

# 基于 HPI 接口的高速数据传输设计

## Design of High-Speed Data Transmission Based on HPI

■ 烟台大学光电信息学院 张玉猛 邵左文 贾伟

### 引言

DSP 芯片能够大大提高数字信号处理的效率,但在主机与 DSP 构成的系统中,当 DSP 与主机间需要大数据量传输时,数据传输速率就会成为程序运行速度的瓶颈。所以在程序调试过程中,实现主机与 DSP 之间的快速数据传输,不仅可以提高程序运行效率,还可以大大节省调试程序的时间。

TMS320C6000 系列的 HPI (Host Port Interface) 接口不仅可以方便主机对 DSP 的控制,还可以实现主机与 DSP 内存的快速数据传输。这里用双 TMS320C6416(600MHz) 来进行实验,通过 HPI 接口实现了主 DSP (下文中都称为“主机”)和从 DSP 的快速数据传输,并通过实验测试了 HPI 接口的数据传输速率。

### 系统介绍

#### HPI 概述

HPI (Host-Port Interface) 主机接口,是 TI 高性能 DSP 上配置的与主机进行通信的片内外设。通过 HPI 接口,主机可以非常方便地访问 DSP 的所有地址空间,从而实现对 DSP 的控制。

TMS320C6416 的 HPI 接口是一个 16bit/32bit 宽的并行端口。主机 (host) 对 CPU 地址空间的访问是通过 EDMA 控制器实现的。HPI 接口的访问主要通过三个专用寄存器来实现,它们分别是 HPI 控制寄存器 (HPIC)、HPI 地址寄存器 (HPIA) 和 HPI 数据寄存器 (HPID)。

#### HPI 接口信号简介

(1) HD[31:0] (数据总线)

(2) HCNTL[1:0] (控制 HPI 访问类型)

如前所述,对 HPI 的访问需要通过三个寄存器,即 HPI 地址寄存器 (HPIA)、HPI 数据寄存器 (HPID) 和 HPI 控制寄存器 (HPIC) 来实现。HCNTL[1:0] 就是用于选择这三个寄存器的专用引脚。

(3) HHWIL (半字指示选择)

HHWIL 指示当前的为第一个或是第二个半字传输,但需要注意的是,它并不代表是最高有效的 (most significant) 还是最低有效的 (least significant),而决定的依据是 HPIC 中的 HWOB 位的状态。在 HPI32 模式下,不使用此信号。

(4) HR/W (读/写操作指示)

(5) HRDY (输出准备好)

(6) HCS, HDS1, HDS2 (选通信号)

当 HCS 有效,并且 HDS1 和 HDS2 中仅有一个有效时,内部触发信号 HSTROBE 有效。这三个信号的组合逻辑其实就是片选和读/写信号构成的组合逻辑,因此,可直接与主机的片选和读/写信号相连。

(7) HAS (地址输入选通)

(8) HINT (向主机输出的中断)

#### HPI 接口寄存器简介

如上所述,主机通过 HPI 接口对 DSP 的访问实际上是通过三个寄存器来实现的,下面就针对这三个专用寄存器进行介绍。

(1) HPI 控制寄存器 (HPIC)

HPIC 中每一位都有特定的功能,在对 HPI 进行访问的过程中需要特别注意。简要介绍一下这些功能位的作用。

① HWOB (半字顺序位)

如果 HWOB=1, 第一个半字为最低有效;如果 HWOB=0, 第一个半字为最高有效。HWOB 对地址和数据都起作用,如果采用 HPI16 模式,在访问数据或者地址寄存器之前,应该首先初始化 HWOB 位。

② DSPINT (主机产生的 Processor-to-CPU 中断,用于 HPI 启动方式中将 DSP 内核从复位状态中唤醒)

③ HINT (DSP-to-Host 中断,即通

表 1 外设选择

PCI_EN Pin[AA4]	McBSP2_EN Pin[AF3]	HPI	GP[15:9]	PCI	EEPROM	McBSP2
0	0	✓	✓			✓
0	1	✓	✓			✓
1	0			✓	✓	
1	1			✓		✓

表 2 主机 EMIFA CE1 参数配置

Configuration	Read Setup	Read Strobe	Read Hold	Write Setup	Write Strobe	Write Hold

向此位写入特定值来产生对主机的中断)

#### (2) HPI 地址寄存器(HPIA)

存放 32bit 数据, 指向将要访问的 DSP 地址空间中的地址。

#### (3) HPI 数据寄存器(HPID)

在写操作中存放将要写入 HPIA 所指向地址的数据, 在读操作中为 HPIA 所指向地址中的数据。

## 系统设计

### 硬件设计

#### 外设选择

在 C6416 中, 一些外设共用某些引脚。其中 HPI, GP [15: 9], PCI, EEPROM 以及 McBSP2 共用一组引脚, DSP 在复位时通过锁存 PCI\_EN 及 McBSP2\_EN 引脚的值来选择使用何种外设。如表 1, 在本设计中, 将这两个选择引脚都拉低。

#### 数据总线

C6416 HPI 数据总线具有 32 个外部引脚 HD[31:0]。因此, C6416 HPI 支持 16 位或 32 位的数据总线。当用 16 位宽的主机接口时, C6416 HPI 称为 HPI16; 当用 32 位宽的主机接口时, C6416 HPI 称为 HPI32。C6416 HPI 通过复位时的自举和器件配置

引脚 (HD5) 选择采用 HPI16 还是 HPI32。

HPI16 具有 16 位数据总线, HPI16 将两个连续的 16 位传输组成一个 32 位数据传送到 CPU。为了和其他 C6000 器件兼容, 无论复位时选择何种 Endian 模式, HPI16 都使用 HD [15:0] 作为数据引脚。HPI32 具有 32 位的数据总线, 使用该增加的总线宽度, 所有传输均为一个 32 位的字传输, 而不是两个连续的 16 位半字。在本设计中采用 HPI32 总线模式。

#### 主机 EMIF 配置

对主机, EMIFA (64 位总线) 和

EMIFB (16 位总线) 都可以与 HPI 相连。在 HPI16 模式下, 可以采用 EMIFB, 在 HPI32 模式下, 可以采用 EMIFA。在本设计中, 影射 EMIFA CE1 存储器空间为主机接口, 对 CE1 空间控制寄存器的设置见表 2。

#### 硬件连接

从 C6416 HPI 寄存器的编址方式可以看出, 主机需两根地址线寻址到 HPI 接口的控制寄存器、地址寄存器和数据寄存器, 因此选择主机的地址线 A3、A2 连接 C6416 HPI 的 HCNLT1、HCNTL0。HPI 的选通由 HCS、HDS1、HDS2 三根信号线共同作用, 最后的 HPI 使能信号(STROBE) 为 HDS1 异或 HDS2 后, 再与 HCS 进行与非运算的结果。若将 HPI 接口安排在主机的 ACE1 区域(即地址范围 0X90000000-0X9FFFFFFF), 则直接将主机的片选信号 ACE1 接到 HCS, 而将读写信号 RE、WE 分别接到 HDS1、HDS2。对于 HR/W 信号, 可以连接到地址线 A4, 当 A4=1 时, 代表读操作, 反之为写操作。

若使用 HPI16, 可以使用主机的地址线 A1 接 HHWIL 来完成高低字节的识别: 当 A1=0 时, 表示为第一字节; A1=1 时表示第二字节。

由于主机的 ARDY 信号和 C6416 HPI 接口的 HRDY 信号逻辑刚好相

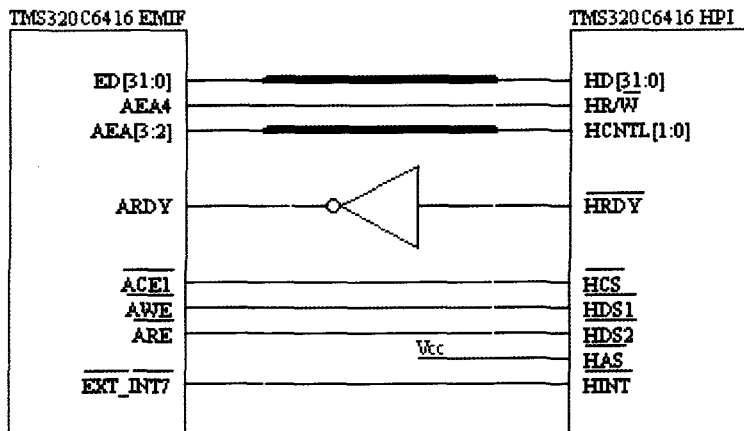


图 1 硬件连接图

表 3 写、读自动增模式的数据传输速率

传输模式	SRC /DST	H PI	HPI Width	Burst Length	EMIF	Latency	Xfer	Throughput (MB/s)
写自动增 模式	EMIFA/ SDRAM	25	32	8	133	36	180	88.9
					100	36	180	88.9
				512	133	36	12272	99.8
					100	36	12272	99.8
			16	8	133	60	372	44.4
					100	60	372	44.4
				512	133	60	24560	49.9
					100	60	24560	49.9
读自动增 模式	EMIFA/ SDRAM	25	32	8	133	167	180	55.3
					100	193	180	51.5
				512	133	165	12207	99.3
					100	193	12223	99.0
			16	8	133	190	372	34.2
					100	215	372	32.7
				512	133	190	24495	49.8
					100	215	24496	49.7

反, 因此要将 HRDY 信号经过反相后再接 ARDY 信号。C6416 HPI 的 HINT 信号可以直接连接到主机的 EXT\_INT7 引脚上实现 HPI 对主机的中断信号连接。HPI 接口信号线中的 HAS 线直接拉高。C6416 HPI 与主机的连接电路如图 1 所示。

### 软件设计

主机必须按照顺序进行 HPI 访问:

- (1) 初始化 HPI 控制 (HPIC) 寄存器;
- (2) 初始化 HPI 控制 (HPIA) 寄存器;
- (3) 写数据到 HPI 数据 (HPID) 寄存器或从 HPID 寄存器读取数据。

一旦 HPI 被初始化, 主机就可以固定地址模式或自动增加模式对 DSP 地址进行读写访问, 下面以 HPI 接口地址增加模式写操作为例来介绍。

```
(int *)ptr_hpi=(int *) (0x90000000);
```

```
void c6416_write_section (int *source,int dest_addr,int length)
```

```
{ int i;
```

```
//初始化 HPI 口的 HPIC 寄存器 HCNTL1 HCNTL0
```

```
*ptr_hpi++=0x00000002;// 0 0
```

```
//初始化 HPI 口的 HPIA 寄存器
```

```
*ptr_hpi++= dest_addr; // 0 1
```

```
//写 HPI 口的 HPID 寄存器
```

```
for(i=0;i<length;i++)
```

```
*ptr_hpi=*source++; // 1 0
```

```
}
```

### 实验结果

HPI 的数据传输速率可以通过在单位时间内传输的数据量来表示, 其计算公式如下

$$\text{数据传输速率} = \frac{(\#words)(4)}{(cpuclock)(latency+xfer)} [\text{bytes/s}]$$

注: #words: 数据(字)传输量

cpuclock: CPU 时钟周期

latency: 从主机开始传输到目标准备转移第一个数据所需周期数

xfer: 传输 n 个字所需周期数

实验中测得的 HPI 数据传输速率见表 3。

### 结束语

本文介绍了主机接口 HPI 的特点, 通过实验得出 HPI 的数据传输速率。该方案灵活简单, 适用于含有 HPI 接口的 DSP 应用系统, 从而为开发人员提供了一种全新的数据共享传输方案。其特点是通用、高速且不需辅助硬件, 具有很好的实际应用前景。GEC