Table of Contents

Introduction ........................................................................................................... 3

1 Configuration Options .......................................................................................... 4

2 Peripheral Signals Assignment Table ................................................................. 5

3 Connectivity Preview ............................................................................................ 6

4 Resource Conflicts ............................................................................................... 7
   Error Example ........................................................................................................ 8
   Warning Example .................................................................................................. 9

5 Port Description .................................................................................................... 10

A Product Support .................................................................................................. 13
   Customer Service ................................................................................................. 13
   Customer Technical Support Center ..................................................................... 13
   Technical Support ............................................................................................... 13
   Website .................................................................................................................. 13
   Contacting the Customer Technical Support Center ........................................... 13
   ITAR Technical Support ....................................................................................... 14
Introduction

The SmartFusion2 Microcontroller Subsystem (MSS) provides one USB hard peripheral. On the MSS canvas, you must enable or disable (default) the USB instance based on whether it is being used in your current application. When disabled, the USB instance is held in reset (lowest power state). By default, when enabled, the USB ports are configured to connect to the device Multi Standard I/Os (MSIOs) using the ULPI mode. Note that MSIOs allocated to the USB instance are shared with other MSS peripherals. These shared I/Os are available to connect to MSS GPIOs and other peripherals when the USB instance is disabled or if the USB instance ports are connected to the FPGA fabric.

The functional behavior of each USB instance must be defined at the application level using the SmartFusion2 MSS USB Driver provided by Microsemi.

In this document, we describe how you can configure the MSS USB instance and define how the peripheral signals are connected.

For more details about the MSS USB hard peripheral, please refer to the SmartFusion2 User Guide.
Interface Selection - Use this option to select between the ULPI and UTMI modes (Figure 1-1). When using the ULPI mode, all USB signals are connected to MSIOs. When using the UTMI option, all USB signals are connected the FPGA fabric. In either case, the 'Main Connection' column is not editable.

ULPI Options - If you have selected the ULPI mode, you can:

- Select the MSIO group the USB MSIO signals are connected to (A, B, C or D). Each group defines a different set of device of pins on your device package. Not all groups may be available on a given device/package combination. Group D usually causes the least number of conflicts with other peripherals and GPIO. You can try each group and see which pins cause conflict and iterate through different assignments with the conflicting peripherals and GPIO. Note than in the M2S050 devices only the D group is available where in the M2S025, M2S010 only the A, B and C I/O groups are available.

Note: If the USB instance is enabled, M3_CLK must be greater than 30.0 MHz. This restriction will be enforced in the MSS CCC Configurator.
The SmartFusion2 architecture provides a very flexible schema for connecting peripherals signals to either MSIOs or the FPGA fabric. Use the signal assignment configuration table to define what your peripheral is connected to in your application. This assignment table has the following columns:

**MSIO** - Identifies the peripheral signal name configured in a given row.

**Main Connection** - Indicates whether the signal is connected to an MSIO or the FPGA fabric. For the USB peripheral, the main connection is a function of the mode - ULPI or UTMI - and is not editable.

**Direction** - Indicates if the signal direction is IN, OUT or INOUT.

**Package Pin** - Indicates the package pin associated with the MSIO when the signal is connected to an MSIO.
The Connectivity Preview panel in the MSS USB Configurator dialog shows a graphical view of the current connections for the highlighted signal row (Figure 3-1).

Figure 3-1 • I2C Connectivity Preview
Because MSS peripherals (MMUART, I2C, SPI, CAN, GPIO, USB, Ethernet MAC) share MSIO and FPGA fabric access resources, the configuration of any of these peripherals may result in a resource conflict when you configure an instance of the current peripheral. Peripheral configurators provide clear indicators when such a conflict arises.

Resources used by a previously configured peripheral result in three types of feedback in the current peripheral configurator:

- **Information** - If a resource used by another peripheral does not conflict with the current configuration, an information icon appears in the connectivity preview panel, on that resource. A tooltip on the icon provides details about which peripheral uses that resource.

- **Warning/Error** - If a resource used by another peripheral conflicts with the current configuration, a warning or error icon appears in the connectivity preview panel, on that resource. A tooltip on the icon provides details about which peripheral uses that resource.

When errors are displayed you will not be able to commit the current configuration. You can either resolve the conflict by using a different configuration or cancel the current configuration using the Cancel button.

When warnings are displayed (and there are no errors), you can commit the current configuration. However, you cannot generate the overall MSS; you will see generation errors in the Libero SoC log window. You must resolve the conflict that you created when you committed the configuration by re-configuring either of the peripherals causing the conflict.

The peripheral configurators implement the following rules to determine if a conflict should be reported as an error or a warning:

1. If the peripheral being configured is the GPIO peripheral then all conflicts are errors.
2. If the peripheral being configured is not the GPIO peripheral then all conflicts are errors unless the conflict is with a GPIO resource in which case conflicts will be treated as warnings.
Error Example

The SPI_0 peripheral is used and uses the device PAD bounded to package pin Y30. Configuring the USB peripheral such that the DIR port is connected to an MSIO (ULPI, I/O Group A) results in an error. Figure 4-1 shows the error icon displayed in the connectivity assignment table for the DIR port.

Figure 4-1 • Error Displayed in the Connectivity Assignment Table

Figure 4-2 shows the error icon displayed in the preview panel on the PAD resource for the DIR port.

Figure 4-2 • Error in the Preview Panel
Warning Example

The GPIO peripheral is used and uses the device PAD bounded to package pin V24 (GPIO_3). Configuring the USB peripheral such that the DATA1 port is connected to an MSIO (ULPI, I/O Group A) results in a warning. Figure 4-3 shows the warning icon displayed in the connectivity assignment table for the DATA1 port.

---

**Figure 4-3 • Warning Displayed in the Connectivity Assignment Table**

Figure 4-4 shows the warning icon displayed in the preview panel on the PAD resource for the DATA1 port.

---

**Figure 4-4 • Warning in the Preview Panel**
## 5 – Port Description

### Table 5-1 • ULPI Ports

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Port Group</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULPI_DIR</td>
<td>USB_ULPI_PADS</td>
<td>In</td>
<td>Signal controlling the direction of the data bus. The PHY should drive this signal High when it has data to be transferred. Otherwise, the PHY should drive this signal Low.</td>
</tr>
<tr>
<td>ULPI_NXT</td>
<td>USB_ULPI_PADS</td>
<td>In</td>
<td>Data control; driven high by the PHY to throttle all data types except interruption data and the results of register reads.</td>
</tr>
<tr>
<td>ULPI_STP</td>
<td>USB_ULPI_PADS</td>
<td>Out</td>
<td>Data end control; driven High for one XCLK cycle to signal the end of a transmit operation. It may also be used to stop the current receive operation. Note: Asynchronous path from DIR</td>
</tr>
<tr>
<td>ULPI_XCLK</td>
<td>USB_ULPI_PADS</td>
<td>In</td>
<td>Transceiver macro-cell clock; 60 MHz</td>
</tr>
<tr>
<td>ULPI_DATA&lt;3:0&gt;</td>
<td>USB_ULPI_PADS</td>
<td>Inout</td>
<td>ULPI input/output data bus to ULPI link wrapper</td>
</tr>
</tbody>
</table>

### Table 5-2 • UTMI Ports

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Port Group</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTMI_LINESTATE[1:0]</td>
<td>USB_UTMI_FABRIC</td>
<td>In</td>
<td>Shows the current state of single-ended receivers. LINESTATE[0] reflects the state of D+; LINESTATE[1] reflects state of D-.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>00: SE0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>01: J State</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10: K State</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11: SE1</td>
</tr>
<tr>
<td>UTMI_XDATAIN[7:0]</td>
<td>USB_UTMI_FABRIC</td>
<td>In</td>
<td>Received data</td>
</tr>
<tr>
<td>UTMI_TXREADY</td>
<td>USB_UTMI_FABRIC</td>
<td>In</td>
<td>Transmit data ready; indicates that the transmitter requires data</td>
</tr>
<tr>
<td>UTMI_RXVALID</td>
<td>USB_UTMI_FABRIC</td>
<td>In</td>
<td>Receive data valid; indicates that valid data has been received</td>
</tr>
<tr>
<td>UTMI_RXACTIVE</td>
<td>USB_UTMI_FABRIC</td>
<td>In</td>
<td>Indicates that a valid packet is being received</td>
</tr>
<tr>
<td>UTMI_RXERROR</td>
<td>USB_UTMI_FABRIC</td>
<td>In</td>
<td>Indicates that the packet being received is about to be aborted due to an error</td>
</tr>
<tr>
<td>UTMI_VBUSVALID</td>
<td>USB_UTMI_FABRIC</td>
<td>In</td>
<td>VBus compared to selected VBus valid threshold (required to be between 4.4 V and 4.75 V)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Above the VBus valid threshold</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Below the VBus valid threshold</td>
</tr>
</tbody>
</table>

10
Table 5-2 • UTMI Ports

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Port Group</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
</table>
| UTMI_AVALID          | USB_UTMI_FABRIC| In        | In VBus compared to session valid threshold for a B device (required to be between 0.8 V and 2 V)  
1: Above the session valid threshold  
0: Below the session valid threshold |
| UTMI_SESEND          | USB_UTMI_FABRIC| In        | In VBus compared to session end threshold (required to be between 0.2 V and 0.8 V)  
0: Above the session end threshold  
1: Below the session end threshold |
| UTMI_HOSTDISCON      | USB_UTMI_FABRIC| In        | Host mode only; must be asserted when a high-speed disconnect occurs (in accordance with the UTMI+ specification).  
Note: Full/low-speed connections are monitored via the LINESTATE signal. |
| UTMI_IDDIG           | USB_UTMI_FABRIC| In        | Indicates USB controller connector type. High = B-type, Low = A-type.                                 |
| UTMI_VSTATUS[7:0]    | USB_UTMI_FABRIC| In        | PHY status data; 8 bits wide as per UTMI+ specifications                                              |
| UTMI_CLK             | USB_UTMI_FABRIC| In        | Transceiver macro-cell clock; 60 MHz                                                                |
| UTMI_SUSPENDM        | USB_UTMI_FABRIC| Out       | Asynchronous suspend mode indicator (derived from signals from both CLK and XCLK flip-flops). When enabled through bit 0 of the Power register, goes Low when the device is in suspend mode. Otherwise High (intended to drive a UTMI PHY). |
| UTMI_OPMODE[1:0]     | USB_UTMI_FABRIC| Out       | Operating mode selector  
00: Normal operation  
01: Non-driving  
10: Bit stuffing and NRZI encoding disabled  
11: Reserved |
| UTMI_XDATAOUT[7:0]   | USB_UTMI_FABRIC| Out       | Data to be transmitted                                                                               |
| UTMI_TXVALID         | USB_UTMI_FABRIC| Out       | Transmit data valid; indicates there is valid data to be transmitted                                 |
| UTMI_XCVRSEL[1:0]    | USB_UTMI_FABRIC| Out       | Transceiver select  
00: HS transceiver  
01: FS transceiver  
10: LS transceiver  
11: FS transceiver, LS packet |
| UTMI_TERMSEL         | USB_UTMI_FABRIC| Out       | Termination select. When 0, high-speed termination is enabled; when 1, full-speed termination is enabled.  
Note: May be used to switch the pull-up resistor on D+. |
| UTMI_DRVBUS          | USB_UTMI_FABRIC| Out       | VBus power enable (used when the USB controller is operating as an A device)                          |
| UTMI_CHRGVBUS        | USB_UTMI_FABRIC| Out       | Charge VBus (used during session request when the USB controller is operating as a B device)         |
Table 5-2 • UTMI Ports

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Port Group</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTMI_DISCHRGVBUS</td>
<td>USB_UTMI_FABRIC</td>
<td>Out</td>
<td>Discharge VBus (used by B devices to ensure that VBus is low enough before starting session request protocol (SRP))</td>
</tr>
<tr>
<td>UTMI_DPPULLDOWN</td>
<td>USB_UTMI_FABRIC</td>
<td>Out</td>
<td>Enable for a pull-down resistor within the transceiver on the D+ line. Low when the USB controller is operating as a peripheral; High when USB controller is operating as a host</td>
</tr>
<tr>
<td>UTMI_DMPULLDOWN</td>
<td>USB_UTMI_FABRIC</td>
<td>Out</td>
<td>Enable for a pull-down resistor within the transceiver on the D– line. Needs to be High when the USB controller is being used for point-to-point communications.</td>
</tr>
<tr>
<td>UTMI_IDPULLUP</td>
<td>USB_UTMI_FABRIC</td>
<td>Out</td>
<td>Enable for IDDIG signal generation</td>
</tr>
<tr>
<td>UTMI_VCONTROL[3:0]</td>
<td>USB_UTMI_FABRIC</td>
<td>Out</td>
<td>PHY control data; 4 bits wide as per UTMI+ specifications</td>
</tr>
<tr>
<td>UTMI_VCONTROLLOADM</td>
<td>USB_UTMI_FABRIC</td>
<td>Out</td>
<td>Active low signal; asserted when new Control information is to be read – if implemented</td>
</tr>
</tbody>
</table>

Note:
- Port names have the name of the USB instance as a prefix, followed by the protocol name, e.g. USB_ULPI_DIR.
- PAD ports are automatically promoted to top throughout the design hierarchy.
A – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call 800.262.1060
- From the rest of the world, call 650.318.4460
- Fax, from anywhere in the world, 408.643.6913

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.
My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select Yes in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.