

AVP32VC33 数字信号处理器

数据手册

Advancechip



Electronics

湖南进芯电子科技有限公司

2018 年 6 月

版本号 Ver1.0

目录

1 产品特征..... 1

1.1 芯片描述 3

1.2 产品编码 5

1.3 引脚信息说明..... 6

1.4 引脚功能描述..... 8

1.4.1 主总线接口信号: 8

1.4.2 控制信号:..... 8

1.4.3 串行口信号..... 8

1.4.4 定时器信号..... 9

1.4.5 电源及振荡器信号 9

1.4.6 JTAG 仿真信号.....10

2 功能概述.....11

2.1 功能简要说明.....12

2.1.1 AVP32VC33 简介12

2.1.2 内部总线（改进的哈弗结构）12

2.1.3 外部总线.....12

2.1.4 总线管理.....13

2.1.5 片内存储（RAM）分布14

2.1.6 CPU 寄存器14

2.1.7 状态寄存器 ST.....15

2.1.8 中断使能寄存器 IE.....16

2.1.9 中断标志寄存器 IF.....17

2.1.10 通用 I/O 引脚 XF0 和 XF1.....18

2.1.11 复位功能18

2.1.12 中断简介18

AVP32VC33 Digital Signal Processor

2.1.13 片内 ROM 引导程序操作	19
2.1.14 JTAG.....	19
2.1.15 定时器简介	19
2.1.16 串行口简介	19
2.1.17 DMA 控制器简介.....	19
2.2 存储器管理	20
2.2.1 中断向量分配存储器地址.....	21
2.2.2 外围寄存器地址分配.....	22
2.2.3 指令高速缓存器	23
2.3 中断管理	24
2.3.1 AVP32VC33 中断向量表.....	24
2.3.2 中断优先级.....	25
2.3.3 中断处理过程.....	26
3 外围设备	27
3.1 定时器.....	27
3.1.2 定时器相关寄存器	28
3.1.3 定时器全局控制寄存器.....	28
3.1.4 定时器周期寄存器和计数寄存器	29
3.1.5 定时器脉冲发生器	30
3.1.6 定时器运行模式	31
3.1.7 定时中断.....	31
3.1.8 定时器初始化	31
3.2 DMA 控制器	32
3.2.1 全局控制寄存器	32
3.2.2 目的寄存器、源地址寄存器和传输计数器	34
3.2.3 DMA 的中断方式	34
3.2.4 DMA 的同步操作模式.....	34

AVP32VC33 Digital Signal Processor

3.2.5 DMA 的字传输过程	36
3.2.6 DMA 的初始化和启动	36
3.3 串行口	37
3.3.1 串行口全局控制寄存器	38
3.3.2 发送端口控制寄存器	40
3.3.3 接收端口控制寄存器	41
3.3.4 收发定时器控制寄存器	42
3.3.5 收发计数器和收发周期寄存器	43
3.3.6 数据发送寄存器和数据接收寄存器	43
3.3.7 串行口的数据传输方式	44
3.3.8 串行口常用的工作方式	48
3.3.9 串行口的中断源	48
3.3.10 串行口初始化与配置	48
3.4 程序引导功能	49
3.4.1 引导方式选择	49
3.4.2 引导程序的工作流程	50
3.4.3 hex 格式转换命令	51
4 开发支持	52
5 电气规范	53
5.1 绝对最大额定值	53
5.2 建议的运行条件	53
5.3 建议的电源电压范围内的电气特性（除非另有说明）	54
5.4 锁相环（PLL）电路时序	55
5.5 时钟电路时序	56
5.6 内存读/写时序	59
5.7 执行 LDFI 和 LDII 时的 XF0 和 XF1 时序	62
5.8 执行 STFI 和 STII 时的 XF0 的时序	63

5.9 执行 SIGI 时的 XF0 和 XF1 时序.....64

5.10 XF_x 配置为输出时加载.....65

5.11 将 XF_x 从输出模式更改为输入模式时加载.....66

5.12 将 XF_x 从输入模式更改为输出模式67

5.13 复位时序.....68

5.14 中断响应时序.....70

5.15 中断确认时序.....71

5.16 串口定时时序.....72

5.17 数据速率时序模式73

5.18 HOLD 时序.....74

5.19 通用 I/O 时序76

5.20 定时器引脚时序78

5.21 SHZ 引脚时序.....79

5.22 IEEE-1149.1 测试访问端口时序.....79

5.23 封装尺寸.....79

1 产品特征

•采用高性能 CMOS 工艺

- 主频 75MHz (周期 13.3ns)
- 低功耗设计 (内核 1.8V , I/O 电压 3.3V)
- 集成片内 1.8V 线性稳压电源 (LDO)

•支持 JTAG 在线仿真

•高性能 32 位 CPU

- 硬件乘法器和算术逻辑单元 ALU , 单周期完成一次乘累加运算
- 哈佛(Harvard)总线架构
- 原子操作
- 快速中断响应和处理
- 统一寄存器编程模式
- 可使用 C/C++ 和汇编语言高效率编程
- 一个 32 位桶形移位器 , 单周期内向左或向右移位 32 位、
- 与 CPU 并行操作片内 DMA 控制器
- 1 个全双工串行口 , 可编程为 32 位、24 位、16 位和 8 位字长
- 2 个 32 位硬件定时器
- 内置 5 倍频锁相环时钟电路

•片内存储器资源

- RAM2 和 RAM3 : 2 块 16K x32 位片内 RAM
- RAM0 和 RAM1 : 2 块 1K x32 位片内 RAM
- 引导 ROM : 1 块 4K x32 位固化在片内的引导存储映射空间
- 外围总线存储器映射寄存器 : 1 块 6Kx32 位存储空间

•外部存储扩展接口

- 微计算机引导模式下 : 1 块 8MB-4KB 和 1 块 8MB-40KB 的可扩展外部空间
- 微处理器方式下 : 1 块 8MB-64B 和 1 块 8MB-40KB 的可扩展外部空间
- 可编程等待状态
- 可编程读/写时序
- 4 个独立片选信号

- 时钟和系统控制**

- 支持动态锁相环(PLL)分频系数调整
- 片内振荡器

- 四个外部中断接口**

- 一个串行端口和 DMA 通道**

- 32 个 I/O 引脚**

- 先进的仿真功能**

- 分析和断点功能
- 基于硬件的实时调试

- 封装选项**

- LQFP 封装

- 开发工具**

- ANSI C/C++ 编译器/汇编语言/连接器
- 支持 Code Composer C3X
- JTAG 仿真器

- 产品等级**

- S： 宇航级（暂无）
- J： 军品级
- G：工业级
- C：消费级

1.1 芯片描述

AVP32VC33,144PIN 封装引脚分配图,如图 1.1 所示

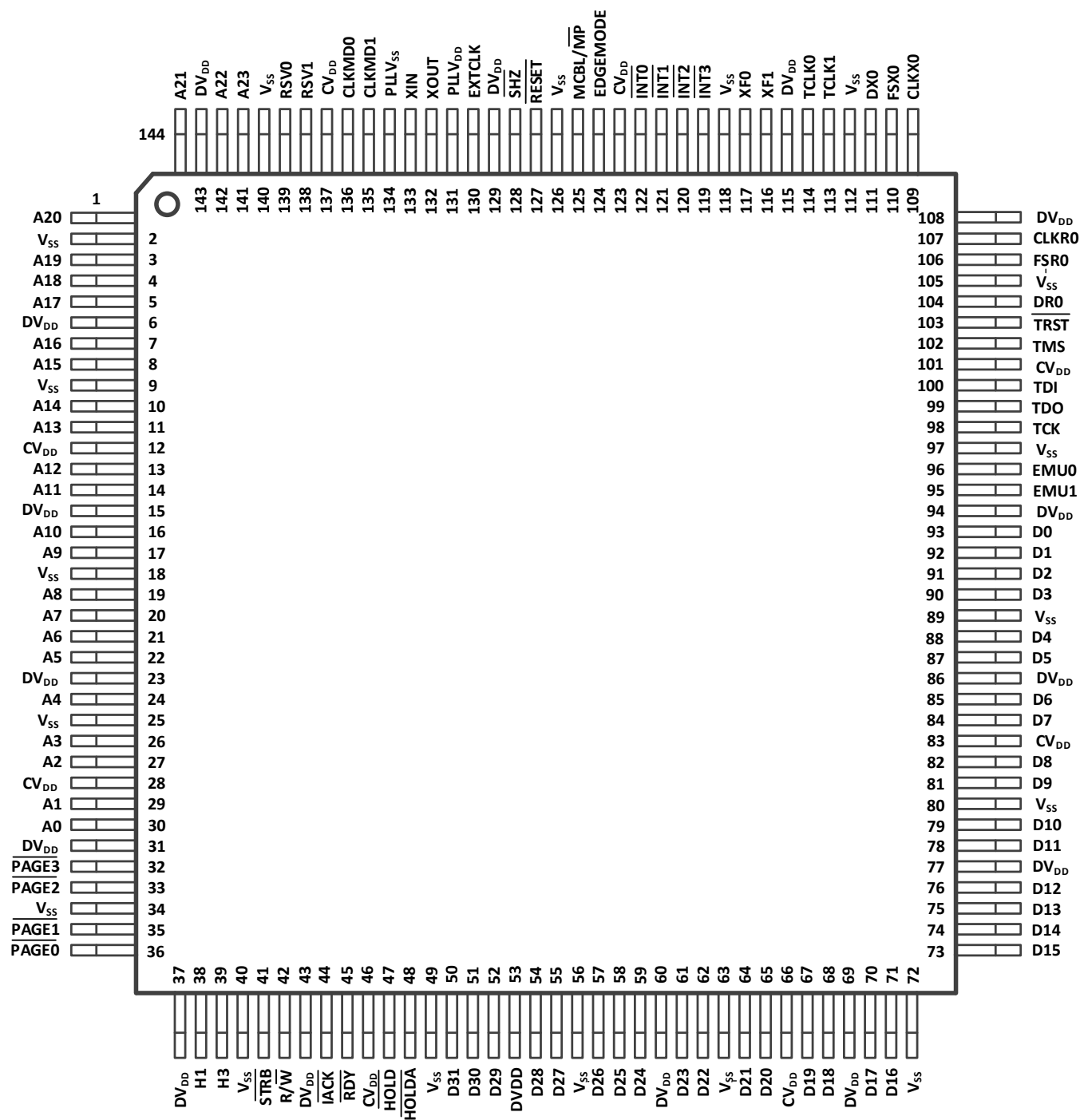


图 1.1 封装引脚图(顶视图)

AVP32VC33 Digital Signal Processor

AVP32VC33,引脚编号与信号名称对应关系,如表 1.1 所示

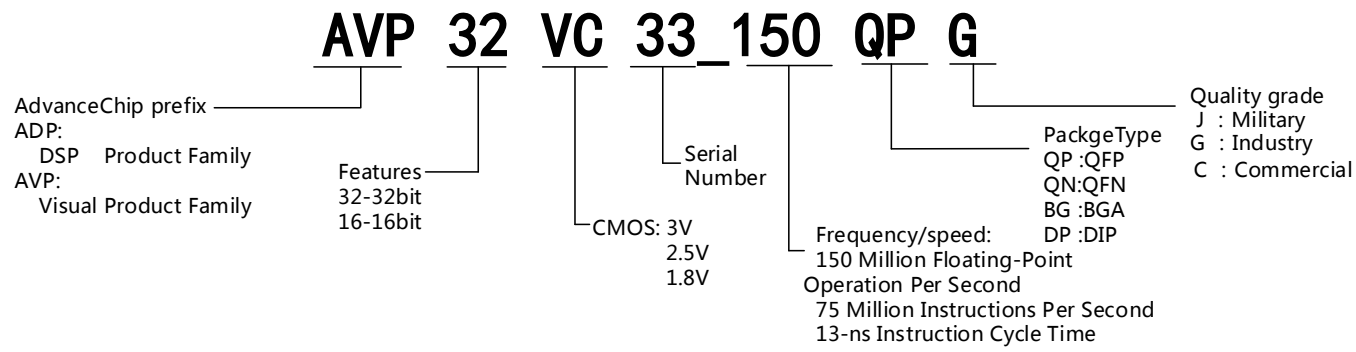
表 1.1 引脚编号与信号名称对应关系表

信号名称	引脚编号	信号名称	引脚编号	信号名称	引脚编号	信号名称	引脚编号
A0	30	D0	93	DV _{DD}	31	R/W	42
A1	29	D1	92		37	RDY	45
A2	27	D2	91		43	RESET	127
A3	26	D3	90		53	RSV0	139
A4	24	D4	88		60	RSV1	138
A5	22	D5	87		69	SHZ	128
A6	21	D6	85		77	STRB	41
A7	20	D7	84		86	TCK	98
A8	19	D8	82		94	TCLK0	114
A9	17	D9	81		108	TCLK1	113
A10	16	D10	79		115	TDI	100
A11	14	D11	78		129	TDO	99
A12	13	D12	76		143	TMS	102
A13	11	D13	75	DX0	111	TRST	103
A14	10	D14	74	EDGEMODE	124	V _{SS}	2
A15	8	D15	73	EMU0	96		9
A16	7	D16	71	EMU1	95		18
A17	5	D17	70	EXTCLK	130		25
A18	4	D18	68	FSR0	106		34
A19	3	D19	67	FSX0	110		40
A20	1	D20	65	H1	38		49
A21	144	D21	64	H3	39		56
A22	142	D22	62	HOLD	47		63
A23	141	D23	61	HOLDA	48		72
CLKMD0	136	D24	59	IACK	44		80
CLKMD1	135	D25	58	INT0	122		89
CLKR0	107	D26	57	INT1	121		97
CLKX0	109	D27	55	INT2	120		105
CV _{DD}	12	D28	54	INT3	119		112
	28	D29	52	MCBL/MP	125		118
	46	D30	51	PAGE0	36		126
	66	D31	50	PAGE1	35		140
	83	DR0	104	PAGE2	33	XIN	133
	101	DV _{DD}	6	PAGE3	32	XOUT	132
	123		15	PLL _{VDD}	131	XF0	117
	137		23	PLV _{SS}	134	XF1	116

AVP32VC33 Digital Signal Processor

1.2 产品编码

ADP32FXX 产品代号编码规则:



1.3 引脚信息说明

AVP32VC33,引脚类型和功能说明,如表 1.2 所示

表 1.2 引脚类型与功能说明表

引脚名	I/O 特性	功能说明	高阻条件
主总线接口信号 (65 个引脚)			
D31-D0	I/O/Z	32 位数据线	S H R
A23-A0	O/Z	24 位地址线	S H R
R/W	O/Z	读写信号。“读”操作时输出高电平,“写”操作时输出低电平	S H R
PAGE0-PAGE3	O/Z	页选通信号。用于对 16MB 存储空间分页选通,每页容量为 4MB	S H R
STRB	O/Z	外部存取选通信号	S H
RDY	I	就绪信号。表示外部设备已经为主总线接口操作做好准备,低电平有效	
HOLD	I	保持信号。当 HOLD 为低电平时,结束任何正在进行的处理。在 HOLD 变为高电平之前,D(0-31)、A(0-23)、STRB、R/W 被置为高阻状态,主总线接口的所有操作被保持	
HOLDA	O/Z	保持信号。本信号是响应 HOLD 低电平而产生的,它标志着 D(0-31)、A(0-23)、STRB、R/W 已被置为高阻状态,总线上的所有操作被保持	S
控制信号 (11 个引脚)			
RESET	I	复位信号。低电平时,处理器被置于复位状态,当其变为高电平时,从复位向量所指定的位置开始执行指令	
EDGEMODE	I	中断边缘方式控制信号。高电平有效,EDGEMOD=1 表示允许以边缘方式触发中断	
INT3-INT0	I	外部中断信号	
IACK	O/Z	中断确认。IACK 是由 IACK 指令生成的,可用于指示何时执行代码段	S
MCB/MP	O/Z	微计算机引导方式和微处理器方式控制信号	
SHZ	I	高阻控制信号。低电平有效,可使所有引脚处于高阻状态,存储器与存储器原有内容丢失	
XF1、XF0	I/O/Z	外部标志信号,这两个引脚可作为通用 I/O 引脚,或使用于多个 DSP 间的通信	S R
串行口信号 (6 个引脚)			
CLKR0	I/O/Z	串行口的接收时钟信号	S R
CLKX0	I/O/Z	串行口的发送时钟信号	S R
DR0	I/O/Z	接收串行数据信号	S R
DX0	I/O/Z	发送串行数据信号	S R
FSR0	I/O/Z	接收帧同步脉冲信号	S R

AVP32VC33 Digital Signal Processor

FSX0	I/O/Z	发送帧同步脉冲信号	S R
定时器信号（2 个引脚）			
TCLK0	I/O/Z	定时器 0 的时钟信号。作为输入，TCLK0 是定时器 0 的定时器时钟或计数脉冲；作为输出，TCLK0 输出定时器 0 产生的脉冲	S R
TCLK1	I/O/Z	定时器 1 的时钟信号。作为输入，TCLK1 是定时器 1 的定时器时钟或计数脉冲；作为输出，TCLK0 输出定时器 1 产生的脉冲	S R
电源及振荡器信号（53 个引脚）			
H1	O/Z	CPU 工作时钟。通过外部引脚输出的 CPU 的工作时钟，H1 与 H3 相位相反	S
H3	O/Z	CPU 工作时钟。通过外部引脚输出的 CPU 的工作时钟，H3 与 H1 相位相反	S
CV _{DD}	I	CPU 的内核供电电压。该电压为 1.8V	
DV _{DD}	I	给 I/O 引脚提供电源电压。该电压为 3.3V	
V _{SS}	I	接地	
PLL _{VDD}	I	锁相环电源。连接到 CVDD（1.8V）	
PLL _{VSS}	I	锁相环地	
EXTCLK	I	外部参考时钟输入引脚	
XOUT	O	参考时钟输出	
XIN	I	参考时钟输入	
CLKMD0、CLKMD1	I	时钟选择方式，用于对时钟倍频，倍频后的时钟作为 CPU 的工作时钟	
RSV0、RSV1	I	保留引脚	
JTAG 仿真（7 个引脚）			
EMU1、EMU0	I/O	仿真引脚。通过电阻上拉后连接到 DVDD	
TDI	I	测试数据输入	
TDO	O	测试数据输出	
TCK	I	测试时钟	
TMS	I	测试选择方式	
$\overline{\text{TRST}}$	I	测试复位	

表 1.2 中介绍了 AVP32VC33 芯片各个引脚的 I/O 特性及其功能说明，其中 I/O/Z 表示引脚的输入输出类型和特性，I 表示输入，O 表示输出，Z 表示高阻，I/O/Z 表示该引脚有输入、输出和高阻三种状态。表最右边一栏表示出现高阻的状态，其中 S、H、R 分别表示 SHZ、HOLD、RESET 有效。

由表可知 AVP32VC33 的 144 个引脚可按照功能分为六大类，分别是（1）主总线接口：65 个引脚；（2）控制信号：11 个引脚；（3）串行口信号：6 个引脚；（4）定时器信号：2 个引脚；（5）电源及振荡器信号：53 个引脚；（6）JTAG 信号：7 个引脚。

1.4 引脚功能描述

1.4.1 主总线接口信号:

- 1、数据线 D0~D31，共 32 条，表明 AVP32VC33 总线带宽为 32 位；
- 2、地址线 A0~A23，共 24 条，可寻址空间为 16MB；
- 3、读写信号 R/W：高电平为读，低电平为写；
- 4、外部存取选通信号 STRB：对所有外部存取操作输出低电平；
- 5、就绪信号 RDY：主要用于与外部低速器件接口，为输入信号；
- 6、HOLD 和 HOLDA 分别为保持和保持回答信号：HOLD 是输入的控制信号，低电平有效，HOLDA 为处于保持状态时的回答信号，如果因为某种连接错误原因使 HOLDA 变为低电平，则此时 DSP 不工作；
- 7、页选通信号 PAGE0~PAGE3：将 16MB 地址空间分为 4 页，每页 4MB 字节，如表所示：

表 1.3 页选通信号范围

页选通信号	开始地址	结束地址
PAGE0	000000h	3FFFFFFh
PAGE1	400000h	7FFFFFFh
PAGE2	800000h	BFFFFFFh
PAGE3	C00000h	FFFFFFh

1.4.2 控制信号:

- 1、复位信号 RESET：低电平有效，当 RESET=0 时，DSP 进入复位状态，地址线、数据线和各个输出信号均处于高阻状态；
- 2、高阻控制信号 SHZ：SHZ 可以与 RESET 配合使用，也可以单独使用，使地址总线、数据总线和各输出信号处于高阻状态，当 SHZ=0 时，除了使各种输出信号处于高阻状态，同时也使 DSP 的存储器和寄存器内容丢失。（应注意 SHZ 与 HOLD 的区别，HOLD 只能使主总线接口信号处于高阻状态）
- 3、EDGEMODE 是中断的边缘触发方式控制信号，若 EDGEMODE=1，在中断引脚上出现的下降沿触发中断。如果 EDGEMODE=0，则中断引脚上的低电平将连续的使中断寄存器中的中断标志位置 1，CPU 或 DMA 响应中断后再两个 CPU 周期内对该标志位清零。
- 4、INT3~INT0 是外部中断信号，作用一是用于上电瞬间进行程序的引导，作用二是接收外部事件发出的中断请求，使 DSP 响应外部中断。
- 5、MCBL/MP 是微计算机引导方式和微处理器方式控制信号，若 MCBL/MP=1，则为微计算机引导方式，若 MCBL/MP=0，则为微处理器工作方式，这两种方式下，DSP 存储器分配上不同的。

1.4.3 串行口信号

串行口信号共 6 个，对应于 6 个引脚，分别是 CLKR0、CLKX0、DR0、DX0、FSR0、FSX0，构成一个全双工串口，任何时候都可以同时进行数据的接收和发送，接收和发送的数据字长可以是 8 位、16 位、24 位或 32 位；

1.4.4 定时器时钟信号

有 2 个 32 位定时器，为定时器 0 和定时器 1，分别记为 TCLK0 和 TCLK1，既可以单独设定为定时器输出信号，也可以单独设置为通用 I/O 引脚。

1.4.5 电源及振荡器信号

1、AVP32VC33 有 DV_{DD} 和 CV_{DD} 两种电源，其中 DV_{DD} 为 3.3V，是 I/O 引脚的供电电源， CV_{DD} 为 1.8V，对 DSP 芯片内部 CPU 供电，AVP32VC33 的 144 个引脚中， DV_{DD} 有 16 个引脚， CV_{DD} 有 8 个引脚， V_{SS} 有 18 个引脚。

2、振荡器：即为时钟发生器，为 AVP32VC33 芯片提供时钟，时钟发生器由芯片的片内振荡电路和锁相环电路组成。时钟发生器工作时，必须对其提供一个参考时钟，可以采用两种方法来产生这一时钟，

方法一如图 1.2 所示，在 EXTCLK 引脚上直接接入一外部输入时钟，如一体化振荡器的输出信号等，这时 XIN 接地，XOUT 悬空，

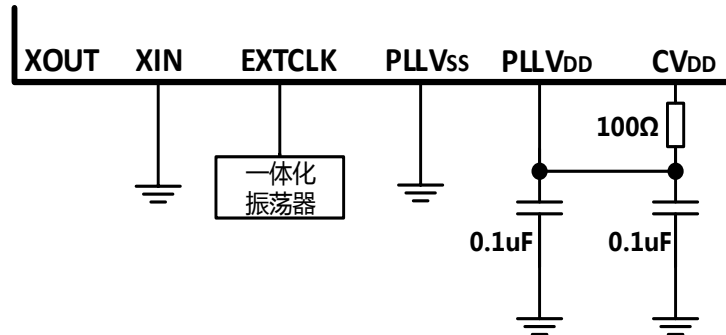


图 1.2 外部时钟连接方法

方法二：将 EXTCLK 接地，通过 XIN 和 XOUT 引脚接入晶体或陶瓷振荡器，如图 1.3 所示，该电路采用 1~20MHz 的晶体振荡器的典型的串联阻抗为 25Ω ，当频率低于 8MHz 时，阻抗将更高。为了对振荡电路滤波并使相位保持稳定，振荡电路的阻抗应为晶体振荡器阻抗的 10~40 倍，为此在电路中引入了补偿电阻 R_d 构成了滤波电路， C_2 进一步起滤波的作用。经过以上滤波后，XIN 引脚上输入的电压幅度是晶体振荡器所提供电压幅度的 75%~85%。锁相环电路的作用是锁定参考时钟的相位，一旦达到同步或锁定后，将自动跟踪参考时钟的相位，并对输入的参考时钟进行倍频，使倍频后的时钟与参考时钟实现同步。与锁相环电路相关的引脚 PLL_{VDD} 、 PLL_{VSS} 的连接方法如图 1.4 所示。随晶体振荡器频率的不同，外接元件 R_d 、 C_1 、 C_2 也应不同的值，具体选择方法如表 1.4。

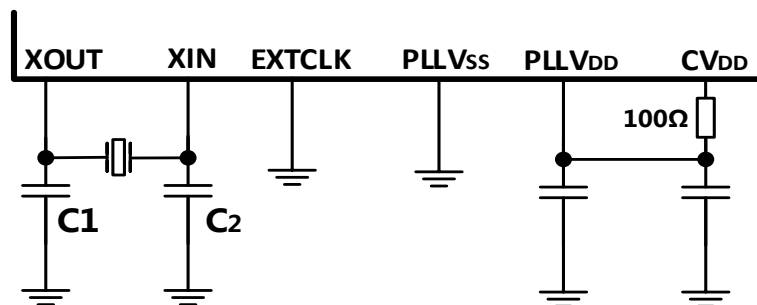


图 1.3 晶体振荡器连接方法

表 1.4 晶体振荡电路参数选择

频率/MHz	R _d /Ω	C ₁ /PF	C ₂ /PF	CL/PF	RL/Ω
2	4.7k	18	18	12	200
5	2.2k	18	18	12	60
10	470	15	15	12	30
15	0	15	15	12	25
20	0	9	9	10	25

3、时钟方式选择

通过引脚 CLKMD0 和 CLKMD1 可以对 CPU 的工作时钟进行选择，即对参考时钟进行倍频，可选的倍数如表 1.5 所示。由表可以看出，CPU 时钟可以是参考时钟频率的 1 倍、1/2 倍和 5 倍，但锁相环电路只有在 5 倍频的情况下才处于工作状态。

表 1.5 时钟控制引脚的接法与倍频的关系

CLKMD0	CLKMD1	锁相环电路状态	内部振荡电路状态	倍数
0	0	不工作	不工作	1
0	1	不工作	工作	1/2
1	0	不工作	工作	1
1	1	工作	工作	5

1 倍频时 CLKMD0 和 CLKMD1 有两种不同的接法，但二者都为低电平时，内部振荡电路不工作，其他三种情况下，内部振荡电路都处于工作状态。但并不意味着一定要采用晶体振荡器作为参考时钟，即也可以采用外部电路从 EXTCLK 引脚提供参考时钟，这时，内部振荡电路的工作可以理解为只是接通了这片内对这一局部电路供电的电源通道。

4、H1 和 H3

H1 和 H3 是通过外部引脚输出的 CPU 的工作时钟，只作为输出用，当 $\overline{\text{SHZ}}$ 有效时可处于高阻状态。H1 和 H3 的作用一是作为电路调试时检测 CPU 的工作时钟，作用二是为 DSP 以外的电路提供时钟信号。H1 和 H3 的频率一定是 CPU 的工作时钟频率，周期一定是 CPU 的指令周期。

1.4.6 JTAG 仿真信号

AVP32VC33 的 JTAG 口的作用主要是用于将程序下载到芯片中，并对软硬件进行仿真。AVP32VC33 的片内 ROM 存放的是出厂时固化的引导程序，所以通过 JTAG 口只能将程序下载到 DSP 的 RAM 内，一旦掉电，下载到 RAM 中的程序将会自动丢失，所以，AVP32VC33 中的 JTAG 口只对 DSP 进行测试和仿真。脱离仿真器后 DSP 的自动运行需要在每次上电时，通过复位和中断引脚，将存放于芯片外部存储器中的程序自动引导到 DSP 的内部 RAM 中。

2 功能概述

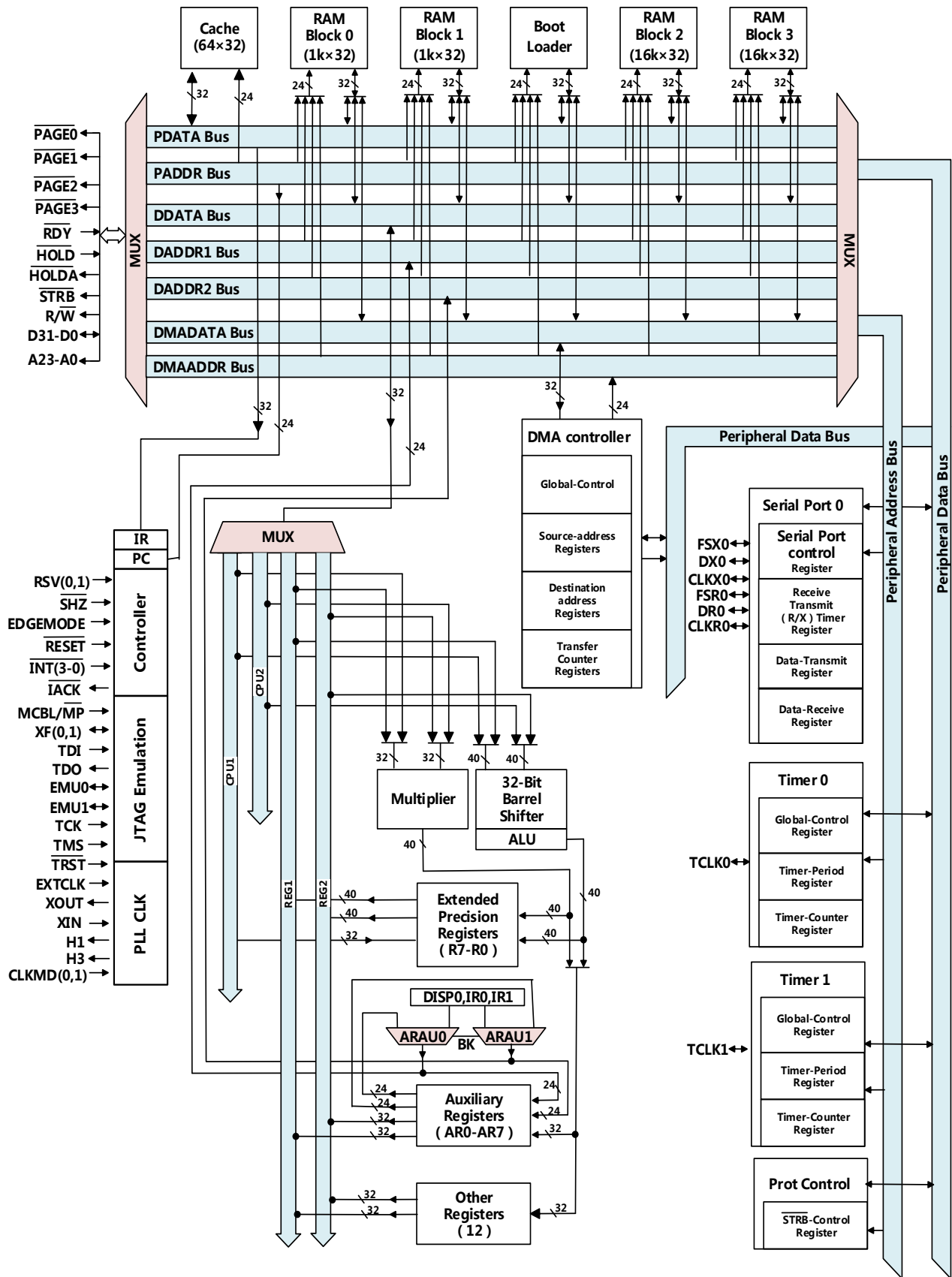


图 2.1 功能框图

2.1 功能简要说明

2.1.1 AVP32VC33 简介

AVP32VC33 是进芯电子浮点 DSP 平台上的一款产品，采用 0.18 μ m 四层金属 CMOS 制造 32 位浮点处理器技术，具有很高的速度和精度，其内部总线和特殊数字信号处理指令集可灵活地执行每秒高达 1.5 亿次浮点运算 (MFLOPS)。AVP32VC33 可以在一个整数或浮点数据上执行并行乘法和 ALU 运算周期，并且每个处理器还具有通用寄存器文件，程序高速缓存，专用 ARUA，内部双存取存储器，支持并行 I/O 的一个 DMA 通道以及较短的机器周期，高性能和易用性等这些优势。

2.1.2 内部总线 (改进的哈佛结构)

AVP32VC33 采用的是改进的哈佛结构，既采用了独立的程序总线和数据总线，又在程序总线和数据总线中采用了交叉连接，允许数据存放在程序存储器中，并被算术运算指令直接使用，增强了芯片的灵活性，提高了运行速度。

内部总线包括程序总线、数据总线和 DMA 总线，是与 CPU 寄存器和程序计数器(PC)相关联的总线，内部总线的特点是：程序总线与数据总线是分开的，地址总线与数据总线是分开的。内部总线的具体划分是：

程序总线： 程序的地址总线 PADDR

程序的数据总线 PDATA

数据总线： 数据的地址总线 DADDR1 和 DADDR2

数据的数据总线 DDATA

DMA 总线： DMA 的地址总线 DMAADDR

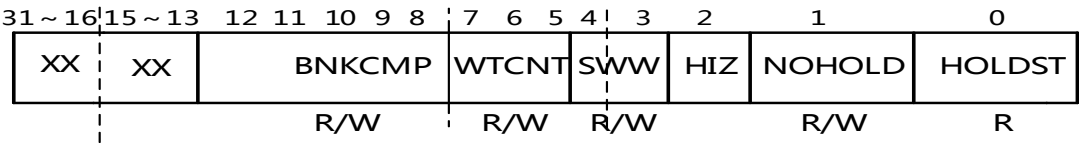
DMA 的数据总线 DMADATA

2.1.3 外部总线

在 AVP32VC33 中采用了多种总线，其目的是提高运算能力和速度，但对用户而言，主要涉及 24 条地址线和 32 条数据线，外部引脚通过多路选择开关连接到内部总线上，简化了芯片的外部结构，同时也提高了 DSP 的性能。这种通过多路切换开关输出的总线称为外部总线。外部总线有两种，一是主总线，二是外围总线。主总线包括 32 位数据线、24 位地址线、读写信号 $\overline{R/W}$ ，外部存取选通信号 $\overline{STRB}$ ，页选通信号 $\overline{PAGE0} \sim \overline{PAGE3}$ 、就绪信号 $\overline{RDY}$ 、保持信号 $\overline{HOLD}$ 、保持回答信号 $\overline{HOLDA}$ 。其中页选通信号是由芯片内的地址线 A22 和 A23 经过译码产生的，外部存取选通信号也是由地址线决定的。外围总线是通过多路切换开关输出的总线，主要用于对定时器、串行口控制寄存器和 DMA 寄存器的控制，这些寄存器不属于 CPU 寄存器，其地址在 DSP 内部 RAM 中。

2.1.4 总线管理

总线管理主要涉及外部总线的管理，达到访问外部存储器和外围器件的目的。通过主总线控制寄存器和使用外部输入信号，用户可控制总线的等待状态，从而允许 CPU 访问低速存储器和外围设备。外部总线接口控制寄存器的地址为：808064h，该寄存器为 32 位的寄存器，结构如图 2.2 所示。表 2.1 列出了该寄存器中各位的功能。



R/W：读/写；xx：保留位，读作 0

图 2.2 外部总线接口控制寄存器结构

表 2.1 外部总线控制寄存器各位的名称

位	名称	复位值	功能描述
0	HOLDST	0	保持状态标志位。为 1 时表明 CPU 正在响应 $\overline{\text{HOLD}}$ 请求
1	NOHOLD	0	允许 HOLD 控制位。为 1 时允许 $\overline{\text{HOLD}}$ 使总线处于保持状态
2	HIZ	0	内部保持控制位。用软件使 HIZ=1, CPU 进入保持阶段，作用与外部 $\overline{\text{HOLD}}$ 但优先级更高
3~4	SWW	11	软件等待状态发生器。与 WTCNT 配合使用，这两位定义了等待状态的模式 00：等待，直到硬件等待信号 $\overline{\text{RDY}}$ 有效 01：等待，直到软件指定等待的状态数减少为 0 10：硬件与软件任一有效 11：硬件与软件都有效
5~7	WTCNT	111	软件等待状态数。在软件等待模式下，内部等待所插入的状态数，范围为 0~7 个周期
8~12	BNKCMP	10000	这 5 位组成的字段用于定义块大小，该段的段值也就是地址的最高有效位数。例如：当 BNKCMP=2 时，最高有效位地址的数目为 2，即 A23、A22，块的大小为 16 MB/4=4 MB。

表 2.2 BNKCMP 与用于定义块大小的最高有效位的对应关系

BNKCMP	最高有效位	块的大小/B
00000	无	16M
00001	A23	8M
00010	A23~A22	4M
00011	A23~A21	2M
00100	A23~A20	1M
00101	A23~A19	512K
00110	A23~A18	256K
00111	A23~A17	128K
01000	A23~A16	64K
01001	A23~A15	32K
01010	A23~A14	16K
01011	A23~A13	8K
01100	A23~A12	4K
01101	A23~A11	2K
01110	A23~A10	1K
01111	A23~A9	512
10000	A23~A8	256
10001~11111	保留	未定义

2.1.5 片内存储 (RAM) 分布

AVP32VC33 的全部存储器空间是 16Mx32 位，程序、数据和 I/O 空间都包含在这个存储器空间内，表格、系数、程序代码或数据既可以存储在 RAM 中，也可以存储在 ROM 中，这就可以最大程度地使用存储器空间并且按照需要分配存储器空间，但应注意片内可供用户使用的存储空间只有 RAM，片内的 ROM 在芯片出厂时已固化了引导程序，用户不能对其进行修改。

供用户使用的片内 RAM 共 4 块，RAM0 和 RAM1 均为 1Kx32 位，RAM2 和 RAM3 均为 16Kx32 位，4 块 RAM 的总容量为 34Kx32 位。存储器还有一块 64x32 位的高速缓存器 (Cache)，供指令快速译码使用，另有一块 4Kx32 位的 ROM，用于存放引导程序。用户的程序和数据只能存放在总容量为 34Kx32 位的四个 RAM 中，不能存放在 ROM 和 Cache 内。

2.1.6 CPU 寄存器

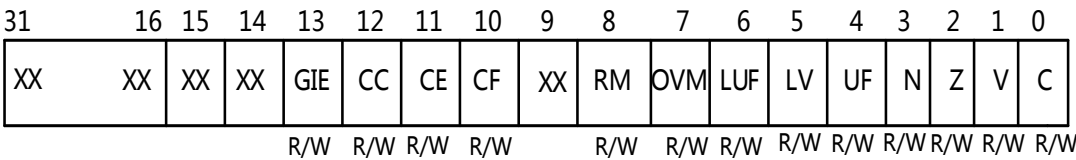
AVP32VC33 的 CPU 寄存器共 28 个，它们不占用 16MB 的存储空间，译码时，每个寄存器都有一个特定的机器代码，可以通过乘法器和算术逻辑单元 (ALU) 进行操作。这 28 个 CPU 寄存器的名称、机器代码和功能描述如表 2.3 所示

表 2.3 CPU 寄存器

寄存器名称	机器代码	功能
R0	00	扩展精度寄存器 0
R1	01	扩展精度寄存器 1
R2	02	扩展精度寄存器 2
R3	03	扩展精度寄存器 3
R4	04	扩展精度寄存器 4
R5	05	扩展精度寄存器 5
R6	06	扩展精度寄存器 6
R7	07	扩展精度寄存器 7
AR0	08	辅助寄存器 0
AR1	09	辅助寄存器 1
AR2	0A	辅助寄存器 2
AR3	0B	辅助寄存器 3
AR4	0C	辅助寄存器 4
AR5	0D	辅助寄存器 5
AR6	0E	辅助寄存器 6
AR7	0F	辅助寄存器 7
DP	10	数据页指针
IR0	11	索引寄存器 0
IR1	12	索引寄存器 1
BK	13	块规模寄存器
SP	14	系统堆核指针
ST	15	状态寄存器
IE	16	中断使能寄存器
IF	17	中断标志寄存器
IOF	18	I/O 控制寄存器
RS	19	块重复起始地址寄存器
RE	1A	块重复结束地址寄存器
RC	1B	块重复计数器

2.1.7 状态寄存器 ST

状态寄存器 ST(Status Register)包含了与 CPU 状态有关的全部信息。通常 CPU 根据指令的操作和运行结果是零还是负等情况来设置状态寄存器的标志位，这就是说，状态寄存器的内容会随着数据传输、存储、算术运算、逻辑操作的结果而变化。当对状态寄存器进行写操作时，状态寄存器的内容恒等于操作数的内容，这保证了状态寄存器内容易于被保护和恢复，在系统复位时，状态寄存器的内容为 0。



R/W：读/写；xx：保留位

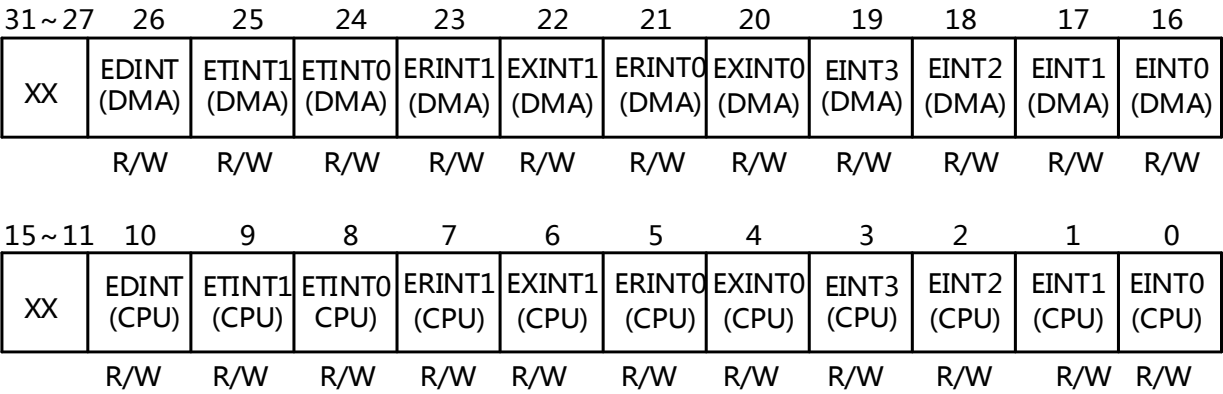
图 2.3 状态寄存器

表 2.4 状态寄存器各位的名称和功能

位	名称	功能
0	C	进位标志位。整数加法时，如果最高有效位发生进位，则 C 置位；整数减法时，如果最高有效位发生借位，则 C 置位；若上述两种情况不发生，C 清零。浮点和逻辑运算不影响进位标志
1	V	溢出标志位。对于整数运算，如果结果不符合目的操作数的特定格式（二进制补码），V 置位，否则 v 清零；对于浮点运算，如果结果的指数大于 127，V 置位，否则 V 清零；逻辑操作始终 V 清零
2	Z	零标志位。对于逻辑、整数和浮点运算，输出为零时 Z 置位，否则 Z 清零
3	N	负数标志位。对于逻辑操作，将输出值最大有效位赋予状态位 N；对整数和浮点运算，如果结果为负，N 置位，否则 N 清零；零作为正值
4	UF	浮点下溢出标志位。如果结果的指数小于-128，则发生浮点溢出，此时 UF 置位，否则 UF 清
5	LV	锁存溢出标志位。溢出标记 V 置位时 LV 置位，否则 LV 不改变；LV 只能在处理器复位或修正状态寄存器 ST 时清零
6	LUF	锁存浮点下溢出标志。浮点下溢出标记 UF 置位时 LUF 置位，否则 LUF 不改变，LUF 只能在处理器复位或修正状态寄存器 ST 时清零
7	OVM	溢出方式标志位。这个标志只影响整数操作，如果 OVM=0，则关闭溢出方式，对溢出的数不作任何处理；如果 OVM=1，则根据正、负、浮点溢出情况分别赋最大或最小值。以正方向溢出的结果被置为 32 位二进制补码的最大正数（7FFFFFFh），以负方向溢出的整数被置为 32 位二进制补码的最大负数（80000000h）。注意：V 和 LV 的功能不依赖于 OVM 的设置
8	RM	重复方式标志位。如果 PC 值在块重复或单指令重复过程中被修改了，则 RM=1，否则为 0
9	保留	保留位。读作 0，写不起作用
10	CF	Cache 冻结位。当 CF=1 时，Cache 被冻结，即不更新；如果 Cache 被允许（CE=1），则允许从 Cache 取数，但不进行 Cache 状态的修改。如果 CF=0，则允许 Cache 刷新。不论 CF=0 或 1，当 CC=1 时，都允许清除 Cache，另外，复位时该位被写为 0，复位后为 1
11	CE	Cache 允许位。当 CE=1 时，允许 Cache 更新；当 CE=0 时，禁止 Cache，Cache 不被修改，也不允许从 Cache 中取指。但当 CC=1 时，允许清除 Cache。CE 和 CF 是关联的，CE=0，Cache 不工作；CE=1 时，若 CF=0，Cache 工作，若 CF=1，Cache 虽工作，但不刷新
12	CC	Cache 清除。当 CC=0 时使 Cache 清零。在该位被写入之后总是被清零，因此读出总是为 0。复位时，该位为 0
13	GIE	全局中断允许位。如果 GIE=1，CPU 响应被允许的中断；如果 GIE=0，CPU 不响应这个被允许的中断。其中断包括 INTO~INT3、串口中断、定时器中断和陷阱
14~15	保留	读作 0
16~31	保留	读作 0

2.1.8 中断使能寄存器 IE

CPU/DMA 中断使能寄存器 IE(Interrupt Enable) 是一个 32 位的寄存器，结构如图 2.4 所示。CPU 中断使能位位于第 10~ 0 位，DMA 中断使能位位于 26 ~16 位，CPU/DMA 中断使能寄存器中的某位为 1 则使能相应的中断，若为 0 则禁止相应的中断。复位时，这个寄存器被写为 0。表 2.5 定义了该寄存器各位的名称和功能。



R/W：读/写；xx：保留位

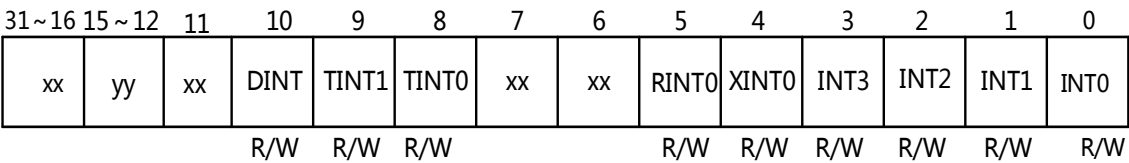
图 2.4 CPU/DMA 中断使能寄存

表 2.5 CPU/DMA 中断使能寄存器各位的名称和功能

位	名称	功能
0	EINT0	外部中断 0 (CPU)
1	EINT1	外部中断 1 (CPU)
2	EINT3	外部中断 2 (CPU)
3	EINT3	外部中断 3 (CPU)
4	EXINT0	串行口 0 发送中断 (CPU)
5	ERINT0	串行口 0 接收中断 (CPU)
6	EXINT1	串行口 1 发送中断 (CPU)
7	ERINT1	串行口 1 接收中断 (CPU)
8	ETINT0	定时器 0 中断 (CPU)
9	ETINT1	定时器 1 中断 (CPU)
10	EDINT	CPU 中断
11~15	保留	未定义
16	EINT0	外部中断 0 (DMA)
17	EINT1	外部中断 1 (DMA)
18	EINT2	外部中断 2 (DMA)
19	EINT3	外部中断 3 (DMA)
20	EXINT0	串行口 0 发送中断 (DMA)
21	ERINT0	串行口 0 接收中断 (DMA)
22	EXINT1	串行口 1 发送中断 (DMA)
23	ERINT1	串行口 1 接收中断 (DMA)
24	ETINT0	定时器 0 中断 (DMA)
25	ETINT1	定时器 1 中断 (DMA)
26	EDINT	允许 DMA 中断
27~ 31	保留	未定义

2.1.9 中断标志寄存器 IF

CPU 中断标志寄存器（IF）是一个 32 位的寄存器，结构如图 2.5 所示。当一个中断发生时，IF 寄存器中相应的位被置 1，通过软件对某一位写 1 也能引起该位相对应的中断，前提是 IE 寄存器中已经设置了使能该位中断。如果把 0 写入到 IF 中的某位，则清除相应的中断。复位时，该寄存器的内容为 0。表 2.6 列出了 CPU 中断标志寄存器各位的名称和功能。



R/W：读/写；xx：保留位，读操作时为 0；yy：保留位，复位时置 0

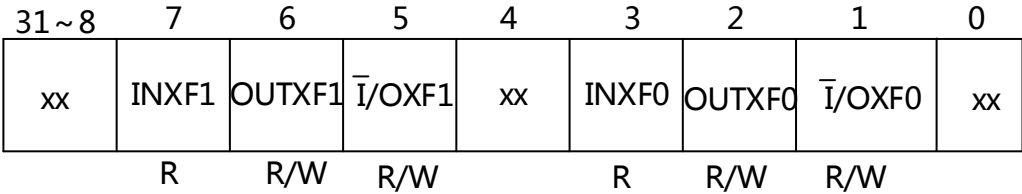
图 2.5 CPU 中断标志寄存器

表 2.6 IF 寄存器各位的名称和功能

位	名称	功能
0	INT0	外部中断 0 标志
1	INT1	外部中断 1 标志
2	INT2	外部中断 2 标志
3	INT3	外部中断 3 标志
4	XINT0	串行口 0 发送中断标志
5	RINT0	串行口 0 接收中断标志
6	保留	未定义
7	保留	未定义
8	TINT0	定时器 0 中断标志
9	TINT1	定时器 1 中断标志
10	DINT	DMA 中断标志
11~31	保留	未定义

2.1.10 通用 I/O 引脚 XF0 和 XF1

通用 I/O 引脚 XF0 和 XF1 可配置为输入或输出,也可以对它们进行读出和写入,由 I/O 控制寄存器(IOF) 控制, I/O 控制寄存器是一个 32 位的寄存器,通过对该寄存器的写操作可以控制 XF0 和 XF1 的功能模式,复位时,该寄存器的内容为 0,结构如图 2.6 所示,表 2.7 列出了该寄存器各位的名称和功能。



R：读；R/W：读/写；xx：保留位

图 2.6 I/O 控制寄存器

表 2.7 IOF 各位的名称和功能

位	名称	功能
0	保留	读作 0
1	$\bar{I}/OXF0$	如果 $\bar{I}/OXF0=0$, 则 XF0 被构置为通用目的输入引脚：如果 $\bar{I}/OXF0=1$, 则 XF0 被构置为通用目的输出引脚
2	OUTXF0	XF0 上的数据输出
3	INXF0	XF0 上的数据输入, 写无效
4	保留	读作 0
5	$\bar{I}/OXF1$	如果 $\bar{I}/OXF1=0$, 则 XF1 被构置为通用目的输入引脚：如果 $\bar{I}/OXF1=1$, 则 XF1 被构置为通用目的输出引脚
6	OUTXF1	XF1 上的数据输出
7	INXF1	XF1 上的数据输入, 写无效
8~31	保留	读作 0

2.1.11 复位功能

使用 $\overline{RESET}$ 时, CPU 会尝试安全的退出可能存在的任何挂起的读取或写入操作进程, 这需要 10 个 CPU 周期, 然后地址、数据和控制引脚将进入一个无效或高阻状态。

使用 $\overline{RESET}$ 和 $\overline{SHZ}$ 时, 器件会立即进入复位状态并保持引脚高阻抗模式。在复位设置为高电平之前, 应禁用 $\overline{SHZ}$ 至少 10 个 CPU 周期。 $\overline{SHZ}$ 可以在上电排序期间使用, 以防止未定义的地址、数据和控制引脚, 避免系统冲突。

2.1.12 中断简介

AVP32VC33 支持多种内部和外部中断, 可用于各种应用。内部中断可通过定时器 0、定时器 1、串行口和 DMA 控制器产生；外部中断共四个, 其引脚名称为 $\overline{INT0} \sim \overline{INT3}$, 外部中断可由 EDGEMODE 引脚控制为沿触发方式或者是低电平触发方式。这些中断可自动设定优先级, 允许不同中断同时发生并按照预先设定的顺序进行服务。

2.1.13 片内 ROM 引导程序操作

当 $\overline{\text{MCBL}}/\overline{\text{MP}}=1$ 时，出厂时固化在片内的引导程序被映射到地址范围为 $0h\sim FFFh$ 的 ROM 中，DSP 上电复位后，该程序自动执行，将存放在 ROM 中的用户程序装载到规定的 DSP 的片内存储器内。当 $\overline{\text{MCBL}}/\overline{\text{MP}}=0$ 时，ROM 未被映射到存储空间中， $0h\sim 03Fh$ 单元包含的中断矢量、陷阱矢量和保留单元，通过外部存取信号（ $\overline{\text{STRB}}$ ）进行存取。

2.1.14 JTAG

AVP32VC33 执行标准 IEEE1149.1UJTAG 接口。通过这个标准可对芯片的硬件电路进行边界扫描和故障检测，具有以下 JTAG 引脚：

- 1、TCK：测试时钟输入；
- 2、TDI：测试数据输入，数据通过 TDI 输入 JTAG 口；
- 3、TDO：测试数据输出，数据通过 TDO 从 JTAG 输出；
- 4、TMS：测试模式选择，TMS 用来设置 JTAG 口处于某种特定的测试模式；
- 5、 $\overline{\text{TRST}}$ ：测试复位时的可选引脚，输入引脚，低电平有效；
- 6、AVP32VC33 除了以上引脚外，还有 EMU0 和 EMU1 引脚，用以激活 JTAG 口，实现仿真功能，这两个引脚通过上拉电阻直接接到 3.3V 电源 DVDD 上。

AVP32VC33 通过以上的 JTAG 口引脚将程序下载到芯片中，对硬件进行仿真和测试。

2.1.15 定时器简介

AVP32VC33 有 2 个 32 位的定时器（即定时器 0 和定时器 1），每个定时器有两种信号模式以及内部或外部时钟模式，使用定时器可以定时一定的周期脉冲或方波信号输出，也可以对外部事件计数。每个定时器都由一个 32 位计数器、一个比较器、一个输入时钟选择器、一个脉冲发生器以及支撑硬件组成。每个定时器由三个寄存器控制它们的工作模式，分别为定时器全局控制寄存器、周期寄存器和计数器，两个定时器各对应一个引脚，分别记为 TCLK0 和 TCLK1，每个引脚可单独设置为定时器方式或通用 I/O 方式。

2.1.16 串行口简介

AVP32VC33 有一个独立的双工串行口，能够同时发送和接收字长为 8、6、24 和 32 位的数据。串行口的时钟源可以通过串口定时器、周期寄存器由内部产生，也可以由外部时钟提供。其中内部产生的时钟是 CPU 工作时钟输出频率 f_{H1} 的 2 分频。串行口引脚共 6 个，分别为 CLKR0、DR0、FSR0、CLKX0、DX0、FSX0，其中发送端口和接收端口各 3 个引脚。

2.1.17 DMA 控制器简介

DMA 控制器是 AVP32VC33 的可编程外围设备，支持单通道的 DMA 传输，能传输大批量数据至存储器的任一地址单元，并且进行这些操作时并不影响 CPU 的传输速度，进而改进系统性能，提高了数据处理能力。DMA 操作可以一直工作，也可以由内部（片内定时器、串行口）中断和外部中断触发。

2.2 存储器管理

AVP32VC33 存储器的分配方式有两种，一种是微计算机引导方式，一种是微处理器方式，方式的选择由 MCBL/MP 引脚上的电平决定。如果将该引脚接为高电平，则为微计算机引导方式，该方式下存储器分配情况如图 2.7；如果将该引脚接为低电平，则为微处理器方式，该方式下存储器分配情况如图 2.8。微计算机引导方式与微处理器方式的最大差异是是否允许引导，如果允许引导则为微计算机引导方式，如果不允许引导则为微处理器方式。

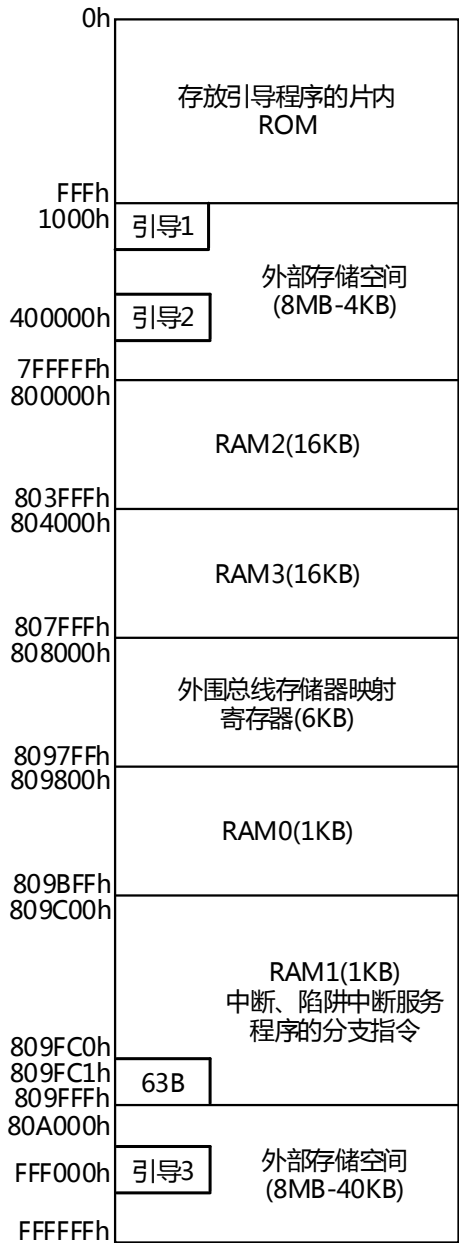


图 2.7 微计算机引导方式

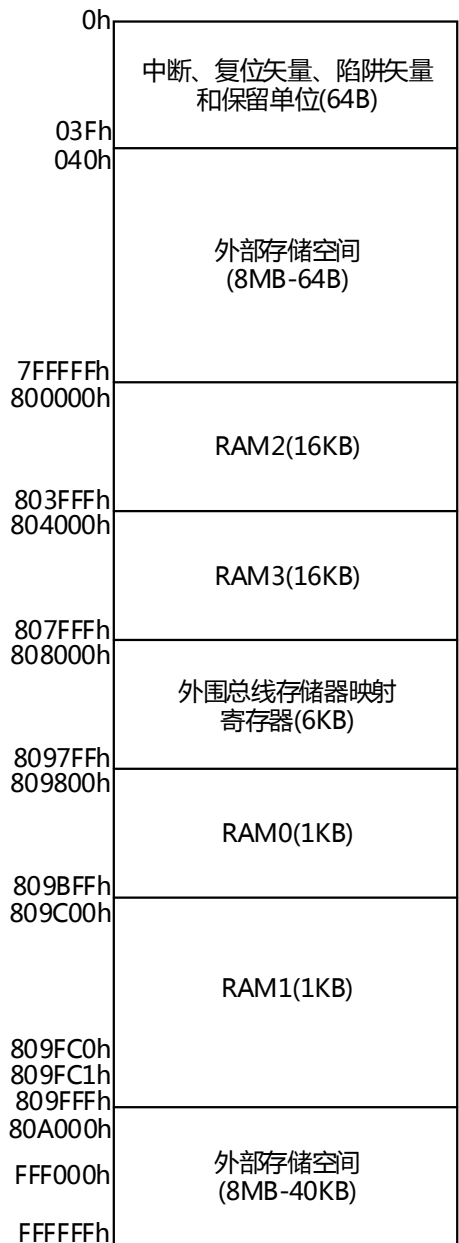


图 2.8 微处理器引导方式

2.2.1 中断向量分配存储器地址

在微计算机引导方式下,中断服务程序和陷阱的分支指令被分配在 809FC1~809FFFh 所指向的 63 个字节中,各向量的具体地址如图 2.9 所示。地址值越低,其优先级越高,由图可知,外部中断 0 的优先级最高。809FC1~809FFFh 所指向的 63 个字节中,809FC7h、809FC8h、809FCCh~809FDFh、809FFCh~809FFFh 等 26 个字节为保留字,中断服务程序的分支地址共 37 个。

在微处理器方式下,复位、中断和陷阱向量被分配在 0~3Fh 的空间内,各向量的具体地址如图 2.10 所示,其中陷阱是指通过软件触发的中断,向量的地址值越低,其优先级越高。由图可知,在所有中断向量中,复位的优先级最高。0~3Fh 所指的 64 个地址中,07h、08h、0Ch~1Fh、3Ch~3Fh 等 26 个字节为保留字,中断向量共 38 个。

809FC1h	INT0
809FC2h	INT1
809FC3h	INT2
809FC4h	INT3
809FC5h	XINT0
809FC6h	RINT0
809FC7h	保留
809FC8h	保留
809FC9h	TINT0
809FCAh	TINT1
809FCBh	DINT
809FCCh	保留
809FDFh	保留
809FE0h	TRAP0
	⋮
809FFBh	TRAP27
809FFCh	保留
809FFFh	

图 2.9 微计算机引导方式

00h	RESRT
01h	INT0
02h	INT1
03h	INT2
04h	INT3
05h	XINT0
06h	RINT0
07h	保留
08h	保留
09h	TINT0
0Ah	TINT1
0Bh	DINT
0Ch	保留
1Fh	保留
20h	TRAP0
	⋮
3Bh	TRAP27
3Ch	保留
3Fh	

图 2.10 微处理器引导方式

2.2.2 外围寄存器地址分配

外围寄存器地址被分配在以地址 808000h 开始的单元中，如图 2.11 所示。这里所述的外围寄存器是相对于 CPU 寄存器而言的，即这些寄存器与 CPU 无直接关联，是分配在 16MB 存储空间内的。外围寄存器也可以理解为控制外围设备的寄存器，这些外围设备包括 DMA、定时器 0、定时器 1、串行口和外部主总线，它们虽然在 DSP 芯片内部，但相对于算术逻辑单元(ALU)、辅助寄存器运算单元（ARAU）和乘法器而言，它们与 CPU 并无直接的关联，因而将其称为外围设备。

808000h	DMA全局控制器寄存器
808004h	DMA源地址寄存器
808006h	DMA目的地址寄存器
808008h	DMA传输计数器
808020h	定时器0全局控制器寄存器
808024h	定时器0计数器
808028h	定时器0周期寄存器
808030h	定时器1全局控制器寄存器
808034h	定时器1计数器
808038h	定时器1周期寄存器
808040h	串行口全局控制器寄存器
808042h	发送端口控制器寄存器
808043h	接收端口控制器寄存器
808044h	收发定时器控制器寄存器
808045h	收发计算器
808046h	收发周期寄存器
808048h	数据发送寄存器
80804Ch	数据接收寄存器
808064h	主总线控制寄存器

图 2.11 外围寄存器的地址分配

2.2.3 指令高速缓存器

指令高速缓冲存储器 (Cache) 用于存放最近使用过的重复率比较高的指令代码，对于最近出现过的指令，可直接从 Cache 中取指，从而减少取指所占用的时间，加速指令的运行。Cache 能以完全自动的方式工作而无需用户干涉，用户唯一需要做的是在状态寄存器中设置好是否允许 Cache 工作的相关控制位。

Cache 的结构如图 2.12 所示，由 64 个 32 位字的 RAM 组成，这 64 个字分成两个段，每段 32 个字，这两个段分别称为段 0 和段 1，每段各有一个 19 位的段基址 (Segment Start Address, SSA) 寄存器，段中的每个字对应一个标志位 P，简称 P 标志，该标志位是 32 位指令字以外的一个单独的位，用以表明指令代码是否存在。

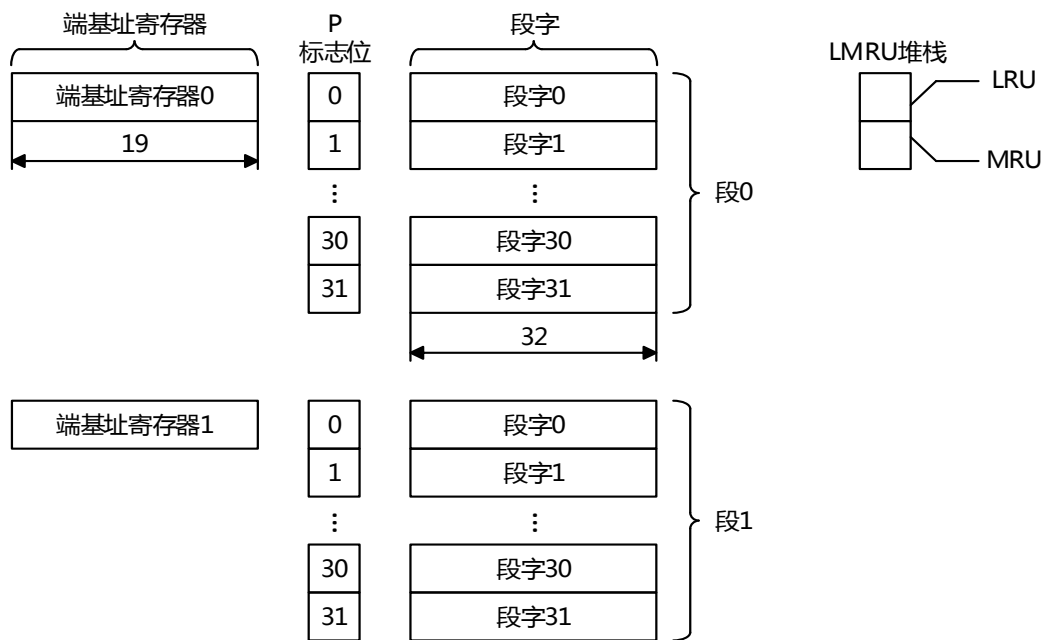


图 2.12 Cache 结构

2.3 中断管理

AVP32VC33 可由定时器、串行口和 DMA 产生内部中断，由 INT0~INT3 四个引脚触发外部中断 0~3。当中断发生时，会对产生中断的中断源进行应答和处理，程序会跳到相应的中断服务子程序的入口地址执行中断程序，但要对产生的中断进行相应的应答，需要与中断相关的三个寄存器进行写操作，三个寄存器分别是状态寄存器（ST）、中断使能寄存器（IE）、中断标志寄存器（IF），它们相应的位可控制中断的操作，操作如下：

- (1) 状态寄存器（ST）中的全局中断使能位 GIE 置 1 时，CPU 响应相应的中断：该置 0 时，所有中断禁止。
 - (2) 通过对中断使能寄存器（IE）的每一位操作，可以使能或禁止 CPU、DMA、串行口、定时器等中断。
 - (3) 中断标志寄存器(IF)包含了中断标志位，指出相应的中断是否已经发生。
- 当 CPU 中断发生时，至少需要满足两个条件，一是将状态寄存器 ST 中的 GIE 位置 1，对所有的中断进行全局使能；二是设置 IE 寄存器，对相应的中断进行使能。

2.3.1 AVP32VC33 中断向量表

中断向量包括了中断服务子程序的入口地址，AVP32VC33 可工作在微计算机引导模式或者微处理模式下，两种模式下的中断服务子程序的入口地址是不同的，表 2.8 和表 2.9 列出了两种工作模式下的中断向量表及其地址。

表 2.8 AVP32VC33 微计算机引导模式下的中断及陷阱向量地址

地址	名称	功能
809FC1H	INT0	外部中断 INT0
809FC2H	INT1	外部中断 INT1
809FC3H	INT2	外部中断 INT2
809FC4H	INT3	外部中断 INT3
809FC5H	XINT0	串口发送缓冲器空时产生内部中断
809FC6H	RINT0	串口接收缓冲器满时产生内部中断
809FC7H ~ 809FC8H	(保留)	
809FC9H	TINT0	定时器 0 内部中断
809FCAH	TINT1	定时器 1 内部中断
809FCBH	DINT	DMA 控制器内部中断
809FCCH ~ 809FCDH	保留	
809FE0H ~ 809FFBH	TRAP0 ~ TRAP27	陷阱 0 ~ 27 指令内部中断
809FFCH ~ 809FFFH	TRAP28 ~ TRAP31 (保留)	

表 2.9 AVP32VC33 微处理器模式下的复位、中断及陷阱向量地址

地址	名称	功能
00H	RESET	外部复位信号
01H	INT0	外部中断 $\overline{\text{INT0}}$
02H	INT1	外部中断 $\overline{\text{INT1}}$
03H	INT2	外部中断 $\overline{\text{INT2}}$
04H	INT3	外部中断 $\overline{\text{INT3}}$
05H	XINT0	串口发送缓冲器空时产生内部中断
06H	RINT0	串口接收缓冲器满时产生内部中断
07H ~ 08H	(保留)	
09H	TINT0	定时器 0 内部中断
0AH	TINT1	定时器 1 内部中断
0BH	DINT	DMA 控制器内部中断
0CH ~ 1FH	保留	
20H ~ 3BH	TRAP0 ~ TRAP27	陷阱 0 ~ 27 指令内部中断
3CH ~ 3FH	TRAP28 ~ TRAP31 (保留)	

2.3.2 中断优先级

当在同一个时钟周期发生多个中断时，CPU 按照中断的优先级来区分中断执行次序。根据中断向量表的位置对中断优先级进行设置，偏移量越靠近向量表基址的中断，其优先级别越高，表 2.10 列出来了中断的优先级顺序。

表 2.10 复位及中断向量优先级

复位/中断	向量地址	优先级	功能
RESET	0	0	外部复位信号
INT0	1	1	外部中断 $\overline{\text{INT0}}$
INT1	2	2	外部中断 $\overline{\text{INT1}}$
INT2	3	3	外部中断 $\overline{\text{INT2}}$
INT3	4	4	外部中断 $\overline{\text{INT3}}$
XINT0	5	5	串行口 0 发送中断
RINT0	6	6	串行口 0 接收中断
保留	7	7	保留
保留	8	8	保留
TINT0	9	9	定时器 0 中断
TINT1	0A	10	定时器 1 中断
DINT	0B	11	DMA 通道中断

2.3.3 中断处理过程

AVP32VC33 允许 CPU 和 DMA 并行响应处理中断，中断处理流程图如图 2.13 所示。在状态寄存器 ST 的 GIE 位和中断使能寄存器相应位都置 1 后，如果中断产生，中断标志寄存器 IF 相应的位置 1，然后处理器开始处理中断，在处理期间，IF 寄存器相应的标志位被清零，同时中断被全局禁止(GIE=0)，然后产生的中断的相应中断向量被加载到处理器，然后 CPU 开始执行中断服务子程序。若希望中断服务子程序可以被中断，那么当进入中断服务子程序后，设置 GIE=1，这样中断服务子程序可以被另外的中断暂停。

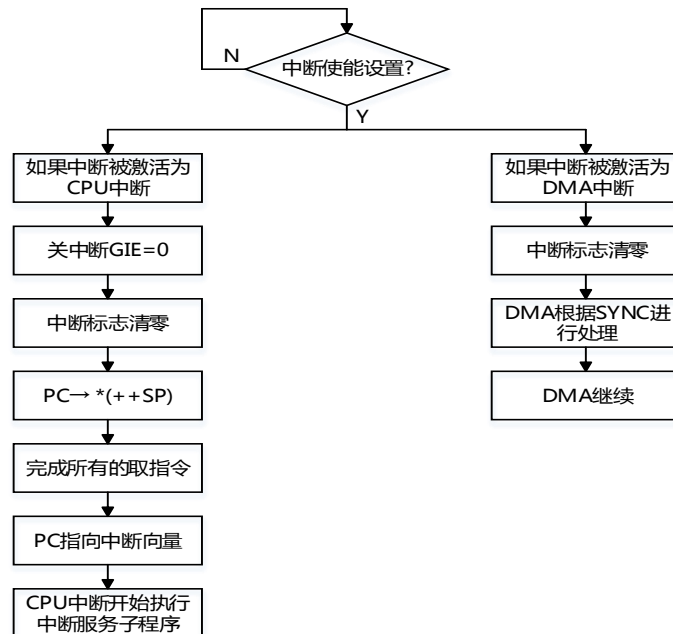


图 2.13 CPU 和 DMA 中断处理示意图

2.3.4 外部中断

AVP32VC33 支持 4 个外部中断 INT0~INT3，外部中断是在内部进行同步的，时钟 H1/H3 驱动 3 个触发器，通过 H1 信号的下降沿或脉冲持续时间对中断 INT0~INT3 进行触发，如果外部中断为电平触发方式时，低电平的持续时间至少为一个 H1/H3 周期，但不能超过两个 H1/H3 周期，否则会被认为是多个中断。如果中断有效，则中断输出将置位相应的中断标志寄存器 IF。

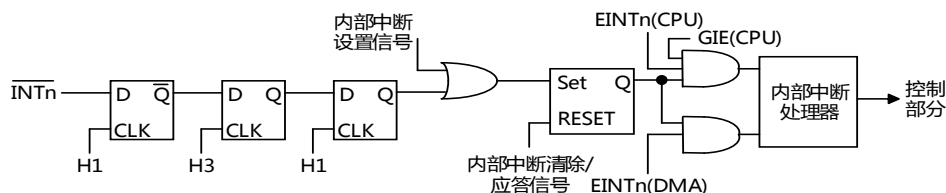


图 2.14 中断逻辑功能框图

当有中断在同一个时钟周期到来时，通过选择其中之一先执行来确定中断优先级。当中断发生时，状态寄存器 ST 的 GIE 位被复位为 0，禁止任何到来的中断。由此阻止了其他的中断 INT0~INT3 取得程序控制权，直到 ST 的 GIE 位被设置为 1 后。中断子程序返回时，RETI 或 RETIcond 指令将 ST 的 GIE 位设置为 1。外部中断可由 EDGEMODE 引脚控制为沿触发方式或者是低电平触发方式，EDGEMODE=1 时，为电平触发，EDGEMODE=0 时，为沿触发。

3 外围设备

AVP32VC33 有两个定时器、一个串行口和一个 DMA(Direct Memory Access)，它们是通过外围总线上的存储器映射寄存器来管理的，因而将其称为外围设备。管理这些外围设备的寄存器共 18 个，其中定时器 0 和定时器 1 各有 3 个寄存器，串行口有 8 个寄存器，DMA 有 4 个寄存器。

3.1 定时器

AVP32VC33 有两个 32 位定时器，两个定时器的功能、结构、控制器的内容和用法是完全相同的，所不同的仅是地址，使用定时器可以定时一定的周期脉冲或方波信号输出，也可以对外部事件计数。使用内部时钟时，定时器可以发出信号作为外部设备（如 A/D 转换器）的控制信号，或者可以中断 DMA 控制器以开始一次数据传输，定时器中断是内部中断的一种。在外部时钟的作用下，定时器可以对外部事件进行计数，并且在指定的事件发生后中断 CPU。

每个定时器都由一个 32 位的计数器、一个比较器、一个输入时钟选择器、一个脉冲发生器及支撑硬件组成，如图 3.1 所示。定时器的工作过程为：时钟选择器用于选择计数器的时钟源，该时钟可以采用 DSP 内部的 CPU 时钟，也可以采用外部的时钟，还可以对输入的内部或外部时钟进行反向。当采用内部时钟时，输入计数器的最高频率是 CPU 时钟频率 F_{H1} 的 1/2，或周期的 2 倍，记作 $2T_{H1}$ ；当采用外部时钟时，外部时钟的最高频率是 CPU 时钟频率 F_{H1} 的 1/2.6。周期寄存器用于存放需要计数的个数，计数器用于累计实际的计数值，比较器对周期寄存器和计数器的值进行比较，当二者相等时，可通过脉冲发生器产生定时器的输出信号，同时将信息反馈到定时器的全局控制寄存器的状态位 TSTAT。

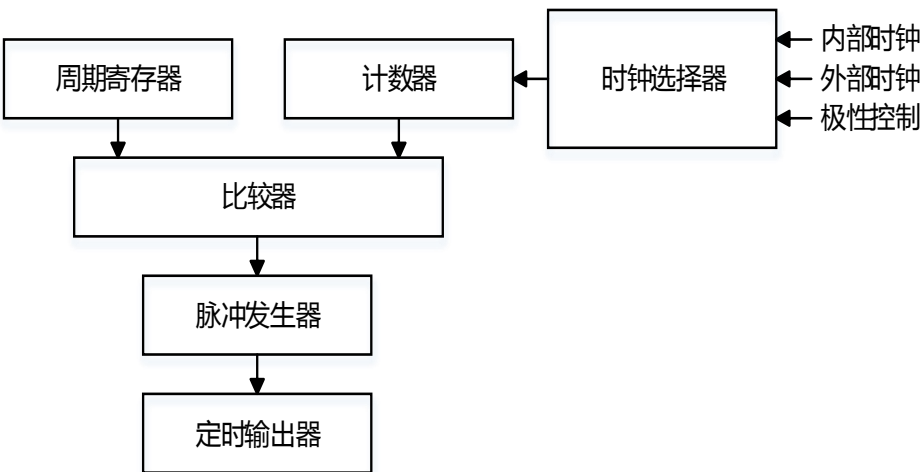


图 3.1 定时器的原理框图

3.1.2 定时器相关寄存器

通过对三个寄存器的读写操作，可以控制定时器的运行模式，这三个寄存器分别是全局控制寄存器、周期寄存器和计数寄存器，其功能如下：

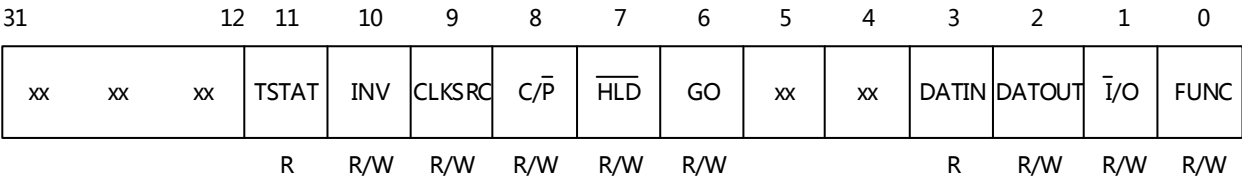
（1）全局控制寄存器（Global-control Register）。该寄存器确定了定时器的工作模式，监控定时器的状况，控制定时器 I/O 引脚的功能。

（2）周期寄存器（Period Register）。该寄存器存放预定的计数值，用来指定定时器发出信号的频率。

（3）计数寄存器（Counter Register）。该寄存器包含了不断增长的计数器的当前值。可以在输入时钟上升沿或下降沿到来的时候增加定时器的值。当它的值等于周期寄存器的值时，计数器被清零，并产生一个内部中断。脉冲发生器可产生两种外部时钟信号的脉冲或时钟。

3.1.3 定时器全局控制寄存器

定时器全局控制寄存器是一个 32 位的寄存器，其结构如图 3.2 所示，该寄存器中只有 10 位有效位，其余均为保留位。



R：只读；R/W：读/写；xx：保留位；

图 3.2 定时器全局控制寄存器的结构

这个 32 位的寄存器中，第 0~3 位是定时器功能和定时器引脚控制位，第 4~7 位为定时器运行和保持控制位，第 8~11 位是定时器的时钟源、时钟极性、输出极性和定时器的状态检测位。复位时，除了 DATIN（被置于 TCLK 读入的值）外，所有的位都被置 0。每一位的具体功能如表 3.1 所示。

表 3.1 定时器全局控制寄存器各控制位的功能

位	名称	复位值	功能描述
0	FUNC	0	定时器引脚的功能控制位。FUNC=0, TCLK(TCLK0 或 TCLK1)为通用 I/O 脚。FUNC=1, TCLKx 为定时器引脚。但作为定时器引脚时，也有输出、输入功能，例如，输入外部的时钟或输出定时信号。FUNC、I/O 及 CLKSRC 配合才能完成一种确定的功能
1	I/O	0	输入或输出控制位。I/O=0 则定时器引脚为输入脚，否则为输出脚
2	DATOUT	0	输出控制位。用以驱动 TCLK 引脚和定时器，也可以把 DATOUT 作为定时器输入
3	DATIN	0	输入状态位。跟踪 TCLKx 的状态，写无效
4 ~ 5	保留	00	保留位。读作 0
6	GO	0	定时器运行控制位。GO=0 对定时器没有影响。GO=1 且 HLD =0 时，计数器在定时器时钟的下一个上升沿被清零并从 0 开始累加计数值。启动计数器计数后，GO 也自动变为 0(GO=0 不影响已经启动了的定时器)

7	$\overline{\text{HLD}}$	0	计数器保持控制位。HLD =0，禁止计数器计数，使其保持当前状态以便当 HLD 变为 1 时，使计数器能够由现行状态继续计数。当定时器处于保持状态时，定时器的寄存器能够被读出或被修正。RESET 的优先权高于 HLD。表 3.2 绘出了 GO 和 HLD 的状态对定时器的影响
8	$\overline{\text{C/P}}$	0	时钟或脉冲方式控制位。C/ P =1 时为时钟方式，TSTAT 标志信号和外部输出为占空比是 50% 的周期信号；C/ P =0 时为脉冲方式，状态标志和外部输出将在每个定时器周期内激发 H1 周期。
9	CLKSRC	0	定时器的时钟源控制位。CLKSRC=1，计数器的时钟来自 DSP 内部的 CPU 时钟，其频率等于 H1 频率一半，反向控制位 INV 对内部时钟源无影响；CLKSRC=0;计数器的计数时钟来自 TCLKx 引脚上的外部信号，其最高频率为 CPU 频率的 1/2.6。FUNC 和 CLKSRC 之间的关系见后图
10	INV	0	反相控制位。若使用了外部时钟源且 INV=1，则计数器的输入时钟和定时器的输出被反向；若 INV=0，不反相。当 TCLKx 工作在通用 I/O 口方式下，INV 位对 I/O 没有影响
11	TSTAT	0	定时器的状态标志位。只读，写无效。它跟踪未反相的 TCLKx 引脚的输出；此位从 0 变为 1 时，CPU 中断置位，写无效。
12 ~ 31	保留	0 ~ 0	保留位。读作 0

表 3.2 GO 和 HLD 对定时器的影响

GO	$\overline{\text{HLD}}$	定时器状态
0	0	定时器处于保持状态，不复位
0	1	定时器继续以前的状态
1	0	定时器处于保持状态，计数器被清零，但 GO 位不被清零，直到定时器取消保持
1	1	定时器复位并开始运行

3.1.4 定时器周期寄存器和计数寄存器

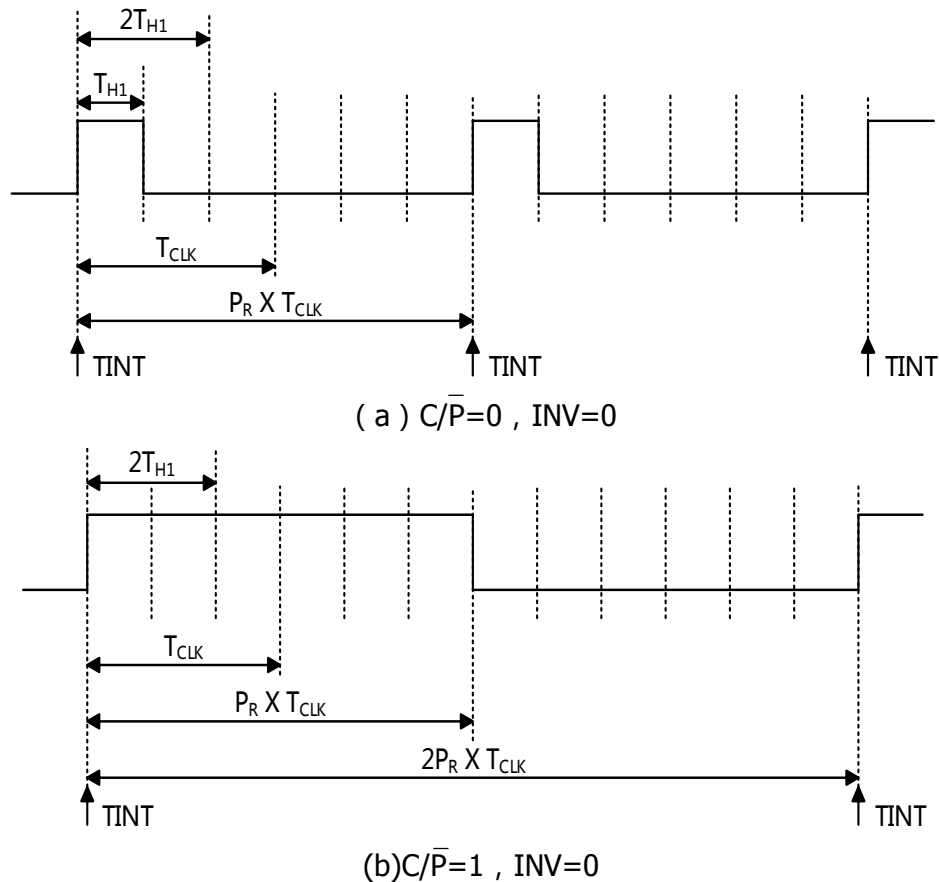
定时器周期寄存器和计数器都是 32 位的寄存器，复位时，它们的内容都为 0。周期寄存器存放预定的计数值，指定定时器产生信号的频率；计数器则对输入的时钟或脉冲由零开始计数，当计数器的计数值等于周期寄存器预存的数值时，计数器的内容回到 0。

有些特殊情况会对定时器的工作状态产生以下影响：

- (1) 周期寄存器和计数寄存器的值都为 0。这时，定时器的操作取决于所选择的 C/ $\overline{\text{P}}$ 方式。在脉冲方式下 (C/ $\overline{\text{P}}$ =0), TSTAT 置位并保持置位；在时钟方式下 (C/ $\overline{\text{P}}$ =1)，定时器输出的时钟周期为 2T_{H1}，这是定时器能够输出的最高频率。
- (2) 周期寄存器为 0 但计数器不为 0。这时计数器计数到最大值 (0FFFFFFh) 后变为 0，以后仍按时钟方式或脉冲方式输出相应的定时信号。
- (3) 如果计数器的设置值大于周期寄存器的设置值，则计数器在计数过程中产生溢出，达到它的 32 位最大值 (0FFFFFFh)，然后加到 0 并重新计数，直到计数值与周期寄存器的设置值相等。

3.1.5 定时器脉冲发生器

定时器脉冲发生器能够产生几种不同的外部信号，这些信号可以用 INV 反向。产生的信号分为两种基本方式：脉冲方式和时钟方式。如果采用内部时钟源，在脉冲方式下，定时器输出的脉冲宽度为 T_{H1} ，即一个 CPU 周期的宽度；在时钟方式下，定时器输出占空比为 50% 的波形，如图 3.3 所示。无论哪种方式，定时器所采用的时钟可以来自内部或外部，内部时钟源的频率为 $f_{H1}/2$ ，外部时钟源的最高频率为 $f_{H1}/2.6$ 。



T_{H1} : CPU 的时钟周期； T_{CLK} ：所采用的时钟源的周期，可以是内部和外部的时钟源

P_R ：周期寄存器的值； $TINT$ ：定时器中断，在上升沿发出中断请求

图 3.3 定时器在选择时钟和脉冲方式下的输出信号

定时器定时输出信号的频率是由定时器输入时钟的频率和周期寄存器共同确定的，下面等式对于内部或外部时钟源都是适用的：

$$T_P = P_R \times T_{CLK} \quad (3.1)$$

$$T_C = 2P_R \times T_{CLK} \quad (3.2)$$

其中 T_P 和 T_C 分别表示以脉冲方式和时钟方式输出的定时器信号的周期。当采用内部时钟源时，满足关系式：

$$T_{CLK} = 2 T_{H1} \quad (3.3)$$

3.1.6 定时器运行模式

主要通过对全局控制寄存器中的 CLKSRC、FUNC 和 I/O 三位设置来控制定时器的运行模式，下面定义了 4 种定时器运行模式：

(1) 方式 1：CLKSRC=1 且 FUNC=0

该模式下定时器的时钟源来自内部时钟，定时器引脚 TCLK 为通用 $\bar{I}/O$ 口工作模式，由于 $\bar{I}/O$ 口又分为输入和输出，所以根据 $\bar{I}/O$ 的设置又分为两种模式：

a、 $\bar{I}/O=0$ ，TCLK 被配置为通用输入脚，它的状态能从 DATIN 上读出，DATOUT 对 TCLK 和 DATIN 没有影响；

b、 $\bar{I}/O=1$ ，TCLK 被配置成通用输出脚，DATOUT 通过 TCLK 输出，并可从 DATIN 读出；

(2) 方式 2：CLKSRC=1 且 FUNC=1

该模式下定时器的时钟源来自内部时钟，定时器引脚 TCLK 为定时器引脚，定时结果从 TCLK 输出并由 INV 反相，在 TCLK 上的输出能够从 DATIN 读出；

(3) 方式 3：CLKSRC=0 且 FUNC=0

该模式下定时器的时钟源来自外部信号，定时器引脚 TCLK 为通用 I/O 口工作模式

a、 $\bar{I}/O=0$ ，定时器的输入信号来自 TCLK 引脚，该信号能用 INV 反相，且能够在 DATIN 上读出

b、 $\bar{I}/O=1$ ，TCLK 为输出脚，TCLK 和定时器都由 DATOUT 驱动，DATOUT 从 0 变为 1 时计数器加 1，INV 对 DATOUT 没有影响，DATOUT 的值能够在 DATIN 读出；

(4) 方式 4：CLKSRC=0 且 FUNC=1

该模式下定时器的时钟源来自外部信号，TCLK 驱动定时器

(1) 如果 INV=0，TCLK 的上升沿计数器加 1；

(2) 如果 INV=1，TCLK 的下降沿计数器加 1。TCLK 的值能在 DATIN 中读出。

3.1.7 定时中断

当定时器全局控制寄存器中的状态位 TSTAT 从 0 跳变到 1 时，就会产生定时器中断。在脉冲方式和时钟方式下，定时器的中断周期是不同的，分别由式 (3.1) 和 (3.2) 给出，这时，上述两式中的 T_p 和 T_c 可分别理解为定时器在脉冲方式和时钟方式下的中断周期。这里需要强调的是：定时器的周期寄存器和计数器一旦设置好后，不论采用内部时钟还是外部时钟，都将周而复始地计数，每当计数器的计数值与周期寄存器的值相等时，就自动产生中断，且计数器的内容自动恢复到 0，而周期寄存器的内容则一直保持不变，当重新设置了周期寄存器的值后，定时器的定时周期才改变。定时器触发的中断可以是 CPU 中断，也可以是 DMA 中断，是否允许定时器中断取决于 IE 寄存器。

3.1.8 定时器初始化

通过对定时器控制寄存器赋值操作，可对定时器进行初始化和重新配置。其步骤如下：

(1) 清除全局控制寄存器的 GO、HLD 位，使定时器处于停止状态，这只要对全局控制寄存器写 0 即可。注意：复位时定时器处于停止状态。

(2) 针对想要实现的定时器的工作方式或定时周期，分别设置全局控制寄存器、计数寄存器和周期寄存器的值。

(3) 对全局控制寄存器的 GO 和 $\overline{HLD}$ 位写 1，启动定时器

3.2 DMA 控制器

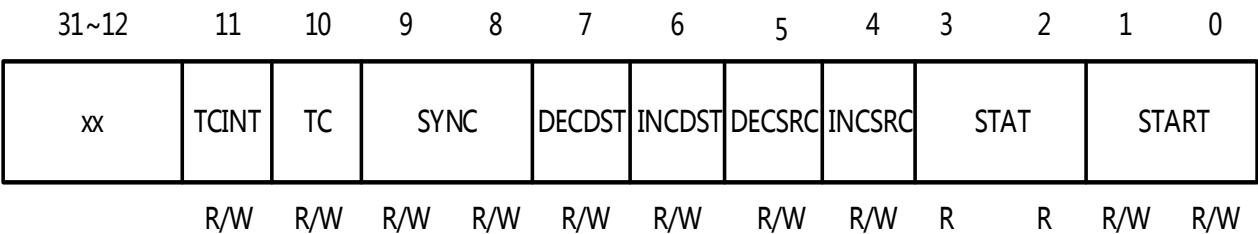
DMA 控制器是 AVP32VC33 的可编程外围设备，其特性如下：

- (1) 对处理器的任意存储单元进行输入 / 输出操作。例如，对片内存储器、片外存储器和片内串口进行输入 / 输出操作；
- (2) DMA 操作的同时并不影响 CPU 的传输速度；
- (3) 存储器到存储器的单 DMA 传输通道；
- (4) 源地址寄存器和目的地址寄存器自动增减；
- (5) 由内部和外部中断来同步数据传输；

AVP32VC33 的 DMA 控制器支持单通道的 DMA 传输，能够对处理器的任意存储单元进行输入/输出操作，并由 4 个寄存器控制其工作模式，4 个寄存器分别为：全局控制寄存器、源地址寄存器、目的地址寄存器和传输寄存器，DMA 操作可以一直工作，也可以由内部中断（片内定时器、串行口）和外部中断触发。

3.2.1 全局控制寄存器

全局控制寄存器控制 DMA 的操作并反映 DMA 在每一个周期内的变化状态，通过对全局控制器的特殊位的操作能够使得源地址和目的地址增加、减少或者同步。系统复位时，DMA 控制寄存器的所有位都置 0。图 3.4 表示了全局控制寄存器的位结构，表 3.3 定义了此寄存器各位的名称及其功能。



R:只读；R/W：读/写；xx：保留位，写无效，读作 0

图 3.4 DMA 全局控制寄存器结构

表 3.3 DMA 全局控制寄存器各位的功能及说明

位	名称	复位值	功能描述
0~1	START	00	控制 DMA 启动、停止，停止时不会丢失任何数据。这两位的组合功能见表 3.4
2~3	STAT	00	显示 DMA 在每一个机器周期内的变化状态，这两位的组合功能与 DMA 状态见表 3.5
4	INCSRC	0	若 INCSRC=1，则源地址每执行一次读操作后加 1
5	DECSRC	0	若 DECSRC=1，则源地址每执行一次读操作后减 1；若 DECSRC=INCSRC，则源地址不被修改

AVP32VC33 Digital Signal Processor

6	INSDST	0	若 INCDST=1 ，则目的地址在每执行一次写操作后加 1
7	DECDST	0	若 DECDST=1 ，则目的地址在每执行一次写操作后减 1： DECDST=INCDST ，则目的地址不被修改
8~9	SYNC	0	同步控制方式，具体操作见表 3.6
10	TC	0	DMA 计数器方式控制。若 TC=0 则当计数器减少到 0 时，不停止 DMA；若 TC=1，则当计数器减少到 0 时，停止 DMA
11	TCINT	0	若 TCINT=1，则允许 DMA 中断；若 TCINT=0，则 DMA 中断无效
12~31	保留	0~0	读作 0

表 3.4 START 位和 DMA 操作（0~1 位）

START	功能
00	停止悬挂的读写操作，DMA 重新启动
01	读或写操作，若已启动，则完成后停止；若还未启动，则不启动，DMA 处于保持状态
10	保留
11	DMA 复位并启动工作

表 3.5 STAT 各位和 DMA 状态（2~3 位）

STAT	功能
00	在读和写之间保持
01	在写和读之间保持
10	保留
11	DMA 忙

表 3.6 SYNC 各位和 DMA 的同步

SYNC	功能
00	不同步，不允许中断
01	源同步，当中断允许时执行一次写操作
10	目的同步，当中断允许时执行一次写操作
11	源和目的同步，当中断使能出现时，执行一次读操作，接下来的中断使能出现时再执行一次写操作

3.2.2 目的寄存器、源地址寄存器和传输计数器

DMA 的目的地址和源地址寄存器是 24 位寄存器，通过对这些寄存器的操作能够控制源地址和目的地址。通过全局控制寄存器中的 DECSRC、INCSRC、DECOST 和 INCDST 这 4 位来实现其地址的增、减，或者保持其不变以选通相应的固定地址的通信通道。其中源寄存器为只读而目的寄存器只能写不能读。在系统复位时，这些寄存器全部置 0。

传输计数器也是一个 24 位寄存器，其内容反映传输数据块的大小，在系统复位时，传输计数器的寄存器置 0，启动 DMA 后，每传输一个字，计数器减 1，当计数器的值变为 0 时，若 DMA 全局控制寄存器中 TCINT 位为 1（即允许中断），则引起一个 DMA 中断，若 DMA 全局控制寄存器中 TCINT 位为 0（即不允许中断），则 DMA 通道将被挂起。如果当计数寄存器的值为 0 时，DMA 通道没有被挂起（通过全局控制寄存器的 TC 位设置），则计数寄存器将继续减少到负值，因此设置计数寄存器为 0 时，DMA 通道将传输最大的字数（1000000h 次）。

3.2.3 DMA 的中断方式

DMA 控制器既可以自身产生一个中断，使 CPU 被 DMA 中断，也可以接收片内定时器和串口产生的内部中断和外部中断，具体方式如下：

- （1）当块传输结束时发送中断给 CPU，由 DMA 全局控制器寄存器的 TCINT 位实现。中断使能寄存器的 EDINT 位必须置位，从而允许 CPU 被 DMA 中断。
- （2）DMA 可以接收外部中断以及定时器、串行口的中断。DMA 接收中断是通过同步来实现的。CPU 中断控制寄存器可以检测到 DMA 控制器接收到的所有中断，并设置适当的中断标志寄存器。如果 CPU 没有流水线冲突或者遇到停止取指的指令，则 DMA 和 CPU 可以响应同一中断。如果同一中断被作为源和目的同步，则单一中断到来时读写周期将同时有效。

3.2.4 DMA 的同步操作模式

（1）当 SYNC=00 时，不产生同步，其传输过程如图 3.5 所示。只要没有冲突，DMA 就可随时执行读、写操作，所有的中断被忽略，但 DMA 中的中断使能位不会发生改变。

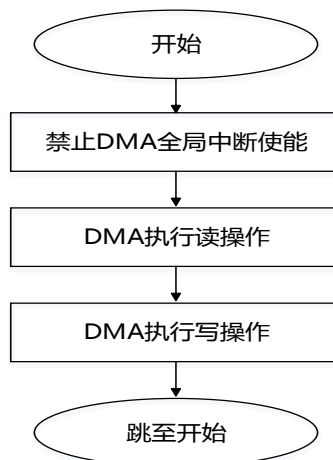


图 3.5 无同步传输方式

(2) 当 SYNC=01 时，DMA 将与源同步，其传输过程如图 3.6 所示。DMA 接收一次中断则执行一次读操作 。之后从总体上禁止所有的 DMA 中断，但 DMA 中断使能后，寄存器的各位不变。

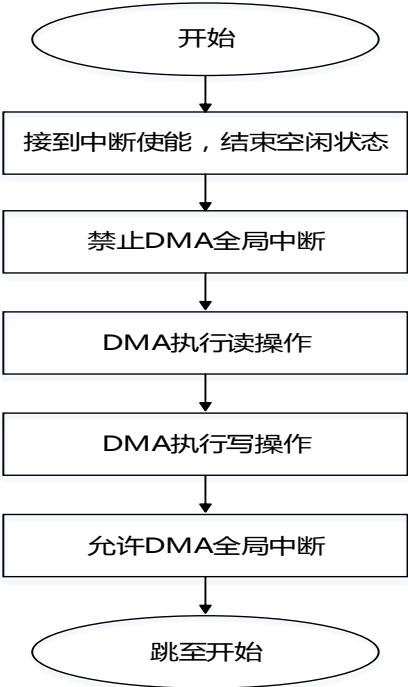


图 3.6 源同步的传输方式

(3) 当 SYNC= 10 时，DMA 与目的同步，其传输过程如图 3.7 所示。开始所有的中断都不考虑直到读操作完成 。且 DMA 中断使能寄存器的各位不会改变 。直到 DMA 接收到一次中断才执行一次写操作 ，但读操作无须等待中断。

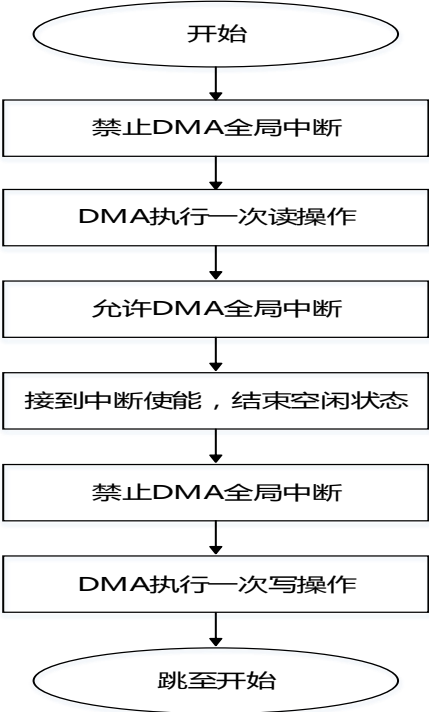


图 3.7 目的同步的传输方式

(4) 当 SYNC=11 时，DMA 将与源和目的同步，其传输过程如图 3.8 所示 。接收到中断时执行读操作，下一次中断再执行写操作 。

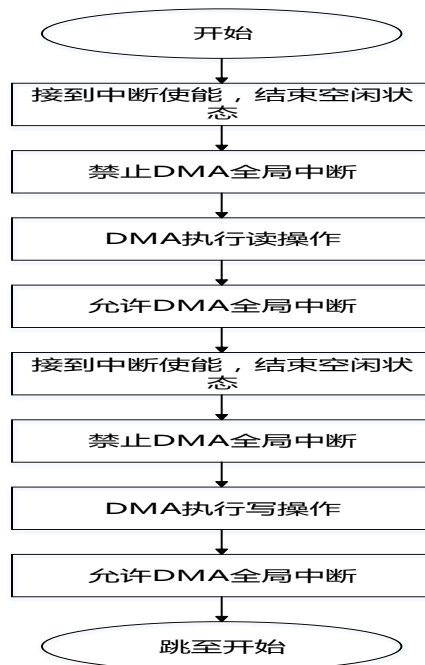


图 3.8 源和目的同步

3.2.5 DMA 的字传输过程

- (1) DMA 通道从源地址寄存器中读取一个字，并把它写入 DMA 通道的一个临时寄存器中。
- (2) DMA 通道一次读操作之后，源地址寄存器是否增加、减少或保持不变取决于 DMA 通道控制寄存器的 INCSRC 或 DECSRC 位的设置。
- (3) 读操作完成后，DMA 通道将临时寄存器中的值写入到目的地址寄存器，指向目的地址。
- (4) 目的地址从目的地址寄存器中取出字后，传输计数寄存器减少，目的地址寄存器增加、减少或保持不变取决于 DMA 通道控制寄存器的 INCDRC 或 DECDRC 位的设置。
- (5) 每次数据写入时，传输计数器的值减少。当传输计数器减至 0 时块传输停止，最后一次传输的写操作完成。DMA 通道设置 DMA 控制寄存器的传输计数器中断 (TCINT) 标志。

数据块传输完后，DMA 控制器可以通过编程实现以下事件：

- (1) 停止传输，直到重新编程 (TC=1)，设置传输计数器。
- (2) 继续传输数据 (TC=0)。
- (3) 产生一个中断信号告知 CPU，数据块传输已经完成 (TCINT=1)。

3.2.6 DMA 的初始化和启动

- (1) 清除 DMA 的全局控制寄存器的 START 位，使 DMA 处于保持状态。注意 DMA 在复位时处于保持状态。
- (2) 配置 DMA 全局控制寄存器 (START=00)，将其以适当的模式加载，通过中断的方式同步 DMA 控制寄存器的读写，配置 DMA 源、目的寄存器和传输计数器。
- (3) 对 DMA 全局控制寄存器的 START 位设置 (START=11)，启动 DMA。

3.3 串行口

AVP32VC33 具有一个双工串行口，能够同时发送和接收字长为 8、16、24 和 32 位的数据。该串行口由 8 个控制寄存器控制，分别是：

(1) 串行口全局控制寄存器：

(2) 发送端口控制寄存器，该寄存器用于控制发送端口的 3 个引脚，它们是：发送帧同步引脚 FSX、数据发送引脚 DX 和发送时钟引脚 CLKX；

(3) 接收端口控制寄存器，该寄存器用于控制接收端口的 3 个引脚，它们是：接收帧同步引脚 FSR、数据接收引脚 DR 和接收时钟引脚 CLKR；

(4) 收发定时器控制寄存器：

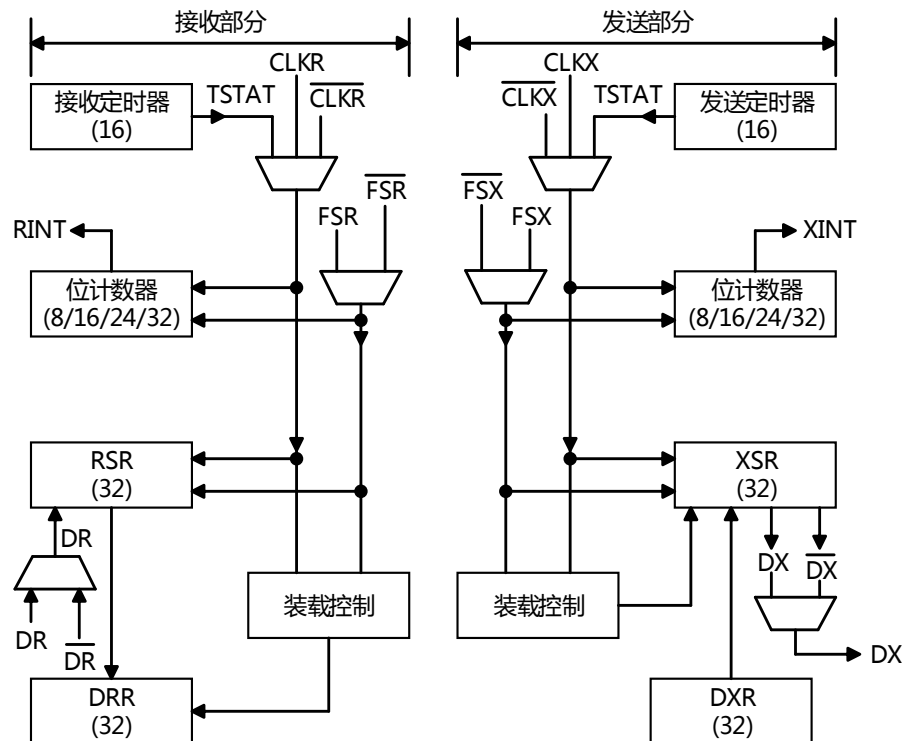
(5) 收发计数器：

(6) 收发周期寄存器：

(7) 数据发送寄存器，该寄存器用于缓存下一次将要被发送的完整的字：

(8) 数据接收寄存器，该寄存器用于存放最新接收到的完整的字。

串行全局控制寄存器控制串行口的全局功能，并决定了串行口的操作模式。两个端口控制寄存器控制六个串行口引脚的功能。数据发送寄存器包含下一个将要发送的完整的字。数据接收寄存器包含最后一次接收的字。三个附加寄存器与串行口定时器的发送 / 接收部分联合使用。串行口的结构示意图如图 3.9 所示。



RINT:接收中断；RSR：接收移位寄存器；DRR：数据接收寄存器

XINT:发送中断；XSR：发送移位寄存器；DXR：数据发送寄存器

图 3.9 串行口结构示意图

3.3.1 串行口全局控制寄存器

串行口全局控制寄存器是一个 32 位寄存器，其结构如图 3.10 所示，该寄存器中各位的名称和功能如表 3.7 所示。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
xx	xx	xx	xx	RRESET	XRESET	RINT	RTINT	XINT	XTINT	RLEN		XLEN		FSRP	FSXP
				R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W		R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRP	DXP	CLKRP	CLKXP	RFSM	XFSM	RVAREN	XVAREN	RCLK SRCE	XCLK SRCE	HS	RSR FULL	XSR EMPTY	FSX OUT	XRDY	PRDY
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R

R：只读；R/W：读/写；xx：保留位

图 3.10 串行口全局控制寄存器

表 3.7 串行口全局控制寄存器各位的名称及功能表

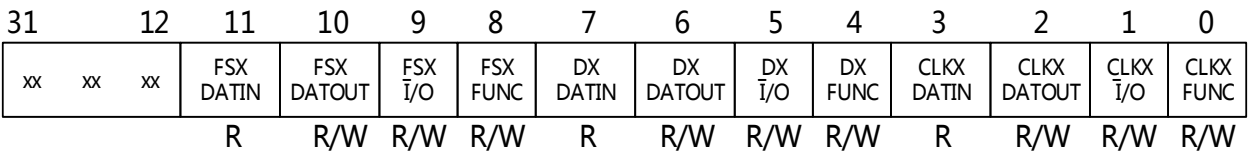
位	名称	复位值	功能描述
0	RRDY	0	接收就绪标志位。若 RRDY=1，接收寄存器内有新的数据，并已处于被读的等待状态。从 DRR 装载到 RRDY=1 有 3 个 CPU 周期的延迟。RRDY 由 0 变为 1 可使 RINT 置位。若 RRDY=0，则接收寄存器内没有新的数据。该位的复位值为 0，接收寄存器被读取后 RRDY=0
1	XRDY	1	发送就绪标志位。若 XRDY=1，则发送寄存器已经把数据的最后一位写到了发送移位器，从发送移位寄存器加载到 XRDY=1 有 3 个 CPU 周期的延迟。XRDY 由 0 变为 1 将置位 XINT。若 XRDY=0,则发送寄存器还没有把数据的最后一位写到发送移位寄存器中，复位时 XRDY=1
2	FSXOUT	0	发送帧同步位。若 FSXOUT=0, FSX 为输入引脚，否则为输出引脚
3	XSREEMPTY	0	发送移位寄存器空标志位。若 XSREEMPTY=0，则表示发送移位寄存器为空；若 XSREEMPTY=1，则表示发送移位寄存器不空。此位在复位时和 XRESET 置位时为 0
4	RSRFULL	0	接收移位寄存器满标志位。若 RSRFULL=1，表示接收寄存器满。在连续方式下，当 RSR 和 DRR 两者都满时，RSRFULL 置 1。在非连续方式下，当 RSR 和 DRR 已满且接收到新的 FSR 时，RSRFULL 置 1。读操作、系统复位和串行口接收复位时，此位都置 0。若 RSRFULL=0，接收寄存器未滿
5	HS	0	握手控制位。若 HS=1，则允许握手方式；若 HS=0，则禁止握手方式
6	XCLKSRCE	0	发送时钟源位。若 XCLKSRCE=1，则使用内部时钟；若 XCLKSRCE=0，则使用外部时钟
7	RCLKSRCE	0	接收时钟源位。若 RCLKSRCE=1，则使用内部时钟；若 RCLKSRCE=0，则使用外部时钟
8	XVAREN	0	发送方式控制位。XVAREN=0 为固定速率发送方式。该方式下，FSX 的有效时间至少持续一个 XCLK 周期，然后在发送开始前变为无效。XVAREN=1 为可变速率发送方式，该方式下，各数据位传输期间 FSX 都有效

AVP32VC33 Digital Signal Processor

9	RVAREN	0	接收方式控制位。RVAREN=0 为固定速率接收方式，RVAREN=1 为可变速率接收方式。对固定速率方式，FSR 的有效时间至少持续一个 XCLK 周期，然后在接收开始前变为无效。若使用可变速率接收方式，则 FSR 在所有各位被接收时都有效
10	XFSM	0	发送帧同步方式位。XFSM=1 为连续方式，XFSM=0 为标准方式。在连续方式中，仅仅数据块的第一个字产生一个同步脉冲；在标准方式中，每一个字都有个相应的同步脉冲
11	RFSM	0	接收帧同步方式位。RFSM=1 为连续方式，RFSM=0 为标准方式。在连续方式中，仅仅数据块的第一个字产生一个同步脉冲；在标准方式中，每一个字都有一个相应的同步脉冲
12	CLKXP	0	发送时钟极性位。若 CLKXP=0，则 CLKX 高电平有效；CLKXP=1，则 CLKX 低电平有效
13	CLKRP	0	接收时钟极性位。若 CLKRP=0，则 CLKR 高电平有效；CLKRP=1，则 CLKR 低电平有效
14	DXP	0	发送数据极性位。若 DXP=0，则 DX 高电平有效；若 DXP=1，则 DX 低电平有效
15	DRP	0	接收数据极性位。若 DRP=0，则 DR 高电平有效；若 DRP=1，则 DR 低电平有效
16	FSXP	0	发送帧同步极性位。若 FSXP=0，则 FSX 高电平有效；若 FSXP=1，则 FSX 低电平有效
17	FSRP	0	发送帧同步极性位。若 FSRP=0，则 FSR 高电平有效；若 FSRP=1，则 FSR 低电平有效
18~19	XLEN	0	发送数据的字长位。00:8 位，01:16 位，10:24 位，11:32 位。不论多少位，在发送寄存器中总是向右对齐
20~21	RLEN	0	接收数据的字长位。00:8 位，01:16 位，10:24 位，11:32 位。不论多少位，在接收寄存器中总是向右对齐
22	XTINT	0	发送定时器中断控制位。若 XTINT=0，则禁止发送定时器中断；若 XTINT=1，则允许发送定时器中断
23	XINT	0	发送中断控制位。若 XINT=0，禁止发送中断；若 XINT=1，允许发送中断。注意：中断使能控制寄存器 IE 中串口的 CPU 中断控制位 EXINT0 是 XINT 和 XTINT 的逻辑“或”；同理，IE 中串口的 DMA 中断控制位 EXINT0 也是 XINT 和 XTINT 的逻辑“或”
24	RTINT	0	接收定时器中断控制位。若 RTINT=0，则禁止接收定时器中断；若 RTINT=1，则允许接收定时器中断
25	RINT	0	接收中断控制位。若 RINT=0，禁止接收中断；若 RINT=1，允许接收中断。注意：中断使能控制寄存器 IE 中串口的 CPU 中断控制位 ERINT0 是 RINT 和 RTINT 的逻辑“或”；同理，IE 中串口的 DMA 中断控制位 ERINT0 也是 RINT 和 RTINT 的逻辑“或”
26	XRESET	0	发送复位位。若 XRESET=0，则串行口的发送端口复位；若 XRESET=1，则退出复位状态。使 XRESET=0 并不改变串行口控制寄存器的任何内容，只是把发送器置于相应的被发送数据帧的开始状态。发送器的复位会产生一个发送中断。此位应该在发送器方式置位的同时复位。在不复位全局控制寄存器的情况下，可以改变 XFSM 的状态
27	RRESET	0	接收复位位，若 RRESET=0，则串行口的接收端口复位。置 RRESET=0 并不改变串行口控制寄存器的任何内容。若 RRESET=1，则退出复位状态
28~31	保留	0	保留位，读作 0

3.3.2 发送端口控制寄存器

发送端口控制寄存器是一个 32 位的寄存器，用于控制 FSX 、 DX 、 CLKX 三个引脚的功能，其结构如图 3.11 所示，该寄存器各位的名称和功能如表 3.8 所示。



R：只读；R/W：读/写；xx：保留位。复位时第 3、7、11 位不确定，其他各位为 0

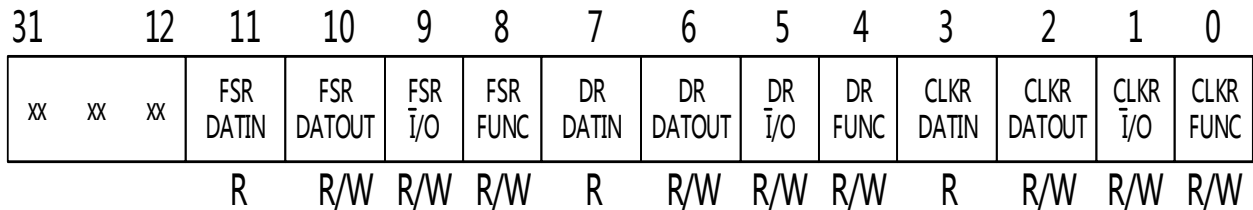
图 3.11 发送端口控制寄存器

表 3.8 发送端口控制寄存器各位的名称及功能

位	名称	复位值	功能描述
0	CLKX FUNC	0	CLKX 引脚的功能控制位。若 CLKX FUNC=0，则 CLKX 为通用 I/O 引脚；若 CLKX FUNC=1，则 CLKX 为串行口引脚
1	CLKX I/O	0	若 CLKX I/O=0，则 CLKX 为通用输入引脚；若 CLKX I/O=1，则 CLKX 为通用输出引脚
2	CLKX DAOUT	0	CLKX 配置成输出引脚时，该位的内容是 CLKX 脚上输出的数据
3	CLKX DATIN	X	CLKX 配置成输入引脚时，该位的内容是 CLKX 脚上输入的数据，写对此位没有影响
4	DX FUNC	0	DX 引脚的功能控制位。若 DX FUNC=0，则 DX 为通用 I/O 脚；若 DX FUNC=1，则 DX 为串行口引脚
5	DX I/O	0	若 DX I/O=0，则 DX 作为通用输入引脚；若 DX I/O=1，则 DX 为通用输出引脚
6	DX DATOUT	0	DX 配置成输出引脚时，该位的内容是 DX 脚上输出的数据
7	DX DATIN	X	DX 配置成输入引脚时，在 CLKX 脚上输入数据在该位，写对此位没有影响
8	FSX FUNC	0	FSX 引脚的功能控制位。若 FSX FUNC=0，则 FSX 为通用 I/O 引脚；若 FSX FUNC=1，则 FSX 为串行口的引脚
9	FSX I/O	0	若 FSX I/O=0，则 FSX 为通用输入引脚；若是 FSX I/O=1，则 FSX 为通用输出引脚
10	FSX DATOUT	0	FSX 配置成输出引脚时，该位的内容是 FSX 引脚上输出的数据
11	FSX DATIN	0	FSX 配置成输入引脚时，该位的内容是 FSX 引脚上输入的数据，写对此位没有影响
12 ~ 31	保留	0 ~ 0	保留位，读作 0

3.3.3 接收端口控制寄存器

接收端口控制寄存器是一个 32 位的寄存器，用于控制 FSR、DR、CLKR 三个引脚的功能，其结构如图 3.12 所示，该寄存器各位的名称和功能如表 3.9 所示。



R：只读；R/W：读/写；xx：保留位。复位时第 3、7、11 位不确定，其他各位为 0

图 3.12 接收端口控制寄存器

表 3.9 接收端口控制寄存器各位的名称及功能

位	名称	复位值	功能描述
0	CLKR FUNC	0	CLKR 引脚的功能控制位。若 CLKR FUNC=0，则 CLKR 为通用 I/O 引脚；若 CLKR FUNC=1，则 CLKX 为串行口引脚
1	CLKR I/O	0	若 CLKR I/O=0，则 CLKR 为通用输入引脚；若 CLKR I/O=1，则 CLKR 为通用输出引脚
2	CLKR DATOUT	0	CLKR 配置成输出引脚时，该位的内容是 CLKR 脚上输出的数据
3	CLKR DATIN	X	CLKR 配置成输入引脚时，该位的内容是 CLKR 脚上输入的数据，写对此位没有影响
4	DR FUNC	0	DR 引脚的功能控制位。若 DR FUNC=0，则 DX 为通用 I/O 脚；若 DR FUNC=1，则 DR 为串行口引脚
5	DR I/O	0	若 DR I/O=0，则 DR 作为通用输入引脚；若 DR I/O=1，则 DR 为通用输出引脚
6	DR DATOUT	0	DR 配置成输出引脚时，该位的内容是 DR 脚上输出的数据
7	DR DATIN	X	DR 配置成输入引脚时，在 CLKR 脚上输入数据在该位，写对此位没有影响
8	FSR FUNC	0	FSR 引脚的功能控制位。若 FSR FUNC=0，则 FSR 为通用 I/O 引脚；若 FSR FUNC=1，则 FSR 为串行口的引脚
9	FSR I/O	0	若 FSR I/O=0，则 FSR 为通用输入引脚；若是 FSR I/O=1，则 FSR 为通用输出引脚
10	FSR DATOUT	0	FSR 配置成输出引脚时，该位的内容是 FSR 引脚上输出的数据
11	FSR DATIN	0	FSR 配置成输入引脚时，该位的内容是 FSR 引脚上输入的数据，写对此位没有影响
12~31	保留	0~0	保留位，读作 0

3.3.4 收发定时器控制寄存器

收发定时器控制寄存器是一个 32 位的寄存器，其结构如图 3.13 所示。其中 0~5 位控制发送定时器，6~11 位控制接收定时器。该寄存器各位的名称和功能如表 3.10 所示。

31	12			11	10	9	8	7	6	5	4	3	2	1	0
xx		xx	xx	RSTAT	XX	RCLK SRC	RC/ $\overline{P}$	$\overline{RHLD}$	RGO	XSTAT	XX	XCLK SRC	XC/ $\overline{P}$	$\overline{XHLD}$	XGO
R					R/W		R/W	R/W	R/W	R		R/W		R/W	R/W

R：只读；R/W：读/写；xx：保留位。复位时各位为 0

图 3.13 收发定时器控制寄存器

表 3.10 收发定时器控制寄存器的各位名称和功能

位	名称	复位值	功能描述
0	XGO	0	发送定时器计数器启动控制位。若 XGO=1 且定时器没有处于保持，则计数器清零并在定时器输入时钟的下一个上升沿开始定时，XGO 位在同一上升沿被清零。写 0 到 XGO 对发送定时器没有影响
1	XHLD	0	发送计数器保持信号位。若 XHLD=0 则计数器被禁止并保持它的现行状态。若 XHLD=1，则内部二分频计数器也会被保持，以便计数器可以从停止的地方恢复计数
2	XC/P	0	发送时钟 / 脉冲方式控制位。若 XC/P = 1，则选择时钟方式，状态标记位所发出的信号和外部输出都是 50% 的占空比；若 XC/P = 0，则在每个定时器周期内的状态标记和外部输出信号的有效时间持续一个 CLKOUT 周期
3	XCLK SRC	0	发送定时器的时钟源控制位。若 XCLK SRC=1，则计数器的时钟源来自内部；若 XCLK SRC=0，则计数器的时钟源来自 CLKX 引脚上的外部时钟信号，外部时钟源的频率不能超过 f _{HI} /2.6
4	保留	0	保留位，读作 0
5	XSTAT	0	发送定时器的状态位。它跟踪未反相的 CLKX 引脚的状态，该位由 0 变为 1 时可置 CPU 中断。写对该位不起作用
6	RGO	0	接收定时器计数器启动控制位。若 RGO=1 且定时器没有处于保持，则计数器清零并在定时器输入时钟的下一个上升沿开始增加计数值，RGO 位在同一上升沿被清零。写 0 到 RGO 对接收定时器没有影响
7	RHLD	X	接收计数器保持信号位。若 RHLD=0，则计数器被禁止并保持现行状态；若 RHLD=1，则内部二分频计数器也被保持，以便计数器可以从停止的地方恢复计数。当定时器处于保持状态时，定时器寄存器可以被读出和修正，RESET 的优先权高于 RHLD
8	RC/P	0	接收方式控制位。若 RC/P = 1 为时钟方式，则状态标记所发出的信号和外部输出信号都是 50% 的占空比；若 RC/P = 0，则在每个定时器周期内的状态标记和外部输出信号的有效时间持续一个 CLKOUT 周期
9	RCLK SRC	0	发送定时器的时钟源控制位。若 RCLK SRC=1，则计数器的时钟源采用内部时钟；若 RCLK SRC=0，则计数器的时钟源来自 CLKR 引脚上的外部时钟信号。外部时钟源的频率不能超过 f _{HI} /2.6
10	保留	0	保留位，读作 0
11	RSTAT	0	接收定时器的状态位。它跟踪没有反相的 CLKR 引脚的输出。该位由 0 变为 1 时，可置 CPU 中断。写对该位不起作用
12~31	保留	0~0	保留位，读作 0

3.3.5 收发计数器和收发周期寄存器

(1) 收发计数器是一个 32 位寄存器，其结构如图 3.14 所示，其中 15~0 位是发送定时器的计数器，31~16 位是接收定时器的计数器，所有的位都可读可写。每个计数器计数到等于周期寄存器的值时，计数器就会置 0，复位时也会置 0。

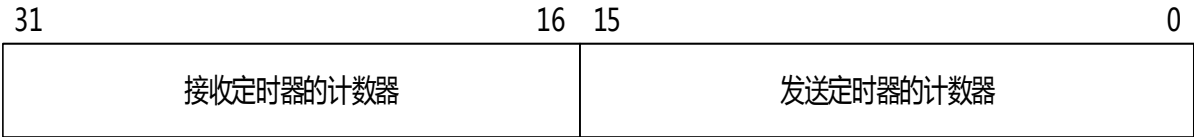


图 3.14 收发计数器

(2) 收发周期寄存器是一个 32 位寄存器，其结构如图 3.15 所示，其中 15~0 位为发送周期寄存器，31~16 为接收周期寄存器，每一位都可被读可写，周期寄存器的作用是规定定时器的周期，复位时此寄存器置 0。

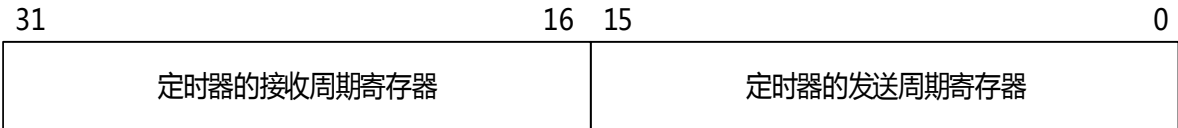


图 3.15 收发周期寄存器

3.3.6 数据发送寄存器和数据接收寄存器

(1) 数据发送寄存器：当数据写入发送寄存器 DXR 时，发送器把已写入的数据传送到发送移位寄存器 XSR 中，然后各位被移出。从写 DXR 到 FSX 的出现有两个 CLKX 周期。在移位寄存器未被移空前，数据字不能再写到移位寄存器中。当数据从 DXR 加载到 XSR 时，XRDY 位置位，此时表明寄存器能用来接收下一个字了。发送移位寄存器 XSR 的操作过程如图 3.16 所示。图中有 4 个抽头点分别对应于四种数据字的端点。发送移位寄存器的移位过程是向左移位，即由低位向高位方向移动。

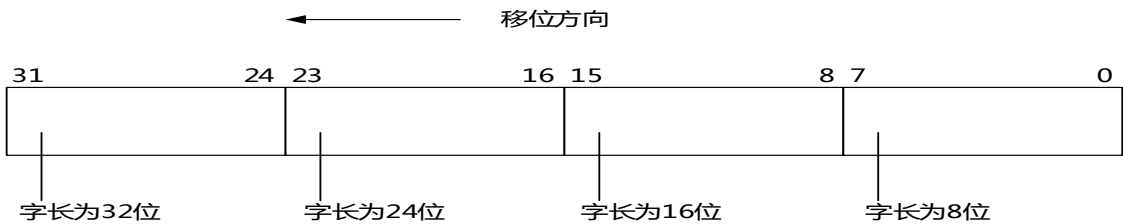


图 3.16 数据发送移位寄存器的移位操作过程

(2) 数据接收寄存器：当串行数据从数据接收引脚 DR 输入时，接收器将其移位到接收移位寄存器 RSR 中；当规定的位数全被移进来后，数据接收寄存器 DRR 从 RSR 中读取数据，并把 RRDY 状态位置位。接收器是一个双缓冲器。如果 DRR 没有被读出而且 RSR 已装满，则接收器就会被冻结，此时从引脚 DR 新进来数据就会被忽略。接收移位寄存器的数据也不会写到 DRR 中去。只有 DRR 被读出后，RSR 中的新数据才能传输到 DRR 中。在数据写入到 DRR 中的同时，也发生了 RSR 到 DRR 的传输，并且 RSR 到 DRR 的传输具有优先权。

数据按位左移，图 3.17 说明了当从串口输入的数据小于 32 位时的情况。在此图中，假定先接收了

一个 8 位的字节 a，这时接收缓冲器的其它三个高字节未被定义，接收到第二个字节 b 时，字节 a 就会向左移，当读取接收寄存器中的数据时，字节 a 和 b 都会被读出。

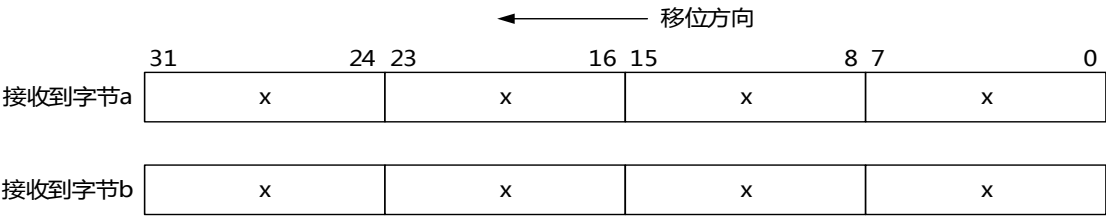


图 3.17 接收移位寄存器的移位操作

3.3.7 串行口的数据传输方式

串行口的数据传输方式可分为两类，即固定数据速率和可变数据速率方式，每种方式又有连续、爆发和标准三种不同的传输方式。

1、固定数据速率方式

固定数据速率的串行口传输有三种方式：爆发方式、标准方式和连续方式。在爆发方式操作中，单个字的传输是由串行口的无效周期分隔开的；在标准方式中，每个字节之间首尾相接，但每个字开始前需要一个帧同步信号；在连续方式中，在相邻字之间的传输不存在间隙，也就是新字的第一位是跟着前一个字的最后一位，但每一个字需要一个帧同步。

(1) 固定爆发方式

在带有固定数据速率时序的爆发方式中，FSX 或 FSR 脉冲启动数据传输，而每一次传输只包含一个字。对于使用内部的 FSX 来说，如图 3.18 所示，发送是通过加载 DXR 启动的。在这种方式下，从 DXR 被加载到 FSX 出现，大约有 2.5 个 CLKX 周期的延迟；若使用外部 FSX，则 FSX 启动数据传输。

在接收操作中，只要传输开始，就不考虑 FSR 直到最后一位。在爆发模式传输中，FSR 必须保持低电平至最后二位的传输，或者将启动另外一次传输。整个字被接收并且传输到 DRR 后，产生一个 RINT。

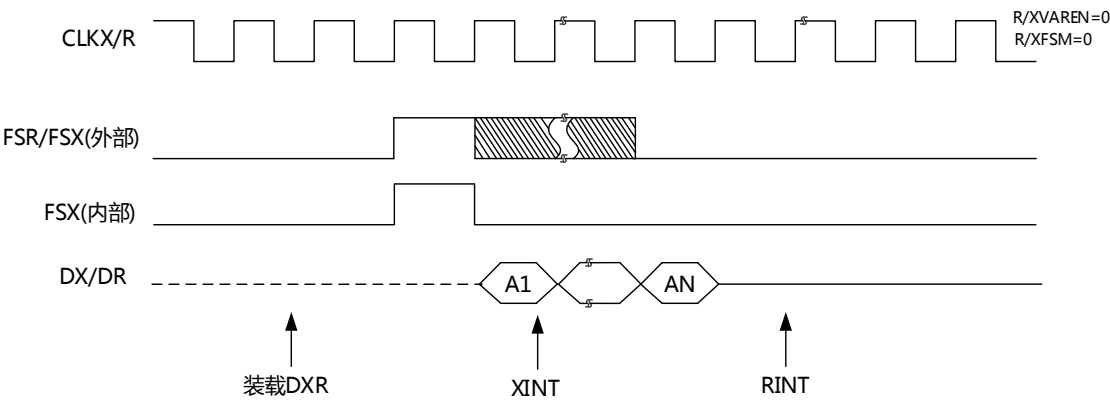


图 3.18 固定爆发方式

(2) 固定标准方式

在固定速率数据方式中，即使 RFSM 或 XFSM=0，只要适当地提供帧同步或者 DXR 在每周期被重新加载（使用一个内部产生的 FSX），就可以进行连续的数据传输，如图 3.19 所示。

对于接收操作，使用外部产生的 FSX，一旦传输开始，只在最后一位传输期间需要帧同步脉冲，以此来启动后续的传输，否则就可以不考虑帧同步输入。如果帧同步脉冲保持高电平，就会产生连续传输。使用内部产生的 FSX，从 DXR 加载到 FSX 出现，大约有 2.5 个 CLKX 周期延迟。这种延迟在每次 DXR 加载的时候都会出现，因此在连续传输期间，若传输 N 位，则加载 DXR 的指令必须早于第 N-3 位执行。由于延迟可能因流水线而发生变化，因此应该为延迟留有保守的安全余量。

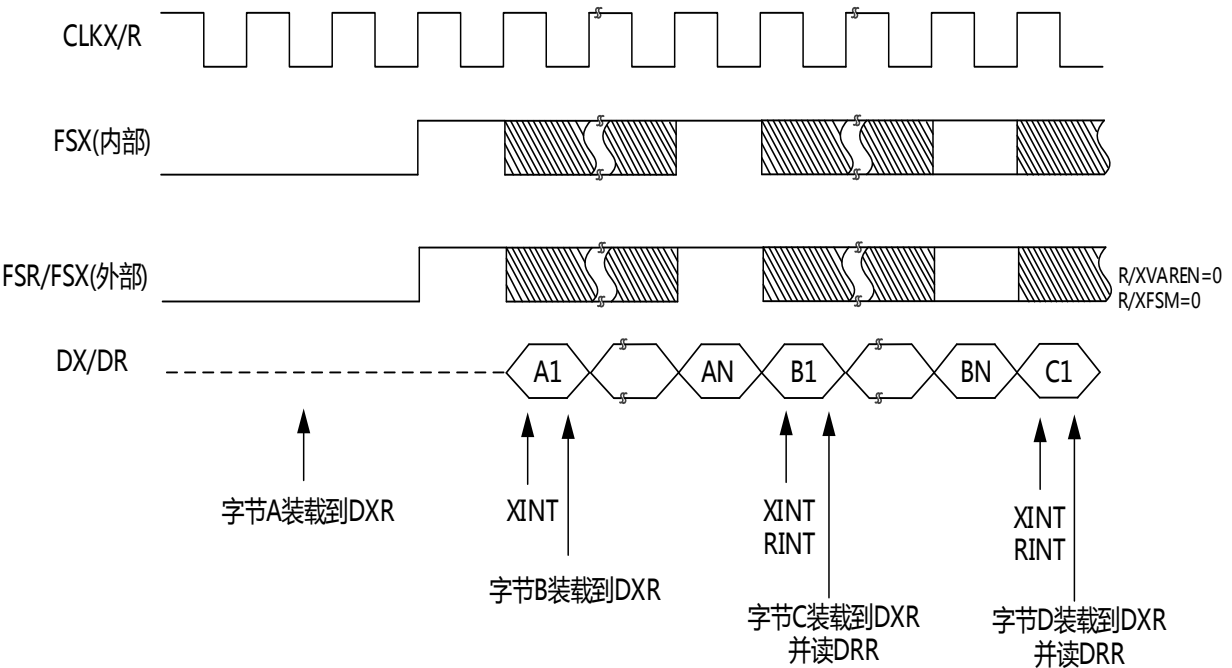
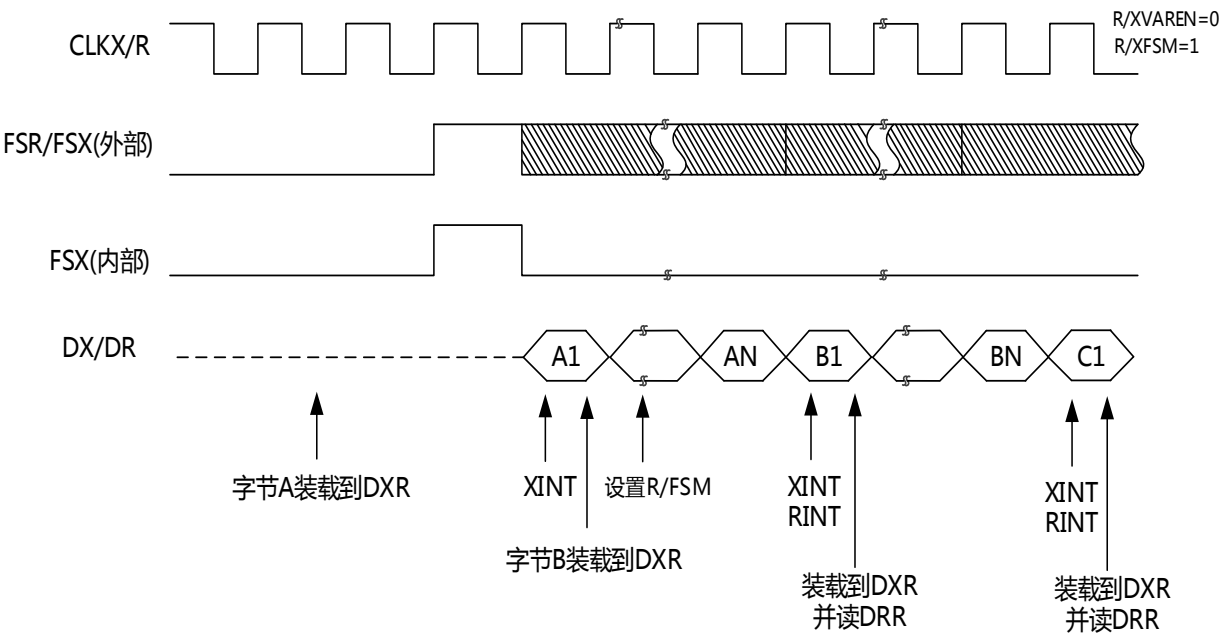


图 3.19 固定标准模式

(3) 固定连续方式

如果把 RFSM 或 XFSM 置 1，就可以在不使用帧同步脉冲的情况下完成串行口的连续传输。在这种方式下，帧同步脉冲仅包含在第一个字的传输中，以后就不再需要使用帧同步脉冲了。在第一个字传输之后，不产生内部的帧同步，也不考虑帧同步输入，另外可以在第一个字传输期间或之前对 R/XFSM 置位。除了发送操作外，必须在第一个字的第 N-1 位传输之前对 R/XFSM 置位。对于固定数据速率模式发送操作，必须在第 N-2 位传输之前对 R/XFSM 置位，清除 R/XFSM 必须在第 N-1 位之前执行，以便在当前周期中识别。该方式的传输时序如图 3.20 所示。

RINT、XINT、数据传入及传出，DXR 和 DRR 的时序分别与具有帧同步的固定数据速率标准模式相同。在这种模式下，从 DXR 装载到内部 FSX 产生也有 2.5 个 CLKX 周期的延迟。如同具有帧同步的固定数据速率模式的连续操作，也必须在第 N-3 位之前重新加载 DXR。

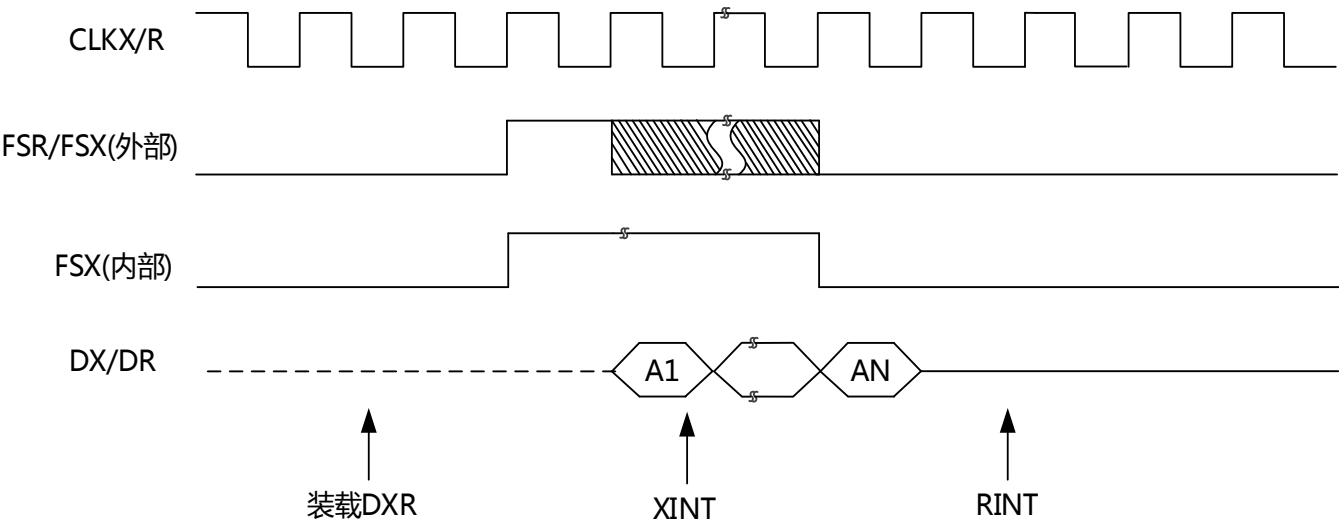


2、可变数据速率方式

可变数据速率方式包括可变爆发方式、可变标准方式和可变连续方式

(1) 可变爆发方式

在可变数据速率爆发方式时序中，FSX 或 FSR 脉冲持续整个数据传输期间，如图 3.21 所示。采用内部 FSX 时，通过装载 DXR 启动传输，DXR 的装载到产生 FSX 有大约 3.5 个 CLKX 周期的延迟。采用外部 FSX 时，通过 FSX 脉冲启动数据的传输，并且从装载 DXR 到 FSX 有效需要 3.5 个 CLKX 周期的延迟。一旦 XSR 从 DXR 加载，则一个 XINT 就会产生。



(2) 可变标准方式

可变数据速率的标准方式的传输时序如图 3.22 所示。可变数据速率的标准方式与固定数据速率的标准传输方式是相同的。不同的是，必须在第 N-4 位之前重新加载 DXR 以维持可变数据速率模式下的连续操作，而必须在第 N-3 位之前重新加载 DXR 以维持固定数据速率模式下的连续操作。

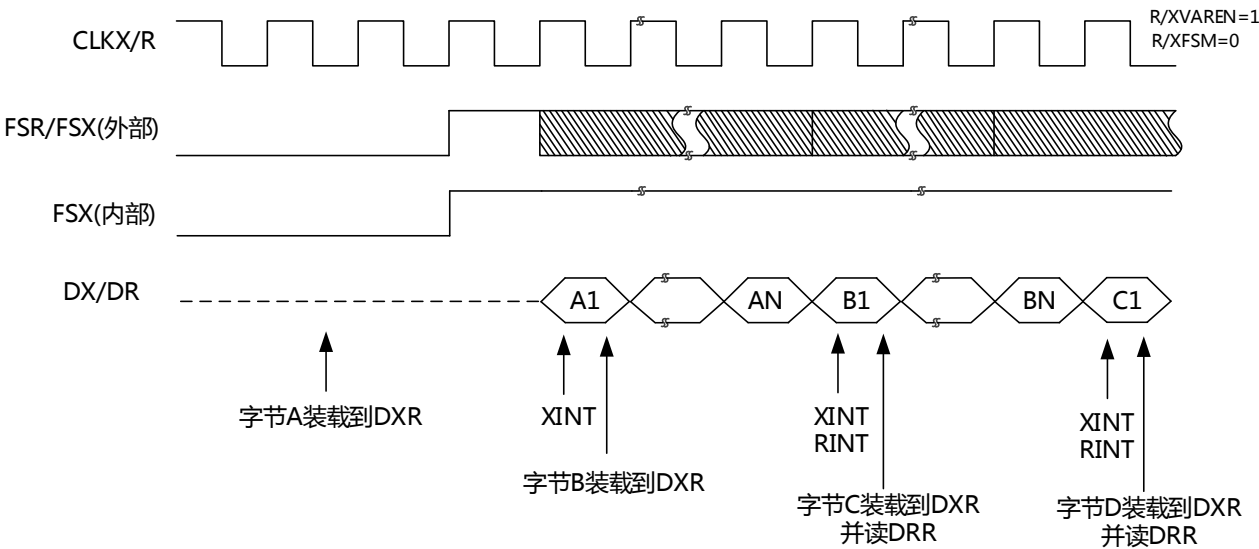


图 3.22 可变标准方式

(3) 可变连续方式

可变数据速率的连续方式的传输时序如图 3.23 所示。可变数据速率的连续方式与固定数据速率的连续传输方式是相同的。使用帧同步的可变数据速率标准模式下，必须在第 N-4 位之前重新加载 DXR 以维持连续操作。另外，在可变数据模式中，当 R/XFRM 被置位或清除时，必须在第 N-1 位之前进行修改，使其在当前传输中生效。

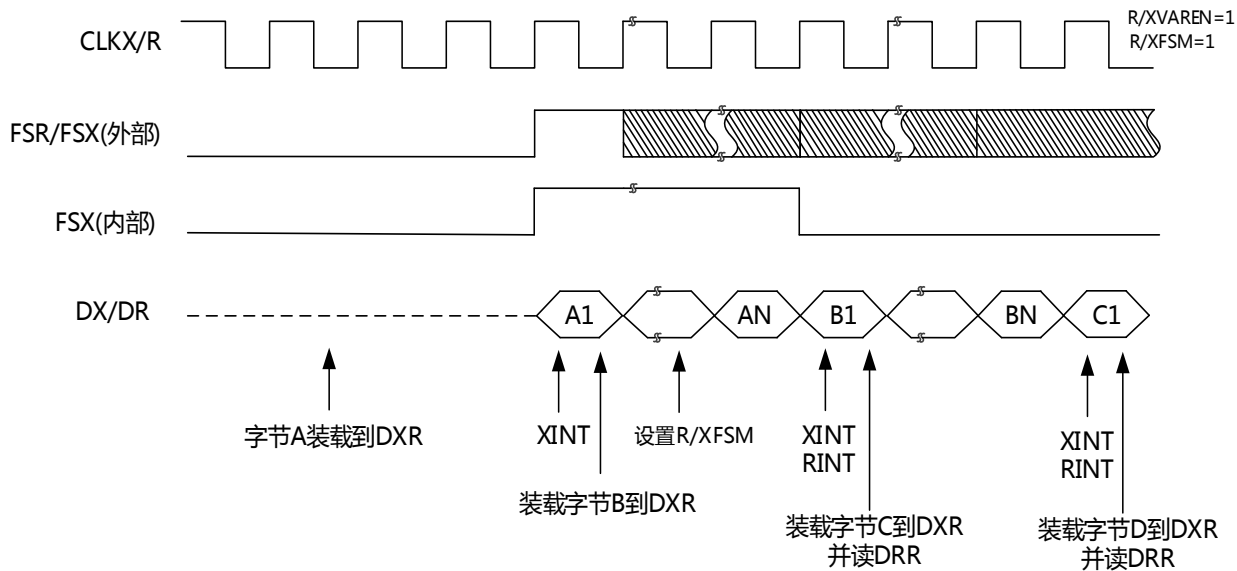


图 3.23 可变连续方式

3.3.8 串行口常用的工作方式

1、连续发送和接收方式

当选择连续方式时，仅仅是数据块的第一个字需要帧同步信号，之后，只要新的数据是在上一个数据的最后一位被发送之前被加载进 DXR，则就可以连续发送。一旦 TXRDY 变为有效并且所有的数据已经送出移位寄存器，则 DX 引脚处于高阻状态，以后写入 DXR 的数据将启动一个新的数据块和新的 FSX。

在连续方式下，接收器会连续地移进新数据并加载 DRR。如果数据接收缓冲器在下一个字移进之前还没有被读取，则随后进来的数据将丢失。RFSM 能用来终止连续接收方式。

2、握手方式

握手方式（HS=1）允许处理器之间直接连接。在这种方式下，首先要发送一个请求位“1”，再发送数据字，发完一个字后，一直到等待接收到一个回答信号“0”后才开始发送下一个字。在串行口接收到一个带开始位“1”的字以后，将发送一个“0”到发送口，这一单独的“0”位就是一个回答位，每当 DRR 读出一字，就发送一个回答位。

在握手方式中，FSX 被自动配置成输出脚，连续方式被自动禁止。在系统复位或 XRESET 复位后，发送器总是允许发送。当进入握手方式时，发送器和接收器必须复位。

3.3.9 串行口的中断源

（1）发送定时器中断：XTSTAT 由 0 变为 1 时，会产生一个单周期的中断脉冲。当 XTINT 为 0 时，这个中断脉冲被禁止。

（2）接收定时器中断：RTSTAT 由 0 变为 1 时，会产生一个单周期的中断脉冲。当 RTINT 为 0 时，这个中断脉冲被禁止。

（3）发送器中断：DXR 的数据传输到 XSR 时会产生一个单周期的中断脉冲。当串行口全局控制寄存器的 XINT=0 时，这个中断脉冲被禁止。

（4）接收器中断：RSR 的数据传输到 DRR 时会产生一个单周期的中断脉冲。当串行口全局控制寄存器的 RINT=0 时，这个中断脉冲被禁止。

3.3.10 串行口初始化与配置

串行口是通过所指定的外围总线上的存储器映射寄存器来控制的，对串行口的初始化和配置步骤如下：

（1）通过清除串行口全局控制寄存器的中的 XRESET 和 RRESET 位，使串行口处于禁止状态，向该寄存器写 0 即可完成。

（2）通过串行口全局控制寄存器（XRESET=RRESET=0）、FSX/DX/CLKX 和 FSR/DR/CLKR 端口控制寄存器来配置串行口。如果必要的话，还需要配置接收发送控制寄存器（XHLD=RHLD=0）、收发计数器和收发周期寄存器。

（3）通过对串行口全局控制寄存器的 XRSET 位和 RRESET 位置位来启动串行口。

3.4 程序引导功能

AVP32VC33 具有片上引导功能，可以通过处理器、存储器（EPRPM 和 FLASH）以及串口将可执行的程序装载到系统指定的高速 RAM 中，然后运行程序。程序引导功能是通过 AVP32VC33 内部固化的引导程序来实现。引导的程序除了程序代码本身外，还需要在程序代码各段前加入少量标识字以提供其运行所必需的信息。可通过 HEX 格式转换工具，加上规定的转换指令，可将.out 程序文件转换成具有特定格式和标识字的机器代码，然后将转换成的机器代码写入 EPROM 和 FLASH 等存储芯片中。

3.4.1 引导方式选择

若 AVP32VC33 执行程序引导功能，必须保证 MCBL/MP 引脚为高电平，此时片内固化的引导程序将起作用。在引导前，首先要选择引导方式，其引导方式分为存储器引导和串行口引导两种。当 MCBL/MP 引脚为高电平时，片内的引导程序被映射到 0h~FFFh 的内存空间内，DSP 上电复位后，通过查询 INT0~INT3 引脚的电平来确定引导方式，表 3.11 给出了可供选择的方式，选择的准则是因 INT0~INT3 引脚的电平。

表 3.11 引导方式选择

INT0	INT1	INT2	INT3	装载模式	被引导程序首地址	数据类型
0	1	1	1	存储器引导	1000h	8、16、32 位
1	0	1	1	存储器引导	400000h	8、16、32 位
1	1	0	1	存储器引导	FFF000h	8、16、32 位
1	1	1	0	串行口引导	串行口	32 位，外部时钟和帧同步

由表 3.11 可知，如果上电后外部中断引脚 INT3 为低电平，则通过串行口来进行引导，所采用的字长只能是 32 位，串行口的工作方式只能是固定速率的爆发方式。图 3.24 给出了串行口引导的流程，传送的数据是从最高有效位（MSB）开始，到最低有效位（LSB）结束。

3.4.2 引导程序的工作流程

引导程序的工作流程如下：

（1）在复位时将 MCBL/MP 引脚置高位以及通过将 INT0~INT3 中相应的引脚接低电平来选择引导方式。在这个引导过程中，MCBL/MP 引脚必须保持高电平，但是当引导结束后可以随意改变其电平。当引导过程结束后可以直接改变 INT0~INT3 引脚信号而不需复位。

（2）中断标志寄存器（IF）的 INT0~INT3 位给出引导方式。

（3）如果只有 IF 寄存器的 INT3 位置位，则引导程序将串口配置为具有外部串口时钟和帧同步信号的 32 位固定爆发读方式，然后通过串口引导源程序。串口传输的数据顺序为先高位后低位。

（4）反之，引导程序将尝试存储器引导方式。存储器引导方式的程序的首地址与相应的中断对应，如表 3.12 所示。存储器引导的源程序包含一个用于指示引导存储器宽度和存储器配置控制字的文件头，该配置字复制写入 STRB 控制寄存器以配置外部主总线接口。

(5) 读完文件头后，引导程序开始复制源程序块。源程序块的前面有两部分：第一部分给出了该块的大小，若大小为零，则表示已经到源程序的末尾；第二部分给出了该块要重复到的存储区的地址。除非地址译码逻辑被重映射，否则引导程序无法将源程序载入地址 1000h 以下的任何存储区。

(6) 引导程序跳转到第一个源程序块的目的地址并开始执行程序。

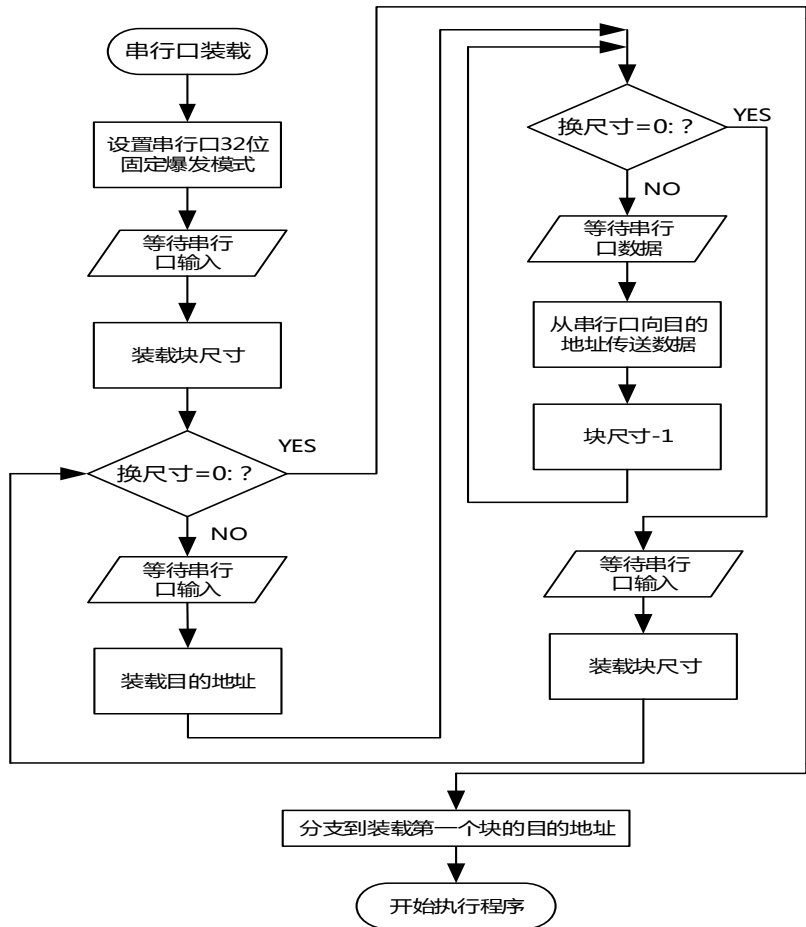


图 3.24 串口引导流程

采用存储器引导方式时，文件头必须包含以下信息：

- (1) 外部存储器的宽度（8/16/32 位）；
- (2) 程序代码块的长度；
- (3) 装入的目的程序的起始地址；
- (4) 存储器访问的定时控制（如等待状态数）。

存储器引导方式时的文件头安排如表 3.12 所示

表 3.12 文件头内容安排

位置	作用	说明
0	Boot 存储器类型（8/16/32）	0x08—8 位；0x10—16 位；0x20—32 位
1	Boot 存储器配置	插入等待状态数
2	程序代码块长度	最大不超过 24 位寻址长度
3	目的程序的起始地址	任意可用的 24 位地址
4	程序代码	用户程序代码和数据

3.4.3 hex 格式转换命令

由于许多 EPROM 不支持汇编器生成的目标执行文件（.out 文件），所以必须把目标执行文件生成十六进制 ASCII 格式才能将其写入 EPROM 中，为此提供了一个转换工具 hex30，只需要编写一个命令文件，通过转换工具 hex30 便可以将目标执行文件转换成十六进制格式。表 3.13 列出了十六进制转换命令选项

表 3.13 十六进制转换命令

位置	选项（指令）	描述
全局控制	-map 文件名	产生一个映像文件
	-o 文件名	指定指出文件
	-q	抑制运行过程中的显示信息
允许写入 ROM 内的数据连续分布	-byte	指定字节数
	-fill	填充值（默认为 0）
	-image	映像模式
	-zero	将初始地址复位为 0
对输出文件的存储字长进行配置	-datawidth	定义合理的数据字长（默认 32 位）
	-memwidth	定义系统存储器字长（默认 32 位）
	-order	指定字的顺序
	-romwidth	指定 ROM 宽度
指定输出格式	-a	ASCII 十六进制格式
	-i	Intel 格式
	-ml	Motorola-SI 格式
	-m2 或-m	Motorola-SI 格式
	-m3	Motorola-SI 格式
	-t	Ti-Tagged 格式
	-x	Tektronix 格式
控制片上引导	-boot	将各段转换到引导文件中
	-bootorg	指定源地址
	-bootpage	指定目标页数
	-cg	设置总线控制寄存器
	-e	指定引导后的 PC 值
	-ivtp	确定中断矢量表
	-tvtp	设置陷阱矢量表

注：-datawidth 是指芯片的字长；
-memwidth 是指存储系统的字长之和； -
-romwidth 是指单片 ROM 的字长；
具体 EPROM 和 FLASH 的烧写步骤和操作在 AVP32VC33 的使用手册中有详细介绍

4 开发支持

软件开发工具

- Code Composer C3X 集成开发环境(IDE)
 - C/C++ 编译器
 - 代码生成工具
 - 汇编器/连接器
 - 周期精确模拟器
- 应用算法
- 示例应用代码

硬件开发工具

- AVP32VC33
- 支持基于JTAG 的仿真器- XDS510 USB,
- 通用5V直流电源，推荐5V/1A以上
- 文档和线缆

5 电气规范

这个部分提供了针对 AVP32VC33 的绝对最大额定值和建议运行条件。

5.1 绝对最大额定值

电源电压范围(DV _{DD})		-0.3V 至 4V
电源电压范围(CV _{DD})		-0.3V 至 2.4V
输入电压范围 V _I		-1V 至 4.6V
输出电压范围 V _O		-0.3V 至 4.6V
连续功耗 (最坏情况)		500mw
操作温度范围	T _C (PGE-商业)	0°C 至 90°C
	T _C (PGEA-工业)	-40°C 至 100°C
贮存温度范围, T _{stg} ⁽³⁾		-55°C 至 150°C

(1) 除非另外说明, 绝对最大额定值的列表在运行温度范围内指定。在超过那些绝对最大额定值下列出的应力下运行有可能会对器件造成永久损坏。这些只是压力等级, 并且在“推荐工作条件”下指出的这些或任何其他条件下, 器件的功能操作不是默示, 长时间运行在最大绝对额定条件下会影响设备的可靠性。所有电压值都是以 V_{SS} 为参考。

(2) 绝对直流输入电平不应超过 DV_{DD} 或 V_{SS} 电源电压的 0.3V 以上。瞬时低电流 <2ns, ,10ms, 允许 <1V 振幅。

5.2 建议的运行条件

		最小值	典型值	最大值	单位
CV _{DD}	核心 CPU 的供电电压	1.71	1.8	1.89	V
DV _{DD}	I/O 引脚的电源电压	3	3.3	3.6	V
V _{SS}	电源接地		0		V
V _{IH}	高电平输入电压	0.7*DV _{DD}	3.3	DV _{DD} +0.3	V
V _{IL}	低电平输入电压	-0.3		0.3*DV _{DD}	V
I _{OH}	高电平输出电流			4	mA
I _{OL}	低电平输出电流			4	mA
T _C 环境温度	工作温度 (商业)	0		90	°C
	工作温度 (工业)	-40		100	
C _L	每个输出引脚的电容负载			30	PF

(1) 所有电压值均相对于 V_{SS};

(2) 绝对直流输入电平不应超过 DV_{DD} 或 V_{SS} 电源电压的 0.3V 以上。瞬时低电流 <2ns, ,10ms, 允许 <1V 振幅。

(3) 所有输入和 I/O 引脚均配置为输入;

(4)除了 SHZ 和 D0~D31 外,所有输入和 I/O 引脚均使用施密特迟滞输入,滞后约为 DV_{DD} 的 10%,并以 0.5*DV_{DD} 为中; (5) CV_{DD} 不能超过 DV_{DD} 0.7V; DV_{DD} 不能超过 CV_{DD} 2.5V.

5.3 建议的电源电压范围内的电气特性（除非另有说明）

参数	测试条件	最小值	典型值	最大值	单位
V_{OH} 高电平输出电压	$DV_{DD}=MIN, I_{OH}=MAX$	2.4			V
V_{OL} 低电平输出电压	$DV_{DD}=MIN, I_{OH}=MAX$			0.4	V
I_Z 高阻抗电流	$DV_{DD}=MAX$	-5		+5	μA
I_I 输入电流	$V_I=V_{SS}$ 到 V_{DD}	-5		+5	μA
I_{IPU} 输入电流（带内部上拉）	带内部上拉的输入	-600		10	μA
I_{IPD} 输入电流（带内部下拉）	带内部下拉的输入	600		-10	μA
I_{BKU} 输入电流（带总线保持器上拉）	Bus Keeper 反对直到条件匹配	-600	3	10	μA
I_{BKD} 输入电流（带总线保持器下拉）		600		-10	μA
I_{DDD} 电源电流	$TC=25^{\circ}C,$ $DV_{DD}=MAX$	$F_x=75MHz$	25	150	mA
I_{DDC} 当前核心电源电流	$TC=25^{\circ}C,$ $CV_{DD}=MAX$	$F_x=75MHz$	60	100	mA
I_{DD} 电源电流, I_{DDD} 加 I_{DDC}	启用 PLL, 振荡器启用		2		mA
	禁用 PLL, 振荡器启用		500		μA
	禁用 PLL, 振荡器禁用, $F_{CLK}=0$		100		
C_i 输入电容	除 XIN 外的所有输入			10	pF
	XIN			10	
C_o 输出电容				10	pF

- （1）所有电压值均相对于 V_{SS} ;
- （2）对于显示为 MIN, MAX, 或 NOM 的测试条件, 请用推荐的操作条件表中指定的适当值;
- （3）对于 VC33, 所有典型值均为 $DV_{DD}=3.3$, $CV_{DD}=1.8V$, TC （外壳温度） $=25^{\circ}C$,
- （4）带内部上拉装置的引脚: TDI, TCK 和 TMS, 带内部下拉器件的引脚: TRST;
- （5）引脚 D0~D31 包括内部总线保持器, 当总线未被驱动时, 它们保持有效的逻辑电平;
- （6）实际工作电流小于该最大值。该值是在特殊生产的最坏情况测试条件下获得的在正常的设备操作期间不会持续;
- （7） f_x 是 PLL 输出时钟频率。

5.4 锁相环 (PLL) 电路时序

参数	最小值	最大值	单位
F_{PLLin} 频率范围，PLL 输入	5	15	MHZ
F_{PLLout} 频率范围，PLL 输出	25	75	MHZ
I_{pll} PLL 电流， CV_{DD} 电源		2	mA
P_{pll} PLL 电源， CV_{DD} 电源		5	mW
PLL_{dc} PLL 输出占空比为 H1	45	55	%
PLL_J PLL 输出抖动， $F_{pllout}=25\text{ MHz}$		400	PS
PLL_{LOCK} 输入周期中的 PLL 锁定时间		1000	cycles

为确保干净的内部时钟参考，必须保持最小的低和高脉冲持续时间，高频率，可能需要快速上升和下降时间以及严格控制的占空比。在较低的 x1 和 x0.5 模式下，这些要求的限制性较低。但是 PLL 必须具有正确操作的输入占空比在 40%到 60%之间。

5.5 时钟电路时序

下表定义了时钟电路信号的时序参数

在片上晶体振荡器推荐工作条件下的切换特性

参数	最小值	典型值	最大值	单位
V _O 振荡器内部供电电压		CV _{DD}		V
F _O 基本模式频率范围	1		20	MHz
V _{bias} 直流偏置点（输入阈值）	40	50	60	%V _O
R _{fbk} 反馈阻力	100	300	500	KΩ
R _{out} 小信号交流输出阻抗	250	500	1000	Ω
V _{xoutac} 带测试晶体的交流输出电压		85		%V _O
V _{xinac} 带测试晶体的交流输入电压		85		%V _O
V _{xoutl} V _{xin} =V _{xinl} , I _{out} =0, F _O =0	V _{SS} -0.1		V _{SS} +0.3	V
V _{xouth} V _{xin} =V _{xinl} , I _{out} =0, F _O =0	CV _{DD} -0.3		CV _{DD} +0.1	V
V _{inl} 当用于逻辑电平输入时，振荡器使能	-0.3		0.2*V _O	V
V _{inh} 当用于逻辑电平输入时，振荡器使能	0.8*V _O		DV _{DD} +0.3	V
V _{xinh} 当用于逻辑电平输入时，振荡器禁用	0.7*D _{VDD}		DV _{DD} +0.3	V
C _{xout} XOUT 内部负载电容	2	3	5	pF
C _{xin} XIN 内部负载电容	2	3	5	pF
t _d (XIN-H1) 延迟时间，XIN 到 H1 x1 和 x5 模式的时间	2	5.5	8	ns
I _{inl} 输入电流，反馈使能，V _{il} =0			50	μA
I _{inh} 输入电流，反馈使能，V _{il} =V _{ih}			-50	μA

- (1) 该电路用于串联谐振基模操作
- (2) 信号幅度取决于所使用的晶体和负载

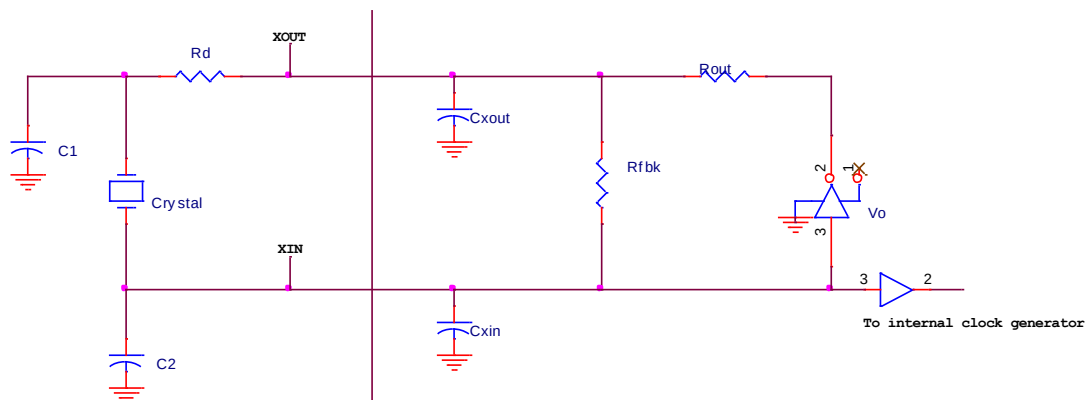


图 5.1

注意：有关 Rd 的值，请参见下表

时钟电路时序 (续)

下表定义了 EXTCLK 的时序要求和开关特性，

AVP32VC33 Digital Signal Processor

所有模式的 EXTCLK 时序要求 (见图 5.2 和 5.3)

参数			最小值	最大值	单位	
$t_{r(EXTCLK)}$ 上升时间，EXTCLK			F=Fmax,x0.5 和 x1 模式	1	ns	
			F<Fmax	4		
$t_{f(EXTCLK)}$ 下降时间，EXTCLK			F=Fmax,x0.5 和 x1 模式	1	ns	
			F<Fmax	4		
$t_{w(EXTCLKL)}$ 脉冲持续时间，EXTCLK 低电平的时间			x5 模式	21	ns	
			x1 模式	5.5		
			x0.5 模式	4.0		
$t_{w(EXTCLKH)}$ 脉冲持续时间，EXTCLK 高电平的时间			x5 模式	21	ns	
			x1 模式	5.5		
			x0.5 模式	4.0		
$t_{dc(EXTCLK)}$ 占空比 EXTCLK[$t_{w(EXTCLK)}/t_{c(H)}$]			x5 PLL 模式	40	%	
			x1 和 x0.5 模式，F=max	45		55
			x1 和 x0.5 模式，F=0 Hz	0		100
$t_c(EXTCLK)$	循环时间， EXTCLK 的时间 AVP32VC33-120	x5 模式	83.3	200	ns	
		x1 模式	16.7			
		x0.5 模式	10			
	循环时间， EXTCLK 的时间 AVP32VC33-150	x5 模式	66.7	200		
		x1 模式	13.3			
		x0.5 模式	10			
F_{ext}	频率范围， $1/t_c$ (EXTCLK) AVP32VC33-120	x5 模式	5	12	MHz	
		x1 模式	0	60		
		x0.5 模式	0	100		
	频率范围， $1/t_c$ (EXTCLK) AVP32VC33-150	x5 模式	5	15		
		x1 模式	0	75		
		x0.5 模式	0	100		

在推荐的工作条件下, 所有模式下 EXTCLK 的切换特性

参数		最小值	典型值	最大值	单位
V_{mid} 中级, 用于衡量工作周期			$0.5 \cdot DV_{DD}$		V
$t_{d(EXTCLK-H)}$ 延迟时间, EXTCLK 到 H1 和 H3 的时间	x1 模式	2	4.5	7	ns
	x0.5 模式	2	4.5	7	
$t_{r(H)}$ 上升时间, H1 和 H3				3	ns
$t_{f(H)}$ 下降时间, H1 和 H3				3	ns
$t_{d(HL-HH)}$ 延迟时间, 从 H1 低电平到 H3 高电平的时间或从 H3 低电平到 H1 高电平的时间		-1.5		1.5	ns
$t_{c(H)}$ 循环时间, H1 和 H3	x5 PLL 模式		$1/(5 \cdot f_{ext})$		ns
	x1 模式		$1/f_{ext}$		
	x0.5 模式		$2/f_{ext}$		

时钟电路时序图;

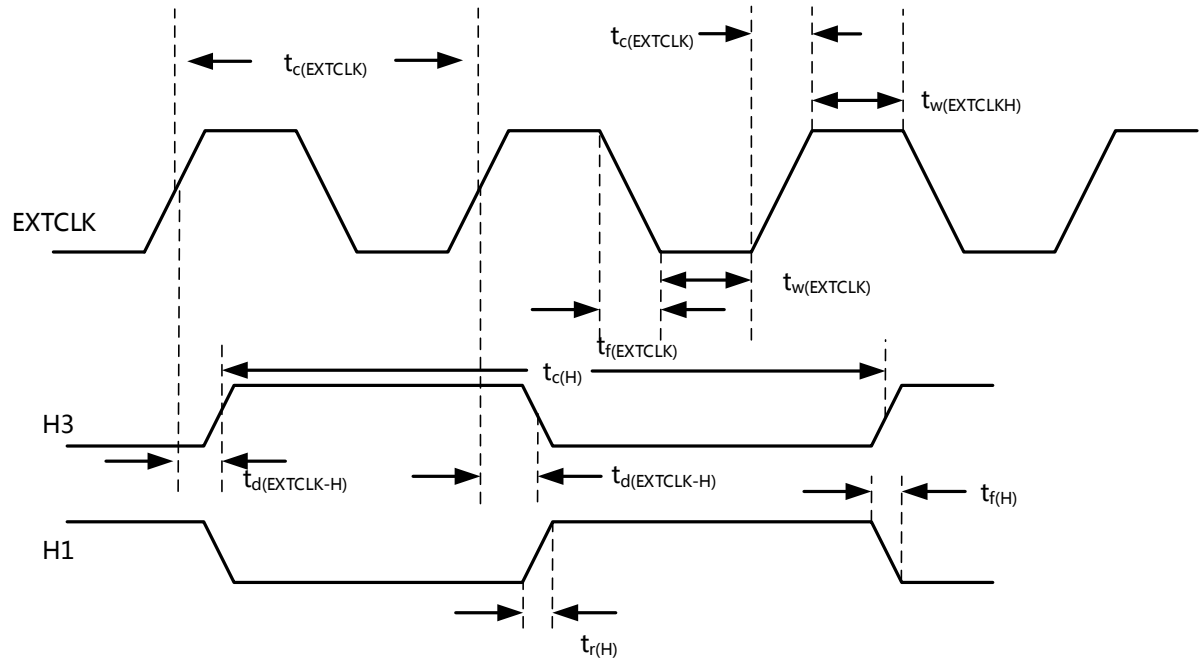
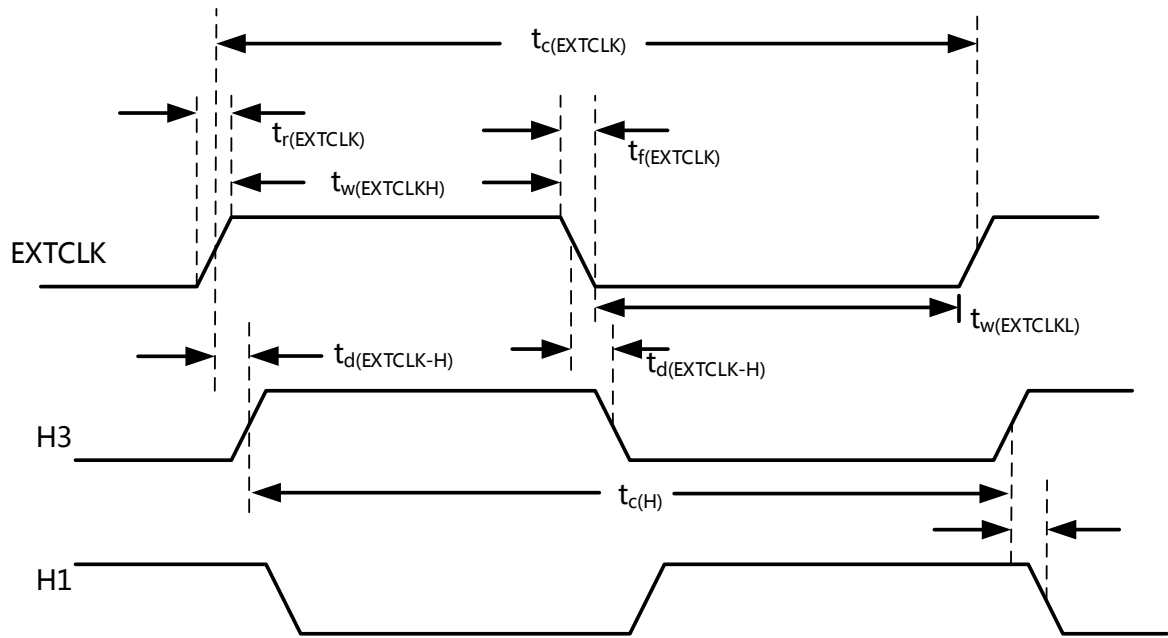


图 5.2 分为两种模式



注意：EXTCLK 持续为低

图 5.3

5.6 内存的读/写时序

下表定义了 STRB 内存的读/写时序参数，

内存的读/写时序要求

参数		AVP32VC33-120		AVP32VC33-150		单位
		最小值	最大值	最小值	最大值	
$t_{su(D-H1L)}$ 设置时间, H1 低电平之前读取数据的时间		5		5		ns
$t_{h(H1L-D)R}$ 保持时间, H1 低电平之后读取数据的时间		-1		-1		ns
$t_{su(RDY-H1H)}$ 设置时间, RDY 在 H1 高电平之前的时间		5		4		ns
$t_{h(H1H-RDY)}$ 保持时间, RDY 在 H1 高电平之后的时间		-1		-1		ns
$t_{d(A-RDY)}$ 延迟时间, 地址有效到 RDY 的时间			P-7		P-6	ns
$t_{v(A-D)}$ 有效时间, 有效数据在地址 PAGE _x 或 STRB 有效之后的时间	0 等待状态, $C_L=30PF$		9		6	ns
	1 等待状态		$t_{c(H)}+9$		$t_{c(H)}+6$	ns

(1) 这些时钟假设在所有引脚上都有类似的 30 pF 负载

(2) $P=t_{c(H)}/2$ (占空比等于 50%时)

在内存的读/写的推荐操作条件下的切换特性

参数		AVP32VC33-120		AVP32VC33-150		单位
		最小值	最大值	最小值	最大值	
$t_{d(H1L-SL)}$ 延迟时间, H1 低电平到 STRB 低电平的时间		-1	4	-1	3	ns
$t_{d(H1L-SH)}$ 延迟时间, H1 低电平到 STRB 高电平的时间		-1	4	-1	3	ns
$t_{d(H1H-RWL)W}$ 延迟时间, H1 高电平至 R/W 低电平的时间 (写入)		-1	4	-1	3	ns
$t_{d(H1L-A)}$ 延迟时间, H1 低电平到地址有效的时间		-1	4	-1	3	ns
$t_{d(H1H-RWH)W}$ 延迟时间, H1 高电平至 R/W 高电平的时间(写入)		-1	4	-1	3	ns
$t_{d(H1H-A)W}$ 延迟时间, H1 高电平到地址有效的时间 (写入)		-1	4	-1	3	ns
$t_{v(H1L-D)W}$ 有效时间, H1 低电平之后写入数据有效的时间			6		5	ns
$t_{h(H1H-D)W}$ 保持时间, H1 高电平之后写入数据有效的时间		0	5	0	5	ns

(1) 这些时序假定所有引脚上的负载相似, 为 30 PF

(2) 高速和低速 (低噪声) 输出缓冲器的输出负载特性如图 5.4 所示, 高速缓冲器用于 A0-A23 , PAGE0-PAGE3 , H1 ,

(3) H3 , STRB 和 R/W , 其他输出使用低速 (低噪声) 输出缓冲区。

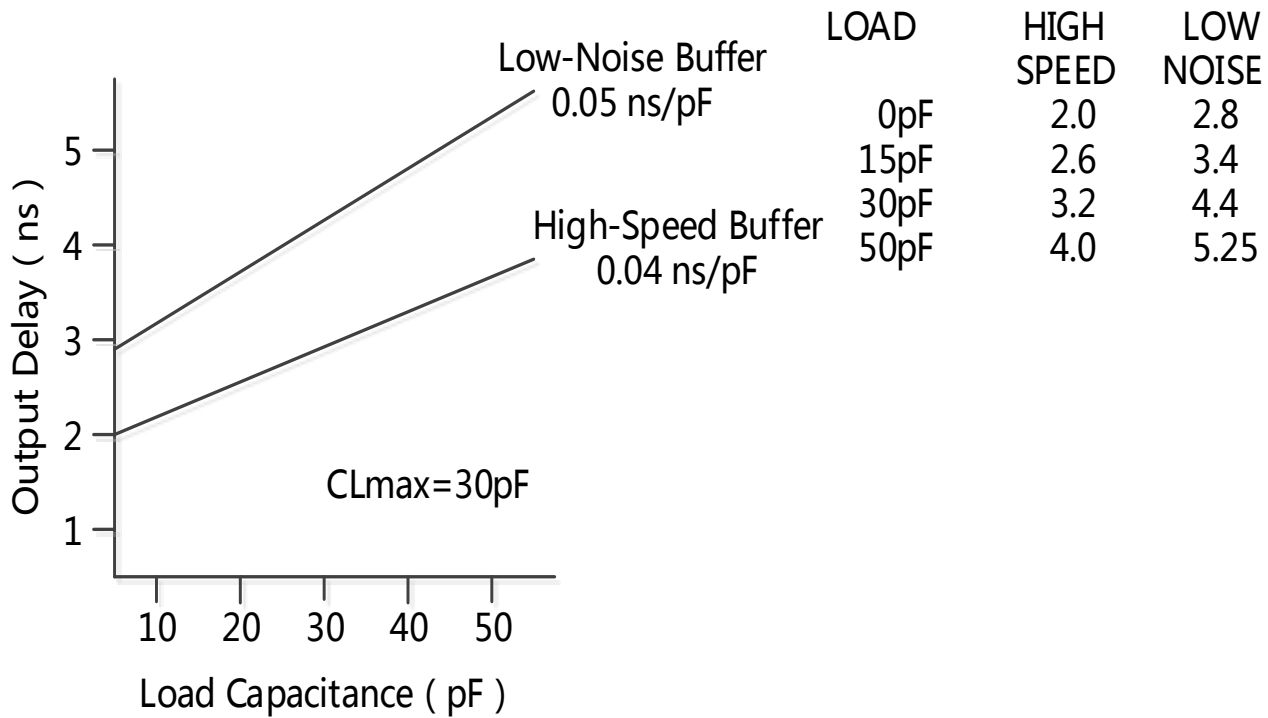
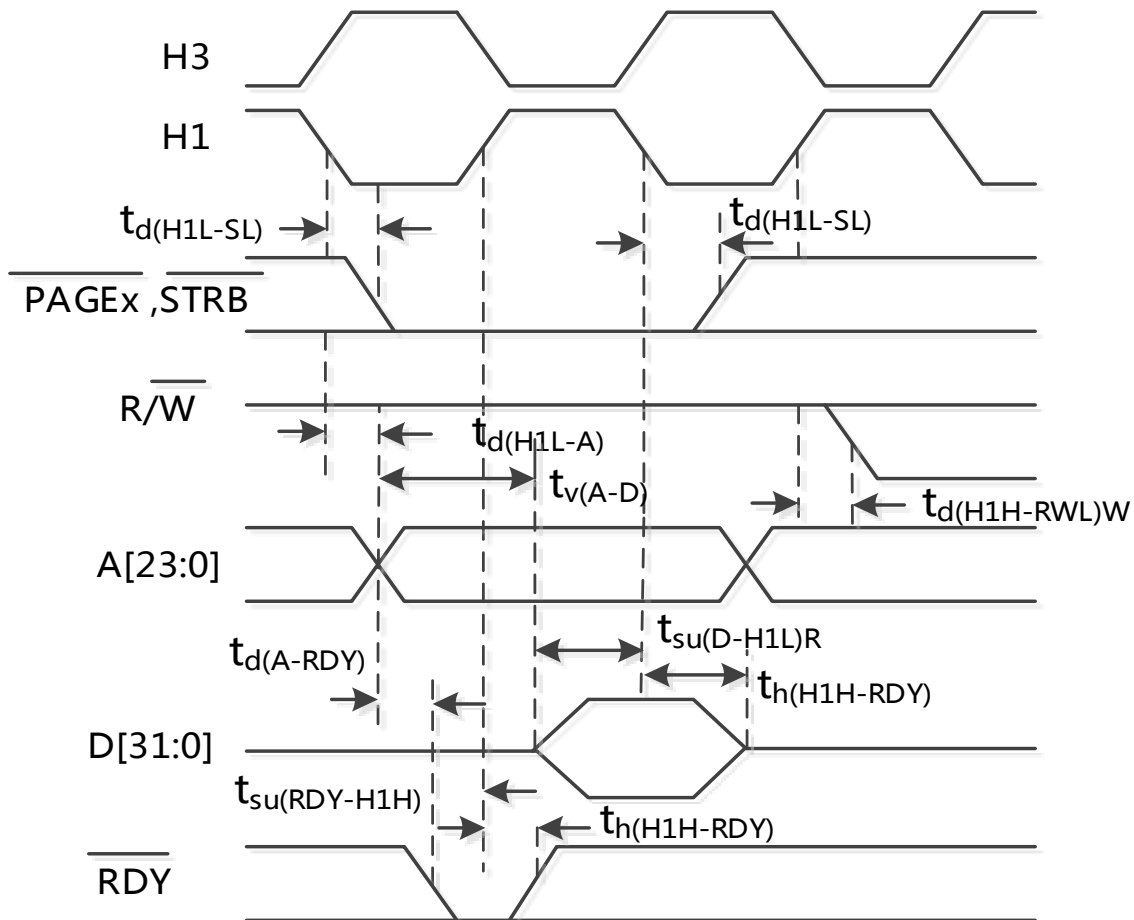


图 5.4 输出负载特性，仅限缓冲器

内存的读/写时序图：



注意：在背对背读取操作期间，STRB 保持低电平

图 5.5 内存时间 (STRB=0 和 PAGEx=0) 读取

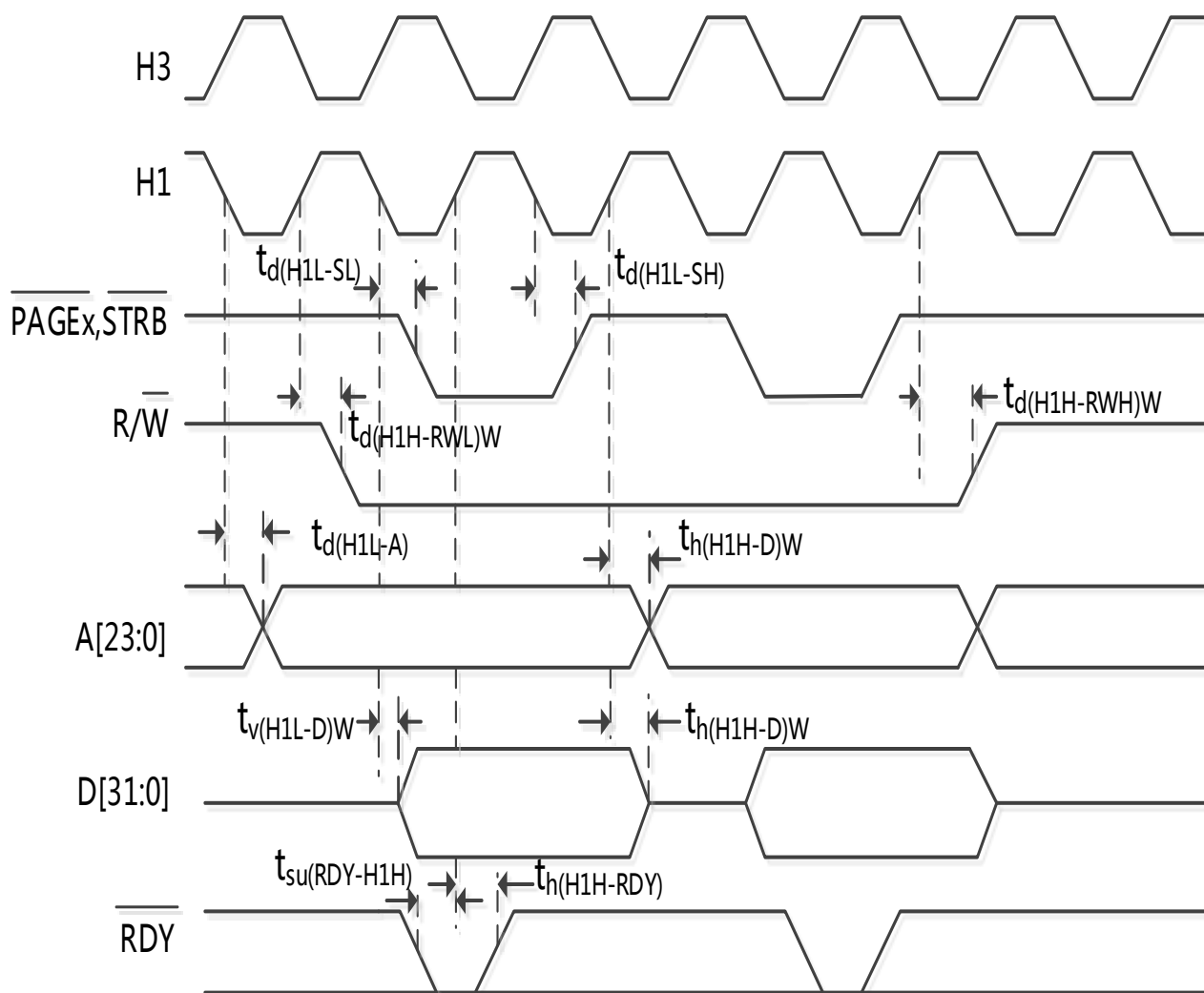


图 5.6 内存时序 (STRB=0 和 PAGES=0) 写入

5.7 执行 LDFI 和 LDII 时的 XF0 和 XF1 时序

执行 LDFI 或 LDII 期间 XF0 和 XF1 时序

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(XF1-H1L)}$ 设置时间,XF1 在 H1 低电平之前有效的时 间	5		4		ns
$t_{h(H1L-XF1)}$ 保持时间 , XF1 在 H1 低电平之后有效的时 间	0		0		ns

在执行 LDFI 和 LDII 时，XF0 和 XF1 在推荐操作条件下的切换特性

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{d(H3H-XF0L)}$ 延迟时间 ,H3 高电平至 XF0 低电平的时间		4		3	ns

执行 LDFI 或 LDII 时 XF0 和 XF1 的时序要求如图 5.6 所示：

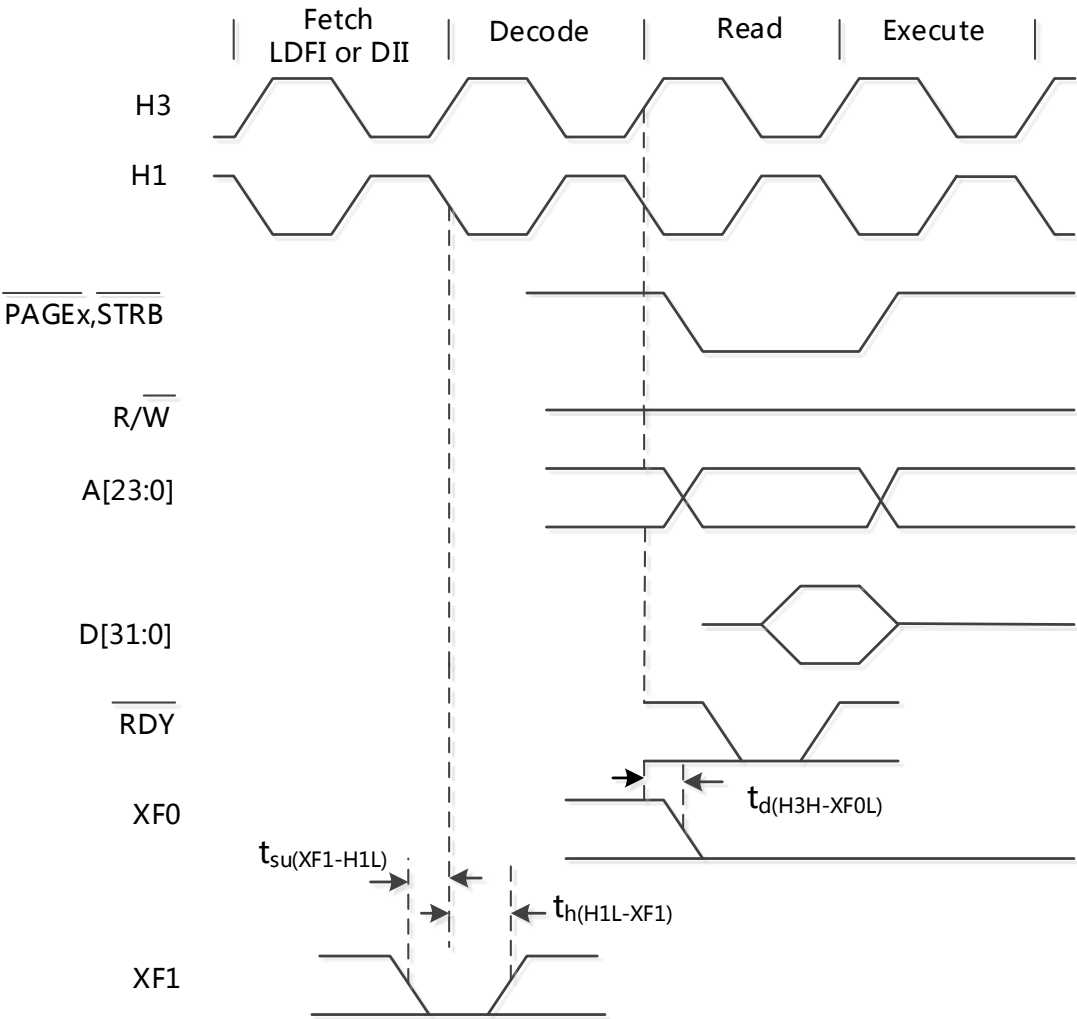


图 5.7 执行 LDFI 或 LDII 时 XF0 和 XF1 的时序

5.8 执行 STFI 和 STII 时的 XF0 的时序

下表定义了执行 STFI 或 STII 期间 XF0 引脚的时序参数，在执行 STFI 或 STII 时，XF0 在推荐操作条件下切换特性如图 5.7。

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{d(H3H-XF0H)}$ 延迟时间，H3 高电平至 XF0 高电平的时间		4		3	ns

在互锁存储指令的执行阶段开始时，XF0 时钟设置为高电平。当没有发生管道冲突时，存储地址也在互锁存储指令的执行阶段开始时被驱动。但是，如果管道冲突阻止了存储指令从执行开始，存储的地址将不会被驱动，直到存储可以执行。

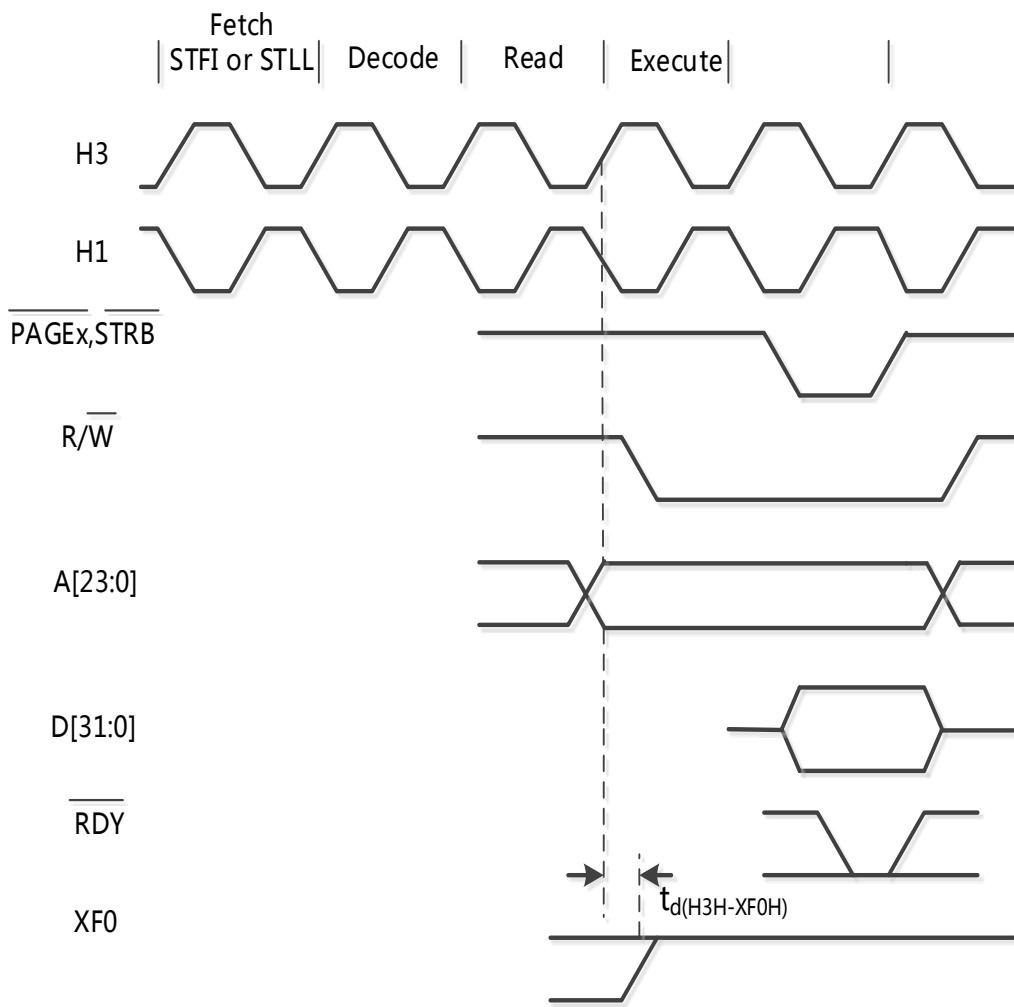


图 5.8 执行 STFI 和 STII 时的 XF0 的时序

5.9 执行 SIGI 时的 XF0 和 XF1 时序

下表定义了执行 SIGI 期间 XF0 和 XF1 引脚的时序参数。

执行 SIGI 时的 XF0 和 XF1 时序要求

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(XF1-H1L)}$ 设置时间,XF1 在 H1 低电平之前有效的时 间	5		4		ns
$t_{h(H1L-XF1)}$ 保持时间, XF1 在 H1 低电平之后有效的时 间	0		0		ns

在执行 SIGI 时，XF0 和 XF1 在推荐操作条件下切换特性

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{d(H3H-XF0L)}$ 延迟时间，H3 高电平至 XF0 低电平的时 间		4		3	ns
$t_{d(H3H-XF0H)}$ 延迟时间，H3 高电平至 XF0 高电平的时 间		4		3	ns

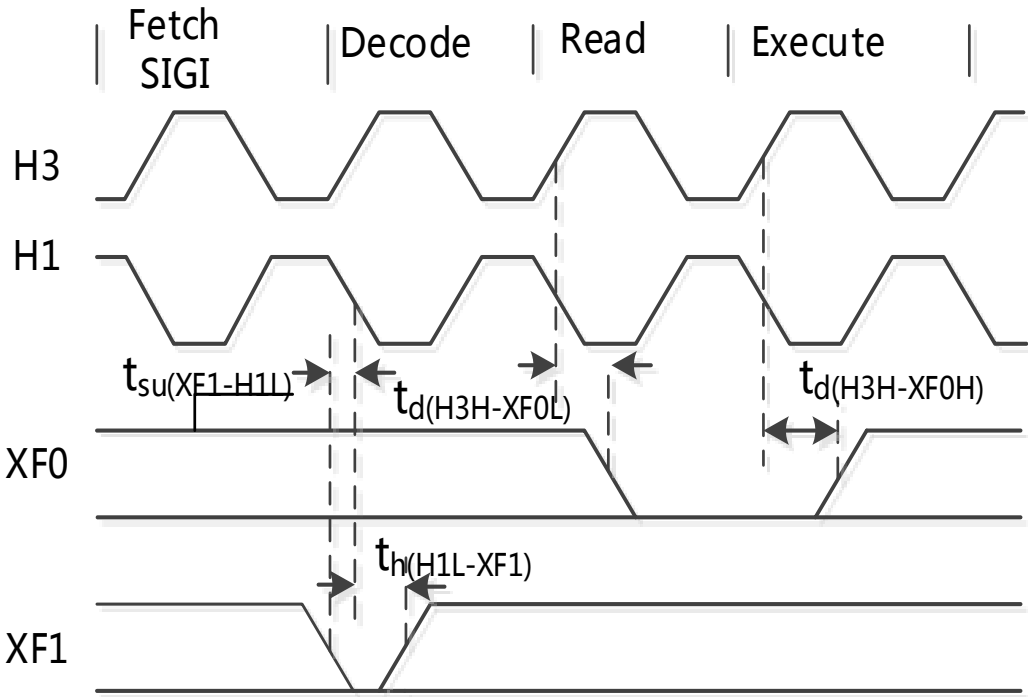
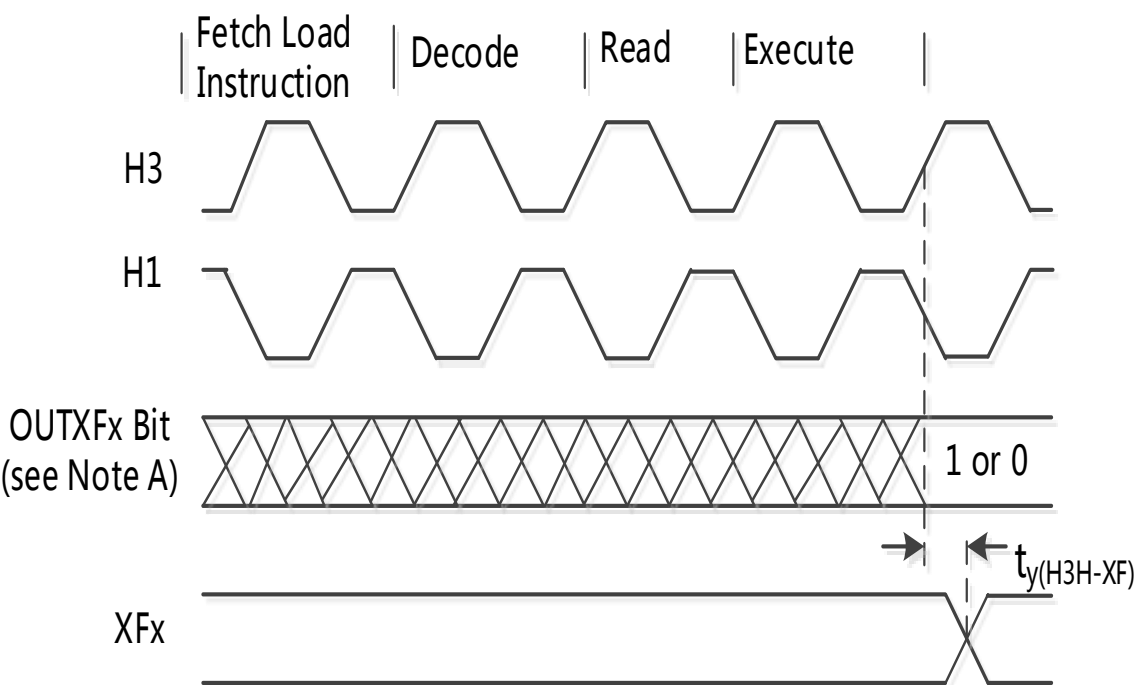


图 5.9 执行 SIGI 时的 XF0 和 XF1 时序

5.10 XFx 配置为输出时加载

下表定义了 XFx 引脚配置为输出时，加载 XF 寄存器的时序参数，在推荐的工作条件下切换特性，以便在加载时 XF 寄存器配置为输出引脚。

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{V(H3H-XF)}$ 有效时间，XFx 在 H3 高电平之后有效的时 间		4		3	ns



注意：OUTXFx 表示 IOF 寄存器的第 2 位或第 6 位

图 5.10 配置为输出引脚时，加载 XF 寄存器的时序

5.11 将 XFx 从输出模式更改为输入模式时加载

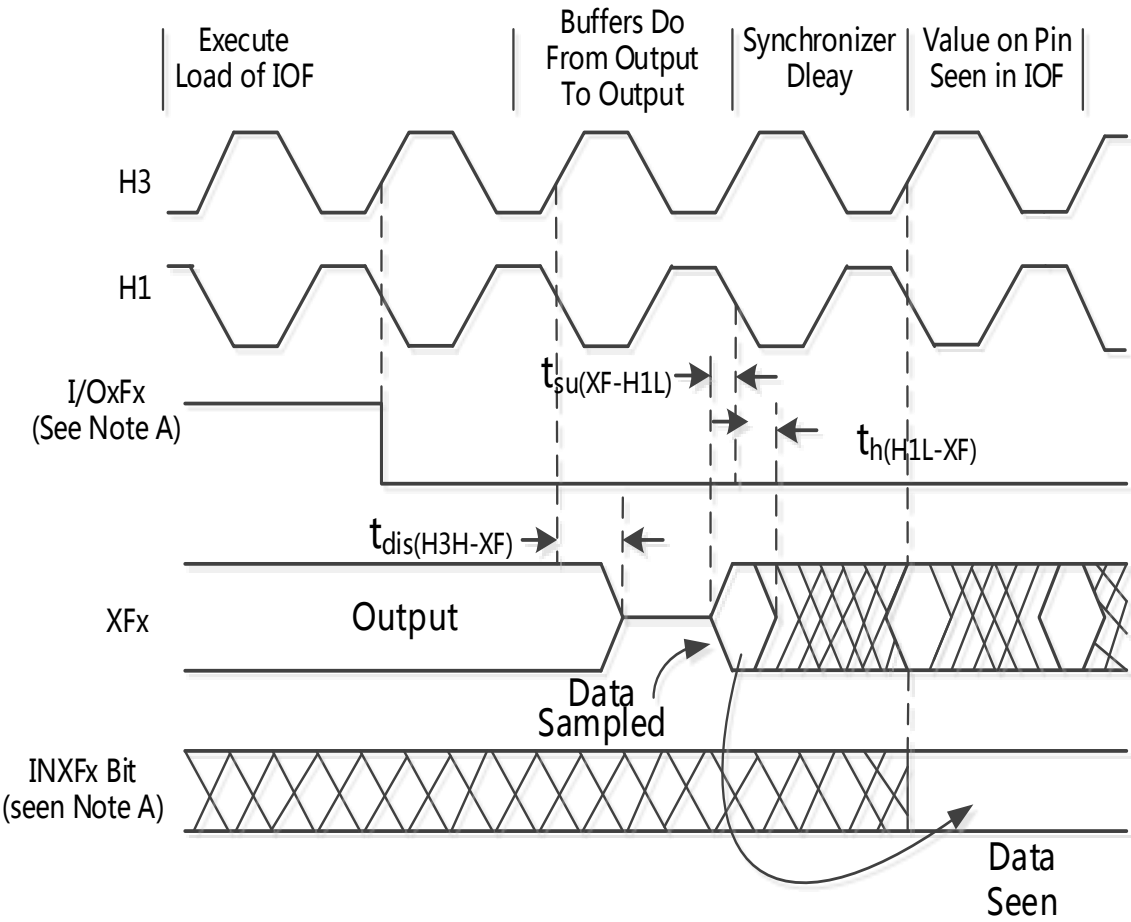
下表定义了将 XFx 引脚从输出模式更改为输入模式的时序参数。

将 XFx 从输出模式更改为输入模式的时序要求

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(XF-H1L)}$ 设置时间,XF _x 在 H1 低电平之前有效的时间	5		4		ns
$t_{h(H1L-XF)}$ 保持时间,XF _x 在 H1 低电平之后有效的时间	0		0		ns

在推荐的工作条件下切换特性，将 XFx 从输出模式更改为输入模式

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{dis(H3H-XF)}$ 禁用时间,XF _x 在 H3 高电平之后有效的时间		6		5	ns



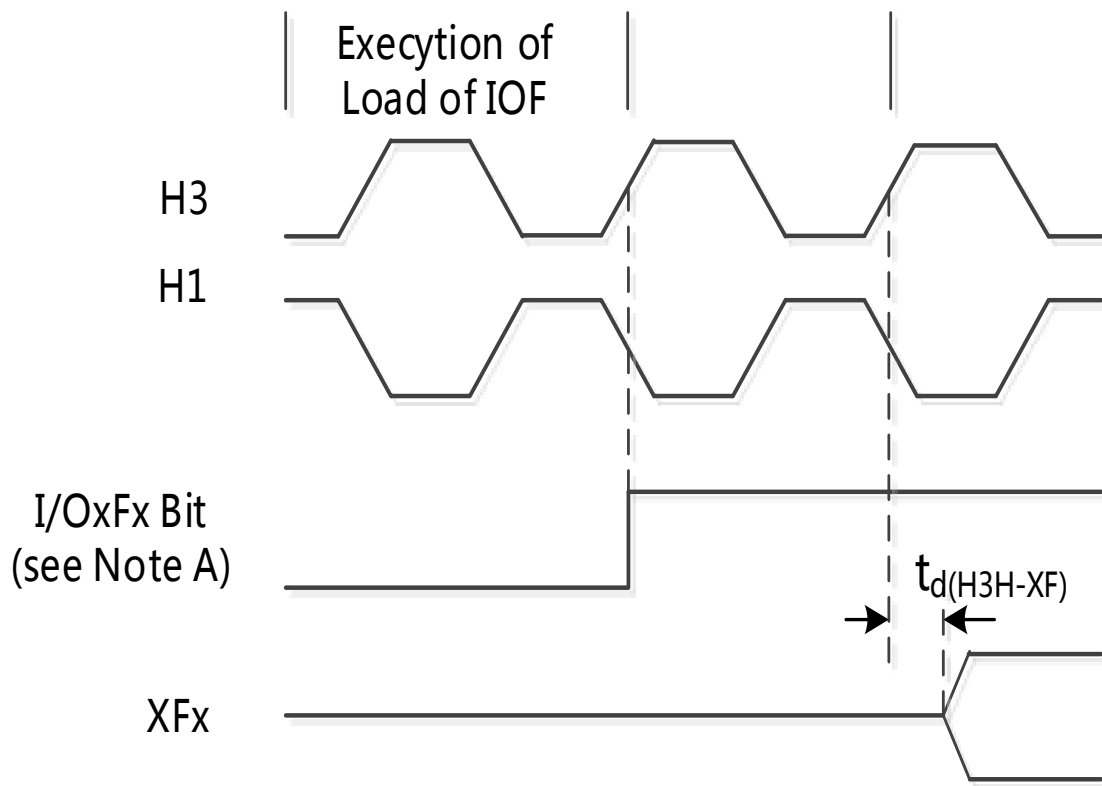
注意：I/OxFx 表示 IOF 寄存器的第 1 位或第 5 位，INXFx 表示 IOF 寄存器的第 3 位或第 7 位

图 5.11 将 XFx 从输出更改为输入模式的时序

5.12 将 XFx 从输入模式更改为输出模式

下表定义了将 XFx 引脚从输入模式更改为输出模式的时序参数。
在推荐的操作条件下切换特性，以便将 XFx 从输入模式更改为输出模式。

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{d(H3H-XF)}$ 延迟时间，H3 高电平至 XFx 从输入到输出切换的时间		4		3	ns



注意：I/OxFx 表示 IOF 寄存器的第 1 位或第 5 位
图 5.12 将 XFx 从输入模式更改为输出模式

5.13 复位时序

RESET 时一个异步输入，可以在时钟周期内随时置位。如果指定的时间满足，则出现图 5.13 所示的确切顺序，否则可能出现一个时钟周期的额外延迟

异步复位信号包括 XF0/1、CLKX0、DX0、FSX0、CLKR0、DR0、FSR0 和 TCLK0/1。复位器件会将总线控制寄存器初始化为七个软件等待状态，从而导致外部访问速度变慢，直到这些寄存器初始化。

HOLD 时一个同步输入，可以在置位期间置位，在 HOLD 被授予之前需要 9 个 CPU 周期

RESET 信号的时序参数

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su}(RESET-EXTCLK)$ 设置时间，RESET 在 EXTCLK 低电平之前有效的时间	6	P-7	5	P-7	ns
$T_{su}(RESETH-H1L)$ 设置时间，RESET 高电平在 H1 低电平之前的时间，在 H1 时钟周期之后复位	6		5		ns

$P=t_c(EXTCLK)$

在 RESET 的推荐工作条件下的切换特性

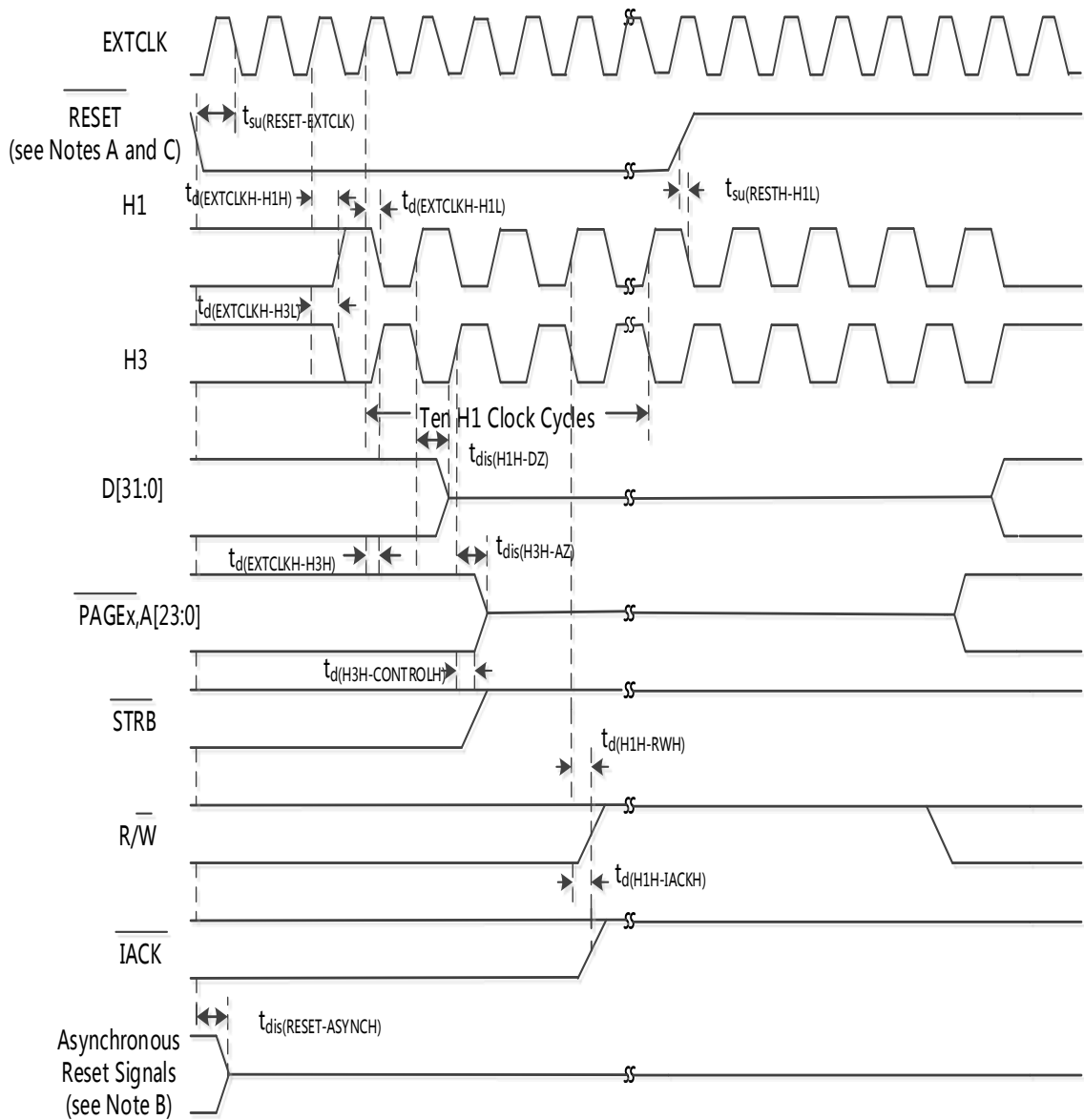
参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_d(EXTCLKH-H1H)$ 延迟时间，EXTCLK 高电平至 H1 高电平的时间	2	7	2	7	ns
$t_d(EXTCLKH-H1L)$ 延迟时间，EXTCLK 高电平至 H1 低电平的时间	2	7	2	7	ns
$t_d(EXTCLKH-H3L)$ 延迟时间，EXTCLK 高电平至 H3 低电平的时间	2	7	2	7	ns
$t_d(EXTCLKH-H3H)$ 延迟时间，EXTCLK 高电平至 H3 高电平的时间	2	7	2	7	ns
$t_{dis}(H1H-DZ)$ 禁用时间，从 H1 高电平得到数据（高阻抗）的时间		7		6	ns
$t_{dis}(H3H-AZ)$ 禁用时间，从 H3 高电平得到地址（高阻抗）的时间		7		6	ns
$t_d(H3H-CONTROLH)$ 延迟时间，H3 高至控制信号高		4		3	ns
$t_d(H1H-RWH)$ 延迟时间，H1 高电平至 R/W 高电平的时间		4		3	ns
$t_d(H1H-IACKH)$ 延迟时间，H1 高电平至 IACK 高电平的时间		4		3	ns
$t_{dis}(RESETL-ASYNCH)$ 禁用时间，从 RESET 低电平得到禁用（高阻抗）异步复位信号		7		6	ns

Dbus 的高阻抗仅限于标称保持器 $Z_{OUT}=15k\Omega$

异步复位信号包括 XF0/1、CLKX0、DX0、FSX0、CLKR0、DR0、FSR0 和 TCLK0/1。

RESET 的时序要求见图 5.13

AVP32VC33 Digital Signal Processor



- 注意：
- a、时钟电路配置为 C31 兼容的 2 分频模式，如果配置为 x1 模式，EXTCLK 直接驱动 H3；
 - b、异步复位信号包括 XF0/1、CLKX0、DX0、FSX0、CLKR0、DR0、FSR0 和 TCLK0/1。
 - c、在微处理模式下，复位向量被提取两次，每次有七个软件等待状态，在为计算机引导模式下，复制向量被提取两次，没有软件等待状态；
 - d、在复位期间，地址和 PAGE0-PAGE3 输出处于高阻态，需要标称 10-22kΩ上拉，如果不是，则在不驱动这些输出时可能发生不希望的伪读。

图 5.13 复位时序

5.14 中断响应时序

下表定义了 INTx 信号的时序参数

参数	AVP32VC33-120			AVP32VC33-150			单位
	最小值	典型值	最大值	最小值	典型值	最大值	
$t_{su(INT-H1L)}$ 设置时间, INT3~INT1 在 H1 低电平之前有效的的时间	5			4			ns
$t_{h(H1L-INT)}$ 保持时间, INT3~INT1 在 H1 低电平之后有效的的时间			0			0	ns
$t_w(INT)$ 脉冲持续时间, 中断只保证一个中断	P+5	1.5P	2P-5	P+5	1.5P	2P-5	ns

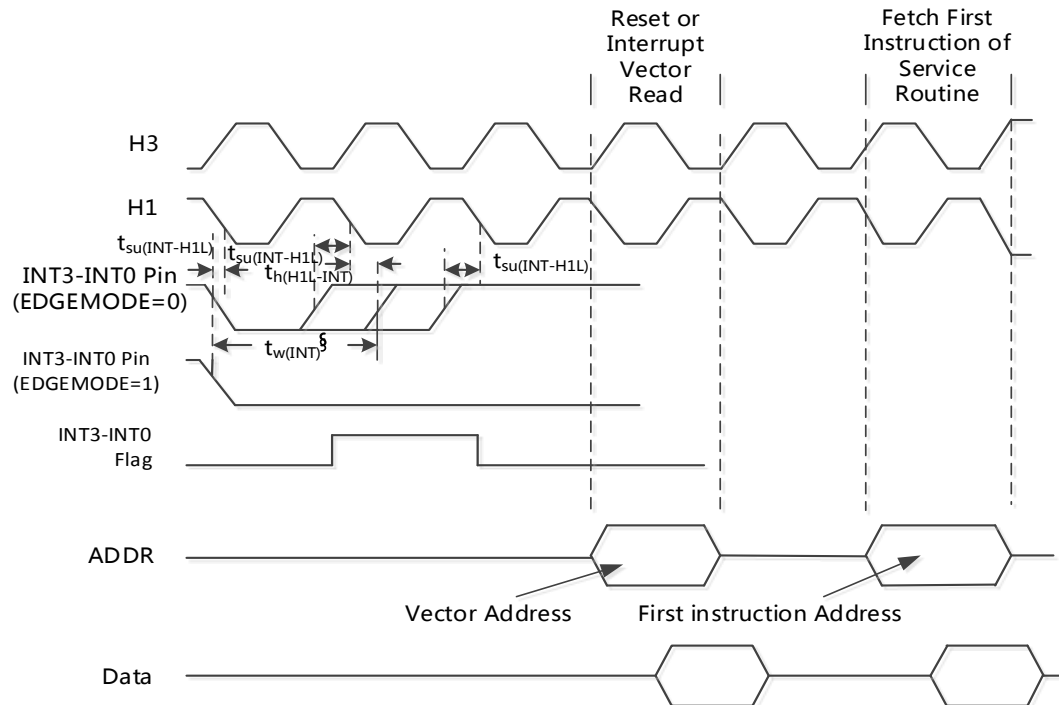
$P=t_{c(H)}$

中断(INTx)引脚上同步输入 ,可在时钟周期内随时置位 ,AVP32VC33 中断可选择为电平或边沿触发 ,在下降沿检测到中断 H1 ,因此 ,必须设置中断并保持在内部 H1 的下降沿以进行正确检测 ,该 CPU 和 DMA 仅响应指令获取边界上检测到的中断。

为使处理器在选择电平模式时仅识别一个中断 ,必须设置并保持中断脉冲 ,以便发生逻辑低电平状态 :

- (1) 至少 H1 下降沿 ;
- (2) 不超过两个 H1 下降沿 ;
- (3) 无法确保其边沿满足 H1 下降沿建立和保持时间的中断源必须进一步限制脉冲宽度 ,如上表中的 $t_w(INT)$ 所定义。

当 EDGEMODE=1 时 ,使用同步逻辑检测 INT0-INT3 引脚的下降沿。脉冲低和高时间应为两个 CPU 时钟或更高。AVP32VC33 可以在具有两个 H1 时钟周期后尽快设置来自同一源的中断标志被清除了。



- (1) H1 的下降沿仅检测到 INTx 下降沿
- (2) H1 的下降沿检测到第二个 INTx 为低电平 ,但标志清除优先。当 INTx 上升时 , H1 的下降沿错过先前的 INTx 低电平

图 5.14 INT3-INT0 响应时序

5.15 中断确认时序

IACK 输出在 IACK 指令的解码阶段的前半个周期（H1 上升）变为有效，并在 IACK 指令的读取阶段的前半个周期（H1 上升）变为无效。

下表定义了 IACK 信号的时序参数

在 IACK 的推荐工作条件下的切换特性

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{d(H1H-IACKL)}$ 延迟时间，H1 高电平至 IACK 低电平	-1	4	-1	3	ns
$t_{d(H1H-IACKH)}$ 延迟时间，H1 高电平至 IACK 高电平	-1	4	-1	3	ns

注意：IACK 指令可以随时执行，以使用 IACK 引脚发出事件信号，IACK 指令通常在中断例程中用于指示发生哪个中断，必须执行 IACK 指令才能产生 IACK 脉冲。

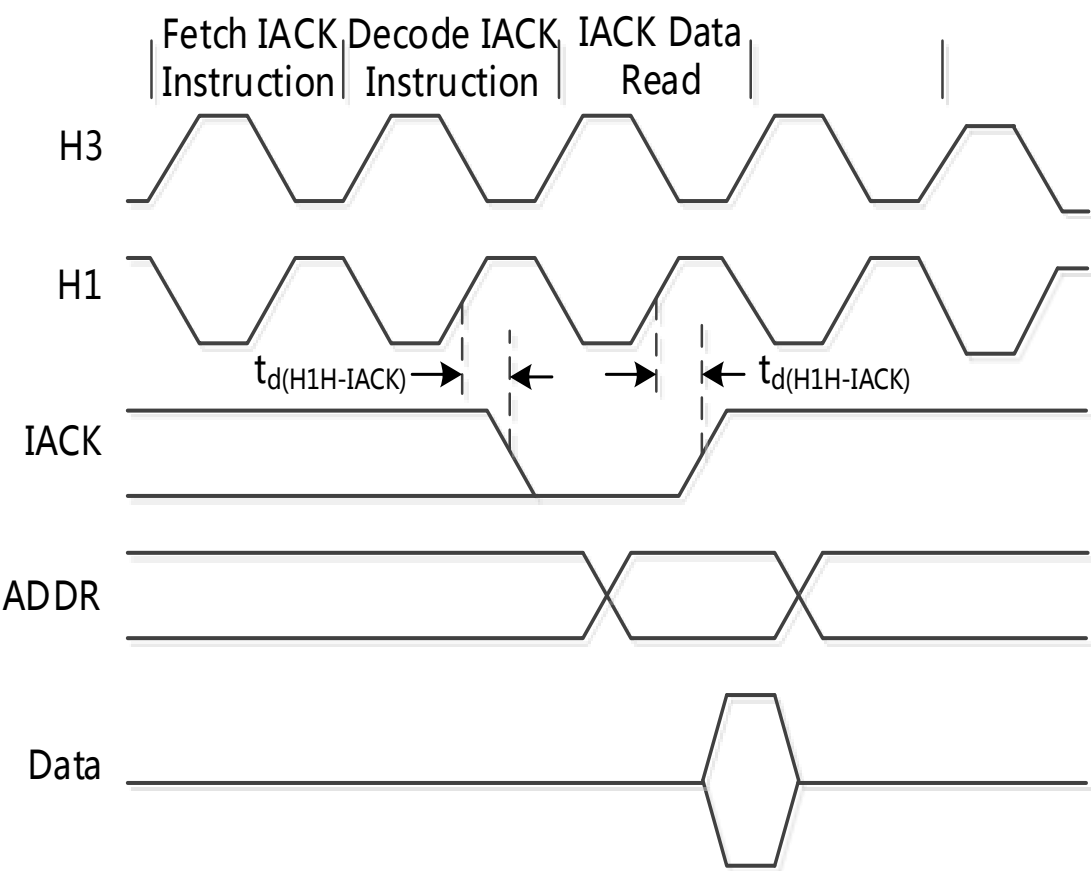


图 5.15 中断确认（IACK）时序

5.16 串口定时时序

下表定义了串行端口的时序参数。

时序要求

参数		最小值	最大值	单位
$t_{c(SCK)}$ 循环时间, CLKX/R	CLKX/R ext	$t_{c(H)} * 2.6$		ns
	CLKX/R int	$t_{c(H)} * 2$	$t_{c(H)} * 2^{16}$	
$t_{w(SCK)}$ 脉冲持续时间, CLKX/R 高/低电平的时间	CLKX/R ext	$t_{c(H)} + 5$		ns
	CLKX/R int	$[t_{c(SCK)}/2] - 4$	$[t_{c(SCK)}/2] + 4$	
$t_{r(SCK)}$ 上升时间, CLKX/R			3	ns
$t_{f(SCK)}$ 下降时间, CLKX/R			3	ns
$t_{su(DR-CLKRL)}$ 设置时间, DR 在 CLKR 低电平之前的时间	CLKR ext	4		ns
	CLKR int	5		
$t_{h(CLKRL-DR)}$ 保持时间, DR 在 CLKR 低电平之后的时间	CLKR ext	3		ns
	CLKR int	0		
$t_{su(FSR-CLKRL)}$ 设置时间, FSR 在 CLKR 低电平之前的时间	CLKR ext	4		ns
	CLKR int	5		
$t_{h(SCKL-FS)}$ 保持时间, FSX/R 在 CLKX/R 低电平后输入的时间	CLKX/R ext	3		ns
	CLKX/R int	0		
$t_{su(FSX-CLKX)}$ 设置时间, 外部 FSX 在 CLKX 之前的时间	CLKX ext	$-[t_{c(H)} - 6]$	$[t_{c(SCK)}/2] - 6$	ns
	CLKX int	$-[t_{c(H)} - 10]$	$t_{c(SCK)}/2$	

在推荐的操作条件下的切换特性

参数		最小值	最大值	单位
$t_{d(H1H-SCK)}$ 延迟时间, H1 高电平至内部 CLK/R 的时间			4	ns
$t_{d(CLKX-DX)}$ 延迟时间, CLKX 到 DX 有效	CLKX ext		6	ns
	CLKX int		5	
$t_{d(CLKX-FSX)}$ 延迟时间, CLKX 到内部 FSX 有效的时间	CLKX ext		5	ns
	CLKX int		4	
$t_{d(CLKX-DX)V}$ 延迟时间, CLKX 到第一个 DX 位之前, FSX 在 CLKX 高电平之前	CLKX ext		5	ns
	CLKX int		4	
$t_{d(FSX-DX)V}$ 延迟时间, FSX 到第一个 DX 位, CLKX 在 FSX 之前的时间			6	ns
$t_{dis(CLKX-DXZ)}$ 禁用时间, 从 CLKX 高电平的最后一个数据位开始 DX 高阻抗			6	ns

5.18 HOLD 时序

HOLD 是一个同步输入，可以在一个时钟周期内随时置位。如果满足指定的时序，则会出现图 5.18 和图 5.19 所示的确切序列，否则，可能出现一个时钟的额外延迟循环。

表 “HOLD/HOLDA 的时序参数” 中定义了 HOLD 和 HOLDA 信号的时序参数。主总线控制寄存器的 NOHOLD 位会覆盖 HOLD 信号，该位置位时，设备不在保持状态，可防止将来的保持周期。

断言 HOLD 可防止处理器访问主总线，程序继续执行，直到请求读取或写入主总线为止。在某些情况下，第一次写入时未察觉的，因此允许处理器继续（内部）直到遇到第二次外部写入。

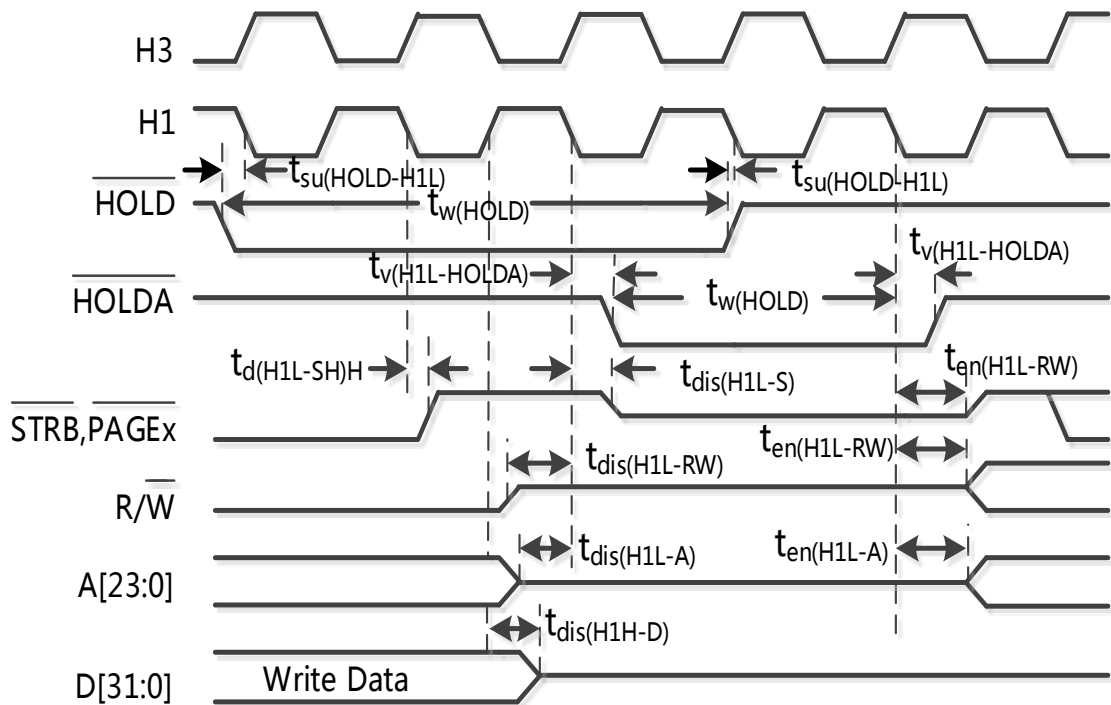
图 5.18 和图 5.19 以及随附的时序用于零等待状态总线配置，由于内部循环之前的一个周期，CPU 在 H1 下降沿内部捕获 HOLD，因此，任何总线配置的最小 HOLD 宽度为 WTCNT3。此外，在 HOLDA 至少激活一个周期之前，HOLD 不应该被置为无效。

HOLD/HOLDA 时序要求

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(HOLD-H1L)}$ 设置时间,HOLD 在 H1 低电平之前的时间	4		3		ns
$t_{w(HOLD)}$ 脉冲持续时间，HOLD 低电平的时间	$3t_{c(H)}$		$3t_{c(H)}$		ns

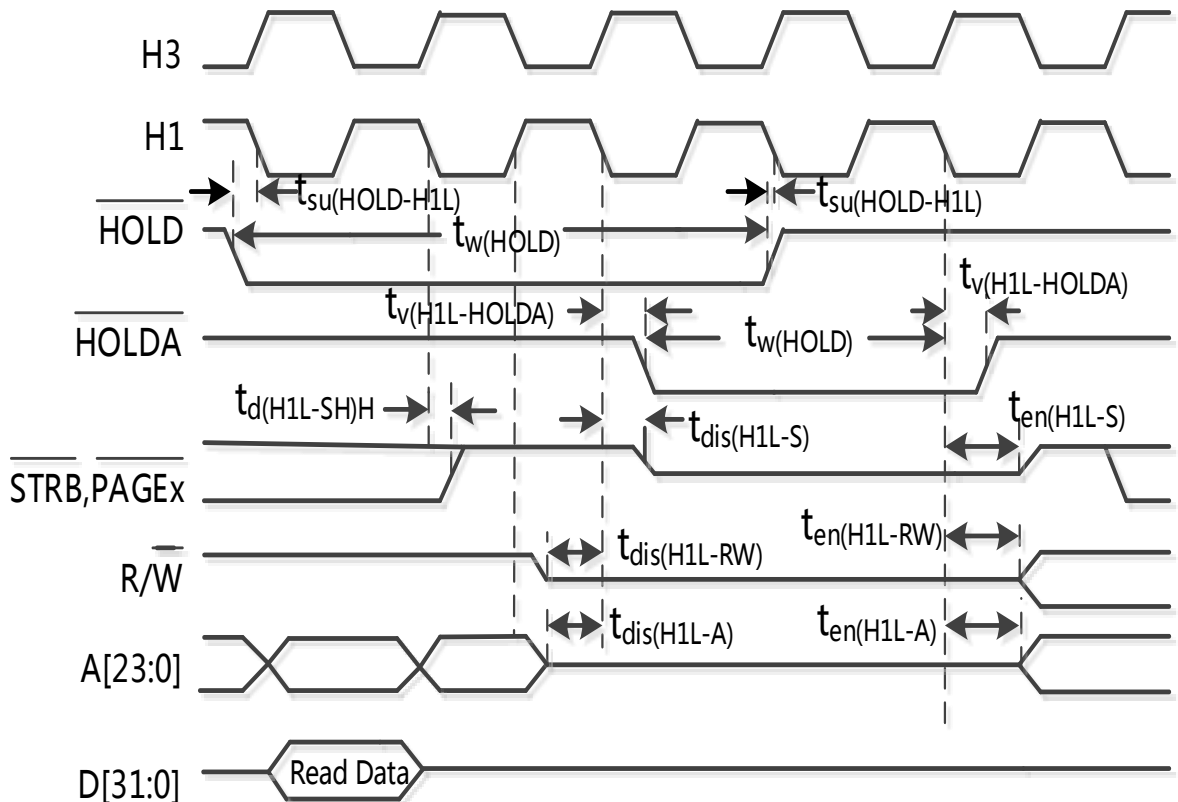
在 HOLD/HOLDA 的推荐工作条件下的切换特性

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{v(H1L-HOLDA)}$ 有效时间,HOLDA 在 H1 低电平之后的时间	-1	4	-1	3	ns
$t_{w(HOLDA)}$ 脉冲持续时间,HOLDA 低电平的时间	$2t_{c(H)}-4$		$2t_{c(H)}-4$		ns
$t_{d(H1L-SH)}_H$ 延迟时间,HOLD 从 H1 低电平到 STRB 高电平的时间	-1	4	-1	3	ns
$t_{dis(H1L-S)}$ 禁用时间,STRB 从 H1 低电平变为高阻态的时间		5		4	ns
$t_{en(H1L-S)}$ 启用时间,STRB 从 H1 低电平启用（有效）的时间		5		5	ns
$t_{dis(H1L-RW)}$ 禁用时间,R/W 从 H1 低电平至高阻态的时间		5		4	ns
$t_{en(H1L-RW)}$ 启用时间,R/W 从 H1 低电平启用（有效）的时间		5		5	ns
$t_{dis(H1L-A)}$ 禁用时间,地址从 H1 低电平到高阻态的时间		5		4	ns
$t_{dis(H1L-A)}$ 启用时间,地址从 H1 低电平启用（有效）的时间		5		5	ns
$t_{dis(H1H-D)}$ 禁用时间,数据从 H1 高电平变为高阻态的时间		5		4	ns



注意：HOLDA 变为低电平以响应 HOLD 变低并继续保持低电平，直到 HOLD 后的一个 H1 周期回升为高电平

图 5.18 HOLD/HOLDA 的时序（写入后）



注意：HOLDA 变为低电平以响应 HOLD 变低并继续保持低电平，直到 HOLD 后的一个 H1 周期回升为高电平

图 5.19 HOLD/HOLDA 的时序（读取后）

5.19 通用 I/O 时序

外设引脚包括 CLKX0、CLKR0、DX0、DR0、FSR0 和 TCLK0/1。与每个外设相关的内部控制寄存器的内容定义了这些引脚的模式。

下表显示了将外设引脚从通用输出引脚更改为通用输入引脚的时序参数，反之亦然。

外设引脚通用 I/O 的时序要求

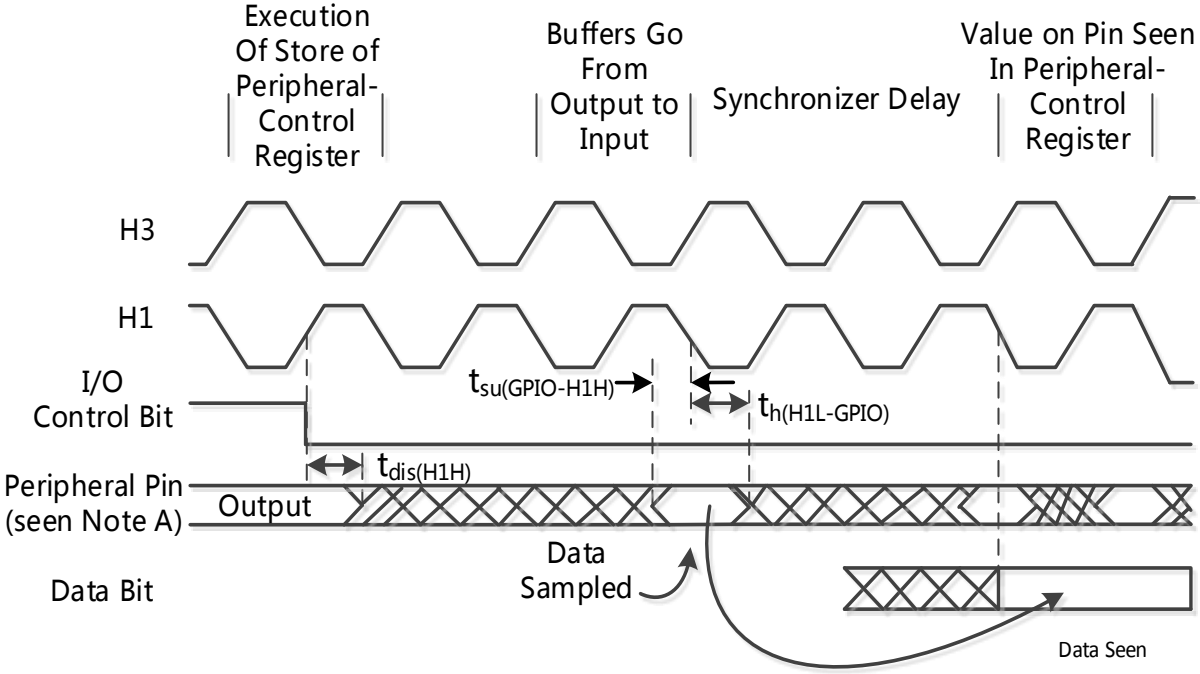
参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(GPIO-H1L)}$ 设置时间 ,H1 低电平之前通用输入的时间	4		3		ns
$t_{h(H1L-GPIO)}$ 保持时间 , H1 低电平之后通用输入的时间	0		0		ns

注意：外设引脚包括 CLKX0、CLKR0、DX0、DR0、FSR0 和 TCLK0/1。这些引脚的模式由与每个外设相关的内部控制寄存器的内容定义。

外围通用 I/O 引脚的推荐工作条件下的切换特性

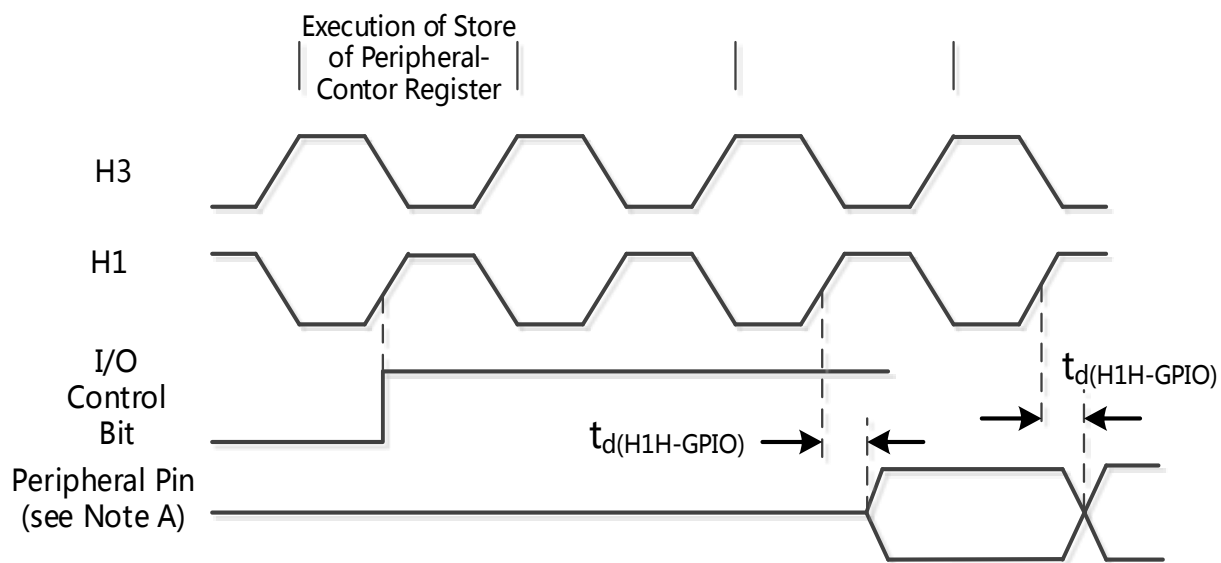
参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(H1H-GPIO)}$ 延迟时间 , H1 高电平到通用输出的时间		5		4	ns
$t_{h(H1H)}$ 禁用时间 , 从 H1 高电平得到通用输出的时间		7		5	ns

注意：外设引脚包括 CLKX0、CLKR0、DX0、DR0、FSR0 和 TCLK0/1。这些引脚的模式由与每个外设相关的内部控制寄存器的内容定义。



注意：外设引脚包括 CLKX0、CLKR0、DX0、DR0、FSR0 和 TCLK0/1。

图 5.20 外设引脚从通用输出更改为输入模式时序



注意：外设引脚包括 CLKX0、CLKR0、DX0、DR0、FSR0 和 TCLK0/1。
图 5.21 外设引脚从通用输入更改为输出模式时序

5.20 定时器引脚时序

有效的逻辑电平周期和极性由内部控制寄存器的内容指定。下表定义了定时器引脚的时序参数。

定时器引脚的时序要求

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(TCLK-H1L)}$ 设置时间，外部 TCLK 在 H1 低电平之前的时间	4		3		ns
$t_{h(H1L-TCLK)}$ 保持时间，外部 TCLK 在 H1 低电平之后的时间	0		0		ns

这些要求适用于同步输入时钟

定时器引脚的推荐工作条件下的切换特性

参数		AVP32VC33-120		AVP32VC33-150		单位
		最小值	最大值	最小值	最大值	
$t_{d(H1H-TCLK)}$ 延迟时间 ,H1 高电平至 TCLK 内部有效的时间			4		3	ns
$t_c(TCLK)$ 周期时间，TCLK	TCLK ext	$t_{c(H)}*2.6$		$t_{c(H)}*2.6$		ns
	TCLK int	$t_{c(H)}*2$	$t_{c(H)}*2^{32}$	$t_{c(H)}*2$	$t_{c(H)}*2^{32}$	
$t_w(TCLK)$ 脉冲持续时间，TCLK	TCLK ext	$t_{c(H)}+6$		$t_{c(H)}+5$		ns
	TCLK int	$[t_{c(TCLK)}/2]-4$	$[t_{c(TCLK)}/2]+4$	$[t_{c(TCLK)}/2]-4$	$[t_{c(TCLK)}/2]+4$	

这些要求适用于同步输入时钟

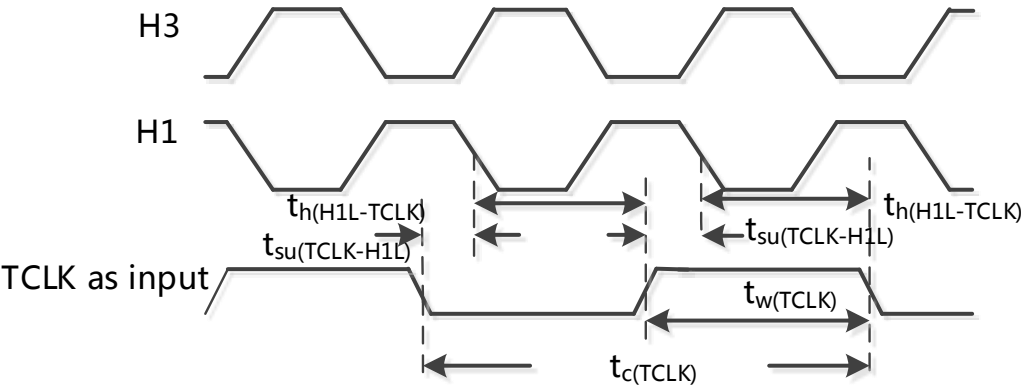


图 5.21 定时器引脚输入时序

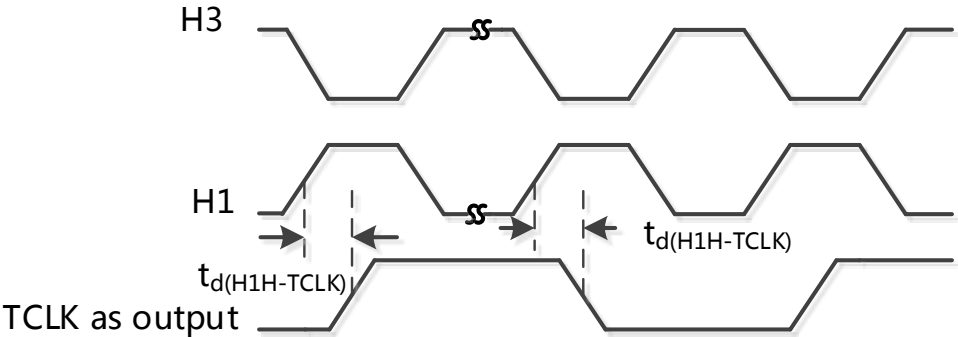


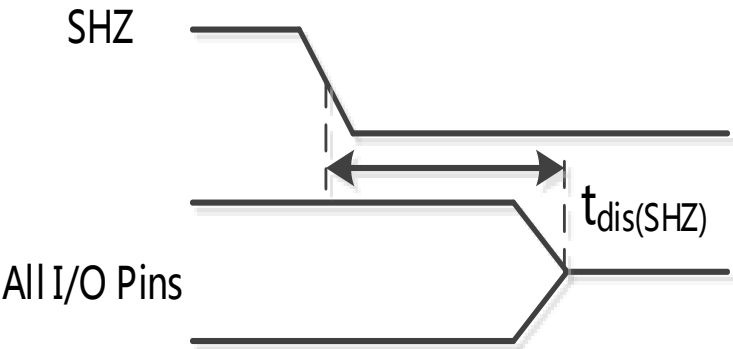
图 5.22 定时器引脚输出时序

5.21 SHZ 引脚时序

下表定义了 SHZ 引脚的时序参数

在 SHZ 的推荐工作条件下切换特性

参数	最小值	最大值	单位
$t_{dis(SHZ)}$ 禁用时间，SHZ 低电平至所有输出，I/O 引脚禁用（高阻抗）的时间	0	8	ns



注意：启用 SHZ 会破坏 AVP32VC33 寄存器和存储器内容。置位 SHZ=1 并复位 AVP32VC33 以将其恢复到已知状态。

图 5.23 SHZ 时序

5.22 IEEE-1149.1 测试访问端口时序

下表定义了 IEEE-1149.1 测试访问端口的时序参数。

IEEE-1149.1 测试访问端口时序

参数	AVP32VC33-120		AVP32VC33-150		单位
	最小值	最大值	最小值	最大值	
$t_{su(TMS-TCKH)}$ 设置时间，TMS/TDI 到 TCK 高电平的时间	5		5		ns
$t_h(TCKH-TMS)$ 保持时间，TMS/TDI 从 TCK 高电平保持的时间	5		5		ns
$t_d(TCKL-TDOV)$ 延迟时间，TCK 低电平到 TDO 有效的时间	0	10	0	10	
$t_r(TCK)$ 上升时间，TCK		3		3	
$t_f(TCK)$ 下降时间，TCK		3		3	

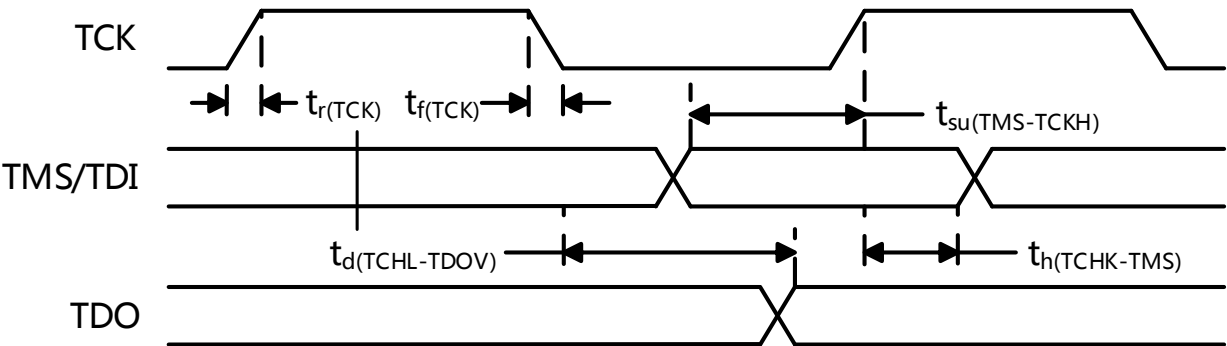
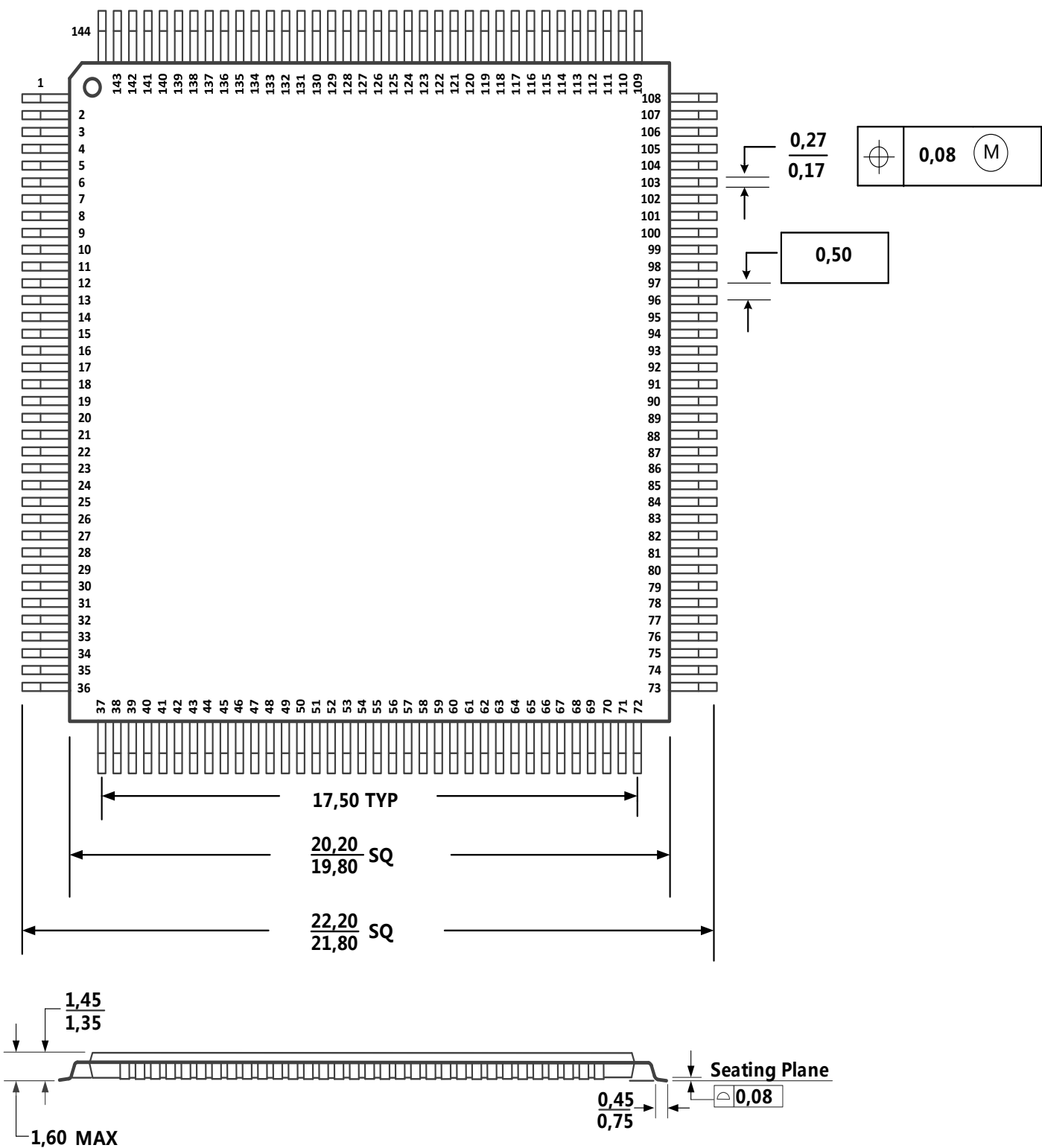


图 5.24 IEEE-1149.1 测试访问端口时序

5.23 封装尺寸



注释：单位 mm

AVP32VC33 Digital Signal Processor

联系方式

公司网址：www.advancechip.com

联系邮箱：sales@advancechip.com

销售联系电话：0731-88731027

公司总部地址：长沙市高新开发区尖山路 39 号中电软件园总部大楼 16 楼 1601-1603