课程复习

主要四部分内容：

（1）DSP2407概述

（2）系统概貌：系统配置和中断、存储器和I/O空间、时钟和低功耗模式、数字输入输出

（3）片内外设：事件管理器、ADC、SCI、SPI、CAN、WD

（4）系统开发和基于C语言的软件设计

第一部分 DSP2407概述

主要内容：

（1）DSP与其他处理器（单片机、PC）相比独有的特点

（2）DSP能进行快速的数字信号处理运算的原因

（3）TMS320LF2407A内部资源

（4）TMS320LF2407A引脚定义

R1.1 DSP的特点

（1）哈佛结构；（2）流水线结构；

（3）硬件乘法器和特殊的数字信号处理指令

R1.2 TMS320LF2407A片内资源

（1）10位（双8路或单16路）A/D转换器，转换时间375nS；

（2）41个可独立编程的数字I/O引脚；

（3）带锁相环PLL的时钟模块；

（4）看门狗定时器模块；

（5）串行通信接口SCI与串行外设接口SPI；

（6）两个事件管理器EVA、EVB，可为所有类型电机提供控制技术，在工业自动化方面的应用奠定了基础：

* 2个16位通用定时器；
* 3个具有死区功能的全比较单元；

* 3个事件捕捉单元，其中2个具有直接连接光电编码器脉冲的能力；

* 8个16位PWM通道--三相反相器控制；

（7）5个外部中断（两个驱动保护、复位、两个可屏蔽中断）；

（8）CAN 2.0B 模块

（9）用于仿真的JTAG接口。

（10）片内存储器：32K字闪存、2.5K字RAM

R1.3 TMS320LF2407A引脚定义

- § 地址、数据、存储器控制信号
- § 事件管理器A（EVA）
- § 事件管理器B（EVB）
- § 模数转换器
- § CAN、SCI、SPI
- § 外部中断、时钟
- § 振荡器、锁相环、闪存、引导及其他
- § 仿真和测试（JTAG）
- § 电源电压

第二部分 系统概貌

主要内容：

（1）DSP2407的中断系统

（2）DSP2407的存储器映射

（3）DSP2407锁相环的工作方式

（4）DSP2407数字I/O引脚的使用

R2.1 DSP2407的中断系统

（1）众多中断源

（2）包含不可屏蔽中断和六个中断优先级的可屏蔽中断

（3）PIE外部中断扩展控制器管理

（4）两级中断模式（外设中断向量寄存器PIVR）

（5）两级中断服务函数（GISR/SISR）

R2.2 DSP2407的存储器映射

R2.2.1 寻址能力

可寻址三个独立的空间：程序空间、数据空间和I/O空间，每个空间最大寻址能力为64K。

绝大部分外设控制寄存器都映射在片内数据空间，只有等待状态发生器映射在片内I/O空间。

R2.2.2 片内存储器资源

FLASH:　32K，程序空间

RAM:　2.5K

SARAM:　2K，程序空间 / 数据空间

DARAM:　544字

B0:　256字，程序空间 / 数据空间

B1:　256字，数据空间

B2:　32字，数据空间

R2.2.3 DSP2407锁相环的工作方式

（1）**内部时钟**：只需外接晶体振荡器

（2）**外部时钟**：把外部时钟接至XTAL1/CLKIN脚

R2.2.4 DSP2407数字I/O引脚的使用

（1）最多**41个**数字I/O引脚，绝大部分有复用功能

（2）区别I/O引脚与I/O空间

（3）使用数字I/O引脚：

- 配置引脚功能
- 设定I/O方向
- 读写数据位

第三部分 片内外设

R3.1 事件管理器主要内容：

（1）事件管理器的功能部件及用途

（2）通用定时器：功能 / 工作方式 / 比较输出

（3）比较单元的功能

（4）捕捉单元和光电正交编码器输入单元的基本原理

R3.1.1 事件管理器的功能部件及用途

有两个事件管理器EVA和EVB，每个事件管理器包括：

（1）2个通用定时器：计数 / 定时、提供时基、比较输出

（2）3个比较单元：死区控制、SVPWM控制、波形发生器

（3）3个捕捉单元：记录事件时刻

（4）1个光电编码器解码模块：获取外部旋转机械的方向、速度、位置等信息

（5）三组中断，占用中断级别2，3，4

比较单元只能用定时器1 / 3，光电编码器只能用于定时器2 / 4。

R3.1.2 通用定时器

（1）通用定时器的功能：计数 / 定时、比较、提供时基

（2）通用定时器的四种工作方式：

- 停止 / 保持模式
- 连续增计数模式
- 定向增减计数模式
- 连续增减计数模式

（3）通用定时器的比较输出

每个通用定时器有1个比较输出引脚，可以输出对称 / 非对称的PWM波形，具有输出控制逻辑

R3.1.3 比较单元

使用定时器1或定时器3作为时基，每个比较单元有两个PWM输出引脚。与定时器的比较输出不同，比较单元的输出具有死区控制和SVPWM控制功能，在早期的产品中也称为全比较单元。

R3.1.4 捕捉单元和光电正交编码器

（1）捕捉单元两级FIFO堆栈

（2）光电正交编码器解码模块可以为定时器2或定时器4提供计数方向和计数时钟，输出时钟是引脚输入信号的四倍频。使用光电正交编码器作为定时器计数时钟和计数方向控制时，定时器必须工作在定向增 / 减计数模式。

R3.2 ADC模块

（1）10bit精度，375ns转换时间

（2）带有内部采样保持器

（3）16路输入，两个8状态排序器（SEQ1和SEQ2）或级联成1个16状态排序器（SEQ）

（4）有多个启动ADC转换的触发源：

§ 软件立即启动

§ EVA事件管理器启动

§ EVB事件管理器启动

§ ADC的SOC引脚启动

R3.3 SCI串行异步通信接口

（1）异步，全双工

（2）可编程的数据位数和停止位、奇偶校验位

（2）16位波特率发生器，最高波特率2.5M

（3）两种唤醒多处理器协议

§ 空闲线模式：适合于大数据块传输

§ 地址位模式：适合于小数据块传输

（4）三种常见的串行通信协议RS-232、RS-422、RS-485

R3.4 SPI串行同步外设接口

（1）通信中分为主控制器 / 从控制器

（2）125种不同的波特率，最大波特率为CLKOUT的四分之一

（3）四种时钟模式

- § 上升沿，无延时
- § 上升沿，有延时
- § 下降沿，无延时
- § 下降沿，有延时

R3.5 WD看门狗定时器

（1）8位计数器，上溢时产生一个系统复位信号

（2）WD的时钟为CLKOUT的1／512，可以进行预定标

第四部分 系统开发和基于C语言的软件设计

主要内容：

（1）建设仿真调试环境

（2）软件开发环境

（3）软件开发语言

（4）COFF文件格式

（5）CCS工程中的各种文件

（6）DSP C语言

R4.1 建设DSP仿真调试环境

（1）仿真RAM

（2）MP/MC*选择

（3）VccP（5V）

R4.2 软件开发环境

（1）CC-Code Composer, no DSP/BIOS

（2）CCS-Code Composer Studio, with DSP/BIOS

R4.3 软件开发语言

（1）汇编语言

高效指令，代码效率高、底层控制灵活、实时性好

指令集掌握困难，程序可读性、可维护性、可移植性差，流程控制困难，开发周期长

适用于运算量大、实时性要求高的场合

（2）C语言

程序可读性、可维护性、可移植性好，修改、升级方便，流程控制容易，开发周期短

某些硬件控制不便，实时性不好

适用于运算量小，实时性要求不高的场合

（3）汇编与C混合编程

综合利用两种语言的优越性，用C语言实现流程控制，用汇编语言实现时序或效率要求严格的核心程序

R4.4 COFF文件格式

（1）按照“段”对程序进行组织，有利于模块化编程

（2）段是程序中的一个数据或代码块，最终实现为存储器中的一部分连续空间

（3） COFF目标文件至少包含以下三个默认段：

.text段（文本段）通常包含可执行代码

.data段（数据段）通常包含初始化的数据

.bss段 （保留空间段）通常为没有初始化的变量保留空间

R4.5 CCS工程中的各种文件

.pjt　　(CCS工程文件）

.mak　　（CC工程文件)

.cmd　　连接命令文件（定义存储器空间并确定输出各段的位置）

.lib　　库文件（开发环境自带）

.h　　头文件（DSP寄存器定义文件等）

.c　　c源文件

.asm　　汇编程序文件（中断向量定义）

.obj　　编译后的目标文件（COFF）

.out　　可在目标DSP上执行的文件（COFF）

R4.6 DSP C语言

DSP C语言以ANSI C为基础，并对ANSI C进行了相应的限定和扩展。

R4.6.1 DSP C语言与ANSI C的主要不同

（1）没有16位以下的数据类型

（2）增加关键字：ioport，interrupt，register等

（3）内嵌汇编语句　asm(" clrc　INTM");

R4.6.2 DSP C语言访问I/O空间

（1）用关键字“ioport”对要访问的地址进行定义

ioport *type* port*hex_num*

（2）定义后的I/O端口访问和一般变量访问没有区别

```
/*************************************************************/
   ioport unsigned int port10; /* 访问I/O端口10h的变量 */
   int func ()
   {
        ...
        port10 = a;    /* 写 a到端口 10h */
        ...
        b = port10;   /* 读取端口10h的值到 b */
        ...
   }
/*************************************************************/
```

R4.6.3 DSP C语言访问数据空间

访问数据空间不需要对要访问的单元预先声明，访问是通过指针的方法实现的。

```
val = *(unsigned int *)addr;
*(unsigned int *)addr = val;
```

用这样的访问方法容易实现循环：

```
/*************************************************************/
pInt= (int * ) 0x201;
for(i = 0;i<4;i++)
{
   * (pInt+i) = i+1;
}
/*************************************************************/
```

R4.6.4 DSP C语言定义中断服务函数

有两种方式定义中断服务函数：a)任何具有名为c_int*d* 的函数（*d*为0到9的数），都被假定为一个中断程序，c_int0函数留作系统复位中断用。

b)利用中断关键词interrupt进行定义。举例如下：

```
/*********************************************************/
void c_int1 ()
{
     ......
}
/*********************************************************/
interrupt void isr ()
{
     ......
}
/*********************************************************/
```

R4.6.5 DSP C语言与汇编语言混合编程

有以下几种混合编程方法：

- C语言调用汇编语言编写的函数
- 使用内嵌汇编语句（asm语句）
- C语言访问汇编语言变量
- 手动修改C语言程序编译后生成的汇编代码