

介绍 **dsp** 知识，为大家提供最新的 **dsp** 资讯，更多内容可以去南京研旭电气科技有限公司的官网 www.njyxdq.com www.f28335.com 或者官方论坛，嵌嵌 **dsp** 论坛

www.armdsp.net 进行交流学习

dsp 论坛 www.armdsp.net

dsp 开发板 www.njyxdq.com

嵌入式开发板 www.f28335.com

DSP HPI 总线与 MPC8272总线接口的 FPGA 实现

摘要：通过对 TI 公司 TMS320C6421 DSP HPI 接口信号和接口总线时序的分析，以 VHDL 语言为工具，使用 Altera 的 FPGA 芯片 EP3C40F780C8，设计完成 MPC8272 总线和 TMS320C6421 DSP HPI 总线之间的通信接口，并在实际的产品中得到运用，给出与整个接口设计相关的 VHDL 源代码。对于类似的 DSP HPI 接口设计，此文章具有参考和指导意义。

一、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。

TMS320C6421的 HPI 接口是一个16bit 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)。

二、HPI 接口信号简介

(1) HD[15 : 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。

HCNTL1

HCNTL0

HPI 访问类型

0

0

主机可读写 HPI 控制寄存器 HPIC

0

1

主机可读写 HPI 数据寄存器 HPID，读操作或写操作后
HPIA 自动增1

1

0

主机可读写 HPI 地址寄存器 HPIA

1

1

主机可读写 HPI 数据寄存器 HPID，读操作或写操作后 HPIA 不变

(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。对于第一个半字，HHWIL 必须被驱动为低电平；对于第二个半字，HHWIL 必须被驱动为高电平。

(4) HR/W (读/写操作指示)

HR/W 为高电平，表示从 HPI 接口读；HR/W 为低电平，表示向 HPI 接口写。

(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。如下图所示：

(7) HAS (地址输入选通)

在 TMS320C6421 HPI 接口中目前没有用，连接到逻辑高电平。

(8) HINT (向主机输出的中断)

三、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。

①HWOB(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。

②DSPINT(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)

③HINT(DSP-to-Host 中断，即通过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 HPID 所指向地址的数据，在读操作中为 HPID 所指向地址中的数据。

四、HPI 接口读写时序

(1)HPI 接口读时序

(2)HPI 接口写时序

五、HPI 接口硬件设计

从 C6421 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 A29、A28连接 C6421 HPI 的 HCNTL1、HCNTL0。选择主机的地址线 A30连接到 C6421 HPI 的 HHWIL，作为半字指示选择。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。HCS、HDS1、HDS2信号由 FPGA 产生。TA 为 MPC8272传输结束标识，HPI 口 HRDY 有效后 FPGA 向 CPU 发送 TA，保证 HPI 数据正确读出。C6416 HPI 的 HINT 信号可以直接连接到主机的 IRQ 引脚上实现 HPI 对主机的中断信号连接，也可通过 FPGA 连接到主机，使控制更灵活。

六、地址空间分配

由于 C6421 为 16 位的 HPI 口，其内部总线为 32 位，所以每次读写要分两次，一次为高 16 位，一次为低 16 位，由 HHWIL 来选择（0—First Half Word，1—Second Half Word），设定 HPIC.HWOB = 0（First Half Word = 高 16 位，Second Half Word = 低 16 位）。C6421 的 HPI 接口映射为 MPC8272 的 4 对地址空间，由 MPC8272 CS3 控制，配置 CS3 为 16 位宽的 GPCM 访问模式，如下地址分配：

HPIC: 0x0D000000 (高 16 位), 0x0D000002 (低 16 位)。

HPIA: 0x0D000004 (高 16 位), 0x0D000006 (低 16 位)。

HPID_FIX: 0x0D00000C (高 16 位), 0x0D00000E (低 16 位), C6421 地址由当前的 HPIA 决定。

HPID_Auto: 0x0D000008 (高 16 位), 0x0D00000A (低 16 位) C6421 地址自动加 1。

七、HPI 接口相关 VHDL 代码

```
output wire c6421_hasn, c6421_hcsn, c6421_hdsn1,  
c6421_hdsn2,
```

```
assign c6421_hasn = 1' B1;
```

```
assign c6421_hcsn = cs3;
```

```
assign c6421_hdsn1 = wen;
```

```
assign c6421_hdsn2 = oen;
```

```
reg [7:0] st_ta;
```

```
reg tax;
```

```
wire tax001 = (zzz_cnt==256) ? 0 : 1 ;
```

```
wire tax002 = (zzz_cnt==256+8) ? 0 : 1 ;
```

```
wire tax003 = (zzz_cnt==256+16)? 0 : 1 ;
```

```
wire tax004 = (zzz_cnt==256+32)? 0 : 1 ;
```

```
reg c6421hcsn_start;

always @(posedge clk_cpu)

if (zzz_cnt==5) c6421hcsn_start = 1;

else          c6421hcsn_start = 0;

always @(posedge clk_cpu) begin

if (c6421hcsn_start) begin

st_ta = STA_CHECKRD;

tax  = 1;

end

else if (STA_CHECKRD == st_ta) begin

if (c6421_hrdyn)
```

```
st_ta = STA_CHECKRD;

else

st_ta = STA_GEN;

tax = 1;

;end

else if (STA_GEN == st_ta) begin

st_ta = STA_END;

tax = 0;

end

else begin

tax = 1;
```

```
end
```

```
end
```

```
assign ta = (~cs3) ? (tax & tax001 & tax002 &  
tax003 & tax004) : 1' BZ;
```

MPC8272功能很强大，C6421 HPI 接口相对于 MPC8272 来说为慢速外部设备，TA 为 MPC8272传输结束标识，HPI 口 HRDY 有效后 FPGA 向 CPU 发送 TA，保证 HPI 数据正确读出。因此在 VHDL 程序中设计了状态机，当 HRDY 信号有效后就立即结束本次访问，否则经过一段时间的延迟后强制结束本次访问，这样可以避免接口长时间占用总线，影响系统性能。

八、结束语

本文使用 VHDL 语言和 FPGA，设计了 MPC8272与 DSP 之间的 HPI 接口。之所以使用 FPGA，是因为在系统中 FPGA 还包含有其他的功能设计。如果只有 MPC8272与 DSP 之间的 HPI 接口设计，使用 CPLD 即可完成，而不必浪费 FPGA 的资源。

也许对你有帮助的:

[基于 DSP 的 IIR 滤波器的设计](#)

[基于 DSP2812 平台及手势识别算法实现的手势电视机红外遥控系统](#)

[在 TMS320F28XXX DSP 上实现从 flash 拷贝整个程序到 ram 运行的方法](#)

介绍 **dsp** 知识, 为大家提供最新的 **dsp** 资讯, 更多内容可以去南京研旭电气科技有限公司的官网 www.njyxdq.com www.f28335.com 或者官方论坛, 嵌嵌 **dsp** 论坛

www.armdsp.net 进行交流学习

dsp 论坛 www.armdsp.net

dsp 开发板 www.njyxdq.com

嵌入式开发板 www.f28335.com