

---

# S3C2410最小系统设计（补充）



# SoC的概念

---

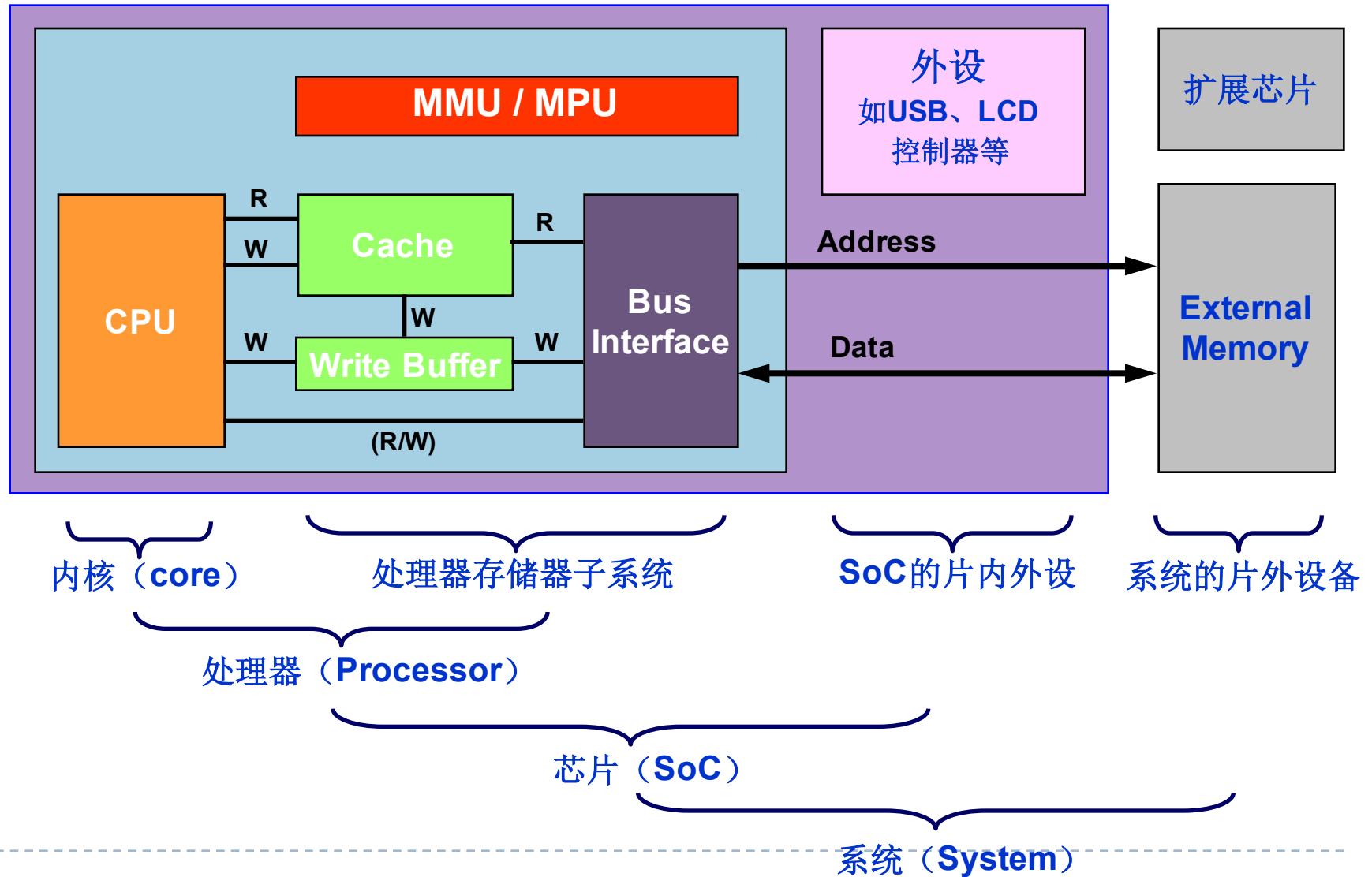
- ▶ **SoC**即**System On Chip**，被称为**系统级芯片或者片上系统**，它是一种面向应用系统的集成电路，包含完整系统的绝大部分功能部件。
- ▶ **SoC嵌入式微处理器**就是一种超大规模集成电路系统，它将许多功能模块做在一个单一的芯片上。

# SoC的概念

---

- ▶ 在SOC系统的设计中，大量可重复利用的IP是非常重要的。
- ▶ ARM公司的RISC架构ARM Core、MIPS公司的MIPS32/64 RISC Core、Freescale的PowerPC等，是目前市场上非常知名的嵌入式微处理器IP
- ▶ 许多芯片设计厂商购买这些微处理器IP，再根据目标处理器的类型、功能和市场（应用领域）的定位，加上购买或自己设计的一些外设IP，就可以快速地形成一个高度集成的SOC嵌入式微处理器。

# 典型的SoC结构



# 什么是最小系统

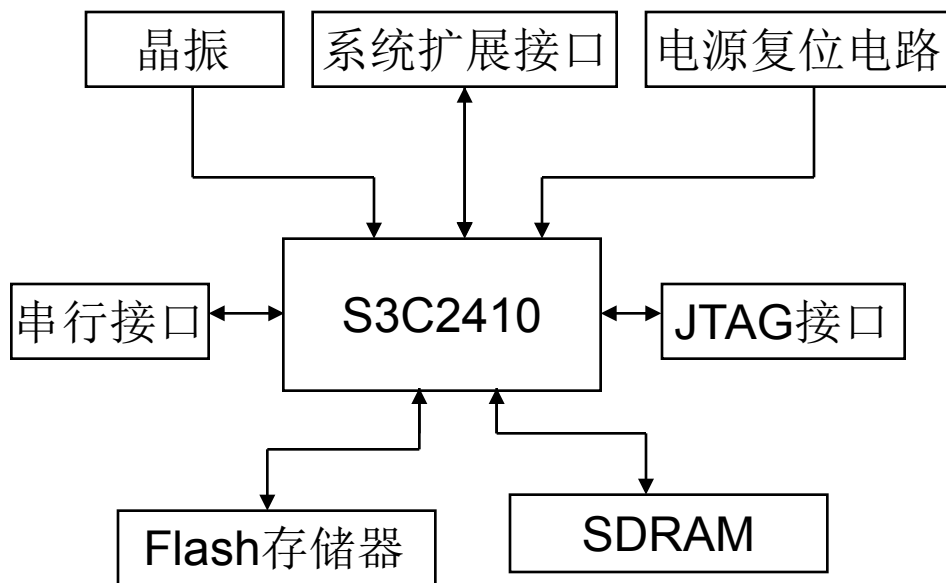
---

嵌入式微处理器芯片自己是不能独立工作的，需要一些必要的外围元器件给它提供基本的工作条件。

一个 **ARM** 最小系统一般包括：

- (1) **ARM** 微处理器芯片
- (2) 电源电路、复位电路，晶振电路，
- (3) 存储器（ **FLASH** 和 **SDRAM** ），
- (4) **UART**（**RS232**及以太网）接口电路。
- (5) **JTAG** 调试接口。

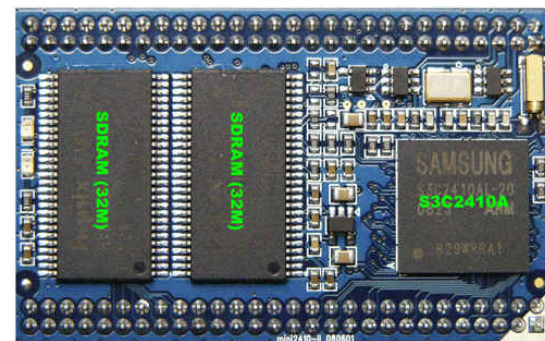




最小系统结构框图



最小系统例板



---

---

## 嵌入式最小系统硬件功能如下：

- **微处理器**：S3C2410是系统工作和控制中心；
- **电源电路**：为S3C2410核心部分提供所需的1.80V工作电压，为部分外围芯片提供3.3V的工作电压；
- **晶振电路**：为微处理器及其他电路提供工作时钟，及系统中S3C2410芯片使用12MHZ及32.768KHz无源晶振；
- **Flash存储器**：存放嵌入式操作系统、用户应用程序或者其他系统在系统掉电后需要保存的用户数据等；



---

---

## 嵌入式最小系统硬件功能如下：

- **SDRAM**：作为系统运行时的主要区域，系统及用户数据、堆栈均位于该存储器中；
- **串行接口**：用于系统与其他应用系统的短距离双向串行通信和构建交叉编译环境；
- **JTAG接口**：对芯片内部所有部件进行访问，通过该接口对系统进行调试、编程等；
- **系统总线扩展**：引出地址总线、数据总线和必须的控制总线，便于用户根据自身的特定需求，扩展外围电路。

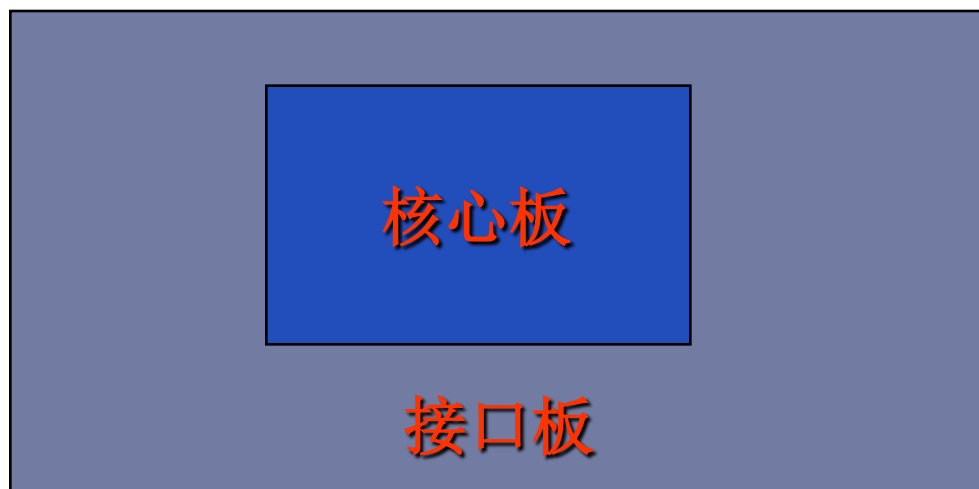


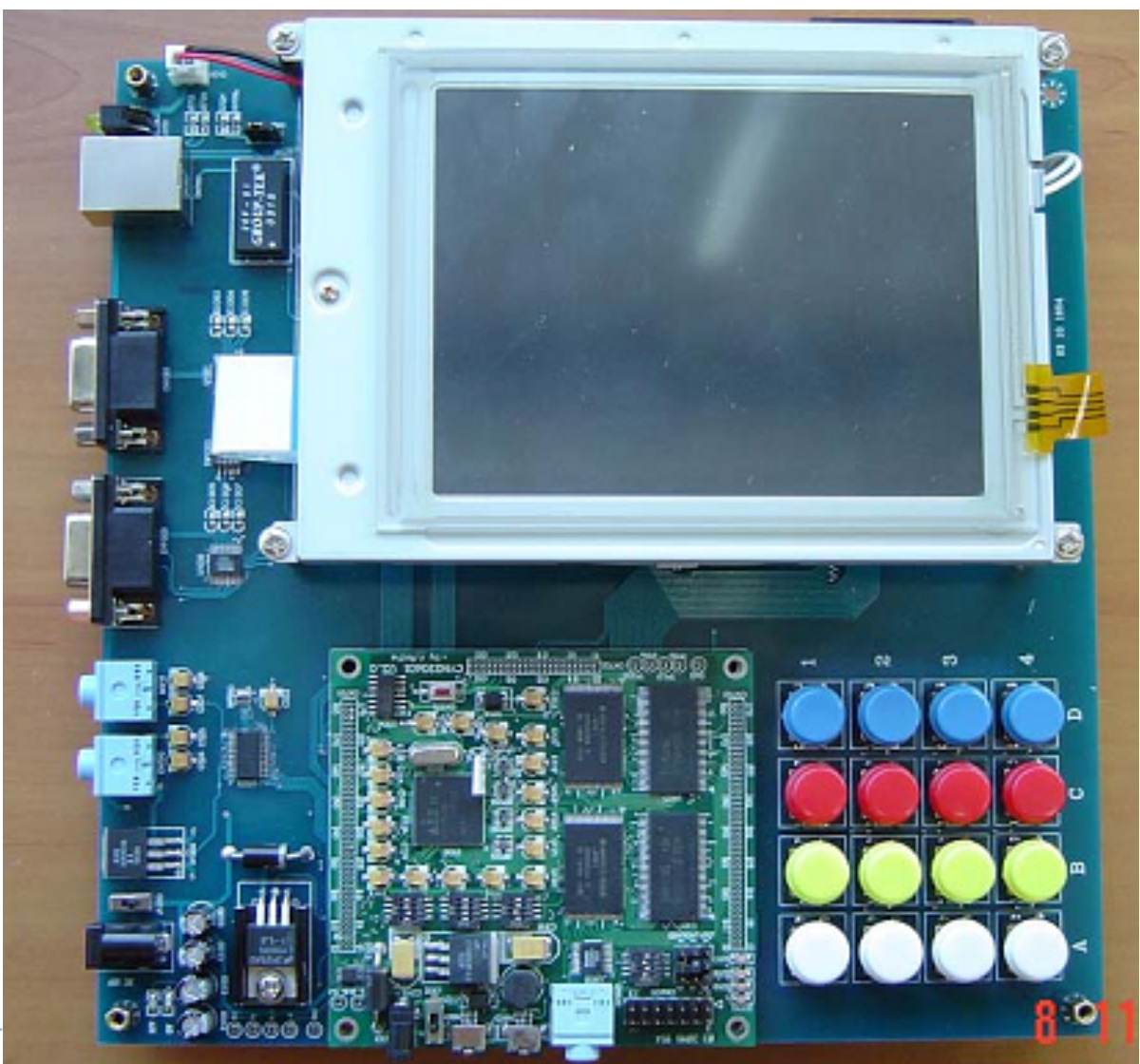


---

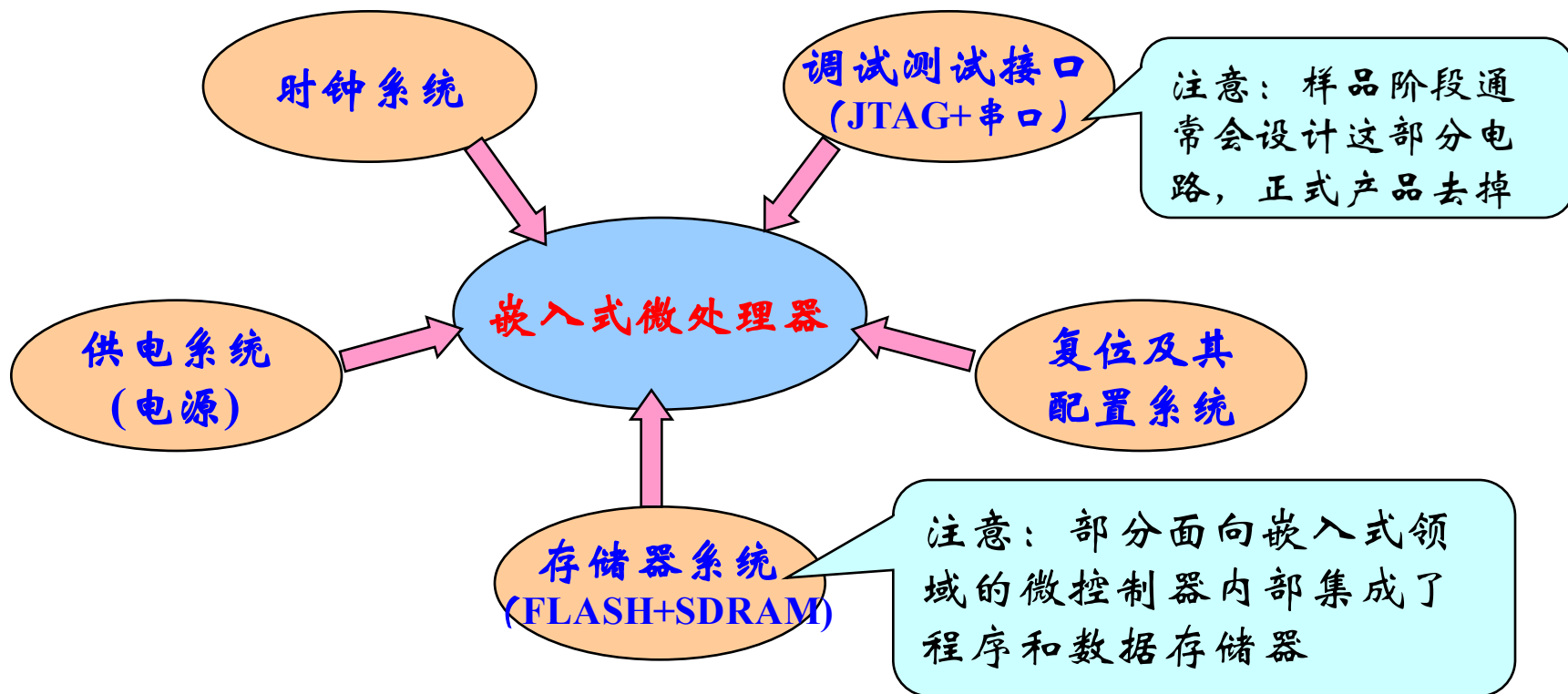
---

在嵌入式系统中，最小系统虽然简单，但是作为整个系统正常运行的基本条件，因此其稳定可靠的运行是至关重要的。因此，在嵌入式系统中，往往将**最小系统制成一个核心板**，其他的各种接口和外围扩展设备都制成一个接口板来组成一个系统。

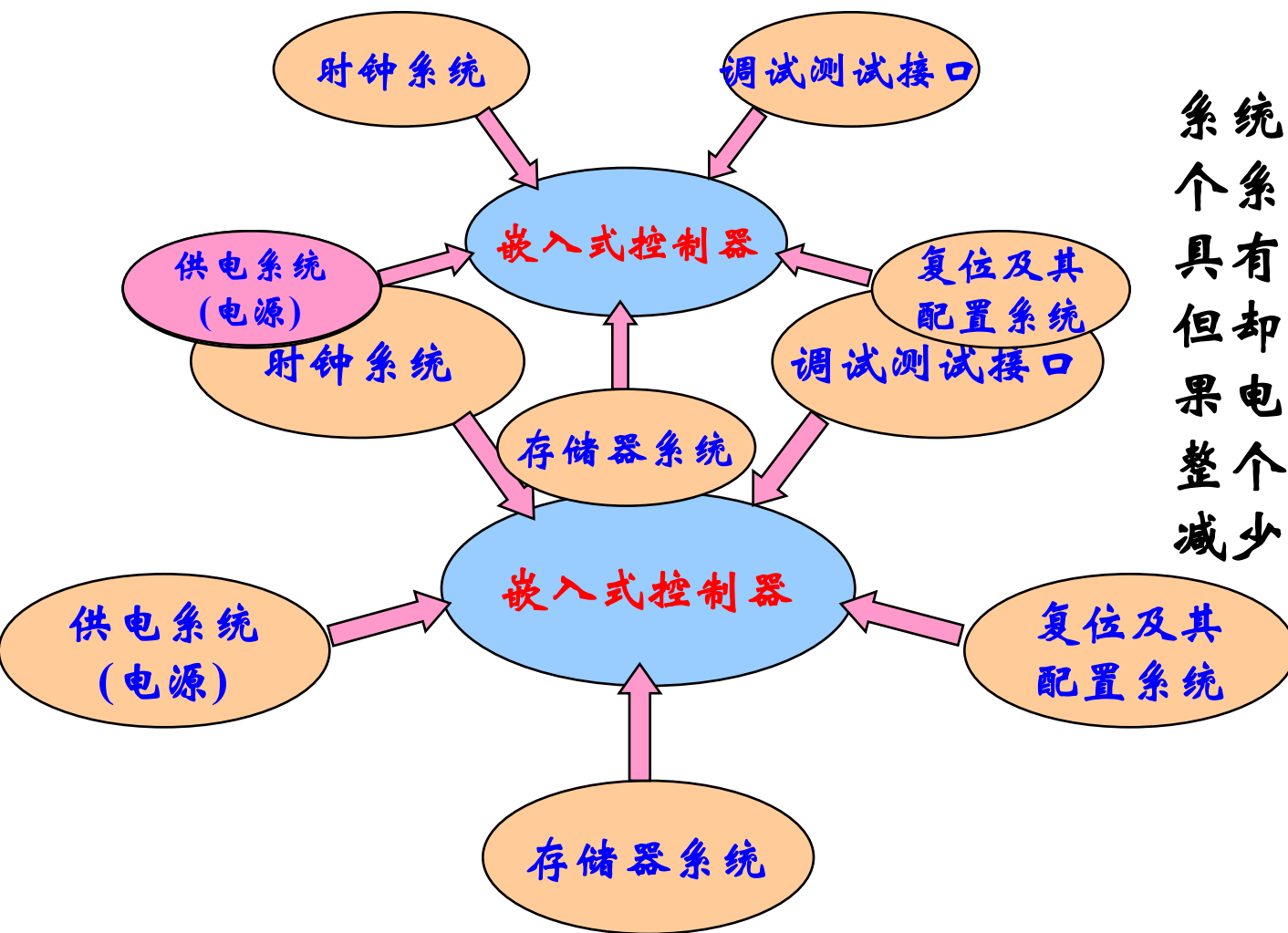




# 最小系统框图

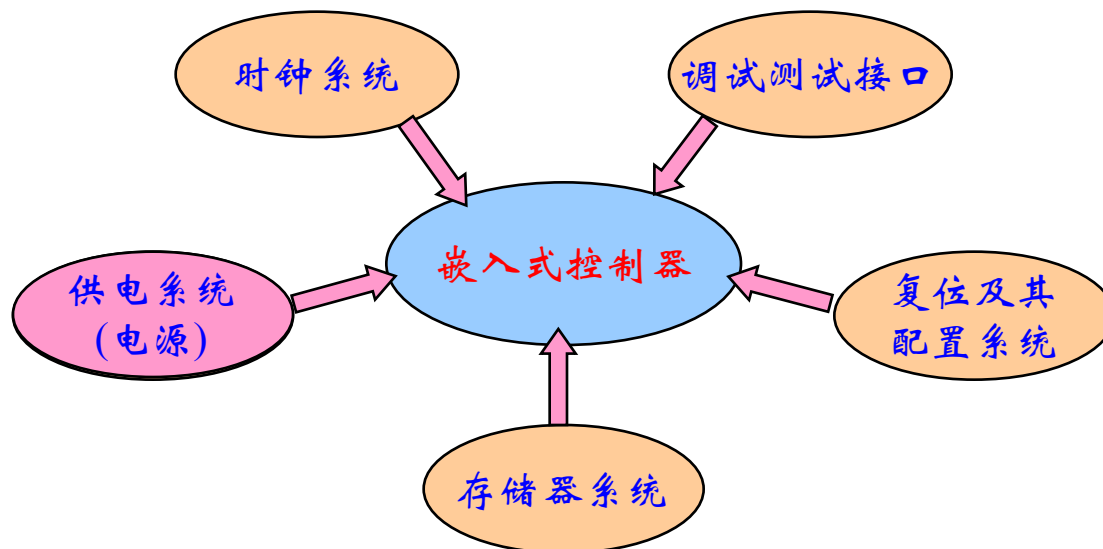


# 一、电源电路设计



电源系统为整个系统提供能量，是整个系统工作的基础，具有极其重要的地位，但却往往被忽略。如果电源系统处理得好，整个系统的故障往往减少了一大半。

## ► 电源电路设计



设计电源时要考虑的因素：

- 1.输出的电压、电流、功率；
- 2.输入的电压、电流；
- 3.安全因素；
- 4.电磁兼容和电磁干扰；
- 5.体积限制；
- 6.成本限制。

## 1. 分析需求

---

---

S3C2410处理器所需要的电源类型：

- 处理器内核所需电源：1.8V
- RTC模块为时钟电路提供的电源：1.8V
- 存储器端口和普通IO口所需的电源：3.3V
- ADC模块所需要的电源：3.3V

由此可见，在电源系统中，至少需要3.3V和1.8V的两种直流稳压电源。

## 2.设计电源电路

---

实际的电源电路设计还需要考虑整个电路板的其他模块对电源的需求，并具体到对1.8V和3.3V的需求。

除处理器外，其他模块对1.8V电源有需求的较少，但对3.3V电源有需求的模块较多。

## 设计电源电路

---

电源设计在确定电压需求之后，需重点分析**功耗要求**。

S3C2410A处理器1.8V消耗的电流约100mA。为了保证可靠性并**留下余量**，则电源系统1.8V能够提供的电流应当大于300mA。

整个系统在3.3V上消耗的电流与其他功能模块和外部条件有很大的关系，这里假设电流不超过300mA，这样，电源系统3.3V能够提供600mA电流即可。

■ 分析得到以下参数：

3.3V电源设计最大电流：600mA；

1.8V电源设计最大电流：300mA。

---



---

---

输入电源选择高质量的**5V**直流稳压电源。

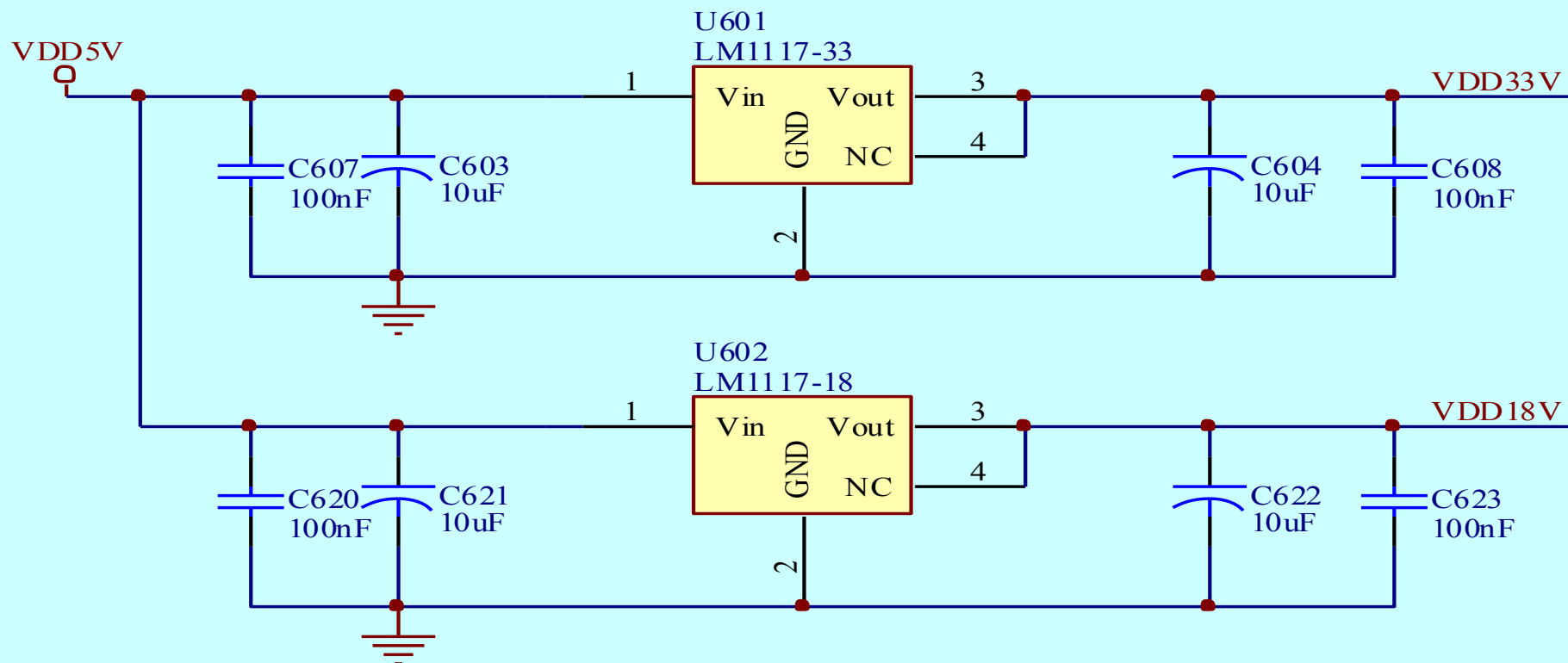
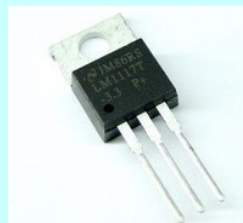
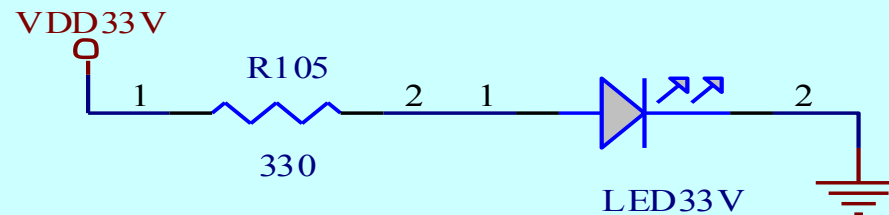
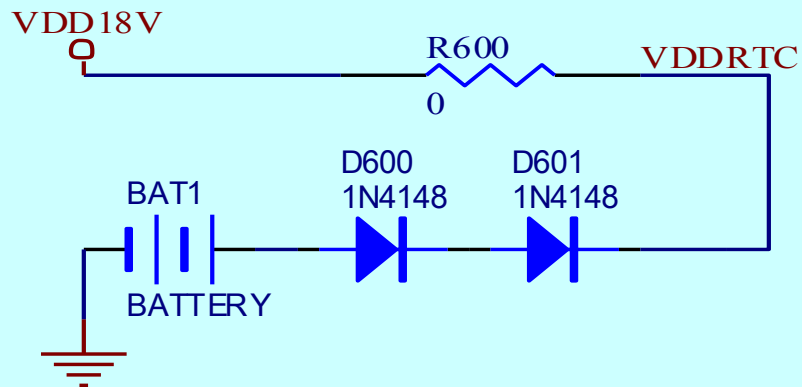
**5V**输入电压经过**DC-DC**转换器可以实现**5V**到**3.3V**和**5V**到**1.8V**的转换。

**DC-DC**模块的输入和输出需加**滤波电容**；

电源设计通常采用**LED**指示灯提示；

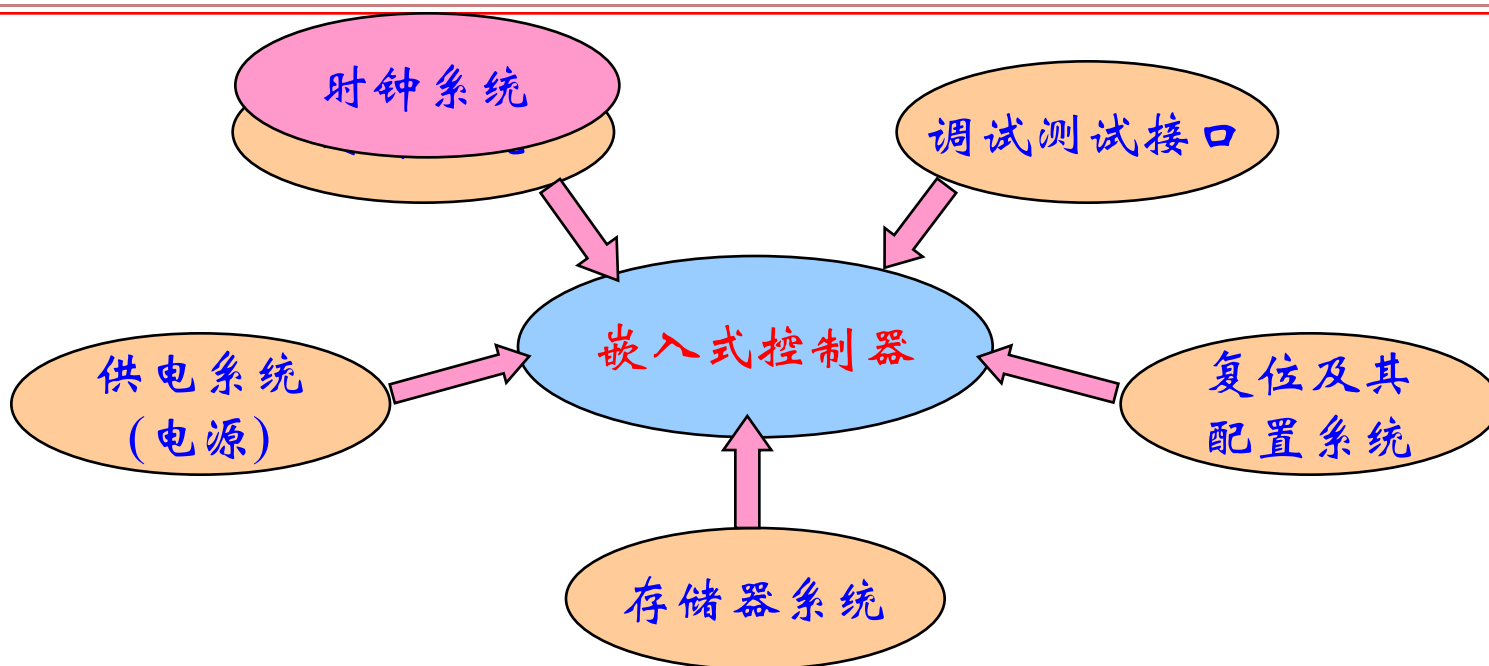
**RTC**模块需要长期供电，即使外部掉电也不能让**RTC**模块停止工作，因此添加**电池**来供电。







## 二、晶振电路设计



目前所有的微处理器均为**时序电路**，需要一个时钟信号才能工作，有些微处理器具有晶体振荡器。简单的方法是利用微处理器**内部的晶体振荡器**，但有些场合需要使用**外部振荡源提供时钟信号**。

---

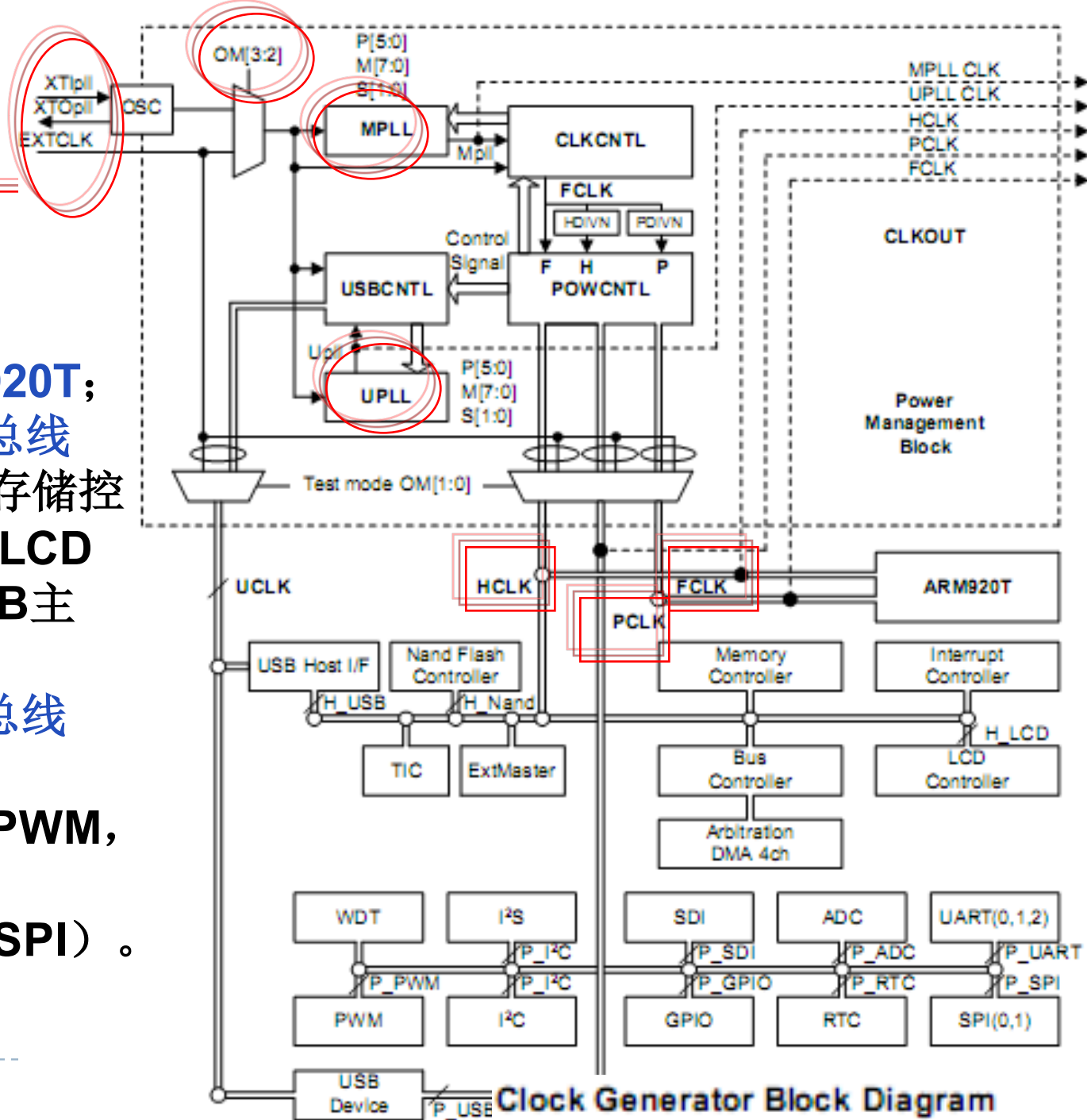
---

**S3C2410**的**时钟控制逻辑**能够产生系统所需要的时钟，包括CPU的**FCLK**，AHB总线接口的**HCLK**，和 APB总线接口的**PCLK**。

**S3C2410A**有两个PLL（**MPLL**和**UPLL**），一个**MPLL**用于 FCLK,HCLK,PCLK，另一个 **UPLL** 用于 **USB** 模块（48MHZ）。



- **FCLK**用于 **ARM920T**;
- **HCLK**用于 **AHB总线**  
(包括 **ARM920T**, 存储控制器, 中断控制器, **LCD** 控制器, **DMA**和 **USB**主机);
- **PCLK** 用于**APB总线**  
(包括外设如 **WDT,IIS,I2C,PWM, PWM, TIMER, MMC, ADC, UART, GPIO, RTC, SPI**) 。



---

---

主时钟源由一个外部晶振或者外部时钟产生。

### 时钟源选择

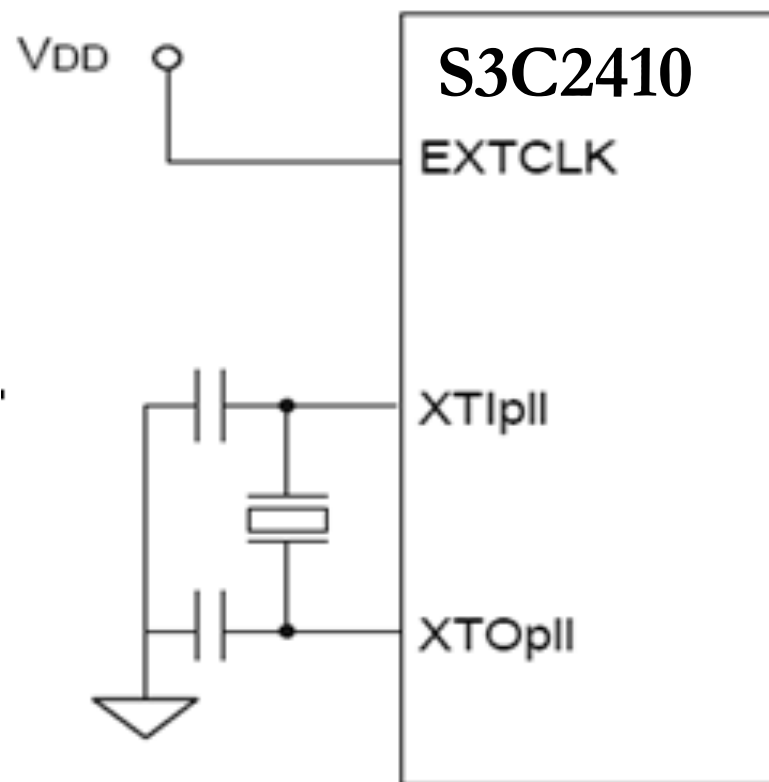
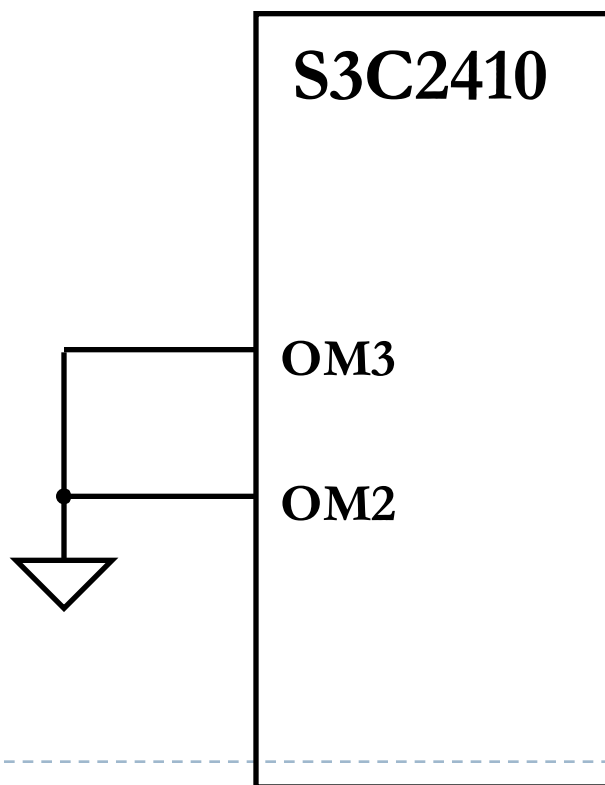
下表 描述了模式控制引脚(**OM3和OM2**)和选择时钟源之间的对应关系。

Mode OM[3:2]	MPLL State	UPLL State	Main Clock source	USB Clock Source
00	On	On	Crystal	Crystal
01	On	On	Crystal	EXTCLK
10	On	On	EXTCLK	Crystal
11	On	On	EXTCLK	EXTCLK

**Clock Source Selection at Boot-Up**

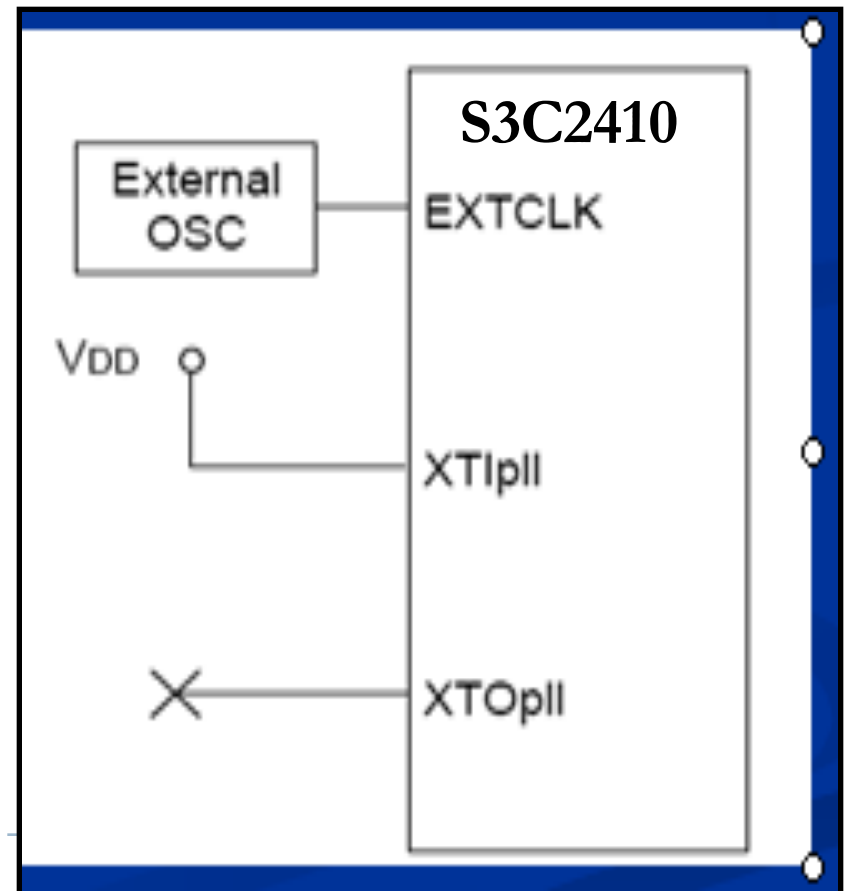
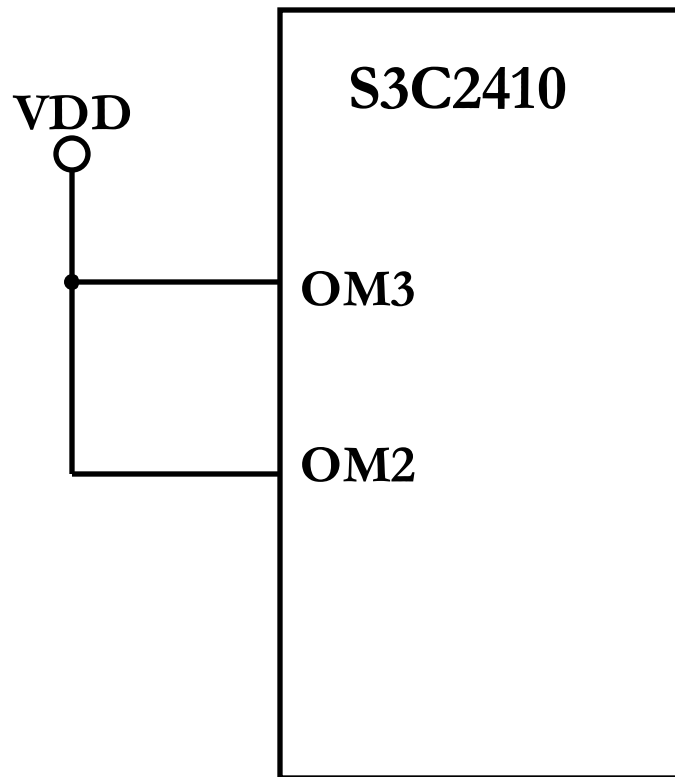


**OM[3:2]=00**时，MPLL和UPLL的时钟均选择外部振荡器；

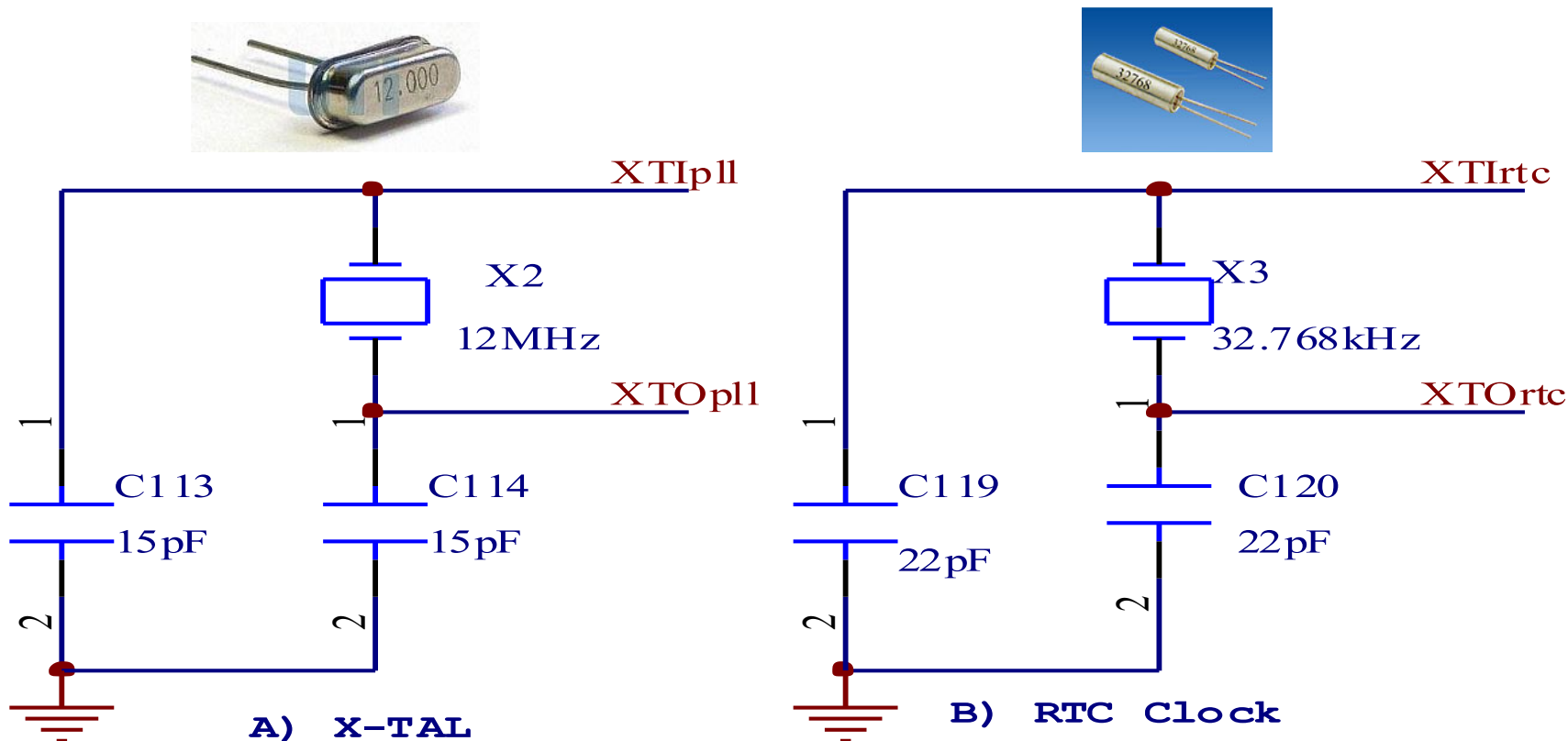




OM[3:2]=11时，MPLL和UPLL的时钟均选择外部时钟源。



## 外部振荡器晶振电路设计



外部振荡器电路有12MHz晶振和2个15pF的微调电容组成，12MHz的时钟信号经过片内的PLL倍频可以达到203MHz。系统所需要的RTC时钟采用相同的方式。

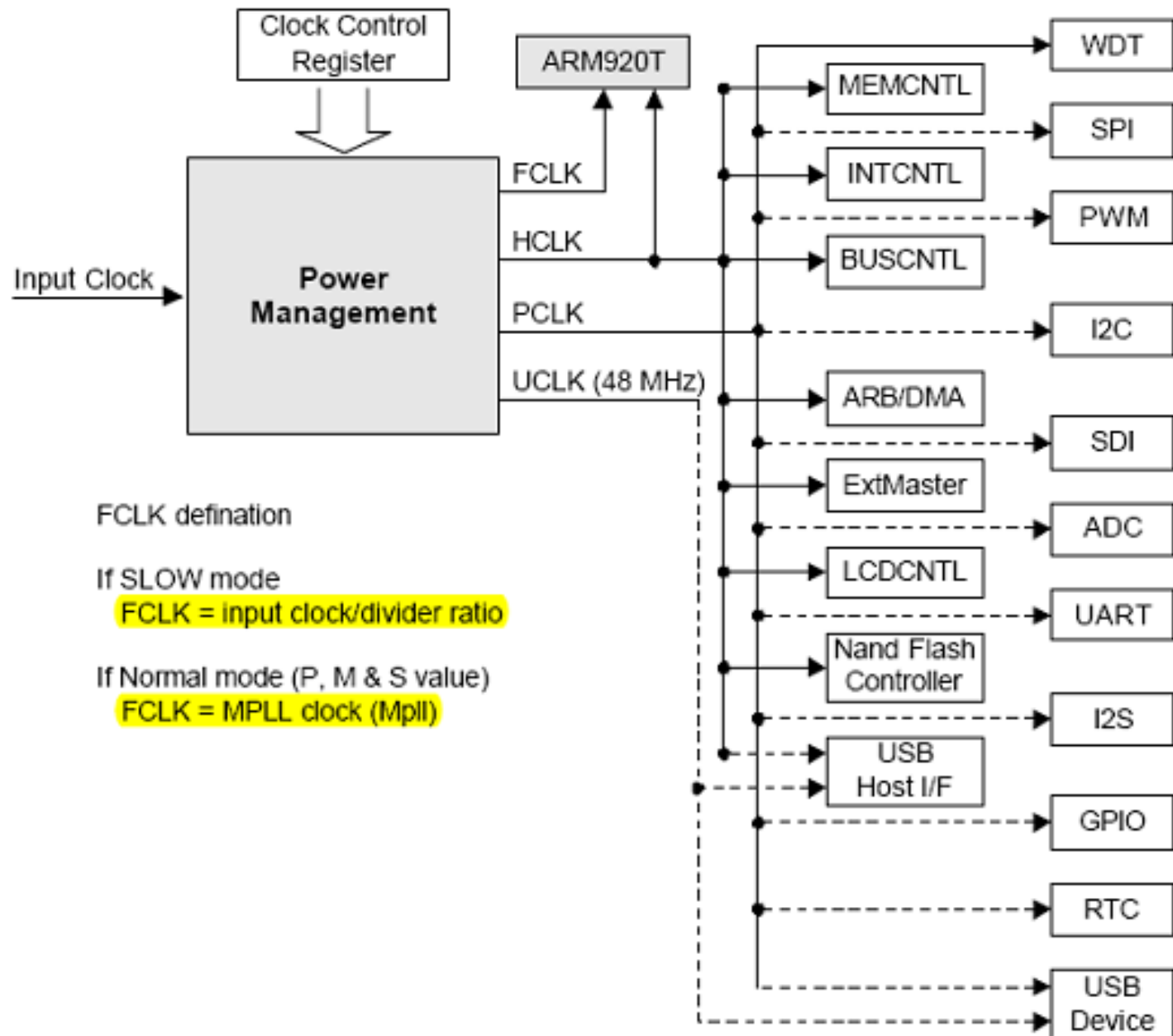
# 系统功率和时钟是密切相关的

---

**S3C2410**有各种针对不同任务提供的最佳功率管理策略，功率管理模块能够使系统工作在如下 4种模式：  
正常模式，低速模式，空闲模式和掉电模式。

在 **S3C2410** 中，功率功率模块通过软件控制系统时钟来达到降低功耗的目的。



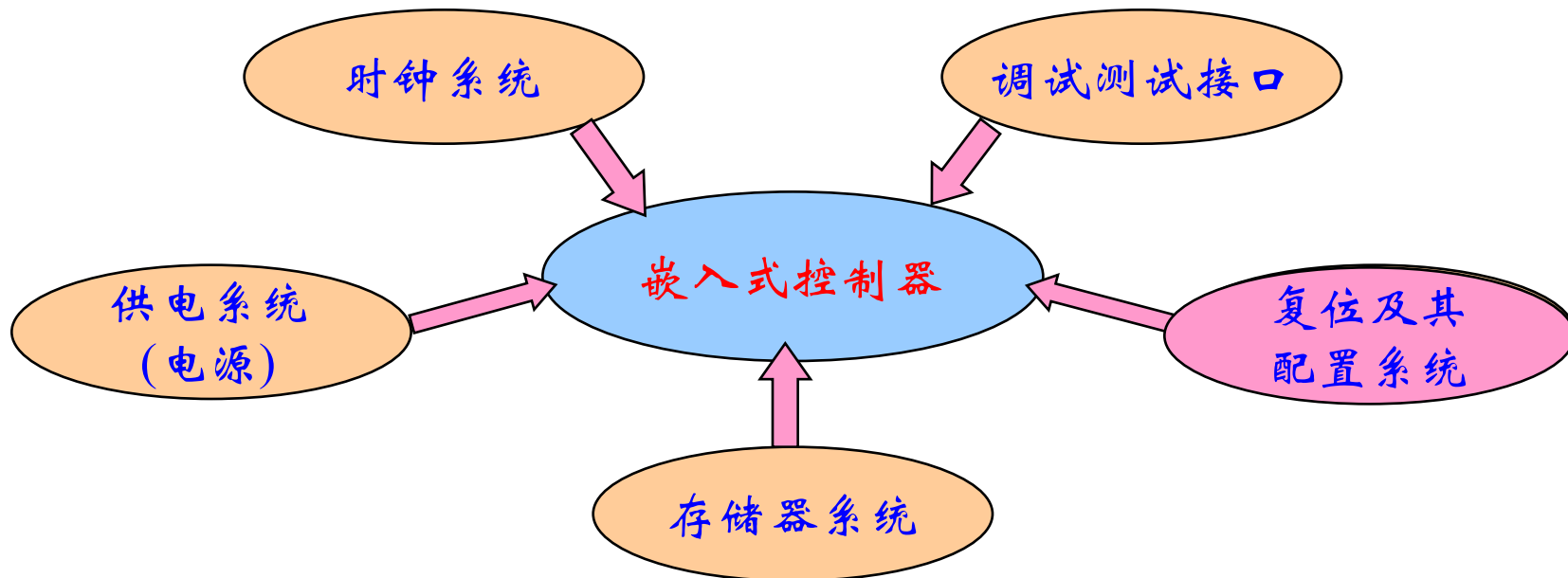


# 时钟控制寄存器 CLKCON

CLKCON	Bit	Description	Initial State
SPI	[18]	Control PCLK into SPI block. 0 = Disable, 1 = Enable	1
IIS	[17]	Control PCLK into IIS block. 0 = Disable, 1 = Enable	1
IIC	[16]	Control PCLK into IIC block. 0 = Disable, 1 = Enable	1
ADC(&Touch Screen)	[15]	Control PCLK into ADC block. 0 = Disable, 1 = Enable	1
RTC	[14]	Control PCLK into RTC control block. Even if this bit is cleared to 0, RTC timer is alive. 0 = Disable, 1 = Enable	1
GPIO	[13]	Control PCLK into GPIO block. 0 = Disable, 1 = Enable	1
UART2	[12]	Control PCLK into UART2 block. 0 = Disable, 1 = Enable	1
UART1	[11]	Control PCLK into UART1 block. 0 = Disable, 1 = Enable	1
UART0	[10]	Control PCLK into UART0 block. 0 = Disable, 1 = Enable	1
SDI	[9]	Control PCLK into SDI interface block. 0 = Disable, 1 = Enable	1
PWMTIMER	[8]	Control PCLK into PWMTIMER block. 0 = Disable, 1 = Enable	1
USB device	[7]	Control PCLK into USB device block. 0 = Disable, 1 = Enable	1
USB host	[6]	Control HCLK into USB host block. 0 = Disable, 1 = Enable	1
LCDC	[5]	Control HCLK into LCDC block. 0 = Disable, 1 = Enable	1
NAND Flash Controller	[4]	Control HCLK into NAND Flash Controller block. 0 = Disable, 1 = Enable	1
POWER-OFF	[3]	Control Power Off mode of S3C2410. 0 = Disable, 1 = Transition to Power_OFF mode	0
IDLE BIT	[2]	Enter IDLE mode. This bit is not cleared automatically. 0 = Disable, 1 = Transition to IDLE mode	0
Reserved	[1]	Reserved	0
SM_BIT	[0]	SPECIAL mode. '0' is recommended normally. This bit can be used to enter SPECIAL mode in only the special condition, OM3=1 & wake-up by nRESET. Please contact us to use this bit.	0

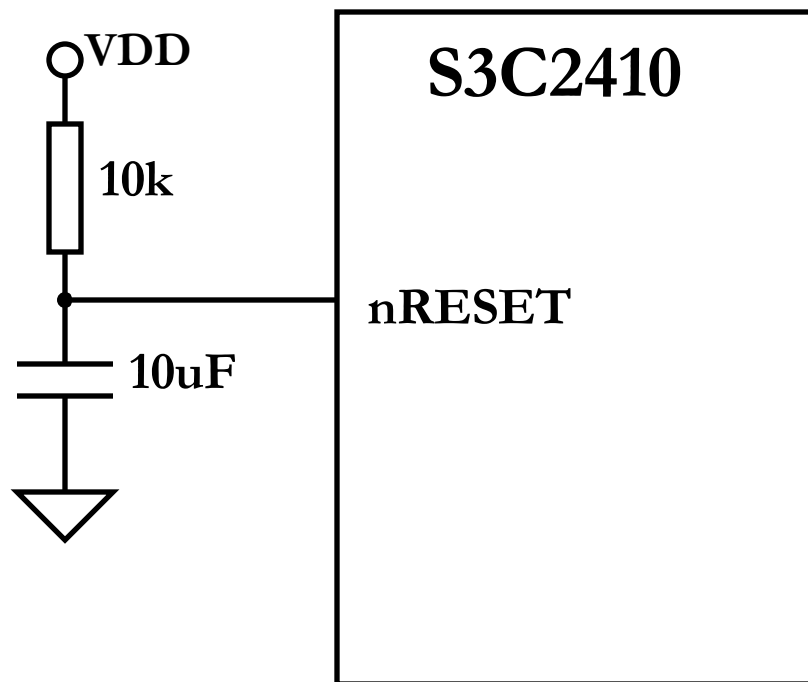
Register	Address	R/W	Description	Reset Value
CLKCON	0x4C00000C	R/W	Clock generator control register	0x7FFF0

### ▶ 三、复位电路设计



微处理器在上电时**状态并不确定**，这造成微处理器不能正确工作。为解决这个问题，所有微处理器均有一个复位逻辑，它**负责将微处理器初始化为某个确定的状态**。这个复位逻辑需要一个复位信号才能工作。大多数微处理器需要外部输入这个信号。

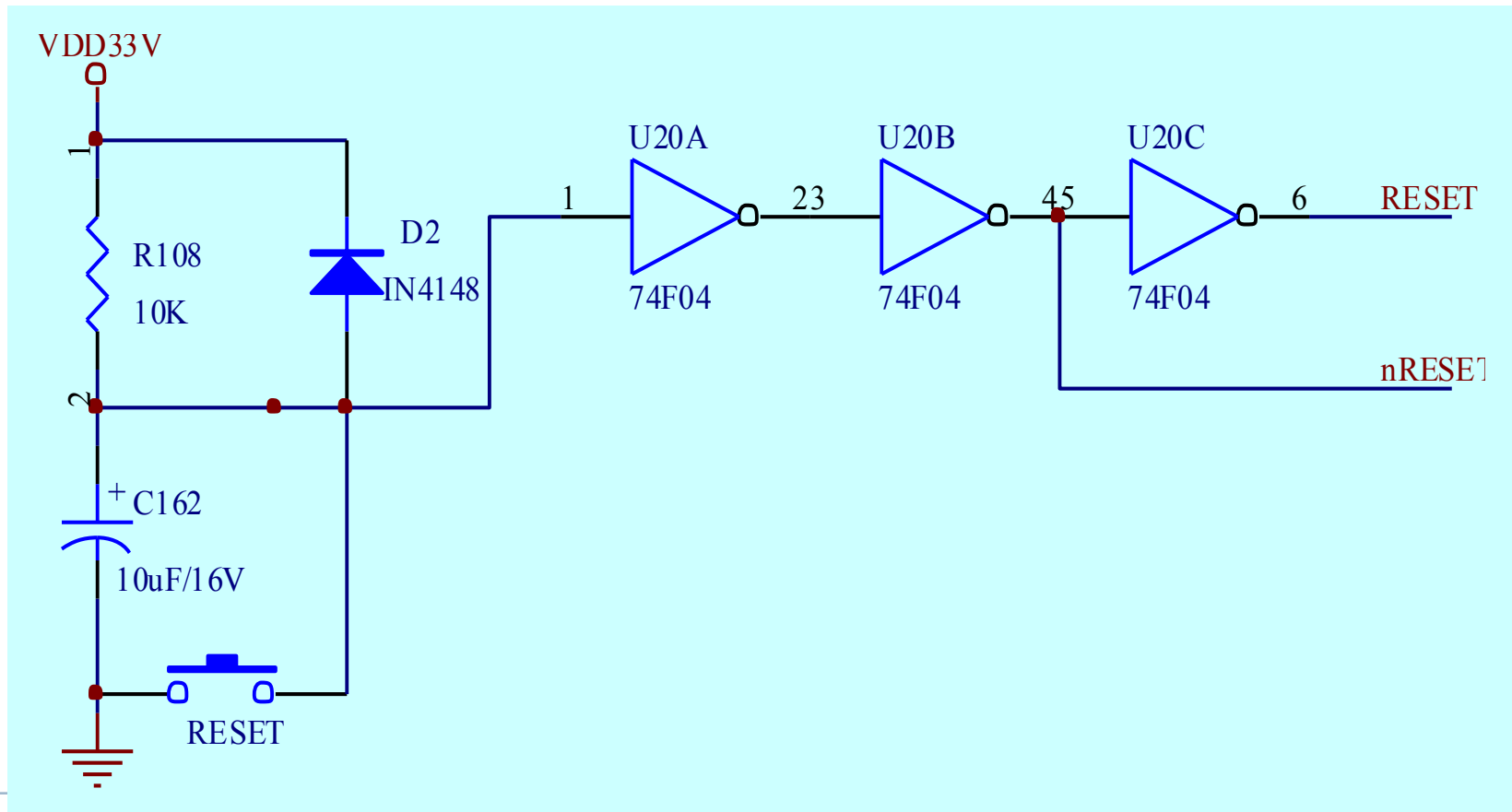
复位电路可以使用简单的**阻容复位 (RC)**，这个电路成本低廉，但不能保证任何情况产生稳定可靠的复位信号，所以一般场合需要使用专门的复位芯片。



阻容复位电路

## 简单实用的复位电路设计

简单实用的复位电路，复位逻辑是比较可靠的。复位电路如图所示：





---

---

该复位电路的**工作原理**如下：在系统上电时，通过电阻R108向电容C162充电，当C162两端的电压未达到高电平的门限电压时，Reset端输出为高电平，系统处于复位状态；当C162两端的电压达到高电平的门限电压时，Reset端输出为低电平，系统进入正常工作状态。

当用户按下按钮RESET时，C162两端的电荷被泻放掉，Reset端输出为高电平，系统进入复位状态，再重复以上的充电过程，系统进入正常工作状态。

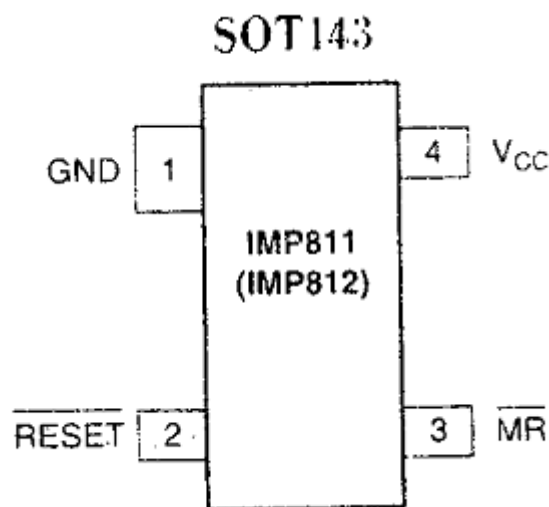
**二极管的作用是：**VDD掉电后，加快电容的放电速度。

---



# 实际工程使用的复位电路设计

**MAX811或IMP811芯片就是常用的复位芯片**，它只有4个管脚。利用该芯片可以同时**上电复位**和**手动复位**，其复位时间不小于140ms。其引脚分布如下：



# 引脚说明

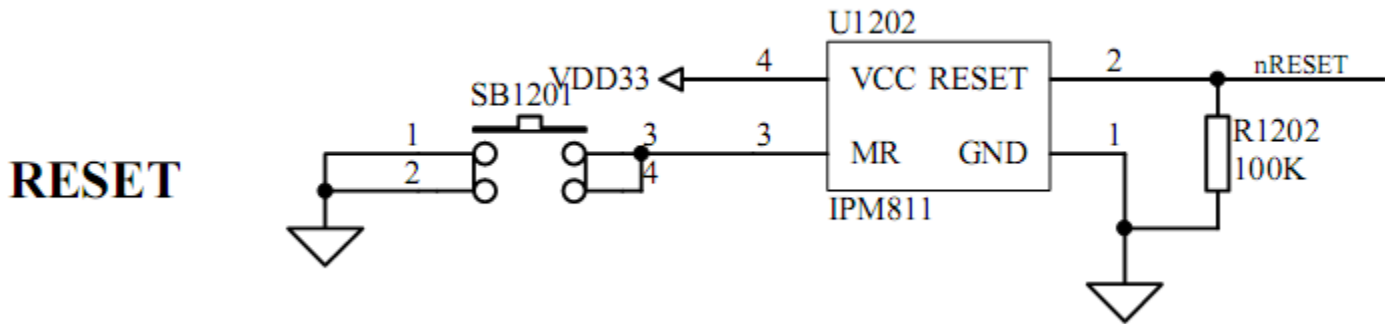
引脚	名称	功能
1	GND	地
2 (IMP811)	$\overline{\text{RESET}}$	当Vcc低于复位门限时，RESET被确定为低电平，并在复位条件中止后保持低电平至少140ms。另外，只要手动复位输入为低电平，那么 $\overline{\text{RESET}}$ 就为低电平有效
2 (IMP812)	RESET	当Vcc低于复位门限时，RESET被确定为高电平，并在复位条件中止后保持高电平至少140ms。另外，只要手动复位输入为低电平，那么RESET就为高电平有效
3	$\overline{\text{MR}}$	手动复位输入。 $\overline{\text{MR}}$ 为逻辑低电平可确定RESET。只要 $\overline{\text{MR}}$ 为低电平，并在 $\overline{\text{MR}}$ 返回高电平之后180ms内，RESET都保持有效。低电平有效输入端具有内部20k $\Omega$ 上拉电阻。如果不用，该输入端应为开路。它可由TTL或CMOS逻辑线驱动或用开关短路到地
4	Vcc	电源输入电压 (3.0V、3.3V、5.0V)



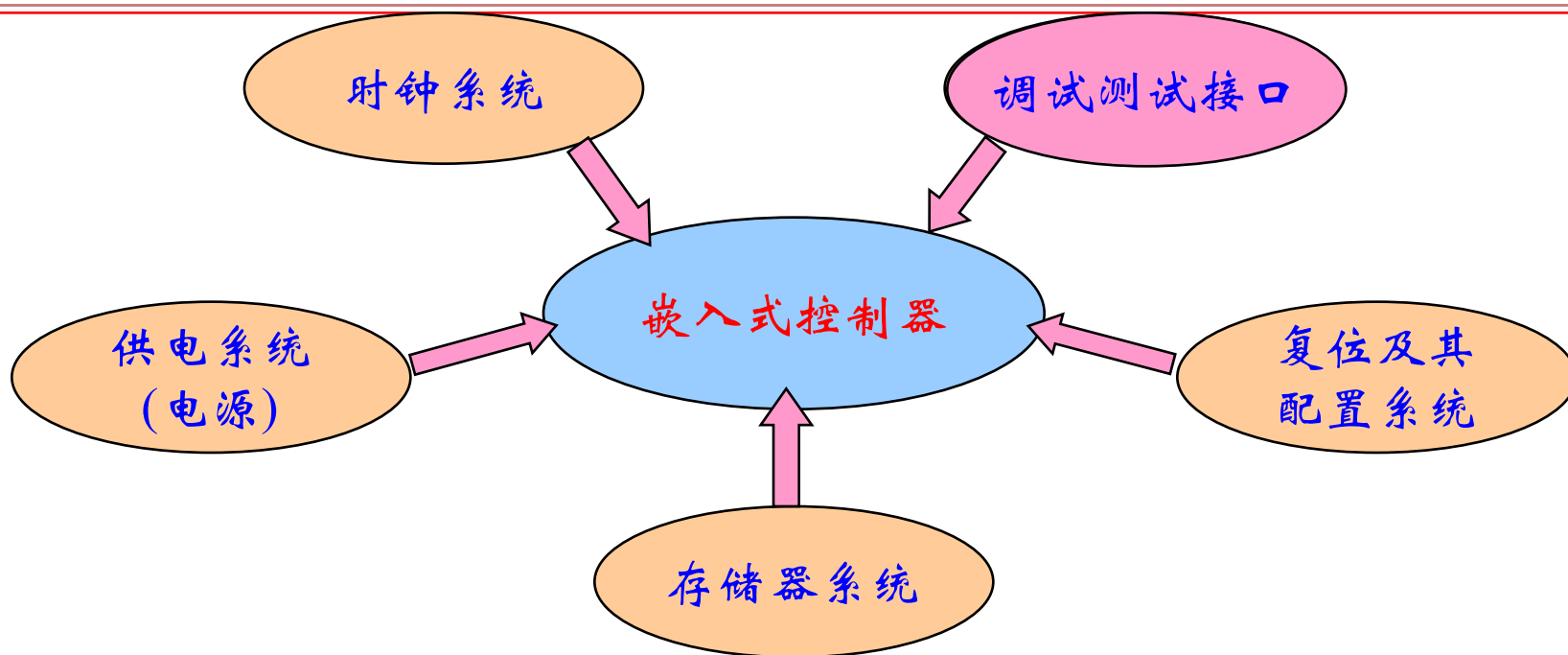
IMP811有六种电压门限以支持3V至5V系统:

复位门限	
后缀	电压 (V)
L	4.63
M	4.38
J	4.00
T	3.08
S	2.93
R	2.63

其应用电路如下:



## ► 四、 JTAG调试接口设计



S3C2410有一个内置JTAG调试接口，通过这个接口可以控制芯片的运行并获取内部信息。

# JTAG概述

---

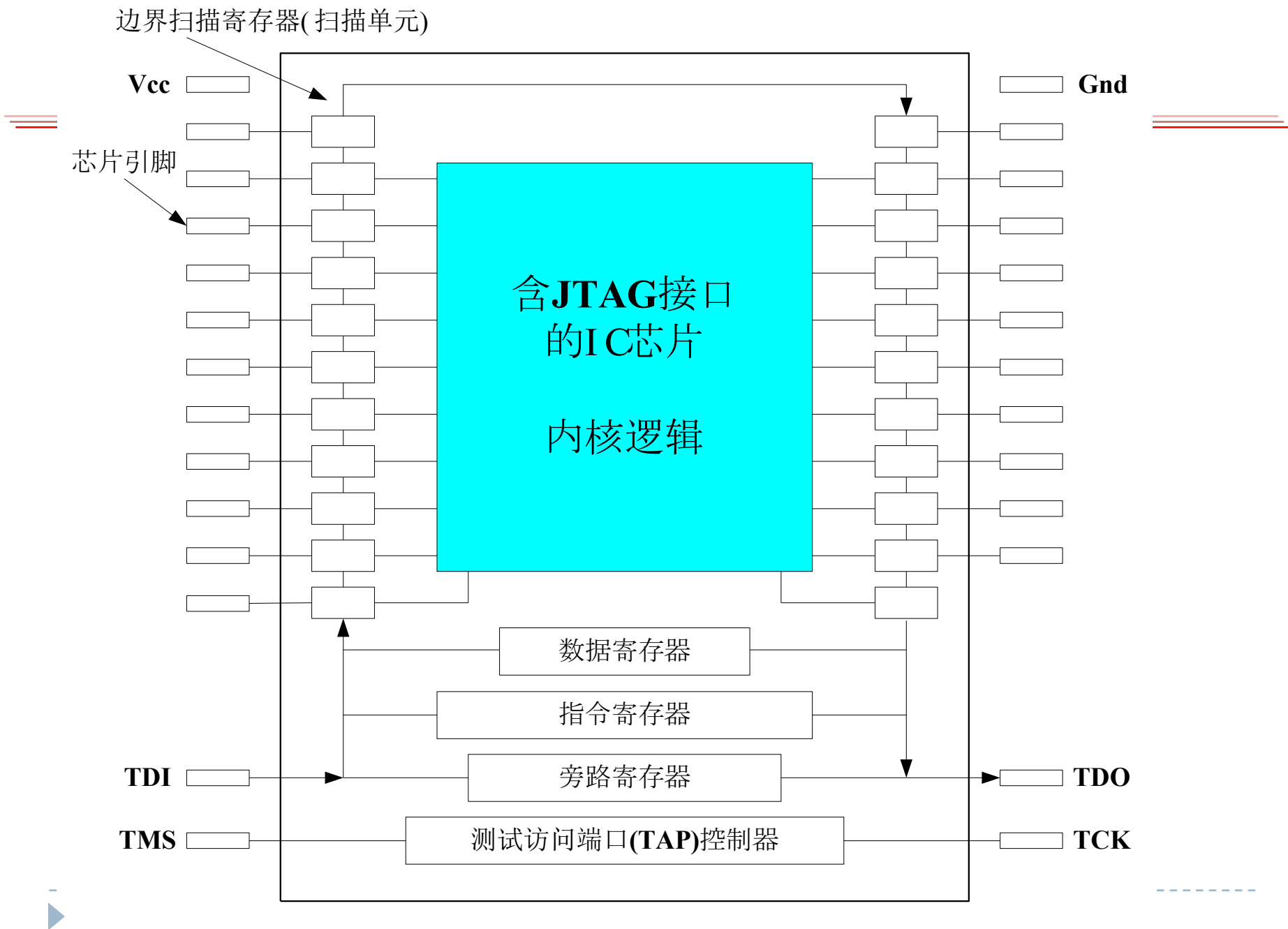
JTAG是Joint Test Action Group（联合测试行动组）的简称。**IEEE1149.1标准**是由JTAG这个组织最初提出来的，最终由IEEE批准并标准化的，所以，IEEE1149.1标准也俗称JTAG标准。

JTAG标准主要用于**芯片内部测试及对系统进行仿真、调试**。在芯片内部封装专门的电路测试访问接口（TAP），通过专用的JTAG测试工具对内部节点进行测试。

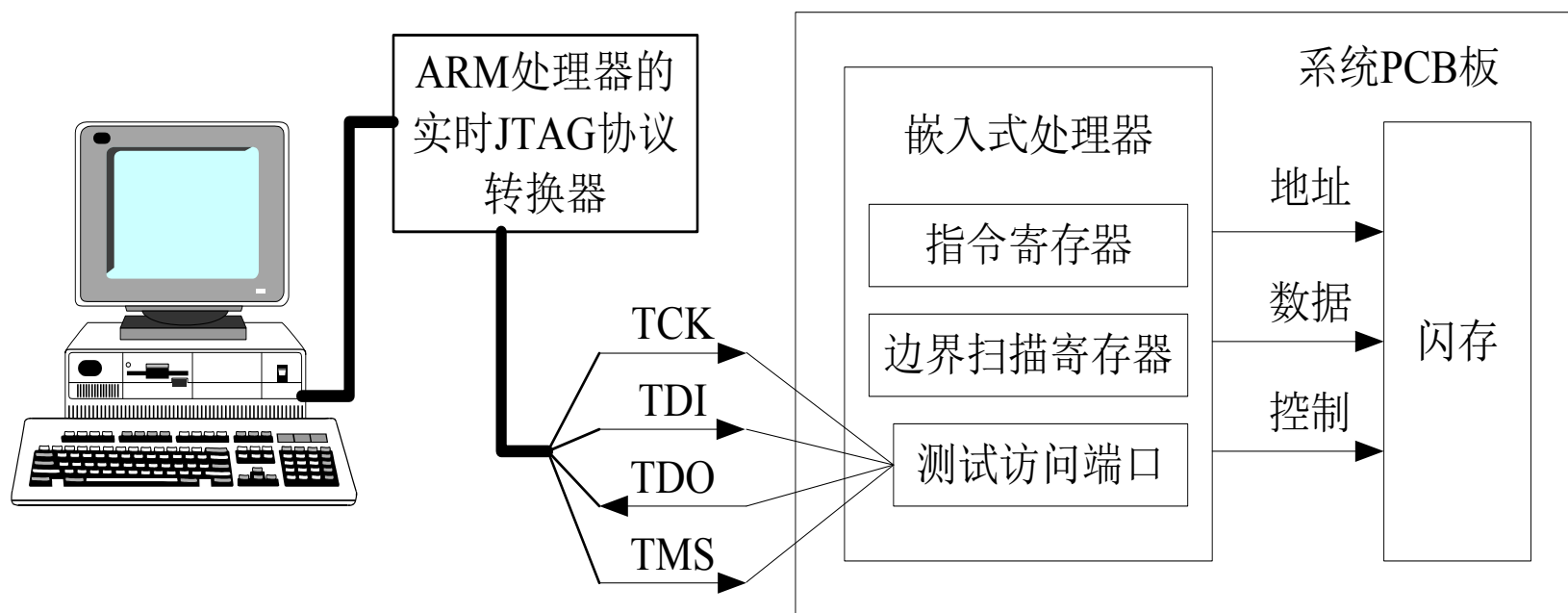
目前，大多数复杂的器件都支持JTAG协议，如ARM、DSP、FPGA等。

---





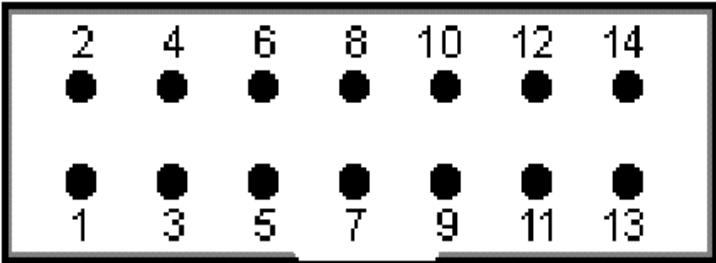
## ARM处理器的实时JTAG仿真器与开发板的连线图





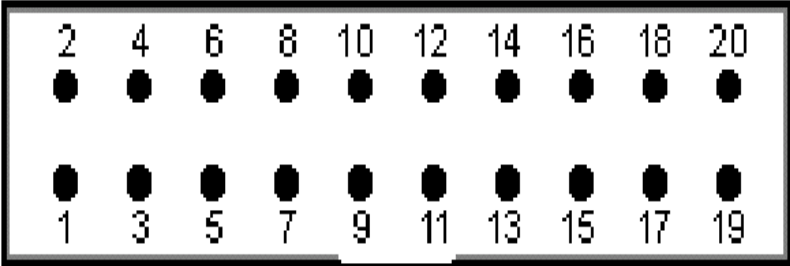
# 接口电路设计

## 1) 14针JTAG接口定义



引 脚	名 称	描 述
1、13	VCC	接电源
3	nTRST	测试系统复位信号
5	TDI	测试数据串行输入
7	TMS	测试模式选择
9	TCK	测试时钟
11	TDO	测试数据串行输出
12	NC	未连接
其他	GND	接地

## 2) 20针JTAG接口定义



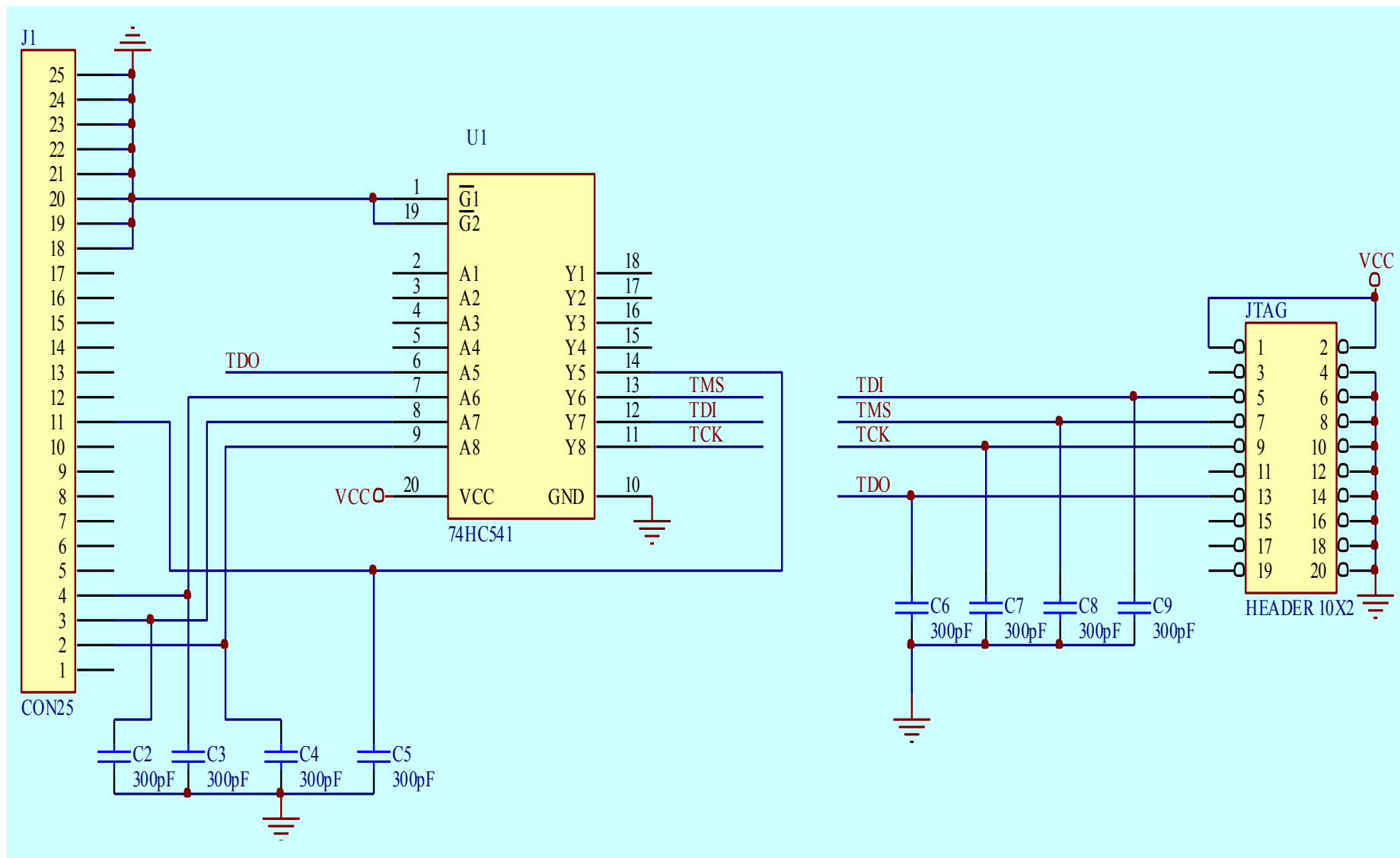
引 脚	名 称	描 述
1	VTref	目标板参考电压，接电源
2	VCC	接电源
3	nTRST	测试系统复位信号
5	TDI	测试数据串行输入
7	TMS	测试模式选择
9	TCK	测试时钟
11	RTCK	测试时钟返回信号
13	TDO	测试数据串行输出
15	nRESET	目标系统复位信号
17、19	NC	未连接
其他	GND	接地

这两类接口之间的信号**电气特性都是一样的**，所以可以把对应的信号直接连接起来进行转化。这意味着即使系统设计是使用了14脚接口，仍可以通过一个简单的电路转换使用采用20脚接口设计的仿真器。

VCC	1	2	VCC
nTRST	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
RTCK	11	12	GND
TDO	13	14	GND
nSRST	15	16	GND
NC	17	18	GND
NC	19	20	GND

VCC	1	2	GND
nTRST	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
TDO	11	12	nSRST
VCC	13	14	GND

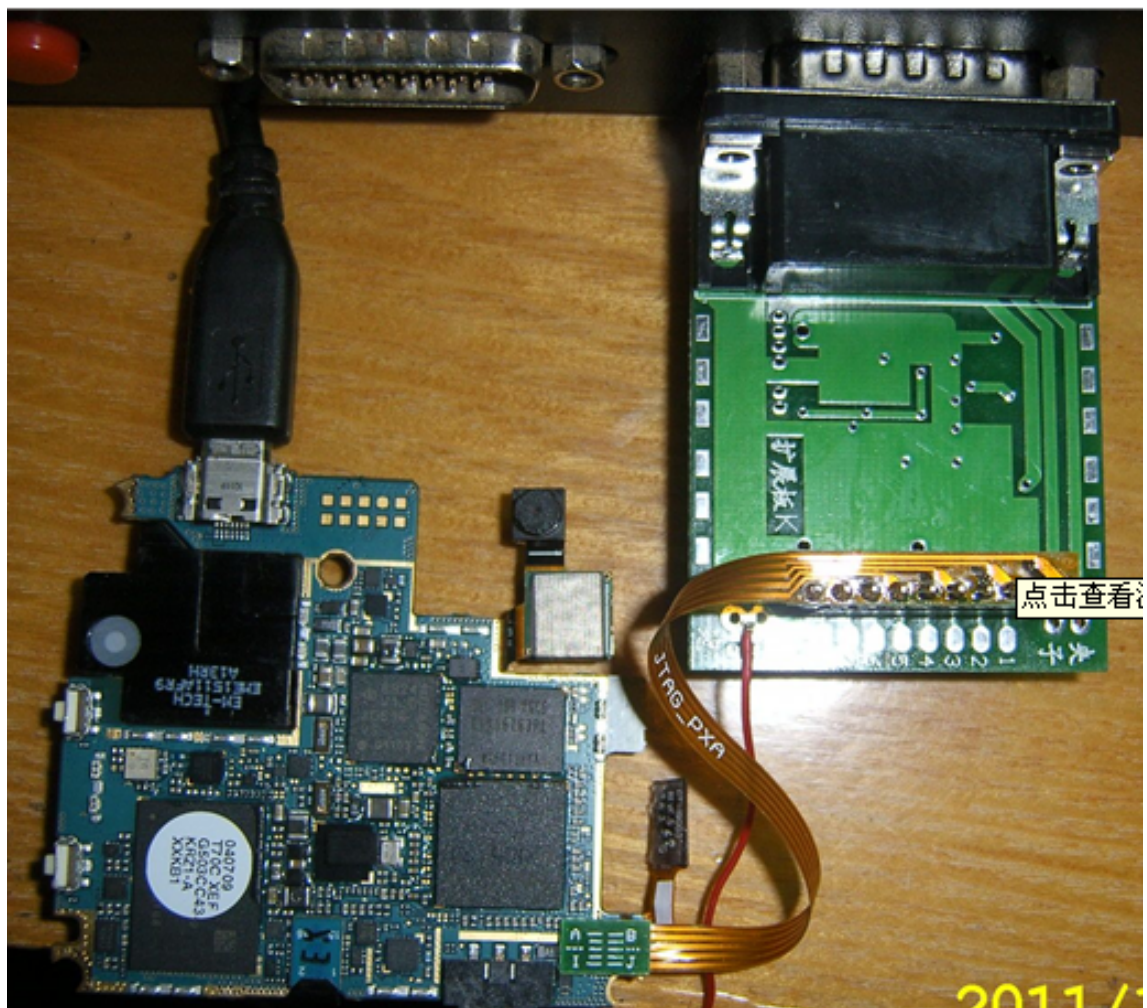
简单的JTAG协议转换接口电路设计如图所示，该电路可作成一小块电路板，包含在下载电缆内，当作JTAG仿真器使用。





**JTAG转换器实物图**

# 调试示意图



# 小结

---

补充了SOC的概念，S3C2410最小系统的组成、功能，以及电源、时钟、复位、JTAG接口的设计。

