



1. Introduction to SOPC Builder

Q1154001-5.1.0

Overview

SOPC Builder is a powerful system development tool for creating systems based on processors, peripherals, and memories. SOPC Builder enables you to define and generate a complete system-on-a-programmable-chip (SOPC) in much less time than using traditional, manual integration methods. SOPC Builder is included in the Quartus® II software and is available to all Altera® customers.

Many designers already know SOPC Builder as the tool for creating systems based on the Nios® II processor. However, SOPC Builder is more than a Nios II system builder; it is a general-purpose tool for creating arbitrary SOPC designs that may or may not contain a processor.

SOPC Builder automates the task of integrating hardware components into a larger system. Using traditional system-on-chip (SOC) design methods, you had to manually write top-level HDL files that wire together the pieces of the system. Using SOPC Builder, you specify the system components in a graphical user interface (GUI), and SOPC Builder generates the interconnect logic automatically. SOPC Builder outputs HDL files that define all components of the system, and a top-level HDL design file that connects all the components together. SOPC Builder generates both Verilog HDL and VHDL equally, and does not favor one over the other.

In addition to its role as a hardware generation tool, SOPC Builder also serves as the starting point for system simulation and embedded software creation. SOPC Builder provides features to ease writing software and to accelerate system simulation.

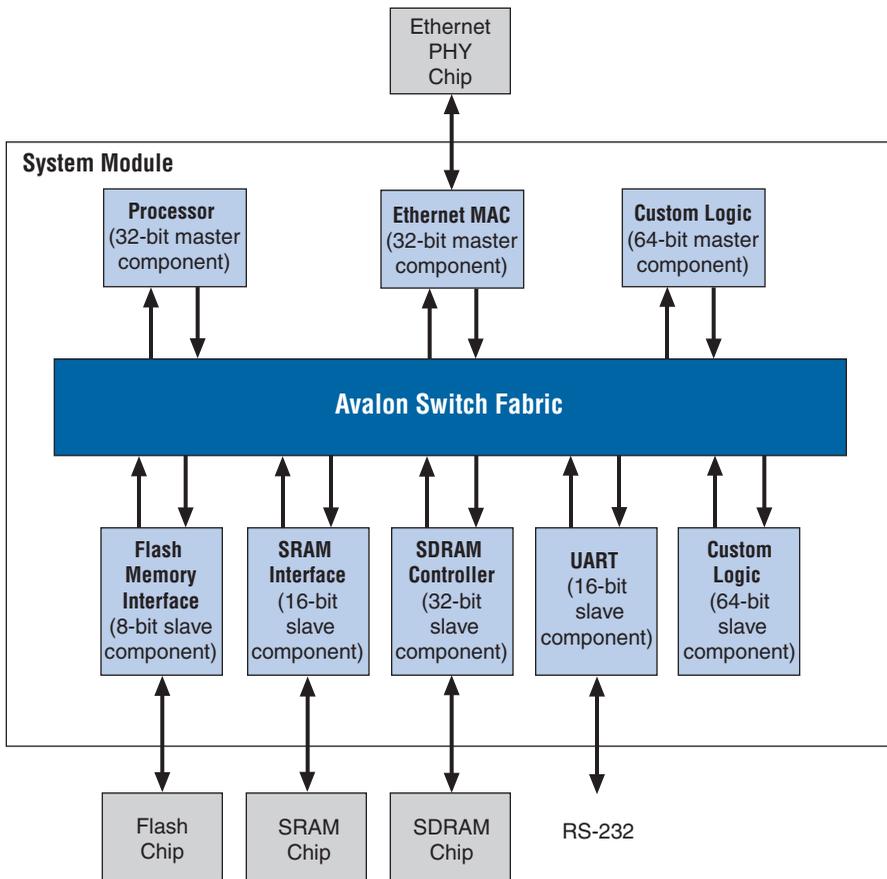
This chapter introduces you to the architectural structure of systems built with SOPC Builder, and describes the primary functions of SOPC Builder.

Architecture of SOPC Builder Systems

This section describes the fundamental architecture of an SOPC Builder system.

An SOPC Builder *component* is a design module that SOPC Builder recognizes and can automatically integrate into a system. SOPC Builder connects multiple components together to create a top-level HDL file called the *system module*. SOPC Builder generates *Avalon® switch fabric* that contains logic to manage the connectivity of all components in the system. [Figure 1–1](#) shows an example of a multi-master system module with Avalon switch fabric connecting the multiple master and slave components.

Figure 1–1. Example of a System Module Generated by SOPC Builder



SOPC Builder Components

SOPC Builder components are the building blocks of the system module. SOPC Builder components use the Avalon interface for the physical connection of components, and you can use SOPC Builder to connect any logical device (either on-chip or off-chip) that has an Avalon interface. The Avalon interface uses an address-mapped read/write protocol that allows master components to read and/or write any slave component.



For details on the Avalon interface, see the *Avalon Interface Specification* at www.altera.com.

A component can be a logical device that is entirely contained within the system module, such as the processor component in [Figure 1-1 on page 1-2](#). Alternately, a component can act as an interface to an off-chip device, such as the SRAM interface component in [Figure 1-1 on page 1-2](#). In addition to the Avalon interface, a component can have other signals that connect to logic outside the system module. Non-Avalon signals can provide a special-purpose interface to the system module, such as the Ethernet MAC in [Figure 1-1 on page 1-2](#).

A component can be instantiated more than once per design.

Altera and third-party developers provide ready-to-use SOPC Builder components, such as:

- Microprocessors, such as the Nios II processor
- Microcontroller peripherals
- Timers
- Serial communication interfaces, such as a UART and a serial peripheral interface (SPI)
- General purpose I/O
- Digital signal processing (DSP) functions
- Communications peripherals
- Interfaces to off-chip devices
 - Memory controllers
 - Buses and bridges
 - Application-specific standard products (ASSP)
 - Application-specific integrated circuits (ASIC)
 - Processors

SOPC Builder Ready Components

Altera awards the SOPC Builder Ready certification to intellectual property (IP) designs that have plug-and-play integration with SOPC Builder. These functions may be accompanied by software drivers, low-level routines, or other software design files.

Altera's OpenCore® and OpenCore Plus evaluation programs allow you to "test drive" an SOPC Builder component both in simulation and in hardware before you buy. You can download evaluations of Altera IP functions directly from www.altera.com/IPMegastore. For IP functions provided by third-party vendors, contact the vendor directly to obtain an OpenCore evaluation.



Check the Altera web site at www.altera.com for up-to-date information about available SOPC Builder Ready components. You can identify SOPC Builder Ready components by the logo shown in [Figure 1-2](#).

Figure 1-2. The SOPC Builder Ready Certification Logo



User-Defined Components

SOPC Builder provides an easy method for you to develop and connect your own components. With the Avalon interface, user-defined logic need only adhere to a simple interface based on address, data, read-enable, and write-enable signals.

You use the following design flow to integrate custom logic into an SOPC Builder system:

1. Define the interface to the user-defined component.
2. If the component logic resides on-chip, write HDL files describing the component in either Verilog HDL or VHDL.
3. Use the SOPC Builder component editor wizard to specify the interface and optionally package your HDL files into an SOPC Builder component.
4. Instantiate your component in the same manner as other SOPC Builder Ready components.

Once you have created an SOPC Builder component, you can reuse the component in other SOPC Builder systems, and share the component with other design teams.



For instructions on developing a custom SOPC Builder component, see the *Developing SOPC Builder Components* chapter in Volume 4 of the *Quartus II Handbook*. For complete detail on the file structure of a component, see the *SOPC Builder Components* chapter in Volume 4 of the *Quartus II Handbook*. For details on the SOPC Builder component editor, see the *Component Editor* chapter in Volume 4 of the *Quartus II Handbook*.

Avalon Switch Fabric

The Avalon switch fabric is the glue that binds SOPC Builder-generated systems together. The Avalon switch fabric is the collection of signals and logic that connects master and slave components, including address decoding, data-path multiplexing, wait-state generation, arbitration, interrupt controller, and data-width matching. SOPC Builder generates the Avalon switch fabric automatically, so that you do not have to manually perform the tedious, error-prone task of connecting hardware modules.

The purpose of SOPC Builder is to abstract away the complexity of interconnect logic, allowing designers to focus on the details of their custom components and the high-level system architecture. Automatically generating the Avalon switch fabric is the keystone to achieving this purpose. Avalon switch fabric in the system module is like air for humans: Its existence is essential, but largely ignored. Because SOPC Builder generates Avalon switch fabric automatically, most users do not interact directly with it or the HDL that describes it.



For further details, see the *Avalon Switch Fabric* chapter in Volume 4 of the *Quartus II Handbook*.

Functions of SOPC Builder

This section describes the fundamental functions of SOPC Builder.

Defining & Generating the System Hardware

The purpose of the SOPC Builder GUI is to allow you to easily define the structure of a hardware system, and then generate the system. The GUI is designed for the tasks of adding components to a system, configuring the components, and specifying how they connect together.

After you add all components and specify all necessary system parameters, SOPC Builder is ready to generate the Avalon switch fabric and output the HDL files that describe the system. During system generation, SOPC Builder outputs the following items:

- An HDL file for the top-level system module and for each component in the system
- A Block Symbol File (.bsf) representation of the top-level system module for use in Quartus II Block Diagram Files (.bdf)
- (Optional) Software files for embedded software development, such as a memory-map header file and component drivers
- (Optional) Testbench for the system module and ModelSim® simulation project files

After you generate the system module, it can be compiled directly by the Quartus II software, or instantiated in a larger FPGA design.



For more detail on the SOPC Builder GUI for defining and generating systems, see the *Tour of the SOPC Builder User Interface* chapter in Volume 4 of the *Quartus II Handbook*.

Creating a Memory Map for Software Development

For each microprocessor in the system, SOPC Builder optionally generates a header file that defines the address of each slave component. In addition, each slave component can provide software drivers and other software functions and libraries for the processor.

The process for writing software for the system depends heavily on the nature of the processor in the system. For example, Nios II processor systems use Nios II processor-specific software development tools. These tools are separate from SOPC Builder, but they do use the output of SOPC Builder as the foundation for software development.

Creating a Simulation Model & Testbench

You can simulate your custom systems with minimal effort immediately after generating the system with SOPC Builder. During system generation, SOPC Builder optionally outputs a push-button simulation environment that eases the system simulation effort. SOPC Builder generates both a simulation model and a testbench for the entire system. The testbench includes the following functionality:

- Instantiates the system module
- Drives all clocks and resets appropriately
- Optionally instantiates simulation models for off-chip devices

Getting Started

One of the easiest ways to get started using SOPC Builder is to read the *Nios II Hardware Development Tutorial* which guides you step-by-step in building a microprocessor system, including CPU, memory, and peripherals. This tutorial and other SOPC Builder example designs are included in the Nios II Development Kit, Evaluation Edition. You can download this kit for free from the Altera Download Center at www.altera.com.

