

计算机组成与嵌入式系统

第九章 DSP处理器技术

钟胜 颜露新

人工智能与自动化学院 飞行器导航制导系

2022年 春季

本章目标

- * 掌握DSP基本概念
- * 熟悉C641X DSP 体系架构
- * 掌握C641X DSP 存储器接口技术

9.1 基本概念

* DSP

Digital Signal Processing

数字信号处理

Digital Signal Processor

数字信号处理器 本课程重点

9.1 DSP芯片的发展历程

信息处理技术诞生并迅速的发展

AMI公司发布DSP芯片S2811，但没有现代DSP芯片的硬件乘法器

日本NEC公司推出具有硬件乘法器的商用DSP芯片MPD7720

第三代DSP芯片问世，运算速度进一步提高，应用范围扩大

1960s

1970s

1978

1979

1989

1982

1980s后期

1990s

DSP芯片的理论和算法基础已成熟,但其应用领域受限

美国Intel公司发布商用可编程器件2920，依然没有硬件乘法器

第一代DSP芯片TMS32010及其系列产品诞生

现了第四代和第五代DSP芯片

9.1 DSP芯片的分类

* 按数据格式分类

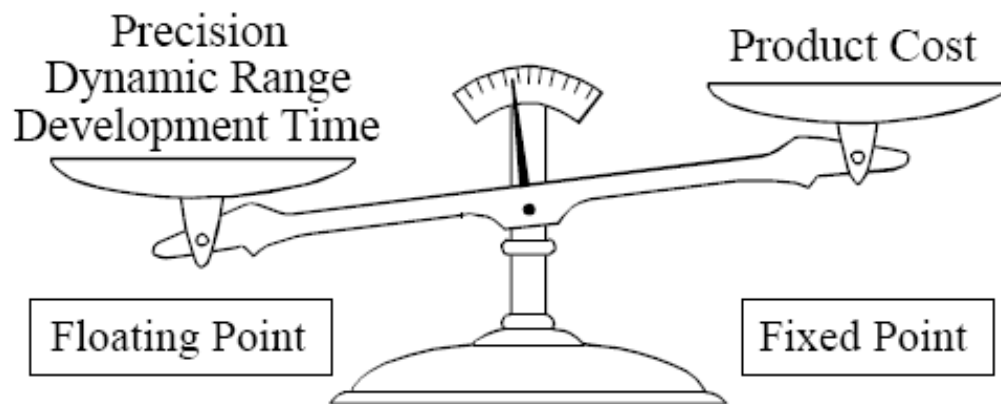
定点DSP芯片

浮点DSP芯片

* 按芯片的用途分类

通用型DSP

专用型DSP



- 浮点DSP有浮点运算硬件，也可进行定点运算
- 定点DSP没有浮点运算硬件，浮点运算需要开发环境支持

9.1 DSP芯片的性能指标

1. **指令周期**：即执行一条指令所需要的时间，通常以ns为单位。如TMS320LC549-80在主频为80MHz时的指令周期为13.5ns
2. **MIPS** (Million Instructions Per Second) :百万条指令/每秒。如TMS320C6416在时钟为1GHz时的峰值性能可达8000MIPS
3. **MOPS** (Million Operations Per Second : 百万次操作/每秒。如TMS320C6201在时钟为200MHz时的峰值性能可达2400MOPS
4. **MFLOPS** (Million Float Operations Per Second : 百万次浮点操作/每秒。如ADSP-TS201S的峰值性能可达19.3GFLOPS

9.1 DSP芯片的性能指标

5. **Mbps**(Million Bits Per Second): 百万位/每秒

如TMS320C6000的总线时钟为200MHz时, 其总线数据吞吐率为800M字节/秒

6. **MAC执行时间**: 完成一次乘-累加运算所需时间

大部分DSP可在单周期内完成一次MAC

7. **FFT执行时间**: 完成一个N点FFT运算所需时间

.....

9.1 主要DSP厂商

- * TI公司
- * Motorola公司
- * ADI公司
- * Hitachi、NEC、Zilog和 STMicroelectroncs等

9.1 国产DSP—仿制之路

- * 上海交大“汉芯”丑闻

2003年2月 推出“汉芯一号”

2006年2月 “汉芯”造假暴露 打磨MOTO

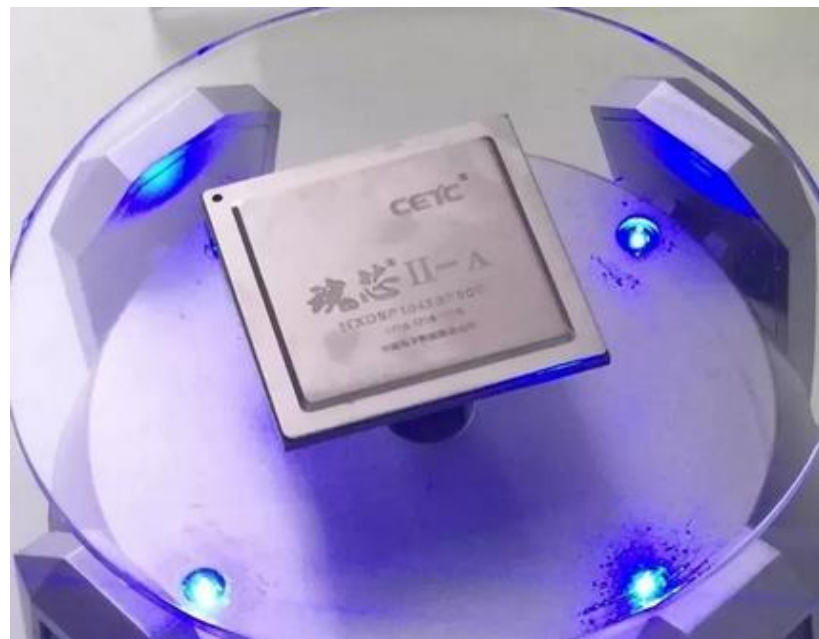
- * 国防科技大学仿制出TI公司C6000系列DSP “飞腾”系列

- * 中科院电子院仿制TI公司TMS320C6455

9.1 国产DSP—自主创新之路



中电14所自主研制的华睿DSP



中电38所自主研制的魂芯DSP

9.1 TI公司主推的三大DSP平台

TMS320C2000™



最佳测控

- 电机控制
- 家用电器
- 变频电源控制
- 工业自动化

TMS320C5000™



低功耗

- 手机
- 电信
- 数码相机
- 便携式媒体播放器
- 医疗器械

TMS320C6000™



高性能

- 无线基站
- 视频流、视频会议
- 视频安防/监控制
- 医疗成像

9.1 C6000系列DSP子系列

- C62x子系列：32位定点DSP、1200~2400MIPS
代表器件： TMS320C6205
代表器件： TMS320C6211
- C64x子系列：32位定点DSP，4000~5760MIPS
代表器件： TMS320C6414、C6415、C6416
数字媒体器件： TMS320DM640、DM641、DM642
- C67x子系列：32位浮点DSP、1200~1800MIPS，
900~1350MFLOPS
代表器件： TMS320C6701
代表器件： TMS320C6711、C6712、C6713
- C674X子系列：低功耗浮点/定点混合
3648MIPS，2746MFLOPS
- C6000多核DSP C647X
- KeyStone 多核DSP C665X C667X

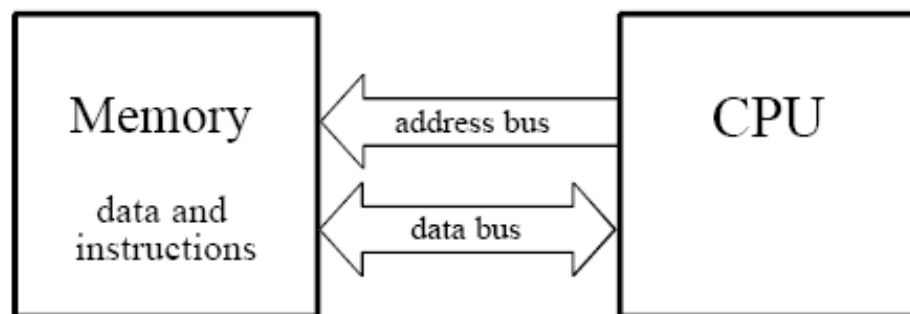
9.1 ADI公司的DSP

- * 21xx系列，主要以218x,219x系列为代表，内部RAM多，接口丰富，适合用于控制
- * SHARC系列，32位浮点，包括前期的2106x，2116x和目前的2126x,2136x
- * TigerSHARC系列，从SHARC发展而来，目前有TS-101s,TS-201,202,203等产品
- * BlackFin系列，16位高性能定点运算和通用微控制器使用方便的性能结合

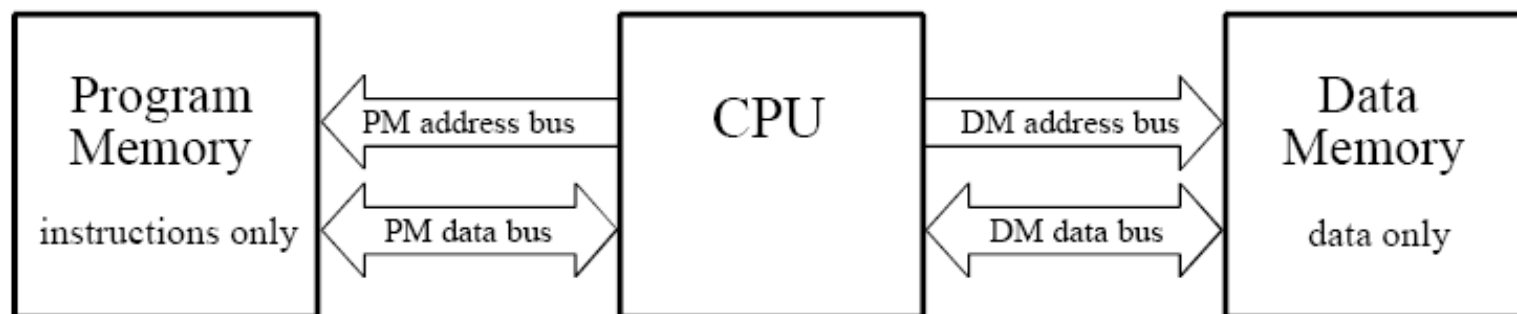
9.1 DSP芯片的特点

(1) 采用超级哈佛结构

a. Von Neumann Architecture (*single memory*)



b. Harvard Architecture (*dual memory*)



9.1 DSP芯片的特点

c. Super Harvard Architecture (*dual memory, instruction cache, I/O controller*)

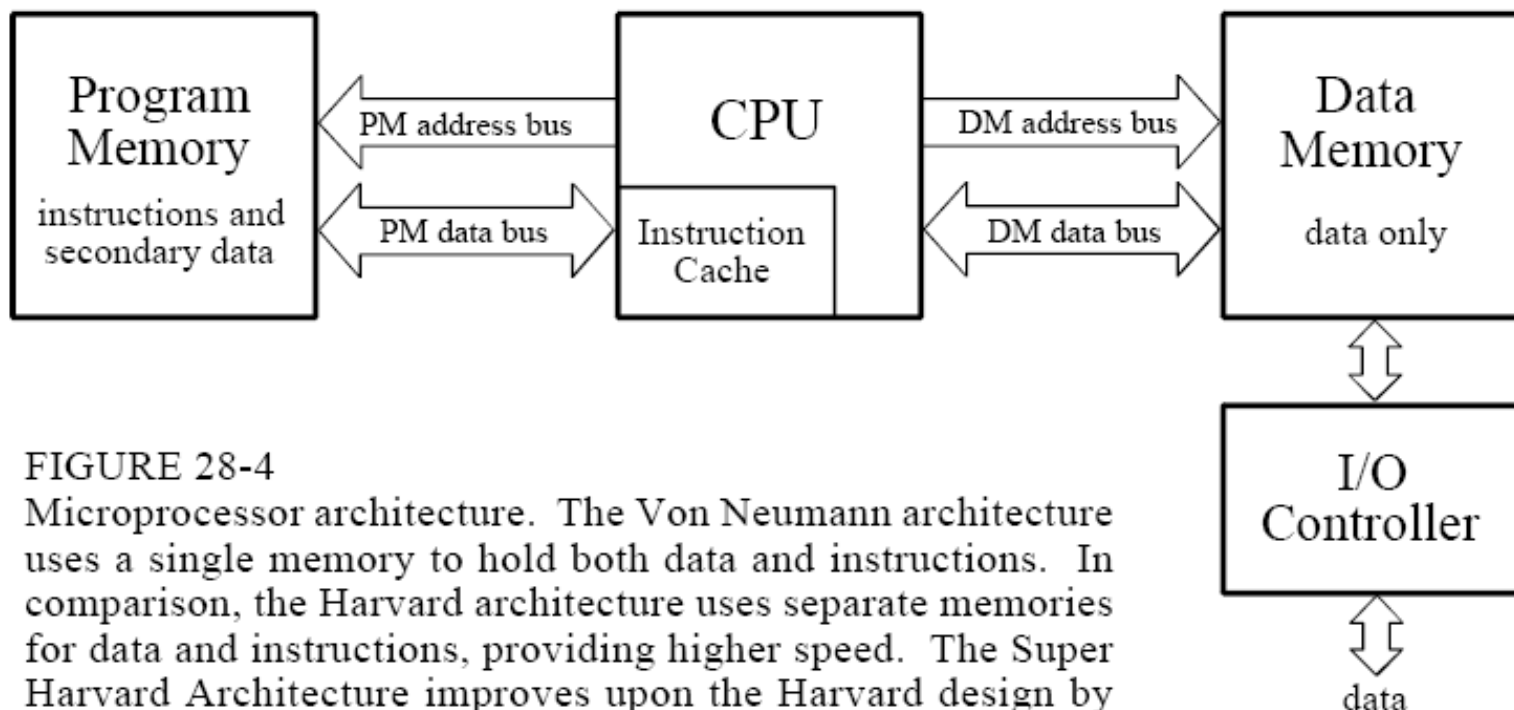


FIGURE 28-4

Microprocessor architecture. The Von Neumann architecture uses a single memory to hold both data and instructions. In comparison, the Harvard architecture uses separate memories for data and instructions, providing higher speed. The Super Harvard Architecture improves upon the Harvard design by adding an instruction cache and a dedicated I/O controller.

9.1 DSP芯片的特点

(2) 采用多总线结构

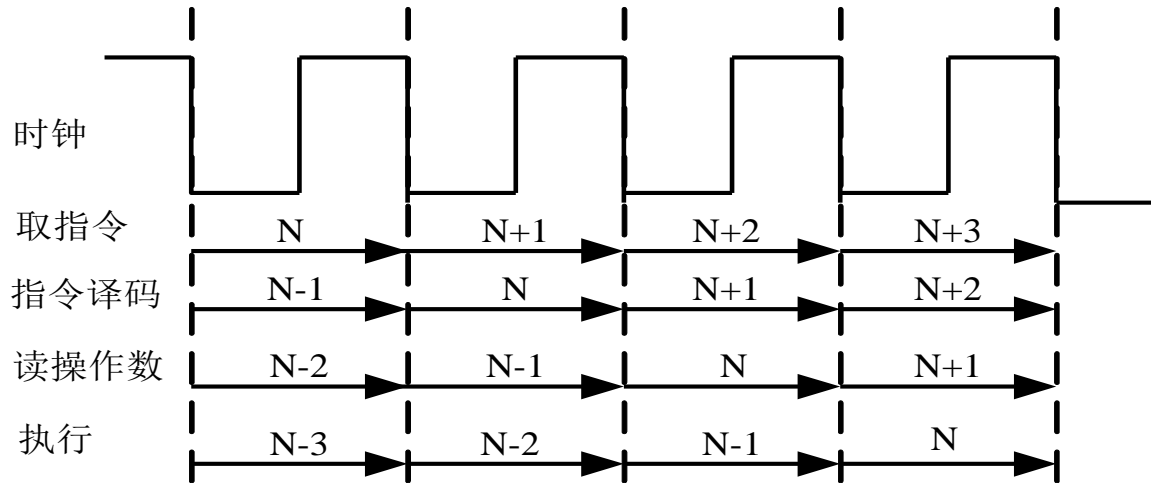
DSP 芯片都采用多总线结构，可同时进行取指令和多个数据存取操作，并由辅助寄存器自动增减地址进行寻址，使 CPU 在一个机器周期内可多次对程序空间和数据空间进行访问，大大提高了 DSP 的运行速度。如：TMS320C54x 系列内部有 P、C、D、E 等 4 组总线，每组总线中都有地址总线 and 数据总线，这样在一个机器周期内可以完成如下操作：

- (1) 从程序存储器中取一条指令
- (2) 从数据存储器中读两个操作数
- (3) 向数据存储器写一个操作数

9.1 DSP芯片的特点

(3) 采用流水线技术

每条指令可通过片内多功能单元完成取指、译码、取操作数和执行等多个步骤，实现多条指令的并行执行，从而在不提高系统时钟频率的条件下减少每条指令的执行时间。以四级流水线例子

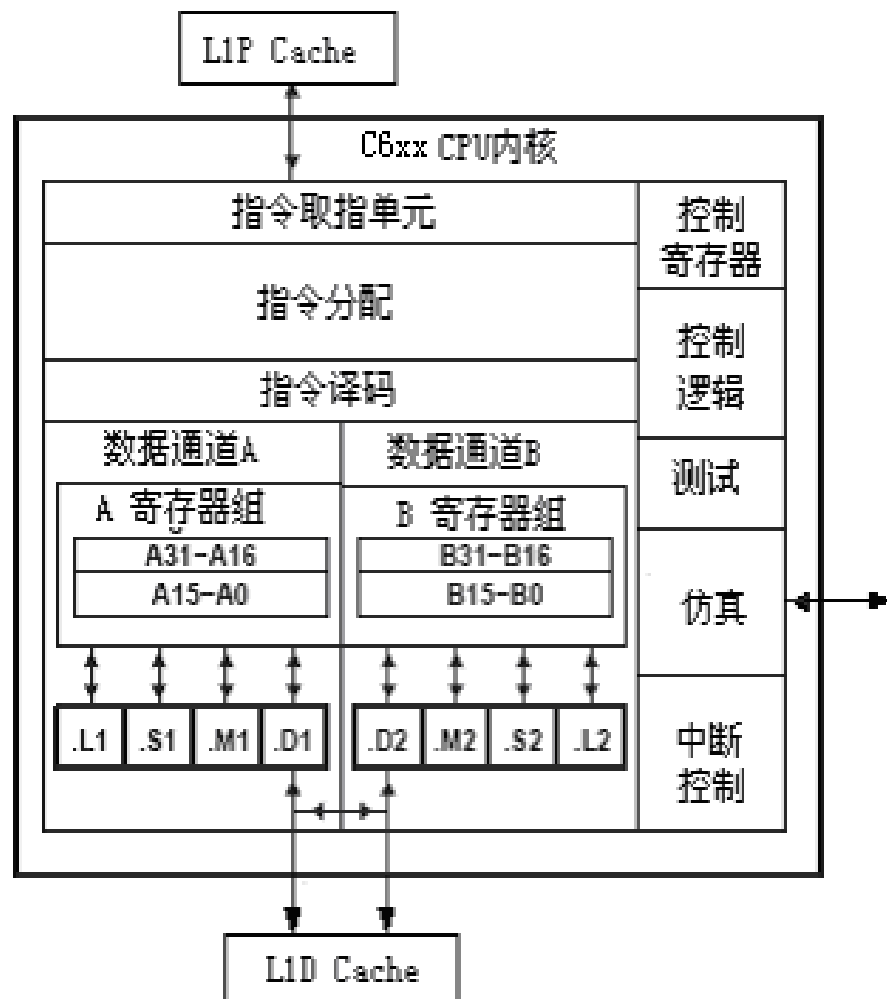


四级流水线操作过程

9.1 DSP芯片的特点

(4) 集成功能强大的CPU

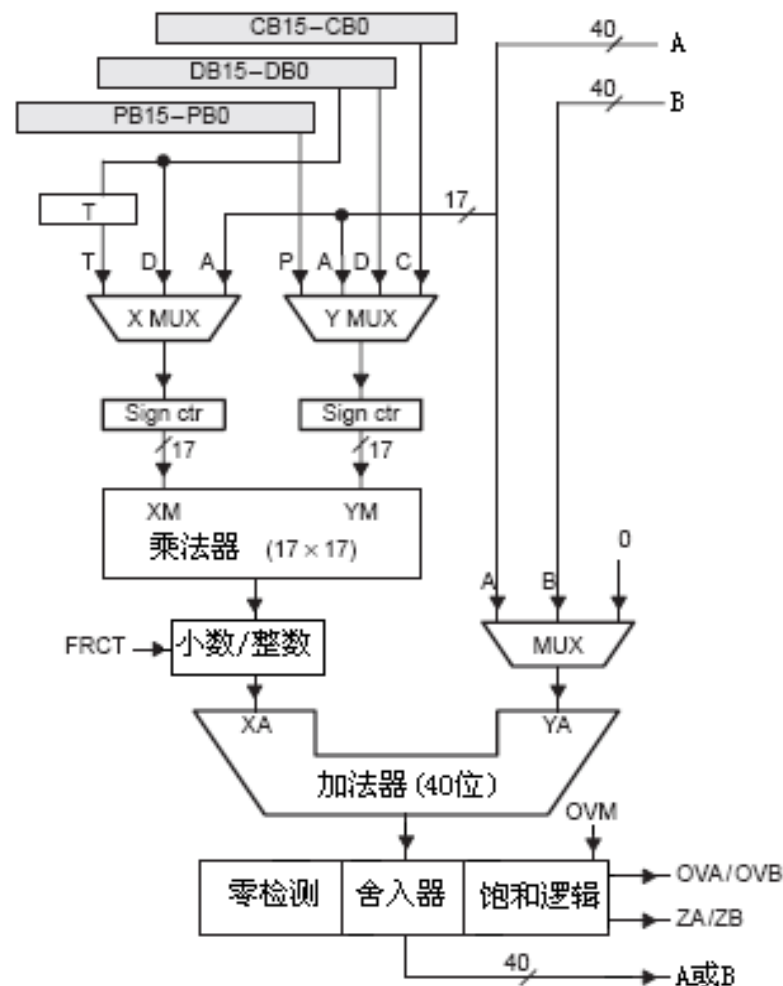
- * DSP的CPU一般包括：算术逻辑运算单元ALU、累加器、硬件乘法器、乘加单元、加法器、桶型移位器、程序地址产生和数据地址产生等部分。
- * 例如：TMS320C6000采用双数据通道和8个功能单元的结构。CPU在单周期内可同时执行8条指令，具有超长指令字(VLIW)操作能力。



9.1 DSP芯片的特点

(5) 集成专用的硬件乘法器

- 通用的微处理器中算法指令需要多个指令周期，如 MCS-51 的乘法需要4个周期
- DSP芯片的特征就是有一个专用的硬件乘法器，乘法可以在一个指令周期内完成，还可以与加法并行进行，完成一个乘法和加法只需一个指令周期。
- TMS320C3x系列DSP芯片中，有一个硬件乘法器
TMS320C6000系列中则有两个硬件乘法器



9.1 DSP芯片的特点

(6) 特殊的专用指令

- 许多DSP芯片具有指令重复循环的专用硬件及重复操作指令，能够自动重复执行单条或一段指令，进入重复机制的指令会自动变为单周期指令,大大减少了执行时间
- 在一条单字单周期指令中分别完成多个操作任务的复合操作指令。由于DSP具有特殊的多总线哈佛结构和CPU硬件，综合了CISC和RISC的优点，大多数指令是复合指令
- TMS320C3x用于卷积和付氏变换的位翻转指令和循环寻址指令，多片DSP间通信的互锁指令等

* 以TMS320C24x的MPYA (乘且累加前次乘积)指令为例

MPYA $\ast+$, AR3; 设AR为AR1, 执行后会发生下列事件:

(1) $(PC) + 1 \rightarrow PC$

(2) $(ACC) + \text{移位后的}(PREG) \rightarrow ACC$

(3) $(TREG) \times (\text{数据存储器}) \rightarrow PREG$

(4) $(AR1) + 1 \rightarrow AR1$

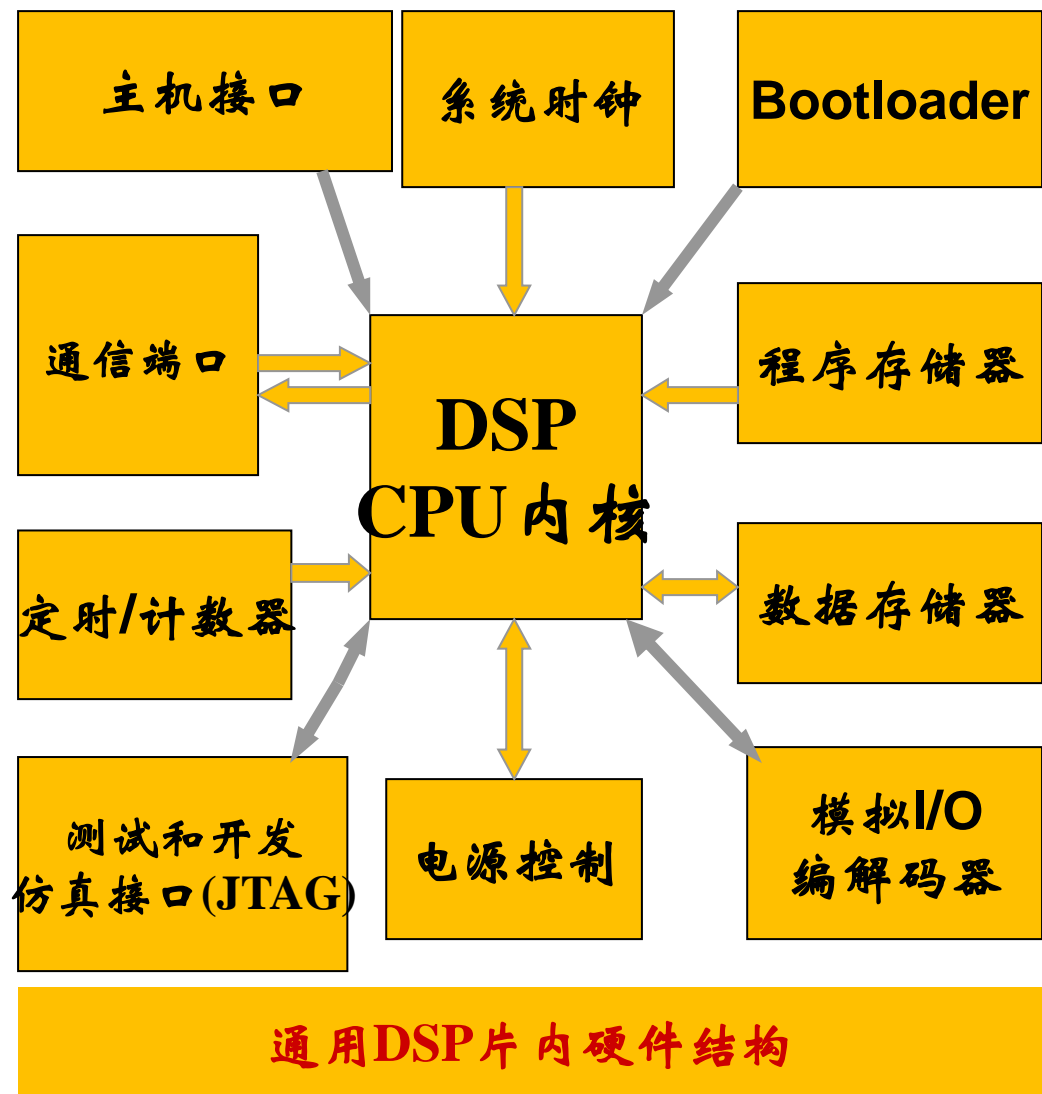
(5) 令AR3为当前辅助寄存器AR

(6) $ARP=011b$,
 $ARB=001b$

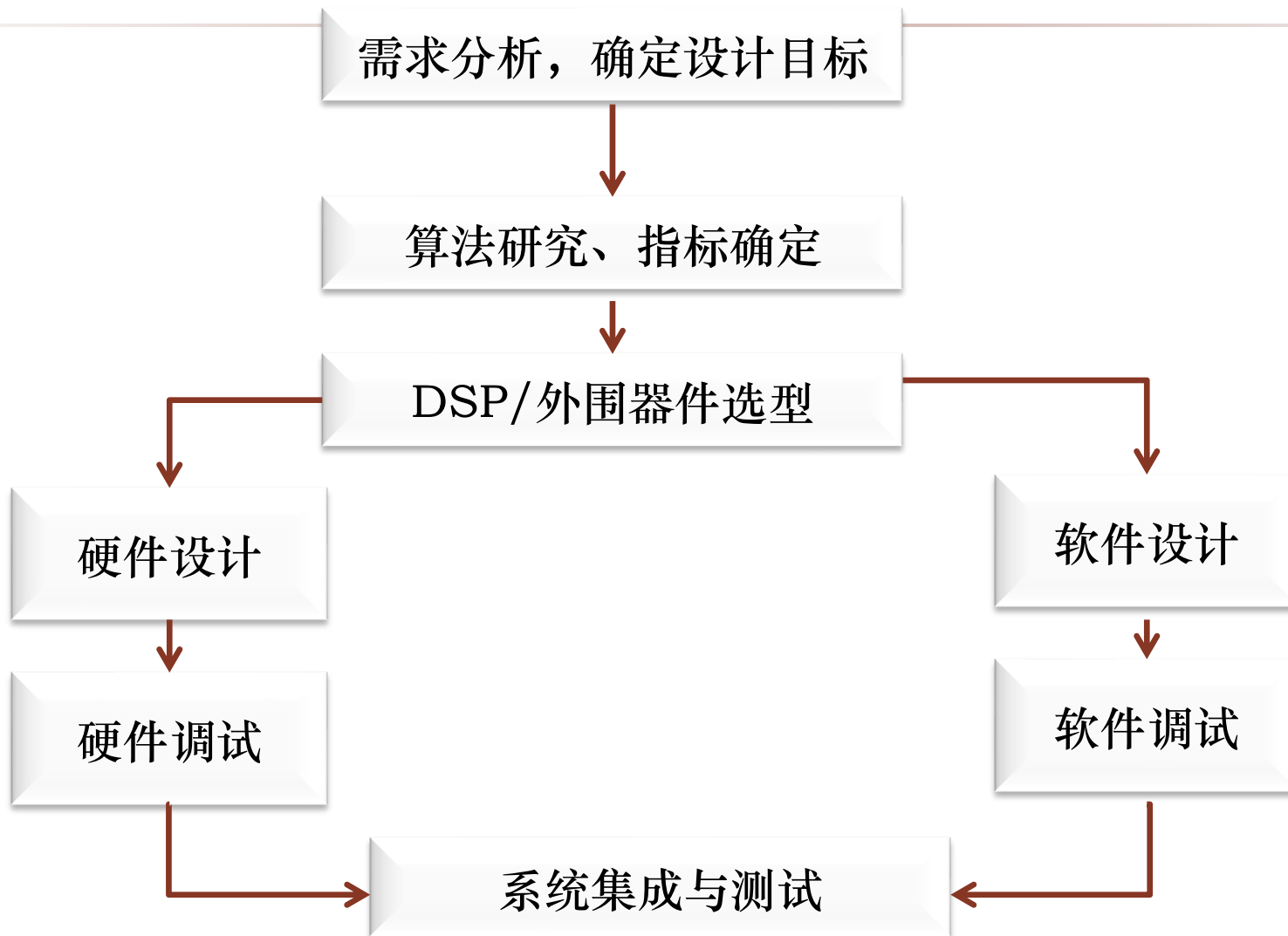
9.1 DSP芯片的特点

(7) 嵌入强大硬件功能

- * DSP片内配置有大量片内外设，不用外扩很多器件，既可组成独立的应用系统。
- * DSP芯片具有强大的扩展接口能力，可有效连接一系列外扩器件。
- * DSP=数字信号处理能力+嵌入式功能



9.2 DSP系统的设计过程



9.2 图像处理DSP选型原则

- * 定点VS浮点

图像处理一般选用定点

动态范围/精度高 选用浮点
(如 密集的FFT运算)

- * 内存大小

片上内存 访问速度快
偏向大内存

- * 外设接口

- * 功耗

- * 成本

9.2 C6000系列DSP开发工具

- * JTAG仿真器（并口，USB口）

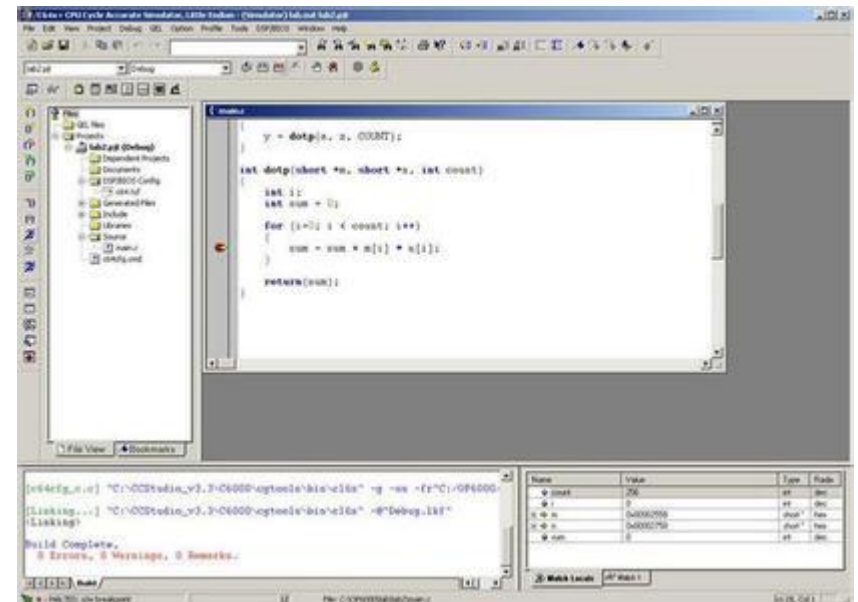
主机与DSP之间的调试通道
DSP调试开发的必选



- * 集成开发环境CCS

Code Composer Studio

TI DSP 开发必需的IDE软件

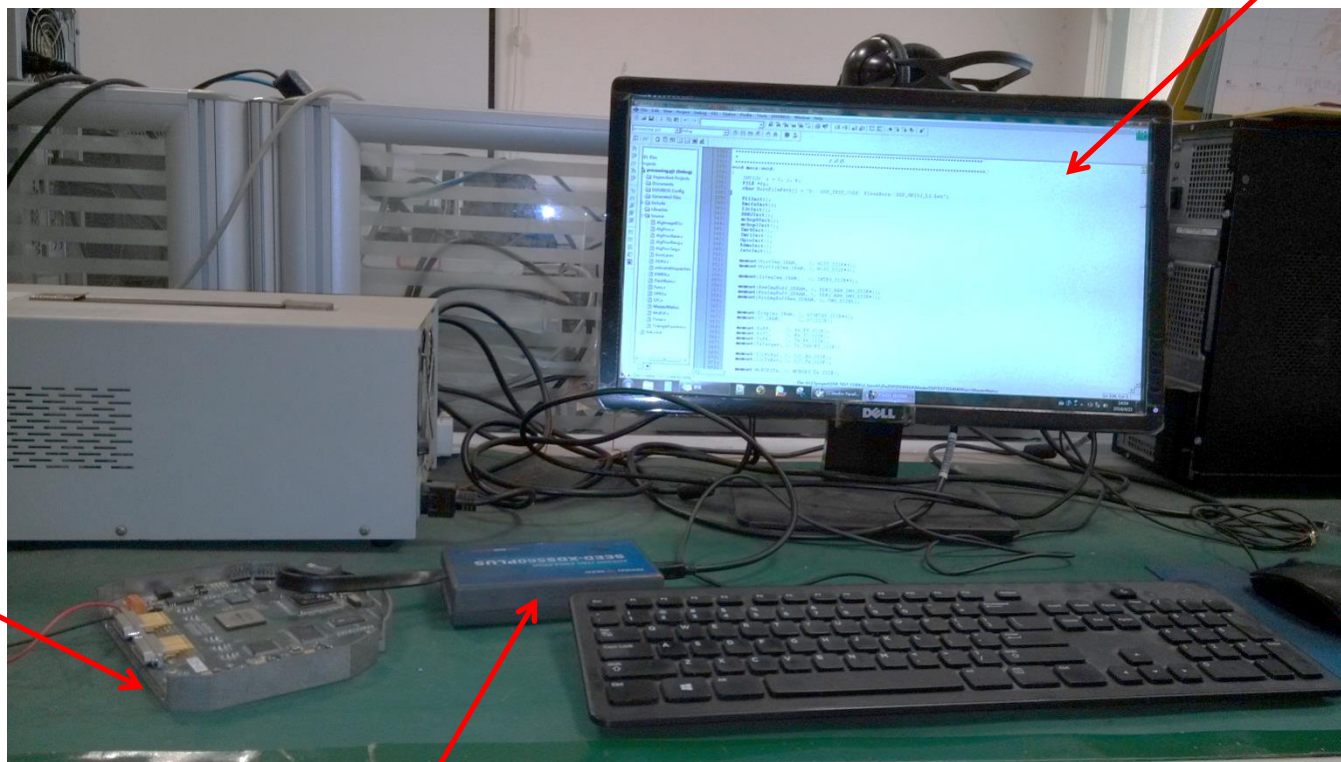


- * 主机（PC机）

9.2 C6000系列DSP开发方式

* 主机（CCS）+JTAG仿真器+目标硬件板

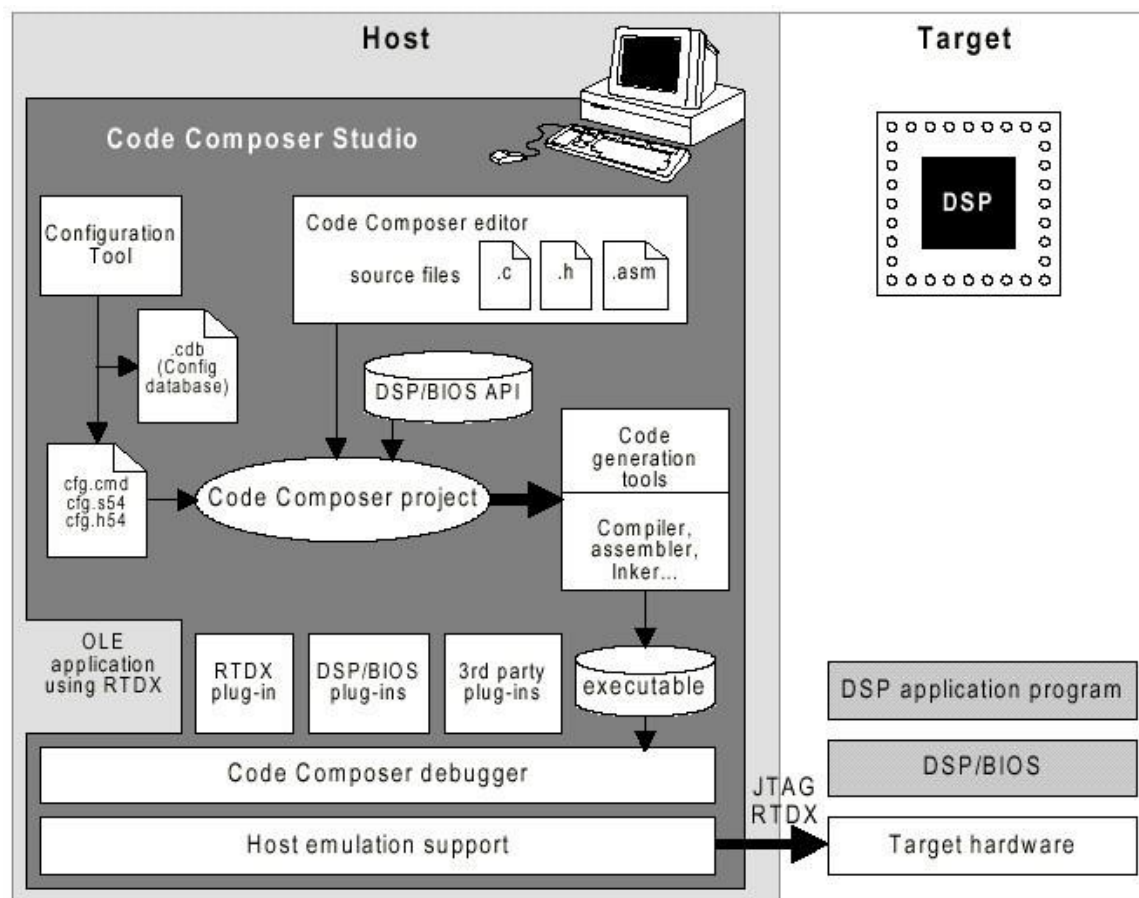
主机（带CCS软件）



JTAG仿真器

9.2 集成开发环境CCS

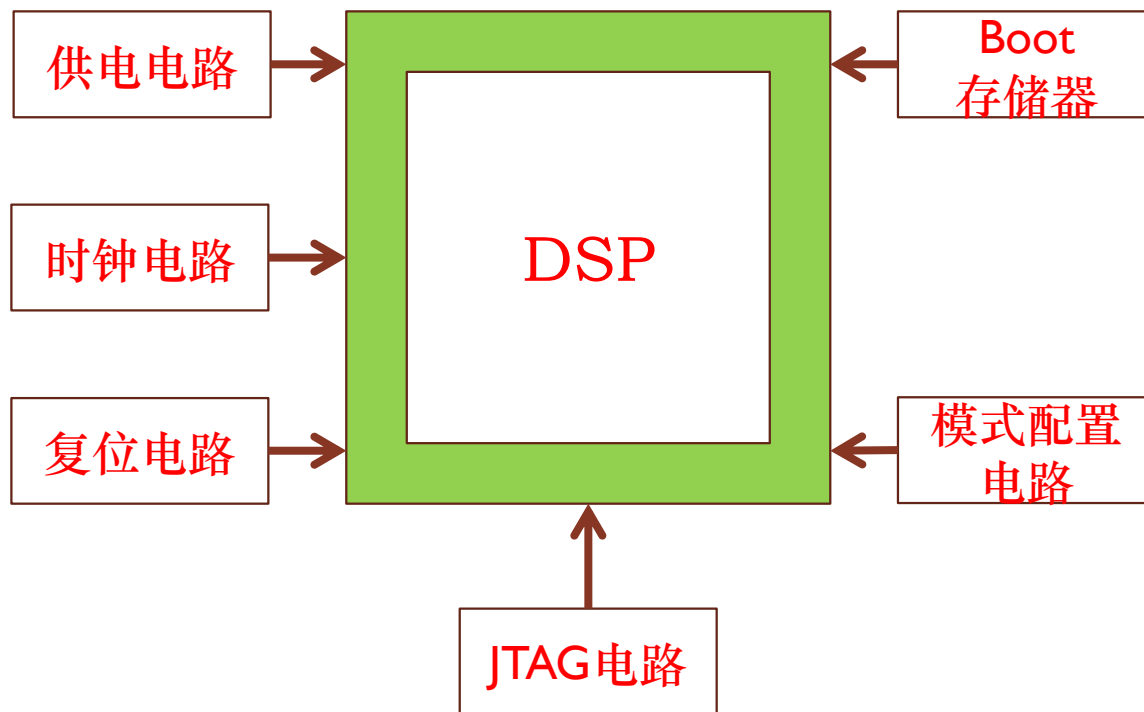
- * 源程序编辑器
- * 编译器
- * 链接器
- * 调试与数据分析
- * DSP/BIOS和API
- * RTDX, 主机接口和API
- * 高效处理库



CCS构成及接口

9.2 DSP硬件最小系统

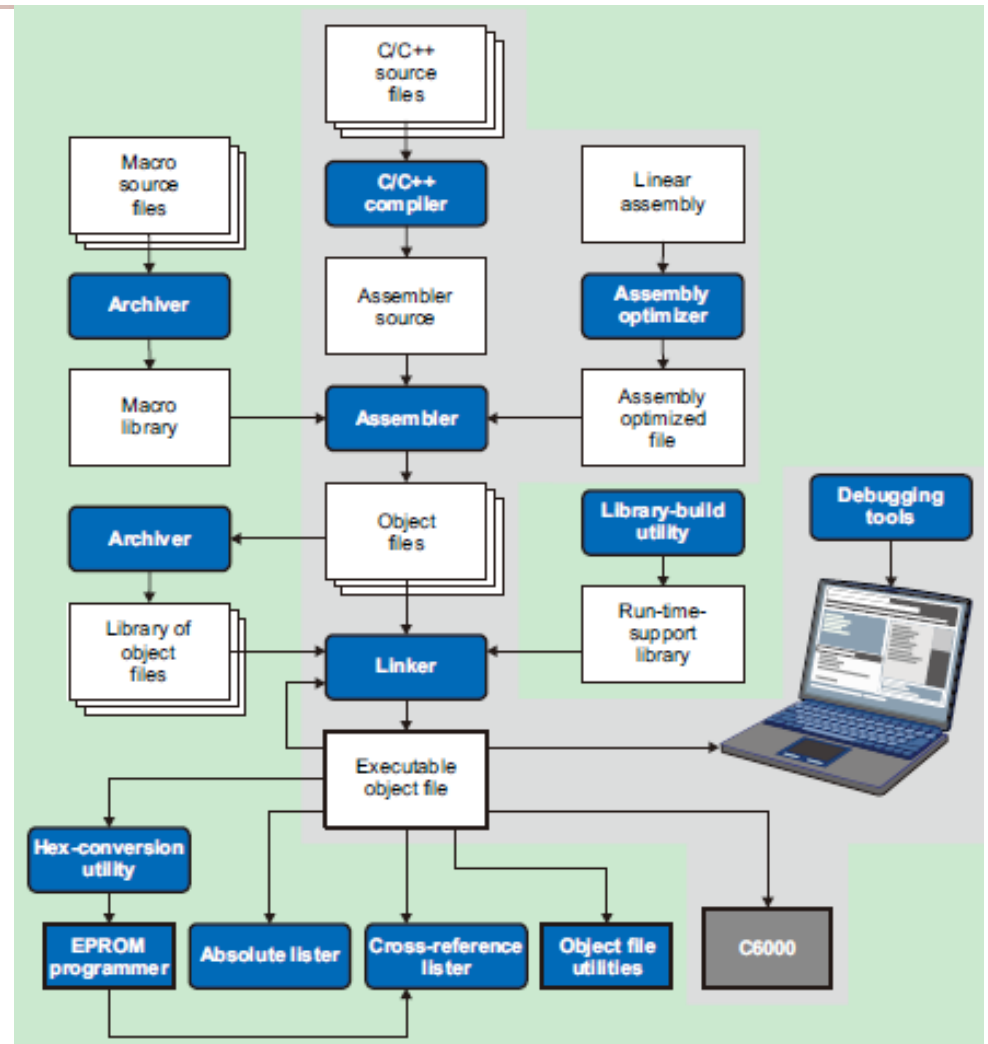
- * 供电电路
- * 时钟电路
- * 复位电路
- * 外围配置电路
- * 启动引导存储器
- * JTAG电路



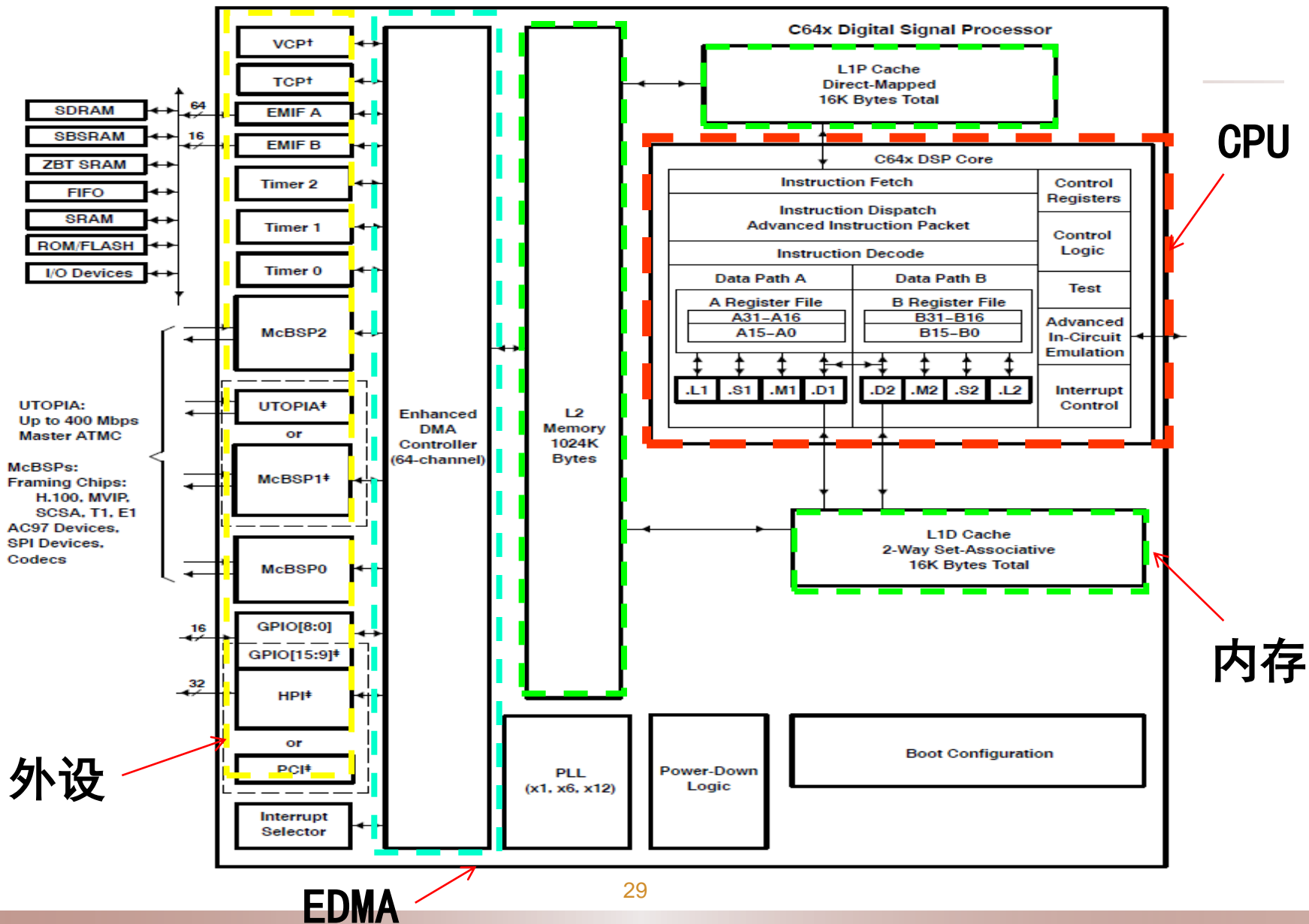
最小系统正常工作是DSP硬件系统正常工作的基础

9.2 DSP软件开发流程

- * 编程语言
 - 常用C语言
 - 汇编语言
 - 线性汇编（混合编程）
- * 算法仿真
 - PC机/MATLAB等
- * 软件仿真（CCS）
- * JTAG调试
 - 下载到DSP内部调试
- * 软件固化
 - 固化到程序存储器



9.3 TMS320C64x体系结构



9.3 C6000系列DSP的功能组成

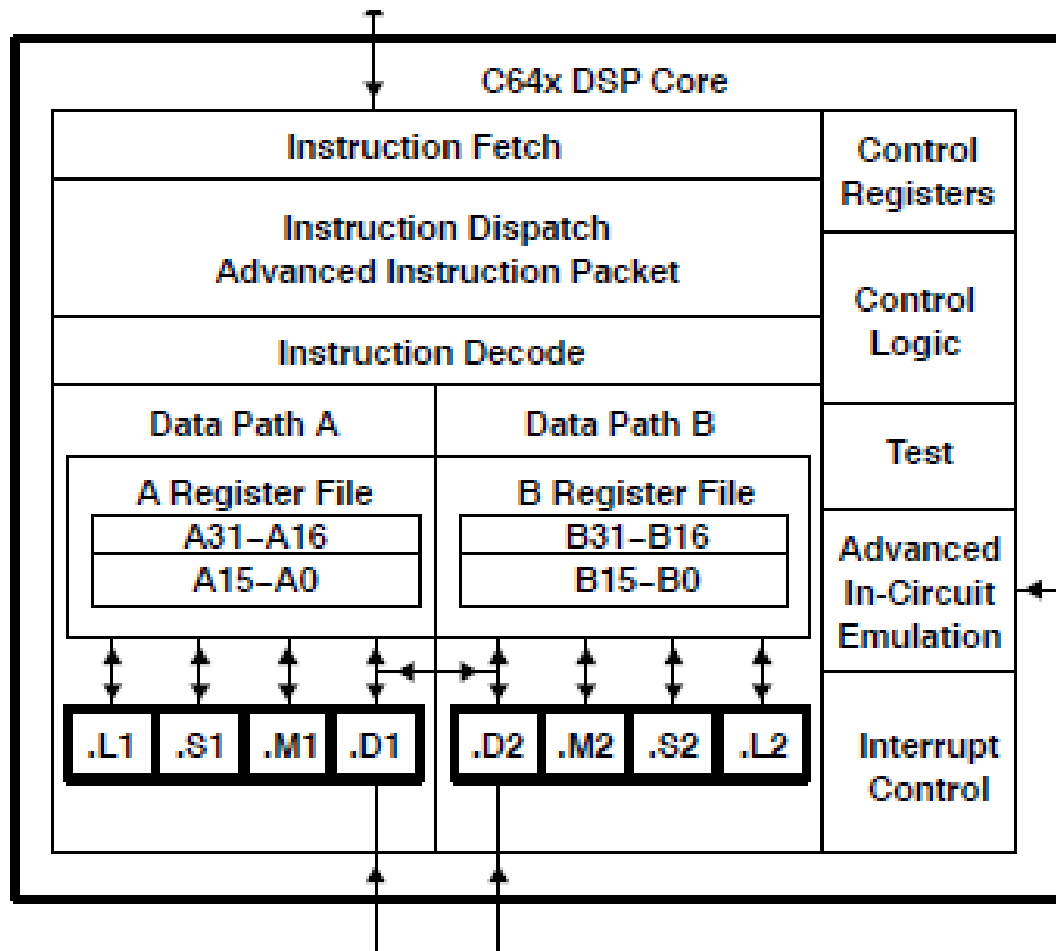
- * CPU或者DSP内核
 - 中断控制逻辑
 - JTAG仿真测试电路
- * 片上存储器
 - 程序存储器 数据存储器
- * EDMA控制器
- * 片上集成外设
- * 时钟管理电路
- * 启动配置电路

9.3 C64x的内核

- * 程序预取单元
- * 指令分包单元
- * 指令译码单元
- * 双数据通路

每通路包含四个功能单元

- * 64个32位寄存器
A/B两组，各32
- * 控制寄存器
- * 控制逻辑
- * JTAG仿真测试单元
- * 中断控制逻辑



9.3 内核功能单元

- * .L 和.S完成算术和逻辑运算(ALU)
 - * .M完成乘法运算
 - * .D完成数据寻址操作
- 寄存器组与内存之间的数据交换

9.3 控制寄存器组

* 寻址模式寄存器	AMR
* 控制状态寄存器	CSR
* 中断标志寄存器	IFR
* 中断设置寄存器	ISR
* 中断清除寄存器	ICR
* 中断使能寄存器	IER
* 中断服务表指针	ISTP
* 中断返回指针	IRP
* 不可屏蔽中断返回指针	NRP
* E1节拍程序计数器	PCE1

9.3 C64X的中断

- * 支持16种不同优先级的中断
 - INT_00 RESET中断 最高优先级
 - INT_15 最低优先级
- * INT_00~INT_03四个为不可屏蔽中断
- * INT4~INT15可编程对应多个中断源
 - INT4~INT7用户使用的中断
- * NMI为用户可使用的不可屏蔽中断，一般不用

9.3 C64X的缺省中断源

CPU Interrupt	Related INTSEL field	INTSEL Reset Value	Interrupt Acronym	Interrupt Description
INT4	INTSEL4	00100b	EXT_INT4	External interrupt pin 4
INT5	INTSEL5	00101b	EXT_INT5	External interrupt pin 5
INT6	INTSEL6	00110b	EXT_INT6	External interrupt pin 6
INT7	INTSEL7	00111b	EXT_INT7	External interrupt pin 7
INT8	INTSEL8	01000b	DMA_INT0/ EDMA_INT	DMA Channel 0 Interrupt/ EDMA interrupt
INT9	INTSEL9	01001b	DMA_INT1	DMA Channel 1 interrupt†‡
INT10	INTSEL10	00011b	SD_INT	EMIF SDRAM timer interrupt (C62x/C67x)
			SD_INTA	EMIFA SDRAM timer interrupt (C64x)
INT11	INTSEL11	01010b	DMA_INT2	DMA Channel 2 interrupt†‡
INT12	INTSEL12	01011b	DMA_INT3	DMA Channel 3 interrupt†‡
INT13	INTSEL13	00000b	DSPINT	Host port to DSP interrupt
INT14	INTSEL14	00001b	TINT0	Timer 0 interrupt
INT15	INTSEL15	00010b	TINT1	Timer 1 interrupt

9.3 中断服务表IST

- * 512Byte的内存块，16个连续的ISFP组成
- * ISFP 中断服务取指包
每个32Byte，包含8条指令
- * 若中断服务程序超过8条指令，设置跳转指令跳转到其它部分；不足8条指令，使用NOP填充
- * 可将IST搬移到任意256Byte边界对齐的存储单元，基地址由ISTB给定
INT4~INT7用户使用的外部中断
- * 复位中断INT_00 必须存放在地址0处
- * 中断服务表指针 ISTP 定位中断服务程序入口

9.3 中断响应过程

* 接收中断请求

有中断请求时，DSP将IFR对应位置1

* 中断确认

软件中断和非屏蔽中断，无条件进入相应ISP

可屏蔽中断，需要满足：

当前优先级为最高级

IER中，对应位为1

CSR中GIE位为0，表示允许可屏蔽中断； 中断相应后，不再响应其它中断
直至当前中断返回

IER中NMIE位为1

* 执行中断服务程序

保护现场，将当前程序指针PC值压入栈顶；

载入中断向量表，将其地址送入PC指针；

执行中断向量表，程序将进入ISR入口；

执行ISP，直至返回；

恢复现场，将栈顶值弹回到PC指针；

继续执行主程序

9.3 C64x内存结构

* 两级存储结构

- * L1+L2

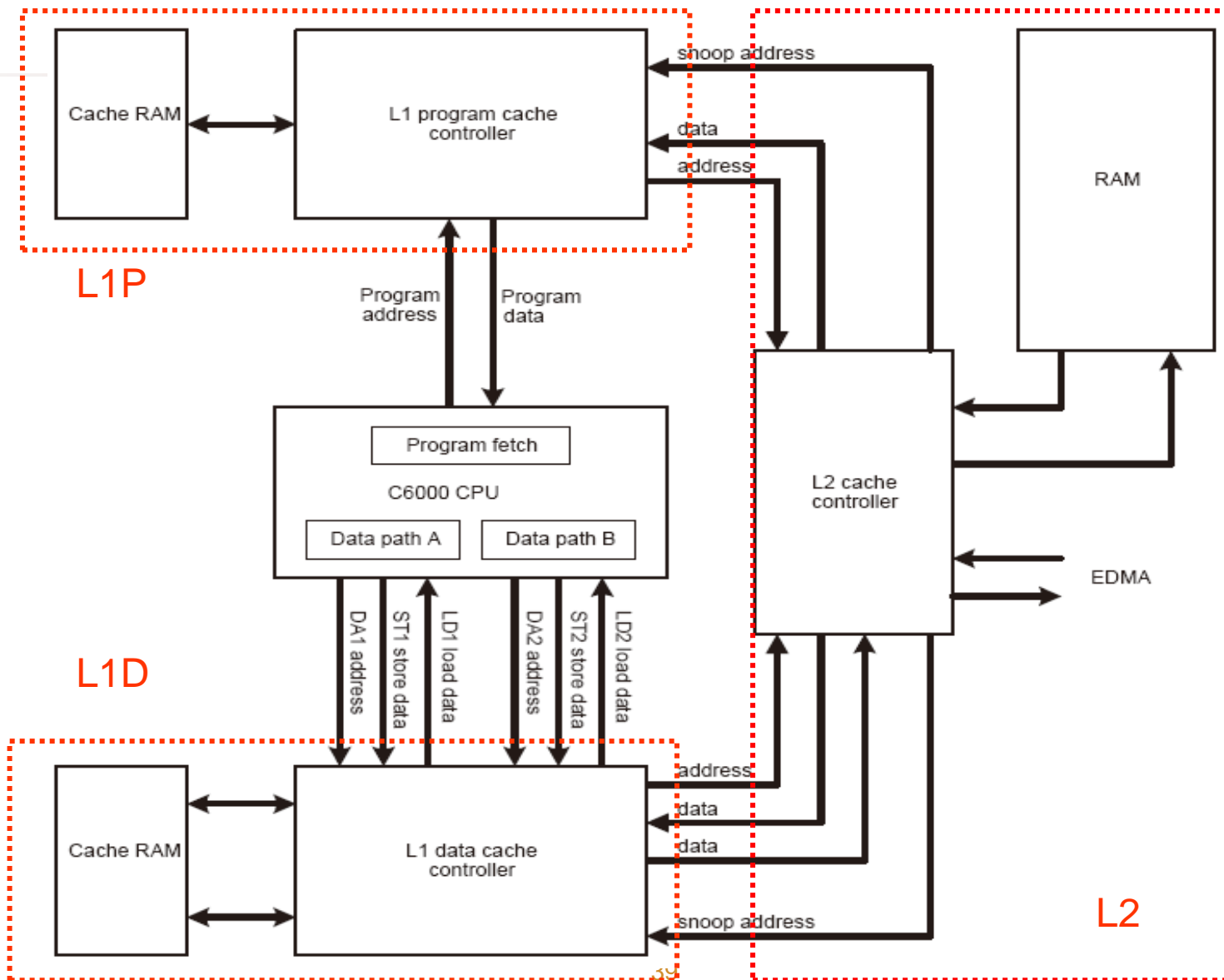
- * L1分L1P (16KB)、L1D (16KB)，靠近内核，访问速度快Cache，用户不可访问

- * L2 (1MB) 靠近外设，访问速度略慢，用户可访问

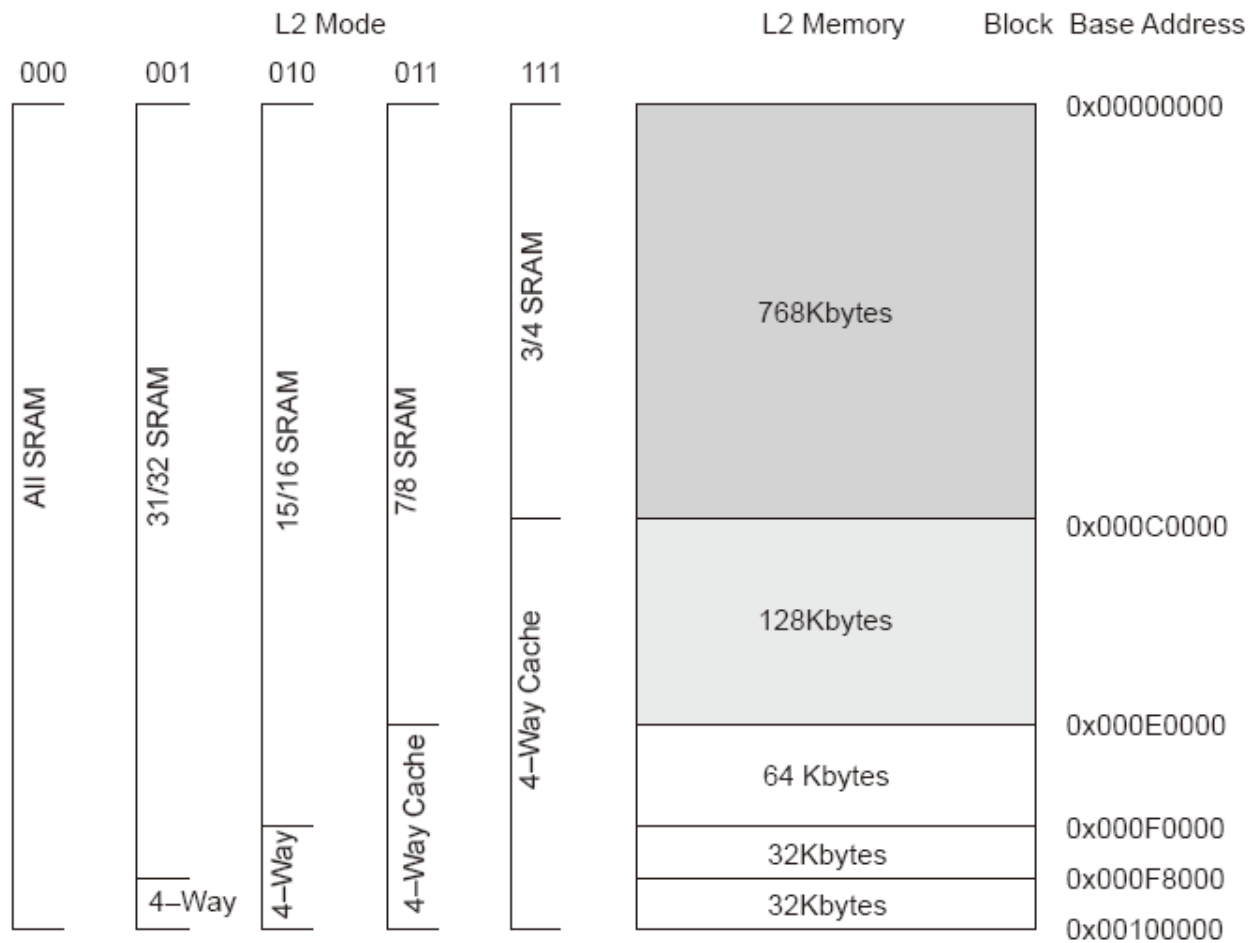
- * L2模式可配置，按字节单位统一编址

- * DMA直接访问L2，内核直接访问L1

9.3 L1/L2两级结构

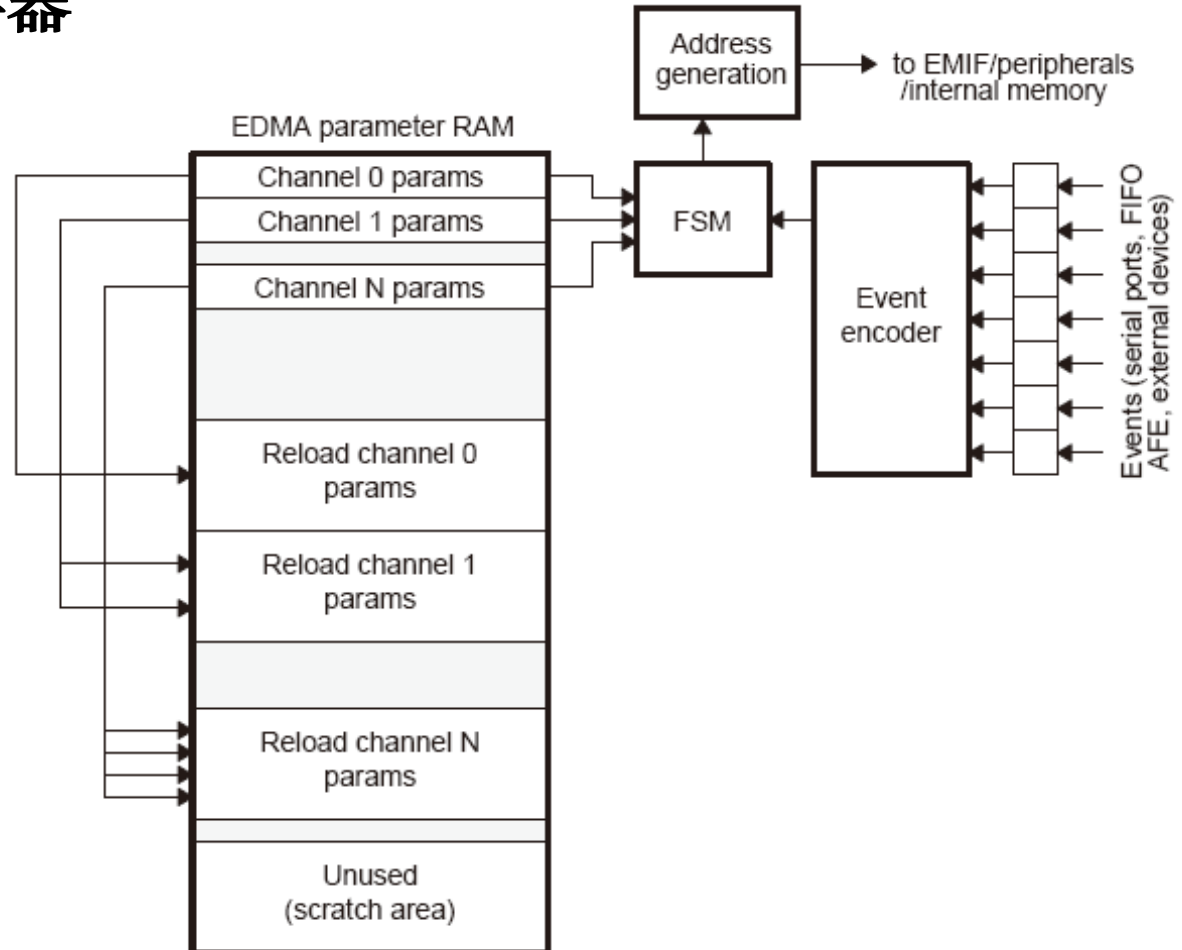


9.3 L2模式可配：调节SRAM和Cache比例



9.3 EDMA控制器组成

- * 事件和中断处理寄存器
- * 事件编码器
- * 参数RAM
- * 地址产生单元
- * 事件结束检测器



9.3 EDMA控制器功能

- * 外设与L2存储器之间的直接数据传输
- * 独立于CPU内核工作
- * 支持8bit、16bit和32bit数据的存取
- * 提供64个独立通道，四个优先级级别

9.3 EDMA的启动

- * 方式一：CPU启动

CPU写ESR寄存器启动1个EDMA通道

适合程序控制的主动数据传输

- * 方式二：同步事件触发

外设中断等事件产生同步信号触发EDMA

适合外设突发数据传输

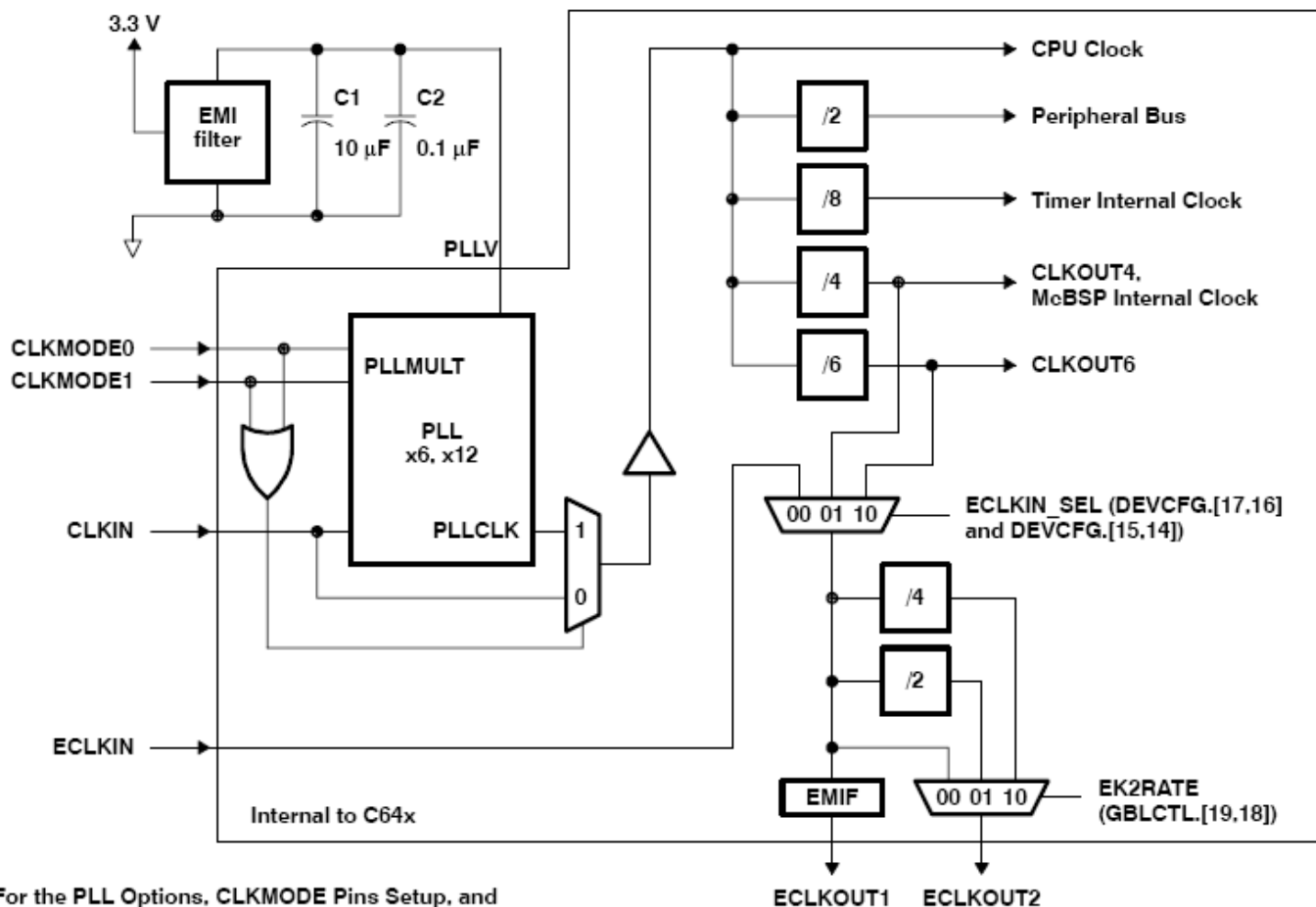
64个通道的启动是独立的

- * 方式三：链触发EDMA

9.3 EDMA与CPU之间关系

- * 双方可独立访问L2，但EDMA优先级低于CPU优先级
- * EDMA的启动需要CPU设置
- * EDMA传输完成后需向CPU递交中断信号

9.3 C64x的时钟网络结构



(For the PLL Options, CLKMODE Pins Setup, and PLL Clock Frequency Ranges, see Table 31.)

9.3 C64x的时钟网络

- * 片上时钟网络是设备协调工作的纽带
- * 时钟网络好比DSP的血管
- * 时钟网络反映了片上设备之间的联系
- * 请重视时钟网络的掌握

GLZ, ZLZ and CLZ PACKAGES – 23 x 23 mm BGA							
CLKMODE1	CLKMODE0	CLKMODE (PLL MULTIPLY FACTORS)	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY RANGE (MHz)	CLKOUT4 RANGE (MHz)	CLKOUT6 RANGE (MHz)	TYPICAL LOCK TIME (μ s) [§]
0	0	Bypass (x1)	30–75.75	30–75.75	7.5–18.9	5–12.6	N/A
0	1	x6	30–75.75	180–454.5	45–113.6	30–75.75	75
1	0	x12	30–60.6	360–727.2	90–181.8	60–121.2	
1	1	Reserved	–	–	–	–	–

9.3 外部存储器接口EMIF

* 可编程，支持多种类型存储器

同步静态存储器

同步动态存储器

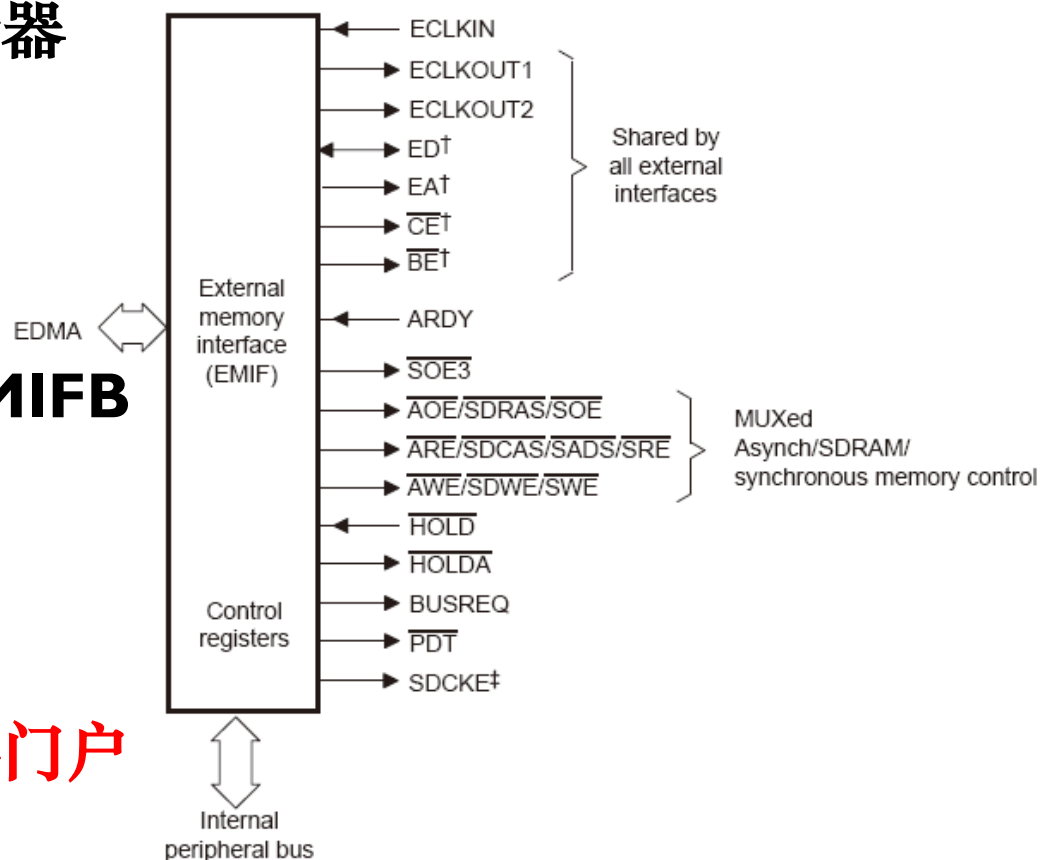
异步存储器

* C64X提供独立EMIFA、EMIFB接口

EMIFA 64位宽

EMIFB 16位宽

* EMIF是DSP数据I/O的重要门户



9.3 EMIF控制寄存器

Byte Address		Abbreviation	EMIF Register Name
EMIF/EMIFA	EMIFB†		
0180 0000h	01A8 0000h	GBLCTL	EMIF global control
0180 0004h	01A8 0004h	CE1CTL	EMIF CE1 space control
0180 0008h	01A8 0008h	CE0CTL	EMIF CE0 space control
0180 000Ch	01A8 000Ch		Reserved
0180 0010h	01A8 0010h	CE2CTL	EMIF CE2 space control
0180 0014h	01A8 0014h	CE3CTL	EMIF CE3 space control
0180 0018h	01A8 0018h	SDCTL	EMIF SDRAM control
0180 001Ch	01A8 001Ch	SDTIM	EMIF SDRAM refresh control
0180 0020h	01A8 0020h	SDEXT§	EMIF SDRAM extension§
0180 0024h to 0180 0040h	01A8 0024h to 01A8 0040h	—	Reserved
0180 0044h	01A8 0044h	CE1SEC¶	EMIF CE1 space secondary control¶
0180 0048h	01A8 0048h	CE0SEC¶	EMIF CE0 space secondary control¶
0180 004Ch	01A8 004Ch	—	Reserved
0180 0050h	01A8 0050h	CE2SEC¶	EMIF CE2 space secondary control¶
0180 0054h	01A8 0054h	CE3SEC¶	EMIF CE3 space secondary control¶

9.3 EMIF控制寄存器

- * 全局控制寄存器 GBLCTL
对每个CE空间的公共参数设置
- * CE空间控制寄存器 CExCTL (x= 0,1,2,3)
对应CE空间的控制寄存器
- * CE空间二级控制寄存器 CExSEC (x= 0,1,2,3)
控制可编程同步存储器访问的周期时序
- * SDRAM控制寄存器 SDCTL
控制SDRAM的全体CE空间的参数
- * SDRAM时序控制寄存器 SDTIM
控制SDRAM刷新周期
- * SDRAM扩展寄存器 SDEXT

9.3 全局控制寄存器 GBLCTL

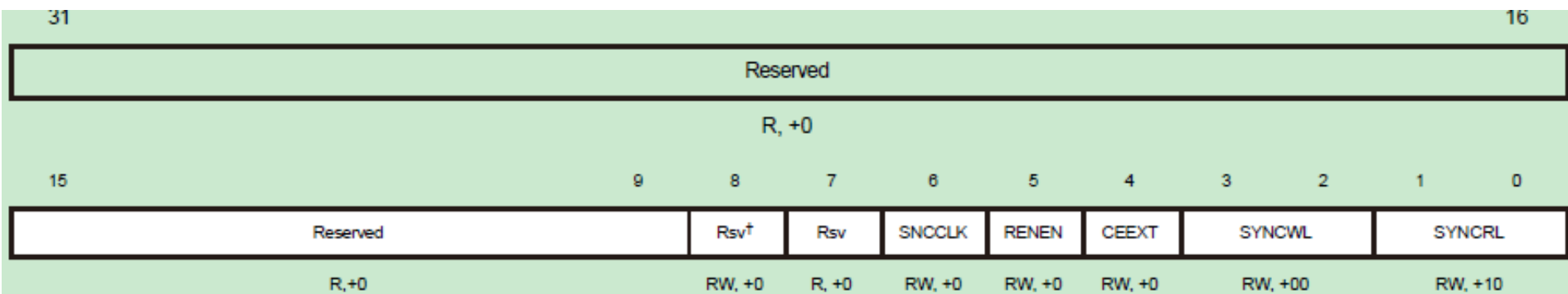
31																20				19	18	17	16
Rsv												EK2RATE		EK2HZ		EK2EN							
R _i +0												RW _i +10		RW _i +0		RW _i +1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Rsv		BRMODE	Rsv	BUSREQ	ARDY	HOLD	HOLDA	NO HOLD	EK1HZ	EK1EN	CLK4EN	CLK6EN	Rsv	Rsv	Rsv								
R _i +0		RW _i +1	R _i +0	R _i +0	R _i +x	R _i +x	R _i +x	RW _i +0	RW _i +1	RW _i +1	RW _i +1	RW _i +1	RW _i +1	R _i +0	R _i +0								

9.3 CE空间控制寄存器 CExCTL

31		28		27		22		21		20		19		16			
Write setup				Write strobe						Write hold		Read setup					
RW, +1111				RW, +111111						RW, +11		RW, +1111					
15		14		13		8		7		4		3		2		0	
TA		Read strobe						MTYPE				Write hold MSB [‡]		Read hold			
R, +11		RW, +111111						RW, +0010 [†]				RW, +0		RW, +011			

MTYPE = 0000b: 8-bit-wide asynchronous interface
 MTYPE = 0001b: 16-bit-wide asynchronous interface
 MTYPE = 0010b: 32-bit-wide asynchronous interface
 MTYPE = 0011b: 32-bit-wide SDRAM
 MTYPE = 0100b: 32-bit-wide SBSRAM (C621x/C671x)
 32-bit-wide programmable synchronous memory (C64x)
 MTYPE = 1000b: 8-bit-wide SDRAM
 MTYPE = 1001b: 16-bit-wide SDRAM
 MTYPE = 1010b: 8-bit-wide SBSRAM (C621x/C671x)
 8-bit-wide programmable synchronous memory (C64x)
 MTYPE = 1011b: 16-bit-wide SBSRAM (C621x/C671x)
 16-bit-wide programmable synchronous memory (C64x)
 MTYPE = 1100b: 64-bit-wide asynchronous interface (C64x only)
 MTYPE = 1101b: 64-bit-wide SDRAM (C64x only)
 MTYPE = 1110b: 64-bit-wide programmable synchronous memory (C64x only)

9.3 CE空间二级控制寄存器 CExSEC



SYNCRL Synchronous interface data read latency

SYNCRL = 00: 0 cycle read latency
 SYNCRL = 01: 1 cycle read latency
 SYNCRL = 10: 2 cycle read latency
 SYNCRL = 11: 3 cycle read latency

SNCCLK Synchronization Clock

SNCCLK = 0, control/data signals for this CE space are synchronized to ECLKOUT1
 SNCCLK = 1, control/data for this CE space are synchronized to ECLKOUT2

SYNCWL Synchronous interface data write latency

SYNCWL = 00: 0 cycle write latency
 SYNCWL = 01: 1 cycle write latency
 SYNCWL = 10: 2 cycle write latency
 SYNCWL = 11: 3 cycle write latency

RENEN Read Enable Enable

RENEN = 0: ADS mode. $\overline{\text{SADS}}/\overline{\text{SRE}}$ signal acts as $\overline{\text{SADS}}$ signal. $\overline{\text{SADS}}$ goes active for reads, writes, and deselect. Deselect is issued after a command is completed if no new commands are pending from the EDMA. (used for SBSRAM or ZBT SRAM interface)

RENEN = 1: Read Enable mode. $\overline{\text{SADS}}/\overline{\text{SRE}}$ signal acts as $\overline{\text{SRE}}$ signal. $\overline{\text{SRE}}$ goes low only for reads. No deselect cycle is issued. (used for FIFO interface)

CEEXT CE extension register

CEEXT = 0: CE goes inactive after the final command has been issued (not necessarily when all the data has been latched).

CEEXT = 1: On read cycles, the CE signal will go active when $\overline{\text{SOE}}$ goes active and will stay active until $\overline{\text{SOE}}$ goes inactive. The $\overline{\text{SOE}}$ timing is controlled by SYNCRL. (used for synchronous FIFO reads with glue, where $\overline{\text{CE}}$ gates $\overline{\text{OE}}$)

9.4 程序存储器—FLASH

- * 1984年东芝舛冈富士雄博士 发明
- * 1988年Intel推出首款NOR FLASH
- * 1989年东芝推出NAND FLASH
- * NOR型和NAND型

内部存储单元连接方式不同（平行连接 顺序连接）

操作接口不同（NOR 随机存储， NAND按页访问）

NOR容量相对小，NAND容量大

9.4 NOR FLASH—SST39VF080介绍

* 地址线

A[19: 0]

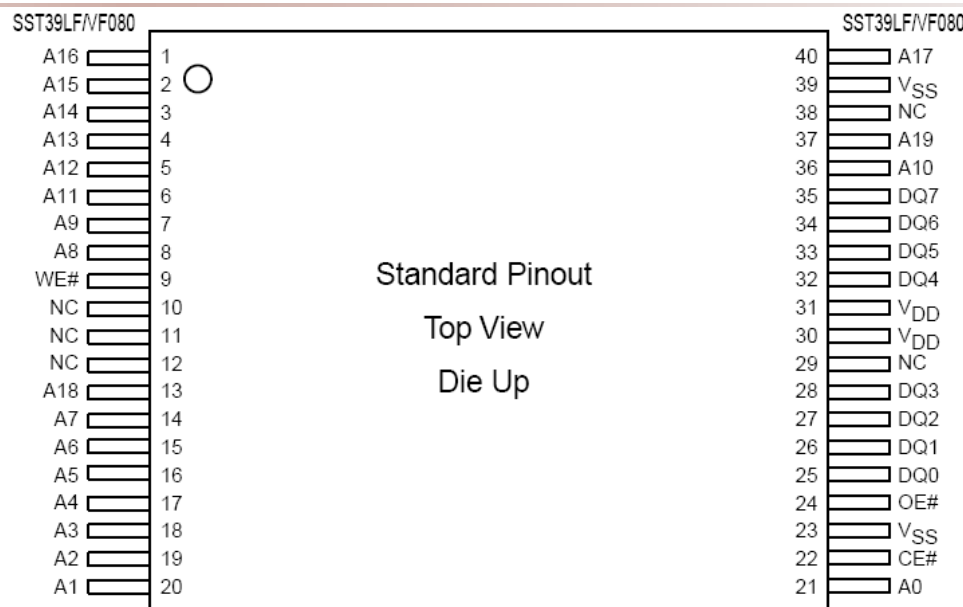
* 数据线

DQ[7: 0]

* 控制线

CE# OE# WE#

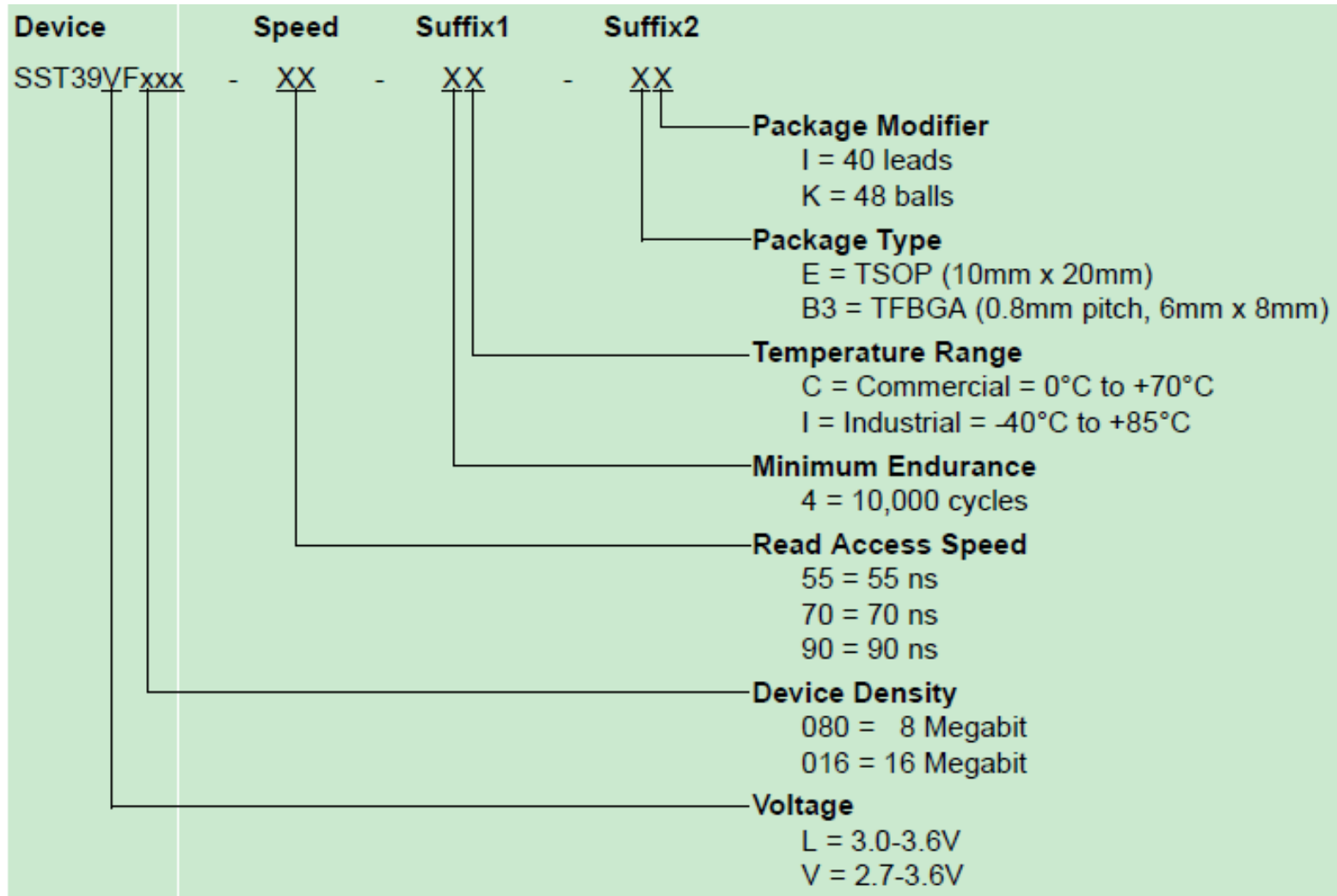
* 电源/地



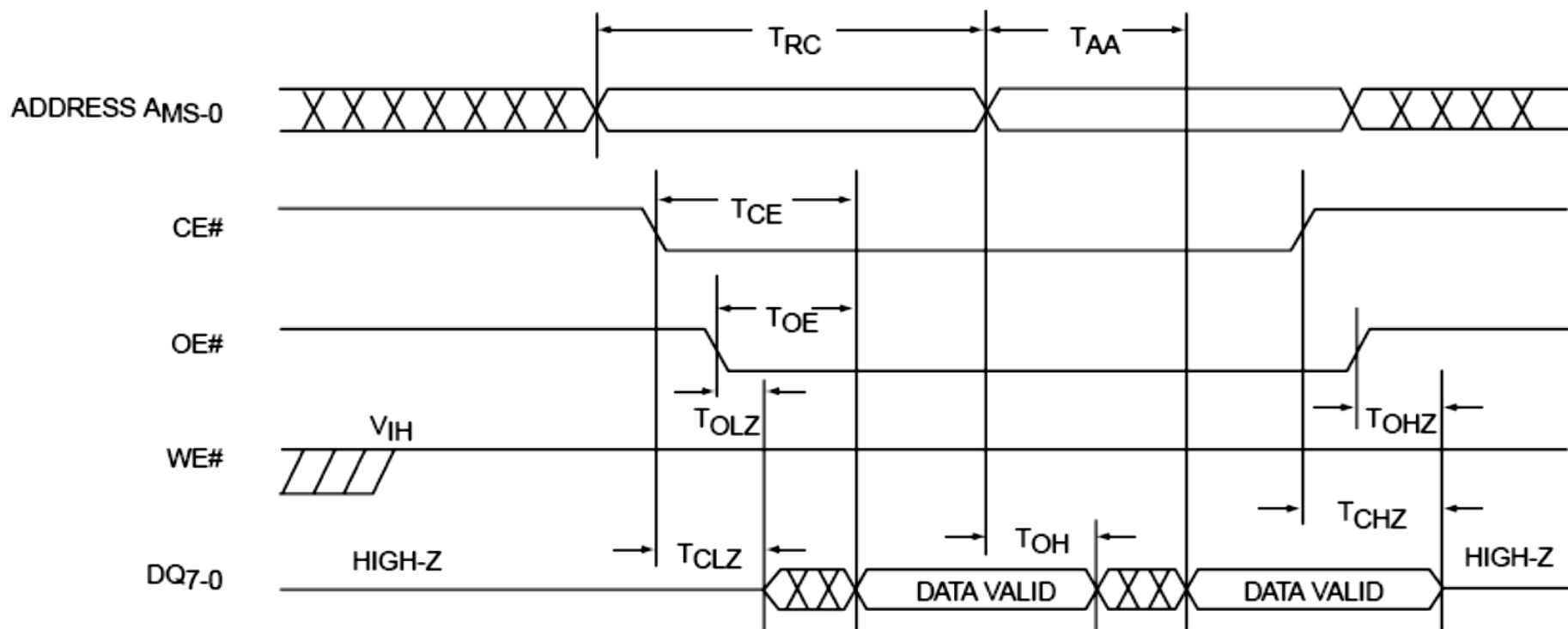
9.4 NOR FLASH—SST39VF080介绍

- * 非易失性存储器，1M×8bit
- * 按扇区（Sector）组织，扇区大小为4KB
- * 16个扇区构成一个块(Block)，64KB
- * 支持扇区、块、整片擦除
- * 只能逐个字节编程
- * 写操作需按照特定协议
- * 读操作速度要求

9.4 NOR FLASH—SST39VF080介绍



9.4 SST39VF080读操作时序



Note: A_{MS} = Most significant address
 $A_{MS} = A_{19}$ for SST39LF/VF080 and A_{20} for SST39LF/VF016.

396 ILL.

TRC时间—读操作速度 FLASH的关键性时序指标

9.4 SST39VF080读操作时序

TABLE 13: READ CYCLE TIMING PARAMETERS

$V_{DD} = 3.0\text{-}3.6\text{V}$ FOR SST39LF080/016 AND $2.7\text{-}3.6\text{V}$ FOR SST39VF080/016

Symbol	Parameter	SST39LF080/016-55		SST39VF080/016-70		SST39VF080/016-90		Units
		Min	Max	Min	Max	Min	Max	
T_{RC}	Read Cycle Time	55		70		90		ns
T_{CE}	Chip Enable Access Time		55		70		90	ns
T_{AA}	Address Access Time		55		70		90	ns
T_{OE}	Output Enable Access Time		30		35		45	ns
T_{CLZ}^1	CE# Low to Active Output	0		0		0		ns
T_{OLZ}^1	OE# Low to Active Output	0		0		0		ns
T_{CHZ}^1	CE# High to High-Z Output		15		20		30	ns
T_{OHZ}^1	OE# High to High-Z Output		15		20		30	ns
T_{OH}^1	Output Hold from Address Change	0		0		0		ns

T13.3 396

1. This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

TRC时间—读操作速度 FLASH的关键性时序指标

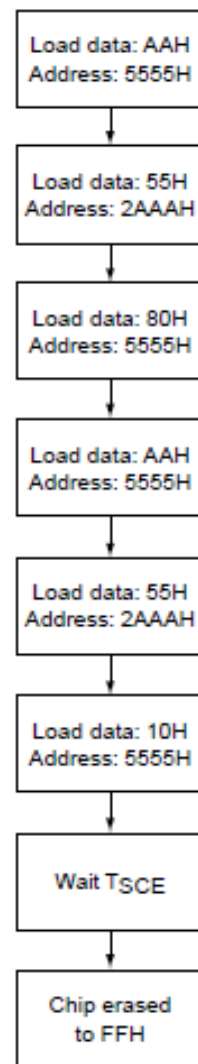
理解电路时序图

- * 时序图三要素关系：
信号组成、逻辑关系、定时关系

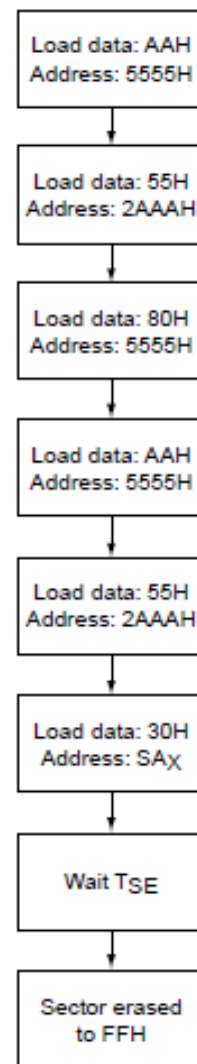
9.4 擦除流程

- * 编程写入前必须擦除
- * 整片/扇区/块 擦除
- 擦除速度差别
- 根据需求选用
- * 擦除后存储内容变为全“1”
- * 擦除等待时间
- * 最多擦除次数有限定

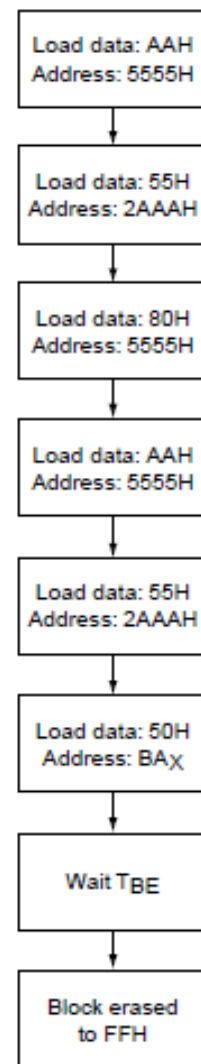
Chip-Erase
Command Sequence



Sector-Erase
Command Sequence

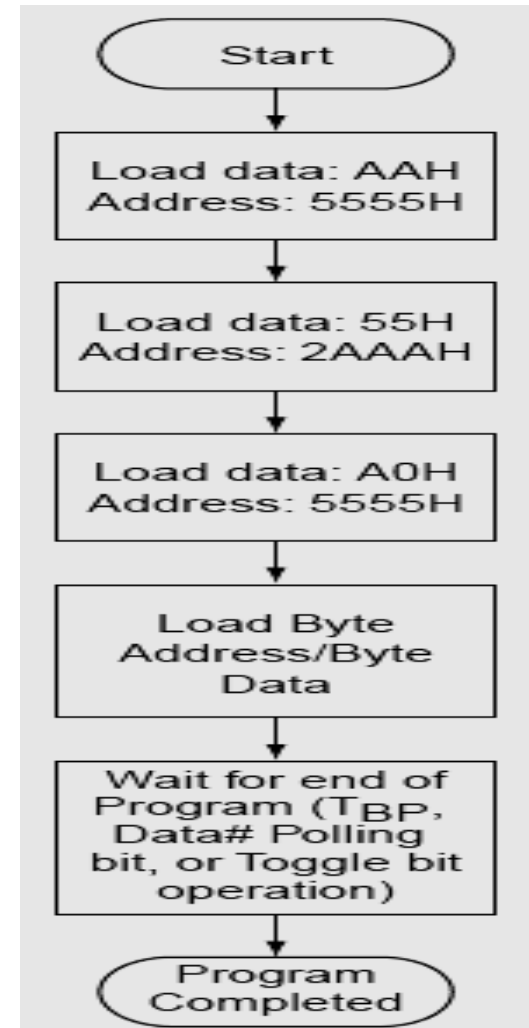


Block-Erase
Command Sequence

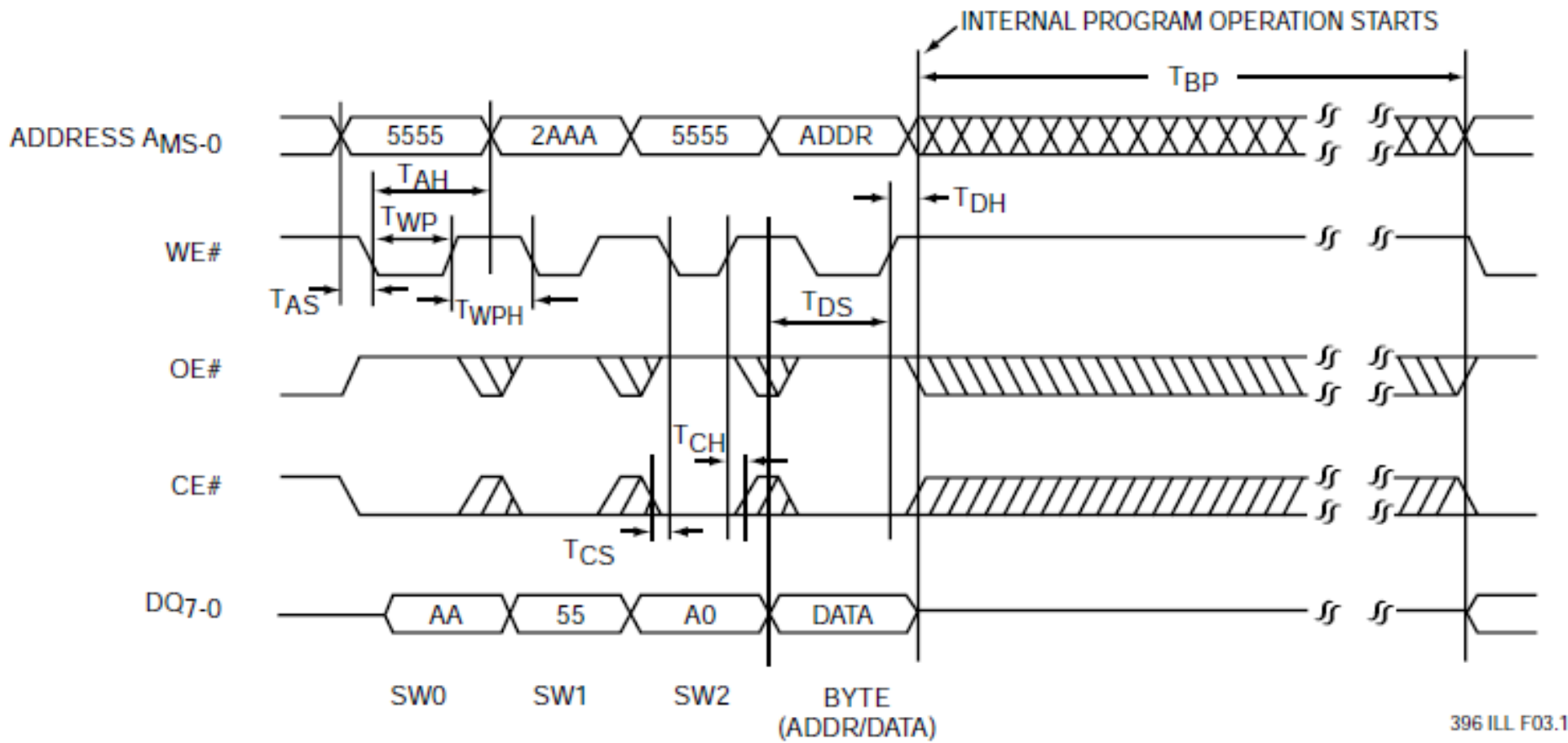


9.4 字节编程流程

- * 按字节编程
- * 编程前必须先擦除
- * 编程使存储比特位
“1” —> “0”
- * 不能 “0” —> “1”
- * 编程等待时间



9.4 字节编程时序



9.4 操作命令序列

SOFTWARE COMMAND SEQUENCE

Command Sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
	Addr ¹	Data	Addr ¹	Data	Addr ¹	Data	Addr ¹	Data	Addr ¹	Data	Addr ¹	Data
Byte-Program	5555H	AAH	2AAAH	55H	5555H	A0H	WA ²	Data				
Sector-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	SA _X ³	30H
Block-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	BA _X ³	50H
Chip-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	5555H	10H
Software ID Entry ^{4,5}	5555H	AAH	2AAAH	55H	5555H	90H						
CFI Query Entry ⁴	5555H	AAH	2AAAH	55H	5555H	98H						
Software ID Exit ⁶ / CFI Exit	XXH	F0H										
Software ID Exit ⁶ / CFI Exit	5555H	AAH	2AAAH	55H	5555H	F0H						

- Address format A₁₄-A₀ (Hex),
Addresses A₁₉-A₁₅ can be V_{IL} or V_{IH}, but no other value, for the Command sequence for SST39LF/VF080.
Addresses A₂₀-A₁₅ can be V_{IL} or V_{IH}, but no other value, for the Command sequence for SST39LF/VF016.
- WA = Program Byte address
- SA_X for Sector-Erase; uses A_{MS}-A₁₂ address lines
BA_X for Block-Erase; uses A_{MS}-A₁₆ address lines
A_{MS} = Most significant address
A_{MS} = A₁₉ for SST39LF/VF080 and A₂₀ for SST39LF/VF016
- The device does not remain in Software Product ID mode if powered down.
- With A_{MS}-A₁ = 0; SST Manufacturer's ID = BFH, is read with A₀ = 0
SST39LF/VF080 Device ID = D8H, is read with A₀ = 1
SST39LF/VF016 Device ID = D9H, is read with A₀ = 1
- Both Software ID Exit operations are equivalent

为什么如此？

9.4 EMIF可编程异步接口

- * 连接所有的异步存储器

如SRAM、EEPROM、FLASH等

- * 接口时序可编程

以ECLKOUT时钟周期为单位

建立时间(Setup): 访问开始到读/写有效之间时间 最小1

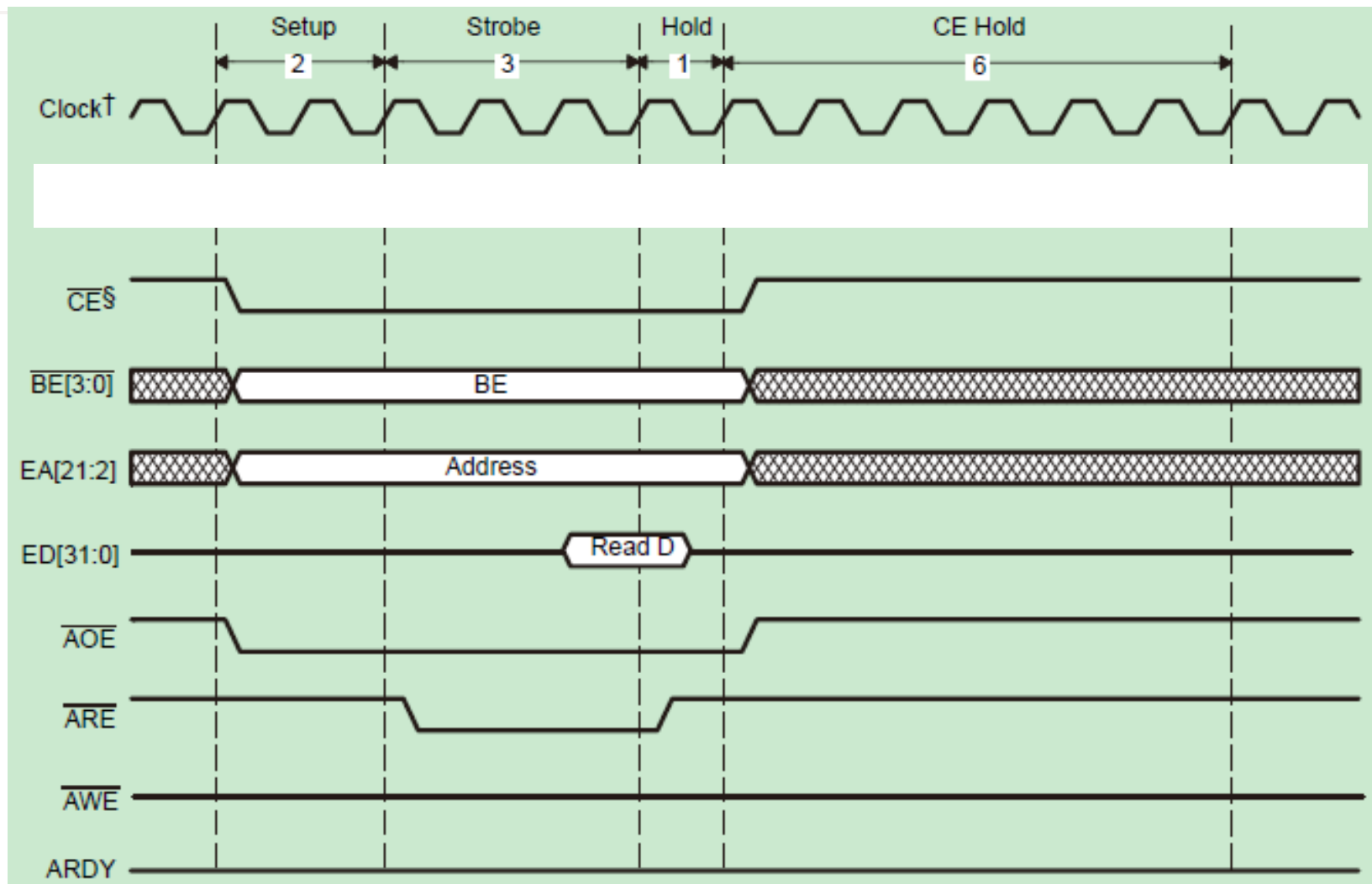
选通时间(Strobe): 读/写有效到无效之间时间 最小1

保持时间(Hold): 读/写无效到访问结束之间时间 最小0

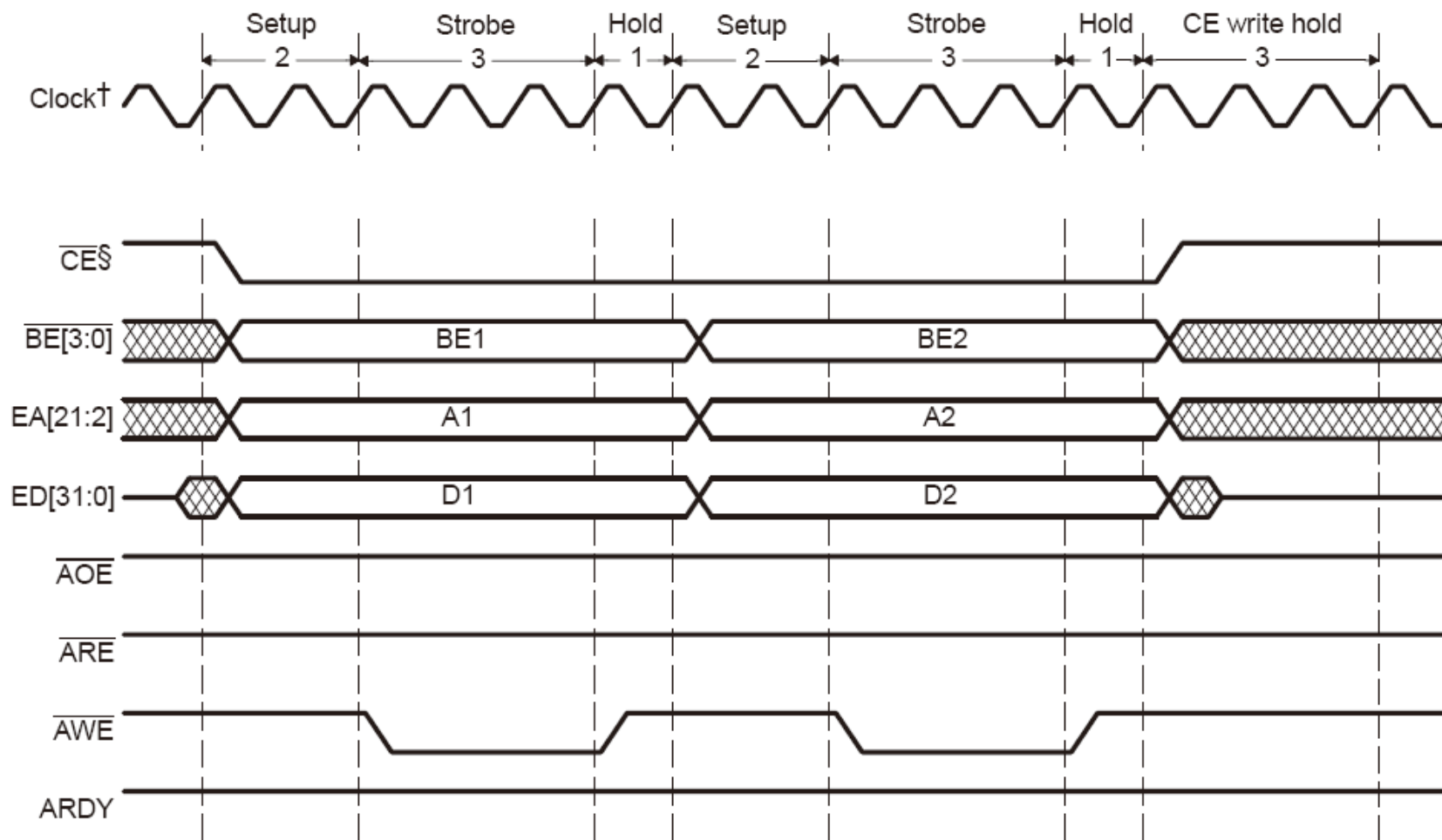
- * 接口信号

AOE# AWE# ARE# ARDY EA ED BE[3: 0]#

9.4 EMIF异步接口读时序



9.4 EMIF异步接口写时序



9.5 数据存储器— FIFO

- * FIFO(First In First Out) **双端口**存储器
- * **数据流管道**：数据写入D端，数据读出Q端
- * 与双口RAM相比连线少
- * 无地址线，**数据不能随机访问**
- * 同步FIFO和异步FIFO(较少用)



FIFO基本模型

9.5 数据存储器— FIFO

* 两种模式

标准模式 standard mode

FWFT(First Word First Through) 模式

写入操作没有差异，读出操作差异：

标准模式在读使能有效的下一周期数据有效

FWFT模式在读使能有效的当前周期数据有效

* 同步操作 vs 异步操作

读/写两端使用同源时钟—同步操作

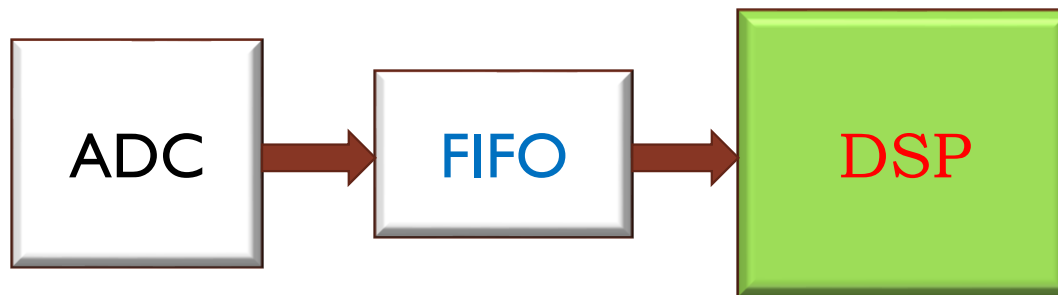
读/写两端使用不同源时钟—异步操作

9.5 FIFO的使用场合

* 速度匹配、数据互连

解决两个设备数据互联问题

解决两个设备速度差异，保持同步



9.5 FIFO的使用场合

- * FPGA内部实现使用FIFO

FPGA厂家都提供FIFO的IP核，用户调用
FPGA图像处理频繁使用FIFO

- * FIFO使用的要点—判断空/满标志，避免异常

FIFO满异常— 已满，仍然写入
FIFO空异常— 已空，仍然读出
FIFO异常后只能通过复位恢复

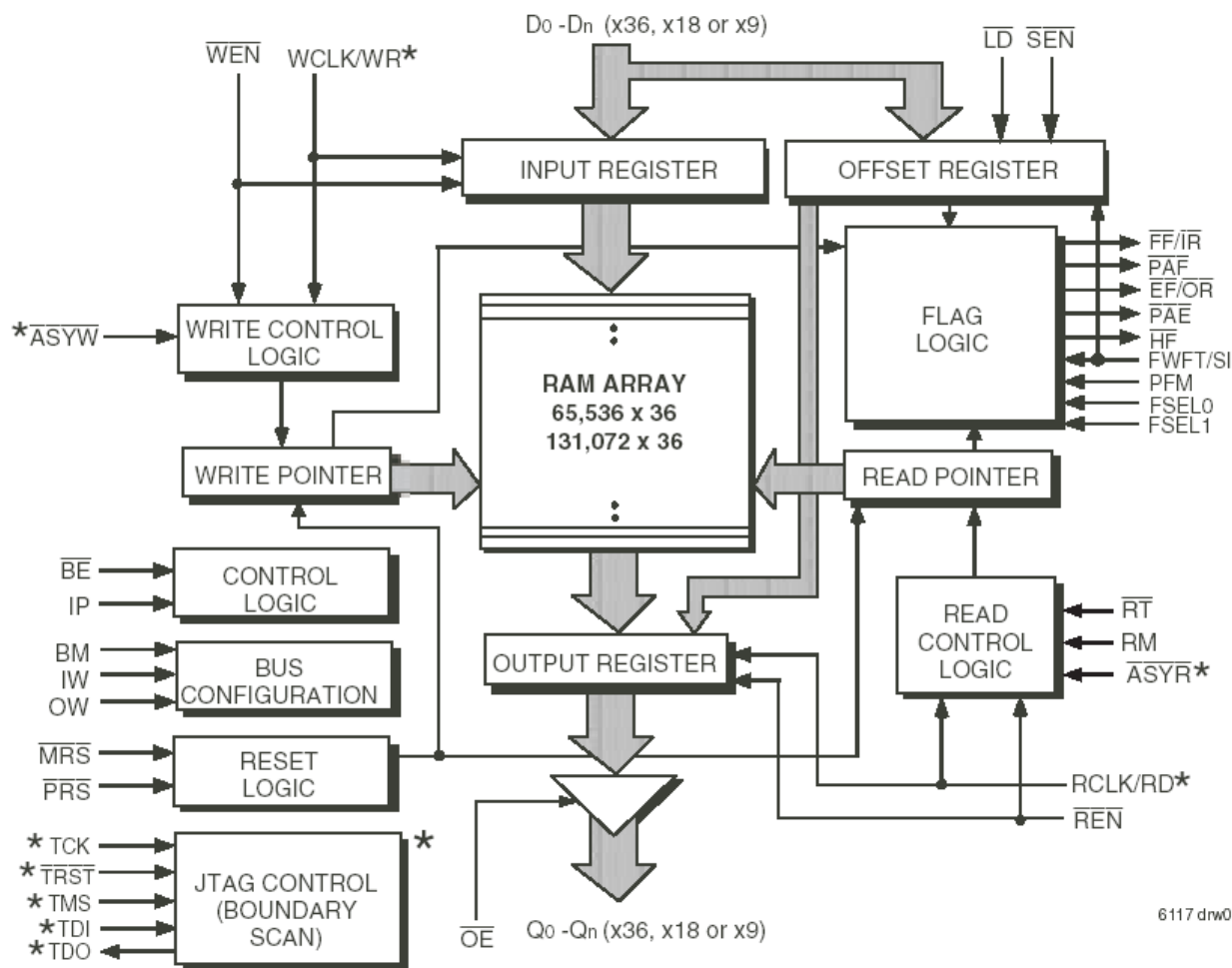
9.5 FIFO器件 IDT72V36110

- * 128K×36bit
- * 最快达166MHz操作时钟
- * 可编程的Almost Full/Almost Empty信号
- * 独立读写时钟
- * 输入D/输出Q位宽

BM	IW	OW	Write Port Width	Read Port Width
L	L	L	x36	x36
H	L	L	x36	x18
H	L	H	x36	x9
H	H	L	x18	x36
H	H	H	x9	x36

9.5 FIFO-IDT72V36110功能框图

*Available on the PBGA package only.



6117 drw01

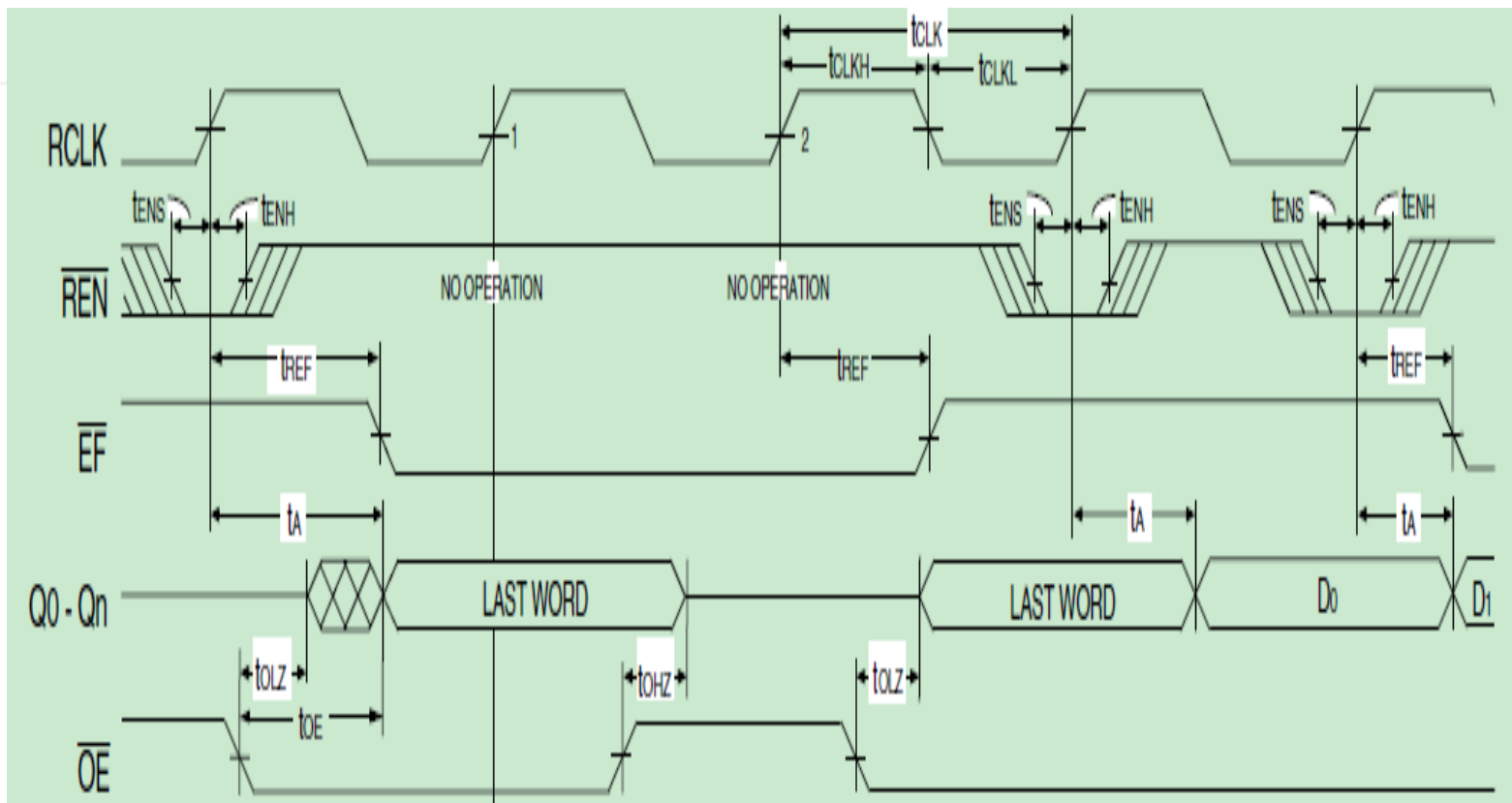
9.5 FIFO-IDT72V36110模式设置

IDT72V36100, 72V36110			
$\overline{\text{LD}}$	FSEL1	FSEL0	Offsets n,m
L	H	L	16,383
L	L	H	8,191
L	H	H	4,095
H	H	L	2,047
H	L	L	1,023
H	L	H	511
H	H	H	255
L	L	L	127
$\overline{\text{LD}}$	FSEL1	FSEL0	Program Mode
H	X	X	Serial ⁽³⁾
L	X	X	Parallel ⁽⁴⁾

NOTES:

1. n = empty offset for $\overline{\text{PAE}}$.
2. m = full offset for $\overline{\text{PAF}}$.
3. As well as selecting serial programming mode, one of the default values will also be loaded depending on the state of FSEL0 & FSEL1.
4. As well as selecting parallel programming mode, one of the default values will also be loaded depending on the state of FSEL0 & FSEL1.

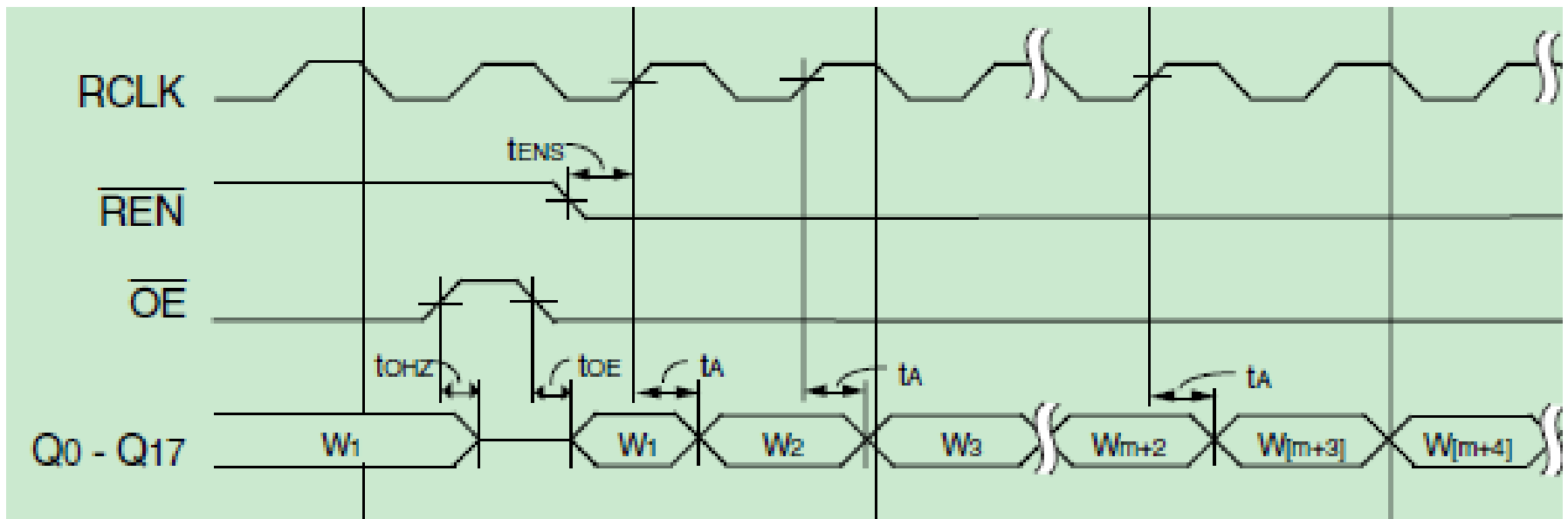
9.5 标准模式读时序



在当前时钟上升沿采样到读有效（低电平）、下一个时钟上升沿才读取到有效数据

读有效时刻延时一个时钟周期读到有效数据

9.5 FWFT模式读时序



在时钟上升沿采样到读有效（低电平）时刻，即读取到数据

读有效时刻读到数据，无时钟周期延时

9.5 EMIF可编程同步接口

- * 连接同步存储器

同步FIFO、ZBTSRAM、SBSRAM等

- * 接口时序可编程

- * 接口控制信号

SOE# SOE3# SWE# SADS/SRE#

ECLKOUT1、ECLKOUT2

9.5 同步FIFO接口

- * 支持标准模式和FWFT模式的FIFO

读时序差异 写入没有差异

- * 接口时序可编程

设置CE_xSEC中字段

RENEN字段设置为1

SYNCRL 读出延迟 最大3个周期

SYNCWL 写入延迟 最大3个周期

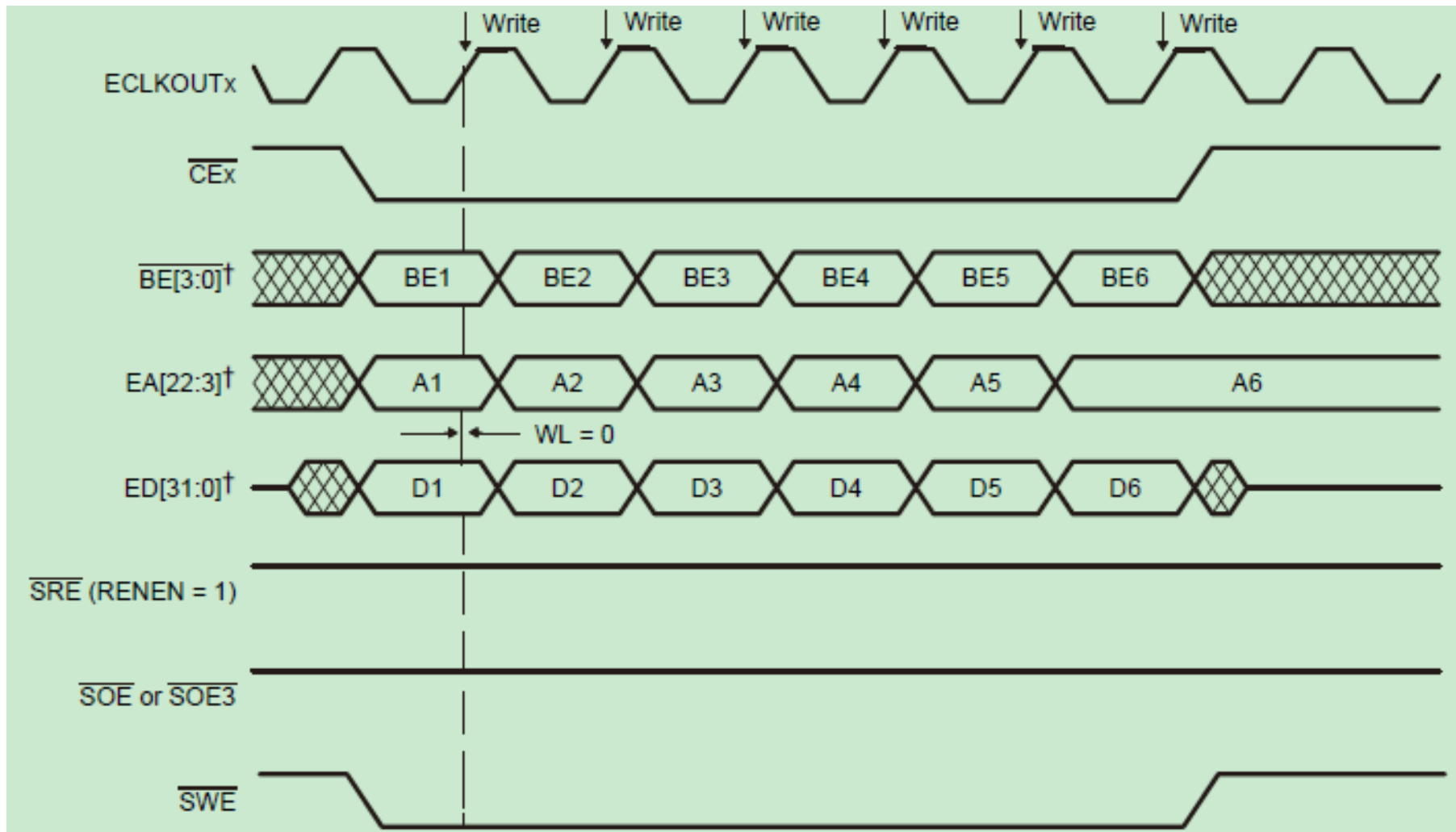
CEEXT CE#是否提前产生

- * 接口控制信号

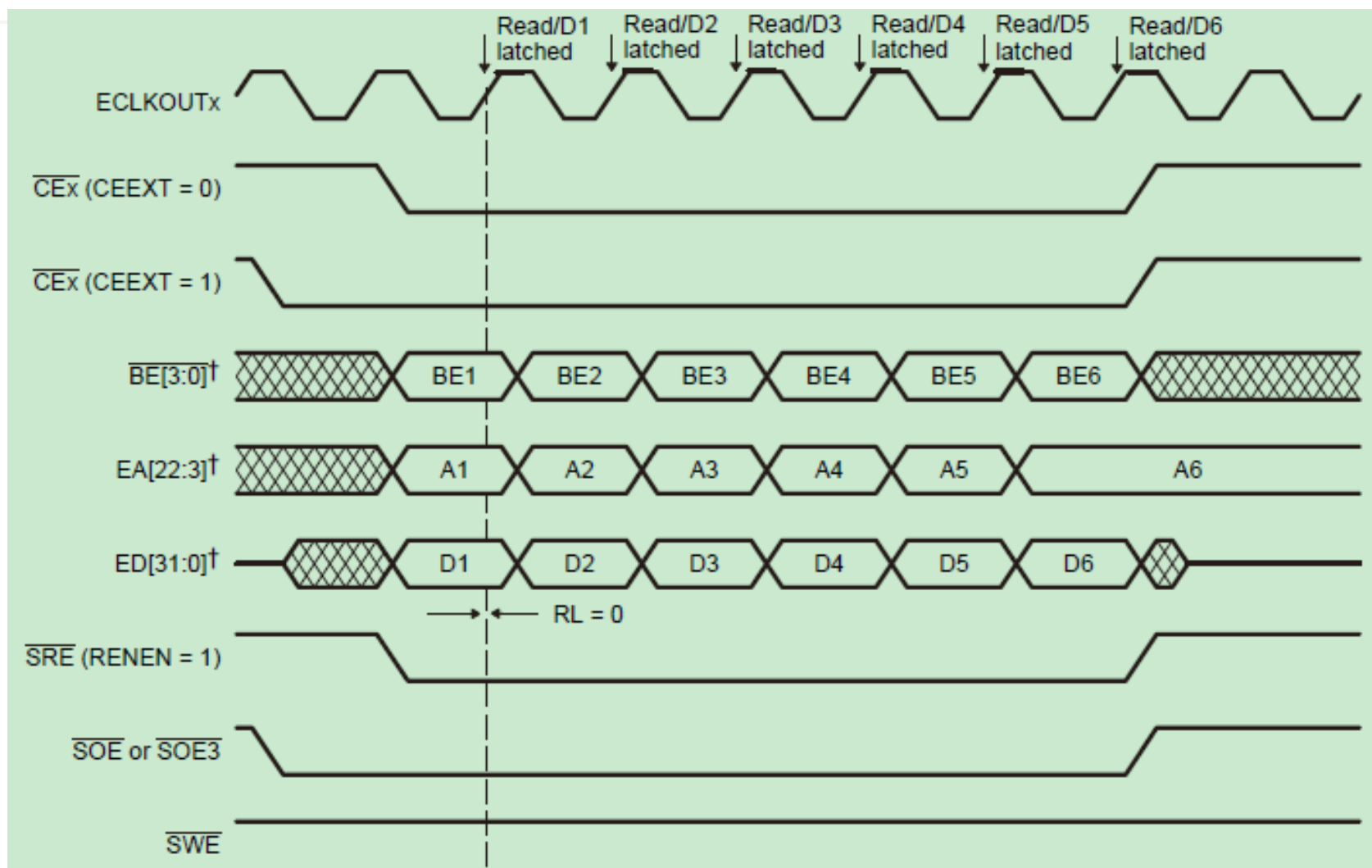
SOE# SOE3# SWE# SADS/SRE#

ECLKOUT1、ECLKOUT2

9.5 可编程同步接口-写FIFO

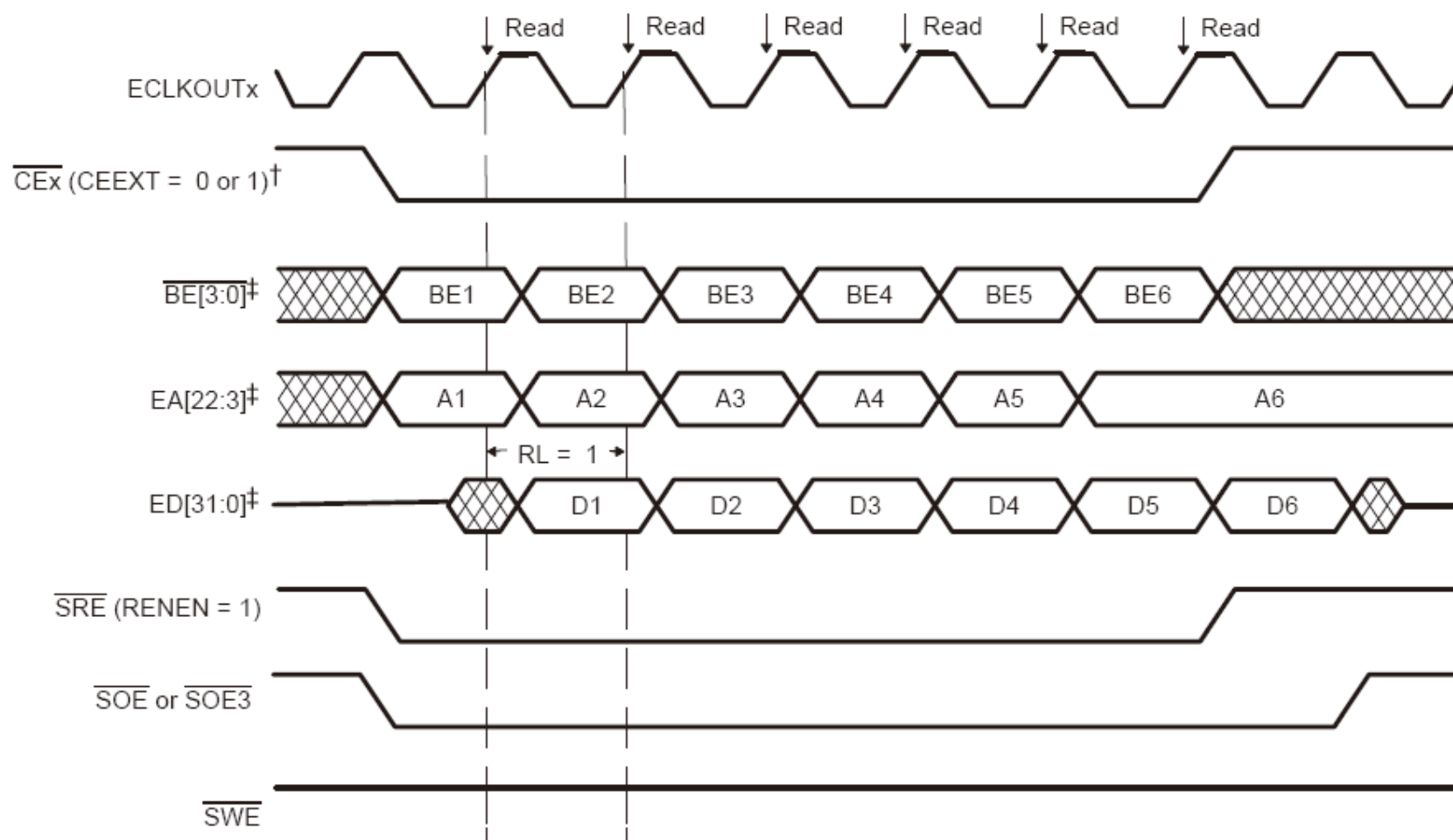


9.5 可编程同步接口—读FWFT模式FIFO



9.5 可编程同步接口—读标准模式FIFO

Figure 10–51. TMS320C64x Standard Synchronous FIFO Read



总 结

- ▣ DSP基本概念、特点、开发流程
- ▣ C641X DSP 体系架构
- ▣ FLASH、FIFO存储器接口技术
- ▣ DSP可编程存储器接口技术