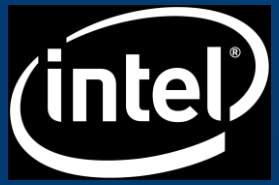


INTRODUCTION TO EMBEDDED DESIGN WITH NIOS

August 2018

OBJECTIVES

Upon completion of this workshop, you should have an understanding of:

1. Platform Designer system development tools - basic features and how to run the tool
2. How to develop applications on the Nios® II processor
3. Eclipse Integrated Development Environment (IDE) and development of “bare metal” applications utilizing Nios and associated FPGA hardware

COMPANION LAB

1. Obtain an FPGA development kit

terasIC
www.terasic.com

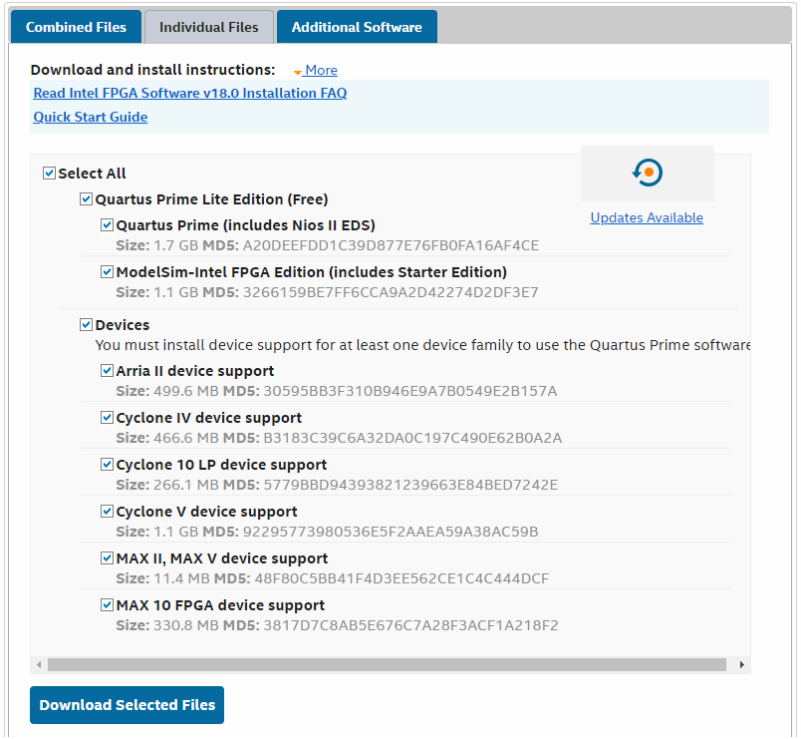
Search for “DE10-Lite”

2. Download the attached lab manual

3. Download Quartus Prime Lite

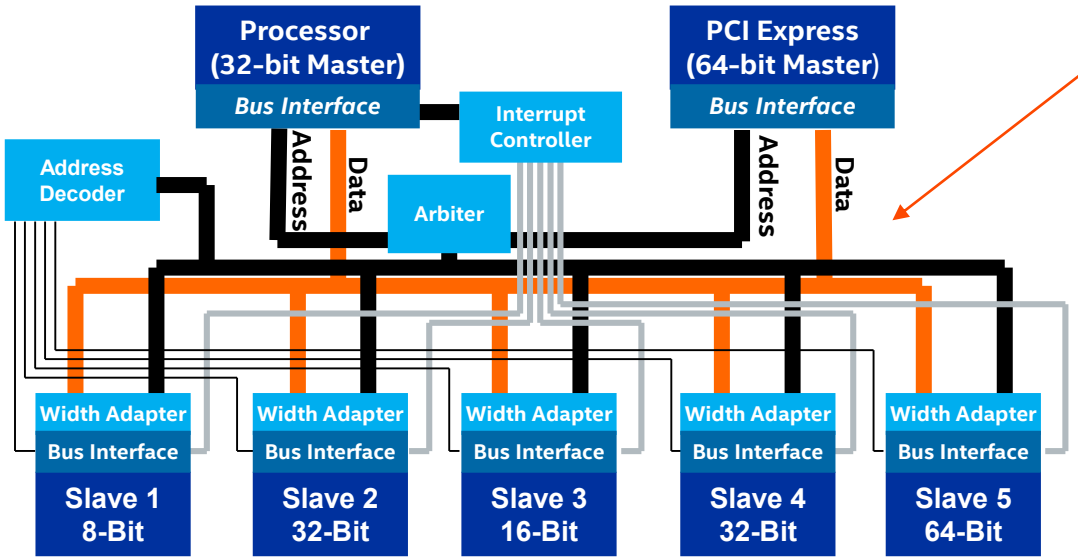
To improve download time, only select the device support that matches the device you are using.

4. Run installation while viewing training



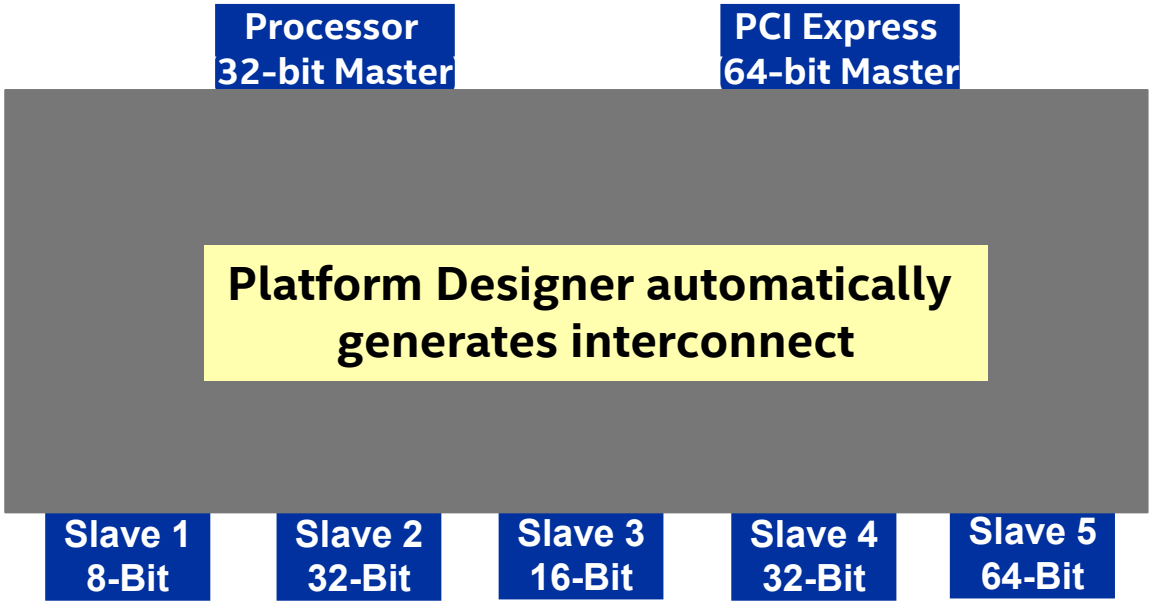
PLATFORM DESIGNER SYSTEM INTEGRATION

TRADITIONAL SYSTEM DESIGN



- Different interfaces (some standard, some non-standard)
- Significant engineering work required to design custom interface logic
- Integrating design blocks and intellectual property (IP) is tedious and error-prone

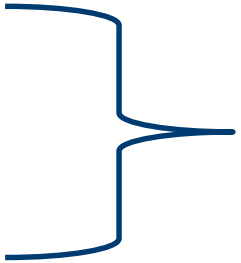
AUTOMATIC INTERCONNECT GENERATION



- Avoids error-prone integration
- Saves development time with automatic logic & HDL generation
- Enables you to focus on value-add blocks

PLATFORM DESIGNER FEATURES

- Easy IP integration with switch fabric connectivity
- Dynamic system generation
- High-level system visualization
- Custom IP authoring
- IP verification and bus functional models (BFMs)
- Simulation support for ModelSim® and other 3rd-party simulation tools
- Real-time system debug through tools like System Console

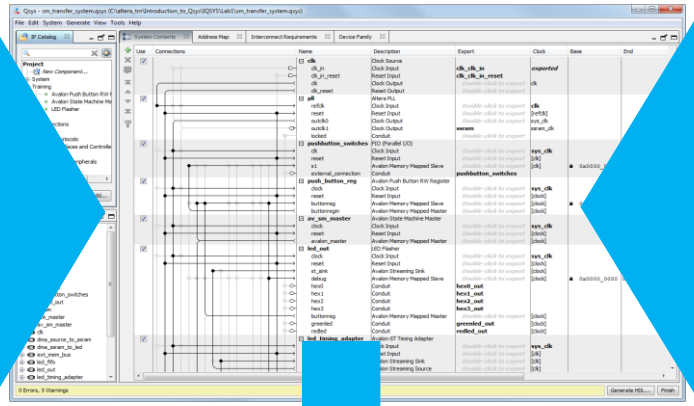


Today's Lab

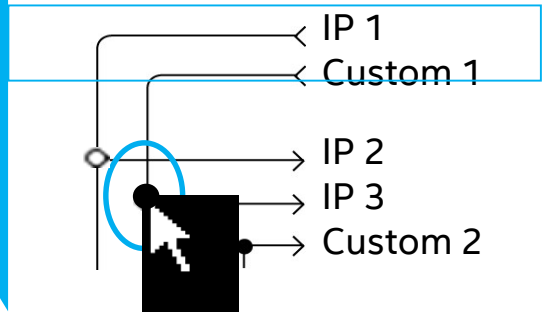
EASY TO USE SYSTEM-INTEGRATION UI

Catalog of available IP

- *Interface protocols*
- *Memory*
- *DSP*
- *Embedded*
- *Bridges*
- *PLL*
- ***Custom systems***



Connect custom IP and systems



Accelerate development

HDL

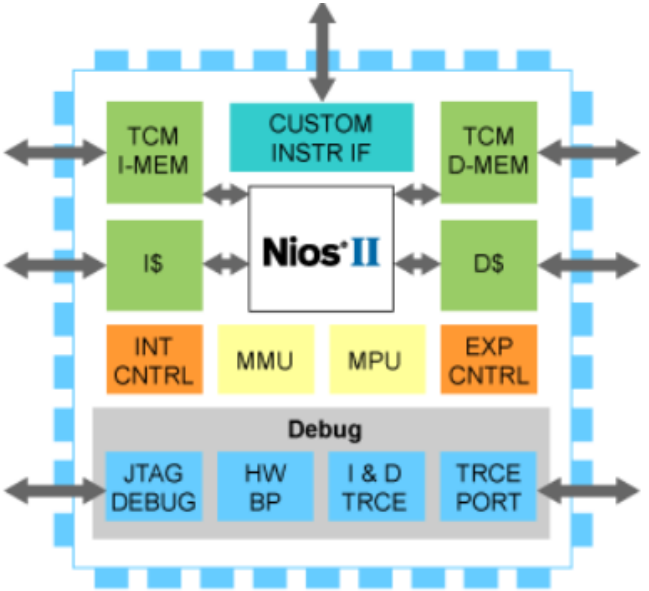
Simplify integration

Automate error-prone integration tasks

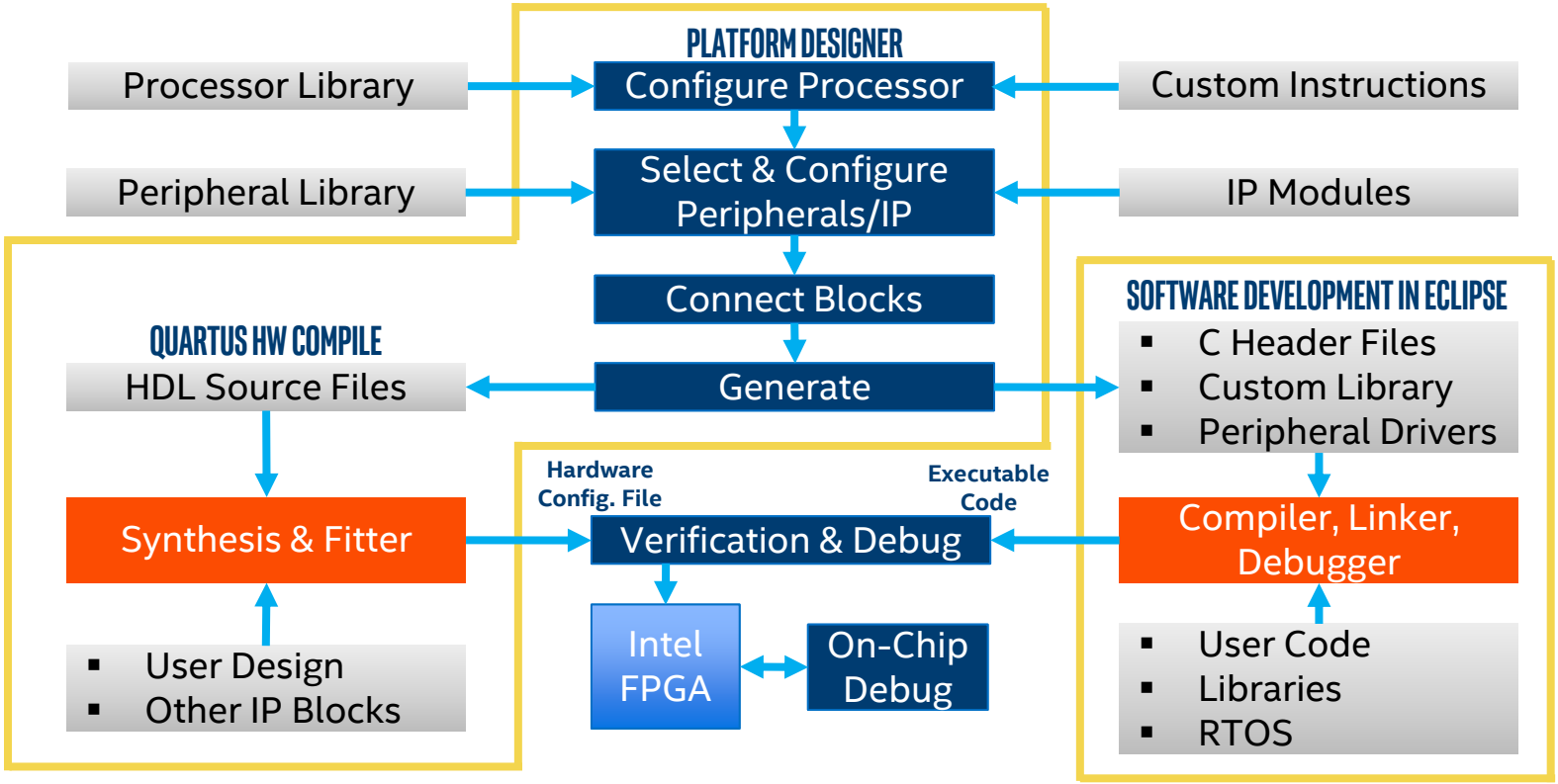
NIOS® II PROCESSOR

NIOS II PROCESSOR

- Intel's 2nd Generation 32-Bit RISC Soft Microprocessor
- Two Variants: Fast - Optimized for Speed; Economy - Optimized for Size
- Configurable features
  - Cache size
  - Tightly coupled memories
  - Arithmetic implementation
  - Interrupt controller
  - MMU/MPU
- Custom instructions and peripherals
- Compatible with all Intel FPGAs



NIOS II DESIGN FLOW



NIOS II EMBEDDED DESIGN SUITE FEATURES

Software Build Tools (SBT)

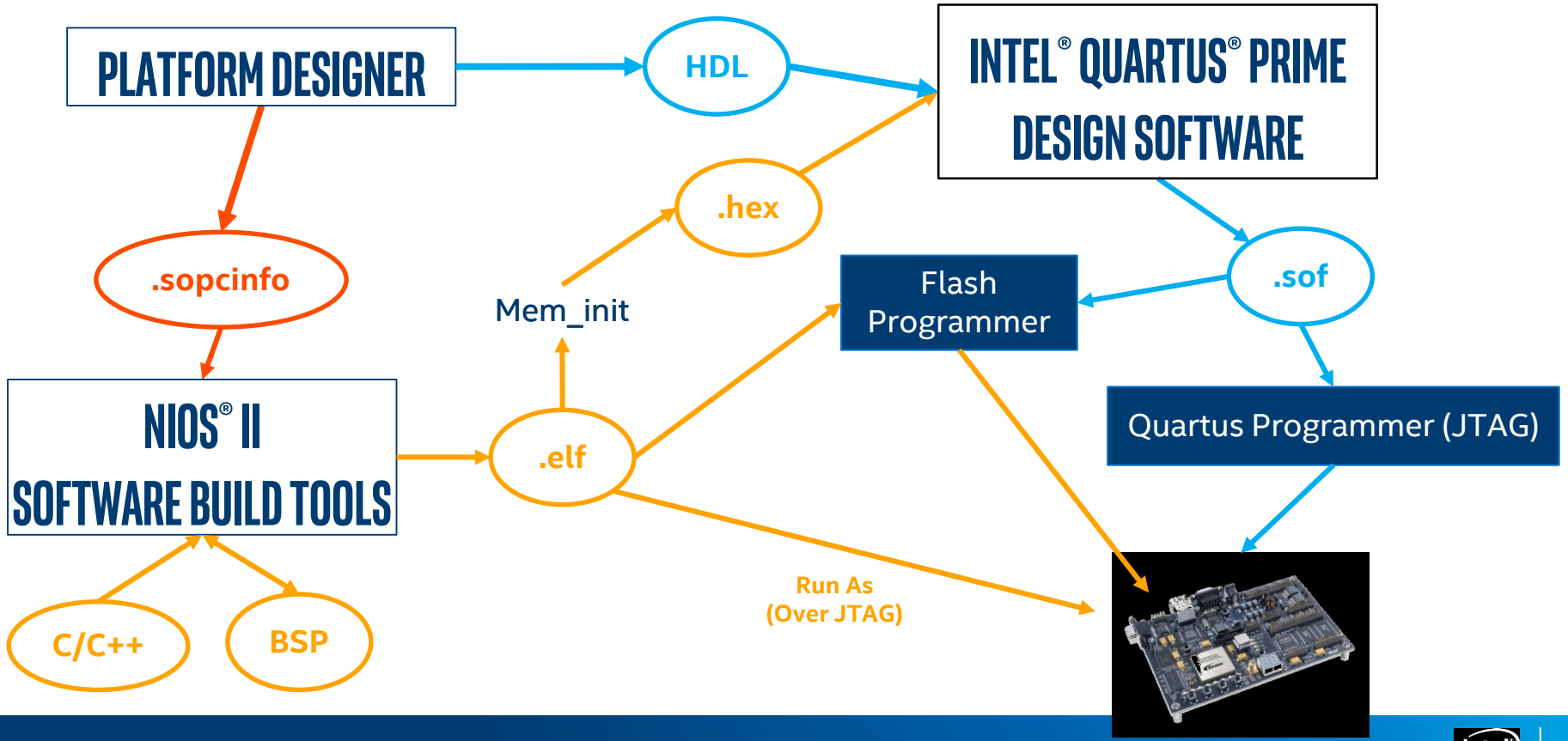
- Proprietary and open-source tools for creating Nios® II programs
- Commands, utilities & scripts
- GUI for developing and debugging software
- Plug-ins to industry standard Eclipse IDE
- Automated board support package (BSP) creation

Hardware Abstraction Layer (HAL)

- Unix-style API
- Interface with peripherals, timers, interrupts, etc.
- Device access and initialization
- All hardware related software libraries automatically updated

_exit()	gettimeofday()	sbrk()	open()
close()	ioctl()	stat()	opendir
closedir()	isatty()	usleep()	read()
fstat()	kill()	wait()	readdir()
getpid()	lseek()	write()	rewinddir()

NIOS II SBT FLOW



ECOSYSTEM & SUPPORT

DEBUGGING

- Quartus Prime debugging tools available for all Nios systems
- System Console
  - Perform register/address level transactions through a scriptable Tcl environment without software
- External memory interface (EMIF) debug toolkit
- Transceiver toolkit
- SignalTap™ II logic analyzer
  - Signal and logic level FPGA hardware debug

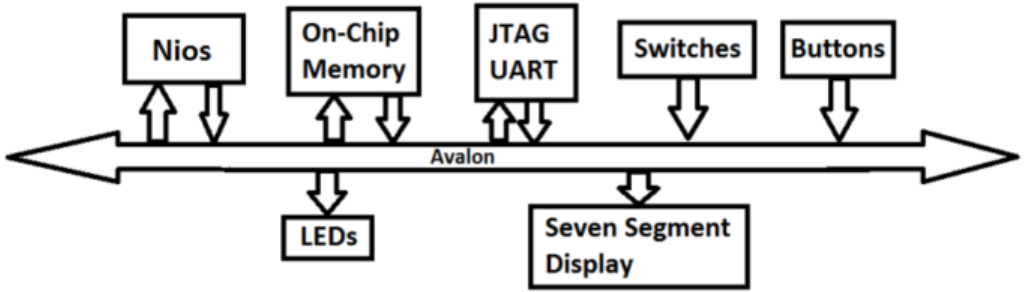
ADDITIONAL NIOS® II RESOURCES

- Nios® II Documentation on intel.com
  - Including One-Click Download zip file of all available documentation
  - Nios® II Hardware and Software Developer's Handbooks
- Intel Forum - Thousands of users and topics
- Intel FPGA Wiki - User generated documentation
- Design Store - Lots of designs (search for Nios, Qsys, or Platform Designer)
- Eclipse Help Content and Welcome Page

LIVE DEMO

EMBEDDED NIOS LAB

1. Build system block diagram in Qsys
2. Connect to a top level Verilog module
3. Export memory map into Eclipse environment
4. Build project and compile code
5. Download hardware.
6. Download software
7. Try some other cool code and download it.



TIPS

1. Follow connections in ***exactly***! Double check inputs vs. outputs.
2. Name things exactly! No spaces in files OR folder names.
3. The instruction and data master is connected to the on chip memory (not just data master)!
4. Remember **Platform Designer → System → Assign Base Addresses** after edits
5. When connecting Eclipse to the board, “Refresh Target” is hidden on the right.
6. To reprogram HW, first stop eclipse from running software (use “stop” button) or the USB blaster will be hung!
7. If HW fails to program, hit Start a second time!
8. Have fun and try more embedded projects on your own!