

【成绩】

【教师签名】

【实验目的】

1. 掌握循环操作指令的运用;
2. 掌握用汇编语言编写DSP程序的方法;
3. 掌握(用汇编) TMS320C54x中的双操作数指令;
4. 掌握TMS320C54x中的并行运算指令;
5. 掌握TMS320C54x中的长字指令。

【实验原理及内容】

1. 循环操作: TMS320C54x具有丰富的程序控制与转移指令,利用这些指令可以执行分支转移、循环控制以及子程序操作。本实验要求编写一程序完成 $y = \sum_{i=0}^{N-1} x_i$ 的计算。这个求和运算可以通过一个循环操作指令BANZ来完成。BANZ的功能是当辅助寄存器的值不为0时转移到指定标号执行。
2. 双操作数乘法: TMS320C54x片内的多总线结构,允许在一个机器周期内通过两个16位数据总线(C总线和D总线)寻址两个数据和系数。双操作数指令是用间接寻址方式获得操作数的,并且只能用AR2到AR5的辅助寄存器。双操作数指令是用间接寻址方式获得操作数,双操作数指令占用较少的程序空间,而获得更快的运行速度。
3. 并行运算: TMS320C54x片内有1条程序总线,3条数据总线和4条地址总线。这3条数据总线(CB、DB和EB)将内部各单元连接在一起。其中,CB和DB总线传送从数据存储器读出的操作数,EB总线传送写入存储器中的数据。并行运算就是同时利用D总线和E总线。并行指令有并行加载和乘法指令,并行加载和存储指令,并行存储和乘法指令,以及并行存储和加/减法指令4种。所有并行指令都是单字单周期指令。并行运算时存储的是前面的运算结果,存储之后再行加载或算术运算,这些指令都工作在累加器的高位,且都受ASM影响。
4. 长字运算: 长操作数指令中的一个重要问题是,高16位和低16位操作数在存储器中如何排列。一般情况下,高16位操作数放在存储器中的低地址单元,低16位操作数放在存储器中的高地址单元。例如一个长操作数16782345H,它在存储器中的存入方式是: (0060H)=1678H(高字), (0061H)=2345H(低字)。

【实验设备】

1. 一台装有CCS软件的计算机。

【实验方案及步骤】

一、循环操作:

1. 编译前: ①启动CCS, 选择模拟CPU (C5416 Device Simulator);
②打开工程文件, 打开".asm"文件和".cmd"文件;
2. 编译和单步调试:
① Build工程, 编译无错误后, 选择"Load Program"命令装载程序, 然后打开存储器数据窗口以及CPU寄存器窗口。
② 单步运行程序, 观察寄存器和存储器的变化, 理解循环指令的三条语句, 注意其3个要点。
③ 改由重复指令实现循环操作, 重新Build工程、装载程序运行后得到与之前相同的结果, 记录原始语句、修改后的语句, 并记录运行后变量x、y的地址和数值, 以及寄存器的变化情况。

二、双操作数乘法:

1. 打开工程文件, Build工程、装载程序、运行观察结果。
2. 改为单操作数乘加指令来实现:
① 先将其中一个操作数装入T寄存器中; ② 去掉原双操作数MAC指令中装入T寄存器的那个操作数; ③ 用BANDZ指令重新构造循环。
3. 重新运行程序后记录观察变量x、y、z的地址和数值。

三、并行运算:

1. 打开工程文件, Build工程、装载程序、运行观察结果。
2. 改用非并行指令来实现:
① 去掉分隔符, 将一条并行指令变成2条指令; ② 修改为使用累加器的高位。
3. 观察记录寄存器、存储器的变化。

四、长字运算:

1. 打开工程文件, Build工程、装载程序、运行观察结果。
 2. 改为单字指令来实现:
① 每条长字指令改为2条单字指令实现; ② 左移16位装入高位; ③ 用ADDS加法指令实现低位的装入。
3. 运行后记录哪些寄存器、存储器的数值发生变化。
- ### 五、分析实验结果, 完成实验报告。

【实验电路图】

【实验数据处理及分析】

1. 循环操作:

程序运行后, 辅助寄存器 AR2 的值随着循环次数的增加而减少, 因为在循环开始前装入了循环次数 4 在 AR2 中, 循环结束后又用 BANZ 指令使 AR2 的值减一后又进行下一次循环, 直到 AR2 中的值为零时跳出循环; 而 AR3 则存放每一次循环的累加值, 因此是递增的。

2. 并行运算:

将一条并行指令 "ST A, *AR5 || LD *AR2+, B" 改为非并行指令, 不仅需要去掉分隔符 "||", 还需把使用累加器的高位的形式表示出来变成 "STH A, *AR5"。

3. 长字运算:

将长字指令 "DLD *(X), A"
DADD *(Y), A
DST A, *(Z)

LD *(X), 16, A
ADD *(X+1), A
ADD *(Y), 16, A
ADD *(Y+1), A
STH A, *(Z)
STL A, *(Z+1)

【实验结论】

1. 将循环操作指令改为重复指令后, 虽然实现逻辑相似, 但代码结构会更加清晰, 易于阅读和维护。
2. 将双操作数乘法指令改为单操作数乘法指令可以简化某些计算场景。例如, 需要计算一个数的平方时, 单操作数乘法指令可以直接完成, 而无需额外的寄存器来暂存其中一个操作数。但对于一般的乘法运算, 双操作数乘法指令更为适用。
3. 将并行运算指令改为非并行运算指令, 会增加流水线中的依赖关系, 可能导致流水线停滞, 降低处理器的指令级并行度, 进而影响执行效率。
4. 使用长操作数指令时, 一般高16位操作数放在存储器中的低地址单元, 低16位操作数放在存储器中的高地址单元。

【思考题】

1. 在§3.1中, 循环控制是在循环体结束处通过指令“BNZ loop, *AR2-”检测循环变量实现的, 如改为在循环体开始处检测循环变量, 如何修改?

答: 可以在“STL A, *(y)”指令前加一个“end-loop”标签, 然后在循环体的上一条指令变为“BNZ *AR2, end-loop”, 当AR2不为零时就会结束循环。

2. 双操作数指令和并行指令说明了TMS320C54X的总线结构有何特点?

答: ① 多总线结构, 允许在一个机器周期内通过两个16位数据总线(C总线和D总线)寻址两个数据和系数;

② 片内有1条程序总线, 3条数据总线和4条地址总线。这3条数据总线(CB、DB和EB)将内部各单元连接在一起。其中, CB、DB总线传送从数据存储器读出的操作数, EB总线传送写到存储器中的数据;

③ 还有数据预取机制、高效的内存访问、指令与数据分离等特点。