Peter the Great St. Petersburg Polytechnic University
Institute of Computer Science and Cybersecurity
Graduate School of Computer Technologies and Information Systems

Lecture: Nios II Software Tools and Design Flow

Subject: Automation of discrete device design (in English)

Completed by student of group 5130901/10101 _____ Nepomnyaschiy M.T.
(signature)

Lecturer _____ Antonov A.P.
(signature)

Saint Petersburg
2024

Table of contents

List of illustrations

Creating a System Design with Platform Designer

This lecture provides information on the Nios II Software and Design Flow, detailing the tools and processes for developing software and designing systems around the Nios II processor.

Part 2 – Nios Software Design Overview

2.1. Nios II Processor

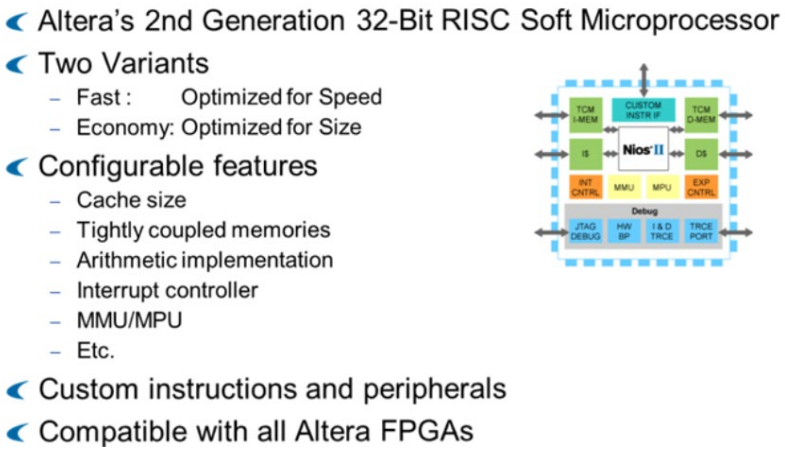

Figure 1 – Nios II Processor

The Nios II Processor is a type of microprocessor developed by Altera, now part of Intel. It can be programmed onto various Altera Field-Programmable Gate Arrays (FPGAs). This processor is designed to be highly adaptable and comes in two versions: one optimized for speed and the other for minimizing space usage. It's commonly used in embedded systems and is categorized as a powerful microcontroller-class processor.

One of the key features of the Nios II Processor is its configurability. Users have the flexibility to customize various aspects such as cache sizes for both instruction and data, enable additional memory ports for faster access, choose different implementations for arithmetic and interrupt control, and even incorporate memory management or protection units as needed. Moreover, since it's a softcore processor, developers can create and integrate custom instructions and peripherals to enhance performance and delegate specific tasks efficiently.

2.2. Nios II Processor System – Design Flow

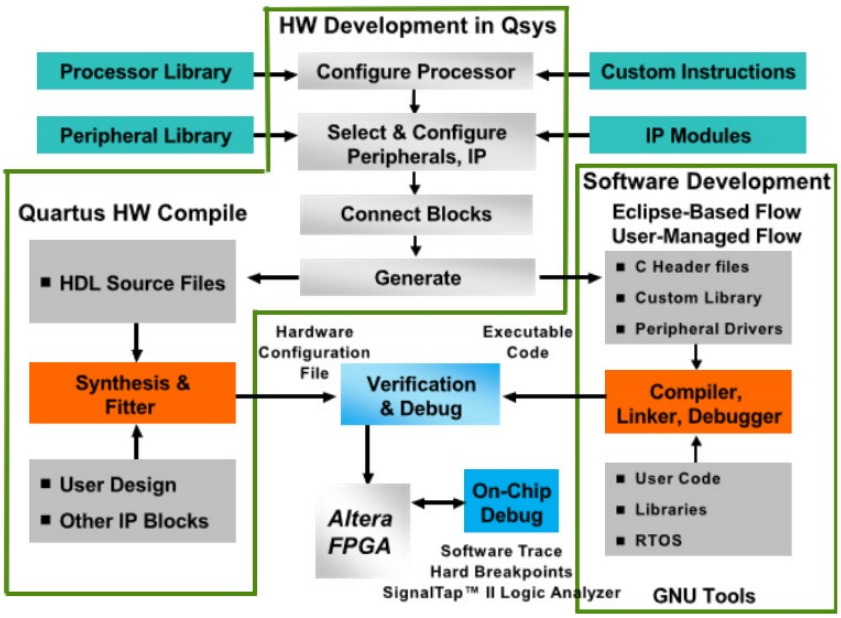

Figure 2 – Nios II (Design Flow)

The system design flow for a Nios II processor-based system starts by setting up the processor's options using the Qsys tool in Quartus Prime software.

In Qsys, users define custom instructions, peripherals, and other components connected to the processor. Once everything is configured, these components are linked to the Nios II processor's core for processing data. This integrated system is called the Nios Qsys System, serving as a foundation for both hardware and software development.

On the hardware side, Quartus Prime software converts the configured system into Hardware Description Language (HDL). This HDL is then compiled to create a specific FPGA hardware configuration file. This file is loaded onto the FPGA, either directly or by storing it in the flash memory. Additionally, Qsys generates files for software development, aiding in the creation of a customized Board Support Package.

For software development, projects are created using these generated files. This allows for the inclusion of drivers, libraries, and C source code. Developers can also add operating systems and middleware as needed. After compiling the software using the GNU compiler, debugging occurs on the physical chip. Finally, the flash memory is programmed with the completed FPGA image software. This comprehensive process ensures the creation of a fully functional and optimized Nios II processor-based system tailored to specific project requirements.

2.3. Nios II Embedded Design Suite

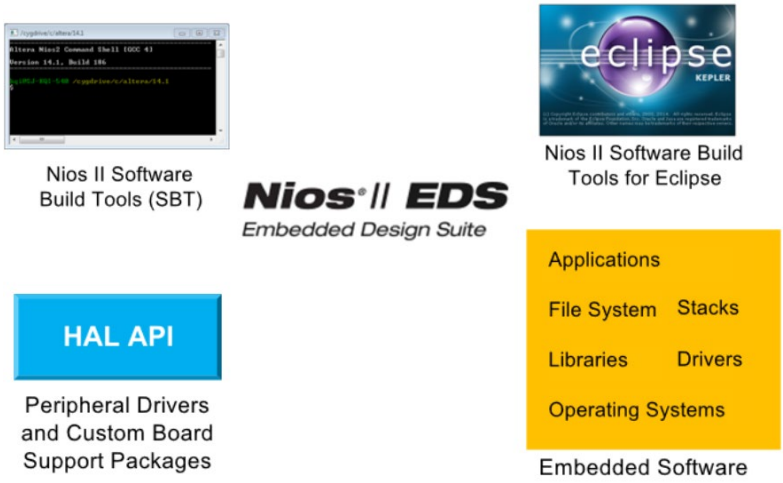

Figure 3 – Nios II Embedded Design Suite

The Nios II Embedded Design Suite (EDS) serves as the core tool for software development and debugging in Nios II processor-based systems.

EDS provides all the necessary functionalities for software development tasks. Within the suite, the Software Build Tools facilitate program compilation, with the Nios SBT for Eclipse offering a graphical user interface (GUI) for ease of use. These tools automatically generate a board support package (BSP) containing essential device drivers and component support. The Hardware Abstraction Layer (HAL) API simplifies component interaction through a Unix-like interface.

Additionally, developers can integrate various off-the-shelf embedded software such as operating systems, network stacks, and file systems into their projects. The Software Build Tools enable straightforward debugging via a JTAG connection, featuring support for breakpoints, run control, and views for registers and memory. With its comprehensive features, the Nios II EDS stands as the singular tool necessary for Nios II software development and debugging needs.

Part 3 – Tools and Design Flow

3.1. Nios II Embedded Design Suit Features

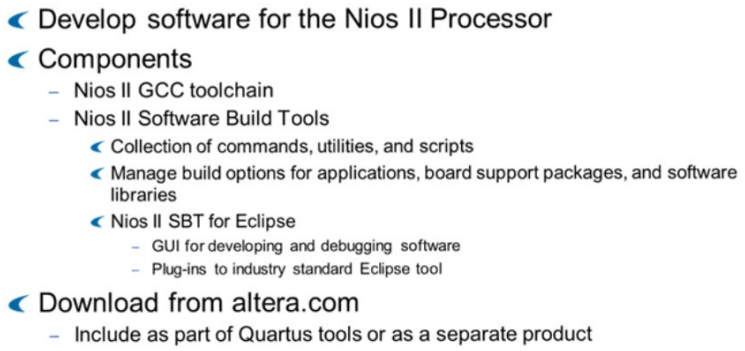

Figure 4 – Nios II Embedded Design Suit Features

To develop software for the Nios II processor, you mainly need the Nios II EDS. This suite includes various tools, both proprietary and open-source, such as the GNU C/C++ toolchain, which you use to create programs for Nios II. The heart of the Nios II EDS is the Software Build Tools, which help you build applications, board support packages, and software libraries using commands, utilities, and scripts. The Nios II SBT for Eclipse is a part of this suite, and it provides plugins for Eclipse tools, making software development and debugging easier. You can get the Nios II EDS for free from altera.com, and it comes bundled with the Quartus software download, so you don't have to worry about extra costs.

3.2. Nios II Software Build Tools Features

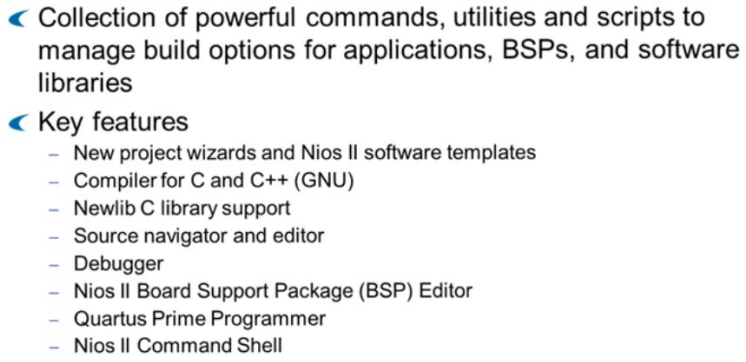

Figure 5 – Nios II Software Build Tools Features

The main component of the Nios II EDS is the Software Build Tools (SBTs), which automate the creation of board support packages (BSPs) for Nios II processor-based systems. BSPs provide a C/C++ runtime environment, shielding developers from hardware complexities. Altera BSPs include the hardware abstraction layer (HAL), optional real-time operating systems (RTOS), and device drivers for system peripherals. There are two flows to utilize these tools: Command Line and Eclipse-based. The Eclipse-based flow operates within the Eclipse environment, enabling software development tasks with GUI convenience and plugins like graphical BSP editor and Quartus Prime programmer. The Command Line flow involves typing Nios II SBT commands directly or embedding them in scripts, utilizing the Nios II Command Shell for tasks like program modification, building, and execution. Both flows offer detailed control over the software build process and share the same utilities and scripting capabilities.

3.3. Integration with FPGA Hardware Tools

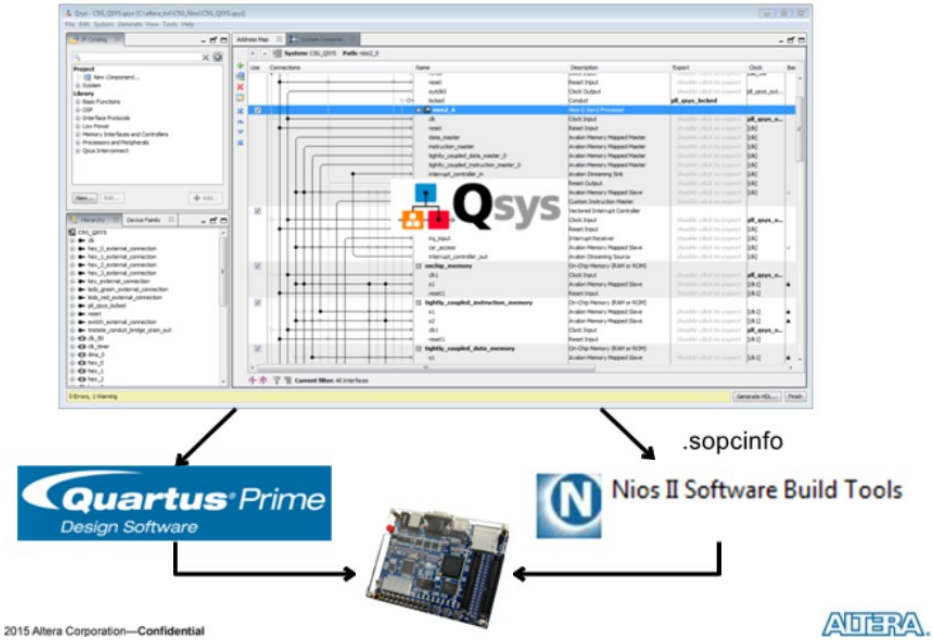

Figure 6 – Integration with FPGA Hardware Tools

Before we talk about software tools, it's important to understand how hardware is set up. In Nios II systems, hardware is created using a tool called Qsys. Here, you set up the Nios II processor and its accessories. Once you're done, Qsys generates files that are used to make the hardware work (HDL files). These files are then processed in another tool called QP Design Software. After this, you get a special file that contains important info about the system. This info helps software tools to set up everything correctly. Once both hardware and software are ready, they can be put onto the board, and you can see your software running on the Nios II processor in the FPGA.

3.4. Nios II Software Build Tools Flow

Figure 7 – Nios II Software Build Tools Flow

There are some steps to create and debug a software application for the Nios II processor using the Software Build Tools.

First, create a software project. Then, develop the software, customizing the board support package if needed. Configure the FPGA with Quartus-generated bitstream. Debug the software using JTAG for typical visibility into CPU and control over execution. Finally, once both the software and hardware are complete, you can program them onto the onboard flash for permanent storage.

3.5. Create Software Projects

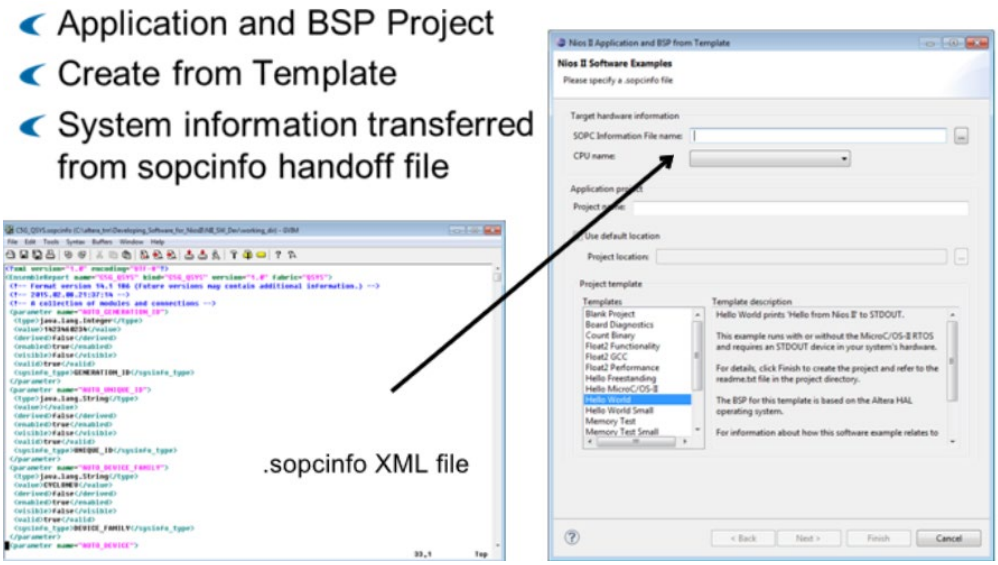

Figure 8 – Create Software Projects

To create a software project in Eclipse for the Nios II processor, it's important to start by using a template. This template simplifies the process by asking for the location of your .sopcinfo file, which is generated when you set up your hardware system using Qsys. The .sopcinfo file, in XML format, describes your hardware system and helps configure the software. Using this information, the tool generates a system library called the Board Support Package (BSP) tailored to your specific hardware configuration. Additionally, the tool creates a linker script based on details such as connected memory components. You can customize the BSP by adjusting settings, adding software packages, and specifying peripherals for input and output. When you start building, the Software Build Tools create the BSP system library based on your preferences. This library includes all your chosen options and generates a system header file with parameter definitions.

3.6. Application vs BSP Project

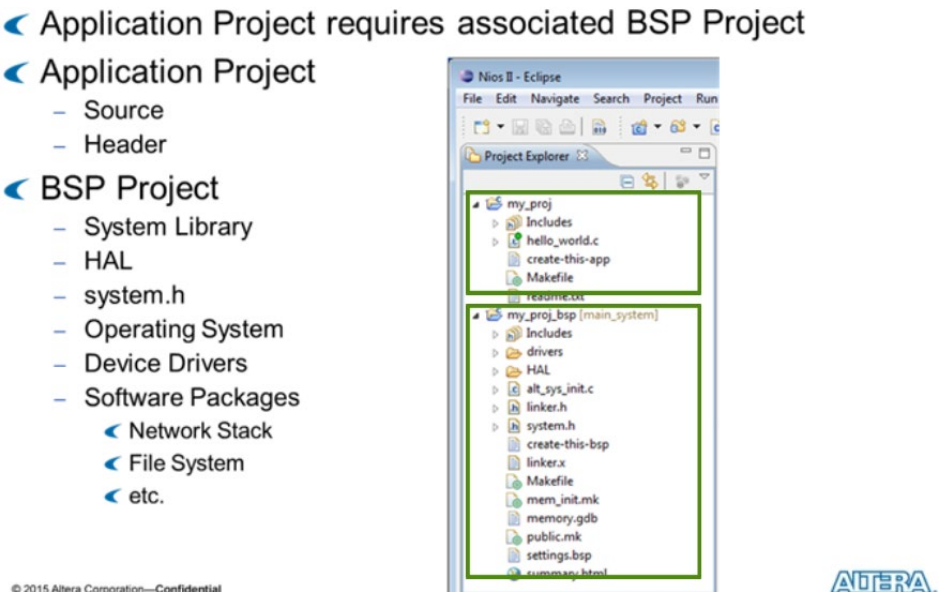

Figure 9 – Application vs BSP Project

Once your project is created in Eclipse, you can explore it using two views: the Nios II view and the Debug view. The Nios II view is the default view, allowing you to navigate through your projects and source code. When you create a project, you'll see two projects in the Project Explorer Pane of Eclipse: the software application and the Board Support Package (BSP). A Nios II C/C++ application project comprises source and header files, along with a makefile. Typically, one source file contains the main() function. The application can include code calling functions from other software libraries and BSPs. The makefile compiles the source code, links it with the BSP and optional libraries, creating an ELF file. On the other hand, a Nios II BSP project serves as a specialized library containing system-specific support code, such as device drivers, operating system files, header files, and the Hardware Abstraction Layer (HAL). Additionally, it includes any added packages like network stacks or file systems. The BSP project source is automatically generated by the Software Build Tools.

3.7. BSP Editor

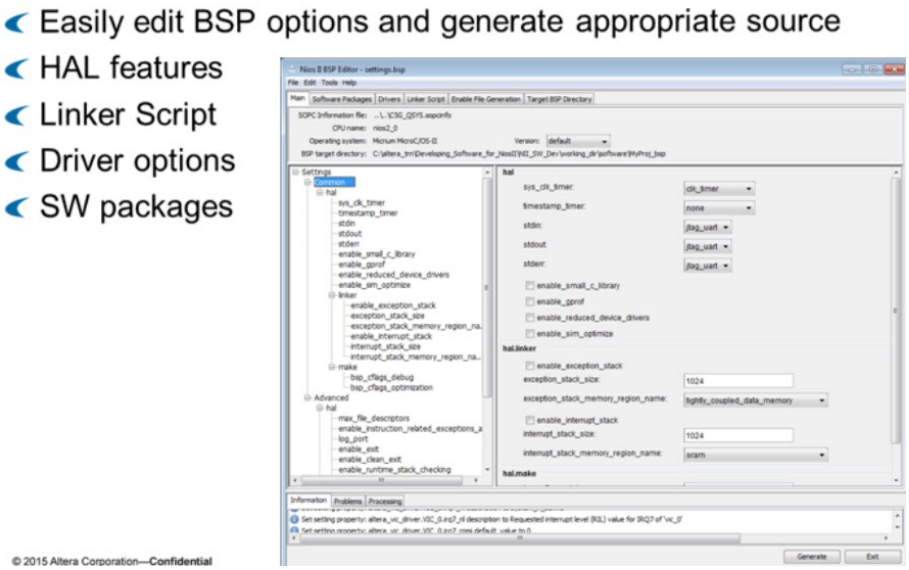

Figure 10 – BSP Editor

The Nios II Software Build Tools feature a BSP Editor to simplify BSP project creation and customization. This tool automatically generates necessary source files for the BSP and allows easy editing and saving. You can choose HAL feature options, set memory regions, and include device drivers for HAL-compliant components. Additionally, software packages like interniche's niche stack can be enabled. Just hit generate, and the BSP project updates with the new files.

3.8. Debug Software Application

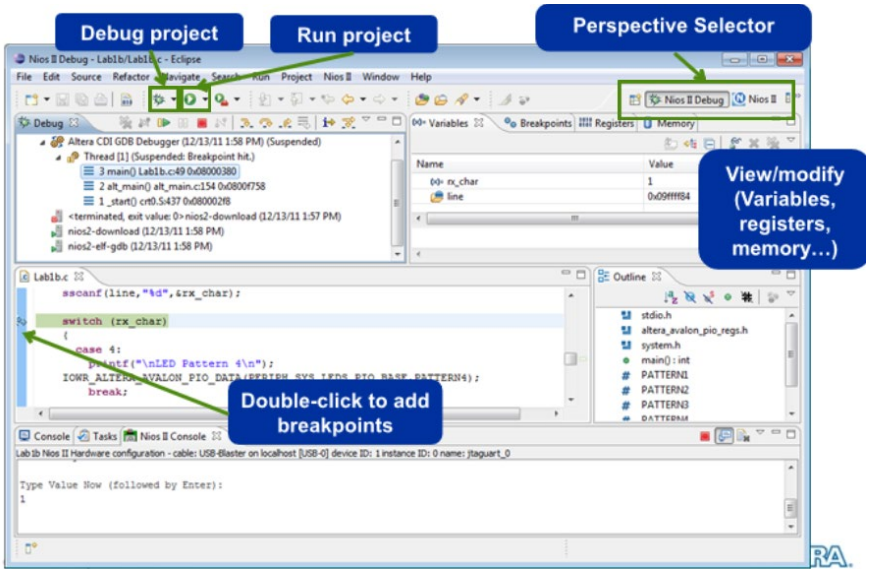

Figure 11 – Debug Software Application

The Nios II EDS for Eclipse makes debugging Nios applications easy. When you open the application, it switches to the Nios II Debug perspective, offering windows for breakpoints, control, and code display. You can control debug sessions and operations like running or stopping code. The interface shows source code and allows adding breakpoints with a click. You can also view variables, registers, and memory. The GUI is customizable, and icons at the top enable configuring debug settings and launching debug sessions, where the software is downloaded and executed on the Nios II Processor while letting you see what's happening in real-time.

3.9. Program to Flash

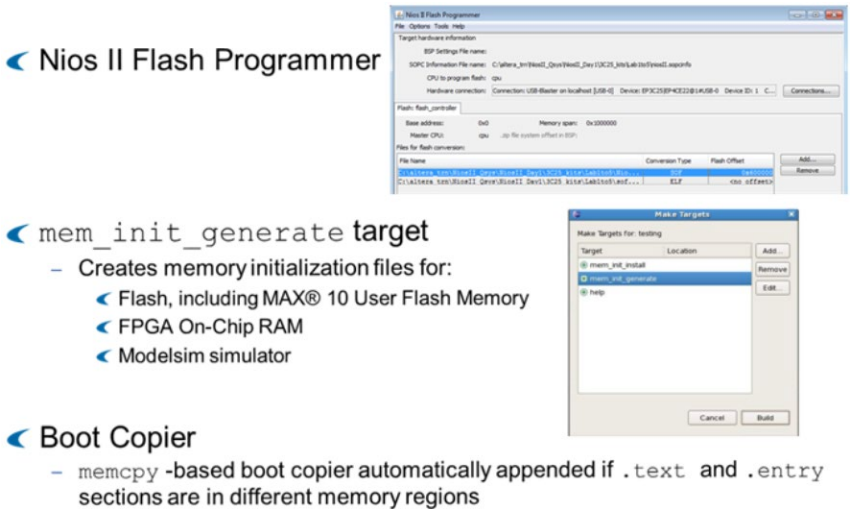

Figure 12 – Program to Flash

The next step after debugging is to flash the board. There are several methods for this, depending on your device.

The Nios II Flash Programmer is a versatile tool that can be used with any Flash component connected to the Nios II Processor. This Eclipse plugin allows you to convert ELF and SOF files into a flash image, which can then be written into the flash memory by the Nios II Processor.

Alternatively, you can use the mem_init_generate feature to generate memory initialization files, especially useful for certain memory components like Max10 User Flash Memory. This process ensures your software runs smoothly, even if it's stored in a different memory component.

3.10. Summary of Nios II SBP Flow

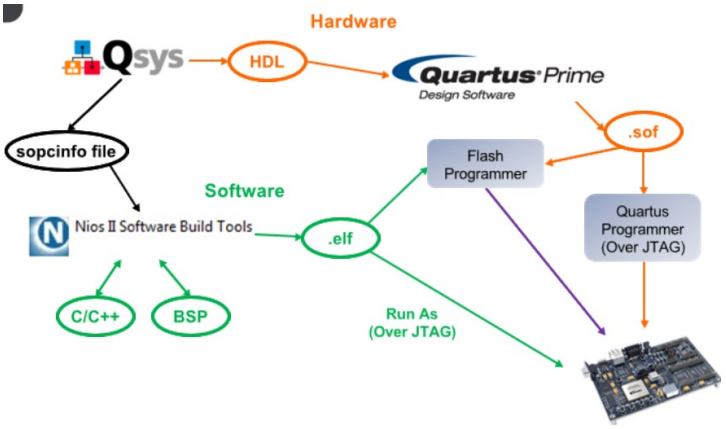

Figure 13 – Summary of Nios II SBP Flow

The Nios II SBT Flow starts with configuring hardware in Qsys and generating a sopcinfo file. SBT for Eclipse uses this to create a BSP supporting custom hardware. You add application files and compile them into an executable. Quartus Prime generates FPGA configuration files for hardware setup. Using Quartus programmer, FPGA is configured and Nios system downloaded. Debugger assists in downloading, running, and debugging applications on hardware. Finally, software executable and FPGA configuration file are used to create a flash image for standalone operation by downloading into the board's flash memory.

Part 4 – Embedded Software And Ecosystem

4.1. Hardware Abstraction Layer

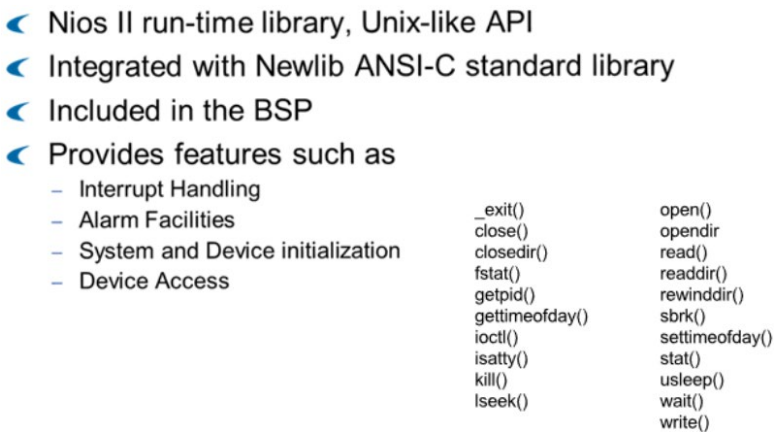

Figure 14 – Hardware Abstraction Layer

The Hardware Abstraction Layer (HAL) provided by the Software Build Tool simplifies interfacing with hardware by offering a Unix-style API. It automatically handles hardware settings and updates related software libraries as needed, eliminating manual adjustments. The HAL provides functions for tasks like interrupt handling, timer management, device access, and system initialization. Additionally, you can easily register custom functions for custom components, enabling straightforward access using C or Unix calls.

4.2. Debugging

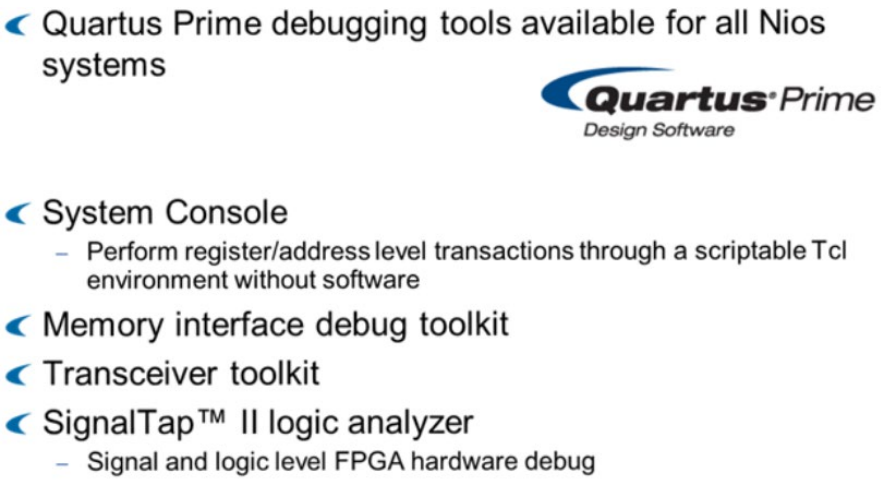

Figure 15 – Debugging

When debugging a Nios II system, Quartus Prime Debugging tools like System Console provide register and address-level access, allowing monitoring system state and performing peripheral reads/writes without software. This enables monitoring system state and performing peripheral reads/writes without software. Additionally, the EMIF Toolkit diagnoses memory interface calibration issues, while the Transceiver Toolkit validates transceiver link signal integrity. The Signal Tap II Logic Analyzer displays internal FPGA logic signals and logic levels. These essential tools are included with Quartus Prime Software.

4.3. Device Driver Support

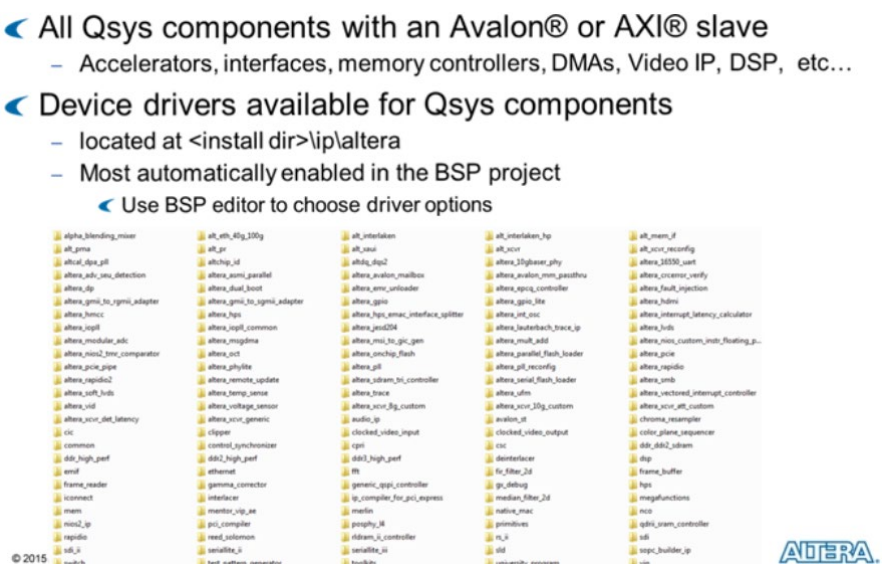

Figure 16 – Device Driver Support

The Nios II Processor can work with a variety of peripherals from Qsys components with Avalon or AXI slave ports. These include accelerators, memory controllers, video IPs, and more. Most off-the-shelf components come with device drivers available in the Quartus installation directory. These drivers are automatically included in your Nios BSP project when you add the component to your Qsys system. You can also customize driver options using the BSP editor if needed.

4.4. Operating Systems

uC/OS-II evaluation included with the Nios II EDS

Operating System	Associated Processor	Supplier
eCos	Nios® II	eCosCentric
eCos	Nios II	Zylin
embOS	Nios II	Segger
Euros RTOS	Nios II	Euros
Linux	Nios II	Timesys
Linux	Nios II	Wind River
Linux	Nios II	SLS
Linux	Nios II	CodeSourcery
Linux	Nios II	Open Source Community
MicroC/OS-II *(1)*	Nios II	Micrium
osCAN *(2)*	Nios II	Vector
ThreadX	Nios II	Express Logic
µCLinux	Nios II	SLS
µCLinux	Nios II	Open Source Community

Figure 17 – Operating Systems

If you decide to use an OS with your Nios II Processor, there are several options available. MicroC/OS-II from Micrium is one of them, and it's included with the Software Build Tool. However, it's worth noting that only the developer license is included. If you plan to ship your product, you'll need to obtain a license from Micrium separately.

4.5. Middleware and Graphic Libraries

Company Name	OS Supported	Network Stack	File System	Graphics Library	USB Stack	Miscellaneous
eCosCentric	eCos	Built in	Built in	-	-	-
Express Logic	ThreadX	NetX *(1)*	FileX *(1)*	PegX *(1)*	USBX *(1)*	-
InterNiche	Any	NicheStack TCP/IP Network Stack – Nios II Edition *(2)*	-	-	-	-
Mentor Graphics	Nucleus Plus	Nucleus Net	Nucleus File *(1)*	Nucleus GRAFIX *(1)*	Nucleus USB *(1)*	-
Micrium	MicroC/OS-II	MicroC/TCP-IP	MicroC/FS	MicroC/GUI	MicroC/USB	MicroC/CAN
Micro Digital	Any	-	-	-	-	GoFast floating-point library
Timesys	Linux	Built in	Built in	-	-	-
Wind River	Linux	Built in	Built in	-	-	-
SLS	µCLinux /Linux	Built in	Built in	-	USB 2.0	-
Community supported(www.alterawiki.com) (Nios Forum area of the Altera Forum)	µCLinux /Linux	Built in	Built in	-	-	-
Altera	Any	-	Read-only zip file system *(3)*	-	-	-

Included with the Nios II EDS

Figure 18 – Middleware and Graphic Libraries

This chart provides an overview of middleware and graphics libraries available from various vendors. Notably, the Read-only Zip File System from Altera and the NicHESTack TCP/IP network stack Nios II Edition are included in the Nios II Software Build Tools (SBT) and can be enabled from the Board Support Package (BSP). However, for all other software packages, it's advised to contact the vendors directly for more information.

4.6. Third Party Software Development Tools

Company	Product	Description
Wind River	Workbench for Nios II embedded processor	Software development tools for embedded Linux on the Nios II processor.
Mentor Embedded	Sourcery CodeBench	GNU toolchain support for embedded Linux on the Nios II processor.
Mentor Embedded	Sourcery CodeBench	GNU toolchain support for the dual-core ARM Cortex-A9 MPCore processor-based SoC Virtual Target.
Altium	Tasking VX-toolset	Optimizing C compiler, assembler, linker, and locator.
MIPS Technologies (formerly FS2)	System Navigator	The System Navigator probe for Nios II processors is designed to support the special features and integrated peripherals of the Nios II cores embedded in Altera FPGAs.
Open-source community	Linux toolchain	Linux toolchain from the open-source community.
Open-source community	µCLinux toolchain	µLinux toolchain from the open-source community.

Figure 19 – Third Party Software Development Tools

Here are some additional third-party software development tools available for the Nios II processor if you need more options beyond Altera's Software Build Tools:

- Windriver Workbench: Suitable for software running on embedded Linux.
- Mentor: Provides GNU toolchain support for embedded Linux.
- Altium and MIPS technologies: Offer tool chains for various environments.
- Open source community: Offers various tool chains for different environments.

Conclusion

The Nios II Software and Design Flow offers a comprehensive set of tools and processes for developing software and designing systems around the Nios II processor. Starting with Qsys for hardware configuration, users can easily create custom systems with various peripherals and components. The Software Build Tools (SBT) provide a user-friendly environment for software development, including the creation of board support packages (BSPs) and integration of operating systems. The debugging tools offer visibility into system hardware, essential for troubleshooting and optimizing performance. Additionally, a wide range of middleware and graphics libraries are available, enhancing functionality. For specialized needs, third-party tools can be integrated into the workflow. Overall, the Nios II Software and Design Flow streamline the development process, allowing users to efficiently design, develop, debug, and deploy software for Nios II-based systems.

In simple terms, the Nios II Software and Design Flow is like a toolbox for building and programming electronic devices. It helps people create the brains (software) and bodies (hardware) of their devices, making sure everything works together smoothly.