
ST-NXP Wireless

IMPORTANT NOTICE

Dear customer,

As from August 2nd 2008, the wireless operations of NXP have moved to a new company, ST-NXP Wireless.

As a result, the following changes are applicable to the attached document.

- **Company name - NXP B.V.** is replaced with **ST-NXP Wireless**.
- **Copyright** - the copyright notice at the bottom of each page "© NXP B.V. 200x. All rights reserved", shall now read: "© ST-NXP Wireless 200x - All rights reserved".
- **Web site** - <http://www.nxp.com> is replaced with <http://www.stnwireless.com>
- **Contact information** - the list of sales offices previously obtained by sending an email to salesaddresses@nxp.com , is now found at <http://www.stnwireless.com> under Contacts.

If you have any questions related to the document, please contact our nearest sales office.
Thank you for your cooperation and understanding.

ST-NXP Wireless

AN10026

Interfacing ISP1160x to Freescale DragonBall EZ RISC processor

Rev. 02 — 24 January 2007

Application note

Document information

Info	Content
Keywords	isp1160; isp1160/01; host controller; usb; universal serial bus
Abstract	This application note explains the critical issues in the ISP1160x embedded design, using the Freescale DragonBall EZ RISC processor as an example. Remark: ISP1160x denotes any NXP embedded USB host controller whose name starts with 'ISP1160'; this includes ISP1160/01, and any future derivatives.

Revision history

Rev	Date	Description
02	20070124	Second release. Updated Fig 2 .
01	20030107	First release.

Contact information

For additional information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Remark: ISP1160x denotes any NXP embedded USB host controller whose name starts with 'ISP1160'; this includes ISP1160/01, and any future derivatives.

1. Overview

When the ISP1160x is integrated into a Personal Digital Assistant (PDA) or Handheld Personal Computer (HPC), it is usually connected to the external bus interface of a RISC processor. This application note deals with the critical issues in the ISP1160x embedded design, using the Freescale DragonBall EZ RISC processor as a concrete example.

2. ISP1160x processor interface signals

The ISP1160x processor bus interface is designed for a simple direct connection with a RISC processor. The data transfer can be done in Programmed I/O (PIO) or Direct Memory Access (DMA) mode. The estimated maximum data transfer rate on the ISP1160x generic processor bus is approximately 15 MB/s. This is based on an ISP1160x internal clock frequency value of 48 MHz. To achieve the maximum data transfer rate on the host processor bus, the ISP1160x contains a ping-pong structured RAM that allows alternative access from the RISC processor or from the internal Host Controller. The Host Controller uses 2 kB of the ping memory and 2 kB of the pong memory in its allocated memory.

The main ISP1160x signals you must take into consideration to connect to a Freescale DragonBall EZ RISC processor are:

- A 16-bit data bus (D[15:0]) for the ISP1160x, which is "little endian" compatible.
- An address line (A0) necessary for complete addressing of the ISP1160x internal registers:
 - **A0 = 0** Selects the Data Port of the Host Controller
 - **A0 = 1** Selects the Command Port of the Host Controller
- One $\overline{CS}$ line used to select the ISP1160x in a certain address range of the host system. This input signal is active LOW.
- $\overline{RD}$ and $\overline{WR}$ are common read and write signals. These signals are active LOW.
- A DMA channel standard control lines: DREQ, $\overline{DACK}$ and EOT. Because the DragonBall EZ processor does not contain a DMA controller, these signals will not be used in a minimal hardware implementation.
- Interrupt line INT1, which has programmable level/edge and polarity (active HIGH or LOW).
- The CLKOUT signal has a maximum value of 48 MHz.
- The $\overline{RESET}$ signal is active LOW.

3. Freescale DragonBall EZ

Freescale MC 68EZ328, also called DragonBall EZ, is a processor from the second generation of the DragonBall EZ family. The MC68EZ328 combines a Freescale MC68EC000 processor with intelligent peripheral modules and typical system interface logic. The operating frequency of this processor is 16 MHz, obtained by an internal PLL, which accepts as input a 32.768 kHz crystal, oscillator or a clock signal.

The main internal blocks of the DragonBall-EZ processor to consider when analyzing the bus interface, on which ISP1160x is connected, are as follows:

- The 8-bit or 16-bit 68000-bus interface block.
- The clock synthesizer and power control block.
- The chip-select generation block.
- The interrupt controller.

The DragonBall-EZ processor generates eight programmable general-purpose chip-select signals, which are arranged in four groups of two (CSA[1:0], CSB[1:0], CSC[1:0] and CSD[1:0]). Each chip-select block allows several features to be selected, which may be specific to devices connected to each selected area:

- Bus size: 8 bits or 16 bits can be independently selected for each area. The default setting corresponds to a 16-bit bus size.
- Number of wait states inserted in a bus cycle can be set from zero to six.
- Each area selected by a CSn can be defined as read-only or read/write access.
- Different types of memory are supported for an area selected by a certain CSn: DRAM, ROM, SRAM and Flash memory.
- The size of any memory area selected by a chip select signal can be selected from a set of predefined range 32 kB, 64 kB, 128 kB, 256 kB, 512 kB, 1 MB, 2 MB, 4 MB.

For a CSn signal to correctly operate, you must program the “Chip-Select Group Base Address registers” and the “Chip Select registers”, accordingly.

4. Considerations in timing diagrams and WAIT states

The following is a short study of ISP1160x timing diagrams.

According to ISP1160x data sheet specifications, a read operation requires the following timing parameters (the requirements of the write operation are similar); see [Fig 1](#).

- $t_{RL} = 33 \text{ ns}$ ($\overline{RD}$ LOW pulse width: minimal value ISP1160x requires)
- $t_{RHRL} = 110 \text{ ns}$ ($\overline{RD}$ HIGH to next $\overline{RD}$ LOW: minimal value ISP1160x requires)
- $t_{RHDZ} = 3 \text{ ns}$ ($\overline{RD}$ hold time, minimal value that can be expected from ISP1160x)
- $t_{RC} = 143 \text{ ns}$ (will result as a sum of t_{RL} and t_{RHRL})
- $t_{SHSL} = 300 \text{ ns}$ (first $\overline{RD}/\overline{WR}$ after command)

For a detailed analysis of a timing diagram, consider the access of an ISP1160x 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 (read/write) may take place.

Remark: The index of each register differs according to whether it is a read or a write operation.

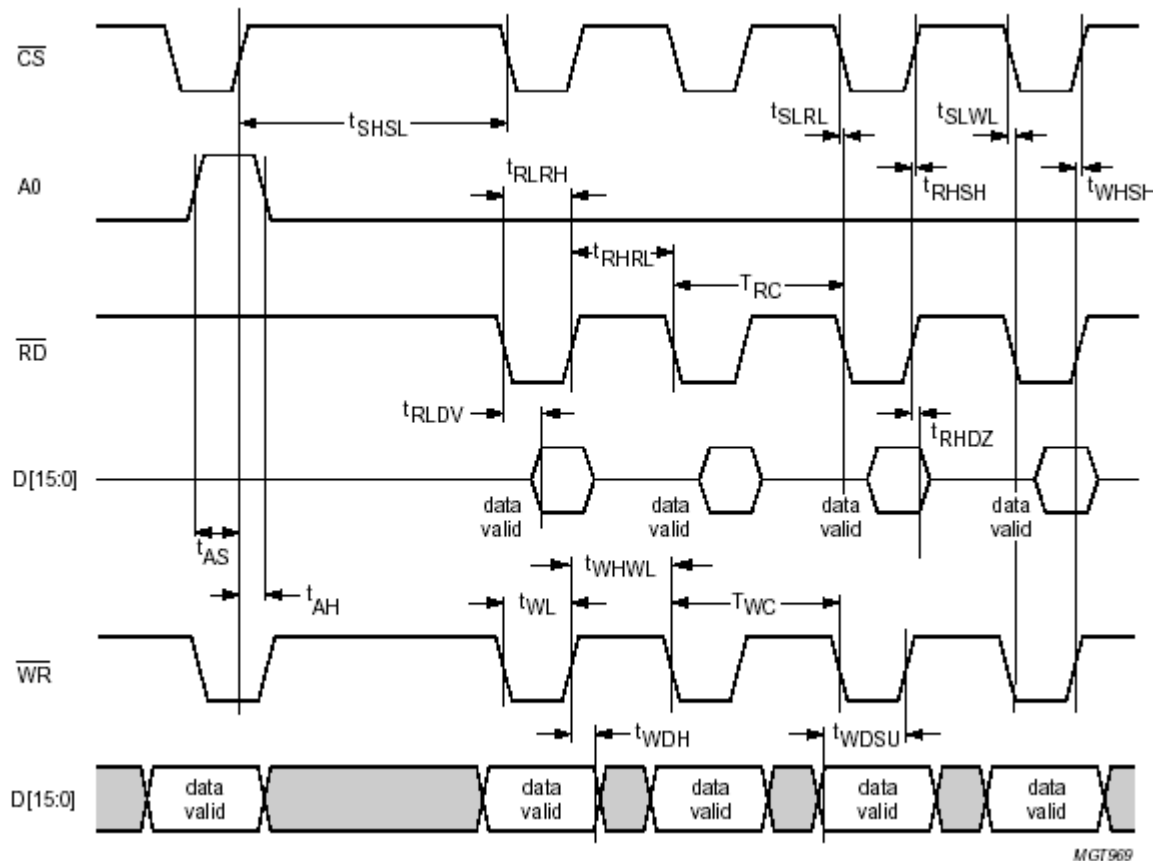


Fig 1. Programmed interface timing (16-bit read/write)

When the ISP1160x connection area is defined as SRAM, the ISP1160x will operate correctly for bus clock CKIO of 33 MHz. Timing measurements show that inserting wait-states in standard bus cycles of DragonBall EZ is unnecessary. Nevertheless, wait-state insertion is described, to cater for cases when faster bus cycles are used to access the ISP1160x.

Wait states insertion can be implemented through hardware or software. Both solutions will delay the rising edge of $\overline{RD}$ or $\overline{WR}$ to the next CKIO cycle and will determine an elongation of the $\overline{RD}$ or $\overline{WR}$ LOW pulse that can be calculated as:

$$t_w = W \times T(\text{CKIO}); \text{ where: } W \text{ is the number of wait states desired}$$

$T(\text{CKIO})$ is the cycle length of CKIO.

Remark: The value of t_{RHRL} will not be modified by the number of wait states inserted by any of the solutions mentioned earlier. The value of this parameter must be calculated and correctly adjusted according to the number and length of instructions executed by the DragonBall EZ processor between two successive accesses to the ISP1160x. The 'software solution' for wait-state insertion in a bus cycle is simple and is preferred in a minimal configuration, if additional wait states are necessary.

5. Using interrupts

The ISP1160x generates an interrupt on the INT pin. This INT signal can directly be connected to any available IRQ signal of the DragonBall processor. The ISP1160x INT can be programmed as active on level or edge and HIGH or LOW, as specified in the HcHardwareConfiguration register.

6. Suspend and resume

You can enable the ISP1160x to enter the reset, resume, operational and suspend functional states by programming the HcControl register of the Host Controller.

Another way to wake up the ISP1160x from suspend mode is to use input signals H_WAKEUP (for the Host Controller). Monitoring the H_SUSPEND pin for the host status can determine the ISP1160x actual status, without having to access internal status registers.

The ISP1160x may wake up when its $\overline{\text{CS}}$ input signal becomes active.

7. Schematic diagram

The schematic diagram on the following page shows the ISP1160x connected to a Freescale DragonBall EZ processor, in a minimal hardware configuration. For a more detailed description of the connection of the ISP1160x to a RISC processor and a study of each category of signals and timing diagrams, refer to application note *Interfacing ISP1160x to Hitachi SH7709 RISC Processor*.

In this configuration, the ISP1160x is selected by CSB0# signal, which is asserted according to the values programmed in the Chip Select Group Base Address and Chip Select registers corresponding to CSB0#.

To correctly access the ISP1160x, it is assumed that the area selected by CSB0# is programmed for the SRAM memory type and for 16-bit bus width accesses.

Interrupts INT are connected to the IRQ2 line of the DragonBall EZ processor. The interrupt inputs of the DragonBall EZ processor can be set as edge or level sensitive, and of positive or negative polarity by programming its Interrupt Control register. The ISP1160x also allows programming polarity (LOW or HIGH) and signaling mode (level or pulse) for generated interrupts INT, by setting the bits of the HcHardwareConfiguration register of the Host Controller. This ISP1160x feature allows a simple connection of the IRQ line, without any additional logic. The interrupts of the DragonBall processor can be masked in the Interrupt Mask register. The status of an interrupt line can be checked by monitoring the values of the Interrupt Status register and the Interrupt Pending register of the DragonBall-EZ processor.

Analysis of the timing diagrams of the ISP1160x and the DragonBall-EZ processor, running at a standard frequency of 16 MHz, shows that inserting wait states in a standard bus cycle is unnecessary. If a similar RISC processor with a higher bus frequency is used, you can insert wait-states during a bus cycle (accessing a certain system resource selected by CSn) by specifying the number of wait states in the corresponding (chip select) n register of the DragonBall processor.

The ISP1160x uses input signals $\overline{\text{H_OC1}}$ and $\overline{\text{H_OC2}}$ to detect an overcurrent on downstream facing ports. When an overcurrent is detected on a downstream port, power is turned off at that port only because of separate overcurrent detection and protection circuits implemented in the ISP1160x. Connecting voltages of two downstream ports

VBUS_DN1 and VBUS_DN2 to the $\overline{H_OC1}$ and $\overline{H_OC2}$ pins enable current to be detected by sensing the voltage drop on Q1 and Q2 that are MOSFET transistors with very low switch-on resistance R_{ds} . Choose Q1 or Q2, depending on the maximum current you want, which in turn determines the value of $R_{ds(on)}$. For example, if a voltage drop of 75 mV triggers the overcurrent circuitry and the allowed maximum current is about 0.5 A, $R_{ds(on)}$ will be approximately 150 M Ω . Connecting ISP1160x input pins $\overline{H_OC1}$ and $\overline{H_OC2}$ to +5 V will disable the ISP1160x internal overcurrent protection. You can opt for an external overcurrent protection circuit.

The $\overline{RESET}$ input signal of the ISP1160x is connected to the system RESET signal, which also generates the RESET# input signal for the DragonBall-EZ processor.

Interfacing ISP1160x to Freescale DragonBall EZ RISC processor

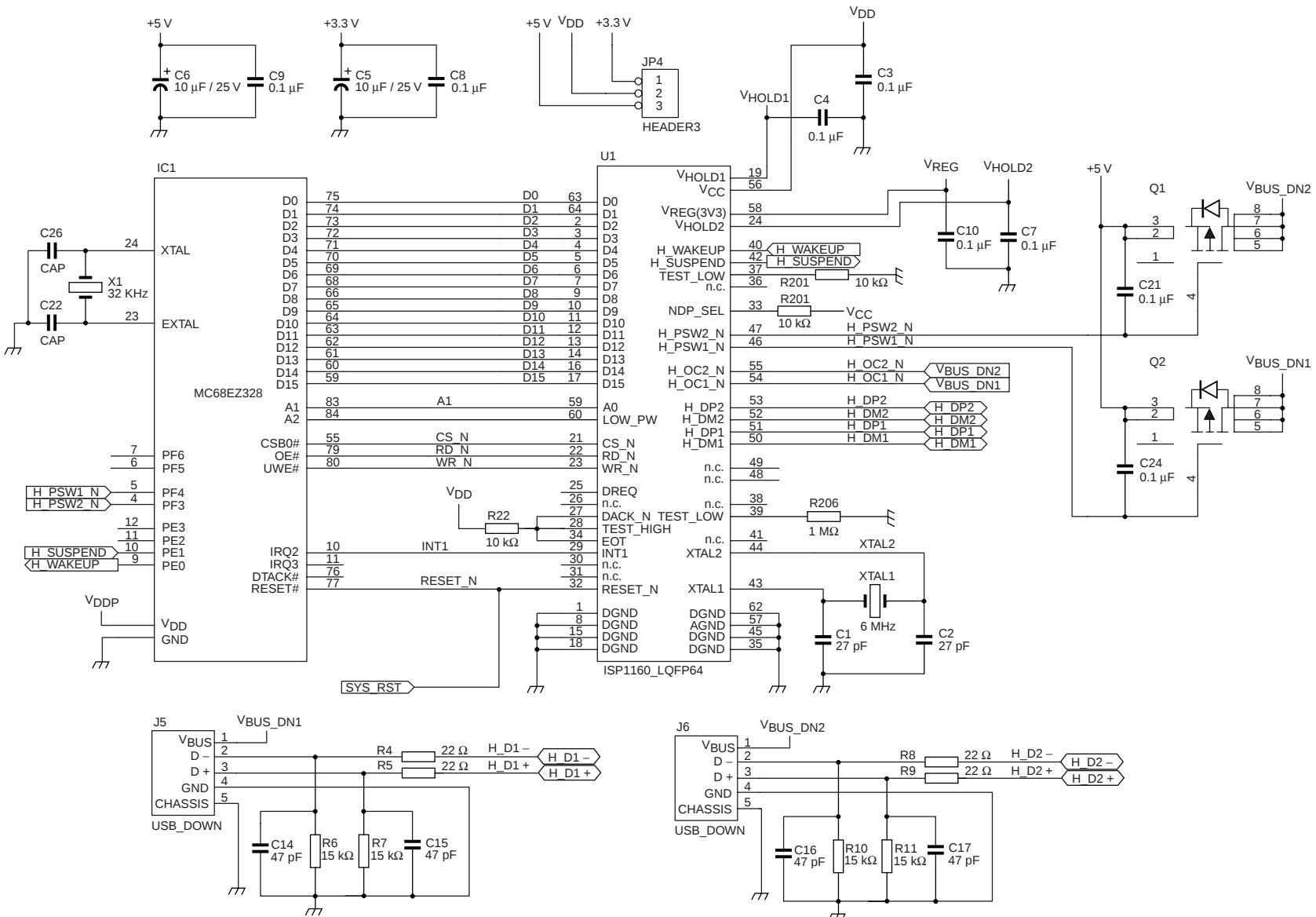


Fig 2. Schematics

8. References

- Universal Serial Bus Specification Rev. 2.0
- ISP1160 Embedded Universal Serial Bus Host Controller data sheet
- ISP1160/01 Embedded Universal Serial Bus Host Controller data sheet
- Interfacing ISP1160x to Hitachi SH7709 RISC Processor application note.

9. Legal information

9.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

9.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

9.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

10. Contents

1. Overview3

2. ISP1160x processor interface signals3

3. Freescale DragonBall EZ3

4. Considerations in timing diagrams and WAIT states4

5. Using interrupts.....6

6. Suspend and resume6

7. Schematic diagram6

8. References9

9. Legal information10

9.1 Definitions.....10

9.2 Disclaimers.....10

9.3 Trademarks10

10. Contents.....11

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.



© NXP B.V. 2007. All rights reserved.

For more information, please visit: <http://www.nxp.com>
For sales office addresses, email to: salesaddresses@nxp.com

Date of release: 24 January 2007
Document identifier: AN10026_2