Semiconductors

AN10021_3

Interfacing ISP1161x to Intel® StrongARM® SA1110 Processor

January 2003

Application Note

Rev. 3.3

Revision History:

Rev.	Date	Descriptions	Author
3.3	Dec 2002	Updated the schematic and Figure 4-1.	Jason Ong
3.2	Sep 24, 2002	Updated the template	Kunzang Dolma
3.1	July 17, 2002	Updated the following: • Schematic • Template • Changed the file name from INTFG_1161_TO_STRONGARM-03.pdf to AN10021-01.pdf	Jason Ong, Seow Aun Kie
3.0c	Oct 8, 2001	Updated schematic	Jason Ong
3.0b	Aug 2001	Changed to Philips template	Yuk-lin Ong
3.0a	June 7, 2001	Revised chapters and added Appendix A	Jason Ong
2.0	May 29, 2001	Updates after actual implementation	Ng Chee Yu
1.0	Aug 18, 2000	First draft	Socol Constantin

Note: ISP1161x denotes any Philips USB single-chip host 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

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

1. Overview

The unique design of the Philips ISP1161x allows it to be used both as a Host Controller (with two downstream facing ports) and a Device Controller (with one upstream facing port). These ports may be independently accessed, enabling simultaneous connection as a Host Controller and a Device Controller.

When ISP1161x is integrated into a personal digital assistant (PDA) or handheld personal computer (HPC), it is usually connected to the external bus interface of a Reduced Instruction Set Computer (RISC) processor. This application note presents the critical issues in the ISP1161x embedded design, using the Intel StrongARM SA1110 processor as a concrete example.

The actual implementation of SA1110-ISP1161x uses an Intel StrongARM-1110 Hardware Development Platform, a Philips ISP1161x evaluation kit, and a Philips ISP1161x Bridging Board for SA1110.

2. ISP1161x Interface Signals to a RISC Processor Bus

The processor bus interface of the ISP1161x is designed for a simple direct connection with a RISC processor. The data transfer can be done in the Programmed I/O (PIO) or direct memory access (DMA) mode. The estimated maximum data transfer rate on the generic processor bus of the ISP1161x is approximately 15 Mbyte/s. This is based on an ISP1161x internal clock frequency value of 48 MHz. To achieve the maximum data transfer rate on the host processor bus, the ISP1161x contains ping pong structured RAM that allows alternative access from the RISC processor or from the internal Host Controller and the Device Controller. The ping pong memory is allocated separately for the Host Controller and the Device Controller. The Host Controller uses 2 kbytes of the ping memory and 2 kbytes of the pong memory in its allocated memory. The Device Controller uses 1.5 kbytes for each of the ping and pong memory in its own memory.

The main ISP1161x signals you should take into consideration for connecting to a StrongARM SA1110 processor are:

- A 16-bit data bus (D[15:0]) for ISP1161x, which is "little endian" compatible.
- The two address lines (A0 and A1) necessary for complete addressing of the ISP1161x internal registers:
  - **A0 = 0 and A1 = 0**—Selects the Data Port of the Host Controller
  - **A0 = 1 and A1 = 0**—Selects the Command Port of the Host Controller
  - **A0 = 0 and A1 = 1**—Selects the Data Port of the Device Controller
  - **A0 = 1 and A1 = 1**—Selects the Command Port of the Device Controller.
- One $\overline{\text{CS}}$ line used to select ISP1161x in a certain address range of the host system. This input signal is active LOW.
- $\overline{\text{RD}}$ and $\overline{\text{WR}}$ are common read and write signals. These signals are active LOW.
- Two interrupt lines:
  - INT1 (used by the Host Controller) and
  - INT2 (used by the Device Controller).

  Both have programmable level/edge and polarity (active HIGH or LOW).
- The $\overline{\text{RESET}}$ signal is active LOW.

3. Intel StrongARM SA1110 Processor

This section describes the main features of the Intel StrongARM SA1110 processor for connecting to ISP1161x.

The "Memory and PCMCIA Control Module" of the Intel StrongARM processor is responsible for generating all signals for interfacing with ISP1161x. The following features are useful for direct connection to ISP1161x:

- The "Memory Control Module" generates all the signals necessary to control different types of external devices:
  - DRAM (up to four banks of FPM, EDO and SDRAM)
  - Static memory (up to three banks of ROM, Flash, SRAM and SMAROM, selected by nCS0, nCS1 and nCS2 signals)
  - Static memory and variable latency I/O devices (up to three banks of ROM, Flash, SMROM and SRAM, such as variable latency I/O devices, selected by the nCS3, nCS4 and nCS5 signals).

Additional wait-states can be inserted by programming the internal registers of SA1110, if necessary.

- The data bus size of these memory areas can be set as 16-bit or 32-bit wide. ISP1161x uses the 16-bit wide data bus size.
- The memory access is defined as "little endian" or "big endian" types, according to the value of the "big endian bit" in the control register. By default, at power-on or after a reset, the SA1110 processor uses the "little endian" memory access scheme that corresponds to the ISP1161x requirement.

4. Considerations in Timing Diagrams and WAIT States

The following is a short study of the timing diagrams of the main bus cycles of ISP1161x and Intel SA1110:

The memory clock determines the timing diagram of the external bus cycle of the Intel StrongARM SA1110 processor, which is equal to two CPU clock cycles. Timing during $\overline{RD}/\overline{WR}$ accesses is determined by the settings of the MSC0, MSC1 and MSC2 registers that correspond to the chip select pairs nCS(5,4), nCS(3,2) and nCS(1,0), respectively. All timing fields are specified as numbers of memory clock cycles. Each register contains two identical configuration fields corresponding to each nCS within one of the pairs mentioned earlier. By programming the MSC0, MSC1 and MSC2 registers, you can modify the assert time of each beat of a burst RD/WR, the deassert time between each beat of a burst RD/WR, and the hold-off time after a write to subsequent accesses.

According to the ISP1161x datasheet specifications, a read operation requires the following timing parameters (the write operation is similar); see Figure 4-1:

- t_{RL} = 33 ns ($\overline{RD}$ LOW pulse width—minimal value required by ISP1161x)
- t_{RHRL} = 110 ns ($\overline{RD}$ HIGH to next $\overline{RD}$ LOW—minimal value required by ISP1161x)
- t_{RHDZ} = 3 ns ($\overline{RD}$ hold time, minimal value that can be expected from ISP1161x)
- t_{RC} = 143 ns (will result as a sum of t_{RL} and t_{RHRL})
- t_{SHSL} = 300 ns (first $\overline{RD}/\overline{WR}$ after command).

For a detailed analysis of a timing diagram, consider the access of an ISP1161x internal register (for example, the Control Register of the Host Controller). It requires two phases: writing the address (index) of the selected register into the Command Port; then only data transfer access (RD/WR) may take place.

Note: the index of each register is different, depending on whether it is a RD or a WR operation.

The timing diagram in Figure 4-1 describes the two phases of accessing ISP1161x:

- The first phase is accessing the Command (control) Port of ISP1161x to write the address (index) of the data port that will be accessed. In this phase, $\overline{CS}$ is active. The data lines D[15:0] contain the desired address. The $\overline{WR}$ pulse will be activated and will latch the data. Note the value of t_{SHSL} that represents the minimum time required between occurrence of the first phase and the second phase. As an example of the Host Controller "Control Register", a value of 01H will be transferred during a $\overline{RD}$ operation and 81H during a $\overline{WR}$ operation.
- The second phase consists of the access (read or write) to the data port selected by the address latched in the previous phase. Two timing diagrams are combined again in this second phase: one for the read access and one for write access. A series of $\overline{RD}$ and $\overline{WR}$ pulses are shown in the diagram to define the timing requirements between two consecutive accesses to ISP1161x: t_{RHRL}, t_{WHWL}, t_{RC}, t_{WC}, t_{RLDV}, as specified in the datasheet.

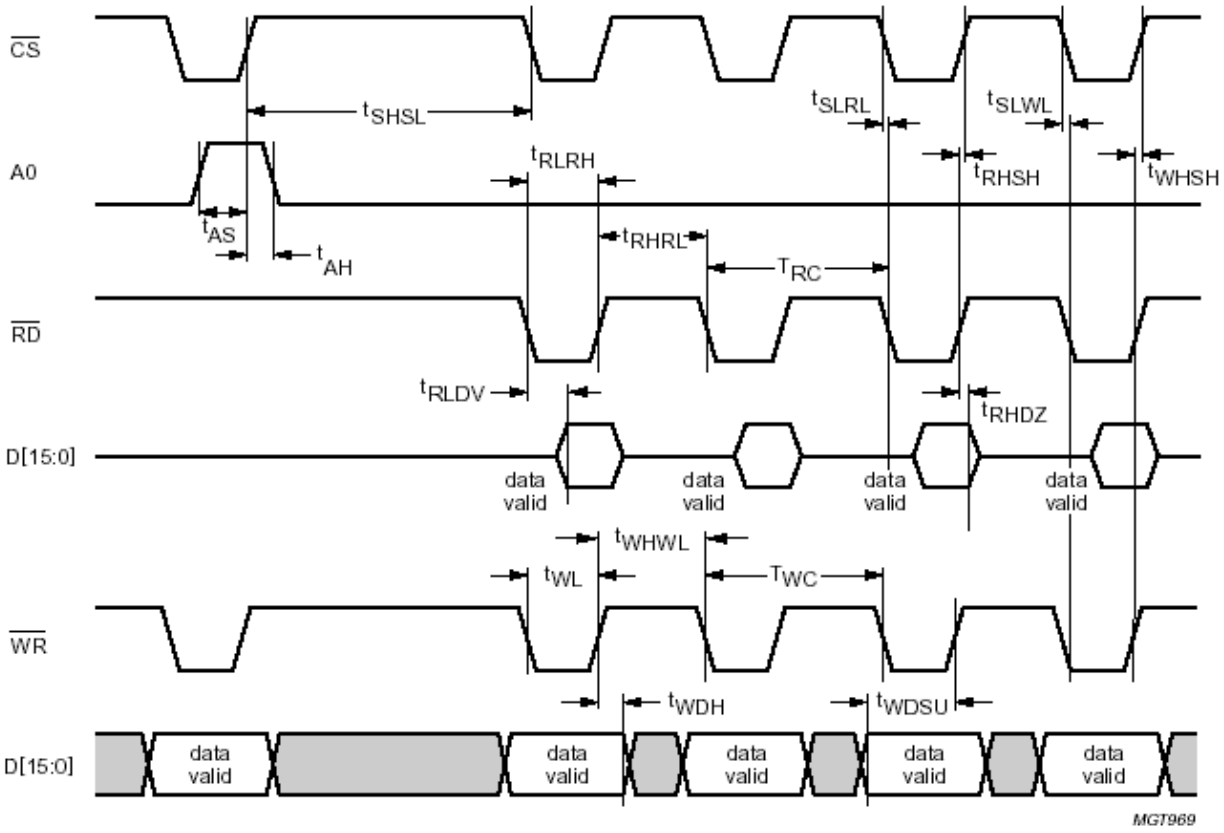


Figure 4-1: Programmed Interface Timing (16-bit Read/Write)

5. Using interrupts

ISP1161x generates two interrupts on the INT1 and INT2 pins, allocated for the Host Controller and the Device Controller, respectively. These interrupts occur depending on the setting of the interrupt registers.

Connect the ISP1161x interrupts—INT1 and INT2—to one of the 28 GPIO lines of the SA1110 processor. Each GPIO line of the SA1110 can be programmed to detect a rising or falling edge and generate an interrupt. The type of edge detection is programmed through the GPIO rising-edge detect register (GRER) and GPIO falling-edge detect register (GFER). The state of the edge detect can be determined by reading the edge-detect status register (GEDR).

Both INT1 and INT2 of ISP1161x are programmable as active on level or edge and HIGH or LOW, as specified in the *HcHardwareConfiguration Register and DcHardwareConfiguration Register* for each of the Host Controller and the Device Controller, separately. You must match the settings of the ISP1161x interrupt lines—INT1 and INT2—with the settings of the GRER and GFER registers of SA1110 used for programming the type of edge detection.

You can use ISP1161x's INT1 and INT2 output signals to wake up the host system's processor (in this case SA1110) from the idle or sleep mode. The GPIO pins are defined as category three signals and are actively sampled by SA1110 even during the sleep mode. The sleep mode of SA1110 offers the greatest power savings. In this mode, the internal sleep state machine of SA1110 is running off the 32.768 kHz crystal oscillator and watches for a preprogrammed event to occur, which will initiate the wake-up sequence. The V_{DDX} I/O voltage supply of SA1110 must be present during the sleep mode to enable this wake-up method.

6. Suspend and Resume

You can enable ISP1161x to enter the Reset, Resume, Operational and Suspend functional states by programming the *HcControl Register* of the Host Controller or the *DcMode Register* of the Device Controller.

Another way to wake up ISP1161x from the suspend mode is to use the input signals H_WAKEUP (for the Host Controller) and D_WAKEUP (for the Device controller). These signals may be connected to any available GP I/O lines of SA1110.

Monitoring the H_SUSPEND pin (for the host status) and the D_SUSPEND pin (for the device status) can determine the actual status of ISP1161x, without having to access internal status registers. Connecting these signals to any available GP I/O port of the SA1110 is an easy way to determine ISP1161x's status.

ISP1161x may wake up when its $\overline{CS}$ input signal becomes active, if this is desired, by programming a **1** in bit 3 of the *DcHardwareConfiguration Register* of the Device Controller. Alternatively, if the same bit is programmed to a value of **0**, asserting the $\overline{CS}$ signal does not cause ISP1161x to wake up.

7. Schematic

The schematic on the following page shows ISP1161x connected to a StrongARM SA1110 processor in a minimal hardware configuration.

In this example schematic, ISP1161x is simply selected by nCS5. To correctly access ISP1161x, it is assumed that the memory space selected by nCS5 is programmed for 16-bit access and "little endian".

Interrupts INT1 and INT2 are arbitrarily connected to IRQ2 and IRQ3 lines of SA1110.

The $\overline{\text{RESET}}$ input signal of ISP1161x is generated by the RESET_OUT# signal of the SA1110 processor, when its RESET_IN signal is active. The RESET_OUT# signal is also asserted for soft reset events (sleep and watchdog). Connecting the $\overline{\text{RESET}}$ input of the ISP1161x to RESET_IN of SA1110 may be a better solution if INT1 and INT2 are used to wake up SA1110.

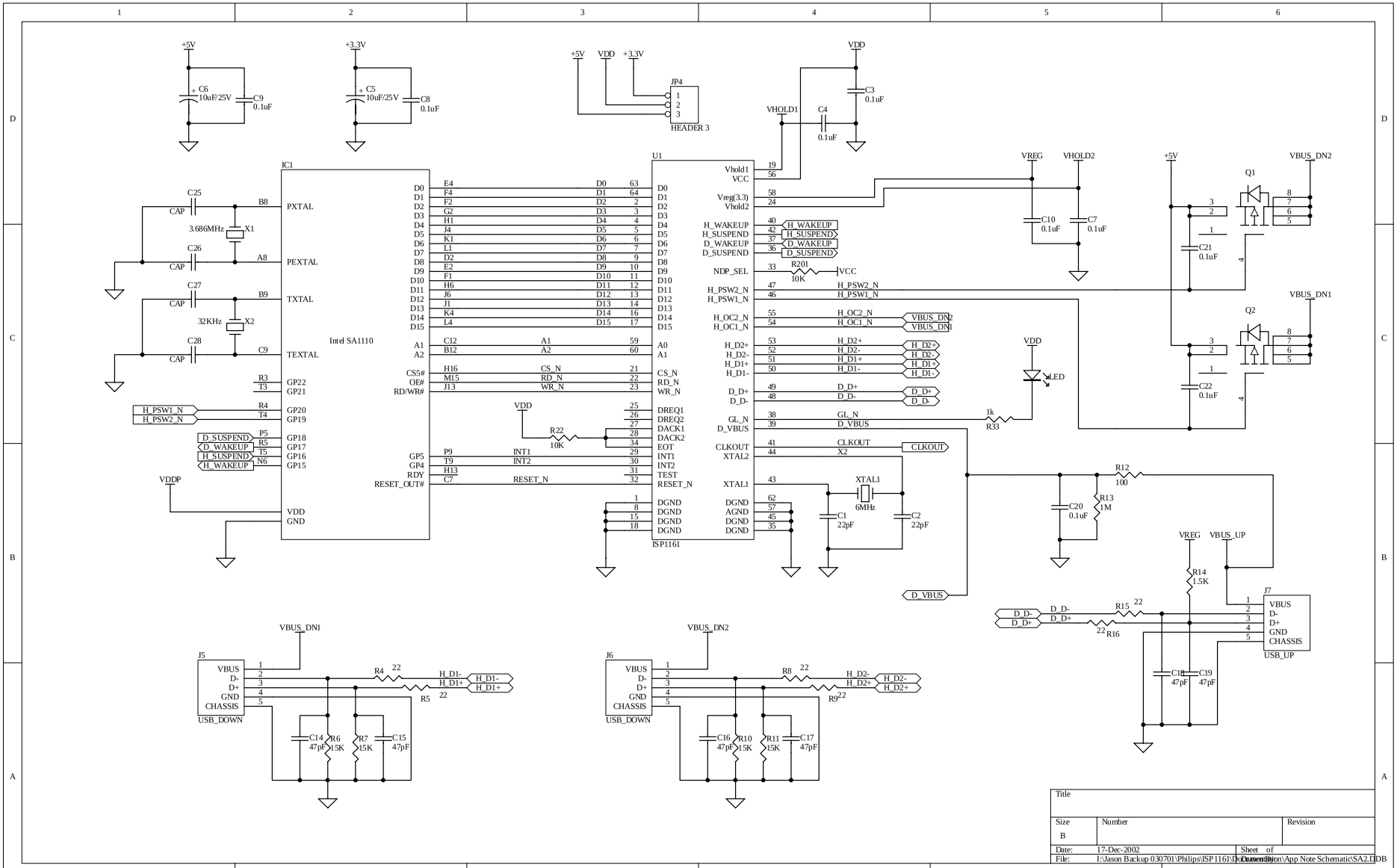
Pins H_PSW1 and H_PSW2 are connected to lines 4 and 5 of the same I/O port. This connection creates an alternative way to determine the power status of each downstream port.

Input signals $\overline{\text{H\_OC1}}$ and $\overline{\text{H\_OC2}}$ are used by ISP1161x to detect an overcurrent on the downstream ports. As separate overcurrent detection and protection circuits are implemented for each ISP1161x downstream port, detection of an overcurrent on a downstream port will have power turned off at that port only. Connecting the voltages of the two downstream ports VBUS_DN1 and VBUS_DN2 to $\overline{\text{H\_OC1}}$ and $\overline{\text{H\_OC2}}$ pins enables detection of the current value by sensing voltage drop on Q1 and Q2, where Q1 and Q2 are PMOS transistors with very low switch-on resistance Rds(on). Selection between Q1 and Q2 depends on the desired maximum current value and this determines the value of Rds(on). For example, if the allowed maximum current is approximately 0.5 A, a voltage drop of 75 mV will trigger the overcurrent circuitry and Rds(on) of approximately 150 MΩ will result. Connecting the ISP1161x input pins $\overline{\text{H\_OC1}}$ and $\overline{\text{H\_OC2}}$ to +5 V will disable the internal overcurrent protection of the ISP1161x. An external overcurrent protection circuit may also be used.

Selection of the number of downstream ports can be done in this configuration by programming the respective GPIO output of the SA1110 to a certain value. This will determine the desired LOW or HIGH level on ISP1161x's NDP_SEL input signal, and one or two downstream ports will be accordingly selected.

Detection of a connection on the upstream port is achieved by connecting VBUS_UP to pin D_VBUS of the ISP1161x. R12 and C20 will act as a low-pass filter that eliminates the possibility of sensing false voltage drop because of load current variations or noise on VBUS_UP. It is recommended, if possible, to implement a hybrid power solution, by using VBUS_UP to power the ISP1161x and an external power source for the rest of the system.

The $\overline{\text{GL}}$ output signal indicates, through an LED, the status of the USB device and helps in troubleshooting the USB connection.


+5V
+3.3V
VDD
C6 10uF/25V
C9 0.1uF
C5 10uF/25V
C8 0.1uF
JP4
HEADER 3
C3 0.1uF
C4 0.1uF
VHOLD1
IC1
Intel SA1110
PXTAL
PEXTAL
TXTAL
TEXTAL
3.686MHz
32KHz
X1
X2
C25
C26
C27
C28
CAP
B8
A8
B9
C9
R3
T3
GP22
GP21
R4
T4
GP20
GP19
P5
R5
T5
N6
GP18
GP17
GP16
GP15
H_PSW1_N
H_PSW2_N
D_SUSPEND
D_WAKEUP
H_SUSPEND
H_WAKEUP
VDDP
VDD
GND
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
A1
A2
CS5#
OE#
RD/WR#
GP5
GP4
RDY
RESET_OUT#
U1
ISP1161
Vhold1
VCC
Vreg(3.3)
Vhold2
NDP_SEL
R201 10K
H_WAKEUP
H_SUSPEND
D_WAKEUP
D_SUSPEND
H_PSW2_N
H_PSW1_N
H_OC2_N
H_OC1_N
H_D2+
H_D2-
H_D1+
H_D1-
D_D+
D_D-
GL_N
D_VBUS
CLKOUT
X2
XTAL2
XTAL1
DREQ1
DREQ2
DACK1
DACK2
EOT
INT1
INT2
TEST
RESET_N
CS_N
RD_N
WR_N
DGND
AGND
R22 10K
6MHz
C1 22pF
C2 22pF
VREG
VHOLD2
C10 0.1uF
C7 0.1uF
VBUS_DN2
VBUS_DN1
Q1
Q2
C21 0.1uF
C22 0.1uF
LED
1k
R33
R12 100
R13 1M
C20 0.1uF
VBUS_UP
R14 1.5K
R15 22
R16 22
J7
USB_UP
VBUS
D-
D+
GND
CHASSIS
J5
USB_DOWN
J6
R4 22
R5 22
R8 22
R9 22
H_D1-
H_D1+
H_D2-
H_D2+
C14 47pF
R6 15K
R7 15K
C15 47pF
C16 47pF
R10 15K
R11 15K
C17 47pF
C18 47pF
C19 47pF
Title
Size
B
Number
Revision
Date: 17-Dec-2002
Sheet of
File: I:\Jason Backup 030701\Philips\ISP1161\Documentation\App Note Schematic\SA2.DDB

Appendix A. Hardware Installation

To install ISP1161x onto the Intel StrongARM SA1110:

1. Connect the ISP1161x Evaluation Board to the StrongBridge Board.
2. Connect the StrongARM Development Kit to the StrongBridge Board.
3. Connect the StrongARM Companion Chip Development Kit to the StrongARM Development Kit Board.

The ISP1161x interface to Intel StrongARM SA1110 Evaluation Kit is ready for testing!

For a clearer graphical installation, see the following figures:

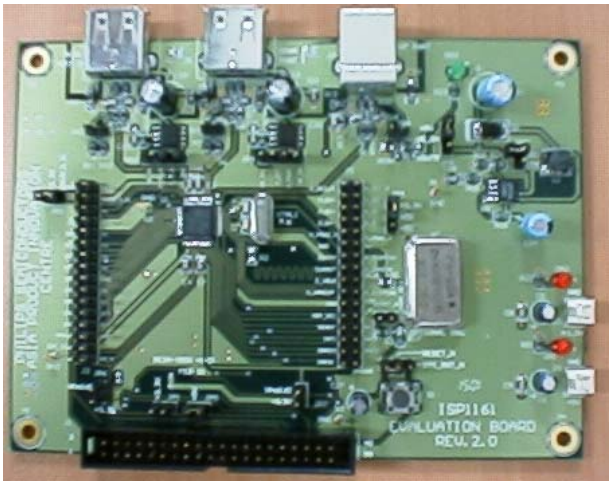


Figure A-1: ISP1161x Evaluation Board

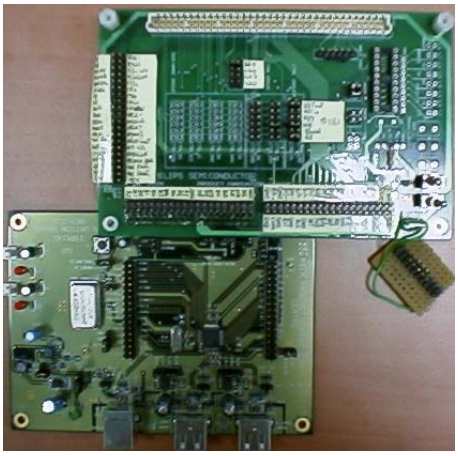


Figure A-2: ISP1161x Evaluation Board with StrongBridge Board

Figure A-3: ISP1161x Evaluation, StrongBridge and StrongARM Development Kit

Figure A-4: ISP1161x Evaluation, StrongBridge, StrongARM Development Kit and StrongARM Companion Chip Development Kit

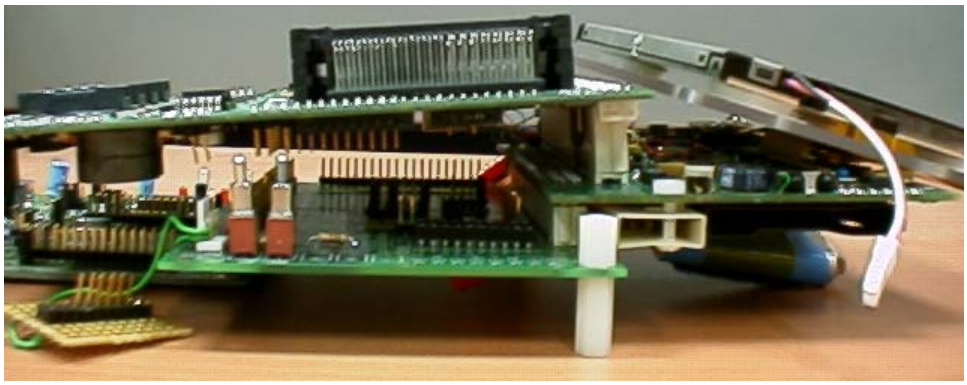


Figure A-5: Side View of ISP1161x Interface to Intel StrongARM SA1110 Evaluation Kit

Appendix B. Software Installation

Host Requirements

To develop on the Microsoft® Windows® CE platform:

1. Install Platform Builder on a PC running on Windows NT® 4.0 or Windows 2000. The current version for Platform Builder is 3.0
2. After the destination location has been chosen, the CESH Update Screen appears. Determine whether this is a new install or update of Microsoft Windows CE Platform Builder:
   - For new installs, choose **Use Ethernet**.
   - For updates, choose **Do not change Current CESH settings**.

Network Requirements

To use the Ethernet boot application to download the Windows CE image or to perform Ethernet debugging, the host and target systems must choose one of these network configurations:

- A network HUB with a DHCP server.
- A 10BASE-T unshielded twisted pair (UTP) crossover Ethernet cable between the host and SA1110 development board (provided with the SA1110 development board kit).

Often, the amount of network traffic present on a corporate network HUB can impede the downloading efforts. To avoid network traffic, it is recommended that you use this 10BASE-T UTP crossover Ethernet cable.

Appendix C. References

- *Universal Serial Bus Specification Rev. 2.0*
- *ISP1161A1 Full-speed Universal Serial Bus single-chip host and device controller* datasheet
- *ISP1161A Full-speed Universal Serial Bus single-chip host and device controller* datasheet
- *ISP1161 Full-speed Universal Serial Bus single-chip host and device controller* datasheet

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