Semiconductors

UM10009_4

ISP1362 PCI Eval Board

October 2003

User's Guide

Rev. 4.0

Revision History:

Revision	**Date**	**Description**	**Author**
4.0	October 2003	Updated the CPLD code.	Alvim LIM
3.0	March 2003	Added 6-page schematics at the end of the document	Yuk-lin ONG
2.0	July 2002	• Updated Table 4-1, Section 4.6, Appendix A and Appendix D. • Removed Figure 4-3.	Jason ONG and SEOW Aun Kie
1.0	June 2002	First release.	Jason ONG, Alvin LIM and David WANG

We welcome your feedback. Send it to wired.support@philips.com.

Philips Semiconductors - Asia Product Innovation Centre
Visit www.semiconductors.philips.com/buses/usb or www.flexiusb.com

PHILIPS

This is a legal agreement between you (either an individual or an entity) and Philips Semiconductors. By accepting this product, you indicate your agreement to the disclaimer specified as follows:

DISCLAIMER

PRODUCT IS DEEMED ACCEPTED BY RECIPIENT. THE PRODUCT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, PHILIPS SEMICONDUCTORS FURTHER DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANT ABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE PRODUCT AND DOCUMENTATION REMAINS WITH THE RECIPIENT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL PHILIPS SEMICONDUCTORS OR ITS SUPPLIERS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THIS AGREEMENT OR THE USE OF OR INABILITY TO USE THE PRODUCT, EVEN IF PHILIPS SEMICONDUCTORS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CONTENTS

FIGURES

TABLES

1. Introduction

The ISP1362 is a single-chip Universal Serial Bus (USB) Host Controller (HC), Device Controller (DC) and On-The-Go (OTG) Controller.

The Host Controller portion of the ISP1362 complies with *Universal Serial Bus Specification Rev. 2.0*, supporting data transfer at full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s). The Device Controller portion of the ISP1362 also complies with *Universal Serial Bus Specification Rev. 2.0*, supporting data transfer at full-speed (12 Mbit/s). The OTG Controller is fully compliant with *On-The-Go Supplement to the USB 2.0 Specification Rev. 1.0a*.

The target applications of the ISP1362 are embedded systems, portable devices, digital still cameras, and so on. It has a 16-bit data bus for interfacing to the microprocessor and separate I/O addresses.

The ISP1362 PCI board is a stand-alone PC evaluation (eval) kit and is the successor of the ISP1362 ISA board. The ISP1362 ISA board required confusing jumper settings to enable its interrupts and DMA channels on the ISA bus. PCI abolishes these settings because it is on a Plug and Play platform—motherboard resources are allocated automatically by the PCI BIOS. The PCI bus also has much higher data throughput, which could not be accomplished using the ISA bus architecture. This will allow you to test the ISP1362 output levels to its limits as required by various users.

Figure 1-1 shows the ISP1362 PCI board.

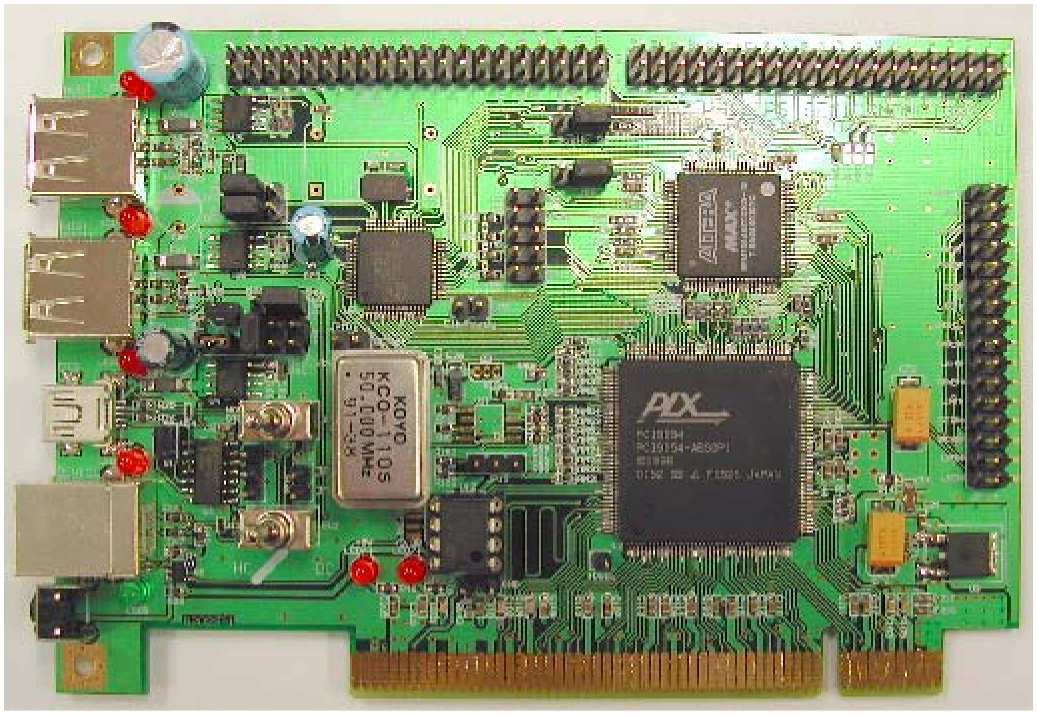


Figure 1-1: ISP1362 PCI board

2. PCI eval board block diagram

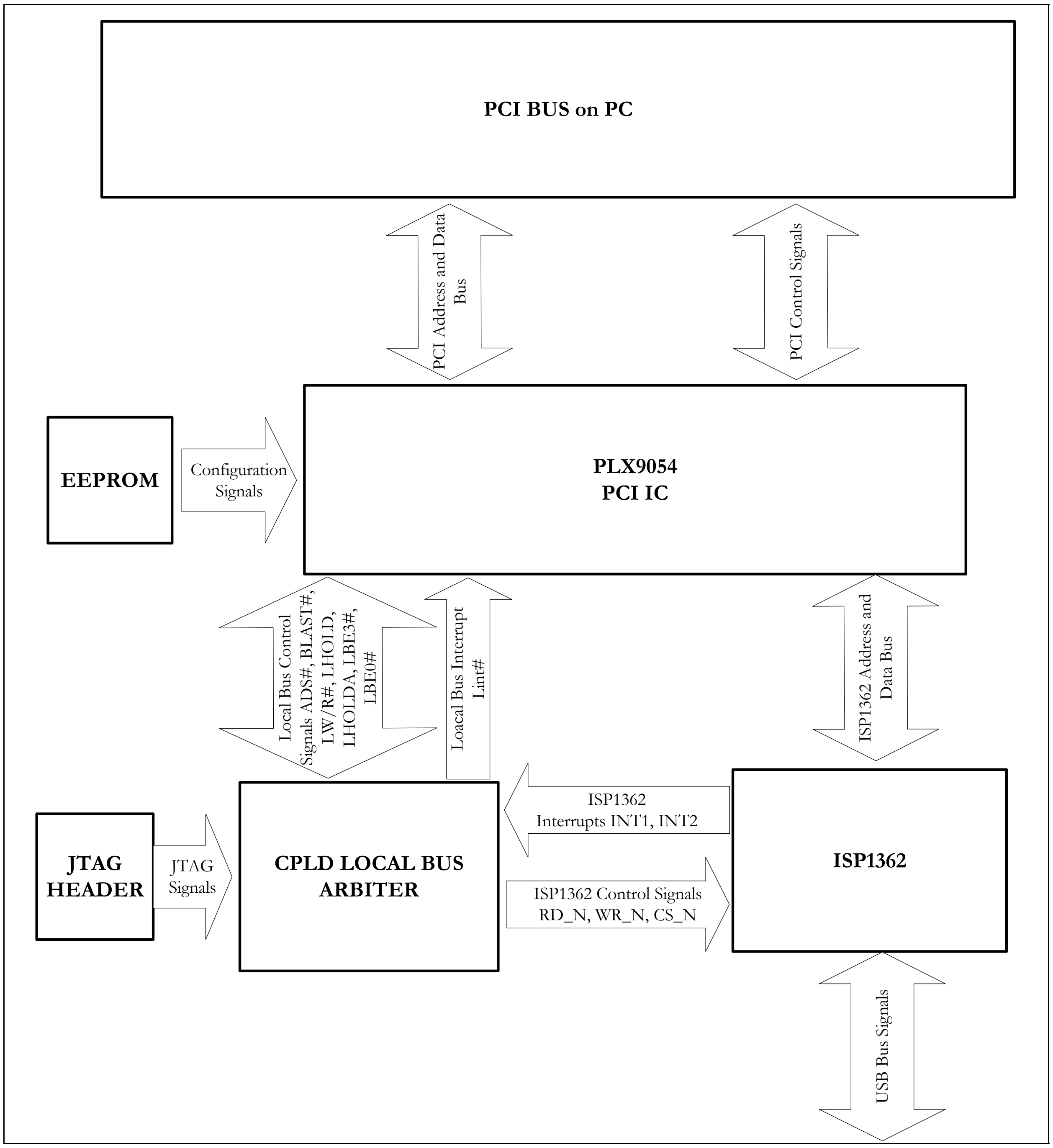

PCI BUS on PC
PCI Address and Data Bus
PCI Control Signals
EEPROM
Configuration Signals
PLX9054
PCI IC
Local Bus Control Signals ADS#, BLAST#, LW/R#, LHOLD, LHOLDA, LBE3#, LBE0#
Loacal Bus Interrupt Lint#
ISP1362 Address and Data Bus
ISP1362 Interrupts INT1, INT2
JTAG HEADER
JTAG Signals
CPLD LOCAL BUS ARBITER
ISP1362 Control Signals RD_N, WR_N, CS_N
ISP1362
USB Bus Signals

Figure 2-1: Block diagram

3. System requirements

3.1. System requirements for the OTG Controller

- Two PC motherboards with PCI slots
- Two PCs running on Microsoft® Windows® 98.

For specific information, refer to the respective ISP1362 PCI/DOS User's Guide.

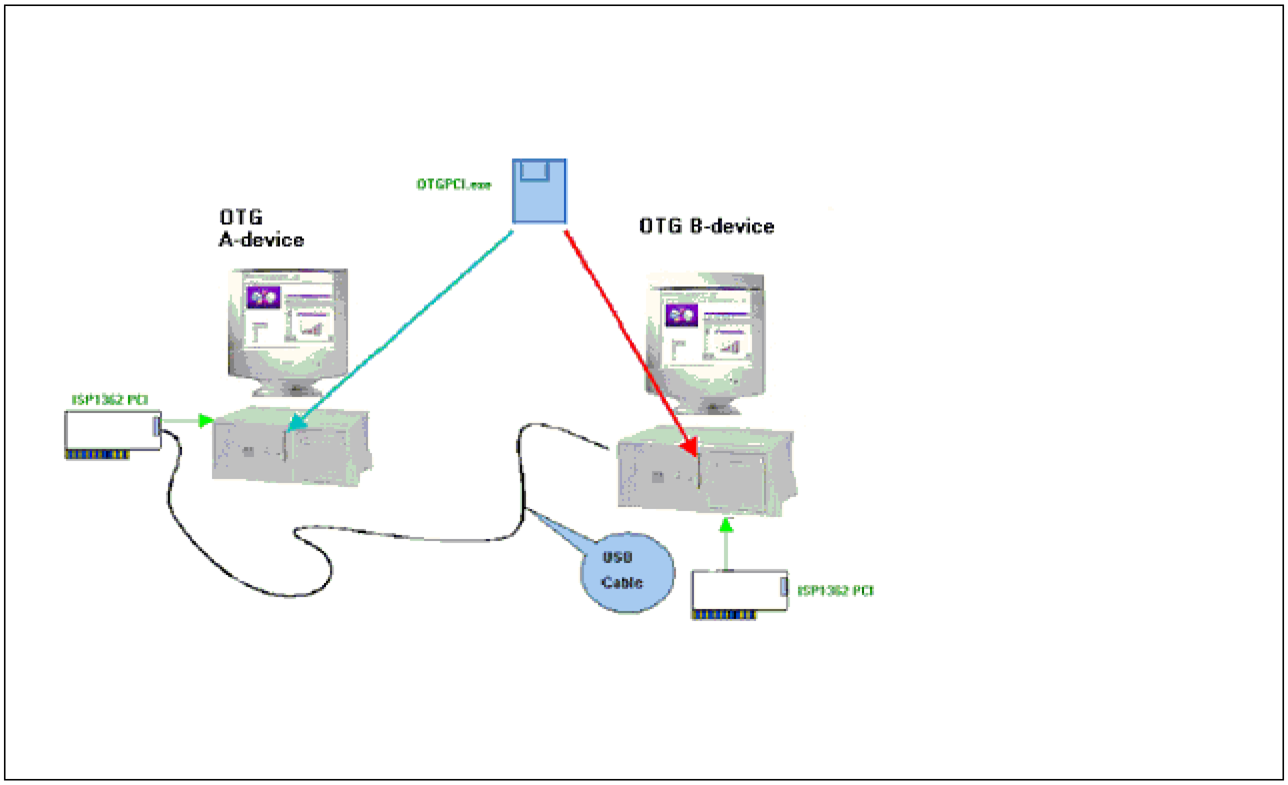


Figure 3-1: System requirements for the ISP1362 OTG Controller

3.2. System requirements for the Host Controller

- One PC motherboard with PCI slot
- One PC running on Microsoft Windows 98.

For specific information, refer to the respective ISP1362 PCI/DOS User's Guide.

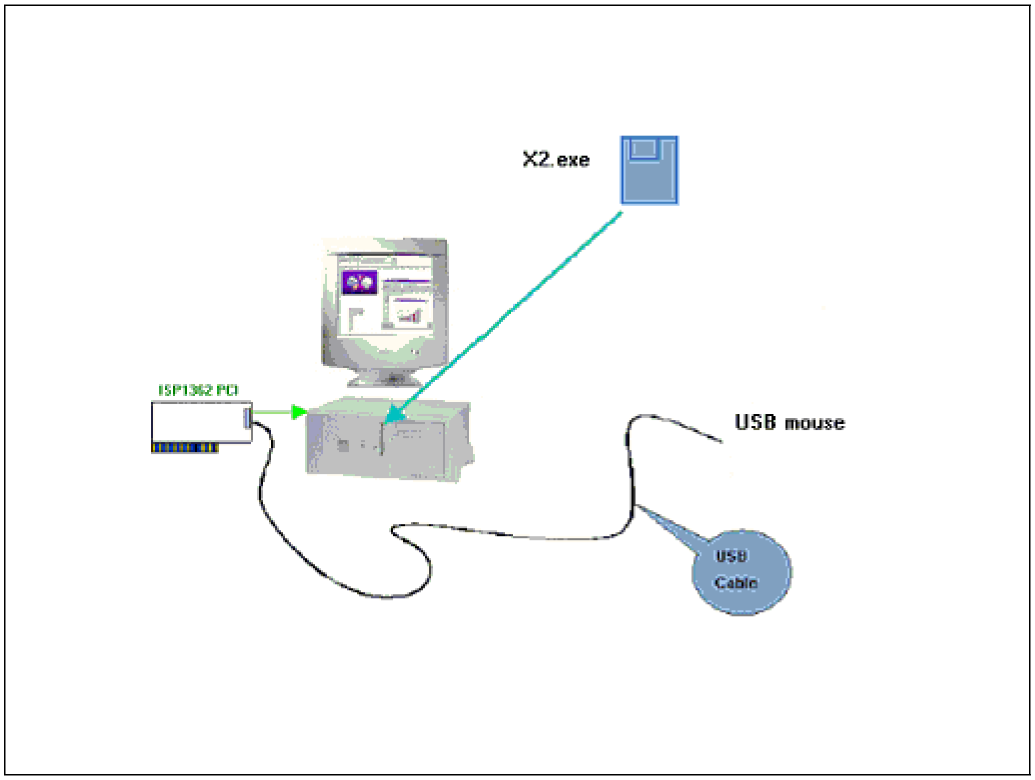


Figure 3-2: System requirements for the ISP1362 Host Controller

3.3. System requirements for the Device Controller

For the host PC:

- PC with USB motherboard or add-on card
- Microsoft Windows 98 Second Edition (SE) or Windows 2000.

For the device PC:

- PC with Microsoft DOS 6.x and a PCI slot
- ISP1362 PCI eval board.

For the firmware development:

- X86 CPU platform: Borland® Turbo C++ 3.0 or later
- ISP1362 eval CD.

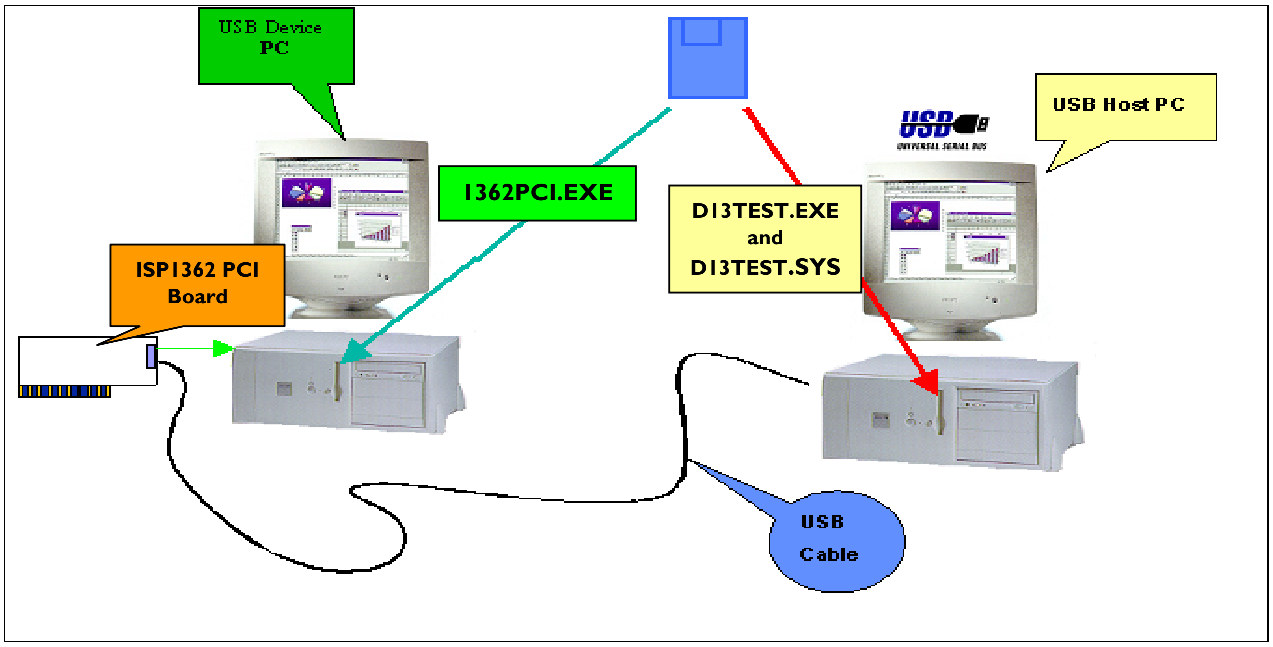


Figure 3-3: System requirements for the ISP1362 Device Controller

Table 3-1 provides a summary of the system requirements for the OTG Controller, the Host Controller and the Device Controller in the ISP1362.

Table 3-1: Summary of the system requirements for the ISP1362

Item	ISP1362 OTG Controller		ISP1362 Host Controller	ISP1362 Device Controller	
	Host PC	**Peripheral PC**	**Host PC**	**Host PC**	**Device PC**
OS used	Windows 98	Windows 98	Windows 98	Windows 2000 or Windows 98	Windows 98
Software used	OTGPCI.EXE	OTGPCI.EXE	X2.EXE	D13Test.EXE and D13Test.SYS	1362PCI.EXE

4. ISP1362 PCI eval kit

4.1. Installing the PCI eval kit

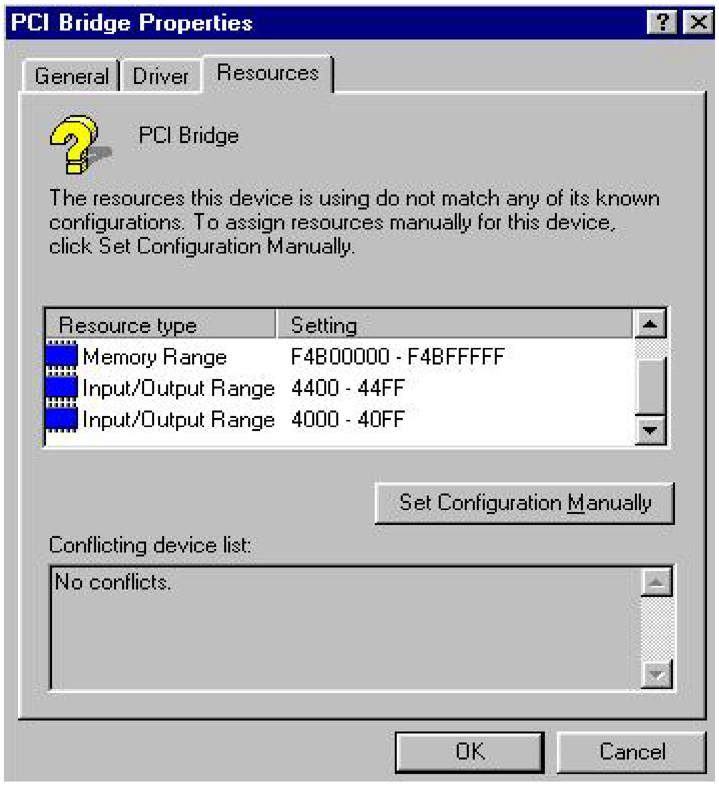
For testing the OTG Controller of the ISP1362, two ISP1362 PCI boards must be installed on two PCs. For testing only the Host Controller of the ISP1362, an ISP1362 PCI board must be installed on a PC. For testing only the Device Controller of the ISP1362, an ISP1362 PCI board must be installed on a device PC.

To install the ISP1362 PCI eval board on a PC:

1. Plug the PCI in the PC PCI slot.
2. Boot the PC to Microsoft Windows 98.
3. The PC must find a new PCI bridge device. Click **Next** a few times. Windows may not be able to locate the driver for the PCI bridge device. This is okay because no driver is required.
4. Reboot the PC.

4.2. PC resources assignment

Since the ISP1362 PCI eval board is on a Plug and Play platform as compared to its predecessor (the ISA board), I/O resources and interrupts are automatically allocated by the PC BIOS during booting. Parallel I/O access is performed between the PC and the ISP1362 PCI board. Figure 4-1shows the PC resources for the PCI bridge.



The resources this device is using do not match any of its known configurations. To assign resources manually for this device, click Set Configuration Manually.

Resource type	Setting
Memory Range	F4B00000 - F4BFFFFF
Input/Output Range	4400 - 44FF
Input/Output Range	4000 - 40FF

Conflicting device list:

No conflicts.

Figure 4-1: PC resources for the PCI bridge

4.3. Power supply and LED indicators

In the ISP1362 PCI eval board, the power supply input comes from +5.0 V of the PC PCI bus. Therefore, there is no need for any other external power supply input. +3.3 V is then regulated on the board from the +5.0 V supply.

There are some LEDs on the board to indicate the power supply status:

- DS1 is the +5.0 V indicator.
- DS2 is the +3.3 V indicator.
- DS3 is the GoodLink™ indicator for the Device Controller.

4.4. Headers, connectors and jumpers

There are two 20-bit interface headers, JP22 and JP23, and one 16-bit interface header, JP14; see Figure 4-2. The 20-bit headers contain the necessary address and data bus, as well as the control and output signals of the ISP1362. The 16-bit header serves as a convenient taping point to debug the local bus signals on the printed circuit board assembly (PCBA). Besides, header JP1 is used for the JTAG programming of the Altera® programmable logic device (PLD) on the PCB. The PCI board provides two USB downstream port connectors (P1 and P2), one OTG connector (P4) and one USB upstream port connector (P3), to interface to other USB peripherals.

There is also a reset switch (S1) for hardware reset of the PLD and the ISP1362.

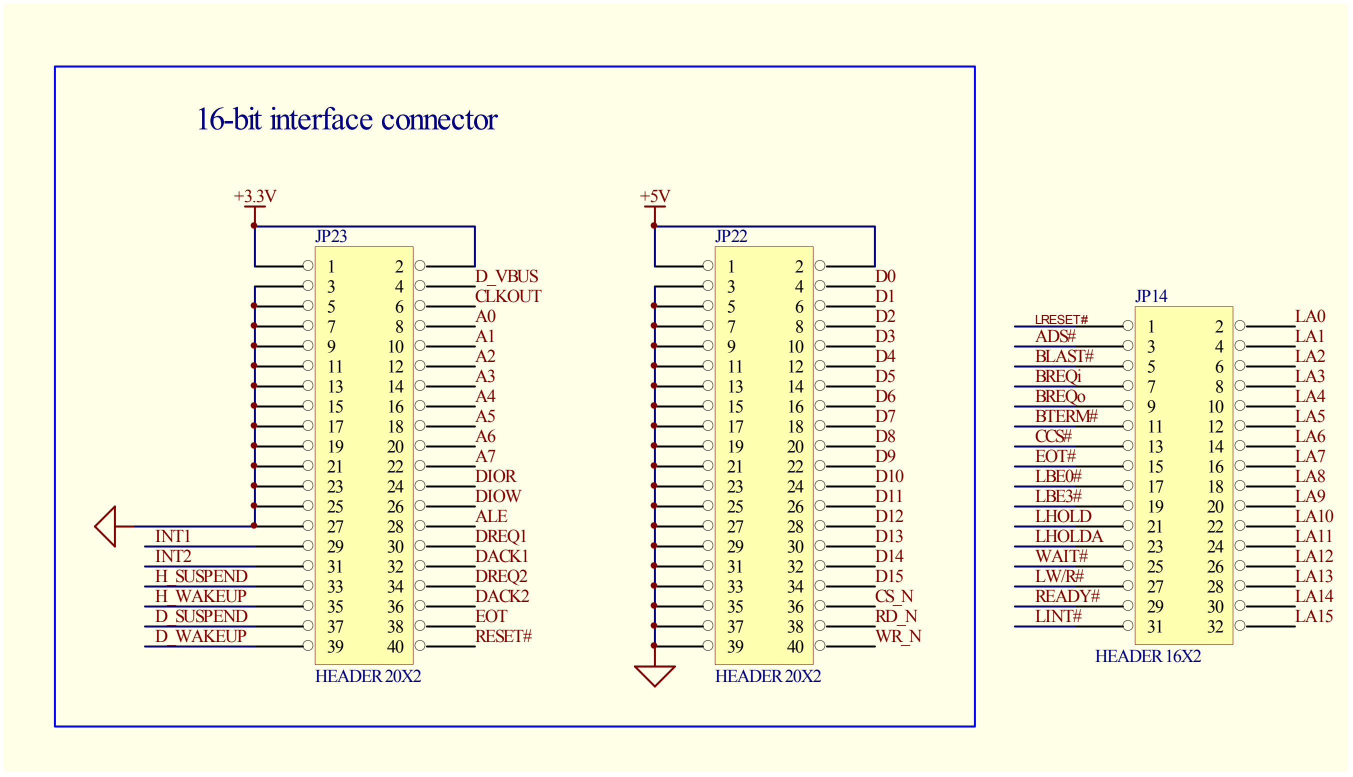


Figure 4-2: Interface headers

Table 4-1 shows the jumper and switch settings that must be configured before using the ISP1362 eval board.

Table 4-1: Jumper and switch settings

Jumper	Description	Setting
JP3	OTG mode pin select	Short <1–2> for the non-OTG mode Short <2–3> for the OTG mode [default]
JP4	V_{BUS} select	Short <5–6> for all modes [default]
JP6	ClkReady (TEST0)	Open for normal operation [default]
JP7	D_SUSWKUP_SET	Short <1–2> for normal operation [default] Short <2–3> to wake-up the Device Controller
JP8	H_SUSWKUP_SET	Short <1–2> for normal operation [default] Short <2–3> to wake-up the Host Controller
JP9	ID pin control	Open for the OTG mode and the device mode [default] In the host mode, short <1–2>
JP10	OCRef switch	Short for normal operation [default]
JP18	OTG OC disable	Short for all modes [default]

Jumper	Description	Setting
JP19	Host 2 enable or disable	Short <1–3> and <2–4> to enable the host port 2 [default] Short <3–5> and <4–6> to disable the host port 2
SW1	Switch 1	Switch to the OTG mode [default]
SW2	Switch 2	Switch to the Host Controller or the Device Controller in the non-OTG mode

The ISP1362 has two USB ports: port 1 and port2. Port 1 can be configured as an OTG, host or device port. The following tables provide simple reference to jumper and switch settings to configure port 1 as an OTG, host or device port. Port 2 can be used only as a host port.

Table 4-2: Default jumper and switch settings for the OTG mode

Jumper	Description	Setting
JP3	OTG mode select	Short <2–3> for the OTG mode [default]
JP9	ID pin control	Open for the OTG mode [default]
SW1	Switch 1	Switch to OTG

Table 4-3: Default jumper and switch settings for the host mode

Jumper	Description	Setting
JP3	OTG mode select	Short <1–2> for the non-OTG mode
JP9	ID pin control	In the host mode, short <1–2>
SW1	Switch 1	Switch to non-OTG
SW2	Switch 2	Switch to HC

Table 4-4: Default jumper and switch settings for the device mode

Jumper	Description	Setting
JP3	OTG mode select	Short <1–2> for the non-OTG mode
JP9	ID pin control	In the device mode, leave it open
SW1	Switch 1	Switch to non-OTG
SW2	Switch 2	Switch to DC

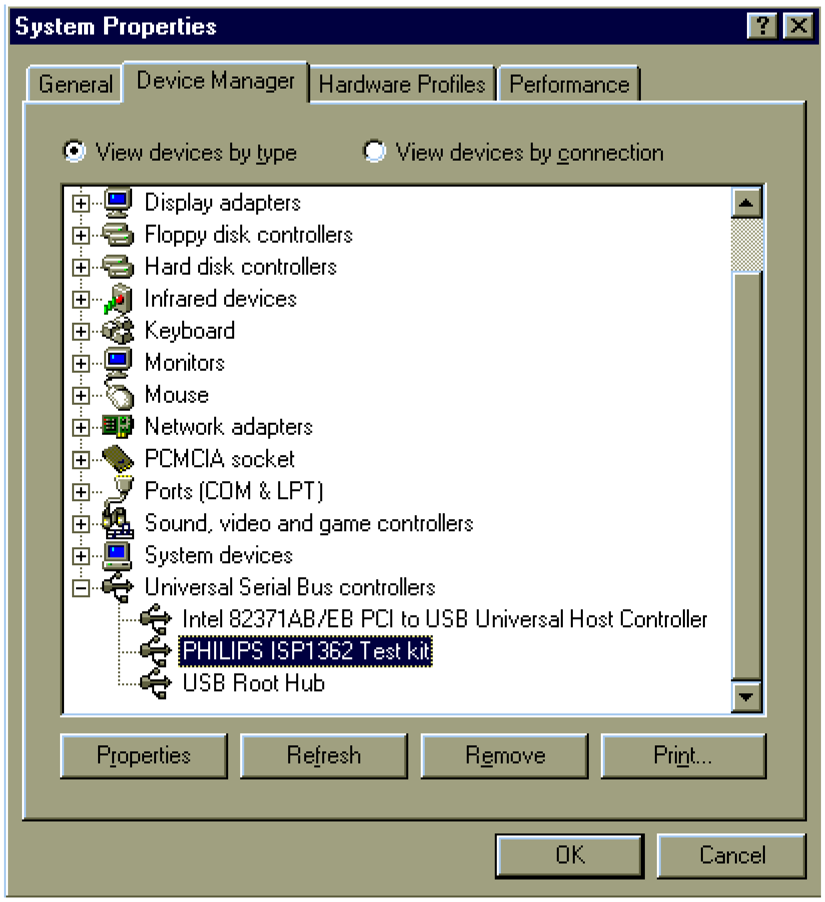
4.5. Charge pump of the OTG Controller

The present configuration of the ISP1362 PCI board supplies only 8 mA current through pin V_{BUS} of the OTG connector, with a 22 nF capacitor C77. To supply more current, a larger capacitor is required. For details, refer to the *ISP1362 Single-chip Universal Serial Bus On-The-Go controller* data sheet.

4.6. Installing the Device Controller hardware, firmware, INF and driver

1. Configure the jumper and switch settings in the device mode as given in Table 4-4.
2. Switch off the device PC.
3. Remove all the unnecessary cards on the device PC.
4. Plug ISP1362 PCI board in the PCI slot of the device PC.
5. Switch on the device PC.
6. Run the firmware (1362PCI.EXE) on the device PC under the DOS mode.
7. Plug the USB cable between the ISP1362 board and the USB host PC. If it is the first time the eval board is connected to the host PC, the host OS Device Manager will prompt for installation of the INF and the driver.
8. Select the location of D13Test.INF and D13Test.SYS from the ISP1362 eval kit and complete the installation procedure.

Figure 4-3 shows a successful installation of the Philips ISP1362 device driver.

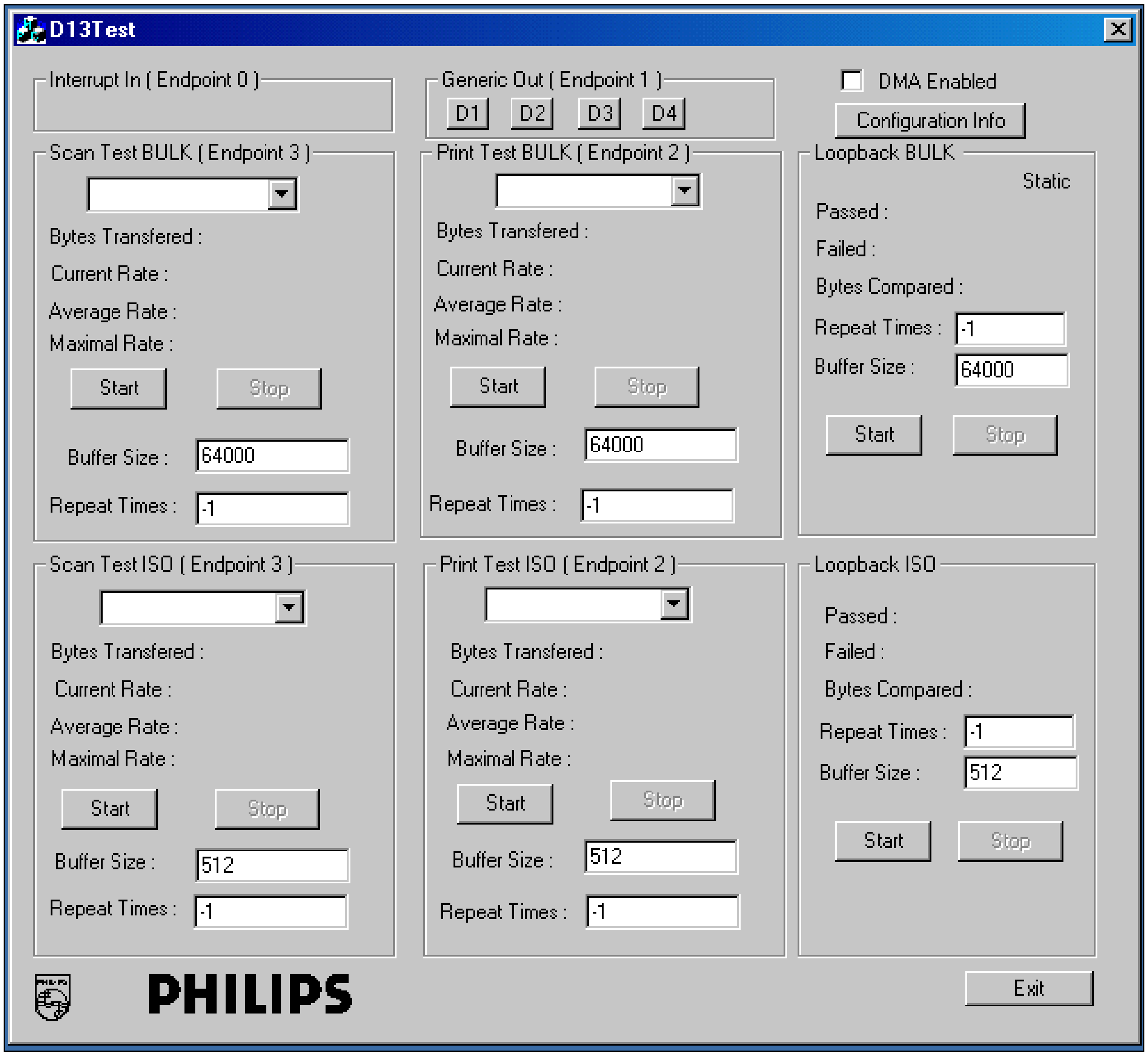


Display adapters
Floppy disk controllers
Hard disk controllers
Infrared devices
Keyboard
Monitors
Mouse
Network adapters
PCMCIA socket
Ports (COM & LPT)
Sound, video and game controllers
System devices
Universal Serial Bus controllers
Intel 82371AB/EB PCI to USB Universal Host Controller
PHILIPS ISP1362 Test kit
USB Root Hub

Figure 4-3: Philips ISP1362 device driver

4.7. Using the Device Controller host applet

The test applet, D13Test.EXE, exercises all the ISP1362 endpoints as shown in Figure 4-4.



Scan Test BULK (Endpoint 3)
Bytes Transfered :
Current Rate :
Average Rate :
Maximal Rate :
Buffer Size : 64000
Repeat Times : -1

Print Test BULK (Endpoint 2)
Bytes Transfered :
Current Rate :
Average Rate :
Maximal Rate :
Buffer Size : 64000
Repeat Times : -1

Loopback BULK
Static
Passed :
Failed :
Bytes Compared :
Repeat Times : -1
Buffer Size : 64000

Scan Test ISO (Endpoint 3)
Bytes Transfered :
Current Rate :
Average Rate :
Maximal Rate :
Buffer Size : 512
Repeat Times : -1

Print Test ISO (Endpoint 2)
Bytes Transfered :
Current Rate :
Average Rate :
Maximal Rate :
Buffer Size : 512
Repeat Times : -1

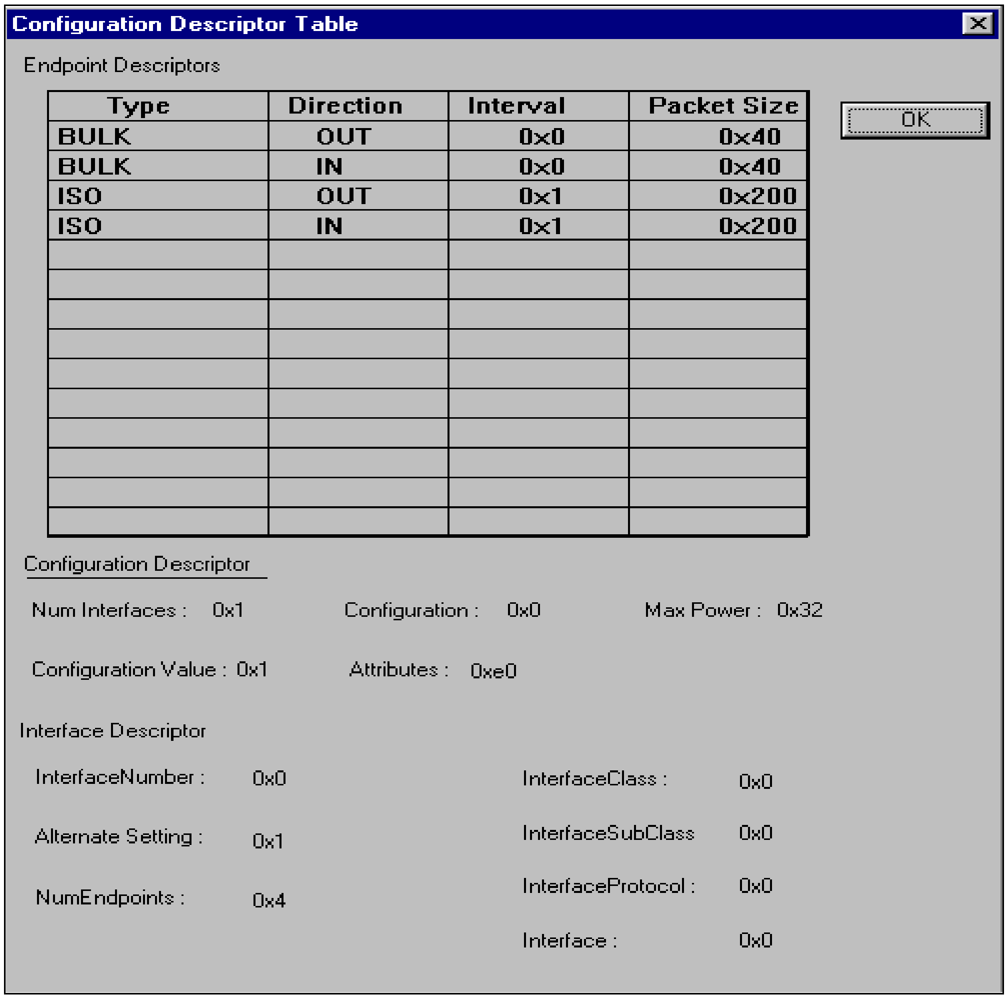
Loopback ISO
Passed :
Failed :
Bytes Compared :
Repeat Times : -1
Buffer Size : 512

Figure 4-4: D13TEST applet

Note: In the D13Test applet, Interrupt In (Endpoint 0) and Generic Out (Endpoint 1) are not in use.

Further testing of control endpoints can be done using standard USB CV test program that can be downloaded from the http://www.usb.org/ web site.

With a device already connected, if you click the **Configuration Info** button, the Configuration Descriptor Table that contains the current configuration descriptor is displayed as given in Figure 4-5.



Configuration Descriptor Table

Endpoint Descriptors

Type	Direction	Interval	Packet Size
BULK	OUT	0x0	0x40
BULK	IN	0x0	0x40
ISO	OUT	0x1	0x200
ISO	IN	0x1	0x200

Configuration Descriptor

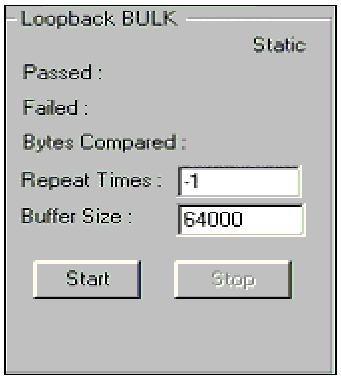
Num Interfaces :	0x1	Configuration :	0x0	Max Power :	0x32
Configuration Value :	0x1	Attributes :	0xe0		

Interface Descriptor

InterfaceNumber :	0x0	InterfaceClass :	0x0
Alternate Setting :	0x1	InterfaceSubClass	0x0
NumEndpoints :	0x4	InterfaceProtocol :	0x0
		Interface :	0x0

Figure 4-5: Configuration description table

Bulk loopback test, bulk print test and bulk scan test can be easily performed by entering the desired transfer byte value in the **Buffer Size** field and the number of transfers that you would like to have in the **Repeat Times** field. The buffer size is limited to 64000 bytes. A Repeat Times value of "-1" will result in a continuous test, and a Repeat Times value of "0" will cause the applet to stop responding.



Loopback BULK

Static

Passed :

Failed :

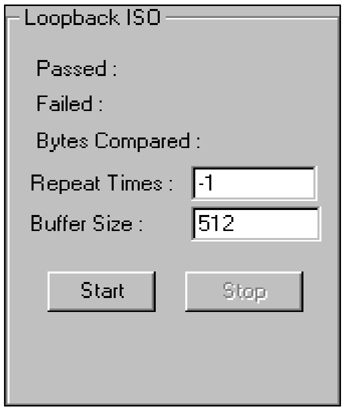
Bytes Compared :

Repeat Times : -1

Buffer Size : 64000

Figure 4-6: Loopback bulk

ISO loopback test, ISO print test and ISO scan test can be easily performed by entering the value 512 in the **Buffer Size** field, and the number of transfer that you would like to have in the **Repeat Times** field. The buffer size for all ISO tests must be specified as 512 bytes. A Repeat Times value of "-1" will result in a continuous test, and a Repeat Times value of "0" will cause the applet to stop responding.


Loopback ISO

Passed :

Failed :

Bytes Compared :

Repeat Times : -1

Buffer Size : 512

Figure 4-7: Loopback ISO

Table 4-5 shows the description of endpoints operations on the ISP1362 eval board.

Table 4-5: Description of endpoints operations

The test applet and the ISP1362 eval board support three test modes: loopback, print and scan. The firmware uses I/O accesses on this endpoint.

Endpoint Number	Endpoint Type	Operations
5	ISO OUT	This pipe is defined as isochronous OUT pipe.
6	ISO IN	This pipe is defined as isochronous IN pipe.
3	Bulk OUT	This pipe is defined as bulk OUT pipe.
4	Bulk IN	This pipe is defined as bulk IN pipe.

Three test modes:

- **Scan mode**: In this mode, the ISP1362 eval board acts like a scanner. It sends data packets to the host PC as fast as possible. This mode is used to evaluate the isochronous IN and bulk IN transfer rates.
- **Print mode**: In this mode, the ISP1362 eval board acts like a printer. It receives data packets from the host PC as fast as possible. This mode is used to evaluate the isochronous OUT and bulk OUT transfer rates.
- **Loopback mode**: In this mode, the ISP1362 eval board receives data packets on the isochronous OUT (or bulk OUT) endpoint and sends them back to the host PC on the isochronous IN (or bulk IN) endpoint. This mode is used to test the data integrity of transfers.

The **DMA Enabled** check box allows you to switch between the bulk DMA transfer and the bulk PIO transfer. The ISP1362 PCI eval board, however, supports only the PIO mode. Therefore, irrespective of what mode you select in D13Test.EXE, data transfer will always be in the PIO mode.

The **Buffer Size** setting in the test applet is determined by the firmware and hardware ability of the eval board. For the PC kit, the maximum size is limited to 64000 bytes for the bulk transfer. For the ISO transfer, the transfer size is predetermined as 512 bytes.

5. References

- *ISP1362 Single-chip Universal Serial Bus On-The-Go controller* data sheet
- *ISP1362 PCI/DOS User's Guide*
- *Universal Serial Bus Specification Rev. 2.0*
- *On-The-Go Supplement to the USB 2.0 Specification Rev. 1.0a*
- *PLX Technology PCI 9054 Data Book.*

Appendix A. ISP1362 PCI eval board PLD code

```
--*********************************************************************************************--
--** The PLX9054 project is a design which implements glue logic between PLX9054 and ISP1362 **--
--**                                                                                         **--
--** Version 1.2c      To cater for WINCE MultiThread                                        **--
--** Version 1.1       Add PCI_disable_function                                              **--
--**                                                                                         **--
--** Function:                                                                               **--
--**    1. Only implement parallel I/O write/read, no DMA operation in this version          **--
--**    2. When parallel I/O write, only non-burst mode is implemented, Burst mode and       **--
--**       BTERM mode  are not implemented in this version                                   **--
--**    3. Because ISP1362 runs with first command access followed by data access, PCI       **--
--**       accesses consist of one single write PCI transfer with one or more read/write     **--
--**       data PCI transfer                                                                 **--
--**    4. READY# signals are used to control interconnect timings                           **--
--*********************************************************************************************--

library ieee;
use ieee.std_logic_1164.all;

entity plx9054 is
   port (
      --** PLX9054 interface
      clk          : in    std_logic;                     -- system clock
      reset_n      : in    std_logic;                     -- system reset
      ads_n        : in    std_logic;                     -- PLX address strobe
      lhold        : in    std_logic;                     -- PLX hold request
      lholda       : out   std_logic;                     -- EPLD hold acknowledge
--       blast_n      : in    std_logic;                     -- PLX burst last
      lwr_n        : in    std_logic;                     -- PLX write/read, '1' for write, '0'
for read
      ready_n      : out   std_logic;                     -- PLX data ready
      bterm_n      : out   std_logic;
      breqi        : out   std_logic;
      otg_dmpu_l   : out   std_logic;
      otg_vbus_pd  : out   std_logic;
      otg_id_gnd   : out   std_logic;
      ccs_n              : out   std_logic;
      lint_n      : out   std_logic;
      laddr        : in    std_logic_vector(2 downto 0);  -- PLX address
      data               : inout   std_logic_vector(15 downto 0); -- PLX data bus
--       data_tmp     : out   std_logic;
      --** ISP1362 interface
      cs_n         : out  std_logic;                      -- ISP1362 chip select
      rd_n         : out  std_logic;                      -- ISP1362 read
      wr_n         : out  std_logic;                      -- ISP1362 write
      dack1_n      : out  std_logic;                      -- ISP1362 HC Dack1
      dack2_n      : out  std_logic;                      -- ISP1362 DC Dack2
      eot_n        : out  std_logic;                      -- ISP1362 EOT#
      a_sel        : out  std_logic_vector(1 downto 0);   -- ISP1362 access mode:
                                                          -- a1: '0' for USB host controller
                                                          --     '1' for USB device controller
                                                          -- a0: '1' for accessing command port
                                                          --     '0' for accessing data port
      int          : in   std_logic_vector(2 downto 1)    -- ISP1362 interrupt
      );
end plx9054;

architecture rtl of plx9054 is

signal  a0_delay              :       std_logic;
signal  a1_delay              :       std_logic;
signal  disable_PCI_int       :       std_logic;
signal  data_out_enable       :   std_logic;
signal   data_in                      :       std_logic_vector(15 downto 0);
signal   data_out             :       std_logic_vector(15 downto 0);

begin

       data    <= data_out when (data_out_enable = '1') else "ZZZZZZZZZZZZZZZZ";
       data_in <= data;
```

```
--              a_sel(0) <= laddr(0);
--              a_sel(1) <= laddr(1);

-- ISP1362 Interrupt process
   interrupt_process: process(reset_n, clk)
begin
   if reset_n = '0' then
        lint_n                <= '1';
               disable_PCI_int       <= '0';
               a_sel(0)              <= '0';
               a_sel(1)              <= '0';
               data_out_enable <= '0';
   elsif (clk'event and clk = '1') then
               if data_out_enable ='1' then
                       data_out_enable <= '0';                       --v1.2c
               end if;
               a0_delay <= laddr(0);
               a1_delay <= laddr(1);
--             a_sel(0) <= laddr(0) or a0_delay;
--             a_sel(1) <= laddr(1) or a1_delay;

               if laddr(2) = '0' then
                       a_sel(0) <= a0_delay;
                       a_sel(1) <= a1_delay;
               end if;

               if (ads_n = '0' and laddr(2) = '1' and laddr(1) = '0' and laddr(0) = '0') then
                       disable_PCI_int       <= '1';
               elsif (ads_n = '0' and laddr(2) = '1' and laddr(1) = '0' and laddr(0) = '1') then
                       disable_PCI_int       <= '0';
               elsif (ads_n = '0' and laddr(2) = '1' and laddr(1) = '1' and laddr(0) = '0') then
--                     disable_PCI_int <= '1';
                       data_out_enable <= '1';
                       if ( int(1) = '0' ) then
                               data_out <= "0000000000000001";              -- output 0x0001
                       elsif ( int(2)='0' ) then
                               data_out <= "0000000000000010";              -- output 0x0002
                       elsif ( int(1) = '0' and int(2) = '0' ) then
                               data_out <= "0000000000000011";              -- output 0x0003
                       else
                               data_out <= "0000000000000000";              -- output 0x0000
                       end if;
--             elsif (ads_n = '0' and laddr(2) = '1' and laddr(1) = '1' and laddr(0) = '1') then
--                     data_out_enable <= '0';
               end if;

               if ( (int(1) = '0' or int(2) ='0') and disable_PCI_int = '0') then
                       lint_n  <= '0';                -- Low Level of (INT1 or INT2) will generate
local interrupt
               else
                       lint_n  <= '1';
               end if;

--             if (data_out_enable ='1') then
--                     data <= data_out;
--             elsif (data_out_enable ='0') then
--                     data <= "ZZZZZZZZZZZZZZZZ";
--             end if;

--             data_in <= data;

   end if;

end process interrupt_process;

--------------------------------------------------------------------------------------
 -- Main State Machine executes
state_machine_process: process(reset_n, clk)
--   type sm_state   is (plx_start, cmd_ads, cmd_blst, cmd_wait, data_ads,data_blst,
--                        data_hold, data_wait, s5, s6);
   type sm_state   is (plx_start, cmd_ads, cmd_blst, cmd_wait);
   variable state  :  sm_state;

begin
   if reset_n = '0' then
         state := plx_start;
--       data_tmp    <= blast_n or data(0) or data(1) or data(2) or data(3) or data(4) or data(5)
or data(6) or data(7) or data(8)
```

```
--                                             or data(9) or data(10) or data(11) or data(12) or
data(13) or data(14) or data(15);
         otg_vbus_pd <= '0';
         otg_id_gnd  <= '1';
         otg_dmpu_l  <= '1';
         dack1_n     <= '1';
         dack2_n        <= '1';
         eot_n          <= '1';
      breqi       <= '0';
      bterm_n     <= '1';
      ccs_n       <= '1';
      ready_n     <= '1';
      cs_n        <= '1';
      rd_n        <= '1';
      wr_n        <= '1';
   elsif (clk'event and clk = '1') then
      -- synchronize PLX external control signals

      if lhold = '1' then
               lholda    <= '1';
      else
               lholda     <= '0';
      end if;

------------------------------------------------------------------------------------------
      -- state machine start
      case state is
                 when plx_start =>                                    -- s0
                   ready_n   <= '1';
            cs_n        <= '1';
            wr_n      <= '1';
            rd_n      <= '1';
--             if(ads_n = '0' and laddr(2)='0') then
               if(ads_n='0') then
               state      := cmd_ads;
            else
               state := plx_start;
            end if;

         when cmd_ads =>
               ready_n <= '1';
            cs_n       <= '1';
            wr_n    <= '1';
            rd_n    <= '1';
                      state   := cmd_blst;

--            if blast_n = '0' then
--               state     := cmd_blst;
--                    elsif blast_n = '1' then
--                           state   := cmd_wait;
--              else state     := cmd_ads;
--            end if;

         when cmd_blst =>
                   ready_n <= '0';

                      if laddr(2)='1' then    -- v1.2b higher address should not assert cs
--                    if ((laddr(2)='1' and laddr(1)='0') or (laddr(2)='1' and laddr(1)='1' and
laddr(0)='0')) then
                      cs_n    <= '1'; -- enabling and disable pci interrupt should not assert cs
                      else
                   cs_n       <= '0';
                      end if;

            if lwr_n = '1' then
              wr_n  <= '0';
              rd_n     <= '1';
            elsif lwr_n = '0' then
              rd_n   <= '0';
              wr_n     <= '1';
            end if;
            state       := cmd_wait;  -- to accomodate wr, rd pulse width of 1362

                when cmd_wait =>
                   ready_n <= '0';

                      if laddr(2)='1' then          -- v1.2b higher address should not assert cs
```

```
--                      if ((laddr(2)='1' and laddr(1)='0') or (laddr(2)='1' and laddr(1)='1' and
laddr(0)='0')) then
                        cs_n    <= '1'; -- enabling and disable pci interrupt should not assert cs
                        else
                     cs_n       <= '0';
                        end if;

            if lwr_n = '1' then
              wr_n      <= '0';
              rd_n            <= '1';
            elsif lwr_n = '0' then
              rd_n      <= '0';
              wr_n            <= '1';
            end if;
               state := plx_start;

     end case;

   end if;

end process state_machine_process;

end rtl;
```

Appendix B. ISP1362 bill of materials

Part Type	Quantity	Designator
PCB ISP 1362 PCI Eval Rev 1.1	1	—
Bracket for 02023	1	—
Screw Pan M3.0x 6mm	2	—
Capacitor SMD 0603 18pF / 50V 5%	2	C73 C74
Capacitor SMD 0603 47pF / 50V 5%	5	C38 C83 C91 C95 C190
Capacitor SMD 0603 100pF / 50V 5%	1	C27
Capacitor SMD 0603 10nF / 50V 10% X7R	25	C24 C41 C42 C43 C44 C45 C47 C48 C49 C50 C51 C52 C106 C107 C108 C109 C110 C111 C112 C113 C114 C115 C116 C117 C118
Capacitor SMD 0603 22nF / 50V 10%	1	C77
Capacitor SMD 0603 47nF / 25V 10%	5	C101 C102 C103 C104 C105
Capacitor SMD 0603 100nF / 50V 20%	40	C1 C8 C9 C25 C28 C29 C30 C31 C32 C33 C34 C35 C36 C37 C39 C40 C58 C59 C60 C61 C62 C63 C64 C65 C71 C75 C76 C78 C80 C81 C82 C85 C86 C87 C89 C90 C205 C209 C210 C211
Capacitor Ele GPS-R 47μF / 25V	1	C88
Capacitor Ele GPS-R 47μF / 16V	1	C79
Capacitor Ele GPS-R 220μF / 16V	1	C180
Capacitor Tan SMD-A 10μF / 10V 20%	1	C26
Capacitor Tan SMD-D 10μF / 25V 10%	2	C70 C72
Socket IC Turn Pin 8-Way	1	U12
Header Pin 0.100" 1-Way Gold	3	G1 JP11 JP12
Header Pin 0.100" 1x2-Way Gold	4	JP6 JP10 JP18 JP26
Header Pin 0.100" 1x3-Way Gold	5	JP3 JP7 JP8 JP9 JP15
Header Pin 0.100" 2x3-Way Gold	2	JP4 JP19
Header Pin 0.100" 2x5-Way Gold	1	JP1
Header Pin 0.100" 2x16-Way Gold	1	JP14
Header Pin 0.100" 2x20-Way Gold	2	JP22 JP23
0.100" Micro Shunt	8	JP3(p1-2) JP4(p1-2) JP4(p5-6) JP7(p1-2) JP8(p1-2) JP10 JP19(p1-3) JP19(p2-4)
USB Type B RA 4-Way 61729-0010B	1	P3
USB Type A RA 4-Way 87520-0010B	2	P1 P2

Part Type	Quantity	Designator
LED 3mm Green Diffused	1	LED5
LED 3mm Red Diffused	6	LED1 LED2 LED3 LED4 LED5 LED7
Fuse SMD NFM40P12C223	1	CF1
Transistor NDS9435A	3	U4 U5 U6
Transistor ZVP2106	1	Q1
Transistor ZVN4206A	1	Q2
Resistor SMD 0603 1/16W 5% 0R	23	R5 R6 R7 R9 R10 R12 R13 R15 R16 R25 R26 R27 R28 R29 R30 R31 R32 R62 R152 R153 R163 FB4 FB5
Resistor SMD 0603 1/16W 1% 22R	6	R67 R68 R75 R81 R82 R86
Resistor SMD 0603 1/16W 1% 499R	1	R99
Resistor SMD 0603 1/16W 5% 510R	1	R18
Resistor SMD 0603 1/16W 1% 1K	7	R22 R48 R52 R53 R54 R55 R150
Resistor SMD 0603 1/16W 1% 1K5	2	R40 R100
Resistor SMD 0603 1/16W 5% 2K	1	R23
Resistor SMD 0603 1/16W 1% 3K9	1	R74
Resistor SMD 0603 1/16W 1% 4K7	4	R1 R2 R3 R4
Resistor SMD 0603 1/16W 1% 10K	41	R17 R20 R21 R38 R41 R45 R47 R51 R56 R57 R58 R59 R61 R63 R88 R89 R90 R91 R97 R98 R104 RN11 RN12 RN13 RN14 RN21 RN22 RN23 RN24 RN31 RN32 RN33 RN34 RN41 RN42 RN43 RN44 RN51 RN52 RN53 RN54
Resistor SMD 0603 1/16W 1% 15K	4	R42 R43 R44 R60
Resistor SMD 0603 1/16W 1% 22K	1	R33
Resistor SMD 0603 1/16W 1% 100K	6	R34 R35 R36 R37 R39 R69
Resistor SMD 0603 1/16W 5% 1M	2	R93 R94
Resistor SMD 0805 1/10W 5% 0R	1	R141
Switch Tact 6mm Right Angle B: 3.85mm	1	S1
Switch Toggle Miniature 48Vac 0.05A,DPDT	2	SW1 SW2
Inductor Solid SMD-4516 BLM41A600SPT	3	FB1 FB2 FB3
IC 74HCT00D	1	U1
IC M93C56BN6	1	U12
IC LM1117DT33	1	U3
PCI9054AB50PI(176 Pin PQFP)	1	U2
IC EPM7064AETC100-10	1	U9

Part Type	Quantity	Designator
IC MAX6306UK30D2	1	U8
Oscillator 50.000MHz	1	U10
Crystal 12.000MHz FA-365	1	X2
Diode Zener STZ5.6N	1	D3
IC ISP1362	1	U7
OTG Socket	1	P4

Appendix C. ISP1362 PCI binary file for EEPROM

```
5406    10B5    0680    000B    0000    010A    0000    0000
0000    0000    FFFF    FF01    2000    0001    0161    000C
0030    0500    0000    0000    0000    0010    8941    0041
FF00    0000    4000    0000    5000    0000    0000    2000
0000    0000    9054    10B5    FFFF    FF01    2000    0001
0000    0141    0000    4C06
```

After reset, the PCI chip (PCI 9054) on the ISP1362 PCI eval board reads the contents of EEPROM for its PCI Configuration registers. For more details, refer to the *PCI 9054 Data Book from PLX Technology*.

Appendix D. ISP1362 PCI eval board schematics

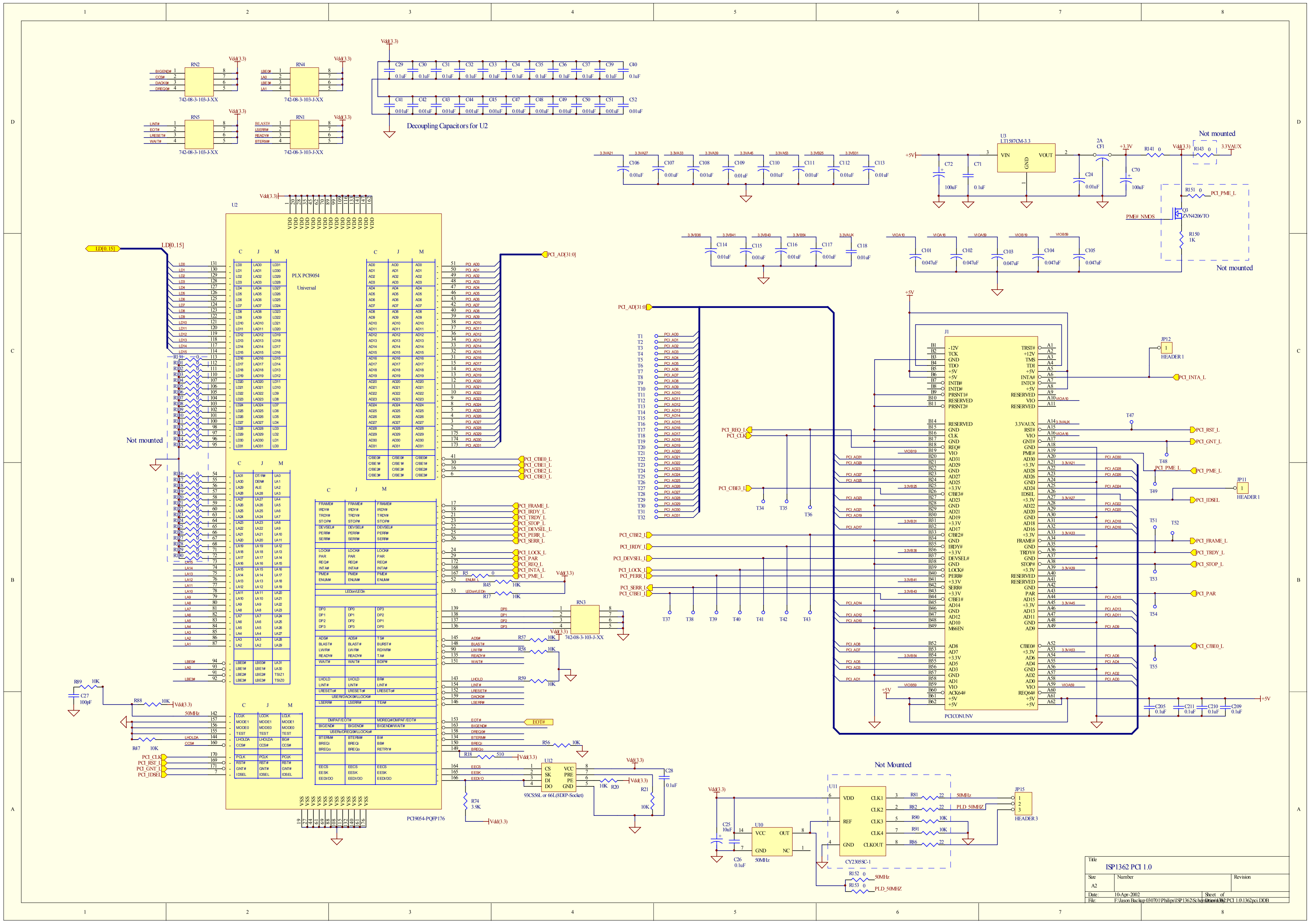


Figure D-1: Schematic of PLX9054

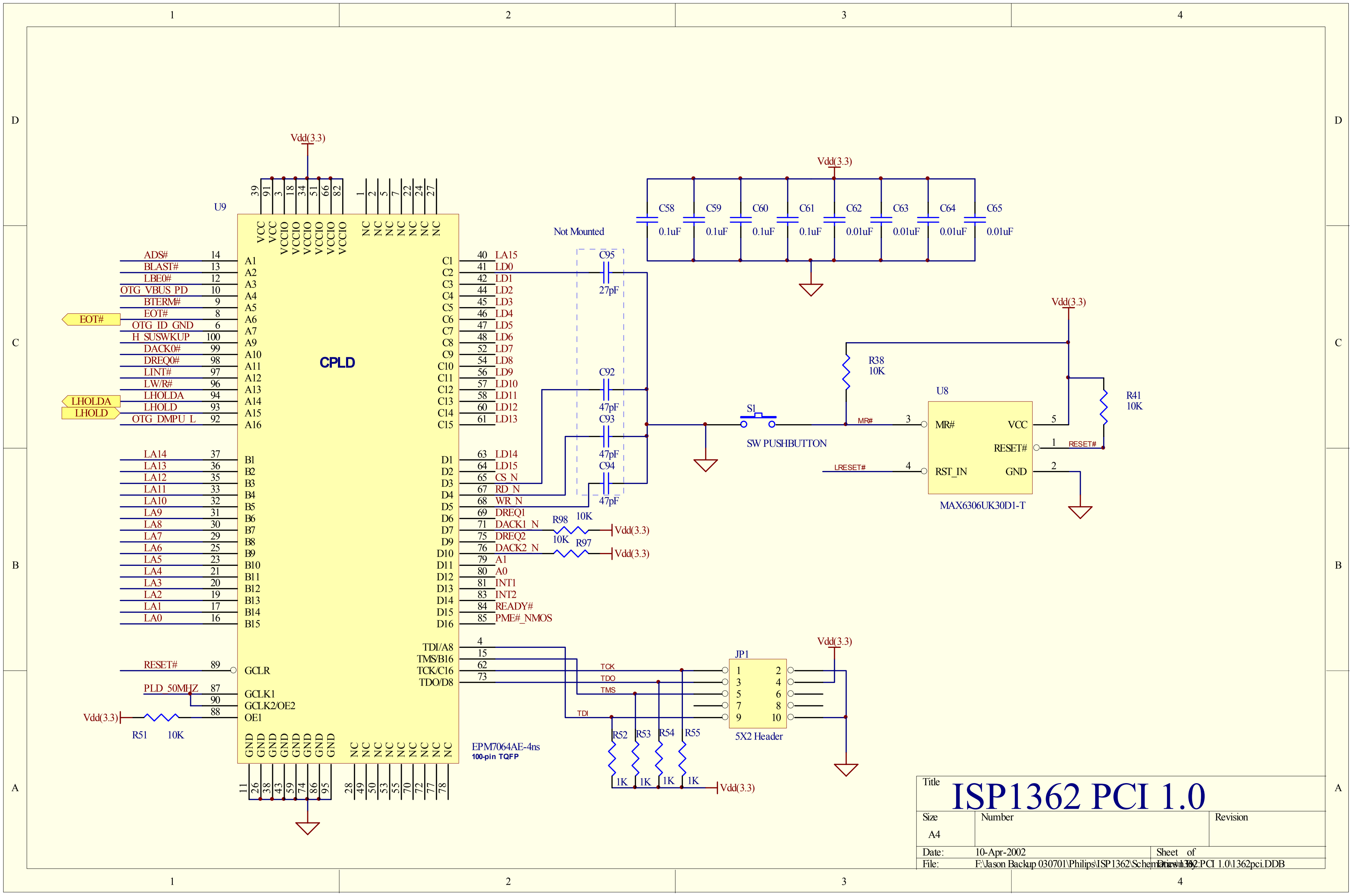


Figure D-2: Schematic of PLD

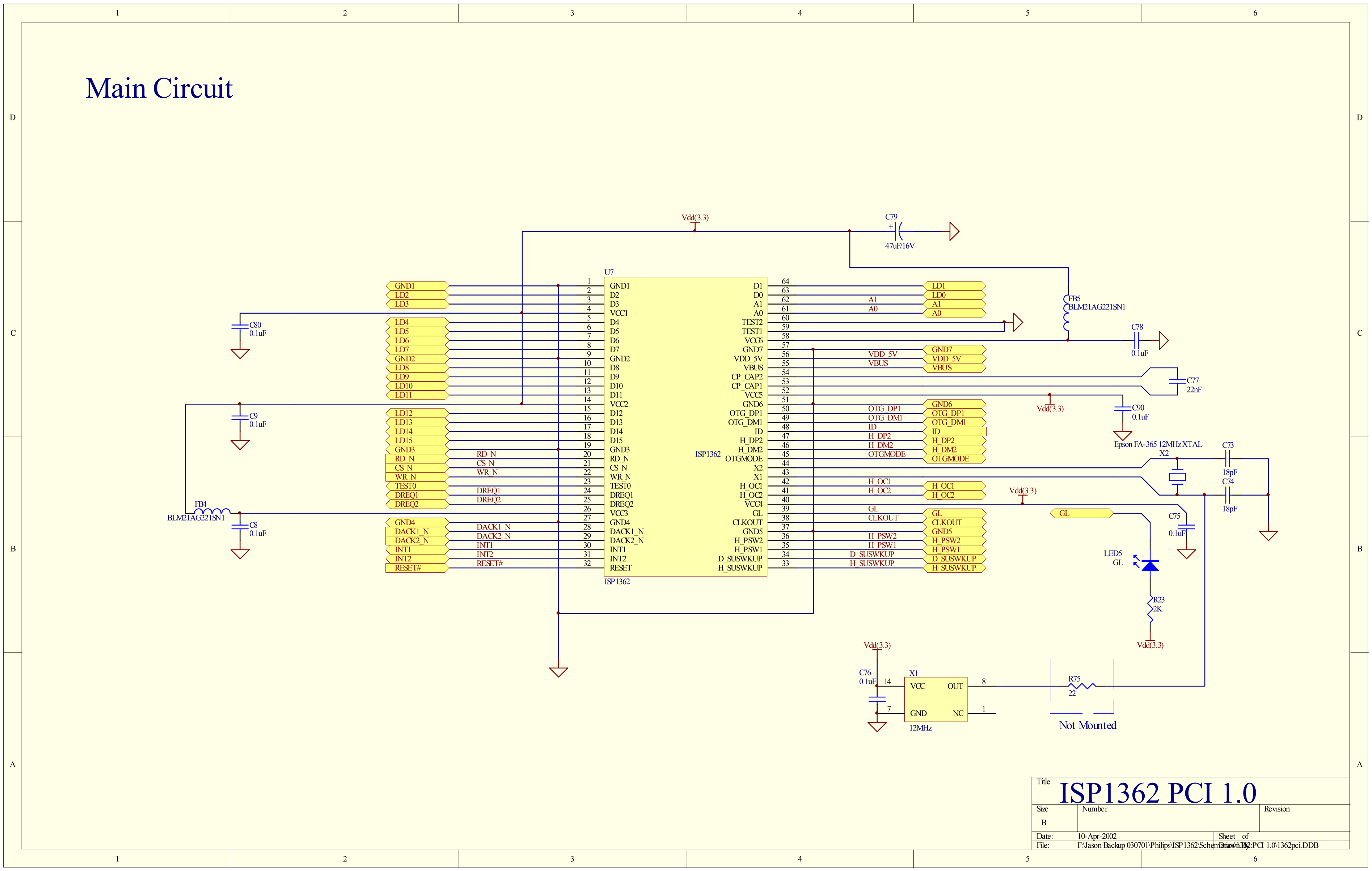


Figure D-3: Schematic of the ISP1362

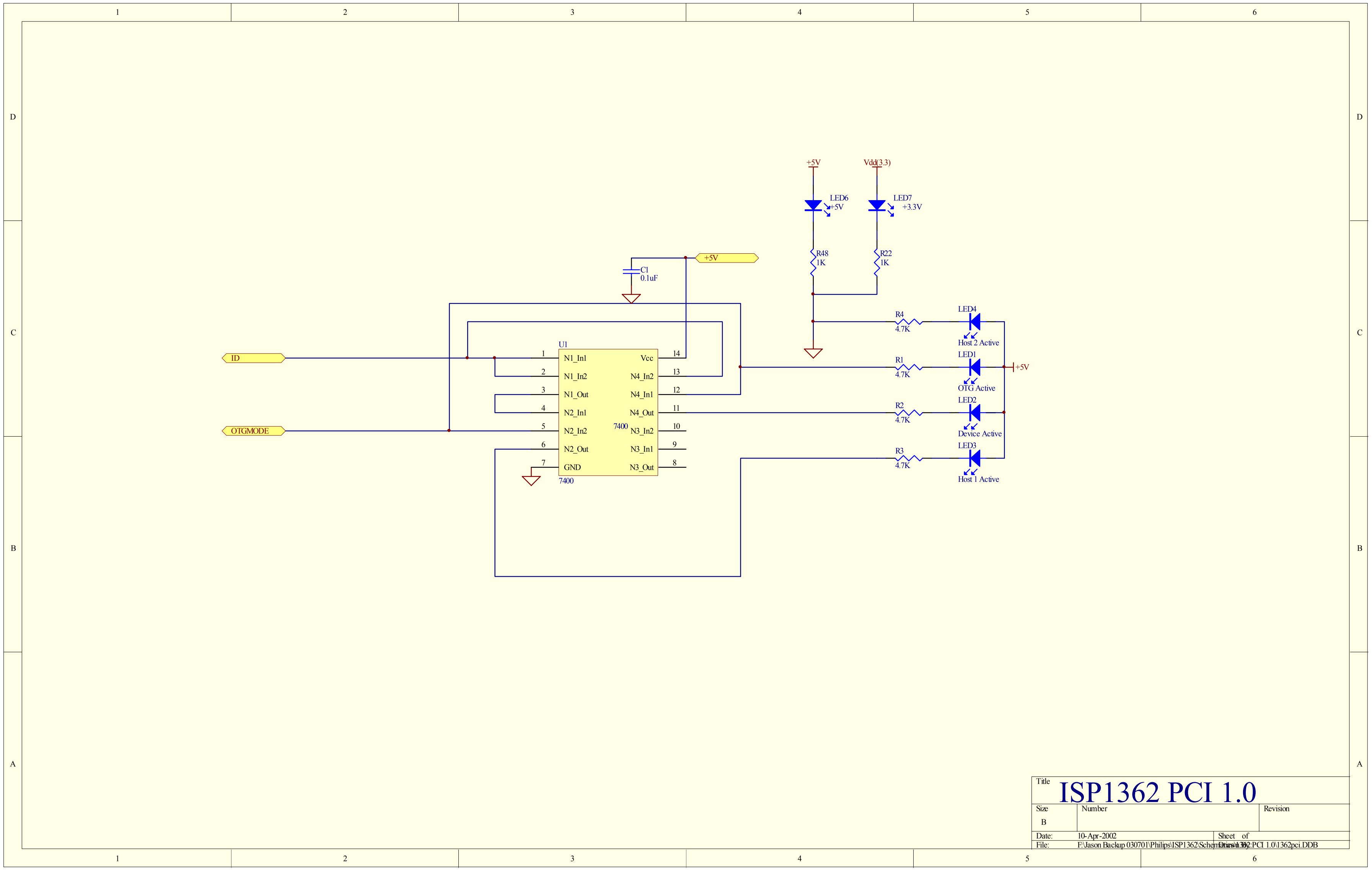


Figure D-4: Schematic of Active Port Indicators

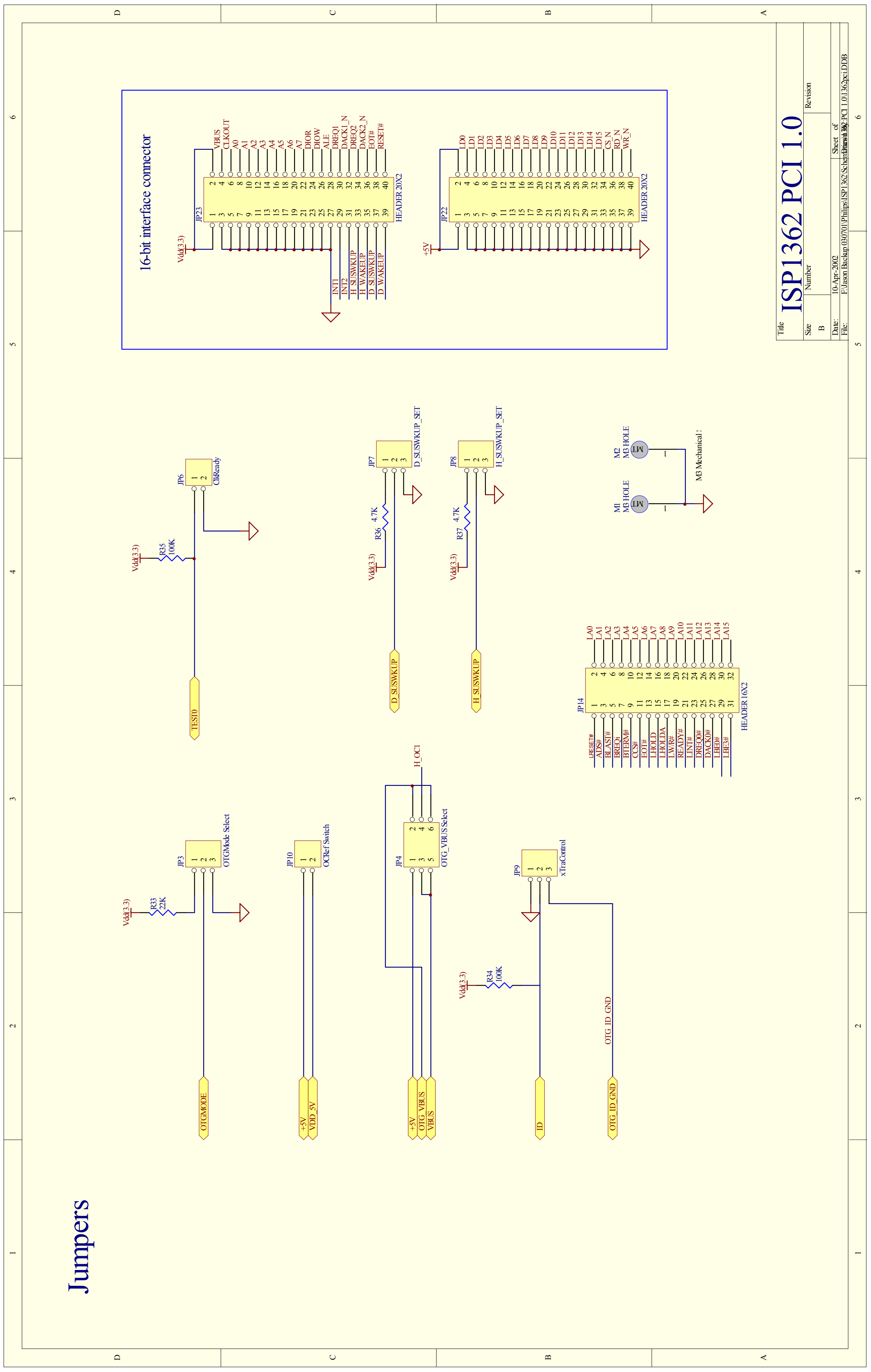


Figure D-5: Schematic of Jumpers

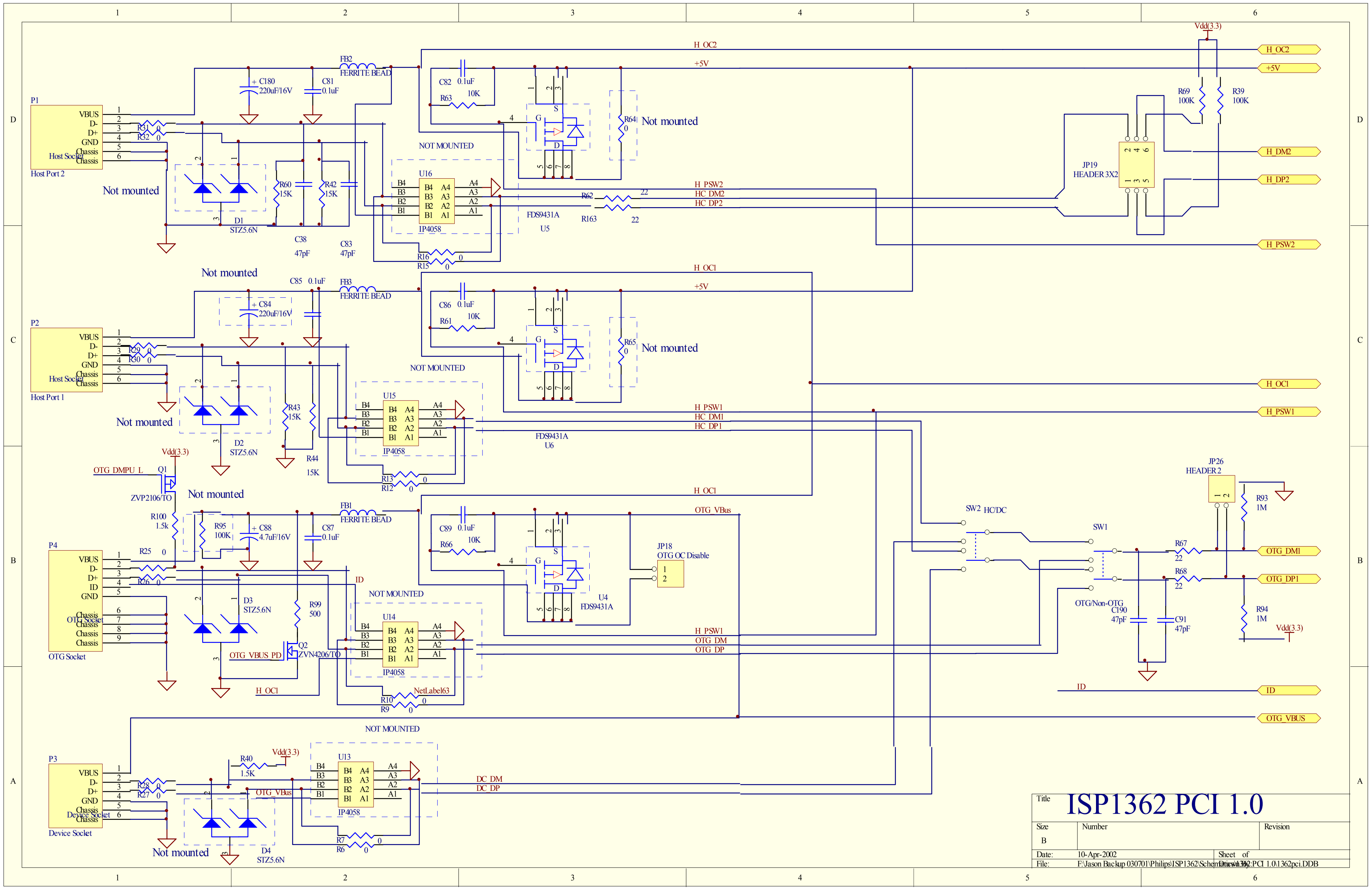


Figure D-6: Schematic of USB Ports

Philips Semiconductors

Philips Semiconductors is a worldwide company with over 100 sales offices in more than 50 countries. For a complete up-to-date list of our sales offices please e-mail
sales.addresses@www.semiconductors.philips.com.
A complete list will be sent to you automatically.
You can also visit our website
http://www.semiconductors.philips.com/sales/

www.semiconductors.philips.com