



暨南大学  
JINAN UNIVERSITY

# 第6章 嵌入式系统硬件 (3)

杨光华

物联网与物流工程研究院 / 电气信息学院

办公室：行政楼 631

电邮：ghyang@jnu.edu.cn

电话：8505687

声明：课件中的部分文字、图片、视频等源于网络，相应版权属于原创人

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器

# 10 输入/输出端口

---

## 主要内容

概述

寄存器

应用举例

# 一、概述

**S3C2410X有117个输入/输出端口。这些端口是：**

**A口（GPA）：23个输出口**

**B口（GPB）：11个输入/输出口**

**C口（GPC）：16个输入/输出口**

**D口（GPD）：16个输入/输出口**

**E口（GPE）：16个输入/输出口**

**F口（GPF）：8个输入/输出口**

**G口（GPG）：16个输入/输出口**

**H口（GPH）：11个输入/输出口**

这些端口都具有多功能，通过引脚配置寄存器，可以将其设置为所需要的功能，如：I/O功能、中断功能等等。

## 二、端口寄存器及引脚配置

每一个端口都有4个寄存器，它们是：引脚配置寄存器 **GPxCON**、数据寄存器 **GPxDAT**、引脚上拉寄存器**GPyUP**等 (其中x为A~H， y为B~H)

Register	Address	R/W	Description	Reset Value
GPXCON	0x560000x0	R/W	端口X配置寄存器	X
GPXDAT	0x560000x4	R/W	端口X数据寄存器	X
GPXUP	0x560000x8	R/W	端口X上拉寄存器	X
RESERVED	0x560000xC	R/W	端口X保留寄存器	-

# 端口寄存器

## (1) 输入/输出口配置寄存器(GPACON-GPHCON)

在S3C2410A中大多数**输入/输出口引脚是复用的**，因此需要用输入/输出口控制寄存器来确定每个引脚的功能

如果GPF0 – GPF7 和 GPG0 – GPG7为Power-OFF模式下的唤醒信号，那么这些管脚必须被配置成**中断模式**

# 端口寄存器

## (2) 输入/输出口数据寄存器(GPADAT-GPHDAT)

如果管脚被置成输出方式则可以**向相应的位写数据**；  
如果管脚被置成输入方式则可以**从相应的位读数据**

## (3) 输入/输出口上拉电阻使能寄存器(GPBUP-GPHUP)

该寄存器能够使能或禁止每个端口组是否需要**上拉电阻**。0表示使能，1表示禁止

如果该端口上拉电阻使能，则上拉电阻始终有效，无论是否设置了管脚的功能

# 1、端口A寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPACON	0x56000000	R/W	端口A引脚配置寄存器	0x7FFFFFFF
GPADAT	0x56000004	R/W	端口A数据寄存器	-
RESERVED	0x56000008	-	端口A保留寄存器	-
RESERVED	0x5600000C	-	端口A保留寄存器	-

## GPADAT寄存器为准备输出的数据

其值为23位[22: 0]

注意： (1) 当A口引脚配置为非输出功能时，其输出无意义；  
(2) 从引脚输入没有意义。



# 1、端口A寄存器及引脚配置

位号	位 名	位值:0	1	位号	位 名	位值:0	1
22	GPA22	输出	nFCE	10	GPA10	输出	ADDR25
21	GPA21	输出	nRSTOUT	9	GPA9	输出	ADDR24
20	GPA20	输出	nFRE	8	GPA8	输出	ADDR23
19	GPA19	输出	nFWE	7	GPA7	输出	ADDR22
18	GPA18	输出	ALE	6	GPA6	输出	ADDR21
17	GPA17	输出	CLE	5	GPA5	输出	ADDR20
16	GPA16	输出	nGCS5	4	GPA4	输出	ADDR19
15	GPA15	输出	nGCS4	3	GPA3	输出	ADDR18
14	GPA14	输出	nGCS3	2	GPA2	输出	ADDR17
13	GPA13	输出	nGCS2	1	GPA1	输出	ADDR16
12	GPA12	输出	nGCS1	0	GPA0	输出	ADDR0
11	GPA11	输出	ADDR26		FCE:	Flash片选	

## 2、端口B寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPBCON	0x56000010	R/W	端口B引脚配置寄存器	0x0
GPBDAT	0x56000014	R/W	端口B数据寄存器	-
GPBUP	0x56000018	R/W	端口B上拉寄存器	0x0
RESERVED	0x5600001C	-	端口B保留寄存器	-

**GPBDAT---为准备输出或输入的数据**

**其值为11位[10: 0]**

**GPBUP---端口B上拉寄存器，位[10: 0]有意义。**

**0：对应引脚设置为上拉**

**1：无上拉功能**

**注意：**当B口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

# 端口B引脚配置寄存器

位号	位 名	位值:00	01	10	11
21,20	GPB10	输入	输出	nXDREQ0	Reserved
19,18	GPB9	输入	输出	nXDACK0	Reserved
17,16	GPB8	输入	输出	nXDREQ1	Reserved
15,14	GPB7	输入	输出	nXDACK1	Reserved
13,12	GPB6	输入	输出	nXBACK	Reserved
11,10	GPB5	输入	输出	nXBREQ	Reserved
9,8	GPB4	输入	输出	TCLK0	Reserved
7,6	GPB3	输入	输出	TOUT3	Reserved
5,4	GPB2	输入	输出	TOUT2	Reserved
3,2	GPB1	输入	输出	TOUT1	Reserved
1,0	GPB0	输入	输出	TOUT0	Reserved

### 3、端口C寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPCCON	0x56000020	R/W	端口C引脚配置寄存器	0x0
GPCDAT	0x56000024	R/W	端口C数据寄存器	-
GPCUP	0x56000028	R/W	端口C上拉寄存器	0x0
RESERVED	0x5600002C	-	端口C保留寄存器	-

**GPCDAT---为准备输出或输入的数据**

**其值为16位[15: 0]**

**GPCUP---端口C上拉寄存器，位[15: 0]有意义。**

**0：对应引脚设置为上拉**

**1：无上拉功能**

**注意：**当C口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

# 端口C引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPC15	输入	输出	VD7	保留	15,14	GPC7	输入	输出	LCDVF2	保留
29,28	GPC14	输入	输出	VD6	保留	13,12	GPC6	输入	输出	LCDVF1	保留
27,26	GPC13	输入	输出	VD5	保留	11,10	GPC5	输入	输出	LCDVF0	保留
25,24	GPC12	输入	输出	VD4	保留	9,8	GPC4	输入	输出	VM	保留
23,22	GPC11	输入	输出	VD3	保留	7,6	GPC3	输入	输出	VFRAME	保留
21,20	GPC10	输入	输出	VD2	保留	5,4	GPC2	输入	输出	VLINE	保留
19,18	GPC9	输入	输出	VD1	保留	3,2	GPC1	输入	输出	VCLK	保留
17,16	GPC8	输入	输出	VD0	保留	1,0	GPC0	输入	输出	VEND	保留

## 4、端口D寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPDCON	0x56000030	R/W	端口D引脚配置寄存器	0x0
GPDDAT	0x56000034	R/W	端口D数据寄存器	-
GPDUP	0x56000038	R/W	端口D上拉寄存器	0xF000
RESERVED	0x5600003C	-	端口D保留寄存器	-

**GPDDAT---为准备输出或输入的数据**

其值为16位[15: 0]

**GPDUP---端口D上拉寄存器，位[15: 0]有意义。**

0：对应引脚设置为上拉

1：无上拉功能

初始化时，[15: 12]无上拉功能，而[11: 0]有上拉

**注意：**当D口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

# 端口D引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPD15	输入	输出	VD23	nSS0	15,14	GPD7	输入	输出	VD15	保留
29,28	GPD14	输入	输出	VD22	nSS1	13,12	GPD6	输入	输出	VD14	保留
27,26	GPD13	输入	输出	VD21	保留	11,10	GPD5	输入	输出	VD13	保留
25,24	GPD12	输入	输出	VD20	保留	9,8	GPD4	输入	输出	VD12	保留
23,22	GPD11	输入	输出	VD19	保留	7,6	GPD3	输入	输出	VD11	保留
21,20	GPD10	输入	输出	VD18	保留	5,4	GPD2	输入	输出	VD10	保留
19,18	GPD9	输入	输出	VD17	保留	3,2	GPD1	输入	输出	VD9	保留
17,16	GPD8	输入	输出	VD16	保留	1,0	GPD0	输入	输出	VD8	保留

## 5、端口E寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPECON	0x56000040	R/W	端口E引脚配置寄存器	0x0
GPEDAT	0x56000044	R/W	端口E数据寄存器	-
GPEUP	0x56000048	R/W	端口E上拉寄存器	0x0
RESERVED	0x5600004C	-	端口E保留寄存器	-

**GPEDAT---为准备输出或输入的数据**

其值为16位[15: 0]

**GPEUP---端口E上拉寄存器，位[15: 0]有意义。**

0：对应引脚设置为上拉

1：无上拉功能

初始化时，各个引脚都有上拉功能。

**注意：**当E口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。



# 端口E引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPE15	输入	输出	IICSDA	保留	15,14	GPE7	输入	输出	SDDAT0	保留
29,28	GPE14	输入	输出	IICSCL	保留	13,12	GPE6	输入	输出	SDCMD	保留
27,26	GPE13	输入	输出	SPICLK0	保留	11,10	GPE5	输入	输出	SDCLK	保留
25,24	GPE12	输入	输出	SPISIO	保留	9,8	GPE4	输入	输出	ISSDO	保留
23,22	GPE11	输入	输出	SPISO0	保留	7,6	GPE3	输入	输出	ISSDI	保留
21,20	GPE10	输入	输出	SDDAT3	保留	5,4	GPE2	输入	输出	CDCLK	保留
19,18	GPE9	输入	输出	SDDAT2	保留	3,2	GPE1	输入	输出	ISSCLK	保留
17,16	GPE8	输入	输出	SDDAT1	保留	1,0	GPE0	输入	输出	ISLRCK	保留

## 6、端口F寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPFCON	0x56000050	R/W	端口F引脚配置寄存器	0x0
GPFDAT	0x56000054	R/W	端口F数据寄存器	-
GPFUP	0x56000058	R/W	端口F上拉寄存器	0x0
RESERVED	0x5600005C	-	端口F保留寄存器	-

**GPFDAT---为准备输出或输入的数据**

其值为8位[7：0]

**GPFUP---端口F上拉寄存器，位[7：0]有意义。**

0：对应引脚设置为上拉

1：无上拉功能

初始化时，各个引脚都有上拉功能。

**注意：**当F口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

# 端口F引脚配置寄存器

位号	位 名	位 值			
		00	01	10	11
15,14	GPF7	输入	输出	EINT7	保留
13,12	GPF6	输入	输出	EINT6	保留
11,10	GPF5	输入	输出	EINT5	保留
9,8	GPF4	输入	输出	EINT4	保留
7,6	GPF3	输入	输出	EINT3	保留
5,4	GPF2	输入	输出	EINT2	保留
3,2	GPF1	输入	输出	EINT1	保留
1,0	GPF0	输入	输出	EINT0	保留

## 7、端口G寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPGCON	0x56000060	R/W	端口G引脚配置寄存器	0x0
GPGDAT	0x56000064	R/W	端口G数据寄存器	-
GPGUP	0x56000068	R/W	端口G上拉寄存器	0xF800
RESERVED	0x5600006C	-	端口G保留寄存器	-

**GPGDAT---为准备输出或输入的数据**

其值为16位[15: 0]

**GPGUP---端口G上拉寄存器，位[15: 0]有意义。**

0：对应引脚设置为上拉

1：无上拉功能

初始化时，[15:11]引脚无上拉功能，其它引脚有。

**注意：**当G口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

# 端口G引脚配置寄存器

位号	位 名	位 值				位号	位名	位 值			
		00	01	10	11			00	01	10	11
31,30	GPG15	输入	输出	EINT23	nYPON	15,14	GPG7	输入	输出	EINT15	SPICLK1
29,28	GPG14	输入	输出	EINT22	YMON	13,12	GPG6	输入	输出	EINT14	SPISI1
27,26	GPG13	输入	输出	EINT21	nXPON	11,10	GPG5	输入	输出	EINT13	SPISO1
25,24	GPG12	输入	输出	EINT20	XMON	9,8	GPG4	输入	输出	EINT12	LCD- PEN
23,22	GPG11	输入	输出	EINT19	TCLK1	7,6	GPG3	输入	输出	EINT11	nSS1
21,20	GPG10	输入	输出	EINT18	保留	5,4	GPG2	输入	输出	EINT10	nSS0
19,18	GPG9	输入	输出	EINT17	保留	3,2	GPG1	输入	输出	EINT9	保留
17,16	GPG8	输入	输出	EINT16	保留	1,0	GPG0	输入	输出	EINT8	保留

LCD-PEN:POWER\_ENABLE

nSS0:SPIO\_SELECT

## 8、端口H寄存器及引脚配置

Register	Address	R/W	Description	Reset Value
GPHCON	0x56000070	R/W	端口H引脚配置寄存器	0x0
GPHDAT	0x56000074	R/W	端口H数据寄存器	-
GPHUP	0x56000078	R/W	端口H上拉寄存器	0x0
RESERVED	0x5600007C	-	端口H保留寄存器	-

**GPHDAT---为准备输出或输入的数据**

**其值为11位[10：0]**

**GPHUP---端口H上拉寄存器，位[10：0]有意义。**

**0：对应引脚设置为上拉**

**1：无上拉功能**

**注意：**当H口引脚配置为非输入/输出功能时，其寄存器中的值没有意义。

# 端口H引脚配置寄存器

位号	位 名	位值:00	01	10	11
21,20	GPH10	输入	输出	CLKOUT1	Reserved
19,18	GPH9	输入	输出	CLKOUT0	Reserved
17,16	GPH8	输入	输出	UCLK	Reserved
15,14	GPH7	输入	输出	RXD2	nCTS1
13,12	GPH6	输入	输出	TXD2	nRTS1
11,10	GPH5	输入	输出	RXD1	Reserved
9,8	GPH4	输入	输出	TXD1	Reserved
7,6	GPH3	输入	输出	RXD0	Reserved
5,4	GPH2	输入	输出	TXD0	Reserved
3,2	GPH1	输入	输出	nRTS0	Reserved
1,0	GPH0	输入	输出	nCTS0	Reserved

UCLK为USB的

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器



# 11 定时器

---

## 主要内容

概述

结构

寄存器

应用举例

# 一、概述

## 1、S3C2410X定时器的主要特性

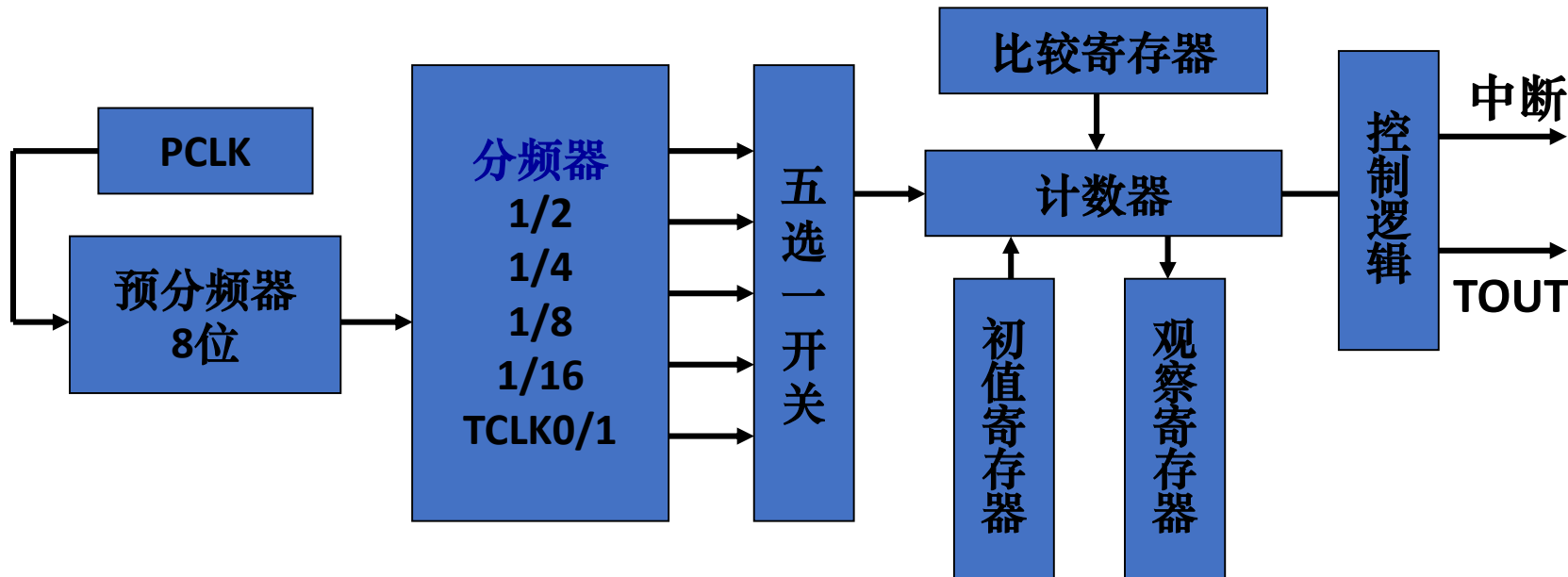
- 5个16位定时器
- 2个8位预分频器和2个4位分频器
- 可编程PWM输出占空比
- 具有初值自动重装连续输出模式和单脉冲输出模式
- 具有死区生成器

# S3C2410X定时器的主要特性

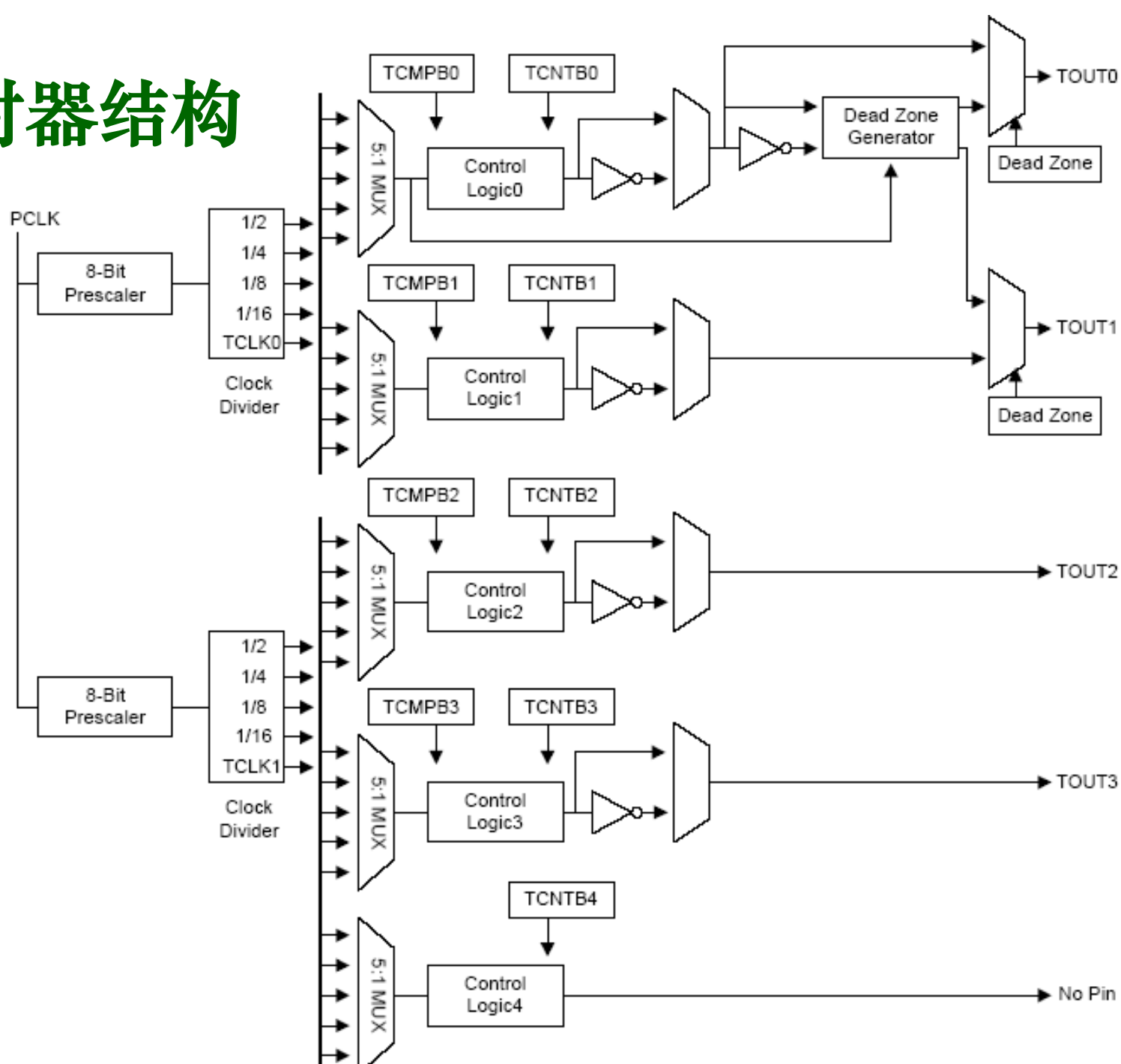
- **S3C2410有5个16位的定时器**，定时器0－3具有PWM（脉宽调制）功能。定时器4是一个内部定时器，没有输出引脚，供内部使用。定时器0有死区产生器，通常用于大电流设备控制。
- **有2个8位预分频器和2个4位分频器**。定时器0和定时器1分享同一个8位的预分频器和分频器，定时器2、3、4分享另一个预分频器和分频器，分频器有2、4、8、16这4种分频值。定时器从分频器接收自己的时钟信号，时钟分频器从相应的预分频器接收时钟信号。

# 定时器结构

- **时钟控制**：每个定时器设置有 预分频器、分频器
- **定时器组成（5部分）**：减法计数器、初值寄存器、比较寄存器、观察寄存器、控制逻辑等部分构成

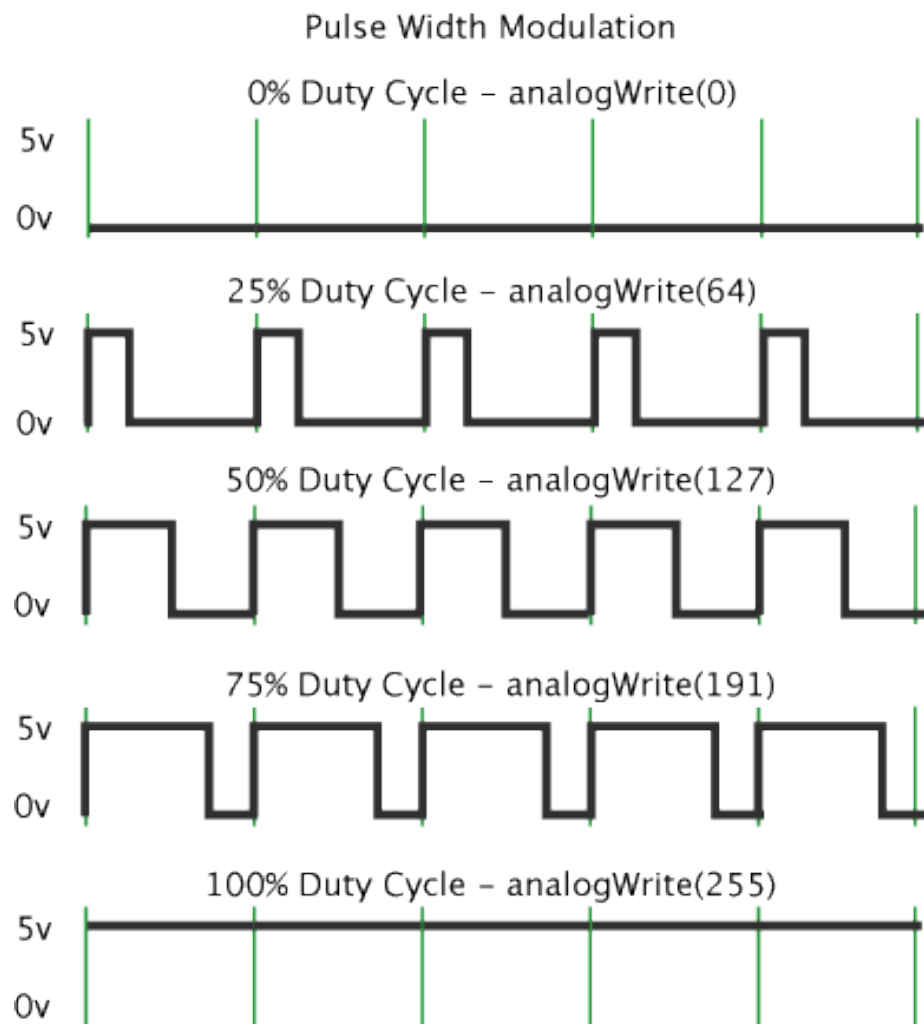


# 定时器结构



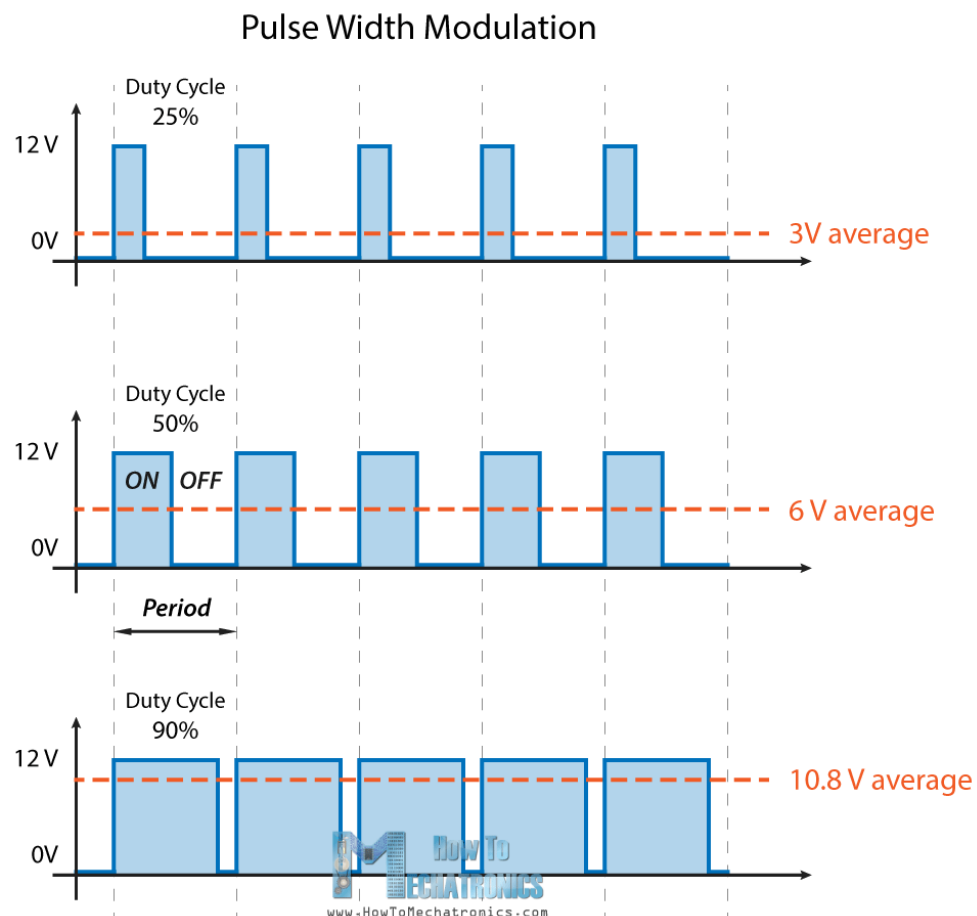
# PWM（脉宽调制）概念

- PWM（脉宽调制）：只改变方波信号的占空比，而不改变幅度和周期
- PWM信号的产生和传输都是数字式的
- PWM技术可有效实现模拟信号，广泛应用于从测量、通信到功率控制与变换的许多领域中



# 用PWM实现模拟信号

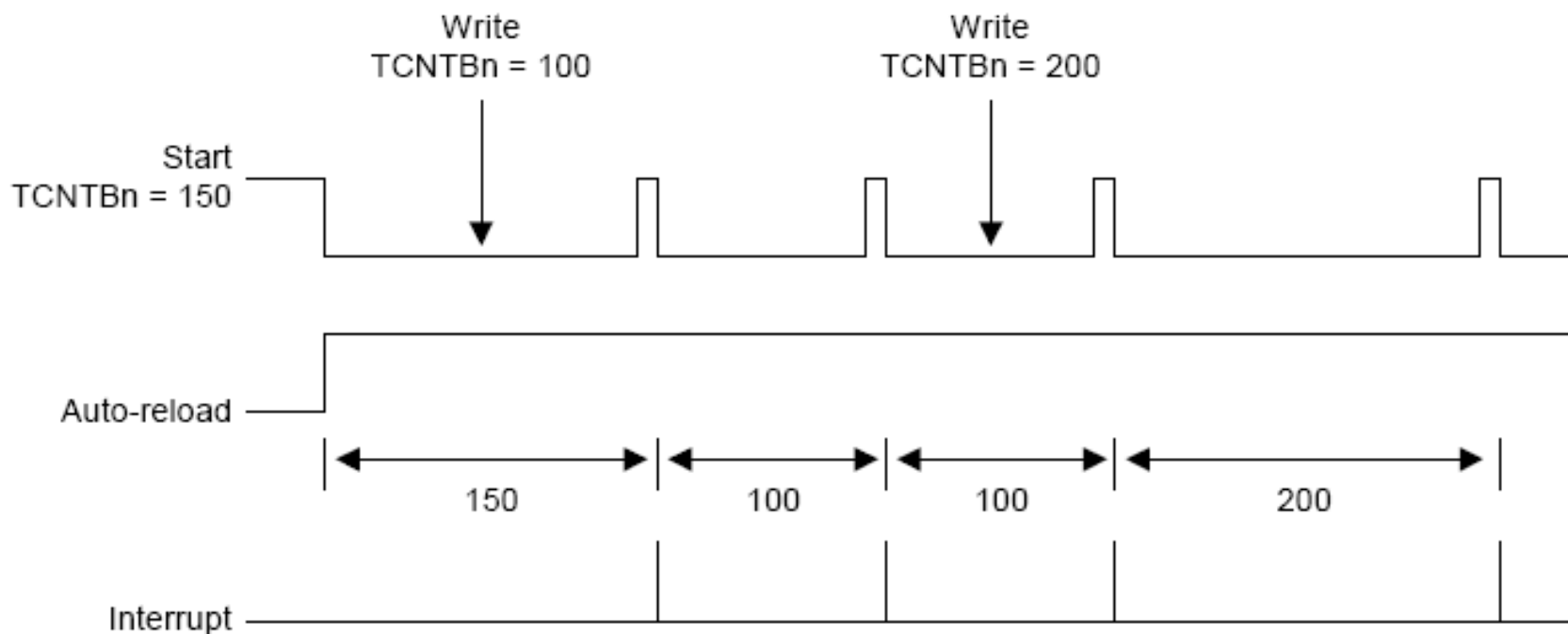
- 如果调制信号的频率远大于信号接受者的分辨率，则接收者获得的是信号的平均效果，其信号大小的平均值与信号的占空比有关
- 其平均值与占空比成正比，信号的占空比越大，平均信号越强
- 只要带宽足够（频率足够高或周期足够短），任何模拟信号都可以使用PWM 来实现



## 2、工作原理

### (1) 定时器工作过程

装入初值、启动计数，计数结束产生中断请求，并且可以重装初值连续计数。如下图所示：



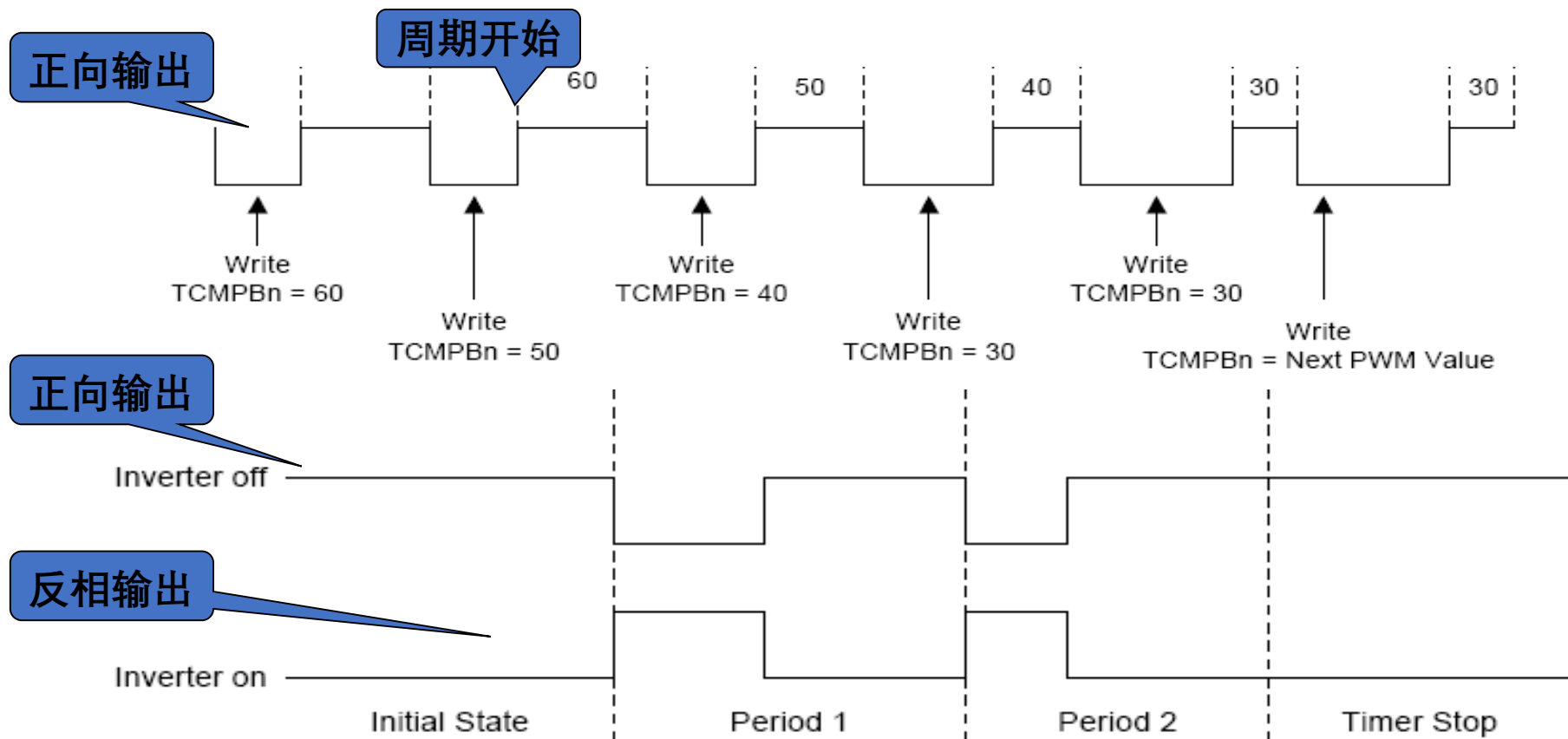


## (2) 初值自动重装、手动装载和双缓冲

- **初值自动重装：** 5个定时器都具有此功能。当计数器中值减到0后，若设置了自动重装功能，则在下一计数周期开始前将初值装入计数器重新计数。
- **初值手动装载功能：** 在启动计数前，必须使用手动装载功能将初值装入计数器，而初值自动重装仅是一次计数结束后重新装入初值。
- **双缓冲功能：** 如果定时器正在工作，此时写入新的数据到TCNTBn、或者到TCMPBn，该写入的数据不影响本次定时器的操作。当定时器到达0后下一次运行定时器时，新写入的TCNTBn、或者TCMPBn才生效。

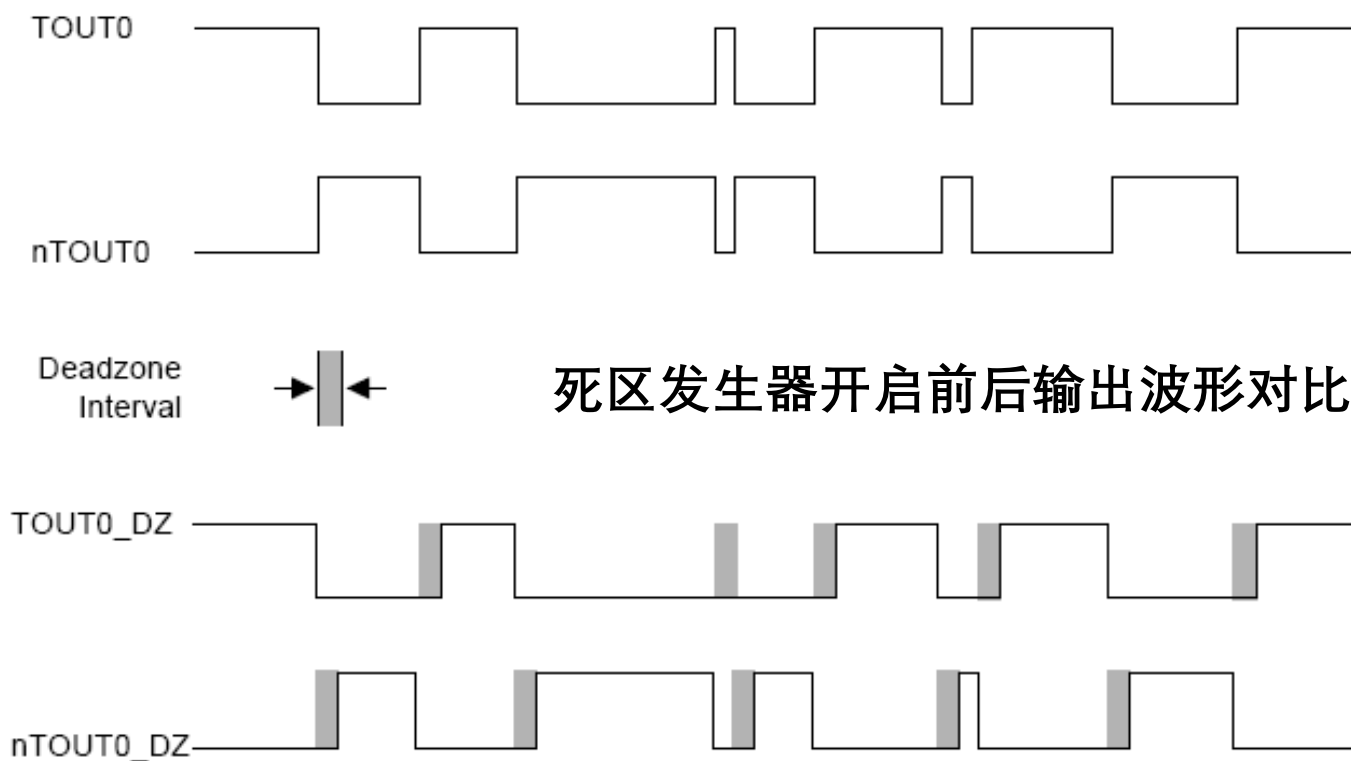
### (3) PWM输出

- 寄存器TCMPB: 当计数器TCNT中的值减到与TCMPB的值相同时, TOUT的输出值取反
- 改变TCMPB的值, 便改变了输出方波的占空比
- TOUT的输出可以设置为反相输出



## (4) 死区产生器

- **死区**：是一小段时间间隔，在这个时间间隔内，禁止两个开关同时处于开启状态。死区是在功率设备控制中常采用的一种技术，防止两个开关同时打开起反作用。
- Timer0具有死区发生器功能，可用于控制大功率设备



## (5) DMA请求模式

- **DMA功能：** 5个定时器都有DMA请求功能，但是在同一时刻只能设置一个使用DMA，通过设置其DMA模式位来实现
- **DMA请求过程：** 定时器可以在任意时间产生DMA请求，并且保持DMA请求信号（nDMA\_REQ）为低直到定时器收到ACK信号。当定时器收到ACK信号时，它使请求信号变得无效。
- **DMA请求与中断的关系：** 如果一个定时器被配置为DMA模式，该定时器不会产生中断请求了。其他的定时器会正常的产生中断

### 3、计数时钟和输出计算

1) 定时器输入时钟频率 $f_{Tclk}$ （即计数时钟频率）：

$$f_{Tclk} = [f_{pclk} / (\text{Prescaler} + 1)] \times \text{分频值}$$

式中：Prescaler，预分频值，0---255；分频值为1/2、1/4、1/8、1/16

2) PWM输出时钟频率：

$$\text{PWM输出时钟频率} = f_{Tclk} / \text{TCNTBn}$$

3) PWM输出信号占空比（即高电平持续时间所占信号周期的比例）：

$$\text{PWM输出信号占空比} = \text{TCMPBn} / \text{TCNTBn}$$

## 定时器最大、最小输出周期

设PCLK的频率为50MHz，经过预分频和分频器后，送给定时器的可能计数时钟频率由表4-7-1给出。

表4-7-1 定时器最大、最小输出周期

分频值	最小输出周期 (预分频器=0、 TCNTBn=1)	最大输出周期 (预分频器=255、 TCNTBn=6553 5)	最小输出周期 (预分频器=0、 TCNTBn=6553 5)	最小输出周期 (预分频器=0、 TCNTBn=255)
1/2	25.00MHz(0.04μs)	0.6710s	381Hz	97656
1/4	12.50MHz(0.08μs)	1.3421s	191Hz	48828
1/8	6.250MHz(0.16μs)	2.6843s	95Hz	24414
1/16	3.125MHz(0.32μs)	5.3686s	48Hz	12207

## 三、定时器专用寄存器

共有6种、17个寄存器

Register	Address	R/W	Description	Reset Value
TCFG0	0x51000000	R/W	配置寄存器 0	0x00000000
TCFG1	0x51000004	R/W	配置寄存器 1	0x00000000
TCON	0x51000008	R/W	控制寄存器	0x00000000
TCNTBn	0x510000xx	R/W	计数初值寄存器(5个)	0x0000
TCMPBn	0x510000xx	R/W	比较寄存器(4个)	0x0000
TCNTOn	0x510000xx	R	观察寄存器(5个)	0x0000

TCNTBn---Timern计数初值寄存器（计数缓冲寄存器），16位

TCMPBn---Timern比较寄存器（比较缓冲寄存器），16位

TCNTOn---Timern计数读出寄存器，16位

# 1、TCFG0---预分频器配置寄存器

31	.....	24	23	.....	16	15	.....	8	7	.....	0
保留（为0）			Dead zone length			Prescaler1			Prescaler0		

## Dead zone length---死区宽度设置位

其值N为： 0~255，以timer0的定时时间为单位

死区宽度为：  $(N+1) \times \text{timer0的定时时间}$

## Prescaler1---timer2、3、4的预分频值

其值N为： 0~255

输出频率为：  $\text{PCLK} \div (N+1)$

## Prescaler0--- timer0、1的预分频值

其值N为： 0~255

输出频率为：  $\text{PCLK} \div (N+1)$



## 2、TCFG1---DMA模式与分频选择寄存器

31 ... 24	23 ... 20	19...16	15...12	11...8	7 ... 4	3 ... 0
保留（为0）	DMA mode	MUX4	MUX3	MUX2	MUX1	MUX0

### DMA mode---DMA通道选择设置位

0000：不使用DMA方式，所有通道都用中断方式

0001：选择timer0

0010：选择timer1

0011：选择timer2

0100：选择timer3

0101：选择timer4

011X：保留

### MUX4~ MUX0---timer4~timer0分频值选择

第14讲到此

1/2

0001：1/4

0010：1/8

0011：1/16

01XX：选择外部TCLK0、1（对timer0、1是选TCLK0，对timer4、3、2是选TCLK1）

### 3、TCN---定时器控制寄存器

31...23	22	21	20	19	18	17	16	15	14	13
保留	TL4	TUP4	TR4	TL3	TO3	TUP3	TR3	TL2	TO2	TUP2

12	11	10	9	8	7...5	4	3	2	1	0
TR2	TL1	TO1	TUP1	TR1	保留	DZE	TL0	TO0	TUP0	TR0

TL4~TL0---计数初值自动重装控制位

0: 单次计数

1: 计数器值减到0时，自动重新装入初值连续计数。

TUP4~TUP0---计数初值手动装载控制位。

0: 不操作

1: 立即将TCNTBn中的计

数初值装载到计数寄存器TCNTn中。

**说明:** 如果没有执行手动装载初值，则计数器启动时无初值。

### 3、TCON---定时器控制寄存器（续）

31...23	22	21	20	19	18	17	16	15	14	13
保留	TL4	TUP4	TR4	TL3	TO3	TUP3	TR3	TL2	TO2	TUP2

12	11	10	9	8	7...5	4	3	2	1	0
TR2	TL1	TO1	TUP1	TR1	保留	DZE	TL0	TO0	TUP0	TR0

**TR4~TR0---TIMER4~TIMER0运行控制位**

0：停止

1：启动对应的TIMER

**TO3~TO0--- TIMER4~TIMER0输出控制位**

0：正相输出

1：反相输出

**DZE---TIMER0死区操作控制位**

0：禁止死区操作

1：使能死区操作

# 四、定时器的使用

## 1、定时器初始化方法

- (1) 写TCFG0，设置计数时钟的预分频值和Timer0死区宽度；
- (2) 写TCFG1，选择各个定时器的分频值和DMA、中断服务；
- (3) 对TCNTBn和TCMPBn分别写入计数初值和比较初值；
- (4) 写TCON，设置计数初值自动重装、手动装载初值、设置反相输出；
- (5) 再写TCON，清除手动装载初值位、设置正相输出、启动计数。

## 2、定时器停止运行方法

写TCON，禁止计数初值自动重装。（一般不使用运行控制位停止运行）

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器

# 12 UART通用异步串行接口

---

## 主要内容

概述

UART专用寄存器

应用举例

RS232与UART

重点:

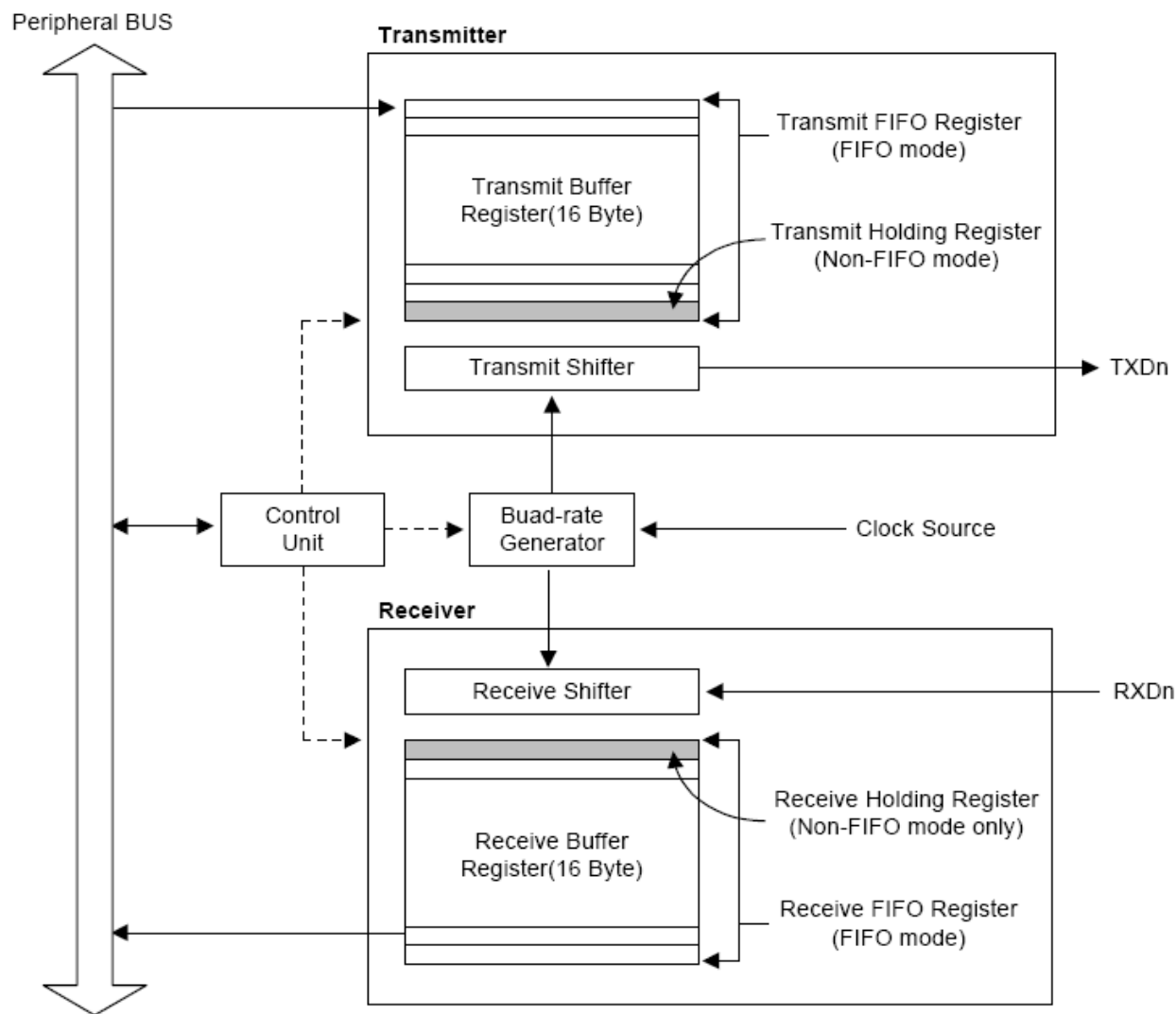
- (1) UART原理
- (2) 奇偶校验
- (3) RS232 与 UART 关系

# 一、概述

- UART实际上是**增加了一些功能的并—串行转换器**
- UART发送器本质上是一个**移位寄存器**，可以并行装载数据，然后在串行时钟脉冲的控制下再将数据一位一位按顺序移出；反过来，接收器是把串行比特流接收到一个移位寄存器中，然后由处理器读取
- **S3C2410 有三个独立的通用异步串行口（UART）**：UART0、UART1、UART2
- 每个串口都可以在中断和DMA 两种模式下进行收发。UART支持的最高波特率达230.4kbps

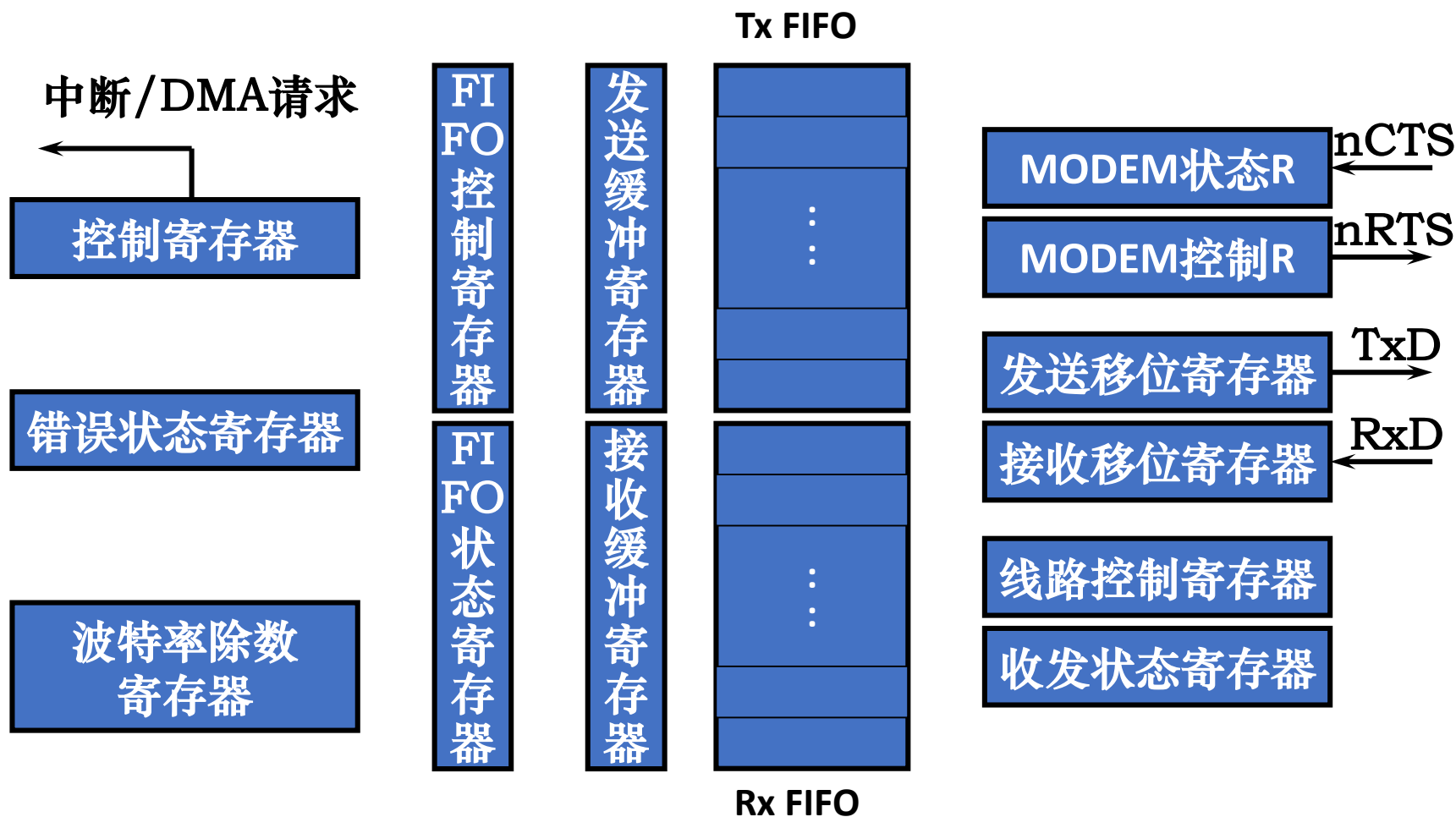
# 串行口结构

- UART主要有4部分构成：**波特率发生器、接收器、发送器和控制单元**
- 发送器和接收器各包含1个16字节的FIFO寄存器和移位寄存器





# 接收器/发送器的结构



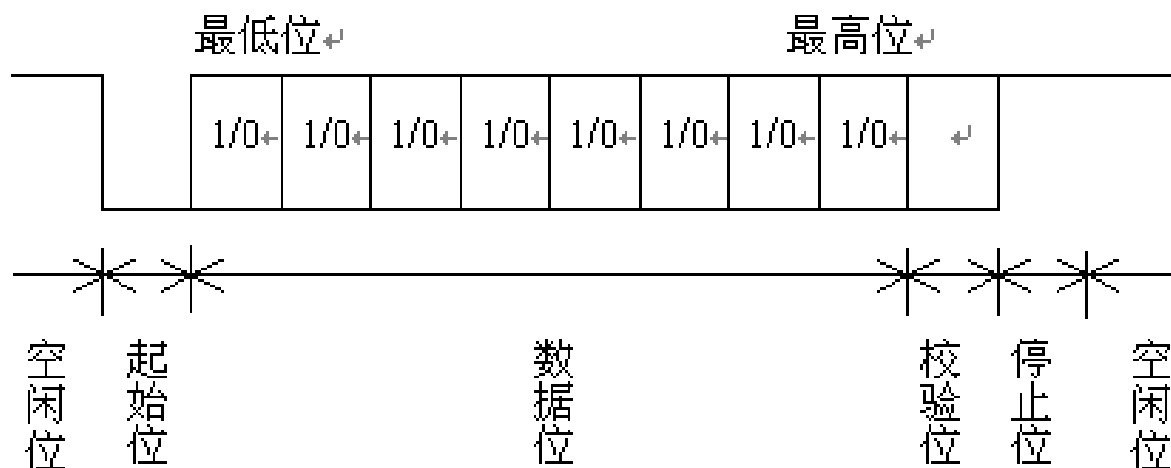
# 帧格式

- UART将数据组成帧进行发送和接收，帧格式可通过线路控制器（**ULCONn**）来设置，一帧包括：

1个起始位，5~8个数据位，

1个可选奇偶校验位，1~2个停止位

- 中止信号：发送超过一帧0作为中止信号



# 误码检测

- 在任何存在着潜在噪声的介质（比如串行线）中进行数据传输时，都有可能出现误码，需进行检测
- UART在传输的字符中专用一位来作误码检测---奇偶校验位
- 奇偶校验位的作用是使发送字符中“1”的个数为偶数（偶校）或奇数（奇校）
  - 发送方计算待发送字符的奇偶性，并据此对校验位进行设置，校验位作为待发送字符的一部分一同被发送；
  - 接收方对收到的字符再次进行奇偶计算，并和接收到的校验位进行比较，相符则传输正确，否则传输出错
  - 接收时，收方必须预先知道发送方采用的是哪种校验方式.

# 误码检测

- 例 设待发送字符为

01000000

- 若用偶校，校验位应为1，字符与校验位一同发送，实发送字符为010000001；
- 若用奇校，校验位应为0；字符与校验位一同发送，实发送字符为010000000；

- 奇偶校验纠错能力有限

- 发方用偶校

- 010000001，1的个数为偶，正确！
- 000000001，1的个数为奇，错误！
- 010101001，1的个数为偶，正确？

- 发方用奇校

- 010000000，1的个数为奇，正确！
- 010000001，1的个数为偶，错误！
- 010101000，1的个数为奇，正确？

要求高时可用循环冗余校  
验**CRC**

# 串行口的波特率发生器

- UART 的波特率发生器为传输提供串行移位时钟
- 波特率发生器以PCLK或UCLK为时钟源
- 波特率数值决定于波特率除数寄存器（UBRDIVn）的值：

$$\text{UBRDIVn} = (\text{int}) (\text{CLK} / (f_B * 16)) - 1$$

其中CLK为所选择的时钟频率， $f_B$ 为波特率。

$$f_B = \text{CLK} / 16 / (\text{UBRDIVn} + 1)$$

例如，如果波特率为115200bps 且PCLK 或UCLK 为40MHz,则：

$$\begin{aligned}\text{UBRDIVn} &= (\text{int}) (40000000 / (115200 * 16)) - 1 \\ &= (\text{int}) (21.7) - 1 \\ &= 21 - 1 = 20\end{aligned}$$

# 串行口波特率误差极限

在应用中，实际波特率往往与理想波特率有差别，其误差不能超过一定的范围，其极限为：UART传输10bit数据的时间误差应该小于1.87%（3/160）。

- 实际的传输10bit所需时间

$$t_{\text{true}} = (\text{UBRDIVn} + 1) \times 16 \times 10 / \text{PCLK}$$

- 理想情况下传输10位需要的时间

$$t_{\text{ideal}} = 10 / \text{baud\_rate}$$

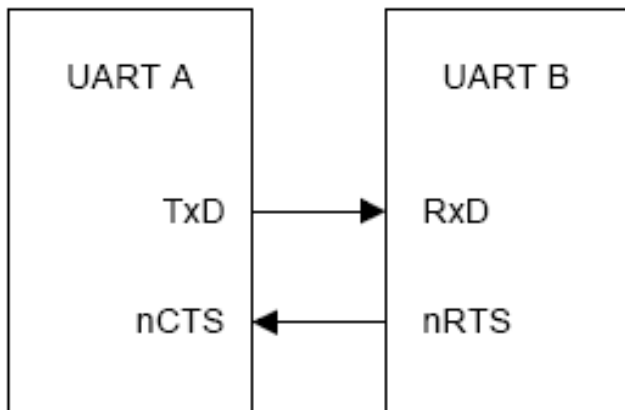
$$\text{UART error} = ((t_{\text{true}} - t_{\text{ideal}}) / t_{\text{ideal}}) \times 100\%$$

# 串行口的自动流控制功能

- 自动流控制是利用信号nRTS、nCTS来实现的。在接收数据时，只要接收FIFO中有两个空字节就会使nRTS有效，使对方发送数据；在发送数据时，只要nCTS有效，就会发送数据。其实现过程如下图所示

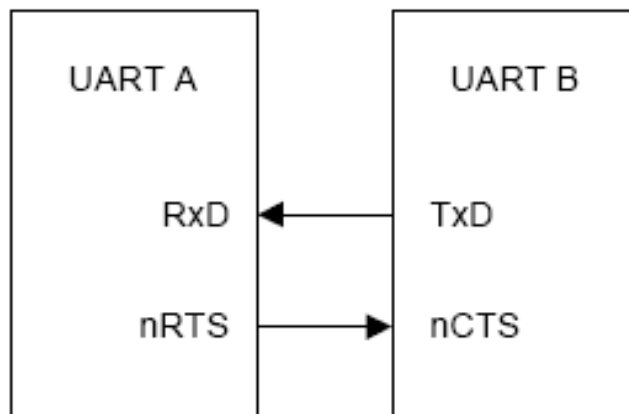
**nRTS：请求对方发送**

Transmission case in  
UART A



**nCTS：清除请求发送**

Reception case in  
UART A



# 中断或DMA请求

- 每个UART都有3类、7种事件产生中断请求或者DMA请求
  - **接收中断**：若接收模式设为中断，当接收缓冲器寄存器收到数据或接收FIFO达到预先设定的触发级时，将产生接收中断请求
  - **发送中断**：若发送模式设为中断，当发送缓冲器寄存器空或发送FIFO达到预先设定的触发级时，将发生发送中断请求
  - **错误中断**



# 中断或DMA请求

- 错误中断：一共有4种错误中断

- 溢出错误
- 奇偶检验错误
- 帧格式错误
- 传输中断信号错误

- 说明：

- 对于“奇偶校验错误、帧格式错误、传输中断信号错误”中断，在数据接收时就产生了，但是在数据接收产生时并非出现中断请求，而是在读出错误数据时才出现中断请求。
- 如果设置的是DMA模式，而不是中断请求模式，对于以上所出现的中断请求，应该是DMA请求。
- 传输中断信号定义：在超出一帧的时间内，全部输出低电平。

## 二、UART专用寄存器

3个UART，每个都有11个专用寄存器，共29个寄存器

Register	Address	R/W	Description	Reset Value
ULCON <sub>n</sub>	0x5000x000	R/W	线路控制寄存器	0x00
UCON <sub>n</sub>	0x5000x004	R/W	控制寄存器	0x00
UFCON <sub>n</sub>	0x5000x008	R/W	FIFO控制寄存器	0x00
UMCON <sub>n</sub>	0x5000x00C	R/W	MODEM控制寄存器*	0x00
UTRSTAT <sub>n</sub>	0x5000x010	R	发送/接收状态寄存器	0x6
UERSTAT <sub>n</sub>	0x5000x014	R	Rx错误状态寄存器	0x0
UFSTAT <sub>n</sub>	0x5000x018	R	FIFO状态寄存器	0x00
UMSTAT <sub>n</sub>	0x5000x01C	R	MODEM状态寄存器*	0x0
UTXH <sub>n</sub>	0x5000x020/23	W	发送缓冲寄存器	-
URXH <sub>n</sub>	0x5000x024/27	R	接收缓冲寄存器	-
UBRDIV <sub>n</sub>	0x5000x028	R/W	波特率除数寄存器	-

### 三、UART应用举例

- 编写一程序，使用S3C2410X的UART2进行串行数据收发，要求用脉冲请求中断的方式、使用收/发FIFO，8个数据位、1个停止位、不校验，波特率为125kb/s。设Pclk为50MHz。（提示：主程序对UART2初始化、引脚配置、中断初始化等，并进行一次发送；中断服务程序进行数据收发，标签清除中断请求标志和中断服务标志）

#### (1) 计算波特率除数：

由公式：
$$\text{UBRDIVn} = (\text{int}) \left( \frac{\text{CLK}}{f_B * 16} \right) - 1$$

这里：Pclk=50MHz,  $f_B = 125\text{kb/s}$

计算得：UBRDIVn=25 -1=24

#### (2) UART2控制寄存器：

线路控制寄存器：ULCON2=0b 0 000 0 11=0x03

含义：非红外、不校验、1个停止位、8个数据位

控制寄存器： UCON2=0b 0 0 0 0 0 0 0 0 1 0 1=0x05

含义：选Pclk、发/收中断脉冲请求、关闭接收超时中断、允许接收错误中断、不回送、不发送暂停信号、发/收用中断方式。

FIFO控制寄存器： UFCON2=0b 10 01 0 0 0 1=0x91

含义：发/收FIFO选8字节触发、保留位为0、不复位发/收FIFO、使能FIFO。

### (3) 引脚配置

需要设置TxD2、RxD2，它们对应GPH6、GPH7，在GPH配置寄存器GPHCON中的位置为：

0b 1 0 1 0 \* \* \* \* \* \* \* \*

方法： GPHCON= GPHCON&~(0xF<<12)|(0xA<<12)

## (4) 中断寄存器设置

中断模式寄存器:  $\text{INTMOD} \&= \sim(1 \ll 15)$

INT\_UART2位于第15位, 将UART2设置为IRQ中断

中断屏蔽寄存器:  $\text{INTMSK} \&= \sim(1 \ll 15)$

中断优先级寄存器PRIORITY:

不设置, 使用固定优先级。

子中断屏蔽寄存器:  $\text{INTSUBMSK} \&= \sim(7 \ll 6)$

INT\_ERR2、INT\_TXD2、INT\_RXD2位于子中断屏蔽寄存器中的8、7、6位。

## (5) 在中断服务程序中对寄存器的操作

清除中断标志寄存器相应位:

$\text{SRCPND} \&= \sim(1 \ll 15)$

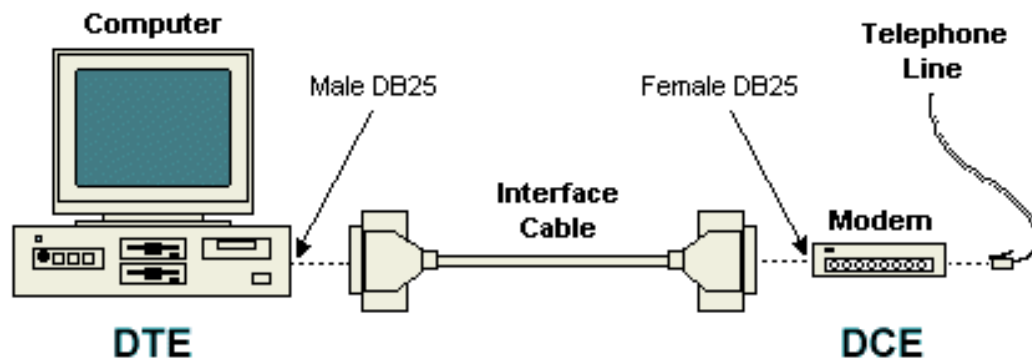
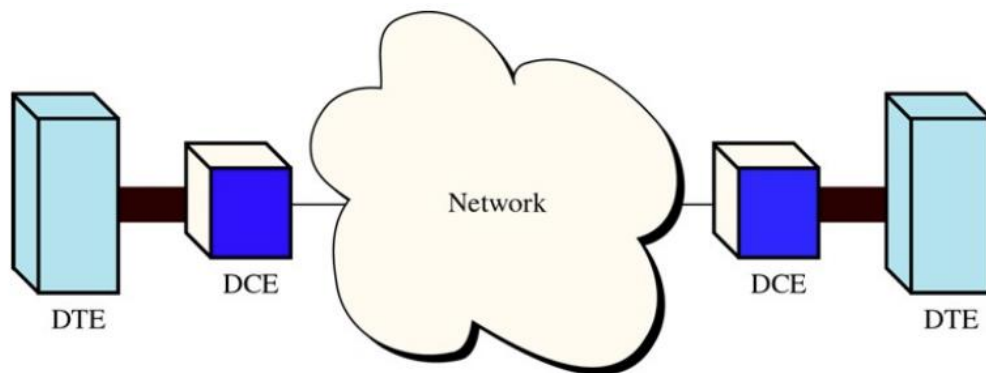
清除中断服务寄存器相应位:

$\text{INTPND} \&= \sim(1 \ll 15)$

## 四、RS-232与UART

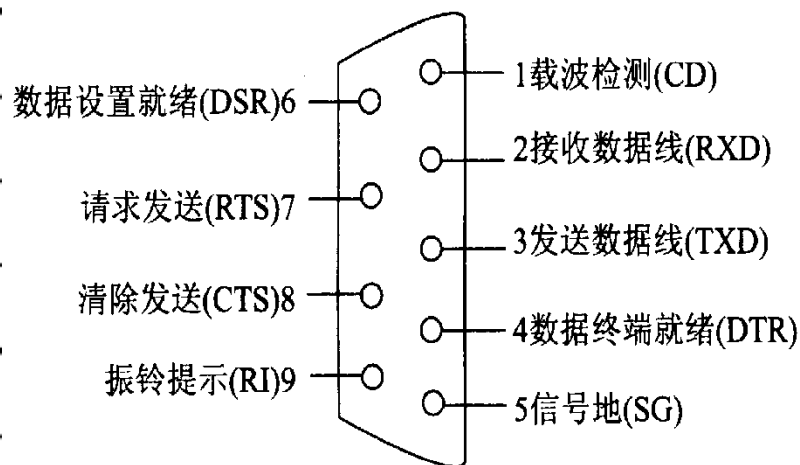
- RS-232C概述

- 数据终端设备DTE 和数据通信设备DCE之间数据交换接口标准，现在也用于DTE之间的数据交换
- 标准对接口的**机械特性、电气特性、信号功能**等进行了规范
- PC上COM1及COM2即是RS-232C接口



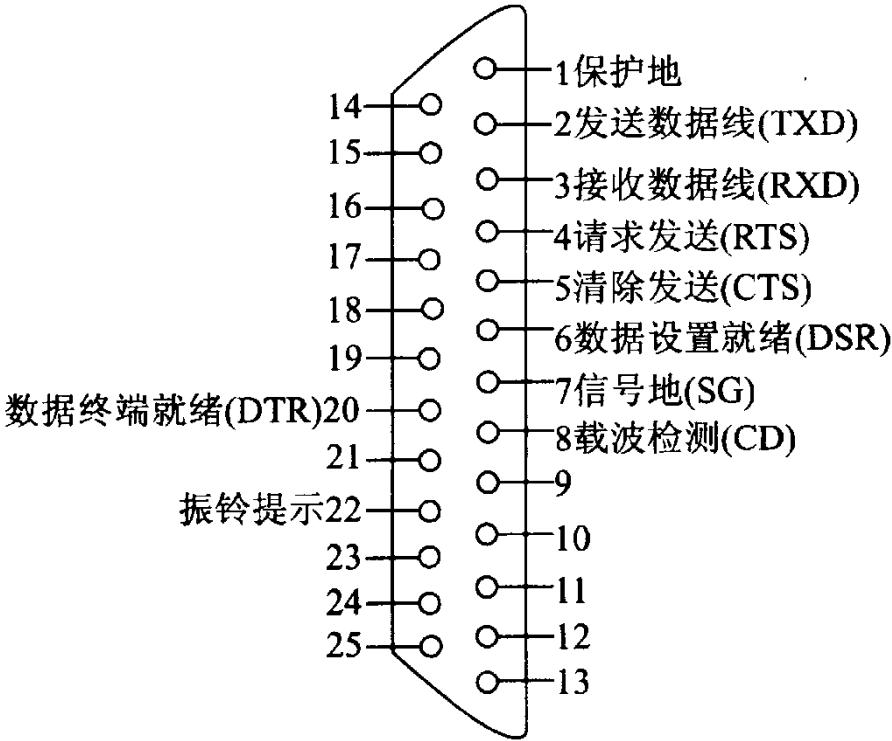
# RS-232C接口的机械特性

9 针串口 (DB9)		
针号	功能说明	缩写
1	数据载波检测	DCD
2	接收数据	RXD
3	发送数据	TXD
4	数据终端准备	DTR
5	信号地	SG
针号	功能说明	缩写
6	数据设备准备好	DSR
7	请求发送	RTS
8	清除发送	CTS
9	振铃指示	RI



# RS-232C接口的机械特性

25 针串口 (DB25)		
针号	功能说明	缩写
8	数据载波检测	DCD
3	接收数据	RXD
2	发送数据	TXD
20	数据终端准备	DTR
7	信号地	GND
针号	功能说明	缩写
6	数据准备好	DSR
4	请求发送	RTS
5	清除发送	CTS
22	振铃指示	RI





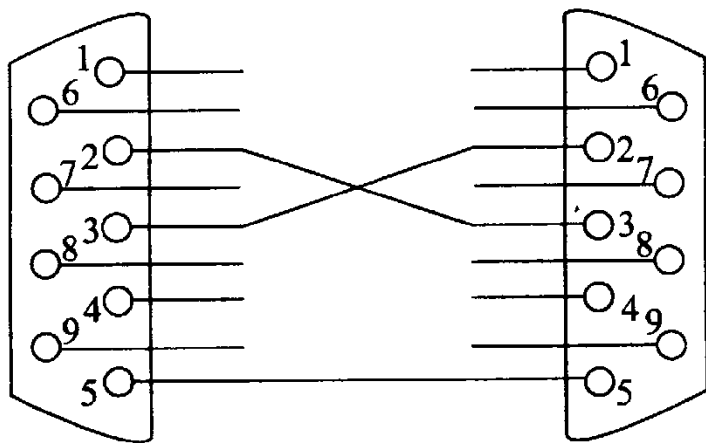
# RS-232引脚功能

- RS-232C DB-9各引脚功能如下

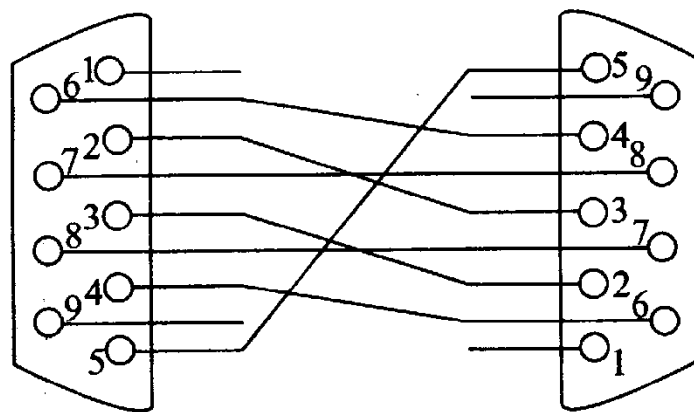
- CD：载波检测。主要用于Modem通知计算机其处于在线状态，即Modem检测到拨号音
- RXD：接收数据线。用于接收外部设备送来的数据
- TXD：发送数据线。用于将计算机的数据发送给外部设备
- DTR：数据终端就绪。当此引脚高电平时，通知Modem可以进行数据传输，计算机已经准备好
- SG：信号地
- DSR：数据设备就绪。此引脚为高电平时，通知计算机Modem已经准备好，可以进行数据通信
- RTS：请求发送。此引脚由计算机来控制，用以通知Modem马上传送数据至计算机；否则，Modem将收到的数据暂时放入缓冲区中
- CTS：清除发送。此引脚由Modem控制，用以通知计算机将要传送的数据送至Mo-dem
- RI：振铃提示。Modem通知计算机有呼叫进来，是否接听呼叫由计算机决定

# RS-232连接方式

- DTE之间通信时，RS-232C有两种连接方式：
  - 简单连接又称三线连接，即只连接发送数据线、接收数据线和信号地
  - 完全连接方式,如右图所示



简单连接



完全连接方式

# RS-232驱动程序界面—PC超级终端

每秒位数 (B) :

2400



数据位 (D) :

8



奇偶校验 (P) :

无



停止位 (S) :

1



数据流控制 (F) :

无



Xon / Xoff

硬件

无

# UART电平

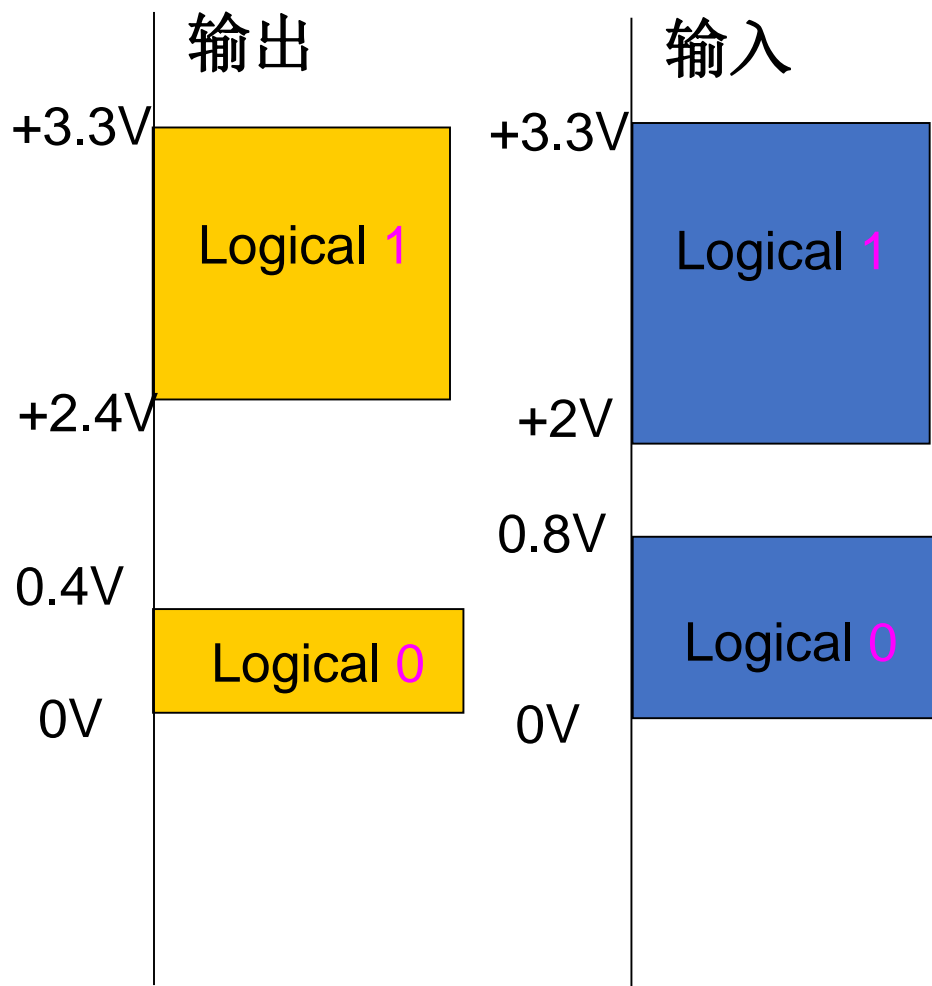
- 处理器上的UART接口通常为LVTTTL电平

- 输出

- 逻辑0 : 0V~0.4V
- 逻辑1 : 2.4V~3.3V

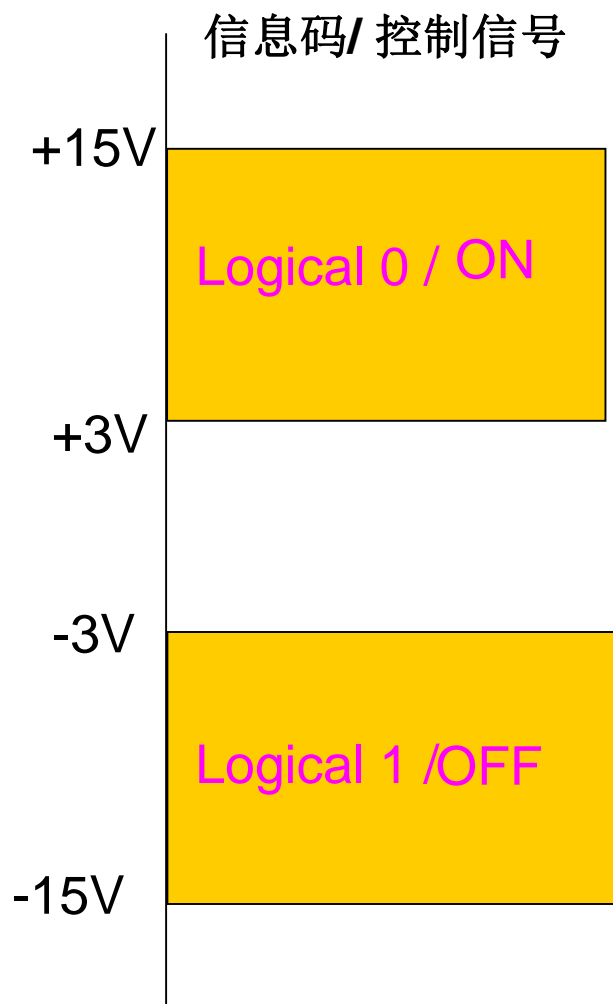
- 输入

- 逻辑0 : 0V~0.8V
- 逻辑1 : 2V~3.3V



# RS-232C接口的电气特性

- TxD和Rx D（信息码）
  - 逻辑1（MARK） = -3V~-15V
  - 逻辑0（SPACE） = +3~+15V
- RTS、CTS、DSR、DTR和DCD等控制线
  - 信号有效（ON） = +3~+15V
  - 信号无效（OFF） = -3~-15V



# RS-232接口硬件设计

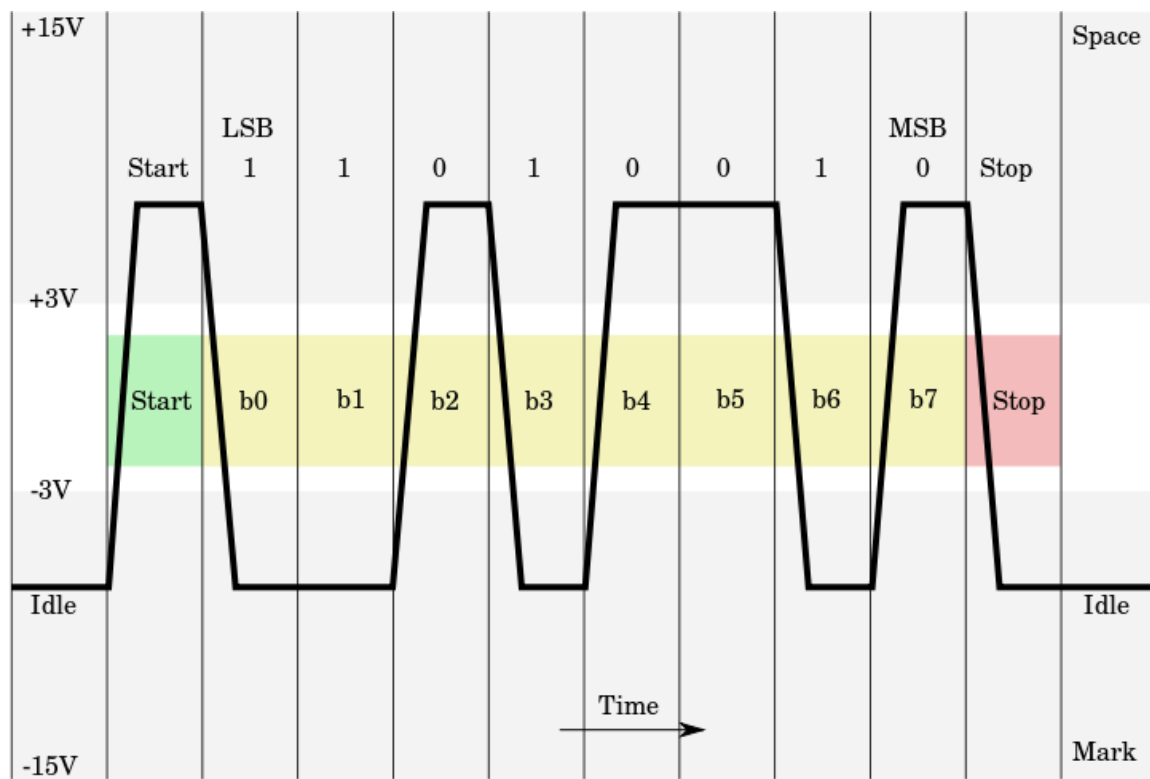
- UART LVTTTL 电平

- 逻辑0：0V~0.8V
- 逻辑1：2V~3.3V

- RS-232 电平

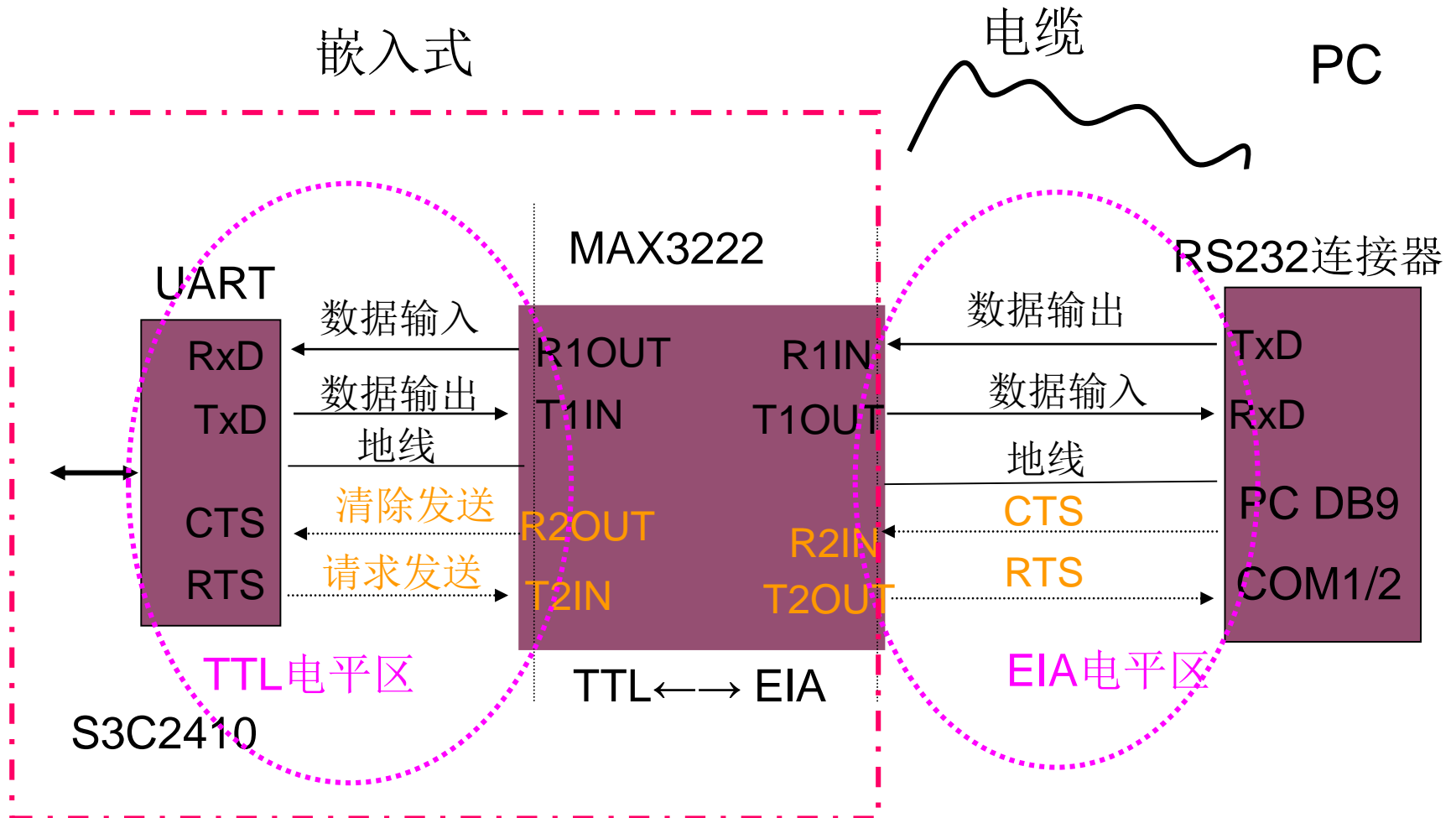
- 逻辑0：+3~+15V
- 逻辑1：-3V~-15V

- 欲将UART 与RS-232C连接起来通信，必须进行电平变换

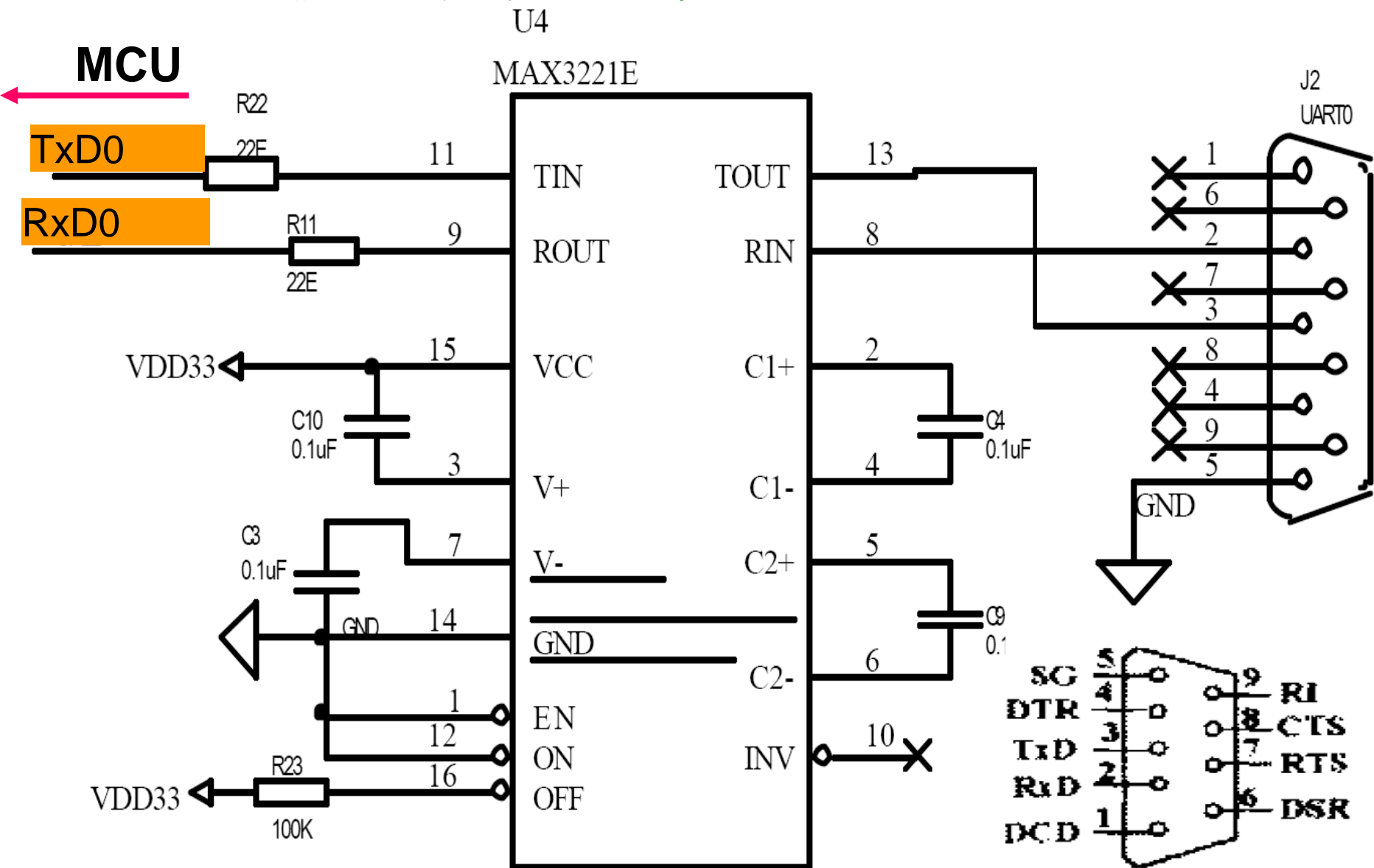


# RS-232接口硬件设计

- MAX3222芯片可完成TTL $\leftrightarrow$ EIA双向电平转换



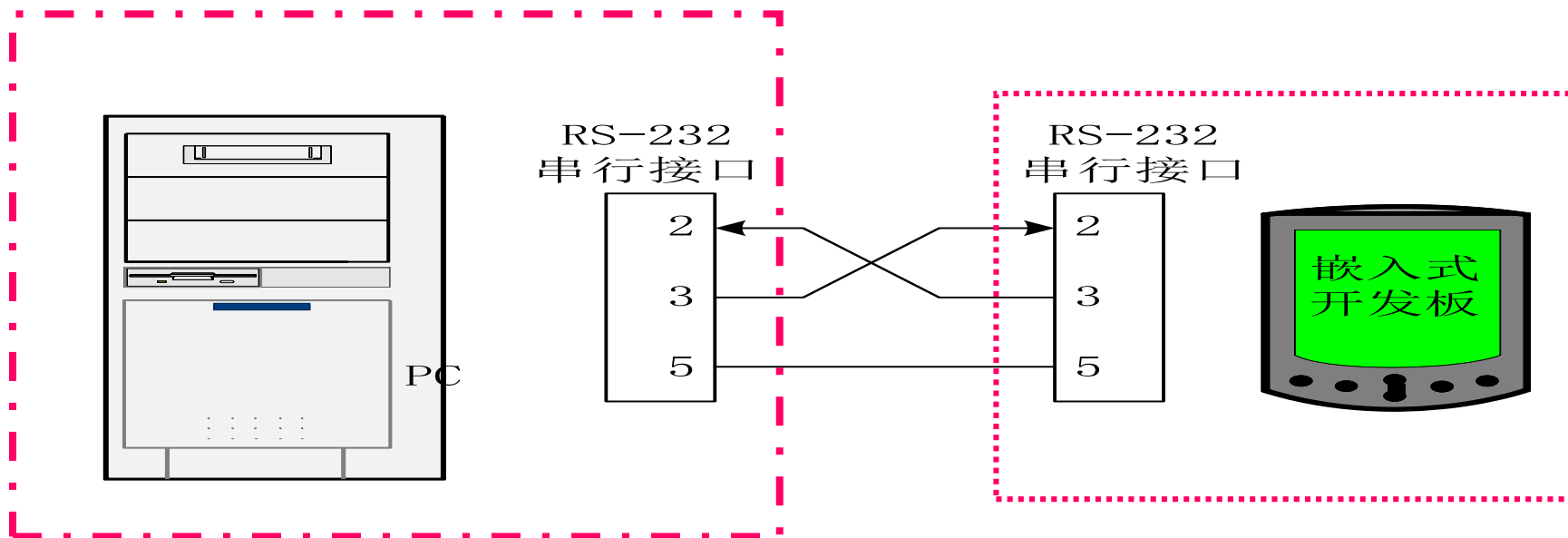
# RS232接口硬件设计





# RS232接口硬件设计

## 嵌入式开发板和PC机的通讯电缆连接方式



## 嵌入式开发板与PC机的串行通讯

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器

# 13 SPI接口

---

## 主要内容

概述

结构

寄存器

**重点：**

- (1) SPI的接口方式**
- (2) SPI与UART的区别**

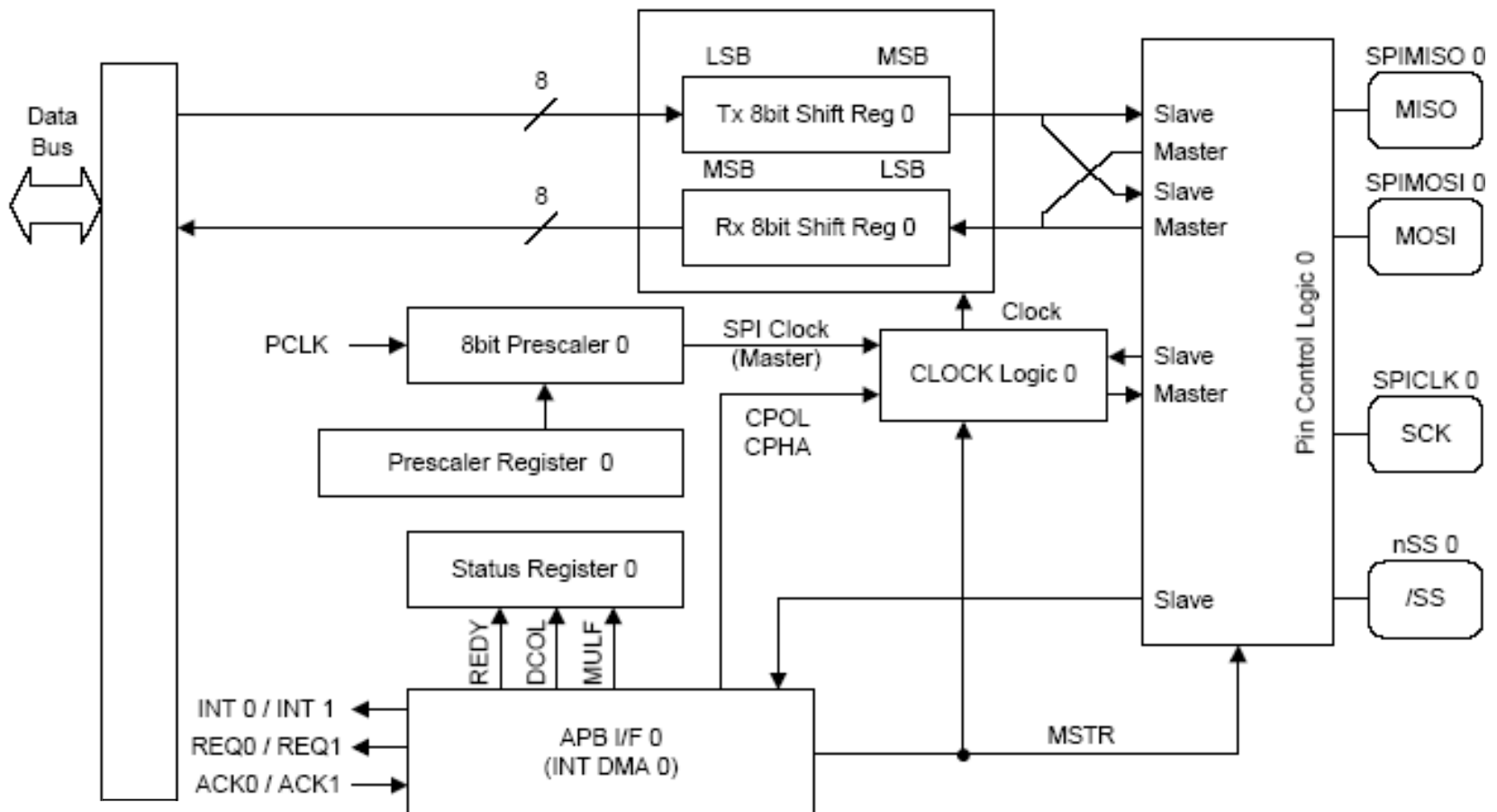
# 一、SPI接口概述

- **SPI接口**：SPI（Serial Peripheral Interface，串行外设接口）是Motorola公司开发的一种同步串行外设接口标准
- **SPI接口信号**：有4个信号：数据发送、数据接收、时钟和片选。在时钟信号的作用下，发送的同时，接收对方发来的数据；也可以只发送、或者只接收。SPI的波特率可以达到20Mb/s以上
- S3C2410有2个SPI接口，既可以作为主SPI使用，也可以作为从SPI使用
- **主要特点**：
  - 与SPI 协议 (ver. 2.11) 兼容；
  - 8位发送移位寄存器；
  - 8位接收移位寄存器；
  - 8位预分频器；
  - 具有查询、中断、DMA传输模式

## 二、S3C2410 SPI结构与工作原理

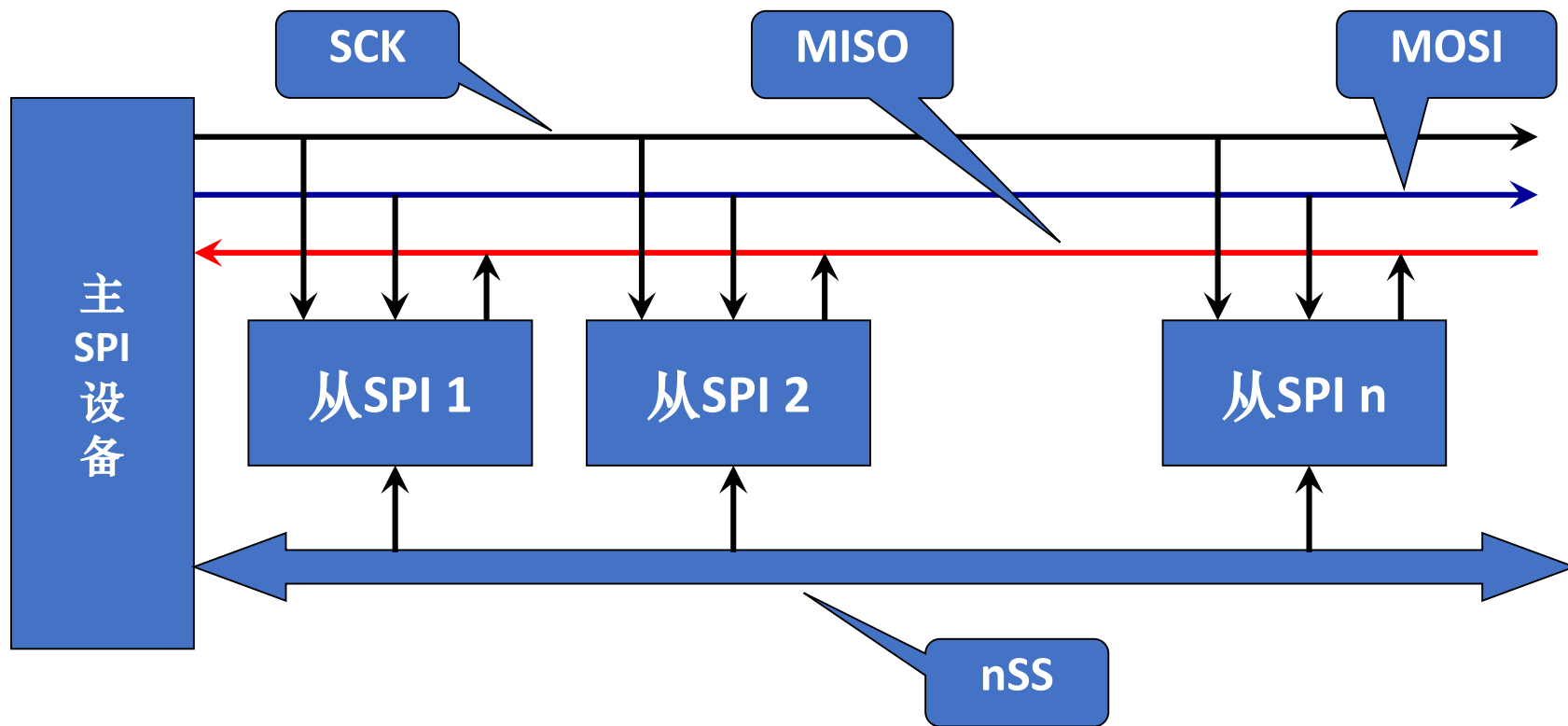
### 1、SPI结构

SPI主要由4部分构成：时钟分频器、发送移位寄存器、接收移位寄存器、控制逻辑等。如下图所示



## 2、SPI设备系统组成

系统可以多个SPI设备组成，任何一个设备都可以为主SPI，但是任一时刻只能有一个主SPI设备。如下图所示。



### 3、SPI工作时序

Format	CPOL和CPHA	第一位数据输出	其它位数据输出	数据采样
A	CPOL=0,CPHA=0	第1个SCK上升沿前	SCK下降沿	SCK上升沿
B	CPOL=0,CPHA=1	第1个SCK上升沿	SCK上升沿	SCK下降沿
A	CPOL=1,CPHA=0	第1个SCK下降沿前	SCK上升沿	SCK下降沿
B	CPOL=1,CPHA=1	第1个SCK下降沿	SCK下降沿	SCK上升沿

SPI Mode	Timing Diagram
<b>Mode 0</b> Positive Pulse. Latch, then Shift. (CPHA=0, CPOL=0)	
<b>Mode 1</b> Positive Pulse. Shift, then Latch.	
<b>Mode 2</b> Negative Pulse. Latch, then Shift.	
<b>Mode 3</b> Negative Pulse. Shift, then Latch.	

## 三、SPI专用寄存器

2个UART，每个都有6个专用寄存器，共12个寄存器

Register	Address	R/W	Description	Reset Value
SPCON <sub>n</sub>	0x590000x0	R/W	SPI控制寄存器	0x00
SPSTAN <sub>n</sub>	0x590000x4	R	SPI状态寄存器	0x01
SPPIN <sub>n</sub>	0x590000x8	R/W	SPI引脚控制寄存器	0x02
SPPREN <sub>n</sub>	0x590000xC	R/W	SPI预分频寄存器	0x00
SPTDAT <sub>n</sub>	0x590000x0	R/W	SPI数据发送寄存器	0x00
SPRDAT <sub>n</sub>	0x590000x4	R	SPI数据接收寄存器	0x00



## 1、SPI控制寄存器 (SPCON)

Register	Address	R/W	Description	Reset Value
SPCON0	0x59000000	R/W	SPI0 控制寄存器	0x00
SPCON1	0x59000020	R/W	SPI1 控制寄存器	0x00

字段名	位	意 义	初值
SMOD	6:5	SPI模式选择。00：查询模式； 01：中断； 10：DMA模式； 11：保留	00
ENSCK	4	时钟SCK控制。 0：禁止； 1：允许。	0
MSTR	3	SPI主、从选择。 0：从SPI； 1：主SPI	0
CPOL	2	时钟极性选择。 0：高电平有效； 1：低	0
CPHA	1	时钟相位选择。 0：格式A； 1：格式B	0
Tx Auto Garbage Data Mode Enable (TAGD)	0	仅接收模式控制。 0：正常收发； 1：仅接收（此时(自动)发送任意数据）	0

## 2、SPI状态寄存器（SPSTA）

Register	Address	R/W	Description	Reset Value
SPSTA0	0x59000004	R	SPI0 状态寄存器	0x01
SPSTA1	0x59000024	R	SPI1 状态寄存器	0x01

字段名	位	意 义	初值
reserved	7:3	保 留	
Data Collision Error Flag (DCOL)	2	数据写碰撞(正在发送时写SPTDAT)错误标志。 0：无错； 1：碰撞错误	0
Multi Master Error Flag (MULF)	1	多主SPI错误标志。 0：无错； 1：多主SPI错误。	0
Transfer Ready Flag (REDY)	0	收发就绪标志。 0：未就绪； 1：收或发就绪。 写SPTDAT后该位自动清0	1

### 3、SPI引脚控制寄存器 (SPPIN)

Register	Address	R/W	Description	Reset Value
SPPIN0	0x59000008	R/W	SPI0 引脚控制寄存器	0x02
SPPIN1	0x59000028	R/W	SPI1 引脚控制寄存器	0x02

字段名	位	意 义	初值
reserved	7:3	保 留	
Multi Master error detect Enable (ENMUL)	2	引脚多主SPI错误测试设置。 0：禁测；1：允许多主错误测试。 测试结果在SPSTAn中的MULF位	0
reserved	1	该位应该为1	1
Master Out Keep (KEEP)	0	1字节发完后MOSI的控制与释放 0：释放；1：保持MOSI原电平	0

## 4、SPI预分频寄存器（SPPRE）

Register	Address	R/W	Description	Reset Value
SPPRE0	0x5900000C	R/W	SPI0 预分频寄存器	0x00
SPPRE1	0x5900002C	R/W	SPI1 预分频寄存器	0x00

字段名	位	意 义	初值
Prescaler value	7:0	预分频值	

波特率计算：

$$\text{Baud rate} = \text{Pclk} / 2 / (\text{Prescaler value} + 1)$$

提示：SPI的波特率必须<25MHz。

## 5、SPI发送数据寄存器（SPTDAT）

Register	Address	R/W	Description	Reset Value
SPTDAT0	0x59000010	R/W	SPI0 发送数据寄存器	0x00
SPTDAT1	0x59000030	R/W	SPI1 发送数据寄存器	0x00

字段名	位	意 义	初值
Tx data	7:0	SPI所发送的8位数据	0x00

**提示：**在同时双向传输时，从SPI设备必须先把所发送的数据写到“发送数据寄存器”SPTDATn中，然后再写主SPI设备的“发送数据寄存器”SPTDATn。

## 6、SPI接收数据寄存器 (SPRDAT)

Register	Address	R/W	Description	Reset Value
SPRDAT0	0x59000014	R	SPI0 接收数据寄存器	0x00
SPRDAT1	0x59000034	R	SPI1 接收数据寄存器	0x00

字段名	位	意 义	初值
Tx data	7:0	SPI所接收的8位数据	0x00

**提示：**在同时双向传输时，启动发送后，应该先读取数据，然后再发送数据。

# 四、SPI应用方法

## 1、一般操作步骤

如果SPI控制寄存器SPCON已经设置过，则写数据发送寄存器SPTDAT启动发送。对SPI操作步骤如下：

- (1) 设置预分频寄存器SPPRE；
- (2) 设置控制寄存器SPCON；
- (3) 设置一个GPIO引脚，使选中的MMC或SD卡的片选信号nSS有效；
- (4) 向数据发送寄存器SPTDAT写10次0xFF，对MMC或SD卡初始化；
- (5) **发送数据：**先要查询Tx/Rx REDY是否为1，然后向数据发送寄存器SPTDAT写数据；

## (6) 接收数据:

**一般方式**（同时收发，TAGD=0）：向数据发送寄存器 SPTDAT写0xFF，查询并确认Rx REDY为1，然后从数据接收寄存器中读取数据。

**仅接收方式**（TAGD=1）：并确认Rx REDY为1，然后从数据接收寄存器中读取数据。读取数据的同时启动一次发送。

(7) 设置GPIO引脚，使选中的MMC或SD卡的片选信号nSS无效，结束传输。



## 2、DMA模式发送

对SPI的DMA模式发送操作步骤如下：

- (1) 设置预分频寄存器SPPRE；
- (2) 设置控制寄存器SPCON，并且设为DMA模式；
- (3) SPI请求DMA服务；
- (4) DMA发送1字节数据给SPI；
- (5) SPI给卡发送数据；
- (6) 返回到（3），直到DMA的计数器为0；

### 3、DMA模式接收

对SPI的DMA模式接收操作步骤如下：

- (1) 设置预分频寄存器SPPRE；
- (2) 设置控制寄存器SPCON，并且设为DMA模式；
- (3) 将SPI设置为仅接收模式，设置引脚控制寄存器SPPIN中的TAGD为1；
- (4) 当SPI从卡接收到1字节数据后，请求DMA服务；
- (5) DMA从SPI读取数据；
- (6) SPI自动启动一次无用数据(0xFF)的发送；
- (7) 返回到（4），直到DMA的计数器为0；
- (8) 设置控制寄存器SPCON，改设为其它模式，如查询、中断模式；
- (9) 如果Rx REDY被设置为1，则需要读取最后接收的数据。

注意：DMA第1次接收的是数据的无效的。

# 第6章 嵌入式系统硬件

---

## 主要内容

1. S3C2410概述
2. 电源电路模块
3. 复位电路模块
4. JATG接口模块
5. 时钟与电源管理
6. S3C2410X的存储器
7. DMA控制器
8. A/D转换与触摸屏
9. 中断控制器
10. 输入/输出端口
11. 定时器、PWM
12. UART通用异步串行接口
13. SPI串行总线接口
14. I2C(IIC)串行总线接口
15. 实时钟RTC
16. USB接口
17. 看门狗
18. LCD控制器

# 14 I<sup>2</sup>C(IIC)串行总线接口

---

## 主要内容

- 1、概述
- 2、结构与工作原理
- 3、寄存器
- 4、应用方法

**重点：**

工作原理时序；控制寄存器；应用

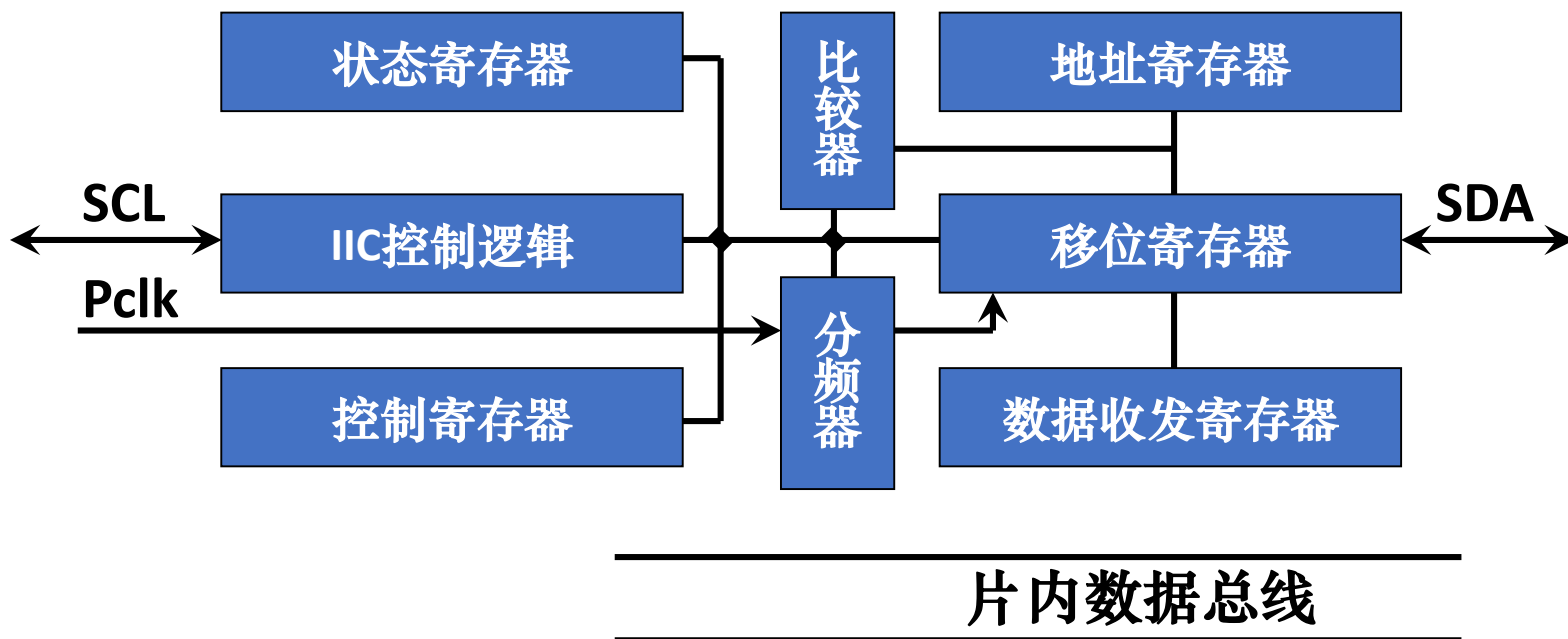
# 一、概述

- **IIC总线的产生和应用**：IIC总线是PHILIPS公司开发的一种串行总线。IIC总线应用越来越广泛，现在在很多器件上都配置有IIC总线接口，如EEPROM、时钟芯片等。
- **IIC总线信号**：为两线，一个能够双向传输的数据线SDA、另一个能够双向传输的时钟线SCL。**是信号线最少的串行总线。**
- **S3C2410的IIC总线的特点**：
  - (1) 有一个IIC总线接口
  - (2) IIC总线的速度：可以标准速度传输（100kb/s），也可以高速传输（高达400kb/s）
  - (3) 可以查询方式和中断方式工作
  - (4) 可以主设备身份传输，也可以从设备身份传输，因此共有4种操作模式：
    - 主机发送模式、主机接收模式
    - 从机发送模式、从机接收模式

## 二、S3C2410的IIC结构与工作原理

### 1、S3C2410的IIC结构

S3C2410的IIC主要有5部分构成：数据收发寄存器、数据移位寄存器、地址寄存器、时钟发生器、控制逻辑等部分。如下图所示。

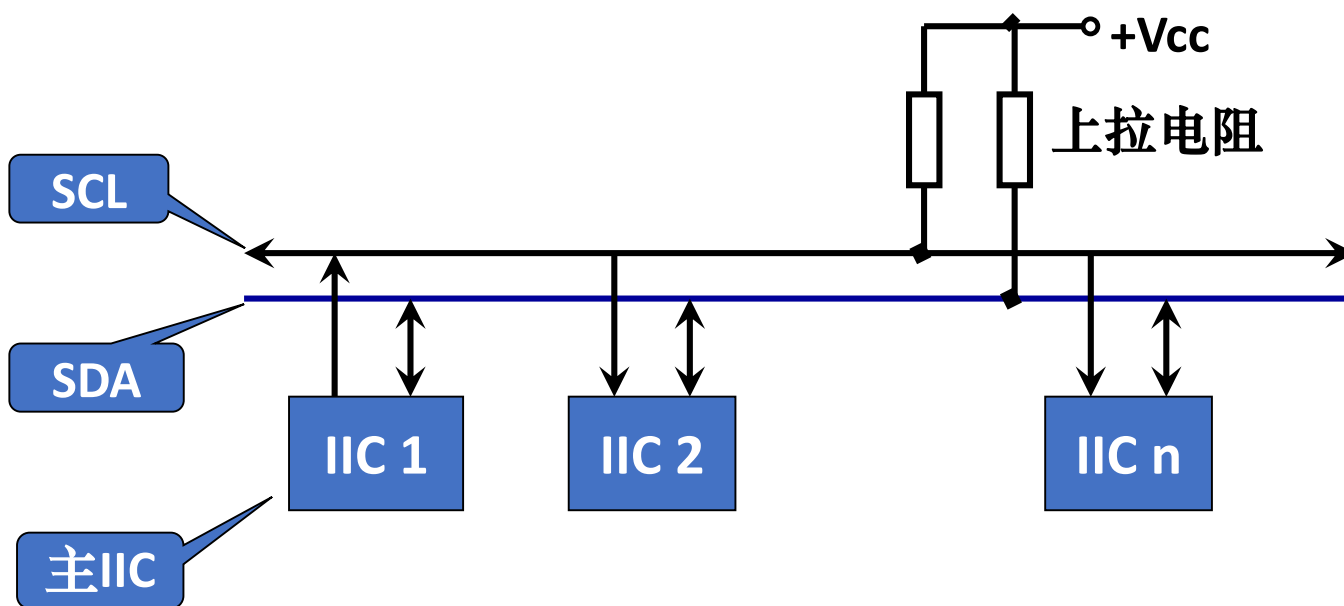


## 2、IIC总线系统组成

- IIC总线是多主系统：系统可以有多个IIC节点，任何一个设备都可以为主IIC；但任一时刻只能有一个主IIC设备，IIC具有总线仲裁功能，保证系统正常运行
- 主IIC设备发出时钟信号、地址信号和控制信号，选择通信的从IIC设备和控制收发

# IIC总线系统组成

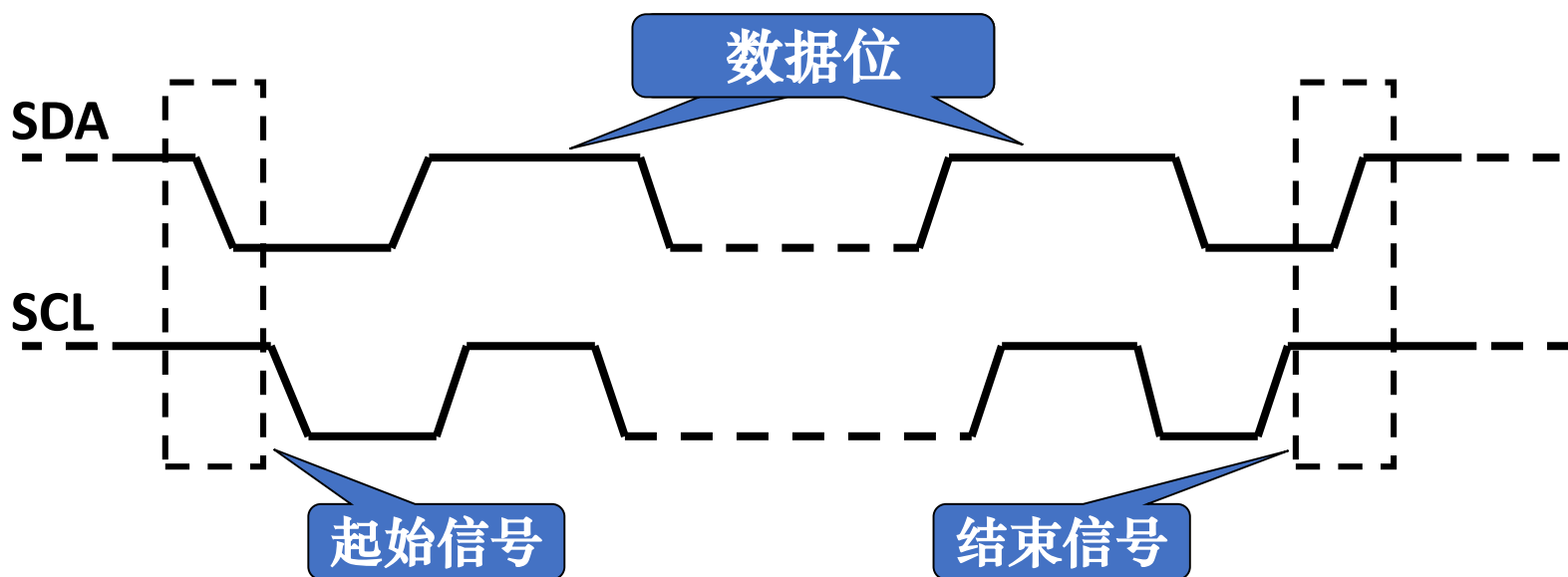
- 系统要求：（1）各个节点设备必须具有IIC接口功能；  
（2）各个节点设备必须共地；（3）两个信号线必须接上拉电阻。如下图所示





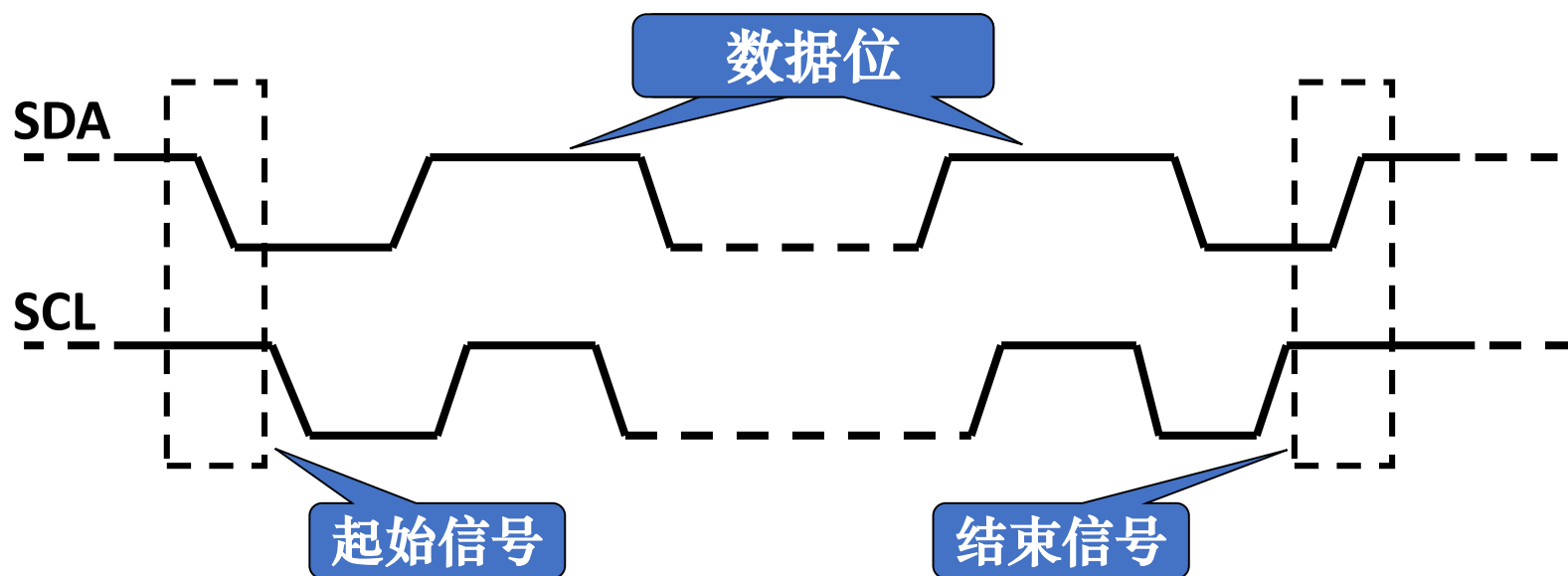
### 3、IIC总线的工作原理

- I2C是**两线**，SDA（数据）和SCL（时钟），也是一种**同步传输协议**。主机在发送开始信号之后，先发送7个比特的地址位和1个比特的读写位，每个从机有自己的I2C地址，当发现该条指令是发给自己的时候，拉低SDA线（即回复ACK信号），然后主机发送或接收数据，完成传输。传输完成之后，主机发送停止位，完成该次传输。



## (1) IIC总线对数据线上信号的定义

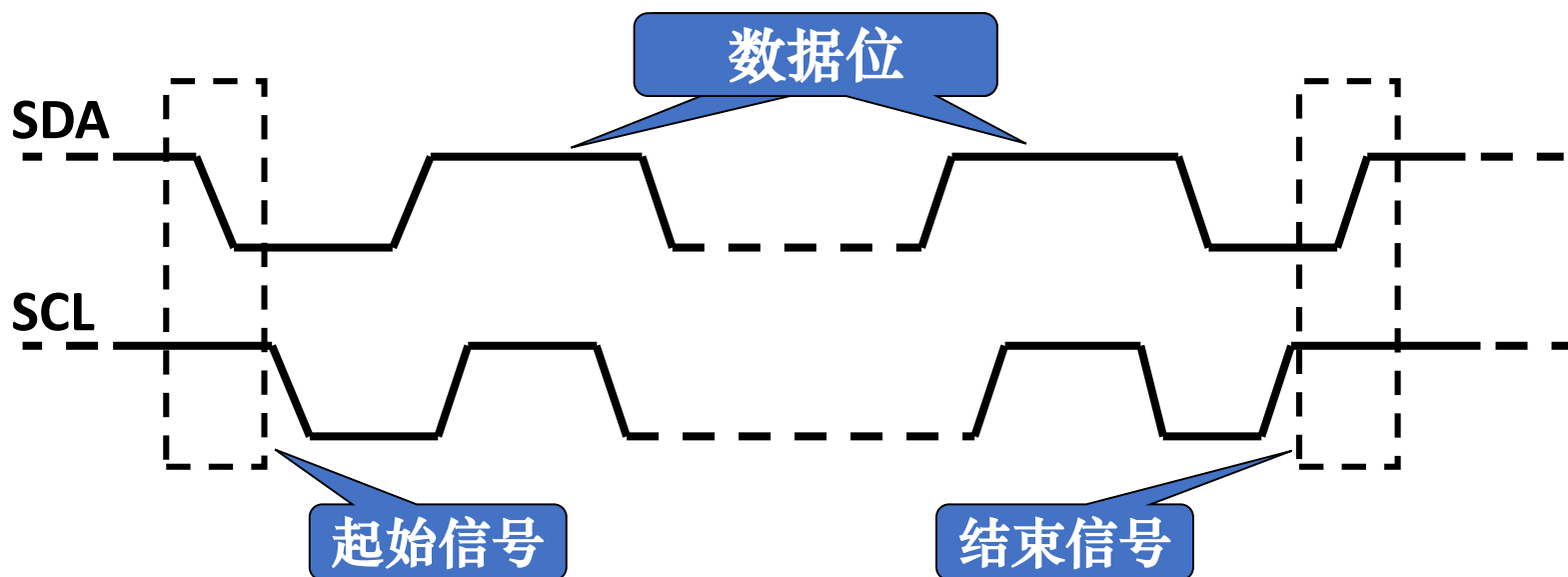
- 1) **总线空闲状态**：时钟信号线和数据信号线均为高电平
- 2) **起始信号**：即启动一次传输，时钟信号线是高电平时，数据信号线由高变低
- 3) **停止信号**：即结束一次传输，时钟信号线是高电平时，数据信号线由低变高



4) **数据位信号**：时钟信号线是低电平时，可以改变数据信号线电位；时钟信号线是高电平时，应保持数据信号线上电位不变，即时钟是高电平时数据有效。

5) **应答信号**：占1位，数据接收者接收1字节数据后，应向数据发出者发送一应答信号。低电平为应答，继续发送；高电平为非应答，结束发送。

6) **控制位信号**：占1位，主IIC设备发出的读写控制信号，高为读、低为写（对主IIC设备而言）。控制位在寻址字节中



**7) 地址信号：**为从机地址，占7位，如下表所示，称之为“寻址字节”，各字段含义如下：

D7	D6	D5	D4	D3	D2	D1	D0
DA3	DA2	DA1	DA0	A2	A1	A0	R/ $\overline{W}$

**器件地址 (DA3---DA0)：**是IIC总线接口器件固有的地址编码，由器件生产厂家给定。如IIC总线EEPROM AT24C××的器件地址为1010等。

**引脚地址 (A2、A1、A0)：**由IIC总线接口器件的地址引脚A2、A1、A0的高低来确定，接电源者为1，接地者为0。

**读写控制位 (R/ W)：**1表示主设备读，0表示主设备写。

7位地址和读写控制位组成1个字节。

## (2) IIC总线数据传输格式

### 1) 一般格式:

S	从IIC地址(7位)	R/W	A	传输数据 .....	A	P
---	------------	-----	---	------------	---	---

### 2) 主控制器写操作格式:

S	从IIC地址 $\overline{W}$	A	数据1	A	数据2	A	.....	数据n	A/ $\overline{A}$	P
---	-----------------------	---	-----	---	-----	---	-------	-----	-------------------	---

红色起始信号S、地址信号、控制信号W、各个数据、结束信号P，均为主IIC设备发送、从IIC设备接收；黑色的应答信号A/A为从IIC设备发送、主IIC设备接收。

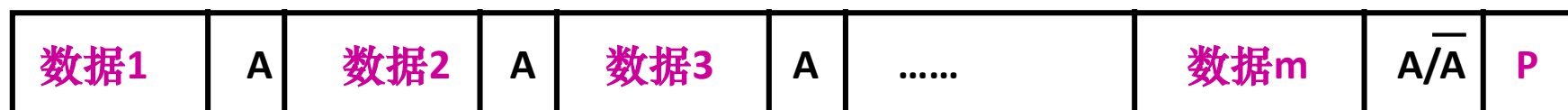
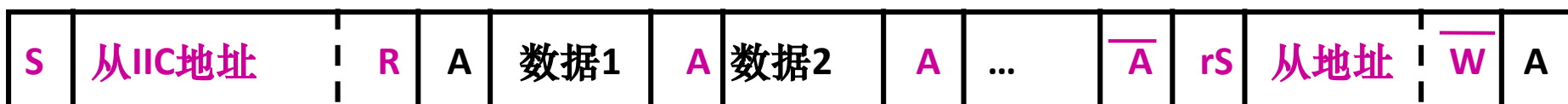
### 3) 主控制器读操作格式:

S	从IIC地址 R	A	数据1	A	数据2	A	.....	数据n	$\overline{A}$	P
---	----------	---	-----	---	-----	---	-------	-----	----------------	---

红色的信号均为主IIC设备发送、从IIC设备接收；黑色的信号均为从IIC设备发送、主IIC设备接收。

#### 4) 主控制器读/写操作格式:

- 由于在一次传输过程中要改变数据的传输方向，因此**起始信号和寻址字节都要重复一次**，而中间可以不要结束信号。
- 在一次传输中，可以有**多次启动信号**。



### (3) 读写操作

- 在发送器模式下，数据被发送之后，IIC 总线接口会等待直到 IICDS（IIC 数据移位寄存器）被程序写入新的数据。在新的数据被写入之前，SCL 线都被拉低。新的数据写入之后，SCL 线被释放
- S3C2410X 可以利用中断来判断当前数据字节是否已经完全送出。在 CPU 接收到中断请求后，在中断处理中再次将下一个新的数据写入 IICDS，如此循环
- 在接收模式下，数据被接收到后，IIC 总线接口将等待直到 IICDS 寄存器被程序读出。在数据被读出之前，SCL 线保持低电平。新的数据从读出之后，SCL 线才释放
- S3C2410X 也利用中断来判别是否接收到了新的数据。CPU 收到中断请求之后，处理程序将从 IICDS 读取数据

## (4) 总线仲裁

- 总线仲裁发生在两个主IIC设备中。如果一个主设备欲使用总线，而测得SDA为低电平，则该主设备仲裁不能够使用总线启动传输。这个仲裁过程会延长，直到信号线SDA变为高电平
- 每次操作都要进行仲裁



### 三、IIC专用寄存器

S3C2410有4个专用寄存器

Register	Address	R/W	Description	Reset Value
IICCON	0x54000000	R/W	IIC总线控制寄存器	0x0X
IICSTAT	0x54000004	R/W	IIC总线控制/状态寄存器	0x0
IICADD	0x54000008	R/W	IIC总线地址寄存器	0xXX
IICDS	0x5400000C	R/W	IIC数据发送/接收寄存器	0xXX

# 1、IIC控制寄存器 (IICCON)

字段名	位	意 义	初值
Acknowledge generation	7	应答使能。0：禁止应答；1：自动应答 应答电平：Tx时为高；Rx时为低	00
Tx clock source selection	6	发送时钟分频选择。 0: $IICCLK = fPCLK / 16$ ; 1: $IICCLK = fPCLK / 512$	0
Tx/Rx Interrupt	5	收发中断控制位。0：禁止；1：允许	0
Interrupt Pending flag	4	中断标志位。读：0无，1示有中断请求 写：写0清除中断标志，写1不操作	0
Transmit clock value	3:0	发送时钟预分频值。 $Tx\ clock = IICCLK / (IICCON[3:0] + 1)$	0

说明：

- (1) 应答使能问题：一般情况下为使能；在对EEPROM读最后1个数据前可以禁止应答，便于产生结束信号
- (2) 中断事件：1) 完成收发；2) 地址匹配；3) 总线仲裁失败
- (3) 中断控制位问题：设为0时，中断标志位不能正确操作，故总设为1
- (4) 时钟预分频问题：当分频位选择为0时，预分频值必须大于1

## 2、IIC控制状态寄存器（IICSTAT）

字段名	位	意 义	初值
Mode selection	7:6	工作模式选择。00：从收；01：从发 10：主收；11：主发	00
Busy / START STOP condition	5	忙状态/启、停控制。读：1示忙；0示闲 写：0产生结束信号，1产生启动信号	0
Serial output	4	数据发送控制。0：禁止；1：允许发送	0
Arbitration Status flag	3	仲裁状态标志。0：仲裁成功； 1：仲裁失败（因为在连续I/O中）	0
Address-as-slave status flag	2	从地址匹配状态。0：与IICADD不匹配 1：匹配。在收到SART/STOP时清0	0
Address zero status flag	1	地址状态标志。0：收到的为非0地址 1：收到0地址。在收到SART/STOP时清0	0
Last-received bit status flag	0	最后收到位状态。0：最后位为0，收到 ACK；1：最后位为1，未收到ACK。	0

## 2、IIC控制状态寄存器（续）

字段名	位	意 义	初值
Mode selection	7:6	工作模式选择。00：从收；01：从发 10：主收；11：主发	00
Busy / START STOP condition	5	忙状态/启、停控制。读：1示忙；0示闲 写：0产生结束信号，1产生启动信号	0
Serial output	4	数据发送控制。0：禁止；1：允许发送	0

IICSTAT控制字：

启动主设备发送：0xF0；结束主设备发送：0xD0

启动主设备接收：0xB0；结束主设备接收：0x90

### 3、IIC地址寄存器 (IICADD)

Register	Address	R/W	Description	Reset Value
IICADD	0x54000008	R/W	地址寄存器	0xXX

字段名	位	意 义	初值
Slave address	7:1	7位从地址。	0xXX
Not mapped	0	不用	-

#### 说明：

- (1) 对从设备，该地址有意义，对主设备其值无意义。
- (2) 只有在不发送数据时（数据传输控制位IICSTAT[4]=0）才能对其写；任何时间都可以读。

## 4、IIC数据发送/接收寄存器 (IICDS)

Register	Address	R/W	Description	Reset Value
IICADD	0x5400000C	R/W	数据发送/接收移位寄存器	0xXX

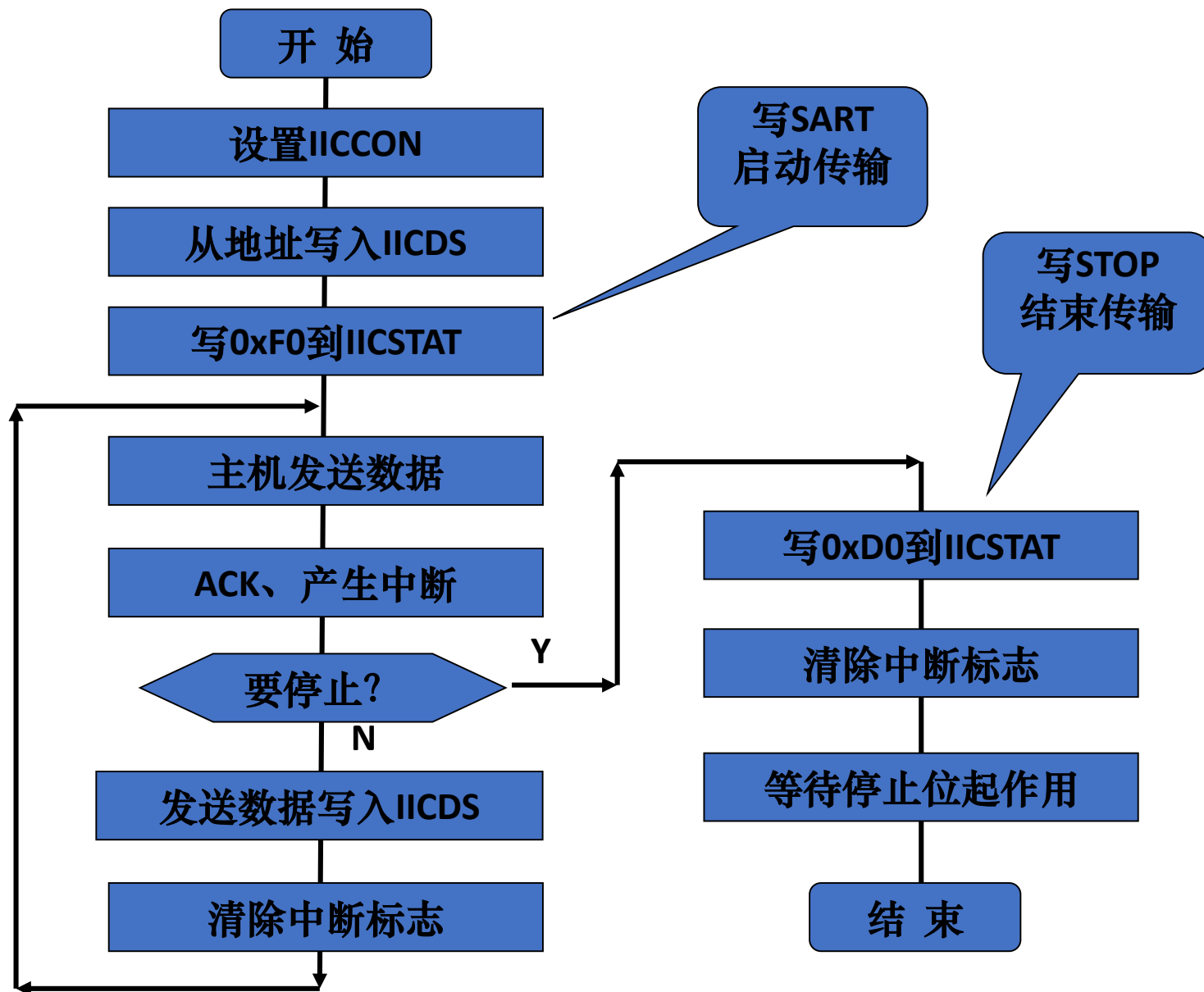
字段名	位	意 义	初值
Data shift	7:0	8位移位接收或移位发送的数据	0xXX

### 说明：

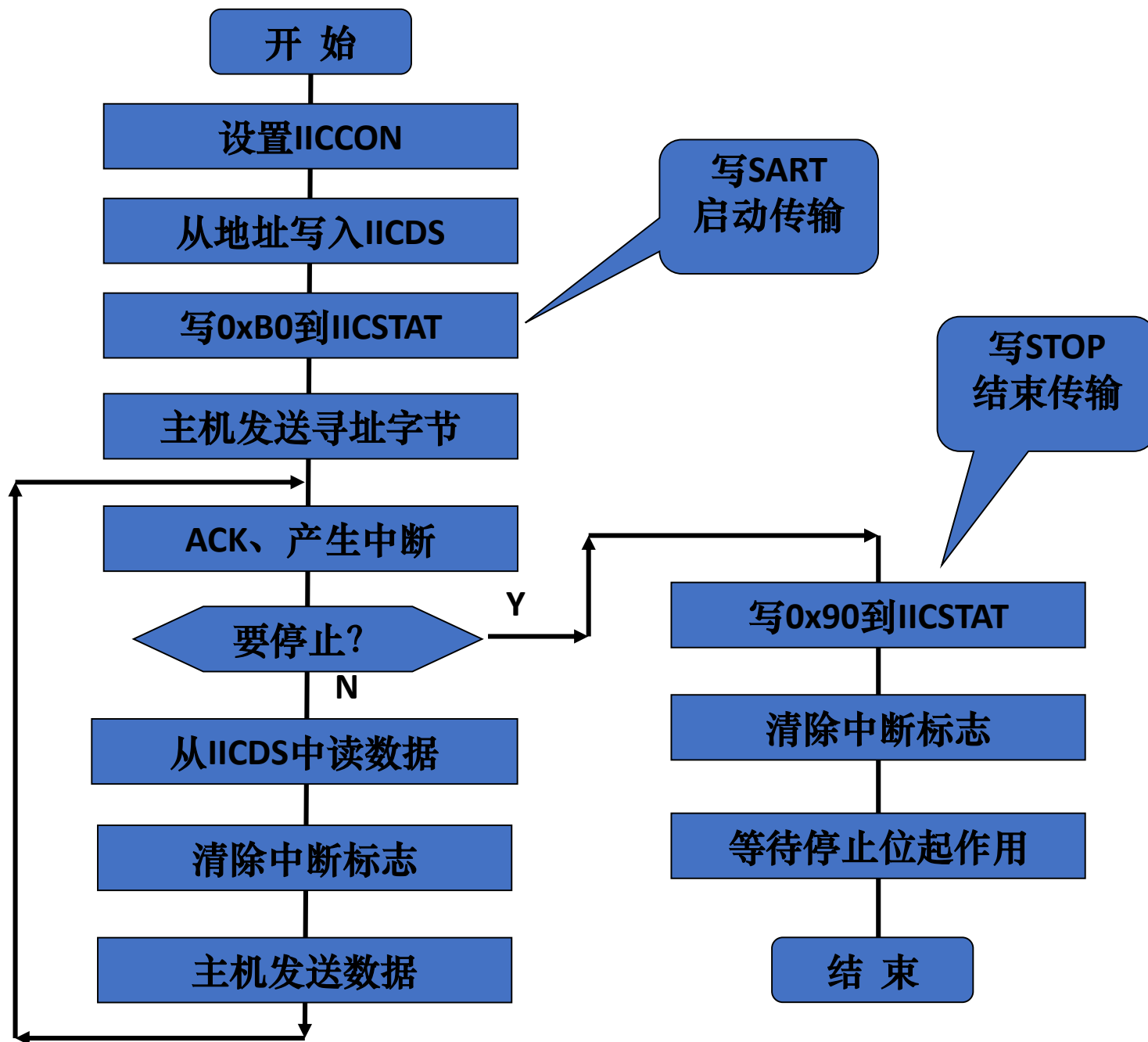
- (1) 在本设备接收时，对其作读操作得到对方发来的数据。任何时间都可以读。
- (2) 在本设备发送时，对其写操作，将数据发向对方。
- (3) 欲发送数据，必须使数据传输控制位IICSTAT[4] = 1才能对其写。

## 四、IIC操作方法

### 1、主模式发送流程

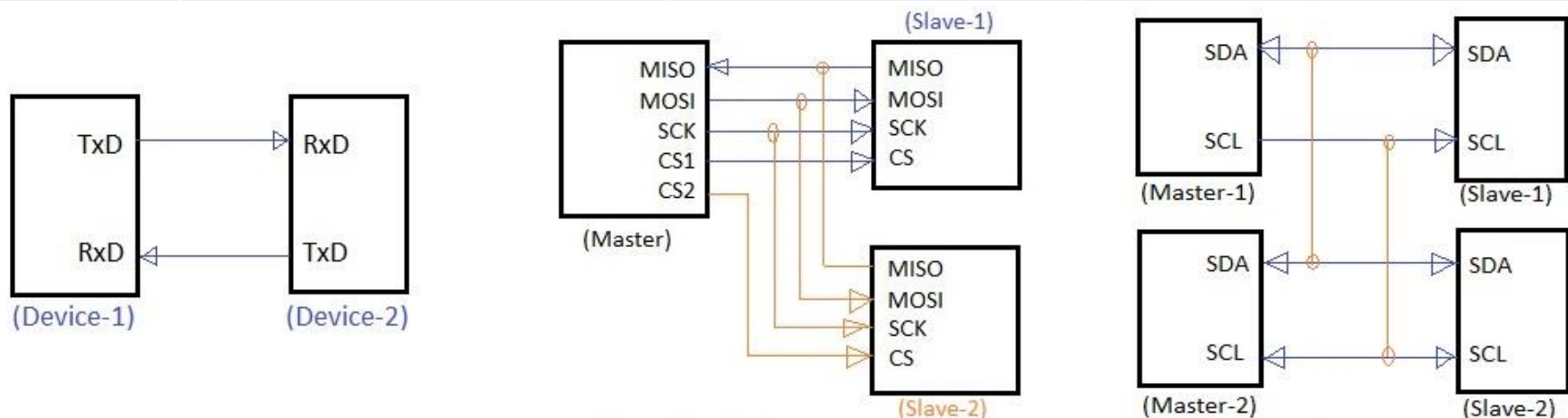


## 2、主模式接收流程





特性	UART	SPI	I2C
名称	Universal Asynchronous Receiver Transmitter 通用异步收发器	Serial Peripheral Interface 串行外设接口	Inter-Integrated Circuit
接口	TxD、RxD	SCLK、MOSI、MISO、CS	SDA、SCL
通信方式	异步、全双工	同步、全双工	同步、半双工
数据率	低：<115200bps, 提前商定	高：10Mbps-20Mbps	中：100Kbps, 400Kbps, 3.4Mbps
主设备数量	不适用、对等总线	唯一（非对等主从总线）	一个或多个（非对等主从总线）
从设备选择	不适用、对等总线	通过片选线来区分	通过地址来区分
硬件复杂度	低	较高	低
协议复杂度	低	较低	较高
特性	常用于处理器与其他外设进行通信；外接不同的电平转换IC可以组建RS232 485等通信接口	一般用于同一板卡上芯片之间的通信，较少用于远距离通信	一般用于同一板卡上芯片间通信，较少用于远距离通信；开漏输出，必须外接上拉电阻





Thank you