DSP 教材课后习题整理

> 文 / W.D.Gaster，2025.12.23
>
> 说明：本文档题目来自教材《TMS28335 DSP 原理、开发与应用》符晓、朱洪顺编著，清华大学出版社。

一、电机控制DSP简介

1. 简要描述DSP芯片的主要特点。

DSP芯片（数字信号处理器）是专为快速实现数字信号处理算法而设计的微处理器，其主要特点包括：

- **哈佛总线结构**：程序和数据拥有独立的总线与存储空间，可同时访问，提高吞吐率。
- **专用硬件乘法器与乘加操作**：单指令周期内可完成一次乘法或一次乘累加（MAC）运算。
- **支持流水线操作**：使取指、译码、执行等操作重叠进行，提高指令执行效率。
- **片上快速RAM**：可通过独立总线同时访问多个存储块。
- **硬件支持高效循环与寻址**：具有低开销循环跳转的硬件支持和单周期内操作的多个硬件地址产生器。
- **快速中断处理**：具备快速的中断处理和硬件I/O支持。

2. 搜集主流电机控制芯片的分类与特点。

电机控制芯片通常指用于驱动和闭环控制的专用集成电路或微控制器。

- **按电机类型分类**：
  - **直流电机驱动芯片**：驱动有刷/无刷直流电机，常用PWM调速，具有过流保护功能。
  - **三相无刷直流电机（BLDC）驱动芯片**：提供精确的电流与速度控制，集成PWM、换相逻辑及保护电路。
  - **步进电机驱动芯片**：提供全步、半步、微步控制，具有细分和高精度定位功能。
  - **交流伺服电机驱动芯片**：用于高精度位置、速度控制的交流伺服电机，通常支持闭环反馈（如编码器接口）。
- **按集成度分类**：
  - **专用驱动IC**：集成功率级（如MOSFET驱动器）和保护电路，需配合主控MCU工作。
  - **集成控制器（如DSP/MCU）**：以TI C2000系列为代表，单芯片集成高性能CPU、PWM、ADC、编码器接口等，实现全数字控制。

3. 整理出C2000系列DSP的具体分类与型号。

TI C2000系列是面向实时控制（如电机、数字电源）的32位微控制器（也称DSC）。以F2833x/F2823x子系列为例，其型号对比如下：

型号	内核类型	片上Flash (16位)	片上SARAM (16位)	主要特征
F28335/F28235	浮点(F28335) / 定点(F28235)	256K	34K	高性能，外设最全

型号	内核类型	片上Flash (16位)	片上SARAM (16位)	主要特征
F28334/F28234	浮点(F28334) / 定点(F28234)	128K	34K	性能与配置居中
F28332/F28232	浮点(F28332) / 定点(F28232)	64K	26K	入门级，满足基本控制需求

> **说明**：F2833x子系列包含**浮点运算单元（FPU）**，而F2823x子系列为**定点内核**，这是它们最核心的区别。

4. 以常见的176引脚 PGF/PTP 薄型四方扁平封装（LQFP）的 F28335 为例，找出与SCI有关的通用I/O引脚。

在F28335上，串行通信接口（SCI） 即UART，其信号通过多路复用的GPIO实现。对于常见的176引脚PGF/PTP封装，主要SCI模块的引脚复用关系如下：

- **SCIA**：可映射到 `GPIO28(SCITXDA) / GPIO29(SCIRXDA)` 或 `GPIO35(SCITXDA) / GPIO34(SCIRXDA)`。
- **SCIB**：可映射到 `GPIO9(SCITXDB) / GPIO11(SCIRXDB)` 或 其他备用引脚。
- **SCIC**：可映射到 `GPIO62(SCITXDC) / GPIO63(SCIRXDC)`。

> **注意**：具体使用哪个引脚，需要通过GPIO多路开关控制寄存器（GPxMUX）进行软件配置，将相应引脚功能设置为SCI模式。

5. TMS320F28335的地址线是什么？并据此计算可寻址的最大地址范围。

- **地址线**：程序读总线22位地址线，数据读总线32位地址线，数据写总线32位地址线。
- **最大寻址范围**：程序空间以16位字word为单位，$2^{22} \times 2 = 8388608Bytes = 8MB$，数据空间以字节为单位，$2^{32}Byte = 4GB$

6. TMS320F28335的数据线有多少根?

程序读总线32位数据线，数据读总线32位数据线，数据写总线32位数据线。

7. 描述模拟电源与数字电源，模拟地与数字地的区别。

- **模拟电源/地 (AVDD/AVSS, VDD1A8/VSS1A8等)**
  - **用途**：为芯片内部的模拟电路供电，例如模数转换器（ADC）、锁相环（PLL）等。
  - **要求**：对噪声极其敏感。需要干净、稳定的电源，通常需通过磁珠或电感从数字电源隔离，并配合高品质的滤波电容。
- **数字电源/地 (DVDD/DVSS, VDD/VSS等)**
  - **用途**：为数字逻辑电路（如CPU、存储器、数字外设）供电。
  - **特点**：由于数字电路开关动作，会产生大量高频噪声。
  - **设计关键**：必须在单点（通常是芯片下方）将模拟地和数字地连接，以避免数字噪声串扰影响模拟电路的精度，形成“星型”接地。

8. 描述主频与时钟周期的关系，并分别计算 CPU 时钟频率为150MHz、100MHz、50MHz和30MHz情况下的时钟周期。

- **关系**：时钟周期是时钟频率的倒数。T=1/f。
- **不同CPU频率下的时钟周期**：
  - 150MHz：周期 = 1 / 150,000,000 ≈ 6.67 ns
  - 100MHz：周期 = 1 / 100,000,000 = 10 ns
  - 50MHz：周期 = 1 / 50,000,000 = 20 ns
  - 30MHz：周期 = 1 / 30,000,000 ≈ 33.33 ns

9. 查找TMS320F28335的各种封装，并描述各自的优点、缺点。

封装型号	描述	优点	缺点
PGF (176-LQFP)	176引脚薄型四方扁平封装	成本较低，易于手工焊接和视觉检查。	散热性能一般，引脚间距相对较大，板子面积较大。
PTP (176-HLQFP)	176引脚热增强型四方扁平封装	带有暴露的散热焊盘，散热性能优于PGF。	比PGF略贵，焊接需处理散热焊盘。
ZJZ (176-BGA)	176焊球塑料球栅阵列	占用PCB面积小，电气性能好（引线电感小）。	无法手工焊接，需专业设备；焊点目视检查困难。
ZHH (179-BGA)	179焊球MicroStar BGA™	尺寸更小，球间距更细。	焊接和PCB布线要求最高，返修难度极大。

10. 与JTAG相关的引脚有几个?

5个必需引脚：**TMS**：测试模式选择；**TCK**：测试时钟；**TDI**：测试数据输入；**TDO**：测试数据输出；**TRST**：测试复位
此外，需要**EMU0**和**EMU1**仿真器引脚。因此，相关的引脚共7个。

11. TMS320F28335的片上存储器有哪些?

1. **Flash存储器**：256K × 16位，用于存储用户程序，可分区擦写。
2. **SARAM**：共34K × 16位，划分为多个块，可灵活配置为程序或数据存储器。
3. **用户OTP**：1K × 16位一次性可编程存储器。
4. **BootROM**：8K × 16位，由TI固化引导加载程序。

12. TMS320F28335的片上RAM被划分为哪些区间?

- **M0 SARAM**：1K × 16位 (地址范围: 0x00 0040 - 0x00 03FF)
- **M1 SARAM**：1K × 16位 (地址范围: 0x00 0400 - 0x00 07FF)
- **L0-L7 SARAM**：每个块4K × 16位
  - **L0-L3**：受代码安全模块(CSM)保护，具有双映射地址。
  - **L4-L7**：也可作为DMA存储器块使用。

13. TMS320F28335的外部存储器被划分为哪些区间?

外部存储器通过XINTF接口扩展，其地址空间被划分为多个可配置的区（Zone），每个区可独立设置等待状态等时序参数。包括：**XINTF Zone 0** 4K × 16位、**XINTF Zone 6** 1M × 16位、**XINTF Zone 7** 1M × 16位。

14. 代码安全模块(CSM)的作用是什么?

主要作用是**防止逆向工程，保护知识产权。**

15. 描述TMS320F28335主要的片上外设。

- **ePWM**：6个增强PWM模块。
- **eCAP**：6个增强的捕捉模块。
- **eQEP**：2个增强的正交编码模块。
- **ADC**：增强AD采样模块，12位精度、16位通道、80ns转换时间。
- **Watchdog Timer**：看门狗模块。
- **McBSP**：两个多通道串行缓存接口，连接高速外设。
- **SPI**：1个串行外设接口。
- **SCI**：3个串行通信接口，完成UART功能，外接电平转换电路，可以实现RS232或RS485通信。
- **I2C**：集成电路模块总线。
- **CAN**：2个增强的控制局域网功能。
- **GPIO**：增强的通用IO接口，控制寄存器可以切换三种不同的信号模式。
- **DMA**：6通道直接存储器存取，不经过CPU直接与外设、存储器进行数据交换。

二、软件开发平台与编程方法

1.概述基于CCS的软件开发流程

CCS（Code Composer Studio）是德州仪器（TI）为旗下DSP、MCU等处理器提供的集成开发环境。基于CCS的典型开发流程遵循“编辑-编译-链接-调试”的循环模式，其核心阶段如下：

1. **工程创建与配置**：在CCS中创建一个新工程，选择对应的目标器件（如TMS320F28335），并配置编译器、链接器选项及头文件、库文件路径。
2. **源代码编辑**：编写或修改C/C++及汇编语言源代码文件（`.c`，`.cpp`，`.asm`）和链接命令文件（`.cmd`）。
3. **编译与汇编**：编译器（C/C++）和汇编器（汇编）分别将高级语言和汇编语言源代码翻译成机器可识别的**目标文件**（`.obj`），此阶段进行语法检查并生成可重定位的代码段和数据段。
4. **链接**：链接器根据`.cmd`文件的指令，将所有`.obj`文件以及库文件（`.lib`）中的代码段和数据段，具体地分配到目标处理器的物理存储器地址上，解决符号引用，最终生成一个完整的、可下载到芯片执行的**可执行输出文件**（`.out`）。
5. **调试与验证**：通过仿真器（如XDS100/200）将`.out`文件下载到目标板或模拟环境中，利用CCS的调试工具（如断点、单步执行、内存/寄存器查看、图形显示）进行功能与性能验证。
6. **优化与迭代**：分析程序运行结果，优化算法、代码结构或存储器配置，重复步骤2-5直至满足设计要求。

2.描述编译器、汇编器和链接器的作用。

- **编译器**：将用**C/C++等高级语言**编写的源代码转换为目标处理器的**汇编语言代码**。它负责语法和语义分析、代码优化（如优化循环、删除无用代码），并生成与具体硬件指令集相关的中间汇编代码。
- **汇编器**：将**汇编语言源代码**（`.asm`）直接翻译成**机器语言目标文件**（`.obj`）。它执行一对一的指令翻译，将助记符（如`MOV`，`ADD`）转换为处理器可执行的二进制操作码。
- **链接器**：将编译和汇编生成的多个分散的`.obj`文件，以及库文件，组合成一个单一的**可执行程序**。它的核心作用是**地址解析**：根据`.cmd`文件的存储器映射描述，为所有的程序代码、变量和数据分配**具体的物理存储器地址**，并解析不同文件模块之间的符号引用（如函数调用、全局变量访问）。

3.描述cmd文件的用途。

`.cmd`文件（链接命令文件）是CCS开发流程中链接阶段的关键配置文件，其核心用途是**定义和分配系统的物理存储空间**。它主要包含两部分：

1. **MEMORY指令**：描述目标芯片**实际可用**的物理存储器资源，包括各存储器块（如Flash, RAM）的名称、起始地址和长度。

   ```
   MEMORY
   {
      PAGE 0: /* 程序存储空间 */
      RAML0      : origin = 0x008000, length = 0x001000
      ROM        : origin = 0x3FF000, length = 0x000FC0

      PAGE 1: /* 数据存储空间 */
      RAMM1      : origin = 0x000400, length = 0x000400
   }
   ```

2. **SECTIONS指令**：规定如何将输入文件（`.obj`）中各种类型的**程序段和数据段**（如`.text`代码段，`.data`已初始化数据段，`.bss`未初始化变量段）放置到MEMORY指令定义的物理存储器块中。

   ```
   SECTIONS
   {
      .text        : > RAML0,    PAGE = 0  /* 将代码段放入RAML0 */
      .cinit       : > ROM,      PAGE = 0  /* 将常量初始化表放入ROM */
      .stack       : > RAMM1,    PAGE = 1  /* 将栈放入RAMM1 */
   }
   ```

   简言之，`.cmd`文件是连接**程序员逻辑代码视图**与**芯片物理硬件布局**的桥梁，确保程序在正确的地址上运行。

4. 计算 16 位字长的情况下，-5、10 的 2 的补码形式。

- **+10**：原码为`0000 0000 0000 1010`，即其补码（`0x000A`）。
- **-5**：`5`的原码为`0000 0000 0000 0101`，反码为`1111 1111 1111 1010`，则补码为`1111 1111 1111 1011`（`0xFFFB`）

5. 计算 16 位字长的情况下，-1.9、0.9 的 2 的补码形式。

- **Q14格式**：总位数：1 位符号 + 1 位整数 + 14 位小数，表示范围 $[-2, 2-2^{-14})$
- **+0.9**：`0.9 * 16384 = 14745.6`，取整得`14746`（`0x399A`）。
- **-1.9**：`1.9 * 16384 = 31129.6`，取整得`31130`（`0111 1001 1001 1010`），补码为`1000 0110 0110 0110`（`0x8666`）。

> 在DSP定点运算中，小数的补码表示本质上是将其**缩放至整数范围后的整数的补码。**

6. 设累加器为 16 位，以二进制原码的形式，计算十进制整数 5 和 10 的乘积，计算过程以二进制形式表示。

1. **转换为原码**：
   - `5` 的原码：`0000 0000 0000 0101`
   - `10` 的原码：`0000 0000 0000 1010`
2. **执行无符号二进制乘法**（原码乘法忽略符号位，对数值位相乘）：

```
        0101  (5)
      x 1010  (10)
      -------
        0000
       0101
      0000
  +  0101
  ----------------
    0110010  (50)
```

3. **最终答案**：`0000 0000 0011 0010`（即十进制50）

7. 设累加器为 16 位，以二进制原码的形式，计算十进制小数 1.5 和 2.5 的乘积，计算过程以二进制形式表示。

1. **二进制表示**：`1.5`二进制为`1.1`，`2.5`二进制为`10.1`
2. **二进制乘法**：

```
      1.1    (1.5)
×    10.1    (2.5)
------------
      1 1
    0 0
+ 1 1 0
------------
  1 1.1 1
```

3. **Q12定点格式**：`3.75*4096=15360`，因此二进制小数`11.11`对应的Q12定点格式为`0011 1100 0000 0000`

8. 设字长为 16 位，小数部分为 6 位，计算浮点数 0.7 转换为定点数之后的结果。

Q6格式：定点数为 `0.7 × 64 = 44.8`，取整得`45`，因此定点数为`45`对应的二进制 **0000 0000 0010 1101**。

9. 设字长为 16 位，小数部分为 10 位，计算有符号定点数 1100 转换为浮点数之后的结果。

1. **符号扩展**：`1111 1111 1111 1100`（`0xFFFC`即十进制的`-4`）。
2. **Q10格式**：对应的浮点值$x = -4/1024 \approx -0.00390625$。

10. 设字长为 32 位，小数部分为 24 位，计算浮点数 9.9 转换为定点数之后的结果。

Q24格式：`9.9 × 16,777,216 = 166,094,438.4`，取整`166,094,438`（`0x9E6B266`）。

11. 设字长为 32 位，小数部分为 28 位，计算有符号定点数 1 1100 0000 转换为浮点数之后的结果。

符号扩展：`1111 1111 1111 1111 1111 1111 1100 0000`（`0xFFFF FFC0`，对应十进制`-64`的补码），浮点数值`(-64) / 268,435,456 = -0.000000238418579101562 5`，约 -2.38×10^{-7}。

12. 把十进制小数 1.1 转换为二进制数。

1. 整数部分`1`的二进制为`1`。
2. 小数部分`0.1`转换为二进制（乘2取整）：

```
0.1 × 2 = 0.2 → 整数部分 0
----
0.2 × 2 = 0.4 → 整数部分 0
0.4 × 2 = 0.8 → 整数部分 0
0.8 × 2 = 1.6 → 整数部分 1
0.6 × 2 = 1.2 → 整数部分 1
----
0.2 × 2 = 0.4 → 整数部分 0
0.4 × 2 = 0.8 → 整数部分 0
0.8 × 2 = 1.6 → 整数部分 1
0.6 × 2 = 1.2 → 整数部分 1
(循环节为0011)
...
```

3. 所以`0.1`的二进制为`0.0001100110011...`（循环节为`0011`）。
   则`1.1`（十进制）= `1.0001100110011...`（二进制循环小数）。

13. 简要描述 IEEE-754 单精度浮点格式的基本特点。

- **总位数**：32位。
- **位分配**：
  - 符号位（S）：1位（第31位），0正1负。
  - 指数位（E）：8位（第30-23位），用偏移码表示，偏移量为127。
  - 尾数位（M）：23位（第22-0位），用隐藏位规格化表示，实际尾数为`1.M`（规格化数时）。
- **表示范围**：约 ±1.4×10⁻⁴⁵ 到 ±3.4×10³⁸。

- **特殊值**：
  - 指数全0且尾数全0：表示零（±0）。
  - 指数全0但尾数非0：表示纯小数。
  - 指数全1且尾数全0：表示无穷大（±∞）。
  - 指数全1且尾数非0：表示NaN（非数字）。
- **精度**：有效数字约6-7位十进制数。

14. 总结 F28335 支持的定点整数类型，并描述它们的区别。

类型	位数	表示范围（有符号）	表示范围（无符号）
char	16位	-32,768 ~ 32,767	0 ~ 65,535
short	16位	-32,768 ~ 32,767	0 ~ 65,535
int	16位	-32,768 ~ 32,767	0 ~ 65,535
long	32位	-2,147,483,648 ~ 2,147,483,648	0 ~ 4,294,967,295
long long	64位	-9 223 372 036 854 775 808~ 9 223 372 036 854 775 807	0 ~ 18 446 744 073 709 551 615

核心区别：主要在于**数据宽度、表示范围和存储空间占用。**

15. 总结 F28335 支持的浮点数类型，并描述它们的区别。

类型	位数	格式	大致范围（绝对值）	有效数字（十进制）
float	32位	IEEE-754 32bit	约 1.4e-45 ~ 3.4e38	6-7位
double	64位	IEEE-754 64bit	约 4.9e-324 ~ 1.8e308	15-16位
long double	64位	IEEE-754 64bit	约 4.9e-324 ~ 1.8e308	15-16位

核心区别：主要在于**数据宽度、表示范围和存储空间占用。**

四、F28335系统时钟与中断控制

1. F28335片上的PLL有哪些工作模式?

主要有三种工作模式：

- **PLL关闭**：将PLLSTS寄存器中的PLLOFF位置1可关闭，减少系统噪声并减少功率损耗。进入前需将PLLCR设为0x0000.
- **PLL旁路**：上电复位或XRS'复位后，PLL进入。时钟信号绕过PLL模块但未关闭。
- **PLL使能**：向PLLCR寄存器写入非零的数使能，写入数据后进入旁路模式至系统稳定。

2. F28335有哪些单独的外设时钟信号?

F28335的时钟系统除了为CPU提供SYSCLKOUT外，还为不同的外设模块提供了独立的时钟信号，以实现灵活的功耗管理和速度匹配。主要的外设时钟域包括：

- **LOSPCP（低速外设时钟预分频器）**：产生LSPCLK，主要为SCI（串口）、SPI、McBSP等低速通信外设提供时钟。
- **HISPCP（高速外设时钟预分频器）**：产生HSPCLK，为ePWM、eCAP、eQEP、HRPWM、ADC等对时序精度要求高的高速控制外设提供时钟。
- **ADC时钟**：具有独立的分频器，可以从HSPCLK或SYSCLKOUT分频得到，专供ADC模块使用，以满足其特定的转换时序要求。
- **外设模块自身时钟使能**：每个外设（如ePWM、SCI等）都拥有独立的时钟使能位（通常在PCLKCRx寄存器中）。即使该外设的时钟域（如HSPCLK）已开启，也必须使能此外设的时钟，它才能工作。这是实现外设级低功耗的关键。

3. 描述XCLKOUT与SYSCLKOUT的关系。

- **SYSCLKOUT**：是芯片内部CPU及大部分内核逻辑的时钟信号，是系统运行的核心主频。所有外设时钟（LSPCLK, HSPCLK）都源于它。
- **XCLKOUT**：是一个可以从芯片引脚输出的时钟信号，主要用于外部电路或芯片的同步或调试。
- **关系**：通过对SYSCLKOUT进行分频（分频系数可配置，如/1, /2, /4等），得到XCLKOUT，可以关闭XCLKOUT的输出以节省功耗和减少噪声。

4. F28335片上的PLL有哪些低功耗工作模式?

PLL本身的低功耗工作模式与其工作模式紧密相关，主要通过以下方式实现：

1. **PLL旁路模式**：当PLLCR寄存器被设置为0（即DIV=0）时，PLL**电路被关闭**，系统运行在旁路模式。由于PLL本身是一个高功耗的模拟电路，关闭它可以**显著降低芯片功耗。**此时系统频率较低（OSCCLK/4）。
2. **IDLE/STANDBY/HALT模式下的时钟控制**：当CPU通过执行`IDLE`指令进入**低功耗模式**时，系统时钟（SYSCLKOUT）**可以停止**，但外设时钟可能根据配置保持运行。在这种全局低功耗状态下，PLL的功耗也相应降低。从这些模式唤醒时，PLL可能需要时间重新锁定。
    - **STANDBY模式**：需要外部中断唤醒，唤醒后系统时钟恢复。
    - **HALT模式**：功耗最低，几乎所有内部时钟都停止。
3. **动态改变PLL倍频系数**：在系统运行时，可以通过软件**动态降低PLL的倍频系数**，从而降低SYSCLKOUT频率，实现动态降频以节省功耗。

5. 总结看门狗模块的作用。

看门狗定时器模块的核心作用是**提高系统的可靠性，防止软件跑飞或陷入死循环。**

- **工作原理**：它是一个**递减计数器。**如果软件正常运行，必须在计数器**减到0之前**（即超时之前）周期性地向看门狗复位键（WDKEY）写入正确的序列（`0x55 + 0xAA`）来“喂狗”，使计数器复位重载，避免其超时。
- **触发后果**：如果软件因故障未能及时喂狗，计数器超时后，看门狗会产生一个**系统复位信号**（或一个中断，可配置），强制整个芯片**复位重启**，使系统从已知的初始状态重新开始运行，从而从故障中恢复。
- **关键配置**：其超时时间可通过预分频器和周期寄存器进行配置，以适应不同应用的监控需求。

6. CPU定时器有几个？它们的区别是什么？

F28335内部有3个32位的CPU定时器，分别为Timer0、Timer1、Timer2。它们的区别为：

属性	归属	复位状态	中断向量	典型用途
Timer 2	保留给 TI 实时操作系统（如 DSP/BIOS、TI-RTOS）使用	默认禁用（需软件使能）	INT13（TINT0）	RTOS 系统节拍（tick）
Timer 0	用户可用	默认禁用	XINT13（通过 PIE）	用户自定义定时任务
Timer 1	用户可用	默认禁用	XINT14（通过 PIE）	用户自定义定时任务

共同点：三者结构相同，都是32位递减计数器，具有可配置的预分频器和周期寄存器。

7. 描述EALLOW保护的作用。

EALLOW（写使能）保护是一种**硬件安全机制**，用于保护芯片中**关键的系统控制寄存器**免受意外或恶意的软件写入操作。

- **保护对象**：主要是那些配置时钟、PIE向量表、仿真、代码安全模块（CSM）等影响芯片全局行为或安全性的寄存器。
- **操作流程**：
    1. **使能写操作**：在需要修改受保护的寄存器之前，必须执行一条特殊的汇编指令 `EALLOW`。
    2. **进行修改**：此时，软件可以正常写入这些寄存器。
    3. **禁止写操作**：修改完成后，必须立即执行 `EDIS` 指令，重新锁住这些寄存器，防止后续代码误写。
- **作用**：防止程序跑飞或代码漏洞对关键系统配置进行篡改，从而**避免系统崩溃、外设行为异常或安全漏洞**，增强了系统的健壮性。

8. 有哪些寄存器是被EALLOW机制所保护的？

被EALLOW保护的寄存器主要分布在以下几个模块中：

1. **器件仿真寄存器**：控制TAG调试和仿真功能的寄存器。
2. **Flash/OTP配置寄存器**：控制Flash擦写、等待状态、功耗模式的寄存器。
3. **代码安全模块（CSM）寄存器**：包括密码寄存器和相关控制寄存器。
4. **系统控制寄存器**：
    - **PLL控制、时钟使能、低功耗模式控制**相关寄存器。
    - **GPIO多路复用控制（GPxMUX, GPxDIR等）** 和上拉控制寄存器。
    - **XINTF（外部接口）的时序配置寄存器。**
5. **PIE中断向量表**：存放所有中断服务程序入口地址的PIE向量表（PIEVECTTABLE）。

9. 描述PIE模块的作用。

PIE（外设中断扩展）模块是F28335中断管理系统的核心枢纽，其主要作用如下：

- **中断源扩展**：CPU本身只有少数几个（如16个）核心中断线（INT1-INT14, NMI, etc.）。而F28335拥有海量的外设，每个都可能产生中断。PIE模块将众多外设中断（最多96个）复用映射到这有限的CPU中断线上。
- **中断向量表管理**：PIE模块内部维护着一个256 x 16位的RAM向量表。每个外设中断（如EPWM1_INT, SCIA_RX_INT）在此表中都有唯一的中断服务程序入口地址。这比传统固定ROM向量表灵活得多，允许用户在运行时动态修改中断服务程序。
- **中断仲裁与优先级**：对于映射到同一根CPU中断线上的多个外设中断，PIE模块内的硬件会根据它们在组内的固定顺序（如INTx.1到INTx.8）提供次级优先级。结合CPU中断线本身的优先级，形成了两级优先级体系。
- **中断标志管理**：负责接收、锁存和清除来自各个外设的中断请求标志。

10. PIE模块共有几个中断源?

F28335的PIE模块最多支持96个可屏蔽的外设中断源，被分为12组（INT1 - INT12），每组包含8个中断源（如INT1.1, INT1.2, ..., INT1.8）。

11. 任选一个外设中断信号，用自己的语言描述CPU是如何响应外设中断的。

以 ePWM1的周期中断（EPWM1_INT，假设它被分配为PIE组1的第1个中断，即INT1.1）为例，描述CPU的响应过程：

1. **中断产生**：ePWM1计时器计数到周期值，硬件自动将ePWM1的中断标志位（EPWM1_INT_FLG）置1。
2. **PIE接收**：ePWM1_INT信号线连接到PIE模块的INT1.1输入。PIE模块检测到该信号后，将组1的组中断标志位（PIEIFR1.1）置1。
3. **向CPU申请**：如果此时组1的**组中断使能位（PIEIER1.1）也为1**，且该组的总使能有效，PIE模块就会向CPU的INT1中断线发出一个中断请求。
4. **CPU响应**：如果CPU全局中断使能（INTM位为0），且INT1中断未被屏蔽，CPU在完成当前指令后，保存现场（将关键寄存器压栈），然后跳转到INT1的中断服务程序入口。
5. **PIE向量调度**：在INT1的公共中断服务程序中，软件需要查询PIE模块的组应答寄存器（PIEACK）。PIE硬件会自动将优先级最高的、已置位且已使能的组内中断源所对应的向量地址提供给CPU。
6. **执行具体ISR**：CPU根据PIE提供的地址，跳转到专门处理 `EPWM1_INT` 的具体中断服务程序（ISR）中执行用户代码（例如，更新PWM占空比）。
7. **清除中断标志**：在ISR结束前，软件必须手动清除三个标志以允许下次中断：
   - 外设级：清除ePWM1模块本身的 `EPWM1_INT_FLG`。
   - PIE级：清除PIE的 `PIEIFR1.1` 位。
   - 向PIE发送应答：向 `PIEACK.1` 位写1，告知PIE组1的中断已处理完毕。
8. **中断返回**：ISR执行完毕后，通过 `IRET` 指令恢复现场，CPU返回到被中断的主程序继续执行。

五、通用输入/输出端口

1. F28335的88个GPIO引脚被分为几组端口?

在寄存器层面，这88个引脚被划分为 A、B、C三组端口进行管理：

- **GPIOA组**：包含 **GPIO0 ~ GPIO31**，共32个引脚。
- **GPIOB组**：包含 **GPIO32 ~ GPIO63**，共32个引脚。
- **GPIOC组**：包含 **GPIO64 ~ GPIO87**，共24个引脚。

2. 每组GPIO有哪些寄存器?

每组GPIO都拥有相同类型的一组寄存器，用于配置和控制该组内的每一个引脚：

1. **功能选择寄存器GPxMUX1/GPxMUX2**：决定引脚是作为通用数字I/O（置0）还是作为外设功能（置1，如PWM, SCI等）使用。
2. **方向控制寄存器GPxDIR**：当引脚配置为数字I/O时，此寄存器决定引脚方向。1 = 输出，0 = 输入。
3. **数据寄存器**
   - **GPxDAT**：直接读写引脚的电平状态。作为输入时，读取外部电平；作为输出时，写入的值驱动引脚输出。
   - **GPxSET**：**置位寄存器。**向某位写1，对应的`GPxDAT`位被置1（输出高电平），写0无效。
   - **GPxCLEAR**：**清零寄存器。**向某位写1，对应的`GPxDAT`位被清零（输出低电平），写0无效。
   - **GPxTOGGLE**：**翻转寄存器。**向某位写1，对应的`GPxDAT`位状态取反，写0无效。
4. **上拉控制寄存器GPxPUD（只有AB有）**：控制内部上拉电阻。1 = 禁止上拉，0 = 使能上拉（引脚作为输入时有效）。
5. **控制寄存器GPxTRL和输入限定寄存器GPxSEL1/2（只有AB有）**：对输入信号进行限定，TRL开启输入限定，SEL选择限定的类型，以消除外部噪声信号。

3. 描述GPIO输入限定功能的作用。

GPIO输入限定功能（Input Qualification）主要用于提高输入信号的抗干扰能力，消除引脚输入的噪声信号。

- **配置选项**：通过`GPxQSEL`寄存器为每个引脚选择量化模式：
  - **异步输入**：不需要同步信号或自身具有同步信号
  - **仅与SYSCLKOUT同步**：与时钟同步，由于输入异步，会产生一个SYSCLKOUT周期的延时
  - **通过采样窗限定**：经过对输入信号进行限定的采样窗，只有输入信号在采样窗内保持不变，采样窗后信号才允许改变，得到最终信号。

4. 简要描述GPIO的配置步骤。

1. **选择 GPIO 工作模式**：首先搞清每个 GPIO引脚所具有的功能，并通过配置 GPxMUXn寄存器选择其工作在外设I/O模式或数字I/O模式。默认情况下，GPIO被配置成数字I/O模式，且为输入状态。
2. **使能或禁止内部上拉电阻**：通过对相应的内部上拉控制寄存器GPxPUD进行配置，可使能或禁止内部上拉功能。
3. **选择输入/输出方向**：如果一个 GPIO被配置成数字I/O模式，还需要为其配置输入/输出方向，通过写GPxDIR寄存器，完成输入/输出方向的配置。
4. **选择输入限定模式**：当一个数字I/O被配置成输入状态，可以为其选择限定模式。默认情况下，所有的输入信号与系统时钟 SYSCLKOUT同步。

5. **选择低功耗模式的唤醒端口**：通过配置 GPIOLPMSEL寄存器，可以指定一个GPIO引脚，用其将CPU从 HALT和STANDBY低功耗模式中唤醒。
6. **为外部中断源选择输入引脚**：为XINT1~XINT7及XNMI外部中断选择合适的输入引脚。

5. 如果希望配置GPIO为SCI-A的发送引脚，可以配置哪些GPIO?

SCI-A（串行通信接口A）的发送引脚 `SCITXDA` 可以通过GPIO复用功能映射到多个不同的物理引脚上，主要可以映射到以下GPIO引脚：

1. **GPIO28**：这是 `SCITXDA` 最常用、最标准的映射引脚之一。
2. **GPIO35**：这是 `SCITXDA` 的一个备用功能映射引脚。

六、模/数转换模块

1. 设 A/D 输入引脚上的电压分别为 0V、1V、1.5V、2.5V、3V，分别计算对应的转换结果寄存器的值。

TMS320F28335内置一个12位分辨率的ADC模块。其基准电压通常为内部或外部提供的3.0V。

- **转换公式**：

  $$转换结果 = \frac{输入电压 - V_{REFLO}}{V_{REFHI} - V_{REFLO}} \times 2^{12} = \frac{输入电压}{3.0} \times 4095$$，结果取整数部分。

- **计算过程与结果**：

输入电压 (V)	计算过程 (十进制)	ADC结果寄存器值 (十进制)	ADC结果寄存器值 (十六进制，12位有效)
0.0	(0.0 / 3.0) × 4095 = 0	**0**	0x000
1.0	(1.0 / 3.0) × 4095 ≈ 1365	**1365**	0x555
1.5	(1.5 / 3.0) × 4095 = 2047.5	**2048**	0x800
2.5	(2.5 / 3.0) × 4095 ≈ 3412.5	**3413**	0xD55
3.0	(3.0 / 3.0) × 4095 = 4095	**4095**	0xFFF

2. 假设 A/D 输入引脚的电压可能会超过 0～3V 的范围，请设计一个输入电压限制电路，以保证输入电压被限制在该范围内。

为了保护ADC输入引脚不被过压损坏，需设计一个钳位保护电路。一个经典且有效的设计是使用二极管钳位结合串联限流电阻。

- **电路原理图**：

```
V_in ──┬──────R_s──────┬──────── 至 ADC 引脚 (V_adc)
       |               |
     C_bypass        V_clamp
       |               |
      GND             GND
```

（其中 `V_clamp` 部分是由两个二极管组成的钳位网络）

- **元件选择与作用**：
  1. **串联限流电阻 (R_s)**：阻值通常为100Ω ~ 1kΩ。主要作用：
     - 限制当输入电压过高时流入钳位二极管和ADC引脚的最大电流。
     - 与ADC内部的采样电容构成低通滤波器，有助于滤除高频噪声。
  2. **钳位二极管**：
     - **D1**：阴极接ADC引脚，阳极接GND。当ADC引脚电压低于-0.3V~-0.7V（二极管正向压降Vf）时导通，将电压钳位在 `-Vf`。
     - **D2**：阳极接ADC引脚，阴极接3.3V（或ADC基准电压VREFHI）。当ADC引脚电压超过 `3.3V + Vf` 时导通，将电压钳位在 `3.3V + Vf`。
     - *选择快速开关肖特基二极管（如BAT54S）因其Vf小、响应快。也可选用集成钳位器件如TVS二极管。*
  3. **旁路电容 (C_bypass)**：在ADC引脚与GND之间接一个 `10pF ~ 100pF` 的小电容，可进一步滤除高频干扰，并配合R_s稳定采样时刻的电压。
- **工作过程**：当 `V_in` 在0V~3V安全范围内时，二极管截止，`V_adc ≈ V_in`。当 `V_in > 3.3V+Vf`，D2导通；当 `V_in < -Vf`，D1导通，从而确保 `V_adc` 始终被限制在安全区间内。

3. 假设转换结果寄存器的值为 2000，计算对应的 A/D 输入引脚上的电压值。

已知ADC结果为2000（十进制），基准电压$V_{REFHI} = 3.0V$，$V_{REFLO} = 0V$。

计算公式（转换公式的逆运算）：

$$\text{输入电压} = \frac{ADC\text{结果}}{2^{12}} \times (V_{REFHI} - V_{REFLO}) = \frac{2000}{4095} \times 3.0\,V \approx 0.4884 \times 3.0\,V \approx 1.465\,V$$

4. 简要描述级联发生器和双序列发生器的区别。

F28335的ADC模块有两种多通道采样排序模式，由级联发生器和双序列发生器控制，核心区别在于排序器的灵活性和通道容量。

特性	级联发生器模式	双序列发生器模式
工作方式	将两个8状态排序器（SEQ1和SEQ2）合并成一个16状态的超大排序器（SEQ）。	两个排序器独立工作：SEQ1（8状态）和SEQ2（8状态）。
触发源	使用一套触发源（如ePWM、软件等）启动整个16通道的转换序列。	SEQ1和SEQ2有各自独立的触发源，可被不同事件异步触发。
通道容量	最多可连续自动转换16个通道（任意顺序）。	每个排序器最多连续转换8个通道，总共也是16通道，但分为两组。

特性	级联发生器模式	双序列发生器模式
灵活性	灵活性较低，适合需要一次性采样大量通道（>8）且触发源单一的场合。	灵活性**高**，适合需要分组采样、由不同事件触发的复杂应用（如同时采样电机三相电流和电压）。
结果存放	转换结果按顺序存入`ADCRESULT0~15`。	SEQ1的结果存入`ADCRESULT0~7`，SEQ2的结果存入`ADCRESULT8~15`。
典型应用	需要同步采样超过8个模拟信号的系统。	电机控制（多路电流、电压分时/同步采样）、多任务数据采集系统。

> **核心区别总结**：**级联模式是“单队列、大容量”，双序列模式是“双队列、可独立触发”。**

5. ADC的控制寄存器有哪些?

F28335的ADC模块由一组控制寄存器管理，以下是关键寄存器及其核心作用：

寄存器类别	寄存器名称（缩写）	主要功能描述
控制寄存器	**ADCTRL1**	配置ADC基本模式：复位、挂起/恢复、采样模式、启动/停止等。
	ADCTRL2	配置中断使能、排序器模式（级联/双序列）、触发源选择（ePWM, GPIO, 软件等）。
	ADCTRL3	配置ADC时钟预分频器、内部基准带隙/参考电路上电控制。
最大转换通道设定寄存器	**ADCMAXCONV**	定义了一个转换序列内所完成的转换次数
通道选择排序寄存器	**ADCCHSELSEQ1~4**	共4个16位寄存器，定义最多16次转换（每个4位字段指定一个模拟输入通道）。
自动序列发生器状态管理器	**ADCASEQSR**	设置一次自动转换序列的最大转换次数（1~16）。
状态寄存器	**ADCST**	ADC状态寄存器，包含排序器状态、中断标志位等。
参考电压选择寄存器	**ADCREFSEL**	00选择内部参考电压，01选择引脚上2.048V，10选择引脚上1.500V，11选择引脚上1.024V
结果寄存器	**ADCRESULT0~15**	16个16位寄存器，存放12位ADC转换结果。
校正寄存器	**ADCOFFTRIM**	存储最低位开始计算的偏移量，以2的补码形式。

6. ADC模块中包含哪些中断源?

ADC模块的中断主要与排序器的工作完成事件相关，具体如下：

1. **SEQ1中断**：当排序器SEQ1完成一次完整的转换序列（转换次数达到MAX_CONV1设定值）时产生。

2. **SEQ2中断**：当排序器SEQ2完成一次完整的转换序列（转换次数达到MAX_CONV2设定值）时产生。在级联模式下此中断无效。
3. **EOC（转换结束）中断**：在每个通道转换完成后都可以产生一个中断。此模式会频繁产生中断，通常用于需要实时处理每个数据点的场景。
4. **溢出中断**：当ADC结果寄存器（ADCRESULTx）中的数据尚未被CPU读取就被新的转换结果覆盖时产生，用于提示数据丢失。

七、增强型脉宽调制模块

1. ePWM有哪些子模块?

F28335的增强型脉宽调制（ePWM）模块由以下7个功能独立又相互协作的子模块构成，它们通过时基同步信号串联起来：

1. **时间基准（TB）子模块**：产生核心的时基计数器（`TBCTR`）和同步信号，是整个ePWM模块的"心脏"。
2. **比较功能（CC）子模块**：将`TBCTR`值与两个比较寄存器（`CMPA`，`CMPB`）比较，产生相应的事件。
3. **动作限定（AQ）子模块**：根据TB和CC产生的事件，决定如何驱动ePWM的输出引脚（置高、置低、翻转）。
4. **死区产生（DB）子模块**：对AQ输出的两路PWM信号（EPWMxA, EPWMxB）插入可编程的死区时间，防止半桥/全桥电路中的上下管直通。
5. **斩波控制（PC）子模块**：产生高频载波信号，用于驱动栅极驱动变压器或生成脉冲序列。
6. **事件触发（ET）子模块**：管理中断和ADC启动（SOC）触发事件。
7. **故障捕获（TZ）子模块**：响应外部故障引脚（如过流、过压）信号，可快速强制PWM输出为安全状态。

2. 描述ePWM的基准时钟与系统时钟之间的关系，以及如何设置相应的寄存器。

- **关系**：ePWM模块的基准时钟（TBCLK）来源于系统时钟（SYSCLKOUT）。它是SYSCLKOUT经过一个可配置的预分频器（由`TBCTL[CLKDIV]`和`TBCTL[HSPCLKDIV]`控制）分频后得到的。

  > **注意**：在F28335中，ePWM模块属于高速外设，其时钟源为**HSPCLK**，而HSPCLK又由SYSCLKOUT通过高速外设时钟预分频器（`HISPCP`寄存器）分频得到。简化理解中，常将HSPCLK视为TBCLK的输入源。

- **寄存器设置**：
  1. **设置高速外设时钟分频（HISPCP）**：在系统控制寄存器中配置，将HSPCLK设置为所需频率（如与SYSCLKOUT相等或分频）。
  2. **设置时基时钟预分频（TBCTL寄存器）**：
     - `TBCTL[CLKDIV]`：设置时基时钟预分频值（/1, /2, /4, ..., /128）。
     - `TBCTL[HSPCLKDIV]`：设置高速时钟预分频（通常设置为/1）。

- **周期计算公式**：

$$TBCLK = \frac{SYSCLKOUT}{HISPCP \times CLKDIV}$$

3.ePWM的计数器有哪些工作模式？它们各自的特点是什么？

ePWM时基计数器（`TBCTR`）有4种工作模式，由`TBCTL[CTRMODE]`控制：

1. **递增计数模式（Up-Count）**：
   - **特点**：计数器从0递增到周期值（`TBPRD`），然后复位回0，重复。产生非对称PWM波形（即占空比从0%到100%变化时，脉冲中心位置会移动）。
   - **适用**：普通PWM生成，如简单的功率调节。
   - **周期公式**：$T_{PWM} = (TBPRD + 1) \times T_{TBCLK}$
2. **递减计数模式（Down-Count）**：
   - **特点**：计数器从周期值（`TBPRD`）递减到0，然后重载`TBPRD`，重复。也产生非对称PWM。
   - **适用**：与递增模式类似，但计数方向相反。
   - **周期公式**：$T_{PWM} = (TBPRD + 1) \times T_{TBCLK}$
3. **递增-递减计数模式（Up-Down Count）**：
   - **特点**：计数器从0递增到`TBPRD`，然后递减回0，重复。产生对称PWM波形（即占空比变化时，脉冲中心位置固定）。这是最常用的模式。
   - **适用**：电机控制、全桥变换器等需要对称PWM的场合，可减少谐波。
   - **周期公式**：$T_{PWM} = 2 \times TBPRD \times T_{TBCLK}$
4. **停止/冻结模式（Stop/Freeze）**：
   - **特点**：计数器停止计数，保持当前值。用于调试或紧急停止。

4. ePWM当前寄存器与映射寄存器的联系和区别。

- **当前寄存器（Active Register）**：用来控制系统硬件的运行，反映硬件当前状态。
- **映射寄存器（Shadow Register）**：用来暂存数据，在特定时刻将数据传送到当前寄存器中，对硬件没有任何直接作用。
- **联系与区别**：
  - **联系**：在特定的**时基事件**（如计数器为0、周期匹配）发生时，硬件会自动将映射寄存器的值加载到当前寄存器中，实现同步更新。
  - **区别**：映射寄存器用于安全地更新参数，避免写入当前寄存器可能导致的脉冲干扰。这是ePWM模块实现无毛刺、同步参数更新的关键机制。

5.比较功能子模块有哪些比较模式？它们各自的特点是什么？

比较功能子模块通过将计数器`TBCTR`与比较寄存器`CMPA`/`CMPB`比较来产生事件。其**模式**由`CMPCTL`寄存器控制：

加载模式（Load Mode）	触发条件	特点	适用场景
CTR = Zero	时间基准计数器归零时加载	更新发生在周期起点，适合不对称 PWM	通用电机控制、DC-DC 变换器
CTR = PRD	计数器等于周期值时加载	更新发生在周期终点（Up-count 模式下等效于 Zero）	较少单独使用

加载模式（Load Mode）	触发条件	特点	适用场景
CTR = Zero or PRD	在 Zero 或 PRD 时加载	兼容 Up/Down 模式，在对称 PWM 中常用	对称 PWM、正弦波调制
立即加载（无影子）	写入即生效	可能造成毛刺	调试或特殊需求

比较行为还受 时间基准（TB）计数模式 影响：

TB 计数模式	比较特点
Up-count	CMPA/CMPB 在 0→PRD 区间有效，适合不对称 PWM
Down-count	CMPA/CMPB 在 PRD→0 区间有效，较少用
Up-Down-count	每个周期比较事件发生两次（上升沿和下降沿各一次），适合对称 PWM 或中心对齐

6.动作限定子模块有哪些工作模式？它们各自的特点是什么？

动作限定（AQ）子模块根据事件（如`TBCTR`等于`CMPA`、`TBCTR`等于`CMPB`、`TBCTR`等于0、`TBCTR`等于`TBPRD`）来决定ePWM输出引脚的动作。其核心工作模式是针对每个事件可以独立配置的动作：

1. **无操作**：不改变输出。
2. **置高**：强制输出为高电平。
3. **置低**：强制输出为低电平。
4. **翻转**：输出电平取反。

7.死区产生子模块的作用是什么？

死区产生（DB）子模块的核心作用是在互补的PWM信号对（EPWMxA和EPWMxB）之间插入一段可控的延迟时间（死区时间）。

- **目的**：防止在桥式电路中，控制上下两个开关管的互补PWM信号由于器件开关延迟等原因出现同时导通的“直通”现象，从而避免短路烧毁功率管。
- **实现**：它可以对上升沿、下降沿或双边沿进行独立的延迟，生成最终带有死区的PWM输出信号（EPWMxA和EPWMxB）。

10.设系统时钟为150MHz,ePWM的基准时钟的频率为系统时钟频率的一半，如果要实现5kHz的不对称 PWM波形输出，则 ePWM的比较功能子模块的工作模式可以配置为哪一种？周期寄存器 TBPRD 的值应该设置为多少？给出计算过程。

- **已知**：$SYSCLKOUT = 150MHz, TBCLK = SYSCLKOUT/2 = 75MHz$。PWM频率 $f_{PWM} = 5kHz$，波形为非对称。
- **比较模式**：在此模式下，只需一个比较点即可设定占空比，通常使用`CMPA`，并配置为递增计数模式。
- **计算周期值（TBPRD）**：PWM周期 $T_{PWM} = (TBPRD + 1) \times T_{TBCLK}$，则 $TBPRD = T_{PWM} \times TBCLK - 1 = \dfrac{75 \times 10^6}{5 \times 10^3} - 1 = 14999$

11.设系统时钟为100MHz，ePWM 的基准时钟的频率与系统时钟频率相同，如果要实现10kHz的对称 PWM波形输出，则ePWM的比较功能子模块的工作模式可以配置为哪一种？周期寄存器 TBPRD 的值应该设置为多少？给出计算过程。

- **已知**：$SYSCLKOUT = 100MHz, TBCLK = SYSCLKOUT = 100MHz$。PWM频率 $f_{PWM} = 10kHz$，波形为对称。
- **比较模式**：采用递增-递减计数模式。
- **计算周期值（TBPRD）**：PWM周期 $T_{PWM} = (2 \times TBPRD) \times T_{TBCLK}$，则 $TBPRD = \dfrac{(TBCLK/f_{PWM})}{2} = \dfrac{(100 \times 10^6/10 \times 10^3)}{2} = 5000$

12.在ePWM的基准时钟为150MHz的情况下，为了实现双边的死区控制，分别给出1μs、5μs 和 10μs 的死区时间所对应的 DBFED 和 DBRED 寄存器的值。

死区时间 T_{dead}与寄存器值 DBRED (上升沿延迟) 和 DBFED (下降沿延迟) 的关系为：
$T_{dead} = DBxED \times T_{DBCLK}$

- **通用公式**：$DBxED = \dfrac{T_{dead}}{T_{TBCLK}} = T_{dead} \times 150 \times 10^6$

1. $T_{dead} = 1\mu s$：$DBxED = 1 \times 10^{-6} \times 150 \times 10^6 = 150$
2. $T_{dead} = 5\mu s$：$DBxED = 5 \times 10^{-6} \times 150 \times 10^6 = 750$
3. $T_{dead} = 10\mu s$：$DBxED = 10 \times 10^{-6} \times 150 \times 10^6 = 1500$

13.如果希望强制ePWM1A这个引脚的电压为高电平，应该如何配置相关的寄存器?

强制输出是通过**动作限定（AQ）子模块**的软件强制功能实现的。

将 `GPAMUX1` 的 `GPIO0` 位字段设置为 `1` 复用为 `EPWM1A`，将 `AQSFRC` 的 `ACTSFA` 字段值设置为 `0x2` (强制为高)，将 `AQSFRC` 的 `RLDCSF` 字段值设置为 `3` 立即生效。

14.如何配置相关的寄存器，使得ePWM1~ePWM6全部同步到同一个时钟信号下?

同步的目的是让所有ePWM模块的时基计数器从同一个起点开始计数。

1. **配置主模块Master**：将ePWM1配置为同步信号源。在ePWM1的 `TBCTL` 寄存器中，设置 `PHSEN=1`（使能相位加载），并根据需要设置 `TBCTL[SYNCOSEL]` 为输出同步信号（如 `CTR=0` 时）。
2. **配置从模块Slaves**：在ePWM2~ePWM6各自的 `TBCTL` 寄存器中， 设置 `PHSEN=1`（使能相位加载）， 设置 `TBCTL[SYNCOSEL]` 为选择外部同步输入（设置为接收来自前一个ePWM模块的同步信号 `SYNCI` ）， 将各自的 `TBPHS`（相位寄存器）设置为0，确保与主模块同步启动。

八、增强型正交编码脉冲模块

1. 设编码器有 1024 个槽，电机的转速为 3000r/min，计算编码器输出的脉冲频率。

1. 转速单位换算：$n = 3000r/min = 50r/s$
2. 每转脉冲数：对于增量式编码器，每转产生 N 个脉冲（A相或B相单独）。
3. 输出频率：$f = n \times N = 50r/s \times 1024脉冲/r = 51,200Hz$

   编码器输出的单相脉冲频率为 **51.2 kHz**。

2. 描述M法与T法测速各自的优缺点。

方法	测量原理	优点	缺点
M法	在固定时间 Tc 内，计数编码器脉冲数 M1。转速 $n \propto M_1$。	1. 中高速时测量精度高、分辨率好。 2. 测量时间固定，便于控制系统定时采样。 3. 硬件实现简单。	1. 低速时分辨率急剧下降（脉冲数少）。 2. 在极低速时，若 Tc 内无脉冲，则无法测量（速度为零）。
T法	测量编码器相邻脉冲的周期 T（通常用高频时钟脉冲数 M2 填充）。转速 $n \propto 1/M2$。	1. 低速时测量精度高、分辨率好。 2. 每个脉冲都能得到速度值，极低速也能测量。	1. 高速时分辨率下降（周期短，填充脉冲数少）。 2. 测量时间不固定（随速度变化），不利于定时控制。 3. 硬件电路或软件处理稍复杂。

> **总结：**M法适用于中高速，T法适用于中低速。实际系统中常采用M/T法（混合法），在高低速下都能保持较好的性能。

3. 设编码器有 512 个槽，计算 M 法测速的情况下，eQEP 的最小转速分辨率。

设eQEP的测速周期为 T_c，通常由eQEP的单位定时器（UTIME） 设置。

M法公式：$n = \frac{60 \times M_1}{N \times T_c}$，最小分辨率对应 $M_1 = 1$，则 $\Delta n_{min} = \frac{60}{N \times Tc} = \frac{60}{512 \times T_c}$

4. 位置计数器有哪些输入模式?

eQEP模块的位置计数器（QPOSCNT）的输入模式，决定了它如何对编码器的A、B相输入信号进行计数。主要模式由 `QDECCTL[QSRC]` 位配置：

1. **正交计数模式（Quadrature-count mode）**：
   - **模式**：根据A、B两相脉冲的相位关系（正交） 和先后顺序来判定方向并进行递增/递减计数。
   - **特点**：利用正交信号，可实现4倍频（在A、B相的每个上升沿和下降沿都计数），将分辨率提高4倍。同时自动识别方向。
2. **方向计数模式（Direction-count mode）**：
   - **模式**：将A相作为时钟脉冲，B相作为方向信号。B相电平高低决定对A相脉冲进行递增或递减计数。
   - **特点**：逻辑简单，分辨率等于编码器PPR。
3. **增计数模式（Up-count mode）**：
   - **模式**：只对A相脉冲进行递增计数，忽略B相。
   - **特点**：用于只需要单方向计数的场合。
4. **减计数模式（Down-count mode）**：
   - **模式**：只对A相脉冲进行递减计数，忽略B相。
   - **特点**：用于只需要单方向计数的场合。

5. 位置计数器有哪些运行模式？简要描述它们各自的特点。

位置计数器的运行模式决定了其计数行为的边界和归零方式，主要由 `QEPCTL[PCRM]` 位配置：

PCRM值	模式名称	特点
00	索引脉冲到来时发生复位	位置计数器在检测到编码器的索引脉冲（Z相）上升沿时，立即复位为0。适用于每转进行一次位置校准的场合，如电机零位对齐。
01	达到最大计数值时复位	当位置计数器的值达到预设的最大位置值（由 `QPOSMAX` 寄存器设定）时，自动复位为0。常用于单圈绝对角度测量（如0~360°），实现循环计数。
10	第一个索引到来时复位	仅在系统上电后**第一次**检测到索引脉冲时复位位置计数器，之后不再响应索引信号。用于初始化阶段确定机械零点，后续进入自由运行模式跟踪多圈位置。
11	单位时间输出事件时复位	在每个单位时间事件（通常由 eQEP 的 QCTMR 定时器产生，如1ms测速周期）发生时，位置计数器被复位。此模式较少使用，主要用于特殊测速算法中需要周期清零的场景。

6. 设电机的转子有 8 个极，位置计数器工作在模式 1，定时器基准单元的溢出周期为 5ms，单位位移量为 32 个正交脉冲，捕获单元定时器的时钟频率为 2.34375 MHz，计算位置计数器的最大值，以及 M 法和 T 法测速时各种的转速表达式。

- **已知条件**：极对数 $p = 8/2 = 4$，位置计数器达到最大计数值时复位，M法的采样时间$T = 5ms$，单位位移量$\Delta\theta = 32$个正交脉冲，捕获单元定时器时钟频率 $f_{cap} = 2.34375MHz$。
- **计算位置计数器的最大值**：由于极对数为4，因此每转正交脉冲数为$N = 32 \times 4 = 128$，位置计数器的最大值$QPOSMAX = N - 1 = 127$。
- **M法表达式**：M法在固定时间 $T = 5ms$ 内，对脉冲计数为 M，
  $n_M = \frac{60 \times M}{N \times T} = \frac{60M}{128 \times 0.005} = \frac{60M}{0.64} \approx 93.75M$(r / min)
- **T法表达式**：测量产生32个正交脉冲所需的时间，计数值为 M_t，
  $n_T = \frac{60 \times 32 \times f_{cap}}{N \times M_t} = \frac{60 f_{cap}}{4M_t} = \frac{35.15625 \times 10^6}{M_t}$(r / min)

十、串行通信接口模块

1. 简要描述SCI数据帧的格式。

SCI（串行通信接口）采用标准的异步串行通信协议，其数据帧格式如下（从起始位开始到停止位结束）：

1. **起始位**：1个逻辑低电平位，标志一帧数据的开始。
2. **数据位**：5~8个数据位（可配置，常用8位）。最低有效位（LSB） 先发送。
3. **奇偶校验位**：1个可选位（可配置为奇校验、偶校验或无校验）。用于简单的错误检测。
4. **停止位**：1个或2个逻辑高电平位（可配置），标志一帧数据的结束。
5. **额外地址位**：在地址模式下，1个额外地址位用来分辨地址与数据。

2.设SCI的通信传输速率分别为9600b/s、115200b/s，计算对应的比特流选择寄存器的值，LSPCLK的值可自选，并给出实际波特率和理想波特率之间的误差。

SCI的波特率由低速外设时钟 `LSPCLK` 和波特率选择寄存器 `SCIHBAUD` 和 `SCILBAUD`（16位组合为 `BAUD`）共同决定。计算公式为：

$SCI波特率 = \frac{LSPCLK}{(BRR+1)\times 8}$。其中，`BRR` 是 `BAUD` 寄存器的值（0~65535）。

由此可推导出 `BRR` 的计算公式：$BRR = \frac{LSPCLK}{SCI波特率\times 8} - 1$

由此可以算得$实际波特率 = \frac{LSPCLK}{(BRR+1)\times 8}$，以及$误差 = \frac{实际波特率 - 理想波特率}{理想波特率}\times 100\%$

假设 LSPCLK = 37.5 MHz（这是在SYSCLKOUT=150MHz且低速外设预分频器LOSPCP默认配置为/4时的典型值）。

- 对于 **9600 b/s**：

  $BRR_{9600} = \frac{37.5\times 10^6}{9600\times 8} - 1 = \frac{37.5\times 10^6}{76800} - 1 \approx 488.28125 - 1 = 487.28125$，取整得 $BRR = 487$ (0x01E7)。

  $实际波特率 = \frac{37.5\times 10^6}{(487+1)\times 8} = \frac{37.5\times 10^6}{3904} \approx 9605.53\,b/s$

  $误差 \approx \frac{9605.53 - 9600}{9600}\times 100\% \approx 0.0576\%$

- 对于 **115200 b/s**：

  $BRR_{115200} = \frac{37.5\times 10^6}{115200\times 8} - 1 = \frac{37.5\times 10^6}{921600} - 1 \approx 40.6901 - 1 = 39.6901$，取整得 $BRR = 40$ (0x0027)。

  $实际波特率 = \frac{37.5\times 10^6}{(40+1)\times 8} = \frac{37500000}{328} \approx 114329.27\,b/s$

  $误差 \approx \frac{114329.27 - 115200}{115200}\times 100\% \approx -0.756\%$

可以看出，对于9600和115200波特率的误差都在UART容忍范围内（2%），通信可靠。

3. 画出SCI自动波特率检测程序流程图。

```
graph TD
    A[开始自动波特率检测] --> B[配置SCI，使能接收];
    B --> C{等待接收特定同步字符};
    C -- 超时 --> Z[检测失败， 退出];
    C -- 收到字符 --> D[获取接收该字符期间的SCICLK周期计数];
    D --> E[根据计数值和字符长度计算波特率];
    E --> F[根据计算结果更新波特率寄存器BAUD];
    F --> G[检测成功， 退出];
```

4. SCI模块中包含哪些中断源?

SCI模块主要产生以下几种中断：

1. **接收中断RXRDY**：当 SCIRXBUF 接收到一帧完整的数据，并将其从RXSHF装载到SCIRXBUF中。
2. **发送中断TXRDY**：当 TX INT ENA置位时，发送器将SCITXBUF 中的数据装载到TXSHF中。

3. **传输间断中断BRKDT**：数据传输出现间断时（SCIRXD在一个停止位丢失后保持10位低电平时间）。

5.当数据位少于8位时，SCI的发生与接收缓冲区中是如何对数据进行处理的?

当SCI配置的数据位少于8位（如5、6、7位）时，数据在发送和接收缓冲区中的处理遵循 **“右对齐，高位补零”** 的原则：

- **发送**：用户将数据写入 SCITXBUF（一个8位寄存器）的低有效位。硬件发送时，只会取 `SCITXBUF` 中对应数据位数的低位进行发送，高位被忽略。
  - *示例*：配置为7位数据。若写入 `SCITXBUF = 0x8F`（二进制 `10001111`），则实际发送出去的数据位是低7位 `0001111`（0x0F）。
- **接收**：接收移位寄存器收到数据后，会将有效数据位存入 SCIRXBUF（一个8位寄存器）的低有效位，高位自动补0。
  - *示例*：配置为6位数据。若收到二进制数据 `111011`，则 `SCIRXBUF` 的值为 `00011101`（0x1D）。