

---

# ARM微架构

---

创客学院 武老师

# 流水线

## ■ 一个工人



第一个货物： 组装 包装 装箱

第二个货物：                    组装 包装 装箱

## ■ 三个工人



第一个货物： 组装 包装 装箱

第二个货物：        组装 包装 装箱

第三个货物：            组装 包装 装箱

第四个货物：                组装 包装 装箱

第五个货物：                    包装 组装 装箱

# 指令流水线

地址      内存

0x00

+

0x04

-

0x08

\*

0x0C

<<

0x10

&

0x14

|

...

... ..

总线

0x10

指令计数器PC

&

指令寄存器IR

<<

指令译码器

控制器

+

-

×

寄存器

运算器

取  
译  
执

# 指令流水线

| Cycle     |   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----------|---|---|---|---|---|---|---|---|---|---|
| Operation |   |   |   |   |   |   |   |   |   |   |
| ADD       | F | D | E |   |   |   |   |   |   |   |
| SUB       |   | F | D | E |   |   |   |   |   |   |
| ORR       |   |   | F | D | E | M |   |   |   |   |
| AND       |   |   |   | F | D | E |   |   |   |   |
| ORR       |   |   |   |   | F | D | E |   |   |   |
| EOR       |   |   |   |   |   | F | D | E |   | W |

F – 取指      D – 解码      E – 执行

# 指令流水线

→ △ 有助 类比 时间安排  
可在多级级

## ■ ARM指令流水线

ARM7采用3级流水线

ARM9采用5级流水线

Cortex-A9采用8级流水线

注1: 虽然流水线级数越来越多, 但都是在三级流水线的基础上进行了细分

## ■ PC的作用 (取指)

不管几级流水线, PC指向的永远是当前正在取指的指令, 而当前正在执行的指令的地址为PC-8

# 指令流水线

■ 指令流水线机制的引入确实能够大大的提升指令执行的速度但在实际执行程序的过程中很多情况下流水线是无法形成的比如芯片刚上电的前两个周期、执行跳转指令后的两个周期等所以指令流水线的引入以及优化只能使平均指令周期不断的接近1而不可能真正的达到1,且流水线级数越多芯片设计的复杂程度就越高,芯片的功耗就越高

| 地址   | 内存  |
|------|-----|
| N    | +   |
| N+4  | -   |
| 0x08 | *   |
| 0x0C | <<  |
| 0x10 | &   |
| 0x04 |     |
| ...  | ... |

(不会到3倍) (C中调用就是跳以这种情况不少)

# 多核处理器

## ■ 多核处理器

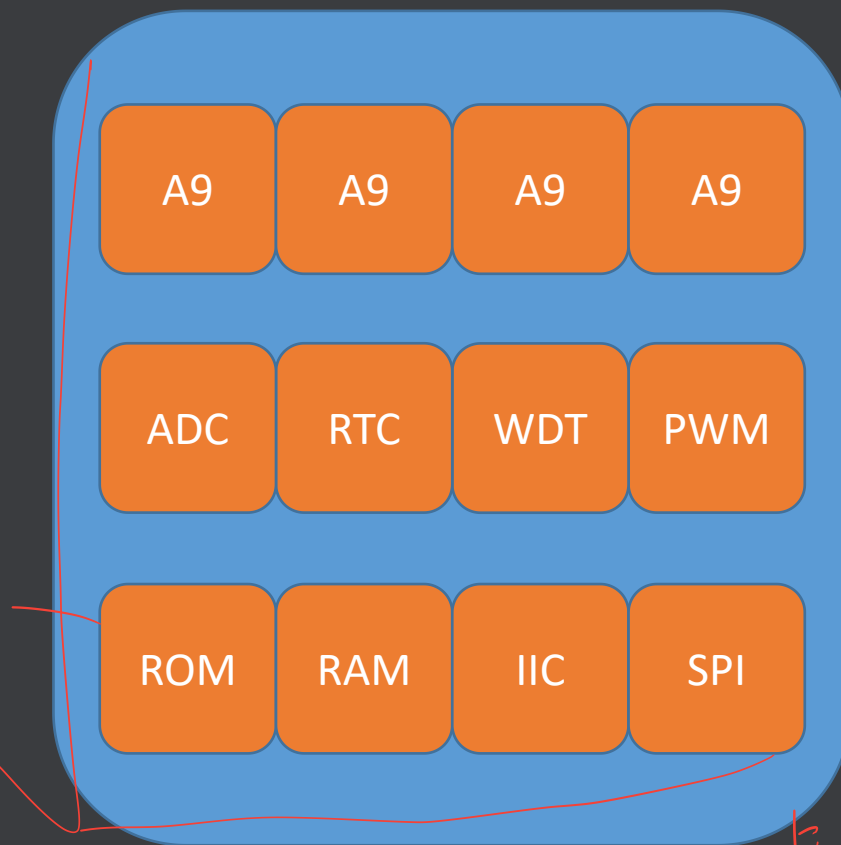
即一个SOC中集成了多个CPU核

## ■ 作用

不同的线程可以运行在不同的核心中  
做到真正的并发

## ■ 资源

多核处理器共用外设与接口资源



（觉确实 SOC  
是把内存、IO  
接口等能放过  
来就放过来了）

例学了上层也有助于学下层设计原因。（是不经总线？更快了？）

扫一扫，获取更多信息



THANK YOU