**《TMS320C54x DSP应用技术教程》部分习题**

1. **绪论**
2. **简述Digital Signal Processing 和Digital Signal Processor 之间的区别与联系。**

**答：前者指数字信号处理的理论和方法，后者则指用于数字信号处理的可编程微处理器，简称数字信号处理器。数字信号处理器不仅具有可编程性，而且其数字运算的速度远远超过通用微处理器，是一种适合于数字信号处理的高性能微处理器。数字信号处理器已成为数字信号处理技术和实际应用之间的桥梁，并进一步促进了数字信号处理技术的发展，也极大地拓展了数字信号处理技术的应用领域。**

1. **什么是DSP技术？**

**答：DSP技术是指使用通用DSP处理器或基于DSP核的专用器件，来实现数字信号处理的方法和技术，完成有关的任务。**

1. **DSP的结构特点有哪些？**

**答：哈佛结构和改进的哈佛结构、多总线结构、流水线技术、多处理单元、特殊的DSP指令、指令周期短、运算精度高、硬件配置强。**

1. **什么是哈佛结构和冯.诺依曼结构？它们有什么区别？**

**答：DSP处理器将程序代码和数据的存储空间分开，各空间有自己独立的地址总线和数据总线，可独立编址和独立访问，可对程序和数据进行独立传输，这就是哈佛结构。以奔腾为代表的通用微处理器，其程序代码和数据共用一个公共的存储空间和单一的地址和数据总线，取指令和取操作数都是通过一条总线分时进行的，这样的结构称为冯·诺伊曼结构。**

**采用哈佛结构，可同时取指令和取操作数，并行地进行指令和数据的处理，从而可以大大地提高运算的速度，非常适合于实时的数字信号处理。而冯·诺伊曼结构当进行高速运算时，取指令和取操作数是分时操作的，这样很容易造成数据传输通道的瓶颈现象，其工作速度较慢。**

**10.DSP可以按几种方式进行分类？**

**答：1）按照DSP处理器工作时的数据格式划分，可以将DSP处理器分为定点DSP和浮点DSP。**

**2）按照DSP处理器的用途划分，可以将DSP处理器分为通用型DSP和专用型DSP。**

**3）按照DSP处理器的工作时钟和指令类型划分，可以将DSP处理器分为静态DSP处理器和一致性DSP处理器。**

**4）按照不同生产厂家的产品系列划分，可以将DSP处理器分为TI公司的TMS320系列、AD公司的ADSP21系列、AT&T公司的DSP16/32系列，Motolora公司的MC5600/MC9600系列、NEC公司的μPD77系列等。**

1. **DSP系统设计概述**
2. **DSP与通用的CPU和微控制器（MCU）相比有什么特点？**

**答：DSP处理器与通用CPU和微控制器（MCU）相比，从系统管理的角度看，通用CPU具有强大的优势；从系统简单、易于开发的角度看，MCU提供了相应用户电路，具有良好的实用性；但如果需要实现复杂数学计算，或需要进行高速数字运算的数字信号处理系统（例如语音识别、图像实时处理和多媒体处理等），则需要使用DSP处理器来完成，这是由于DSP处理器在结构上采用了许多专门技术和措施，来提高数据运算处理速度，使其能实时快速地实现各种数字运算。**

1. **简述DSP系统的基本构成。**

**答：典型的DSP系统基本结构框图如图所示。从结构框图可以看出，典型的DSP系统包括数字信号处理器DSP、存储器、A/D和D/A转换器、模拟控制和处理电路、各种控制口与通信口，同时还需要电源管理以及为并行处理或协处理提供的同步电路等。**

抗混叠

滤波器

程序存储器

输入

A/D 转换器

数字信号处理器

D/A 转换器

平滑

滤波器

输出

数据存储器

主机/用户接口

通信接口

**4.简述DSP 系统的一般设计过程。**

**答：DSP系统的设计开发过程可以用下图所示的流程图来表示，该流程图将设计过程大致分为如下几个阶段：**

1. **定义系统性能指标**
2. **确定算法并进行算法模拟**
3. **选择DSP处理器**
4. **设计DSP应用系统（软硬件设计）**
5. **系统集成和测试**

根据系统需求

定义系统技术性能指标

确定算法并进行算法模拟

选择DSP处理器

硬件设计

硬件调试

软件编程

软件调试

系统集成和测试

**5.TI公司的DSP主要有哪几大类？**

**答：TI公司为各种应用开发提供了多种数字信号处理（DSP）平台，其中包括TMS320C2000系列、TMS320C5000系列、TMS320C6000系列、DaVinci数字媒体处理器和OMAP应用处理器。**

1. **在设计DSP系统时，如何选择DSP？**

**答：对不同的应用场合，DSP处理器的选取也是不一样的。一般来说，选择DSP处理器时应考虑以下因素：DSP处理器的运算速度、DSP处理器的运算精度、DSP处理器的硬件资源、DSP处理器的开发工具、DSP处理器的功耗、DSP处理器的价格及售后技术服务、DSP处理器的支持多处理器功能、DSP应用系统的运算量和其他因素。**

1. **TMS320C54x的硬件结构**

**2.TMS320C54x的总线有哪些？它们各自的作用和区别是什么？**

**答：TMS320C54x 片内有8条16位主总线，即1条程序总线（PB）、3条数据总线（CB、DB和EB）和4条地址总线（PAB、CAB、DAB和EAB），这些总线形成了支持高速指令执行的硬件基础。8条16位总线的功能如下。**

**1）1条程序总线（PB）：程序总线（PB）传送由程序存储器取出的指令操作代码和立即操作数。**

**2）3条数据总线（CB、DB和EB）：3条数据总线（CB、DB和EB）将内部各单元（如CPU，数据地址生成电路，程序地址生成电路，片内外围设备以及数据存储器）连接在一起。其中，CB和DB用来传送从数据存储器读出的数据；EB用来传送写入存储器的数据。**

**3）4条地址总线（PAB、CAB、DAB和EAB）：4条地址总线（PAB、CAB、DAB和EAB）用于传送执行指令所需要的地址。**

**3.TMS320C54x的CPU包含哪些部分？它们的功能是什么？**

**答：TMS320C54x的CPU基本组成如下：**

**运算部件：**

**1）40位算术逻辑单元（ALU）：功能：TMS320C54x使用40位算术逻辑单元（ALU）和两个40位累加器（ACCA和ACCB）来完成二进制补码的算术运算和逻辑运算，且大多数都是单周期指令。**

**2）2个40位累加器A和B：用于存储ALU或乘法器/加法器单元输出的数据。累加器也能输出数据到ALU或乘法器/加法器中。**

**3）40位桶形移位寄存器：功能：桶形移位器能把输入的数据进行0到31位的左移和0到16位的右移。**

**4）17×17位乘法器和一个专用40位加法器：C54x CPU有一个17×17位的硬件乘法器，与40位的专用加法器相连，可以在单周期内完成一次乘法累加运算。**

**5）比较、选择和存储单元（CSSU）：C54x CPU有一个17×17位的硬件乘法器，与40位的专用加法器相连，可以在单周期内完成一次乘法累加运算。**

**6）指数编码器：指数编码器是用于支持单周期指令EXP的专用硬件。功能支持指令EXP和NORM完成规格化定点数操作。**

**控制部件：**

**1）状态寄存器ST0：ST0反映寻址要求和计算的中间运行状态**

**2）状态寄存器ST1：ST1反映寻址要求，计算的初始状态设置，I/O及中断控制。**

**3）处理器模式状态寄存器PMST：PMST主要设定并控制处理器的工作方式，反映处理器工作状态。PMST中的数据决定了C54X芯片的存储器配置情况，PMST寄存器通过存储器寻址的寄存器指令装载。**

1. **TMS320C54x 的CPU中累加器A和B的保护位AG和BG的作用是什么？**

**答：累加器A和B的保护位AG和BG作为计算时的高位余量位，用于防止在迭代运算（如自相关）中产生溢出，在进行有符号运算时为扩展符号位。**

1. **当标志位FRCT=1时，TMS320C54x CPU中的乘法器的乘积将作怎样的调整？说出这些调整在小数运算中的实际意义是什么？**

**答：由于乘法器在进行两个16位二进制补码相乘时会产生两个符号位，为提高运算精度，在状态寄存器ST1中设置小数方式控制位FRCT。当FRCT=1时，乘法器结果自动左移一位，消去多余的符号位，相应的定标值加1。**

1. **已知累加器A的值为FF FF81 5432h，暂存器T的值为0010h，执行指令EXP A和NORM A 后，累加器A和暂存器T的值各为多少？**

**答：累积器A的值为FF 8154 3200h，暂存器T的值为8h。**

**9.TMS320C54x 的总存储空间为多少？可分为哪几类？它们的大小是多少？**

**答：TMS320C54x 的总存储空间为192K字，分为3个可选择的存储空间：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。**

1. **简述三种存储器空间各自的作用是什么？**

**答：1）程序存储器空间用来存放要执行的指令和执行中所需的系数表。**

**2）数据存储器空间用来存放执行指令所要用的数据，包括需要处理的数据或数据处理的中间结果。**

**3）I/O存储器空间可与存储器映射外围设备相接口，也可以作为附加的数据存储空间使用。**

**12. I/O空间是在片内还是在片外？访问I/O的实质是什么？**

**答：I/O空间是在片外，访问I/O是对I/O映射的外部器件进行访问，而不是访问存储器。**

1. **数据页0（0h~7Fh）能否被映射到程序存储空间？**

**答： 不能**

1. **TMS320C54x 存储空间的配置是受、OVLY和DROM 3个位控制的。如果想使片上RAM同时映射到数据存储空间和程序存储空间，那么、OVLY和DROM的值应该如何设置？**

**答：OVLY=1**

**17. TMS320C54x片内外设主要有哪些？**

**答：通用I/O端口、时钟发生器、软件可编程等待状态发生器、可编程块切换逻辑、定时器、主机接口、串行口、DMA控制器等。**

**21. TMS320C54x复位的条件有哪些？通常TMS320C54x有几种复位方式？各是什么？上电复位后，第一条程序指令所在的地址是多少？**

**答：TMS320C54x复位期间，处理器进行以下操作：**

1. **处理器工作方式状态寄存器PMST中的中断向量指针IPTR被设置成1FFh。**
2. **处理器工作方式状态寄存器PMST中被设置成与引脚状态相同的值。**
3. **程序计数器PC设置成FF80h。**
4. **扩展程序计数器XPC被清零（如果XPC可用）。**
5. **不管位的状态如何，将FF80h加到地址总线上。**
6. **数据总线变为高阻状态。**
7. **控制线处于无效状态。**
8. **产生应答信号。**
9. **状态寄存器ST1中的中断方式位INTM置1，关闭所有可屏蔽中断。**
10. **中断标志寄存器IFR被清零，以清除中断标志。**
11. **单指令重复计数器（RC）被清除。**
12. **产生同步复位信号，用于初始化片内外设。**
13. **状态寄存器ST0=1800h，即以下的状态位被设置成它们的初始值：ARP=0，TC=1，C=1，OVA=0，OVB=0，DP=0。**
14. **状态寄存器ST1=2900h，即以下的状态位被设置成它们的初始值：BRAF=0 ，CPL=0，XF=1，HM=0，INTM=1，OVM=0，SXM=1，C16=0，FRCT=0，CMPT=0，ASM=0。**
15. **处理器工作方式状态寄存器PMST以下的状态位被设置成它们的初始值：OVLY=0，AVIS=0，DROM=0，CLKOFF=0。**

**需要注意的是，复位期间，其余的状态位和堆栈指针SP没有被初始化，需要使用用户程序对它们进行初始化。如果=0，那么处理器从片内ROM开始执行程序，否则，处理器从片外程序存储器开始执行程序。**

**TMS320C54x的复位有两种方式，分别为软件复位和硬件复位。软件复位是通过执行指令实现处理器的复位；硬件复位是通过硬件复位电路实现处理器的复位。硬件复位电路包括上电复位、手动复位和自动复位。**

**上电复位后，第一条程序指令所在的地址是FF80h。**

**22. 在闲置方式1（IDLE1）、闲置方式2（IDLE2）、闲置方式3（IDLE3）和保持方式这4种省电方式中，哪一种最省电？哪几种能够被内部中断唤醒？**

**答：闲置方式3（IDLE3）最省电。闲置方式1（IDLE1）能够被内部中断唤醒**

**23. TMS320C54x中断分为哪几类？其中的可屏蔽中断在什么情况下可以被CPU响应？**

**答：TMS320C54x 支持软件中断和硬件中断。软件中断由程序指令产生（INTR或TRAP）。硬件中断由设备的一个信号产生，硬件中断包含两种类型，分别为外部硬件中断和内部硬件中断。外部硬件中断由外部中断接口的信号触发；内部硬件中断由片内外设的信号触发。**

**硬件或软件中断发送了一个中断请求后，CPU必须要决定是否响应该中断。软件中断和不可屏蔽中断立即被响应，而可屏蔽的硬件中断只有在满足以下3种条件时才能被响应：**

1. **优先级最高。当同时有多个硬件请求中断时，TMS320C54x根据优先级对其进行响应。**
2. **状态寄存器ST1中的INTM位为0。表示允许可屏蔽中断，可以用“RSBX INTM”指令来对INTM复位。**
3. **中断屏蔽寄存器IMR中的相应位为1。**

**CPU响应中断时，让PC转到适当的地址取出中断向量，并发出中断向量信号，清除响应的中断标志位。**

**26. TMS320C54x的流水线操作分为几个阶段？分别叙述流水线操作各阶段的功能？**

**答：流水线的6个操作阶段分别为预取指（P）、取指（F）、译码（D）、访问（A）、读数（R）和执行（X）。每个流水线操作阶段各占用一个机器周期。各操作阶段的功能如下：**

1. **程序预取指（P）：将下一条指令的地址，放在程序地址总线（PAB）上。**
2. **程序取指（F）：从程序总线（PB）上取指令字，并将该指令字放入指令寄存器（IR）中。**
3. **译码（D）：将指令寄存器（IR）中的内容译码，确定要访问存储器的类型以及数据地址产生单元（DAGEN）和CPU的控制时序。**
4. **访问（A）：数据地址产生单元（DAGEN）在数据地址总线（DAB）输出要读的操作数的地址。如果还有第二个操作数，则在另一个数据地址总线CAB上输出相应的地址。同时更新间接寻址模式下的辅助寄存器（ARx）和堆栈指针（SP）。**
5. **读数（R）：从数据总线DB和CB上读取操作数，完成操作数的读取。同时，操作数的写入开始。如果需要写数据，则写数据的地址放在数据写地址总线（EAB）上。对存储器映射寄存器而言，数据是从存储器中读取，写数据时通过DB写入选择的存储器映射寄存器。**
6. **执行（X）：在这个阶段完成指令的执行，并将数据放在数据写总线（EB）上完成操作数的写入。**
7. **TMS320C54x 指令系统**
8. **TMS320C54x提供哪几种数据寻址方式？各有什么特点？应该应用在什么场合？**

**答：TMS320C54x 提供了以下7种基本的数据寻址方式：**

1. **立即数寻址：指令中直接包含了所需要的操作数。**
2. **绝对寻址：指令中包含所要寻址的存储单元的地址。**
3. **累加器寻址：利用累加器的数值作为地址来读写程序存储器。**
4. **直接寻址：指令中包含数据存储器地址的低7位。这7位作为偏移地址与数据页指针DP或堆栈指针SP相结合共同构成16位的数据存储器实际地址。**
5. **间接寻址：根据辅助寄存器的内容来寻找存储器映射寄存器地址，地址的低7位来自指令或某个辅助寄存器的低7位，高9位设置为0。**
6. **存储器映射寄存器寻址：修改存储器映射寄存器中的值而不影响当前数据页指针DP或堆栈指针SP的值。**
7. **堆栈寻址方式：地址来自堆栈指针SP，把数据压入和弹出系统堆栈。**

**2.绝对地址寻址有哪几种？它们可以访问哪些地址空间？有什么特点？其中的长立即数寻址方式的指令能不能与循环指令RPT和RPTZ一起使用？**

**答：绝对寻址有以下四种类型：**

1. **数据存储器地址寻址：数据存储器地址（dmad）寻址是用一个符号（符号地址）或一个表示16位地址的立即数来指明寻址的数据存储单元的16位绝对地址。**
2. **程序存储器地址寻址：程序存储器地址（pmad）寻址是用一个符号（符号地址）或一个表示16位地址的立即数来指明寻址的程序空间的地址。**
3. **端口地址寻址：端口地址（PA）寻址是用一个符号或一个数值来指明其外部I/O口地址。**
4. **长立即数\*(lk)寻址：长立即数\*(lk)寻址是用一个符号或一个数值来指明寻址的数据存储空间的地址。这种寻址方式可以用在所有支持单数据存储器（Smem）操作数的指令中。**

**使用长立即数寻址方式的指令不能与单循环指令RPT和RPTZ一起使用。**

1. **直接寻址方式有两种方式，它们是什么？如何控制？当SP=2000h，DP＝2，偏移地址为25h时，分别寻址的是哪个存储空间的哪个地址单元？**

**答：直接寻址方式包括DP直接寻址和SP直接寻址两种方式。当CPL=0时，寻址方式为以DP为基地址的直接寻址方式，简称DP直接寻址；当CPL=1时，寻址方式为以堆栈指针SP为基地址的直接寻址方式，简称SP直接寻址。**

**当CPL=0时，DP＝2，dmad=25h，实际地址为0100h+25h=0125h。**

**当CPL=1时，SP＝2000h，dmad=25h，实际地址为2000h+25h=2025h。**

1. **当使用位倒序寻址时，应使用什么辅助寄存器？试述地址以位倒叙方式产生的过程。**

**答： 位倒序寻址是DSP的一种特殊处理方式，是专门为快速傅立叶变换FFT而设计的，这种寻址方式可以显著提高程序的执行速度和存储区的利用效率。使用时，AR0存放的整数N为FFT点数的一半，另一个辅助寄存器ARx指向数据存放的单元，当使用位倒序寻址把AR0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。**

1. **在循环寻址方式中，如何确定循环缓冲的起始地址？如循环缓冲大小为32，其起始地址必须从哪开始？**

**答：长度为R的缓冲器必须从N位地址的边界开始（即循环缓冲器基地址的N个最低有效位必须为0），N是满足2N>R条件的最小整数。长度R=32的循环缓冲器必须从地址XXXX XXXX XX00 00002（N=6，26>32，该地址的最低6位为0）开始，同时32必须存入BK。**

1. **若辅助寄存器AR0的值为0x0010h，AR3的值为0x0310h，循环缓冲起始地址为0300h，BK=31，请分别给出下列寻址方式修改后的辅助寄存器的值。**
2. **\*AR3+% (2) \*AR3+0% (3) \*AR3-0% (4) \*+AR3(-2) (5) \*AR0(0100H)**

**答：寻址方式修改后的辅助寄存器的值**

**(1) AR3＝0X0311H （0X0300H<0X0311H<0X031FH)**

**(2) AR3＝0X0310H+0X0010H=0X0320H>0X031FH 则：AR3＝0X0320H-1FH= 0X0301H**

**(3) AR3=0X0310H-0X0010H=0X0300H**

**(4) AR3=0X0310-2= 0X030EH**

**(5) AR0=0X0100H**

1. **双数据存储器操作数间接寻址使用哪几种类型？所用辅助寄存器只能是哪几个？其特点是什么?**

**答：双数据存储器操作数间接寻址类型为\*ARx、\*ARx-、\*ARx+、\*ARx+0%。所用4个辅助寄存器为AR2、AR3、AR4、AR5。双数据存储器操作数间接寻址的特点是：占用程序空间小，运行速度快，在一个机器周期内通过2个16位数据总线（C和D）读两个操作数。**

1. **堆栈寻址的作用是什么？压栈和弹出堆栈操作是如何实现的？**

**答：当发生中断或子程序调用时，堆栈用来自动地保存程序计数器PC的值。堆栈也可以用来保护现场或传送参数。堆栈寻址，就是利用堆栈指针，按照先进后出的原则来寻址。SP总是指向压入堆栈的最后一个数据。堆栈寻址的作用是保护调用，中断现场信息，进行数据传输。在执行压入堆栈操作时，SP先减1，然后将数据压入堆栈；在执行弹出堆栈操作时，数据从堆栈中弹出后，SP再加1。**

1. **TMS320C54x的软件开发与设计**
2. **简述TMS320C54x 应用软件开发过程。**

**答：TMS320C54X DSP软件开发的流程主要包括以下步骤：**

**(1)用C语言或汇编语言编写程序,后缀分别为.C或 .asm, 在asm文件中,除了DSP的指令外还有伪指令;**

**(2)用汇编器由asm文件生成obj文件, 或者用C编译器把C文件翻译成asm文件，再生成obj文件；**

**(3)用链接器根据链接命令文件(cmd文件)将多个obj文件及库文件链接起来,并分配各程序段、数据段的地址，生成的out文件可供模拟/仿真；**

**(4)用模拟器/仿真器对out文件进行分析验证；**

**(5)用代码格式转换工具将out文件转化成可写入EPROM的数据文件。**

1. **简述汇编器、链接器的功能。**

**答：汇编器（Assembler）用于把汇编语言源文件翻译成机器语言目标文件，机器语言格式为公用目标格式（COFF）。链接器（Linker）用于把多个目标文件组合成单个可执行目标模块。它一边创建可执行模块，一边完成重定位以及决定外部参考。链接器的输入是可重定位的目标文件和目标库文件。**

**4.链接器对段是如何处理的？**

**答：链接器是开发TMS320C54x器件必不可少的开发工具之一，它对段处理时有2个主要任务：一个是将一个或多个COFF目标文件中的各种段作为链接器的输入段，经链接后在一个执行的COFF输出模块中建立各个输出段；另一个是在程序装入时对其重新定位，为各个输出段选定存储器地址。**

**链接器通过链接命令文件来控制对段的处理，命令文件为ASCⅡ文件，可包含以下内容：**

**控制链接的输入文件名、指定目标文件、存档库或其他命令。**

**链接器选项，它们在命令文件中的使用方法与在命令行中相同。**

**MEMORY和SECTIONS链接伪指令，用来指定目标存储器结构和地址分配。**

**10.请解释链接命令“lnk500 file1.obj file2.obj -o link.out -m lm.map”完成的功能。**

**答：该命令是将两个目标文件file1.obj和file2.obj进行链接，生成一个可执行的输出文件link.out和一个映射文件lm.map。**

1. **CCS集成开发环境及其使用**
2. **CCS集成开发环境由哪几部分构成？它都有哪些功能？与原先的DSP开发软件相比有哪些优势？**

**答：CCS由以下5部分组件构成：**

1. **代码生成工具。它是CCS所提供的开发环境的基础，用来对C语言、汇编语言或混合语言编程的DSP源程序进行编译汇编，并链接成为可执行的DSP程序。主要包括汇编器、链接器、C/C++编译器和建库工具等。**
2. **CCS集成开发环境。CCS集成开发环境集编辑、编译、链接、软件模拟、硬件在线仿真调试和实时跟踪等功能于一体，包括编辑工具、工程管理工具和调试工具等。**
3. **DSP/BIOS实时内核插件及其应用程序接口API。它们主要为实时信号处理应用而设计，包括DSP/BIOS的配置工具、实时分析工具等。**
4. **实时数据交换的RTDX插件和相应的程序接口API。它们可对目标系统数据进行实时监视，实现DSP与其他应用程序的数据交换。**
5. **由TI公司以外的第三方提供的应用模块插件。**

**早期的DSP开发工具没有集成化，需要在DOS环境下键入比较复杂的命令，使用起来不方便，调试、开发的效率也不高。 Code Composer Studio（简称CCS）是TI公司推出的用于开发DSP的集成开发环境，它采用Windows风格界面，集编辑、编译、链接、软件模拟、硬件仿真调试以及实时跟踪等功能于一体，支持汇编语言与C语言及二者的混合编程，极大地方便了DSP的开发与设计。**

1. **CCS的Simulator和Emulator有何区别，在哪些情况下适合使用Simulator调试程序，哪些情况下必须使用Emulator调试程序？**

**答：Simulator软件模拟器模式，即脱离DSP处理器，在PC机上模拟DSP的指令集和工作机制，主要用于前期算法实现和调试；Emulator硬件在线编程模式，即实时运行在DSP处理器上，与硬件开发板相结合在线编程和调试应用程序。**

**8.链接配置文件（\*.cmd）的作用是什么？在开发工程中可以不用该文件吗？**

**答：cmd文件是链接命令文件,包括两条命令:**

**MEMORY命令,用来定义目标系统的存储器配置图,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;**

**SECTIONS命令,此命令告诉链接器如何将输入段组合成输出段,以及将输出段放在存储器的什么位置.**

**链接器在处理段的时候,有2个主要任务:把一个或多个COFF目标文件中的各种段作为链接器的输入段,经链接后在一个可执行的COFF输出模块中建立各个输出段;为各个输出段选定存储器地址**

**在开发工程中可以不用该文件。**

1. **从Sample.c源程序到最终的Sample.out，中间需要经过哪些步骤？**

**答：编译、汇编、链接。**