

# 迅为电子Linux驱动教程

## ——物理地址虚拟地址

北京迅为电子有限公司





## 本节目标

- 了解物理地址
- 了解虚拟地址

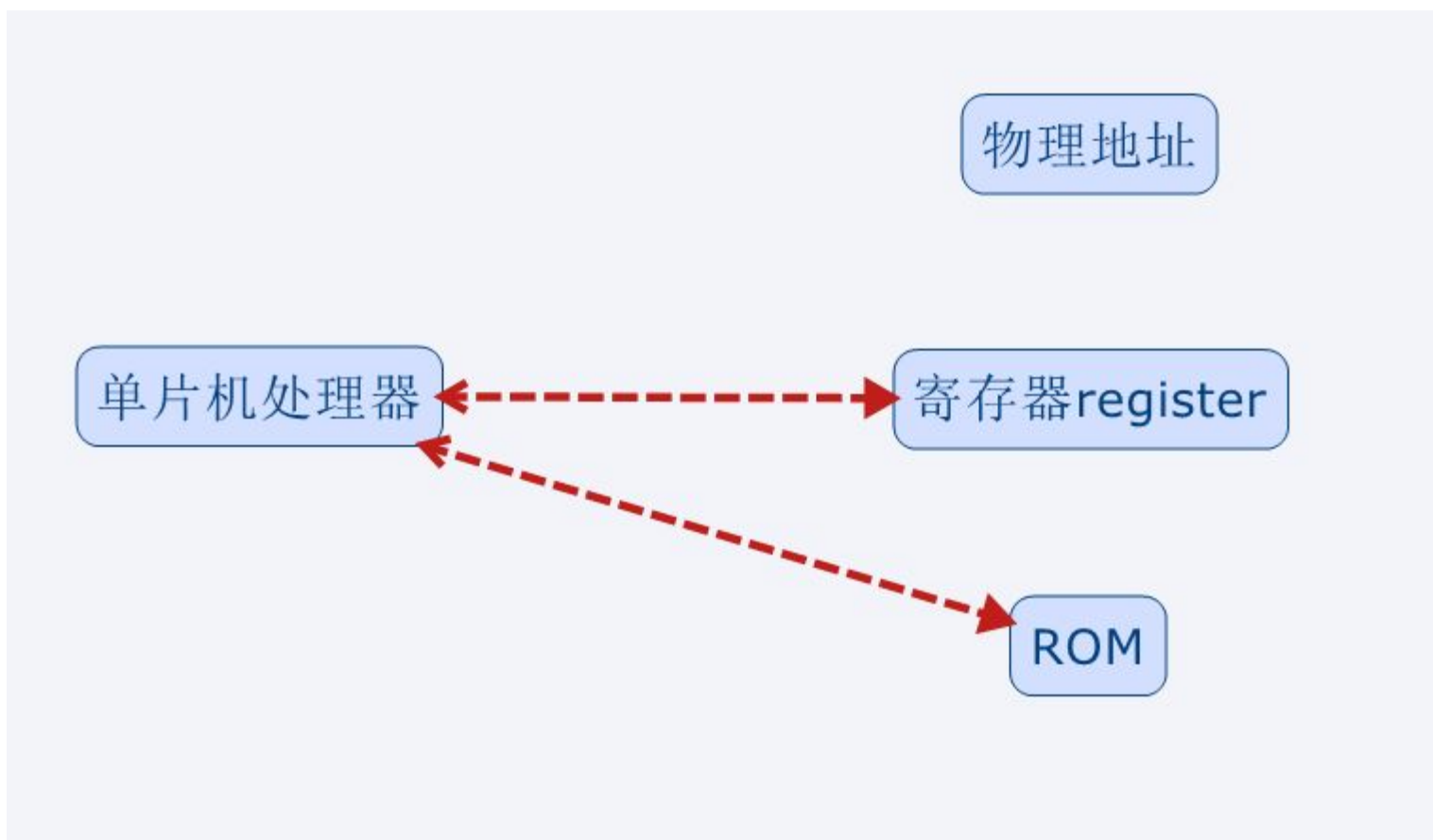


## 这两期视频教程的目的

- 让大家能够接受或者理解下面的内容
  - 对宏定义EXYNOS4\_GPL2(0)的操作就是对4412芯片管脚AC21(POP)寄存器的操作
- 上层程序员（包括驱动工程师）不用关心物理地址和虚拟地址具体是多少，只需要对一组宏定义操作，就是对4412内部寄存器操作

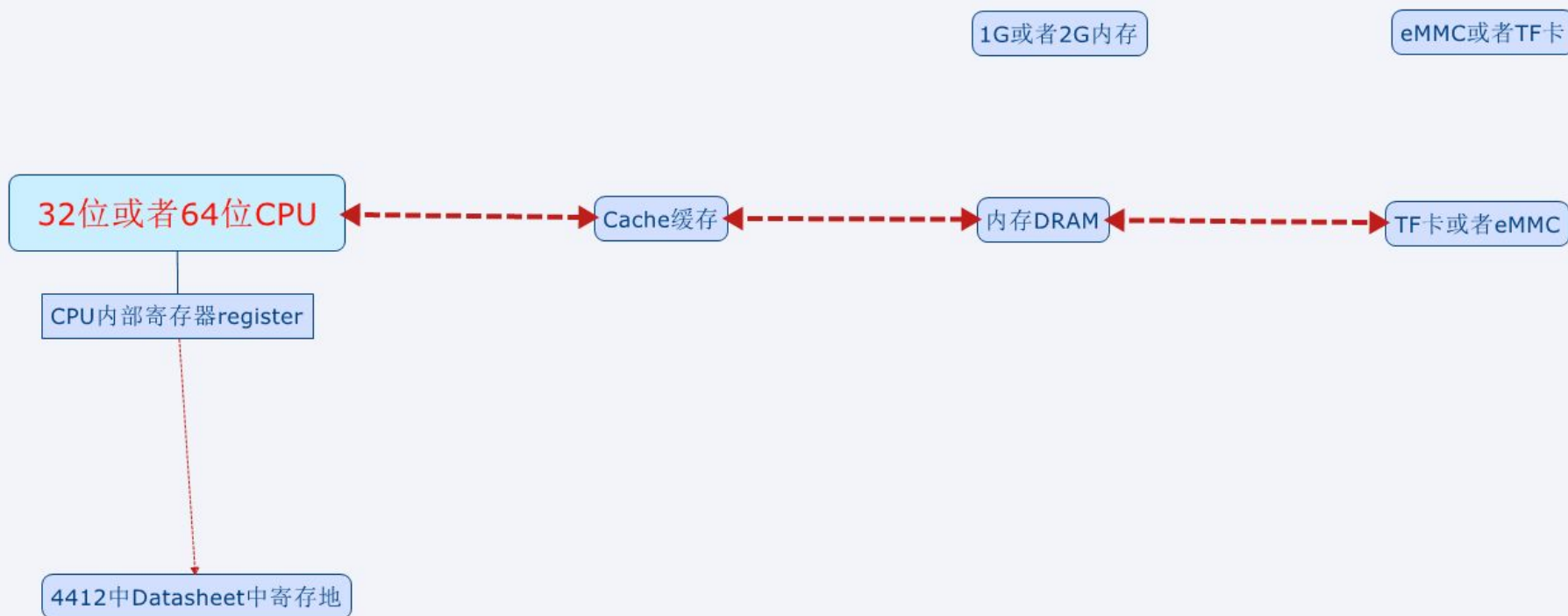


# 单片机处理器





# 现代中央处理器CPU





# 物理地址

- 4412中的物理地址
  - CPU内部寄存器register
  - eMMC
  - TF卡
- 从前面的内容看来，现代CPU和单片机就是相比多了缓存和内存管理单元，好像还是很好理解的。
  - 缓存比内存贵，速度快，所以它在挨着CPU的第一级
  - 内存比存储介质贵，速度快，所以它在缓存和内存之间



# MMU内存管理单元

## ——产生的历史背景

- MMU是中央处理器中用来管理虚拟存储器、物理存储器的控制线路，同时负责**虚拟地址映射为物理地址**
- 在原始的PC上，程序员编程的时候遇到一个问题
  - 程序大于内存
  - 当时的解决办法是程序员将程序“分割”成一段一段的，运行的时候CPU先运行一段，然后关闭前一段，再运行下一段。
  - 虽然运行的时候，CPU可以自动处理，但是“分割”的工作由程序员完成，很痛苦，很枯燥。
- 在这个环境下引入了内存管理单元——MMU



# MMU内存管理单元 ——地址范围

- 内存管理单元的基本思路
  - 数据和堆栈的大小总和可以超过物理存储器的大小，操作系统把当前使用的部分留在内存中，其它的保留在硬盘上。
- 内存管理单元——地址范围
  - 32位的CPU地址范围是 $0—2^{32}$
  - 64位的CPU地址范围是 $0—2^{64}$





# MMU内存管理单元

## ——页和页帧

- 虚拟地址空间被划分为页的单位，相应的物理地址空间被划分为页帧。这个就是分页机制
  - 页的大小必须一样，页帧的大小也必须一样
- 如果一台32位的机器有1G内存，它最大可运行4G大小的程序。
  - 这台机器首先要有4G的存储空间



# MMU内存管理单元

## —虚拟地址和物理地址的映射

- MMU本质上是一个表格
  - MMU表格一边是CPU发送指令对应的虚拟地址，一边存储的是物理地址
  - 虚拟地址转化为物理地址，内存通过物理地址去读取实际的数据
- 在4412处理器中，内部寄存器也是通过虚拟地址和物理地址映射之后才拿来使用的。



## 小结

- 大家可以通过网络的资料进一步了解这几个基本概念，这些概念对于我们操作GPIO有很大的帮助



谢谢！