



# **CNN Coprocessor Unit IP Core**

## **User Guide**

FPGA-IPUG-02170-1.0

November 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, with all faults and associated risk the responsibility entirely of the 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

Acronyms in This Document .....	5
1. Introduction .....	6
1.1. Features .....	6
1.2. Conventions .....	6
1.2.1. Nomenclature .....	6
2. Functional Descriptions .....	7
2.1. Overview .....	7
2.2. Signal Description .....	9
2.2.1. Clock and Reset .....	9
2.2.2. AHB-Lite Slave Interface .....	9
2.2.3. AHB-Lite Master Interface .....	10
2.3. Attribute Summary .....	10
3. Generating the CCU IP .....	11
3.1. Generating the CCU IP using Lattice Radiant Software .....	11
3.2. Generating the CCU IP using Lattice Propel Builder .....	13
4. Operating the CCU .....	16
5. Calculating the CCU .....	17
Appendix A. Resource Utilization .....	19
References .....	20
Technical Support Assistance .....	21
Revision History .....	22

## Figures

Figure 2.1. CCU Block Diagram .....	7
Figure 3.1. Module/IP Block Wizard in Lattice Radiant .....	11
Figure 3.2. Configure User Interface of CCU IP Core .....	12
Figure 3.3. Check Generating Result.....	12
Figure 3.4. Module/IP Block Wizard in Lattice Propel Builder .....	13
Figure 3.5. Configuring the Parameters.....	14
Figure 3.6. Verifying the Result.....	14
Figure 3.7. Specifying the Instance Name.....	15
Figure 3.8. Generated Instance.....	15
Figure 5.1. CCU Calculation Flow .....	17
Figure 5.2. acc_out<1..4> Calculation Flow .....	18

## Tables

Table 2.1. CCU Registers .....	7
Table 2.2. Clock and Reset Port .....	9
Table 2.3. AHB-Lite Slave Interface.....	9
Table 2.4. AHB-Lite Master Interface.....	10
Table 2.5. Attributes Table .....	10
Table 2.6. Attributes Description.....	10
Table A.1. Device and Tool Tested.....	19
Table A.2. Resource Utilization.....	19

## Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AHB-L	Advanced High-performance Bus – Lite
CCU	CNN Coprocessor Unit
HDL	Hardware Description Language
MAC	Multiply-accumulate

# 1. Introduction

The CNN Coprocessor Unit (CCU) IP core is a calculation engine that accelerates the input channel (depth) loop of convolution function. It performs multiply-accumulate operation on the input and filter values. Offset value can be set for both the input and filter values.

The operation of the IP core can be controlled and monitored by register access. And the output of the calculation is stored in a register. This IP core is intended to offload the calculation from the processor.

## 1.1. Features

The CCU soft IP has the following features:

- Supports both unsigned input/filter values and signed input/filter values
- Supports four simultaneous MAC operations for each cycle
- Slave AHB-Lite interface for accessing the registers for configuration and operation control
- Master AHB-Lite interface for retrieving the input and filter values

## 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 CCU has an AHB-L slave interface for accessing the registers to configure and control the CCU operation. It also has AHB-L master interface for retrieving data needed for CCU calculation. Interface is shown in [Figure 2.1](#).

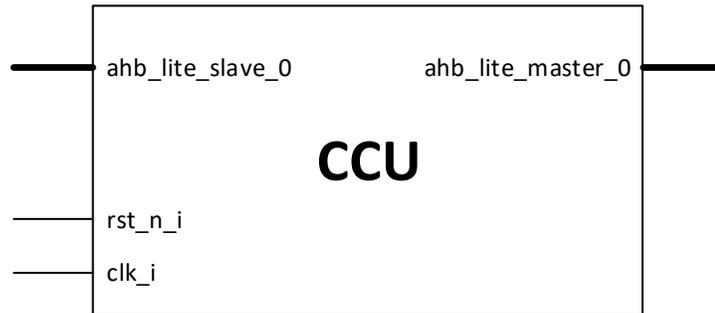


Figure 2.1. CCU Block Diagram

[Table 2.1](#) provides the descriptions of CCU registers.

Table 2.1. CCU Registers

Offset	Name	Description															
0x000	PDMACR	<p><b>CCU control register</b> This register is used to trigger the start of CCU operation.</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:1]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[0]</td> <td>START</td> <td>WO</td> <td>1</td> <td>0x0</td> </tr> </tbody> </table> <p>START:</p> <ul style="list-style-type: none"> <li>0 – no effect</li> <li>1 – Start CCU operation</li> </ul>	Field	Name	Access	Width	Reset	[31:1]	Reserved	--	--	--	[0]	START	WO	1	0x0
Field	Name	Access	Width	Reset													
[31:1]	Reserved	--	--	--													
[0]	START	WO	1	0x0													
0x004	PDMASR	<p><b>CCU status register</b> Indicates the status of CCU operation</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:1]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[0]</td> <td>DONE</td> <td>RO</td> <td>1</td> <td>0x1</td> </tr> </tbody> </table> <p>DONE:</p> <ul style="list-style-type: none"> <li>0 – CCU process is NOT completed</li> <li>1 – CCU process is completed</li> </ul>	Field	Name	Access	Width	Reset	[31:1]	Reserved	--	--	--	[0]	DONE	RO	1	0x1
Field	Name	Access	Width	Reset													
[31:1]	Reserved	--	--	--													
[0]	DONE	RO	1	0x1													

Offset	Name	Description															
0x008	SIGSELR	<p><b>Sign Select configuration register</b> Configures the sign of input and filter values</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:1]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[0]</td> <td>SIGN_SEL</td> <td>RW</td> <td>1</td> <td>0x0</td> </tr> </tbody> </table> <p>SIGN_SEL:</p> <ul style="list-style-type: none"> <li>0 – Unsigned</li> <li>1 – Signed</li> </ul>	Field	Name	Access	Width	Reset	[31:1]	Reserved	--	--	--	[0]	SIGN_SEL	RW	1	0x0
Field	Name	Access	Width	Reset													
[31:1]	Reserved	--	--	--													
[0]	SIGN_SEL	RW	1	0x0													
0x00C	INOFFSETCR	<p><b>Input Offset configuration register</b> Configures the offset of input values</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:9]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[8:0]</td> <td>INPUT_OFFSET</td> <td>RW</td> <td>9</td> <td>0x00</td> </tr> </tbody> </table> <p>Input Offset Value:</p> <ul style="list-style-type: none"> <li>2's complement signed number</li> <li>Range: -256 ~ 255</li> </ul>	Field	Name	Access	Width	Reset	[31:9]	Reserved	--	--	--	[8:0]	INPUT_OFFSET	RW	9	0x00
Field	Name	Access	Width	Reset													
[31:9]	Reserved	--	--	--													
[8:0]	INPUT_OFFSET	RW	9	0x00													
0x010	FILOFFSETCR	<p><b>Filter Offset configuration register</b> Configures the offset of filter values</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:9]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[8:0]</td> <td>FILTER_OFFSET</td> <td>RW</td> <td>9</td> <td>0x00</td> </tr> </tbody> </table> <p>Filter Offset Value:</p> <ul style="list-style-type: none"> <li>2's complement signed number</li> <li>Range: -256 ~ 255</li> </ul>	Field	Name	Access	Width	Reset	[31:9]	Reserved	--	--	--	[8:0]	FILTER_OFFSET	RW	9	0x00
Field	Name	Access	Width	Reset													
[31:9]	Reserved	--	--	--													
[8:0]	FILTER_OFFSET	RW	9	0x00													
0x014	INDEPTHCR	<p><b>Input Depth Configuration register</b> Configures the input depth value</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:10]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[9:0]</td> <td>INPUT_DEPTH_BY_2_M1</td> <td>RW</td> <td>10</td> <td>0x00</td> </tr> </tbody> </table> <p>Input Depth Value:</p> <ul style="list-style-type: none"> <li><math>input\_depth = (input\_depth\_by\_2\_m1 + 1) / 2</math></li> <li>Range: 0 ~ 1023</li> <li>Covers up to 512 input depth</li> </ul>	Field	Name	Access	Width	Reset	[31:10]	Reserved	--	--	--	[9:0]	INPUT_DEPTH_BY_2_M1	RW	10	0x00
Field	Name	Access	Width	Reset													
[31:10]	Reserved	--	--	--													
[9:0]	INPUT_DEPTH_BY_2_M1	RW	10	0x00													
0x018	INPUTADDRCR	<p><b>Input Data Address Configuration register</b> Configures the start address of input data</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:17]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[16:0]</td> <td>INPUT_DATA_ADDR</td> <td>RW</td> <td>17</td> <td>0x00</td> </tr> </tbody> </table>	Field	Name	Access	Width	Reset	[31:17]	Reserved	--	--	--	[16:0]	INPUT_DATA_ADDR	RW	17	0x00
Field	Name	Access	Width	Reset													
[31:17]	Reserved	--	--	--													
[16:0]	INPUT_DATA_ADDR	RW	17	0x00													

Offset	Name	Description															
0x01C	FILTERADDRCR	<b>Filter Data Address Configuration register</b> Configures the start address of filter data  <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:17]</td> <td>Reserved</td> <td>--</td> <td>--</td> <td>--</td> </tr> <tr> <td>[16:0]</td> <td>FILTER_DATA_ADDR</td> <td>RW</td> <td>17</td> <td>0x00</td> </tr> </tbody> </table>	Field	Name	Access	Width	Reset	[31:17]	Reserved	--	--	--	[16:0]	FILTER_DATA_ADDR	RW	17	0x00
Field	Name	Access	Width	Reset													
[31:17]	Reserved	--	--	--													
[16:0]	FILTER_DATA_ADDR	RW	17	0x00													
0x020	ACCOUTR	<b>Accumulator Output register</b> Indicates the result of the CCU calculation  <table border="1"> <thead> <tr> <th>Field</th> <th>Name</th> <th>Access</th> <th>Width</th> <th>Reset</th> </tr> </thead> <tbody> <tr> <td>[31:0]</td> <td>ACC_OUT</td> <td>RO</td> <td>32</td> <td>0x00</td> </tr> </tbody> </table> <p>Accumulator Output Value:</p> <ul style="list-style-type: none"> <li>Result of CCU calculation is valid when PDMA SR.DONE = '1'</li> </ul>	Field	Name	Access	Width	Reset	[31:0]	ACC_OUT	RO	32	0x00					
Field	Name	Access	Width	Reset													
[31:0]	ACC_OUT	RO	32	0x00													

## 2.2. Signal Description

Table 2.2 to Table 2.4 list the ports of the CCU 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
reset_n_i	In	1	Reset input (active low)

### 2.2.2. AHB-Lite Slave Interface

Table 2.3. AHB-Lite Slave Interface

Name	Type	Width	Description
ahbl_s00_hsel_i	In	1	Standard AHB-Lite Slave interface
ahbl_s00_haddr_i	In	AHBL_ADDR_WIDTH	Standard AHB-Lite Slave interface
ahbl_s00_hwrite_i	In	1	Standard AHB-Lite Slave interface
ahbl_s00_hsize_i	In	3	Standard AHB-Lite Slave interface
ahbl_s00_hburst_i	In	3	Standard AHB-Lite Slave interface
ahbl_s00_hprot_i	In	4	Standard AHB-Lite Slave interface
ahbl_s00_htrans_i	In	2	Standard AHB-Lite Slave interface
ahbl_s00_hmastlock_i	In	1	Standard AHB-Lite Slave interface
ahbl_s00_hready_i	In	1	Standard AHB-Lite Slave interface
ahbl_s00_hwdata_i	In	AHBL_DATA_WIDTH	Standard AHB-Lite Slave interface
ahbl_s00_hreadyout_o	Out	1	Standard AHB-Lite Slave interface
ahbl_s00_hresp_o	Out	1	Standard AHB-Lite Slave interface
ahbl_s00_hrdata_o	Out	AHBL_DATA_WIDTH	Standard AHB-Lite Slave interface

**Note:** Refer to AMBA 3 AHB-Lite Protocol version 1.0 for more information.

### 2.2.3. AHB-Lite Master Interface

**Table 2.4. AHB-Lite Master Interface**

Name	Type	Width	Description
ahbl_m00_hready_i	In	1	Standard AHB-Lite Master interface
ahbl_m00_hresp_i	In	1	Standard AHB-Lite Master interface
ahbl_m00_hrdata_i	In	AHBL_DATA_WIDTH	Standard AHB-Lite Master interface
ahbl_m00_haddr_o	Out	AHBL_ADDR_WIDTH	Standard AHB-Lite Master interface
ahbl_m00_hwrite_o	Out	1	Standard AHB-Lite Master interface
ahbl_m00_hsize_o	Out	3	Standard AHB-Lite Master interface
ahbl_m00_hburst_o	Out	3	Standard AHB-Lite Master interface
ahbl_m00_hprot_o	Out	4	Standard AHB-Lite Master interface
ahbl_m00_htrans_o	Out	2	Standard AHB-Lite Master interface
ahbl_m00_hmastlock_o	Out	1	Standard AHB-Lite Master interface
ahbl_m00_hwdata_i	Out	AHBL_DATA_WIDTH	Standard AHB-Lite Master interface

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

### 2.3. Attribute Summary

The configurable attributes of the CCU soft IP are shown in [Table 2.5](#) and are described in [Table 2.6](#).

The attributes can be configured through the Lattice Propel™ Builder software.

**Table 2.5. Attributes Table**

Attribute	Selectable Values	Default	Dependency on Other Attributes
AHBL_ADDR_WIDTH	--	32	—
AHBL_DATA_WIDTH	--	32	—
LMMI_ADDR_WIDTH	--	16	—
SIGN_SEL_INIT	--	0	—
INPUT_OFFSET_INIT	--	0	—
FILTER_OFFSET_INIT	--	0	—
INPUT_DEPTH_BY_2_M1_INIT	--	0	—
INPUT_DATA_ADDR_INIT	--	0	—
FILTER_DATA_ADDR_INIT	--	0	—

**Table 2.6. Attributes Description**

Attribute	Description
AHBL_ADDR_WIDTH	Address bus width of AHB-Lite Slave and Master Interface
AHBL_DATA_WIDTH	Data bus width of AHB-Lite Slave and Master Interface
LMMI_ADDR_WIDTH	Address bus width of the internal LMMI Interface
SIGN_SEL_INIT	Initial value of SIGSELR.SIGN_SEL register field value after reset
INPUT_OFFSET_INIT	Initial value of INOFFSETCR.INPUT_OFFSET register field value after reset
FILTER_OFFSET_INIT	Initial value of FILOFFSETCR.FILTER_OFFSET register field value after reset
INPUT_DEPTH_BY_2_M1_INIT	Initial value of INDEPTHCR.INPUT_DEPTH_BY_2_M1 register field value after reset
INPUT_DATA_ADDR_INIT	Initial value of INPUTADDRCR.INPUT_DATA_ADDR register field value after reset
FILTER_DATA_ADDR_INIT	Initial value of FILTERADDRCR.FILTER_DATA_ADDR register field value after reset

### 3. Generating the CCU IP

#### 3.1. Generating the CCU IP using Lattice Radiant Software

This section provides information on how to generate the CCU IP Core module using the Lattice Radiant™ software.

To generate the CCU IP Core:

1. Create a new Lattice Radiant software project or open an existing project.
2. Double-click **CNN Coprocessor Unit** under **Processors\_Controllers\_and\_Peripherals**. This opens the **Module/IP Block Wizard**. Enter the **Component Name** and **Create in** fields as shown in [Figure 3.1](#). Click **Next**.

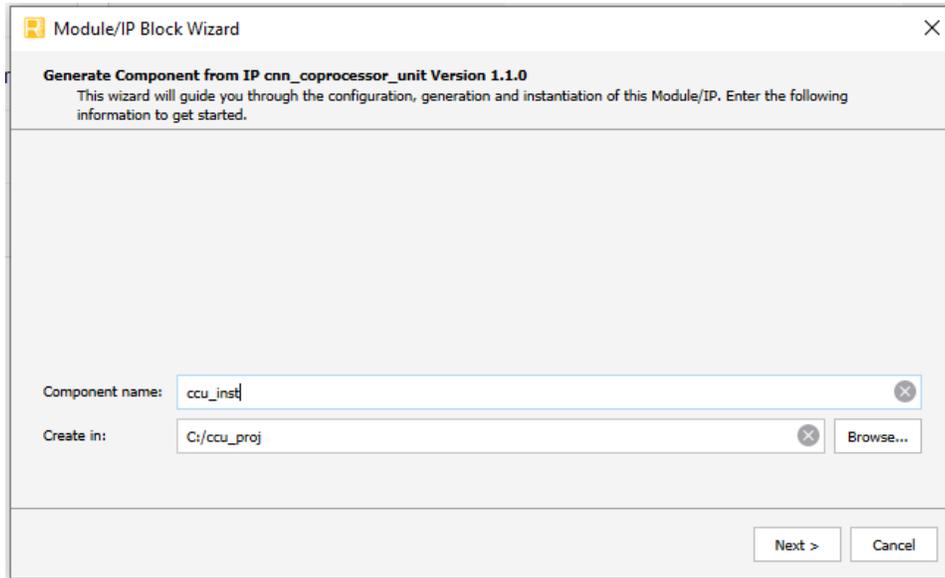
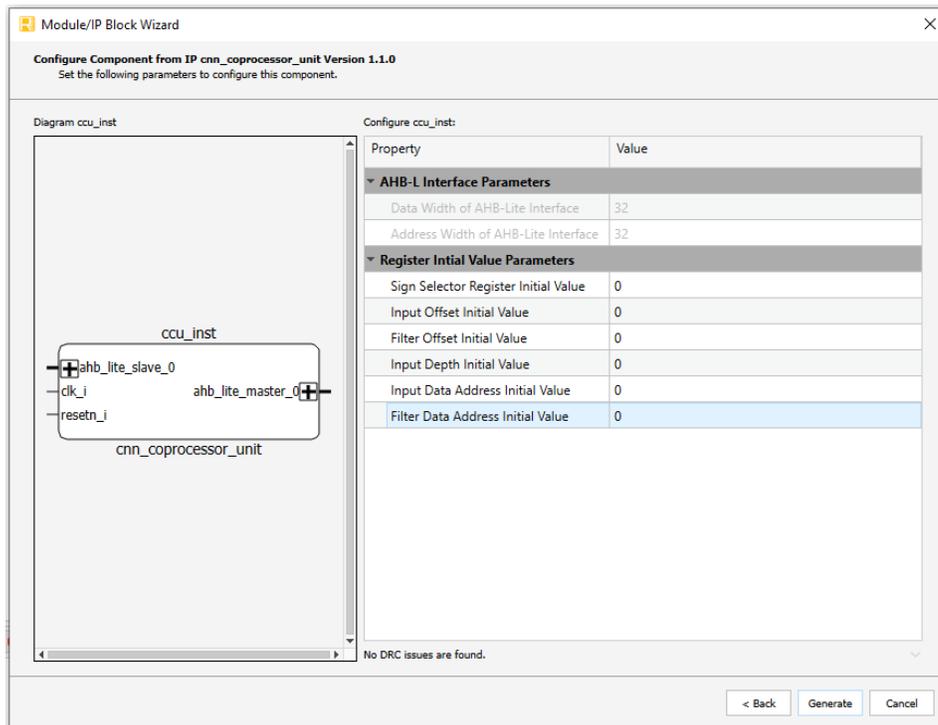


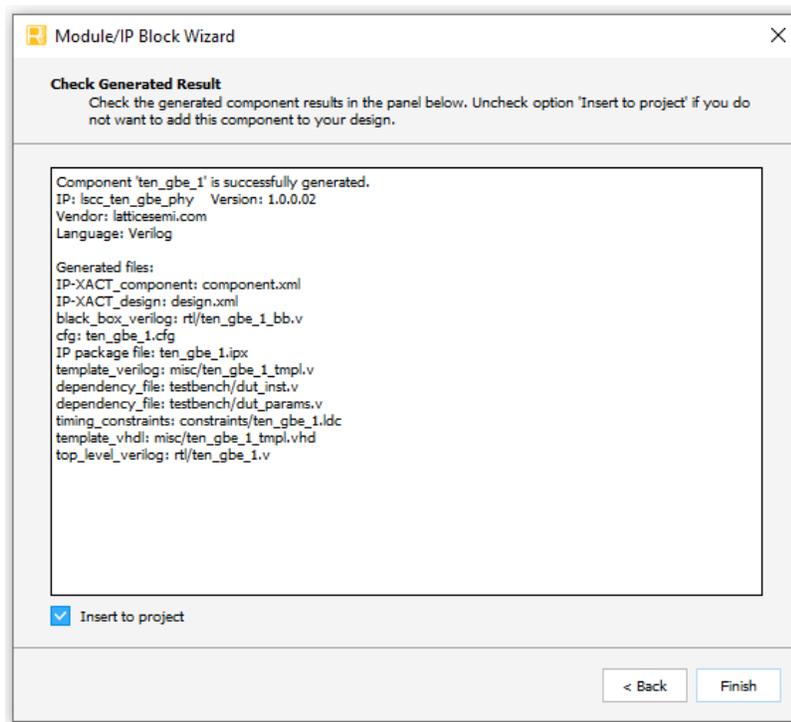
Figure 3.1. Module/IP Block Wizard in Lattice Radiant

- In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected CCU IP Core. As a sample configuration, see [Figure 3.2](#). For configuration options, see the [Attribute Summary](#) section.



**Figure 3.2. Configure User Interface of CCU IP Core**

- Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results as shown in [Figure 3.3](#).



**Figure 3.3. Check Generating Result**

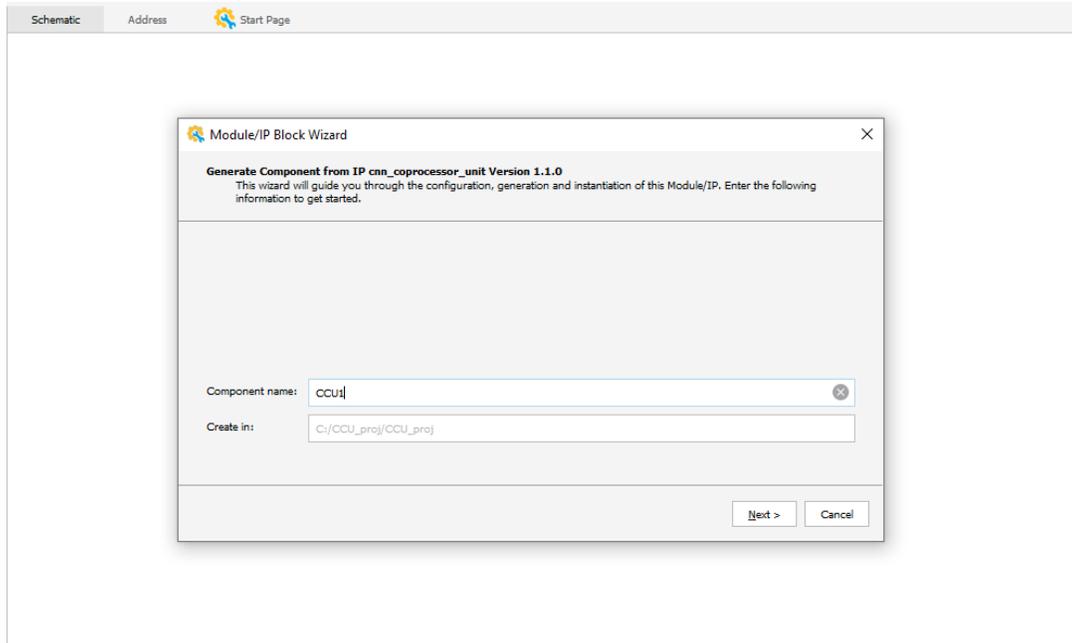
5. Click **Finish**. Once this module is in the Lattice Radiant software project, it can be instantiated in other modules within the project. You can view the files and instance/s added in the **File List** tab.

### 3.2. Generating the CCU IP using Lattice Propel Builder

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

To generate the CCU IP Core:

1. In Propel Builder, create a new design and select the CCU package.
2. Enter the module name and click **Next** as shown in [Figure 3.4](#).



**Figure 3.4. Module/IP Block Wizard in Lattice Propel Builder**

3. Configure the parameters as shown in [Figure 3.5](#).
4. Click **Generate**.

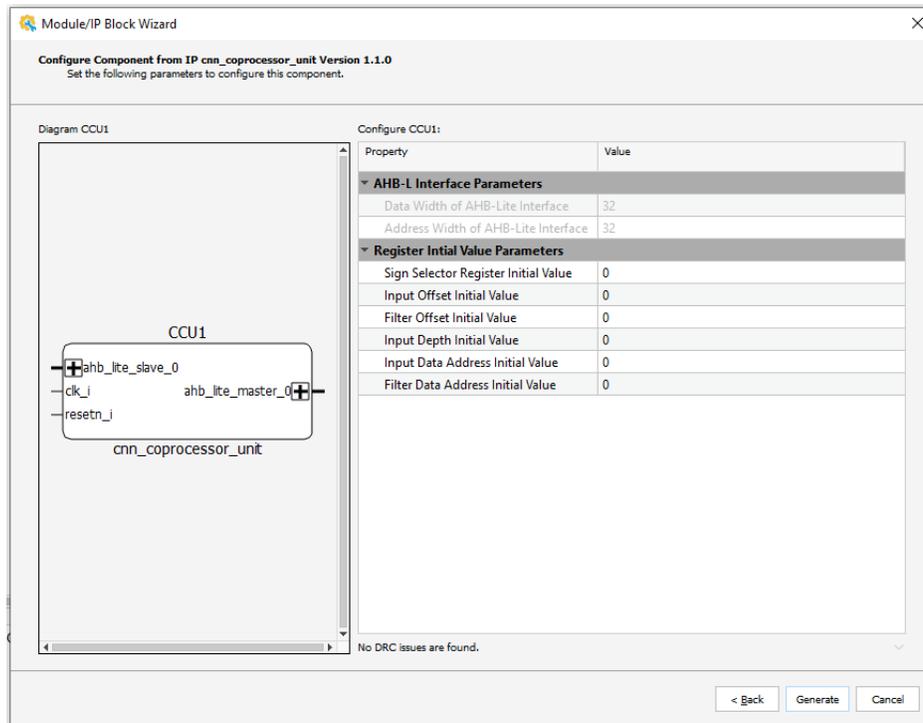


Figure 3.5. Configuring the Parameters

5. Verify the information. Click **Finish**.

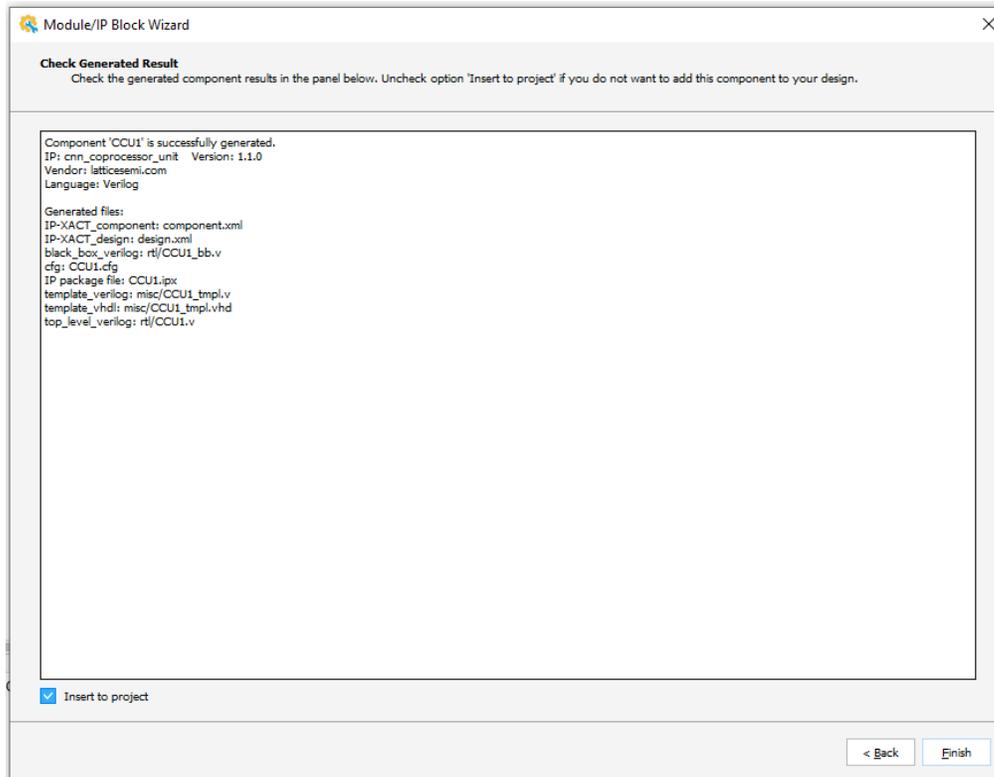
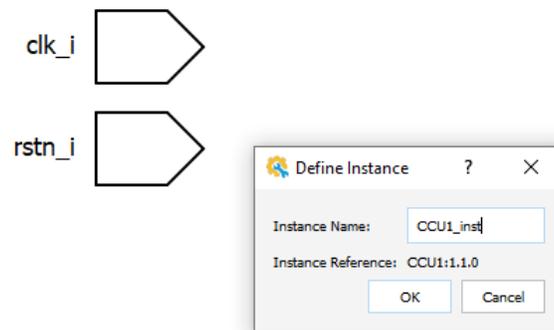


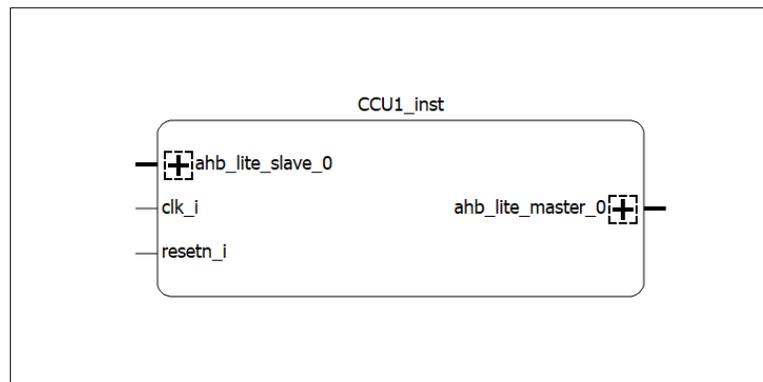
Figure 3.6. Verifying the Result

6. Confirm or modify the module instance name. Click **OK**.



**Figure 3.7. Specifying the Instance Name**

The CCU IP instance is successfully generated.



**Figure 3.8. Generated Instance**

## 4. Operating the CCU

This section describes how to operate the CCU soft IP.

To operate the CCU soft IP, perform the following steps:

1. Program the following configuration registers if previous/init value needs to be updated. Note that the init value can be configured during IP generation.

- SIGSELR
- INOFFSETCR
- FILOFSETCR
- INDEPTHCR
- INPUTADDRCR
- FILTERADDRCR

2. Trigger the start of CCU calculation by writing '1' to PDMACR register.
3. Poll the PDMASR register until done = '1'.
4. Read the results of CCU calculation in ACCOUTR register.

## 5. Calculating the CCU

This section describes the calculation performed by CCU Soft IP. Writing '1' to PDMACR.start triggers the start of CCU calculation. The CCU calculation is described in the following pseudo-code.

Figure 5.1 shows the flow chart of CCU calculation. Figure 5.2 shows how the `acc_out<1..4>` is calculated.

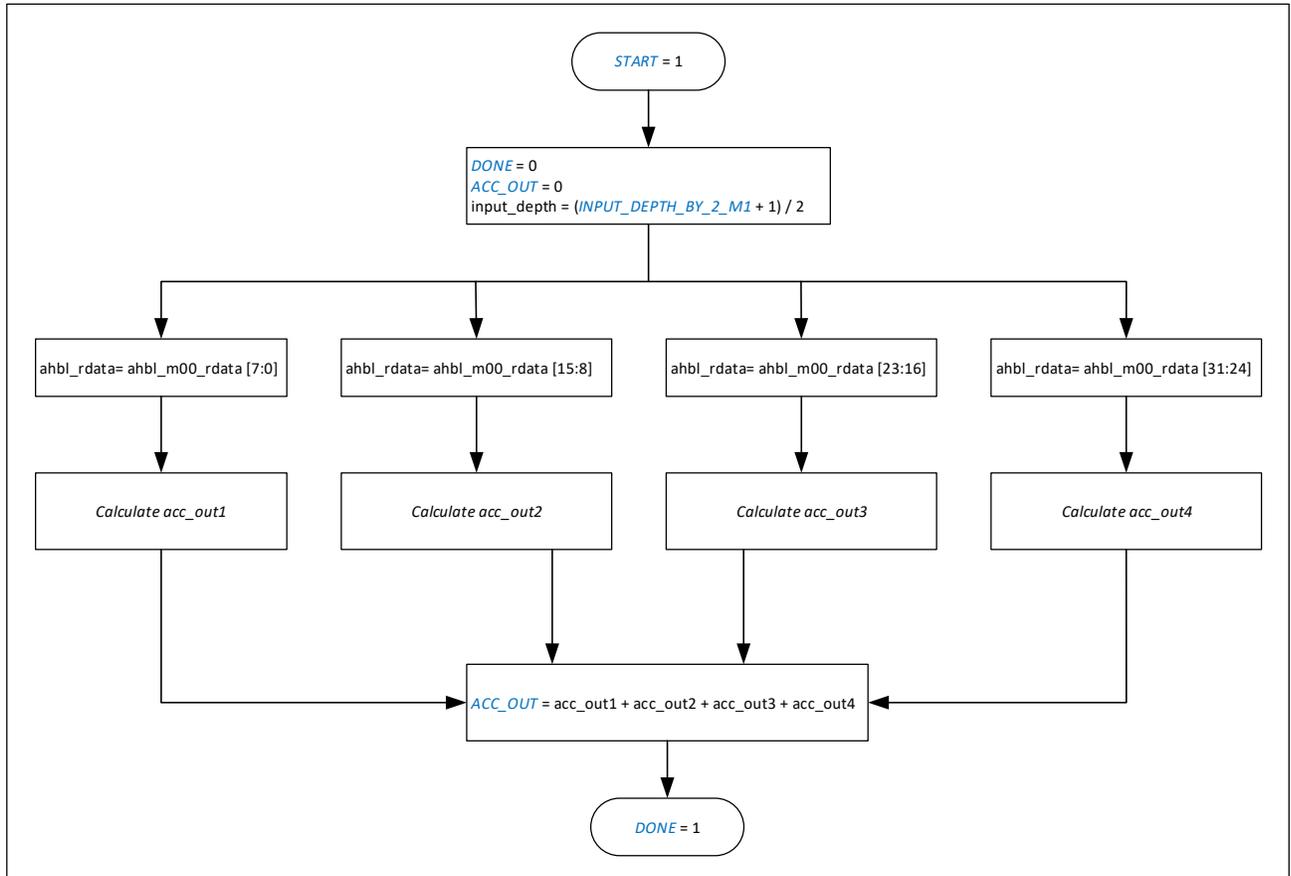


Figure 5.1. CCU Calculation Flow

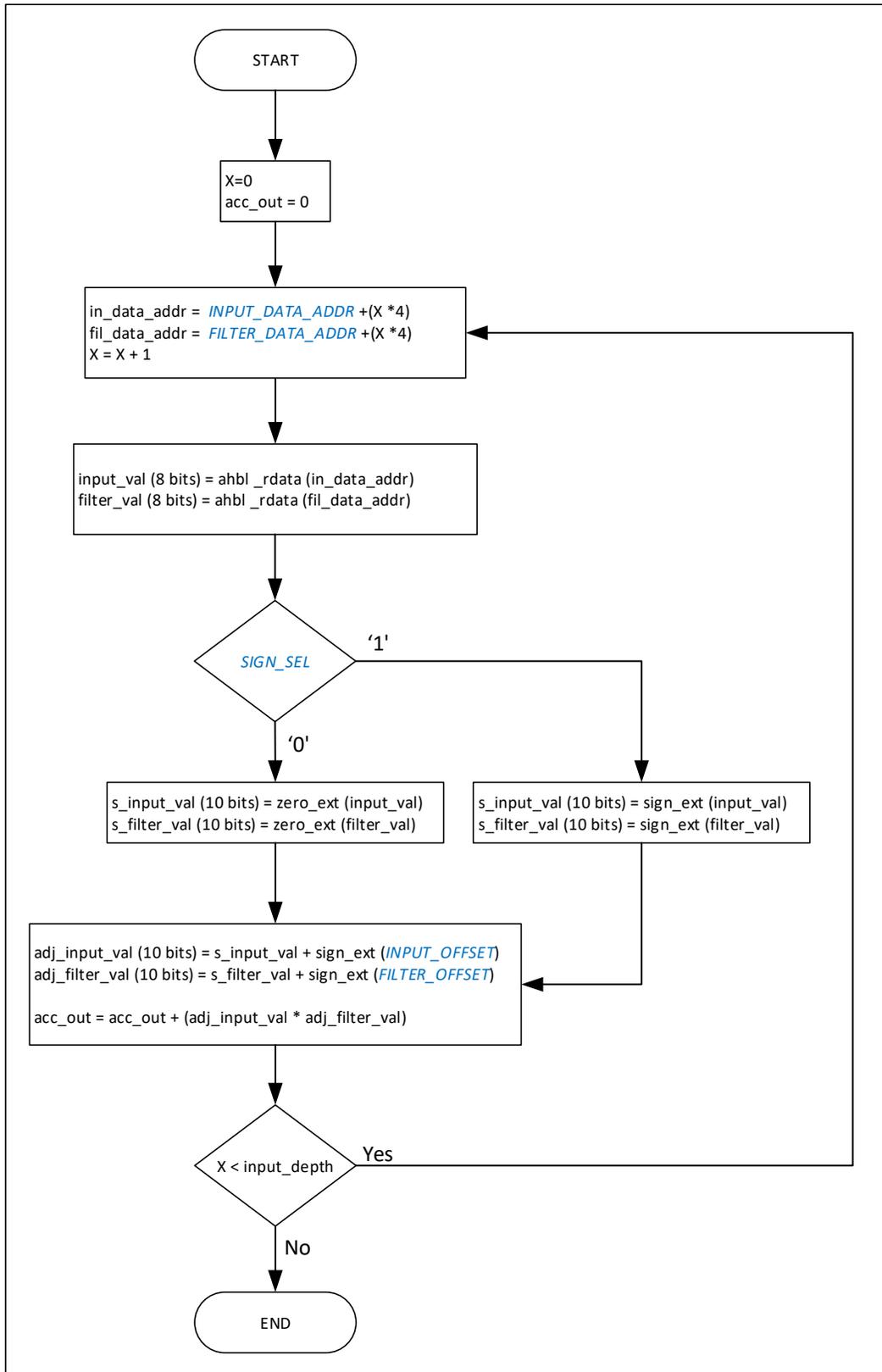


Figure 5.2.  $acc\_out\langle 1..4 \rangle$  Calculation Flow

## Appendix A. Resource Utilization

Table A.2 shows the resource utilization using the device and tools listed in Table A.1. No effect on the resource utilization when the attribute values were changed.

**Table A.1. Device and Tool Tested**

	Value
Lattice Radiant Software Version	ng3.1 (for Windows)
Device Used	LFCPNX-100-9ASG256C
Performance Grade	9_High-Performance_1.0V
Synthesis Tool	Lattice Synthesis Engine (LSE)
	Synplify Pro® (R) R-2021.03LR-SP1-Beta2, Build 097R, Aug 12 2021

**Table A.2. Resource Utilization**

Configuration	Synthesis Tool	Register	LUTs	EBR	DSP
Any combination of attribute values	Synplify Pro	597	994	0	4
	LSE	698	1127	0	4

## References

[AMBA 3 AHB-Lite Protocol V1.0](#)

## Technical Support Assistance

Submit a technical support case through [www.latticesemi.com/techsupport](http://www.latticesemi.com/techsupport).

## Revision History

### Revision 1.0, November 2021

Section	Change Summary
All	Initial release.



[www.latticesemi.com](http://www.latticesemi.com)