Application Note

Interfacing ISP1181 to

Hitachi SH7709 RISC Processor

Rev 1.0

December 1, 2000

Revision History:

Rev.	Date	Descriptions	Author
1.0	12/01/00	First draft	Rajaram Veerappan

Interfacing ISP1181 to Hitachi SH7709 RISC Processor

Table of Contents

Interfacing ISP1181 to Hitachi SH7709 RISC Processor

1. Introduction

The ISP1181 is a Universal Serial Bus (USB) interface device that complies with Universal Serial Bus Specification Rev 1.1. It provides full speed USB communication capability to microcontroller- or microprocessor-based systems. The ISP1181 communicates with the system microcontroller or microprocessor through a high-speed general-purpose parallel interface. It also supports fully autonomous DMA operation.

Because of its high performance and versatile interface configurations, the ISP1181 is ideally suited for applications in many PC peripherals such as printers, scanners, external mass storage devices, digital still cameras (DSCs), personal digital assistants (PDAs), and communication devices. This means it can be connected to the external bus interface of any microcontrollers or RISC processors. This application note will address the most important issues in such a design, using the Hitachi SH7709 RISC processor as an example.

2. ISP1181 interface signals to a RISC processor bus

The processor bus interface of the ISP1181 is designed for a simple glueless interconnection with any RISC processor. The data transfer can be done in PIO or DMA mode; the estimated maximum data transfer rate is around 11.1 Mbyte/second. The ISP1181 has 14 programmable endpoints and 2 fixed control IN/OUT endpoints. The integrated physical size of the FIFO is 2462 bytes.
The main signals of the ISP1181 which must be taken care of while it is connected to a Hitachi SH7709 RISC processor will be:
A 16-bit data bus: D0-D15 for ISP1181. ISP1181 is “little endian” compatible.

- A0: An address line to select Command or Data.
  - A0 = 1 will select the Command port of the device controller.
  - A0 = 0 will select the Data Port of the Device Controller
- RD_N and WR_N - are common read and write signals for PIO and DMA modes. These signals are active low.
- DMA channel standard control lines: DREQ, DACK and EOT. These signals have programmable polarity levels.
- INT: The Interrupt signal has programmable triggering(level/edge) and Programmable polarity (active high or low).
- READY: a Low level indicates that the ISP1181 is processing a previous command and is not ready for the next PIO command or data transfer; a high level signals that the ISP1181 will complete the PIO data transfer.
- CS_N: Chip-select signal which is active Low.
- The CLKOUT signal, with a maximum value of 48 MHz, may be used efficiently as an input clock for the Hitachi SH7709.
- The RESET signal is active low.
- WAKEUP: Can be connected to any GPIO pin of the processor. It is an input pin used to generate Remote Wakeup from ‘suspend’ state. If not used, the pin is to be tied LOW (connect to DGND).
- SUSPEND: Can be connected to any GPIO pin of the processor.

The other input pins which need to be taken care of:

- ALE: Address Latch Enable input pin to be tied LOW (Connect to DGND).
- SDRD_N, SDWR_N: Unused pins to be tied HIGH (Connect to Vcc or Vreg(3.3))
- BUS_CONF0, BUS_CONF1: To be tied LOW (connect to DGND) for 16-bit I/O port and 16-bit DMA port configuration.

Note: ISP1181 can function normally with a supply voltage of 3.3 V as well as with 5.0 V. This enables easy implementation with different configurations.

3. Hitachi SH7709

The main features of the Hitachi SH7709 processor (a member of the SH-3 family) of special interest for connecting to the ISP1181 are highlighted as follows. This RISC processor contains two bus state controllers: one used by the CPU or cache memory (named BSC) and the other used by the internal DMA controller (named BSCP).

The Bus State Controller (BSC) of the SH7709 will generate all control signals on the external processor interface that will be used for the ISP1181 connection. The BSC divides the physical address space into six areas: each contains a maximum of 64 Mbytes. Using one of the CS1 - CS6 chip-select signals provided on the external bus is a simple way to select a certain area - allocated for a specific device.

Each of these areas has several features that can set by software:
- Bus size: 8, 16 or 32 bits can be set independently for each area
- Number of wait cycles can be set independently for each area
- Setting the type of space for each area will enable a direct connection to several possible types of memory: SRAM, DRAM, SDRAM, and burst ROM. The SH7709 will generate the necessary signals to control each of these types of memories.
- Each area can be used in either little or big endian mode. For a correct data alignment matching data width and endian is necessary. The ISP1181 connection will require a 16-bit or little endian configuration of the selected memory area.

Certain areas will be normally reserved in a system design, due to the presence of other system resources that are allocated in those specific areas. For example, the MS7709ASE Solution Engine board from Hitachi will allow a simple connection of the ISP1181 in area 2 or in area 5, just by using the CS2 or CS5 signals, as these 2 areas are not used by other system devices.

However, using one of the CSn signals without additional address selection logic may not be very efficient as this will not allow the addition of other devices into the same memory area. It is a good practice to allocate the ISP1181 to area 4, which is also used on the Hitachi MS7709ASE board by a Super I/O chip set and a 10BASE-T chipset, still leaving an available space of 32 MB, that can be allocated to other devices. Obviously, in this situation, adding simple selection logic is necessary, in order to include the ISP1181 in a certain address space.

4. Considerations on timing diagrams and WAIT states

A short study of the timing diagrams of the main bus cycles of both the ISP1181 and the SH7709 is done as follows:
The timing diagram of the external bus cycle of the SH7709 is based on the frequency of the system clock CKIO signal. In all operating modes of the CPG (clock pulse generator) of the SH7709 (using an external oscillator, external clock input on EXTAL pins or using CKIO pin as clock input), the CKIO bus clock frequency is always in the range of: 10 MHz to 40 MHz. Currently, in the majority of PDA designs, the most encountered frequency values of the CKIO signal are around 20–25 MHz.

According to the ISP1181 datasheet specifications, a read/write operation requires the following main timing parameters:
- tRLRH = 25 ns (RD# low pulse width – minimal value required by ISP1181),
- tWLWH = 22 ns.(WR LOW pulse width – minimal value required by ISP1181),

Interfacing ISP1181 to Hitachi SH7709 RISC Processor

- tRC,tWC = 180 ns (READ cycle time, WRITE Cycle time – minimal value required by ISP1181 for 16-bit configuration; For 8-bit, the cycle time is 90 ns)
- tSHDZ = 3 ns (Max) (Data bus high impedance after CS_N HIGH - Maximum value that can be expected from the ISP1181)
- tWHDZ = 3 ns.(Min) (Minimum Value required by the ISP1181 -Data hold after WR_N HIGH)
- tDVWH = 5 ns (Min) (Data setup before WR_N HIGH – Minimum value required by the ISP1181)

Consider the access of one of the internal registers of the ISP1181 for a detailed analysis of the timing diagram. An access to one of the internal registers will require two phases: first writing the address of the selected register into the Command Port and, second, the data transfer access (RD_N/WR_N) may take place.

The following timing diagram describes the 2 phases of accessing the ISP1181:
- The first phase is accessing the Command Port of ISP1181 to write the address of the data port that will be accessed. In this phase you can notice that 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 Tcy(WC-WD) that represents the minimum time required between occurrence of the first phase and the occurrence of the second phase.
- The second phase consists of the access (read or write) to the data port selected by the address latched in the previous phase.

The following is the access Cycle Timing:

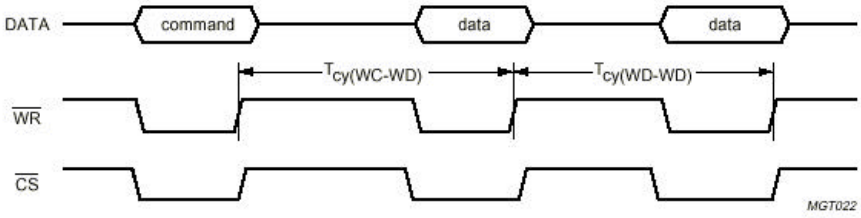
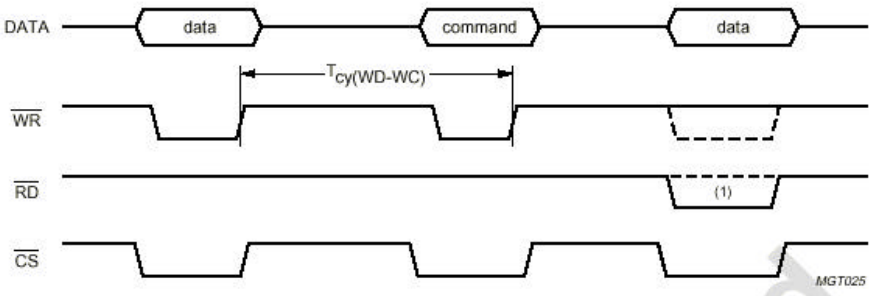


Fig 1 . Write command + write data cycle timing.



(1) Example: read data.

Fig 2 . Write data + write command cycle timing.

Interfacing ISP1181 to Hitachi SH7709 RISC Processor

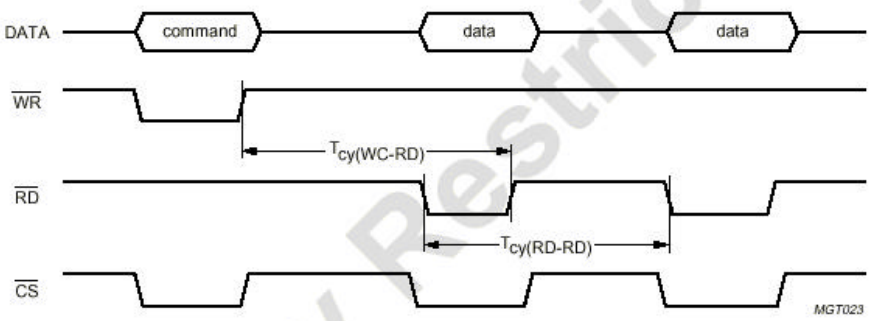
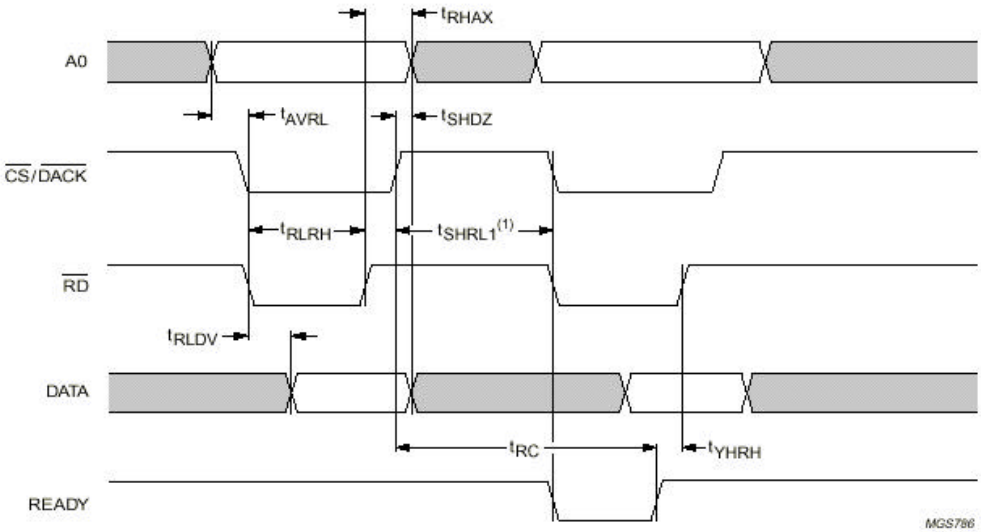


Fig 3 . Write command + read data cycle timing.

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min [1]	Max	Min [1]	Max	
Write command + write data (see Figure 1 and Figure 2)							
$T_{cy(WC-WD)}$	cycle time for write command, then write data		100 [2]	-	205	-	ns
$T_{cy(WD-WD)}$	cycle time for write data		90	-	205	-	ns
$T_{cy(WD-WC)}$	cycle time for write data, then write command		90	-	205	-	ns
Write command + read data (see Figure 3)							
$T_{cy(WC-RD)}$	cycle time for write command, then read data		100 [2]	-	205	-	ns
$T_{cy(RD-RD)}$	cycle time for read data		90	-	205	-	ns

[1] If the access cycle time is less than specified, the READY signal will be LOW until the internal processing has finished.
[2] Commands Acknowledge Setup, Clear Buffer, Validate Buffer and Write Endpoint Configuration require 180 ns to complete.

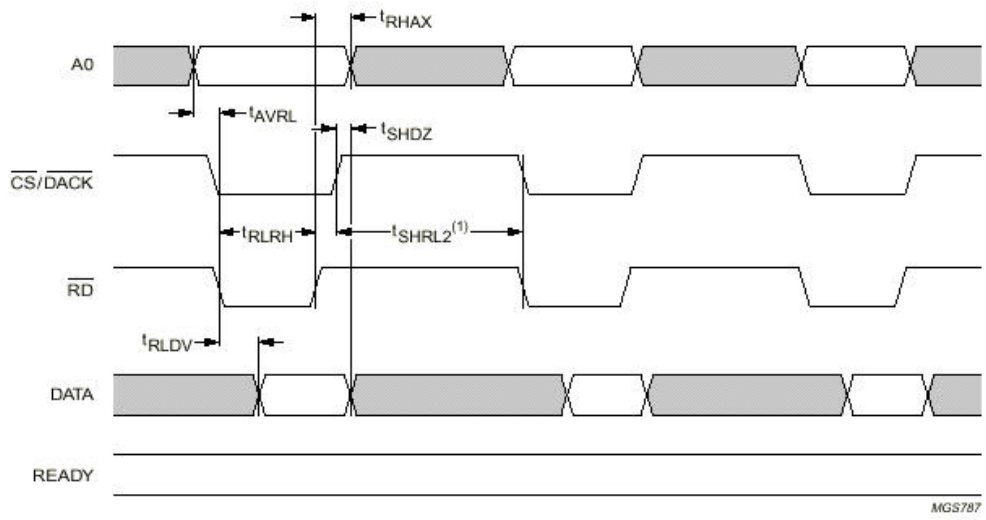
-The detailed Parallel I/O interface timing signals are given as follows:



(1) For t_{SHRL} both CS and RD must be LOW.

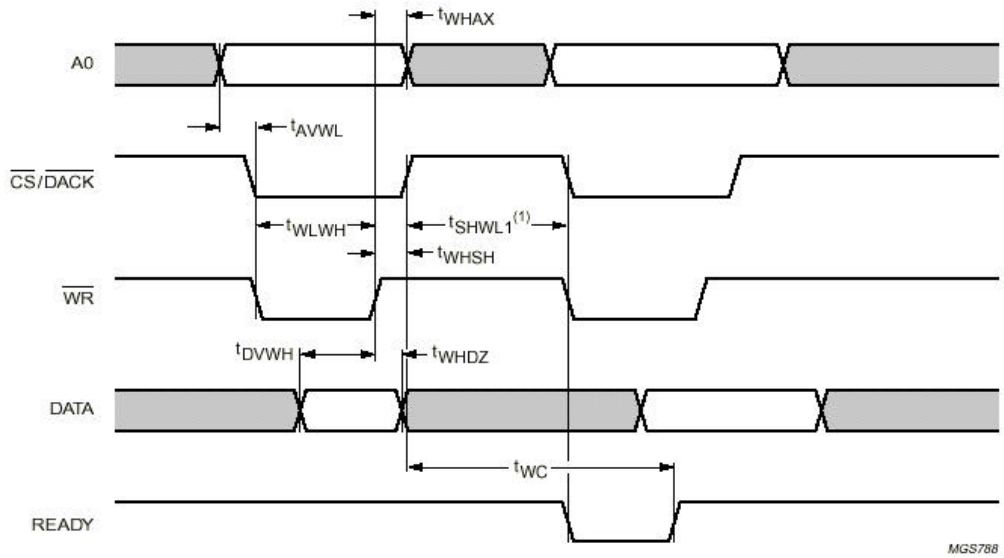
Parallel interface read timing (I/O and 8237 compatible DMA) with READY.

Interfacing ISP1181 to Hitachi SH7709 RISC Processor



(1) For t_{SHRL} both $\overline{CS}$ and $\overline{RD}$ must be LOW.

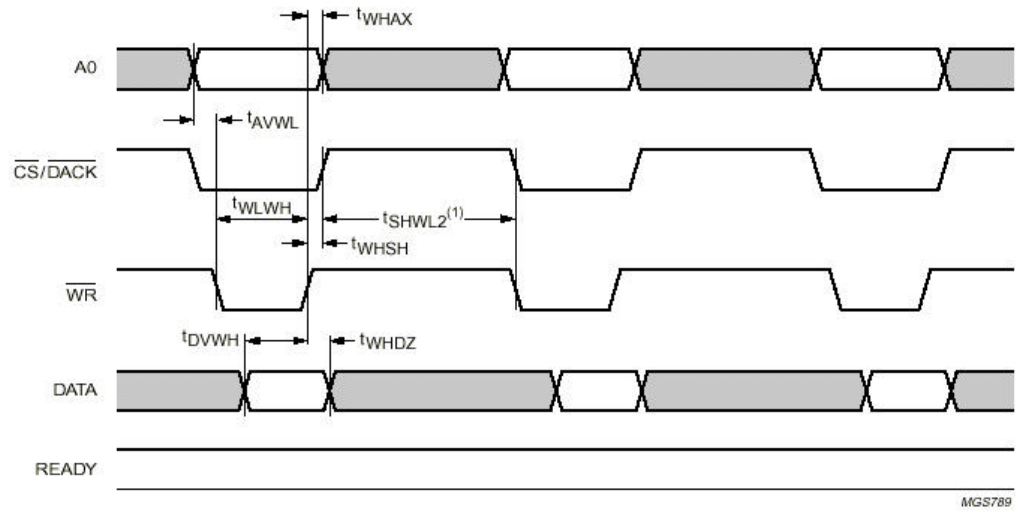
Parallel interface read timing (I/O and 8237 compatible DMA) without READY.



(1) For t_{SHRL} both $\overline{CS}$ and $\overline{WR}$ must be LOW.

Parallel interface write timing (I/O and 8237 compatible DMA) with READY.

Interfacing ISP1181 to Hitachi SH7709 RISC Processor



(1) For t_{SHRL} both $\overline{CS}$ and $\overline{WR}$ must be LOW.

Parallel interface write timing (I/O and 8237 compatible DMA) without READY.

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Read timing							
t_{RHAX}	address hold time after $\overline{RD}$ HIGH		3	-	3	-	ns
t_{AVRL}	address setup time before $\overline{RD}$ LOW		0	-	0	-	ns
t_{SHDZ}	data outputs high-impedance time after $\overline{CS}$ HIGH		-	3	-	3	ns
t_{RLRH}	$\overline{RD}$ pulse width		25	-	25	-	ns
t_{RLDV}	data valid time after $\overline{RD}$ LOW		-	22	-	22	ns
t_{SHRL1}	read interval after $\overline{CS}$ HIGH [1]	READY pulsing	22	-	22	-	ns
t_{SHRL2}	read interval after $\overline{CS}$ HIGH [1]	READY = HIGH	90	-	180	-	ns
t_{SHYH}	output READY HIGH after $\overline{CS}$ HIGH		-	90	-	180	ns
t_{YHRH}	$\overline{RD}$ HIGH time after output READY HIGH		0	-	0	-	ns
t_{RC}	read cycle duration		-	90	-	205	ns

Interfacing ISP1181 to Hitachi SH7709 RISC Processor

Write timing

t_{WHAX}	address hold time after $\overline{WR}$ HIGH		3	-	3	-	ns
t_{AVWL}	address setup time before $\overline{WR}$ LOW		0	-	0	-	ns
t_{SHWL1}	write interval after $\overline{CS}$ HIGH [2]	READY pulsing	22	-	22	-	ns
t_{SHWL2}	write interval after $\overline{CS}$ HIGH [2]	READY = HIGH	90/180 [3]	-	180	-	ns
t_{WLWH}	$\overline{WR}$ pulse width		22	-	22	-	ns
t_{WHSH}	chip deselect time after $\overline{WR}$ HIGH		0	-	0	-	ns
t_{DVWH}	data setup time before $\overline{WR}$ HIGH		5	-	5	-	ns
t_{WHDZ}	data hold time after $\overline{WR}$ HIGH		3	-	3	-	ns
t_{WC}	write cycle duration		-	90/180 [3]	-	205	ns

[1] Measured from $\overline{CS}$ going HIGH to $\overline{CS}$ and $\overline{RD}$ both going LOW.
[2] Measured from $\overline{CS}$ going HIGH to $\overline{CS}$ and $\overline{WR}$ both going LOW.
[3] Commands Acknowledge Setup, Clear Buffer, Validate Buffer and Write Endpoint Configuration require 180 ns to complete.

The values of the ISP1181 parameters mentioned earlier determined by the timing diagram of SH7709, in a standard bus cycle with no wait states and CKIO = 25MHz (in the worst case scenario – according to datasheet specifications) will be:

- tHRL = 40 ns (SH7709 low pulse width),
- tHRHRLmin = 60 ns (estimated approximate minimal value of RD# high to next RD# low) and
- tHRHDZ = 0 ns. (RD# hold time, required by SH7709, in which data must still be valid after the rising edge of RD# signal).

Defining the connection area of the ISP1181 as “ordinary memory”, a correct operation of the ISP1181 is ensured even if CKIO is higher than 33 MHz and timing measurements show that insertion of wait-states in the standard bus cycles of the SH7709 is not necessary. A description of the possible methods of wait-state insertion and correct usage of the READY signal is done below anyway, taking into consideration the situation when faster bus cycles will be used for accessing the ISP1181.

Wait states insertion into a RD_N/WR_N bus cycle can be achieved using two solutions: hardware or software implementation. Both solutions will delay the rising edge of RD_N to the next CKIO cycle and will determine an elongated RD_N low pulse, determined by:

tW = W x T(CKIO); where: (W) is the number of wait states desired
T(CKIO) – is the cycle length of CKIO

Note: observe that the value of tRHRL will not be modified by the number of wait states inserted by any of the solutions described below. It must be calculated and correctly adjusted according to the number and length of instructions executed by the SH7709 between two successive accesses of the ISP1181.

The “software solution” for wait-state insertion in a bus cycle is simple and is preferable in this configuration if additional wait states are necessary. The SH7709 allows insertion of a certain number of wait-states in a normal bus cycle by programming certain values into its internal *WCR2 register*. In this case, using the external WAIT signal of SH7709 is not necessary, as a predefined number of wait states will be inserted in each bus cycle. This method allows the designer to use minimal hardware implementation of the whole system. It is interesting to observe that the number of wait states can be differently defined for each of the 6 areas contained in the physical address space of the SH7709.

Programming the wait state control register WCR2 to insert a number of wait states for a slower device allocated in a certain area will not influence other faster resources selected in a different area.

The “hardware solution” for wait state insertion may also be simple in certain configurations, as shown below, and may imply just directly connecting the READY pin of the ISP1181 to the WAIT input pin of the SH7709. The READY signal generated by ISP1181 is active high and that corresponds to the WAIT logic of the SH7709. READY signal will be asserted by ISP1181 when the length of the bus cycle will determine a data transfer rate higher than 11.1 MB/s. Note: according to datasheet specifications, the WAIT input signal of the SH7709 is synchronous, and, to ensure a correct operation of the whole system, it is necessary to meet the setup (tWTS) and hold (tWTH) times relative to CKIO rising edges.

5. Using interrupts

ISP1181 will generate interrupt on INT pin – Occurrence of these interrupt depend on the setting of the following registers: *Interrupt Register, Interrupt Enable Register* and the *Hardware Configuration Register*. These registers can be programmed according to the requirement.

Connection of the INT signal can be done directly to any of the available IRQ signal of the SH7709. Choosing IRQ0 – IRQ4 pins of SH7709 as interrupt input lines connected to INT of the ISP1181 will allow waking up the SH7709 from the standby status when either one of these lines will become active. In this case, it is necessary to use a 32-kHz crystal connected to XTAL2/EXTAL2 pins and a battery connected to RTCVCC pin of SH7709.

INT of the ISP1181 is programmable as active on level / edge and high or low - as specified in the *Hardware Configuration Registers*.

The Hitachi SH7709 also allows sensing the interrupt on the rising or falling edge or on low level, by programming the desired values in the *Interrupt Control Register*; individual control of the characteristics of each interrupt line is possible. Obviously, it is necessary to match the settings of the ISP1181 interrupt line INT with the setting of the SH7709 IRQ line.

Detection of interrupt occurrence on IRQ line of SH7709 is done on the falling edge of the CKIO clock. Although the timing diagram of SH7709 specifies a setup time (tIRQS=10ns) and a hold time (tIRQH=4nsec) the IRQn signals are of the asynchronous type. This means that if these values are not met, the detection of the interrupt will be done in the next bus cycle (on the next falling edge of CKIO).

6. The DMA operation

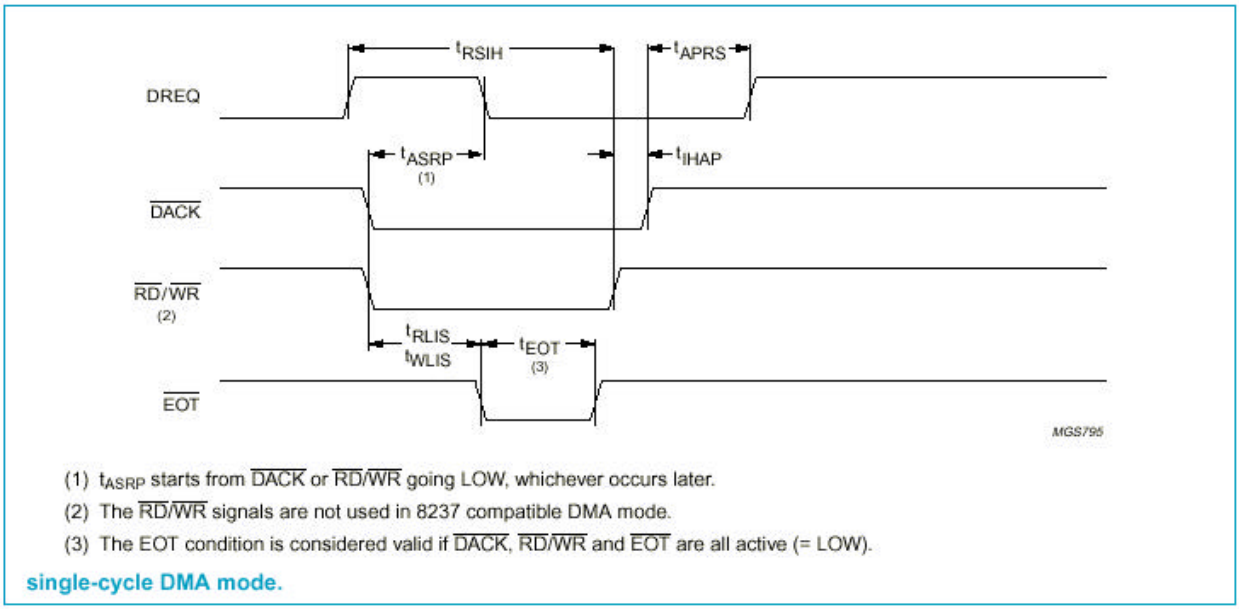
The DMA handler of ISP1181 can work in “DACK only mode” or in “8237 compatible mode”, according to the setting of the *Hardware configuration Register*. Also, the active levels of the DMA signals (DACK, DREQ and EOT) can be programmed using the Hardware Configuration Register. Programming the *DMA Configuration Register* and the *DMA Counter Register* is also necessary for defining the parameters of the DMA operation (transfer counter enable, DMA enable, burst length etc.). Details regarding the programming of DMA registers can be found in the ISP1181 datasheet and in the programming manual.

The DMA controller (DMAC) of SH7709 includes 4 DMA channels. Note: external requests (from DREQn pins) are accepted only from channels 0 and 1. An interrupt request can be generated to the CPU after transfer ends by a specified count.

Interfacing ISP1181 to Hitachi SH7709 RISC Processor

Due to the flexible interface of both the ISP1181 and the SH7709, the DMA signals DREQ and DACK of the ISP1181 can be simply connected to the DMA signals DREQ0 and DACK0 of SH7709. The DMAC of SH7709 allows selecting the detection of DREQ on the falling edge or low level and also the DACK output can be programmed as active low or high. These settings are contained in the respective *DMA Channel Control Register* of SH7709. The EOT signal of ISP1181 is not used in this configuration, as there is no corresponding pin on the DMAC interface of SH7709. Since the EOT pin is not connected anywhere, a pull-up resistor must be provided on this pin.

As shown in the timing diagram below, by asserting the DACK0 signal, the host system will allow data transfer to take place, allocating the bus for the ISP1181. Both RD_N and WR_N data transfer cycles are contained in the timing diagram below. The CS_N signal is not used by the internal selection logic of the ISP1181, and the DACK0 signal will be used instead to define the bus owner that issued the DREQ and is currently involved in the data transfer. No other resource of the system will be accessed as long as the DMA cycle is in progress and occupies the bus; the DACK0 active low pulse determines the time allocated to the DMA data transfer.



(1) t_{ASRP} starts from $\overline{DACK}$ or $\overline{RD}/\overline{WR}$ going LOW, whichever occurs later.
(2) The $\overline{RD}/\overline{WR}$ signals are not used in 8237 compatible DMA mode.
(3) The EOT condition is considered valid if $\overline{DACK}$, $\overline{RD}/\overline{WR}$ and $\overline{EOT}$ are all active (= LOW).

single-cycle DMA mode.

The timing diagram of SH7709 specifies tDRQS = 12 ns (DREQ setup time) and tDRQH = 8 ns (DREQ hold time) relative to a falling edge of CKIO. If the DRQ signal generated by the ISP1181 does not meet the tDRQS timing in the current bus cycle, it will be sensed on the next falling edge of CKIO, and the DACK will be generated accordingly.
The requirement tDRQH = 8 ns is determined by the sum: DACK delay time from the falling edge of CKIO + tDLDL (the time between DACK is driven low by SH7709 until DREQ is deactivated by ISP1181). As tDLDL is 10 ns to 22 ns the requirement tDRQH = 8 ns will be always satisfied.

A DMA burst access of up to 16 cycles for the 1181 DMA handler can be defined in the *DMA Configuration Register*.
If necessary, wait states can be added to the basic DMA cycle to create longer RD_N/WR_N and DACK pulses. Wait states insertion can be achieved in the same way as described in chapter 4.

7. Suspend /Resume

The ISP1181 can be determined to enter different functional states such as SUSPEND and WAKEUP by programming the *Mode Register* ('GOSUSP') and the *Hardware configuration* (bits PWROFF, WKUPCS). Also, issuing the soft reset command (0xF6) will reset the ISP1181.

Another alternative to determine ISP1181 to "wake-up" from Suspend mode is to set the WKUPCS (*Hardware Configuration register*) bit '1' which will do a remote wake-up via a low level on input CS_N.

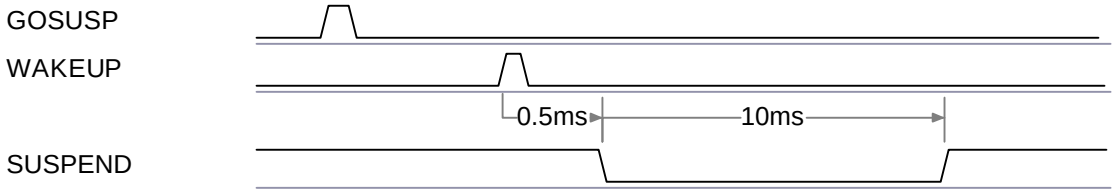
The behavior of the SUSPEND pin can be changed by the setting of the bit PWROFF (*Hardware Configuration register*) bit.

The ISP1181 detects USB suspend condition in the following 3 cases:

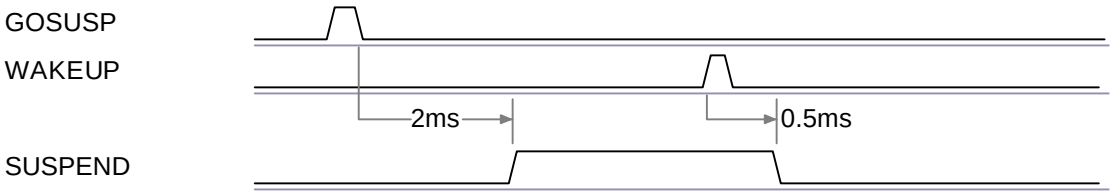
1. J state is present on the bus for 3 ms;
2. VBUS is lost (there is weak pull-up/pull-down on D+/D- when VBUS is lost);
3. *SoftConnect*™ is disabled (*Mode Register* bit SOFTCT = 0) when this feature is used (*Hardware Configuration* bit EXTPUL = 0). ISP1181 is effectively disconnected from USB bus in this situation.

After detecting the 'suspend' condition, writing a logic '1' (bit GOSUSP in the *Mode register*) followed by a logic '0' will activate the ISP1181 into 'suspend' mode and it will be in 'suspend' mode for at least 5 ms, before reacting to the global resume, bus traffic, wakeup on chip-select or wakeup pin activities.

When the PWROFF bit is set as '0' in the *Hardware configuration register*, the SUSPEND pin will behave in the following manner.:



When the PWROFF bit is set as '1' in the Hardware Configuration register, the SUSPEND pin will be behaving in the following manner:



Interestingly, the ISP1181 can wake up when its CS_N input signal becomes active by programming a '1' in bit 3 (WKUPCS) in the *Hardware Configuration register* .

8. Schematic Diagram

The following schematic diagram shows the connection of the ISP1181 to a SH7709 processor, in a minimal hardware configuration.

In the example schematic, the ISP1181 is selected by CS5 (in a more complex configuration some glue logic may be required to generate a CS signal composed of CSn and several address lines). To access the ISP1181 correctly, it is presumed that Area 5 is programmed for the SRAM memory type and 16-bit accesses.

Interrupt INT is arbitrarily connected to IRQ2 of SH7709. A 32 kHz oscillator is used as a second clock generator to give the possibility to wake up the SH7709 when the interrupts is generated, as mentioned in chapter 5.

The DREQ and DACK lines are directly connected to the DREQ and DACK lines of the SH7709 DMA channels 0. The EOT signal, which is not used in this configuration, is connected to a pull-up resistor to keep this input pin in a controlled inactive logic state.

The READY signal of ISP1181 is directly connected to the WAIT input of SH7709, as the CLKOUT signal of the ISP1181 is used as an input clock for the RISC processor. The CLKOUT frequency must be adjusted for a value below 40 MHz, as mentioned earlier. For example, a value of 24 MHz may be used; in this case a multiplier value of 4 will determine an internal frequency of 96 MHz for the SH7709. The estimated maximum transfer rate allowed by the ISP1181 will not be decreased due to the reduction of the CLKOUT frequency value.

The WAKEUP and SUSPEND pins are connected to the I/O port PTC of SH7709. A direct control is possible using this configuration: the SUSPEND state of the ISP1181 can be directly monitored and an immediate WAKEUP can be obtained, without accessing the ISP1181 internal registers. An alternative method to determine ISP1181 WAKEUP is using CS_N signal, as shown in chapter 7.

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

The RESET input signal of ISP1181 is generated by the RSTOUT signal generated by the SH7709.

Interfacing ISP1181 to Hitachi SH7709 RISC Processor

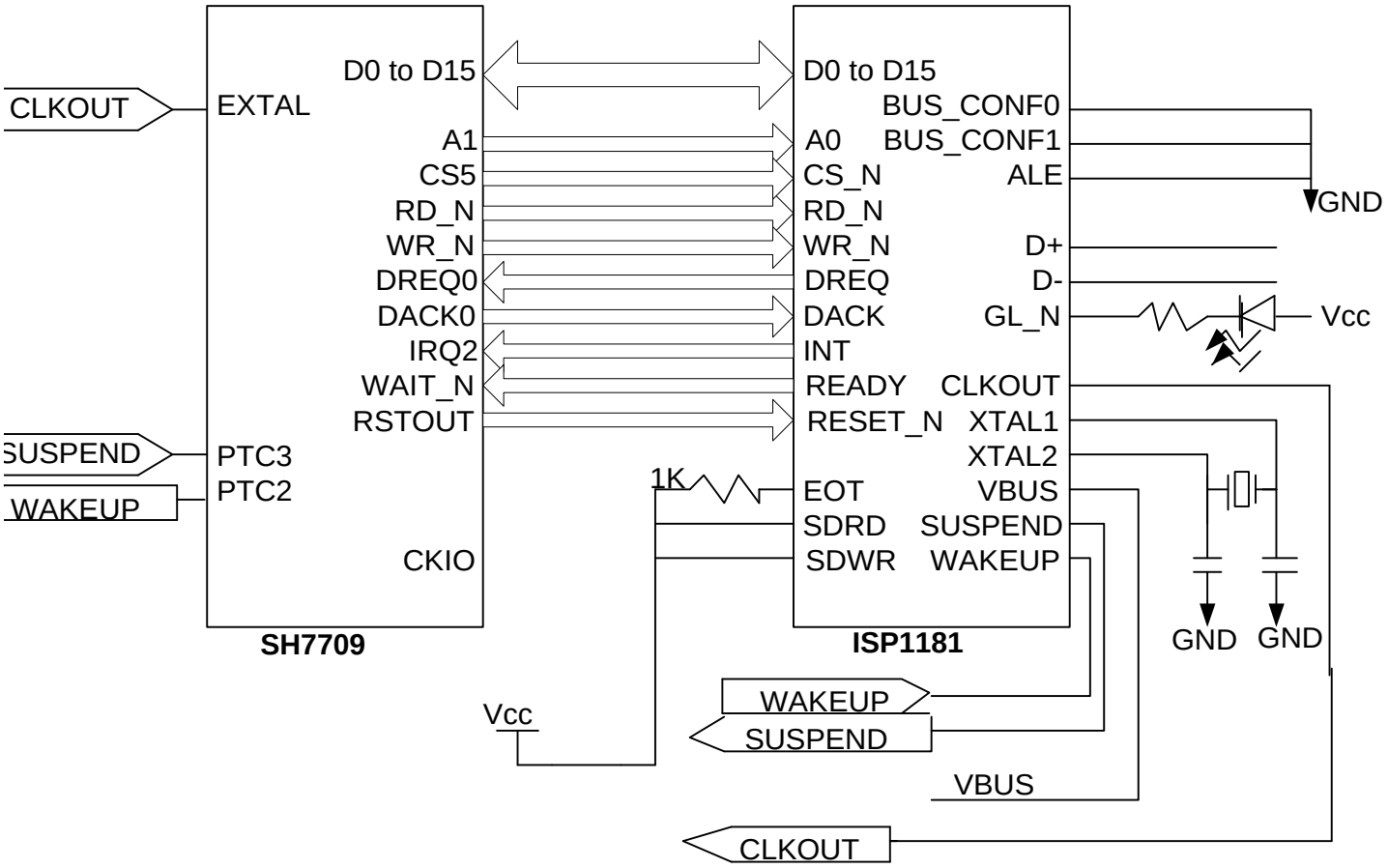


Figure: ISP1181 interface to SH7709 RISC processor