

Chapter 01

1. 嵌入式系统一般定义 (PDF 25)

以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统

注： 也称“嵌入到对象体系中的专用计算机系统”

嵌入式系统的 3 个基本要素：嵌入性，专用性，计算机系统

对象系统：嵌入式系统所嵌入的宿主系统

2. 嵌入式系统组成 (PDF 31)

嵌入式硬件：嵌入式微处理器，外围硬件设备

软件：嵌入式操作系统，嵌入式应用软件

3. 通用计算机与嵌入式系统对比 (PDF 27-28)

特征	通用计算机	嵌入式系统
形式和类型	<ul style="list-style-type: none">看得见的计算机按其体系结构、运算速度和结构规模等因素分为大、中、小型机和微机	<ul style="list-style-type: none">看不见的计算机形式多样，应用领域广泛，按应用来分
技术要求	<ul style="list-style-type: none">高速、海量的数值计算	<ul style="list-style-type: none">对象的智能化控制能力
技术发展方向	<ul style="list-style-type: none">总线速度的无限提升存储容量的无限扩大	<ul style="list-style-type: none">与对象系统密切相关的嵌入性能、控制能力与控制的可靠性
组成	<ul style="list-style-type: none">通用处理器、标准总线和外设软件和硬件相对独立	<ul style="list-style-type: none">面向应用的嵌入式微处理器，总线和外部接口多集成在处理器内部软件与硬件是紧密集成在一起的

特征	通用计算机	嵌入式系统
资源情况	<ul style="list-style-type: none">资源不受限	<ul style="list-style-type: none">资源受限
开发方式	<ul style="list-style-type: none">开发平台和运行平台都是通用计算机	<ul style="list-style-type: none">采用交叉开发方式，开发平台一般是通用计算机，运行平台是嵌入式系统
二次开发性	<ul style="list-style-type: none">应用程序可重新编制	<ul style="list-style-type: none">一般不能再编程
对代码质量的要求	<ul style="list-style-type: none">没有特别的要求，在存储空间上也没有太多的限制	<ul style="list-style-type: none">要求高，要求可执行程序占用的存储空间少
对稳定性和性能要求	<ul style="list-style-type: none">在稳定性和性能方面的要求没有嵌入式系统那么严格	<ul style="list-style-type: none">由于多任务同时运行，产生的错误比较多，因此在稳定性和性能方面的要求特别高

4. 嵌入式系统与单片机对比 (PDF 34)

特征	单片机	嵌入式系统
体系结构	<ul style="list-style-type: none">• 多为4位、8位、16位机，不适合运行操作系统，难以进行复杂的运算及处理功能	<ul style="list-style-type: none">• 主流是以32位嵌入式微处理器为核心的硬件设计和基于实时操作系统(RTOS)的软件设计
设计方法	<ul style="list-style-type: none">• 大多采用软硬件流水设计	<ul style="list-style-type: none">• 强调基于平台的设计• 软硬件协同设计
设计核心与工作量	<ul style="list-style-type: none">• 软硬件设计所占比例基本相同	<ul style="list-style-type: none">• 核心是软件设计工作，占70%左右的工作量
学习方法及途径	<ul style="list-style-type: none">• 一般从硬件入手，从硬件体系结构、汇编语言到硬件设计、软件设计	<ul style="list-style-type: none">• 可以从软件入手，从应用层编程到操作系统移植、硬件平台设计较好，按单片机设计的学习流程较难掌握

5. 嵌入式系统的特点 (PDF 50)

- 1) 嵌入式系统通常是形式多样、面向特定应用的
- 2) 嵌入式系统得到多种类型的处理器和处理器体系结构的支持
- 3) 嵌入式系统通常极其关注成本
- 4) 嵌入式系统有实时性和可靠性的要求
- 5) 嵌入式系统使用的操作系统一般是适应多种处理器、可剪裁、轻量型、实时可靠、可固化的嵌入式操作系统
- 6) 嵌入式系统开发需要专门工具和特殊方法

6. 嵌入式系统的分类

1) 按系统响应时间分类 (PDF 62)

- i. **强实时系统**，其系统响应时间在毫秒或微秒级
- ii. **一般实时系统**，其系统响应时间在几秒的数量级上，其实时性的要求比强实时系统要差一些
- iii. **弱实时系统**，其系统响应时间约为数十秒或更长。这种系统的响应时间可能随系统负载的轻重而变化

2) 按确定性来分类 (PDF 63)

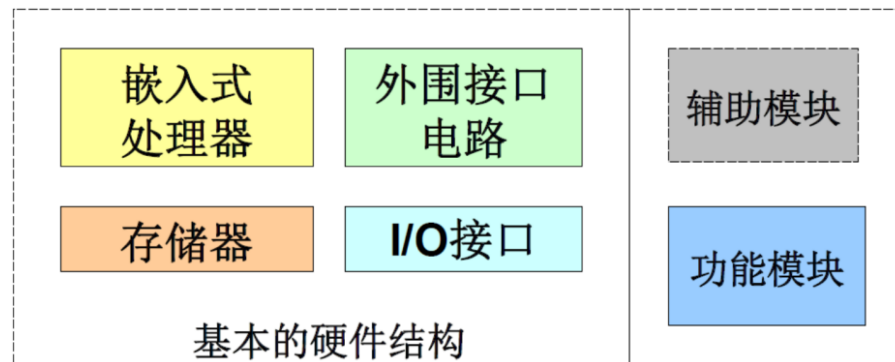
- i. **硬实时**：系统对系统响应时间有严格的要求，如果系统响应时间不能满足，就要引起系统崩溃或致命的错误

- ii. **软实时**: 系统对系统响应时间有要求, 但是如果系统响应时间不能满足, 不会导致系统出现致命的错误或崩溃

7.

Chapter 02

1. 嵌入式系统硬件体系结构 (PDF 14)



2. 什么是嵌入式处理器、主要特点 (PDF 16)

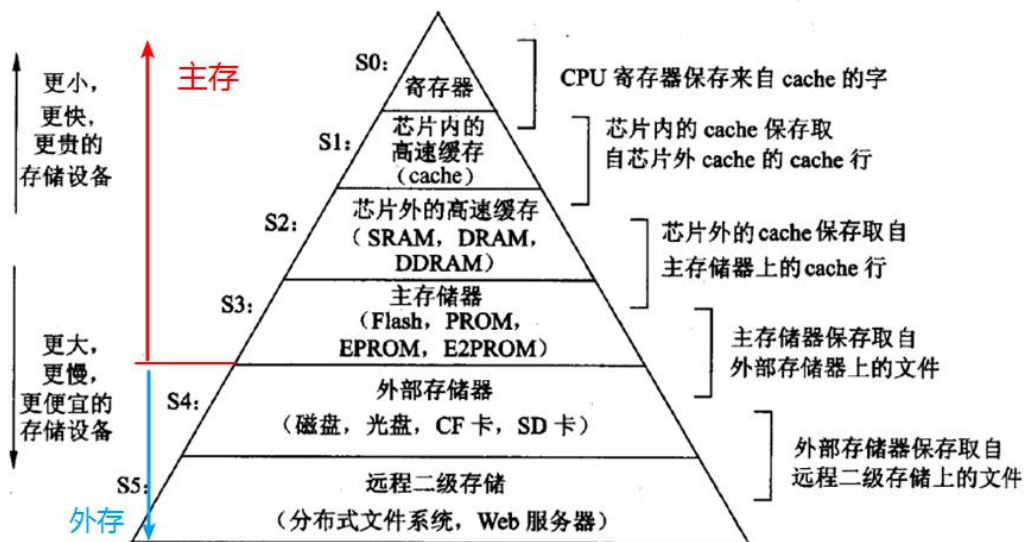
- 1) **嵌入式处理器**: 嵌入式处理器是指应用在嵌入式系统中的微处理器
- 2) **特点**: 品种多, 体积小, 成本低, 集成度高

3. 嵌入式处理器分类 (PDF 18)

- 1) **MPU** (Micro Processor Unit, 微处理器)
- 2) **MCU** (Micro Controller Unit, 微控制器)
- 3) **DSP** (Digital Signal Processor, 数字信号处理器)
- 4) **SoC** (System on Chip, 片上系统)

4. 存储器系统的层次结构 (PDF 21)

存储器系统的层次结构



5. 主流易失性存储器与非易失性存储器 (PDF 19)

1) 易失性存储器:

- i. **SRAM** (静态存储器), 速度快, 价格高, Cache
- ii. **DRAM** (动态存储器), 内存

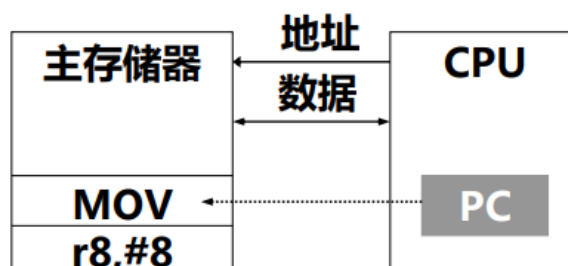
2) 非易失性存储器: ROM、EPROM、EEPROM、FLASH

6.

Chapter 03

7. 冯·诺依曼结构 与 哈佛结构 (PDF 6-8)

1) 冯·诺依曼结构:

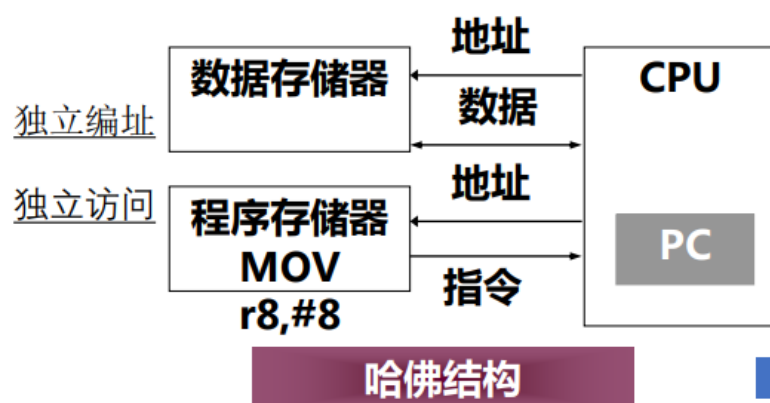


冯·诺依曼结构

- i. 程序指令和数据共用一个存储空间
- ii. 程序指令存储地址和数据存储地址指向同一个存储器的不同物理位置

- iii. 采用单一的地址及数据总线，程序指令和数据的宽度相同
- iv. 处理器执行指令时，先从存储器中取出指令解码，再取操作数执行运算，即使单条指令也要耗费几个甚至几十个周期，在高速运算时，在传输通道上会出现瓶颈效应
- v. Intel, ARM7, MIPS 均采用这种架构

2) 哈佛结构:

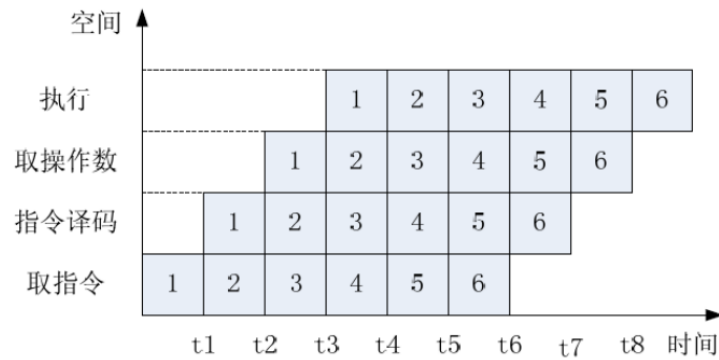


- i. 程序和数据存储器在两个分开的物理空间中，取指令和执行能完全重叠
- ii. 允许在一个机器周期内同时获取指令字和操作数，从而提高了执行速度，使数据吞吐率提高 1 倍
- iii. 所有的 DSP 处理器, ATME1 公司的 AVR 系列, ARM9、ARM10 和 ARM11

8. 复杂指令集系统 CISC 与精简指令集系统 RISC 区别 (PDF 17)

类别	CISC	RISC
指令系统	指令数量很多	较少，通常少于100
执行时间	有些指令执行时间很长，如整块的存储器内容复制；或将多个寄存器的内容复制到存储器	没有较长执行时间的指令
编码长度	编码长度可变，1~15字节	编码长度固定，通常为4个字节
寻址方式	寻址方式多样	简单寻址
操作	可以对存储器和寄存器进行算术和逻辑操作	只能对寄存器进行算术和逻辑操作，Load / Store体系结构
编译	难以用优化编译器生成高效的目标代码程序	采用优化编译技术，生成高效的目标代码程序

9. 解释流水线技术的概念和核心思想 (PDF 22)



1) **概念:** 流水线技术是将一个重复的时序分解成若干个子过程, 而每一个子过程都可有效地在其专用功能段上与其他子过程同时执行

2) **核心思想:**

- i. 将一条指令分解成一连串执行的子过程
- ii. 在 CPU 中把一条指令的串行执行子过程, 变为若干条指令的子过程的重叠执行

10. 说明数据存储格式

1) **小端模式 (little endian) (PDF 33)**

- i. 低字节数据存放在内存低地址处 (低—低)
- ii. 高字节数据存放在内存高地址处 (高一高)

2) **大端模式 (big endian) (PDF 35)**

- i. 高字节数据存放在内存低地址处 (高一低)
- ii. 低字节数据存放在内存高地址处 (低—高)

11. 嵌入式微处理器的特点 (PDF 47)

- 1) 体积小、重量轻、成本低、可靠性高
- 2) 功耗低
- 3) 工作温度、抗电磁干扰、可靠性等方面增强

12. 主流嵌入式微处理器

目前主流的嵌入式微处理器系列主要有 ARM 系列、MIPS 系列、PowerPC 系列、SuperH 系列和 X86 系列等

13.

Chapter 04

1. ARM 含义、ARM 处理器特点 (PDF 15, 19)

- 1) 含义: 一个公司的名字, 一类微处理器的通称, 一种技术的名字
- 2) 特点: 体积小、低功耗、低成本、高性能

2. Cortex 处理器的分类与应用领域 (PDF 30)

- 1) **Cortex-A** (Application): 用于高性能的开放应用平台, 如平板电脑、智能手机; Cortex-A8、A9、A15
- 2) **Cortex-R** (Real-time): 用于高端、对实时性有要求的嵌入式系统, 如汽车 ABS 系统, 交换机, 航空系统; Cortex-R4
- 3) **Cortex-M** (MCU): 用于对成本敏感、深度嵌入的嵌入式系统, 智能家电; Cortex-M3

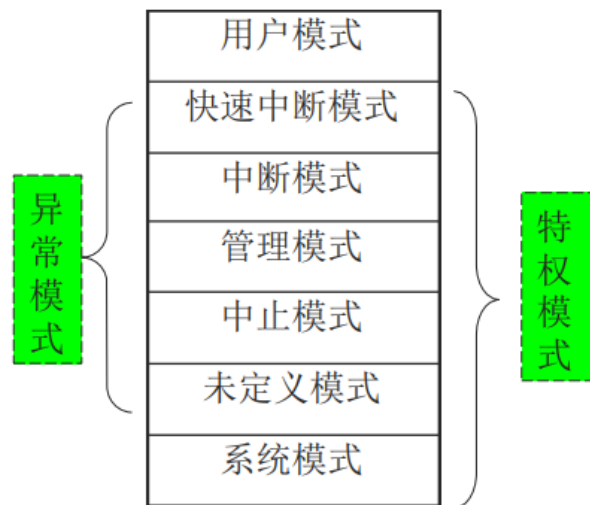
注: Cortex 即 ARMv7

3. ARM 内核后缀命名含义与扩展支持 (PDF 37,38)

ARM {x}{y}{z} {T}{D}{M}{I}{E}{J}{F}{-S}

- 1) **x**: 系列编号, ARM7/ARM9/ARM11...
- 2) **y**: 内部存储管理 MMU 和保护单元 MPU
- 3) **z**: 片内高速缓存情况
- 4) **ARM v4**:
 - i. T -- 支持 16bit Thumb 指令集
 - ii. D -- 片上调试 JTAG
 - iii. M -- 支持快速乘法
 - iv. I - 嵌入式跟踪宏单元 Embedded ICE
- 5) **ARM v5**:
 - i. E -- 增强型 DSP 指令集
 - ii. J -- Java 硬件加速器-Jazelle
 - iii. F -- 向量浮点运算单元
- 6) **ARM v6**:
 - i. S - 提供可综合的 VHDL 文件
 - ii. Z - 安全处理技术 TrustZone
- 7) 如: ARM1156T2, ARM1176JZ...

4. ARM 处理器的工作模式 (PDF 91)



5. ARM 的两种工作状态 (PDF 98,97)

- 1) 当 ARM 微处理器执行 32 位的 ARM 指令集时，工作在 ARM 状态
- 2) 当 ARM 微处理器执行 16 位的 Thumb 指令集时，工作在 Thumb 状态
- 3) ARM 微处理器在开始执行代码时，处于 ARM 状态
- 4) 在程序的执行过程中，微处理器可以随时在两种工作状态之间切换
- 5) 处理器工作状态的转变并不影响处理器的工作模式和相应寄存器中的内容

6. ARM 的寄存器组织 (PDF 102, 118)

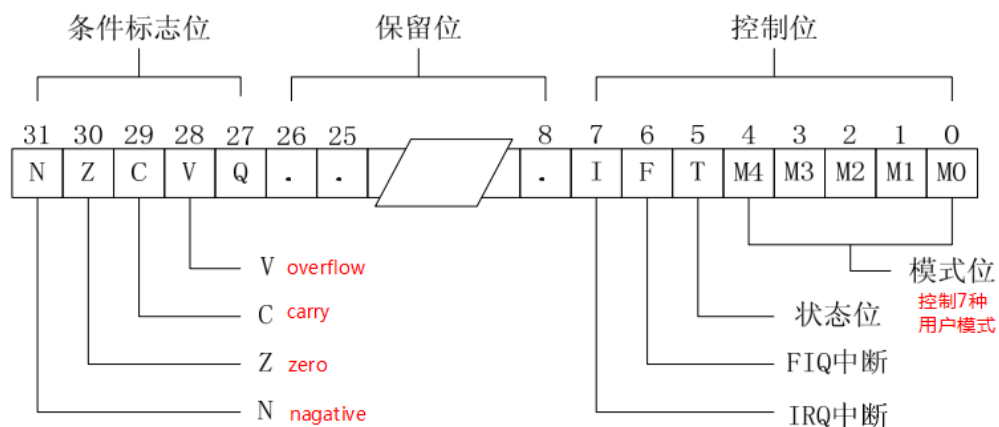
ARM 有 37 个 32 位长的寄存器:



1) 通用寄存器 (X 31)

- 未分组寄存器 R0~R7
- 分组寄存器 R8~R14
- 程序计数器 PC (R15)

2) 程序状态寄存器 (X 6)



- 当前程序状态寄存器 **CPSR** (X 1) (current program status register)
- 备份的程序状态寄存器 **SPSR** (X 5) (saved program status register)

7. 解释异常的概念和异常处理过程 (PDF 127)

- 异常**: 由内部或外部源产生并引起处理器处理的一个事件

2) 异常处理过程:

- i. 当异常中断发生时，程序执行完当前指令后，根据引起异常的模式，转去相应的异常中断处理程序处执行
- ii. 在处理异常之前，处理器的状态必须保留，以便在异常处理完后，能够重新执行原来的程序

8. 异常中断类型与中断向量地址 (PDF 129)

异常中断类型	异常中断模式	向量地址	优先级（1最高）
复位	管理模式	0x00000000	1
未定义指令	未定义模式	0x00000004	6
软件中断（SWI）	管理模式	0x00000008	6
指令预取中止	中止模式	0x0000000C	5
数据访问中止	中止模式	0x00000010	2
保留		0x00000014	
外部中断请求IRQ	IRQ模式	0x00000018	4
快速中断请求FIQ	FIQ模式	0x0000001C	3

9. 中断向量表: (PDF 130)

- 1) 各中断向量地址组成异常中断向量表，中断向量表指定了各异常中断及其处理程序的对应关系，它通常存放在存储地址的低端
- 2) 在 ARM 体系中，中断向量表的大小为 **32 字节**。其中每个异常中断占据 4 个字节大小，保留了 4 个字节空间

10. 异常中断的优先级: (PDF 132)

- 1) 当几个异常中断同时发生时，就必须按照一定的次序来处理这些异常中断，这就是异常中断的优先级
- 2) 在 ARM 中通过给各异常中断赋予一定的优先级序号来实现这种处理次序

11.

Chapter 05

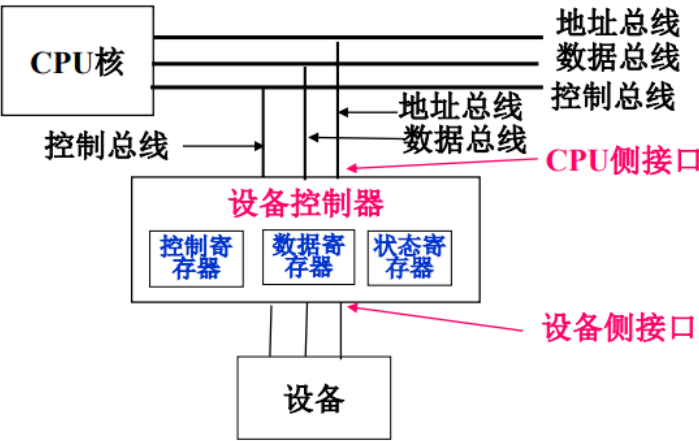
1. S3C2410 的概况 (PDF 5)

S3C2410 是 Samsung 公司推出的 16/32 位 RISC 处理器，主要面向手持设备以

及高性价比、低功耗的应用。CPU 内核采用 ARM920T 处理器

2. 设备控制器 (PDF 7)

- 1) 概念: 设备控制器是 CPU 核与外部设备间的桥接器, 在 CPU 侧与设备侧各有一个接口
- 2) 组成结构:



- 3) 寄存器:
 - i. 数据寄存器: 用于存储 CPU 核发往设备的数据或设备向 CPU 核提供的数据
 - ii. 控制寄存器: 用于接收 CPU 核发往设备的命令, 并对命令进行译码
 - iii. 状态寄存器: 用于存储设备的状态, 供 CPU 核读取

3.

Chapter 06

1. 线性稳压器 vs 开关稳压器 (PDF 12, 13)

	线性稳压器	开关稳压器
优	不在电路板上产生噪声 元件数量少 价格不高	效率更高 不产生那么多的热量 从任意输入电压产生任意输出电压
劣	效率不高 产生热量 不能输出比输入高的电压	噪声大 元件数量多

2. 多功能监控电路的功能 (PDF 25)

- 1) **电源测控**: 供电电压出现异常时提供预警指示或中断请求信号,
- 2) **方便系统实现异常处理**
- 3) **数据保护**: 当电源或系统工作异常时 对数据进行必要的保护, 如
- 4) **写保护、数据备份或切换后备电池**
- 5) **看门狗定时器**: 当系统程序跑飞或死锁时, 复位系统
- 6) 其它的功能: 如**温度测控、短路测试**等

3. JTAG 测试技术概念、用途、接口信号 (PDF 21)

- 1) JTAG (Joint Test Action Group, 联合测试行动小组) 是一种国际标准测试协议, 主要用于芯片内部测试、系统仿真与调试及 **bootloader** 的下载
- 2) JTAG 接口信号
 - i. **nTRST**: (Optional) JTAG 复位信号, 复位 JTAG 的状态机和内部的宏单元(Test Reset)
 - ii. **TMS**: 测试模式选择(Test Mode Select), 通过 TMS 信号控制 JTAG 状态机的状态
 - iii. **TCK**: JTAG 的时钟信号(Test Clock)
 - iv. **TDI**: 数据输入信号(Test Data-In)
 - v. **TDO**: 数据输出信号(Test Data-Out)
 - vi. **VCC**: 接电源
 - vii. **GND**: 接地线
 - viii. **NC**: 未连接

4. 4 种电源管理模式 (PDF 43)

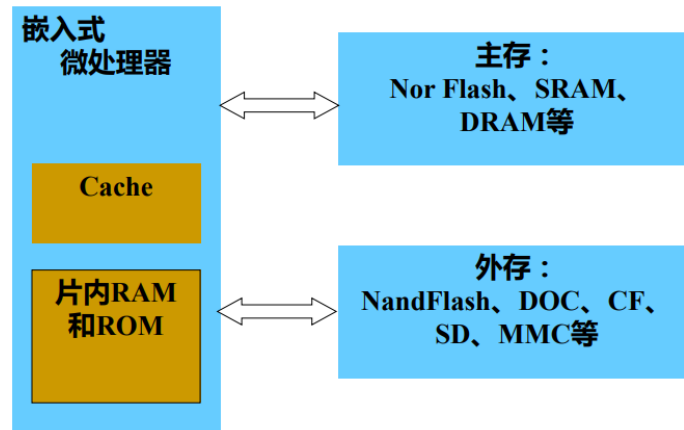
- 1) **正常模式**: 锁相环工作; 为 CPU 和所有片内外设提供时钟。
此模式功耗最大
- 2) **慢时钟模式**: 锁相环不工作,CPU 等直接使用原始时钟或原始时钟的分频工作。
此模式工作时钟频率低而使功耗低, 并且锁相环不工作也使功耗降低。
- 3) **空闲模式**: 停止为 CPU 提供时钟, CPU 不工作(其外设均工作)。
退出方法: 任何中断请求可唤醒 CPU 工作, 退出空闲模式
- 4) **断电模式**: 时钟模块断电, 除了唤醒电路之外所有部分均不供电。系统需

分成两部分供电。

此模式功耗最低。必须设置有外中断

退出方法：用中断唤醒。（外部中断 EINT0---15；实时钟报警中断）

5. 主存 VS 外存 (PDF 5)



1) 主存

- i. 主存是处理器能直接访问的存储器，用来存放系统和用户的程序和数据
- ii. 嵌入式系统的主存可位于 SoC 内和 SoC 外，片内存储器存储容量小、速度快，片外存储器容量大
- iii. 可以做主存的存储器有：

RAM 类：SRAM、DRAM、SDRAM 等

ROM 类：Nor Flash、EPROM、E2PROM、PROM 等

2) 外存

- i. 外存储器也称辅助存储器，简称外存或辅存
- ii. 外存主要指那些容量比主存大、读取速度较慢、通常用来存放需要永久保存的或相对来说暂时不用的各种程序 and 数据的存储器

6. ROM 的分类与特点 (PDF 17)

1) 分类

- i. **Mask ROM** (掩模 ROM, 由厂家写入数据, 用户无法修改)
- ii. **PROM** (Programmable ROM, 可编程 ROM, 1 次)
- iii. **EPROM** (Erasable Programmable ROM, 可擦写 ROM)
- iv. **EEPROM** (电可擦除可编程 ROM)

v. **Flash ROM (闪速存储器)**

NOR Flash (逻辑或)	NAND Flash (逻辑与)
<p>a) 支持随机访问，速度快、电压低、功耗低、稳定性高，主要用于主存</p> <p>b) 应用程序可以直接在闪存内运行，不需要再把代码读到系统 RAM 中运行</p> <p>c) 优点：传输效率很高，在 1~4MB 的小容量时具有很高的成本效益</p> <p>d) 缺点：写入和擦除速度很低</p> <p>e) 用于对数据可靠性要求较高的代码存储、通信产品、网络处理等领域</p>	<p>a) 支持顺序访问</p> <p>b) 具有高存储密度，容量大、回写速度快、芯片面积小等特点，主要用于外存</p> <p>c) 闪存和需要特殊的系统接口</p> <p>d) 用于对存储容量要求较高的 MP3 等；也可组成各种类型的电子盘如 USB 盘、CF、SD 和 MMC 存储卡等</p>

	NOR	NAND
写入/擦除一个块的操作时间	1 ~ 5s	2 ~ 4ms
读性能	1200 ~ 1500KB	600 ~ 800KB
写性能	<80KB	200 ~ 400KB
接口/总线	SRAM接口/独立的地址数据总线	8位地址/数据/控制总线，I/O接口复杂
读取模式	随机读取	串行地存取数据
成本	较高	较低，单元尺寸约为NOR的一半，生产过程简单，同样大小的芯片可以做更大的容量
容量及应用场合	1 ~ 64MB，主要用于存储代码	8MB ~ 4GB，主要用于存储数据
擦写次数(耐用性)	约10万次	约100万次
位交换(bit位反转)	少	较多，关键性数据需要错误探测/错误更正(EDC/ECC)算法
坏块处理	无，因为坏块故障率少	随机分布，无法修正

S3c2410 的自动导入模式: (PDF 53)

a) 当 OM1、OM0 都是低电平时，S3C2410 从 NAND Flash 启动

- b) 复位后，NAND Flash 开始的 4k 代码会被自动地复制到内部 SRAM 中的 Steppingstone，传送后，引导代码在 Steppingstone 中执行
- c) 使用这 4k 代码来把更多的代码从 NAND Flash 中读到 SDRAM 中去运行

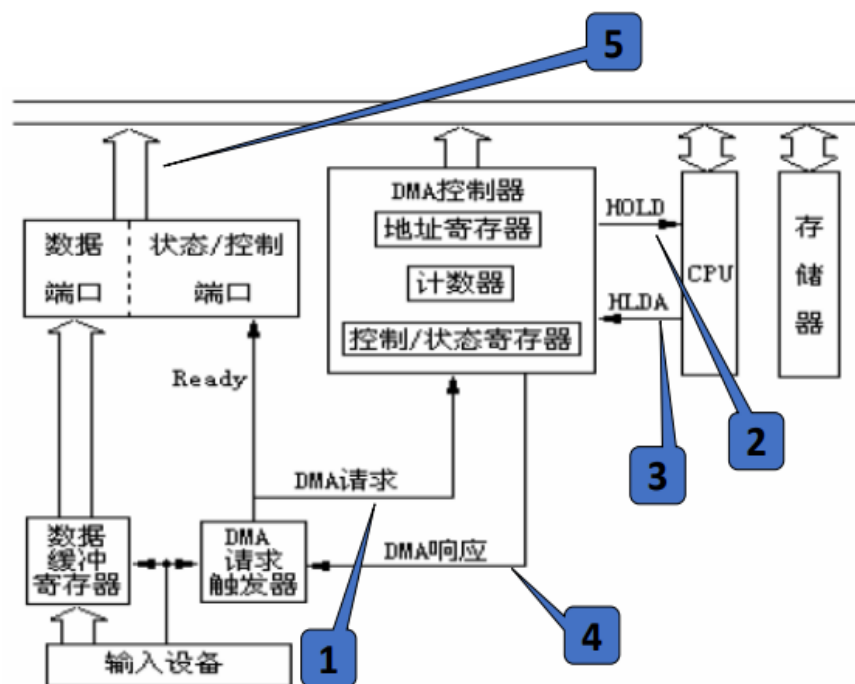
2) 特点

- i. 在烧入数据后，无需外加电源来保存数据
- ii. 断电数据不丢失，但速度较慢
- iii. 适合存储需长期保留的不变数据

7. DMA 的优点: (79)

可以不通过 CPU 的中断来实现数据的传输

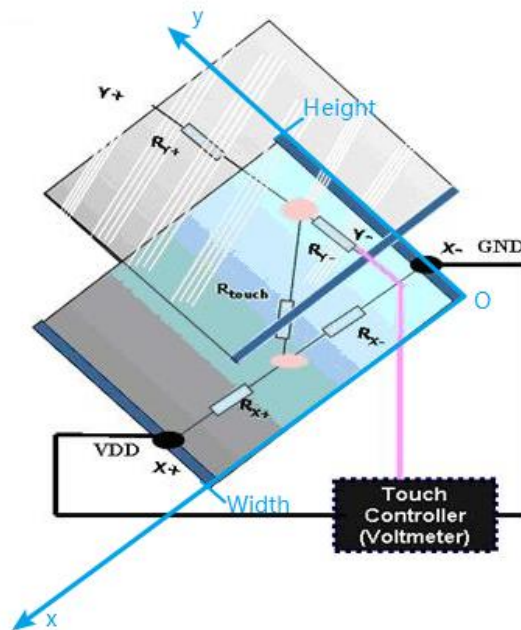
8. DMA 工作过程: (81)



- 1) 外设向 DMAC (C -- Controllor?) 发出请求
- 2) DMAC 通过 HOLD 向 CPU 发出总线请求
- 3) CPU 响应释放三总线，并且发应答 HLDA
- 4) DMAC 向外设发 DMA 应答
- 5) DMAC 发出地址、控制信号，为外设传送数据
- 6) 传送完规定的的数据后，DMAC 撤销 HOLD 信号，CPU 也撤销 HLDA 信号，

并且恢复对三总线的控制

9. 利用 ADC 测量电阻式触摸屏按压点的方法 (PDF 106)

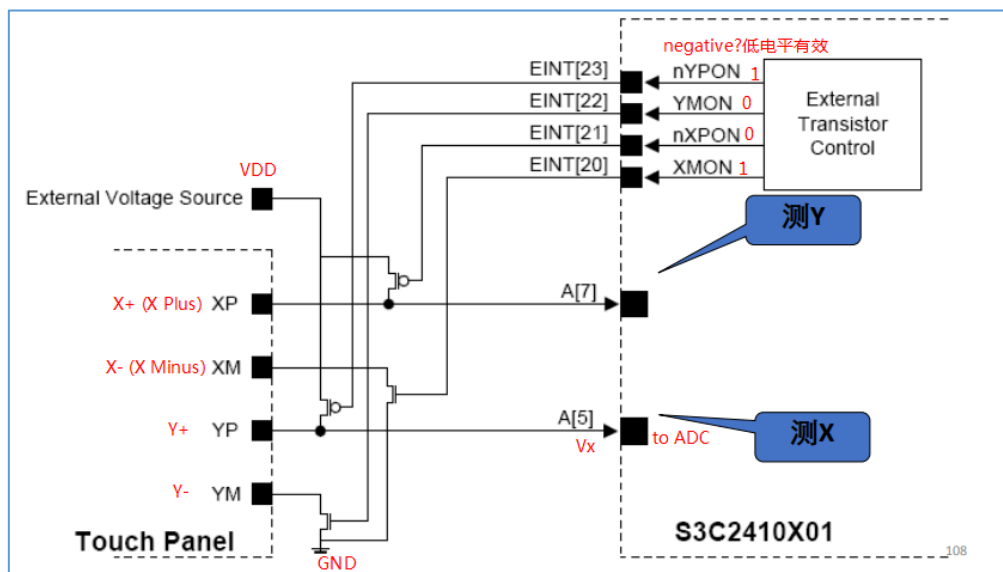


如图: 当按压屏幕时, 上下两层出现通路 R_{touch} , 于是可在 Y- (或 Y+) 可测得电压 V_x , 并且:

$$\frac{V_x}{VDD} = \frac{x}{Width}$$

于是水平方向距离: $x = Width * V_x / VDD$,

当测量 V_x 时, $nXPON=0, XMON=1$; $nYPON=1, YMON=0$



同理, 换 Y-接 GND, Y+接 VDD, 在 X- 得到电压 V_y , 可测得竖直方向距离:

$y = \text{Height} * V_y / V_{DD}$, (x, y) 即确定二维平面的一点.

10. 中断仲裁: (PDF 122)

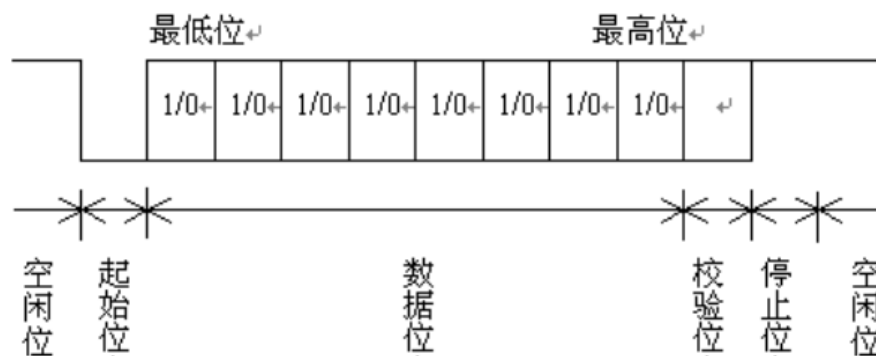
当中断控制器接收到多个中断请求时, 其内的优先级仲裁器裁决后向 CPU 发出优先级最高的中断请求信号或快速中断请求信号

11. PWM (PDF 30)

- 1) **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
- 2) **应用:** PWM 技术可有效实现模拟信号, 广泛应用在从测量、通信到功率控制与变换的许多领域中

12. UART

- 1) **工作原理:** 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
- 2) **组成部分:** 波特率发生器、接收器、发送器和控制单元
注: 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
- 3) **帧格式:** 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

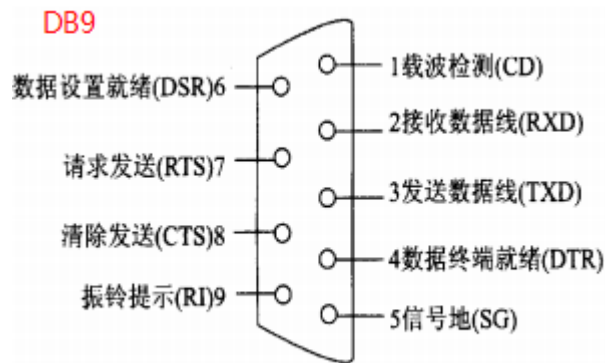


注: 发送超过一帧 0 作为中止信号

- 4) **奇偶校验:** 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)
应用: UART 在传输的字符中专用一位来作误码检测 --- 奇偶校验位
局限性: 检错能力有限 (可用 CRC)
- 5) **信号电平 (与后面 RS232 对比)**

13. RS232

1) 接口标准 (PDF 63)?



2) 信号电平 (PDF 70)

i. UART 电平: LVTTTL

a) 逻辑 0: 0V ~ 0.8V

b) 逻辑 1: 2V ~ 3.3V

ii. RS-232 电平:

a) 逻辑 0: +3V ~ +15V

b) 逻辑 1: -3V ~ -15V

iii. 欲将 UART 与 RS-232C 连接起来通信, 必须进行电平变换

注: MAX3222 芯片可完成 TTL <--> EIA 双向电平转换

14. SPI (PDF 76, 78, 79)

1) 介绍: 是 Motorola 公司开发的一种同步串行外设接口标准, 它以主从方式工作, 这种模式通常有一个主设备和一个或多个从设备, 需要至少 4 根线 (单向传输时 3 根也可以):

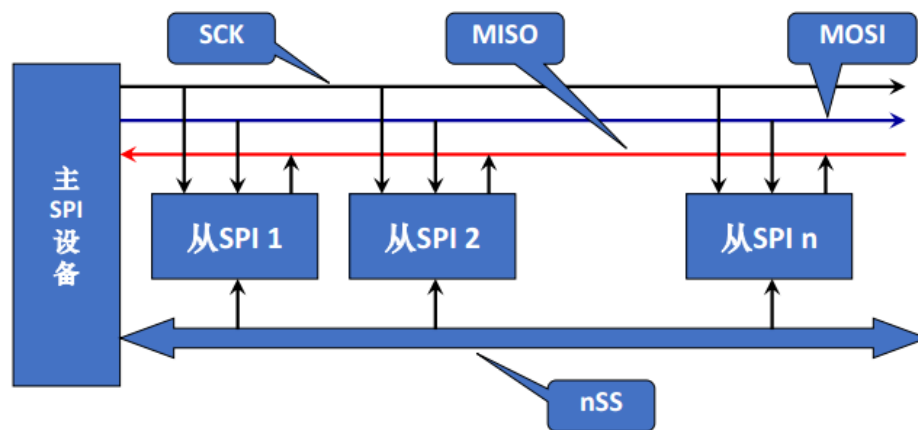
i. **SDI** - Serial Data In, 数据输入;

ii. **SDO** - Serial Data Out, 数据输出;

iii. **SCLK** - Serial Clock, 时钟 (由主设备产生);

iv. **CS** - Chip Select, 片选 (由主设备控制)。

2) 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。

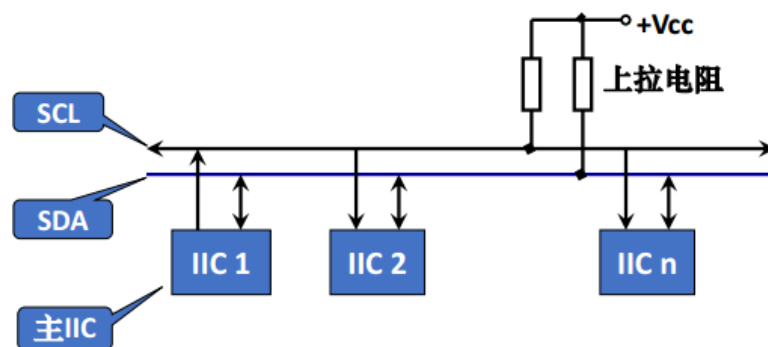


3) 工作原理

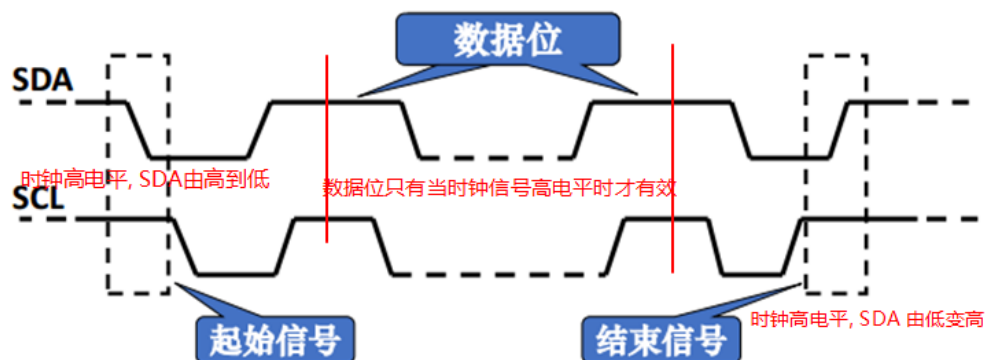
?

15. I2C (PDF 96)

1) 组成

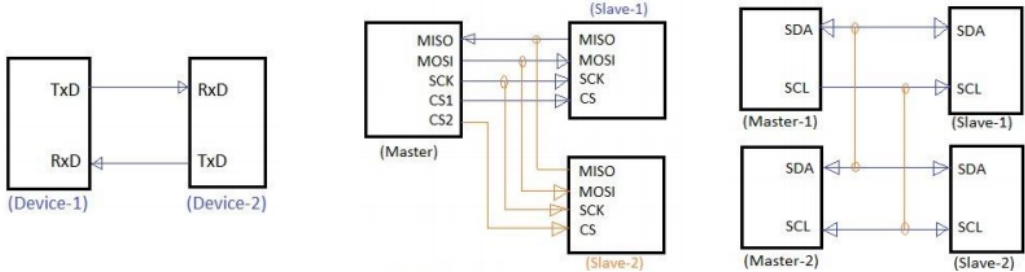


- 2) 工作原理: I2C 是两线，**SDA** (数据) 和 **SCL** (时钟)，也是一种同步传输协议。主机在发送开始信号之后，先发送 7 个比特的地址位和 1 个比特的读写位，每个从机有自己的 I2C 地址，当发现该条指令是发给自己的时候，拉低 SDA 线 (即回复 ACK 信号)，然后主机发送或接收数据，完成传输。传输完成之后，主机发送停止位，完成该次传输。



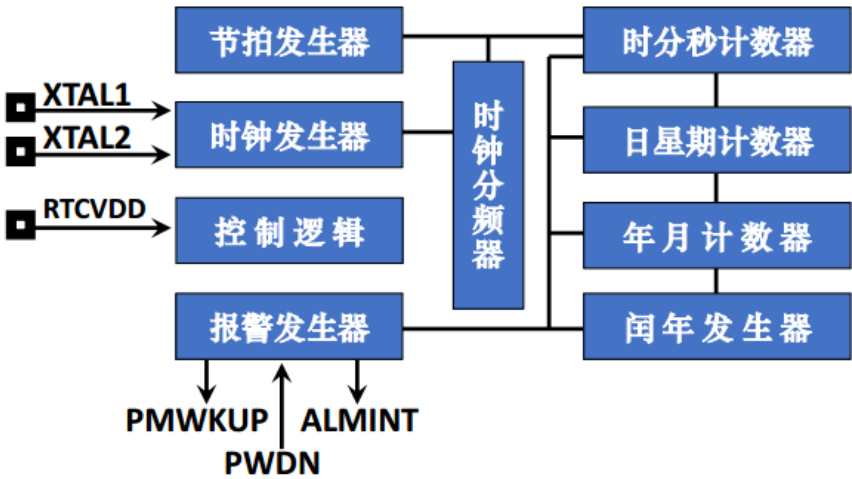
16. UART, SPI, I2C 对比 (PDF 113):

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



17. RTC (实时时钟) (PDF 4):

- 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（电度表、燃气表、水表等）、家用电器（电视机、机顶盒、DVD）等，应用非常广泛
- 结构：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



3) 为何多选用 32.768KHz 晶振

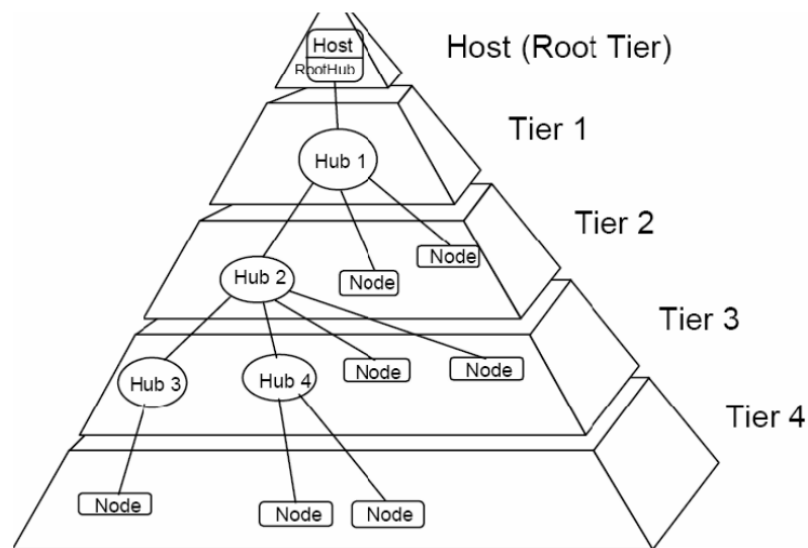
$32.768K = 32768 = 2^{15}$, 32.768KHz 的晶振产生的振荡信号经过石英钟内部分频器进行 15 次分频后得到 1Hz 秒信号, 即秒针每秒中走一下, 石英钟内部分频器只能进行 15 次分频, 要是换成别的频率的晶振, 15 次分频后就不是 1Hz 的秒信号, 钟就不准了。

<https://blog.csdn.net/lifengxun20121019/article/details/8274749>

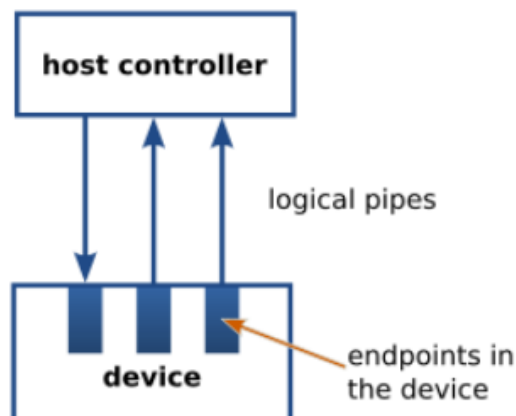
18. USB

1) 拓扑结构 (PDF 43)

树状结构:



2) USB 接口逻辑组成 (PDF 44):



- i. **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)

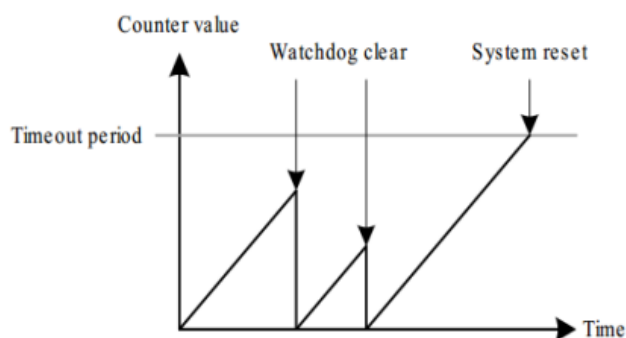
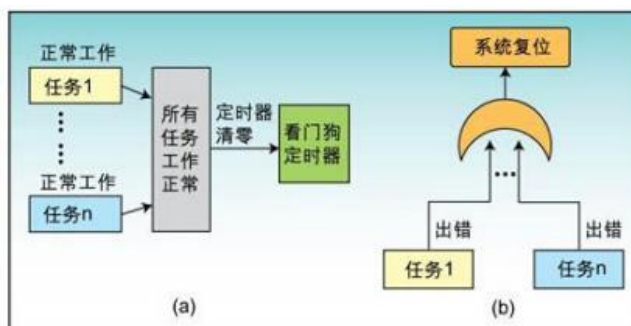
- ii. **USB 设备 (device):** 所有的 USB 设备均可接收数据, 根据数据包的地址判断是否保存
- iii. **端点 (endpoint):** 端点是位于 USB 设备中、与 USB 主机进行通信的基本单元。USB 设备可以有多个端点, 各端点的地址由设备地址和端点号确定。在 USB 设备中, 端点就是一个数据缓冲区
- iv. **管道 (pipe):** 是主机与设备之间数据通信的逻辑通道。

3) USB 总线特点 (PDF 46)

- i. USB 端口不区分设备
- ii. 即插即用、可热插拔
- iii. 传输速度高
- iv. 易扩展, 可扩展到 127 个 USB 设备
- v. 对设备提供电源
- vi. 成本低等

19. 看门狗(WATCHDOG) (PDF 58, 59)

- 1) **定义:** 一种定时器电路, 具有监视并恢复程序正常运行的功能。
- 2) **用途:** 当系统“跑飞”而进入死循环时, 恢复系统的运行
- 3) **原理:**



- i. 设系统程序完整运行一周期的时间是 T_p , 看门狗定时周期为 T_i ($T_i > T_p$),
- ii. 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
- iii. 若系统不能在 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

20.

Chapter 07

1. 软件 (PDF 4)

是计算机系统中与硬件相互依存的另一部分, 它包括程序 (program)、相关数据 (data)及其说明文档 (document)

2. 嵌入式软件分类 (PDF 5)

1) 系统软件

控制、管理计算机系统的资源, 如: 嵌入式操作系统, 嵌入式中间件 (CORBA, Java)

2) 支撑软件

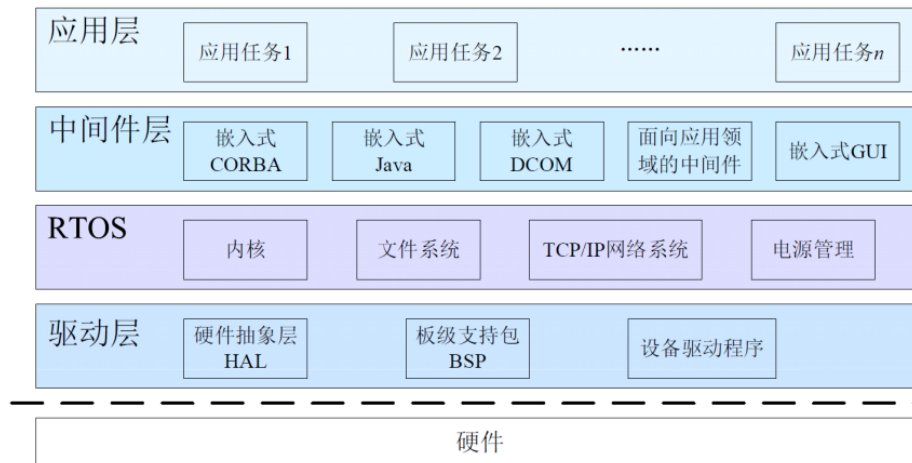
辅助软件开发的工具, 如: 系统分析设计工具, 仿真开发工具, 交叉开发工具等

3) 应用软件

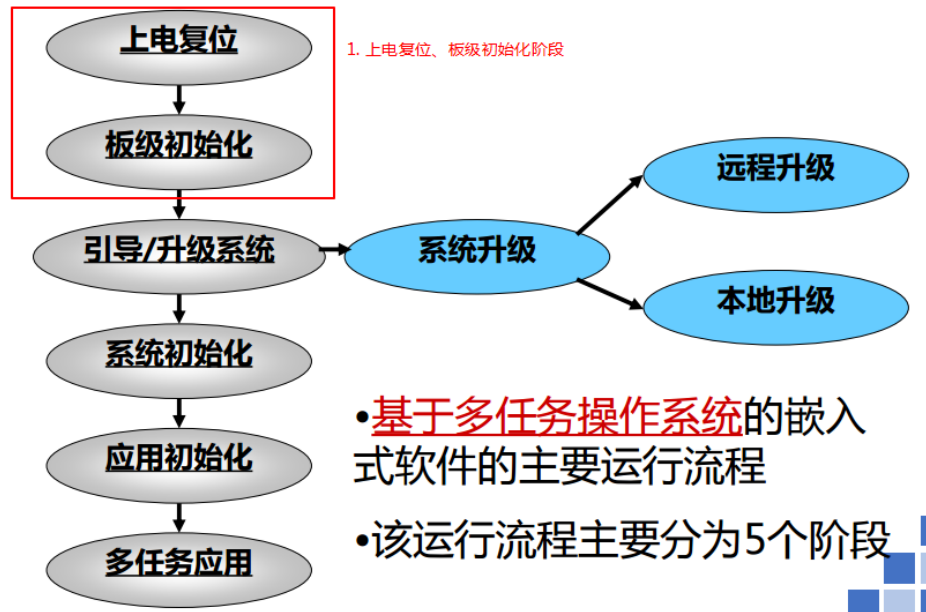
面向应用领域, 如: 手机软件, 路由器软件, 交换机软件等

3. 嵌入式软件系统

1) 体系结构 (PDF 7):



- i. **驱动层:** 提供所有的外围设备的支持和给上层软件提供设备的操作接口
 - ii. **RTOS:** 主要完成设备和资源的管理
 内核(必需), 完成基本的功能(任务调度和管理、内存管理等)
 文件系统、驱动程序、网络通信, 可根据需要进行裁减
 - iii. **中间件层:** 中间件是一类连接软件组件和应用的计算机软件, 它包括一组服务, 以便运行在一台或多台机器上的多个软件通过网络进行交互
 - iv. **应用层:** 操作系统之上运行的用户程序
 - a) 主要由多个相对独立的应用任务组成
 - b) 每个任务完成特定的工作, 如 I/O 任务、计算的任务、通信的任务
 - c) 由操作系统调度各个任务的运行
- 2) 主要运行流程 (PDF 12):



4. 嵌入式操作系统

1) 特点 (PDF 21):

- i. 可固化
- ii. 可配置、可剪裁
- iii. 独立的板级支持包，可修改
- iv. 不同的 CPU 有不同的版本
- v. 应用的开发需要有集成的交叉开发工具

2) 组成 (PDF 24):



- i. 内核是嵌入式操作系统的基础，也是必备的部分，还提供特定的应用

编程接口, 主要功能: **任务管理, 内存管理, 通信同步与互斥机制, 中断管理, 时间管理, 任务扩展 ..**

- ii. **任务管理:** 是内核的核心功能, 具有任务调度、创建任务、删除任务、挂起任务、解挂任务、设置任务优先级等功能
 - a) 通用计算机的操作系统追求的是最大的吞吐率, 为了达到最佳整体性能, 其调度原则是公平, 采用 Round-Robin 或可变优先级调度算法, 调度时机主要以时间片为主驱动
 - b) 而嵌入式操作系统多采用**基于静态优先级的可抢占的调度**, 任务优先级是在运行前通过某种策略静态分配好的, 一旦有优先级更高的任务就绪就马上进行调度

5. 交叉开发环境 (PDF 48)

1) **定义:** 用于嵌入式软件开发的所有工具软件的集合

2) **主要工具:**

- i. 文本编辑器
- ii. 交叉编译器
- iii. 交叉调试器
- iv. 仿真器
- v. 下载器等

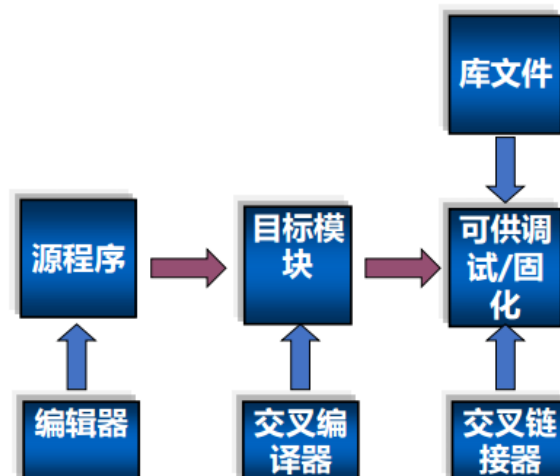
3) **组成:** 宿主机和目标机组成, 宿主机与目标机之间在物理连接的基础上建立起逻辑连接

注: 宿主机 (Host): 是用于开发嵌入式系统的计算机。一般为 PC 机, 具备丰富的软硬件资源, 为嵌入式软件的开发提供全过程支持

目标机 (Target): 是嵌入式系统 / 软件的运行环境, 其硬件软件是为特定应用定制的. 在开发过程中, 目标机端需接收和执行宿主机发出的各种命令如设置断点、读内存、写内存等, 将结果返回给宿主机, 配合宿主机各方面的工作

6. 嵌入式软件实现阶段的开发过程 (PDF 52)

1) **软件的生成:** 在宿主机上进行, 利用各种工具完成应用程序的**编辑、交叉编译和链接**工作, 生成可供调试或固化的目标程序



注: **交叉编译**: 把在宿主机上编写的高级语言程序编译成可以运行在目标机上的代码, 即在宿主机上能够编译生成另一种 **CPU** (嵌入式微处理器) 上的二进制程序

2) 调试: 通过**交叉调试器**完成软件的调试工作。调试完成后还需进行必要的测试工作

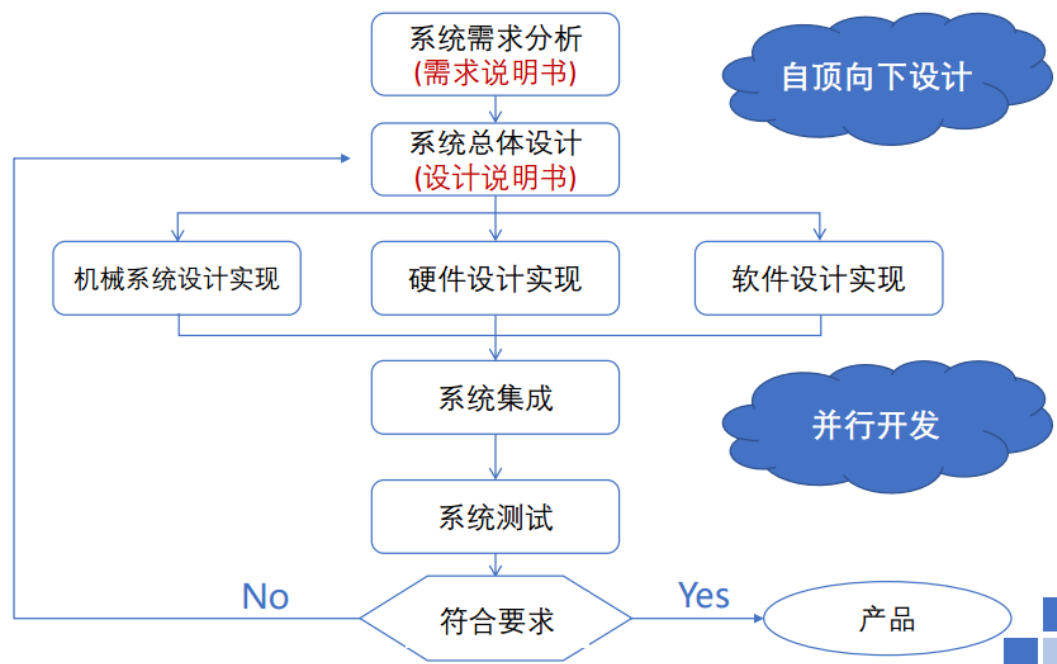
注: 常见调试方法:

- i. **指令集模拟器**: 一种利用 PC 机端的仿真开发软件模拟调试的方法
 - ii. **驻留监控软件**: 驻留监控程序运行在目标板上, PC 机端调试软件可通过并口、串口、网口与之交互, 以完成程序执行、存储器及寄存器读写、断点设置等任务
 - iii. **JTAG 仿真器**: 通过 ARM 芯片的 JTAG 边界扫描口与 ARM 核进行通信, 不占用目标板的资源, 是目前使用最广泛的调试手段
 - iv. **在线仿真器 (In-Circuit Emulator, ICE)**: 使用仿真头代替目标板上的 CPU, 可以完全仿真 ARM 芯片的行为。但结构较复杂, 价格昂贵, 通常用于 ARM 硬件开发中
- 3) 固化运行**: 先用一定的工具将应用程序固化到目标机上, 然后启动目标机, 在没有任何工具干预的情况下应用程序能自动地启动运行

7.

Chapter 08

1. 嵌入式系统开发流程 (PDF 7)



2. 嵌入式系统设计要解决的主要问题 (PDF 5)

- 1) 需要用什么样的系统结构来实现?
- 2) 如何满足时限要求, 如何处理多项功能在时间上的协调一致关系?
- 3) 如何保证系统可靠地工作?
- 4) 如何满足系统接口要求, 嵌入式应用直接和系统接口输入输出信息?
- 5) 如何降低系统的功耗?
- 6) 如何使系统可升级?

3. 系统需求分析 (PDF 8)

- 1) 目的: 用户和设计者有效交流、沟通, 明确设计目标
- 2) 产出物: 需求说明书
- 3) 需求的种类:
 - i. 功能性需求 (输入, 输出, 功能)
 - ii. 非功能性需求 (成本, 大小, 重量, 设计时间..)

4. 系统总体设计 (PDF 15)

- 1) 作用: 实现系统的蓝图, 系统整体结构的一个计划 (形成设计说明书)
- 2) 内容:
 - i. 系统分析
 - ii. 系统软硬件整体结构的设计

- iii. 软硬件划分
- iv. 标准构件和自行设计构件的确定

5. 软硬件划分 (PDF 18)

- 1) 决策依据: 软/硬件划分通常由**速度、灵活性以及开销**来决策
- 2) 划分决策的前提: **双重性** (硬件/软件都可实现)
- 3) 通常由软件实现的功能:
 - i. 操作系统功能 (任务调度, 资源管理, 设备驱动)
 - ii. 协议栈 (TCP/IP)
 - iii. 应用软件框架
 - iv. 除基本系统、物理接口、基本逻辑电路, 许多由硬件实现的功能都可以由软件实现
- 4) 具有双重性的功能
 - i. 算法 (加密/解密, 编码/解码, 压缩/解压 ...)
 - ii. 数学运算 (浮点运算, FFT ...)

6. 处理器的选择依据 (PDF 24)

- 1) 应用类型
- 2) 处理性能
- 3) 主频及功耗
- 4) 封装
- 5) 软件支持工具
- 6) 是否内置调试工具
- 7) 供应商是否提供开发板
- 8) 其它因素
- 9) 只要可能, 尽量选择普通的硬件

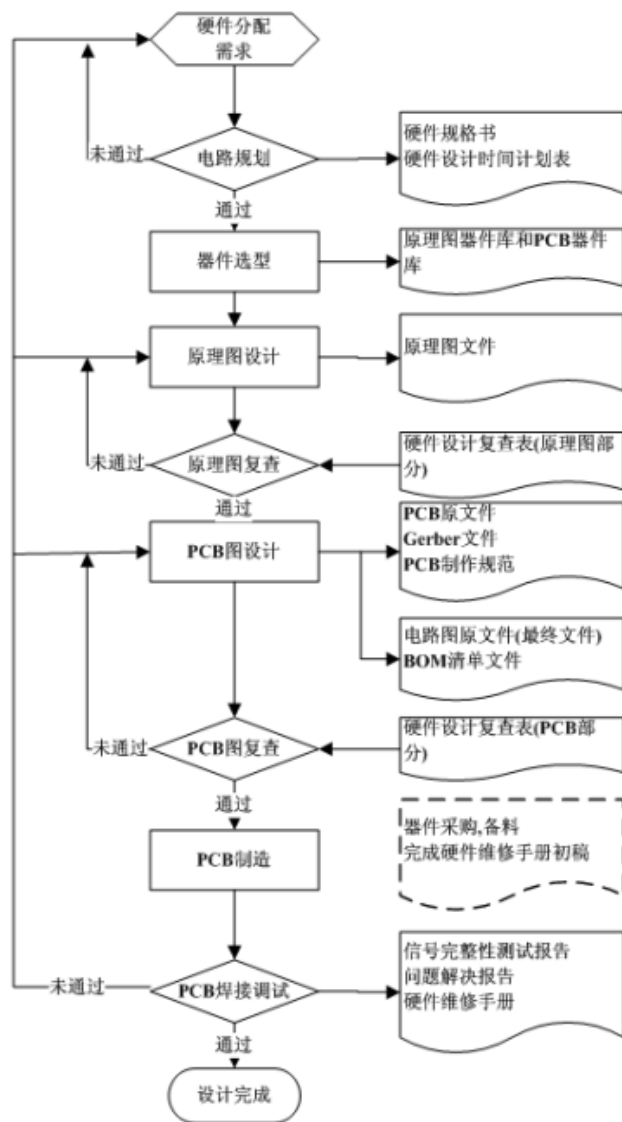
7. 操作系统选定的选择依据 (PDF 25)

- 1) 操作系统提供的开发工具
- 2) 操作系统向硬件接口移植的难度
- 3) 操作系统的内存要求
- 4) 开发人员是否熟悉此操作系统及其提供的 API

- 5) 是否提供硬件的驱动程序，如 SD 卡、LCD 屏幕等
- 6) 操作系统的可剪裁性
- 7) 操作系统的实时性能

8. 硬件设计制作-流程 (PDF 32)

- 1) 硬件分配需求
- 2) 电路规划
- 3) 器件选型
- 4) 原理图设计 (先整体, 后局部)
- 5) 原理图复查
- 6) PCB 图设计
- 7) PCB 图复查
- 8) PCB 制板
- 9) 器件采购、备料
- 10) PCB 焊接及调试



9. Gerber 文件的作用 (PDF 53)

PCB 图检查无误后，生成 Gerber 文件，将此文件发送给 PCB 厂商进行 PCB 制作，作用如下：

- 1) 对原设计文件进行保密
- 2) 文件中各项设计参数的标注是明确的，避免引起 PCB 厂商对设计的误解

10. 原材料清单 BOM (PDF 54)

- 1) 根据原理图生成相应的原材料清单 BOM (Bill Of Material)，然后根据 BOM 清单进行器件采购工作，之后再根据所需的数量进行器件的备料，等待 PCB 板制作完成后即可进行焊接工作
- 2) BOM 清单中包含了器件的详细信息，包括器件名称、说明、以前采购的价格及供应商名称等。另外在焊接时可对照 BOM 清单进行器件焊接，

清晰、直观、不易出错

3) 与 PCB 的制作过程同步进行

4) 示例:

序号↕	分类↕	数量↕	值↕	说明↕	器件编号↕	单 价↕	金 额↕	供应 商↕	联系 方式↕
01↕	电阻↕	38↕	22_5%↕	RES 22 5% 1/10W SMD 0603↕	R3 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16 R17 R18 R19 R20 R21 R22 R23 R24 R25 R26 R29 R30 R31 R32 R33 R34 R35 R36 R37 R43 R46 R52 R53 R56 R57↕	0.0 04↕	0.1 52↕	↕	↕
02↕	电阻↕	1↕	470_5%↕	RES 470 5% 1/10W SMD 0603↕	R61↕	0.0 04↕	0.0 04↕	↕	↕

11. 软硬件协同设计 (PDF 70)

1) 目的: 使软件和硬件最佳组合

2) 过程

- 需求分析
- 软硬件协同设计
- 软硬件实现
- 软硬件协同测试和协同验证

12.