

4.0元

试题编号:

DSP

重庆邮电大学 2012—2013 学年 第 学期

《DSP 原理及应用》期末考试题 (A 卷)

题 号	一	二	三	四	五	六	七	八	总 分
得 分									
评卷人									

## 一、 填空题（每空 2 分，共 20 分）

- 1、在 C 语言和 C55x 汇编语言的混合程序设计中,C 函数的参数和返回值传递到 C55x 的寄存器中。在函数 “long func(int \*p1, int i2, int i3, int i4)” 中, \*p1 传递到 ARO 寄存器, i2 传递到 T0 寄存器, i4 传递到 AR1 寄存器, 返回值由 AC0 寄存器传递。
- 2、汇编语言 “mov \*AR0, AC0” 使用的寻址方式是 间接寻址模式, “mov #0x3, DPH” 使用的寻址方式是 直接寻址模式, “mov \*(#0x011234), T2” 使用的寻址方式是 绝对寻址模式。
- 3、TMS320C54 有两个通用引脚, BIO 和 XF, BIO 输入引脚可用于监视外部接口器件的状态; XF 输出引脚可以用于与外部接口器件的握手信号。
- 4、C55x 的链接器命令文件中, SECTIONS 命令的主要作用是 告诉链接器如何将输入段组合成输出段,以及在存储器何处存放输出。MEMORY 命令的主要作用是 定义目标系统的存储器配置图, 包括对存储器各部分的命名, 以及规定它们的起始地址和长度。

## 二、简述题（共 40 分）

1. 当 TMS320C54x CPU 接收到可屏蔽的硬件中断时, 满足哪些条件中断才能被响应? (6 分)

答: (1) 出现多个中断时, 此中断的优先级最高

(2) INTM=0 允许全局中断

(3) IMR 中的响应相应位为 1, 开放此中断。

1、TMS320C55x 芯片的总线结构有何特点，主要包括哪些总线？它们的功能是什么？（6 分）

答：TMS320C55x DSP 采用先进的哈佛结构并具有十二组总线，其独立的程序总线 and 数据总线允许同时读取指令和操作数，实现高度的并行操作。

采用各自分开的数据总线分别用于读数据和写数据，允许 CPU 在同一个机器周期内进行两次读操作数和一次写操作数。独立的程序总线 and 数据总线允许 CPU 同时访问程序指令和数据。

包括 12 条总线，分别是：PAB 和 PB、BAB 和 BB、CAB 和 CB、DAB 和 DB、EAB 和 EB、FAB 和 FB。

功能：

2、DSP 为了降低功耗采取了哪些措施？（6 分）

答：双电压供电；多种工作模式

3、TMS320C55x 的总存储空间为多少？可分为哪 3 类，它们的大小是多少？存储器空间的各自作用是什么？（6 分）

答：程序空间 16M Byte；I/O 空间 64K Words；数据空间 8M Words

4、TMS320C55x 有哪些寻址方式，它们是如何寻址的？试为每种寻址方式列举一条指令（6 分）

答：直接寻址模式，`mov #K16, DP;`

间接寻址模式，`mov *AR0, AC0;`

绝对寻址模式，`mov *(#0x011234), T2;`

MMR 寻址模式，`mov *abs16(#AR2), T2;`

寄存器位寻址模式，`btstp @30, AC1;`

圆形寻址模式。

5、将 C 源程序转换成可执行文件需要经过哪些步骤？（6 分）

6、

答：创建 C 源文件；创建工程文件；创建连接器命令文件；编译整个工程文件；链接；生成可执行文件

7、常用的 TMS320C55x 汇编命令有哪些，它们的作用是什么？（5 分）

汇编伪指令	作用	举例
.title	紧跟其后的是用双引号括起的源程序名	.title "example.asm"
.end	结束汇编命令	放在汇编语言源程序的最后
.text	紧随其后的是汇编语言程序正文	.text 段是源程序正文。经汇编后，紧随.text 后的是可执行程序代码
.data	紧跟其后的是已初始化数据	有两种数据形式：.int 和.word
.int	.int 用来设置一个或多个 16 位无符号整型量常数	table: .word 1, 2, 3, 4 .word 8, 6, 4, 2
.word	.word 用来设置一个或多个 16 位带符号整型量常数	表示在标号为 table 的程序存储器开始的 8 个单元中存放初始化数据 1、2、3、4、8、6、4 和 2，table 的值为第一个字的地址
.bss	.bss 为未初始化变量保留存储空间	.bss x, 4 表示在数据存储器中空出 4 个存储单元存放变量 x1、x2、x3 和 x4，x 代表第一个单元的地址
.sect	建立包含代码和数据的自定义段	.sect "vector" 定义向量表，紧随其后的是复位向量和中断向量，名为 vectors
.usect	为未初始化变量保留存储空间的自定义段	STACK .usect "STACK", 10h 表示在数据存储器中留出 16 个单元作为堆栈区，名为 STACK（栈顶地址）

### 三、程序设计题（共 40 分）

1、用 C55x 汇编语言实现计算  $y = \sum_{i=1}^4 a_i x_i$  的程序。（10 分）

答：mpym \*AR0+, \*AR1+, AC0

mpym \*AR0+, \*AR1+, AC1

add AC1, AC0

mpym \*AR0+, \*AR1+, AC1

add AC1, AC0

mpym \*AR0+, \*AR1+, AC1

add AC1, AC0

或者：

mpym \*AR0+, \*AR1+, AC0

||rpt #2

macm \*AR0+, \*AR1+, AC0

2、用 C55x 汇编语言实现计算  $y_1 = x_1 * a_1 - x_2 * a_2$  的程序。（10 分）

答：mpym \*AR0+, \*AR1+, AC3

masm \*AR0+, \*AR1+, AC3

3、以 8 点的 FFT 为例，用 C55x 的汇编语言实现 FFT 的算法，并且解释该算法的实现方法。（20 分）

答：.sect "fft\_code"

\_fft:

```
aadd #(ARGS-Size+1), SP      ; Adjust stack for local vars
mov  mmap(ST1_55), AR2       ; Save ST1, ST3
mov  mmap(ST3_55), AR3
mov  AR2, fft.d_ST1
mov  AR3, fft.d_ST3
btst @#0, T1, TC1            ; Check SCALE flag set
mov  #0x6340, mmap(ST1_55)    ; Set CPL, XF, SATD, SXAM, FRCT (SCALE=1)
mov  #0x1f22, mmap(ST3_55)    ; Set: HINT, SATA, SMUL
xcc  do_scale, TC1
mov  #0x6300, mmap(ST1_55)    ; Set CPL, XF, SATD, SXAM (SCALE=2)
```

do\_scale

```
mov  T2, fft.d_T2            ; Save T2
||  mov  #1, AC0
mov  AC0, fft.d_L             ; Initialize L=1
||  sfts AC0, T0               ; T0=EXP
mov  AC0, fft.d_N             ; N=1<<EXP
mov  XAR1, XCDP               ; CDP = pointer to U[]
mov  XSP, XAR4
add  #fft.d_temp, AR4         ; AR4 = pointer to temp
mov  XAR0, XAR1               ; AR1 points to sample buffer
mov  T0, T1
mov  XAR0, XAR5               ; Copy external bits to XAR5
```

```

outer_loop                                ; for (L=1; L<=EXP; L++)
    mov  fft.d_L, T0                      ; note: Since the buffer is
|| mov  #2, AC0                           ;      arranged in re, im pairs
    sfts AC0, T0                          ;      the index to the buffer
    neg  T0                               ;      is doubled
|| mov  fft.d_N, AC1                      ;      But the repeat coutners
    sftl AC1, T0                          ;      are not doubled
    mov  AC0, T0                          ; LE=2<<L
|| sfts AC0, #-1
    mov  AC0, AR0                         ; LE1=LE>>1
|| sfts AC0, #-1
    sub  #1, AC0                          ; Init mid_loop counter
    mov  mmap(AC0L), BRC0                 ; BRC0=LE1-1
    sub  #1, AC1                          ; Init inner loop counter
    mov  mmap(AC1L), BRC1                 ; BRC1=(N>>L)-1
    add  AR1, AR0
    mov  #0, T2                           ; j=0
|| rptblocal mid_loop-1                   ; for (j=0; j<LE1; j++)
    mov  T2, AR5                          ; AR5=id=i+LE1
    mov  T2, AR3
    add  AR0, AR5                          ; AR5 = pointer to X[id].re
    add  #1, AR5, AR2                      ; AR2 = pointer to X[id].im
    add  AR1, AR3                          ; AR3 = pointer to X[i].re
|| rptblocal inner_loop-1                 ; for(i=j; i<N; i+=LE)
    mpy  *AR5+, *CDP+, AC0                 ; AC0=(X[id].re*U.re
:: mpy  *AR2-, *CDP+, AC1                  ;      -X[id].im*U.im)/SCALE
    masr *AR5-, *CDP-, AC0                 ; AC1=(X[id].im*U.re
:: macr *AR2+, *CDP-, AC1                  ;      +X[id].re*U.im)/SCALE

```

```

    mov    pair(hi(AC0)),dbl(*AR4); ACOH=temp.re AC1H=temp.im
||  mov    dbl(*AR3),AC2
    xcc    scale,TC1
||  mov    AC2>>#1,dual(*AR3)      ; Scale X[i] by 1/SCALE
    mov    dbl(*AR3),AC2
scale
    add    T0,AR2
||  sub    dual(*AR4),AC2,AC1      ; X[id].re=X[i].re/SCALE-temp.re
    mov    AC1,dbl(*AR5+T0)        ; X[id].im=X[i].im/SCALE-temp.im
||  add    dual(*AR4),AC2          ; X[i].re=X[i].re/SCALE+temp.re
    mov    AC2,dbl(*AR3+T0)        ; X[i].im=X[i].im/SCALE+temp.im
inner_loop                                ; End of inner loop
    amar   *CDP+
    amar   *CDP+                    ; Update k for pointer to U[k]
||  add    #2,T2                    ; Update j
mid_loop                                ; End of mid-loop
    sub    #1,T1
    add    #1,fft.d_L                ; Update L
    bcc    outer_loop,T1>0          ; End of outer-loop
    mov    fft.d_ST1,AR2             ; Restore ST1,ST3,T2
    mov    fft.d_ST3,AR3
    mov    AR2,mmap(ST1_55)
    mov    AR3,mmap(ST3_55)
    mov    fft.d_T2,T2
    aadd   #(Size-ARGS-1),SP        ; Reset SP
    ret
    .end

```

# 《DSP 原理及应用》 期末考试卷

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

## 一、 填空题（共 30 分，每空 1 分）

- DSP 的狭义理解为 数字信号处理器，广义理解为 数字信号处理方法。
- 在直接寻址中，指令代码包含了数据存储器地址的低 7 位。当 ST1 中直接寻址编辑方式位 CPL =0 时，与 DP 相结合形成 16 位数据存储器地址；当 ST1 中直接寻址编辑方式位 CPL =1 时，加上 SP 基地址形成数据存储器地址。
- TMS320C54 有两个通用引脚，BIO 和 XF，BIO 输入引脚可用于监视外部接口器件的状态；XF 输出引脚可以用于与外部接口器件的握手信号。
- 累加器又叫做 目的寄存器，它的作用是存放从 ALU 或乘法器/加法器单元 输出的数据。它的存放格式为

39~32	31~16	15~0
保护位	高阶位	低阶位
- 桶形移位器的移位数有三中表达方式：立即数；ASM；T 低 6 位
- DSP 可以处理双 16 位或双精度算术运算，当 C16=0 位双精度运算方式，当 C16=1 为双 16 位运算方式。
- 复位电路有三种方式，分别是 上电复位；手动复位；软件复位。
- 立即数寻址指令中在数字或符号常数前面加一个 # 号，来表示立即数。
- 位倒序寻址方式中，AR0 中存放的是 FFT 点数的一半。
- 一般，COFF 目标文件中包含三个缺省的段：. text 段；. data 段和 . bss 段。
- 汇编源程序中标号可选，若使用标号，则标号必须从 第一列 开始；程序中可以有注释，注释在第一列开始时前面需标上 星号或分号，但在其它列开始的注释前面只能标 分号。
- ' C5402 有 23 条外部程序地址线，其程序空间可扩展到 1M，内程序区在 第 0 页
- 指令执行前有关寄存器及数据存储器单元情况如下图所示，请在下图分别填写指令执行后有关寄存器及数据存储器单元的内容。

ADD \*AR3+, 14, A

A	00 0000 1200
---	--------------

A	00 0000 2700
---	--------------

C	1
---	---

C	1
---	---

AR3	0100
-----	------

AR3	0101
-----	------

数据存储器

0100h	1500
-------	------

0100h	1500
-------	------

## 二、 简答（共 40 分）

1.TMS320C54x 有多少条 16 位总线？这些总线各有什么作用？（6 分）

答：’ C54x 共有 4 组 8 条 16 位总线

1 条程序总线（PB）：传送取自程序存储器的指令代码和立即操作数。

3 条数据总线（CB、DB、EB）：CB 和 EB 传送从数据存储器读出的操作数；EB 传送到存储器中的数据。

4 条地址总线（PAB、CAB、DAB、EAB）传送相应指令所学要的代码。

2. TMS320C54x 片内存储器一般包括哪些种类？如何控制存储器片内或片外的分配？（6 分）

答：TMS320C54x 的片内存储空间分为 3 个可选的存储空间：64K 的程序空间，64K 的数据空间和 64K 的 I/O 空间，所有 TMS320C54x 芯片都包括 RAM、SARAM、DARAM。

程序空间：MP/MC=1 4000H~FFFFH 片外

MP/MC=0 4000H~EFFFH 片外 FF00H~FFFFH 片内

OVLY=1 0000H~007FH 保留 0080H~007FH 片内

OVLY=0 0000H~3FFFH 片外

数据空间：DROM=1 F000H~FEFFH 只读空间 FF00H~FFFH 保留

DROM=0 F000H~FEFFH 片外

3.根据你的理解，试列举 DSP 芯片的特点？（5 分）

答：哈佛结构；多总线结构；指令系统的流水线操作；专用的硬件乘法器；特殊的 DSP 指令；快速的指令周期；丰富的外设

4.循环寻址中，如何确定循环缓冲的起始地址？如果循环缓冲大小为 17，其起始地址必须从什么地址开始？BK 初始化为多少？（4 分）



5.TMS320C54x 硬件复位地址为多少？如何确定中断向量地址？计算 INT0（IPTR=001H）的中断向量地址。（6 分）

答：复位后，复位向量地址为：0080H

确定地址方式：IPTR+左移 2 位后的中断向量序列号

10H 左移 2 位后成为 40H，IPTR=001H，则中断向量地址为 00C0H

6.若辅助寄存器 AR0 的值为 0x0005H，AR3 的值为 0x030FH，请分别给出下列寻址方式修改后的辅助寄存器的值。（6 分）

\*AR3+ ; AR3=0310H

\*AR3+0 ; AR3=0314H

\*AR3（15） ; AR3=0324H

7.分别解释以下指令的功能。（6 分）

LD #80h, A; 把立即数 80H 装入累加器 A

LD 80h, A; 把 80H 为地址的数据装入累加器 A

LD #80h, 16, A; 把立即数 80H 左移 16 位后装入累加器 A

### 三、 程序填空（共 22 分，每空 2 分）

1. 实现计算  $z=x+y-w$  的程序。

```
. title "example1.asm"
. mmregs
STACK . usect "STACK", 10h
. bss x, 1
. bss y, 1
. bss w, 1
. bss z, 1
. def start
. data
table: . word 10, 26, 23
. text
start: STM #0, SWWSR
      STM #STACK+10h, SP ; 初始化堆栈指针 SP
      STM #x,AR1
      RPT #2
      MVPD table,*AR1+ ; 数据从程序存储器传送到数据存储器
      CALL SUMB
```

```

end:          B          end
SUMB:        LD          @x, A
            ADD          @y, A
            SUB          @w,A      ; 实现减法运算
            STL          A, @z
            RET
            . end

```

2. 实现对数组  $X[5]=\{1, 2, 3, 4, 5\}$  的初始化, 然后将数据存储器中的数组  $X[5]$  复制到数组  $Y[5]$ 。最后实现数组  $X$  和数组  $Y$  中对应元素相乘并累加, 即  $z=1*1+2*2+3*3+4*4+5*5$

```

            . mmregs
            . bss          x, 5
            . bss          y, 5
            . bss          z, 1
            . def          start
table:       . word        1, 2, 3, 4, 5
            . text
start:       STM    #x,     AR1
            RPT     4
            MVPD    table, *AR1+    ; 从程序存储器传送到数据存储器数组 X[5]
            STM    # x,    AR2
            STM    # y,    AR3

            RPT     4
            MVDD   *AR1, *AR3      ; 从数据存储器数组 X[5]传送到 Y[5]

            SSBX      FRCT          ; 准备整数乘法

            STM    # x,    AR2
            STM    # y,    AR3
            STM    # z,    AR4
            RPTZ    4
            MAC      *AR2+,*AR3+,A
            STL     A, *AR4
End:         B      end
            . end

```

#### 四、分析程序回答问题（共 8 分）

```

            . def _c_int00
            . mmregs
stack:      . usect  "stack",10h
            . bss a,4

```

```

        . bss x,4
        . bss y,1
        . data
table:   . word 1,2,4,3
        . word 8,6,4,2
        . text
_c_int00: STM      #stack+10h, sp
        STM      #a, AR1
        RPT      #7
        MVPD     table,*AR1+
        CALL     sum
end:     B        end
sum:     STM      #a, AR3
        STM      #x, AR4
        RPTZ     A, #3
        MAC      *AR3+,*AR4+,A
        STL      A, @y
        ret
        . end

```

(1) 说明该程序的功能：(4 分)

答：  $a1 \times x1 = a2 \times x2 + a3 \times x3 + a4 \times x4$

(2) 说明以下各量的理论值 (4 分)

A: 2A

AR1: 88H

a1 的地址: 81H

AR3: 84H

内容: 1

试题编号：

## 2009—2010 学年 第 1 学期

### 《DSP 原理及应用》期末考试题 (A 卷)

题 号	一	二	三	四	五	六	七	八	总 分
得 分									
评卷人									

#### 一、填空（每空 1 分，共 30 分）

1. TI 公司的定点 DSP 产品主要有 TMS320C2000 系列、TMS320C5000 系列和 TMS320C6000 系列。
2. 'C54x DSP 中传送执行指令所需的地址需要用到 PAB、CAB、DAB 和 EAB 4 条地址总线。
3. DSP 的内部存储器类型可分为随机存取存储器 (RAM) 和只读存储器 (ROM)。其中 RAM 又可以分为两种类型：单寻址 RAM (SARAM) 和 双寻址 RAM (DARAM)。
4. 'C54x DSP 的内部总存储空间为 192K 字，分成 3 个可选的存储空间：64K 字的 程序存储空间、64K 字的数据存储空间和 64K 字的 I/O 空间。
5. 从功能结构上，'C54X DSP 的 CPU 可以划分成 运算部件 和 控制部件 两大部分。
6. 'C54x DSP 的寻址方式有七种，分别为立即寻址、绝对寻址、累加器寻址、直接寻址、间接寻址、存储器映象寄存器寻址、堆栈寻址。
7. 在'C54x DSP 寻址和指令系统中，Xmem 和 Ymem 表示 16 位双寻址操作数，Dmad 为 16 位立即数，表示 数据存储器地址，Pmad 为 16 位立即数，表示 程序存储器地址。
8. 程序计数器的值可以通过 复位操作、顺序执行指令、分支转移，累加器转移，块重复，子程序调用，从累加器调用子程序，中断等操作改变。
9. 'C54x DSP 芯片采用了 6 级流水线的工作方式，即一条指令分为 预取指、取

指、译码、寻址、读数和执行 6 个阶段。

10. 解决 MMR 写操作的流水线冲突时，一般可用采用推荐指令和插入空操作指令的方法。
11. 'C54x DSP 定时器由 3 个 16 位存储器映射寄存器组成：定时器寄存器（TIM）、定时器周期寄存器（PRD）和定时器控制寄存器（TCR）。
12. 主机接口（HPI, Host Port Interface）是 TMS320C54x 系列定点芯片内部具有的一种接口部件，主要用于 DSP 与其他总线或 CPU 进行通信。
13. 'C54x DSP 的指令系统有助记符指令和代数指令两种形式。
14. COFF 目标文件中.text 段通常包含可执行代码，.data 段通常包含已初始化的数据，.bss 段中通常为未初始化的数据保留空间。
15. DSP 芯片的开发工具可以分为代码生成工具和代码调试工具两类。

## 二、简答题(每题 5 分，共 20 分)

1. 什么是定点 DSP 芯片和浮点 DSP 芯片？各有什么优缺点？

解：按数据的定点格式工作的 DSP 芯片称为定点 DSP；

按数据的浮点格式工作的 DSP 芯片称为浮点 DSP；

定点 DSP 的价格便宜，功耗低，但运算精度低；

浮点 DSP 的价格较高，C 语言编程调试方便，运算精度高。

2. 简述流水线操作的基本原理。

解：流水线操作是各指令以机器周期为单位相差一个时钟周期，连续并行工作的情况。其本质是 DSP 多条总线彼此独立地同时工作，使得同一条指令在不同机器周期内占用不同总线资源。同时，不同指令在同一机器周期内占用不同总线资源。

3. 'C54x DSP 有哪些重复操作？各有什么优点？

解：有单条指令重复执行和程序块重复执行两种重复操作。

单条指令重复操作功能，可以使乘法/累加和数据块传送那样的多周期指令在执行一次之后变成单周期指令，大大提高了这些指令的执行速度。

利用块重复操作进行循环，是一种零开销循环。

4. 软件可编程等待状态发生器的功能是什么？

解：软件可编程等待状态产生器可以将外部总线周期扩展到 7 个机器周期（C549、C5402、C5410 和 C5420 为 14 个机器周期），这样'C54x DSP 可以方便地与慢速的片内存储器 and I/O 器件接口。

三、问答题(10 分)

1. 'C54x DSP 的串行口有哪些类型？各有什么特点？

解：有标准同步串口 SP、缓冲同步串口 BSP、多路缓冲串口 McBSP、时分多路同步串口 TMD 四种。

缓冲串口（BSP）是一个增强型的标准串口，它由一个全双工双缓冲串口和一个自动缓冲单元（ABU）组成。

由于其中的串行口与标准串口的功能相同，因此在标准模式下，缓冲串口的操作与标准串口的工作方式是一样的。不过无论是标准模式还是自动缓冲模式，BSP 都提供了一些增强功能。主要包括了可编程控制的串口时钟、可选择时钟和帧同步信号的正负极性，能够以每帧 8 位、10 位、12 位和 16 位传输数据。通过配置 BSP 的控制寄存器，BSP 还能实现忽略帧同步信号的数据传输。

时分复用串行口 TDM 采用时分复用技术，将多个外部器件复用与'C54x 进行串行通信，每一个时隙对应于其中的 1 路通信。

TDM 可以和外部的多个应用接口实现方便灵活的数据交换。'C54x 最多可以和 8 个外部器件接口通信。

多通道缓冲串口（McBSP）是在缓冲串口的基础上发展起来的增强版。

McBSP 具有高速、全双工、可与各种总线标准的器件直接接口等特点，它为 DSP 使用者在不同方面的应用提供了方便，尤其适合在通信领域的应用。

四、阅读指令并填空(每题 5 分，共 25 分)

1. BANZ 2000h, \*AR3-

指令执行前		指令执行后	
PC	<div>1000</div>	PC	<div>2000</div>
AR3	<div>005</div>	AR3	<div>004</div>

2. CMPR GT, AR3

指令执行前	指令执行后
TC <input type="text" value="1"/>	TC <input type="text" value="0"/>
AR0 <input type="text" value="FFFF"/>	AR0 <input type="text" value="FFFF"/>
AR3 <input type="text" value="7FFF"/>	AR3 <input type="text" value="7FFF"/>

### 3. ADD \*AR3+, 14, A

指令执行前	指令执行后
A <input type="text" value="00 0000 1200"/>	A <input type="text" value="00 0540 1200"/>
C <input type="text" value="1"/>	C <input type="text" value="0"/>
AR3 <input type="text" value="0100"/>	AR3 <input type="text" value="0101"/>
SXM <input type="text" value="1"/>	SXM <input type="text" value="1"/>

Data Memory

0100h <input type="text" value="1500"/>	0100h <input type="text" value="1500"/>
---	---

### 4. SUB A, -8, B

指令执行前	指令执行后
A <input type="text" value="00 0000 1200"/>	A <input type="text" value="00 0000 1200"/>
B <input type="text" value="00 0000 1800"/>	B <input type="text" value="00 0000 17EE"/>
C <input type="text" value="x"/>	C <input type="text" value="1"/>
SXM <input type="text" value="1"/>	SXM <input type="text" value="1"/>

### 5. MPY \*AR3+, A

指令执行前	指令执行后
A <input type="text" value="00 0000 1200"/>	A <input type="text" value="00 0000 0040"/>
T <input type="text" value="0004"/>	T <input type="text" value="0004"/>
AR3 <input type="text" value="0100"/>	AR3 <input type="text" value="0101"/>

Data Memory

0100h <input type="text" value="0010"/>	0100h <input type="text" value="0010"/>
---	---

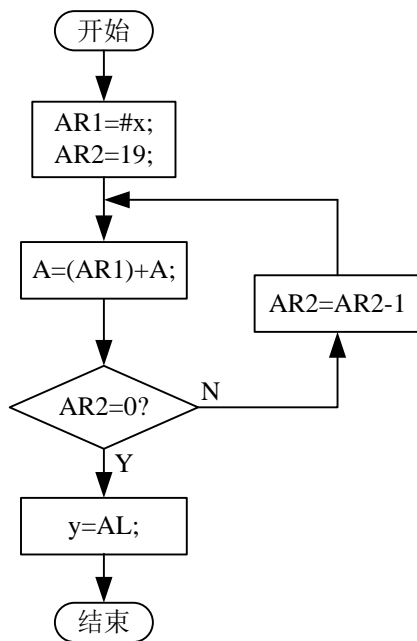
## 五、程序题(15 分)

写出计算  $y = \sum_{i=1}^{20} x_i$  的主要汇编程序，并给程序添加注释，画出程序的流程图。

解：程序及注释：

.bss	x, 20	;给变量 x 分配 20 个字单元
.bss	y, 1	;给变量 y 分配 1 个字单元
STM	#x, AR1	;取 x 地址
STM	#19, AR2	;设置循环次数
LD	#0, A	;累加器清零
loop:	ADD *AR1+, A	;累加 x 分量
	BANZ loop, *AR2-	;循环是否结束，没有结束则继续累加
	STL A, @y	;把结果存入 y

流程图：





# 《DSP 原理及其应用》期末考试试题

## (综合大作业)

### 一、名词解释 (共 5 题, 共 10 分)

1. DSP\_\_\_\_\_

2. HPI\_\_\_\_\_

3. CSS\_\_\_\_\_

4. 哈佛体系结构\_\_\_\_\_

5. CPLD\_\_\_\_\_

### 二、选择题 (共 10 题, 每题 2 分, 共 20 分)

1. TMS320C54X DSP 主机接口 HPI 是\_\_\_\_\_位并行口。

A、 32                      B、 16                      C、 8                      D、 2

2. TMS320C5000 系列的主要应用领域为\_\_\_\_\_。

A、测控领域              B、无线通信和有线通信设备中  
C、无线基站              D、图像处理

3. 寄存器 ST1 的 CNF 位决定 B0 的作用, 当 CNF=1 时, B0 映射\_\_\_\_\_。

A、程序存储空间      B、数据存储空间  
C、I/O 空间              D、以上答案均不正确

4. 看门狗定时器时钟频率是 CPU 时钟频率的\_\_\_\_\_。

A、1/128 倍              B、1/256 倍              C、1/512 倍              D、1/1024 倍

5. TMS320C54X DSP 的 32 个中断源分为 14 级, 其中\_\_\_\_\_级别最高。

A、INTR                  B、NMI                  C、RS                  D、INT0

6. DSP 系统开发的硬件实现中, PCB 图设计属于\_\_\_\_\_。

A、系统分析              B、系统综合              C、技术指标              D、确定方案

7. 仿真系统中, 主机通过仿真器与目标系统的\_\_\_\_\_接口相连。

A、JTAG              B、PCI              C、并口              D、USB 接口

8. 下列段不属于 COFF 目标文件的是\_\_\_\_\_。

A、.text              B、.data              C、.bss              D、.prj

9. 目标文件的.data 段对应的虚拟的目标存储器是\_\_\_\_\_。

A、RAM              B、EEPROM              C、ROM              D、FLASH

10. TMS320C54X DSP 具有 ( ) 个 ( ) 位的累加器。选择\_\_\_\_\_。

A、2, 40              B、1, 40              C、2, 32              D、1, 32

### 三、填空题 (共 10 题, 每题 2 分, 共 20 分)

1. DSP 与单片机、嵌入式微处理器的最大区别是能够\_\_\_\_\_、\_\_\_\_\_地进行数字信号处理运算。

2. DSP 的基本结构是采用\_\_\_\_\_结构, 即程序和数据是分开的。

3. DSP 实现 FIR 滤波器延时一个采样周期的方法有两种, 一是采用\_\_\_\_\_缓冲区法, 二是采用\_\_\_\_\_缓冲区法。

4. 汇编语言 “mov \*AR0, AC0” 使用的寻址方式是\_\_\_\_\_, “mov #0x3, DPH” 使用的寻址方式是\_\_\_\_\_。

5. TMS320C54X DSP 共有 129 条指令, 分为\_\_\_\_\_大类。

A、6              B、5              C、4              D、3

6. 当工作于\_\_\_\_\_计数模式时, 通用定时器会产生一个对称波形。

7. TMS320C54XDSP 多通道缓冲串口 (MCBSP) 发送和接收通道数最多可达\_\_\_\_\_路。

A、128              B、64              C、32              D、16

8. TMS320C54X DSP 采用改进的哈佛结构, 围绕着 ( ) 组 ( ) 位总线建立。选择\_\_\_\_\_。

A、8, 16              B、16, 8              C、8, 8              D、16, 16

9. 初始化段包含数据或代码, 包括\_\_\_\_\_段、\_\_\_\_\_段以及由汇编器伪指

令.sect 产生的命名段。

10. 同传统的单片机中断处理方式类似，DSP 中断的处理也有两种方法，分别是\_\_\_\_\_和\_\_\_\_\_。

#### 四、简答题（共 6 题，每题 5 分，共 30 分）

1. TI 公司 DSP 集成开发环境（CCS）有哪几种工作模式？各自特点是什么？
2. 列出事件管理器通用定时器计数操作的 4 种模式？
3. 说明 C5402 的内部总线结构？
4. 简述 DSP 系统开发主要包含的步骤。
5. 简述 C5402 流水线操作的过程和特点。
6. 什么是哈佛结构和冯. 诺依曼结构？他们有什么区别？

#### 五、综合应用题（共 2 题，每题 10 分，共 20 分）

1. 利用嵌入式 DSP 开发一个数码显示的多点温度检测装置，请给出模块化设计框图。并将其  
中得主要模块试作简单说明。

解答： 根据题目要求，给出系统结构简单框图如下：

2. 已知事件管理器通用定时器工作于连续递增计数模式，且定时器的使能信号和定时器时钟信号如下图所示，当周期寄存器的值  $TxPR=3$  时，画出通用定时器连续递增计数模式的示意图（至少画出 2 个计数周期）。

# 《DSP 原理及应用》期末考试题

1、对于 TMS320C54x 系列 DSP 芯片，下列说法正确的是 ( C )

(A) 专用型 DSP                      (B) 32 位 DSP                      (C) 定点型 DSP                      (D) 浮点型 DSP

2、要使 DSP 能够响应某个可屏蔽中断，下面的说法正确的是 (B )

- A. 需要把状态寄存器 ST1 的 INTM 位置 1，且中断屏蔽寄存器 IMR 相应位置 0
- B. 需要把状态寄存器 ST1 的 INTM 位置 1，且中断屏蔽寄存器 IMR 相应位置 1
- C. 需要把状态寄存器 ST1 的 INTM 位置 0，且中断屏蔽寄存器 IMR 相应位置 0
- D. 需要把状态寄存器 ST1 的 INTM 位置 0，且中断屏蔽寄存器 IMR 相应位置 1

3、若链接器命令文件的 MEMORY 部分如下所示：

```
MEMORY
{
    PAGE 0:    PROG:          origin=C00h,    length=1000h
    PAGE 1:    DATA:         origin=80h,     length=200h
}
```

则下面说法不正确的是 (A)

- A、程序存储器配置为 4K 字大小                      B、程序存储器配置为 8K 字大小
  - C、数据存储器配置为 512 字大小                      D、数据存储器取名为 DATA
- 5、C54X DSP 的流水线是由 ( B ) 级 (也即是由多少个操作阶段) 组成。
- (A) 4                      (B) 6                      (C) 8                      (D) 10

6、假定 AR3 中当前值为 200h，AR0 中的值为 20h，下面说法正确的是 ( )

- A、在执行指令\*AR3+0B 后，AR3 的值是 200h；
- B、在执行指令\*AR3-0B 后，AR3 的值为 23Fh；
- C、在执行指令\*AR3-0B 后，AR3 的值是 180h；

7、下面对一些常用的伪指令说法正确的是：( D )

- A、.def 所定义的符号，是在当前模块中使用，而在别的模块中定义的符号；
- B、.ref 所定义的符号，是当前模块中定义，并可在别的模块中使用的符号；
- C、.sect 命令定义的段是未初始化的段；
- D、.usect 命令定义的段是未初始化的段。

8、在采用双操作数的间接寻址方式时，要使用到一些辅助寄存器，在此种寻址方式下，下面的那些辅助寄存器如果使用到了是非法的 ( D )

- A、AR2                      B、AR4                      C、AR5                      D、AR6

## 二、填空题(每空 2 分，共 20 分)

1、DSP 芯片按照其用途分类，可以分为通用型和 专用型 两种。

2、在堆栈操作中，PC 当前地址为 4020h，SP 当前地址为 0033h，运行 PSHM AR2 后，PC=4021h，SP=0032h。(PSHM AR2 为单字指令)

3、TMS320C54xDSP 芯片四种串行口类型是指 SP、BSP、McBSP 和 TDMcBSP。

4、请简要说明 TMS320C5402VCDSP 以下引脚的功能：

$\overline{RS}$  : 复位引用脚 ,  $\overline{IOSTRB}$ : I/O 选通信号引脚。

5、TMS320C54xDSP 的内部总线主要包括 程序总线、数据总线 和 地址总线。

### 三、问答题与程序阅读题（共计 24 分）

1、简述 DSP 程序的段 .text, .data, .bss, 各包含什么内容。（6分）

答：.text 代码段，该段包含程序代码

.data 数据段，该段包含已初始化的数据

.bss 变量段，该段为未初始化的变量保留空间

2、比较伪指令 .usect 和 .sect。（6分）.

答：.sect: 定义初始化了的带名称的段 Unsect: 在一个未初始化的段中保留空间

3、TMS320C54X 芯片的流水线共有多少个操作阶段？每个阶段执行什么任务？（8分）

答：共有 6 个操作阶段:1.预取址 2, 取址 3, 译码 4, 寻址 5, 读数 6, 执行

4、假设 AR3 的当前值为 200h, 当使用以下 TMS320C54XX 寻址模式后其中的值为多少？假定 AR0 的值为 20h。(4 分)

(1) \*AR3+0 (2) \*AR3-0 (3) \*AR3+ (4) \*AR3

### 四、 简答（共 40 分）

1.TMS320C54x 有多少条 16 位总线？这些总线各有什么作用？（6 分）

答：’ C54x 共有 4 组 8 条 16 位总线

1 条程序总线（PB）：传送取自程序存储器的指令代码和立即操作数。

3 条数据总线（CB、DB、EB）：CB 和 EB 传送从数据存储器读出的操作数；EB 传送到存储器中的数据。

4 条地址总线（PAB、CAB、DAB、EAB）传送相应指令所学要的代码。

2。TMS320C54x 片内存储器一般包括哪些种类？如何控制存储器片内或片外的分配？（6 分）

答：TMS320C54x 的片内存储空间分为 3 个可选的存储空间：64K 的程序空间，64K 的数据空间和 64K 的 I/O 空间，所有 TMS320C54x 芯片都包括 RAM、SARAM、DARAM。

程序空间：MP/MC=1 4000H~FFFFH 片外

MP/MC=0 4000H~EFFFH 片外 FF00H~FFFFH 片内

OVL=1 0000H~007FH 保留 0080H~007FH 片内

OVLY=0	0000H~3FFFH	片外		
数据空间: DROM=1	F000H~FEFFH	只读空间	FF00H~FFFH	保留
DROM=0	F000H~FEFFH	片外		

3. 当 TMS320C54x CPU 接收到可屏蔽的硬件中断时, 满足哪些条件中断才能被响应? (6 分)

答: (1) 出现多个中断时, 此中断的优先级最高

(2) INTM=0 允许全局中断

(3) IMR 中的响应相应位为 1, 开放此中断。

4. TMS320C54x 硬件复位地址为多少? 如何确定中断向量地址? 计算 INT0 (IPTR=001H) 的中断向量地址。(6 分)

答: 复位后, 复位向量地址为: 0080H

确定地址方式: IPTR+左移 2 位后的中断向量序列号

10H 左移 2 位后成为 40H, IPTR=001H, 则中断向量地址为 00C0H

# 2008-2009 学年第二学期期末考试试题答案

## 《 DSP 原理及其应用》 (B 卷)

题号	一	二	三	四	总分
得分					

### 一、填空题（请将答案填在相应的答题线上。每空 2 分，共 30 分）

1. 采用 Q0.15 格式的十六进制数 3000H 与 Q15.0 格式的 005DH 的十进制和为：  
93.375。
2. 德州仪器的 DSP 芯片分为三大系列：C2000、C5000、C6000，其中 C2000 系列多取代 MCU 用于复杂的工业控制系统中。
3. ALU 的运算结果送至累加器 A、B，如果 A 有溢出发生，则置溢出标志位 OVA=1。
4. TMS320C54x 芯片中负责存储器配置的三个控制位是： $\overline{MP/\overline{MC}}$ 、OVLY 和 DROM。
5. TMS320C54x 的片内 RAM 和 ROM 都以块的方式来组织数据的存取，并且可以采用分页扩展技术将程序存储空间扩展到 8M 字。
6. 单操作数间接寻址的两个特殊功能是：循环寻址和位码倒序寻址。
7. 在执行重复指令 RPT 或 RPTZ 过程中，对所有可屏蔽中断是否响应：不响应。
8. TMS320C5416 的 COFF 文件格式为：COFF2。
9. TMS320C54x 的 COFF 文件都包含三种形式的段伪指令：.bss、.text、data。
10. 伪指令 “.bss x, 5” 的功能是：定义长度为 5 个字单元的数组 x。
11. DSP 汇编程序的四种基本结构是：顺序结构、分支结构、循环结构、子程序结构。
12. 中断系统中最高优先级的中断是复位中断（或 RS）。
13. TMS320C54x 的两个中断管理寄存器分别是：IFR 和 IMR。

14. 在 CCS 开发环境中，一般通过设置断点、  探针（或探测点）  和评价点，以步进方式调试程序。
15. 在 CCS 开发环境中，源程序或工程文件经构建后生成的输出文件格式为  .out  。

**二、简答题（第 1 小题 2 分，第 2 小题 3 分，第 3 小题 3 分，第 4 小题 7 分，第 5 小题 7 分，第 6 小题 8 分，共 30 分）**

1. 什么是数字信号处理器？

答：

数字信号处理器指的是实现我们所需要的数字信号处理功能的硬件载体（专用微处理器）。 （2 分）

2. 试举例说明数字信号处理器的应用领域有哪些。

答：

数字信号处理器广泛应用于：

**移动通信、数字图像、航天军工、数字娱乐产品、医疗器械等领域及各种智能设备。**

（写出 5 个以上应用领域即给 3 分）

3. DSP 的软件开发中可以使用哪几种源程序？哪种源程序更适合作为系统文件？

答：

1) 可以使用三种源程序：**汇编程序.asm**、**C 语言程序.c**、以及混合源代码； （2 分）

2 ) **C 语 言 程 序 .c** 更 适 合 作 为 系 统 文 件 。  
（1 分）

4. 回答下列 DSP 存储器的有关问题：

（1）TMS320C54x 中，控制位 OVLY 是如何配置存储器空间的？

（2）在多级流水线结构中，双寻址存储器（DARAM）容易引发时序冲突的原因是什么？解决的方法有哪些？



答：

(1) OVLY -- 决定 0000H~3FFFH 程序空间的片内外分配  
(3 分)

OVLY=1, 0000H~007FH 保留 (程序无法占用)

0080H~3FFFH 为片内 DARAM

OVLY=0, 0000H~3FFFH 全部为片外程序空间

(2) 原因： (2 分)

双寻址存储器 (DARAM) 在单周期内, 允许 CPU 对 DARAM 访问两次, 第一次在前

半周期, 第二次在后半周期。但在同半个周期内做两个操作, 容易引发时序冲突。

解决方法有： (2 分)

a. CPU 重新排定访问次序

b. 写操作延迟一个周期

c. 插入空周期

5. 回答以下 C54x 指令系统的有关问题：

(1) C54x 的指令系统按功能可分为哪四个基本类型？

(2) MV 系列指令中, 在数据存储器内部传送数据应使用哪个？在 MMR 空间互传数据应使用哪个？

(3) 重复指令的作用及优点是什么？

答：

(1) C54x 的指令系统按功能可分为四个基本类型：

算术运算指令、逻辑运算指令、程序控制指令、存储及转移指令。

(2 分)

(2) MVDD; MVMM

(2 分)

(3) 对于一些多周期指令, 重复操作使这些指令在第一次执行后变为单周期指令, 从而降低程序的时间开销。并且, RPT 指令在重复执行时, 不响应可屏蔽中断。

(3 分)

6. 回答以下 C54x 汇编链接原理的有关问题：

- (1) 链接命令文件.cmd 中包含哪两个伪指令？它们的作用分别是什么？
- (2) 试述汇编器和链接器的作用。

答：

- (1) 链接命令文件.cmd 中包含伪指令 MEMORY 和 SECTIONS：

(2 分)

MEMORY 伪指令 —— 用来划分目标系统的存储器配置空间，包括对存储器各部分命名，  
以及规定它们的起始地址和长度。

(1 分)

SECTIONS 伪指令 —— 用来指定链接器将输入段组合成输出段方式，分配各段进入空间。

(1 分)

- (2) 汇编器的作用：

将各部分程序代码和数据连在一起，构成目标文件.obj。

(2 分)

链接器的作用：

分配存储单元，即将目标文件中的段重新定位到目标系统的存储器中。(2 分)

### 三、指令分析与计算（写出指令的功能表达式及结果。第 1 小题 8 分，第 2 小题 12 分，共 20 分）

1. MAC \*AR1+, \*AR2-, A, B

指令执行前		指令执行后	
A	00 0000 1000H	A	保持
B	00 0000 2200H	B	00 000D 2400H
T	0003H	T	0345H
FRCT	0	FRCT	保持
AR1	0100H	AR1	0101H
AR2	0200H	AR2	01FFH
0100H	0345H	0100H	保持
0200H	0400H	0200H	保持

1 分

1 分

1 分

1 分

- (1)  $A + (AR1) * (AR2) \rightarrow B$

(2 分)

$$1000H+0345H*0400H=1000H+D,1400H= D,2400H \rightarrow B$$

(2) (AR1)=0345H  $\rightarrow$  T

(1 分)

(3) AR1+1  $\rightarrow$  AR1=0101H, AR2-1  $\rightarrow$  AR2=01FFH

(1 分)

(即：代数功能表达式写全给 4 分)

(表达式不写但答案正确的最多给 6 分)

...

2. LDM \*AR1, A  
BIT \*AR1, 7  
MIN B  
STH A, 2

指令执行前		指令执行后		
A	00 0000 0000H	A	00 0000 2700H	1 分
B	00 0050 F900H	B	00 0000 2700H	1 分
TC	X	TC	1	2 分
DP	1	DP	保持	
AR1	10E2H	AR1	0062H	2 分
0002H	0999H	0002H	保持	
0062H	2700H	0062H	保持	
0082H	0516H	0082H	0000H	2 分
10E2H	0833H	10E2H	保持	

(1) LDM 的真实寻址是：(0062H)

所以(0062H)=2700H  $\rightarrow$  A=00 0000 2700H, 0062H  $\rightarrow$  AR1=0062H (1 分)

(2) (AR1) (15-7)=(0062H) (8)=2700H(8)=1  $\rightarrow$  TC (1 分)

(3)  $\min\{A, B\} = A \rightarrow B = 2700H$  (1 分)

(4) 直接寻址@2 的真实地址是 0082H: Hi (A)=0000H  $\rightarrow$  0082H (1 分)

(代数功能表达式写全做多给 4 分)(表达式不写但答案正确的最多给 10 分)

#### 四、综合程序题（根据题目要求作答。第 1 小题 5 分，第 2 小题 15 分，共 20 分）

1. 在 TMS320C54x 系列中没有专门的除法指令，试述间接实现除法的方法有哪些？

应采用哪个指令组合？

答：

在 TMS320C54x 系列中，实现除法的运算有两种间接方法：

- 借助倒数的乘法进行 (1 分)
- 16 次重复 SUBC 减法运算，指令组合为： (2 分)  
“ RPT #15  
SUBC \*AR2, B ” (2 分)

2. 编写汇编程序，建立数组  $x = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ，并从  $x$  中读数据到  $y$ ，使得  $y = \{1, 4, 7\}$ 。

(至少需写出关键指令与伪指令步骤)

(正确写出程序关键步骤就给满分，其余情况酌量给分)

```
.title      " read_data.asm"
.mmregs

.bss x,9
.bss y,3

.data
table:     .word    1,2,3,4,5,6,7,8,9

.text
start:     STM      #3, AR0
           STM      #x, AR1
           STM      #x, AR2
           RPTZ     A, #8
           MVPD     table, *AR2+    ; establish vector x and then AR2 point to
y
           RPT      #2
           MVDD     *AR1+0, *AR2+ ; copy data from x to y

end:       B        end
           .end
```

# 《DSP 原理及应用》期末总复习资料

## 填空:

- OVLY= (0)，片内 RAM 仅配置到数据存储空间。
- DROM= (1)，片内 ROM 配置程序和数据存储空间。
- ST1 的 CPL= (1) 表示选用对栈指针 SP 的直接寻址方式。
- ST1 的 C16= (1) 表示 ALU 工作在双精度算术运算式。
- 软件中断是由 (INTR) (TRAP) (RESET) 产生的。
- 时钟发生器包括一个 (内部振荡电路) 和一个 (锁相环电路)。
- 状态寄存器 ST1 中 CPL=0 表示 (使用 DP)，CPL=1 表示 (使用 SP)
- 累加器寻址的两条指令分别是 (READA Smem) (WRITA Smem)
- 链接器对段的处理主要通过 (MEMORY) 和 (SECTIONS) 两个命令完成。
- 所有的 TMS320C54x 芯片内部都包含 (程序) 存储器和 (数据) 存储器。
- 所有的 COFF 目标文件都包含以下三种形式的段: (.text 文本段 .data 数据段 .bss 保留空间段)。
- TMS320C54x 有 8 组 16 位总线 (1 组程序总线, 3 组数据总线, 4 组地址总线)。
- TMS320C54x DSP 具有两个 (40) 位累加器。累加器 A 的 (AG 或 32~39) 位是保护位。
- 对于 32 位数寻址时, 如果寻址的第一个字处在偶地址, 那么第二个就处在 (下一个高) 地址; 如果寻址的第一个字处在奇地址, 那么第二个就处在 (前一个低) 地址。
- 
- DSP 芯片特点: 有 (改进的哈佛结构)、(低功耗设计) 和 (高度并行性) (多处理单元) (特殊 DSP 指令) 等特点。
- DSP 片内寄存器在 C 语言中一般采用 (指针) 方式来访问, 常常采用的方法是将 DSP 寄存器地址的列表定义在 (头文件)。
- TMS320C54x 有 3 个 16 位寄存器作为状态和控制寄存器 (ST0) (ST1) (PMST)。
- TMS320C54x 的三类串行口: (标准同步串行口) (缓冲串行口) (时分多路串行口)。
- TMS320C54x 的工作方式状态寄存器 PMST 提供了三个控制位, 包括 (MP/非 MC)、(OVLY)、(DROM)。

- MEMORY 的作用（是用于描述系统实际的硬件资源，用来定义用户设计的系统中所包含的各种形式的存储器，以及他们占据的地址范围）。
- SECTIONS 的作用（是用于描述段如何定位到家当的硬件资源上。将输出段定位到所定义的存储器中）。
- 直接寻址就是在指令中包含有数据存储器地址的（低 7 位）用作偏移地址，并与（基地地址）组成 16 位的数据存储器地址。
- 时钟发生器为 TMS320C54x 提供时钟信号，时钟发生器可以由（内部振荡电路）或（外部时钟电路）驱动。
- 桶型移位寄存器能把输入的数据进行（0-31）位的左移和（0-15）位右移。
- 即寻址方式中的立即数有两种数值形式，数值的位数为（3、5、8、9）位时为短立即数，数值的位数为（16）位时为长立即数。
- MP/非 MC=（1），允许片内 ROM 配置到程序存储空间。
- MP/非 MC=（0），禁止片内 ROM 配置到程序存储空间。
- OVLY=（1），片内 RAM 配置程序和數據存储空间。
- DROM=（0），禁止 ROM 配置到到数据存储空间。

## 指令：

- ADD #4568H,8,A,B //将 4568H 左移 8 位与 A 相加，赋给 B。
- ADD A,-8,B//将 A 右移 8 位加上 B，保存于 B。
- ADD \*AR3+,14,A//将 AR3 左移 4 加上 A→A。
- DST B,\*AR1-//B 存到长字单元→AR1 中，且 AR 减 1。
- LDM AR1,A //存储器映像寄存器寻址方式，将映像寄存器加载累加器 AR1→A 的低位，其余位置 0。
- LD #10H,4,A//将 10H 左移 4 位加载到累加器 A 中。
- LDM MMR,A//
- MAC \*AR3+,\*AR4+,A,B//AR3.AR4+A→B，AR3 与 AR4 都加 1。
- MVKD SAMPLE,\*AR5//SAMPLE→AR5 中。

●MVDD \*AR3+,\*AR5+ //数据存储器内部传送数据 AR3→AR5,且指令结束后 AR3 与 AR5 加 1。

●MPYA TEMP2//B<=A.Temp2,T=Temp2。

●PPT #99

NOP//重复 NOP100 次。

●STL A,@quot//将累加器 A 的低 16 位字存放在 quot 的存储单元中。

●STL B,\*AR3+//将累加器的低位移位后加载到 AR3 所指地址。

●WRITA SMEM //将 A 的内容写入 SMEM 中。

●●●●●●●●

●LD #0032H,4,A//立即数 0032H 先左移 4 位后,再加载累加器 A.

●STM #1000H,\*AR2//立即数 1000H 存储到 AR2 指向的地址单元

●MAC #345,A,B//立即数 345 与 T 寄存器内值相乘后与累加器 A 值相加,结果加载累加器 B.

●MVDD \*AR3-,100H //AR3 指向的地址单元的值传送给地址 100H 单元,AR3 中地址减一。

●LDM \*AR1,A//AR1 指向的地址单元的值加载到累加器 A.

## 简答:

■CPU 响应中断有哪些条件? 中断处理一般过程是什么?

对于软件中断和非可屏蔽中断, CPU 立即响应。如果是可屏蔽中断, 只有满足以下条件才能响应:

1 优先权利最高;

2 ST1 中的 INTM=0 即允许可屏蔽中断;

3 IMR 中相位为 1, 允许可屏蔽中断。

中断流程:

1 接受中断请求;

2 响应中断;

3 执行中断服务程序。

■简述链接器命令文件中的 MEMORY 和 SECTIONS。

MEMORY 命令用于描述系统实际的硬件资源, 用来定义用户设计的系统中所包含的各种形式的存储器, 以及他们占据的地址范围。

SECTIONS 命令用于描述段如何定位到家当的硬件资源上。将输出段定位到所定义的存储器中。

#### ■简述 TMS320C54x 芯片的主要特点。

- 1 CPU(中央处理单元)利用其专用的硬件逻辑和高度并行性提高芯片的处理性能。
- 2 存储器具有 192k 字可寻址存储空间 (包括 64 K 字程序存储空间、64 K 字数据存储空间和 64 K 字 I/O 空间)。。
- 3 高度专业化的指令集能够快速地实现算法用于高级语言编程优化。
- 4 片内外设和专用电路采用模块化的结构设计,可以快速的推出新的系列产品。
- 5 TMS320C54x 执行单周期定点指令时间为 25/20/15/12.5/10ns,每秒指令数位 40/66/100MIPS。
- 6 TMS320C54x 电源由 IDLE1、IDLE2、IDLE3 功耗下降指令控制功能,以便 DSP 工作在节电模式下,使之更适合于手机。
- 7 在片仿真接口,片上的 JTAG 接口符合 IEEE1149.1 边界扫描逻辑接口标准,可与主机连接,用于芯片的仿真和测试。

#### ■简述 TMS320C54x 芯片存储器的分配方法。

TMS320C54x 采用改进的哈佛结构。存储空间由三个独立可选的存储空间组成: 64k 字的程序存储空间、64k 的数据存储空间、64k 的 I/O 存储空间。

TMS320C54x 的工作方式状态寄存器 PMST 提供了三个控制位: MP/非 MC、OVLY、DROM,用于在存储空间中配置片内存储器。

**MP/非 MC:** 微处理器/微型计算机工作方式位。

**OVLY:** RAM 重叠位。

**DROM:** 数据 ROM 位。

#### ■简述直接寻址及其两种方式:

直接寻址就是在指令中包含有数据存储器地址的低 7 位,用着 7 位作为偏移地址,并与基地址值组成一个 16 位的数据存储器地址。直接寻址分为数据页指针直接寻址、堆栈指针直接寻址两种。

**数据页指针直接寻址:** 当状态寄存器 ST1 中的 CPL 位等于 0 时,ST0 中的 DP 值与指令中的 7 位地址一起形成 16 位数据存储器地址;

**堆栈指针直接寻址:** 当 ST1 中的 CPL 位等于 1 时,将指令中的 7 位地址与 16 位堆栈指针 SP 相加,形成 16 位的数据存储器地址。

#### ■简述位码倒寻址的主要用途及实现方法。



位码倒寻址的主要用于 FFT 算法中。

实现方法：ARO 存放的整数 N 是 FFT 点数的一半，一个辅助寄存器只想一个数据存放的物理单元，当使用位码倒寻址把 ARO 加到辅助寄存器中时，地址以位倒序的方式产生，即进位是从左向右，而不是通常的从右向左。

■堆栈寻址的作用是什么？压栈和弹出堆栈是如何实现的？

系统堆栈用来在中断或调用子程序期间自动存放程序计数器，也能用来存放用户当前的程序环境或传递数据值。处理器使用一个 16 位存储器映像寄存器的一个堆栈指针来寻址堆栈，SP 始终指向存放在对战中的最后一个单元。当调用一个子程序或一个中断响应发生时，PC 会被自动压栈，堆栈指针 SP 指向存放最后一个数据的堆栈单元；返回时，返回地址从堆栈中弹出并装入 PC。

堆栈存放数据是从高地址向低地址进行的，压入堆栈时，先减小 SP，再将数据压入堆栈；弹出对战士，先从堆栈弹出数据，在增加 SP 值。

■简述 TMS320C54x 宏命令及其功能。

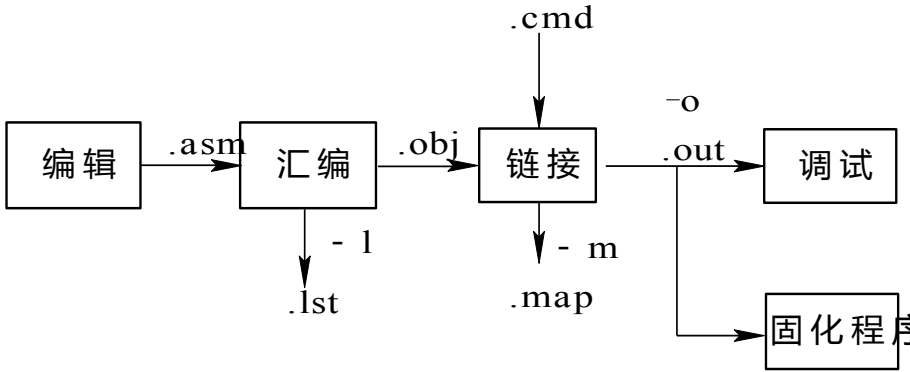
宏命令是源程序中具有独立功能的一段程序代码，它可以根据用户的需要，由用户创建自己的指令。宏命令一经定义，便可在以后的程序中多次调用，从而可以简化和缩短源程序。

功能：定义自己的宏，重新定义已存在的宏、简化长的或复杂的汇编代码、访问由归档器创建的宏库、处理一个宏中的字符串、控制展开列表。

使用过程：宏定义，宏调用，宏展开。

■画出汇编程序的编辑、汇编和链接过程图，并说明各部分的作用？

3. 答：



编辑：程序代码的编写，生成 .asm 文件。

汇编：利用 TMS320C54x 的汇编器对已经编好的一个或者多个源文件分别进行汇编，并生成 .lst(列表)文件和 .obj(目标)文件。

链接：利用 TMS320C54x 的链接器 LNK500，根据链接器命令文件(.cmd)对已经汇编过的一个或是多个目标文件(.obj)进行链接，生成 .map 文件和 .out 文件。

■简述流水线的 6 个独立阶段并介绍其功能。

- 1 程序预取指：将所要取指的地址放在程序地址总线上；
- 2 取指：从程序总线上取指令字，并装入指令寄存器；
- 3 指令译码：对 IR 中的内容译码，产生执行指令所需要的一系列控制信号；
- 4 寻址：数据地址产生单元在数据地址总线上输出读操作数的地址。；
- 5 读：从数据总线和控制总线上读操作数；
- 6 执行指令：从数据总线上写数据。

■154■■■简述 TMS320C54x 芯片的中断系统。

中断系统是计算机发展史上的一个里程碑。也是衡量微处理器性能好坏的一项主要指标。中断系统既支持硬件中断，又支持软件中断。无论是哪种中断都可以分为可屏蔽中断和非可屏蔽中断。中断系统包括：中断结构、中断流程、中断编程。

■简述串行通信，并描述 TMS320C54x 的三个串口。

串行通信是发送器将并行数据逐位移出成为串行数据流，接收器将串行数据流以一定的时序和一定的格式呈显在连接接收/发器的数据线上。

三类串行口：

标准同步串行口(SPI)：有两个独立的缓冲器用于传送数据，接收缓冲器和发送缓冲器，每个缓冲器有一条可屏蔽的中断线。串行数据可以按 8 位字或 16 位字转换。

缓冲串行口(BSP)：在标准同步串行口的基础上增加了一个自动缓冲单元，BSP 是一种增强型标准串行口，它是全双工的，并有两个可设置大小的缓冲区。

时分多路串行口(TDM)：允许同一个串口以分时方式传送多路数据，TDM 为多处理器通信提供了一种简单而有效的方式。

■■■■■■■

■段的作用是什么？COFF 目标文件包含哪些段？

所谓段，是指连续地占有存储空间的一个代码块或数据块，是 COFF 文件中最重要的概念。一

个目标文件中的每一个段都是分开的和各不相同的。所有的 COFF 目标文件都包含一下三种形式的段：.text 文本段 .data 数据段 .bss 保留空间段

■TMS320C54x 有哪几种基本的寻址方式？

立即寻址、绝对寻址、累加器寻址、直接寻址、间接寻址、存储器映像寄存器寻址、堆栈寻址。

■一个典型的 DSP 系统通常由哪些部分组成？画出原理框图。

A/D、DSF、D/A。

1001010011

1001010111010100101

100100011101000010110

100101011 1010100101

1000

■简述存储器映像寄存器寻址地址产生及其地址产生的两种方式。

存储器映像寄存器是用来修改存储器映像寄存器的，但不影响当前数据页指针 DP 或堆栈指针 SP 的值。

1 在直接寻址方式下，不管当前 DP 或 SP 为何值，是数据寄存器地址的高 9 位强制置 0，数据存储器地址的低 7 位则来自于指令字；

2 在间接寻址方式，只使用当前辅助寄存器的低 7 位作为数据存储器地址的低 7 位，地址的高 9 位为 0，指定的辅助寄存器的高 9 位在寻址后被强制置 0。

■简述链接器的作用和功能。

链接器的作用就是根据链接命令或链接命令文件 (.cmd)，将一个或多个 COFF 目标文件链接起来，生成存储器映像文件 (.map) 和可执行文件的输出文件 (.out)。

功能：

1 将各个段配置到目标系统的存储器中；

2 对各个符号和段进行重新定位，并给他们制定一个最终的地址；3 结局输入文件之间未定义的外部引用问题。

■HPI 接口有几个寄存器？他们的作用是什么？

1 HPI 存储器：用于 TMS320C54x 与主机间传送数据。地址从 1000H 到 17FFH，空间容量为 2K 字。

2 HPI 地址寄存器：由主机对其直接访问，存放当前寻址 HPI 存储单元的地址。

3 HPI 数据锁存器：有主机对其直接访问，存放当前进行读/写的的数据。

4 HPI 控制寄存器：TMS320C54x 和主机都能对其直接访问，用于主处理与 DSP 相互握手，实现相互中断请求。

5 HPI 控制逻辑：用于处理 HPI 与主机之间的接口信号。

## 程序：

●设计一存储空间为 100 个单元的堆栈。将栈底地址指针送 SP,对其初始化。

Size .set 100;设置堆栈空间的大小为 100

Stack .usect "STK",size ;设置堆栈段的首地址和堆栈空间

STM #stack+size,SP

●利用 SUBC 完成整除法,41H/7H=9H,余数是 2H。

LD #0041H, B

STM #0100H, AR2

STM #0110H, AR3

ST #0007H, \*AR2

RPT #15

SUBC \*AR2, B

STL B,\*AR3+

STH B,\*AR3

●编写求解加、减法的程序,计算  $z=x+y-w$ 。

.title "ex41.asm"

.mmregs

STACK .usect "STACK",10H

.bss x,1

.bss y,1

.bss w,1

.bss z,1

.def start

Table: .word 6, 7,9

.text

Start: STM #0,SWWSR

STM #STACK+10H,SP

STM #x,AR1

RPT #2

MVPD table,\*AR1+

LD @x,A

ADD @y,A

SUB @w,A

STL A,@z

End: B end

.end

●假设目标存储器的配置如下：

程序存储器： EPROM E000h~FFFFH(片外)

数据存储器： SPRAM 0060H~007FH (片内);DARAM 0080H~017FH (片内)

要求编写链接命令文件 ex421.cmd。此命令用来链接 ex421.obj 和 vector.obj 两个目标文件(输入文件),并生成一个映像文件 ex421.map,以及一个可执行的输出文件 ex421.out。标号"start"是程序的入口。

答案：

ex421.obj

vectors.obj

-o ex421.out

-m ex421.map

-e start

MEMORY

{

PAGE 0: EPROM : org=0E000H,len=100H

```

vecs : org=0FF80H,len=04H

vecs1 : org=0FFC8H,len=04H

PAGE 1:

SPRAM : org=0060H,len=20H

DARAM : org=0080H,len=100H

}

```

## SECTIONS

```

{

.text : >EPROM PAGE 0

.bss : >SPRAM PAGE 1

.data : >EPROM PAGE 0

STACK : >DARAM PAGE 1

.vectors: >vecs PAGE 0

INT_2 : >vecs1 PAGE 0

}

```

● 阅读下列汇编源程序, 在每条语句后写出注释, 并叙述程序的功能。

```

.title "c jy4.asm"

.mmregs

STACK .usect "STACK",10H

.bss a,4 ;为 a 分配 4 个存储单元

.bss x,4 ;为 x 分配 4 个存储单元

.bss y,1 ;为结果 y 分配 1 个存储单元

.def start

.data ;定义数据代码段

table: .word 1*32768/10 ;在 table 开始的 8 个

.word 2*32768/10 ;地址放数据

.word -3*32768/10

.word 4*32768/10

.word 8*32768/10

.word 6*32768/10

.word -4*32768/10

.word -2*32768/10

```

.text ;定义可执行程序代码段

start:SSBX FRCT ;设置 FRCT 位,表示进行小数乘

STM #x,AR1 ;将 x 的首地址传给 AR1

RPT #7 ;重复 8 次下条指令

MVPD table,\*AR1+ ;将程序空间 8 个数传给数据存储器

STM #x,AR2 ;将数据存储器第一个数 x1 的地址传给 AR2

STM #a,AR3 ;将数据存储器第五个数 a1 的地址传给 AR3

RPTZ A,#3 ;将 A 清零,重复 4 次下条指令

MAC \*AR2+,\*AR3+,A ;执行乘法累加和,结果放在 A 中

STH A,@y ;将 A 的高端字存入结果 y,低端字省去

end: B end ;原处循环等待

.end

●编写一段程序,将 PM 中的 10 个数首先传送到 DM 中(以地址 0016H 开始),再将该数据传送到地址以 0058H 开始的 DM 中。(PM: 程序存储器;DM: 数据存储器)。

●●●●●●●●

●编程实现小数乘法,要求将变量 temp1 装入累加器高 16 位与 temp2 相乘,结果存入 temp3。

SSBX FRCT ; 置 FRCT 标志位,准备小数乘法

LD temp1, 16, A ; 将变量 temp1 装入累加器 A 的高 16 位

MPYA temp2 ; 完成 temp2 与累加器 A 的高 16 位相乘,结果放入 B 累加器,并将 temp2 装入 T 寄存器

STH temp3 ; 将乘积结果的高 16 位存入变量 temp3

●将数组 x[5] 初始化为{1, 2, 3, 4, 5}。(共 10 分)

.data; 定义初始化数据段起始地址

TBL:.word 1,2,3,4,5 ; 为标号地址 TBL

; 开始的 5 个单元赋初值

.sect ".vectors" ; 定义自定义段,并获得该段起始地址

B START ; 无条件转移到标号为 START 的地址

.bss x,5 ; 为数组 x 分配 5 个存储单元

.text ; 定义代码段起始地址

START:STM #x,AR5 ; 将 x 的首地址存入 AR5

RPT #4 ; 设置重复执行 5 次下条指令

MVPD TBL,\*AR5+

end: B end

.end

● 阅读下列汇编源程序，在每条语句后写出注释，并叙述程序的功能。

.title “cjyl.asm” ; 为汇编源程序取名

.mmregs ; 定义存储器映象寄存器

STACK .usect “STACK”,30H; 设置堆栈

.bss x,10 ; 为数组 x 分配 10 个存储单元

.bss y,10 ; 为数组 y 分配 10 个存储单元

.data

table:.word 1,2,3,4,5,6,7,8,9,10

.def start ; 定义标号 start

.text

start:STM #0,SWWSR ; 复位 SWWSR

STM #STACK+30H,SP; 初始化堆指针

STM #x,AR1 ; 将目的地首地址赋给 AR1

RPT #19 ; 设定重复传送的次数为 20 次

MVPD table,\*AR1+ ; 程序存储器传送到数据存储器

STM #x,AR2 ; 将 x 的首地址存入 AR2

STM #y,AR3 ; 将 y 的首地址存入 AR3

RPT #19 ; 设置重复执行 20 次下条指令

MVDD \*AR2+,\*AR3+; 将地址 x 开始的 20 个值复制到地址 y 开始的 20 个单元

end: B end

.end

程序功能：将数据存储器中的数组 x[10]复制到数组 y[10]。