Online Tutorials

Atom processor
FPGA material
Yocto tools
Atom processor

Based on 32nm process technology, the processor series feature new levels of performance-per-watt opening the door to always-on, always-connected embedded devices. When paired with Intel® NM10 Express chipset, the platform provides rich I/O capabilities and flexibility via high-bandwidth interfaces such as PCI Express, Serial ATA and USB 2.0.

For N2600 (CedarView) information:

Video: Intel® Atom™ Processor N2000 and D2000 series with Intel® NM10 Express Chipset Based Platform Overview

Atom processor

Intel® Rapid Start Technology provides fast resume; Intel® Smart Connect Technology enables an always-updated experience even during standby.

Integrated hardware-accelerated decoder for smooth full HD (up to 1080p) video playback and streaming.

Wide range of display outputs - DisplayPort (eDP/DP), LVDS, VGA and HDMI enabled.

Intel® Deep Power Down Technology significantly reduces power usage during idle periods.

Support for up to 4GB of DDR3 system memory.
Online Tutorials

Atom processor

FPGA material

Yocto tools
Learn how to verify a custom component with Qsys and the Avalon Verification IP Suite. You can use Qsys to generate a testbench system for the design under test and perform a functional simulation with the ModelSim simulator. The Qsys generated testbench uses the Avalon Verification IP Suite components.
Using the Quartus II Software: An Introduction (ODSW1100) 1.5 Hours Online Course
http://www.altera.com/education/training/courses/ODSW1100

Course Description

You will become familiar with the easy-to-use Quartus® II software design environment. You will learn about the steps of the basic FPGA design flow and how to use the Quartus II software in the flow, illustrating the ease of going from design entry to device programming all from within one tool. You will learn about the Tasks window which makes it easy to locate basic functions in the Quartus II software user interface. For example, the Tasks window makes it easy to create new projects, make pin assignments, and locate Quartus II software compilation output information. The Quartus II software version 10.1 is used for demonstration purposes.

At Course Completion

You will be able to:
- Use the New Project Wizard to create new projects
- Use the MegaWizard® plug-in manager to easily create and configure megafunctions
- Review output information in the compilation report
- Control design processing and hardware options using settings and assignments
- Create I/O assignments using the Pin Planner
- Perform a Quartus II software simulation
- Program an Altera® FPGA or CPLD with the Quartus II Programmer
Introduction to Qsys (OQSYS1000)
1.5 Hours Online Course

Course Description

The Qsys system integration tool saves design time and improves productivity by automatically generating interconnect logic to connect intellectual property (IP) functions and subsystems. You will receive an introduction to system design and get an overview of the Qsys system integration tool in the Quartus® II software v. 12.0. You will learn the benefits of standard interfaces, key Qsys features, and how to migrate from SOPC Builder to Qsys. You will learn about the Qsys high-performance interconnect, user interface, and design flow. You will learn how to integrate hierarchy into your design and about creating reusable custom IP. Finally, you will learn about verification, using system console and functional simulation.

At Course Completion

You will be able to:
- Describe the benefits of the Qsys system integration tool
- Recognize the design productivity benefits of using standard interfaces
- Compare Qsys with SOPC Builder
- Create a system in Qsys with the user interface
- Simulate and verify a Qsys system

http://www.altera.com/education/training/courses/OQSYS1000
Getting Started with PCI Express Designs in Altera Transceiver Devices (OPCI1101)

2 Hours Online Course

http://www.altera.com/education/training/courses/OPCI1101

Course Description

In this online course, you will see how to build PCI Express® solutions targeting the Cyclone® IV GX, Arria® II GX/GZ, Stratix® IV GX/GT, and HardCopy® IV GX devices. You will learn about the PCI Express Hard IP blocks and how to customize its layers using the Quartus® II software to create your own PCI Express design. You will see how to integrate a PCI Express solution into an embedded system using the Qsys system integration tool. You may download exercise instructions so you can practice developing and verifying a PCI Express system on your own.

At Course Completion

You will be able to:

- Describe the features and functionality of the PCI Express hard IP block found in select Altera® devices
- Build a PCI Express solution containing the PCI Express hard IP block
- Successfully integrate a PCI Express hard IP block solution into a Qsys system and describe how it interfaces with the Qsys interconnect
Online Tutorials

Atom processor

FPGA material

Yocto tools
Yocto Project
http://www.yoctoproject.org/

Documentation

Documentation for the Yocto Project consists of manuals, the Yocto Wiki, training videos, the Yocto Project website, and documentation for specific tools such as Hob and the Build Appliance. Aside from these sources of information, you can also get a run-down of the current release's features, updates, and known issues by checking out the Release Notes. And, you view a list of commonly asked questions with their answers by looking at the FAQ.
Eclipse IDE Plug-in

The Eclipse IDE Plug-in integrates the functionality of the Yocto Project ADT and toolchain into the Eclipse IDE. This allows its users direct access to these ADT capabilities:

• The cross compiler
• Use of the debugging and profiling tools
• Use of the emulator
GDB, the GNU Project debugger, allows you to see what is going on `inside' another program while it executes -- or what another program was doing at the moment it crashed.

GDB can do four main kinds of things (plus other things in support of these) to help you catch bugs in the act:

Start your program, specifying anything that might affect its behavior.

Make your program stop on specified conditions.

Examine what has happened, when your program has stopped.

Change things in your program, so you can experiment with correcting the effects of one bug and go on to learn about another.

The program being debugged can be written in Ada, C, C++, Objective-C, Pascal (and many other languages). Those programs might be executing on the same machine as GDB (native) or on another machine (remote). GDB can run on most popular UNIX and Microsoft Windows variants.