

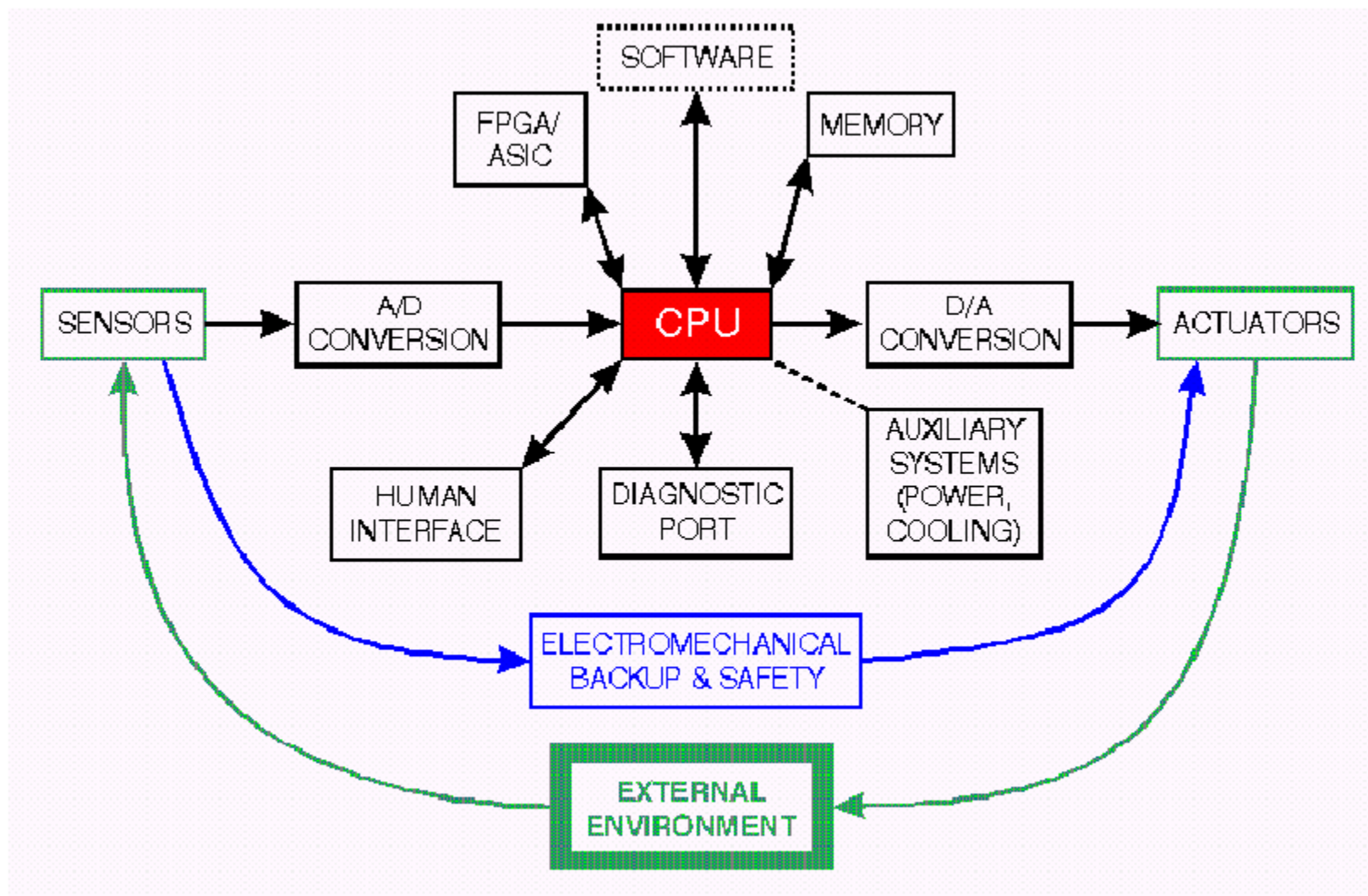
第二章

嵌入式微处理技术

2.1 概述

- 嵌入式微处理器是嵌入式系统的核心部件，深入了解和掌握嵌入式微处理器的相关技术是从事嵌入式系统分析、设计和应用的必然要求。本章主要介绍嵌入式微处理器的基本分类和典型技术，并对应用广泛的ARM嵌入式微处理器的历史作一简要梳理，后续章节会更多的讲解ARM处理器的概念、技术和新进展。

2.1 概述



2.2 嵌入式处理器的基本分类

- 嵌入式系统的核心部件是嵌入式微处理器。
- 嵌入式微处理器的分类随着集成电路技术的发展和分类方法的差异而有所不同。
 - 早期，一般将嵌入式微处理器分为两种：
 - 内部仅主要包含CPU核心组件算术逻辑单元、控制单元与寄存器的微处理器称为一般用途型微处理器或通用微处理器
 - 如4004, Z80, 8085, 8086等
 - 除了CPU核心单元外，还集成了ROM、RAM及I/O等组件于同一芯片内部的微处理器称为单片机或单芯片型微处理器（Single Chip Microcontroller）
 - 如TMS1000, COPS, 8051, 8096, 68200等。

2.2 嵌入式处理器的基本分类

- 21世纪初开始，人们根据嵌入式处理器的设计目标、性能、功能、应用场合及处理器发展历史，将嵌入式处理器约略分为五类(各类别间非不相交)：

- **嵌入式微控制器MCU（即单片机）**

- 嵌入式微控制器将整个计算机系统集成到一块芯片中，一般以某一种微处理器内核为核心，芯片内部还集成ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、WatchDog、I/O、串行口、脉宽调制输出、A/D、D/A、Flash RAM、E2PROM等各种必要功能和外设以适应不同的应用需求。
- MCU的最大特点是单片化，体积大大减小，从而使功耗和成本下降、可靠性提高。微控制器的片上外设资源比较丰富，适合于自动控制类应用。
- 嵌入式微控制器的典型产品有：8051、P51XA、MCS-251、MCS-96/196/296、C166/167、MC68HC05/11/12/16、68300等。

2.2 嵌入式处理器的基本分类

- 嵌入式微处理器单元EMPU (Embedded Micro Processor Unit)

- 第一种EMPU指的是嵌入式系统中使用的通用处理器，这些处理器并不是专门为嵌入式系统设计的，却被用于嵌入式系统中。在应用中，将微处理器装配在专门设计的电路板上，配上ROM/RAM等构成单板计算机，它只保留和嵌入式应用有关的母板功能，这样可以大幅度减小系统体积和功耗。
 - X86处理器（8086、80186、80286、80386，Pentium等）就是典型代表。
- 第二种EMPU是为各种嵌入式设备共用目的而专门设计的高性能专用处理器，为了满足嵌入式应用的特殊要求，这类EMPU虽然在功能上和通用微处理器基本相似，但在性能、功耗、体积、测试等方面进行了改进设计，工作温度、抗电磁干扰、可靠性等方面一般也做了各种增强。
 - ARM、PowerPC、ColdFire、MIPS、Imax等是这类嵌入式微处理器单元的典型代表。

2.2 嵌入式处理器的基本分类

- **嵌入式DSP处理器 (Embedded Digital Signal Processor)**

- 这种处理器是专门用于嵌入式系统数字信号处理的，它对系统结构和指令集做了特殊设计，具有专门的硬件乘法器，采用哈佛结构，支持流水线作业，使其适合于执行数字滤波、FFT、谱分析等方面的DSP算法，其程序编译效率高，指令执行速度快。
- 嵌入式DSP处理器有两个发展来源：
 - 一是DSP处理器经过单片化、EMC改造、增加片上外设成为嵌入式DSP处理器，如德州仪器TI的TMS320C2000/C5000等属于此范畴。
 - 二是在通用单片机或嵌入式片上系统(System On Chip) SOC中增加DSP协处理器，例如Intel的MCS-296和Infineon(Siemens)的TriCore等属于此类。
- DSP芯片的分类
 - 按基础特性（工作时钟、指令类型）分为静态DSP芯片和一致性DSP芯片。
 - 按数据格式分为定点DSP芯片和浮点DSP芯片。
 - 按用途分为通用型DSP芯片和专用型DSP芯片。

2.2 嵌入式处理器的基本分类

- **Soc嵌入式系统微处理器 (Soc Embedded System Processor)**

- SoC嵌入式系统微处理器是随着VLSI技术和EDA工具的发展而出现的一类单芯片嵌入式系统微处理器，它在一个芯片内部集成了更多更复杂的功能模块，这些功能模块除了包括微处理器核心（如ARM RISC、MIPS RISC、DSP等）外，还可能包括通信接口模块（如USB、TCP/IP、蓝牙接口等）、I/O接口模块、多媒体压缩解压模块等，这样做既大大缩小了系统所占体积，又加快了模块间信号传递速度，降低了外部噪声的干扰，减少了芯片引脚数，简化了制造过程，还可以通过改变芯片内工作电压，降低功耗。这些功能模块的成熟设计即是所谓的IP核（Intellectual Property）。

- **SoPC可编程片上系统 (System On Programmable Chip)**

- 这是Altera公司（现被Intel收购）首先提出的一种基于可编程逻辑器件的SoC解决方案，支持IP核的重用，方便系统集成，能够进行系统软硬件协同设计和验证，最大限度提高系统的性能，节约设计成本，加快设计速度。
- SoPC一般至少含有一个嵌入式处理器内核，一定容量的片内RAM资源，足够的片上可编程逻辑资源，处理器调试接口和FPGA编程接口等。
- Altera、Xilinx、Lattice、QuickLogic等全球重要的FPGA及EDA公司都有其SoPC系统解决方案。

2.2 嵌入式处理器的基本分类

- 教材中还另外列出了两类嵌入式微处理器：

- **嵌入式双核/多核处理器**

- 双核/多核处理器是这样一种处理器，其基于单个半导体的一个处理器芯片上拥有两个或多个计算引擎，这些计算引擎能独立或协同地完成各自的工作任务，从而提高处理器整体的计算性能。其优势在于可以解决通过提升工作频率来提高性能而引起的热障问题。
- 嵌入式领域典型的产品是双核/多核DSP处理器，其主要结构是一个或多个微控制器（MCU）核外加一个DSP核。
 - 例如：美国AD公司Blackfin系列产品中的ADSP-BF561 就是一种适合于多格式音视频、语音和图像处理的嵌入式双核DSP处理器，它可工作在两种模式之下：模式一，A核运行操作系统、网络协议栈和控制任务，B核运行信号处理的RISC/DSP任务。模式二，连个核按照SMP方式执行，信号处理任务可以被平均分配在两个核上执行。
 - 又如：杰尔系统公司（Agere System）的Vision X115就是一款三核DSP处理器，其ARM7核用于通信，其DSP16K核用于视频信号处理，ARM9 核则专门用于各种应用处理。该三核处理器能够处理每秒24 帧的1/4 VGA规范（320X240 像素）的视频信号流。

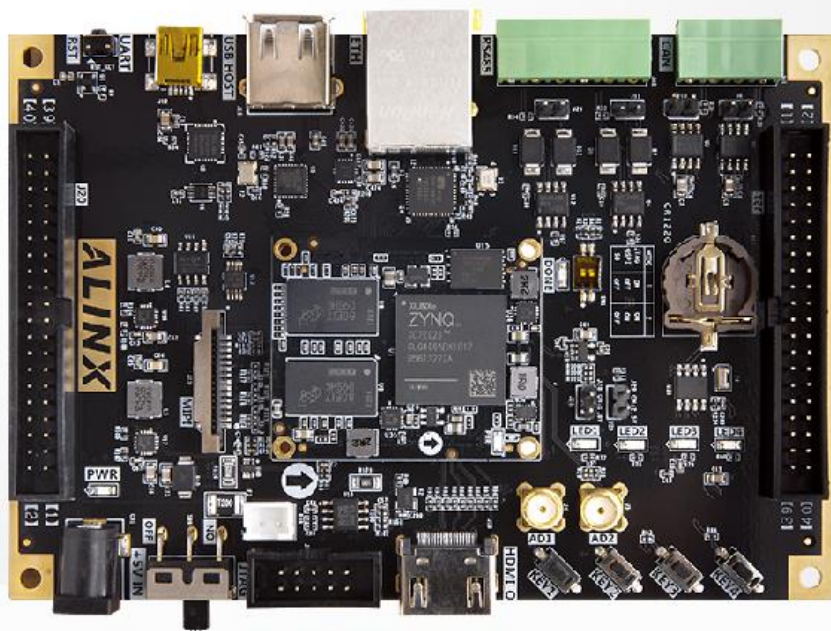
2.2 嵌入式处理器的基本分类

- 全可编程片上系统(可扩展处理平台)

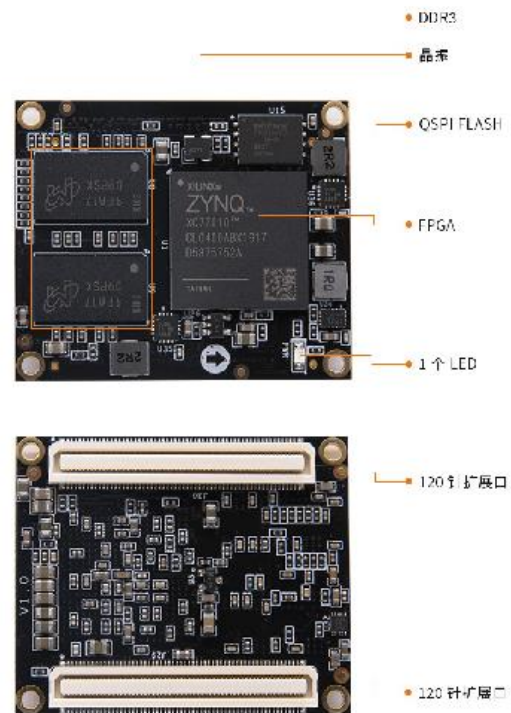
- 这也可归类为Soc或SoPC类别,单独列出概因其首先由Xilinx公司推出的Zynq-7000(S) FPGA芯片内集成了单个或多个软件可编程的ARM处理器硬核和一个硬件可编程的FPGA,这种片上系统故称为全可编程Soc。这种”处理器硬核+FPGA”的配置构成了一个有突破意义的可扩展的处理平台,开辟出一种新型嵌入式系统开发模式。
- Zynq-7000内部结构如下图,它将双ARM Cortex A9 MPCore处理器核与低功耗可编程逻辑和硬 I P 外设集成在同一个F P G A器件上。
- 研发人员可以在P C的 I S E集成开发环境下,用硬件描述语言编写自己的硬件电路,按照需要添加Xilinx公司的 I P 核,将自己编写的硬件电路和现成的 I P 核连接在Arm Cortex A9处理器的引脚上,编译连接生成固核,下载到Zynq-7000开发板上,使其成为一个数字计算机,然后即可完成配置,编写程序下载到该计算机上进行测试和运行,这样做可以快速地完成嵌入式产品原型开发,加速产品上市时间

2.2 嵌入式处理器的基本分类

Zynq-7000 EPP

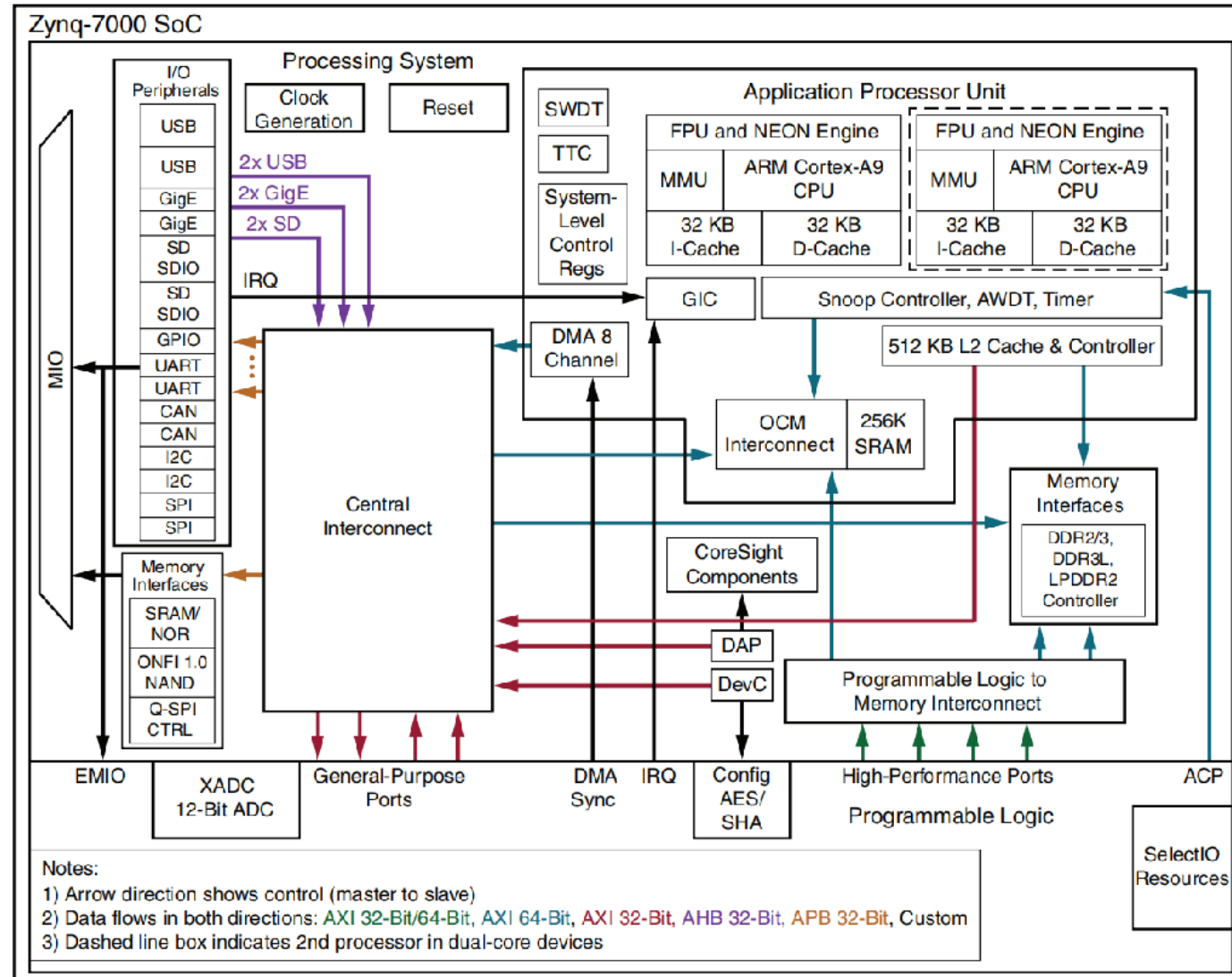


功能接口
INTERFACE FUNCTION



* 核心板详细了解可点击查看 >

2.2 嵌入式处理器的基本分类



DS190_01_070218

Figure 1: Architectural Overview

2.3 嵌入式微处理器的典型技术

嵌入式微处理器具有功能专用、低功耗和低成本的特点，这源于它采用了一些通用处理器中没有或不明显的技术解决方案，这些嵌入式微处理器的典型技术包括：

- 1) I/O端口统一编址与特殊功能寄存器；
- 2) 哈佛架构；
- 3) 桶型移位器；
- 4) 正交指令集；
- 5) 双密度指令集；
- 6) 看门狗定时器；
- 7) 边界对准与端序；
- 8) 地址重映射；
- 9) FIFO缓冲寄存器；
- 10) 主存控制器。

2.3.1 I/O端口统一编址与特殊功能寄存器

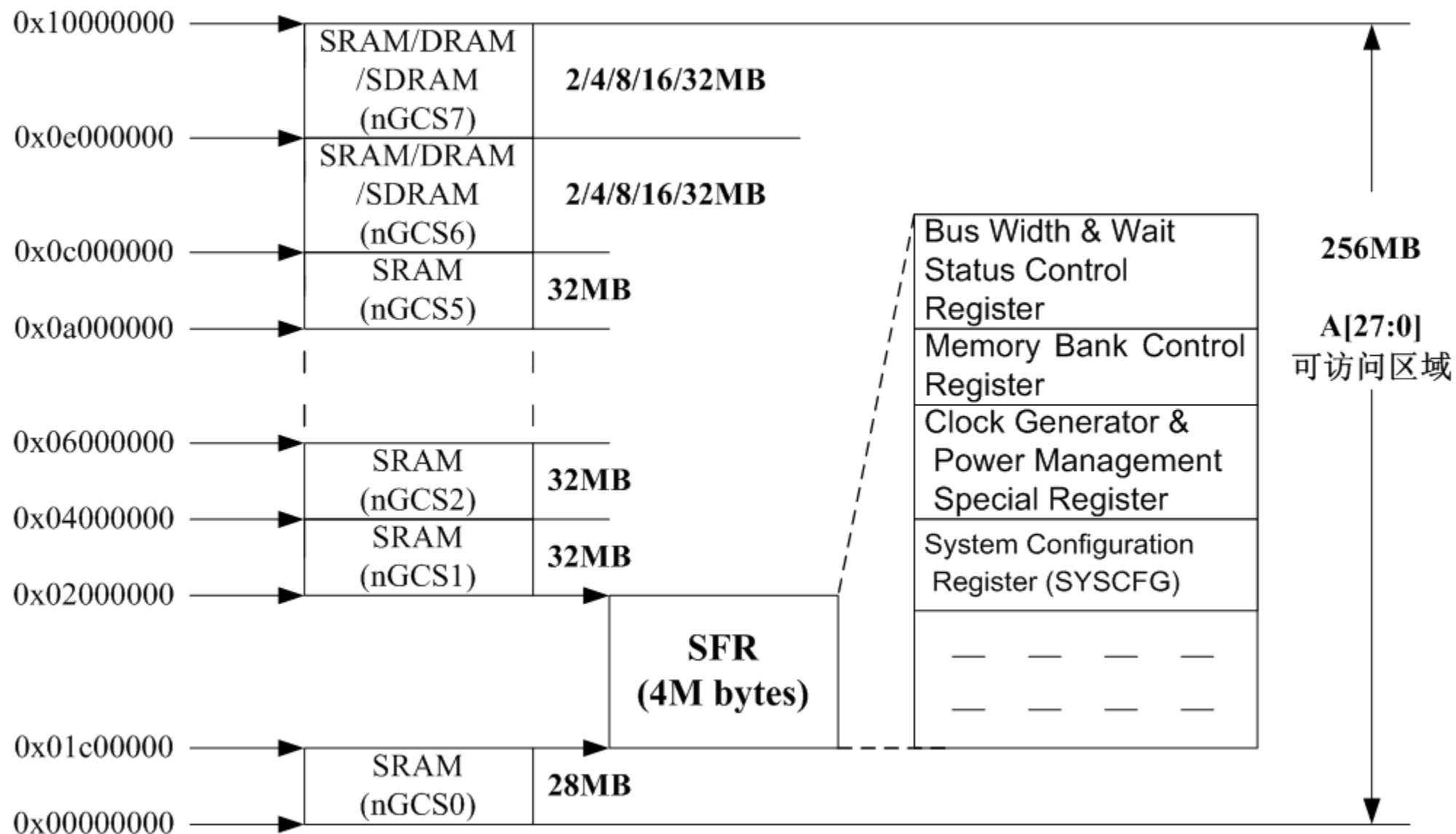
- 计算机的两种地址空间

- 主存储器地址空间，简称主存地址空间，由地址线数量决定。
- I/O端口地址空间，简称I/O地址空间，与地址线数量无直接关系。

- I/O地址空间的两种编址方案

- 与主存地址线无关的独立编址方案
 - 如x86处理器系列：32根地址线的主存寻址空间是4GB，MOV指令只访问这些存储单元。而I/O地址空间是另外独立的64K个字节单元，这些I/O端口的数据寄存器或控制逻辑端口内容需专门的IN/OUT指令来实现读写操作。
- 与主存地址线合在一起的统一编址方案
 - 如8051系列和ARM系列处理器的I/O编址（见下图）。其I/O地址空间占用部分主存空间，故I/O端口数据的存取和主存单元数据存取使用同一种数据传送指令。

2.3.1 I/O端口统一编址与特殊功能寄存器



2.3.1 I/O端口统一编址与特殊功能寄存器

- 特殊功能寄存器（SFR）

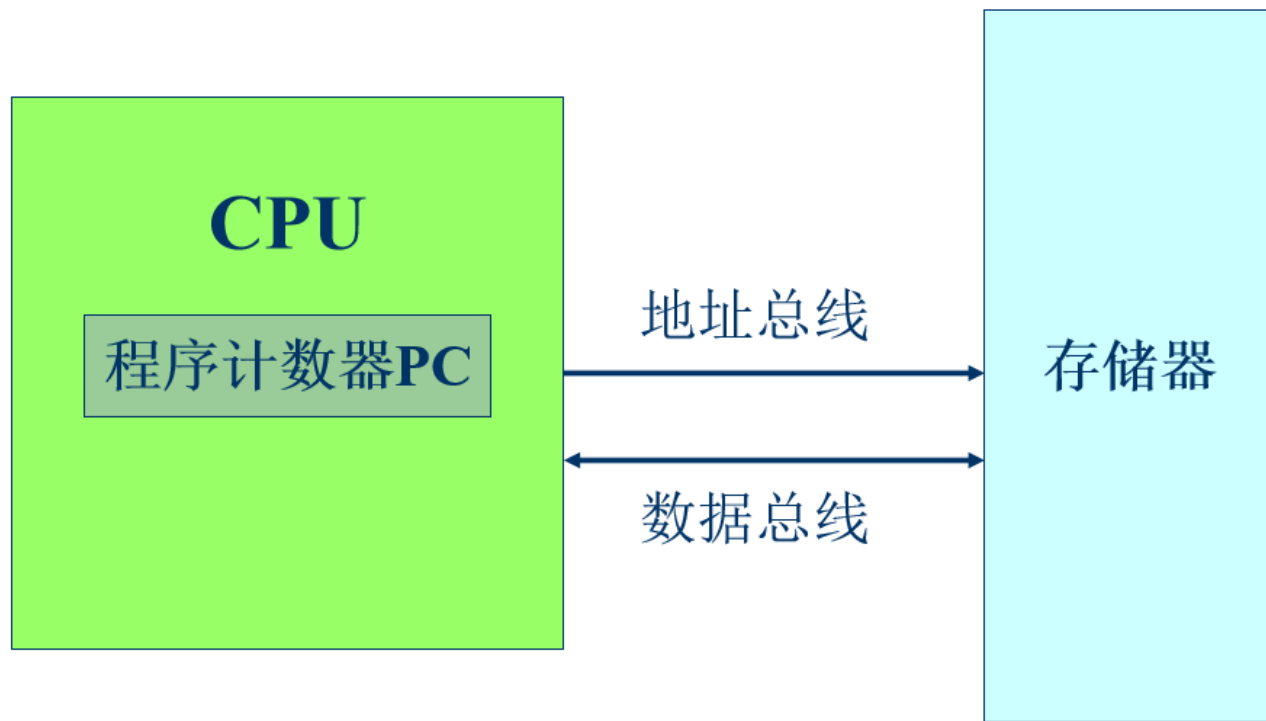
- SFR就是接口控制器的端口寄存器
 - 计算机系统少不了I/O端口和I/O接口，如UART接口、LCD控制器接口、中断控制器接口等，这些接口模块都集成了各自的I/O端口寄存器，用于在CPU和接口之间传递数据、状态和控制信息。
- SFR实际为片上（片内）的物理存储单元
 - 8051与ARM处理器的这些端口寄存器就集成在存储器芯片内，与存储器单元划分区段统一编址。
 - 与普通的内存单元相比，这些端口寄存器的特殊之处在于：
 - 单元地址的存储位数特殊，可能是8比特，也可能使其它位数如3或4比特。
 - 功能特殊，不仅用于端口数据的读写，还可表示端口的状态信息和控制信息。
- 在前述三星公司S3C44B0X处理器的例子中，地址0x01C00000到0x01FFFFFF之间的4MB空间就是SFR的地址空间，这中间安排了S3C44B0X所有片内外设控制器的数据端口、控制端口和状态端口，每个端口特殊功能寄存器占用32比特，每一个外设控制器的SFR端口地址和比特位含义与功能都有详细的定义。

2.3.2 冯·诺依曼结构与哈佛结构

- 冯·诺依曼结构是一种计算机体系结构, 也叫普林斯顿结构。
- 冯·诺依曼结构基本特点:
 - 计算机只有一个主存储器, 程序指令存储和数据存储分开都存放在同一存储器中。
 - 只有一种访问主存储器的指令, 且仅在取指周期从主存储器中读出的数据才是机器指令。
 - CPU通过数据总线与存储器交换信息。
- 通用计算机通常采用这种冯·诺依曼结构。

2.3.2 冯·诺依曼结构与哈佛结构

冯·诺依曼结构示意图



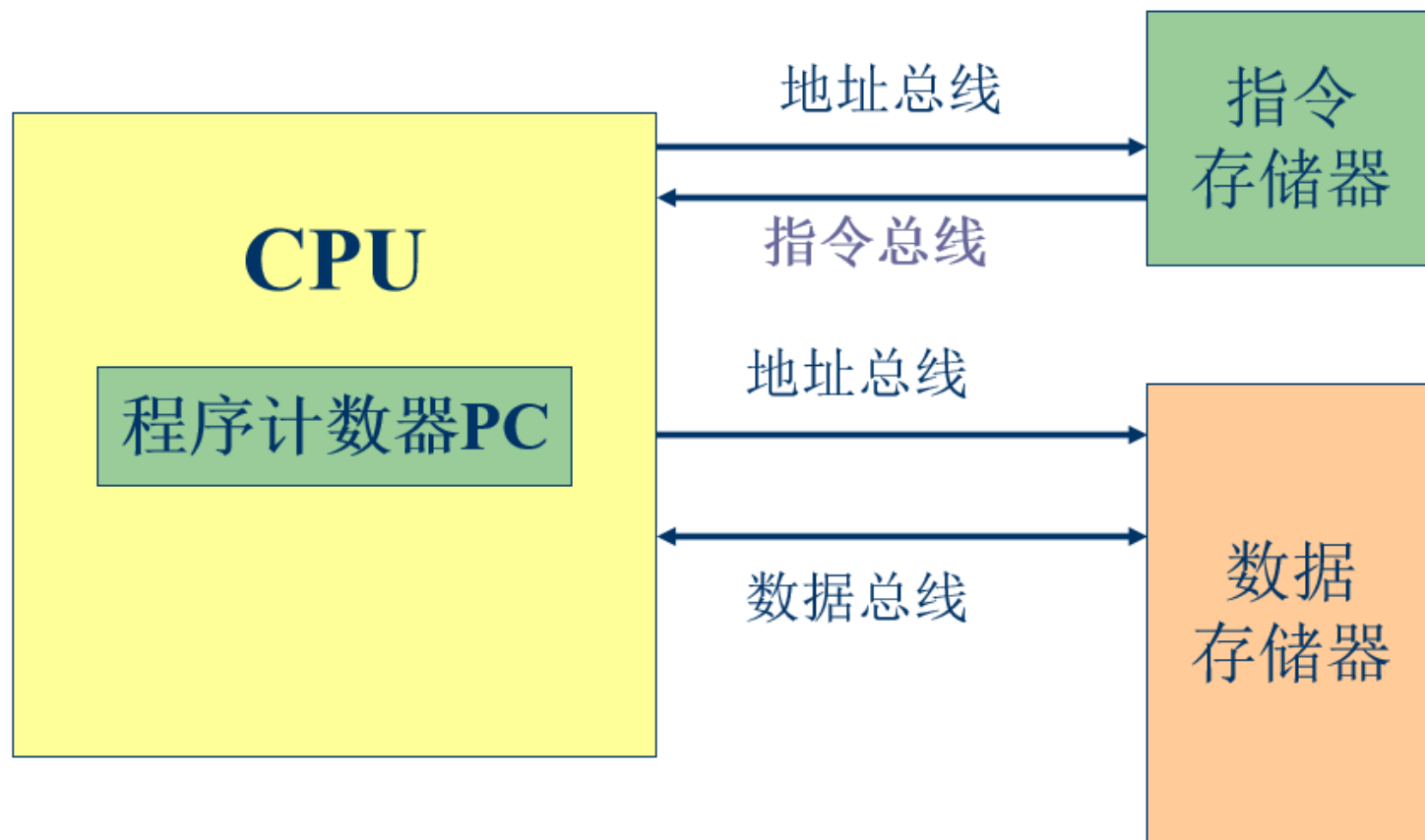
2.3.2 冯·诺依曼结构与哈佛结构

哈佛结构计算机

- 哈佛结构也是一种计算机体系结构。
- 哈佛结构基本特点：
 - 程序指令存储和数据存储分开的存储器结构。
 - 冯·诺依曼结构中的数据总线在哈佛结构中被分为指令总线和数据总线。
 - 可以使指令和数据有不同的数据宽度。
 - 如Microchip公司的PIC16芯片的程序指令是14位宽度，而数据是8位宽度。
 - 优点：
 - 较高执行效率和数据吞吐率。
- 嵌入式计算机通常采用哈佛结构。
 - 例如：MCS-51 ， MC68， Z8， ARM9/10/11等。

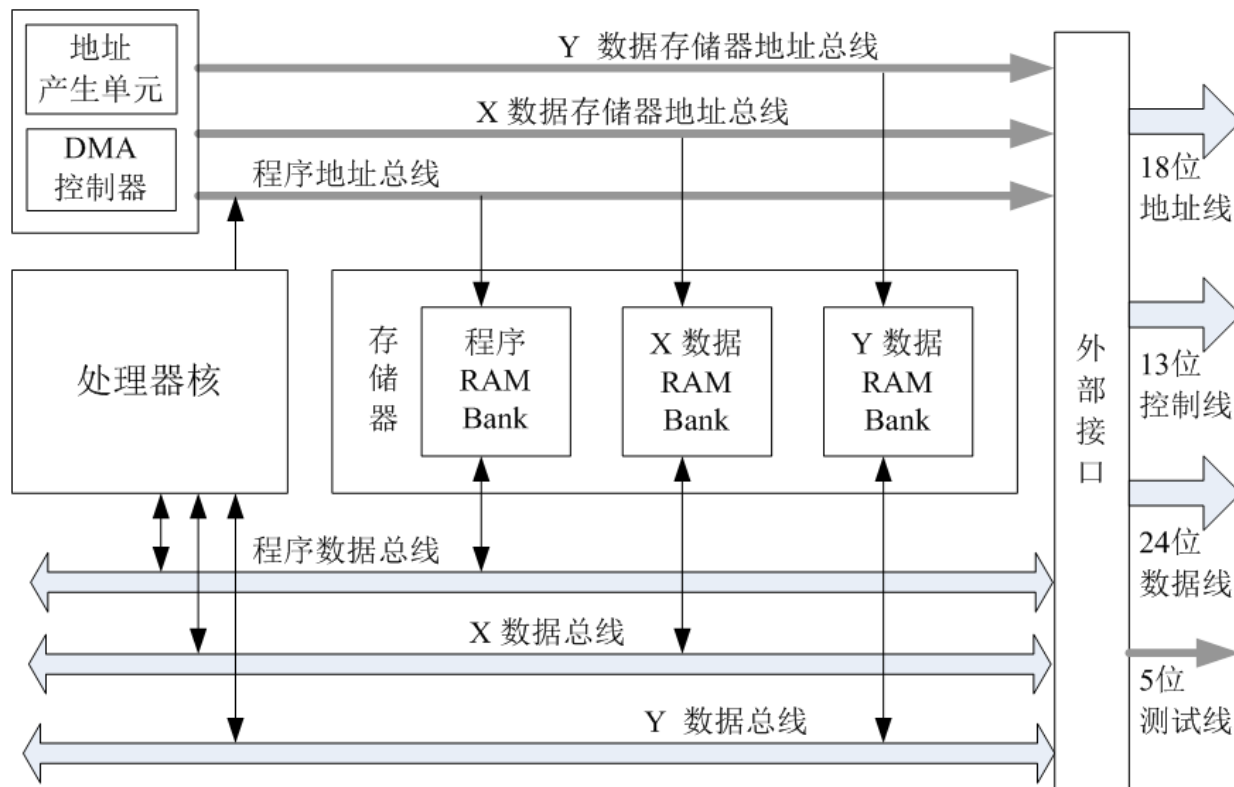
2.3.2 冯·诺依曼结构与哈佛结构

哈佛结构示意图



2.3.2 冯·诺依曼结构与哈佛结构

Motorola公司DSP56311型数字信号处理器的存储系统结构



2.3.3 桶型移位器

- 移位操作是计算机CPU内部的重要运算之一，一般由连接在算术逻辑部件ALU内部或外部的移位器来执行，移位器用硬件实现。
- 传统计算机的ALU输出端附加移位控制逻辑的结构属于ALU内部移位器，它只能完成直送、左移一位、右移一位的操作。传统计算机的ALU外部移位器在一个时钟周期只进行一位右移、左移、循环右移、循环左移的操作。
- 传统移位器如果需要移动多位就需要消耗多个时钟周期，效率较低，于是人们研究出了一个时钟周期内可以进行字宽限度内任意位数移位或循环移位操作的硬件移位器，这种移位器就是桶型移位器（barrel-shifter）。
 - 32位桶型移位器最早在80386 处理器上得到使用，作为常用IP核，广泛集成在微控制器和DSP处理器内。

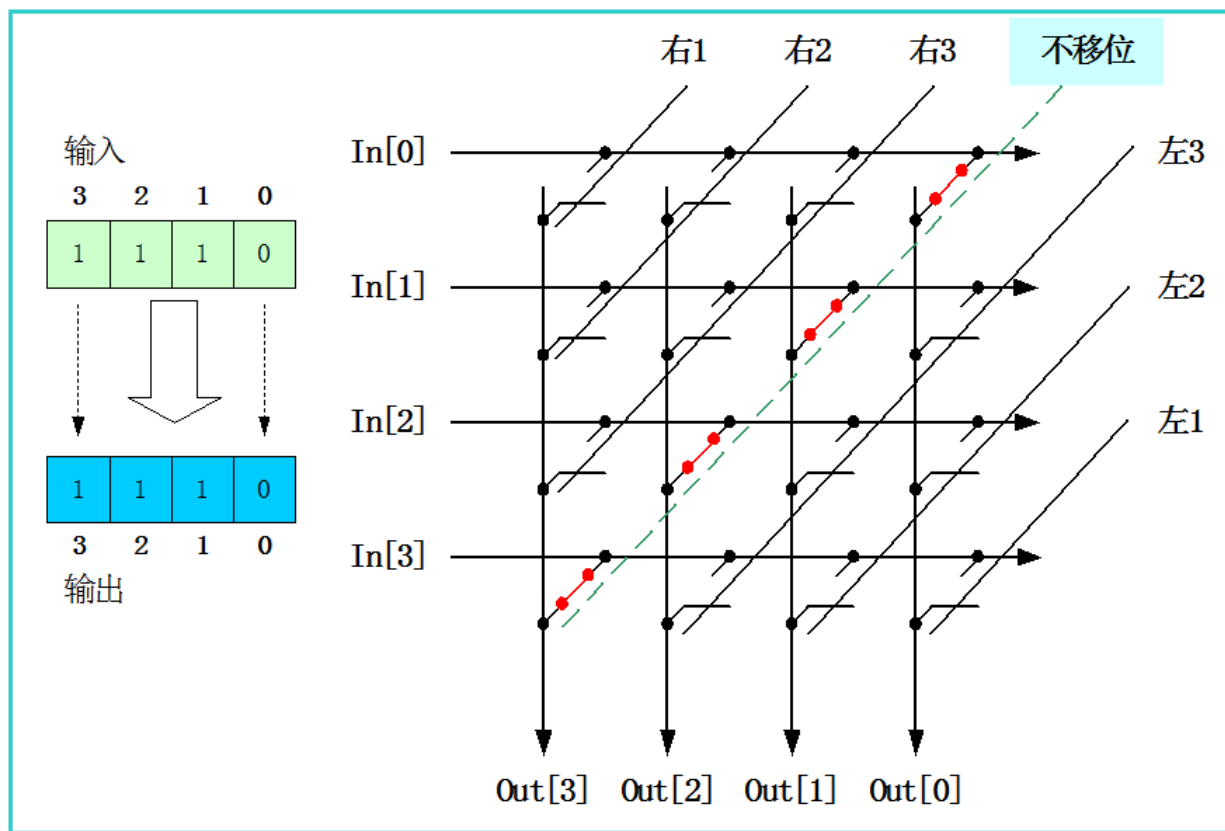
2.3.3 桶型移位器

桶型移位器的工作原理

- 桶型移位器一般位于ALU前端。
- 进行桶型移位操作的机器指令需指明移动方向（左、右）、移动类型（循环、算术、逻辑）和移动位数。
- 利用电子开关矩阵和一组组合逻辑电路可实现一个时钟周期内输入数据到任意位移位输出的变换。

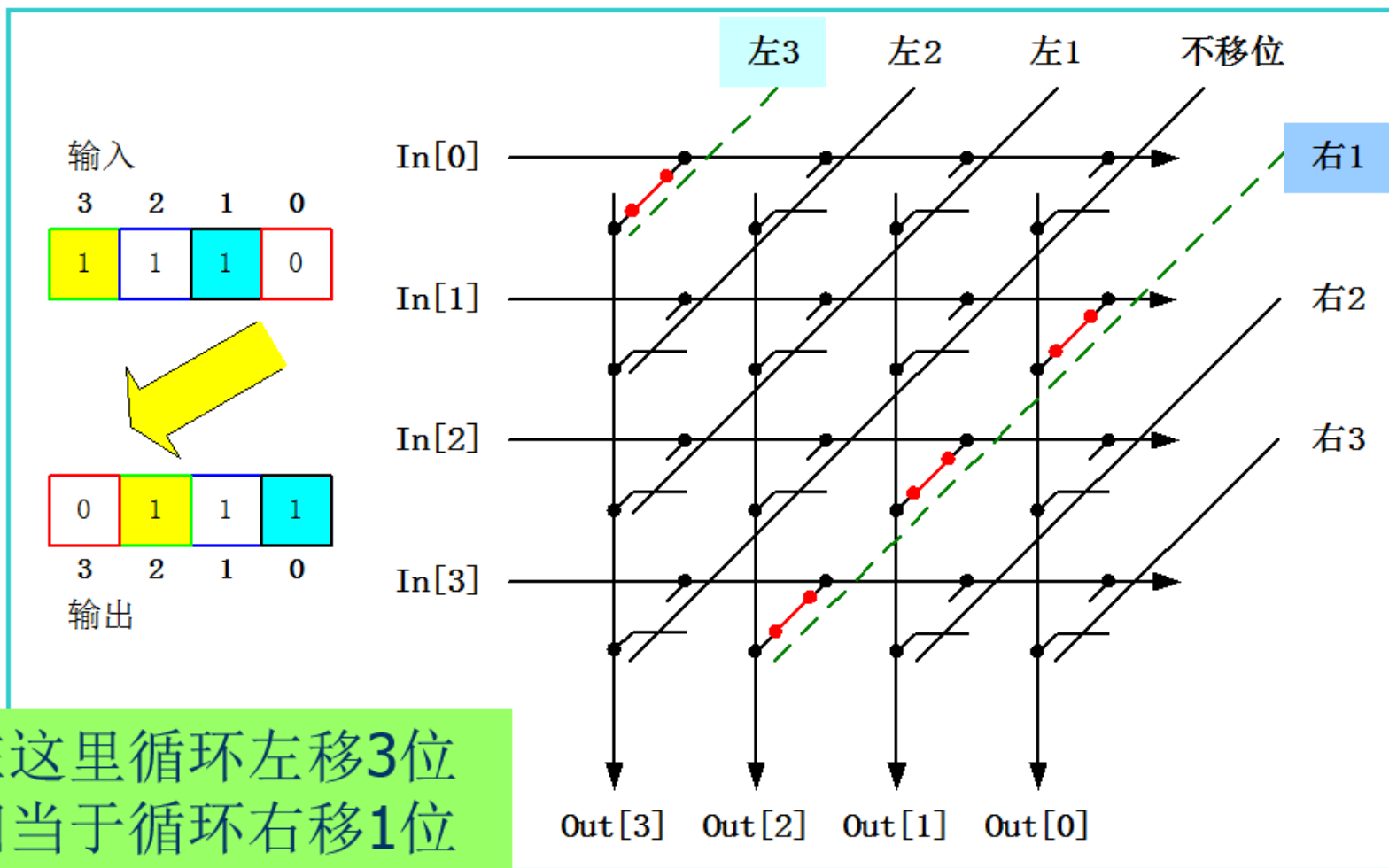
2.3.3 桶型移位器

不移位操作示意图



2.3.3 桶型移位器

循环左移3位操作示意图



2.3.4 正交指令集

- 正交指令集是这样一种指令集，其绝大多数指令长度相同，指令的操作码和操作数寻址字段的长度相对固定，在寻址字段中，所有寄存器的寻址可替换使用，指令的操作码、寻址方式、第一操作数和第二操作数四个字段的取值相互独立。
 - 例如：PDP-11， 680X0， ARM , VAX等处理器的指令集。
- 例2-1：一个正交的两地址指令集中的运算类指令常采用以下格式：
定长操作码 + 寻址方式编码 + 定长格式的目的地寄存器集 + 定长格式的源寄存器集
- 例2-2：ARM处理器有16个通用寄存器，分别命名为R0到R15。ARM处理器的数据处理类指令中的立即数移位指令格式如下：

d31~28	d27~25	d24~21	d20	d19~16	d15~12	d11~7	d6~4	d3-0
执行条件	000	操作码	S	Rn	Rd	#shift	SH	Rm

2.3.4 正交指令集

- 例2-2的说明

- 其中的执行条件字段是该指令执行的条件，共有15种条件，加上无条件，一共有16种可能，用4位二进制数表示，可以取其中一种；操作码字段有4位，表示执行的是哪一种算术逻辑运算，一共有16种运算，可以取其中任何一种；SH字段表示移位类型，可以有5种移位指令的编码可供选择，这些指令都用到了桶型移位器；Rn、Rd、Rm字段是操作数寄存器字段，可以取16个寄存器R0~R15中的任何一个；#shift是立即数移位长度值。

- 正交指令举例：

- SUBNES R10, R5, R2, ASR #5
- 这条指令的操作码是SUB, 即减法运算；NE是条件标志，表示不相等；S表示本指令执行完毕后更新标志位；第一操作数Rn是R5；第二操作数表示将Rm的内容算术右移动5位，即R2的值除以32；目标寄存器Rd是R10。指令执行 $R10 \leftarrow R5 - R2/32$ 。

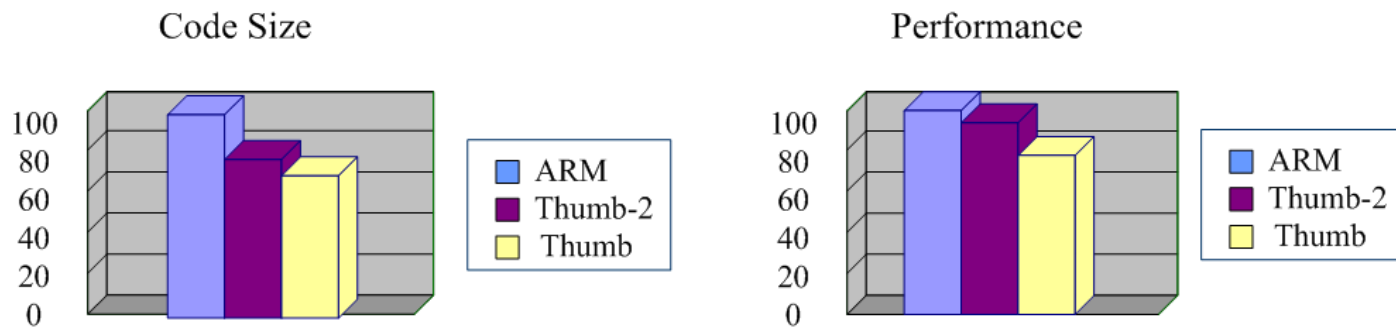
2.3.5 双密度指令集

- 指令密度定义
 - 对同等机器指令操作序列，单位内存空间所容纳的机器指令数。
- 指令密度是计算机指令集的一个技术指标，与处理器的体系结构密切相关。
- 指令密度有高密度指令和低密度指令之分。与低密度指令相比较，高密度指令执行同样的机器指令序列所需要的指令存储空间较小。
- 双密度指令集处理器：指令集包含有指令密度不相同的两套指令集。

2.3.5 双密度指令集

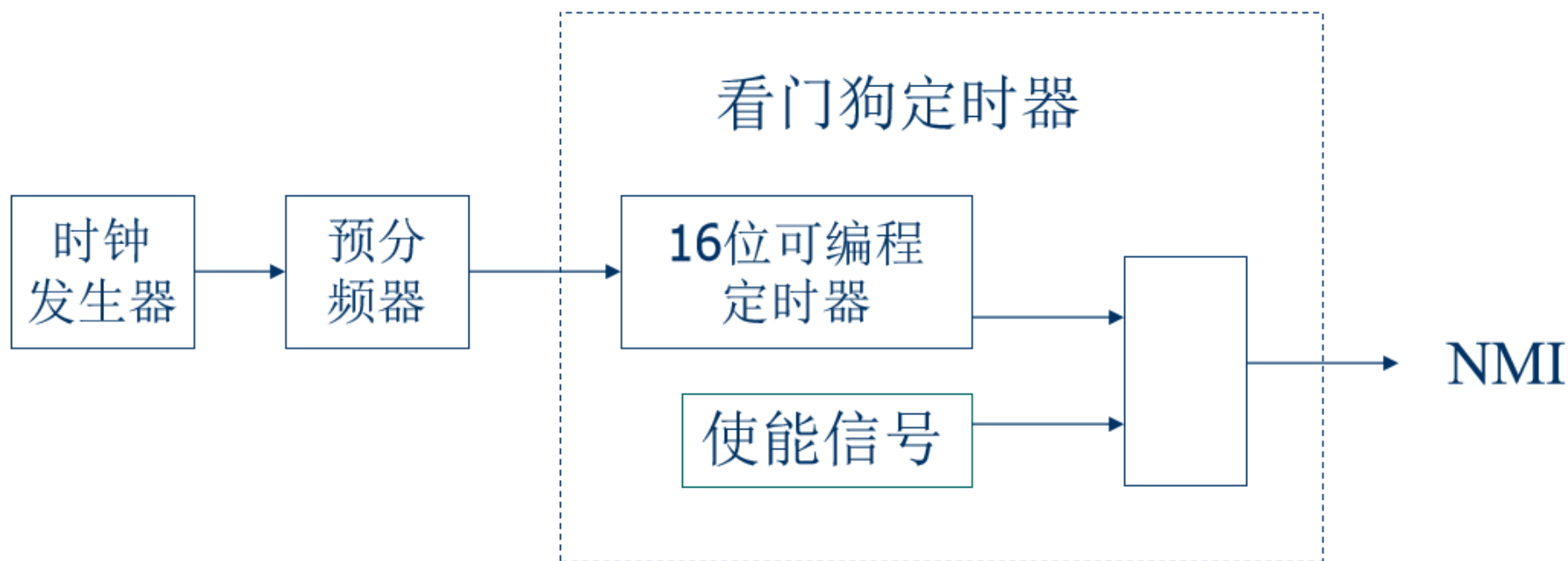
双密度指令集举例

- 对于32位ARM微处理器，配有定长32位的指令集，也配备称为Thumb指令集的16位指令集。Thumb指令集允许软件编码为更短的16位机器指令。
- 早期16位Thumb指令集称为Thumb-1指令集，其指令密度远高于32位指令集。2003年6月ARM公司推出了Thumb-2核心指令集技术。
- 三种指令集的代码尺寸与性能比较参看下图：



2.3.6 看门狗定时器

- 看门狗定时器是嵌入式系统很有特色的硬件部件，用来引导嵌入式微处理器脱离死锁工作状态。
 - 嵌入式系统的工作信号多为低电压信号，且工作于恶劣环境中，易受外部电磁干扰影响，导致系统进入死循环状态，此时看门狗定时器可以捕获和复位已失去控制的处理器。



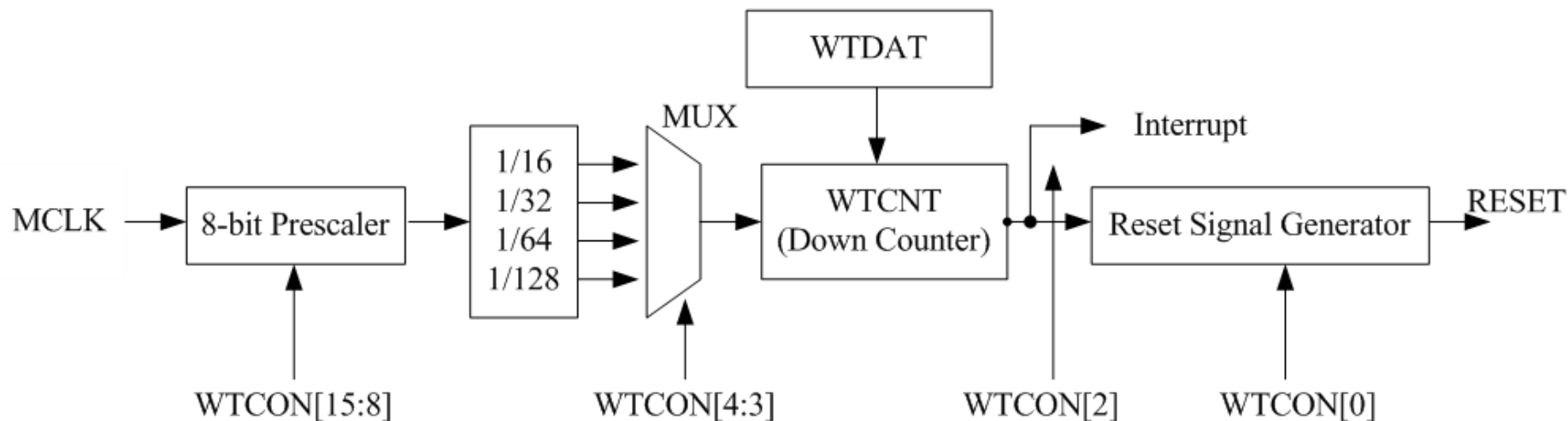
2.3.6 看门狗定时器

工作原理

- 嵌入式系统启动后，初始化程序向看门狗的计数寄存器写入计数初值。此后，每隔一个预定的时间间隔，看门狗计数寄存器执行一次计数。如果软件与设备工作正常，系统程序就会重置计数寄存器的计数初值，让其继续计数，并且一直循环下去。若软件与设备工作发生故障或机器死锁，必然导致计数器计满后不能重新被填入计数初值而产生计数溢出，此时看门狗会产生一个复位信号并重新复位系统。

2.3.6 看门狗定时器

S3C44B0X处理器的看门狗定时器



- 该看门狗是一个**16**位数的间隔定时器，用来产生中断服务请求信号。它可在**128**个存储器时钟周期之内产生复位信号。

2.3.6 看门狗定时器

- 在S3C44B0X 处理器中，预分频值和分频系数指定存放在看门狗定时控制寄存器WTCON中。
- 有效的预分频值范围为 0 到 2^8-1 。分频系数可以选为 16, 32, 64或 128。
- 使用下述公式计算看门狗定时器时钟频率和每个定时器时钟周期的持续时间：

$$t_watchdog = 1 / (MCLK / (Prescaler\ value + 1) / Division_factor)$$

2.3.6 看门狗定时器

SFR名称	地址	读写	控制寄存器 描述	位数	初值
WTCON	0x01D30000	R/W	看门狗定时 控制寄存器	16	0x8021
WTDAT	0x01D30004	R/W	看门狗定时 数据寄存器	16	0x8000
WTCNT	0x01D30008	R/W	看门狗定时 计数寄存器	16	0x8000

2.3.7 边界对准与端序

- 边界地址
 - 从0号单元开始，按照2字节、4字节或者其他2的整倍数长度划分内存的读写块，读写块的首地址就是边界地址。
- 地址对准（边界对准）访问
 - 从边界地址处进行内存访问。
 - 按照地址对准方式进行数据读写可以提高效率。嵌入式系统处理器按照数据总线的宽度进行主存数据读写。

读写方式	2字节对准	4字节对准	8字节对准	16字节对准	32字节对准
边界地址特征	$A[0]=0b0$	$A[1:0]=0b00$	$A[2:0]=0b000$	$A[3:0]=0b0000$	$A[4:0]=0b00000$

2.3.7 边界对准与端序

- 端序，字节序

- 端序是指多字节数据存放时，高字节存放在低地址单元还是高地址单元的存放顺序，也叫字节序。
- 端序有大端序和小端序之分。
 - 在大端序系统中，高字节存放在低地址处。
 - 在小端序系统中，低字节存放在低地址处。
 - PC通常使用小端序，IBM大型机用大端序，ARM和PowerPC支持两种端序中的任意一种。
- 例如：存放4字节数0x5F01003, 两种端序的存放方式如下：

32位字存储单元的最低2位地址	大端序	小端序
第11单元	00000011	00000101
第10单元	0010000	11110000
第01单元	11110000	00010000
第00单元	00000101	00000011

2.3.8 地址重映射

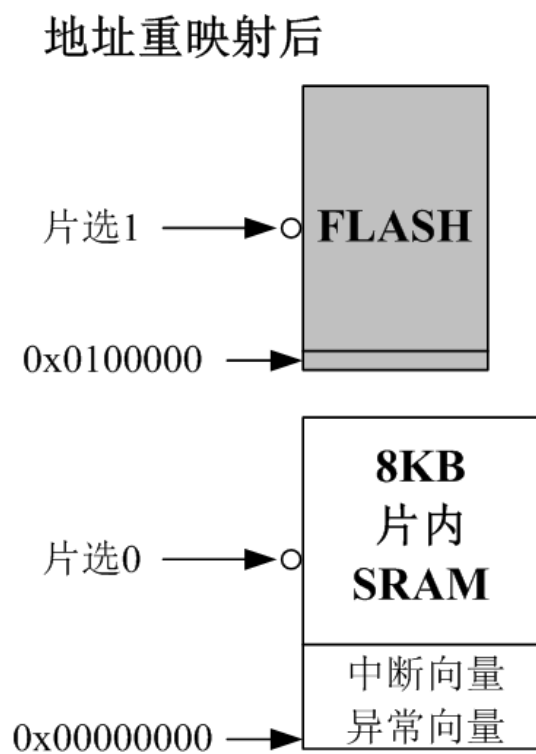
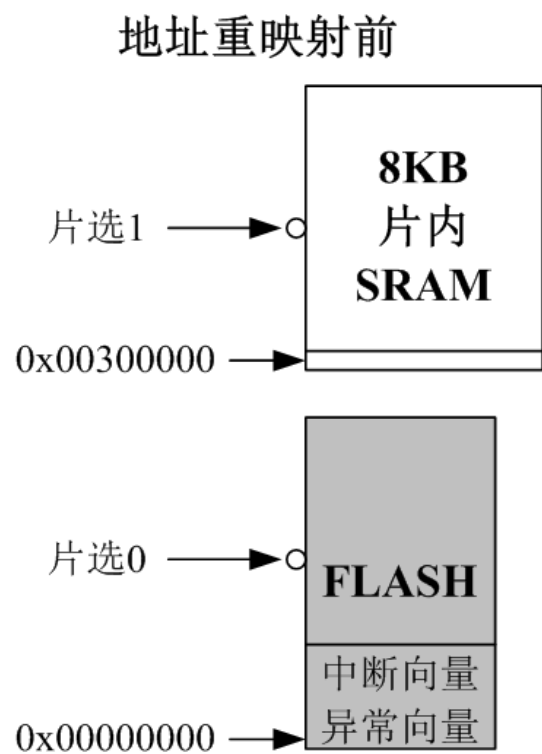
- 一般而言，计算机的物理存储单元与分配给它的地址之间具有一一对应的关系，而且固定不变。如果计算机在运行过程中改变了这种映射关系，那么这种地址改变，就叫地址重映射。
- 嵌入式系统常用到地址重映射技术，特别是在嵌入式系统的启动阶段。
 - 启动阶段的引导启动加载程序（BootLoader）都存放在Nor Flash ROM中。
 - 开始启动时，包含有中断（异常）向量表以及其它启动程序的ROM被分配在0x00000000开始的一段地址空间中，并且中断向量和异常向量表从ROM中的0号地址开始存放。

2.3.8 地址重映射

- 由于Nor 型 Flash ROM的读出速度慢，如果每次中断都要读取ROM上的中断向量会严重影响中断速度。因此在启动程序完成RAM存储器初始化、堆栈和寄存器初始化、将ROM中断向量表以及程序代码与数据复制到RAM低地址后，就需要进行地址空间的重新安排（也即**地址重映射**），把RAM地址空间迁移到0地址开始处，把Flash ROM的地址空间迁移到系统存储器的高地址处，这样就能解决中断速度的问题了。
- 地址空间的重新分配与处理器的硬件结构紧密相关，32位系统中的地址重映射机制分两种：
 - 用处理器内部专门的重映射寄存器完成地址重映射。
 - 如ATMEL 的AT91XX微处理器
 - 处理器内部没有专门的重映射寄存器时，通过改写处理器内部用于控制内存起止地址的 B a n k 寄存器来实现重映射。
 - 如三星公司的S3C4510B

2.3.8 地址重映射

AT91M55800A处理器 地址重映射前后的地址分配



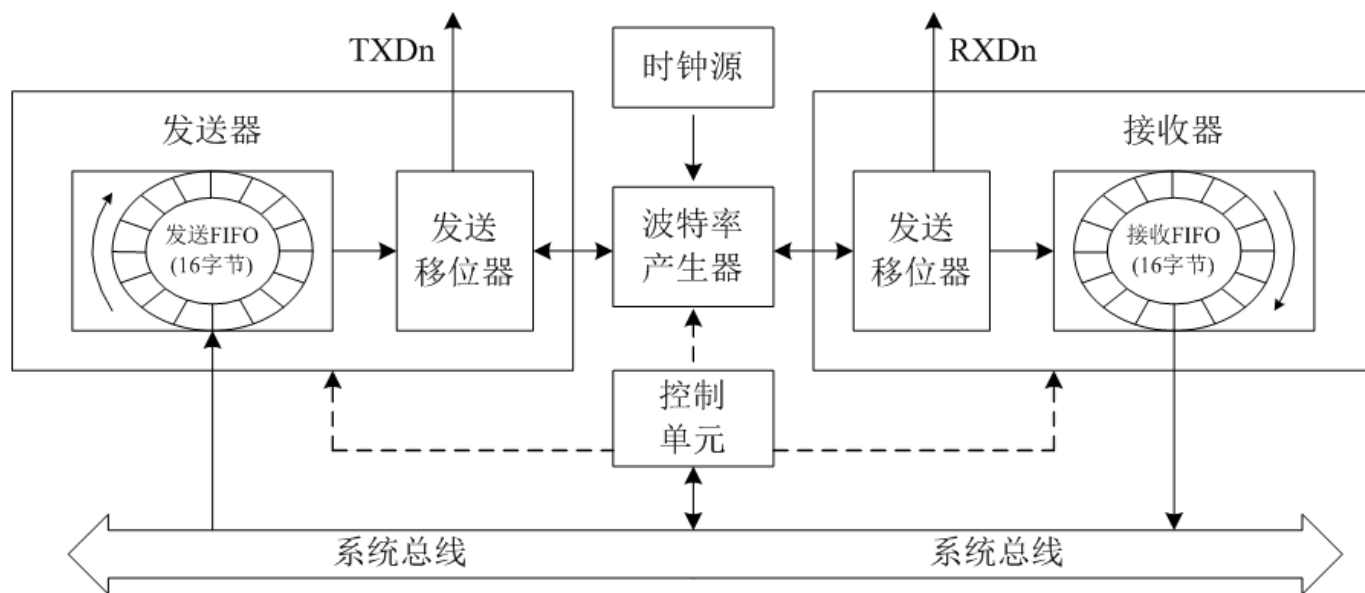
2.3.9 FIFO缓冲寄存器

- FIFO (First In First Out) 也即先入先出
- FIFO寄存器或FIFO缓冲寄存器
 - 按照FIFO方式进行数据读写的寄存器
 - 可被组织成环形结构
- FIFO缓冲寄存器常常用于嵌入式系统中
 - 提高数据传输效率
 - 两种使用形态
 - 一种是分立的专用器件
 - 集成在CPU某个控制器的专用寄存器

2.3.9 FIFO缓冲寄存器

S3C44B0X的UART方框图（含FIFO）

- S3C44B0X处理器内建有两个UART接口。每一个UART的最高传输速率是115.2K bps，并且带有两个接受和发送的FIFO缓冲器。

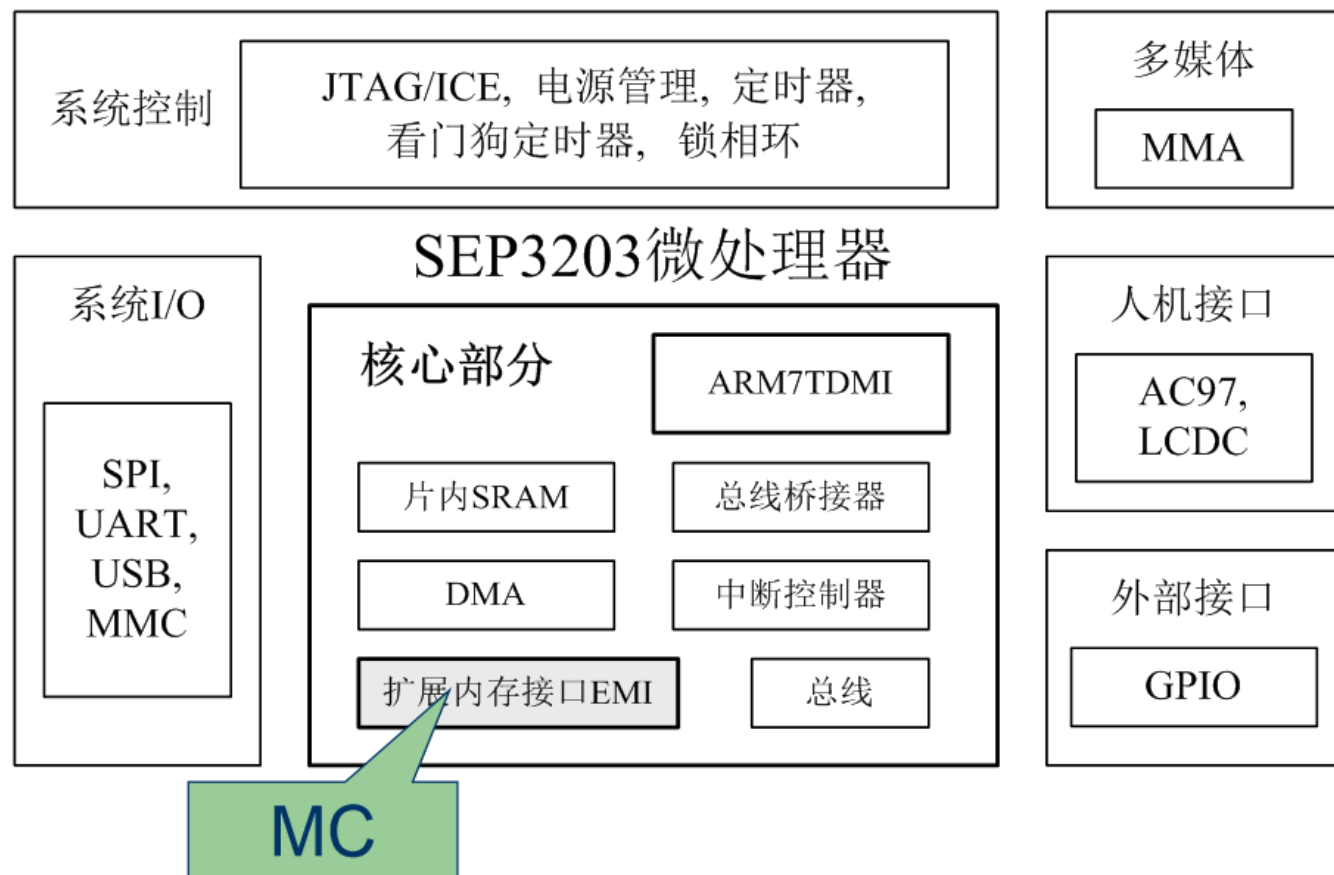


2.3.10 主存控制器

- 主存控制器MC (Memory Controller)
 - MC是CPU所在核心模块中的一个重要子模块。
 - MC功能：
 - 确定处理器所使用的存储区 (Bank) 个数；
 - 确定各存储区起始地址；
 - 确定各BANK寻址空间大小和数据线宽度；
 - 确定字节序；
 - 确定所有Bank的可编程访问周期；
 - 动态存储器的自动刷新等。

2.3.10 主存控制器

东南大学博芯公司 SEP3203微处理器方框图



2.3.10 主存控制器

SEP3203微处理器的MC

- 在SEP3203中主存控制器被称为外部存储器接口 (External Memory Interface, EMI)
 - 可提供7个存储器Bank选择信号，其中六个片选信号支持对ROM、SEAM、SDRAM和Nor Flash存储器进行片选；
 - 对每一个片选可配置16位/32位数据线宽度，可配置对应存储区的起始地址；
 - 每个片选支持的最大寻址范围是64MB；
 - 提供NAND_CS片选信号，支持从8位数据宽度的NAND FLASH直接启动系统；
 - 支持ECC校验的一位纠错；
 - 管理一个20KB 大小的片内SRAM, 实现零等待读写。

2.4 主流嵌入式微处理器

Table 4-1: Real-world architectures and processors

Architecture	Processor	Manufacturer
AMD	Au1xxx	Advanced Micro Devices, ...
ARM	ARM7, ARM9, ...	ARM, ...
C16X	C167CS, C165H, C164CI, ...	Infineon, ...
ColdFire	5282, 5272, 5307, 5407, ...	Motorola/Freescale, ...
I960	I960	Vmetro, ...
M32/R	32170, 32180, 32182, 32192, ...	Renesas/Mitsubishi, ...
M Core	MMC2113, MMC2114, ...	Motorola/Freescale
MIPS32	R3K, R4K, 5K, 16, ...	MTI4kx, IDT, MIPS Technologies, ...
NEC	Vr55xx, Vr54xx, Vr41xx	NEC Corporation, ...
PowerPC	82xx, 74xx, 8xx, 7xx, 6xx, 5xx, 4xx	IBM, Motorola/Freescale, ...
68k	680x0 (68K, 68030, 68040, 68060, ...), 683xx	Motorola/Freescale, ...
SuperH (SH)	SH3 (7702, 7707, 7708, 7709), SH4 (7750)	Hitachi, ...
SHARC	SHARC	Analog Devices, Transtech DSP, Radstone, ...
strongARM	strongARM	Intel, ...
SPARC	UltraSPARC II	Sun Microsystems, ...
TMS320C6xxx	TMS320C6xxx	Texas Instruments, ...
x86	X86 [386, 486, Pentium (II, III, IV)...]	Intel, Transmeta, National Semiconductor, Atlas, ...
TriCore	TriCore1, TriCore2, ...	Infineon, ...

几种常见嵌入式处理器:

ARM

PowerPC

MIPS

2.4 主流嵌入式微处理器

ARM简史

- 1978年3月, Hermann Hauser 与Chris Curry 创办CPU公司 (Cambridge Processing Unit) 。
- 1978年12月, Andy Hopper与CPU两位创始人 一起创立了Acorn Computer公司, 最初是销售家用电脑Atom, 到1982年与BBC合作贩卖教育用计算机BBC Micro。
- 1983年10月, 正式启动Acorn RISC Machine项目。
- 1985年4月, 第一款ARM原型由Acorn Computer Group开发、VLSI Tehnology Inc制造成功, 诞生了首款RISC处理器ARM1。
- 1986年, 推出ARM1的改良版ARM2, 集成了3万个晶体管, 工作时钟为8兆, 数据线为32比特, 采用3微米制程建造, 用于BBC Archimedes 305上。随后又推出了带缓存的ARM3, 工作时钟为25M, 用于台式机Acorn Archimedes540/1 上。
- 1990年, ARM公司 (Advanced RISC Machines Limited) 从Acorn公司脱颖而出。

2.4 主流嵌入式微处理器

- 1991年，ARM推出第一款嵌入式RISC核ARM6，苹果电脑使用ARM610 设计了其Apple Newton PDA。
- 1993年，ARM推出ARM7 核。
- 1995年，Digital Semiconductor 推出第一款StrongARM核。
- 1996年，ARM和VLSI Technology 推出ARM810核。
- 1997年，ARM发布ARM9TDMI核。
- 1998年，ARM推出可综合的ARM7TDMI核。
- 1999年，ARM推出可综合的ARM9E处理器，提高了信号处理能力。
- 2000年，ARM发布SecurCore智能卡系列产品。
- 2001年，ARM发布ARMv6 微体系结构。
- 2002年，ARM发布ARM11 微体系结构。
- 2004年，ARM发布基于ARMV7体系结构的Cortex处理器系列，发布首款ARM Cortex-M3 ，发布第一款集成多处理器MPCore。
- 2005年，ARM发布Cortex-A8处理器。

2.4 主流嵌入式微处理器

- 2007年，发布ARM Cortex-M1处理器，首款专为FPGA中实现设计的处理器。推出的ARM Cortex-A9 可实现可扩展性能和低功耗设计，推出针对智能卡应用的SecurCore 300 处理器。
- 2009年，ARM宣布实现2Ghz频率的Cortex-A9 双核处理器，推出体积最小、功耗最低和能效最高的处理器Cortex-M0。
- 2010年，发布基于ARMV7-A Cortex 微架构的Cortex A15 MPCore处理器，单芯片内拥有1~4个SMP处理核心。
- 2011年，ARMV8架构诞生，先后推出的Cortex-53/57/72/A35/32/73/55 采用该架构，这是ARM公司首个支持64 位字长的处理器架构。
- 2013年，ARMV8-R架构发布，面向汽车及工业控制应用。
- 2015年，在ARMv6-M基础上衍生了ARMv8-M baseline，在ARMv7-M基础上衍生了ARMV8-M mainline，Cortex-M23采用的是ARMV8-M baseline 架构，Cortex-M33采用的是ARMv8-M mainline架构，两款处理器都加入了TrustZone支持，面向IoT物联网。
- 2016年9月ARM被软银收购，并发布基于ARMV8-R架构的Cortex-R52处理器。
- 2021年3月31日， 发布ARMv9架构，面向PC、高性能计算、深度学习等应用。