



暨南大学
JINAN UNIVERSITY

第6章 嵌入式系统硬件 (4)

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

15 实时钟RTC

主要内容

概述

结构

寄存器

应用举例

重点：

(1) RTC原理

(2) RTC应用

一、概述

- **RTC（实时时钟）**：提供可靠的系统时间，包括时、分、秒、和年、月、日等；系统关机状态下也能正常工作（**后备电池供电**），外围也不需太多辅助电路，典型的只需一个高精度32.768KHz 晶振和电阻电容等
- **RTC应用**：现在很多电子产品都有RTC功能，如电子日历（台式、壁式等）、手持数码产品（手机、电子词典、各种学习机、照相机、摄像机等）、电子计量仪表（电表、燃气表、水表等）、家用电器（电视机、机顶盒、DVD等）等，应用非常广泛

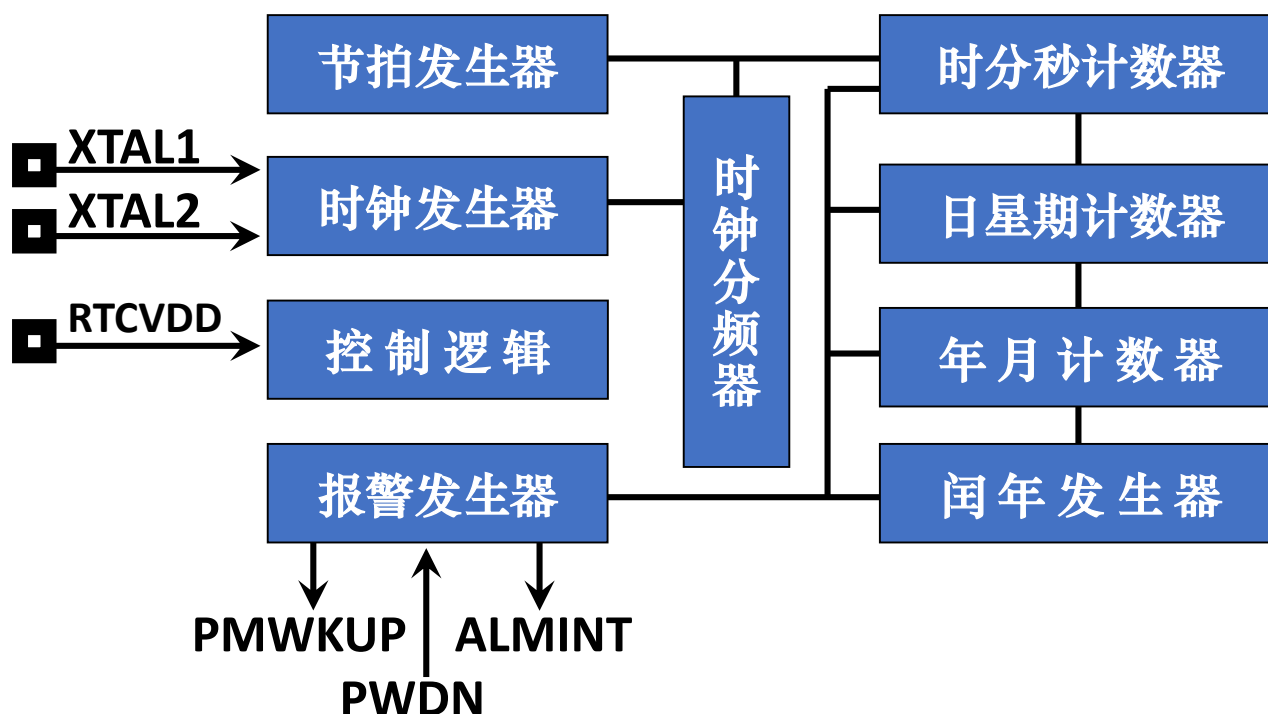
S3C2410 RTC 特点

- **RTC芯片**：独立的RTC芯片，如壁式电子日历、电子计量仪表等；RTC功能集成到其它的芯片中，如手持数码产品等
- **S3C2410的RTC的特点**
 - 时钟数据采用BCD 编码
 - 时钟数据有：秒、分、时、日、月、年、星期
 - 能够对闰年的年月日进行自动处理
 - 具有告警功能，当系统处于关机状态时，能产生告警中断
 - 具有独立的电源输入
 - 提供毫秒级时钟中断，可用于嵌入式操作系统的内核时钟

二、S3C2410的RTC结构

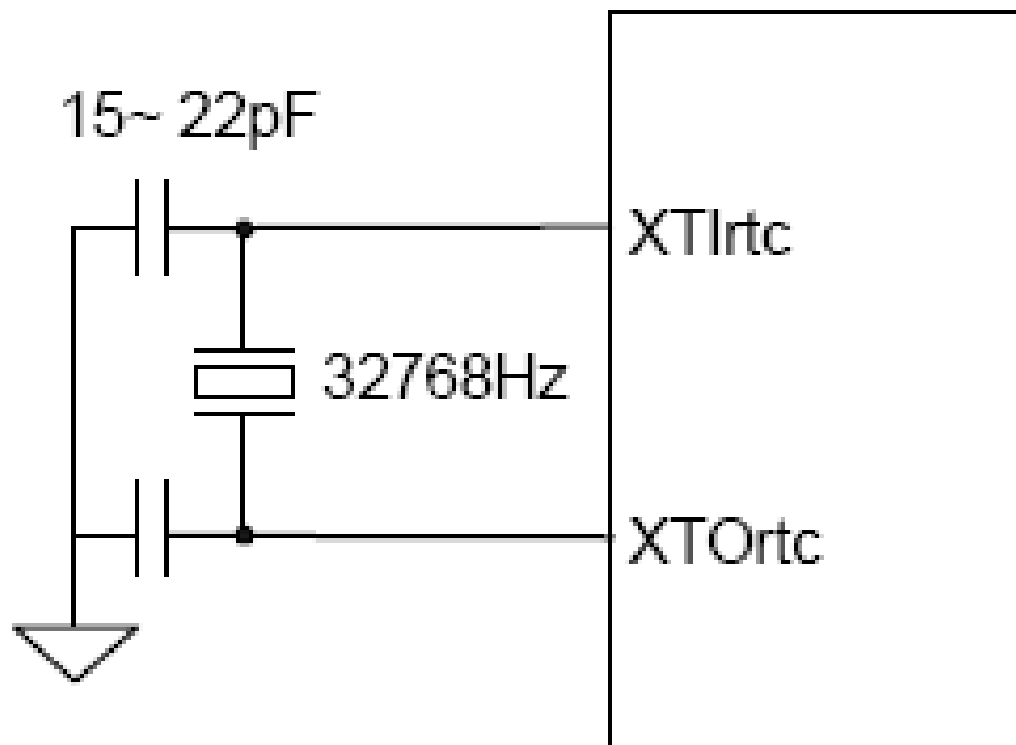
1、S3C2410的RTC结构

- 5部分构成：时钟发生器、节拍发生器、时间与日期计数器（时分秒年月日星期）、报警发生器、控制逻辑等



2、S3C2410 RTC的振荡电路

- S3C2410 RTC的只需外接2个20P左右的小电容、32.768KHz的晶振即可



Why 32.768KHz?

三、RTC专用寄存器

- S3C2410的RTC有17个专用寄存器，均需用字节读写。下表为前10个，有4个为控制寄存器，6个为报警寄存器

Register	Address	R/W	Description	Reset Value
RTCCON	0x57000040/3	R/W	RTC控制寄存器	0x0
TICNT	0x57000044/7	R/W	RTC节拍计数器	0x00
RTCALM	0x57000050/3	R/W	RTC报警控制寄存器	0x00
RTCRST	0x5700006C/F	R/W	RTC循环复位寄存器	0x0
ALMSEC	0x57000054/7	R/W	报警秒数寄存器	0x00
ALMMIN	0x57000058/B	R/W	报警分钟数寄存器	0x00
ALMHOUR	0x5700005C/F	R/W	报警小时数寄存器	0x00
ALMDAY	0x57000060/3	R/W	报警天(日)数寄存器	0x01
ALMMON	0x57000064/7	R/W	报警月数寄存器	0x01
ALMYEAR	0x57000068/B	R/W	报警年数寄存器	0x00

三、RTC专用寄存器

- S3C2410的RTC有17个专用寄存器，均需用字节读写。下表为4个控制寄存器

Register	Address	R/W	Description	Reset Value
RTCCON	0x57000040/3	R/W	RTC控制寄存器	0x0
TICNT	0x57000044/7	R/W	RTC节拍计数器	0x00
RTCALM	0x57000050 /3	R/W	RTC报警 控制寄存器	0x00
RTCRST	0x5700006C /F	R/W	RTC秒循环 复位寄存器	0x0

本表6个寄存器：为报警日期、时间寄存器

Register	Address	R/W	Description	Reset Value
ALMSEC	0x57000054/7	R/W	报警秒数寄存器	0x00
ALMMIN	0x57000058/B	R/W	报警分钟数寄存器	0x00
ALMHOUR	0x5700005C/F	R/W	报警小时数寄存器	0x00
ALMDAY	0x57000060/3	R/W	报警天(日)数寄存器	0x01
ALMMON	0x57000064/7	R/W	报警月数寄存器	0x01
ALMYEAR	0x57000068/B	R/W	报警年数寄存器	0x00

后7个寄存器：为日期、时间寄存器

Register	Address	R/W	Description	Reset Value
BCDSEC	0x57000070/3	R/W	秒当前值寄存器	0xXX
BCDMIN	0x57000074/7	R/W	分钟当前值寄存器	0xXX
BCD HOUR	0x57000078/B	R/W	小时当前值寄存器	0xXX
BCDDAY	0x5700007C/F	R/W	日当前值寄存器	0xXX
BCDDATE	0x57000080/3	R/W	星期当前值寄存器	0xXX
BCDMON	0x57000084/7	R/W	月当前值寄存器	0xXX
BCD YEAR	0x57000088/B	R/W	年当前值寄存器	0xXX

1、RTC控制寄存器 (RTCCON)

Register	Address	R/W	Description	Reset Value
RTCCON	0x57000040(L) 0x57000043(B)	R/W (字节)	RTC控制寄存器	0x0

字段名	位	意 义	初值
CLKRST	3	RTC时钟计数复位。 0：不复位；1 = BCD计数复位。	0
CNTSEL	2	BCD计数选择。0：合并BCD计数； 1 = 保留（单独的BCD计数器）	0
CLKSEL	1	BCD时钟选择。0：XTAL / 32768 1：用XTAL原值（但只用于测试）	0
RTCEN	0	RTC控制使能。0：失能；1：使能 指BCD时间计数和读取操作可以被执行	0

2、RTC节拍时间计数器（TICNT）

Register	Address	R/W	Description	Reset Value
TICNT	0x57000044(L) 0x57000047(B)	R/W (字节)	RTC节拍时间 计数器	0x00

字段名	位	意 义	初值
TICK INT ENABLE	7	节拍中断使能。 0：失能，1：使能。	0
TICK TIME COUNT	6:0	节拍时间计数值（1~127）。	000 0000

说明：这个计数器的值在内部减少，用户不能在工作时读取这个计数器的值。

3、RTC报警控制寄存器 (RTCCON)

Register	Address	R/W	Description	Reset Value
RTCALM	0x57000050(L) 0x57000053(B)	R/W (字节)	RTC报警 控制寄存器	0x0

字段名	位	意 义	初值
Reserved	7	保留 (为0)	0
ALMEN	6	报警总使能位。0: 失能; 1: 使能	0
YEAREN	5	年报警使能位。0: 失能; 1: 使能	0
MONEN	4	月报警使能位。0: 失能; 1: 使能	0
DATEEN	3	日报警使能位。0: 失能; 1: 使能	0
HOUREN	2	时报警使能位。0: 失能; 1: 使能	0
MINEN	1	分报警使能位。0: 失能; 1: 使能	0
SECEN	0	秒报警使能位。0: 失能; 1: 使能	0

4、RTC报警秒数寄存器 (ALMSEC)

Register	Address	R/W	Description	Reset Value
ALMSEC	0x57000054(L) 0x57000057(B)	R/W (字节)	报警秒数 寄存器	0x00

字段名	位	意 义	初值
Reserved	7	保留 (为0)	0
ALMSECH	6:4	报警时间秒十位, BCD值。0 ~ 5	000
ALMSECL	3:0	报警时间秒个位, BCD值。0 ~ 9	0000

5、报警时间分钟数寄存器 (ALMMIN)

Register	Address	R/W	Description	Reset Value
ALMMIN	0x57000058(L) 0x5700005B(B)	R/W (字节)	报警分钟数 寄存器	0x00

字段名	位	意 义	初值
Reserved	7	保留 (为0)	0
ALMMINH	6:4	报警时间分钟十位, BCD值。0 ~ 5	000
ALMMINL	3:0	报警时间分钟个位, BCD值。0 ~ 9	0000

6、报警时间小时数寄存器 (ALMHOUR)

Register	Address	R/W	Description	Reset Value
ALMHOUR	0x5700005C(L) 0x5700005F(B)	R/W (字节)	报警小时 寄存器	0x00

字段名	位	意 义	初值
Reserved	7:6	保留 (为0)	00
ALMHOURH	5:4	报警时间小时十位, BCD值。0 ~ 2	00
ALMHOURL	3:0	报警时间小时个位, BCD值。0 ~ 9	0000

7、RTC报警天数寄存器 (ALMDATE)

Register	Address	R/W	Description	Reset Value
ALMDATE	0x57000060(L) 0x57000063(B)	R/W (字节)	报警日期 天数寄存器	0x01

字段名	位	意 义	初值
Reserved	7:6	保留 (为0)	00
ALMDATEH	5:4	报警日期天数十位, BCD值。0 ~ 3	00
ALMDATEL	3:0	报警日期天数个位, BCD值。0 ~ 9	0001

8、报警时间月数寄存器 (ALMMON)

Register	Address	R/W	Description	Reset Value
ALMMON	0x57000064(L) 0x57000067(B)	R/W (字节)	报警日期 月数寄存器	0x01

字段名	位	意 义	初值
Reserved	7:5	保留 (为0)	000
ALMMONH	4	报警日期月数十位, BCD值。0 ~ 1	0
ALMMONL	3:0	报警日期月数个位, BCD值。0 ~ 9	0001

9、报警时间年数寄存器 (ALMYEAR)

Register	Address	R/W	Description	Reset Value
ALMYEAR	0x57000068(L) 0x5700006B(B)	R/W (字节)	报警年数 寄存器	0x00

字段名	位	意 义	初值
ALMYEARH	7:4	报警日期年数十位， BCD值。0 ~ 9	0000
ALMYEARL	3:0	报警日期年数个位， BCD值。0 ~ 9	0000

说明：年数的千位和百位应该是20。

10、秒循环复位寄存器 (RTCRST)

Register	Address	R/W	Description	Reset Value
RTCRST	0x5700006C(L) 0x5700006F(B)	R/W (字节)	秒循环复位 寄存器	0x00

字段名	位	意 义	初值
Reserved	7:4	保留 (为0)	000
SRSTEN	3	秒循环复位控制位。 0：禁止；1：允许	0
SECCR	2:0	秒循环进位边界。 011：30秒； 100：40秒；101：50秒。	00

说明：对于秒循环进位边界设为其它值，到设定值只复位，但不会向分钟进位。

11、当前时间秒数寄存器 (BCDSEC)

Register	Address	R/W	Description	Reset Value
BCDSEC	0x57000070(L) 0x57000073(B)	R/W (字节)	当前时间 秒数寄存器	-

字段名	位	意 义	初值
Reserved	7	保留 (为0)	-
NOWSECH	6:4	当前时间秒十位, BCD值。0 ~ 5	-
NOWSECL	3:0	当前时间秒个位, BCD值。0 ~ 9	-

12、当前时间分钟寄存器 (BCDMIN)

Register	Address	R/W	Description	Reset Value
BCDMIN	0x57000074(L) 0x57000077(B)	R/W (字节)	当前时间 分钟寄存器	-

字段名	位	意 义	初值
Reserved	7	保留 (为0)	-
NOWMINH	6:4	当前时间分钟十位, BCD值。0 ~ 5	-
NOWMINL	3:0	当前时间分钟个位, BCD值。0 ~ 9	-

13、当前时间小时数寄存器 (BCDHOUR)

Register	Address	R/W	Description	Reset Value
BCDHOUR	0x57000078(L) 0x5700007B(B)	R/W (字节)	当前时间 小时寄存器	-

字段名	位	意 义	初值
Reserved	7:6	保留 (为0)	-
NOWHOURH	5:4	当前时间小时十位, BCD值。0 ~ 2	-
NOWHOURL	3:0	当前时间小时个位, BCD值。0 ~ 9	-

14、当前日期天数寄存器 (BCDDATE)

Register	Address	R/W	Description	Reset Value
BCDDATE	0x5700007C(L) 0x5700007F(B)	R/W (字节)	当前日期 天数寄存器	-

字段名	位	意 义	初值
Reserved	7:6	保留 (为0)	-
NOWDATEH	5:4	当前日期天数十位, BCD值。0 ~ 3	-
NOWDATEL	3:0	当前日期天数个位, BCD值。0 ~ 9	-

15、当前星期寄存器 (BCDDAY)

Register	Address	R/W	Description	Reset Value
BCDDAY	0x57000080(L) 0x57000083(B)	R/W (字节)	当前星期 寄存器	-

字段名	位	意 义	初值
Reserved	7:3	保留 (为0)	-
NOWDAY	2:0	当前星期值。1 ~ 7	-

1: 星期日; 2: 星期一; 3: 星期二;
4: 星期三; 5: 星期四; 6: 星期五;
7: 星期六

16、当前日期月数寄存器（BCDMON）

Register	Address	R/W	Description	Reset Value
BCDMON	0x57000084(L) 0x57000087(B)	R/W (字节)	报警日期 月数寄存器	0x01

字段名	位	意 义	初值
Reserved	7:5	保留（为0）	000
NOWMONH	4	当前日期月数十位， BCD值。0 ~ 1	0
NOWMONL	3:0	当前日期月数个位， BCD值。0 ~ 9	0001

17、当前日期年数寄存器 (BCDYEAR)

Register	Address	R/W	Description	Reset Value
BCDHOUR	0x57000088(L) 0x5700008B(B)	R/W (字节)	当前日期 年数寄存器	0x00

字段名	位	意 义	初值
NOWYEARH	7:4	当前日期年数十位， BCD值。0 ~ 9	0000
NOWYEARL	3:0	当前日期年数个位， BCD值。0 ~ 9	0000

说明：年数的千位和百位应该是20

四、S3C2410 RTC使用方法

1、读/写寄存器

(1) 设置允许读写：对寄存器RTCCON 的0 位写1

(2) 显示时间、日期方法：需要不断地从BCDSEC、BCDMIN、BCDHOUR、BCDDAY、BCDDATE、BCDMON、和BCDYEAR 寄存器读取数据，然后显示出来

(3) 可能会引起显示错误

例如，假设用户在2006年12月31日23点59分59秒读取寄存器BCDYEAR到BCDSEC，在用户读取BCDSEC寄存器时，如果结果是0，那么很有可能年、月、日、时、分已经变成了2007年1月1日0时0分，数据组合在一起可能是错的。读取的数据可能是：

1) 2006年12月1日0时0分

2) 2006年1月1日0时0分，等

解决的方法：当读取到的BCDSEC等于0时，用户应该再读取一次

BCDYEAR到BCDSEC的值。

2、报警功能

RTC的报警寄存器（RTCALM） 决定了报警的使能、禁止、以及报警时间设定的条件

在RTC报警的使能情况下：

（1）在正常工作模式下，报警中断（ALMINT）是激活状态的

（2）在掉电模式下（PWDN信号有效），电源管理唤醒信号（PMWKUP）与报警中断（ALMINT）都是激活状态

3、节拍中断

RTC 节拍用于中断请求

TICNT 寄存器：有中断使能位、节拍时间位

当节拍计数值到达0 时，就会触发节拍中断。节拍中断的间隔时间计算如下：

$$\text{Period} = (n+1)/128 \text{ 秒}$$

n：节拍时间计数值（1~127）

说明： RTC 节拍中断可以作为RTOS（实时操作系统）内核的时间节拍

五、应用举例

例：编写一程序，对S3C2410的RTC进行设置，使用节拍中断，每1秒中断一次，中断后显示出当前的日期和时间。初始日期、时间设置为正确值。

1、设置控制寄存器：

RTCCON=0x0 0 0 1 = 0x01

含义：RTC不复位、正常BCD计数、BCD时钟选择为1/32768、允许读出RTC值

2、设置节拍时间寄存器：

TICNT=0x1 1 1 1 1 1 1 = 0xFF

含义：允许节拍中断、节拍计数值为127，每1秒钟中断一次。

BIT_TICK EQU (0x1<<8)

BIT_ALLMSK EQU (0xffffffff)

BIT_RTC EQU (0x1<<30)

#include "2410addr.h"

#include "2410lib.h"

void Test_Rtc_Tick(void);

void __irq Rtc_Tick(void);

void Test_Rtc_Tick(void)

```
{  
    Uart_Printf("RTC Tick interrupt test for S3C2410 ! \n");  
  
    pISR_TICK = (unsigned)Rtc_Tick;           //设置中断服务程序地址  
    rINTMSK & = ~(BIT_TICK);                 // (0x1<<8) 开时钟节拍中断  
                                              //设置日期与时间
```

```
    rBCDYEAR = 0x07;  
    rBCDMON  = 0x05;  
    rBCDDAY  = 0x03;                          //SUN:1 MON:2 TUE:3 WED:4 THU:5 FRI:6 SAT:7  
    rBCDDATE = 0x22;  
    rBCDHOUR = 0x08;  
    rBCDMIN  = 0x38;  
    rBCDSEC  = 0x25;
```

```
    rTICNT   = (1<<7) + 127;                 //设置节拍值      Period = (n + 1) / 128 =1sec  
    rRTCCON  = 0x01;                         //启动计时、允许读写
```

```
    Uart_Printf("Press any key to exit.\n");  
    Uart_Getch();                             //等待按键
```

```
    rINTMSK |= BIT_TICK;                     //(1<<8)  
                                              //关闭时钟节拍中断
```

```
}
```

```
void __irq Rtc_Tick(void)
```

```
{
```

```
    char  year,month,date,hour,min,sec;
```

```
//读取年月日
```

```
    year  = rBCDYEAR;
```

```
    month = rBCDMON & 0x1F;
```

```
    date  = rBCDDATE & 0x3F;
```

```
//读取时分秒
```

```
    hour  = rBCDHOUR & 0x3F;
```

```
    min   = rBCDMIN & 0x7F;
```

```
    sec   = rBCDSEC & 0x7F;
```

```
//显示日期、时间
```

```
    Uart_Printf("20%2x年%2x月%2x日,  
                %2x: %2x: %2x\n", year,  
                month ,date hour,min,sec);
```

```
    rSRCPND |= BIT_TICK;
```

```
//清除中断请求标志
```

```
    rINTPND |= BIT_TICK;
```

```
//清除中断请求标志
```

```
}
```

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

16 USB接口

主要内容

USB总线概述

结构

寄存器

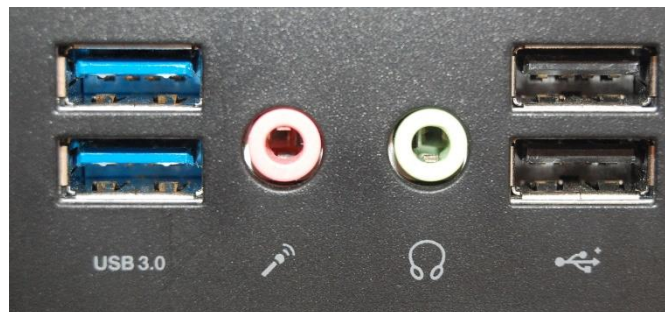
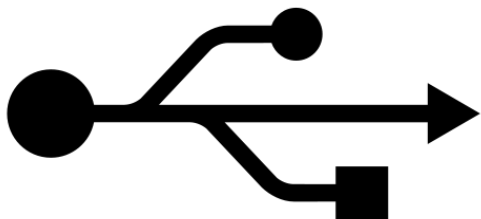
重点：USB拓扑结构、逻辑组成、传输方式

一、USB总线概述

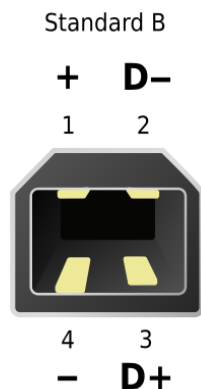
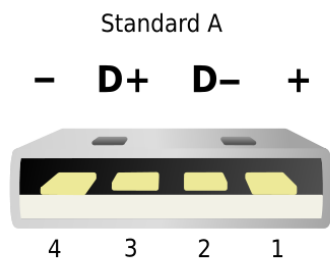
- USB (Universal Serial Bus, 通用串行总线) 是由Intel、Compaq、Microsoft 等公司联合提出的一种串行总线标准，主要用于PC与外围设备互连





Release name ◆	Release date ◆	Maximum transfer rate ◆
USB 0.8	December 1994	
USB 0.9	April 1995	
USB 0.99	August 1995	
USB 1.0-RC	November 1995	
USB 1.0	January 1996	Full Speed (12 Mbit/s)
USB 1.1	August 1998	Full Speed (12 Mbit/s) ^[31]
USB 2.0	April 2000	High Speed (480 Mbit/s)
USB 3.0	November 2008	SuperSpeed (5 Gbit/s)
USB 3.1	July 2013	SuperSpeed+ (10 Gbit/s)
USB 3.2	September 2017	SuperSpeed+ (20 Gbit/s)

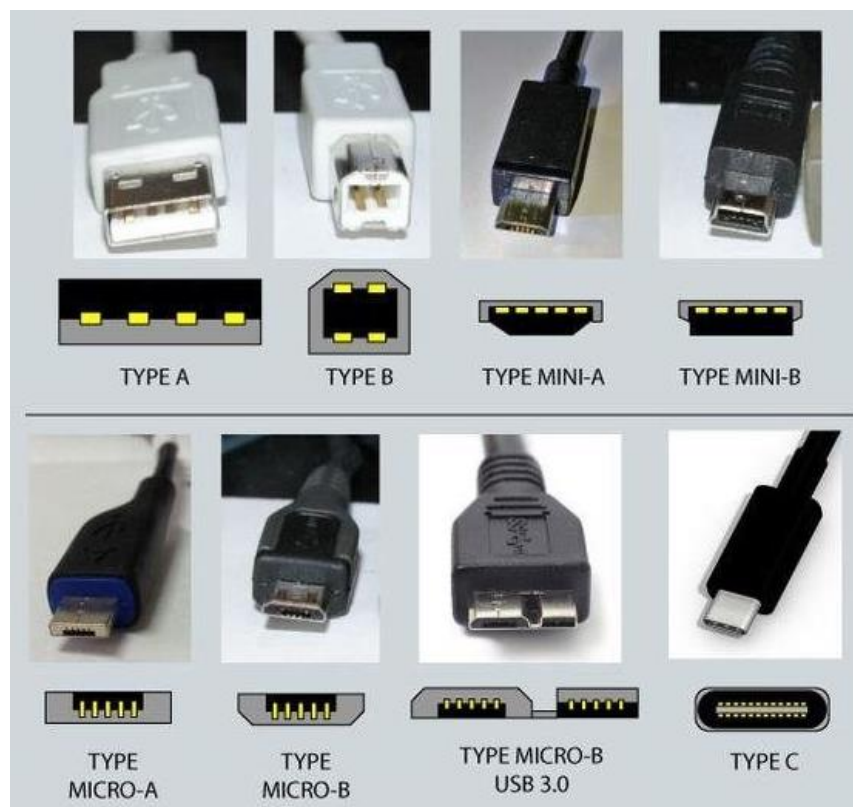
一、USB总线概述



USB



Pin 1	 V_{BUS} (+5 V)
Pin 2	 Data-
Pin 3	 Data+
Pin 4	 Ground



USB传输方式（4种）

1) 同步传输：设备与主机同步，速度高，一次传输，不确保无错误。如用于声音、视频传输

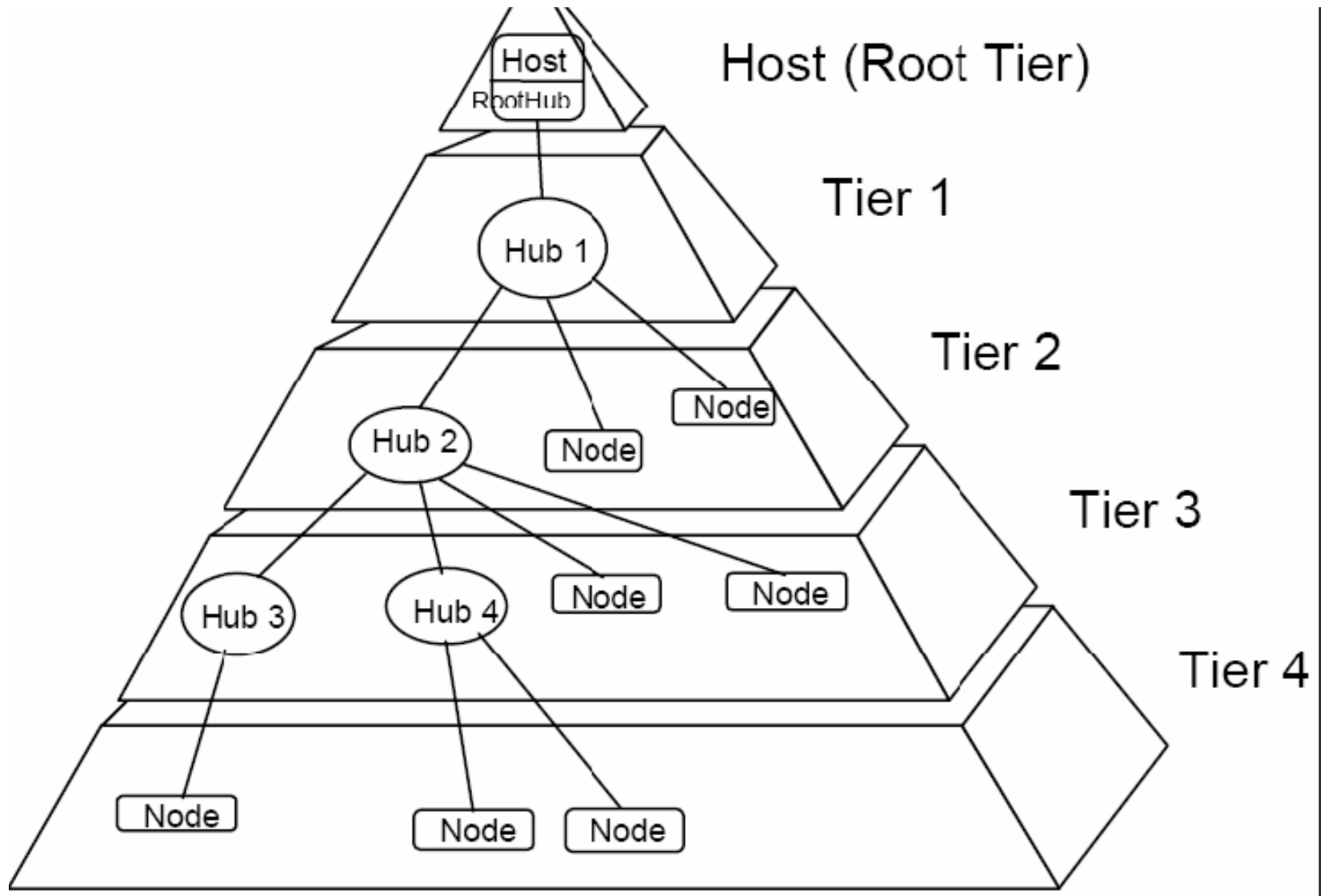
2) 中断传输：实时性强，应用于数据量少、分散、不可预测的数据传输中。如键盘、鼠标、游戏杆操作

USB传输方式（4种）

3) 批量传输：应用于大量数据传输，保证传输数据正确无误。但对数据的实效性要求不高。如打印机、扫描仪等

4) 控制传输：传输的不是数据，而是命令和状态信号，主要用于主机对USB设备进行配置、控制、查询状态等。该方式数据量小、实效性要求也不高

USB拓扑结构（树状结构）

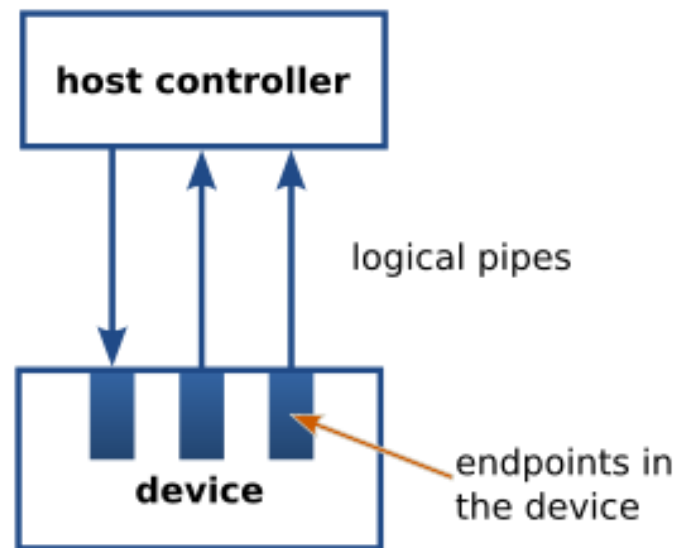


USB接口逻辑组成

1) USB主机 (host) : 控制USB总线上所有的USB设备和所有集线器的数据通信过程

- 检测、连接、断开设备
- 控制数据流
- 收集状态、纠正错误等。

2) USB设备 (device) : 所有的USB设备均可接收数据, 根据数据包的地址判断是否保存

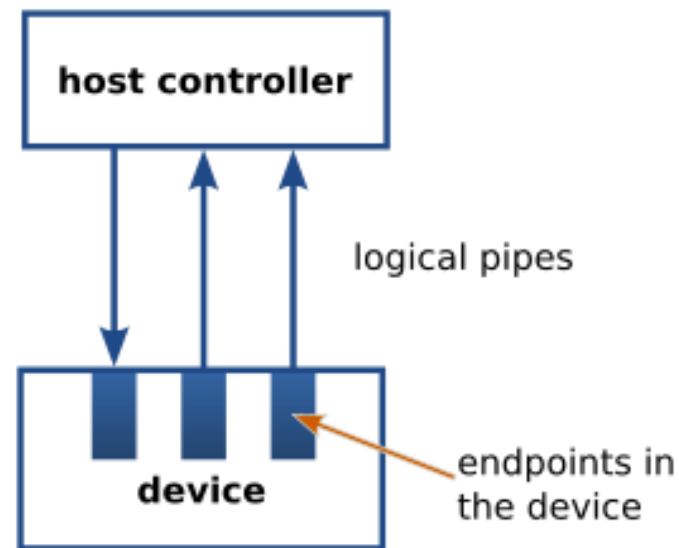


USB接口逻辑组成

3) 端点 (endpoint) : 端点是位于USB设备中、与USB主机进行通信的基本单元。USB设备可以有多个端点，各端点的地址由设备地址和端点号确定

在USB设备中，端点就是一个数据缓冲区

4) 管道 (pipe) : 是主机与设备之间数据通信的逻辑通道。



USB总线主要特点

- USB端口不区分设备
- 即插即用、可热插拔
- 传输速度快
- 易扩展，可扩展到127个USB设备
- 对设备提供电源
- 成本低等

二、S3C2410的USB 接口结构

S3C2410处理器内部集成的USB HOST控制器支持两个USB host 通讯端口和1个USB Device端口

1、USB 控制器的主要特点

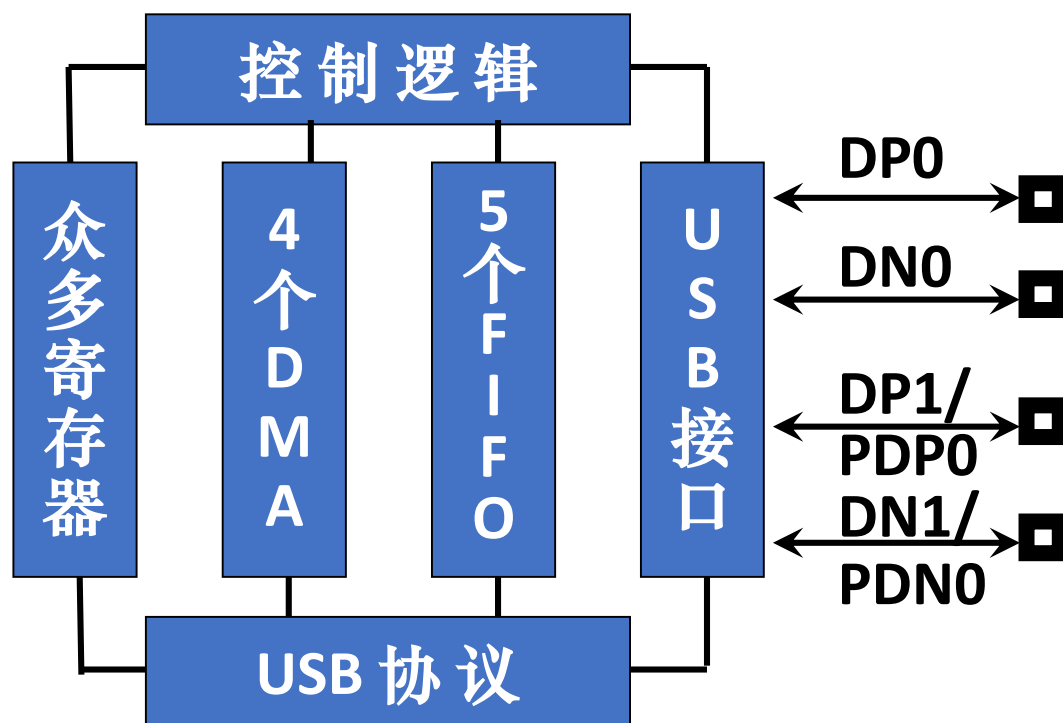
- 符合USB 1.1 协议规范
- 支持USB低速(1.5Mb/s)和全速(12Mb/s)设备连接
- 支持控制、中断、批量数据传输方式（无同步方式）

1、USB 控制器的主要特点

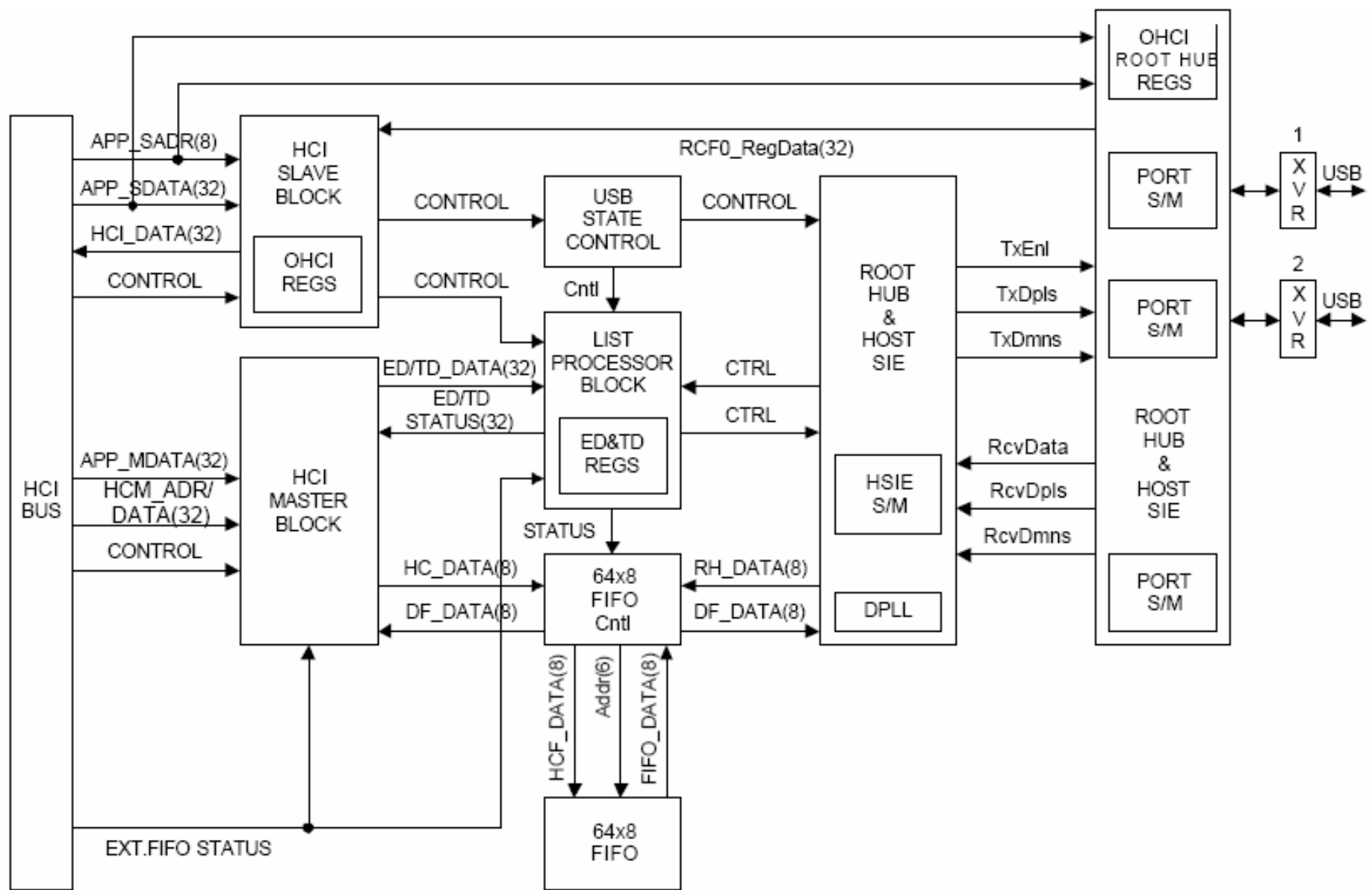
- 集成了5个配置有FIFO缓冲器的节点
 - 1个有16字节的FIFO (EP0)
 - 4个有64字节的FIFO (EP1---EP4)
- 支持DMA方式批量传输 (EP1---EP4)
- 集成了USB收发器
- 支持挂起和远程唤醒功能

2、S3C2410的USB 原理结构

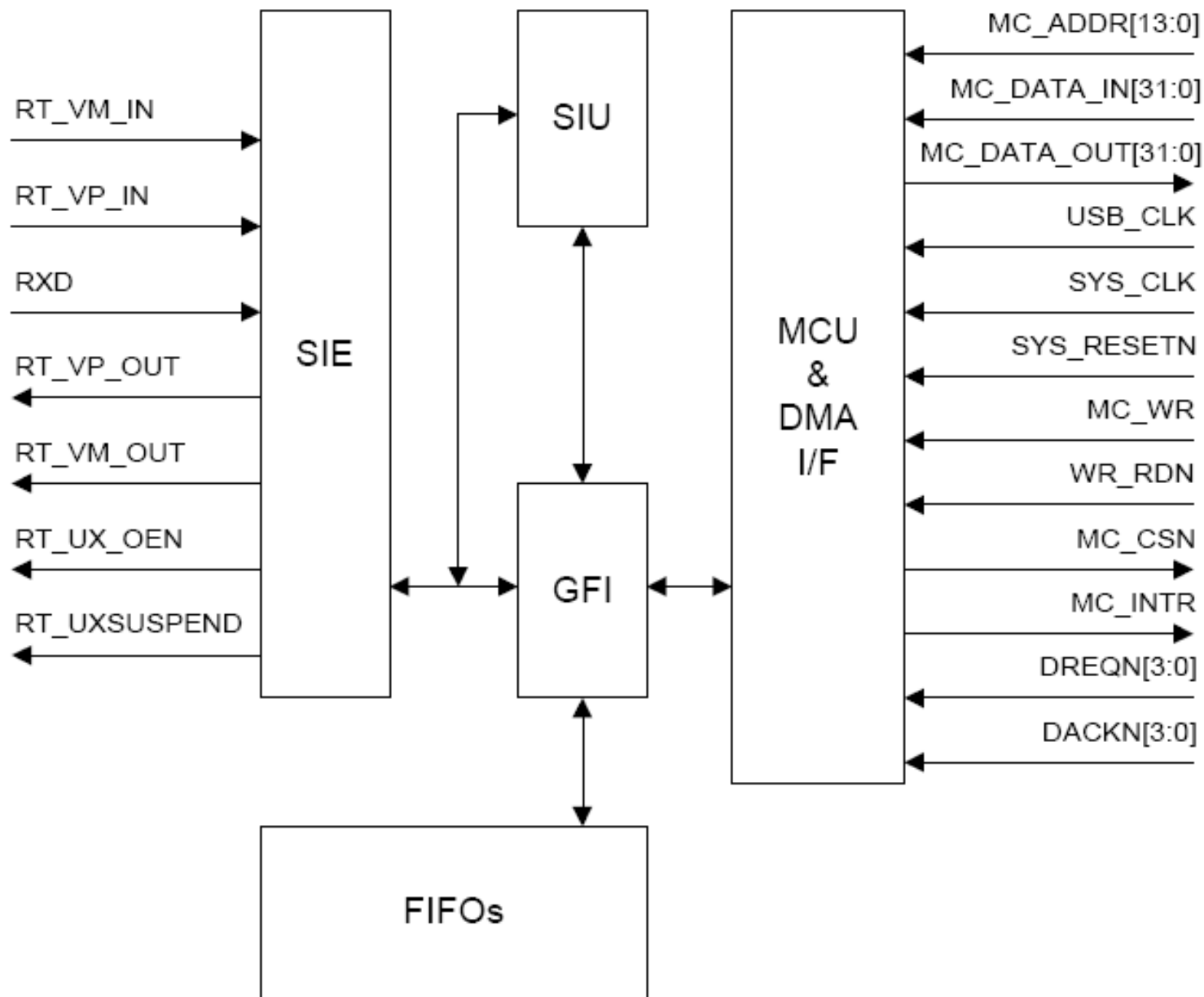
- 主要有5部分构成：控制逻辑、USB协议、5个FIFO、4个DMA、USB接口



S3C2410的USB HOST结构



S3C2410的USB DEVICE结构



三、USB DEVICE专用寄存器

共46个，其基地址为0x52000000

寄存器	描 述	偏址
FUNC_ADDR_REG	功能地址寄存器	140
PWR_REG	电源管理寄存器	144
EP_INT_REG(EP0–EP4)	端点0---4中断寄存器	148
USB_INT_REG	USB中断寄存器	158
EP_INT_EN_REG (EP0–EP4)	端点中断允许寄存器	15C
USB_INT_EN_REG	USB中断允许寄存器	16C

三、USB DEVICE专用寄存器（续1）

寄存器	描 述	偏址
FRAME_NUM1_REG	帧序号低字节寄存器	170
FRAME_NUM2_REG	帧序号高字节寄存器	174
INDEX_REG	索引寄存器	178
EP0_FIFO_REG	端点0 FIFO 寄存器	1C0
EP1_FIFO_REG	端点1 FIFO 寄存器	1C4
EP2_FIFO_REG	端点2 FIFO 寄存器	1C8
EP3_FIFO_REG	端点3 FIFO 寄存器	1CC
EP4_FIFO_REG	端点4 FIFO 寄存器	1D0

三、USB DEVICE专用寄存器（续2）

寄存器	描 述	偏址
EPn_DMA_CON	端点n DMA控制寄存器	2xx
EPn_DMA_UNIT	端点n DMA传输单位寄存器	2xx
EPn_DMA_FIFO	端点n DMA FIFO计数器	2xx
EPn_DMA_TTC_L	端点n DMA传输计数器L	2xx
EPn_DMA_TTC_M	端点n DMA传输计数器M	2xx
EPn_DMA_TTC_H	端点n DMA传输计数器H	2xx

说明： n=1, 2, 3, 4。因此是4组，24个寄存器。

三、USB DEVICE专用寄存器（续3）

寄存器	描 述	偏址
IN_CSR1_REG /EP0_CSR	端点输入控制状态寄存器1 /端点0控制状态寄存器	184
IN_CSR2_REG	端点输入控制状态寄存器2	188
MAXP_REG	端点最大包寄存器	18C
OUT_CSR1_REG	端点输出控制状态寄存器1	190
OUT_CSR2_REG	端点输出控制状态寄存器2	194
OUT_FIFO_CNT1_REG	端点写输出计数寄存器1	198
OUT_FIFO_CNT2_REG	端点写输出计数寄存器2	19C

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

17 看门狗

主要内容

USB总线概述

结构

寄存器

应用举例

重点：WDT定义与工作原理；理解其应用方法

一、看门狗及工作原理

1、看门狗

嵌入式系统运行受到外部干扰或者系统错误，程序有时会出现“跑飞”，导致整个系统瘫痪。在对系统稳定性要求较高的场合，为了防止这一现象的发生，需要“看门狗”(WATCHDOG)电路。看门狗的作用就是当系统“跑飞”而进入死循环时，恢复系统的运行

看门狗：一种定时器电路，具有监视并恢复程序正常运行的功能。

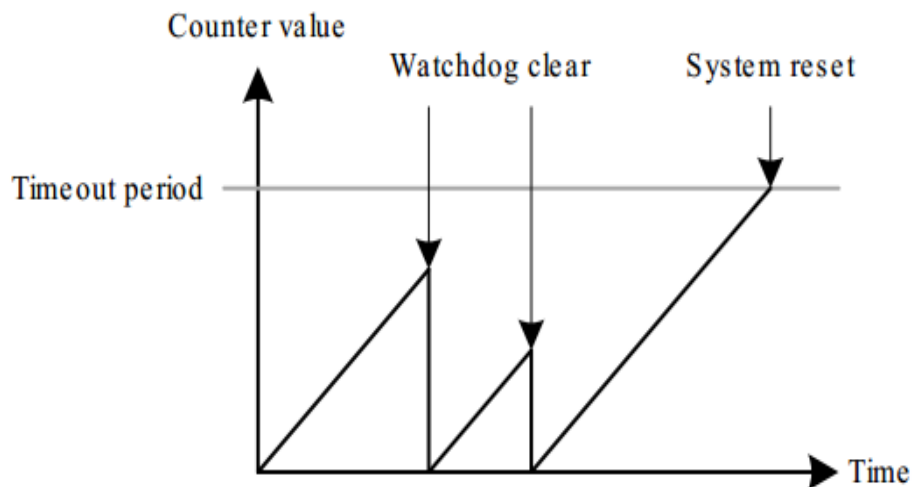
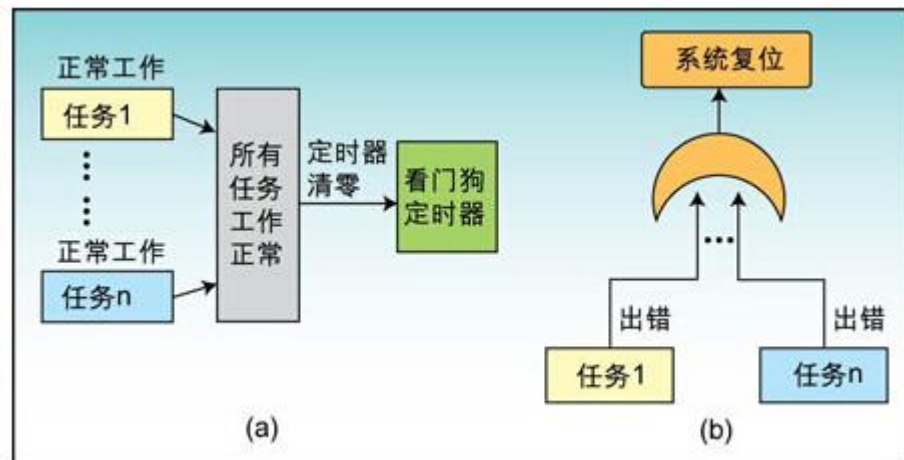
2、看门狗的工作原理

- 基本原理：

设一系统程序完整运行一周期的时间是 T_p ，看门狗的定时周期为 T_i ，要求 $T_i > T_p$

在程序运行一周期后，修改定时器的计数值，只要程序正常运行，定时器就不会溢出

若由于干扰等原因使系统不能在 T_p 时刻修改定时器的计数值，定时器将在 T_i 时刻溢出，引发系统复位，使系统得以重新运行，从而起到监控作用



二、S3C2410 的看门狗

1、S3C2410 看门狗的功能

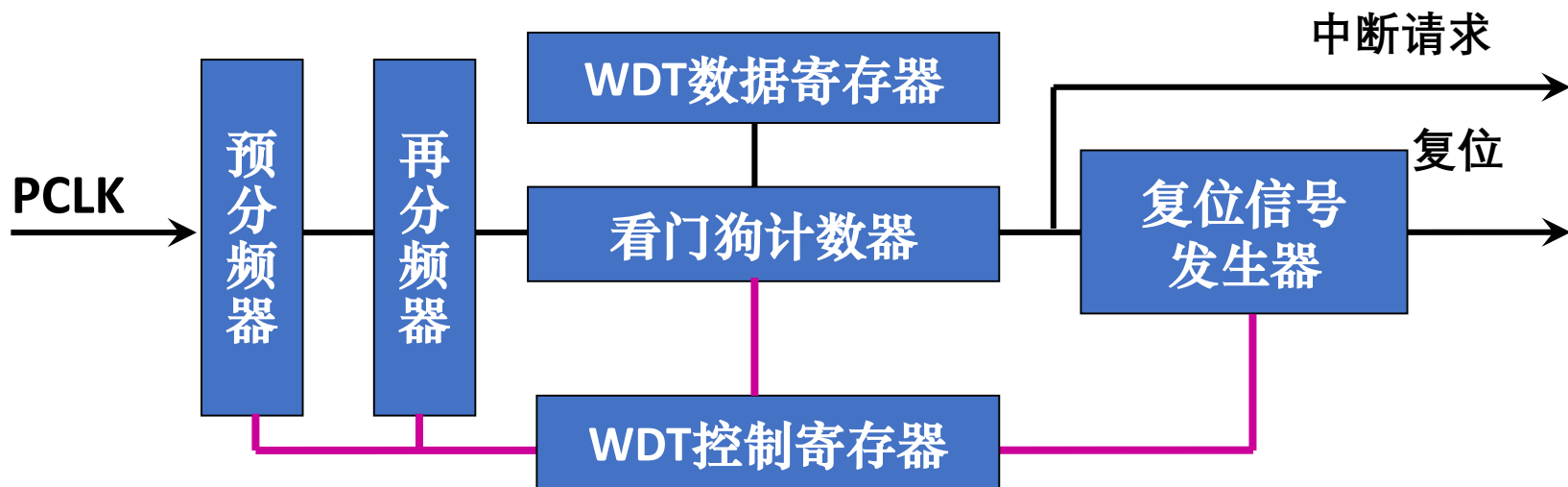
S3C2410 的看门狗定时器有两个功能：

- (1) **定时器功能**：可以作为常规定时器使用，它是一个十六位的定时器，并且可以产生中断，中断名为 INT_WDT，中断号是0x09
- (2) **复位功能**：作为看门狗定使用，当时钟计数减为0（超时）时，它将产生一个128个时钟周期的复位信号

2、S3C2410 看门狗结构与工作原理

由五部分构成：时钟、看门狗计时器、看门狗数据寄存器、复位信号发生器、控制逻辑等

S3C2410 ARM9的看门狗工作原理



3、 S3C2410 看门狗定时时间

预分频器为8位，其值为： 0---255

再分频器可选择值为： 16、 32、 64、 128

输入到计数器的时钟周期为：

$$T_{\text{wtd}} = 1 / [\text{PCLK} / (\text{Prescaler} + 1) / \text{Division_factor}]$$

看门狗的定时周期为：

$$T = \text{WTDAT} \times T_{\text{wtd}}$$

注意：

- 一旦看门狗的定时器启动工作，其数据寄存器（WTDAT）中的值将不会自动读入到时间寄存器中间去（WTCNT）
- 由于这个原因，程序员必须在看门狗计时器启动之前，应该将一个初始值写入到看门狗的时间计数器（WTCNT）中间去
- 即先对时间计数器系初值，再启动看门狗工作

三、看门狗专用寄存器

共3个，其基地址为0x53000000

寄存器	描 述	初值	偏移地址
WTCN	看门狗控制寄存器	0x8021	0
WTDAT	看门狗数据寄存器	0x8000	4
WTCNT	看门狗计数寄存器	0x8000	8

1、看门狗控制寄存器（WTCON）

Register	Off Add	R/W	Description	Reset Value
WTCON	0	R/W	看门狗控制寄存器	0x8021

字段名	位	意 义	初值
Prescaler Value	15:8	预分频值。0---255。	0x80
Reserved	7:6	保留（为0）	00
Watchdog Timer	5	看门狗控制位。 0：禁止；1：允许	1

1、看门狗控制寄存器（WTCON，续）

字段名	位	意 义	初值
Clock Select	4:3	再分频值选择。00：16； 01：32；10：64；11：128	00
Interrupt Generation	2	看门狗中断控制。 0：禁止；1：允许。	0
Reserved	1	保留（为0）	0
Reset Enable	0	看门狗复位功能控制。 0：禁止；1：允许。	1

2、看门狗数据寄存器 (WTDAT)

Register	Off Add	R/W	Description	Reset Value
WTDAT	4	R/W	看门狗数据寄存器	0x8000

说明:

- (1) 该数据寄存器为对看门狗计数器重装计数值，初始值为0x8000
- (2) 在初始化看门狗操作中，WTDATA 的值不会自动加载到定时计数器中
- (3) 在计数溢出后，WTDAT的值将被装载到WTCNT 寄存器中

3、看门狗计数寄存器 (WTCNT)

Register	Off Add	R/W	Description	Reset Value
WTCNT	8	R/W	看门狗计数寄存器	0x8000

字段名	位	意 义	初值
Count Value	15:0	看门狗的当前计数值	0x8000

说明： 在计数中只能读，不能写（写不起作用）

四、应用举例

例：编写一程序，利用S3C2410看门狗中断产生频率为1kHz的方波，并且从GPB0引脚输出。设S3C2410的PCLK为50MHz

(1) 计算数据寄存器值

1) 取再分频值为16，分频后的频率为：

$$50\text{M}/16 = 3125000\text{Hz}$$

2) 取预分频值为25，分频后的频率为：

$$3125000/25 = 125000\text{Hz}$$

3) 取计数值为60，则计数器后的频率为：

$$125000/60 = 2083.3\text{Hz}$$

4) 方波频率为：

$$2083.3/2 = 1042\text{Hz}$$

不可能实现准确的1000Hz方波

(2) 看门狗控制寄存器值

WTCON

=0b 00011000 00 0 00 1 0 0 = 0x1804

含义：预分频值为0x18、保留00、先禁止看门狗定时器工作、选择再分频00（分频值为16）、允许定时器中断、保留0、禁止看门狗复位。

(3) 看门狗数据寄存器值

WTDAT = 60 = 0x3C

```

BIT_WDT EQU          (0x1<<9)
#include <string.h>
#include "2410addr.h"
#include "2410lib.h"
#include "timer.h"

void __irq Wdt_Int(void);
void Test_WDT_IntReq(void)
{
    Uart_Printf("WatchDog Timer Interrupt Request Test! \n");
    rINTMSK &= ~(BIT_WDT);                //开看门狗中断
    pISR_WDT = (unsigned)Wdt_Int;          //设置中断向量
    rGPBCON = rGPBCON&0x03|0x01;          //把GPB0设为输出

    rWTCON = 0x1804;                       //写控制寄存器
    rWTDAT = 60;                           //写数据寄存器
    rWTCNT = 60;                           //写计数器
    rWTCON = rWTCON | (1<<5);              //启动看门狗定时器工作

```



```
Uart_Printf("Press any Key to Exit!\n");
```

```
Uart_Getch(); //等待按键
```

```
rWTCON &= ~(1<<5); //关闭看门狗定时器
```

```
rINTMSK |= (BIT_WDT); //屏蔽看门狗中断
```

```
}
```

```
void __irq Wdt_Int(void)
```

```
{
```

```
rGPBDATA ^= 0x01; //对GPB0取反
```

```
rSRCPND |= BIT_WDT; //清除看门狗中断请求标志
```

```
rINTPND |= BIT_WDT; //清除看门狗中断服务标志
```

```
}
```

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

18 LCD控制器

主要内容

- 1、LCD概述
- 2、S3C2410 LCD控制器
- 3、S3C2410 LCD控制器结构与工作原理
- 4、专用寄存器

要求：了解LCD的工作原理；了解LCD控制器的结构，了解引脚信号

一、LCD概述

- LCD（液晶显示器）主要分成三部分：LCD显示屏、显示控制器、缓冲存储器

1、LCD显示屏（LCD模块）

主要由：玻璃基板、偏振片、电极、胶框、液晶材料、背光等构成。（STN、TFT、OLED等）

2、显示控制器

产生各种信号：场、行扫描及同步信号，显示驱动信号等

显示控制器：如T6963、LPC3600

TN、STN 及 TFT 型液晶显示屏的比较

类 别	TN	STN	TFT
原理	液晶分子扭转 90°	液晶分子扭转 $240^\circ \sim 270^\circ$	液晶分子扭转 90° 以上
特性	黑白、单色，低对比度	黑白、彩色（26 万色），低对比度，较 TN 佳	彩色（1 667 万色），高对比度，较 STN 佳
全色彩化	否	否	全彩色
动画显示	否	否	可以
视角	狭窄（ 30° 以下）	狭窄（ 40° 以下）	较宽（ 80° 以下）
面板尺寸	1~3 英寸	1~12 英寸	12 英寸以上
应用范围	电子表、计算器、简单的掌上游戏机、移动电话	电子词典、移动电话、商务通、低档笔记本电脑	移动电话、笔记本电脑、液晶彩显、液晶彩电



二、S3C2410 LCD控制器

一般功能： S3C2410 LCD控制器具有一般LCD控制器功能，产生各种信号、传输显示数据到LCD驱动器

1、S3C2410特点

(1) 基本特点

1) 有专用DMA

用于向LCD驱动器传输数据

2) 有中断 (INT_LCD)

3) 显示缓存可以很大

系统存储器可以作为显示缓存用

4) 支持多屏滚动显示

用显示缓存支持硬件水平、垂直滚屏

5) 支持多种时序LCD屏

通过对LCD控制器编程，产生适合不同LCD显示屏的扫描信号、数据宽度、刷新率信号等

6) 支持多种数据格式

大端、小端格式，WinCE格式

(2) 支持STN材料LCD

1) 单色显示：每像素2位数据、4级灰度；每像素4位数据、16级灰度。

2) 单色扫描：4位单向、双向扫描，8位单向扫描。

3) 彩色显示：每像素16位数据、65536种色彩，每像素24位数据真色彩。

4) 支持多种LCD屏

640×480、320×240、160×160等

5) 4MB显示缓存

支持256色的像素数：4096×1024, 2048×2048, 1024×4096等

(3) 支持TFT材料LCD

1) 单色显示：每像素1位数据、2位数据、4位数据、8位数据。

2) 彩色显示：每像素16位数据、65536种色彩，每像素24位数据、16M种真色彩。

3) 支持多种LCD屏

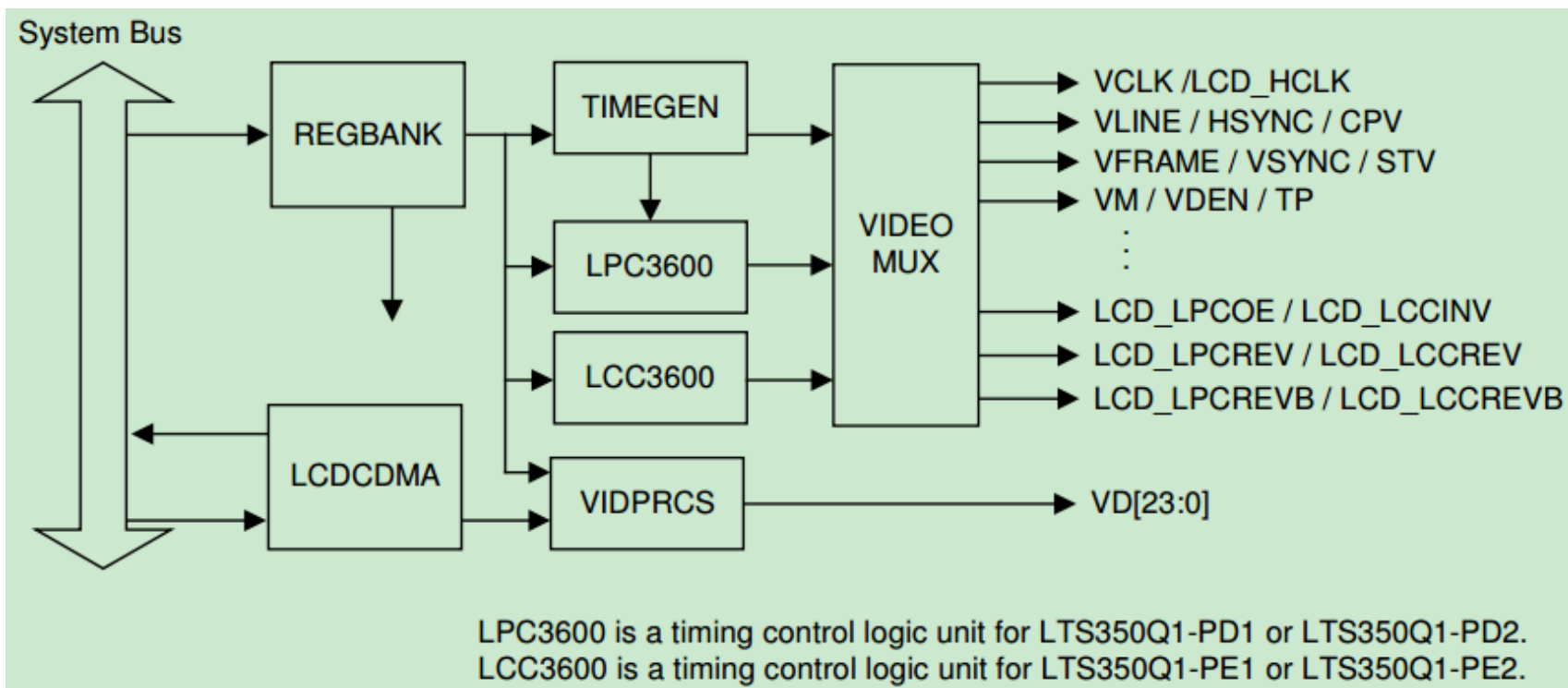
640×480、320×240、160×160等

4) 4MB显示缓存 支持64K色的像素数：2048×1024等

三、S3C2410 LCD控制器结构与工作原理

1、LCD控制器结构

- 主要由6部分组成：时序发生器、LCD主控制器（LPC3600）、DMA、视频信号混合器、数据格式转换器、控制逻辑等



2、LCD控制器引脚信号

共41个信号

VD[23:0]: LCD数据

VDEN: 数据使能

VCLK: 时钟信号

VLIN: 行扫描信号

LEND: 行结束信号

VFRAME: 帧扫描信号

HSYNC: 水平同步信号

VSNC: 垂直同步信号

VM: 显示驱动交流信号

LCDVF0、LCDVF1、LCDVF2: 时序控制信号

LCD_PWREN: 面板电源控制信号

LCD_HCLK: 时钟面板控制信

CPV: 行同步面板控制信号

STV: 帧同步面板控制信号

TP: 显示驱动面板控制信号

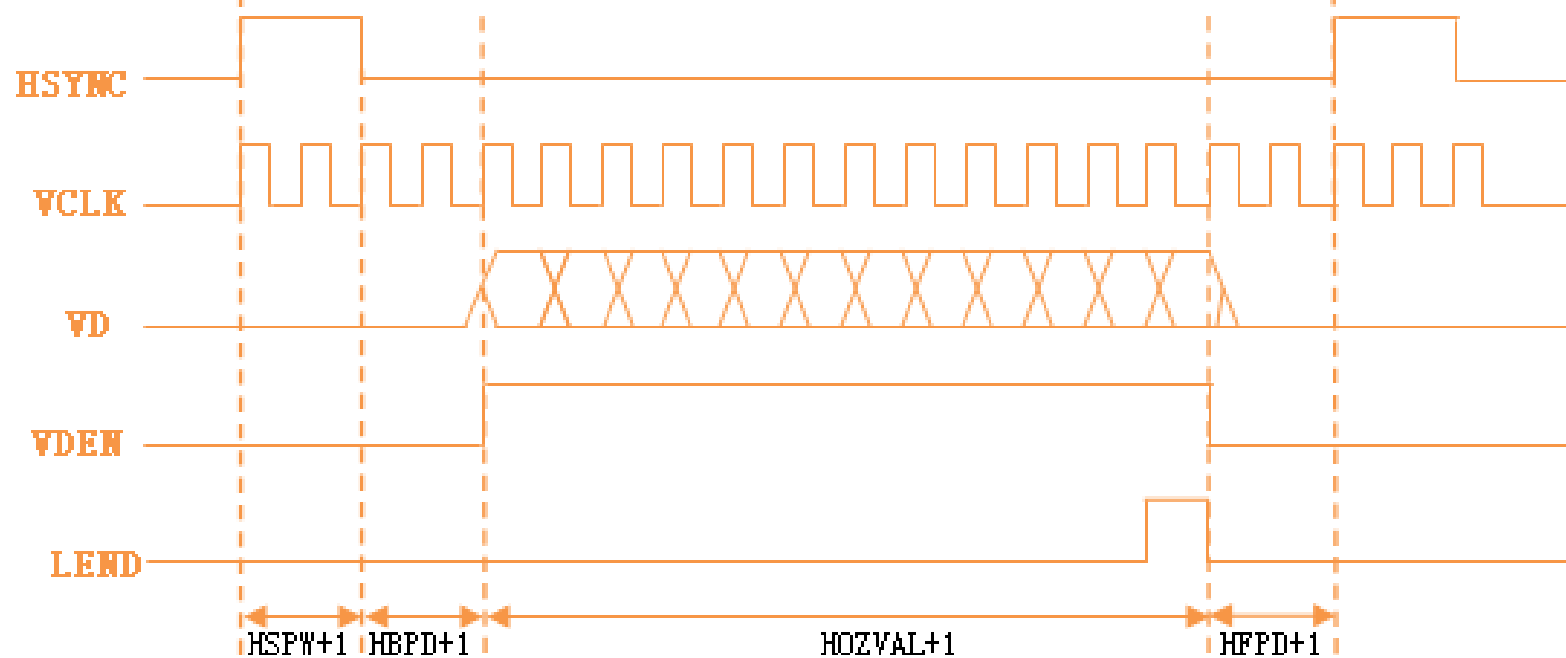
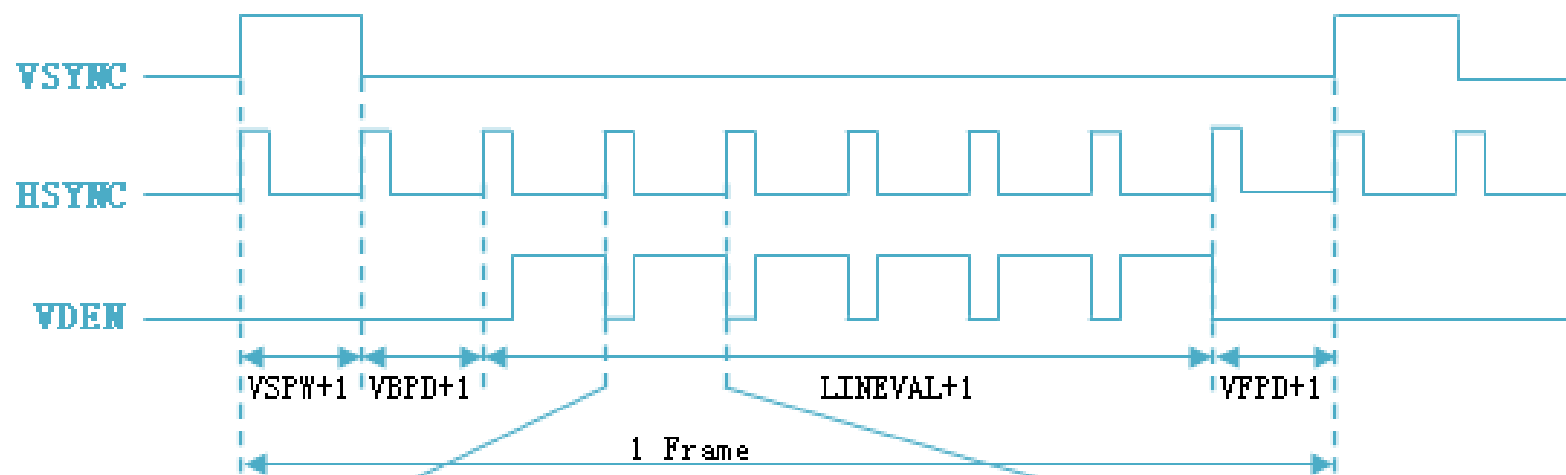
STH: 面板控制信号

→ VCLK / LCD_HCLK
→ VLIN / HSYNC / CPV
→ VFRAME / VSNC / STV
→ VM / VDEN / TP

⋮

→ LCDVF0
→ LCDVF1
→ LCDVF2

→ VD[23:0]



四、LCD控制器专用寄存器

- S3C2410的有17个专用寄存器，分为四类，其基地址均为
0x4D000000

1、控制寄存器（5个，如下表）

寄存器	描 述	初值	偏址
LCDCON1	LCD控制寄存器1	0x00000000	0x00
LCDCON2	LCD控制寄存器2	0x00000000	0x04
LCDCON3	LCD控制寄存器3	0x00000000	0x08
LCDCON4	LCD控制寄存器4	0x00000000	0x0C
LCDCON5	LCD控制寄存器5	0x00000000	0x10

2、地址寄存器

共3个地址寄存器控制寄存器，如下表所示

寄存器	描 述	初值	偏址
LCDSADDR1	帧起始地址寄存器1	0x00000000	0x14
LCDSADDR2	帧起始地址寄存器2	0x00000000	0x18
LCDSADDR3	虚拟屏地址寄存器	0x00000000	0x1C

3、颜色配置寄存器

- 共4个颜色配置寄存器，1个抖动模式寄存器，如下表

寄存器	描 述	初值	偏址
REDLUT	红颜色寄存器	0x00000000	0x20
GREENLUT	绿颜色寄存器	0x00000000	0x24
BLUELUT	蓝颜色寄存器	0x00000000	0x28
DITHMODE	抖动模式寄存器	0x00000000	0x4C
TPAL	临时调色寄存器	0x00000000	0x50

4、中断寄存器

- 共3个中断寄存器，1个LCD控制器寄存器，如下表所示

寄存器	描 述	初值	偏址
LCDSADDR1	中断服务寄存器	0x0	0x54
LCDSADDR2	中断标志寄存器	0x0	0x58
LCDSADDR3	中断屏蔽寄存器	0x3	0x5C
LPCSEL	LPC3600控制器 控制寄存器	0x4	0x60



Thank you