PHILIPS

Philips Semiconductors

Connectivity

Oct 2001

Application Note

Interfacing ISP1161 to Intel StrongARM® SA1110 Processor

Rev. 3.0

Revision History:

Rev.	Date	Descriptions	Author
1.0	Aug 18, 2000	First draft	Socol Constantin
2.0	May 29, 2001	Updates after actual implementation	Ng Chee Yu
3.0a	June 7, 2001	Revised chapters and added Appendix A	Jason Ong
3.0b	Aug 2001	Changed to Philips template	Yuk-lin Ong
3.0c	Oct 8, 2001	Updated schematic	Jason Ong

We welcome your feedback. Send it to apic@philips.com.

Interfacing ISP1161 to Intel StrongARM SA1110

Table of Contents

Interfacing ISP1161 to Intel StrongARM SA1110

1. Overview

The unique design of Philips ISP1161 makes it possible to use it as a host controller (with two downstream ports) as well as a device controller (with one upstream port); these ports may be accessed independently. In this way, a simultaneous connection as a host controller and a device controller can be achieved.

When ISP1161 is integrated into a personal digital assistant (PDA) or handheld personal computer (HPC), in most configurations this means connecting it to the external bus interface of a RISC processor. This application note deals with the critical issues in ISP1161's embedded design, using the Intel StrongARM SA-1110 processor as a concrete example.

The actual implementation of the SA1110-ISP1161 uses an Intel StrongArm-1110 Hardware Development Platform, a Philips ISP1161 evaluation kit, and a Philips ISP1161 Bridging Board for SA1110.

2. ISP1161 Interface Signals to a RISC Processor Bus

The processor bus interface of ISP1161 was designed for a simple direct connection with a RISC processor. The data transfer can be done in PIO or DMA mode. The estimated maximum data transfer rate on the generic processor bus of the ISP1161—based on an ISP1161 internal clock frequency value equal to 48 MHz—is about 14 MB/s. To achieve the maximum data transfer rate on the host processor bus, the ISP1161 contains a Ping-Pong structured RAM which allows alternative access from the RISC processor or from the internal host/device controller. The "Ping-Pong" memory is allocated separately for the host and the device controllers. The host controller uses 2-KB of "Ping" memory and 2-KB of "Pong" in its allocated memory and the device controller uses 1.5 KB for each of the "Ping" memory and "Pong" in its own memory.

The main ISP1161 signals you should take into consideration for connecting to a StrongARM SA1110 processor will be:

- A 16-bit data bus: D0-D15 for ISP1161. ISP1161 is "little endian" compatible.
- Two address lines A0 and A1 necessary for complete addressing of the ISP1161 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 CS_N line used for selection of the ISP1161 in a certain address range of the host system. This input signal is active LOW.
- RD_N and WR_N 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 RESET signal is active LOW.

3. Intel StrongARM SA-1110 processor

The main features of the Intel StrongARM SA1110 processor of special interest for connecting to the ISP1161 are highlighted as follows:

The “Memory and PCMCIA Control Module” of the Intel StrongARM processor is responsible for generating all signals for interfacing with the ISP1161. The following features are useful for a direct connection of ISP1161:

- The “Memory Control Module” generates all the necessary signals to control different types of external devices:
  - DRAM (up to 4 banks of FPM, EDO, and SDRAM)
  - static memory (up to 3 banks of ROM, Flash, SRAM, SMAROM – selected by nCS0, nCS1 and nCS2 signals)
  - static memory and variable latency I/O devices (up to 3 banks of ROM, Flash, SMROM, and SRAM-like variable latency I/O devices, selected by 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. The 16-bit wide data bus size is used by the ISP1161.

- 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, which corresponds to ISP1161 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 ISP1161 and Intel SA1110:

The timing diagram of the external bus cycle of the Intel StrongARM SA1110 processor is determined by the “memory clock”, which is equal to two CPU clock cycles. Timing during RD/WR accesses is determined by the settings of the MSC0, MSC1 and MSC2 registers which 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 it is possible to modify the assert time of each beat of a burst RD/WR, the de-assert time between each beat of a burst RD/WR, and the hold-off time after a write to subsequent accesses.

According to ISP1161 datasheet specifications, a read operation requires the following timing parameters (the write operation is similar), defined in Figure 1:

- tRL = 33 nsec (RD_N low pulse width – minimal value required by ISP1161),
- tRHRL = 110 nsec (RD_N HIGH to next RD_N LOW – minimal value required by ISP1161) and
- tRHDZ = 3 nsec (RD_N hold time, minimal value that can be expected from ISP1161).
- tRC = 143 nsec (will obviously result as a sum of tRL and tRHRL)
- tSHSL = 300 nsec (first RD_N/WR_N after CMD).

Consider the access of one of the internal registers of the ISP1161 (for example the Control Register of the HC) for a detailed analysis of the timing diagram. An access of one of the internal registers of ISP1161 will require two phases: first writing the address (index) of the selected register into the Command Port and then the data transfer access (RD/WR) may take place. Note: the index of each register is different in case of a RD or a WR operation.

The following timing diagram describes the two phases of accessing the ISP1161:

Interfacing ISP1161 to Intel StrongARM SA1110

- The first phase is accessing the Command (control) Port of ISP1161, to write the address (index) of the data port that will be accessed. In this phase CS_N is active, the data lines D0-D15 contain the desired address and then the WR_N pulse will be activated and will latch the data. Note the value of tSHSL that represents the minimum time required between occurrence of the first phase and the occurrence of the second phase. One example of the HC "Control Register" a value of 01H will be transferred during a RD operation and 81H during a WR operation.
- The second phase shown in the same diagram 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 Read access and one for Write access. A series of RD_N and WR_N pulses are shown in the diagram to define the timing requirements between two consecutive accesses to the ISP1161: tRHRL, tWHWL, tRC, tWC, tRLDV, as specified in the datasheet.

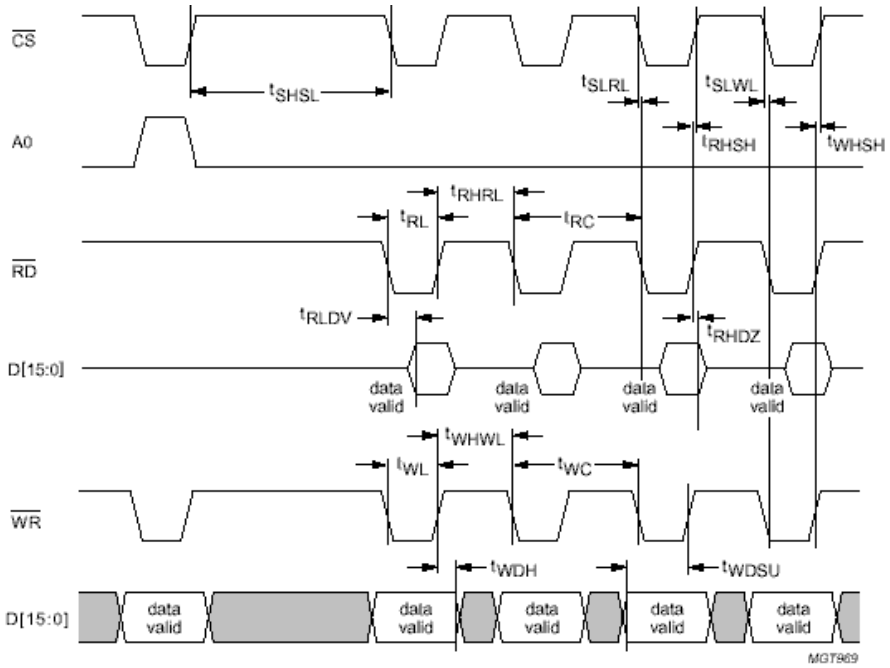


Figure 1. Parallel I/O Interface Timing (16-bit Read/Write)

5. Using interrupts

ISP1161 will generate two interrupts on INT1 and INT2 pins, allocated for the Host and the Device controller, respectively. These interrupts will occur depending on the setting of the following registers: *Interrupt Register, Interrupt Enable Register* and the *Hardware Configuration Register*. These registers are separately programmed for host controller and for device controller operation.

Connect the ISP1161 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 to 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 the ISP1161 are programmable as active on level or edge and HIGH or LOW—as specified in the *Hardware Configuration Registers* for each of the Host and Device controllers, separately. You must match the settings of the ISP1161 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 ISP1161's INT1 and INT2 output signals to "wake up" the host system's processor—in this case the SA1110—from the idle or sleep modes. The GPIO pins are defined as category 3 signals and are actively sampled by the SA1110 even during sleep mode. The sleep mode of SA1110 offers the greatest power savings. In this mode the internal sleep state machine of the 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 VDDX I/O voltage supply of SA1110 must be present during sleep mode to enable this wake-up method.

6. Suspend/Resume

You can make ISP1161 enter different functional states (Reset, Resume, Operational and Suspend) by programming the *Control Register* of the Host Controller or the *Mode Register* of the Device Controller.

Another way to "wake-up" ISP1161 from 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 the SA1110.

Monitoring the H_SUSPEND pin (for Host status) and D_SUSPEND pin (for Device status) can determine the actual status of the ISP1161, without having to access the internal status registers. Connecting these signals to any available GP I/O port of the SA1110 is an easy way to determine the status of the ISP1161.

ISP1161 may "wake up" when its CS_N input signal becomes active, if this is desired, by programming a "1" in bit 3 of the *Hardware Configuration Register* of the Device Controller. Alternatively, if the same bit is programmed to a value of "0", asserting the CS_N signal does not cause ISP1161 to wake up.

7. Schematic Diagram

The following schematic diagram shows the connection of the ISP1161 to a StrongARM SA1110 processor, in a minimal hardware configuration.

In this example schematic, ISP1161 is simply selected by nCS5. To correctly access the ISP1161, it is assumed that the memory space selected by nCS5 is programmed for 16 bit accesses and “little endian”.

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

The RESET input signal of ISP1161 is generated in the schematics by the RSTOUT signal generated by the SA1110 processor, when its RESET_IN signal is active. The RSTOUT signal is also asserted for “soft reset” events (sleep and watchdog). Connecting the RESET input of 1161 to RESET_IN of SA1110 may be a better solution if INT1 and INT2 are used to “wake up” SA1110.

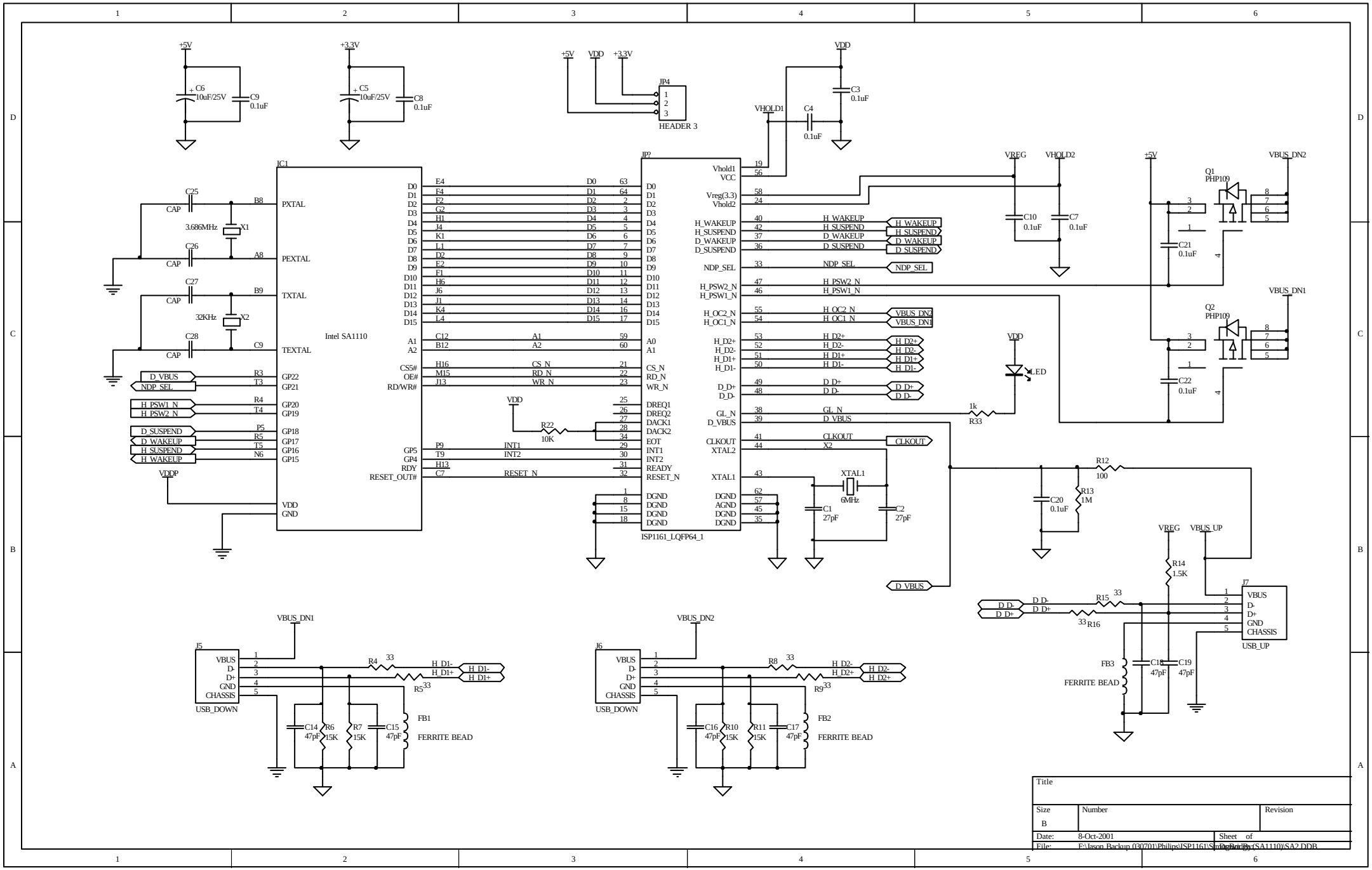
Pins H_PSW1 and H_PSW2 are connected to lines 4 and 5 of the same I/O port; this connection creates an alternative way to determine the power status of each downstream port.

Input signals H_OC1_N and H_OC2_N are used by the ISP1161 to detect an overcurrent on the downstream ports. As separate overcurrent detection and protection circuits are implemented for each downstream port in ISP1161, detection of an overcurrent on one of the downstream ports will power down that port only. Connecting the voltages of the two downstream ports VBUS_DN1 and VBUS_DN2 to H_OC1_N and H_OC2_N pins enables detection of the current value by sensing the voltage drop on Q1 and Q2 which are PMOS transistors with very low switch-on resistance Rds(on). Selection between Q1 and Q2depends on the desired maximum current value and this determines the value of Rds(on). For example, if the allowed maximum current is about 0.5 A, a voltage drop of 75 mV will trigger the overcurrent circuitry and an approximate value of Rds(on) = 150 mΩ will result. Connecting the 1161 input pins H_OC1_N and H_OC2_N to +5V will disable the internal overcurrent protection of ISP1161; 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 ISP1161’s NDP_SEL input signal, and one or two downstream ports will be selected accordingly.

Detection of a connection on the upstream port is achieved by connecting VBUS_UP to pin D_VBUS of the ISP1161. R12 and C20 will act as a low-pass filter that eliminates the possibility of sensing false voltage drop due to 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 ISP1161 and an external power source for the rest of the system.

The GL_N output signal indicates, through an LED, the status of the USB device and helps in troubleshooting the USB connection.


+5V
+3.3V
C6
10uF/25V
C9
0.1uF
C5
10uF/25V
C8
0.1uF
JP4
HEADER 3
VDD
VHOLD1
C4
0.1uF
C3
0.1uF
IC1
Intel SA1110
PXTAL
PEXTAL
TXTAL
TEXTAL
C25
C26
C27
C28
CAP
3.686MHz
X1
32KHz
X2
D_VBUS
NDP_SEL
H_PSW1_N
H_PSW2_N
D_SUSPEND
D_WAKEUP
H_SUSPEND
H_WAKEUP
GP22
GP21
GP20
GP19
GP18
GP17
GP16
GP15
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#
CS_N
RD_N
WR_N
INT1
INT2
RESET_N
R22
10K
JP?
ISP1161_LQFP64_1
Vhold1
VCC
Vreg(3.3)
Vhold2
H_WAKEUP
H_SUSPEND
D_WAKEUP
D_SUSPEND
NDP_SEL
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
XTAL2
XTAL1
DREQ1
DREQ2
DACK1
DACK2
EOT
READY
DGND
AGND
C1
27pF
C2
27pF
6MHz
VREG
VHOLD2
C10
0.1uF
C7
0.1uF
VBUS_DN1
VBUS_DN2
Q1
PHP109
Q2
PHP109
C21
0.1uF
C22
0.1uF
LED
1k
R33
C20
0.1uF
R12
100
R13
1M
VBUS_UP
R14
1.5K
J7
USB_UP
R15
33
R16
33
FB3
FERRITE BEAD
C18
47pF
C19
47pF
J5
USB_DOWN
R4
33
R5
33
H_D1-
H_D1+
C14
47pF
R6
15K
R7
15K
C15
47pF
FB1
FERRITE BEAD
J6
USB_DOWN
R8
33
R9
33
H_D2-
H_D2+
C16
47pF
R10
15K
R11
15K
C17
47pF
FB2
FERRITE BEAD
Title
Size
B
Number
Revision
Date:
8-Oct-2001
Sheet of
File:
F:\Jason_Backup_030701\Philips\ISP1161\Schematic\SA1110\SA2.DDB
Drawn By:

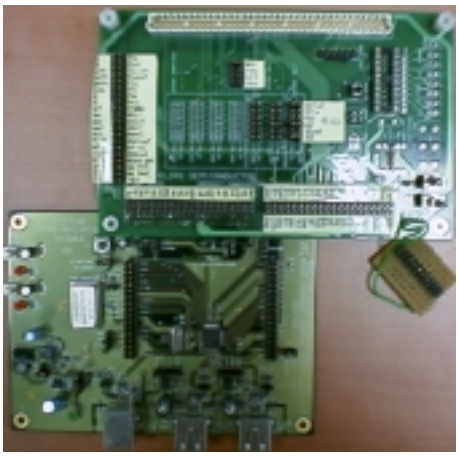
Appendix A –Hardware Installation

The installation of the ISP1161 to Intel StrongARM SA1110 is very simple. The procedure is as follows:
Step 1: Plug the ISP1161 Evaluation Board to the StrongBridge Board.
Step 2: Plug the StrongARM Development Kit to the StrongBridge Board.
Step 3: Plug the StrongARM Companion Chip Development Kit to the StrongARM Development Kit Board.
The ISP1161 interface to Intel StrongARM SA1110 Evaluation Kit is ready for testing!

For a clearer graphical installation, please see the following figures:

Figure 3. ISP1161 Evaluation Board

Figure 4. ISP1161 Evaluation Board with StrongBridge Board



Interfacing ISP1161 to Intel StrongARM SA1110

Figure 5. ISP1161 Evaluation, StrongBridge and StrongARM Development Kit

Figure 6. ISP1161 Evaluation, StrongBridge, StrongARM Development Kit and StrongARM Companion Chip Development Kit

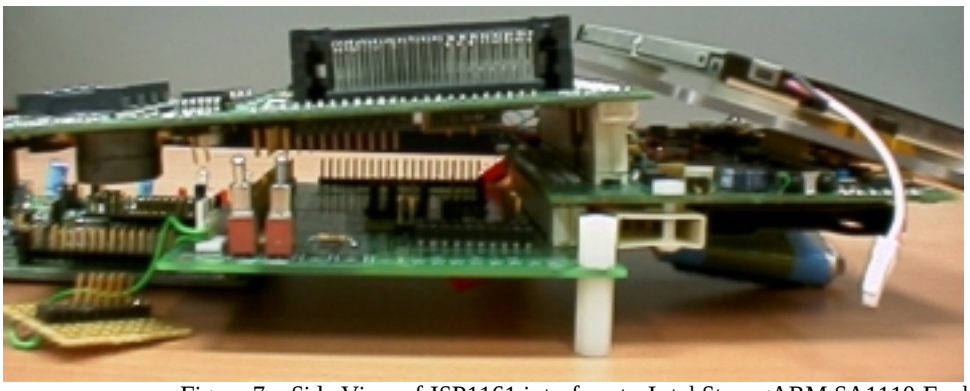


Figure 7. Side View of ISP1161 interface to Intel StrongARM SA1110 Evaluation Kit

Appendix B –Software Installation

Host Requirements

To develop on the WinCE platform:

1. Install Platform Builder in 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 if 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 do Ethernet debugging, the host and target systems must choose **one** of the network configurations listed below:

- A network HUB with a DHCP server
- A 10BASE-T unshielded twisted pair (UTP) “cross-over” Ethernet cable to go between the host and SA-1110 Development Board (provided with the SA-1110 Development Board kit). The amount of network traffic often 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.