# 第一章 嵌入式系统

1. 嵌入式系统三要素：嵌入性，专用性，计算机系统。
2. 概念：嵌入式系统是嵌入到对象体系中、用于执行独立功能的专用计算机系统。
3. 嵌入式技术是后pc时代的核心技术。
4. 从储存介质角度看，嵌入式系统大多采用半导体存储器。
5. 从广义上看，嵌入式系统是一种计算机系统。
6. 嵌入式系统与桌面系统的区别：系统内核小，专用性强，运行环境差异大，可靠性要求高，系统精简和高实时性操作系统，具有固化在非易失性储存其中的代码，嵌入式系统开发工具和代码
7. 构成；一般由嵌入式处理器，外围硬件设备，嵌入式操作系统，用户的应用软件系统。
8. 硬件：以嵌入式处理器为核心，由存储器、I/O单元电路、通信模块、外设、接口组成。
9. 按照嵌入式处理器字长分类可以分为；4位、8位、16位……
10. 按照软件实时性需求分类有：非实时系统、软实时系统、硬实时系统。
11. 按照复杂程度：小型系统、中型系统、复杂系统。

# 第二章 微处理器

1. 冯·诺伊曼结构：将程序（指令序列的集合）和数据存放在同一存储器的不同地址；顺序执行指令；执行过程：取指令（或数据），分析指令，执行指令。
2. 哈弗结构：将程序和数据存放在不同的储存器中；并执行指令；执行过程：取指令（和数据），分析指令，执行指令。
3. CISC:复杂指令集计算机：用最少的语言指令完成所需要的任务。包含大量复杂功能指令和丰富的寻址方式。与嵌入式相悖。

8/2原则：常用指令仅占指令总数的20%，但在程序中出现的频率却占80%。

1. RISC:精简指令集计算机：借助可以在单个周期完成的指令，以降低cpu的复杂程度，将程序的复杂性交给编译器。只包含最有用的指令，只提供简单操作。
2. 注意：计算机的存储器结构和指令系统没有对应。
3. 流水线技术：几条指令并行（与计组类似）。
4. ARM：微处理器设计企业。
5. ARM芯片内部结构：ARM CPU+外部设备=ARM芯片；
6. ARM运行模式：
7. 
8. 除用户模式其他模式为特权模式；特权模式可以随意访问系统资源，任意切换处理器模式；除用户模式和系统模式其他模式成为异常模式，用于处理异常（包括中断）和访问受保护的系统资源。
9. 模式切换：程序切换或者特定的异常（中断）出现后自动进入。管理模式-多种特权模式-用户模式。
10. 两种状态：ARM状态：执行32位的字对齐的ARM指令。

处理器开始执行代码时只能处于ARM状态。

Thunb状态：执行16位的、半字对齐的Thumb指令。

状态切换：切换处理器或状态指令。

执行BX Rn 当状态位bit[0]=1时从ARM到Thumb状态。当状态位bit[0]=0时从Thumb到ARM状态

1. ARM指令集：限定宽度32位，四字节对齐，代码密度低。

Thumb指令集：减少代码量提出；只有一些通用功能。

指令集关系：

1. 寄存器组织：R0-R15通用寄存器
   1. R0-R7未分组寄存器，保存数据或者地址。不同运行模式使用相同的物理寄存器。
   2. R8-R14分组寄存器：
      1. R8-R12:

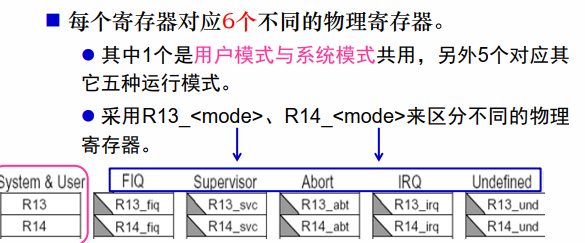
每次访问的物理地址和运行模式相关

每个寄存器对应两个不同的物理寄存器

当使用快速中断（fiq）时访问R8(FIQ)-R12(FIQ)，其他模式使用R8(USR)-R12(USR);

* + 1. R13-R14

每个对应6个物理寄存器



R13常用作堆栈指针，初始化时需要初始化每种模式对应的R13。（sp计数器）

R14常用作链接寄存器，保存子程序返回地址或者，异常（中断）返回地址(LR子程序返回地址)

* 1. R15程序计数器PC 指向同一个物理寄存器

ARM状态下bit[1:0]=00,bit[32:2]存放pc pc的值为正在执行的命令的地址值+8，即正在执行的命令的后两条。-多级流水线技术。

Thumb bit[0]=0,bit[31:1]存放pc

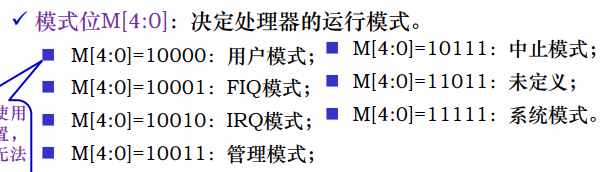
1. 状态寄存器6个
   1. CPSR当前状态：所有模式同一地址，任何模式下都可以访问。
   2. SPSR备份状态：每种异常模式专用物理地址，异常模式发生时保存CPSR，异常退出恢复CPSR；
   3. 条件码标志：N表示带符号运算的正负N=1为负；N=0为正或0

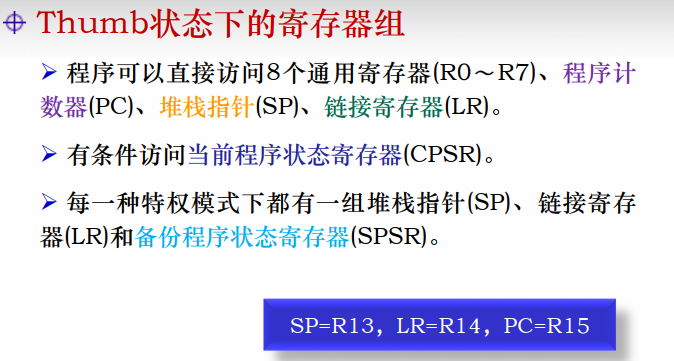
Z=1表示运算结果为0；

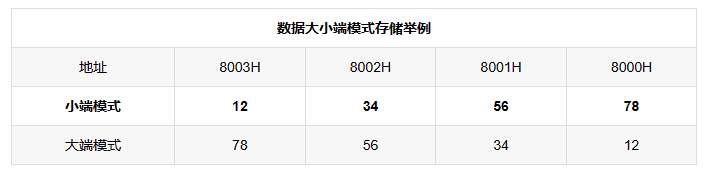
C：加法结果最高位进位时C=1，减法最高位借位时C=0，位移指令时，C为从最高位最后移出的位；其他指令C的值不变。

V：符号位溢出则V=1；

Q代表DSP运算是否溢出，

* 1. 控制位：I=1禁止IRQ，F=1禁FIQ；T=1处理器位Thumb模式，T=0为ARM模式。
  2. 模式位：决定运行模式：M [4:0]
  3. 

1. 
2. 存储器格式：

大端格式：数据的高字节在低地址，低字节在高地址；小端格式相反。

1. 存储器管理单元MMU

将虚拟地址转换为物理地址—将主存地址从虚拟存储空间映射到物理存储空间。

存储器访问权限控制

设置虚拟存储空间的缓冲特性。

1. ARM920T支持的存储块模式：段，大页，小页，极小页。

段：1MB，大页 ：64KB，小页：4KB，极小页：1KB；

1. ARM异常处理
   1. 状态保存：下一条地址存入连接寄存器LR，CPSR复制到SPSR保存；
   2. 根据异常类型强制设置CPSR的运行模式位。强制PC从相关的异常向量地址取下一条指令执行，从而执行相应的异常处理程序。
   3. 将连接寄存器LR的值减去相应的偏移量后送到PC中；将SPSR复制回CPSR；清除中断禁止位。
2. 1

# 第三章 ARM指令

1. 寻址方式：寄存器寻址，立即寻址，寄存器偏移寻址，寄存器间接寻址，基址寻址，多寄存器寻址，堆栈寻址，相对寻址。
   1. 寄存器寻址，指令中操作数给的是寄存器编号，直接取出寄存器的值来操作。MOV R1,R2.
   2. 立即寻址，指令中的操作数就是数据本身，数据包含在指令中。

SUBS R0,R0,#1 //R0 = R0-1;

MOV R0,#0XFF00; 将立即数装入R0；

* 1. 寄存器偏移寻址（ARM特有）：指令在使用操作数前，先移位。

MOV R0,R2,LSL,#3 //R2的值左移3位，结果放入R0

ANDS R1,R1,R2,LSL,R3 // R2的值左移R3位，然后和R1相与结果放入R1。支持的操作：LSL,LSR,ASR,ROR,RRX。

* 1. 寄存器间接寻址：指令操作数给出的时寄存器编号，寄存器中存操作数的地址。LDR R0,[R2] //将R2指向的存储单元的内容读出放入R0
  2. 基址寻址：将基址寄存器的内容与指令中给出的偏移量相加，形成操作数的有效地址。LDR R2,[R3,#0x0C] //读取R3+0x0C中存储的的内容，放入R2.
  3. 多寄存器寻址：LDMIA R1!,{R2-R4,R6} //将R1指向的顺序存储单元中的数据读取到R2~R4和R6，每次读取，R1自动加4；连续的寄存器用-连接，否则用，连接。
  4. 堆栈寻址：（先进后出）FA:满递增；EA：空递增；FD：满递减；ED空递减。
  5. 相对寻址：由PC提供基准地址，操作数为偏移量。两者相加得EA有效地址。

1. 指令集
   1. 分支指令（实现跳转），专门的跳转指令可以实现前后32mb空间跳转。

B label 指令：跳转到label；BL label指令带返回的跳转，跳转时将pc存入LR（R14）中，来实现返回。BX Rm指令：带状态切换的跳转Rm到PC，切换处理器状态。 BLX Rm带返回的切换跳转，先存PC再跳，再回来。

* 1. 数据处理：只对寄存器操作，不对存储器操作，可选用使用S后缀并影响状态标志；比较指令不使用S也会影响状态标志。

数据传送：MOV RD ,OP;将op传到rd寄存器；MVN数据非传送，先将op取反再传送，

比较指令：CMP RN,OP比较指令；CMN RN,OP1取反比较指令；

标志位：正数N=0；结果为0则Z=1；产生借位C=0；符号溢出则V=1；

TST RN,OP位测试指令；TEQ RN,OP相等测试指令；

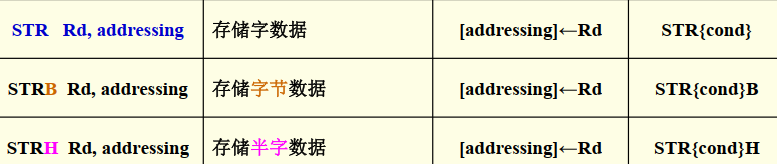






* 1. 存储区访问指令：使用单寄存器加载指令加载数据到PC寄存器，可以实现程序跳转。







上面三条大题可能会考，在ARM中只有MRS指令可以将状态寄存器CPSR或SPSR读出到通用寄存器。

下图重要：



* 1. 1
  2. 常考指令（自己觉得）

LDM(或者STM)(后面加类型) 基址寄存器{！}，寄存器列表{^}

类型有：IA：每次传送后地址加1，递增

IB:：每次传送前地址加1，递增

DA：每次传送后地址减1，递减

DB：每次传送前地址减1，递减

FD：满递减堆栈

ED：空递减堆栈

FA：满递增堆栈

EA：满递增堆栈

{！}为可选后缀，若选用则数据传输完毕后将最后的地址传入基址寄存器。

{^}后缀只在数据传输用，包含PC选用后会将SPSR复制到CPSR。同时改后缀表示传入传出为用户模式下的寄存器。

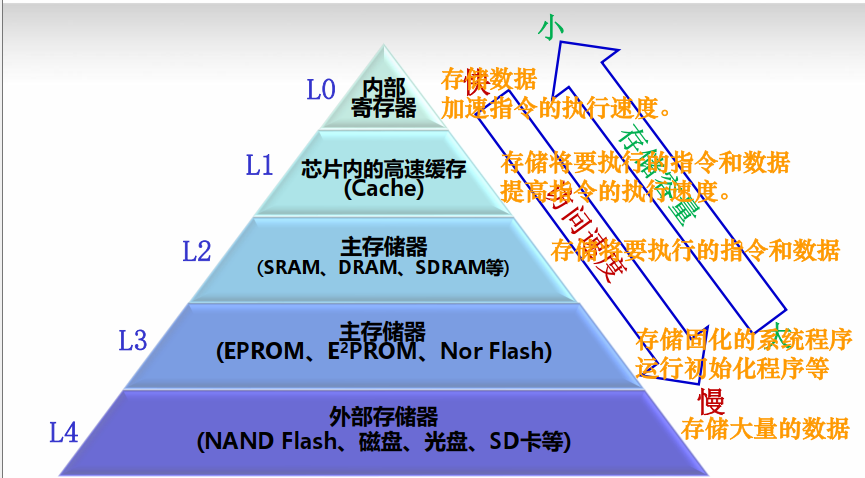
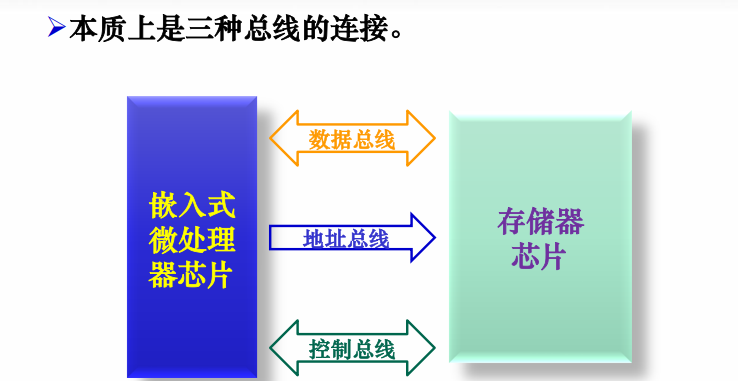
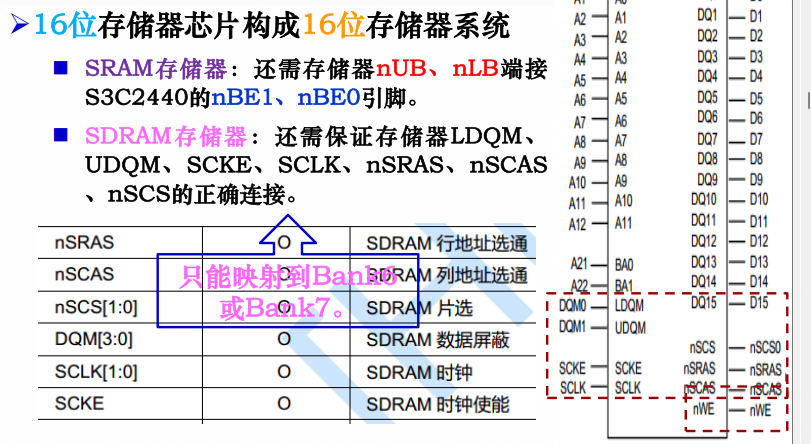
例：STMFD R13!,{R0,R4-R12,LR}//将寄存器列表中的寄存器（R0,R4-R12,LR）存入堆栈

ARM规定寄存器由小到大排列；

LDMFD R13!,{R0,R4-R12,PC}//将堆栈内容恢复到寄存器（R0,R4-R12,LR）

**这章太多建议狠狠的背诵**

# 第四章 存储器

1. 嵌入式存储器大多数用半导体存储器
2. 
3. 常见有8位、十六位（用的最多）、32位。
4. 
5. 外部存储器连接实例
   1. 8位构成8位：储存器DQ0-7与芯片DATA0-7对应连接地址线A0-A15与ADDR0-15对应连接。允许输出nOE接芯片nOE，写信号nWE接芯片nWE，片选信号nCE接芯片nGCSn引脚
   2. 8位并联构成16位：低8位储存器DQ0-7与芯片DATA0-7对应连接，高8位储存器DQ0-7与芯片DATA8-15对应连接，地址线A0-A15与ADDR**1**-16对应连接。允许输出nOE接芯片nOE，低8位写信号nWE接芯片nWBE0，高8位写信号nWE接芯片nWBE1，片选信号nCE接芯片nGCSn引脚（作为整体配置到同一bank中）。
   3. 8位并联构成32位：低8位储存器DQ0-7与芯片DATA0-7对应连接，次低8位储存器DQ0-7与芯片DATA8-15对应连接，次高8位储存器DQ0-7与芯片DATA16-23对应连接，高8位储存器DQ0-7与芯片DATA24-31对应连接；地址线A0-A15与ADDR**2**-17对应连接。允许输出nOE接芯片nOE，低8位写信号nWE接芯片nWBE0，次低8位写信号nWE接芯片nWBE1，次高8位写信号nWE接芯片nWBE2，高8位写信号nWE接芯片nWBE3，片选信号nCE接芯片nGCSn引脚（作为整体配置到同一bank中）。
   4. 16位构成16位：储存器DQ0-15与芯片DATA0-16对应连接地址线A0-A16与ADDR1-19对应连接。允许输出nOE接芯片nOE，写信号nWE接芯片nWE，片选信号nCE接芯片nGCSn引脚
   5. 
6. Nor Flash存储器与NAND Flash存储器芯片

看TMD ppt吧，特别是NAND。

1. 使用NAND Flash作为引导ROM的启动流程

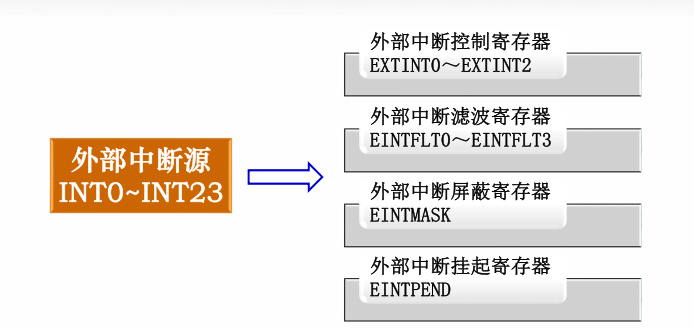
Boot loader程序启动加载引导程序，引导程序通过NAND Flash初始化NAND flash芯片，读取引导扇区，引导程序执行。

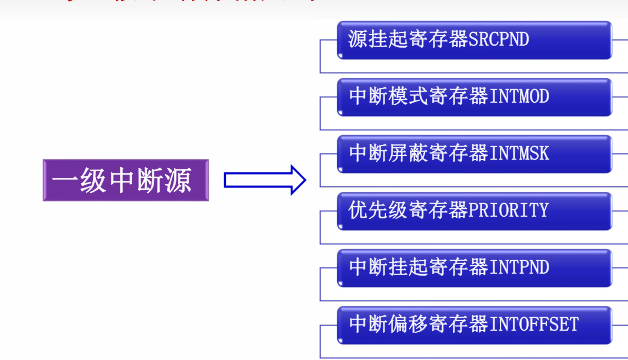
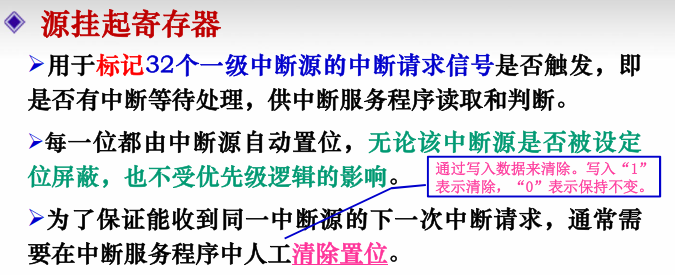
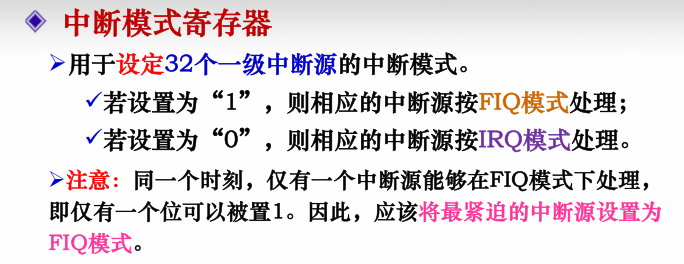
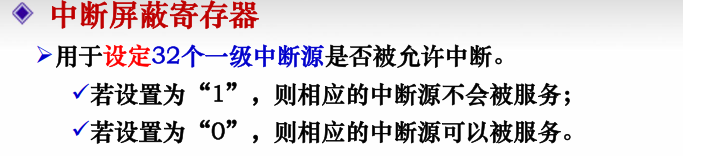
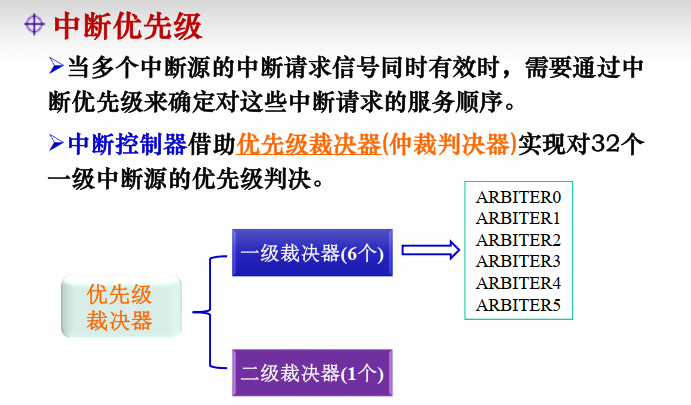
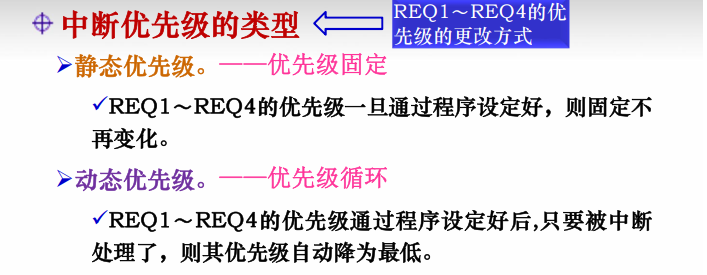
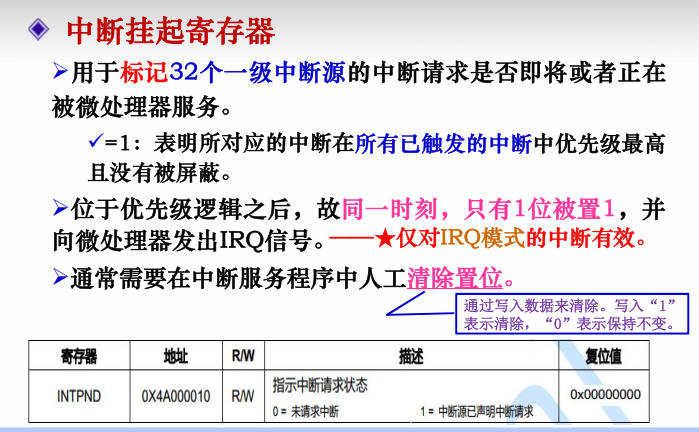
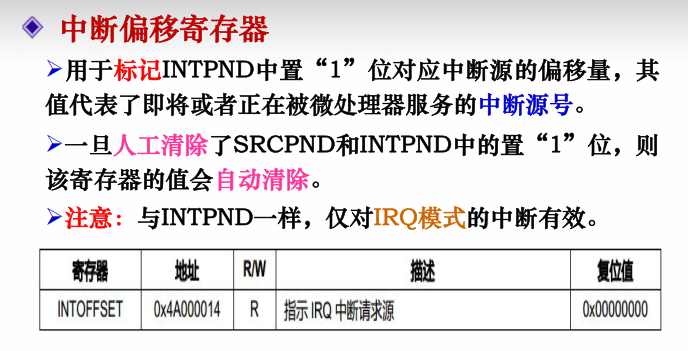
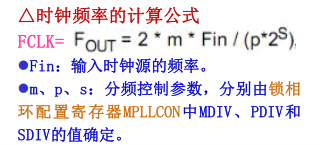
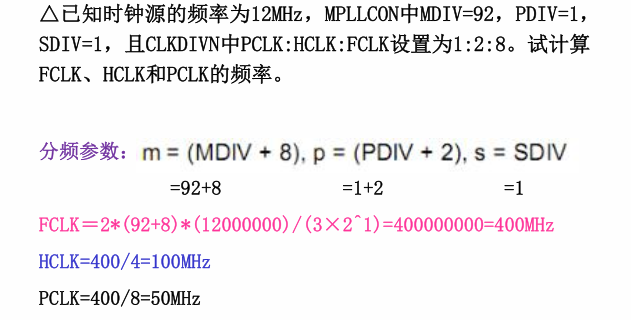
# 第五章 外围设备及接口

1. GPIO：通用输入输出端口，连接各种输入输出设备，实现信号和数据传输。
   1. GPA：23个输入输出引脚，功能1：普通输出口2：外接存储器的地址信号和存储块选择信号。
   2. GPB：11个输入输出引脚，功能1：普通输出口2：用于DMA和总线的请求和应答信号以及各种时钟信号
   3. GPC：16个输入输出引脚，功能1：普通输出口2：LCD的相关信号
   4. GPD：23个输入输出引脚，功能1：普通输出口2：
   5. GPE：23个输入输出引脚，功能1：普通输出口2：
   6. GPF：23个输入输出引脚，功能1：普通输出口2：
   7. GPG：23个输入输出引脚，功能1：普通输出口2：
   8. GPH：23个输入输出引脚，功能1：普通输出口2： 用于异步串行接口uart的控制信号。（2个）
   9. GPI：23个输入输出引脚，功能1：普通输出口2：
   10. GPJ：23个输入输出引脚，功能1：普通输出口2：用于摄像头的数据信号和控制信号。
   11. 
   12. 配置：CON 数据：DAT 上拉：UP 驱动能力：DSC0-DSC1;
2. 中断：在执行过程中，因为某件事发生，收到来自外部的请求信号。若响应信号，则暂停处理当前。
   1. 作用：并行处理：当传输数据才中断，其他时间与外围同时工作。

实时处理：在实时控制时，外围随时提出请求，通过中断快速响应。

故障处理：出现突发故障，利用中断处理。

* 1. ARM920T中断构成（S3C2440为例）：快速中断模式（FIQ）和外部中断模式(IRQ)。中断控制器管理中断信号。中断信号：中断源发出-中断控制器-处理器内核。
  2. 中断控制器功能：外部请求信号管理-中断模式设定-中断信号标记-中断频闭设定-中断优先级管理-中断服务标志。
  3. 
  4. 外部中控制寄存器(EXTINT1-2)：用于设定外部中断请求信号的触发方式：高低电平触发，上升下降或者双边沿触发；
  5. 外部中断滤波寄存器(EINTFLT0-3): 用于设定EINT16-EINT23的滤波器的时钟信号来源和滤波宽度，有效信号要在40ns以上。
  6. 外部中断屏蔽寄存器(EINTMASK)：用于标记20个外部中断请求信号（EINT4-EINT23）是否触发。
  7. 外部中断挂起寄存器(EINTPEND)：用于标记20个外部中断是否触发，供中断服务程序读取和判断。

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 定时部件
    1. 时钟：为微处理器工作提供基本的时钟信号，实现内外设的时序控制。
    2. 定时：产生不同周期或者波形的时钟信号。外部信号计数。时间信息。复位信号。定时，计数，脉宽调制。
    3. 时钟控制模块 发时钟信号 给定时部件。
    4. 三种时钟信号：FCLK内核时钟,HCLK总线时钟,PCLK I/O接口时钟
    5. 分频计算
    6. 
    7. 
11. 定时器：功能：定时，计数，脉宽调制。
    1. 时钟分频器 2 4 8 16 TCLK外部
    2. 执行减一操作的是TCNTn，初始值写入TCNTBn每次开始复制到TCNTn，程序读取TCNTOn间接获取TCNTn。TCNTn自动重载。
    3. 双缓冲功能，定时操作中，重新设置定时器的值，只有当前操作执行完毕后才有效---非立即有效。



1. 实时时钟：年月日时分秒星期8位BCD码表示利用 STRB/LDRB从寄存器取出。
2. 看门狗：WDT用于噪音或系统错误引起故障时，恢复控制器的定时器。
   1. 16位内部定时器：当计数器值为0是（超时），触发中断，将激活128个PCLK时钟周期的内部复位信号。只能使用PCLK的时钟信号。
   2. 