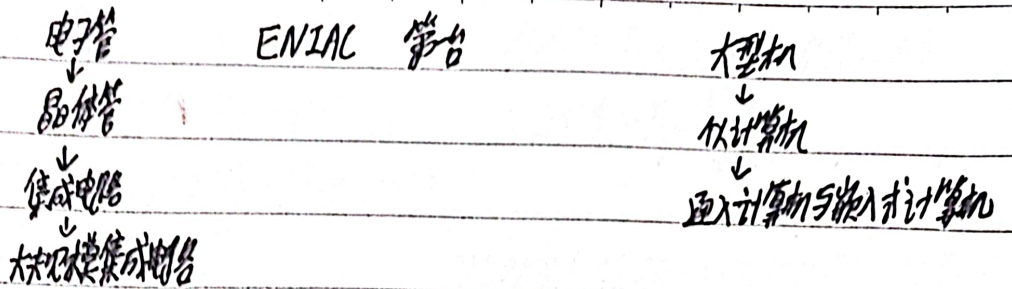


第 章 嵌入式系统导论

① 计算机的发展历史



microprocessor : micro controller 微处理器与微控制器 从4004开始到MC-51到ARM

② 通用计算机与嵌入式区别

- 1) 相对独立软件原料 结合较为紧密
- 2) 应用程序可重新编程 往往不能再编程
- 3) 既作开发机,又作运行机 仅运行机

③ 嵌入式系统的定义

用于控制,监视或辅助设备,机器,工厂运行的设备

一般由嵌入式微处理器,外围硬件设备,嵌入式操作系统及用户应用程序组成 (包括I/O接口,内存,处理器等)

④ 嵌入式系统的发展

1960-1970 航天军警装备 并行实时控制,流水线,操作系统
单总线结构,高速存储器,中断系统...

1971-1989 微处理器问世 4040 8008, 系统总线...

单片机MCU问世 微处理器, I/O接口, A/D/A, RAM等集成. 8048, 8051

数字信号处理器DSP问世

高级语言的编程应用 C语言, 汇编语言

一些专有名词 MCU 微控制器 (804单片机) : 以微处理器MPU为核心; 集成RAM, ROM, 总线, 定时器等, I/O接口

SOC 系统级芯片 微处理器, 存储器, 数字信号处理器...的集中

DSP 数字信号处理器

MPU 微处理器 (与其他总线并叫单片机) (4004, 8008, 8080...)

FPGA 半定制芯片 可编程芯片 可用Verilog HLL编程

CPLD 可编程逻辑芯片 (但资源, 效率低于FPGA)

⑤ 嵌入式系统的特点

- 1) 形式多样且面向特定应用
- 2) 多类型处理器 主要 MM, MIPS
- 3) 极低成本, 实时性, 可靠性要求高
- 4) 需专门开发环境, 开发工具 (不是 C/C++)

成本: 一次性开发成本 + 产品成本 × 产品总量

产品成本: 封装, 原材料, 软件版权成本

最后成本 = $\frac{\text{总成本}}{\text{产品总量}}$

⑥ 嵌入式系统的分类

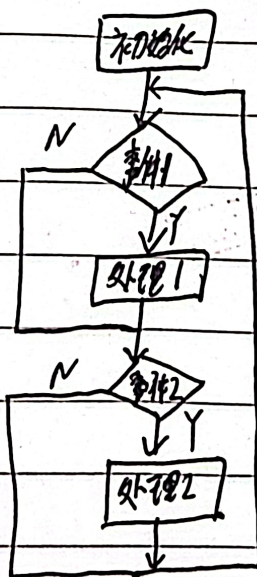
- 1) 按特殊应用分类
 - 信息家电类
 - 移动终端类
 - 通信类
 - 汽车电子类
 - 工业控制类

实时性 { 硬实时: 响应时间不满足会导致致命后果
软实时: 响应时间变高仅造成降低

⑦ 按系统软件复杂程度分类

① 循环轮询系统 (慢速/非常快速的简单系统)

② 前台系统 (控制驱动系统)



initialize ()

while (true)

if (condition-1) action-1;

if (condition-2) action-2;

?

后台任务程序, 使用循环轮询系统

不断运行, 前台由中断处理程序完成。

(ISR 中断服务程序)

中断识别与信号分类

中断嵌套

中断处理

任务程序协调

系统性能由: 中断延迟时间, 响应时间, 恢复时间

③ 单处理器多任务系统

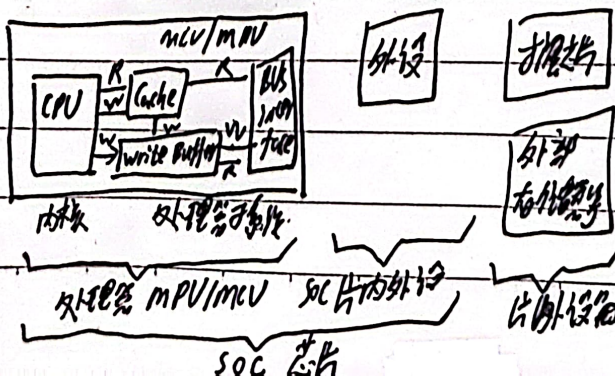
任务是顺序执行, 并行性由操作系统支持

任务间相互通信同步, 由操作系统支持。

④ 多处理器多任务系统

任务并行性均由多个处理器独立运行。

⑦



第二章 嵌入式系统

① 嵌入式系统的基本组成：(嵌入式微处理器、总线、存储器、输入/输出接口和设备)

1) 嵌入式微处理器

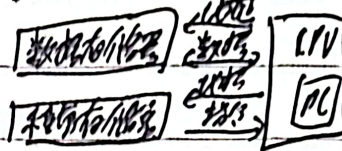
① 结构 冯·诺依曼结构 (Von Neumann 结构)

哈佛结构 (Harvard 结构)

(程序、数据相互独立，两条总线)



指令和数据共用存储器



指令和数据相互独立

② 指令系统 RISC 精简指令系统 reduced

设置两条总线

CISC 复杂指令系统 complex

CISC 复杂指令系统

RISC 精简指令系统

使用部分完成软件功能，硬件复杂度高，集成度高

软件部分完成硬件功能，硬件复杂度高，集成度低

寻址方式多，寄存器较少

流水线降低执行速度，增大代码尺寸

代码尺寸大，指令周期长

寻址方式简单，指令周期短，寄存器多

2) 总线 (片外总线 PCI 广泛使用)

片内总线 AMBA 总线 (ARM 公司)

Wishbone 总线 (OpenCores 组织)

OCIP 总线 (ACP-IP 组织)

总线的种类与微处理器结构有关

3) 存储器 { 主存：代码和数据可直接访问的存储器 ROM, EPROM 等存储器

外存：相对容量大

4) I/O 接口和设备 { I/O 接口主要包含中断控制器、DMA、串并接口、并口等 (DMA: 直接存储器访问)

设备寄存器、计数器、看门狗定时器等... 外设

② 微处理器 (microprocessor) 嵌入式微处理器可分为嵌入式 DSP 与通用的嵌入式微处理器

1) 通用的嵌入式微处理器：一般是集成了通用微处理器的核、总线、外围接口和设备的 SOC 芯片。

2) 嵌入式微处理器的组成：CPU 核心、cache、MMU (内存管理)、总线、I/O 接口、中断控制器 DMA、串并接口、定时器、AD/DA 转换、多任务操作系统...

3) 集成外围逻辑芯片 形式：单芯片式 STM32F107

芯片级 主芯片+从芯片

4) 算木村

受低成本低功耗限制, 往往使用质点运算

(可朱用卿材料批研实验 净点运算)

$$3625_{(10)} + 65_{(10)} =$$

$$\begin{array}{r} 0011.1010(2) \\ + 0110.1000(1) \\ \hline 1010001^0(3) \end{array}$$

$$0.625 = \frac{1}{2} + \frac{1}{8} = 1010$$

$$= 10.125(10)$$

5) 指令集

可以扩展至其他特定领域的运用 (如推广DSP的推广)

MAC 系统条件: 一个显示器或一次生活-精力训练

STMP ~~控制~~ 操作: 允许并指定进行多字节数据流操作。

零开始编程操作：万全发现

多媒体创作指南

6) 性能

MIPS 多种处理器的指令语言描述

一般低端嵌入式微处理器 50MIPS 以内, 标称 150MIPS 以上 (增加时钟频率, 加深流水线深度, 增加 cache 等)

高端微处理器：指令指令集，每个周期执行多条指令，使用多处理器...

7、减小功耗方法

①降低工作强度。

功耗5

成已什

② 提供不同时钟频率

③ 关闭暂时不使用的能量模块

工作电压,工作频率,存储容量,外设数量

④把总行营业部保持最小规模

8) 祈禱

① 粒子密度

② 芯片大小

③ 片上有褶皱大。

(4) 封裝和31脚數

9/ 主活 输入 才 借 款 处理 学

ARM系列

ARM 7

ARM7

ARM ID

ARMIL

cortex --- 系列

MEPS 系列

CRISC处理器, 无互锁流水, 66的标微处理器

X86 系列

Power P(分)

Super H₂O₂

10) ARM 数据类型

数据类型 Byte

数据类型 bits

数据类型 Word

数据类型 32 bits

ARM 37 寄存器

64 个寄存器

31 个通用寄存器

R0-R7 通用

R13 SP 指针/栈指针

R14 保存调用返回地址

R15 程序计数器 PC

ARM - 异常: 由内部/外部原因引起, 异常发生时, CPU 暂停执行当前指令直到指定的向量地址处取指令并执行

异常时 R14 SPSR 保存前状态

异常返回时 SPSR → CPSR, R14 → PC

ARM 寻址空间 $2^{32} = 4G$

I/O 端口编址方式

存储器映射编址: I/O 端口地址与内存地址统一编址, 占内存空间

I/O 映射编址: I/O 端口与内存分开编址

, 不占用内存 64KB

小端数据格式

0x12345678

A0 A1 A2 A3

大端数据格式

0x78563412

A3 A2 A1 A0

ARM - 指令长度

ARM 指令固定 32 位长

Thumb 指令固定 16 位长

Thumb-2 指令 16 位/32 位长

右: MIPS 架构

总线

数据总线

地址总线

控制总线

(MB/s)

总线带宽 = (总线宽度/8) × 总线频率 (MHz)

总线宽度 32 位, 频率 66MHz, 则总线带宽 $32/8 \times 66\text{MHz}$

= 264MB/s

前: 多总线, 高速总线之间用桥接

2) 总线

① 片内总线 AMBA

AHB 高性能

: 组成主核, 从核, 仲裁器, 译码器

APB 低性能

APB 桥是 APB 仲裁器

② 片外总线 PCI

APB 桥

: 负责地址片选信号, 写/读/读/写, 命令信号 Enable

③ 串行总线

UART, I2C, SPI, USB

地址数据信号

UART 数据 8 位

I2C 数据 8 位 (PC485 CAN)

No.

Date.

存储器 SRAM 静态随机存储器

ROM 静态只读存储器

嵌入式非易失性 DOL

SPROM 串行可编程只读存储器

SD

DDR4M 2倍

MMC

QDRAM 4倍

第三章 嵌入式软件系统

(1) 软件：程序，相关数据及说明材料

(2) 软件特点：①是一种逻辑实体，有抽象性。

②无制造过程，可复制性

③对硬件有依赖性。

④无实度，效率低，成本高

(3) 软件分类 { 系统软件：操作系统

开发软件：开发工具、编译、设计器

应用软件

(4) 软件体系结构 { 驱动层：初始化程序，驱动程序

操作系统层：基础服务和库：文件系统、I/O系统、数据库管理

中间层：web, Java...

应用层

(5) 软件运行过程 { 上电复位

系统引导/启动

系统初始化

应用初始化

应用阶段

(6) 嵌入式操作系统 组成：内核，嵌入式TCP/IP网络系统，嵌入式驱动程序

内核：{ 任务管理：任务调度、创建任务、删除任务、设置任务优先级... (通常指操作系统内核)

内存管理

通信、同步和互斥机制

中断管理

时间管理

任务扩展

TCP/IP网络系统：提供TCP/IP协议标准，提供socket编程接口

嵌入式文件系统：文件存储检索更新加锁

uc/os 操作系统: uc/os 是十种嵌入式实时操作系统

可移植, 可固化, 可集成, 可扩展, 支持多任务, 中断管理, 信号服务等

关键技术: 如传递指向数据的指针, 不是真正的指针数据。

嵌入式操作系统发展态势 { 形成产业特性。

高可靠, 高可用, 低功耗。

适应不同嵌入式应用场合。

开发工具 { 需求分析工具

软件设计工具

编码, 调试工具

测试工具

配置管理, 维护工具

交叉开发系统

{ 宿主机 + 目标机构成, 包括文本编辑器, 文本编译器, 仿真器, 下载器

而硬件 → 交叉的物理连接

{ 串口
以太网
OCD 调试: JTAG + BDM

端设备 → 运行环境

交叉编译: 把编译宿主机上高级语言编译成可在目标机上的程序 (二进制的程序)

交叉调试: 调试器运行在宿主机, 而被调试程序运行在目标机。

可通过调试器查看修改目标机的内存, 寄存器... 调试器还外部通信控制被调试程序

方法: ROM-Monitor: 最直观, 但 ROM-Monitor 会占用目标机一定资源

OCD

不占用目标机资源, 但实时性不好

非交叉调试: Simulator

软件仿真器, 模拟目标机

第四章 任务管理与调度

① 嵌入式系统采用多任务处理方式：每个任务小，更易编写、调试、移植性、易扩大，实时性好

② 任务调度的基本单位 { 进程 process：由代码、数据、堆栈和操作系统内核组成，简称为可执行的文件

线程 thread：线程是进程内部一个相对地址的并发性。

在进程中所有线程共享该进程状态和资源

(使用线程优势：创建、销毁、切换、通信更快更简单)

③ 任务 { 动态性

并行性

异步独立性。

任务包括：代码

数据

堆栈

上下环境

任务包括程序，程序是任务组成部分

任务与程序区别：程序静态、任务动态

程序可对应多个任务

任务可创建任务

任务通常有一个无限循环不停

任务有生命周期

④ 任务状态：因为任务拥有的资源情况不断变化，任务状态也不断变化。

等待：任务等待事件发生

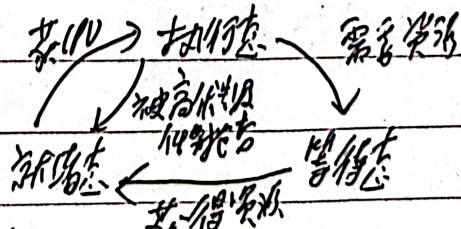
就绪：任务等待获得处理资源

执行：任务获得处理资源后，代码正在执行。

单处理系统：任何时刻仅有一个任务在CPU中执行。

任何可执行任务最终处于就绪状态。

任务状态的转移：(叫任务调度)



任务由任务与一个调度程序组成，调度程序为一时间。

任务管理：通过任务控制块实现

任务切换：任务任务上下文

- 时机
- 1) 中断, I/O
 - 2) 任务任务上下文资源
 - 3) 时间片轮转调度
 - 4) 高优先级任务就绪

任务管理

创建任务

删除任务

挂起任务

→ 等待状态

唤醒任务

→ 就绪状态

设置任务优先级

...

任务调度：任务执行顺序与任务CPU资源分配、任务执行时间控制

任务调度：调度程序从就绪队列中

按照什么原则分配CPU → 任务调度的算法

调度程序不运行任务，只是调度调用

何时分配CPU → 任务调度的时机

调用调度程序具体位置 → 调度点

何时分配CPU → 任务调度的时机

任务等待状态的转换

任务执行就绪

影响任务调度算法（需时间），没有考虑任何情况的调度算法

考虑下列因素

- CPU使用
- 输入/输出设备等待
- 作业时间
- 公平性
- 截止时间

任务调度分类

离线和在线

：离线调度不灵活/但不实时性、是时间驱动的因素

抢占和非抢占

：正在运行的任务可被其他任务打断，非抢占调度只有当任务主动放弃CPU资源才让出CPU

静态和动态

：静态调度在任务前任务排定优先级

最佳和次优

（大部分为抢占式）

内核可抢占：即使有较高优先级内核服务函数也能响应中断，中断服务程序退出时重新调度
 优先级最低的任务就绪，即立即运行，不需回到内核中等待

内核不可抢占：①在运行内核服务函数时响应中断，屏蔽中断

两种

② 执行内核服务函数时可响应中断，但中断退出后不进行任务重新调度

① 基于优先级的可抢占调度

非抢占/抢占 内核中断时 P98/99 U4

更高优先级任务就绪，则当前任务就绪运行，待当前任务执行完，CPU执行更高优先级任务

② 时间片轮转调度

(即更高优先级任务就绪时先执行更高优先级)

当两个或两个以上具有相同优先级的任务就绪时，任务调度按照任务就绪先后顺序调度第几个就绪

任务的一个时间片后执行下一个任务，依次类推，直至执行完一个任务

(使优先级相同的任务平等运行权利)

③ 优先级反转现象

定义：P106/125 U4

原因：

解决：优先级继承协议

优先级天花板协议

P106-125 第 P102-170

第六章 进程互斥通信

多进程系统中进程间的关系

（相互独立：仅利用CPU能力，
互斥：共享CPU外的资源，
同步：协调进程的执行顺序，
通信：交换数据）

同步、互斥控制

（信号量 Semaphore，互斥锁，
条件变量，
自旋锁，
消息队列，消息限制，
通信。）

信号量（互斥信号量：解决互斥问题：实现资源共享的同步访问）

↓
实现互斥的同步
二值信号量 解决同步问题
计数信号量 解决资源计数问题
（互斥信号量为1，表示资源是否进入临界区，但最多只能有一个进程进入临界区。
初值设为0，表示资源初始状态，但没有信号量，
并修改信号量 task 7 4 5 6 7
（对信号量有影响，防止子进程重复访问）
互斥信号量为0，其他信号量为1，可以访问。）
（允许任何进程访问）

消息队列

（进程间通信，前一个可以访问共享资源，
其他进程可以，允许任何进程访问）

消息队列 → 间接通信 → 有缓冲池 → 更好的传递消息的控制

管道 → 仅能单向通信

管道 → 不能接收多个信息 先进先出

共享信号 书 P201-204 111 P117-112
VS

异常中断管理

中断管理概述：处理器与I/O设备速度差异产生中断信号 方法：外设向处理器，由处理器向CPU
 硬件事件由CPU核发出并处理

异常中断概述：导致程序正常执行流程发生变化的事件。

自陷和异常中断

异常中断 { 原因：由CPU内部原因改变如 PPT U6 T4

异常中断(内部中断)

存储体访问的异常，异常事件如 P210。又分为内部中断。

异常中断(外部中断)

即如处理器所具有的外设接口，便可预知发生

异常中断(外部中断)

异常中断正在执行的程序发生变化。

同时，事件

异常 (CPU自动产生的内部，以处理异常事件。

异常分类 { 内部中断 外部中断

可屏蔽中断

由外设引发中断/异常中断

向量中断，直接中断，间接中断

异常中断中断时序 / 异常中断时序 / 异常中断时序

中断时序 139-40

异常中断/异常

中断时序

CPU中断时序

CPU中断时序

异常中断/异常中断 P145书

中断时序

异常中断时序

异常中断时序

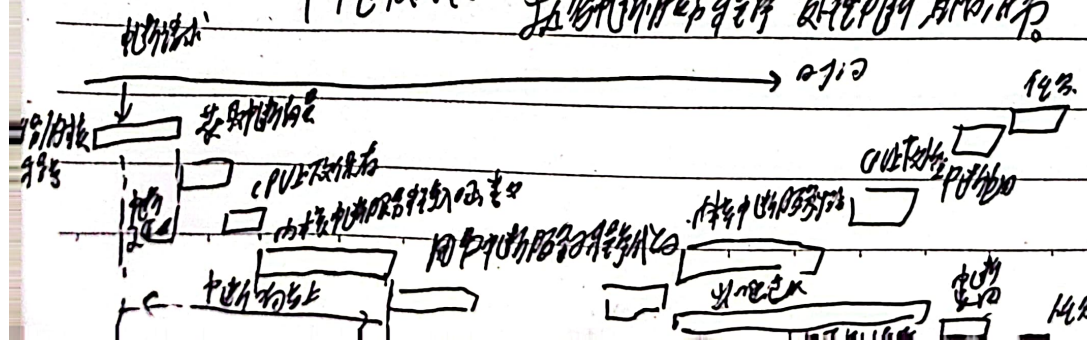
异常中断时序

异常处理流程 { 中断处理：异常中断(异常中断) 异常中断(异常中断) 异常中断(异常中断)

异常中断：处理器内部异常中断，异常中断(异常中断)。

异常处理：异常中断(异常中断) 异常中断(异常中断) 异常中断(异常中断)

异常中断(异常中断) 异常中断(异常中断) 异常中断(异常中断)



第八章 时间管理.

用途: 时间管理是计算机系统时间驱动 (周期性操作/定时操作).

{	相对时间: 2000	时钟源: RTC 实时时钟 → 1. 硬件时钟提供时间 周期性产生时间 而核上时钟 产生时间: 3.6E
	绝对时间: 管理当前时间	

RTC 实时时钟, 时钟周期保持时间的, 独立时钟芯片, → 初始化实时时钟.

时钟系统/时钟源: OS 时钟 (实时时钟) 系统时钟通过 tick 由时钟源提供
与实时时钟同步/时钟源产生的 tick 脉冲频率不同.

与系统时钟中的不同频率提供一个时钟源 clock tick

原因: 系统时钟从时钟源中取出时间单位, 即在每个 tick 时钟到来前又需要时钟源.

tick 抖动: 由于系统时钟源的抖动, 每次定时时钟变化的

时钟中断服务程序与系统时钟在定时时间导致延迟, 这个定时变化的 tick 抖动
导致 tick

(与 CPU 频率有关, 或系统时钟源 3.6E)

时钟源: Pcs/50. U7.

相对时间即系统时间: 1 个 tick 系统相对时间, 由内核把相对时间 → 绝对时间

时间等待处理: 在等待处理中, 每个 tick 相对时间等待处理中等待时间处理.

系统等待处理: 每个表项的值为 0 表示它没有处理的数值.

系统等待处理: 1 个 tick 只修改过系统时间.

→ 3.6E → 3.6E

当系统 0, 系统等待处理.

系统时间单位处理

系统时间单位处理.