Semiconductors

UM10006_2

ISP1161x PCI Eval Board

October 2003

User's Guide

Rev. 2.0

Revision History:

Revision	Date	Description	Author
2.0	October 2003	• Updated the CPLD code. • Upgraded to the latest template. • Changed all occurrences of ISP1161 to ISP1161x, where ISP1161x represents ISP1161A and ISP1161A1 and any future derivatives.	Alvin LIM and Kunzang DOLMA
1.0	April, 2002	First release	Jason ONG and Alvin LIM

Note: ISP1161x denotes any Philips USB single-chip Host Controller and Device Controller whose name starts with 'ISP1161', this includes ISP1161A, ISP1161A1 and any future derivatives.

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

Note: ISP1161x denotes any Philips USB single-chip Host Controller and Device Controller whose name starts with 'ISP1161', this includes ISP1161A, ISP1161A1 and any future derivatives.

1. Introduction

The ISP1161x is a single-chip Universal Serial Bus (USB) Host Controller (HC) and Device Controller (DC) that comes in a small LQFP 64-pin package. This helps to save precious printed circuit board (PCB) space. The Host Controller portion of the ISP1161x 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 ISP1161x also complies with *Universal Serial Bus Specification Rev. 2.0*, supporting data transfer at full-speed (12 Mbit/s).

The target applications of the ISP1161x 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, interrupt request signals, DMA request and acknowledge signals for the Host Controller and the Device Controller. This makes it possible for the microprocessor to simultaneously control both the USB controllers. This advantage allows greater flexibility in systems with the ISP1161x built-in. For example, the ISP1161x system can be connected to a personal computer (PC) or a USB hub that has a USB downstream port, and at the same time it can be connected to a peripheral that has a USB upstream port, such as USB printer, camera, keyboard or mouse. You can also connect several ISP1161x systems together—this is point-to-point USB.

The ISP1161x has two built-in downstream ports for the Host Controller, and one upstream port for the Device Controller. Each downstream port has overcurrent detection functions to monitor the current loading of the downstream port. In the event of abnormal current conditions, the ISP1611x has power switch controls to automatically disable the downstream port's current on the USB V_{BUS}. You may ignore this function if you do not have such demands in your design or you have made your own design provisions.

The ISP1161x PCI board is a stand-alone PC evaluation (eval) kit and is the successor of the ISP1161x ISA board. The ISP1161x 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 and therefore, the motherboard resources are automatically allocated 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 ISP1161x output levels to its limits as required by various users.

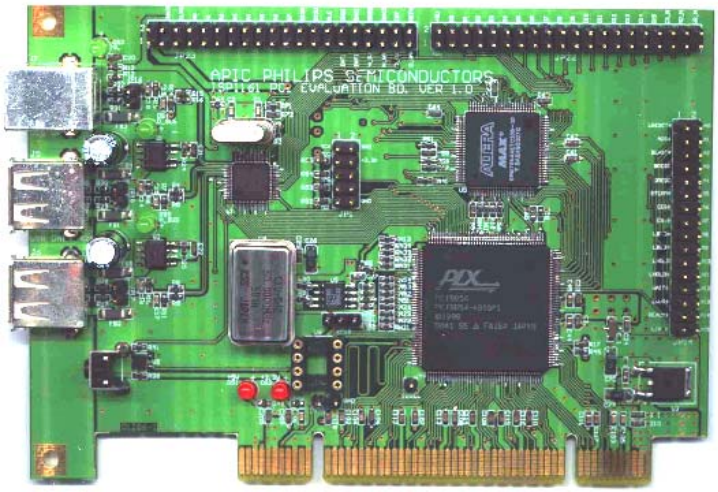

Figure 1-1: ISP1161x PCI eval board

2. Block diagram of the PCI eval board

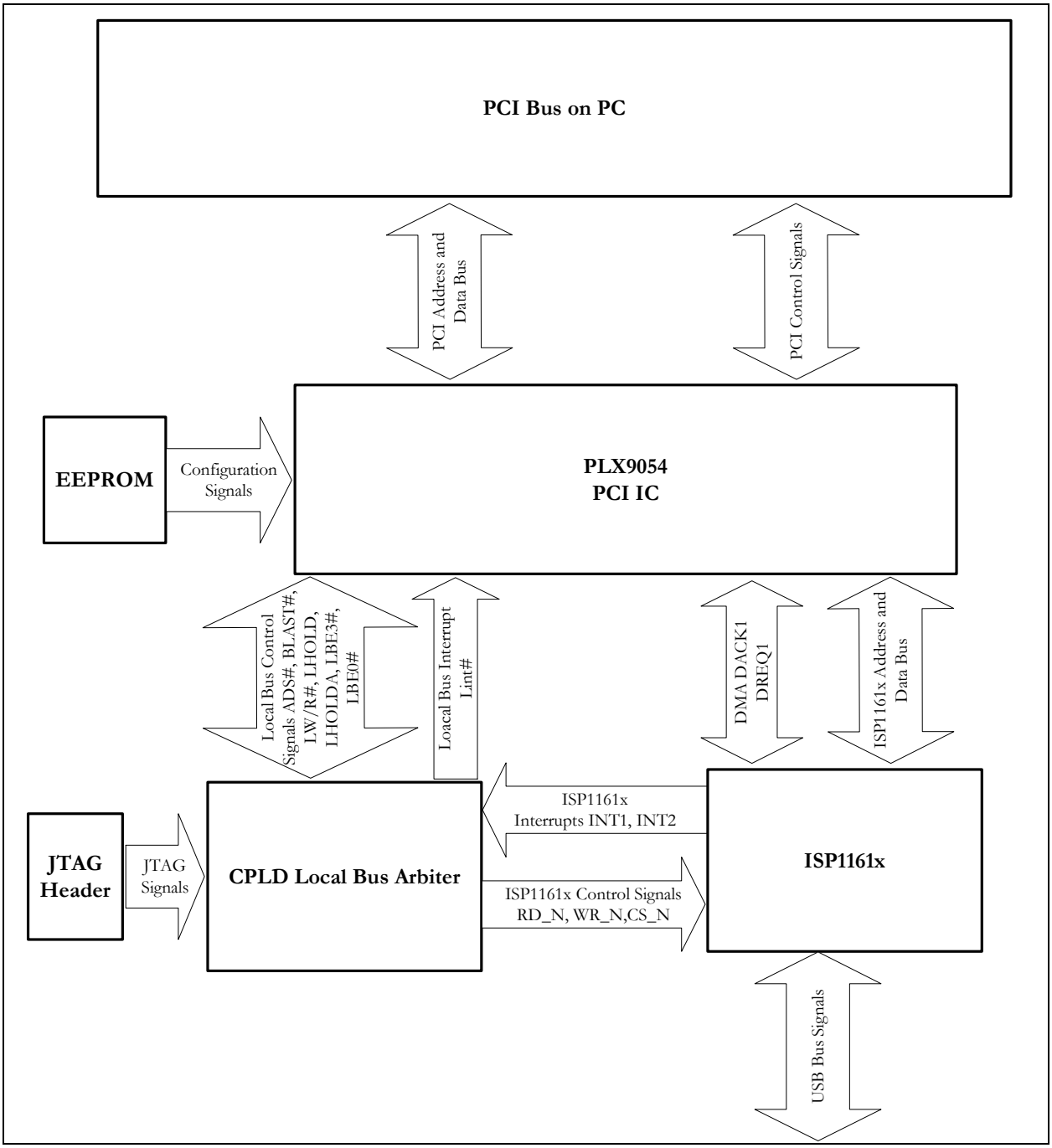

Figure 2-1: Block diagram

3. System requirements

3.1. System requirements for the Host Controller

- PC motherboard with PCI slot
- Microsoft® Windows® 98.

For specific information, refer to the respective ISP1161x software manual.

3.2. System requirements for the Device Controller

For host PC:

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

For device PC:

- PC with Microsoft DOS 6.x and a PCI slot
- ISP1161x PCI eval board.

For firmware development:

- X86 CPU platform: Borland Turbo C++ 3.0 or later
- ISP1161x eval CD.

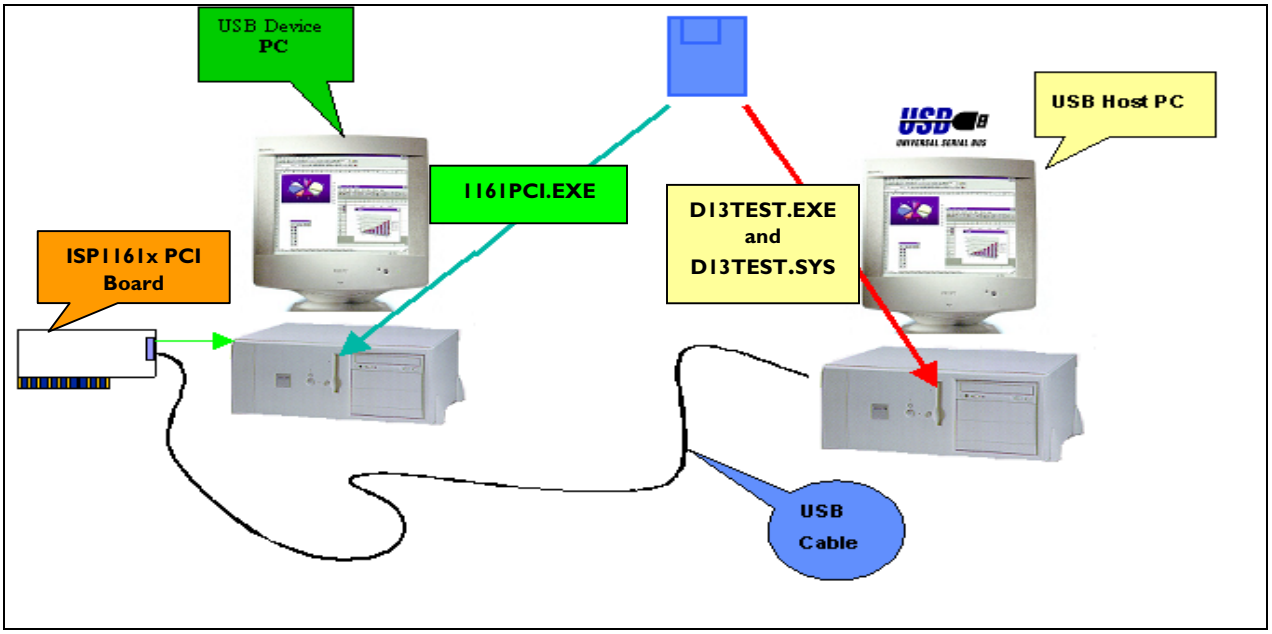

Figure 3-1: Setup environment for the Device Controller

4. ISP1161x PCI eval board

4.1. Installing the PCI eval board

The procedure to install the ISP1161x PCI eval board on a PC is as follows:

1. Plug the PCI in the PC PCI slot and boot the PC to the Microsoft Windows 98 operating system.
2. 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.
3. Reboot the PC.

The ISP1161x PC eval board is now ready for testing.

4.2. PC resources assignment

As the ISP1161x 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 ISP1161x PCI board.

Figure 4-1 shows the PC resources for the PCI bridge.

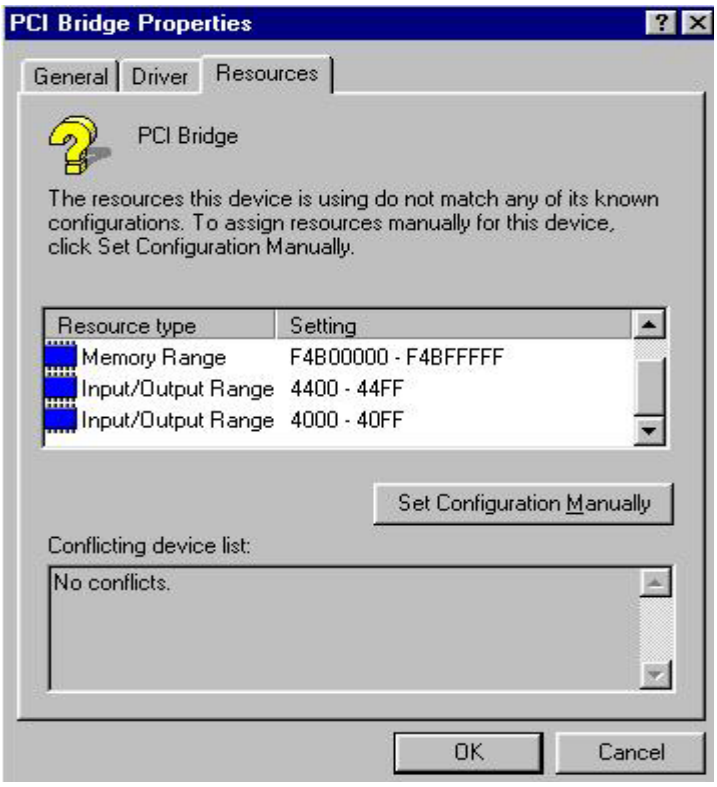

Figure 4-1: PC resources for the PCI bridge

4.3. Power supply and LED indicators

In the ISP1161x 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 board from the +5.0 V supply.

There are LEDs on the board to indicate the power supply status. DS1 is the +5.0 V indicator, and DS2 is the +3.3 V indicator. LED DS3 is the GoodLink™ indicator for the Device Controller. DS4 and DS5 are the indicators for D_SUSPEND and H_SUSPEND, respectively. When the Device Controller or the Host Controller is in the suspend mode, DS4 or DS5 will be lit up, respectively.

4.4. Headers and connectors

There are two 20-bit interface headers, JP22 and JP23, and one 16-bit interface header, JP14. The 20-bit headers contain the necessary address and data bus, as well as the control and output signals of the ISP1161x. The 16-bit header serves as a convenient taping point to debug the local bus signals on the PCB 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, J5 and J6, and one USB upstream port connector, J7, to interface to other USB peripherals.

There is also a reset switch, S1, for the hardware reset of the PLD and the ISP1161x.

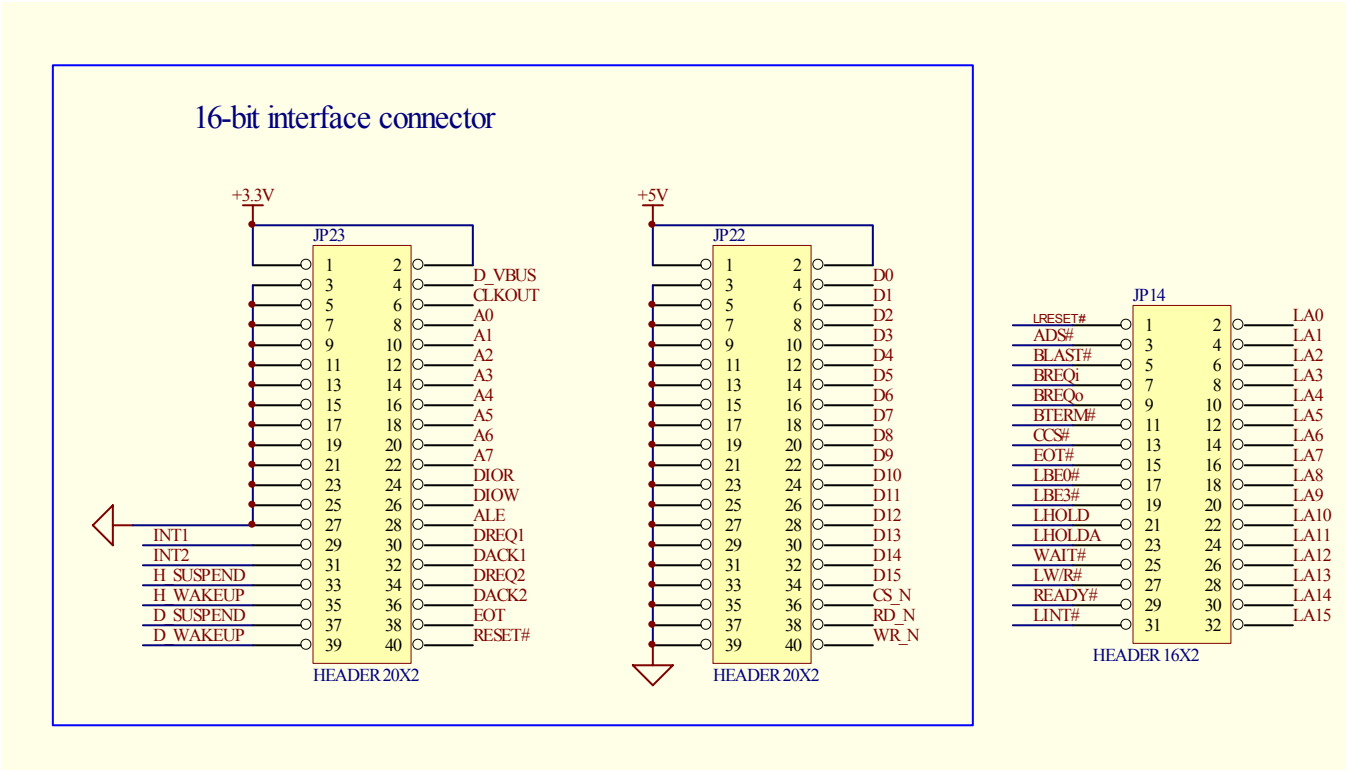

Figure 4-2: Interface headers

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

1. Switch off the device PC.
2. Remove all the unnecessary cards from the device PC.
3. Plug ISP1161x PCI board in the PCI slot of the device PC.

4. Switch on the device PC.
5. Run the firmware, 1161FW.EXE, on the device PC under the DOS mode.
6. Plug the USB cable between the ISP1161x board and the USB host PC.

If it is the first time that the eval board is connected to the host PC, the host OS Device Manager will prompt for installation of the INF and the driver.

7. Select the location of D13Test.INF and D13Test.SYS from the ISP1161x eval CD and complete the installation procedure.

Figure 4-3 shows a successful installation of the Philips Device Controller driver.

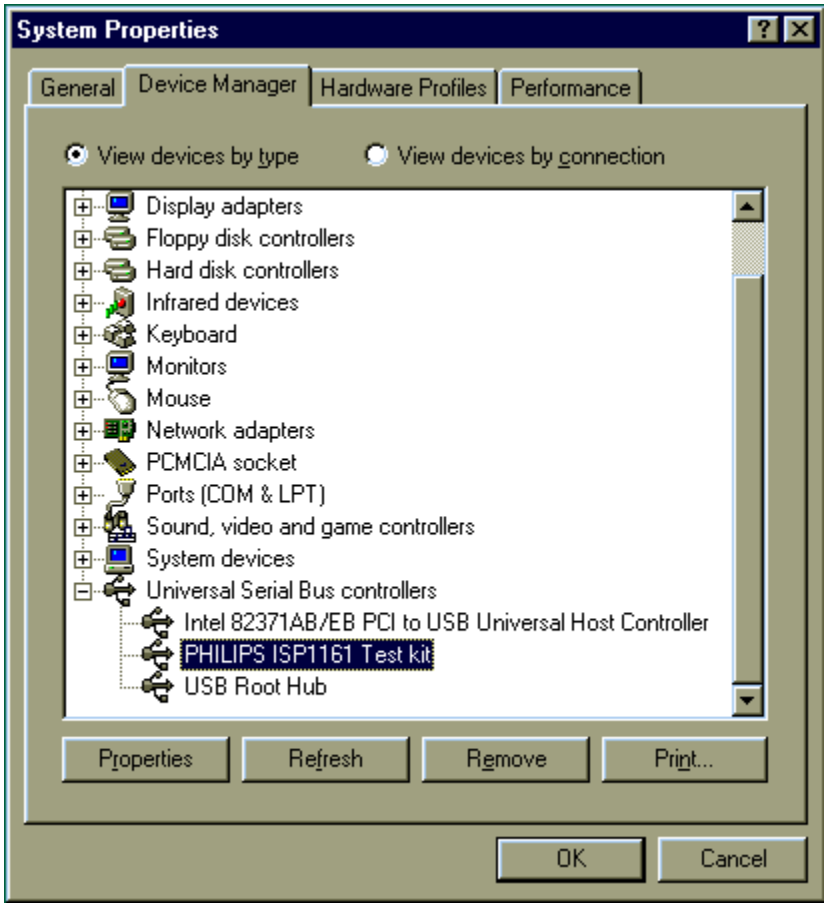

Figure 4-3: Philips Device Controller driver

4.6. Using the Device Controller host applet

The test applet, D13Test.EXE, exercises all the ISP1161x endpoints as shown in Figure 4-4.

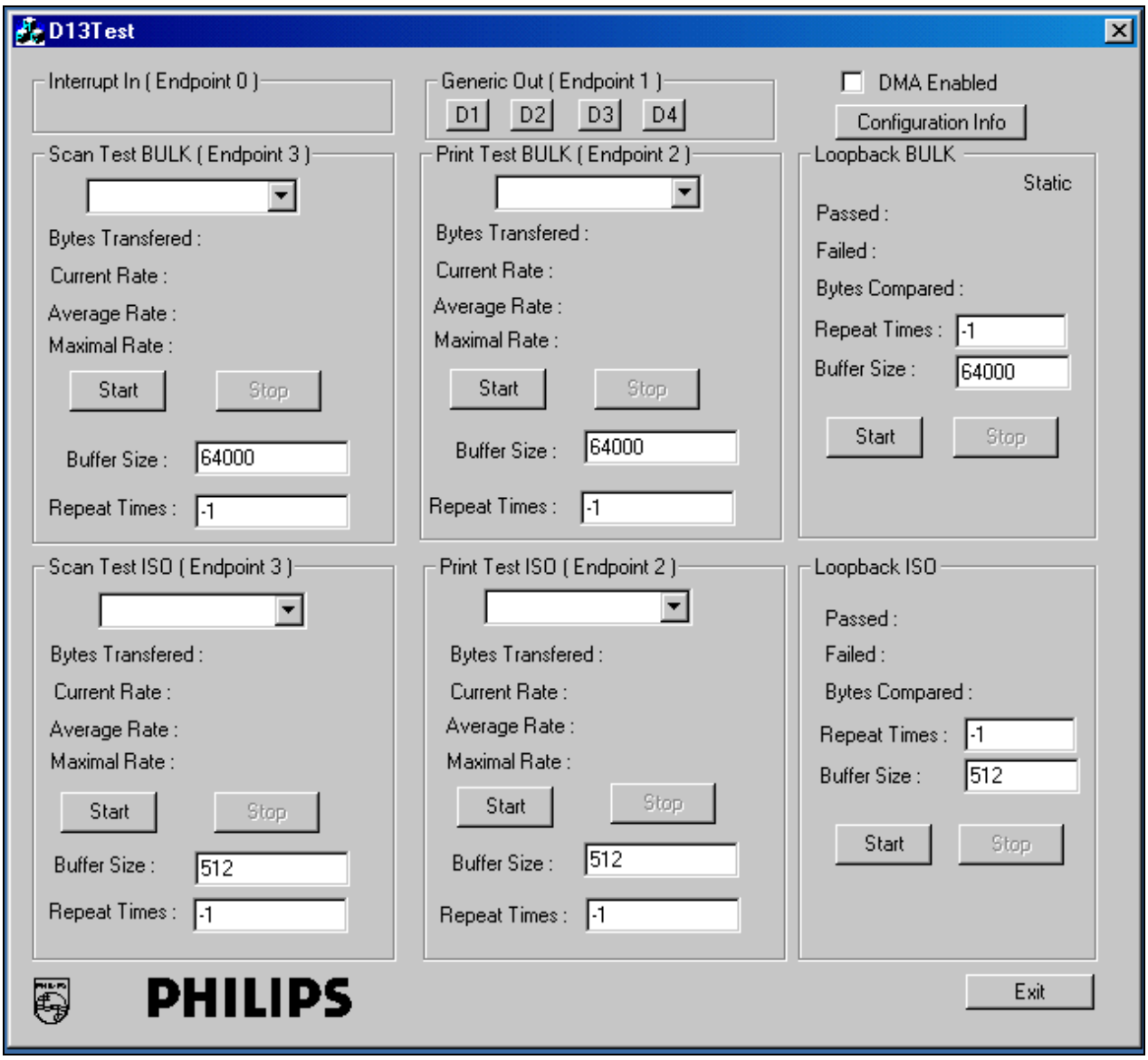

Figure 4-4: Applet: D13Test

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 will appear, as given in Figure 4-5.

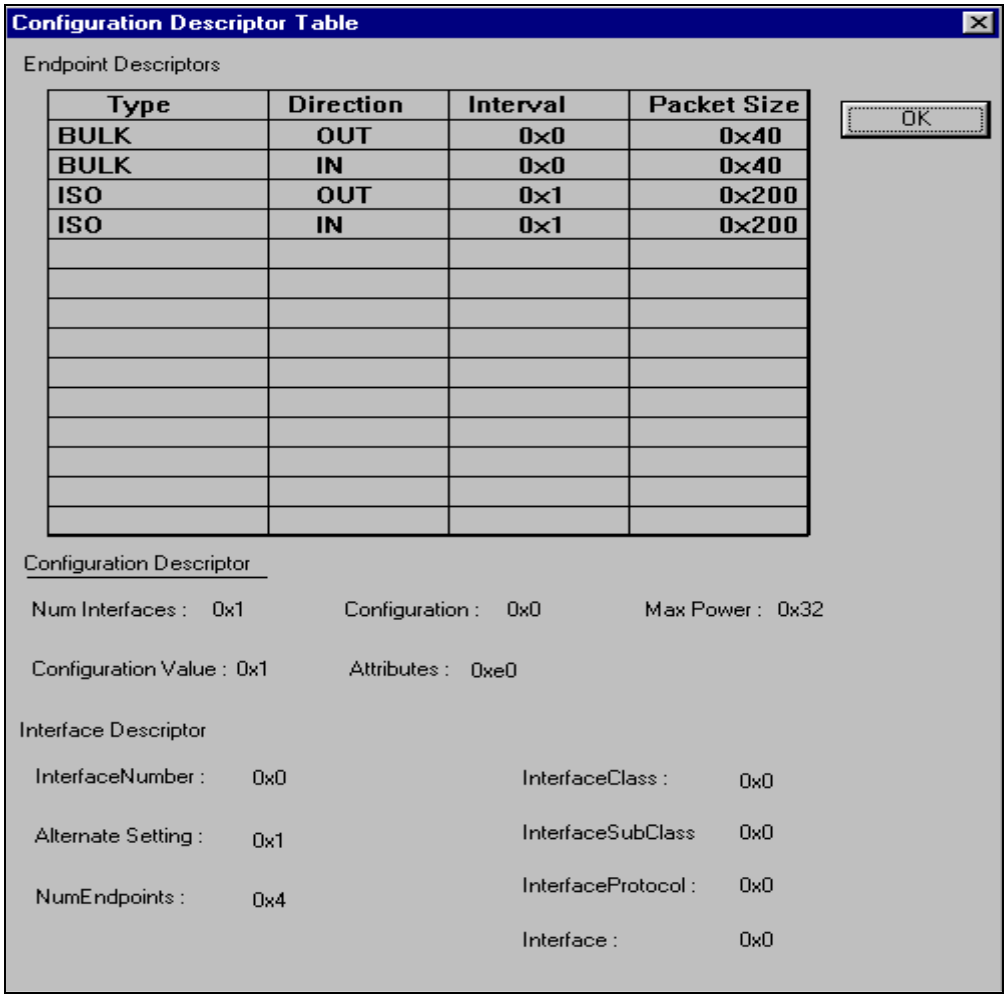

Figure 4-5: Configuration Description Table

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.

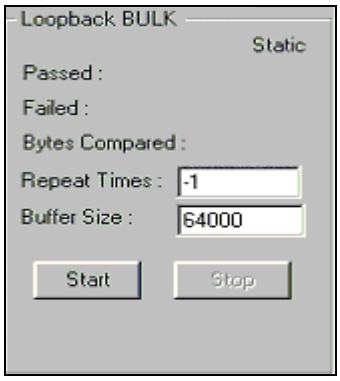

Figure 4-6: Loopback bulk

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 you that would like to have in the **Repeat Times** field. The buffer size for all ISO tests must be entered 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.

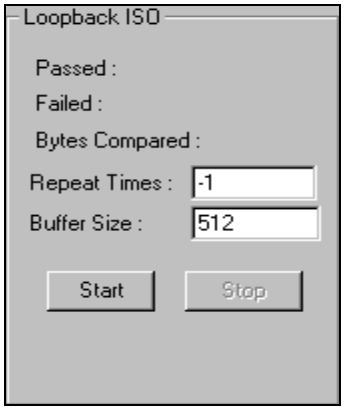

Figure 4-7: Loopback ISO

Table 4-1 shows the description of endpoints operations on the ISP1161x eval board.

Table 4-1: Description of endpoints operations

The test applet and the ISP1161x 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 ISP1161x 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 ISP1161x 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 ISP1161x 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 bulk DMA transfer and bulk PIO transfer. The ISP1161x 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 ISO transfer, the transfer size is predetermined as 512 bytes.

5. References

- *ISP1161A Full-speed Universal Serial Bus single-chip host and device controller* data sheet
- *ISP1161A1 Full-speed Universal Serial Bus single-chip host and device controller* data sheet
- *Universal Serial Bus Specification Rev. 2.0*
- *ISP116x PCI/DOS Mini Evaluation Kit User's Guide*
- *PLX Technology PCI9054 Data Book.*

Appendix A. ISP1161x PCI eval board PLD codes

```
--*************************************************************************************************--
--** The PLX9054 project is a design which implements glue logic between PLX9054 and ISP1161 **--
--**                                                                                             **--
--** Function:                                                                                   **--
--**    1. Only implement parallel I/O write/read, not DMA operation in version 1.0              **--
--**    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 ISP1161 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. WAIT# and READY# signals are used to control interconnect timings                     **--
--**                                                                                             **--
--**    Version 1.0           1st Release                                                       **--
--**    Version 1.1           State Machine Update from ISP1362 PCI v1.1                        **--
--*************************************************************************************************--

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
      wait_n       : out   std_logic;                     -- PLX data wait
      bterm_n      : out   std_logic;
      breqi        : out   std_logic;
      breqo        : out   std_logic;
      ccs_n        : out   std_logic;
      lint_n       : out   std_logic;
      laddr        : in    std_logic_vector(2 downto 0);  -- PLX address
      data         : in    std_logic_vector(15 downto 0); -- PLX data bus
      data_tmp     : out   std_logic;
      --** ISP1161 interface
      cs_n         : out  std_logic;                      -- ISP1161 chip select
      rd_n         : out  std_logic;                      -- ISP1161 read
      wr_n         : out  std_logic;                      -- ISP1161 write
      dack1_n      : out  std_logic;                      -- ISP1161 HC Dack1
      dack2_n      : out  std_logic;                      -- ISP1161 DC Dack2
      eot_n        : out  std_logic;                      -- ISP1161 EOT#
      a_sel        : out  std_logic_vector(1 downto 0);   -- ISP1161 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)     -- ISP1161 interrupt
      );
end plx9054;

architecture rtl of plx9054 is

       signal a0_delay                    : std_logic;
       signal a1_delay                    : std_logic;
       signal disable_PCI_int             : std_logic;

begin

-- ISP1161 Interrupt process
   interrupt_process: process(reset_n, clk)

   begin

   if reset_n = '0' then
        lint_n         <= '1';
               a_sel(0)       <= '0';
               a_sel(1)       <= '0';
```

```
   elsif (clk'event and clk = '1') then
   -- synchronize ISP1161 interrupt
        a0_delay    <= laddr(0);             -- delay 1 clk cycle for addr 0
        a1_delay       <= laddr(1);          -- delay 1 clk cycle for addr 1

                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';
                end if;

                if ( ( int(1) = '0' or int(2) = '0' ) and disable_PCI_int = '0' ) then
                        lint_n  <= '0';
                else
                        lint_n  <= '1';
                end if;
        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);
   variable state  :  sm_state;

   begin
   if reset_n = '0' then
          state := plx_start;
          data_tmp <= data(0) or data(1) or data(2) or data(3) or data(4) or data(11) or
            data(5) or data(6) or data(7) or data(8) or data(9) or data(10) or
            data(12) or data(13) or data(14) or data(15);
          breqo       <= '0';
      breqi       <= '0';
      bterm_n     <= '1';
      ccs_n       <= '1';
      wait_n      <= '1';
          dack1_n     <= '1';
          dack2_n     <= '1';
          eot_n       <= '1';
      ready_n      <= '1';
      cs_n         <= '1';
      rd_n         <= '1';
      wr_n         <= '1';
   elsif (clk'event and clk = '1') then

      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') 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;

           when cmd_blst =>
                     ready_n <= '0';
```

```
            if (laddr(2)='1' and laddr(1)='0') then
                        cs_n    <= '1';
                  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;

                when cmd_wait =>
                   ready_n <= '0';

                        if (laddr(2)='1' and laddr(1)='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. ISP1161x PCI eval board BOM

Part type	Quantity	Designator
PCB ISP1161 Rev. 1.0	1	—
Insulation Pad f/HC-49 Crystal	1	XTAL1
Capacitor SMD 0603 18pF / 50V 5%	2	C1, C2
Capacitor SMD 0603 47pF / 50V 5%	6	C14, C15, C16, C17, C18, C19
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 47nF / 25V 10%	5	C101, C102, C103, C104, C105
Capacitor SMD 0603 100nF / 50V 20%	35	C3, C4, C7, C10, C11, C12, C20, C21, C22, C25, C28, C29, C30, C31, C32, C33, C34, C35, C36, 37, C39, C40, C58, C59, C60, C61, C62, C63, C64, C65, C71, C205, C209, C210, C211
Capacitor Ele GPS-R 220uF / 16V	2	C13, C23
Capacitor Tan SMD-A 10uF / 10V 20%	3	C26, C70, C72
Socket IC Turn Pin 8-Way	1	U6
Header Pin 0.100" 1-Way Gold	3	G1, JP11, JP12
Header Pin 0.100" 1x2-Way Gold	3	JP16, JP17, JP18
Header Pin 0.100" 1x3-Way Gold	1	JP15
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
USB Type B RA 4-Way 61729-0010B	1	J7
USB Type A RA 4-Way 87520-0010B	2	J5, JP6
LED 3mm Green Diffused	3	DS3, DS4, DS5
LED 3mm Red Diffused	2	DS1, DS2
Fuse SMD NFM40P12C223	1	CF1
Resistor SMD 0603 1/16W 5% 0R	10	R12, R27, R28, R29, R30, R31, R32, R62, R63, R64
Resistor SMD 0603 1/16W 1% 22R	9	R4, R5, R8, R9, R15, R16, R81, R82, R86
Resistor SMD 0603 1/16W 5% 510R	1	R18
Resistor SMD 0603 1/16W 1% 1K	9	R22, R23, R48, R52, R53, R54, R55, R84, R85
Resistor SMD 0603 1/16W 1% 1K5	1	R14

Part type	Quantity	Designator
Resistor SMD 0603 1/16W 1% 3K9	1	R74
Resistor SMD 0603 1/16W 1% 10K	43	R3, R17, R20, R21, R24, R25, R38, R41, R45, R47, R51, R56, R57, R58, R59, R71, R72, R88, R89, R90, R91, R97, R98, 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	R6, R7, R10, R11
Resistor SMD 0603 1/16W 5% 1M	2	R13, R19
Switch Tact 6mm Right Angle B: 3.85 mm	1	S1
Inductor Solid SMD-4516 BLM41A600SPT	5	FB1, FB2, FB3, FB4, FB5
IC FM93CS56N	1	U6
IC LM1117DT33	1	U3
IC EPM7064AETC100-10	1	U9
IC MAX6306UK30D2	1	U8
Crystal 6.0000 MHz HC-49/U	1	XTAL1
Oscillator 50.000MHz	1	U4
Diode STZ5.6N	2	D3, D4
Transistor	2	Q1, Q2
IC CY2305SC-1	1	U5
IC ISP1161	1	U1
IC PC19054	1	U2

Appendix C. ISP1161x 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
```

Appendix D. ISP1161x PCI eval board schematics

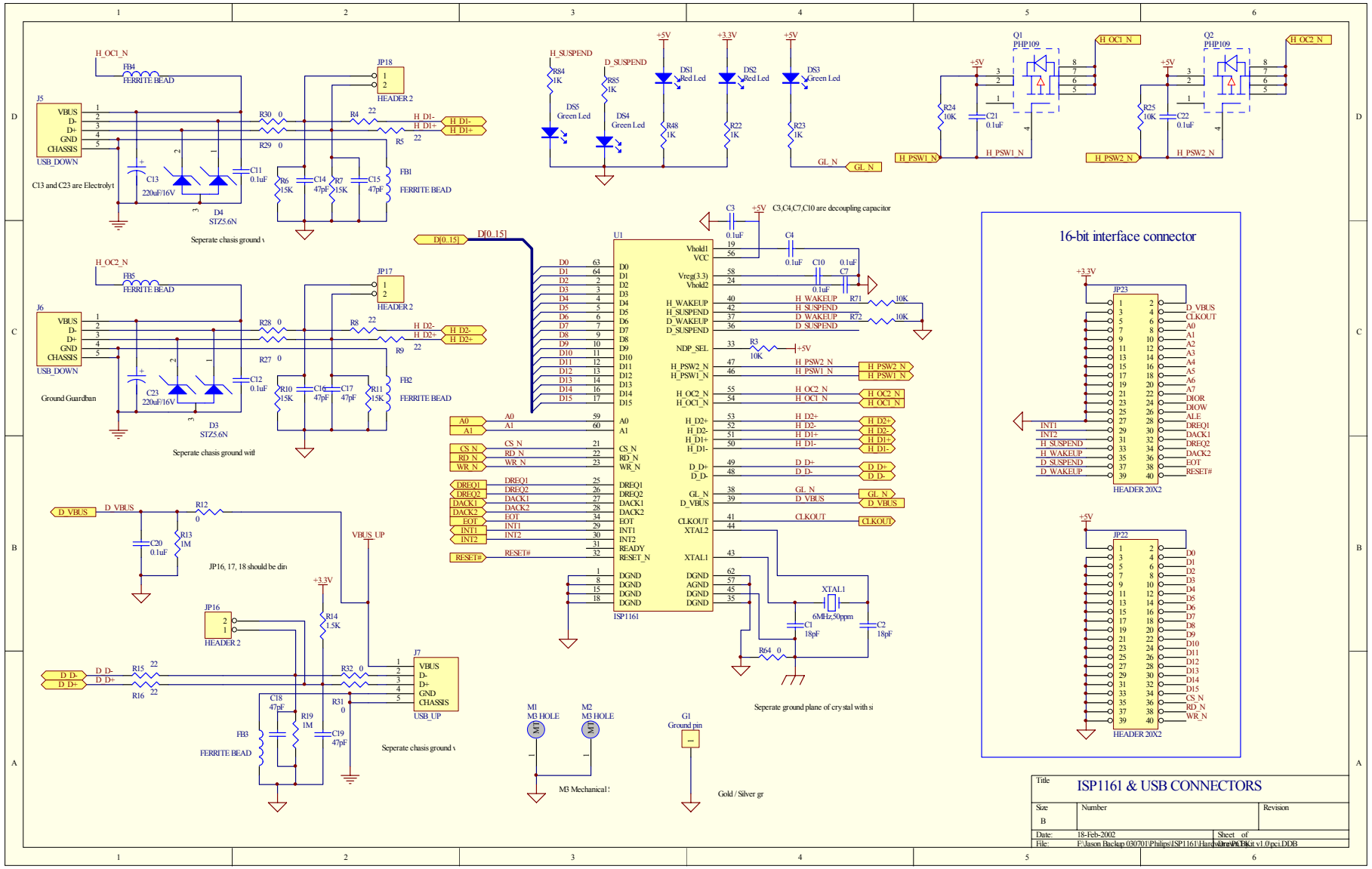

Figure D-1: ISP1161x schematic

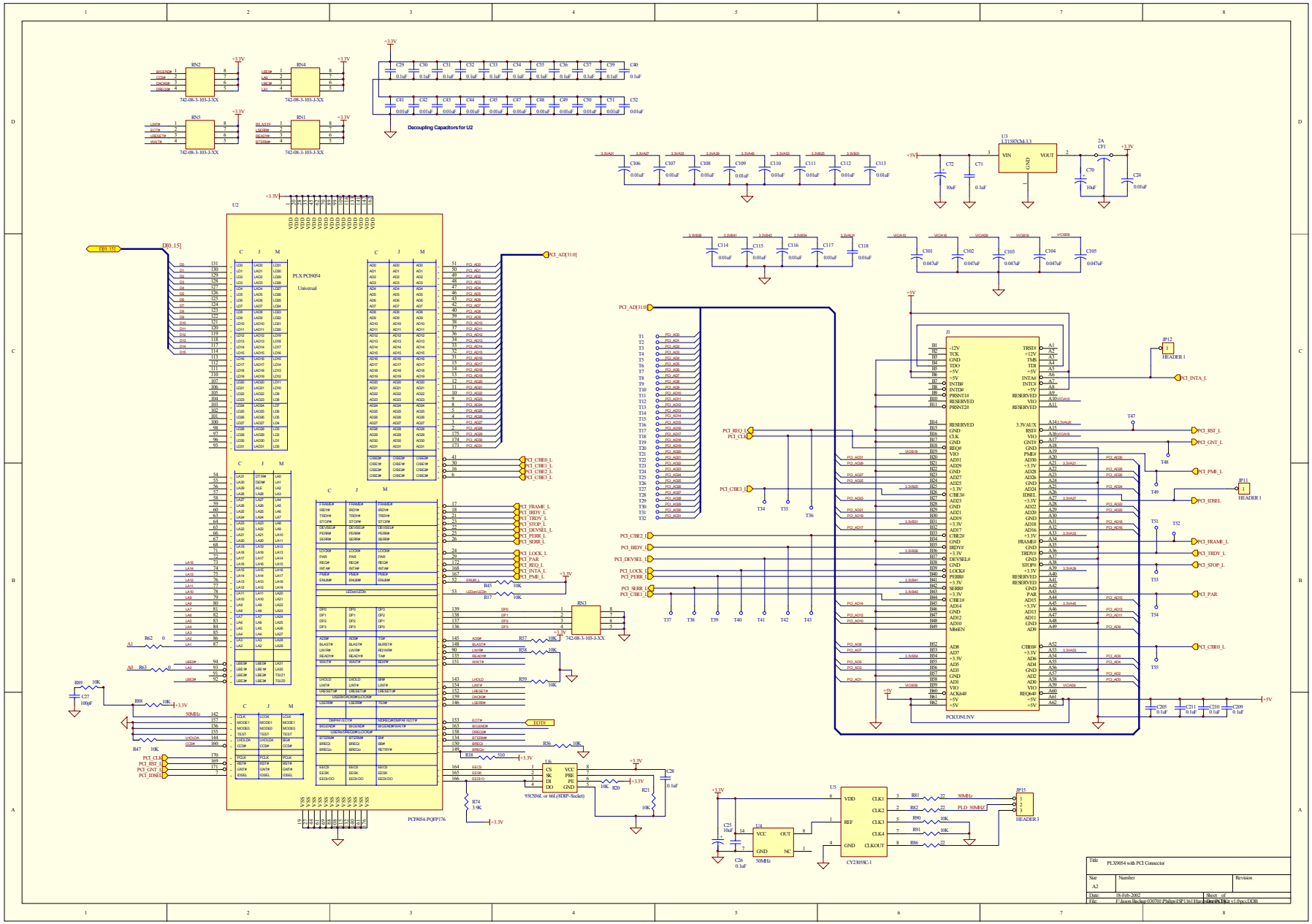

Figure D-2: Schematic of the PCI bridge

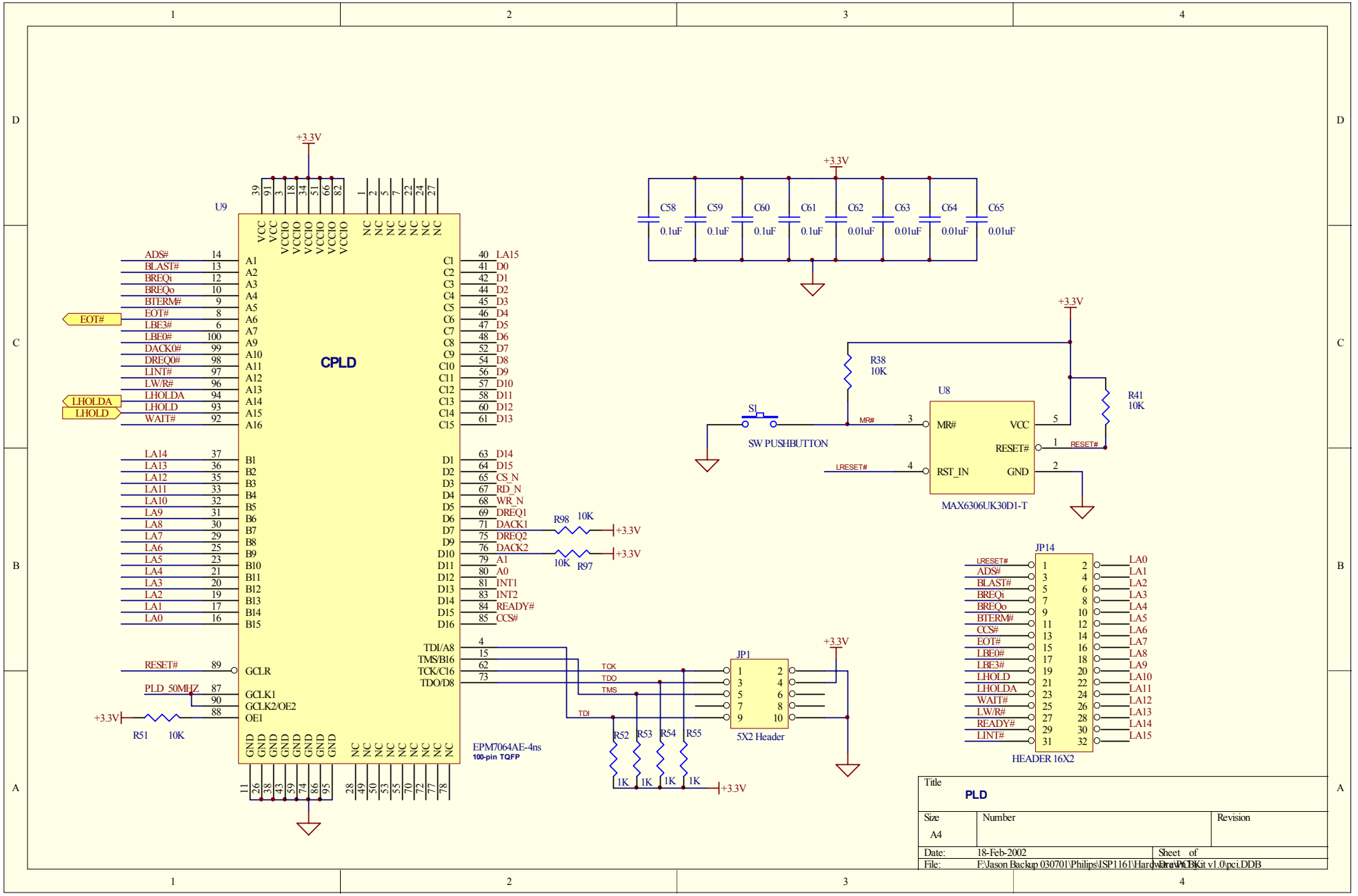

Figure D-3: Schematic of the PLD

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

PHILIPS