 在开发设计第一个ARM芯片时，当时的一些机器如Digital PDP-8、Cray-1和IBM 801在设计时早就提出了RISC的概念，并且在其后发展中有了许多RISC的特征，但RISC惟一的例子仍只有Berkeley的RISC I和II及Stanford的MIPS（Microprocessor without Interlocking Pipeline Stages，无互锁流水线处理器），而它们当时仅仅用于教学和研究。ARM处理器是第一个为商业用途而开发的RISC微处理器。ARM所采用的体系结构，对于当时的RISC体系结构既有继承，又有抛弃，即完全根据实际设计的需要仔细研究，没有机械照搬。ARM的体系结构中采用了若干Berkeley RISC处理器设计的特征，但也放弃了其他若干特征。这些采用的特征有： Load/Store体系结构 固定的32位指令 3地址指令格式 在Berkeley RISC设计采用的特征中被ARM设计者放弃的RISC的技术特征有： 寄存器窗口 在早期的RISC中，由于Berkeley原型机中包含了寄存器窗口，使得寄存器窗口的机制密切地伴随着RISC的概念，成为一般RISC的一大特征。Berkeley RISC处理器的寄存器堆中使用寄存器窗口，使得任何时候总有32个寄存器是可见的。进程进入和退出都访问新的一组寄存器，因此减少了因寄存器保存和恢复导致的处理器和存储器之间的数据拥塞和时间开销。这是拥有寄存器窗口的优点。但是寄存器窗口的存在以大量寄存器占用较多的芯片资源为代价，使得芯片成本增加，因此在ARM处理器设计时未采用寄存器窗口。尽管在ARM中用来处理异常的影子（shadow

）寄存器和窗口寄存器在概念上基本相同，但是在异常模式下对进程进行处理时，影子寄存器的数量是很少的。 延迟转移 由于转移中断了指令流水线的平滑流动而造成了流水线的“ 断流 ”问题，多数RISC处理器采用延迟转移来改善这一问题，即在后续指令执行后才进行转移。在原来的ARM中延迟转移并没有采用，因为它使异常处理过程更加复杂。 所有指令单周期执行 ARM被设计为使用最少的时钟周期来访问存储器，但并不是所有指令都单周期执行。如在低成本的ARM应用领域中普遍使用的ARM7TDMI，数据和指令占有同一总线，使用同一存储器时，即使最简单的Load和Store指令也最少需要访问2次存储器（1次取指令，1次数据读/写）。当访问存储器需要超过一个周期时，就多用一个周期。因此，并不是所有ARM指令都在单一时钟周期内执行的，少数指令需要多个时钟周期。高性能的ARM9TDMI使用分开的数据和指令寄存器，才有可能把Load和Store指令的指令存储器和数据访问存储器操作单周期执行。 最初的ARM设计最关心的是必须保持设计的简单性。ARM的简单性在ARM的硬件组织和实现上比指令集的结构上体现得更明显。把简单的硬件和指令集结合起来，这是RISC体系的思想基础；但是ARM仍然保留一些CISC的特征，并且因此达到了比纯粹的RISC更高的代码密度，使得ARM在开始时就获得其功率效率和较小的核面积的优势。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com