



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY



TMS320C64X 外设接口

— DSP 培训课件之七

上海交大-TI 联合DSP实验室

版权所有



TMS320C64x的主要外设

- ④ C6000 系列 DSPs 不仅运算速度快，片内集成了很多外围设备，支持多种工业标准的接口协议，能够提供高带宽的数据 I/O 能力。这些特点使 C6000 系列 DSPs 获得了很高的综合性能。下图是 C6416 芯片的结构框图。
- ④ 下面是C64x的片内集成外设的简单介绍，详细说明参见文档：《TMS320C6000 DSP Peripherals Overview Reference Guide》，TI文档编号 SPRU190D。

TI DSP培训以及技术服务简介

上海交大BME-美国德州仪器联合DSP实验室成立于2007年，是国内最权威的TI技术服务于培训机构。实验室有TI（C6000，C2000，C5000，达芬奇，多核DSP）全系列开发平台，提供DSP，MSP430等技术培训与技术服务，项目合作等。培训内容有

- 1) CCS开发环境精解与实例；
- 2) DSP/SYS BIOS 实例；
- 3) C6000/C5000/C2000全系列DSP架构以及汇编，C语言，混合编程等；
- 4) HPI，EMIF，EDMA，Timer等外设；
- 5) C6416、DM642，C6678多核EVM开发平台实例；
- 6) Bootloader 原理以及实例等。

常年开班，三人以上集体报名8折优惠，学生5折。

联系电话：13651621236（牛老师），颁发TI授权证书

邮件报名：jhniu@sjtu.edu.cn，niujinhai@yahoo.com.cn



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



颁发TI授权的培训证书

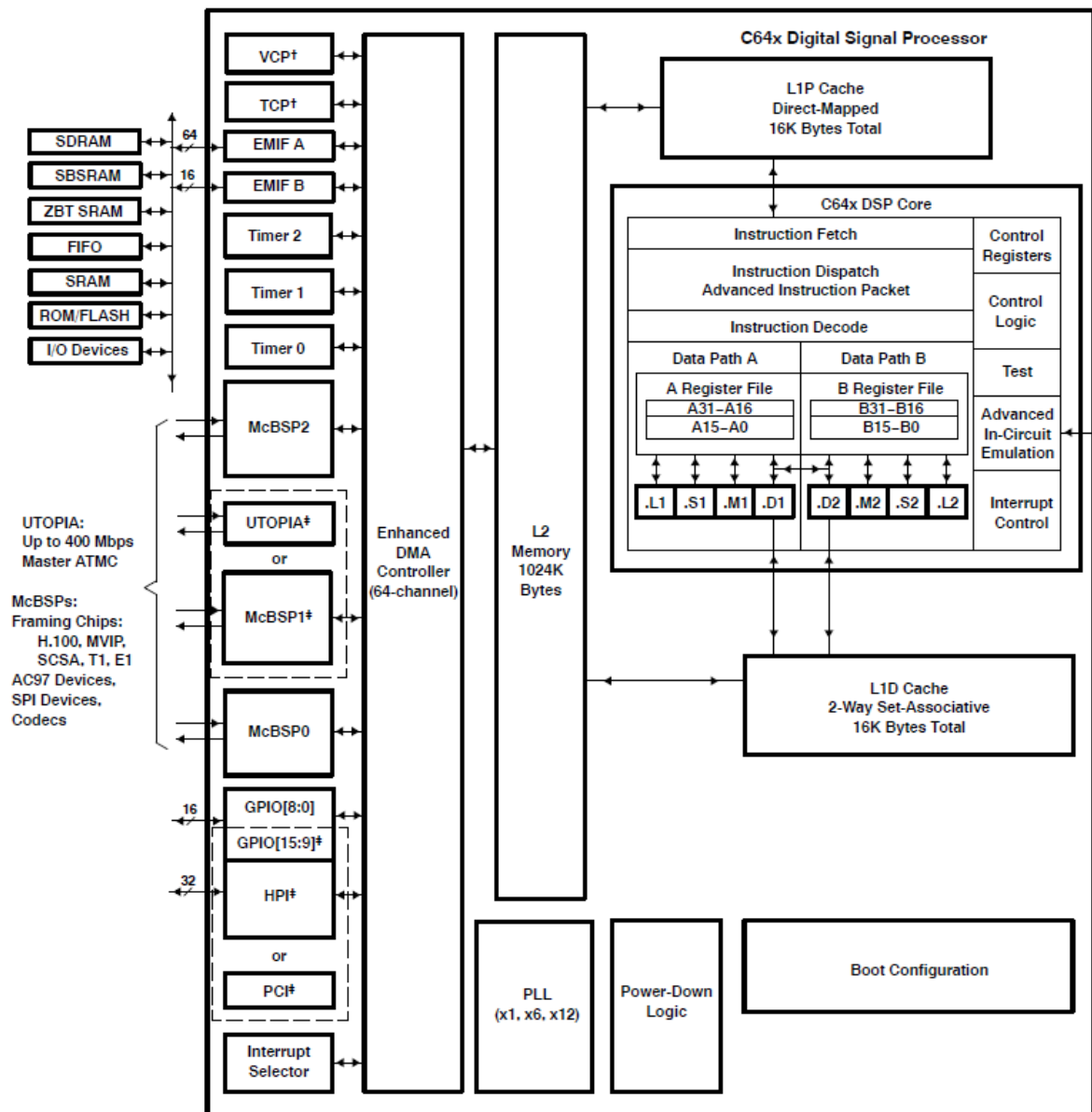


SHANGHAI JIAO TONG UNIVERSITY

the Office of the President



C6416芯片的结构框图



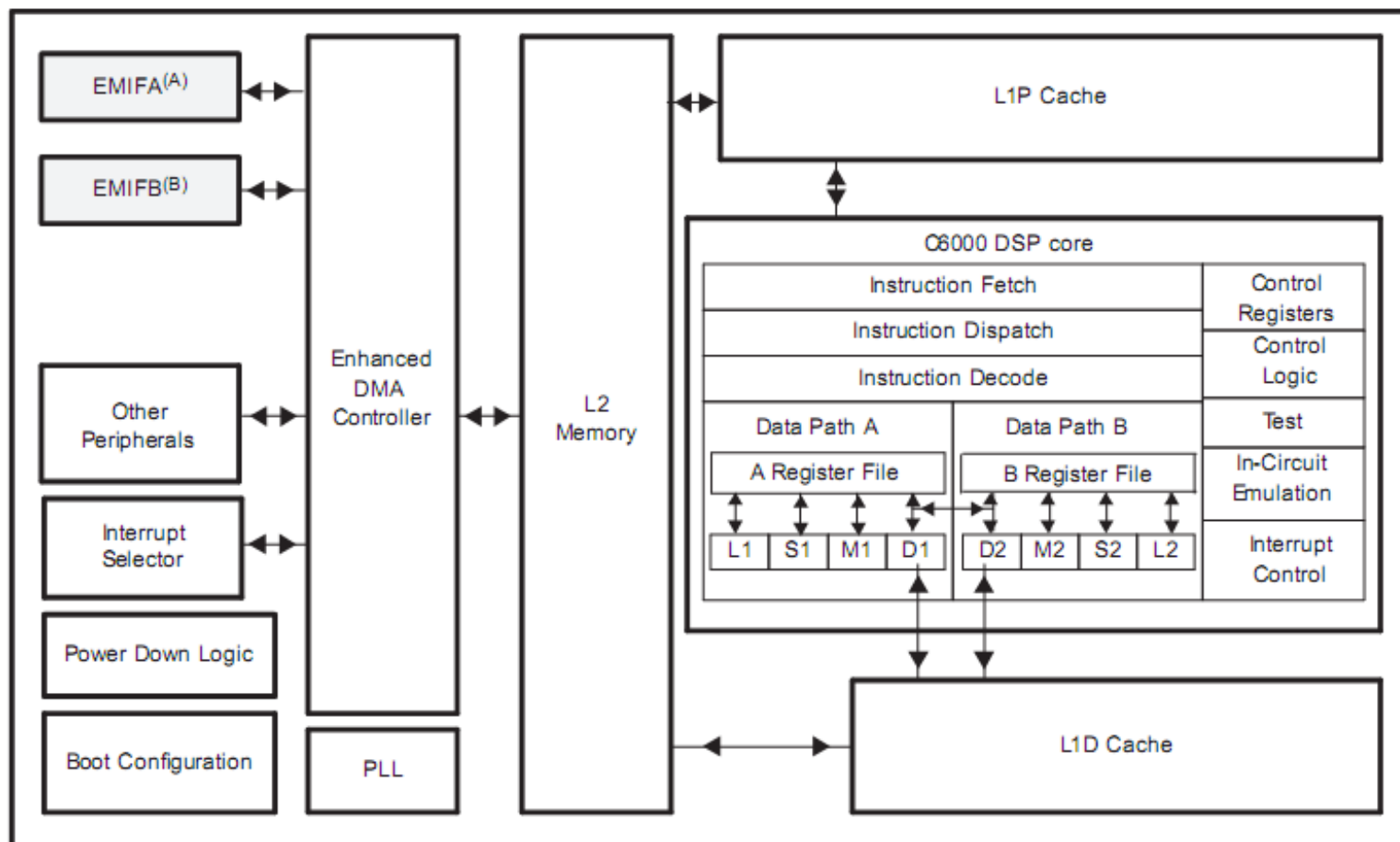


TMS320C64x的各外设及参考文档

外设	首字母缩写词	文献编号	C64x								DM64x			
			10	11	12	13	14	15	16	18	0	1	2	3
增强直接存储器存取控制器	EDMA	SPRU234	√	√	√	√	√	√	√	√	√	√	√	√
以太网媒体接入控制器/ 管理数据输入/输出模块	EMAC	SPRU628			√						√	√	√	√
外部存储器接口	EMIF	SPRU266	√	√	√	√	√	√	√	√	√	√	√	√
通用输入/输出	GPIO	SPRU584	√	√	√	√	√	√	√	√	√	√	√	√
主机端口接口	HP1	SPRU578	√	√	√	√	√	√	√	√		√	√	√
内部集成电路	I2C	SPRU175	√		√	√				√	√	√	√	√
内部集成电路附录	-	SPRZ221	√			√				√				
中断选择器	中断	SPRU646	√	√	√	√	√	√	√	√	√	√	√	√
多通道音频串行端口	McASP	SPRU041	√			√				√	√	√	√	√
多通道缓冲串行端口	McBSP	SPRU580	√	√	√	√	√	√	√	√	√	√	√	√
外设组件互连	PCI	SPRU581		√	√			√	√				√	
断电逻辑与模式	-	SPRU728	√	√	√	√	√	√	√	√	√	√	√	√
计时器，32 位	计时器	SPRU582	√	√	√	√	√	√	√	√	√	√	√	√
Turbo 解码器协处理器	TCP	SPRU534							√					
两级内部存储器	高速缓存	SPRU610	√	√	√	√	√	√	√	√	√	√	√	√
ATM 的通用测试和操作接口	UTOPIA	SPRU583						√	√					
视频端口 /VCX0 内插控制端口	视频端口	SPRU629									√	√	√	√
Viterbi 解码器协处理器	VCP	SPRU533							√	√				



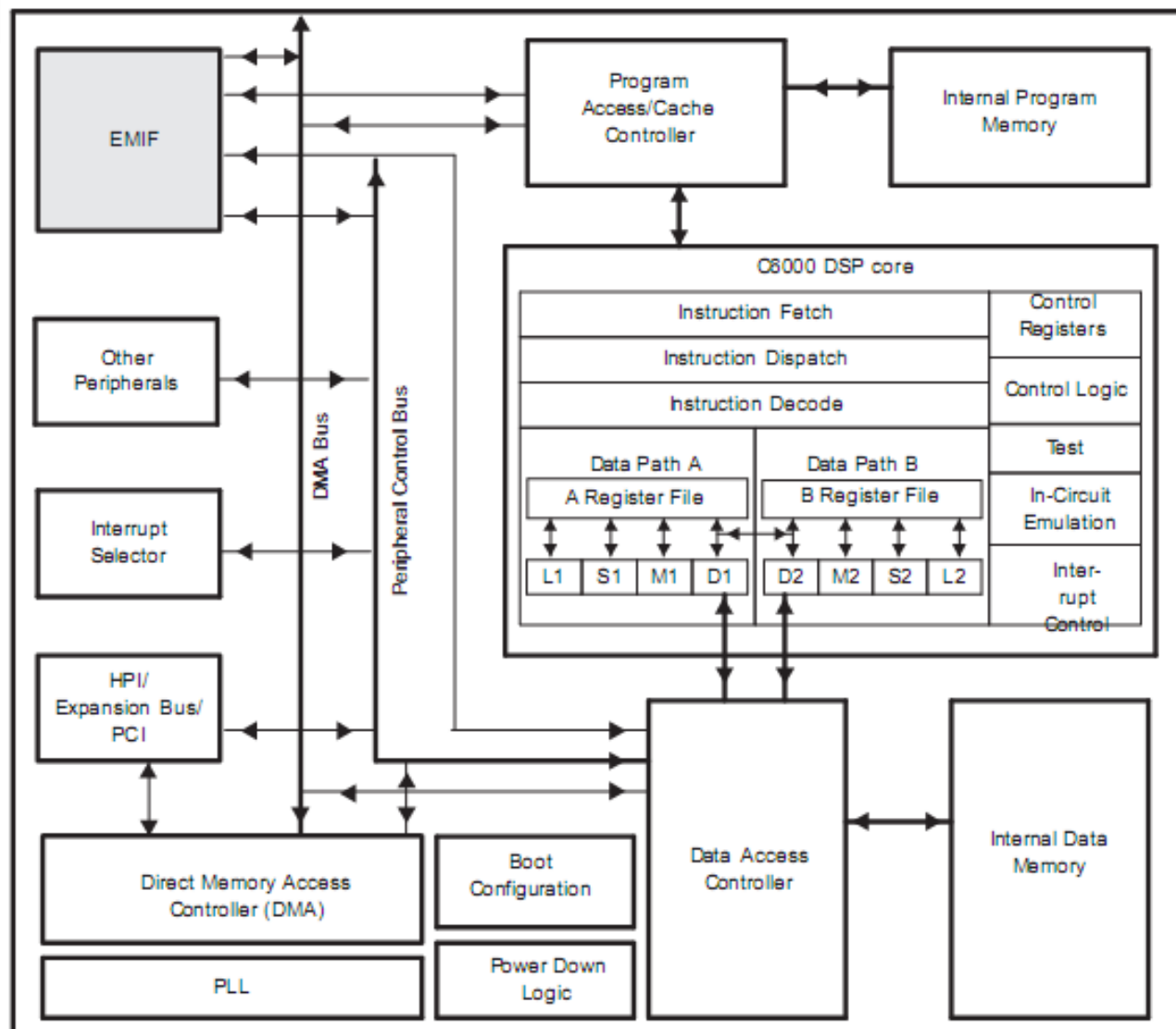
上海交通大学 TMS320C64x DSP 结构框图



TMS320C64x DSP Block Diagram



TMS320C62x/C670x DSP 结构框图



TMS320C620x/C670x DSP Block Diagram



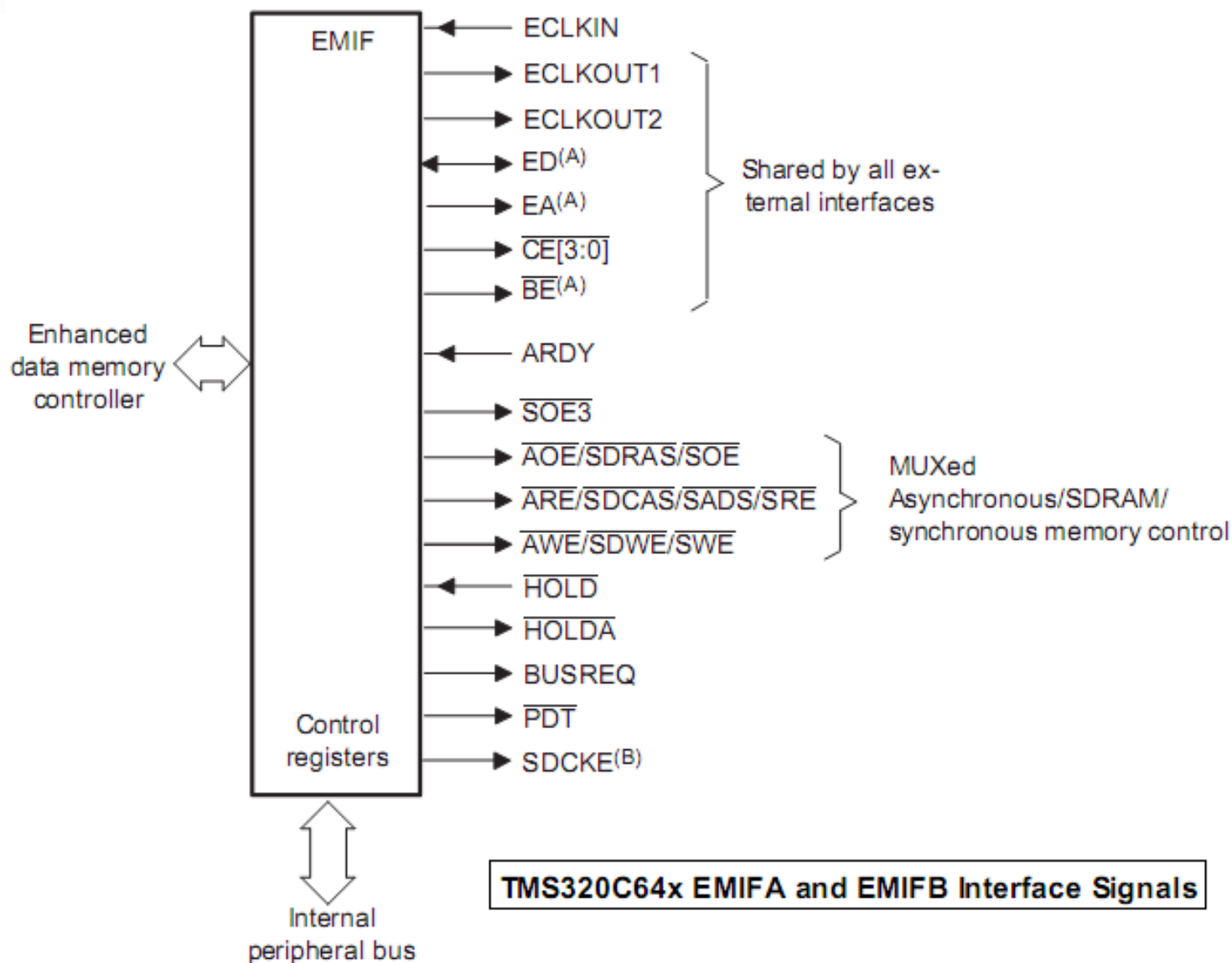
1) 扩展/外部存储器接口 (EMIF和EMIFA)

SHANGHAI JIAO TONG UNIVERSITY

- ④ DSPs访问片外存储器时必须通过外部存储器接口，C6000系列 DSPs 的 EMIF 具有很强的接口能力，不仅具有很高的数据吞吐率，而且可以与目前几乎所有类型的存储器直接接口。这些存储器包括：
 - 流水结构的同步突发静态 RAM (SBSRAM)
 - 同步动态 RAM (SDRAM)
 - 异步器件，包括 SRAM、ROM 和 FIFO等
 - 外部共享存储空间的设备
- ④ C64x具有两个独立的 EMIF：EMIFA和 EMIFB。MIFA 提供 64bit 宽度的外总线数据接口，EMIFB 提供 16bit 宽度的外总线数据接口（仅限于C6414/C6415/C6416）。C64x增强了原有的 SBSRAM 接口，提供可编程的同步接口模式。下图给出了 C64x 的 EMIF 接口信号。



C64x EMIF 的接口信号



TMS320C64x EMIFA and EMIFB Interface Signals

- 虽然C64x提供32bit的地址寻址能力，但是经过EMIF直接输出的地址信号只有EA[21:3]。一般情况下 EA2 信号对应于逻辑地址 A2，但这并不意味着 C64 处理器只能进行 32bit或者 64bit，实际上内部 32bit 地址的最低两位经过译码后由 BEx 输出，是能够控制字节访问的。某些情况下，EA2还可能对应最低位逻辑地址 A0，甚至对应逻辑地址 A11。更高位的逻辑地址经过译码后输出 EA[3:0]。



C64x EMIFA 寻址能力

C64x 的 EMIFA 支持 8/16/32/64bit 的数据访问，EMIFB 支持 8/16bit 的数据访问，同样支持 little-endian 和 big-endian 模式。下表总结了 C64x 的寻址能力。

存储器类型	存储器宽度	每个 CE 空间最大 可寻址范围	EA[21:2](EMIFA) EA[20:1](EMIFB) 输出的逻辑地址	含义
ASRAM	X8	1MB	A[19:0]	字节地址
	X16	2MB	A[20:1]	半字地址
	X32	4MB	A[21:2]	字地址
	X64	8MB	A[22:3]	双字地址
可编程同步 存储器	X8	1MB	A[19:0]	字节地址
	X16	2MB	A[20:1]	半字地址
	X32	4MB	A[21:2]	字地址
	X64	8MB	A[22:3]	双字地址



- EMIF 接口有一组存储器映射的寄存器进行维护与控制，包括配置各个空间的存储器类型和设置读写时序等，如下表所示：

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 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 0044h	01A8 0044h	CE1SEC [¶]	EMIF CE1 space secondary control [¶]
0180 0048h	01A8 0048h	CE0SEC [¶]	EMIF CE0 space secondary control [¶]
0180 0050h	01A8 0050h	CE2SEC [¶]	EMIF CE2 space secondary control [¶]
0180 0054h	01A8 0054h	CE3SEC [¶]	EMIF CE3 space secondary control [¶]



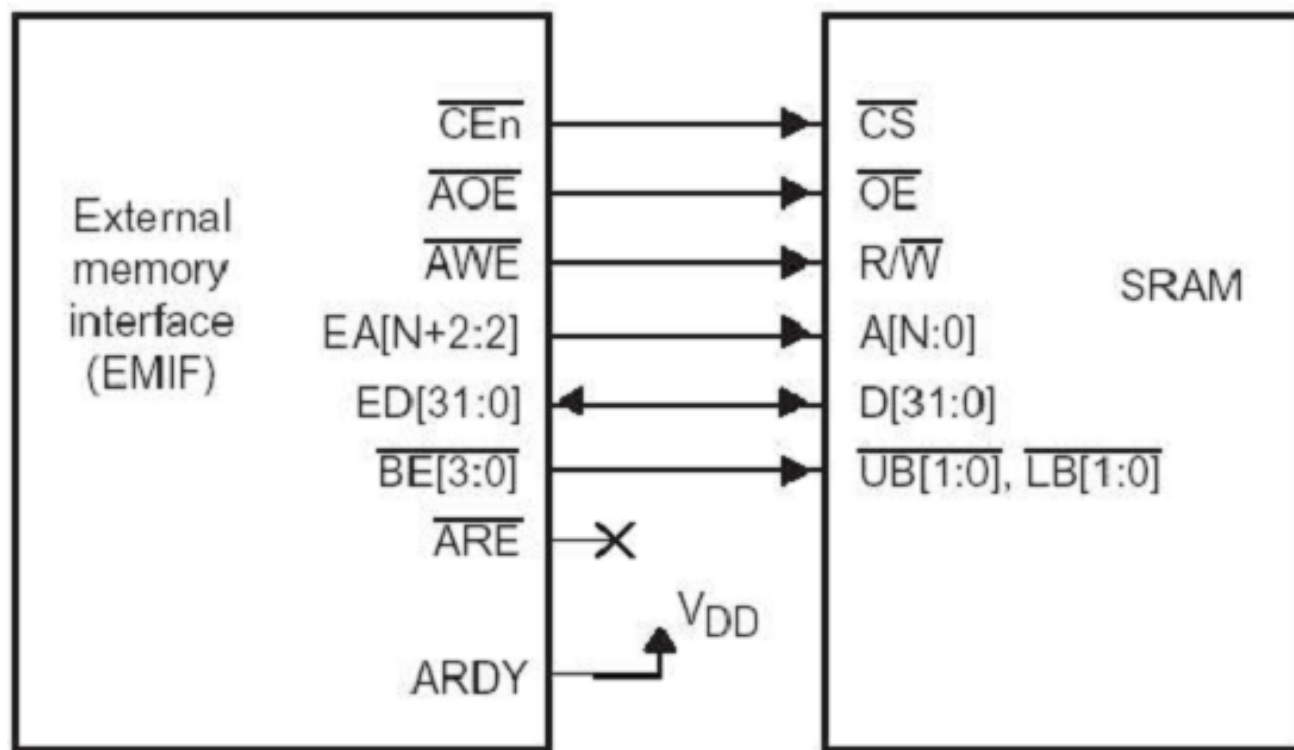
EMIF 异步接口举例

- EMIF 的异步接口提供 4 个控制信号，这 4 个控制信号可以通过不同的组合实现与不同类型的异步器件的无缝接口。EMIF 的 CExCTL 寄存器负责设置异步读写操作的接口时序，以满足对不同速度的异步器件的存取。

EMIF 异步接口信号	功能
/AOE	输出允许，在整个读周期中有效
/AWE	写允许，在写周期中触发阶段保持有效
/ARE	读允许，在读周期中触发阶段保持有效
ARDY	Ready 信号，插入等待



EMIF 与 32bit 异步 SRAM 的接口示意图



EMIF 与 32bit 的 ASRAM 的接口

上图中，只用了/AOE和/AWE 信号，没有使用 /ARE 和 ARDY 信号。



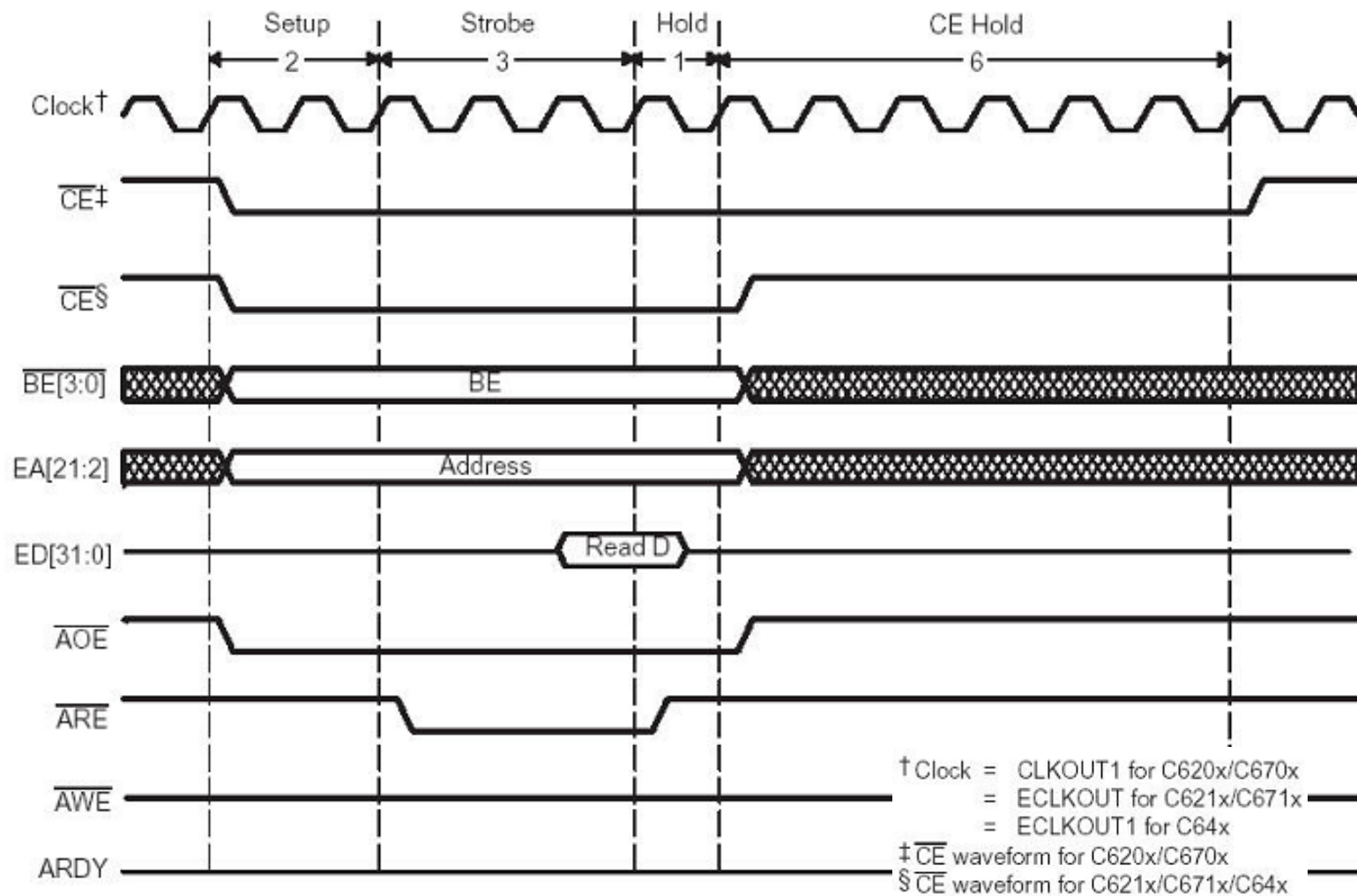
EMIF 读写时序

EMIF 的异步接口的可编程性好，每个读/写周期由 3 个阶段构成：建立时间（Setup）、触发时间（Strobe）和保持时间（Hold）。各自定义如下：

- 建立时间 从存储器访问周期开始到读/写选通信号有效之前
- 触发时间 读/写选通信号从有效到无效
- 保持时间 从读/写选通信号无效到访问周期结束

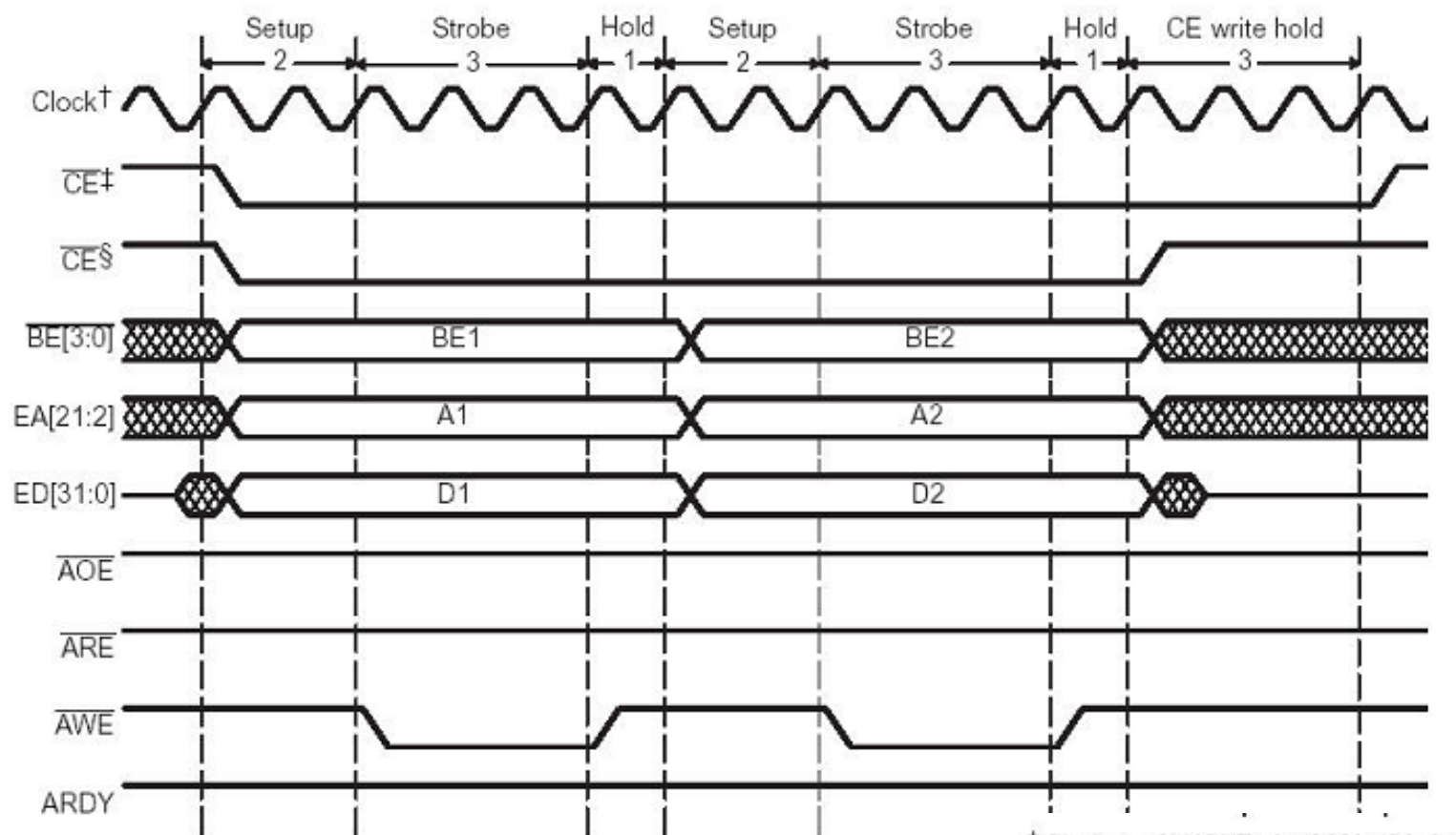


EMIF 读时序





EMIF 写时序



† Clock = CLKOUT1 for C620x/C670x
= ECLKOUT for C621x/C671x
= ECLKOUT1 for C64x

‡ CE waveform for C620x/C670x

§ CE waveform for C621x/C671x/C64x

对于异步接口，时序设计的关键是计算CExCTL寄存器中的Setup/Strobe/Hold这3个控制位的设置值，以及考虑时间裕量tmargin。



EMIF 支持不同位宽 ROM 接口

- EMIF 同时支持 8/16bit ROM 接口。当从些所谓的“窄存储空间”（narrow-width memory space）读取数据时，EMIF 会自动将多余的数据合成一个 32bit 值，送入片内总线。EMIF在这种情形下：
 - 不管存取的数据量和存储器的宽度怎样，读操作始终按 32bit 进行。



异步存储器的字节地址映射

- 输出的地址会自动进行移位调整，这保证了在对窄存储器存取操作时提供正确的地址。对 16bit ROM，地址自动左移一位，对 8bit 的 ROM，地址自动左移 2 位。移出的高位地址将被舍弃。如下表：

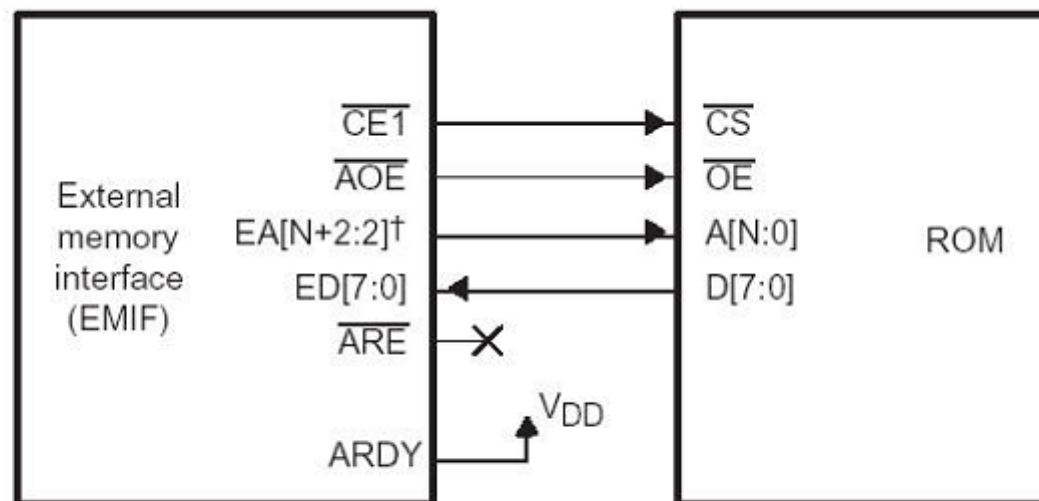
		EA Line																			
		21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Width		Logical Byte Address																			
×32		21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
×16		20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
×8		19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

不同宽度的异步存储器的字节地址映射

- EMIF 总是先读取较低地址的数据，将其排在 LSB，再读取下一个数据



EMIF与 8 位异步 ROM 的接口



[†] For C64x EMIFA, $EA[N+3:3]$ is used.
For C64x EMIFB, $EA[N+1:1]$ is used.

从上图中可看出：

第一，没有用到写信号；

第二，虽然是 8bit/16bit ROM，但是仍然是 EA2（EMIF 一侧）与 A0（ROM 一侧）相连。



主要内容

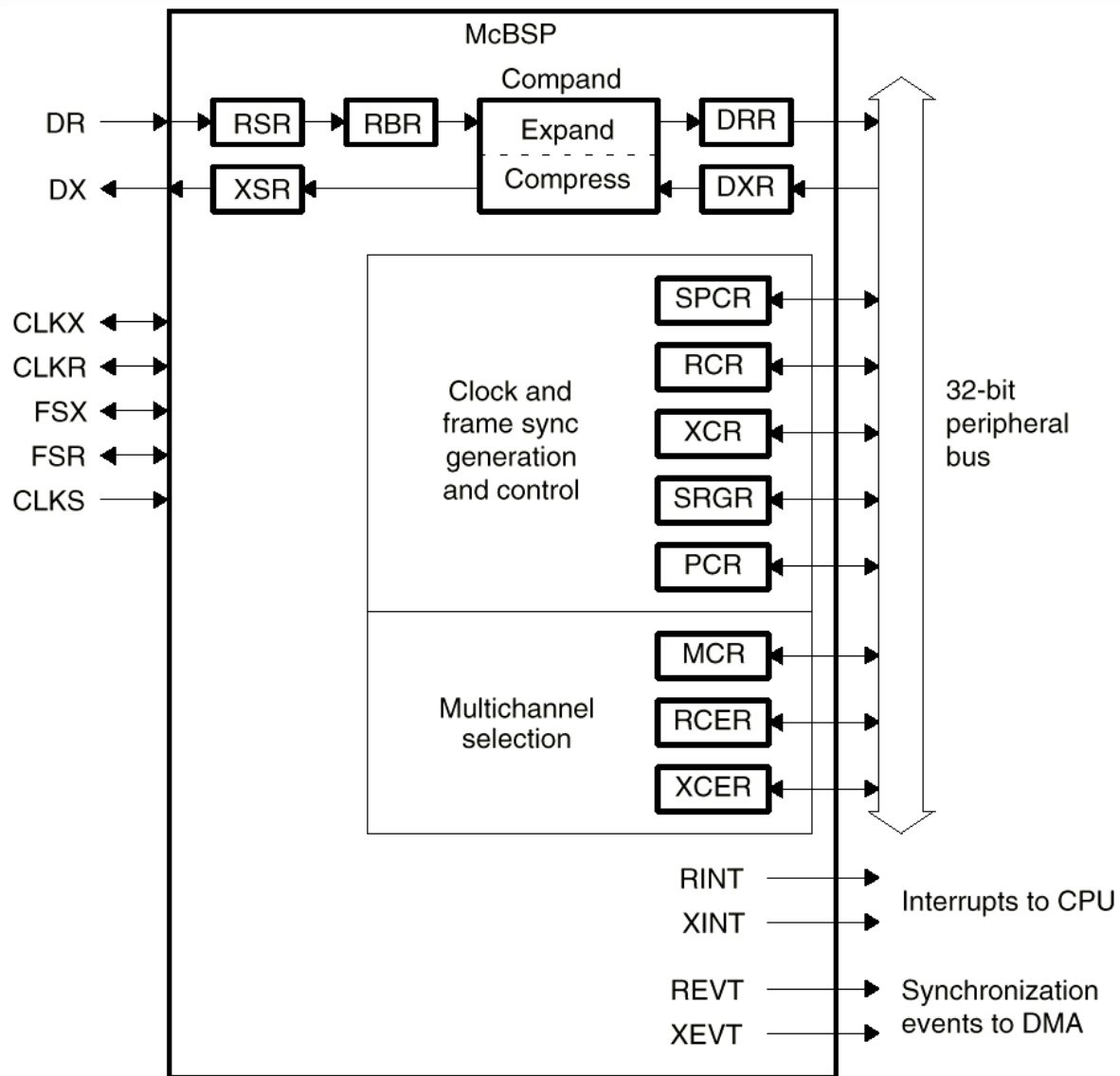
- 串口特点
- 数据的收发操作
- 多通道操作
- 硬件的 μ - A -律压扩
- SPI接口

概述—特点

- ④ 收/发独立、全双工、双缓冲数据通信，允许连续的数据流
- ④ 可以和与标准的编/解码器、AICs接口
- ④ 支持多种协议下的接口传输
 - T1/E1、MVIP、ST-BUS、IOM-2、AC97、IIS、SPI
- ④ 支持多通道操作（128）
- ④ 内置u-律和A-律压扩硬件
- ④ 内部传输时钟和帧同步信号可编程程度高
- ④ 数据字长可以是8-/12-/16-/20-/24-/32-bit



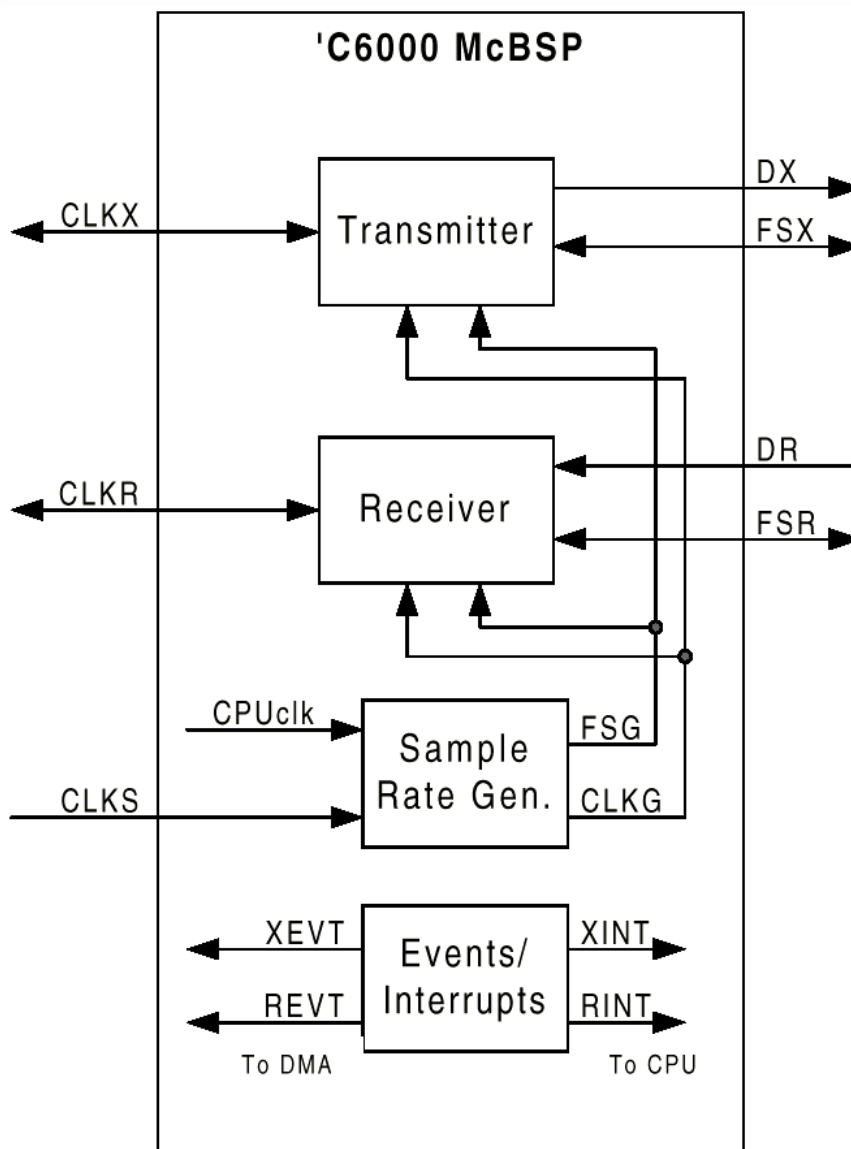
概述—框图



•数据通道
和控制通道



概述—信号



- 收/发
 - 时钟
 - 帧同步
 - 数据
- 外部时钟

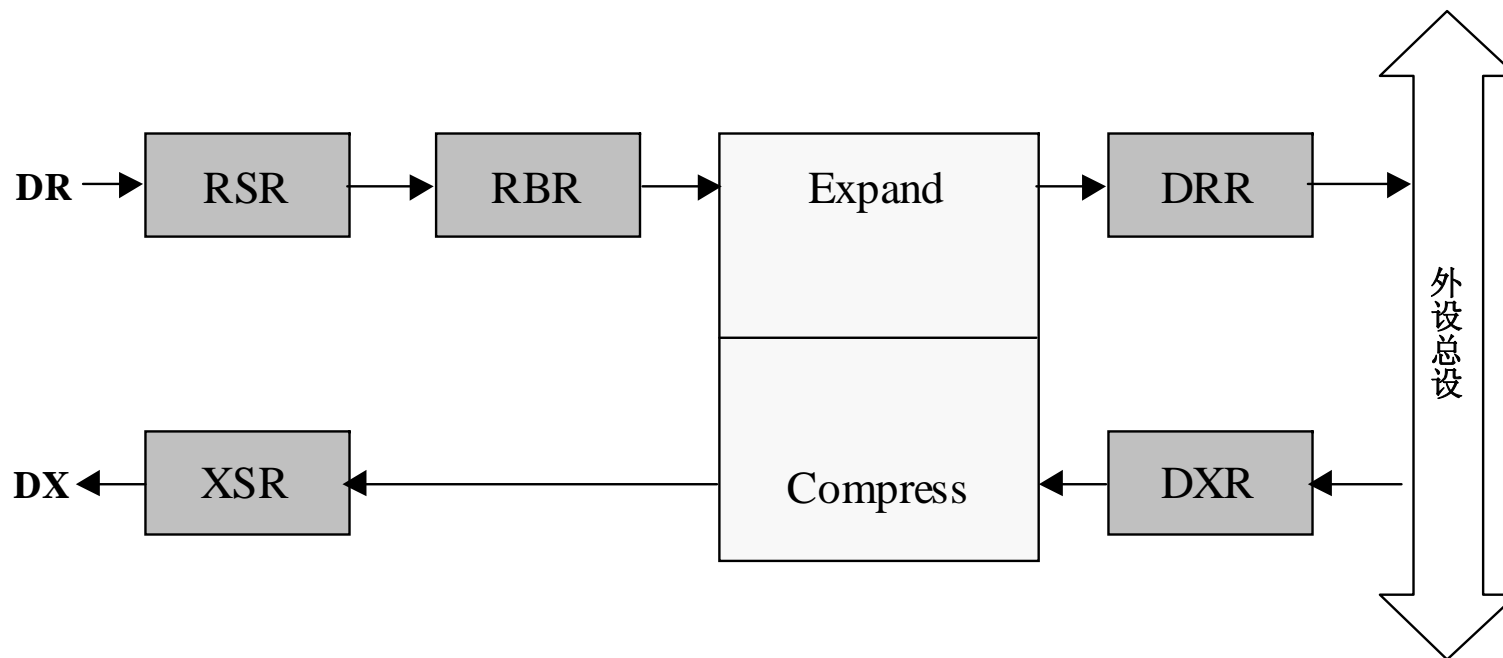


概述—寄存器

地址 (HEX)			缩写	McBSP 寄存器名
McBSP0	McBSP1	McBSP2 ^[注]		
-	-	-	RBR	接收缓冲寄存器
-	-	-	RSR	接收移位寄存器
-	-	-	XSR	发送移位寄存器
018C 0000	0190 0000	01A4 0000	DRR	接收数据寄存器
018C 0004	0190 0004	01A4 0004	DXR	发送数据寄存器
018C 0008	0190 0008	01A4 0008	SPCR	串口控制寄存器
018C 000C	0190 000C	01A4 000C	RCR	接收控制寄存器
018C 0010	0190 0010	01A4 0010	XCR	发送控制寄存器
018C 0014	0190 0014	01A4 0014	SRGR	采样率发生器寄存器
018C 0018	0190 0018	01A4 0018	MCR	多通道控制寄存器
018C 001C	0190 001C	01A4 001C	RCER	接收通道使能寄存器
018C 0020	0190 0020	01A4 0020	XCER	发送通道使能寄存器
018C 0024	0190 0024	01A4 0024	PCR	管脚控制寄存器



传输—缓冲收发



- McBSP的接收操作采取三级缓冲方式
- 发送操作采取两级缓冲方式



传输—初始化

④ 决定与串口交换数据的方式

- CPU
- DMA

④ 串口的初始化

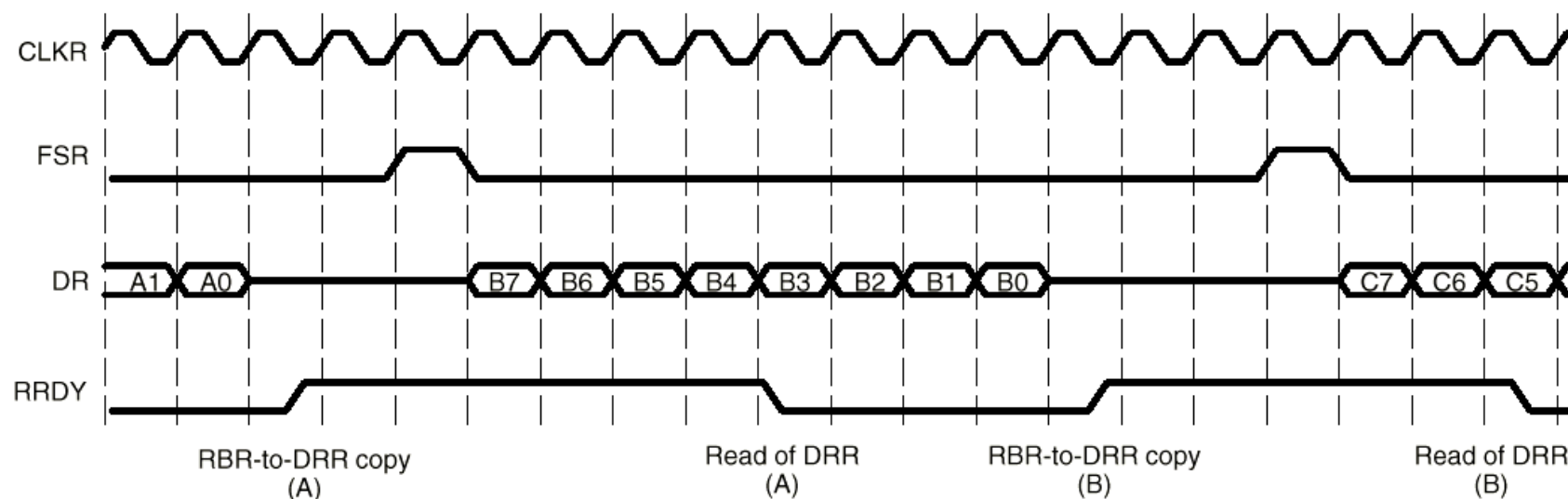
- 串口复位
- 设置寄存器
 - 串口控制寄存器(SPCR) /管脚控制寄存器(PCR)/接收控制寄存器(RCR)/采样率发生器寄存器(SRGR)
 - 时钟和帧信号的来源/频率/极性/信号参数、每帧的数据个数、数据字长、管脚工作模式、中断、同步事件

④ 启动

- 使收/发通道退出复位态，等候帧同步信号
- 使帧信号主控端退出复位态



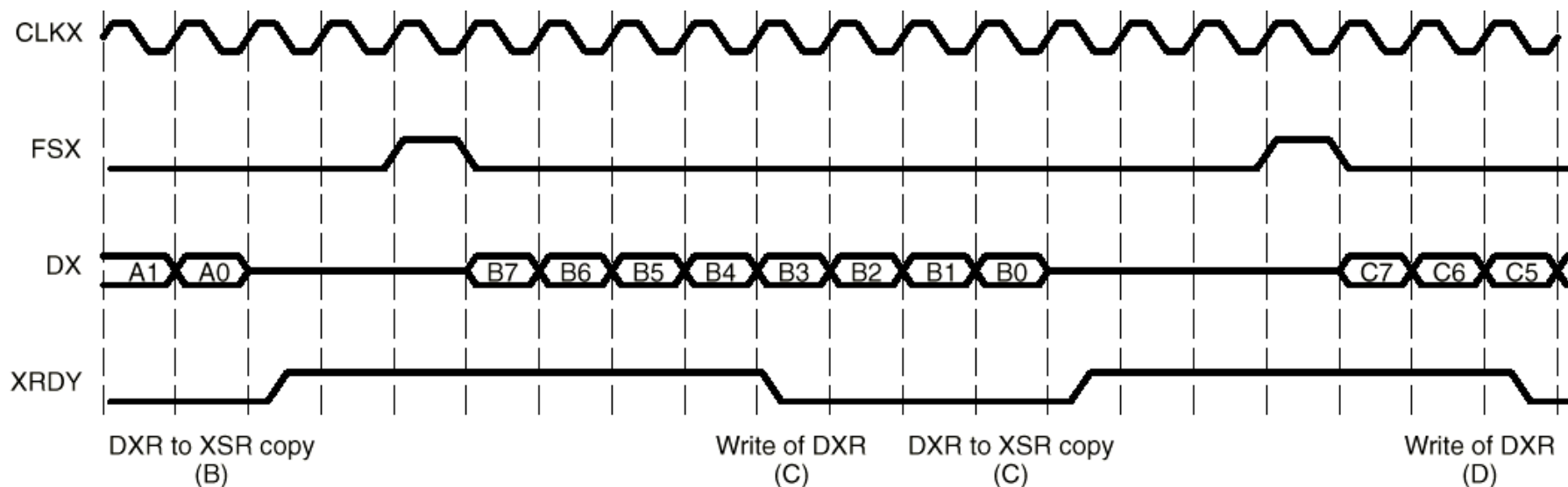
传输—接收



- 帧同步信号FSR激活数据的接收操作
- SPCR寄存器中的RRDY位标示接受状态
- RRDY=1表示数据接收寄存器(DRR)已准备好
- 当数据被读走后，RRDY变成无效=0



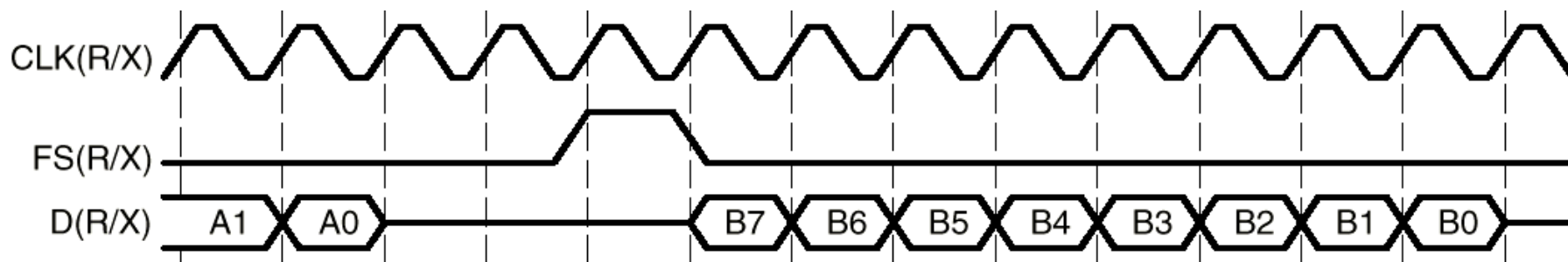
传输—发送



- 帧同步信号FSX激活数据移位输出
- SPCR寄存器中的XRDY位标示发送状态
- XRDY=1表示数据发送寄存器(DXR)已空
- 当数据被写入DXR后, XRDY变成无效=0



传输—设置项

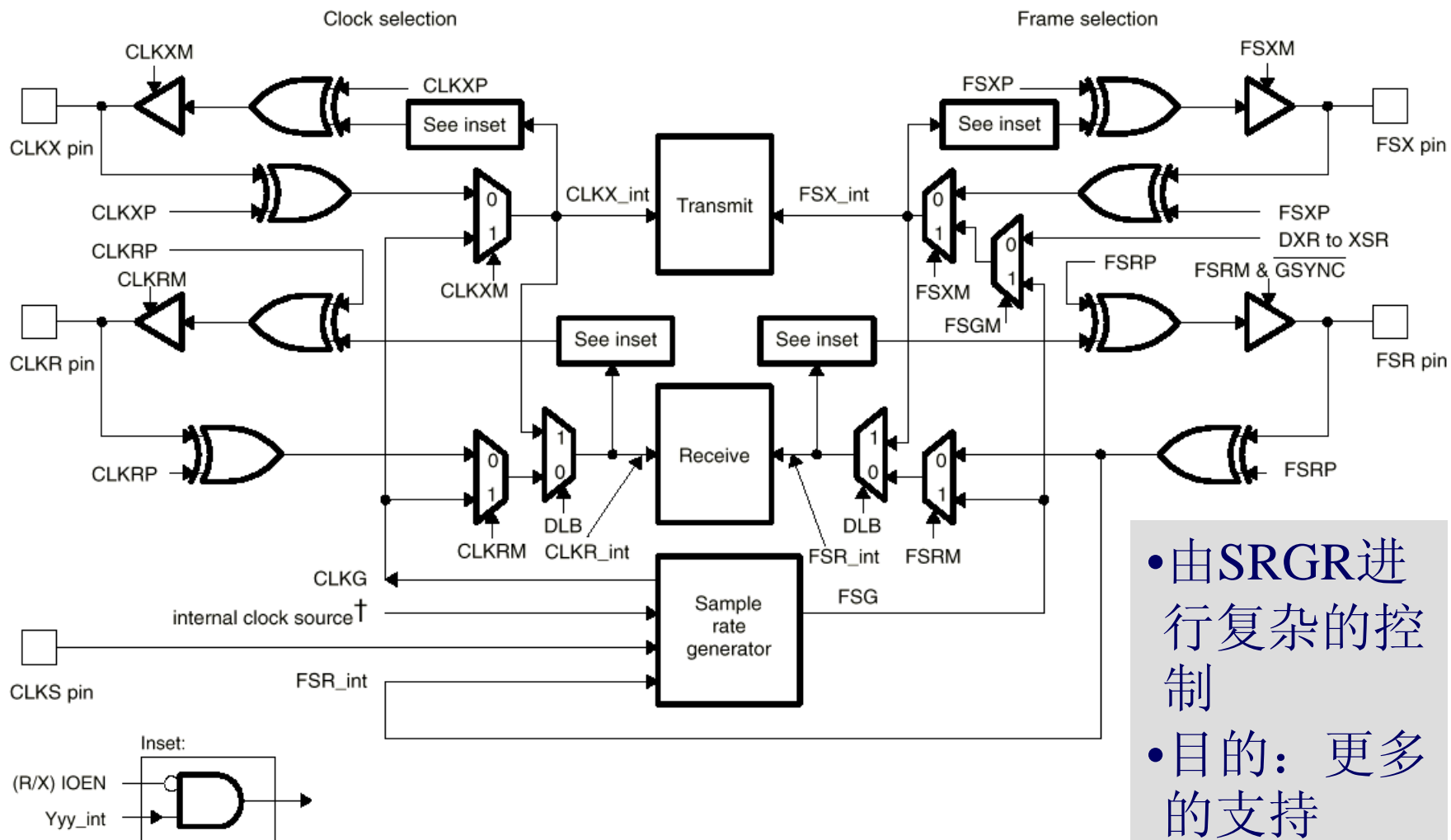


典型传输时序

- 同步时钟的来源/产生方法/周期
- 帧信号的格式（周期，脉宽）
- 信号的极性
- 帧信号与数据的出现关系
- 每帧传输的数据的结构（phase）
- 每帧的数据个数
- 数据字长



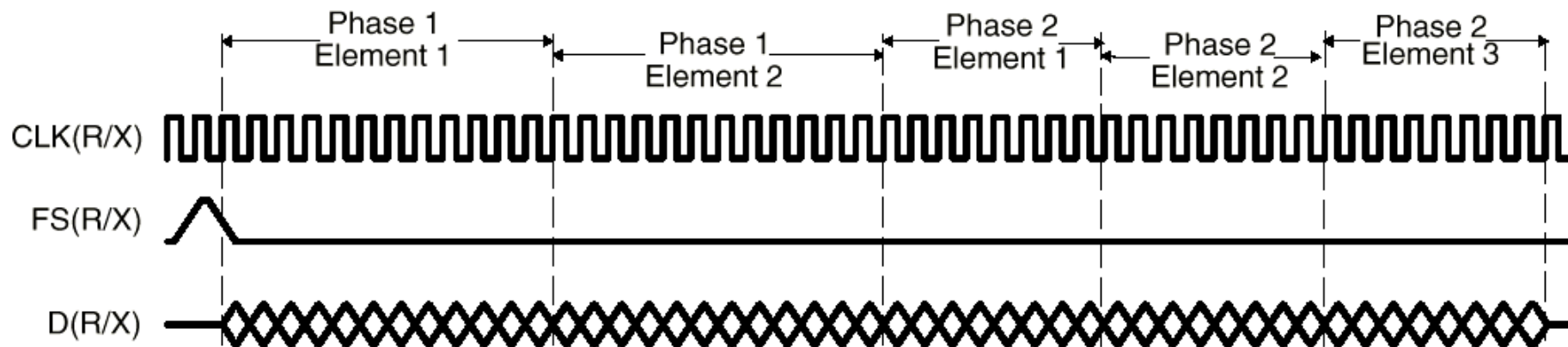
传输—时钟与帧信号



- 由SRGR进行复杂的控制
- 目的：更多的支持



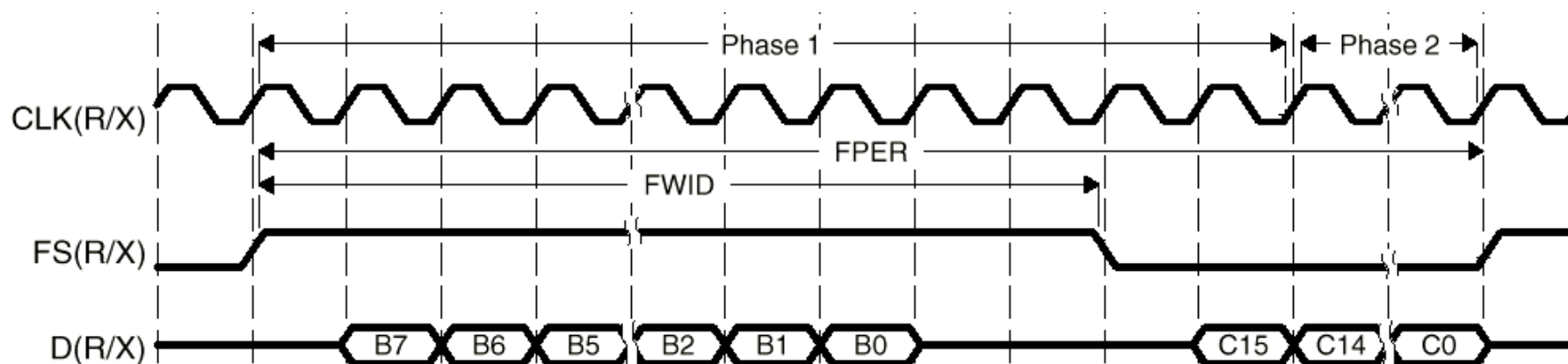
传输—例子1



- 每帧包含两个相(Phase)
- phase 1中包含两个12-bit的数据单元
- phase 2中包含三个8-bit的数据单元
- 一帧中的整个数据流是连续的，数据单元以及相之间没有传输的间隔。



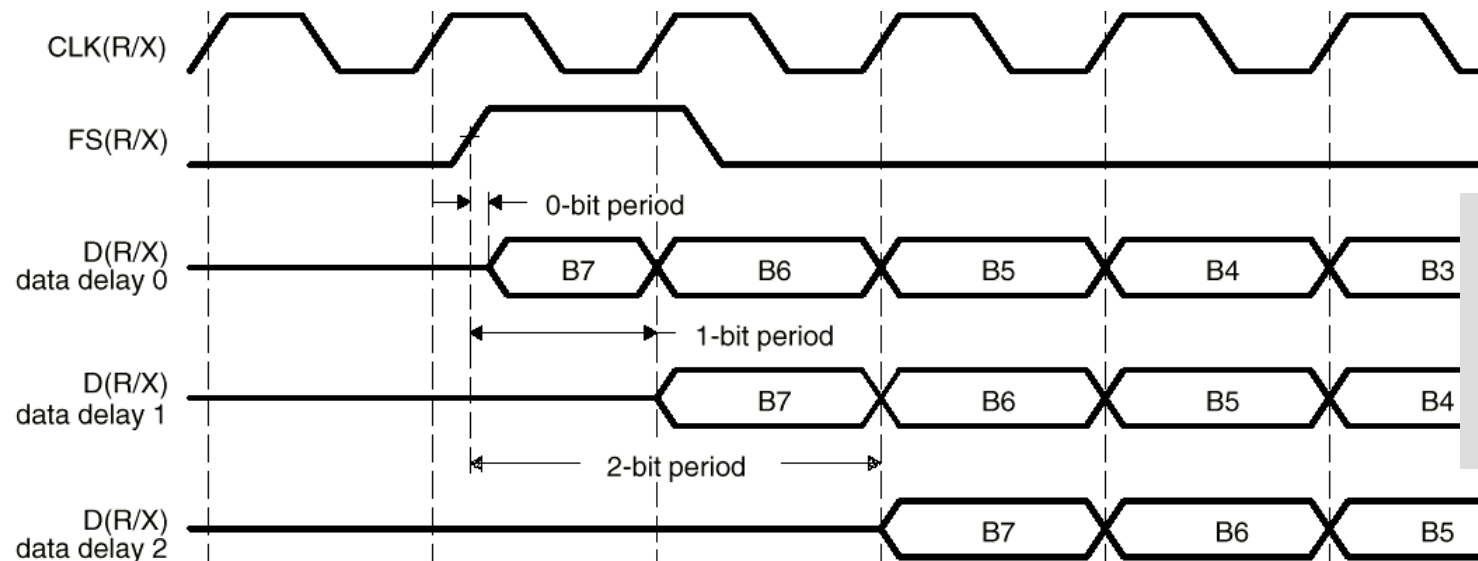
传输—例子2



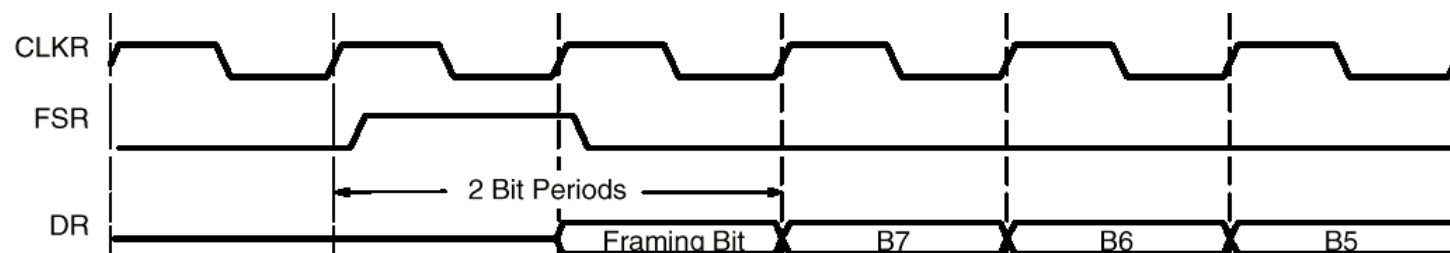
- 每帧包含两个相(Phase)
- phase 1中包含1个8-bit的数据单元
- phase 2中包含1个16-bit的数据单元
- phase 2的起始位置可以定义
- FWID位决定了phase 1的持续时间
- FPER域决定了两相总共的帧周期
- 在phase 1和phase 2之间存在空闲时间（dead time）



传输—例子3



- 数据延迟
- 范围：0~2个周期
- 给用户提供了更大的灵活性

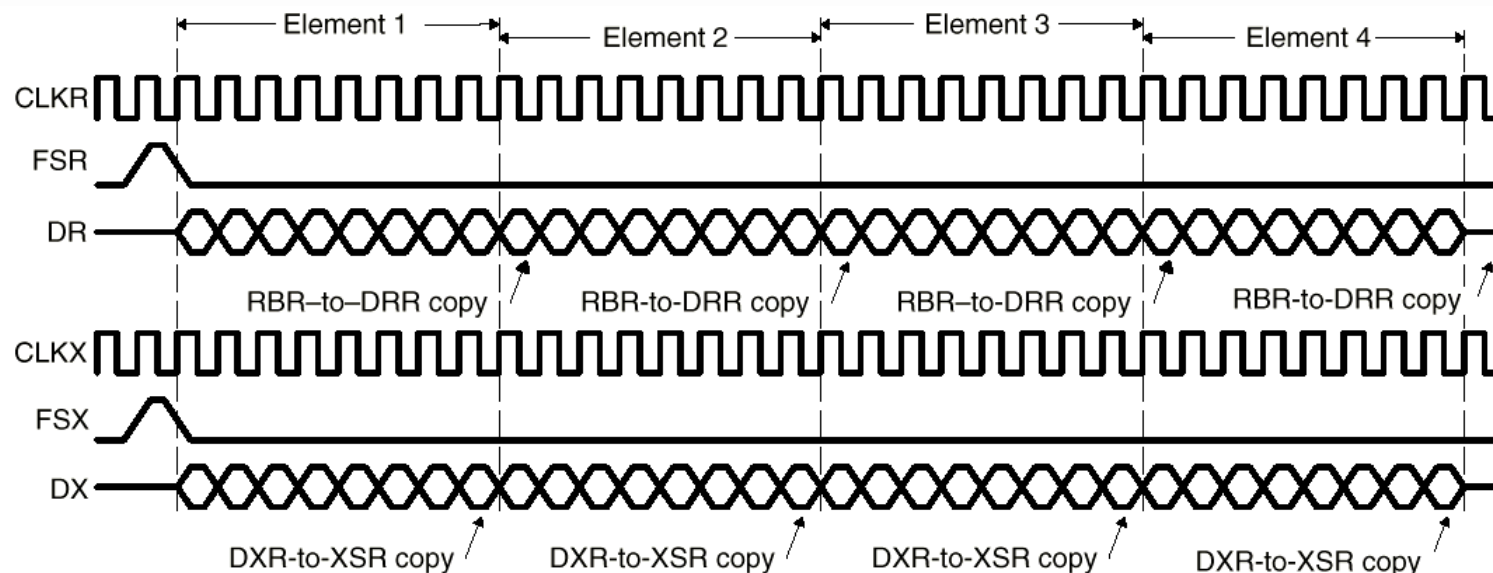


- 忽略帧位数据

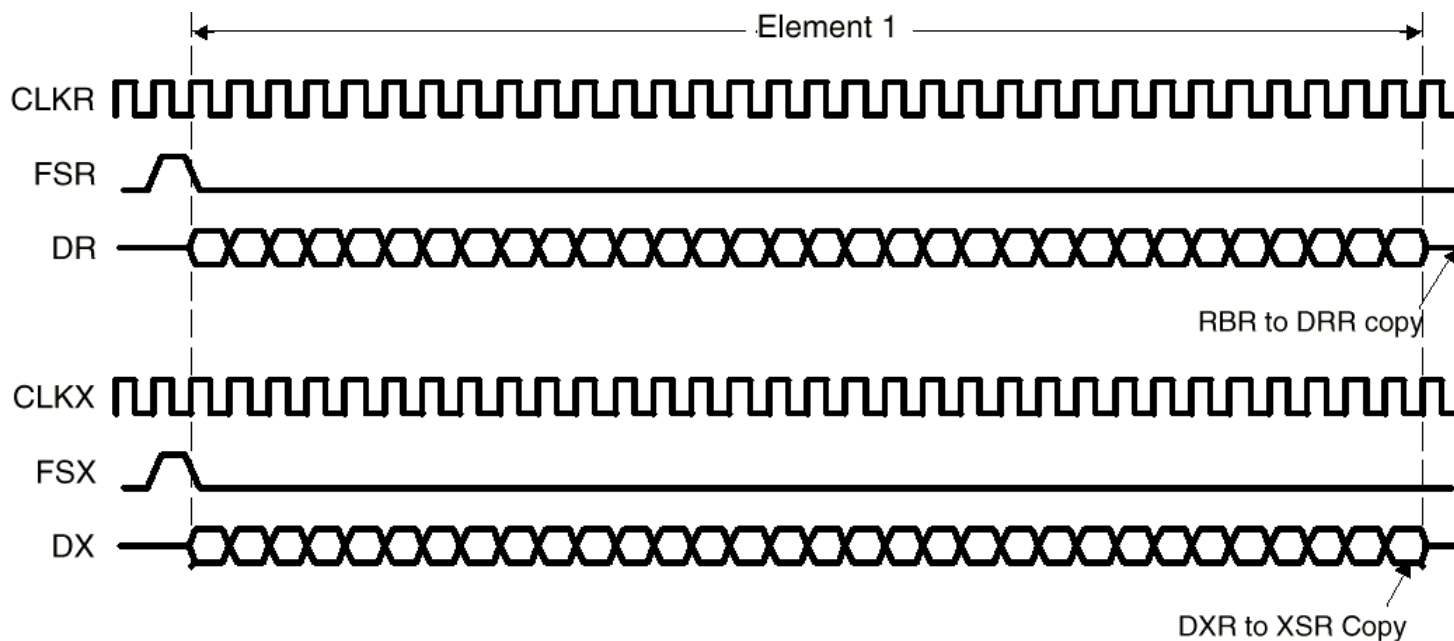


传输—例子4

4个8-bit数据
(R/X)PHASE=0
(R/X)FRLLEN1=3h
(R/X)WDLEN1=0

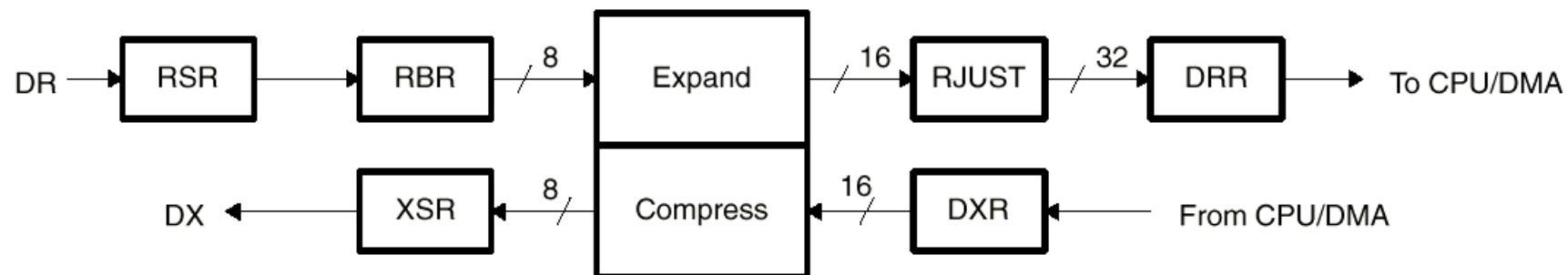


改为1个32-bit数据
(R/X)PHASE=0
(R/X)FRLLEN1=0
(R/X)WDLEN1=5h

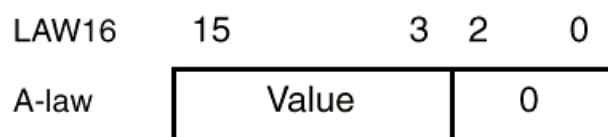
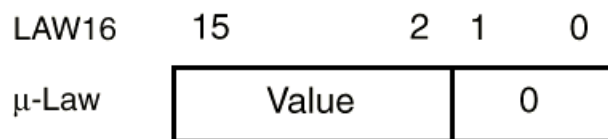




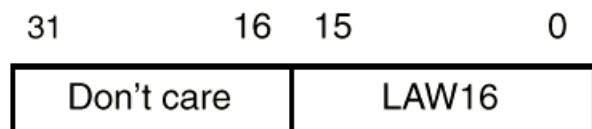
数据压扩



压/扩硬件框图



扩展数据格式



DXR中的数据

- 在(R/X)CR中使能压扩硬件
- 数据从DXR拷贝至XSR的过程中进行压缩
- 从RBR拷至DRR时被扩展
- 压扩数据总是8位
- 扩展数据LAW16内部左对齐



多通道—概述

- ④ 多通道是C6000串口一个非常强的功能
- ④ 单相位 (single-phase) 模式下支持多通道操作
- ④ 多通道
 - 一帧数据传输——一组时分复用数据流
 - 每帧的数据单元个数代表了传输通道的个数
 - 一帧最多可以有128个通道 (7-bit FRLLEN1)
 - 发送和接收可以独立地选择其中某一个或某一些通道中传输数据单元
 - 一次可以选择32个通道被使能发送或接收



控制寄存器

- 多通道控制寄存器 (MCR)
- 发送通道使能寄存器 (XCER)
- 接收通道使能寄存器 (RCER)



选择收/发通道

- 由MCR和(R/X) CER共同决定通道的选择
- 使能子帧
- 选择子帧中的数据通道



多通道—子帧与通道

Subframe #	0	1	2	3	4	5	6	7	0
(R/X)PABLK Partition A elements	0 0-15		1 32-47		2 64-79		3 96-111		0 0-15
(R/X)PBBLK Partition B elements		0 16-31		1 48-63		2 80-95		3 112-127	
FS(R/X)									

- 128个数据通道分为8个子帧
- 8个子帧归为两组A/B（乒乓控制）
- 数据通道的选择：使能使能子帧
+选择子帧中的单元

④ 通道的禁止

- 接收
 - 该数据不会执行RBR-DRR拷贝
 - 该数据不会产生RRDY
- 发送
 - DX高阻
 - 不会执行DXR-XSR拷贝

④ 通道的使能

- 被使能的通道数据正常的收发操作

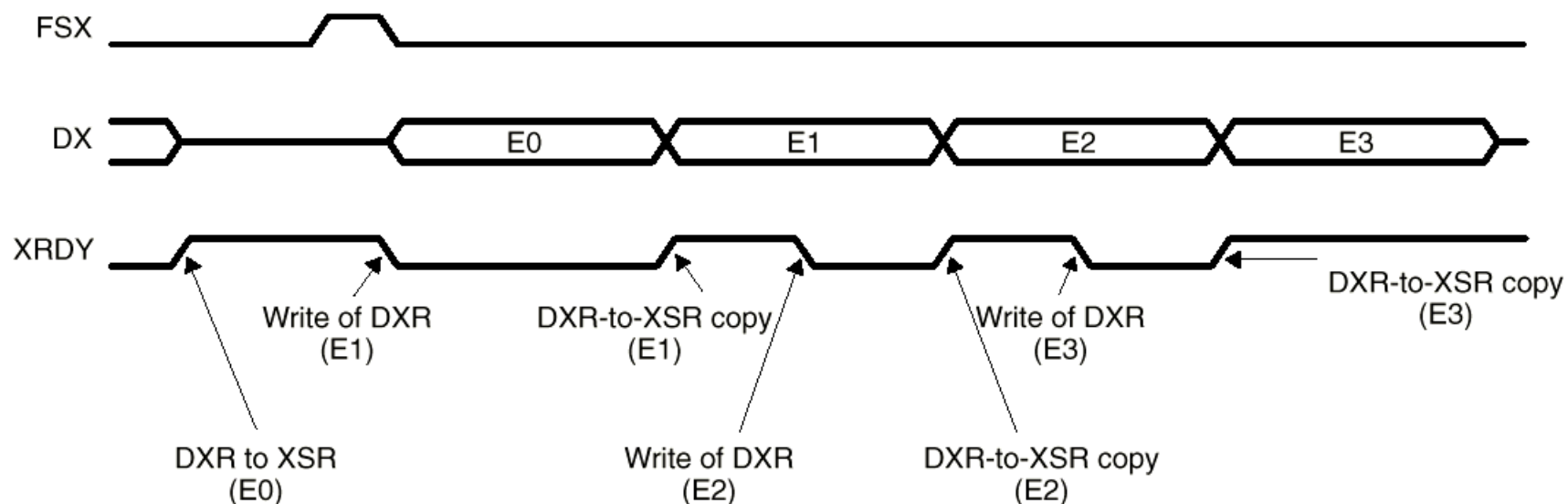
④ 发送的屏蔽

- DX保持高阻，即便通道被使能



多通道—选择

(a) XMCM = 00b



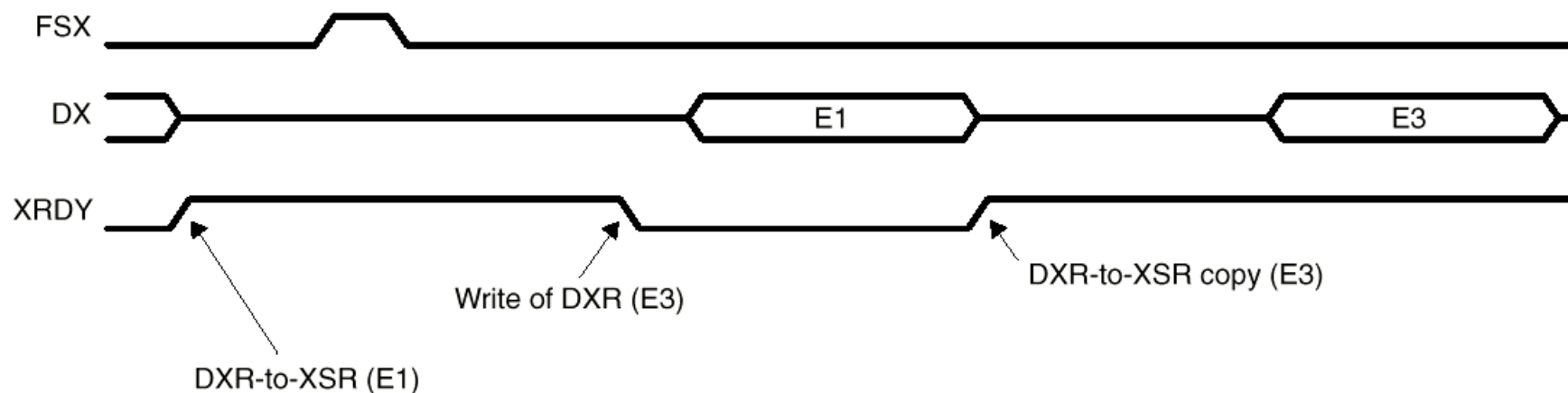
使能所有的数据单元，没有屏蔽

- 都会执行DXR-XSR
- 都被输出



多通道—选择

(b) $XMCM = 01b$, $XPABLK = 00b$, $XCER = 1010b$



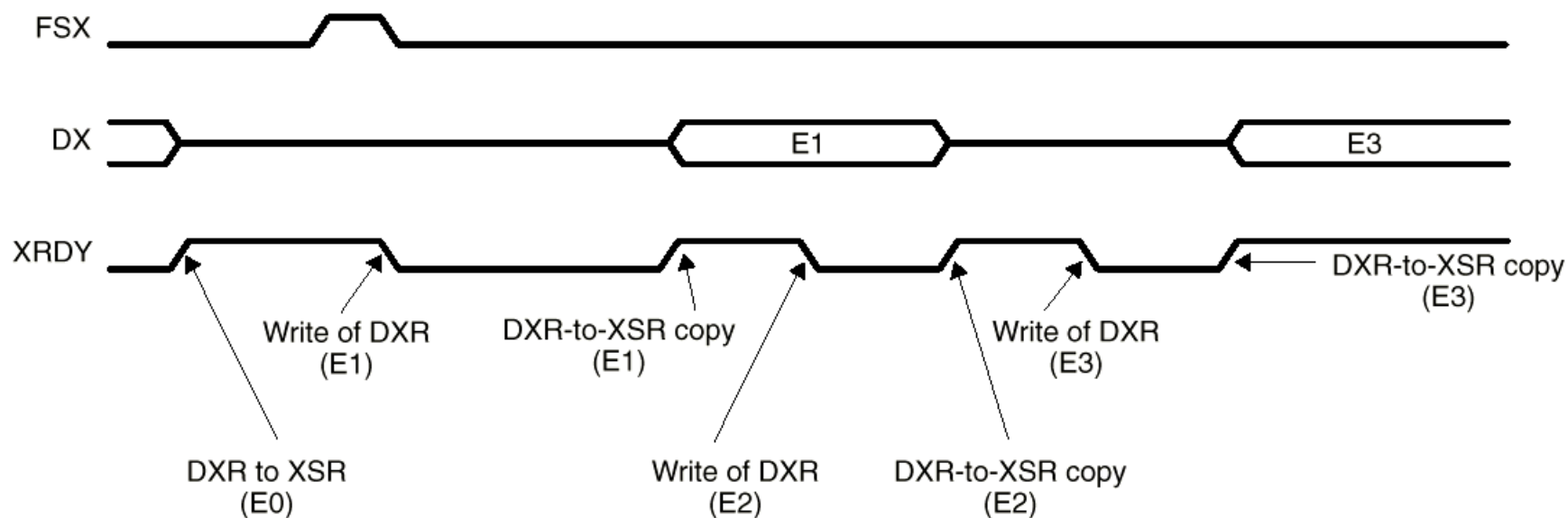
禁止/屏蔽所有单元，选择0号子帧中的通道1和3发送

- 被选择者执行DXR-XSR
- 选择者被输出



多通道—选择

(c) $XMCM = 10b$, $XPABLK = 00b$, $XCER = 1010b$



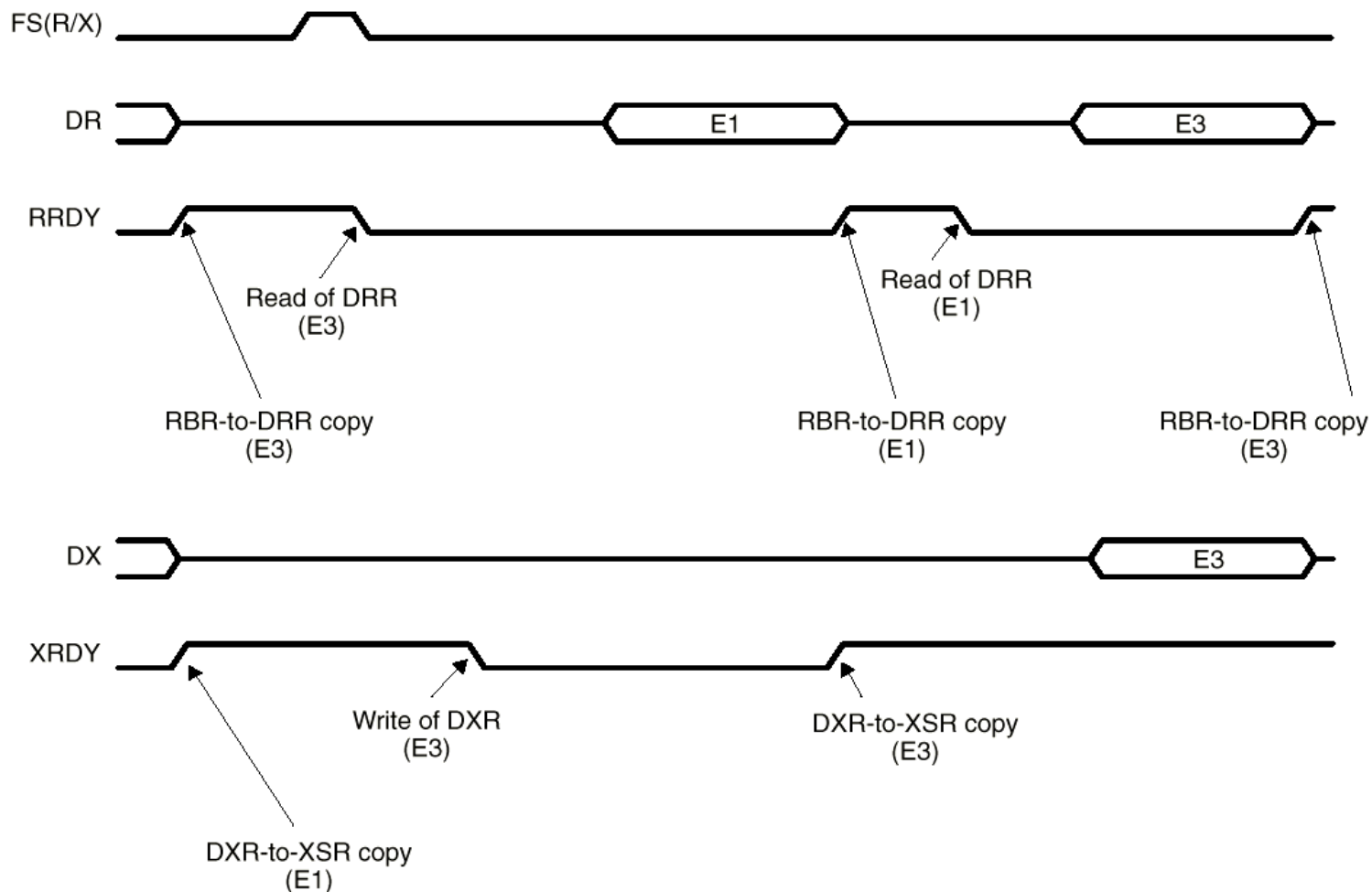
使能/屏蔽所有单元，选择0号子帧中的通道1和3

- 所有通道都执行DXR-XSR
- 只有选择的发送通道被输出



多通道—选择

(d) $XMCM = 11b$, $RPABLK = 00b$, $XPABLK = X$, $RCER = 1010b$, $XCER = 1000b$

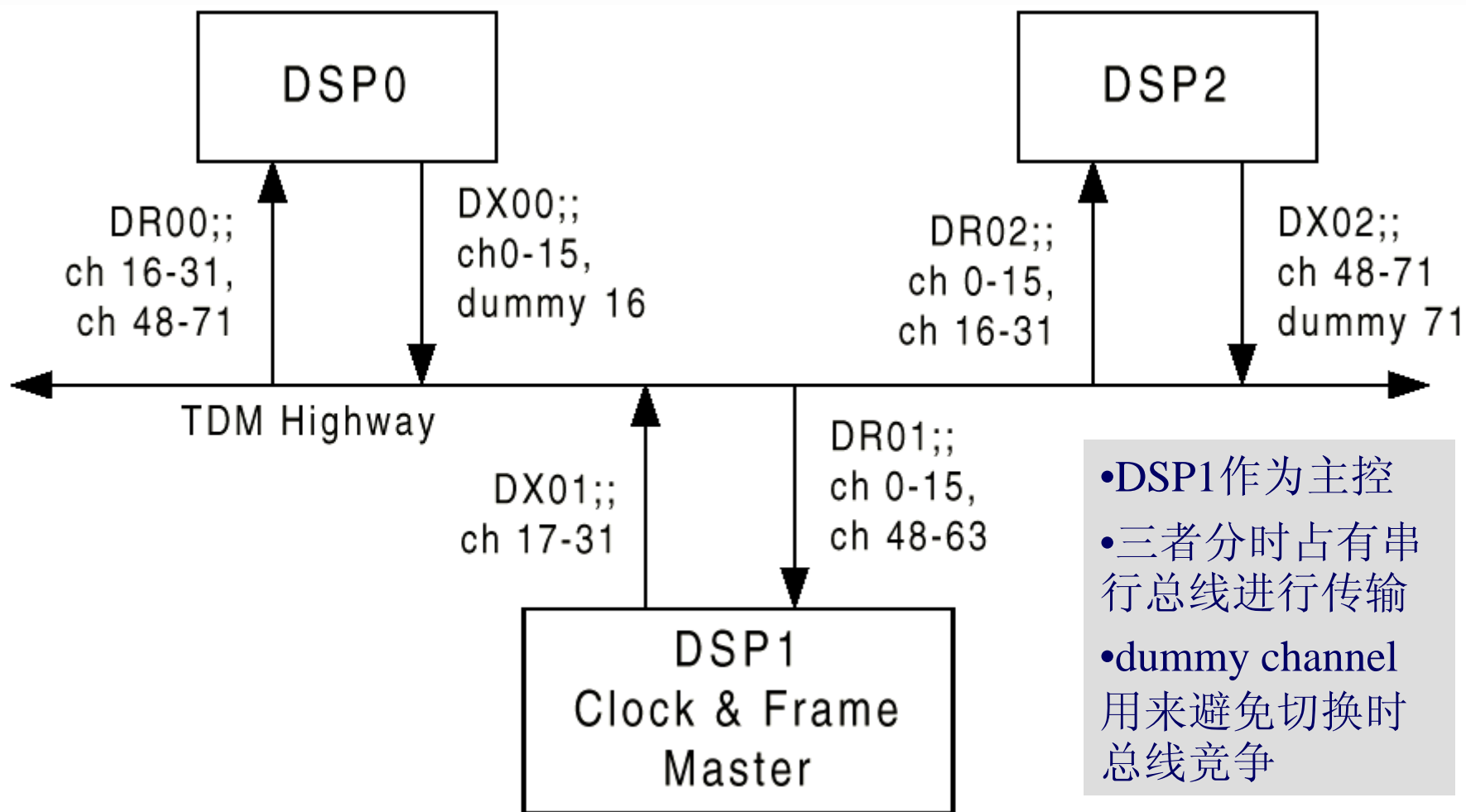


禁止/屏蔽所有单元，选择了0子帧的通道1/3(收)和通道3(发)

- 选择的发送通道被输出
- 屏蔽不影响接收



多通道—例子



- DSP1作为主控
- 三者分时占有串行总线进行传输
- dummy channel 用来避免切换时总线竞争

TDM总线上的多个McBSP

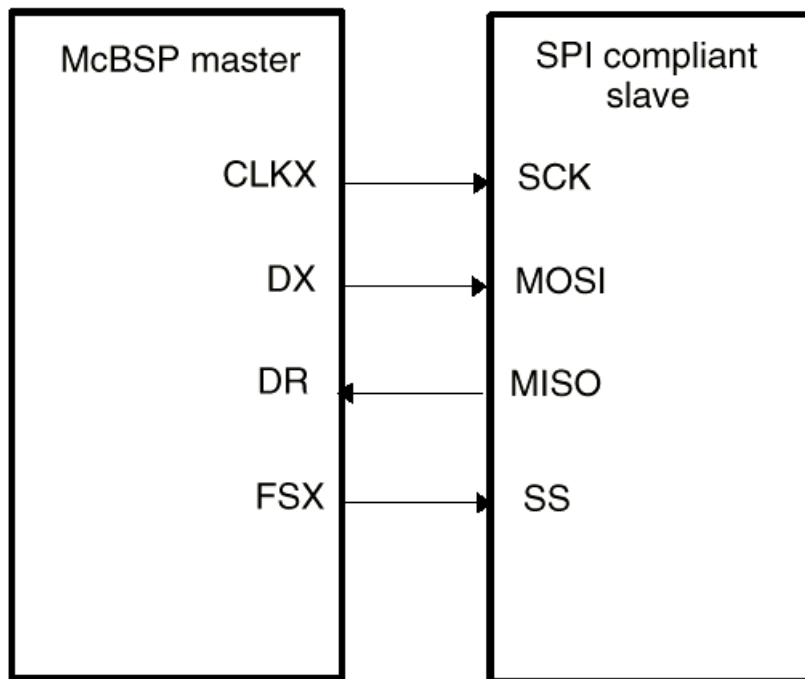


SPI—概念

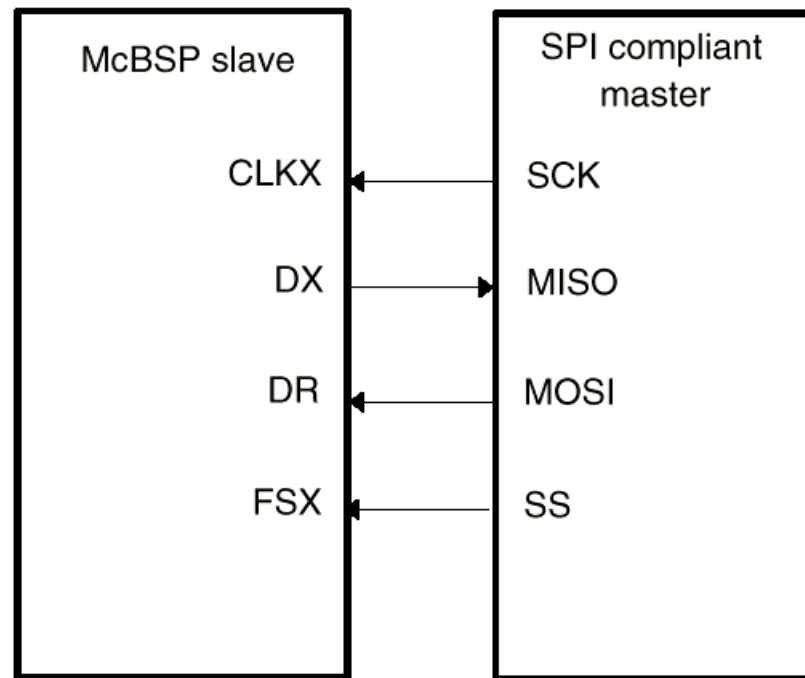
- ④ SPI的定义—— Series Protocol Interface
- ④ SPI的信号—— 4线串行接口
- ④ SPI的模式—— 主模式/从模式
- ④ SPI的特点—— 由主设备时钟信号的出现与否来界定主/从设备间的通信
- ④ C6000对SPI的支持表现在CLKSTP控制
 - 数据时钟的停止控制



SPI—接口



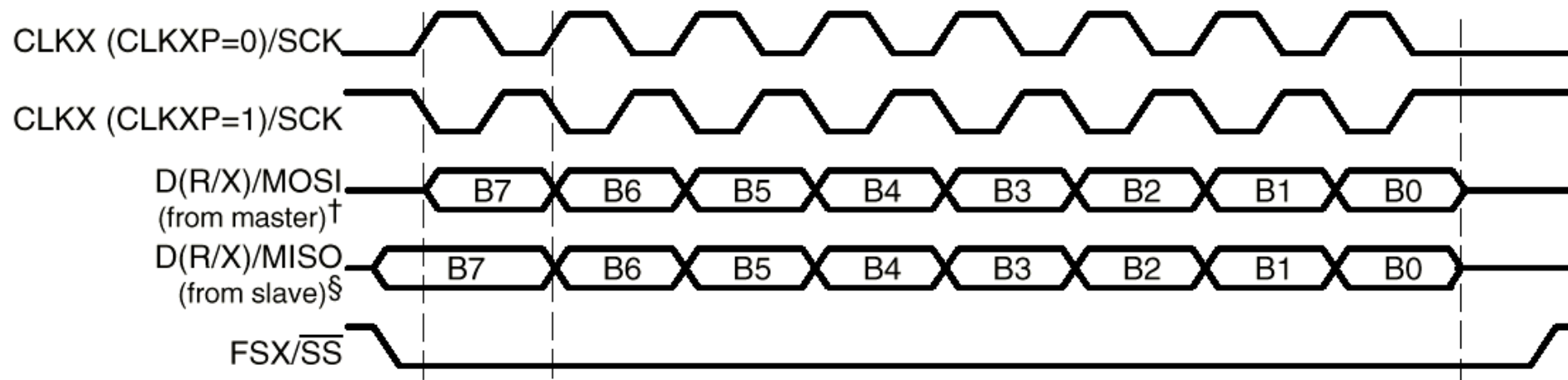
McBSP作主控：
提供时钟和设备使能



McBSP为从模式：
接收时钟和设备使能



SPI—控制



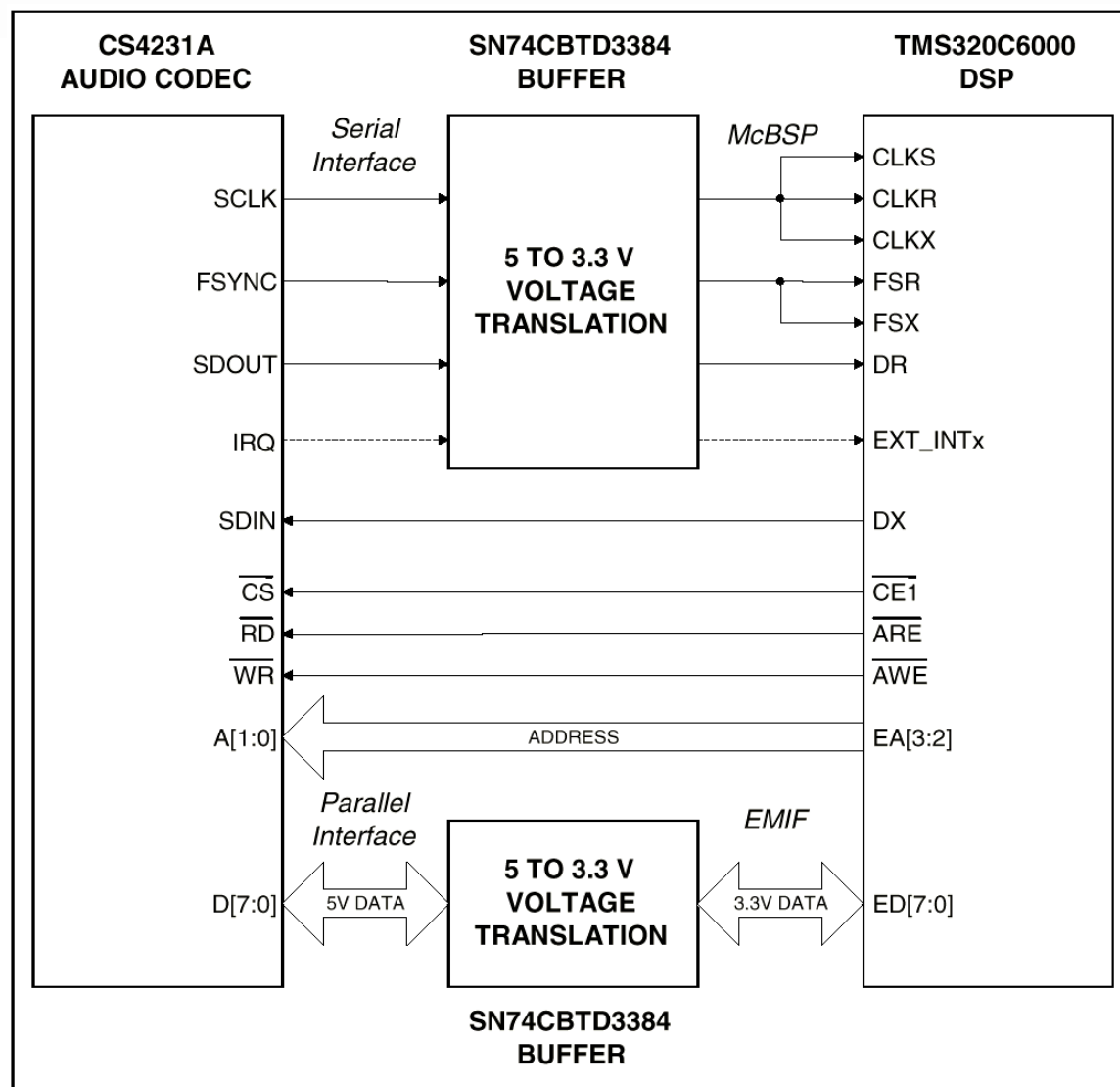
CLKSTP=10b时的SPI 传输

McBSP可设置参数:

- 时钟无效期间的电平
- 收发数据的边沿
- 有无延迟

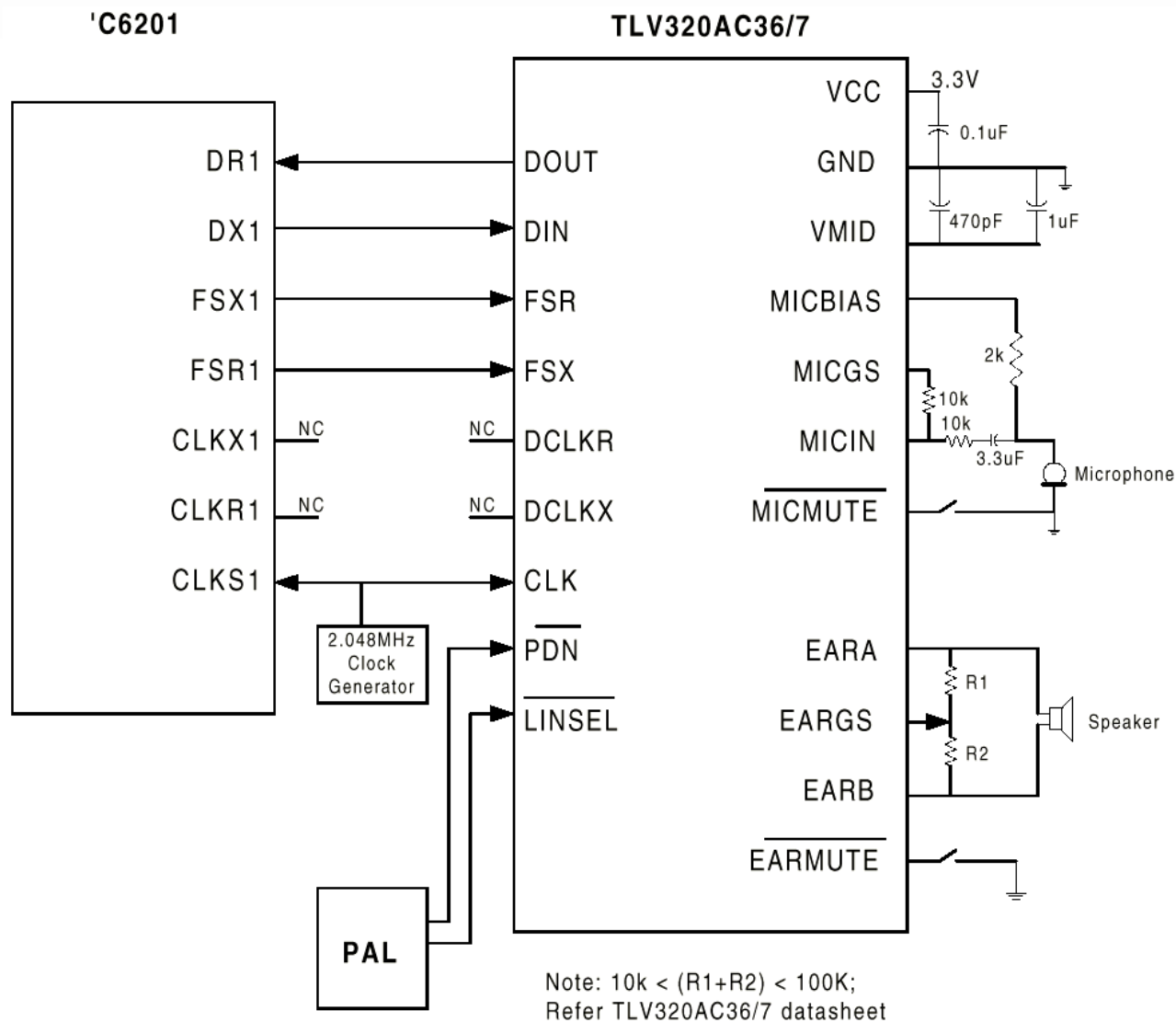


接口实例—AIC





例子实例—VBAP





例子实例—VBAP

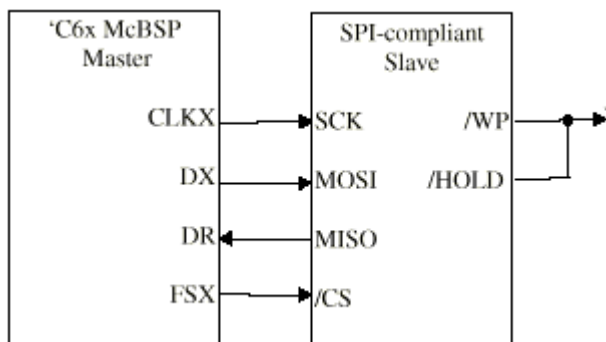
寄存器设置

Register [bit-field #]	Bit-field Name	Value	Description
RCR[20:19]	RCOMPAND	10b	Receive μ -law companding
RCR[17:16]	RDATDLY	01b	1 bit-clock data delay
XCR[20:19]	XCOMPAND	10b	Transmit μ -law companding
XCR[17:16]	XDATDLY	01b	1 bit-clock data delay
SRGR[28]	FSGM	1	FSX generated by FSG
SRGR[27:16]	FPER	0xFF	Frame period of 256 2.048Mhz clock periods to get 8KHz frame sync sampling rate
SRGR[15:8]	FWID	0	Generates 1 clock period active-high pulse.
SRGR[7:0]	CLKGDV	0	2.048 CLKS drives CLKR/X with no divide-down
PCR[11]	FSXM	1	FSX is an output
PCR[10]	FSRM	1	FSR is an output
PCR[9]	CLKXM	1	CLKX is an output generated by CLKS
PCR[8]	CLKRM	1	CLKR is an output generated by CLKS
SPCR[31:0]	all	default	Reset bits will be driven as per initialization procedure



例子实例—SPI ROM

Register	Value in hex	Description
RCR	0x00010000	single phase, one 8-bit element per frame, one bit-clock delay
XCR	0x00010000	single phase, one 8-bit element per frame, one bit-clock delay
SRGR	0x2000005F	<ul style="list-style-type: none">serial clock CLKX generated by CPU clock (CLKSM=1)frame sync FSX generated due to DXR-to-XSR transfer (FSGM=0)clock divide down is 95 for 200 MHz clock to generate 2.08 MHz shift clock (CLKGDV=0x5F)
PCR	0x00000A0C	<ul style="list-style-type: none">FSX is an active low (FSXP=1) output (FSXM=1)FSR is an active low (FSRP=1) input (FSRM=0)CLKX is an output (CLKXM=1) and starts with a rising edge (CLKXP=0)
SPCR[12:10]	0x3	CLKSTP=11b. Since CLKXP=0, this refers to data transmitted on rising edge and received on falling edge of CLKX by the master. This parameter can be changed as per application needs.



McBSP 控制寄存器设置 @ 200MHz CPU clock

- ④ 串口传输过程
- ④ 灵活的时钟设置
- ④ 支持多种协议
- ④ 丰富的内置辅助硬件
- ④ 多通道选择传输
- ④ 接口实例



3) 主机并行接口 (HPI)

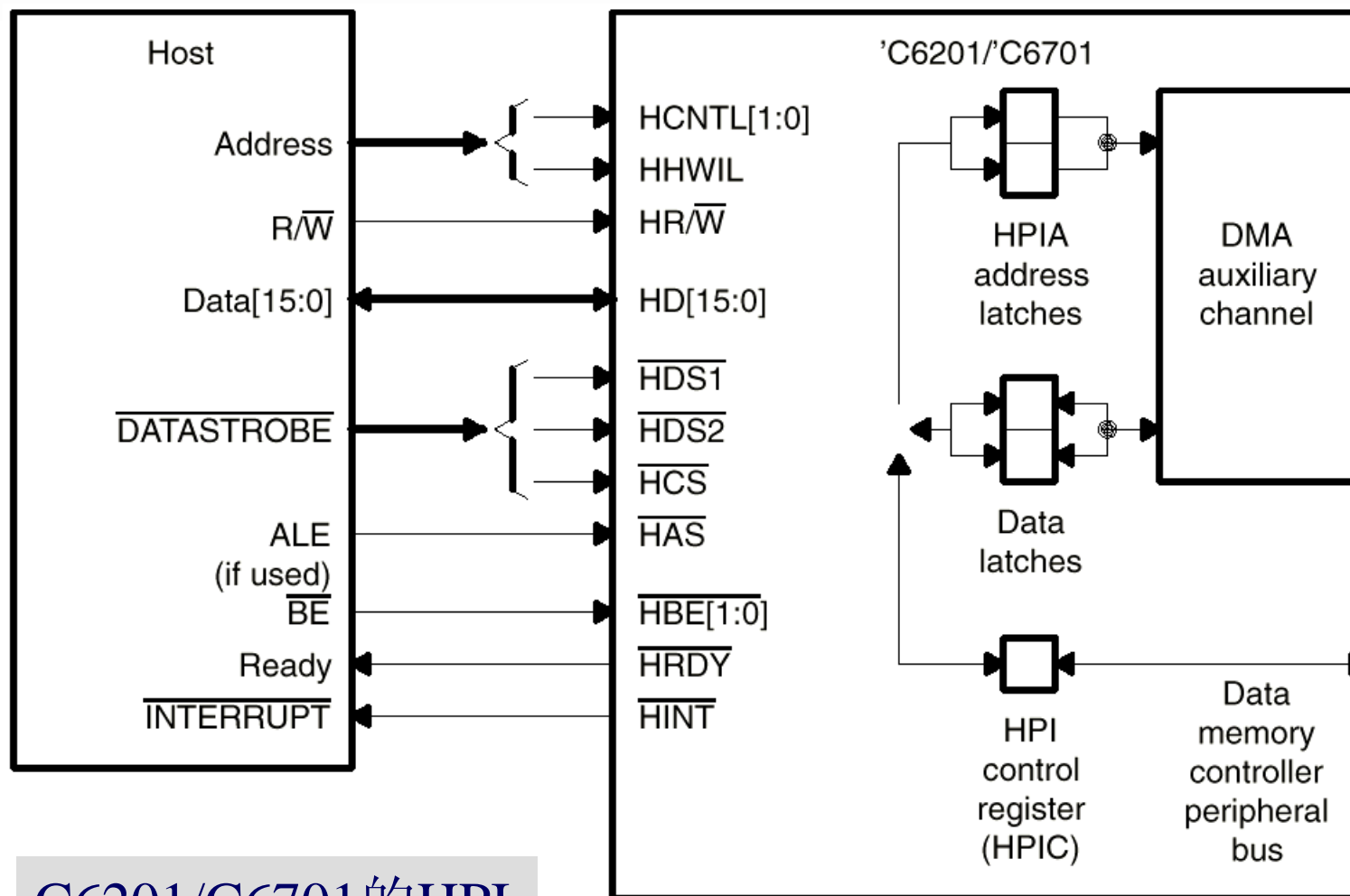
- 主机口 HPI 是一个并行端口。上位机拥有该接口的控制权，通过它可以直接访问 DSP 的存储空间（包括映射的片内外设）。
- HPI 可以通过 DMA/EDMA 控制器实现对 DSP 存储空间的访问。主机和 CPU 都可以访问 HPI 控制寄存器 (HPIC) 和地址寄存器 (HPIA)，主机一方还可以访问 HPI 数据寄存器 (HPID)。

基本简介

- ④ 16-bit宽度的并行端口
- ④ 主机/上位机掌管该接口的主控权
- ④ 通过HPI可以直接访问C6000的存储空间，以及映射的外围设备
- ④ DMA控制器实现HPI与CPU存储空间的互连
- ④ C6211/C6711中的HPI更有所加强
 - 不再借助DMA辅助通道
 - 硬件来处理读/写请求



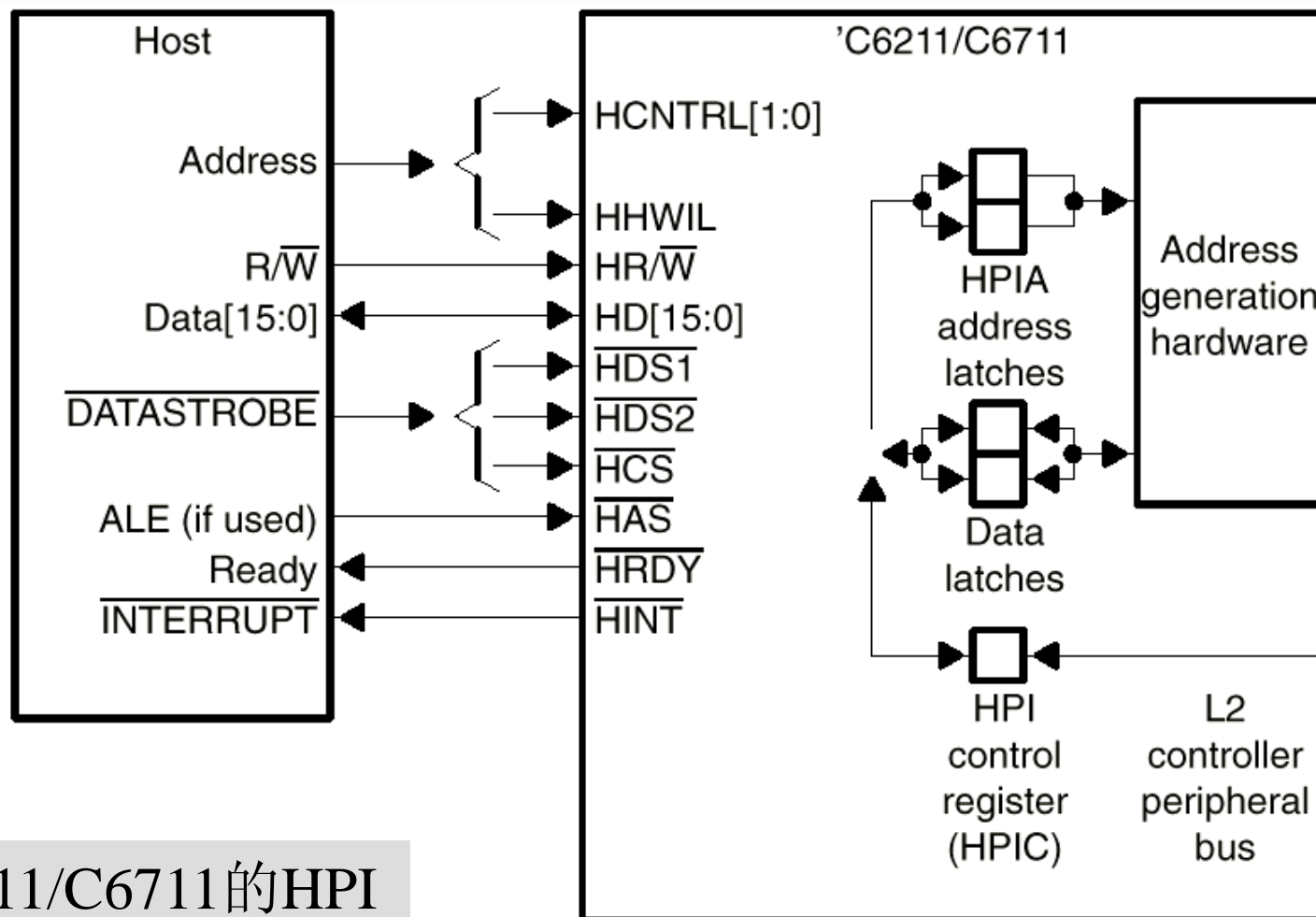
框图



C6201/C6701的HPI



框图

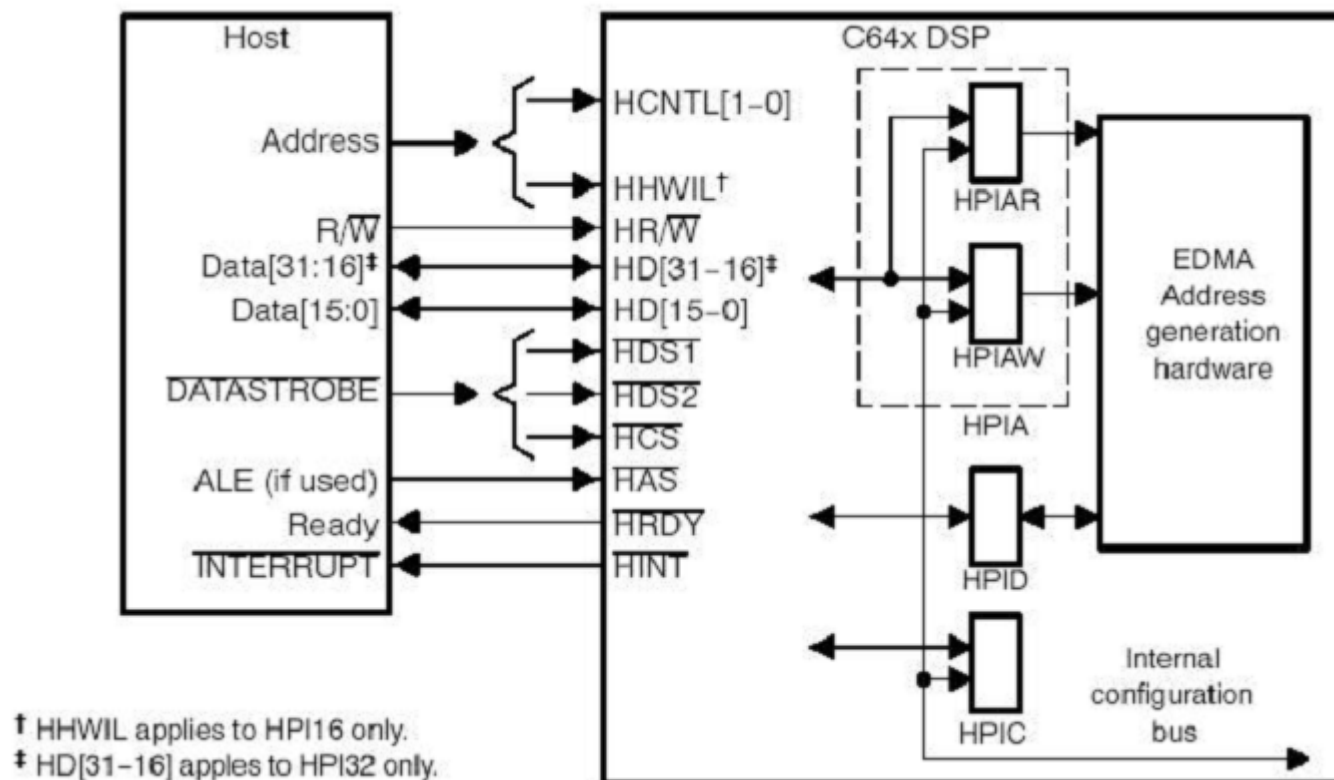


C6211/C6711的HPI



C64x HPI 接口信号

- ④ C64x的 HPI 是 C621x/C671x HPI 的增强版，总线扩展为 32bit，可以设置为 HPI32 和HPI16 这两种模式。C64x的 HPI 接口示意如下图：





- ④ 在控制信号上，为了便于和多种处理器接口，C64x的HPI 接口提供了许多冗余的信号。
- ④ HDx 是数据地址复用的双向总线，传送的数据包括控制寄存器的设置值、初始化的访问地址以及真正的数据。HCNTL[1:0]控制当前访问的是 HPI 的哪一个寄存器。这2 个信号对应的 4 个状态除了区分对 HPIC/HPIA/HPID 寄存器的访问，还提供了一个对 HPID 进行地址自增的访问方式。HHWIL信号的作用是区分第 1 个/第 2个半字的传输。
- ④ 数据选通（/HDS1 和/HDS2）、读写选择和地址选通这几个信号在功能上非常类似，只是在时序上有所差别。这种差别主要是为了兼顾不同工业标准处理器的需要。



C64x HPI 接口信号总结

Signal Name	Signal Type†	Signal Count	Host Connection	Signal Function
HD[15-0] or HD[31-0]‡	I/O/Z	16 or 32‡	Data bus	
HCNTL[1-0]	I	2	Address or control lines	HPI access type control
HHWIL§	I	1	Address or control lines	Halfword identification input
HAS	I	1	Address latch enable (ALE), address strobe, or unused (tied high)	Differentiation between address and data values on multiplexed address/data host
HBE[1-0]¶	I	2	Byte enables	Data write byte enables
HR/W	I	1	Read/write strobe, address line, or multiplexed address/data	Read/write select
HCS	I	1	Address or control lines	Data strobe inputs
HDS[1-2]	I	1 1	Read strobe and write strobe or data strobe	Data strobe inputs
HRDY	O	1	Asynchronous ready	Ready status of current HPI access
HINT	O	1	Host interrupt input	Interrupt signal to host

† I = input, O = output, Z = high impedance

‡ HD[31-16] applies to C64x or HPI32 only.

§ HHWIL does not apply to C64x HPI32.

¶ HBE[1-0] applies to C620x/C670x DSP only.



- ④ HPI 通过 3 个寄存器完成主机与 CPU 的通信：HPI 数据寄存器（HPID）、HPI 地址寄存器（HPIA）和 HPI 控制寄存器（HPIC）。主机对于这 3 个寄存器都可以进行读/写，CPU 只能进行访问。对于 C64x，CPU 和主机都可以访问。
- ④ HPID 用于存放主机存储空间读取的数据，或是主机要向 DSPs 存储空间中写入的数据。HPIA 用于存放当前主机访问 DSPs 存储空间的地址，这是一个 30bit 的值，也就是说一个 32bit 的字，其最低 2 位为 0。
- ④ HPIC 寄存器字长为 32bit，但是高 16bit 和低 16bit 对应同一个物理存储区，因此高 16bit 和低 16bit 的内容一致。写 HPIC 时，必须保证高低 16bit 的内容一致。



接口信号

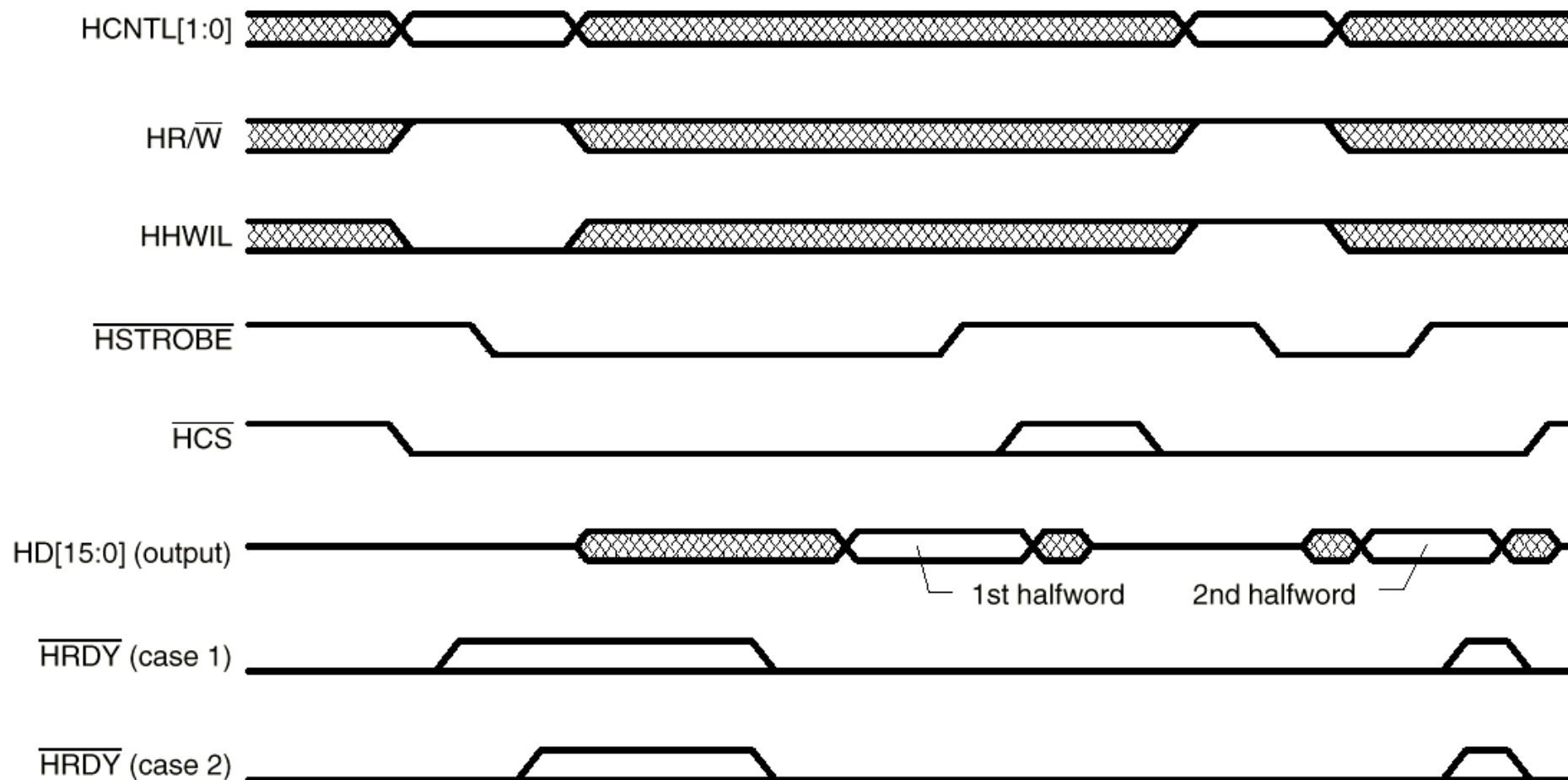
信号	类型	管脚数	主机对应信号	信号功能
HD[15:0]	I/O/Z	16	数据总线	
HCNTL[1:0]	I	2	地址或控制线	HPI 访问类型控制
HHWIL	I	1	地址或控制线	确认半字(16-bit)输入
#HAS	I	1	地址锁存使能 (ALE), 地址触发, 或者不用	对复用地址/数据总线的主机, 区分地址和数据
#HBE[1:0]	I	2	字节使能	写数据字节使能
HR/#W	I	1	读/写触发	读/写选择
#HCS	I	1	地址或控制线	输入数据选通
#HDS[1:2]	I	1	读触发, 写触发 数据触发	输入数据选通
#HRDY	O	1	异步 ready 信号	当前访问 HPI 状态准备好
#HINT	O	1	主机中断输入	向主机发出的中断信号



- ④ 主机对HPI的访问操作
 - 初始化HPIC寄存器
 - 初始化HPIA寄存器
 - 从HPID寄存器读取/写入数据
- ④ 每次数据访问都需要进两次存取
 - C6000芯片结构决定了与主机间所有的数据交换都是32-bit
- ④ HPI的四种数据传输模式
 - 不带地址自增的读操作
 - 带地址自增的读操作
 - 不带地址自增的写操作
 - 带地址自增的写操作



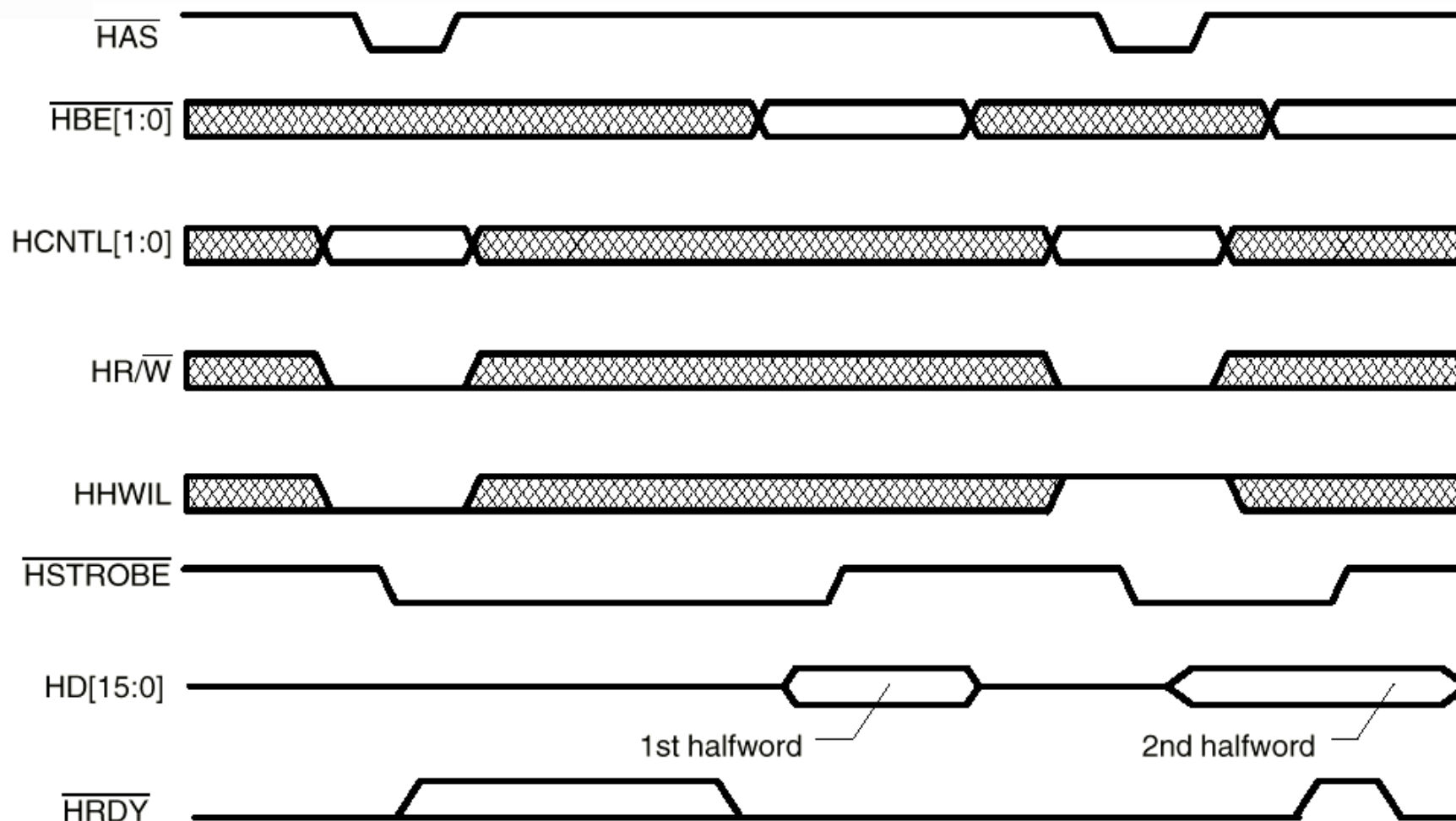
时序



读时序(没有用到/HAS信号)



时序



写时序(用到/ HAS 信号)

存取效率总结

读操作时HPI的最大数据传输率
(主机从DSP存储空间中固定地址读取数据)

不带地址自增的 读操作	传输率 (Mbyte/s)		传输率 (Mbit/s)	
	C6201	C6701	C6201	C6701
	200MHz	167MHz	200MHz	167MHz
片内数据存储器	34.7	28.97	277.6	231.79
片内程序存储器	34.7	28.97	277.6	231.79
SDRAM	24.2	20.2	193.6	161.65
SBSRAM (半速)	25.8	21.54	206.4	172.34
SBSRAM (全速)	27.5	22.96	220	183.7
片内外设总线*	38.0	31.73	304	253.84

注：假设主机端驱动信号的速度没问题

*带地址自增时不包括访问外设



存取效率总结

HPI写操作的最大数据传输率

带/不带地址自增 的写操作	传输率 (Mbyte/s)		传输率 (Mbit/s)	
	C6201	C6701	C6201	C6701
	200MHz	167MHz	200MHz	167MHz
片内数据存储器	50	41.75	400	334
片内程序存储器	50	41.75	400	334
SDRAM	38	31.73	304	258.84
SBSRAM (半速)	40	33.4	320	267.2
SBSRAM (全速)	44.4	37.07	355.2	296.59
片内外设总线	50	41.75	400	334

注：假设主机端驱动信号的速度没问题

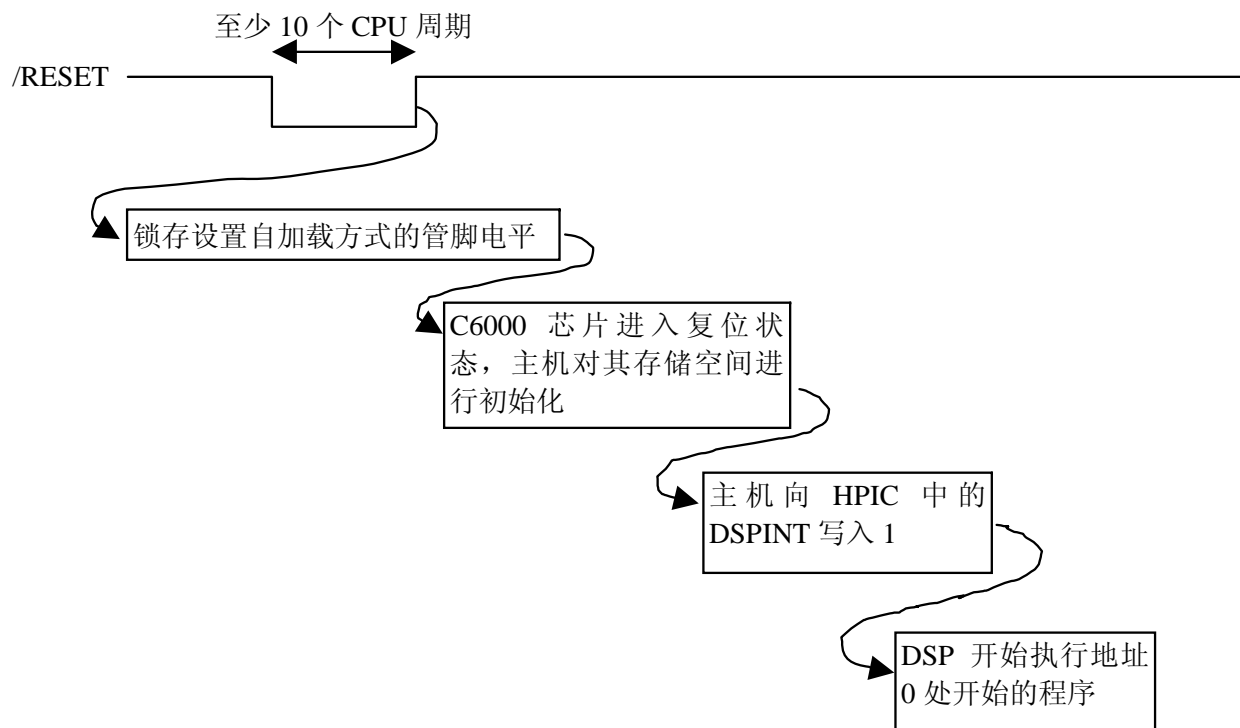


HPI 加载操作

- ④ C6000 DSPs 复位时，如果选择了 HPI Boot 模式，则只有 DSPs 内核进入复位状态，其余模块保持活动状态。这样，主机可以通过 HPI 接口访问 C6000 的整个存储空间，对他们进行初始化。完成有关设置之后，主机向 HPIC寄存器的 DSPINT 位写 1，将 DSPs 从复位状态唤醒，接下来 DSP 将从地址0 开始执行程序。
- ④ 在 HPI boot 模式下，主机对 DSPs 可以进行的操作包括：初始化 CPU和 EMIF 以及向DSPs 加载程序和数据。这些都可以在主机端编程实现。



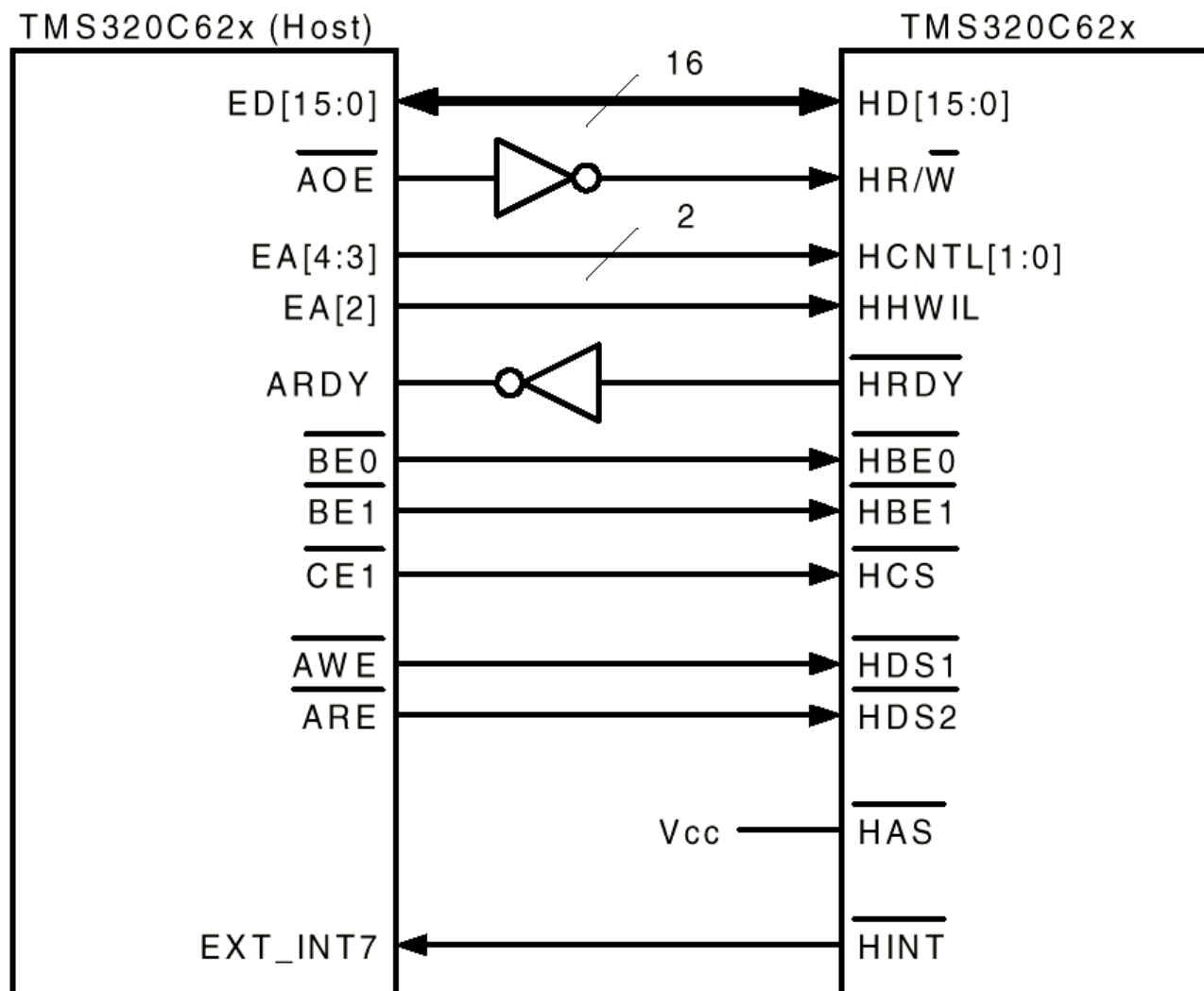
HP I 的自加载



- 复位时，内核复位，外围激活态
- 由主机对DSP进行唤醒



上海交通大学 接口实例— 两个 C6201

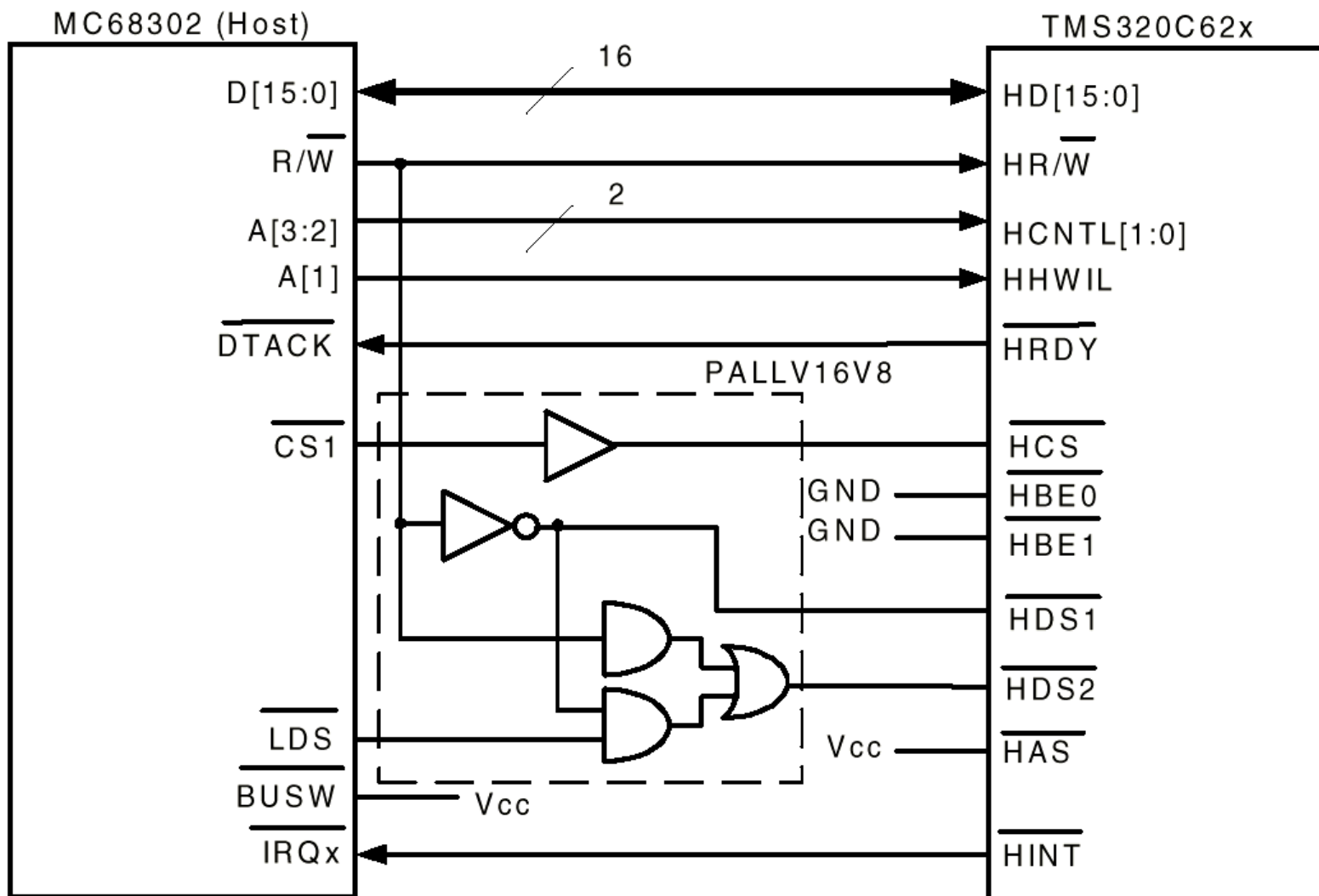


校长办公室

the Office of the President

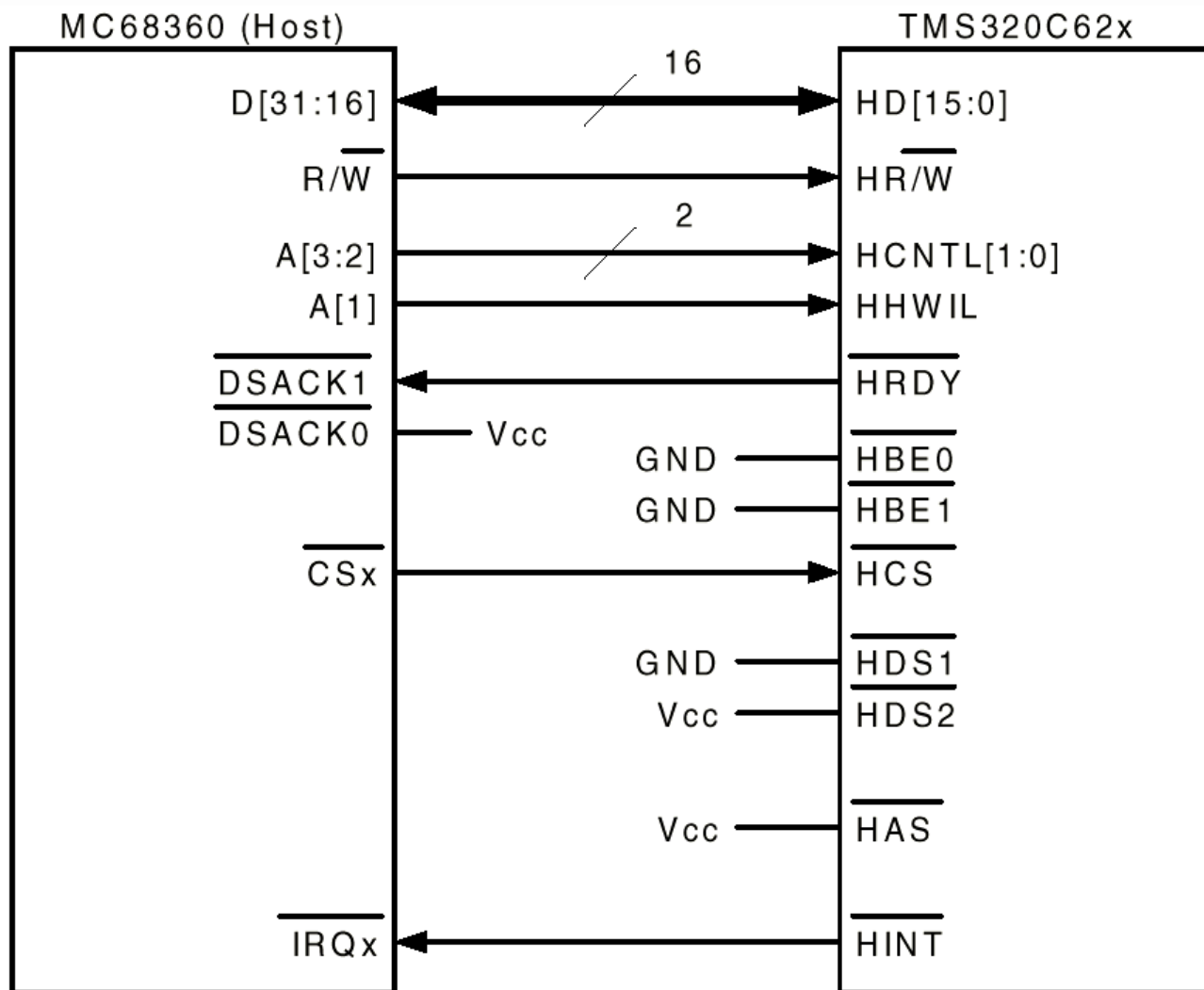


接口实例— MC68000



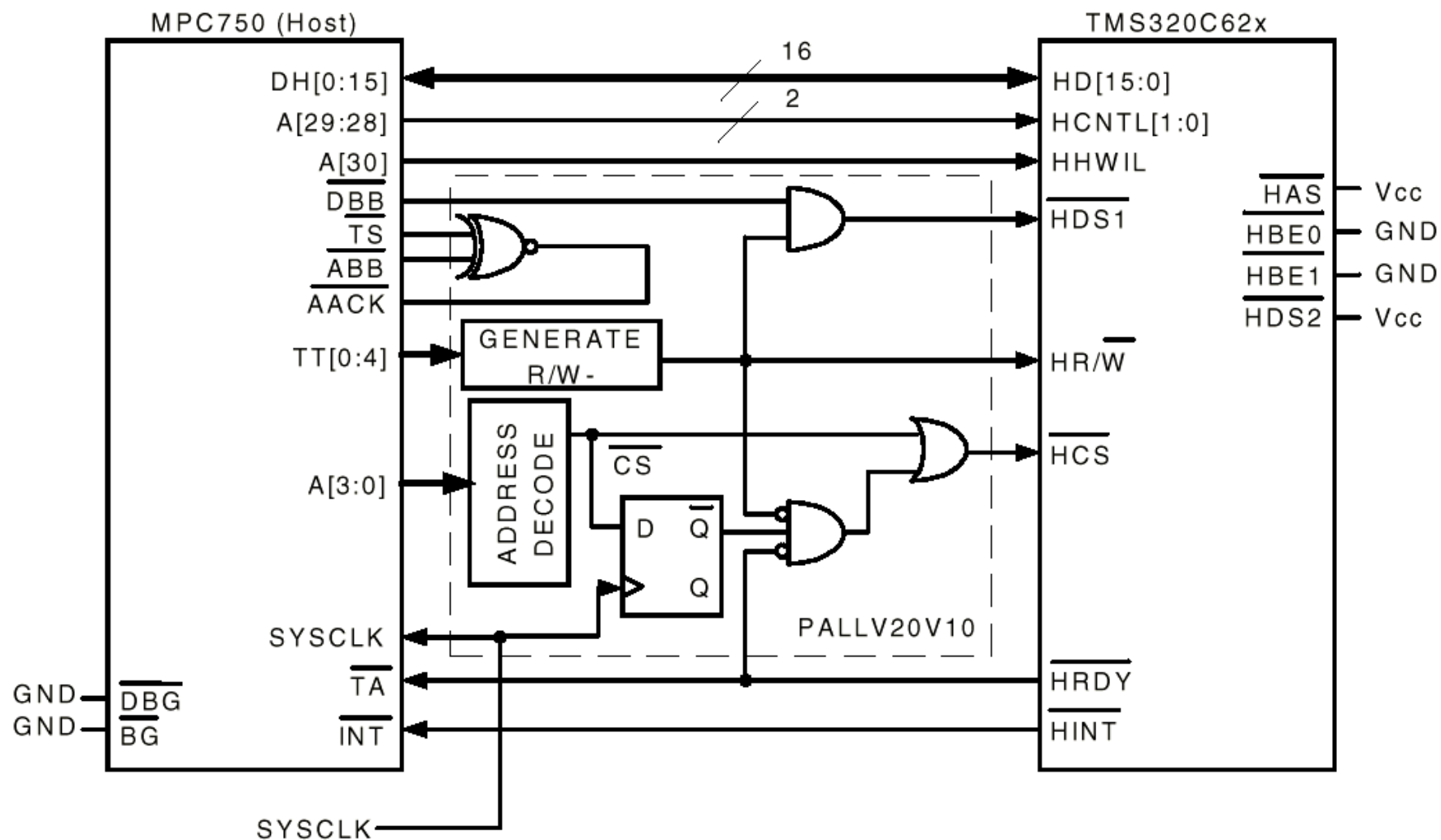


接口实例— MC68360



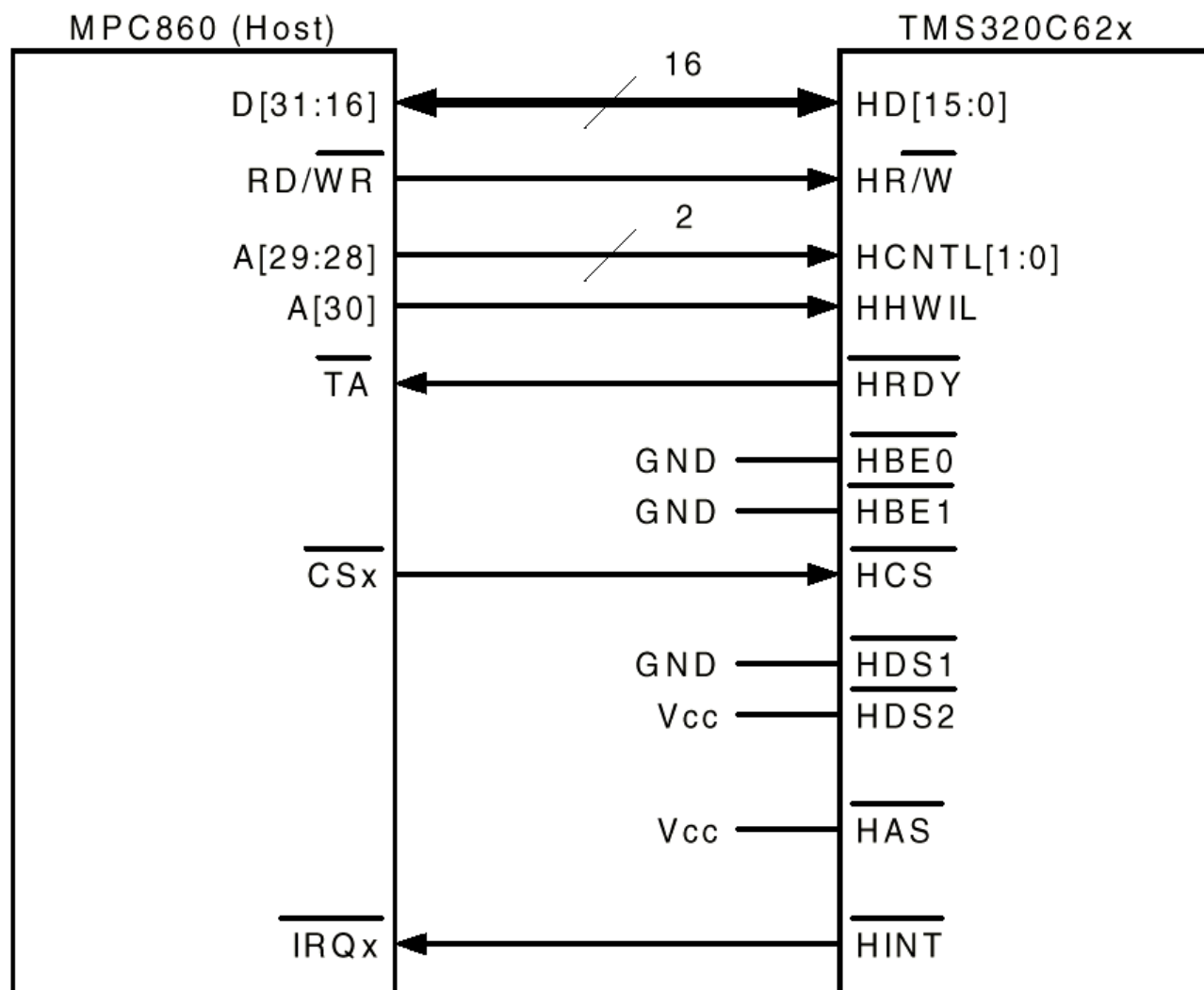


接口实例— MPC750



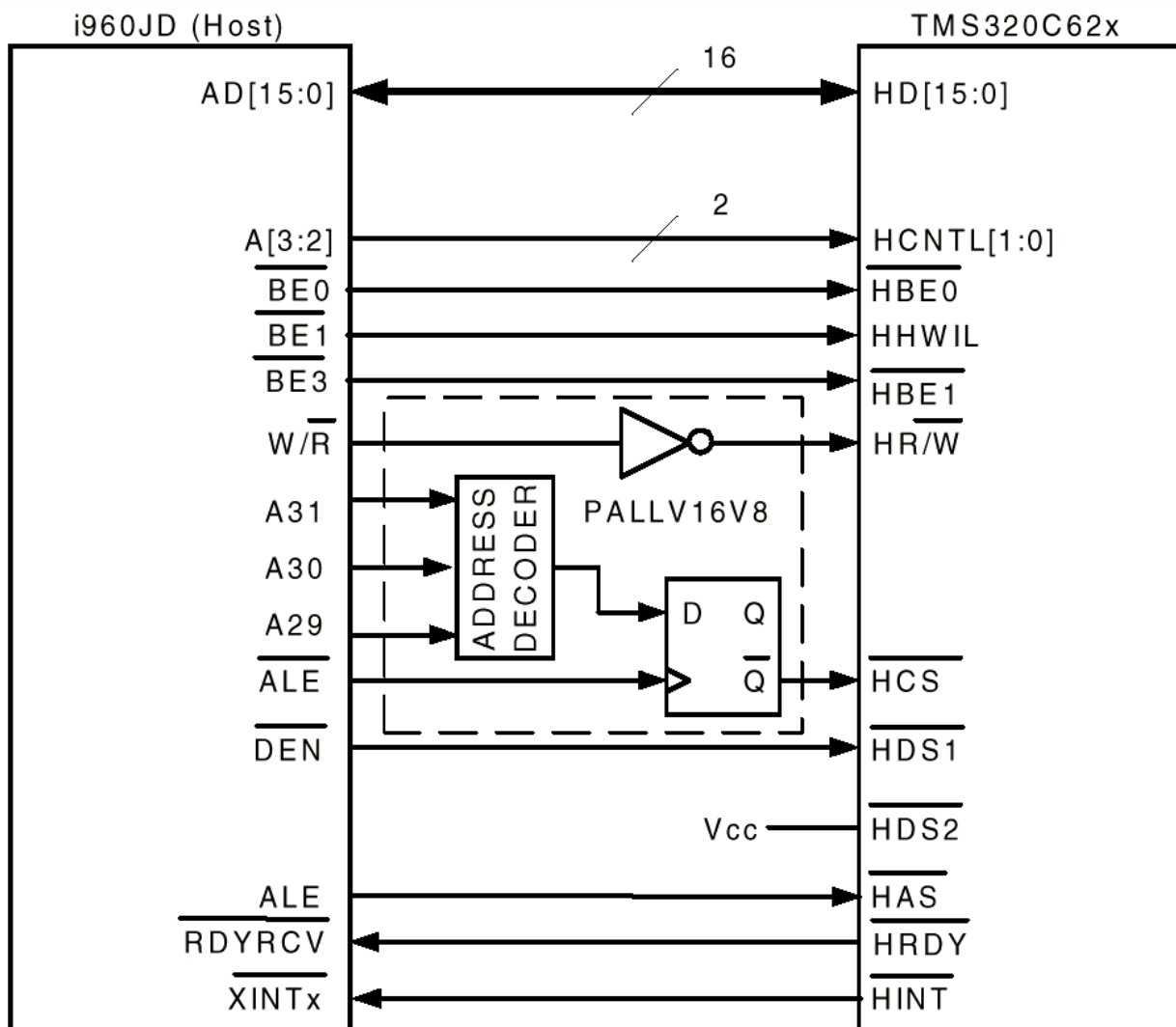


接口实例— MPC860





上海交通大学 接口实例— intel 80960Jx

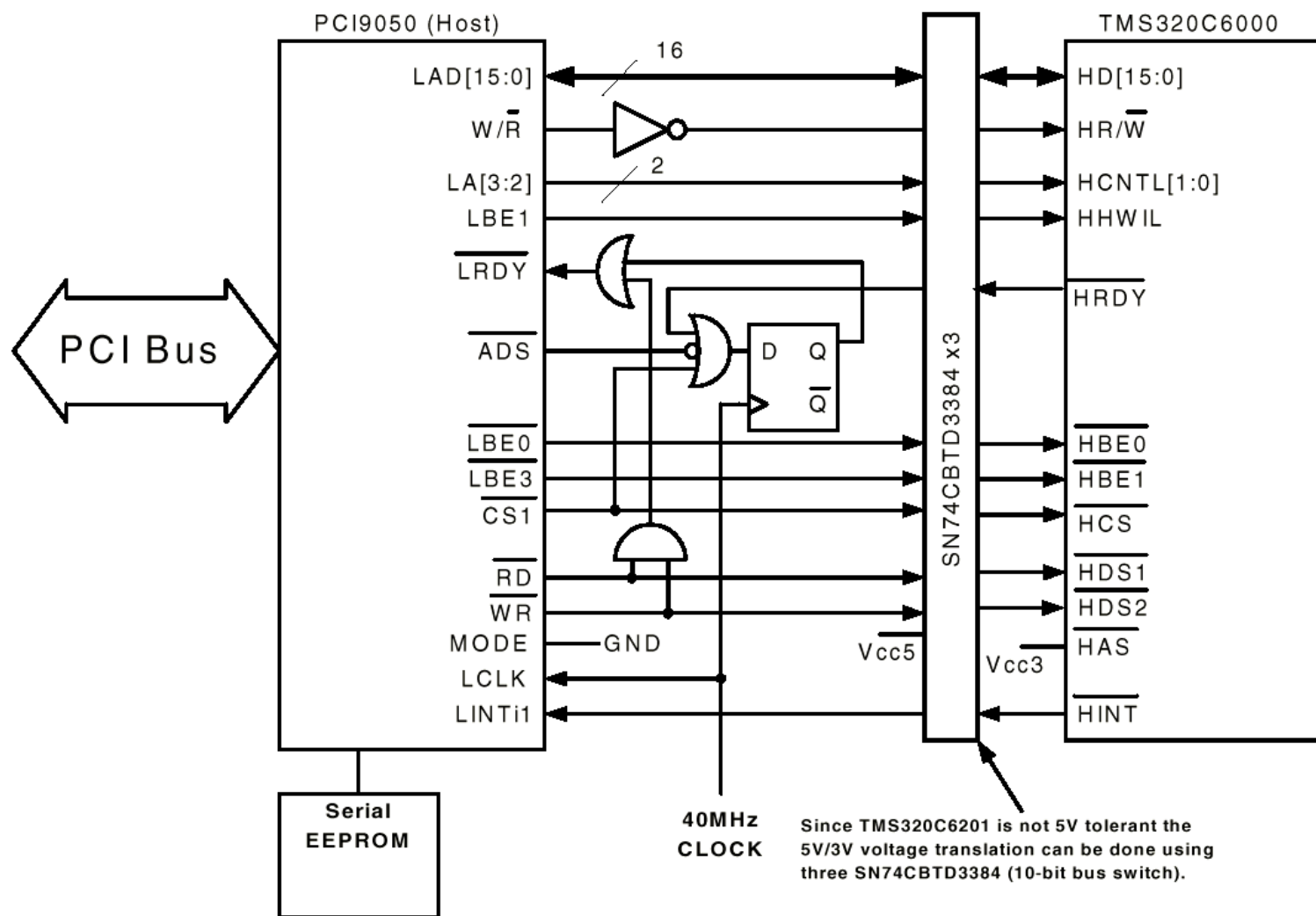


校长办公室

the Office of the President



接口实例— PCI 桥





4) DMA/EDMA

- ④ EDMA 是 C621x/C671x/C64x特有的增强的直接存储器访问方式。它可以在没有 CPU参与的情况下，由 EDMA 控制器完成 DSP 存储空间的数据搬移。
- ④ EDMA 的内容较多，将会在其它章节单独讲述。



5) 其他外设

- 除了前面详细描述의常用外设外，C64x还TIMER、McASP、I²C，GPIO等外设，在此不再赘述。
- TIMER: 定时器
- McASP: Multichannel Audio Serial Port
- I²C: Inter-Integrated Circuit
- GPIO: General-Purpose Input/Output

TI DSP培训以及技术服务简介

上海交大BME-美国德州仪器联合DSP实验室成立于2007年，是国内最权威的TI技术服务于培训机构。实验室有TI（C6000，C2000，C5000，达芬奇，多核DSP）全系列开发平台，提供DSP，MSP430等技术培训与技术服务，项目合作等。培训内容有

- 1) CCS开发环境精解与实例；
- 2) DSP/SYS BIOS 实例；
- 3) C6000/C5000/C2000全系列DSP架构以及汇编，C语言，混合编程等；
- 4) HPI，EMIF，EDMA，Timer等外设；
- 5) C6416、DM642，C6678多核EVM开发平台实例；
- 6) Bootloader 原理以及实例等。

常年开班，三人以上集体报名8折优惠，学生5折。

联系电话：13651621236（牛老师），颁发TI授权证书

邮件报名：jhniu@sjtu.edu.cn，niujinhai@yahoo.com.cn



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



颁发TI授权的培训证书



SHANGHAI JIAO TONG UNIVERSITY

the Office of the President



DSP实验室介绍

- ④ 美国德州仪器（TI）—上海交通大学（SJTU）联合DSP实验室成立于2007年10月，位于上海交大闵行校区，致力于TI DSP技术的推广，以及相关数字信号处理算法的研究与开发，为客户提供优质的产品与服务，涉及的技术领域有，无线通信，音频/视频信号处理，医学信号/图像处理，数字马达控制等。实验室研发与培训教师主要由上海交通大学青年教师承担，同时聘请了多位有企业工作背景的DSP技术专家为实验室的顾问。



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

DSP实验室介绍



校长办公室

the Office of the President



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

End

Thanks

校长办公室

the Office of the President