



APB-TIMER

2024.09.21

一 修订

版本	日期	编辑人	内容
1.00	2024.09.21	陈家耀	创建了第一个正式版本

二 简介和特性

APB-TIMER 是一个带有 APB 从接口的通用定时器，可作为**时钟基准**，或者用于**脉冲检测**、**PWM 生成**。APB-TIMER 具有以下特性：

- 8~32 位定时器
- 支持多达 4 个输入捕获/输出比较通道
- 输入捕获支持边沿滤波，可捕获上升沿/下降沿/双沿
- 可启用计数溢出中断和输入捕获中断

APB-TIMER 由 **APB 寄存器接口**、**基本定时器**和**输入捕获/输出比较通道**组成，如图 2-1 所示。**APB 寄存器接口**实现了控制/状态寄存器片，并生成中断信号。**基本定时器**实现了带预分频和自动装载的向下计数，可作为时钟基准。**输入捕获/输出比较通道**可选择使用输入捕获或输出比较功能，输入捕获/输出比较共用 1 组捕获/比较寄存器，可用于脉冲检测（脉宽、频率测量）和 PWM 生成。

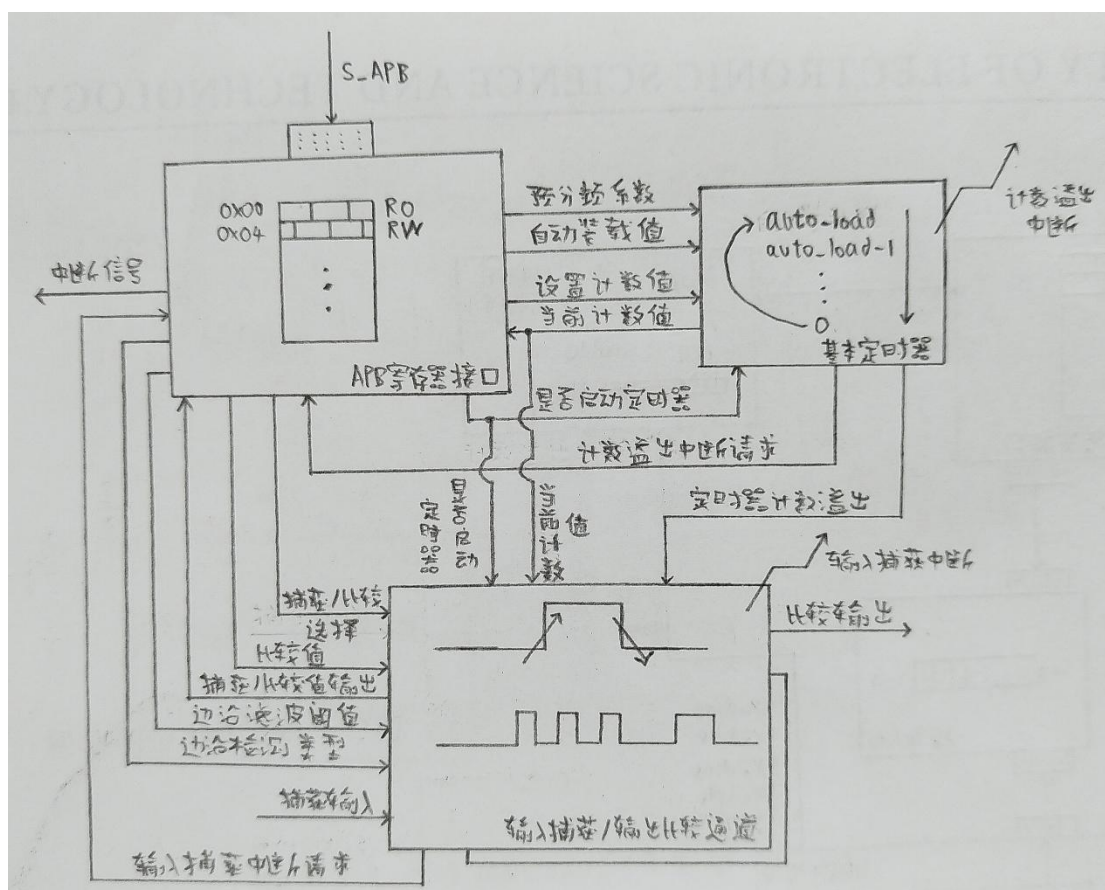


图 2-1 APB-TIMER 组成框图

三 IP 功能

APB-TIMER 包含基本定时器和至多 4 个输入捕获/输出比较通道，其结构如图 3-1 所示。APB-TIMER 精简易用，仅实现了最常用的定时器功能，能够满足 SOC 里对时钟基准、脉冲检测、PWM 生成等基本定时功能的需求。

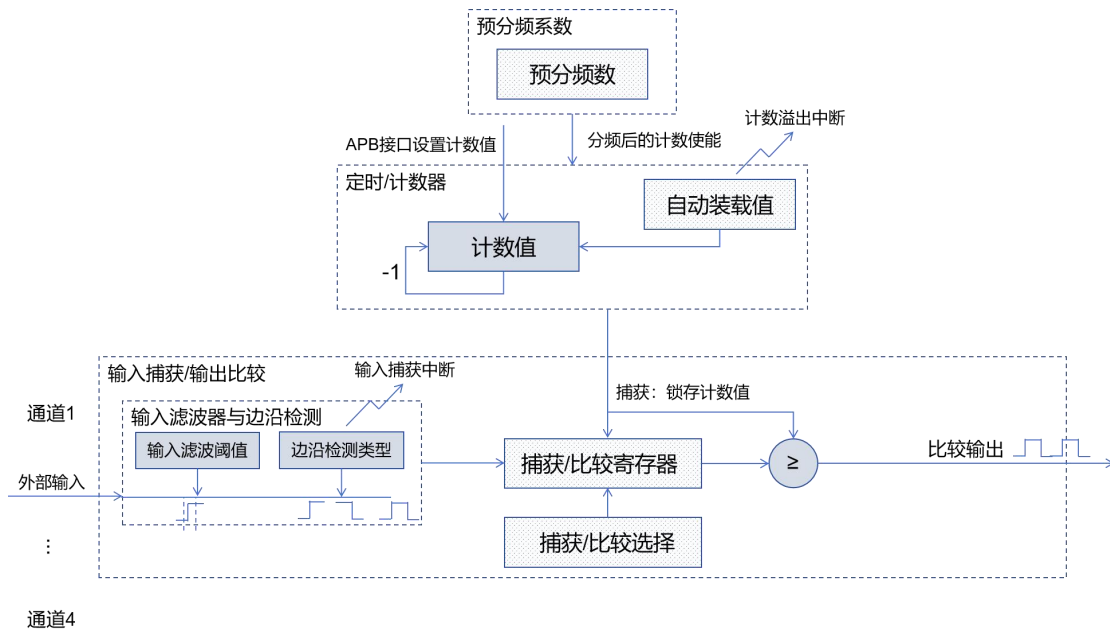


图 3-1 APB-TIMER 结构图

APB-TIMER 具有以下功能：

- 1、**产生时钟基准**。APB-TIMER 具有一组 8~32 位**计数值寄存器**、**预分频数寄存器**和**自动装载值寄存器**，预分频数寄存器带有影子寄存器缓存以保证预分频数只能在定时器溢出或者定时器未启动时改变。基本定时器固定为向下计数，计数到 0 时自动重新装载。可配置预分频数和自动装载值来改变定时器计数溢出的周期，以提供时钟基准。
- 2、**输入捕获**。APB-TIMER 的每个输入捕获/输出比较通道都可以独立地配置为输入捕获模式。APB-TIMER 可配置边沿检测类型（上升沿/下降沿/双沿）和边沿滤波阈值（捕获边沿时的确认周期数，可滤除毛刺）。当捕获到对应的边沿时，将当前的计数值锁存到捕获/比较寄存器。
- 3、**输出比较**。APB-TIMER 的每个输入捕获/输出比较通道都可以独立地配置为输出比较模式。APB-TIMER 将当前计数值与比较值进行比较来得到输出。可配置比较值以改变 PWM 输出的占空比，可配置预分频数和自动装载值来改变 PWM 输出的频率。
- 4、**可启用的计数溢出中断和输入捕获中断**。APB-TIMER 支持启用计数溢出中断以通知 CPU 计数溢出事件的发生，支持启用输入捕获中断以通知 CPU 输入捕获事件的发生，此时 CPU 可读取捕获/比较寄存器的值。

四 IO 描述

表 4-1 APB-TIMER IO 表

端口名	方向	位宽	含义
时钟和复位			
clk	input	1	时钟
resetn	input	1	复位，低有效
APB 从机接口			
paddr	input	32	读写地址
psel	input	1	从机选择，高有效
penable	input	1	传输使能
pwrite	input	1	传输方向，高为写
pwrdata	input	32	写数据
pready_out	output	1	传输完成信号
prdata_out	output	32	读数据
pslverr_out	output	1	传输错误信号，错误时为高
捕获/比较			
cap_in	input	channel_n	捕获输入
cmp_out	output	channel_n	比较输出
中断信号			
itr	output	1	通用定时器中断信号

注：对于不使用输入捕获功能的通道，应将其捕获输入 cap_in 固定为 1'b0。

五 可配置参数描述

5-1 APB-TIMER 可配置参数表

配置参数名	含义	可取值
timer_width	定时器位宽	8~32
channel_n	捕获/比较通道数	0~4
simulation_delay	仿真延时，可用于仿真时模拟 D 到 Q 延迟	0.1f~100.0f

六 应用指南

6.1 RTL 设计指南

APB-TIMER 是标准的通用定时器外设, 请将 APB-TIMER 挂载在 APB 总线上使用, 典型情况是挂载在 AXI-APB 桥或 AHB-APB 桥上作为一个 APB 从机, 如图 6-1-1 所示。关于 AXI-APB 桥或 AHB-APB 桥, 请参见 UG200。

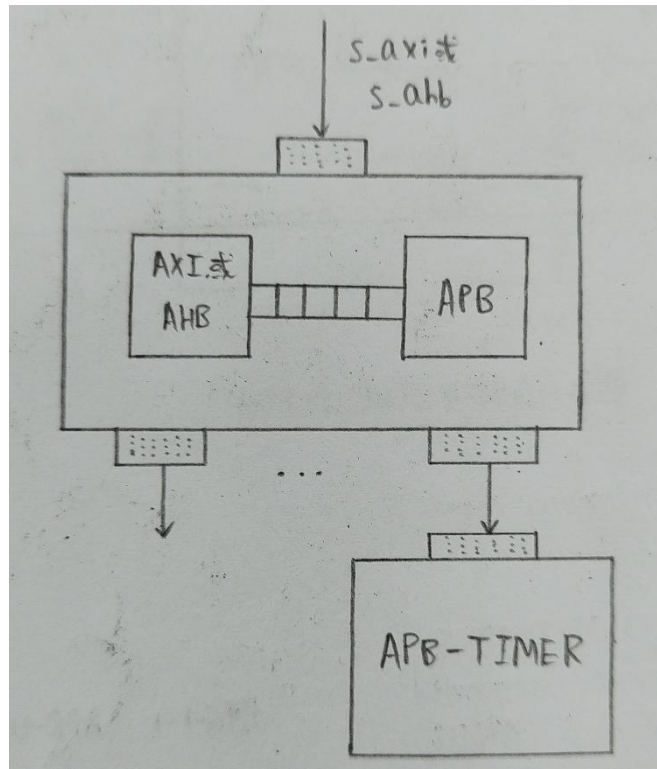


图 6-1-1 APB-TIMER 应用图

6.2 软件编程指南

6.2.1 软件驱动 API

1、类型定义

- ApbTimer 结构体 (APB-TIMER 外设结构体)

ApbTimerHd* hardware: APB-TIMER 寄存器接口 (结构体指针)

uint32_t prescale: 预分频系数 - 1 (备份值)

uint32_t auto_load: 自动装载值 - 1 (备份值)

uint8_t cap_cmp_sel: 捕获/比较选择 (备份值)

uint8_t chn_n: 通道数

- ApbTimerHd 结构体 (APB-TIMER 寄存器映射结构体)

uint32_t prescale: 预分频系数 - 1

uint32_t auto_load: 自动装载值 - 1

uint32_t count: 定时器计数值
uint32_t ctrl: 定时器控制（定时器开关, 捕获/比较选择）
uint32_t itr_en: 中断使能
uint32_t itr_flag: 中断标志
ApbTimerChnHd chn_hd[4]: 捕获/比较通道（寄存器区）

- **ApbTimerChnHd 结构体（APB-TIMER 捕获/比较通道寄存器映射结构体）**

uint32_t cap_cmp: 捕获/比较值
uint32_t cap_config: 捕获配置

- **ApbTimerConfig 结构体（APB-TIMER 初始化配置结构体）**

uint8_t cap_cmp_sel: 捕获/比较选择
uint8_t chn_n: 通道数
uint32_t prescale: 预分频系数 - 1
uint32_t auto_load: 自动装载值 - 1
uint32_t cmp[4]: 比较值
uint8_t edge_detect_type[4]: 边沿检测类型
uint8_t edge_filter_th[4]: 边沿滤波阈值

2、宏定义

- 通道编号

APB_TIMER_CH0 通道 0
APB_TIMER_CH1 通道 1
APB_TIMER_CH2 通道 2
APB_TIMER_CH3 通道 3

- 捕获/比较选择

CAP_SEL_CHN0 通道 0 设为捕获模式
CMP_SEL_CHN0 通道 0 设为比较模式
CAP_SEL_CHN1 通道 1 设为捕获模式
CMP_SEL_CHN1 通道 1 设为比较模式
CAP_SEL_CHN2 通道 2 设为捕获模式
CMP_SEL_CHN2 通道 2 设为比较模式
CAP_SEL_CHN3 通道 3 设为捕获模式
CMP_SEL_CHN3 通道 3 设为比较模式

注：使用时请“或上”每个通道的模式选择，如 **CAP_SEL_CHN0 | CMP_SEL_CHN1 | CAP_SEL_CHN2 | CMP_SEL_CHN3** 表示通道 0、2 选择捕获模式，通道 1、3 选择比较模式。

- 输入捕获边沿检测类型

CAP_POS_EDGE 捕获上升沿
CAP_NEG_EDGE 捕获下降沿
CAP_BOTH_EDGE 捕获双沿

- 中断类型掩码

ITR_TIMER_ELAPSED_MASK 计数溢出中断

ITR_INPUT_CAP_CHN0_MASK 通道 0 输入捕获中断

ITR_INPUT_CAP_CHN1_MASK 通道 1 输入捕获中断

ITR_INPUT_CAP_CHN2_MASK 通道 2 输入捕获中断

ITR_INPUT_CAP_CHN3_MASK 通道 3 输入捕获中断

注：用于中断使能时请“或上”需要使能的通道，用于从中断标志判断中断类型时请“与上”需要判断的中断。

3、宏函数

- **__apb_timer_start(timer)**

简介：APB-TIMER 启动定时器

参数：**timer** APB-TIMER（结构体指针）

返回值：无

- **__apb_timer_stop(timer)**

简介：APB-TIMER 暂停定时器

参数：**timer** APB-TIMER（结构体指针）

返回值：无

- **__apb_timer_reset(timer)**

简介：APB-TIMER 重置定时器

参数：**timer** APB-TIMER（结构体指针）

返回值：无

注意：仅将定时器计数值重置为自动装载值，未暂停定时器

- **__apb_timer_set_cnt(timer, cnt)**

简介：APB-TIMER 设置计数值

参数：**timer** APB-TIMER（结构体指针）

cnt 要设置的计数值

返回值：无

- **__apb_timer_get_prescale(timer)**

简介：APB-TIMER 获取预分频系数

参数：**timer** APB-TIMER（结构体指针）

返回值：预分频系数

- **__apb_timer_get_autoload(timer)**

简介：APB-TIMER 获取自动装载值

参数：**timer** APB-TIMER（结构体指针）

返回值：自动装载值

- **__apb_timer_get_cnt(timer)**

简介：APB-TIMER 获取计数值

参数: **timer** APB-TIMER (结构体指针)

返回值: 当前计数值

4、函数

- **void init_apb_timer**(ApbTimer* timer, uint32_t base_addr, const ApbTimerConfig* config);

简介: 初始化 APB-TIMER

参数: **timer** APB-TIMER (结构体指针)

base_addr APB-TIMER 外设基地址

config APB-TIMER 初始化配置 (结构体指针)

返回值: 无

- **void apb_timer_enable_itr**(ApbTimer* timer, uint8_t itr_en);

简介: APB-TIMER 使能中断

参数: **timer** APB-TIMER (结构体指针)

itr_en 中断使能向量

返回值: 无

- **void apb_timer_disable_itr**(ApbTimer* timer);

简介: APB-TIMER 除能中断

参数: **timer** APB-TIMER (结构体指针)

返回值: 无

- **uint8_t apb_timer_get_itr_status**(ApbTimer* timer);

简介: APB-TIMER 获取中断状态

参数: **timer** APB-TIMER (结构体指针)

返回值: 中断标志向量

- **void apb_timer_clear_itr_flag**(ApbTimer* timer);

简介: APB-TIMER 清除中断标志

参数: **timer** APB-TIMER (结构体指针)

返回值: 无

- **void apb_timer_start**(ApbTimer* timer);

简介: APB-TIMER 启动定时器

参数: **timer** APB-TIMER (结构体指针)

返回值: 无

- **void apb_timer_stop**(ApbTimer* timer);

简介: APB-TIMER 暂停定时器

参数: **timer** APB-TIMER (结构体指针)

返回值: 无

- **void apb_timer_reset**(ApbTimer* timer);

简介：APB-TIMER 重置定时器

参数：**timer** APB-TIMER（结构体指针）

返回值：无

注意：仅将定时器计数值重置为自动装载值，未暂停定时器

- **void apb_timer_set_prescale(ApbTimer* timer, uint32_t prescale);**

简介：APB-TIMER 设置预分频系数

参数：**timer** APB-TIMER（结构体指针）

prescale 预分频系数

返回值：无

- **void apb_timer_set_autoload(ApbTimer* timer, uint32_t auto_load);**

简介：APB-TIMER 设置自动装载值

参数：**timer** APB-TIMER（结构体指针）

auto_load 自动装载值

返回值：无

- **void apb_timer_set_cnt(ApbTimer* timer, uint32_t cnt);**

简介：APB-TIMER 设置计数值

参数：**timer** APB-TIMER（结构体指针）

cnt 计数值

返回值：无

- **int apb_timer_set_cmp(ApbTimer* timer, uint8_t chn_id, uint32_t cmp);**

简介：APB-TIMER 设置某个通道的比较值

参数：**timer** APB-TIMER（结构体指针）

chn_id 通道编号

cmp 比较值

返回值：是否成功

- **int apb_timer_set_cap_config(ApbTimer* timer, uint8_t chn_id, uint8_t edge_detect_type, uint8_t edge_filter_th);**

简介：APB-TIMER 设置某个通道的捕获配置

参数：**timer** APB-TIMER（结构体指针）

chn_id 通道编号

edge_detect_type 边沿检测类型

edge_filter_th 边沿滤波阈值

返回值：是否成功

- **uint32_t apb_timer_get_prescale(ApbTimer* timer);**

简介：APB-TIMER 获取预分频系数

参数：**timer** APB-TIMER（结构体指针）

返回值：预分频系数

- `uint32_t apb_timer_get_autoload(ApbTimer* timer);`
 简介：APB-TIMER 获取自动装载值
 参数：**timer** APB-TIMER（结构体指针）
 返回值：自动装载值
- `uint32_t apb_timer_get_cnt(ApbTimer* timer);`
 简介：APB-TIMER 获取计数值
 参数：**timer** APB-TIMER（结构体指针）
 返回值：当前计数值
- `uint32_t apb_timer_get_cap(ApbTimer* timer, uint8_t chn_id);`
 简介：APB-TIMER 获取某个通道的捕获值
 参数：**timer** APB-TIMER（结构体指针）
 chn_id 通道编号
 返回值：捕获值

6.2.2 软件编程示例

1、时钟基准示例

该示例位于“examples/apb_timer_timebase_example.c”。在本示例中，产生了 0.5s 的时基，实现 LED 灯每隔 0.5s 闪烁。这里假设 APB-TIMER 的时钟为 25MHz，定时器位宽为 16 位。另外，请根据硬件平台更换与全局中断控制器相关的 API。

2、输出比较示例

该示例位于“examples/apb_timer_pwm_example.c”。在本示例中，产生 1KHz 的 PWM 以实现呼吸灯，每隔 0.1s 增加/减小 10% 的 PWM 占空比。这里假设 APB-TIMER 的时钟为 25MHz，定时器位宽为 16 位。使用通道 0 作为输出比较通道。另外，请根据硬件平台更换与延迟（delay）相关的 API。

3、输入捕获示例

该示例位于“examples/apb_timer_input_cap_example.c”。在本示例中，捕获红外接收器产生的脉冲信号，根据 NEC 协议解码出按键编号。这里假设 APB-TIMER 的时钟为 25MHz，定时器位宽为 32 位。使用通道 0 作为输入捕获通道。另外，请根据硬件平台更换与全局中断控制器相关的 API。