嵌入式Linux驱动专题

北京迅为电子有限公司

嵌入式Linux驱动概述

- 学习方法：框架学习法

- 基本概念：也就是掌握uboot，Linux内核，文件系统等几个概念
  - Android
  - QT
  - UBUNTU

- 推荐看一下：“Linux驱动初探_精英版QQ群聊天记录.doc”

Linux驱动程序是什么？

- 基于OS的编程方式和单片机开发（裸机开发）截然不同
- 应用程序的执行是依赖于操作系统的
- 应用程序需要调用Linux操作系统的库函数来实现
- 驱动程序是联接操作系统和硬件之间的桥梁
- 应用程序不能直接操作硬件
- Linux的体系架构使得系统更稳定可靠

Linux驱动程序的基本认识

- Linux驱动包含三种：字符设备驱动、块设备驱动、网络设备驱动
  - 见开发板接口图
  - 大部分设备是字符设备

- 理解和掌握字符设备驱动是更重要的

Linux驱动程序讲解的步骤

- GPIO（General Purpose Input Output Port），通用输入输出口
- 内存管理单元（MMU)
- Linux把设备看成了文件（ open， read， write， ioctrl，close）
- Linux字符设备驱动的编程思想：做一组设备对应的驱动函数
- 设备节点（设备文件）
- 举例说明上层应用程序如何对某个外设进行操作
- 举例说明底层驱动的编写方式以及如何注册到系统（register_chrdev()）
- MISC杂项设备（混杂设备）
- 模块（module）

嵌入式Linux驱动专题(第二讲)

- 学会查看开发板原理图
- 学会查看处理器的数据手册（datasheet）
- 对外部设备操作的步骤（三部曲）

– 通过原理图找到设备连接的PIN脚

– 根据该PIN脚找到控制这个引脚的相关寄存器，并找到寄存器对应的物理地址

– 最后，通过编写程序来实现对该设备的操作

GPIO

•**GPIO （General Purpose Input Output Port）**

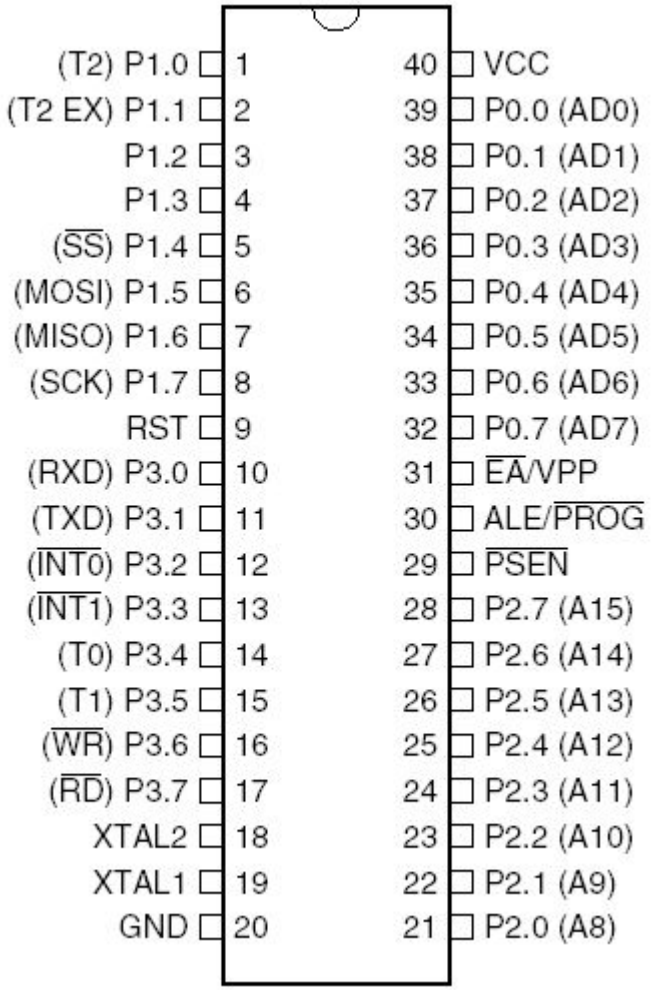
– 通用输入输出口

– 单片机或处理器对外设进行操作的主要方式

– 通过设置**GPIO**的高电平或者低电平来实现对外设的操作

– **GPIO**很多是复用的

•程序对寄存器操作即可实现对**GPIO**的操作



学会查看原理图

- 驱动开发工程师一定要学会看原理图
- **iTOP-4412**光盘当中有两种原理图
  - 核心板原理图（**TOPEET_coreboard4412_scp.pdf**）
  - 底板原理图（**ITOP4412_MAIN_CLASICS_V3_2.pdf**）
- 举例说明

学会查看数据手册（Datasheet)

- 每种芯片都有对应的数据手册
- 一般都是英文的
- 不需要每章都看，根据需要去查阅
- 举例说明

完成对外设的操作

- 在Datasheet中找到相关寄存器的章节，并确定物理地址
- 通过编写程序来实现对该寄存器的操作，最终控制相关硬件
- 对不同外设的控制，思路上是一样的

谢谢！

嵌入式Linux驱动专题(第三讲)

- 回顾上一节内容：对外部设备操作的步骤（三部曲）
- ARM处理器体系架构以及发展历程
- CACHE（高速缓存）
- MMU（内存管理单元）

CPU（中央处理器）

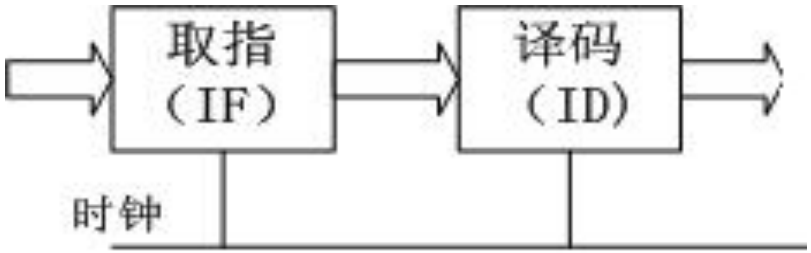
- 遵循 冯•诺依曼结构　‘存储程序’
- 说到底是个数字电路
- 对应一套指令系统
- 不断顺序取指令执行
- 哈佛结构

指令系统

- 软硬件界面
- RISC（精简指令集）/CISC（复杂指令集）
- RISC便于实现流水线，进而提高性能

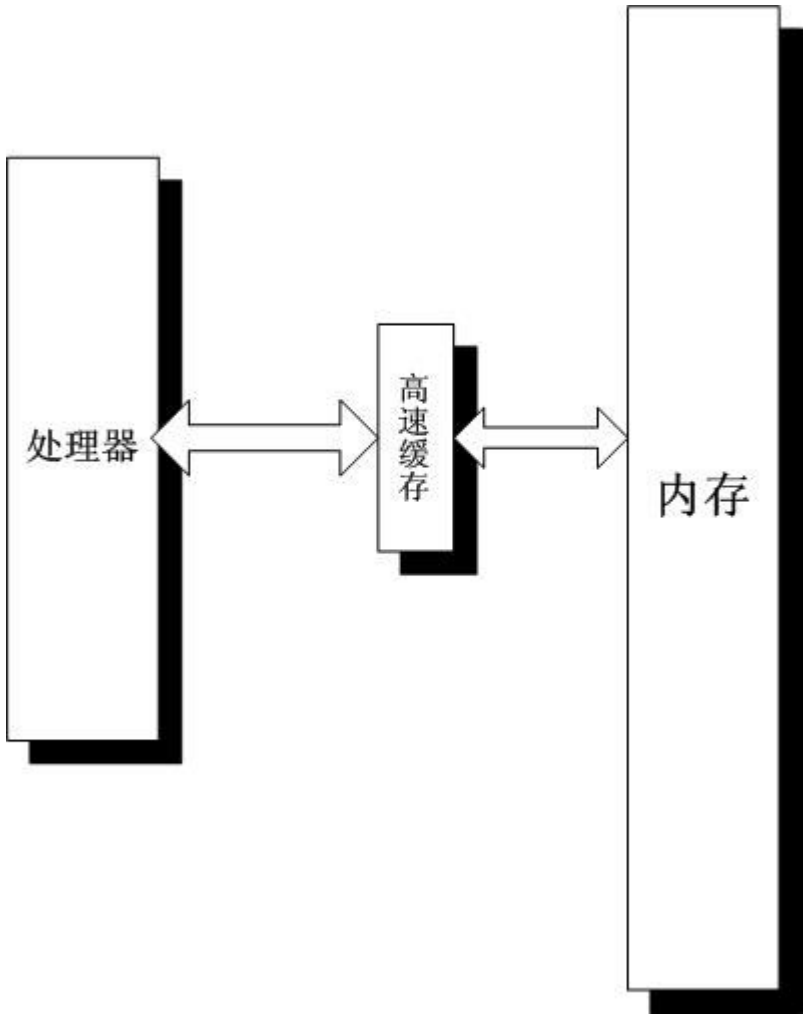
指令流水线

- 可以使一些需要计算机处理的多项操作在时间上重叠进行
- 便于提高电路的工作频率



高速缓存（cache)

- 位于CPU和主存之间的高速存储子系统
- 提高存储器的平均访问速度，从而使存储器的速度和CPU的速度相匹配
- 单位容量的价格很昂贵
- 处理器与存储器性能的差距仍在以每年50%的速度增大

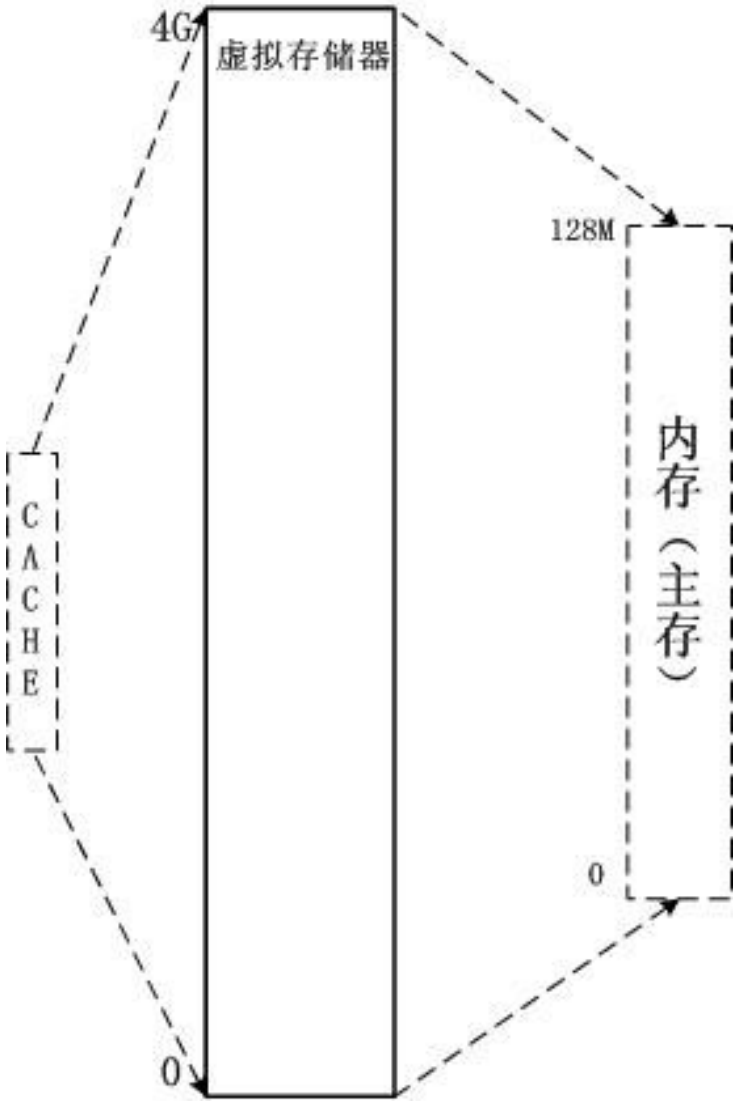


虚拟内存

- 处理器有了MMU，那么就有了虚拟内存的概念
- 虚拟存储器的空间大小取决于计算机的访存能力而不是实际外存的大小
- 使存储系统既具有相当于外存的容量又有接近于主存的访问速度

```
{
   int  *ptr,idata;
   …
   ptr = 0x1000a018;
   idata = *ptr;
   …

}
```



内存管理单元(MMU)

- **Memory Management Unit**
- **虚拟地址到物理地址的转换**
  - **如何通过物理地址找到虚拟地址呢？（ioremap)**
- **辅助实现虚拟内存**
- **辅助实现多任务管理**

MMU的功能

- 完成了从虚拟地址到物理地址的转换
- 《计算机系统结构》郑纬民 清华大学出版社

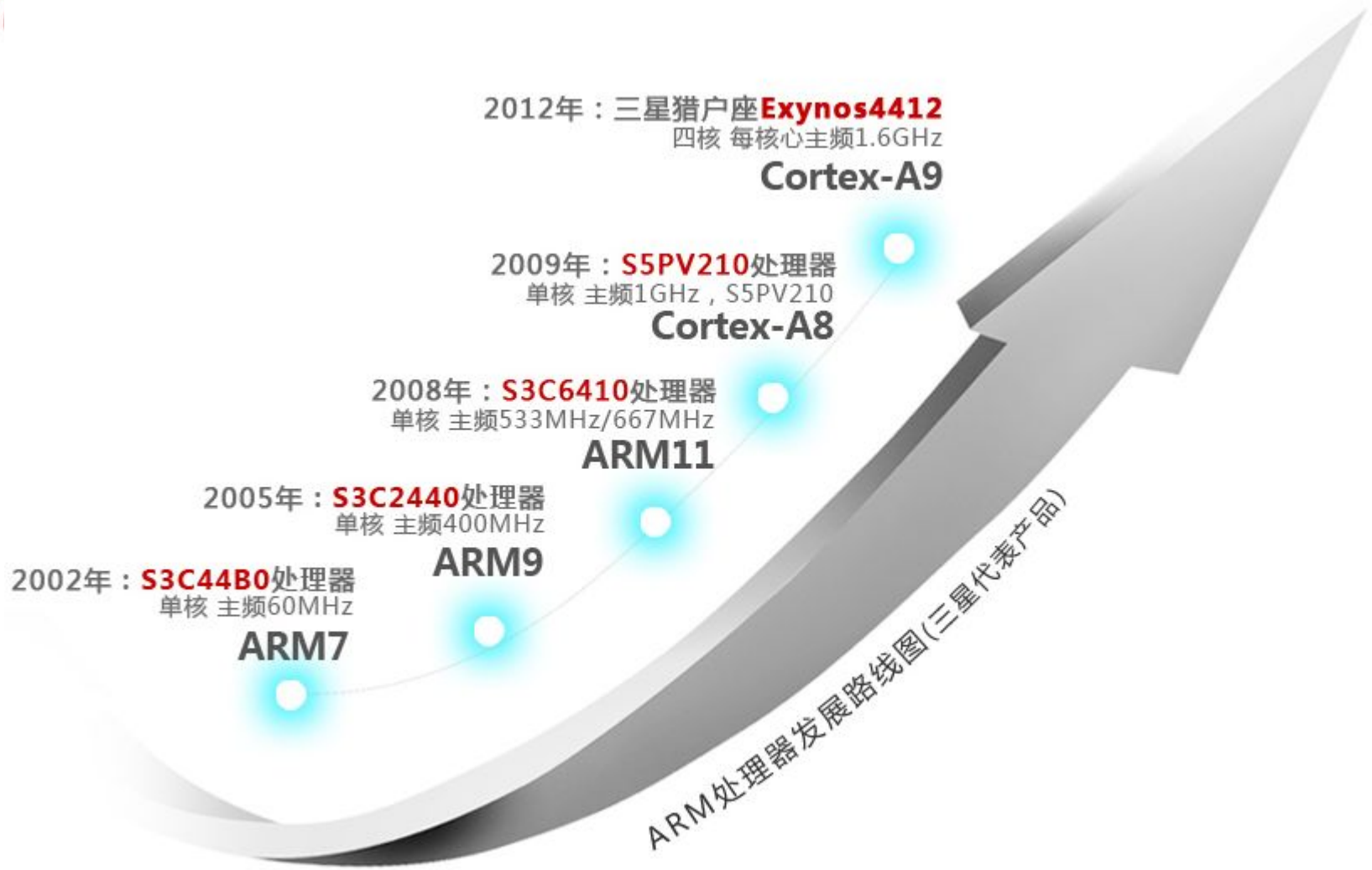
如何评价处理器

- 频率
- 性能 （MIPS/MHz）
  - ARM7 处理速度为0.9MIPS/MHz,常见的主时钟一般为20MHz-133MHz
  - ARM9 处理速度为1.1MIPS/MHz,常见的主时钟一般为100MHz-233MHz
  - CORTEX-A9处理速度2.5 MIPS/MHz，主时钟一般为1000MHz-1600MHz
- 功耗
- 面积（成本）

ARM介绍

- Advanced Risc Machines,Ltd
- ARM公司90年成立，只提供IP（知识产权）
- V3 到 V8 （架构相当于建筑的结构设计）
- 低功耗，低成本
- 大量使用寄存器，指令长度固定
- 领先的性能/功耗（MIPS/Watt）

ARM系列处理器



ARM的野心

主要ARM芯片供应商

- Samsung
- TI
- Freescale
- Intel
- NXP
- Atmel
- 联发科
- 华为
- 瑞芯微
- 全志

ARM体系结构发展（V3）

- V3结构　32位地址。

  T　Thumb状态：16位指令。

  M　长乘法支持（32*32=>64或者32*32+64=>64）。这一性质已经变成V4结构的标准配置。

- ARM7TDMI是1995年推出的该系列第一个处理器内核

ARM体系结构发展（V4）

- V4结构 加入了半字存储操作。

_

D 对调试的支持（Deb ug）
I 嵌入的ICE（In Circuit Emulation）

ARM7TDMI，ARM710T（ARM7TDMI核的处理器）
ARM720T（ARM7TDMI核的处理器）、
ARM9TDMI，ARM910T（ARM9TDMI核的处理器）、
ARM920T（ARM9TDMI核的处理器）

- ARM9系列于1997年问世

ARM体系结构发展（V5）

- 提升了ARM和Thumb指令的交互工作能力。
  E　DSP指令支持。
  J　Java指令支持。

- ARM10TDMI，ARM1020T（ARM10TDMI核处理器）。
  ARM9E，ARM9E-S（ARM9E可综合版本），ARM946（ARM9E核的处理器），ARM966（ARM9E核的处理器），ARM9EJ，ARM9EJ-S（ARM9EJ可综合版本），ARM926EJ（ARM9EJ核的处理器）

- ARM926EJ-S发布于2000年，ARM10发布于1999年

ARM体系结构发展（V6）

- 增加了媒体指令
  属于V6体系结构的处理器核有ARM11。ARM体系结构中有四种特殊指令集：Thumb指令（T），DSP指令（E），Java指令（J），Media指令，V6体系结构包含全部四种特殊指令集。为满足向后兼容，ARMv6也包括了ARMv5的存储器管理和例外处理。这将使众多的第三方发展商能够利用现有的成果，支持软件和设计的复用。

- ARM1136J-S发布于2003年 8级流水线

ARMv7及ARMV8

- Cortex-A8基于下一代ARMv7架构 （05年）
- Cortex-A9基于下一代ARMv7架构（08年）
- Cortex-M3是ARM公司于2004年底推出的首款基于ARMv7-M架构的处理器
- ARM 06年发布了新款Cortex-R4处理器

谢谢！