PIC IP Core - Lattice Propel Builder

User Guide

FPGA-IPUG-02141-1.0

February 2021

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AHB-L	Advanced High-performance Bus – Lite
FPGA	Field Programmable Gate Array
GDB	Gnu Debugger
HDL	Hardware Description Language
LUT	Lookup-Table
PIC	Programmable Interrupt Controller

1. Introduction

The Lattice Semiconductor PIC soft IP is a programmable interrupt controller with 1~8 interrupt inputs and 32-bit AHB-L interface. The number of interrupt inputs is configurable. The interrupt status, masks, and polarities are programmable.

The design is implemented in Verilog HDL. It can be configured and generated using the Lattice Propel™ Builder software. It can be targeted to Mach-NX™ FPGA devices and implemented using the Lattice Diamond® software Place and Route tool integrated with the Synplify Pro® synthesis tool.

1.1. Features

The RISC-V MC soft IP has the following features:

- 1~8 interrupt inputs
- Programmable interrupt status
- Programmable interrupt masks
- Programmable interrupt polarities
- Support for the AHB-L bus standard for register read/write
- Support for Mach-NX

1.2. Conventions

1.2.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

2. Functional Descriptions

2.1. Overview

The PIC aggregates up to eight external interrupt inputs (IRQs) into one interrupt output to processor core. The interrupt status register can be used to read the values of IRQs. Individual IRQs can be configured by programming the corresponding PIC_STATUS, PIC_ENABLE, PIC_SET, and PIC_POL registers. All registers can be accessed through an AHB-L interface, as shown in Figure 2.1.

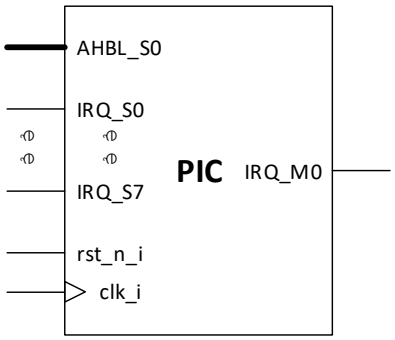

Figure 2.1. PIC Block Diagram

Table 2.1 provides the descriptions of PIC registers.

Table 2.1. PIC Registers

<table>
<tr><th>Offset</th><th>Name</th><th>Description</th></tr>
<tr><td>0x000</td><td>PIC_STATUS</td><td>Interrupt Status Register (read-write) (parameterizable width, min=2, max=8)
Indicate the pending interrupt at corresponding interrupt request port(irq[i]).

<table>
<tr><th>Field</th><th>Name</th><th>Access</th><th>Width</th><th>Reset</th></tr>
<tr><td>[N-1]</td><td>PIC_STATUS [N-1]</td><td>RW</td><td>1</td><td>0x0</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>[1]</td><td>PIC_STATUS [1]</td><td>RW</td><td>1</td><td>0x0</td></tr>
<tr><td>[0]</td><td>PIC_STATUS [0]</td><td>RW</td><td>1</td><td>0x0</td></tr>
</table>

PIC_STATUS[i]:
Read
• 0 – no interrupt at irq[i]
• 1 – interrupt pending at irq[i]
Write
• 0 – no effect
• 1 – clear interrupt status for irq[i]</td></tr>
<tr><td>0x004</td><td>PIC_ENABLE</td><td>Interrupt Enable Register (read-write) (parameterizable width, min=2, max=8)
Enable or Disable the corresponding interrupt request port (irq[i]).

<table>
<tr><th>Field</th><th>Name</th><th>Access</th><th>Width</th><th>Reset</th></tr>
<tr><td>[N-1]</td><td>PIC_ENABLE[N-1]</td><td>RW</td><td>1</td><td>0x0</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>[1]</td><td>PIC_ENABLE[1]</td><td>RW</td><td>1</td><td>0x0</td></tr>
<tr><td>[0]</td><td>PIC_ENABLE[0]</td><td>RW</td><td>1</td><td>0x0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Offset</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td>PIC_ENABLE[i]:
Read
• 0 – irq[i] disabled
• 1 – irq[i] enabled
Write
• 0 – disable irq[i]
• 1 – enable irq[i]</td></tr>
<tr><td>0x008</td><td>PIC_SET</td><td>Interrupt Set Register (write-only) (parameterizable width, min=2, max=8)
Set the interrupt status for corresponding interrupt request port (irq[i]).

<table>
<tr><th>Field</th><th>Name</th><th>Access</th><th>Width</th><th>Reset</th></tr>
<tr><td>[N-1]</td><td>PIC_SET [N-1]</td><td>W</td><td>1</td><td>0x0</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>[1]</td><td>PIC_SET [1]</td><td>W</td><td>1</td><td>0x0</td></tr>
<tr><td>[0]</td><td>PIC_SET [0]</td><td>W</td><td>1</td><td>0x0</td></tr>
</table>

PIC_SET[i]:
Read
• Invalid operation, will get 0.
Write
• 0 – no effect
• 1 – set interrupt status for irq[i] (set PIC_STATUS[i])</td></tr>
<tr><td>0x00C</td><td>PIC_POL</td><td>Interrupt Polarity Register (read-write) (parameterizable width, min=2, max=8)
Indicates the polarity of corresponding interrupt request (irq[i]) port.

<table>
<tr><th>Field</th><th>Name</th><th>Access</th><th>Width</th><th>Reset</th></tr>
<tr><td>[N]</td><td>PIC_POL [N]</td><td>RW</td><td>1</td><td>0x0</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>[1]</td><td>PIC_POL l[1]</td><td>RW</td><td>1</td><td>0x0</td></tr>
<tr><td>[0]</td><td>PIC_POL [0]</td><td>RW</td><td>1</td><td>0x0</td></tr>
</table>

PIC_POL[i]:
Read
• 0 – irq[i] is active high
• 1 – irq[i] is active low
Write
• 0 – Set irq[i] active high
• 1 – Set irq[i] active low</td></tr>
</table>

Note: The register definition of PIC follows Lattice Interrupt Interface (LINTR) Standard, refer to Lattice Memory Mapped Interface (LMMI) and Lattice Interrupt Interface (LINTR) User Guide (FPGA-UG-02039) for more information.

2.2. Signal Description

Table 2.2 to Table 2.4 list the ports of the PIC soft IP in different categories.

2.2.1. Clock and Reset

Table 2.2. Clock and Reset Port

Name	Type	Width	Description
clk_i	In	1	Clock input
rst_n_i	In	1	Reset input (active low)

2.2.2. Register Interface

Table 2.3. Register Port

Name	Type	Width	Description
AHBL_S0/HSELx	In	1	Standard AHB-Lite Slave interface
AHBL_S0/HREADY	In	1	Standard AHB-Lite Slave interface
AHBL_S0/HADDR	In	32	Standard AHB-Lite Slave interface
AHBL_S0/HWRITE	In	1	Standard AHB-Lite Slave interface
AHBL_S0/HSIZE	In	3	Standard AHB-Lite Slave interface
AHBL_S0/HPROT	In	1	Standard AHB-Lite Slave interface
AHBL_S0/HTRANS	In	2	Standard AHB-Lite Slave interface
AHBL_S0/HBURST	In	3	Standard AHB-Lite Slave interface
AHBL_S0/HMASTLOCK	In	1	Standard AHB-Lite Slave interface
AHBL_S0/HWDATA	In	32	Standard AHB-Lite Slave interface
AHBL_S0/HRDATA	Out	32	Standard AHB-Lite Slave interface
AHBL_S0/HREADYOUT	Out	1	Standard AHB-Lite Slave interface
AHBL_S0/HRESP	Out	1	Standard AHB-Lite Slave interface

Note: Refer to AMBA 3 AHB-Lite Protocol V1.0 for more information.

2.2.3. Interrupt Interface

Table 2.4. Interrupt Port

Name	Type	Width	Description
IRQ_Sx (x can be 0~7)	In	1	Interrupt inputs
IRQ_M0	Out	1	Interrupt output

2.3. Attribute Summary

The configurable attributes of the PIC IP are shown in Table 2.5 and are described in Table 2.6.
The attributes can be configured through the Propel Builder software.

Table 2.5. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
PIC_IRQ_NUM	1~8	8	—

Table 2.6. Attributes Description

Attribute	Description
PIC_IRQ_NUM	Number of interrupt inputs

3. PIC IP Generation

This section provides information on how to generate the PIC IP Core module using Propel Builder.

To generate the PIC IP Core module:

1. In Propel Builder, create a new design, then select the PIC package.
2. Enter the module name and click **Next** as shown in Figure 3.1.

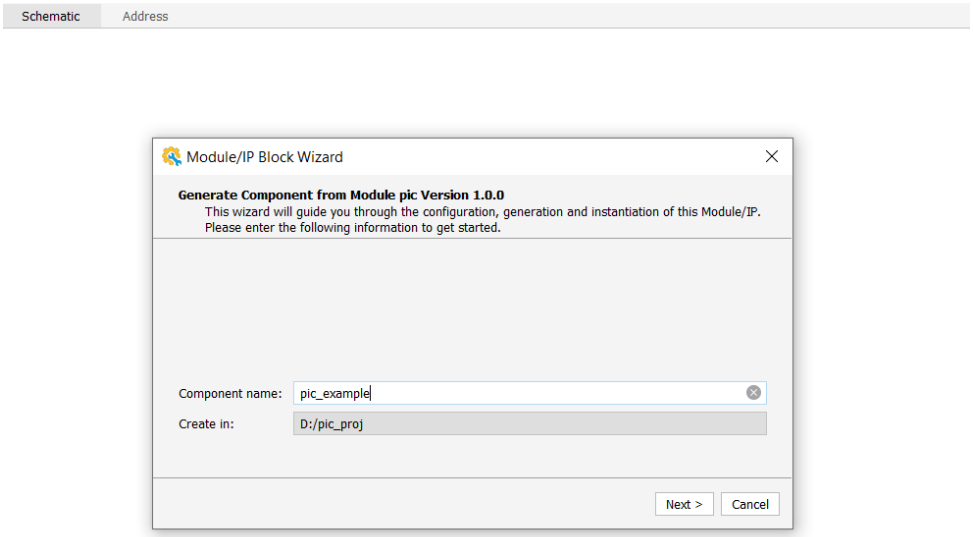

Figure 3.1. Entering the Module Name

3. Configure the parameters as needed, then click **Generate**.

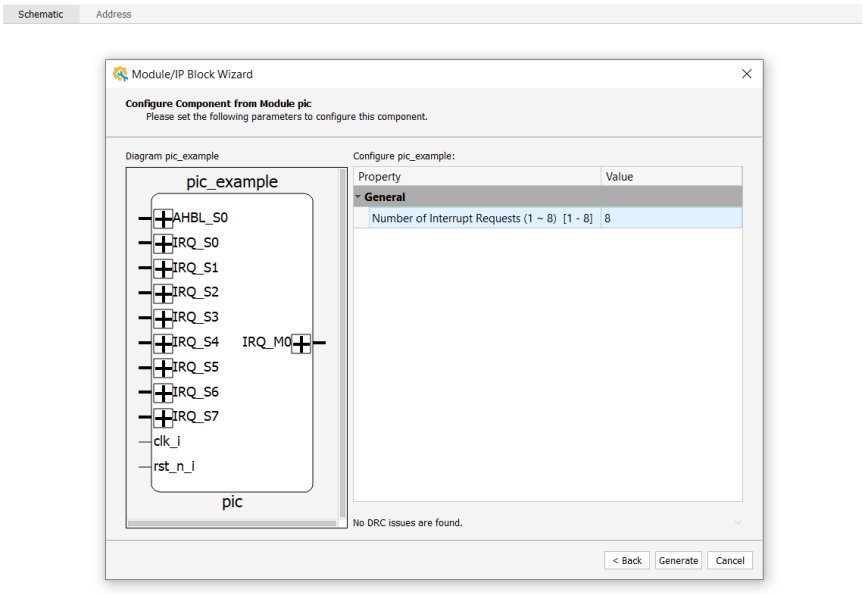

Figure 3.2. Configuring the Parameters

4. Verify the information and click **Finish**.

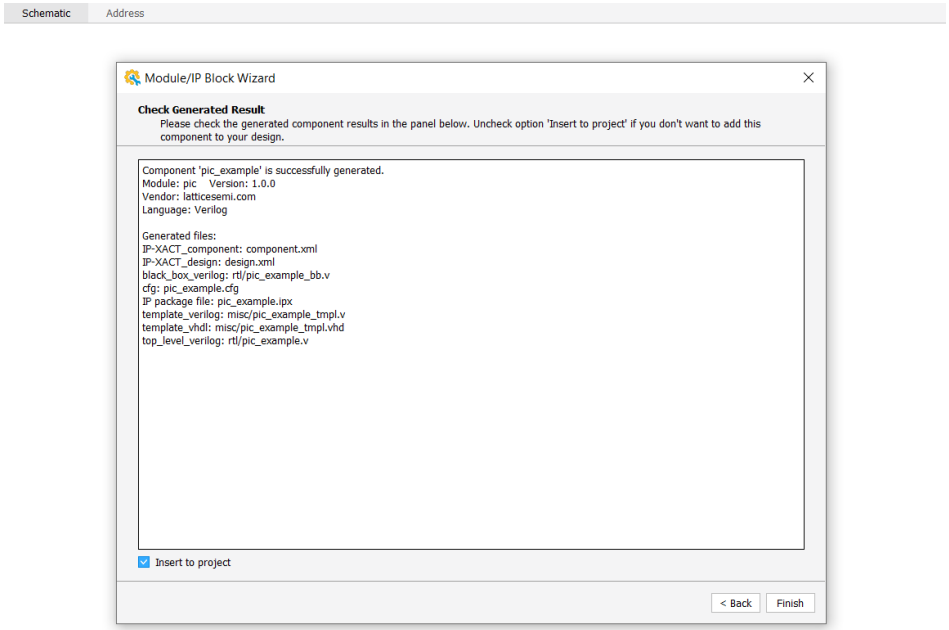

Figure 3.3. Verifying the Result

5. Confirm or modify the module instance name, then click **OK**.

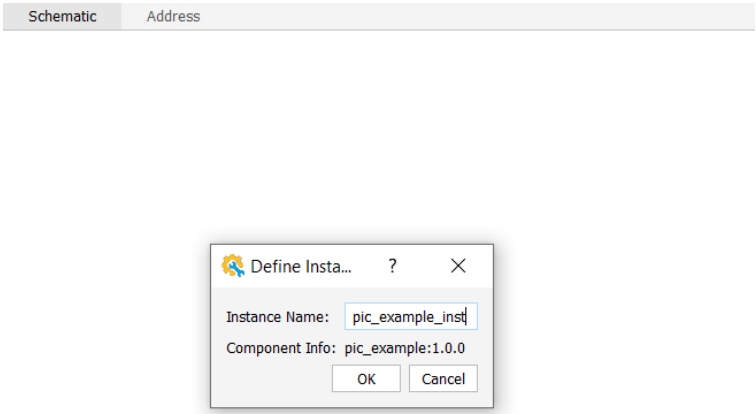

Figure 3.4. Specifying the Instance Name

The PIC IP instance is successfully generated.

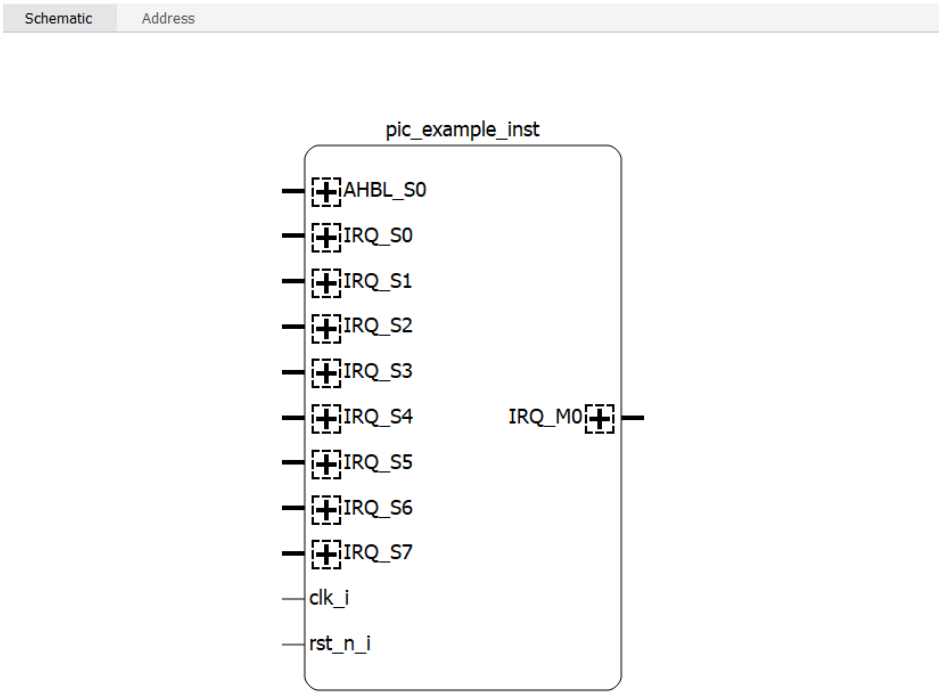

Figure 3.5. Generated Instance

References

- AMBA 3 AHB-Lite Protocol V1.0
- Lattice Memory Mapped Interface (LMMI) and Lattice Interrupt Interface (LINTR) User Guide (FPGA-UG-02039)

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.0, February 2021

Section	Change Summary
All	Initial release

www.latticesemi.com