作 | 表示 | 系数 |
| 0000 | 正负判断 |  |  |
| 0001 | 取反 |  |  |
| 0010 |  |  |  |
| 0011 | E指数 |  |  |
| 0100 | 浮点除（单） |  | 有 |
| 0101 | 浮点乘（单） |  | 有 |
| 0110 | 归0 |  |  |
| 0111 | 浮点除（双） |  |  |
| 1000 | 浮点乘（双） |  |  |
| 1001 |  |  |  |

指令码与操作码

指令码操作码+运算系数+控制信号

硬件上，会有单独的解码模块对于指令码进行解码，基于以下考虑：

1. 外部指令是定长的，同时减少指令位，并具有可扩展性
2. 指令码与操作码的对应关系可以已查找表的形式进行存储，在运行时可以用FIFO进行实现

根据一阶段目标，实现两个基本函数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 基本算子 |  |  |  |  |
| 指令码 | 0001 | 0011 | 0010 | 0100 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 基本算子 |  |  |  |  |  |
| 指令码 | 0000 | 0101 | 0011 | 0010 | 0101 |

硬件设计 主要设计任务

1. 基本算子模块： 输入：16\*n 数据，start，系数，正负掩膜

输出：16\*n 数据，complete

1. 指令解码模块，操作码查找表，操作码分配，控制信号
2. 激活函数数据通路：主要多路选择器和start与complete信号控制
3. 整体ALU控制通路：输入寄存器，输出寄存器，中间结果寄存器（复用问题），多路选择器
4. 控制计数器，用于生成整体控制信号（暂定）