

HPI 接口在 TI SOC 的应用详解

德州仪器半导体技术（上海）有限公司 DSP 技术支持工程师 唐超伦

【摘要】本文从总体上介绍 HPI（Host Peripheral Interface）接口的工作模式，与上位主机的连接方式，主机访问的操作流程；归纳了应用当中常见问题，并提供了分析解决办法。

简介

HPI 接口是 TI 为处理器之间直接互连通讯定义的一种异步接口，大多数 TI DSP 芯片上都有 HPI 接口。HPI 接口是从（Slave）端口，接在主机的扩展内存总线上，DSP 不能通过 HPI 向主机（Host）的访问，只能被主机读写。两个 DSP 的 HPI 接口之间不能通讯。两个 DSP 之间互连，可以将一个 DSP（从）的 HPI 接到另一个 DSP（主）的扩展内存接口（EMIF）上[1]。

1. HPI 工作模式

不同系列 DSP 上的 HPI 接口版本有所不同，区别体现在 DSP 对 HPI 的控制上，如 C6727 上的 UHPI 可通过寄存器使能与关闭 HPI 接口，对主机访问 DSP 内存空间的控制，以及对 HPI 接口信号的功能复用上。但从主机访问的角度，HPI 的工作模式分为：复用模式 (Multiplexed-Mode) 和非复用模式 (Non-multiplexed-Mode)。

复用模式下没有地址线，主机访问 DSP 的地址信息是以数据方式送到 HPIA (HPI 地址寄存)。从硬件信号的角度，地址，数据信号是由同一组数据线传递，所以称为复用模式。

非复用模式的数据线与地址线是分开的，与内存接口连接相似。非复用模式不需要操作 HPIA 寄存器，主机访问的地址信息通过地址总线直接送给 HPI。

所有的 HPI 接口都支持复用模式，但不是所有芯片的 HPI 接口都支持非复用模式 (参考相应的芯片手册确定是否支持)。除了有无 HPIA 的操作区别外，两种模式的操作没有区别。因为非复用模式的操作是复用模式操作的子集，为方便起见，本文以复用模式展开讨论。

2. HPI 硬件信号连接

HPI 接口复用模式连线如图 1 所示，根据在应用当中的必要性分为：必要的，和可选的两组信号。可选的信号以虚线表示。

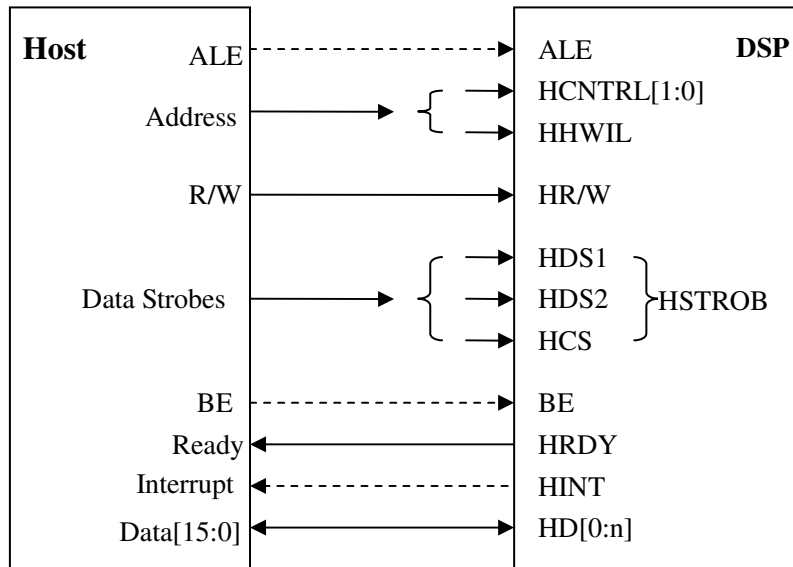


图 1. HPI 接口复用模式硬件连接

- 数据线 HD[0:n]: 在复用模式下，数据线的宽度一般为 CPU 位宽的一半，一个 HPI 访问分为高低半字的两次访问，如 C5000 是 16-bit CPU，HPI 数据线为 8 位，C6000 是 32-bit CPU，其 HPI 数据线为 16 位。C64x 系列的 HPI 支持 32 位，在 32 位模式下一个 HPI 访问不需要分为高低半字两次访问组成一个完整的访问。
- HCNTL0/1, HWIL: HCNTL0/1 选择要访问的 HPI 寄存器，HWIL 控制访问寄存器的高低半字，必须先高后低。一个寄存器的高低半字的两次访问一定要连续完成，中间不能插入其它的 HPI 操作。只有 HPIC 可以只访问半个字。
- HR/W: 指示对 HPI 寄存器进行读，还是写操作。如果主机的读，写信号是分开的，可以利用其中一个信号，但要注意做上拉或下拉处理以控制其在三态时的电平。
- HCS, HDS1/2: 这三个信号根据图 2 的逻辑产生内部 HSTROBE 信号，其逻辑关系是要求 HDS1 和 HDS2 信号相反，HCS 低有效。HSTROB 下降沿的时间点反应的是三个信号中最后跳变的信号。HPI 在 HSTROB 的下降沿采样控制信号 HR/W, HCNTL0/1, HWIL 以判断主机要对哪个寄存器进行读，或者写操作命令。注意控制信号在 HSTROBE 的下降沿之前需要最少 5ns 的 setup 稳定时间，而 HDS1/2 和 HCS 到 HSTROBE 信号内部门电路的延时是皮秒级的，所以控制信号的 setup 时间需要外部时序保证。

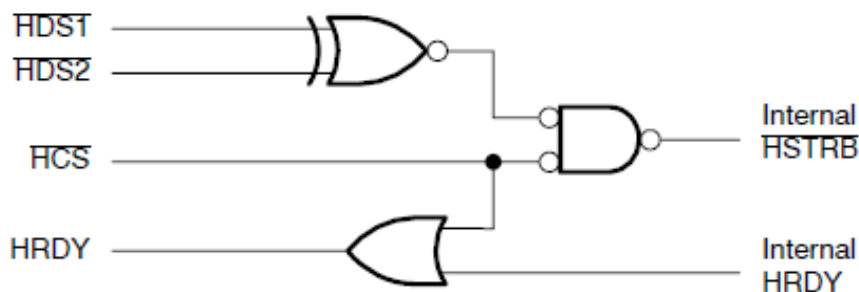


图 2. HSTROBE 信号产生逻辑

- HRDY: HPI 的输出信号，指示当前操作状态，用做硬件流量控制握手信号。
- HINT: 通过 HPI，主机与 DSP 之间可以互发中断。HINT 是 HPI 送给主机的中断信号，DSP 对 HPIC[HINT]位写 1，HINT 信号线上送出高电平信号，主机可利用此信号做为中

断信号输入。DSP 不能清除 HPIC[HINT]状态，主机在响应中断后，需要对 HPIC[HINT]位写 1 清除状态，DSP 才能再次对 HPIC[HINT]置位发中断。主机通过写 HPIC[DSPINT]置 1 给 DSP 产生中断，DSP 在响应中断后，需要对 HPIC[DSPINT]写 1 清除状态，主机才能继续操作 HPIC[DSPINT]给 DSP 发中断。通过 HPI 传输数据，结合互发中断做为软件层的握手信号，可有效提高通讯的效率与灵活性。

- ALE: 存在于地址，数据线复用的主机上用来指示地址信号周期，这种总线复用的主机很少见，所以通常将 ALE 固定上拉处理，只用 HSTROBE 采样控制信号。
- BE: Byte Enable 信号，这个信号只出现在 32 位的 C6000 DSP 上。因为应用当中通常都是对整个 32-bit 字进行访问，所以直接做上拉使能处理。

3. HPI 寄存器地址映射

HPI 口提供给主机端访问的寄存器只有 4 个，通过 HCNTRL [1: 0] 选择。

表 1. HPI 寄存器访问选择

HCNTL1	HCNTL0	寄存器选择
0	0	HPIC
0	1	HPID（地址自增模式）
1	0	HPIA
1	1	HPID（地址非自增模式）

由于在复用模式下数据线通常只有寄存器宽度的一半，所以一个完整的数据访问由高低半字两次访问组成，由 HWIL 信号控制，HWIL 信号必须是先低后高。通常将 HWIL 和 HCNTRL[1:0]接在主机的地址线上，将 4 个寄存器映射为主机端的 8 个内存单元，下表中的地址线连接方法将 8 半字寄存器映射到主机的 8 个连续的内存单元。这里的地址线是用来选择 HPI 的寄存器，与非复用模式下的地址线直接访问 DSP 的内存空间是完全不同的作用。

表 2. HPI 寄存器地址映射

Addr Offset	HCNTL[1:0] A[2:1]	HWIL A[0]	Register	
0	00	0	HPIC first half word	
1	00	1	HPIC second half word	
2	01	0	HPID first half word	地址自增模式
3	01	1	HPID second half word	
4	10	0	HPIA first half word	
5	10	1	HPIA second half word	
6	11	0	HPID first half word	地址非自增模式
7	11	1	HPID second half word	

4. HPI 寄存器功能说明

HPIC (HPI Control Register)

HPI 控制寄存器 HPIC 的位图如表 3 所示，C6000 系列 DSP 的 HPI 寄存器是 32 位的，也只有低 16 位有效，与 C5000 系列 DSP 的 HPIC 寄存器定义的功能保持一致。不同版本的 HPI 接口的 HPIC 位定义的主要不同之处在于软件握手 HRDY 位的有无，其它位名称可能存在细小差异，但功能定义都是一样的。

表 3. HPI Control Register

15	12	11	10	9	8
Reserved		HPIASEL	Reserved	DUALHPIA	HBOBSTAT
R-0		R/W-0	R/W-0	R/W-0	R-0
7	6	5	4	3	2
HPIRST	Reserved	FETCH	HRDY	HINT	DSPINT
R-1h	R-2h	R/W-0	R-1	R/W-1(Host)	R/W-0
R/W-0(CPU)					
				1	0
				DSPINT	HWOB
				R/W-0	R/W-0

主机在对 HPI 进行访问前可以通过 HPIC 配置字节序 (HWOB) 和地址寄存器模式 (DUALHPIA)。默认的配置为 HWOB=0, 即先访问高半; DUALHPIA=0, 即单地址寄存器模式, 读和写操作使用同一个 HPIA; 通常都采用默认的 HPIC 寄存器配置。

HWOB 与硬件信号 HWIL 是没有联系的, HWIL 信号必须保证先低后高分别访问两个半字单元。至于先访问一个字当中的高或低半字, 是由 HWOB 控制的。

HPIA (HPI Address Register)

物理上存在 HPIAR, HPIAW 两个地址寄存器。HPIAR 是读操作地址寄存器; HPIAW 是写操作地址寄存器。由 HPIC 的 DUALHPIA 位来决定是采用双地址寄存器模式还是单地址寄存器模式。如果采用双地址寄存器模式, 在对 HPIA 操作之前, 通过设置 HPIC 的 HPISEL 位选择下一个要访问的 HPIA。通常为了简化在读写操作转换时对 HPIC 的操作, 选用单地址寄存器模式。

HPIA 的地址信息在不同系列 DSP 上有所不同:

在 C5000 上, 数据空间, I/O 空间只能按 16 位字寻址, 程序空间按字节寻址。HPI 的 DMA 访问属于 I/O 空间, HPIA 的值代表 16 位字地址。如主机端要访问 DSP 内存字节地址 0x100, 则要往 HPIA 写的地址值是 0x80。在用 HPI 启动时, 要注意 DSP 代码是按字节地址链接的, 即代码段的地址是字节地址, 主机端要将代码段的地址信息除以 2 再送到 HPIA。

在 C64 上, HPIA 的值代表字节地址, 但是 HPI 访问的数据是 32 位的, 所以 HPI 会忽略 HPIA 地址值的低两位。

在 C64+ 的 DSP 上, HPIA 代表 32 位字地址, HPI 会将字地址左移两位转换成字节地址, 主机若要访问 DSP 字节地址 0x100, 则要赋值 HPIA 为 0x40。

HPID (HPI Data Register)

主机通过 HPI 对 DSP 的内存访问是间接访问, 主机只能访问 HPID, HPID 与 DSP 内存之间是通过 HPI 专属的 DMA 进行数据搬运的。如图 3 所示。

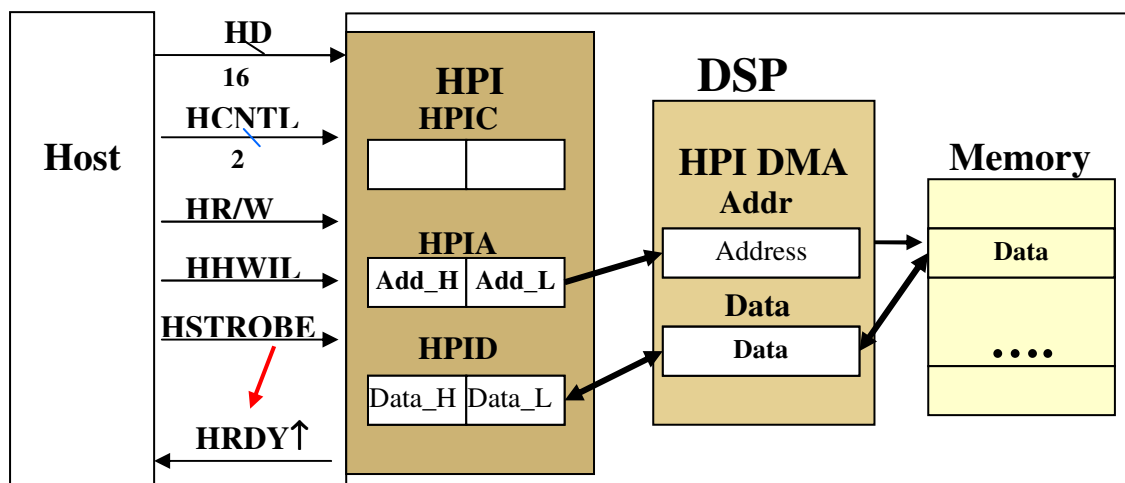


图 3. HPI 读写数据流

HPID 的访问分为自增模式和非自增模式。在自增模式下，访问 HPID 后，HPIA 会自动增加指向下一个字地址，在连续访问时，自增模式因为减少了主机对 HPIA 的操作，可以增加 HPI 数据访问的吞吐率。非自增模式下访问 HPID 后，HPIA 的值保持不变，主机需要更新 HPIA 来访问下一个地址。

在写操作时，主机把数据写到 HPID，HPI 将第二个半字的数据通过 HSTROBE 的上升沿锁存到 HPID 后，将 HRDY 置为忙状态，并启动 HPI DMA 将 HPID 的内容搬到 HPIA 所指向的内存单元，然后清除 HRDY 指示可以进行下一次操作。

在读操作时，在第一个 HSTROBE 的下降沿，HPI 采样到 HR/W 为读命令，则将 HRDY 置为忙状态，启动 HPI DMA 将 HPIA 指向的内存单元的数据搬到 HPID，清除 HRDY 忙状态，主机端方可结束总线访问周期，锁存数据线上的有效数据。

5. HPI 操作流程

主机对 HPI 的一次总线访问周期为分三个阶段：主机发起访问，HPI 响应，主机结束访问周期。

- A. 主机发起访问：即对 HPI 寄存器的读，或者写命令。主机送出的硬件信号为 HSTROBE（由 HCS, HDS1/2 产生），HR/W，HCNTL0/1，HWIL，以及 HD[0:n]。HPI 在 HSTROBE 的下降沿采样控制信号 HR/W，HCNTL0/1，HWIL 判断主机的操作命令。
- B. HPI 响应：HPI 在 HSTROBE 的下降沿采样控制信号，根据控制信号做出相应的响应。如果是写（HR/W 为低）命令，则在 HSTROBE 的上升沿将数据线上的信号锁存到 HCNTL0/1 和 HWIL 指向的寄存器。如果是读命令（HR/W 为高），如果是读 HPIC，或者 HPIA 寄存器，HPI 将寄存器的值直接送到数据总线上；如果读 HPID，HPI 先将 HRDY 置为忙状态，HPI DMA 将数据从 HPIA 指向的内存单元读到 HPID，再送到数据总线上，并清除 HRDY 忙状态，在读 HPID 后半字时，数据从寄存器直接送到数据总线上，不会出现 HRDY 信号忙状态。
- C. 主机结束访问周期：对于写操作，主机将数据送出后，只要满足芯片手册中 HPI 对 HCS 的最小宽度要求，即可结束访问周期。对于读 HPID 操作，要等 HRDY 信号由忙变为不忙，主机才能结束访问周期。
 - a) 两次连续的 HPI 操作的间隔，在芯片手册的 HPI 时序参数表里有要求，最小间隔为两个 HPI 功能模块时钟周期。

6. HPI 常见故障案例分析

在 HPI 应用调试过程中，常遇到的问题分为：写数据不成功，读数据不正确，HRDY 常高。这些问题通常都是由于时序不正确造成的，下面结合实际应用当中的案例进行分析。

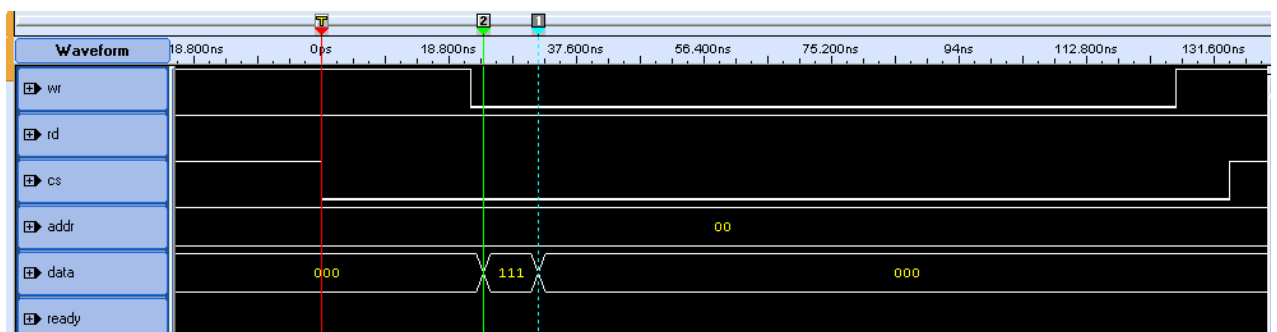
6.1 写数据不成功

案例的硬件连接如下：

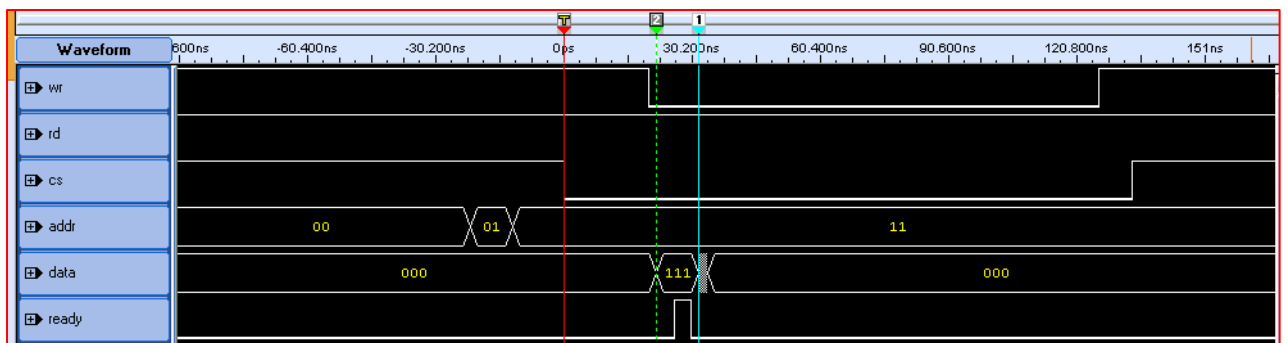
DSP HHWL R3	U19B
DSPARM IRQ4	HHWL/PTRDY
ARM WR P1	HINT/PFRAME
DSP HRDY P4	HR/W/PCBE2
DSP HAS T3	HRDY/PIRDY
DSP CS R2	HAS/PPAR
ARM WR T1	HCS/PPERR
ARM RD T2	HDS1/PSERR
ARM A2 T4	HDS2/PCBE1
ARM A3 R1	HCNTL0/PSTOP
	HCNTL1/PDEVSEL

首先关注核心信号 HSTROBE 由 DSP_CS, ARM_WR (HDS1), ARM_RD (HDS2) 产生, 从下面时序图可以看出 ARM_WR 的下降沿是最后产生的, 所以写操作时 HSTROBE 的下降沿反应的是 ARM_WR 的下降沿。

写 HPIC 的时序截图如下:



写 HPID 的时序截图如下:



两个时序图显示主机送出的数据 111b 在 HSTROBE (ARM_WR) 的下降沿后, 很快被改变成其它值 000b。在写 HPID 的时序截图中, 第一个 HSTROBE 的下降后, HPI 送出 HRDY 信号, 然后数据线被改变, 首先判断 HPI 对主机的命令做出了响应, 通过 HRDY 信号的出现时机, 说明 HPI 判断这是一个读操作, 可以判断为 HSTROBE 的下降沿采样 HR/W 信号不正确。

从硬件连接来看, HSTROBE (HR/W) 要采样 HR/W, HCNTL0/1 来判断主机命令, HR/W 的与 HSTROBE 为同一信号源, 且同为下降沿, HR/W 与 HSTROBE 的下降沿之间的 setup 时间不够, 采样 HR/W 的电平状态出现误判, 认为是高电平读命令, HPI 对读命令的响应则是在第一个 HSTROBE 的下降沿之后送出 HRDY 信号, 并在 HRDY 之后, HPI 送出数据到总线上。

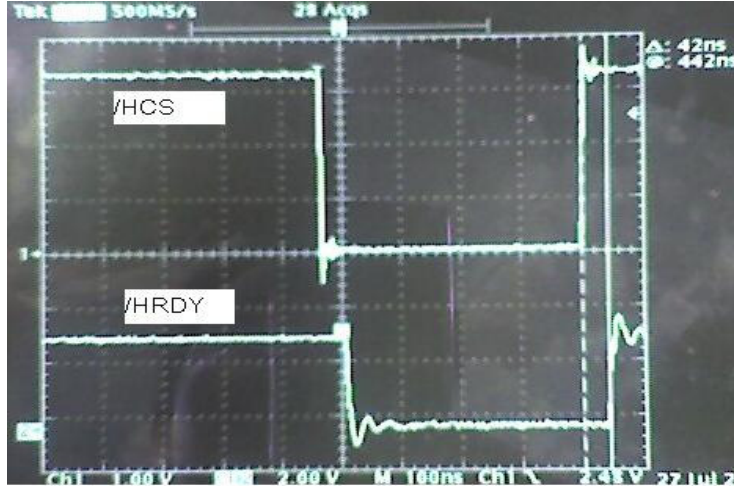
对于该问题, 需要对参与 HSTROBE 逻辑译码的 HR/W 信号的下降沿做延时处理, 可在逻辑电路如 CPLD 或 FPGA 里实现, 以确保 HSTROBE 的下降沿采样到稳定的 HR/W 电平。

6.2 读数据不正确

通常表现为读读 HPIC, HPIA 正常, 但读 HPID 不正常, 前半字为 0, 后半字正确, 对同

一个地址读两次，第二次的数据完全正确。

在案例中，用示波器观察 HCS 与 HRDY 之间的时序关系，发现 HCS 的上升沿在 HRDY 的上升沿之前，即主机在 HPI 数据有效之前结束了访问周期。HRDY 的上升沿其实是因为 HCS 的结束而拉高的，并非数据真正有效。



用户由于没有在硬件上将 HRDY 与主机 PowerPC 的 TA 信号互连，没有硬件握手机制，于是从软件配置上加大主机的总线访问周期，即增加 HCS 的宽度，故障现象没有变化。

原因分析：读 HPID 与 HPIC，HPIA 时序不同，读 HPID 操作需要 HPI DMA 从 HPIA 所指向的地址读数据到 HPID，会有时间上的延时。而读 HPIC 和 HPIA 直接从寄存器读数据，没有延时，所以读 HPIC，HPIA 是正确的。在读 HPID 时，HPI 会在第一个 HSTROBE 的下降沿后将 HRDY 置位，指示数据未准备好的忙状态，主机应当在总线上插入等待周期，数据准备好后 HPI 清除 HRDY，主机才可以结束总线周期，通过 HCS 的上升沿将有效数据锁存。

HSTROBE 的下降沿到数据有效之间的延时与芯片及 HPI 接口的工作频率相关，以 C5502，C5501 为例，在芯片手册中，这个延时参数 H1 在 SYSCLK1 与 CPU 时钟的分频为 4 时，最大延时为 $12 * 2H + 20(ns)$ ， $H = SYSCLK1 / 2$ ，在 HPI 启动期间，PLL 没有倍频，处于旁通状态，系统输入时钟就是 CPU 的工作时钟，SYSCLK1 默认分频为 CPU 时钟的 4 分频，以输入时钟为 25MHz 为例，最大延时为：

$$H1_{Max} = 12 * 2H + 20 = 12 * 2 * \frac{4 * 10^9}{25 * 10^6} * \frac{1}{2} + 20 = 1940(ns)$$

这个时间长度通常超出了主机端总线周期的软件配置范围，所以通过软件配置增加 HCS 的宽度不一定能满足 HRDY 的最大延时要求。在有的 DSP 芯片手册上只提供了 HRDY 的最小延时，最大延时与芯片的优先级设置，及系统配置相关而不确定，比如与系统中其它主模块如 EDMA 同时访问 DDR，那么延时与 HPI 的优先级，EDMA 的优先级，EDMA 的 burst 长度，以及 DDR 的命令排序等配置相关，这样通过延长主机的总线访问周期，更加不可靠。

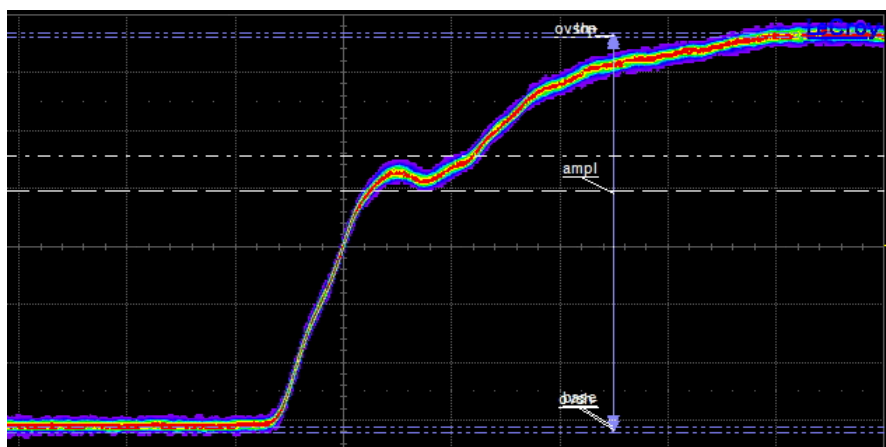
解决办法：在硬件设计之初，一定要利用 HRDY 硬件握手信号[2][3]。虽然有的芯片 HPIC 寄存器提供了 HRDY 软件握手方式，只能做为弥补硬件设计之初遗漏 HRDY 硬件握手信号的权宜之计，软件轮循 HRDY 的办法会带来额外的开销，降低 HPI 总线的吞吐率，增加主机软件实现的复杂度。而且有的芯片 HPI 不支持 HRDY 软件查询方法，只能通过硬件 HRDY 保证数据的有效性。

6.3 HRDY 常高

有的系统在长时间运行中偶尔出现 HRDY 常高，导致主机端总线访问异常，需要重新上电才能恢复 HPI 的正常操作。这种故障是由于 HPI 状态机出现异常。

从实际故障定位中总结出以下几点原因：

- A. HPI 的高低半字访问的顺序访问被其它 HPI 访问打断：在复用模式下，一个完整的 HPI 访问是由高低半字两次访问组成，需要严格保证，否则会破坏 HPI 的状态机，从而导致不可预期的后果。
- B. 主机通过 HPI 访问了 DSP 内部的保留空间，或者破坏了 DSP 的程序，数据空间，导致 DSP 运行异常，进而导致 HPI 状态机异常。
- C. 主机的 HSTROBE 信号有毛刺，或者信号完整性不好，如下图中 HCS（些案例 HSTROBE 是由 HCS 控制）的上升沿的回勾，都会导致 HPI 误判断为主机的新的访问的开始，从而打乱了高低半字的访问顺序要求，导致 HPI 状态机的错乱。



7. 总结

HPI 是一种简单的异步接口，只要设计中满足了时序要求，即可稳定工作。在开发当中遇到数据读写不正确，从 HSTROBE 信号入手检查与之相关的信号的时序关系，便可以找出问题原因。另外，信号完整性是任何系统稳定工作的前提。

关于特定芯片上 HPI 接口的特有功能本文没有针对讨论，如 C6727 的字地址模式和字节地址模式可通过 HPIC 配置；C6727 在 HPI 启动后 ROM bootloader 将 HPI 关闭，需要软件重新使能才能使用等；以及不同芯片的 HPI 启动模式下的跳转方式不同，请参考相应芯片的 HPI 手册及 bootloader 应用手册。

参考文献：

- [1] spr536:TMS320C6000 EMIF to TMS320C6000 Host Port Interface
- [2] spr545: TMS320C6000 Host Port to MC68360 Interface
- [3] spr546: TMS320C6000 Host Port to MPC860 Interface