



# Integer Square Root Core User Guide

---

UG-CORE-SQRT-v3.3  
February 2023  
[www.efinixinc.com](http://www.efinixinc.com)



# Contents

<b>Introduction.....</b>	<b>3</b>
<b>Features.....</b>	<b>3</b>
<b>Resource Utilization and Performance.....</b>	<b>4</b>
<b>Functional Description.....</b>	<b>5</b>
Ports.....	5
Continuous Mode.....	6
Single Mode.....	6
<b>IP Manager.....</b>	<b>7</b>
<b>Customizing the Integer Square Root.....</b>	<b>8</b>
<b>Integer Square Root Example Design.....</b>	<b>9</b>
<b>Integer Square Root Testbench.....</b>	<b>9</b>
<b>Revision History.....</b>	<b>11</b>

# Introduction

The Integer Square Root core calculates the integer square root of an input value.

Use the IP Manager to select IP, customize it, and generate files. The Integer Square Root core has an interactive wizard to help you set parameters. The wizard also has options to create a testbench and/or example design targeting an Efinix® development board.

## Features

- Integer square root function
  - Returns an integer result and any remainder
  - Parameterized input, result, and remainder widths
- User-defined delay for output result
- User-selectable modes:
  - *Continuous*—Pipelined for high speed; supports multiple calculations and uses more resources
  - *Single*—Uses fewer resources; you can specify the number of calculation units
- Verilog RTL and simulation testbench
- Includes example designs targeting the Trion® T20 BGA256 Development Board and Titanium Ti60 F225 Development Board

### FPGA Support

The Integer Square Root core supports all Trion® and Titanium FPGAs.

# Resource Utilization and Performance



**Note:** The resources and performance values provided are just guidance and change depending on the device resource utilization, design congestion, and user design.

## Titanium Resource Utilization and Performance

FPGA	Mode	Logic and Adders	Flip-flops	Memory Blocks	DSP48 Blocks	f <sub>MAX</sub> (MHz) <sup>(1)</sup>	Efinity <sup>®</sup> Version <sup>(2)</sup>
Ti60 F225 C4	Continuous	610	512	0	0	429	2021.2
	Single	189	126	0	0	350	

## Trion Resource Utilization and Performance

FPGA	Mode	Logic Utilizations (LUTs)	Registers	Memory Blocks	Multipliers	f <sub>MAX</sub> (MHz) <sup>(1)</sup>	Efinity <sup>®</sup> Version <sup>(2)</sup>
T20 BGA256 C4	Continuous	560	730	0	0	156	2021.1
	Single	192	126	0	0	112	

<sup>(1)</sup> Using default parameter settings.

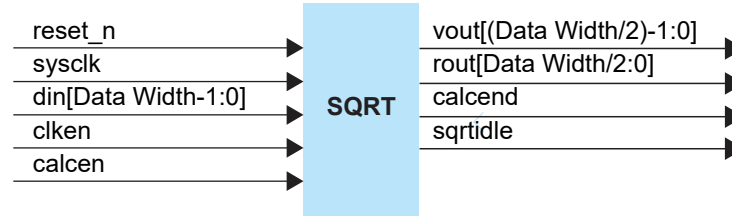
<sup>(2)</sup> Using Verilog HDL.

# Functional Description

The relationship between the integer I, square root result Q, and remainder R is given by these equations:

- $I = Q^2 + R$ , where I, Q, and R are integers  $\geq 0$
- Q is the largest possible value

**Figure 1: Integer Square RootSystem Block Diagram**



## Ports

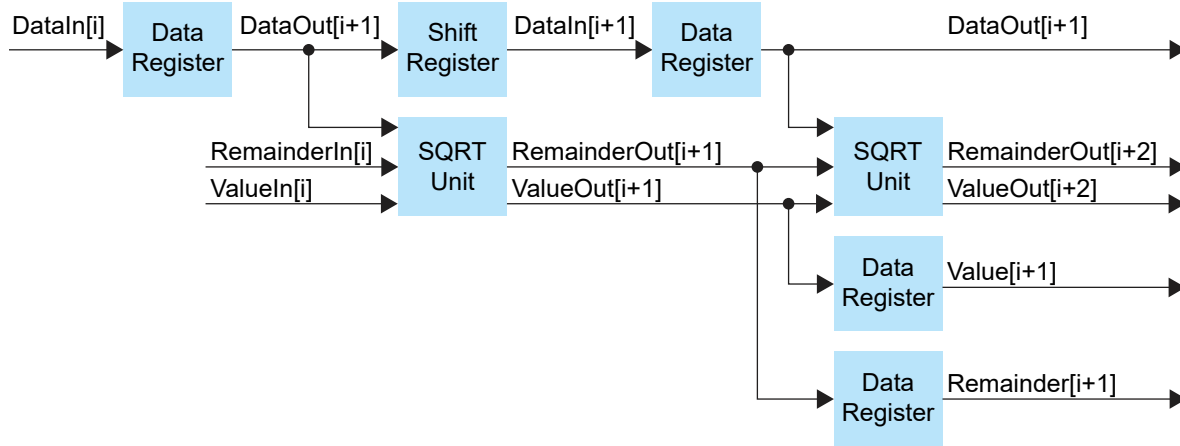
**Table 1: Integer Square Root Core Ports**

Port	Direction	Description
reset_n	Input	Asynchronous system reset 1: Core is active 0: Reset
sysclk	Input	System clock input
din [Data Width-1:0]	Input	Input integer number
calcen	Input	1: Start calculation 0: Stop calculation
clken	Input	1: Enable clock to latch registers 0: Disable clock to latch
vout [(Data Width/2)-1:0]	Output	Square root result
rout [Data Width/2:0]	Output	Remainder
calcend	Output	1: Calculation completed 0: Calculation in progress or no calculation
sqrtidle	Output	1: Core is idle 0: Core is busy

## Continuous Mode

The following figure shows the iterative functions the Integer Square Root core performs.

**Figure 2: Integer Square Root Iterative Blocks (Continuous Mode)**



The Sqrt Unit block performs this calculation:

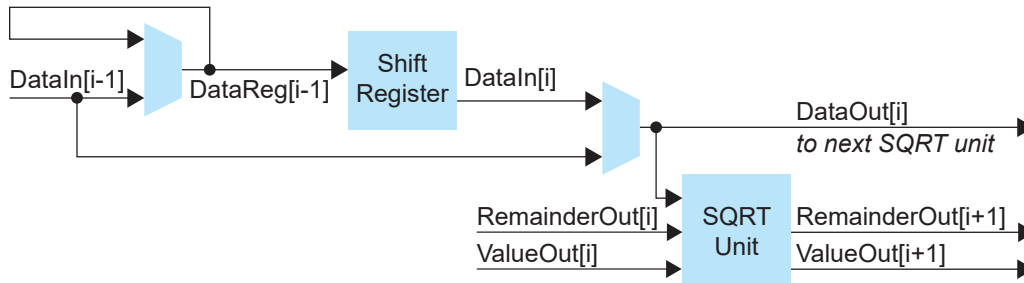
```
When (RemainderIn * 4 + DataIn) > (ValueIn * 4),
    RemainderOut = (RemainderIn * 4 + DataIn) - (ValueIn * 4)
    ValueOut = ValueIn * 4 + 1
Else
    RemainderOut = RemainderIn * 4 + DataIn
    ValueOut = ValueIn * 4
```

The Data Register block determines whether the register is needed on the output for pipelining. The maximum number of pipelines cannot exceed the width of output data (i.e., the input value's width divided by 2). Pipeline registers are evenly distributed in the iterative chain.

## Single Mode

In single mode, the core simplifies the data registers using a multiplexer.

**Figure 3: Integer Square Root Iterative Blocks (Single Mode)**



The Sqrt Unit block performs the same operation as in continuous mode.

You can set the number of calculation units using the `ALUNUMBER_C` parameter. When you use more calculation units, fewer bits are input to each Sqrt Unit block. The Integer Square Root core can run at higher frequencies, but it uses more resources as you increase the number of calculation units.

# IP Manager

The Efinity® IP Manager is an interactive wizard that helps you customize and generate Efinity® IP cores. The IP Manager performs validation checks on the parameters you set to ensure that your selections are valid. When you generate the IP core, you can optionally generate an example design targeting an Efinity development board and/or a testbench. This wizard is helpful in situations in which you use several IP cores, multiple instances of an IP core with different parameters, or the same IP core for different projects.



**Note:** Not all Efinity IP cores include an example design or a testbench.

## Generating a Core with the IP Manager

The following steps explain how to customize an IP core with the IP Configuration wizard.

1. Open the IP Catalog.
2. Choose an IP core and click **Next**. The **IP Configuration** wizard opens.
3. Enter the module name in the **Module Name** box.



**Note:** You cannot generate the core without a module name.

4. Customize the IP core using the options shown in the wizard. For detailed information on the options, refer to the IP core's user guide or on-line help.
5. (Optional) In the **Deliverables** tab, specify whether to generate an IP core example design targeting an Efinity® development board and/or testbench. For SoCs, you can also optionally generate embedded software example code. These options are turned on by default.
6. (Optional) In the **Summary** tab, review your selections.
7. Click **Generate** to generate the IP core and other selected deliverables.
8. In the **Review configuration generation** dialog box, click **Generate**. The Console in the **Summary** tab shows the generation status.



**Note:** You can disable the **Review configuration generation** dialog box by turning off the **Show Confirmation Box** option in the wizard.

9. When generation finishes, the wizard displays the **Generation Success** dialog box. Click **OK** to close the wizard.

The wizard adds the IP to your project and displays it under **IP** in the Project pane.

## Generated Files

The IP Manager generates these files and directories:

- **<module name>\_define.vh**—Contains the customized parameters.
- **<module name>\_tmpl.v**—Verilog HDL instantiation template.
- **<module name>\_tmpl.vhd**—VHDL instantiation template.
- **<module name>.v**—IP source code.
- **settings.json**—Configuration file.
- **<kit name>\_devkit**—Has generated RTL, example design, and Efinity® project targeting a specific development board.
- **Testbench**—Contains generated RTL and testbench files.



**Note:** Refer to the IP Manager chapter of the Efinity® Software User Guide for more information about the Efinity® IP Manager.

# Customizing the Integer Square Root

The core has parameters so you can customize its function. You set the parameters in the General tab of the core's IP Configuration window.

**Table 2: UART Core Parameters**

Parameter	Options	Description
Operation Mode	CONTIN, SINGLE	Indicates the operation mode: CONTIN: Pipelined for high speed; supports multiple calculations and uses more resources. SINGLE: Uses fewer resources; you can specify the number of calculation modules. (Default)
Pipelines	2 - 127	Number of pipelines. Default: 17
Data Width	4 - 256	Input number width. Data Width must be the multiple of 2. When Operation Mode is SINGLE, Data Width must be greater than Calculation Units x 2. Default: 40
Calculation Units	1 - 15	Number of calculation units (for SINGLE mode only). Default: 2



# Integer Square Root Example Design

You can choose to generate the example design when generating the core in the IP Manager Configuration window. Compile the example design project and download the **.hex** or **.bit** file to your board.



**Important:** Efinix tested the example design generated with the default parameter options only.

The example design targets the Trion® T20 BGA256 Development Board and Titanium Ti60 F225 Development Board by implementing a Integer Square Root module in the FPGA.

*Table 3: Titanium Example Design Implementation*

FPGA	Mode	Logic and Adders	Flip-flops	Memory Blocks	DSP48 Blocks	f <sub>MAX</sub> (MHz) <sup>(3)</sup>	Efinity® Version <sup>(4)</sup>
Ti60 F225 C4	Continuous	610	512	0	0	339	2021.2
	Single	139	126	0	0	239	

*Table 4: Trion® Example Design Implementation*

FPGA	Mode	Logic Utilizations (LUTs)	Registers	Memory Blocks	Multipliers	f <sub>MAX</sub> (MHz) <sup>(3)</sup>	Efinity® Version <sup>(4)</sup>
T20 BGA256 C4	Continuous	560	730	0	0	156	2021.1
	Single	192	126	0	0	112	

# Integer Square Root Testbench

You can choose to generate the testbench when generating the core in the IP Manager Configuration window.



**Note:** You must include all **.v** files generated in the **/testbench** directory in your simulation.

Efinix provides a simulation script for you to run the testbench quickly using the Modelsim software. To run the Modelsim testbench script, run `vsim -do modelsim.do` in a terminal application. You must have Modelsim installed in your computer to use this script.

The testbench provides a number of test cases:

- Check the values of the result and remainder for following conditions in continuous mode (**sqr\_ttb\_pl.v**):
  - DataIn = 0
  - DataIn = 1
  - DataIn = 169
  - DataIn = 255
  - DataIn = 257

<sup>(3)</sup> Using default parameter settings.

<sup>(4)</sup> Using Verilog HDL.

- $\text{DataIn} = 2^{32}-1$
- Repeat the test with different pipeline stage parameters
- Repeat the test in simple mode (**sqrt\_tb.v**)

# Revision History

*Table 5: Revision History*

Date	Version	Description
February 2023	3.3	Added note about the resource and performance values in the resource and utilization table are for guidance only.
January 2022	3.2	Updated resource utilization table. (DOC-700)
October 2021	3.1	Added note to state that the $f_{MAX}$ in Resource Utilization and Performance, and Example Design Implementation tables were based on default parameter settings. Updated design example target board to production Titanium Ti60 F225 Development Board and updated Resource Utilization and Performance, and Example Design Implementation tables. (DOC-553)
June 2021	3.0	Added note about including all .v generated in testbench folder is required for simulation. Updated resource utilization and performance table. Updated example design implementation table. Added support for Titanium FPGAs and example design for Titanium Ti60 F225 Development Board. Updated for Efinity v2021.1.
December 2020	2.0	Updated user guide for Efinix® IP Manager which includes added IP Manager topics, updated parameters, and user guide structure.
April 2020	1.1	Updated PIPELINENUM_C parameter values. Updated vout and rout ports width.
October 2019	1.0	Initial release.