

Introduction to Computer System Organization

SZU Review

Chapter7

汇编语言程序设计

1. 分析程序

2. 设计程序

伪指令

.ORIG

格式: .ORIG address

功能: 指定程序起始地址

e.g. .ORIG x3000

.BLKW

格式: LABEL .BLKW n

功能; 分配n个连续的内存 (16 bit)

e.g ARRAY .BLKW 10 ;分配10个字的空间

.FILL

格式: LABEL.FILL value

功能: 用指定值填充一个内存位置

e.g. NUM.FILL #42 ;定义一个值为42的内存位置

.END

格式: .END

功能: 标记程序结束

e.g. .END ;程序结束

.STRINGZ

格式: LABEL .STRINGZ "string"

功能: 分配连续的字符并以null结尾

e.g. MSG.STRINGZ "Hello"; 定义以null结尾的字符串"Hello\n".

(etc.)

汇编语言的程序格式 (4 types)

1. 主程序

```
.ORIG x3000  
  
code area  
  
HALT  
  
data definition area  
  
.END
```

子程序分为系统调用(TRAP)和用户调用(JSR/JSRR)(中断子程序在Chapter8讨论, 此处不作介绍)

2. 子程序(系统调用)

```
.ORIG      X0150;起始地址  
  
store area  
  
function area  
  
restore area  
  
RET  
  
data definition area  
  
.END
```

3. 用户调用:

```
子程序名  
  
store area  
  
function area  
  
restore area
```

```
RET
```

```
data definition area
```

comment: 子程序应置于主程序数据区

e.g.

Quesiton:

设计一个程序"SUMVEC"实现向量减法:

Explain:

入口参数:

```
R0  --> 数组指针
```

```
R1  --> 数组大小
```

返回参数:

```
R2
```

exp:

```
SUMVEC ;程序名
```

```
ST R0,SAVER0
```

```
ST R1,SAVER1
```

```
ST R3,SAVER3 ;存储区
```

```
AND R2,R2,#0
```

```
ADD R1,R1,#0
```

```
BRnz DONE
```

```
LOOP LDR R3,R0,#0
```

```
ADD R2,R2,R3
```

```
ADD R0,R0,#1
```

```

        ADD R1,R1,#-1

        BRp LOOP      ;实现区

LD   R0,SAVER0

LD   R1,SAVER1

LD   R3,SAVER3      ;恢复区

DONE    RET

SAVER0.FILL #0

SAVER1.FILL #0

SAVER3.FILL #0      ;数据定义区

```

comment:

1. 入口参数以及除返回值以外的寄存器,若值被修改,则需"存储--恢复";
2. 返回值寄存器一定不能"存储--恢复";
3. 调用子程序时所使用的JSR指令或JSRR指令,将默认值用寄存器R7保存返回地址,故此时R7寄存器必须"存储--恢复"。

4. 汇编过程

汇编过程 --> 机器代码

1st Scanning :符号表

2nd Scanning :生成机器代码

5. 程序分析

题型: 给出程序功能和部分代码 --> 程序填空

comments: 形式: ①设计 ②分析 ③填空 ④代码(汇编语言) --> 符号表和机器代码