

LEG

目录

- [LEG工作流程](#)
 - [指令读取形式](#)
 - [4字节指令格式](#)
- [操作码功能](#)
 - [ALU功能](#)
 - [立即数](#)
 - [IF跳转](#)
 - [操作码资源使用情况](#)
 - [已使用](#)
- [地址参数含义](#)

LEG工作流程

指令读取形式

LEG每个时刻钟从存储器中读取4个字节并执行对应的指令

4字节指令格式

1. 第1个字节描述操作 (OPCODE)
2. 第2个字节是参数 (地址)
3. 第3个字节是参数 (地址)
4. 第4个字节是结果参数 (地址)

操作码功能

ALU功能

- 0 : ADD (加法)
- 1 : SUB (减法)
- 2 : AND (按位与)
- 3 : OR (按位或)
- 4 : NOT (按位非) 【只对第2字节地址参数起作用】
- 5 : XOR (按位异或)

- 共占用3个bits，占用第1、2、3三位，剩余6、7两个值

立即数

操作码的第8位为高电平时，将第2字节作为立即数，而不是地址值

操作码的第7位为高电平时，将第3字节作为立即数，而不是地址值

- 共占用2个bits，占用第8、7两位

IF跳转

- 之前的4字节指令格式
 1. 第1个字节描述操作 (OPCODE)
 2. 第2个字节是参数 (地址)
 3. 第3个字节是参数 (地址)
 4. 第4个字节是结果参数 (地址)
- 现在的4字节指令格式
 1. 第1个字节描述操作 (OPCODE)
 2. 第2个字节是参数 (地址)
 3. 第3个字节是参数 (地址)
 4. 第4个字节是跳转地址 (地址)

32 : IF_EQUAL (等于)

33 : IF_NOT_EQUAL (不等于)

34 : IF_LESS (小于)

35 : IF_LESS_OR_EQUAL (小于等于)

36 : IF_GREATER (大于)

37 : IF_GREATER_OR_EQUAL (大于等于)

比较内容是无符号整数

- 共占用4个bits，占用第6、1、2、3四位，剩余38、39两个值

操作码资源使用情况

8bits操作码，0~255共256个值

已使用

1. ALU功能：0~5，共6个
2. 立即数：128和64，共2个
3. if跳转：32~37，共6个

地址参数含义

0 : 0号寄存器

1 : 1号寄存器

2 : 2号寄存器

3：3号寄存器

4：4号寄存器

5：5号寄存器

6：计数器

7：输入/输出元件（第2、3字节对应输入，第4字节对应输出）