AN10037

Interfacing ISP1760; ISP1761 to the Intel PXA25x processor

Rev. 07 — 16 March 2010 Application note

Document information

Info	Content
Keywords	isp1760, isp1761; usb, universal serial bus, pxa250, pxa255
Abstract	This application note explains interfacing the ISP1760 and ISP1761 to the Intel PXA250 and PXA255 processors. **Remark**: PXA25x denotes the PXA250 and PXA255 processors.

ST ERICSSON

Revision history

Rev	Date	Description
07	20100316	Updated the filename according to the latest standards. Section 6.1: updated description for pin 117. Fig 4: updated pin 117 termination.
06	20090710	Rebranded to the ST-Ericsson template.
05	20090220	Rebranded to the ST-NXP Wireless template.
04	20060906	Fourth release: • Section 1: updated sixth paragraph. • Section 2.1: added two remarks. • Section 2.2: updated the content. • Section 2.3: updated content. • Rephrased content for clarity.
03	20051205	Third release. • Section 2.2: removed step 7, USBINTR register. • Section 2.3: added “OUT token” to the steps under Get Descriptor. • Added Section 10.
02	20051102	Second release. • Added Section 4 and Section 5. • Updated Section 2, Section 6 and Section 8. • Recreated schematics as per latest standards.
01	20041004	First release

Contact information

For additional information, please visit: **http://www.stericsson.com**

For document related queries, please send an email to: **wired.support@stericsson.com**

Remark: PXA25x denotes the PXA250 and PXA255 processors.

1. Introduction

The ISP1760/1 is a Hi-Speed Universal Serial Bus (USB) host controller that can directly be connected to a generic processor interface. The programmable CPU interface allows you to connect most of the RISC processors available, without additional glue logic. The data bus can be configured as 16-bit or 32-bit, ensuring direct connection to most of the RISC processors.

The ISP1760/1 is compatible with *Enhanced Host Controller Interface Specification Rev. 1.0* and provides three USB ports that support high-speed, full-speed, and low-speed modes.

The internal architecture of the ISP1760/1 contains an internal hub. The three available USB ports are the downstream ports of the internal hub.

An internal Transaction Translator (TT) is implemented to support full-speed and low-speed modes. This allows a cost-effective and simple architecture, avoiding the need of OHCI companion controllers.

The ISP1760/1 is a slave host controller. This means that it does not need access to the local bus of the system to transfer data, unlike in the case of the PCI Hi-Speed USB controllers. Therefore, data to be transferred will be moved to or from the ISP1760/1 internal memory by the system, as described in Section 2.

The access to the ISP1760/1 is ‘memory-mapped’ and data transfers can be done using either Programmed I/O (PIO) or Direct Memory Access (DMA). For the host controller application, if the processor supports DMA, it is good to use DMA.

The DMA of the ISP1760/1 is ‘slave-type’ that can generate DREQ when enabled, with configurable burst size and the maximum number of bytes transferred. If DREQ is not required by the system, the ISP1760/1 can be accessed by the system DMA in PIO mode and the ISP1760/1 DMA programming is not required. Programming the memory register is necessary only during memory read cycles.

Programming the ISP1760/1 for enumeration and data transfer to an external device involves the programming of:

- CPU interface configuration registers that mainly define the behavior of control signals on the interface to the host system.
- EHCI operational registers to initialize EHCI.
- Proprietary Transfer Descriptors (PTDs) found in the ISP1760/1 memory are dedicated memory areas defined at CPU address 0400h to 0FFFh.
- The payload area that contains data to be transferred (1000h to FFFFh).

The payload is located at predefined CPU address, 1000h to FFFFh, specified by the PTD. The ISP1760/1 implements 63 kB of internal memory, which consists of the payload area and PTDs. This memory size is optimized to balance the maximum USB performance, with minimal system loading and cost.

Address lines A[17:1] determine a total address space of 64 kB (16-bit or 32-bit addressing mode) of which 1 kB is occupied by the registers’ address space.

2. Initializing sequence of the ISP1760/1

The following sections explain the steps involved when initializing the ISP1760/1 to enumerate an external Hi-Speed USB device. The steps are listed in sequence and simply following the order and the example values listed should ensure the functionality described. Some simple changes may be necessary, for example, in the HW Mode Control register to match the concrete system implementation.

2.1 Programming the CPU interface

This section provides the steps necessary to program the CPU interface to set up control signals. Depending on your application, you may need to change the sequence. For details on programming the ISP1760/1, refer to the ISP1760 and ISP1761 data sheets. The typical sequence is:

1. SW Reset register (address: 030Ch)

   Example: 0000 0003h

   Result: Both the CPU interface and EHCI registers are reset. An internal reset pulse is generated and both bits are automatically cleared by the hardware.

   Remark: Software must wait in a while loop until the HCRESET bit is cleared.

2. HW Mode Control register (address: 0300h)

   Example: 0000 0125h

   Result: As a result:

   - Global interrupt enabled: IRQ will be asserted as soon as any of the enabled events occur.
   - IRQ is defined as ‘edge triggered’. A pulse of predefined width is generated when IRQ occurs.
   - IRQ is active HIGH.
   - DREQ is active HIGH. DREQ will become 1 when asserted.
   - DACK is active LOW. An active-LOW DACK is expected from the host system.
   - Data bus width is set for 32-bit mode.

3. HW Mode Control register (address: 0300h)

   Example: 8000 0125h

   Result: All ATX are reset.

   Remark: Wait for 15 ms.

4. HW Mode Control register (address: 0300h)

   Example: 0000 0125h

   Result: ATX reset condition is removed.

5. Interrupt Enable register (address: 0314h), as necessary

   Example: 0000 0004h

Result: An INT will be generated only when the DMA transfer is completed.

6. Edge Interrupt Count register (address: 0340h) can also be programmed with suitable values (optional).

   Example: 0000 FFFFh

   Result: The INT pulse width is approximately 1 ms. This is the maximum value that can be programmed using bits [15:0]. In this example, no minimum delay between two subsequent INTs is specified. If a delay is specified, only one INT is generated after the specified time because of the specified events. The Interrupt register contains cumulated INT events (not an INT queue). All bits corresponding to occurred events will be set, regardless of whether the respective bit sources in the Interrupt Enable register were enabled or not. The IRQ line will be asserted only if one of the enabled events has occurred.

7. DMA Configuration register (address: 0330h), if the ISP1760/1 DMA operation is required.

   Example: 0020 000Eh

   Result: As a result:

   - DMA is programmed for write to the ISP1760/1 RAM.
   - DMA is enabled.
   - A 16-cycle DMA burst is selected.
   - The transfer length is 8 kB.

8. Port 1 Control register (address: 0374h), applicable for configuration of port 1 only.

   Example: 0080 0018h

   Result: Port 1 internal multiplexer is switched to the host controller. The V_{BUS} of port 1 will be enabled as soon as the port power is enabled. V_{BUS} will be generated from the external power source.

2.2 Programming the internal EHCI

Programming the internal EHCI concludes with enabling the internal host controller port and connecting the internal hub. For details on programming registers, refer to the ISP1760 and ISP1761 data sheets, and *Enhanced Host Controller Interface Specification Rev. 1.0*.

This sequence must follow the steps mentioned in the preceding section, with minimum steps described as follows:

1. USBCMD register (address: 0020h)

   Example: 0000 0002h

   Result: Host controller is reset.

2. CONFIGFLAG register (address: 0060h)

   Example: 0000 0001h

Result: All ports are routed to EHCI.

3. PortSC1 register (address: 0064h)

   Example: 0000 1000h

   Result: Port power is enabled.

4. PortSC1 register (address: 0064h)

   Example: 0000 1100h

   Result: Port is reset.

   Remark: Wait for 50 ms.

5. PortSC1 register (address: 0064h)

   Example: 0000 1000h

   Result: Port reset condition is removed; port power is still present.

   Remark: Before proceeding to the next step, make sure that reset is completed. The PR (Port Reset) bit must be cleared by the EHCI hardware. For details, refer to *Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0*. At this stage, the Port Enable (PE) bit is set by the hardware.

6. USBCMD register (address: 0020h)

   Example: 0001 0021h

   Result: R/S = 1; ITC[7:0] = 01h.

7. ATL PTD Skip Map register (address: 0154h)

   Example: 1000 0000h

   Result: ATL Skip Map is set.

   Remark: Do not skip the current TD, which will be programmed.

8. ATL PTD Last PTD register (address: 0158h)

   Example: 1000 0000h

   Result: ATL Last PTD Map is set.

   Remark: Set the last TD at a position higher than the current.

2.3 Enumerating the internal hub

The internal hub is always connected on the internal root hub port. The internal hub connection can be verified by reading the PORTSC1 register, after the EHCI RESET condition is removed. For this case, no interrupt will be generated.

This section covers enumerating the internal hub, enabling downstream ports, and enabling the port power of these ports. The enumeration of the internal hub means sending standard USB hub configuration commands. These configuration commands

contain a series of standard requests. Each command has a set-up token phase and an IN token phase.

To achieve correct communication with the internal hub, the PTD must be correctly programmed. The PTD area is situated in memory at address 0400h to 0FFFh, just above the register addressing space. The PTD area consists of Isochronous (ISO), Interrupt (INT), and Asynchronous Transfer List (ATL). Also, for the set-up token, a certain payload will be used, specific to each request. For details, refer to the PTD description in the data sheet.

The set-up token requires the following PTD settings:

0000 0800h—DW1

1003 8000h—DW2

8380 0000h—DW3

0000 0000h—DW4

0000 0000h—DW5

0000 0000h—DW6

0000 0000h—DW7

2100 0041h—DW0

DW0 contains the number of bytes to be transferred. It must be the last value written because it sets PTD valid bit V, which enables the PTD execution. Alternatively, the Skip Map register may be set to skip the current PTD bit first, and then unskip after completing the PTD initialization.

DW1 contains the device address and the endpoint. It must be adjusted after setting the hub address. For example, after Set Address, DW1 must be changed to 0000 0810h. The same applies to the IN token. These are called **set-up token 2** and **IN token 2** in the following example.

DW2 contains Set Address, that is, the ISP1760/1 internal memory address at which the payload must be placed for an OUT transfer and from where data will be retrieved after an IN transfer.

These eight double words of data must be written to the ATL PTD memory area. For example, starting at CPU addresses C00h, C20h, C40h, and so on.

The payload preparation is performed first at the memory address specified in the PTD DW2 and then the PTD is set up. The payload data is the data that will be sent to the internal hub during the hub enumeration and ports power phase. The payload address specified in the PTD is an internal ISP1760/1 address calculated according to the formula given in Section 7.2.2 of the ISP1760 and ISP1761 data sheets. For example, the ISP1760/1 internal address 380h in DW2 corresponds to the external system or CPU address 2000h. As a result, the system CPU must write the payload data at ISP1760/1 internal memory address 2000h. Once the PTD is correctly set up and the payload data is ready, valid bit V in DW0 will be set, which will enable the execution of the PTD. Alternatively, the Skip Map register may be used to disable or enable the PTD execution.

DW3: In the example, the Data Toggle bit is set to 1 by software. When it comes to the set-up token, the Data Toggle bit must be set to 0. The Token[1:0] field, however, takes precedence over the Data Toggle bit. If the Token[1:0] field must be configured to be a set-up token then, regardless of the value in the Data Toggle bit, data sent out will be on Data Toggle 0.

Similarly, the IN token requires the PTD area to be correctly set up. The IN token PTD setting are:

2100 0200h: DW0

0000 2400h: DW1

1003 8100h: DW2

8380 0000h: DW3

0000 0000h: DW4

0000 0000h: DW5

0000 0000h: DW6

0000 0000h: DW7

2100 0201h: DW0

Similar to the set-up token, the last line sets the V bit, enabling the execution of the IN token PTD.

Get Descriptor

Set-up token payload:

Lower 32: 0100 0680h (At the ISP1760/1 internal memory address 2000h.)

Upper 32: 0012 0000h (At the ISP1760/1 internal memory address 2004h.)

IN token

OUT token

Set address:

Set-up token payload:

Lower 32: 0002 0500h (At the ISP1760/1 internal memory address 2000h.)

Upper 32: 0000 0000h (At the ISP1760/1 internal memory address 2004h.)

IN token

Set Hub Configuration:

Set-up token 2 payload:

Lower 32: 0001 0900h

Upper 32: 0000 0000h

IN token 2

Set_Feature (Port_Power #2)

Set-up token 2 payload:

Lower 32: 0008 0323h

Upper 32: 0000 0002h

IN token 2

Clear_Feature (C_Port_Connection Port #2)

Set-up token 2 payload:

Lower 32: 0010 0123h

Upper 32: 0000 0002h

IN token 2

Set_Feature (Port_Reset #2)

Set-up token 2 payload:

Lower 32: 0004 0323h

Upper 32: 0000 0002h

IN token 2

Set-up token 2 is the same as the set-up token example, except that the value of DW1 will be 0000 0810h, as mentioned earlier. The same applies to IN token 2.

These steps listed will enable downstream port 2 and V_{BUS} will be present on this port, that is, port power is enabled.

If a USB device is connected when port 2 is reset, chirp will be generated and the port will be automatically switched to the speed corresponding to the USB device connected to the respective downstream port. The chirp sequence is hardware-controlled and is initiated by the device as a response to the host controller port reset. μSOF will be generated each 125 μs on port 2 at this stage on the downstream port, if a high-speed device is connected and the chirp sequence is successful. Complete enumeration of an external USB device can be consequently performed.

3. ISP1760/1 DMA or PIO data transfers programming

3.1 Microprocessor DMA: ISP1760/1 DMA case

This section explains the programming of the ISP1760/1 DMA in the microprocessor DMA, ISP1760/1 DMA case.

1. Program the DREQ and DACK active-level polarity in the HW Mode Control register.

   Example: 300h ← 125h

   Action:

   GLOBAL_INT_EN: An IRQ will be generated on any enabled event.

   IRQ is level-triggered.

   IRQ is active HIGH.

   DREQ is active HIGH.

   DACK is active LOW.

   CPU interface is set to 32-bit data bus width.

   IRQ, DREQ, and DACK of the peripheral controller are routed to the respective peripheral controller dedicated pins. Common mode is not selected.

Digital overcurrent scheme is implemented.

No ATX is reset.

2. Set the correct DMA start address in the DMA Start Address register.

   Example: 344h ← 400h

   Action: The system DMA must start the transfer from address 400h. This is the external address of the actual CPU memory.

   Remark: Usually memory addresses above 1000h will be specified in this register because the payload area is located above address 1000h.

3. Program the system DMA according to ISP1760/1 DMA settings.

   Programming steps are system-specific. All system DMA parameters must match the ISP1760/1 DMA programmed settings: signal's active polarity, DMA burst length, transfer count, start address, and so on.

   The system DMA can be enabled at this step, if configured to wait for the ISP1760/1 DREQ. If the system DMA will not wait for the ISP1760/1 DREQ, then this step must be performed after step 4.

4. Set the HcDMAConfiguration register.

   Example: 330h ← 0020 000Bh

   Action:

   Selected DMA READ, from the ISP1760/1 memory.

   DMA enabled, DREQ is immediately asserted.

   Selected 8-cycle DMA burst.

   Programmed 8-kB data transfer.

Fig 1 shows the timing diagram of a typical DMA data transfer, corresponding to these settings.

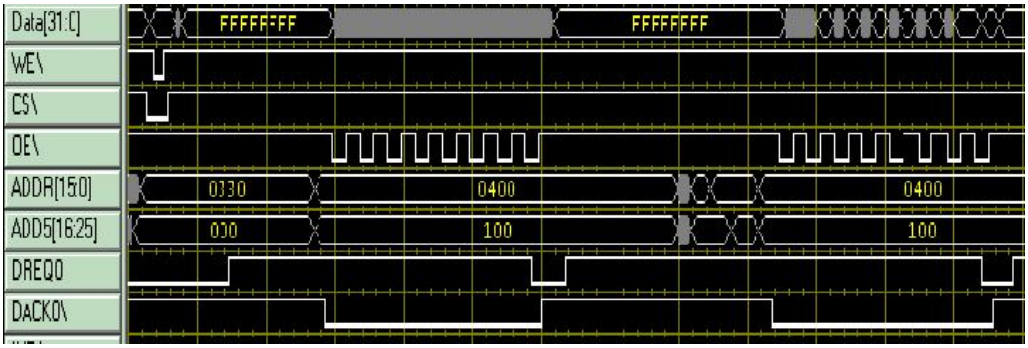

Fig 1. Typical DMA data transfer

It is observed that DREQ is immediately asserted after the ISP1760/1 DMA is enabled and the system will assert DACK in response, executing a burst according to the programmed parameters. DREQ will be deasserted on the last write or read cycle of each burst, and will be reasserted, until the programmed transfer count is completed.

If the IRQ generation at the end of the DMA transfer is necessary, the ISP1760/1 IRQ can be enabled at the End-Of-Transfer (EOT) in the Interrupt Enable register. For example: for 08h, only the DMA EOT IRQ is enabled.

3.2 Microprocessor DMA: ISP1760/1 PIO case

You can obtain a better data transfer rate on the extension bus in this mode by eliminating the DREQ and DACK latencies, and also because of the ISP1760/1-specific timing.

In this case, the ISP1760/1 DMA programming is unnecessary. The system DMA will access the ISP1760/1 memory as in a typical memory-to-memory access. It is not necessary to connect the DREQ signal to the DACK signal. A 10 kΩ pull-up resistor must be placed on the ISP1760/1 DACK input signal, if not used.

The system must generate CS_N instead of DACK, as during a normal PIO access.

The ISP1760/1 will not generate IRQ on completing DMA because there is no transfer count programmed. The system DMA, however, will stop on its programmed transfer count and an internal processor INT on EOT can usually be generated.

There is no preparatory programming necessary in the case of a write to the ISP1760/1 memory.

Special attention must be given when reading from the ISP1760/1 memory because the 33Ch register must be programmed with the correct read start address, before starting the memory read operation. As long as the read address is contiguous, it is not necessary to program register 33Ch again.

For details, refer to the ISP1760 and ISP1761 data sheets.

4. Suspend, resume, and power management

Suspend mode is achieved by appropriate programming of ISP1760/1 EHCI registers and sending right commands to the internal hub. This will determine when internal ISP1760/1 clocks will stop, leaving only an internal, low-frequency, dedicated LazyClock running to detect events that may require a wake-up.

The Power Down Control register of the ISP1760/1 allows power down of the ISP1760/1 blocks to be programmed for maximum power saving during suspend. Programming these bits will disable certain blocks. Consider the consequences of disabling functionality when you perform bit settings. The events that may produce a wake-up are:

- CS_N assertion, usually determined by any access to ISP1760/1 registers or memory.
- A plug or unplug of a USB device.
- Driving LOW the SUSPEND/WAKEUP_N pin.

The SUSPEND/WAKEUP_N pin is a dual-function pin and must usually be connected to an available GPIO pin on the processor. It initially indicates the suspend or awake state of the ISP1760/1 but can also produce the ISP1760/1 wake-up, if driven or forced LOW during suspend (open-drain).

A resume because of any of the defined wake-up events can generate a CLKRDY_IRQ that must be enabled in the Interrupt Enable register, if an IRQ assertion is needed. The IRQ assertion will show that internal CLK signals are restored and normally running for the default 10 ms time-out.

For details on register programming, refer to the ISP1760 and ISP1761 data sheets.

5. ISP1761 port 1 V_{BUS} enabling in various functional modes

When port 1 is defined as a host controller (pure host): The port power, 5 V on the V_{BUS} pin of the socket, is achieved by 'SetPortFeature PORT1_POWER' request sent to the internal hub. Issuing SetPortFeature to the internal hub will result in PSW1_N turning on the external transistor to provide a 5 V V_{BUS} to the USB socket. The OTG Control register must already be programmed as 41Eh (OTGDISABLE | VBUS_DRV | SEL_CP_EXT | DP_PULLDOWN | DM_PULLDOWN). After 41Eh is programmed into the OTG Control register, PSW is **not** turned on to provide power to the V_{BUS} socket. The setting of 41Eh corresponds to choosing an external power source and enabling it; instead of using the internal charge pump, which is limited to 50 mA. The external power source may be implemented using either analog, for example, PMOS, or digital, for example, MIC2026, overcurrent scheme. In this scenario, the port power, 5 V on the V_{BUS} pin, will be enabled only after 'SetPortFeature PORT1_POWER' is completed.

If port 1 is in OTG mode: The ISP1761 is a B-device and SRP must be performed: This is achieved by alternately setting and resetting the VBUS_CHRG and VBUS_DISCHRG bits of the OTG Control register. This will pull up V_{BUS} to 3.3 V through an internal resistor, generated by the internal 3.3 V regulator, or directly pull down V_{BUS} to GND for faster discharge.

Details on SRP can be found in Section 9.3 of the ISP1761 data sheet. Do not enable port power; that is, do not turn on VBUS_DRV.

Port 1 is in OTG mode: It is an A-device and acts as a host controller: Port 1 power setting, 5 V on the V_{BUS} pin, is similar to when port 1 is defined as a pure host. The internal charge pump or the external power source can be selected according to the SEL_CP_EXT bit setting. The external power source must be implemented with the analog overcurrent scheme, using an external PMOS, to allow V_{BUS} sensing through the OC1_N pin directly connected to V_{BUS}. For example, as a host the ISP1761 will need to sense the V_{BUS} pulsing from a B-peripheral.

V_{BUS} sensing for the V_{BUS} pulsing will not be possible with a MIC2026 implementation because V_{BUS} and OC1_N cannot be connected together; see Fig 2. Therefore, the ISP1761 will be unable to monitor the V_{BUS} pin because the B-peripheral pulses V_{BUS} to perform SRP.

To achieve port power and normal functionality, you must perform the internal hub enumeration and 'SetPortFeature PORT1_POWER'.

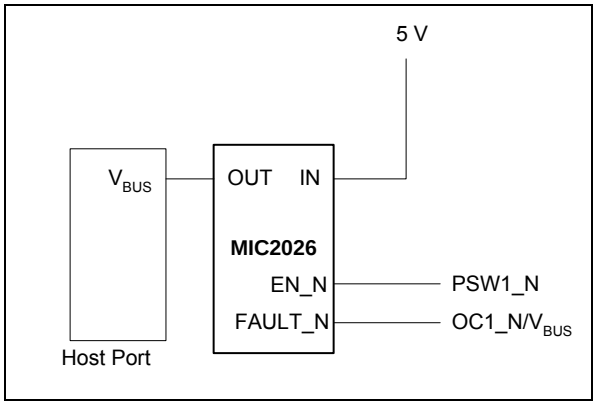

Fig 2. Using digital overcurrent switch

Port 1 is in OTG mode: It is an A-device and acts as a peripheral: In this case, the B-device acts as a host but the OTG A-device is still the one generating V_{BUS}. The peripheral controller is internally routed to port 1, ensuring port 1 power just by programming the OTG Control register.

Remark: 'SetPortFeature PORT1_POWER' is not required.

The required bit settings are:

SW_SEL_HC_DC = 1

VBUS_DRV = 1.

The internal charge pump or the external power source can be selected by setting the SEL_CP_EXT bit. The external power source must be implemented with the analog overcurrent scheme, using an external PMOS, to allow V_{BUS} sensing through the OC1_N pin directly connected to V_{BUS}; see Fig 3.

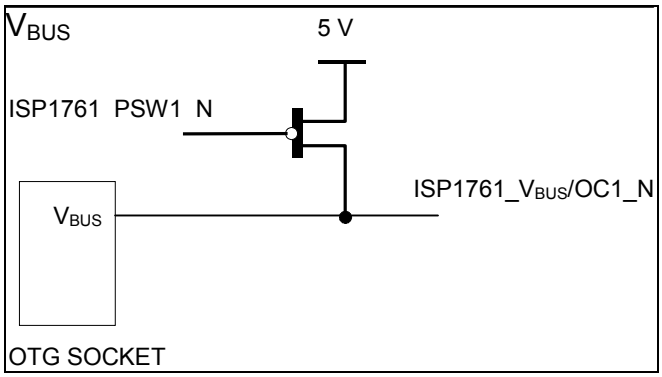

Fig 3. Using external PMOS

As an A-device acting as a peripheral, V_{BUS} is needed because the V_{BUS} sensing will not be possible with a MIC2026 implementation. This is because V_{BUS} and OC1 cannot be connected together. See Fig 2.

6. Schematics description

6.1 ISP1760/1 pin connection and general description

The ISP1760/1 can be accessed on either a 16-bit or 32-bit data bus, address line A0 is never used because 8-bit mode access is not possible.

In the case of a 16-bit data bus, address line A1, the first address line used, must be connected to system's A1.

In the case of a 32-bit data bus, address line A2, the first address line used, must be connected to system's A2. A1 is not used and can be connected to system's A1 or GND.

The ISP1760 and ISP1761 have the same packaging and pinout. Therefore, ISP1761-specific pins that are not used in the ISP1760 are defined as test pins and must be connected as follows:

- Pin 3 GND
- Pin 81 GND
- Pin 117 must be connected to $V_{CC(I/O)}$ through a 10 kΩ
- Pin 120 pull-up to $V_{CC(I/O)}$
- Pin 124 connect a 220 nF capacitor between this pin and pin 125
- Pin 125 connect a 220 nF capacitor between this pin and pin 124
- Pin 126 connect to 3.3 V

When the ISP1760/1 is connected to a system with a 16-bit data bus and the upper 16 data lines are not used, these data lines must be connected to 3.3 V through a 15 kΩ pull-up resistor. A single external resistor can connect upper 16 data lines together.

The respective OCn_N pin is also not used. Therefore, the pin must be pulled up using a 10 kΩ resistor.

The ISP1760/1 has separate power pins for the digital CPU interface, namely pins 10, 40, 48, 59, 67, 75, 83, 94, 104 and 115, and separate power inputs for the digital core and analog blocks, namely pins 6 and 7. Internal linear voltage regulators, 5 V to 3.3 V (analog) and 5 V to 1.8 V (digital core), are powered using the power source on pins 6 and 7.

If a stable 3.3 V power source that can supply a minimum additional current of 150 mA is already available in the design, pins 6 and 7 can directly be powered by 3.3 V. This will reduce the power consumption, avoiding unnecessary power dissipation in the ISP1760/1 internal power supply block.

Pin 9 is the output of the ISP1760/1 internal linear regulator (5 V to 3.3 V) and requires decoupling capacitors of 100 nF and 4.7-µF to-10 µF to be placed as close as possible.

Pins 5, 50, 85, and 118 are connected to the output of the ISP1760/1 internal linear regulator (5 V to 1.8 V). Each pin must be connected to a 100 nF decoupling capacitor, placed as close as possible to it. An additional 4.7 µF-to-10 µF capacitor must be connected to any one of these pins.

Similarly, there are separate GND connections. In a concrete design, these can be connected together to a single GND plane.

Depending on your design, it may be a good idea to connect reference GND pins 15, 22, and 29 to GND through a properly selected inductor, if isolation of potential noise from the local GND plane is necessary.

It is recommended that you place ESD protection components on each USB port.

If the DC_IRQ line is connected to one of the system's IRQ lines and you intend to use it, program correct polarity for both the ISP1760/1 and the processor, even if, only the host controller may be used in the first phase and the peripheral does not need programming. Maintaining default polarity settings may produce signal contention. An undesirable result may be a higher current consumption in suspend mode.

6.2 Overcurrent

The ISP1760/1 supports both analog and digital overcurrent schemes. Both solutions are implemented in the schematics example, considering components can be optionally soldered for one solution or the other. For example, the analog solution means soldering U15, R30, R33, and C43 but not MIC2026 and R21.

The analog overcurrent detection is achieved by sensing the voltage drop on PMOS transistor U15. This will automatically deassert the PSW2_N signal, which will cut off the V_{BUS} voltage on the respective port. The voltage drop detection is in the range of 45 mV to 90 mV.

For an overcurrent limit of 500 mA, a PMOS transistor with R_{DSON} = 100 mΩ must be selected. The overcurrent limit can be adjusted by modifying the series resistor. For example: R30 for port 2 in the schematics.

The digital overcurrent scheme requires a standard power switch with integrated overcurrent detection, such as:

- LM3526 and MIC2526 (two ports) or
- LM3544 (four ports)

The advantage of these integrated power control switches is that they implement thermal shutdown and internal filter of 1 ms to 3 ms to prevent false overcurrent reporting that may appear because of inrush currents, when plugging a USB device.

The digital overcurrent protection logic of the ISP1760/1 uses the following two pins for power switching and overcurrent protection of each USB port:

- PSWn_N: This pin will enable or disable the respective external power switch (MIC2526 and LM3526).
- OCn_N: This is an input on which the respective port power device will signal a fault condition.

For both the analog and digital overcurrent schemes, the port power will be disabled, that is, PSWn_N is deasserted, once an overcurrent or other fault condition is detected on the respective port.

A possible alternative is to use a ‘resettable fuse’ (or polyswitch PTC device) on each port or one for 1 to 3 ports.

The advantage of this solution is cost but it will not allow a permanent cutoff of the port power in the case of an overcurrent event. It will protect the port by switching on and off, as long as the overcurrent condition persists.

6.3 Specific schematics elements

This section explains specific elements in the schematics related to connecting the ISP1760/1 to the Intel PXA25x processor. These schematics provide a concrete example on connecting the ISP1760/1 to the extension bus of the BSQUARE PXA25x development system.

The ISP1760/1 must be defined as 'Variable Latency I/O' in the MSC register of PXA25x because it does not operate on a synchronous interface, but without the implementation of the READY signal.

If necessary, additional wait states defined in PXA25x register MSC0-2 will increase the access time.

The maximum data transfer that can be obtained in a system depends not only on the 'active time' pulse but also on the cycle-to-cycle 'inactive time'. This is determined by the concrete setting possibilities of each system: DMA channel priority, system loading, and execution of other parallel tasks. For example, the timing diagram in Fig 1 shows that the efficiency of the DMA transfer can sensibly be affected by the DACK latency.

Although after a burst of 16, DREQ is immediately asserted again to complete the transfer, DACK is asserted only after some time, depending on the DMA channel priority, SDRAM speed, and other tasks running. This will reduce the maximum data transfer rate and is strictly related to the system's capability to generate DACK.

During the ISP1760/1 memory DMA accesses, only DACK must be active. Perform WR_N or RD_N according to the cycle in progress.

7. PCB design recommendations

Some important checks for a successful PCB design are:

- Typically, a solution using four-layer PCB, signal 1, GND, V_{CC}, and signal 2, is sufficient for proper routing, allowing you to obtain good functionality and meeting all compliance tests requirements. Start your design by placing the ISP1760/1 chip, the major components, and routing of the high-speed DP and DM traces, and clock traces. Also, a complete 'clean' solution to route the power and GND (plane split) must be defined before you start routing the rest of the signals.
- Route the high-speed USB differential pairs over continuous GND or power planes. Avoid crossing anti-etch areas and any breaks in internal planes (plane split). The minimum recommended distance to a plane split is 25 mils. Also, avoid placing a series of VIA holes near the DP and DM lines because these will create 'break areas' in the GND plane below. This is because of the clearance imposed by the manufacturing process around any VIA holes to an internal plane.
- Try to keep the length of the DP and DM traces equal. The maximum trace length mismatch between Hi-Speed USB signal pairs must not be greater than 70 mils.
- Maintain parallelism between USB differential signals, with the trace spacing needed to achieve 90-Ω differential impedance. To achieve the required impedance of the pair traces, it is recommended that you use 8 mils traces and keep the distance between the DP and DM traces at 8 mils. These values may vary, depending on actual PCB parameters.
- Avoid corners when routing the differential pair DP and DM. Any 90° direction change of traces must be accomplished with two 45° turns or by using an arc of an imaginary circle tangent to the DP and DM lines.
- Avoid routing USB differential pairs near I/O connectors, signal headers, crystals, oscillators, magnetic devices, and power connectors.
- Maintain the maximum possible distance between high-speed USB differential pairs, high-speed or low-speed clock, and nonperiodic signals. The minimum recommended distances are as follows:
  - 20 mils between the DP and DM traces and low-speed nonperiodic signal traces.

  - 50 mils between the DP and DM traces and clock or high-speed periodic signal traces.
  - 20 mils between two pairs of the DP and DM traces.
- Avoid creating stubs to connect 15 kΩ pull-down resistors or to test points. If a stub is unavoidable in the design, no stub must be greater than 80 mils.
- Route all the DP and DM lines on one layer. Do not change layers (avoid using VIAs) even to avoid crossing a plane split. It is better to place a non-split plane under high-speed USB signals, ground layer, or power layer. It is recommended that you place ground layer beneath the DP and DM lines.
- The maximum allowed length of the DP and DM lines for onboard solutions (or [trace + cable length] for a front-panel solution) is 18 inches.
- A decoupling capacitor must be placed on V_{BUS} as close as possible to each USB connector. A value of about 150 µF / 10 V is recommended on each port.
- The decoupling capacitors must be placed as close as possible to the ISP1760/1. A good choice is the four corners of the IC because these areas will not normally be occupied by traces or other components, according to the ISP1760/1 pinout.
- For good EMI testing results, it is recommended that you provide a good path from the USB connector shell to the chassis ground. The USB connector shell must be connected to an isolated ground plane.

For more information, refer to Intel document *The USB 2.0 Platform Design Guideline, Rev. 1.0.*

8. Schematics

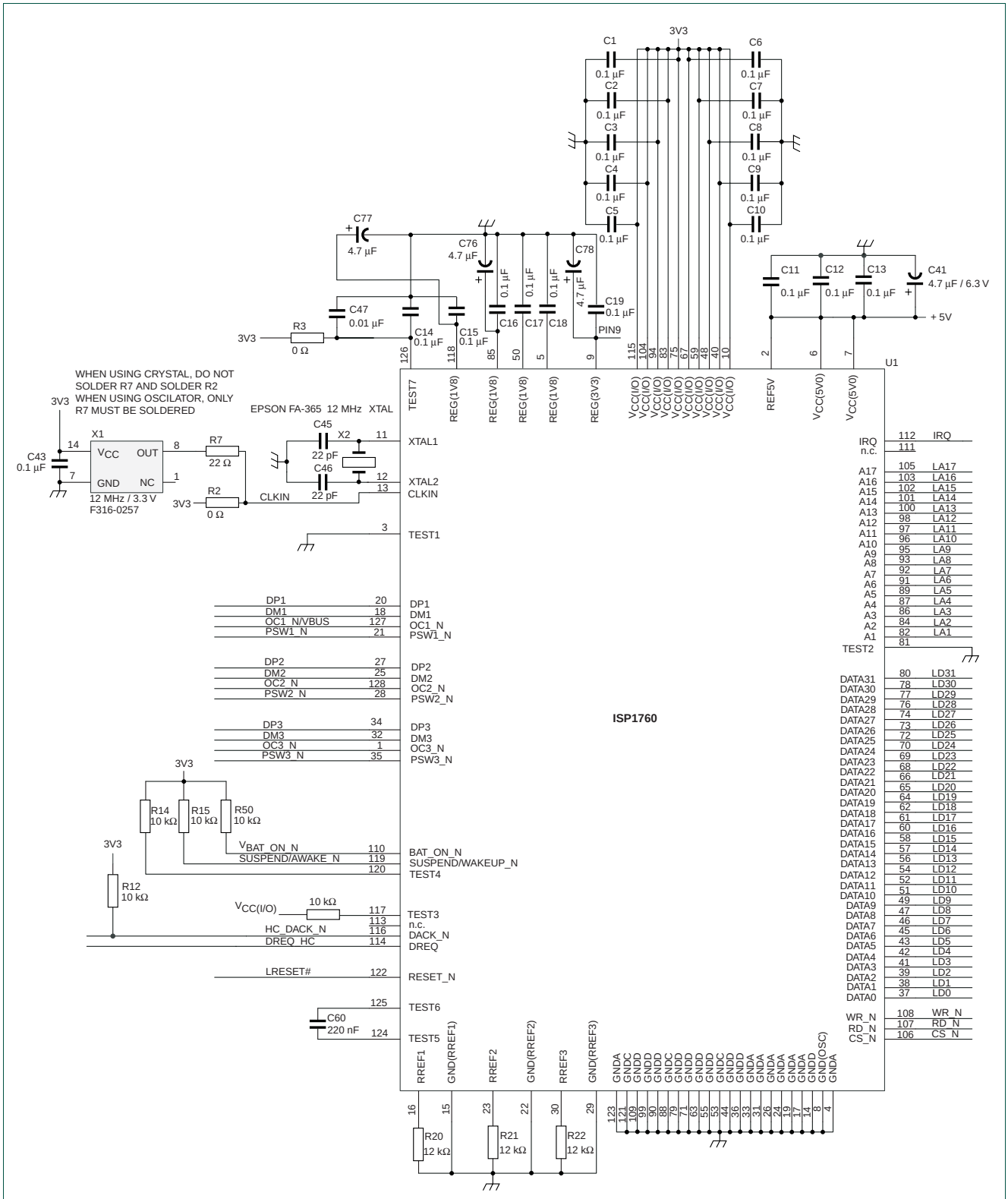

Fig 4. ISP1760

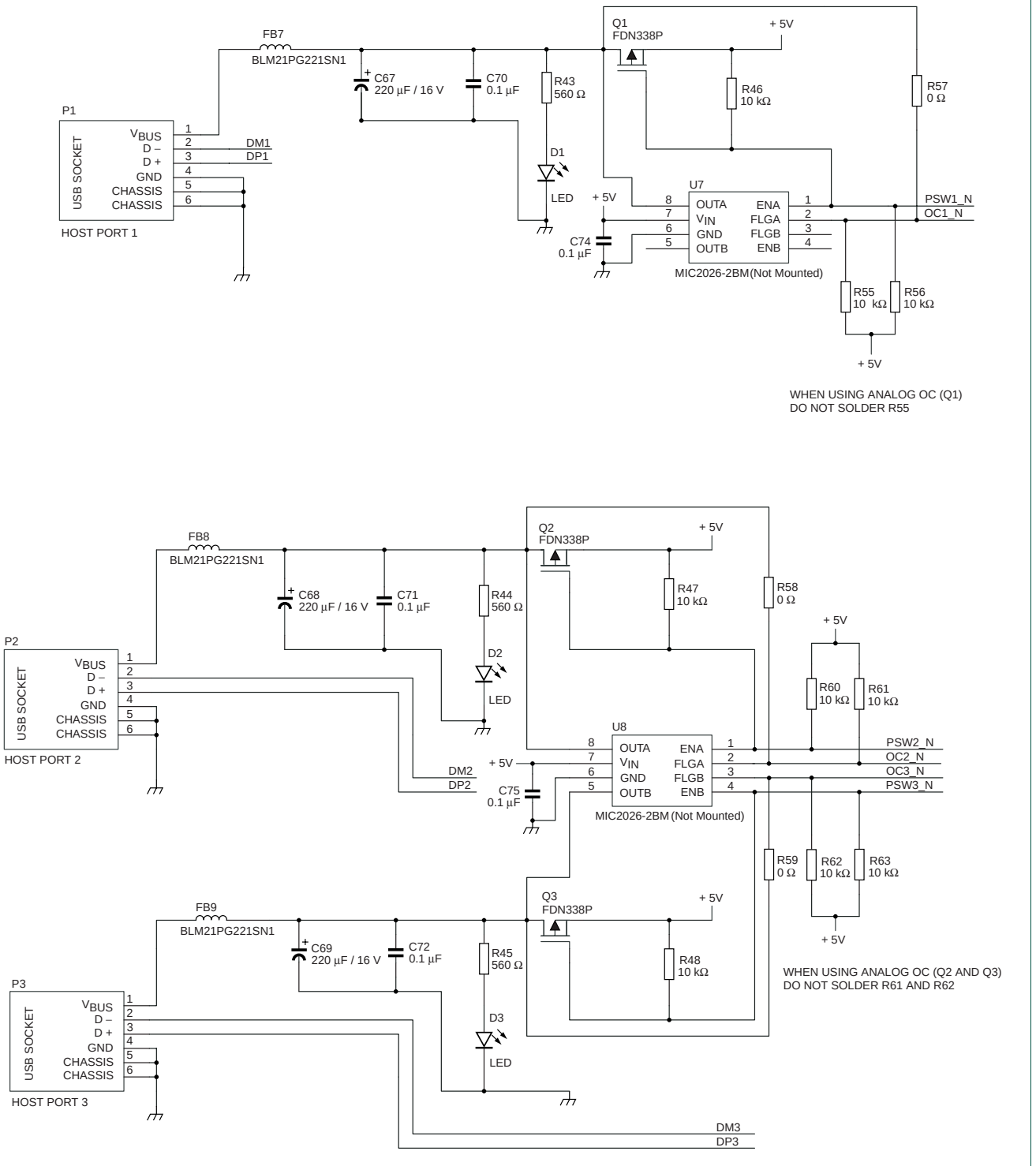

Fig 5. ISP1760 USB port interface

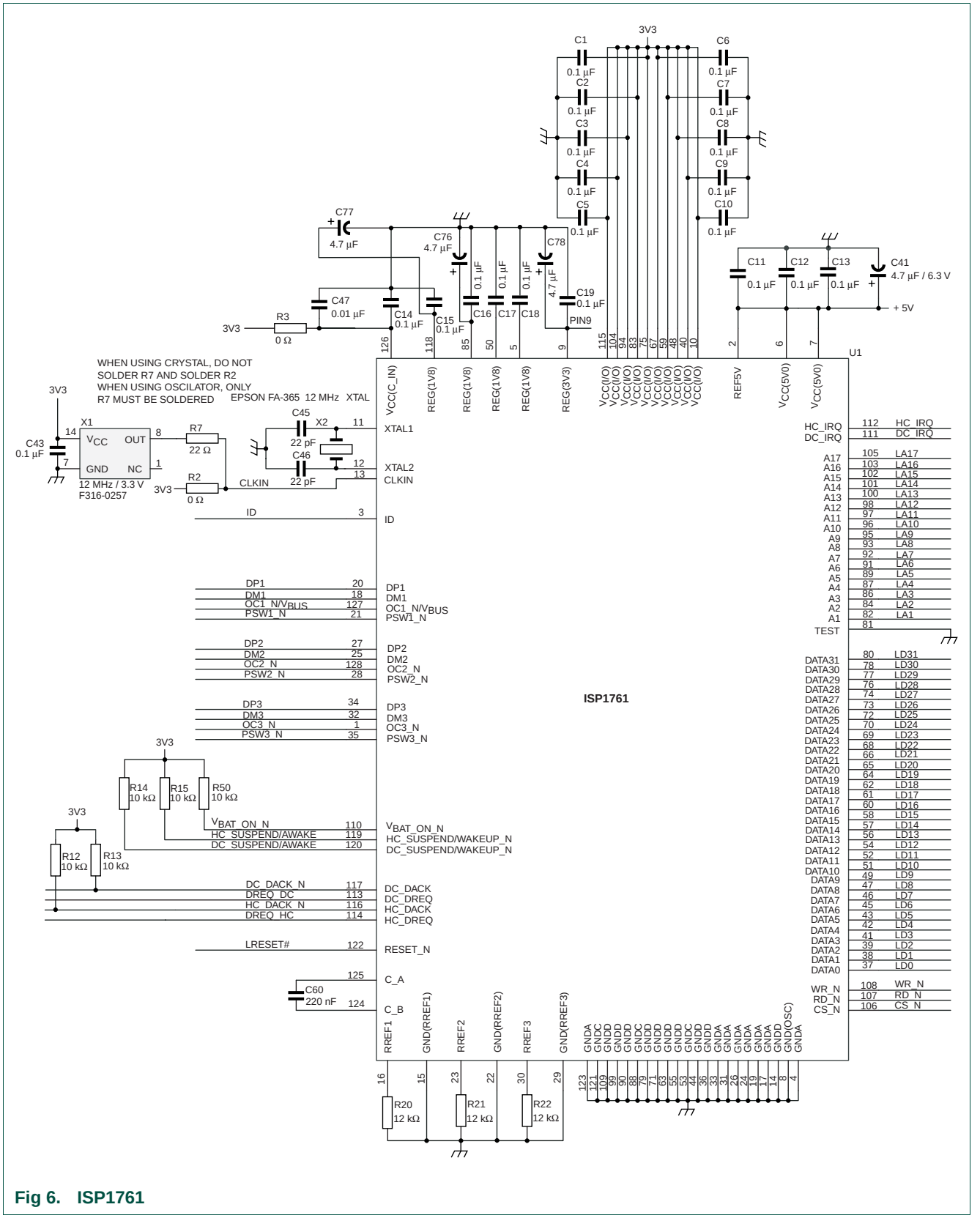

Fig 6. ISP1761

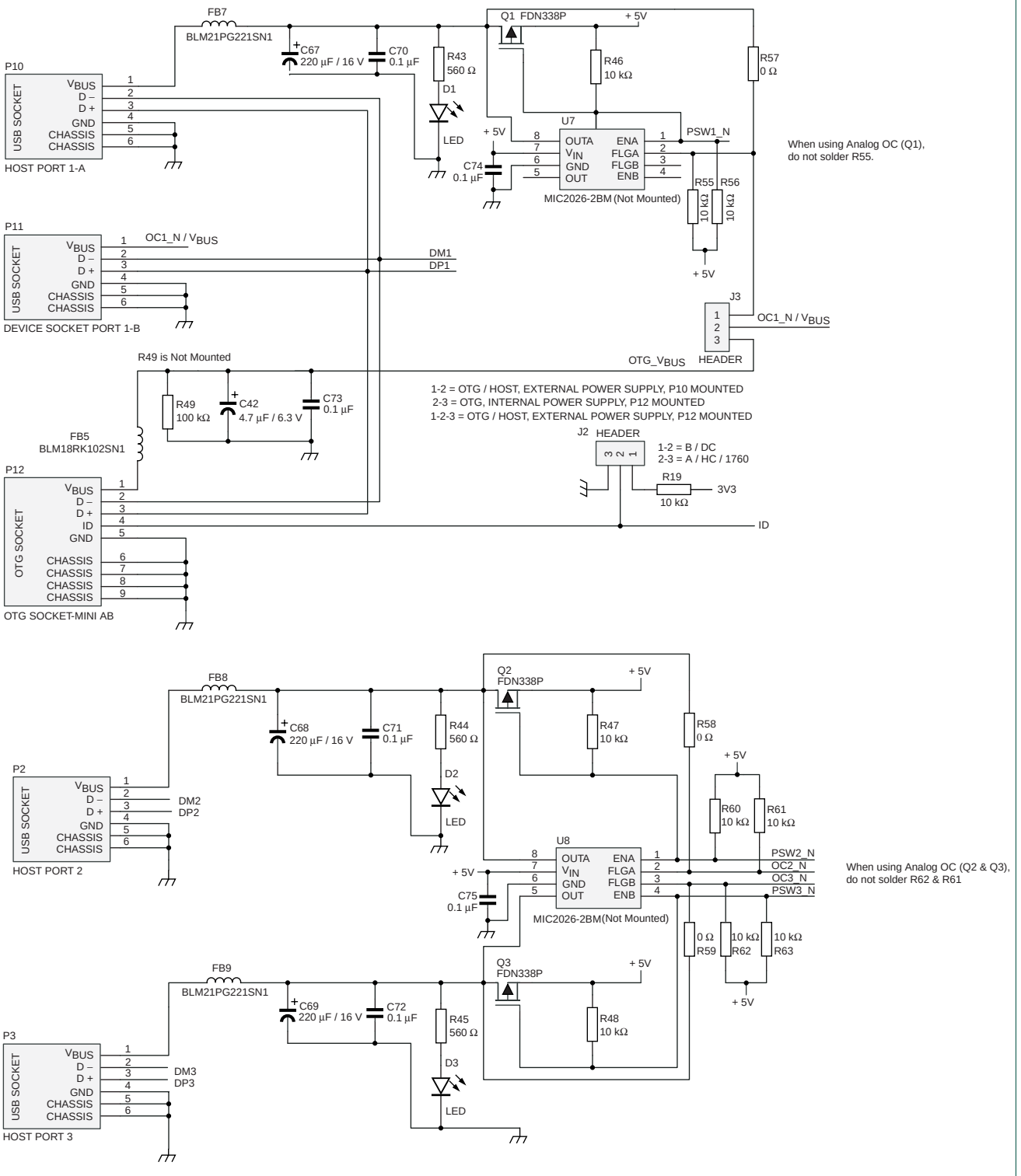

Fig 7. ISP1761 USB port interface

9. Pinning configuration

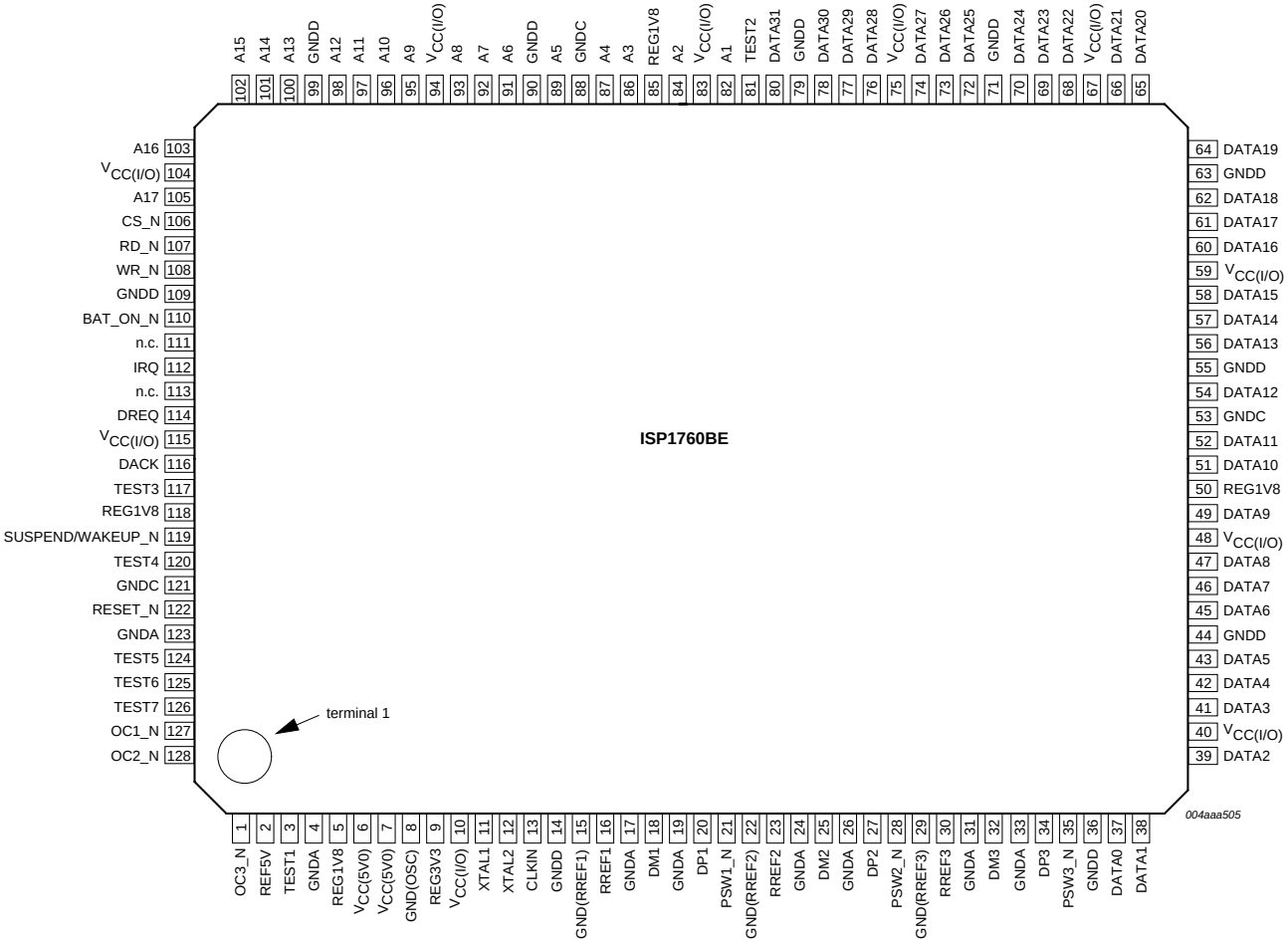

Fig 8. ISP1760 pinning diagram

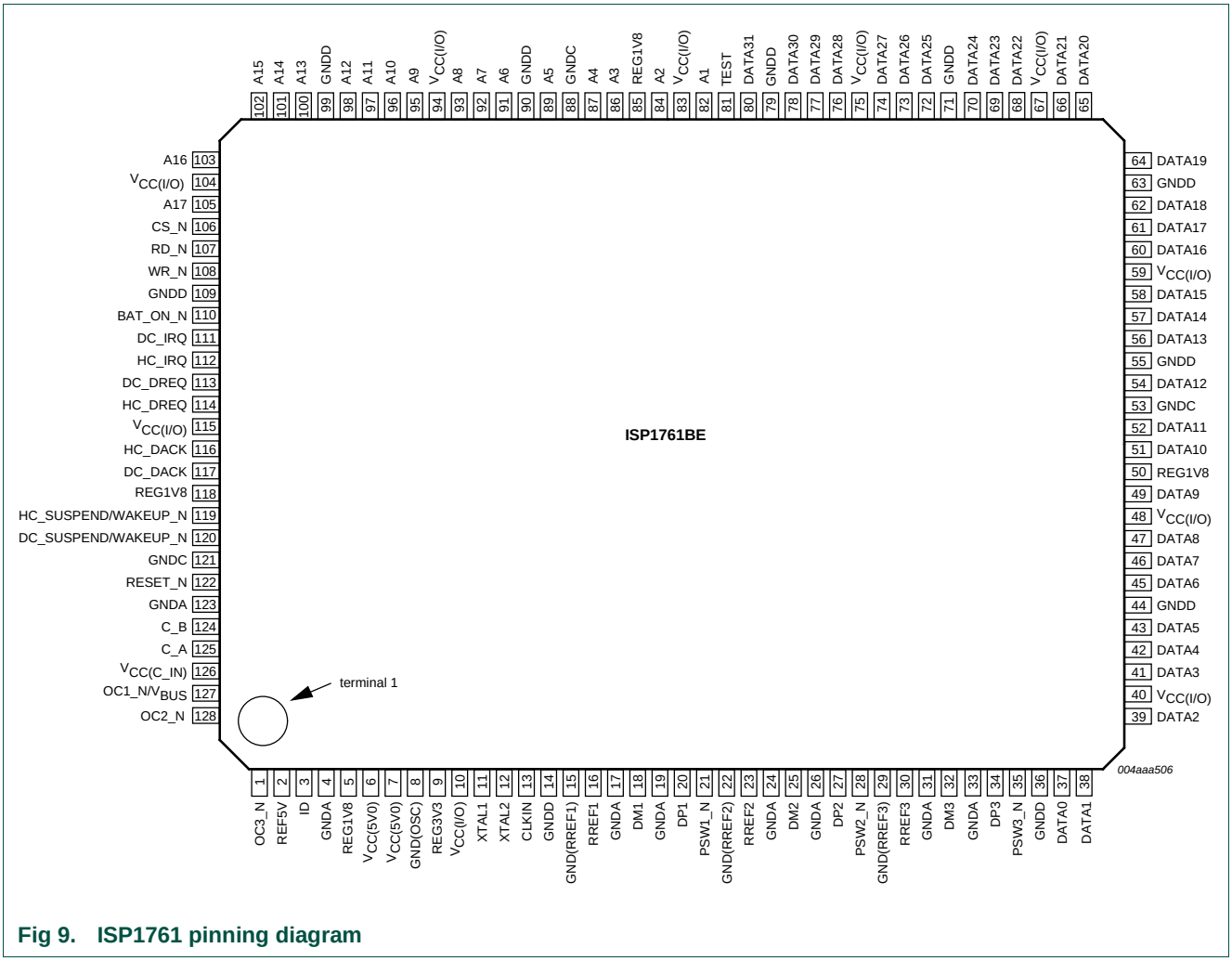

Fig 9. ISP1761 pinning diagram

10. Abbreviations

Table 1. Abbreviations

Acronym	Description
ATL	Asynchronous Transfer List
ATX	Analog USB Transceiver
CPU	Central Processing Unit
DMA	Direct Memory Access
EHCI	Enhanced Host Controller Interface
EOT	End-Of-Transfer
ESD	ElectroStatic Discharge
IC	Integrated Circuit
INT	Interrupt

Acronym	Description
ISO	Isochronous
OHCI	Open Host Controller Interface
PCB	Printed-Circuit Board
PCI	Peripheral Component Interconnect
PIO	Programmed I/O
PMOS	Positive-channel Metal-Oxide Semiconductor
PTD	Proprietary Transfer Descriptor
RISC	Reduced Instruction Set Computer
SRP	Session Request Protocol
TT	Transaction Translator
USB	Universal Serial Bus

11. References

[1] Universal Serial Bus Specification Rev. 2.0

[2] ISP1761 Hi-Speed USB On-The-Go controller data sheet

[3] ISP1760 Hi-Speed USB host controller for embedded applications data sheet

[4] The USB 2.0 Platform Design Guideline, Rev. 1.0

[5] Enhanced Host Controller Interface Specification Rev. 1.0

12. Legal information

Please Read Carefully:

The contents of this document are subject to change without prior notice. ST-Ericsson makes no representation or warranty of any nature whatsoever (neither expressed nor implied) with respect to the matters addressed in this document, including but not limited to warranties of merchantability or fitness for a particular purpose, interpretability or interoperability or, against infringement of third party intellectual property rights, and in no event shall ST-Ericsson be liable to any party for any direct, indirect, incidental and or consequential damages and or loss whatsoever (including but not limited to monetary losses or loss of data), that might arise from the use of this document or the information in it.

Contact information at www.stericsson.com under Contacts

www.stericsson.com

13. Contents

For more information, please visit: http://www.stericsson.com
For document related queries, email to: wired.support@stericsson.com

Date of release: 16 March 2010
Document identifier: CD00222772