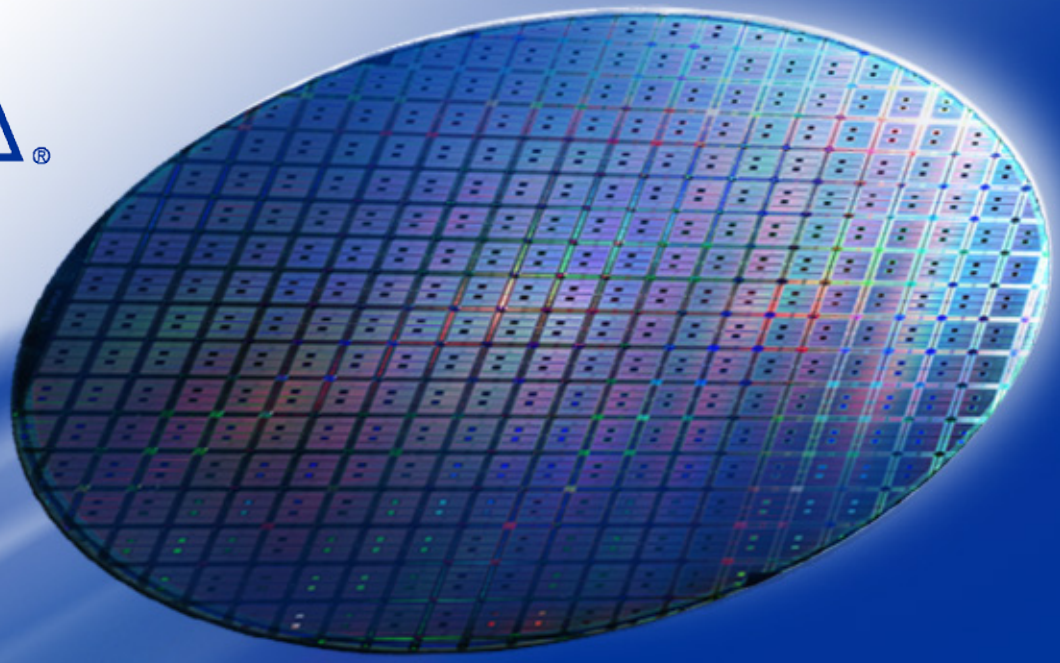


Quartus II Basic Training

Programmable Logic Families

- Structured ASIC
  - HardCopy® II, HardCopy Stratix
- High & Medium Density FPGAs
  - Stratix II, Stratix, APEX™ II, APEX 20K, & FLEX 10K®
- Low-Cost FPGAs
  - Cyclone II & Cyclone
- FPGAs with Clock Data Recovery
  - Stratix II GX
- CPLDs
  - MAX II, MAX 7000 & MAX 3000
- Embedded Processor Solutions
  - Nios II
- Configuration Devices
  - Serial (EPCS) & Enhanced (EPC)

MAX 7000A & MAX 3000A Family Overview

Parameter	MAX 3000A					MAX 7000A				
	EPM3032A	EPM3064A	EPM3128A	EPM3256A	EPM3512A	EPM7032AE	EPM7064AE	EPM7128AE	EPM7256AE	EPM7512AE
Useable Gates	600	1,250	2,500	5,000	10,000	600	1,250	2,500	5,000	10,000
Macrocells	32	64	128	256	512	32	64	128	256	512
Maximum User I/O Pins	34	66	96	158	208	36	68	100	164	212
t_{PD} (ns)	4.5	4.5	5.0	7.5	7.5	4.5	4.5	5.0	5.5	7.5
f_{CNT} (MHz)	227	222	192	127	116	227	222	192	172	116
t_{SU} (ns)	2.9	2.8	3.3	5.2	5.6	2.9	2.8	3.3	3.9	5.6
t_{CO1} (ns)	3.0	3.1	3.4	4.8	4.7	3.0	3.1	3.4	3.5	4.7

Complete Voltage Portfolio

5.0 V

MAX 7000S

- Performance Leader
- Feature Leader
- Wide Range of Package Offerings
- Industrial-Grade Offerings

3.3 V

MAX 7000AE

- High Performance
- Feature Leader
- Wide Range of Package Offerings

MAX 3000A

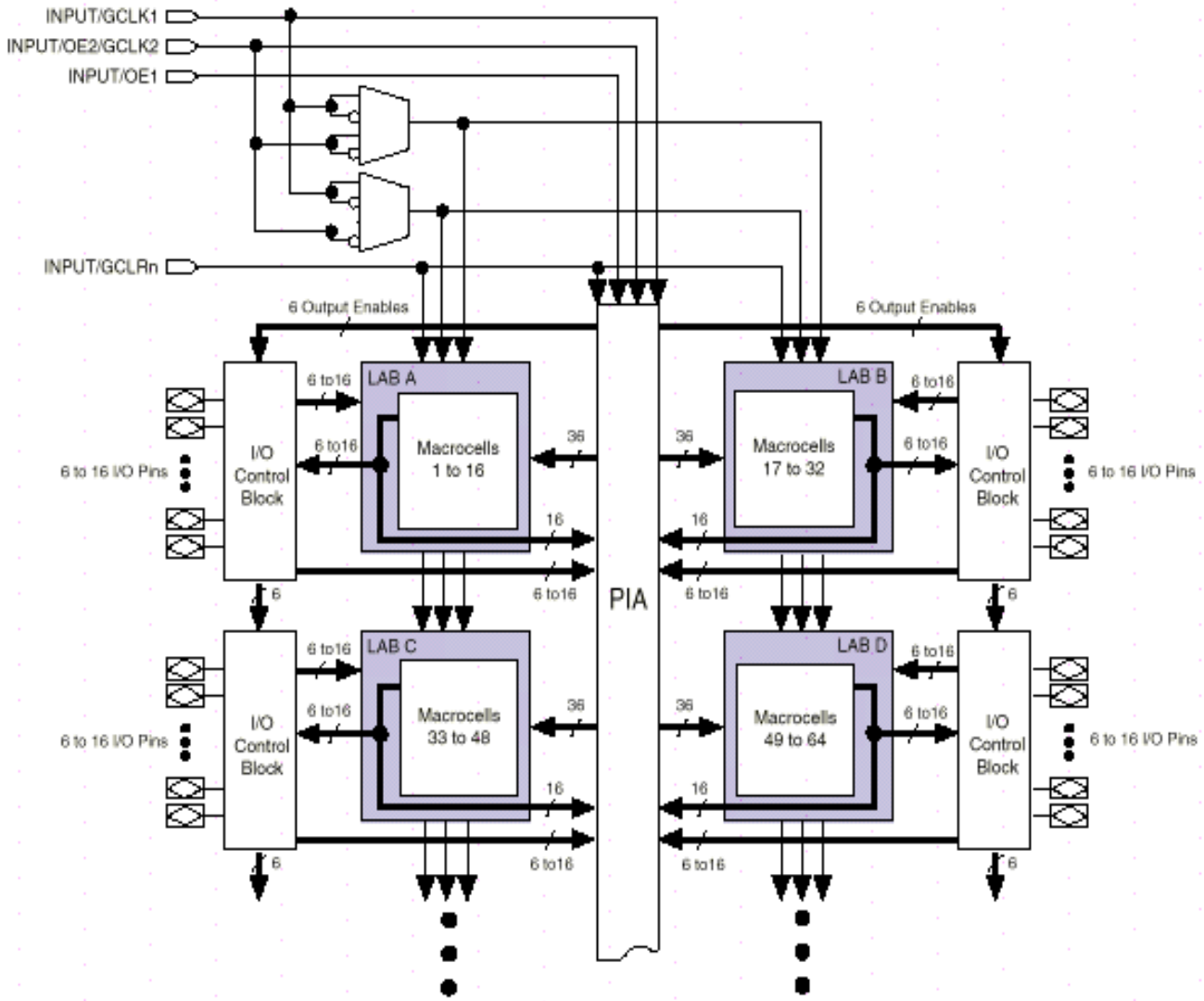
- Price Leader
- Feature & Package Subset of MAX 7000AE

2.5 V

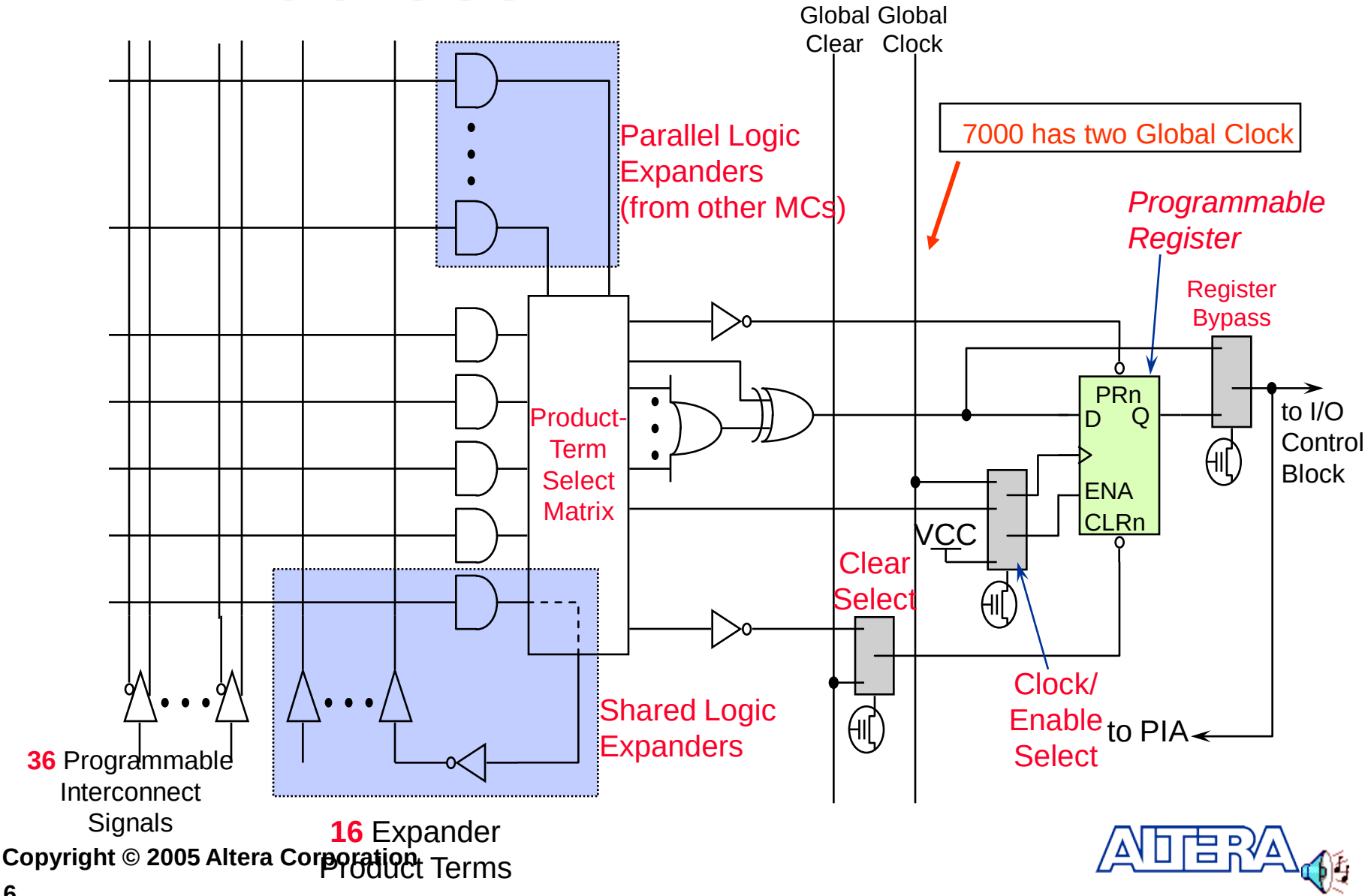
MAX 7000B

- High Performance
- Feature Leader
- Wide Range of Package Offerings

MAX Device Block Diagram



MAX Macrocell



MAX II: The Lowest-Cost CPLD Ever

- New Logic Architecture
  - 1/2 the Cost
  - 1/10 the Power Consumption
  - 2X the Performance
  - 4X the Density
- Non-Volatile, Instant-On
- Supports 3.3-, 2.5- & 1.8-V Supply Voltages

Breakthrough Technology to Expand the Market

Flexible Supply Voltage

- On-Chip Voltage Regulator
- Accepts 3.3-, 2.5- & 1.8-V Supply Inputs
- Internally Converted to 1.8-V Core Voltage

Convenience of 3.3 V with the Power & Performance of 1.8 V

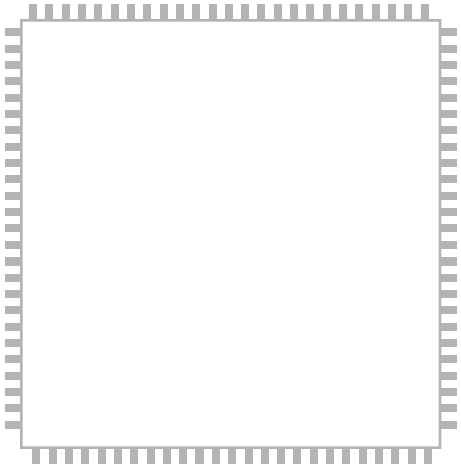
MAX II Device Family

Device	Logic Elements (LEs)	Typical Macro-cells	User I/O Pins	Speed Grades	Fastest t_{pd1} (ns)	User Flash Memory (bits)
EPM240	240	192	80	3, 4, 5	4.7	8,192
EPM570	570	440	160	3, 4, 5	5.5	8,192
EPM1270	1,270	980	212	3, 4, 5	6.3	8,192
EPM2210	2,210	1,700	272	3, 4, 5	7.1	8,192

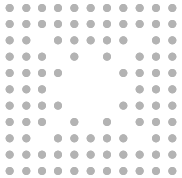
Device	100-Pin TQFP[1] 0.5-mm Pitch 16 x 16 mm	144-Pin TQFP 0.5-mm Pitch 22 x 22 mm	256-Pin FBGA[2] 1.0-mm Pitch 17 x 17 mm	324-Pin FBGA 1.0-mm Pitch 19 x 19 mm
EPM240	80			
EPM570	76	116	160	
EPM1270		116	212	
EPM2210			204	272

New Small Packages

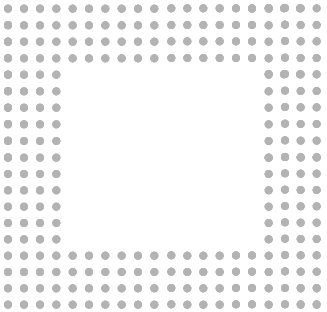
T100
0.5mm TQFP
16x16mm



New Packages

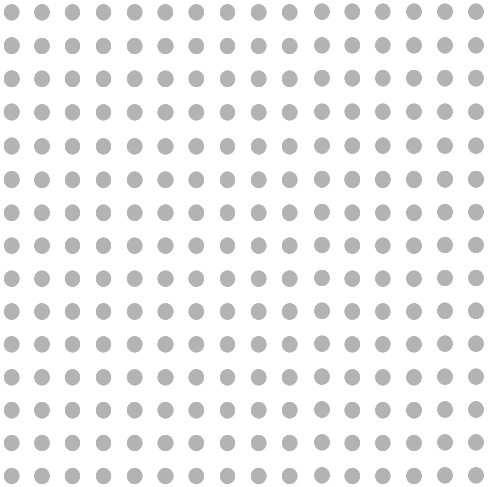


Partial
M100
0.5mm MBGA
6x6mm



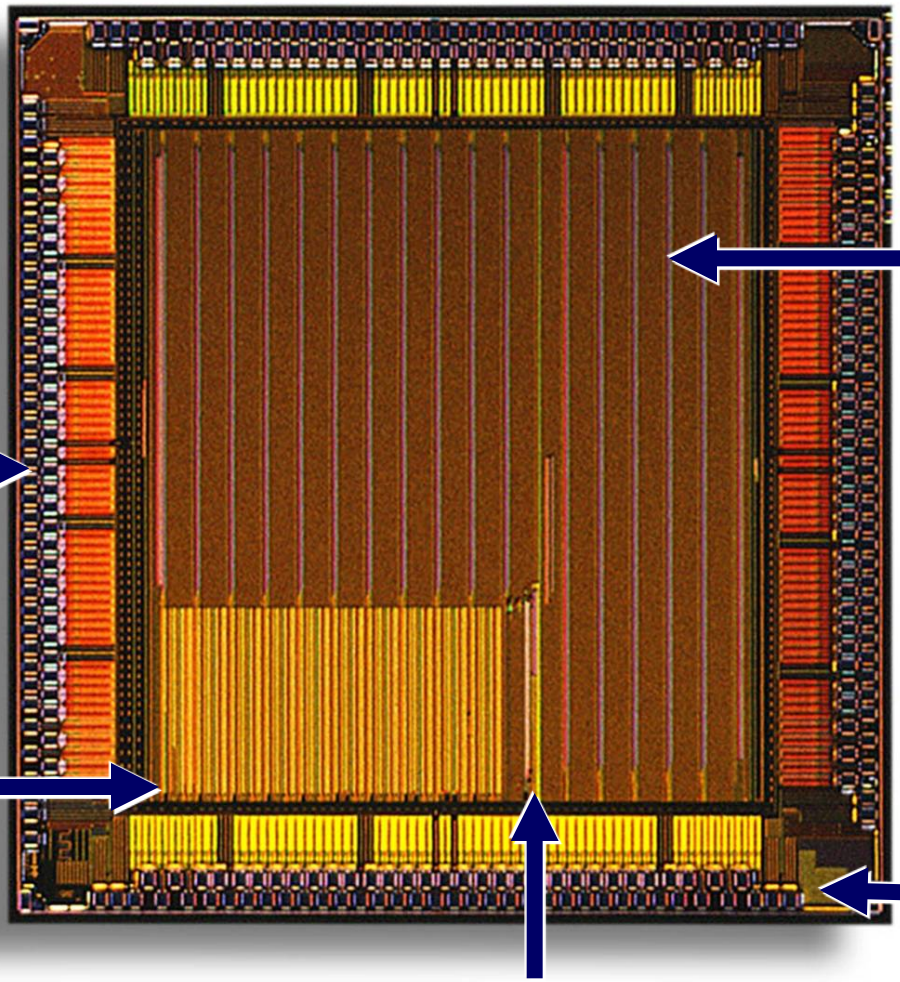
Partial
M256
0.5mm MBGA
11x11mm

F256
1.0mm FBGA
17x17mm



- Packages minimize PCB area and optimize ease-of-use
  - Partial arrays allow for 2 layer PCB break out

MAX II Architecture



Logic Elements (LEs)

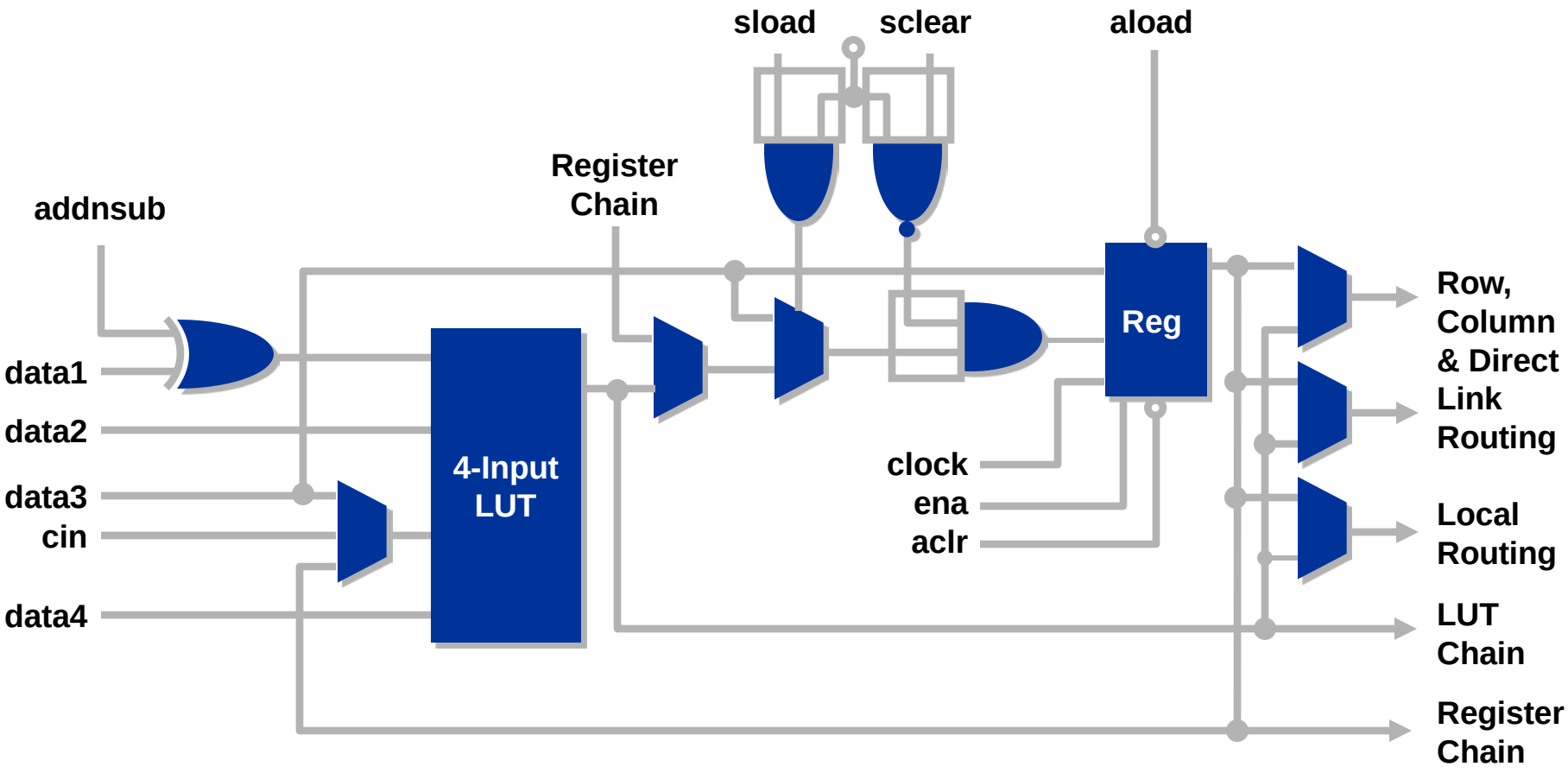
Staggered I/O Pads

Configuration Flash Memory

JTAG & Control Circuitry

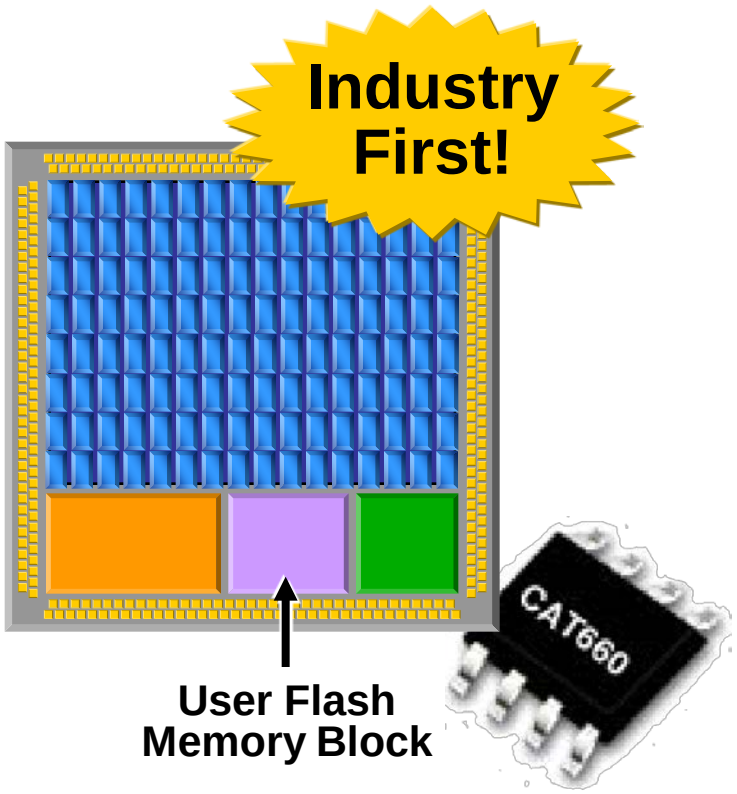
User Flash Memory

MAX II Logic Element (LE)



User Flash Memory

- Feature
  - Flash Memory Storage Bank
  - 8,192 Bits Per Device
  - Interface to SPI, I^2C, Parallel, or Proprietary Buses
- Applications
  - Store Revision & Serial Number Data
  - Store Boot-Up & Configuration Data

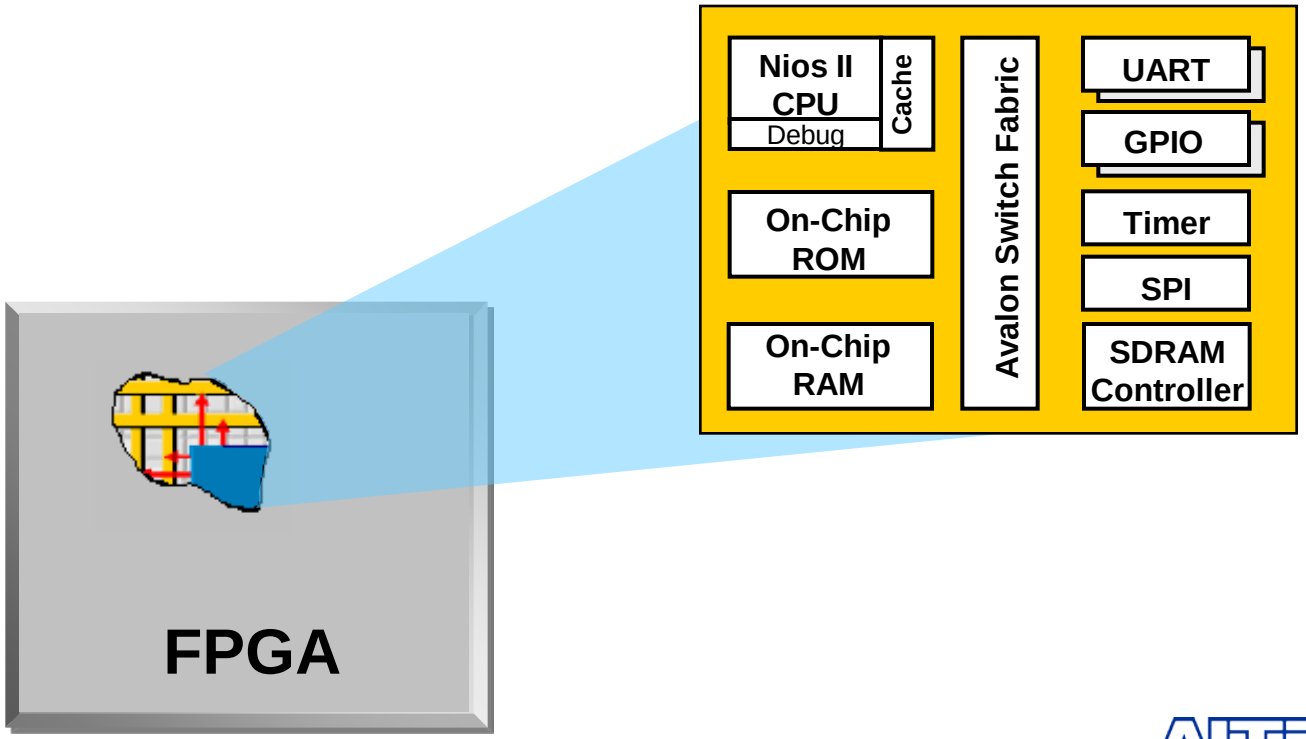


MAX & MAX II Comparison

Parameter	MAX	MAX II
Process Technology	0.3-um EEPROM	0.18-um Flash
Logic Architecture	Product Term	Look-Up Table (LUT)
Density Range	32 to 512 Macrocells	128 to 2210 Macrocells (240 to 2,210 LEs)
Routing Architecture	Global	Row & Column
On-Chip Flash Memory	None	8 Kbits
Maximum User I/O Pins	212	272
Supply Voltage	5.0 V, 3.3 V, 2.5 V	3.3 V/2.5 V, 1.8 V
I/O Voltages	5.0 V, 3.3 V, 2.5 V, 1.8 V	3.3 V, 2.5 V, 1.8 V, 1.5 V
Global Clock Networks	2 per Device	4 per Device
Output Enables (OEs)	6 to 10 per Device	1 per I/O Pin
Schmitt Triggers	None	1 per I/O Pin

What is Nios II?

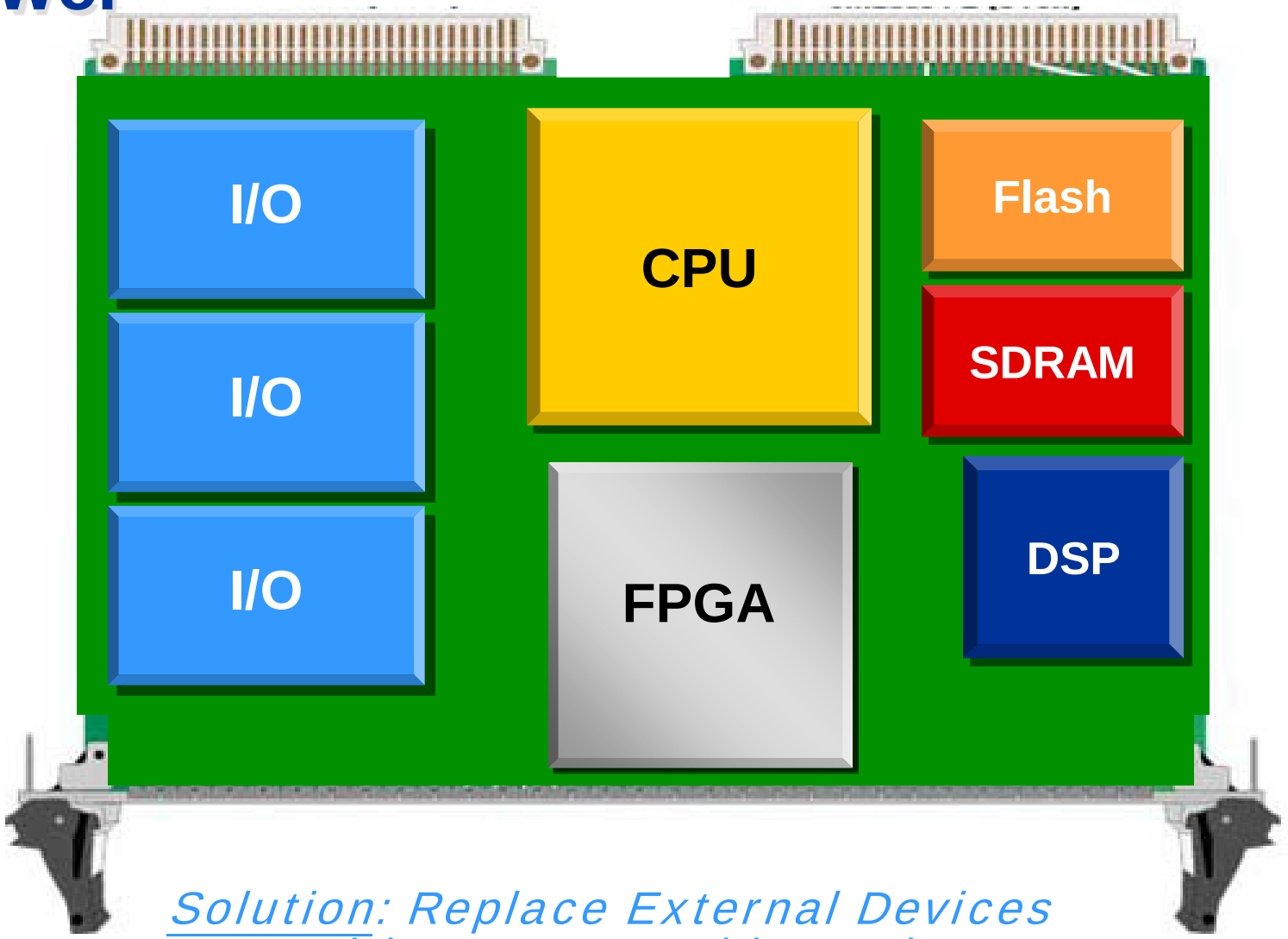
- Altera's Second Generation Soft-Core 32 Bit RISC Microprocessor
  - Nios II Plus All Peripherals Written In HDL
  - Can Be Targeted For All Altera FPGAs
  - Synthesis Using Quartus II Integrated Synthesis



Nios II Processor Architecture

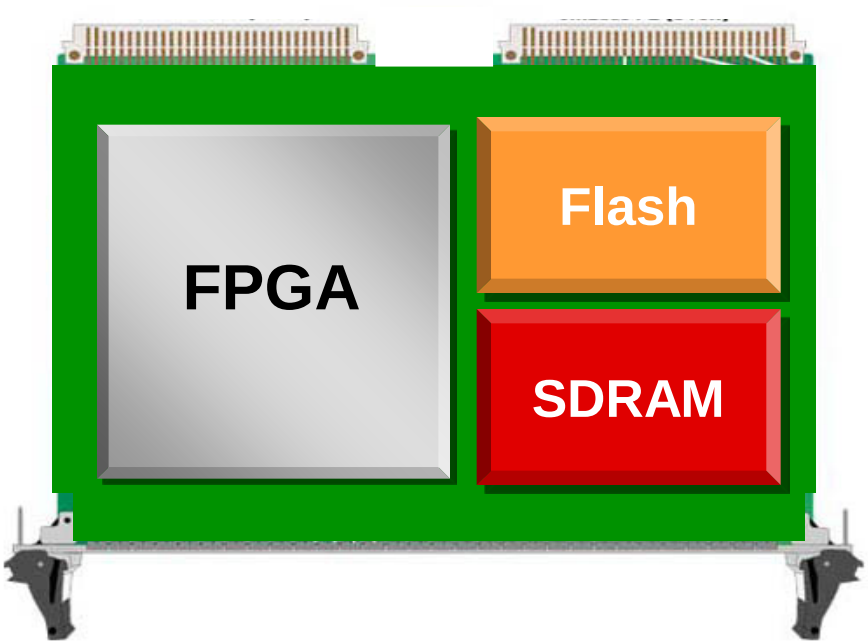
- **Classic Pipelined RISC Machine**
  - 32 General Purpose Registers
  - 3 Instruction Formats
  - 32-Bit Instructions
  - 32-Bit Data Path
  - Flat Register File
  - Separate Instruction and Data Cache (configurable sizes)
  - Tightly-Coupled Memory Options
  - Branch Prediction
  - 32 Prioritized Interrupts
  - On-Chip Hardware (Multiply, Shift, Rotate)
  - Custom Instructions
  - JTAG-Based Hardware Debug Unit

Problem: Reduce Cost, Complexity & Power



Solution: Replace External Devices with Programmable Logic

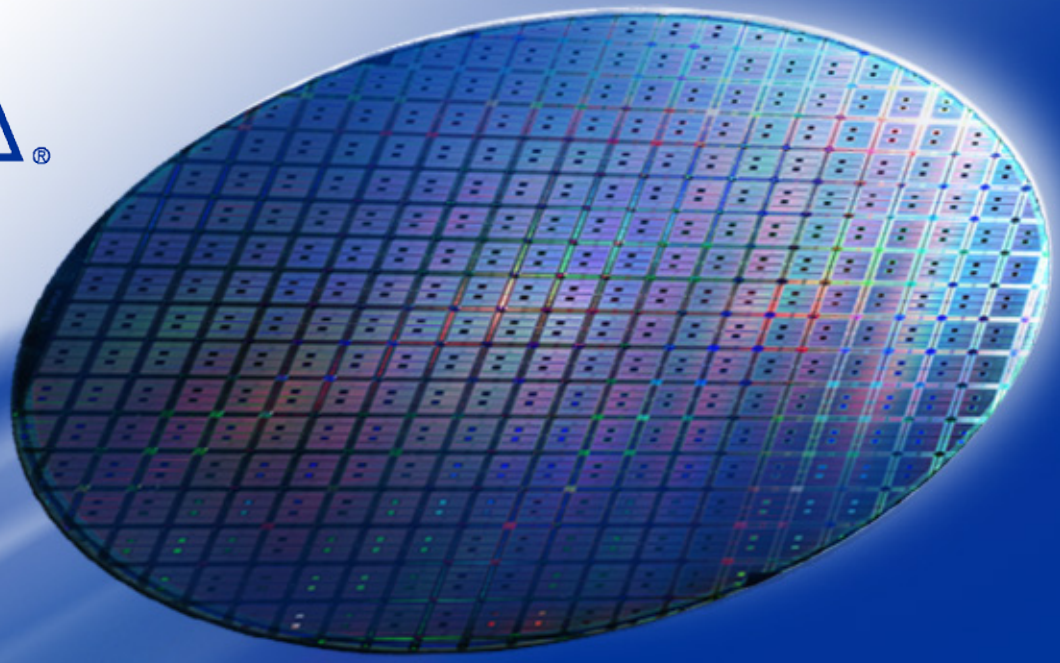
System On A Programmable Chip (SOPC) Power



CPU is a Critical Control Function Required for System-Level Integration

Licensing

- Nios II Delivered As Encrypted Megacore
  - Licensed Via Feature Line In Existing Quartus II License File
  - Consistent With General Altera Megacore Delivery Mechanism
  - Enables Detection Of Nios II In Customer Designs (Talkback)
- No Nios II Feature Line (OpenCore Plus Mode)
  - System Runs If Tethered To Host PC
  - System Times Out If Disconnected from PC After ~ 1 hr
- Nios II Feature Line (Active Subscriber)
  - Subscription and New Dev Kit Customers Obtain Licenses From www.altera.com
  - Nios II CPU RTL Remains Encrypted
- Nios II Source License
  - Available Upon Request On Case-By-Case Basis
  - Included With Purchase Of Nios II ASIC License



Quartus II Basic Training

Quartus II Development System
Feature Overview

Software & Development Tools

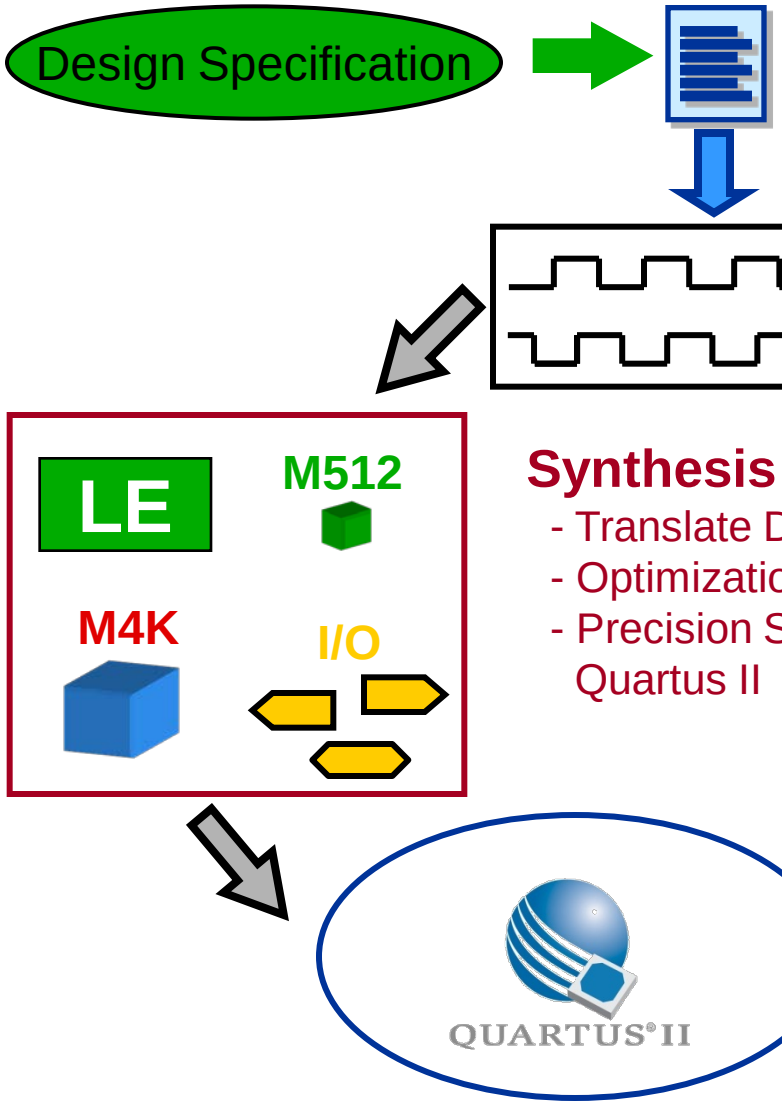
- Quartus II
  - All Stratix, Cyclone & Hardcopy Devices
  - APEX II, APEX 20K/E/C, Excalibur, & Mercury Devices
  - FLEX 10K/A/E, ACEX 1K, FLEX 6000 Devices
  - MAX II, MAX 7000S/AE/B, MAX 3000A Devices
- Quartus II Web Edition
  - Free Version
  - Not All Features & Devices Included
    - See www.altera.com for Feature Comparison

- MAX+PLUS® II
  - All FLEX, ACEX, & MAX Devices

Quartus II Development System

- Fully-Integrated Design Tool
  - Multiple Design Entry Methods
  - Logic Synthesis
  - Place & Route
  - Simulation
  - Timing & Power Analysis
  - Device Programming

Typical PLD Design Flow



Design Entry/RTL Coding

- Behavioral or Structural Description of Design

RTL Simulation

- Functional Simulation (Modelsim®, Quartus II)
- Verify Logic Model & Data Flow (No Timing Delays)

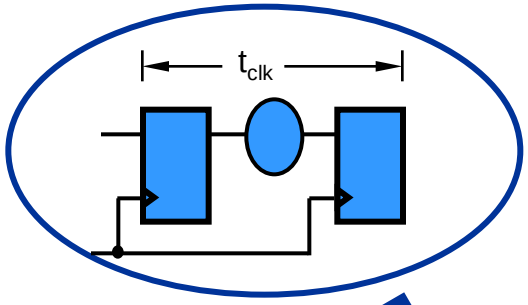
Synthesis

- Translate Design into Device Specific Primitives
- Optimization to Meet Required Area & Performance Constraints
- Precision Synthesis, Synplify/Synplify Pro, Design Compiler FPGA, Quartus II

Place & Route

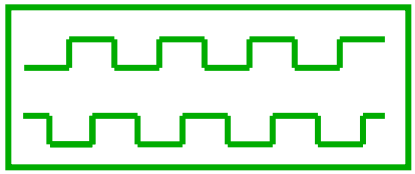
- Map Primitives to Specific Locations inside Target Technology with Reference to Area & Performance Constraints
- Specify Routing Resources to Be Used

Typical PLD Design Flow



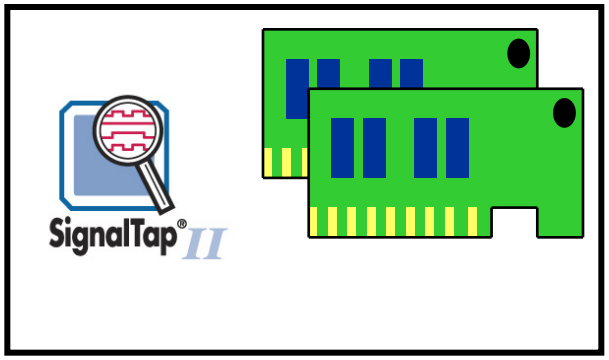
Timing Analysis

- Verify Performance Specifications Were Met
- Static Timing Analysis



Gate Level Simulation

- Timing Simulation
- Verify Design Will Work in Target Technology

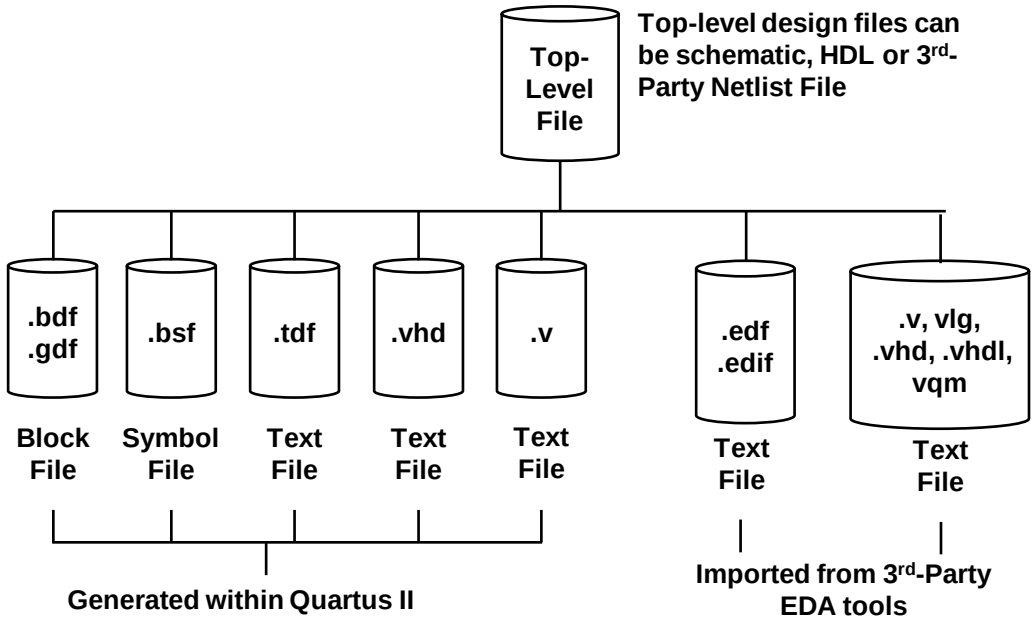


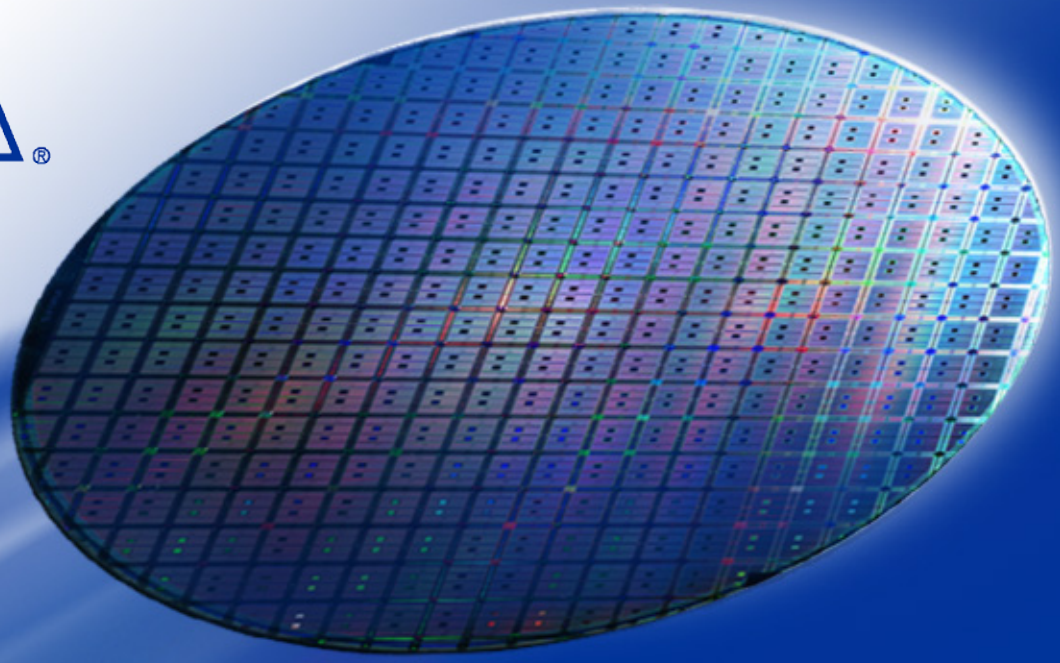
PC Board Simulation & Test

- Simulate Board Design
- Program & Test Device on Board
- Use **SignalTap II** for Debugging

Design Entry Methods

- Quartus II
  - Text Editor
    - AHDL
    - VHDL
    - Verilog
  - Schematic Editor
    - Block Diagram File
    - Graphic Design File
  - Memory Editor
    - HEX
    - MIF
- 3rd-Party EDA Tools
  - EDIF
  - HDL
  - VQM
- Mixing & Matching Design Files Allowed





Quartus II Basic Training

Quartus II Quick Start

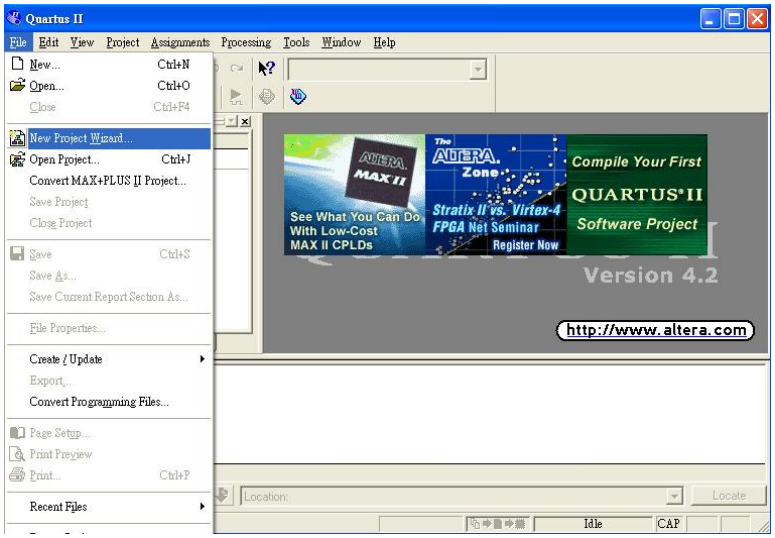
LAB1

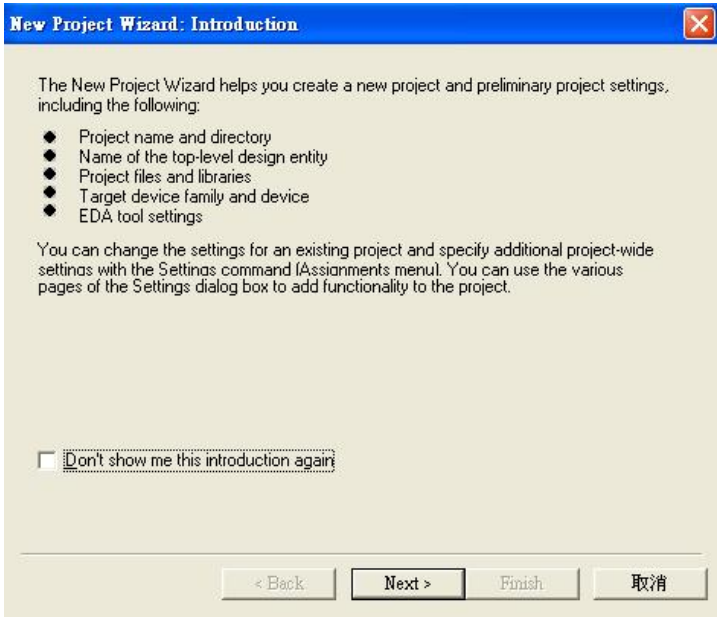
Objectives

- *Create a project using the New Project Wizard*
- *Name the project*
- *Add design files*
- *Pick a device*

Step 1 (Setup Project for QII5_1)

Under File, Select **New Project Wizard**….
A new window appears. If an Introduction screen appears, click Next.

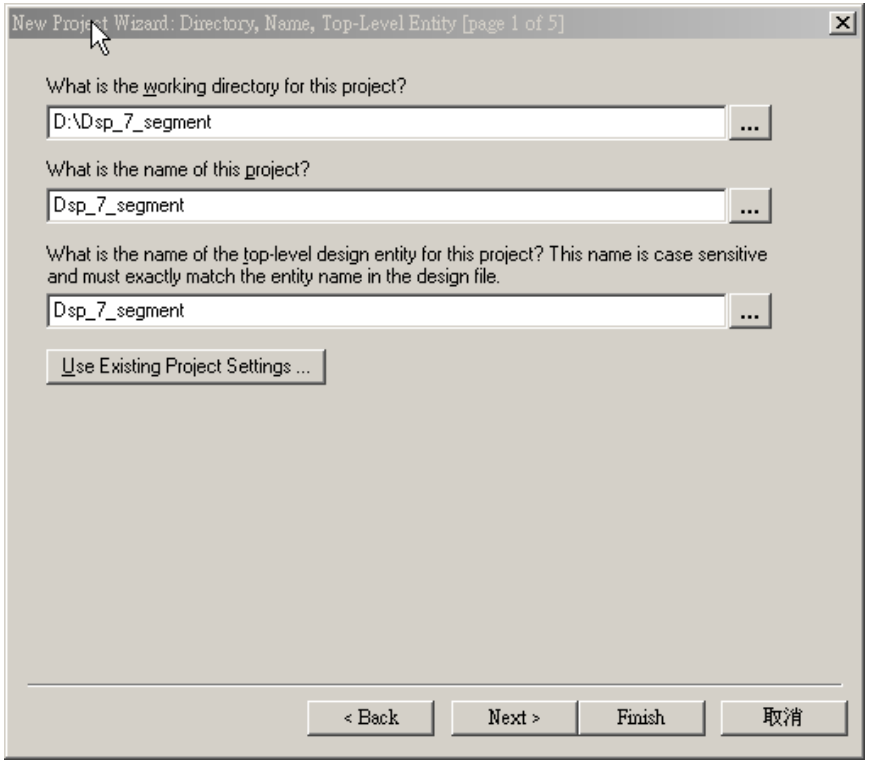




Step 2 (Setup Project for QII5_1)

Page 1 of the wizard should be completed with the following

working directory for this project	**<lab_install_directory> \Dsp_7_segment**
name of project	**Dsp_7_segment**
top-level design entity	**Dsp_7_segment**

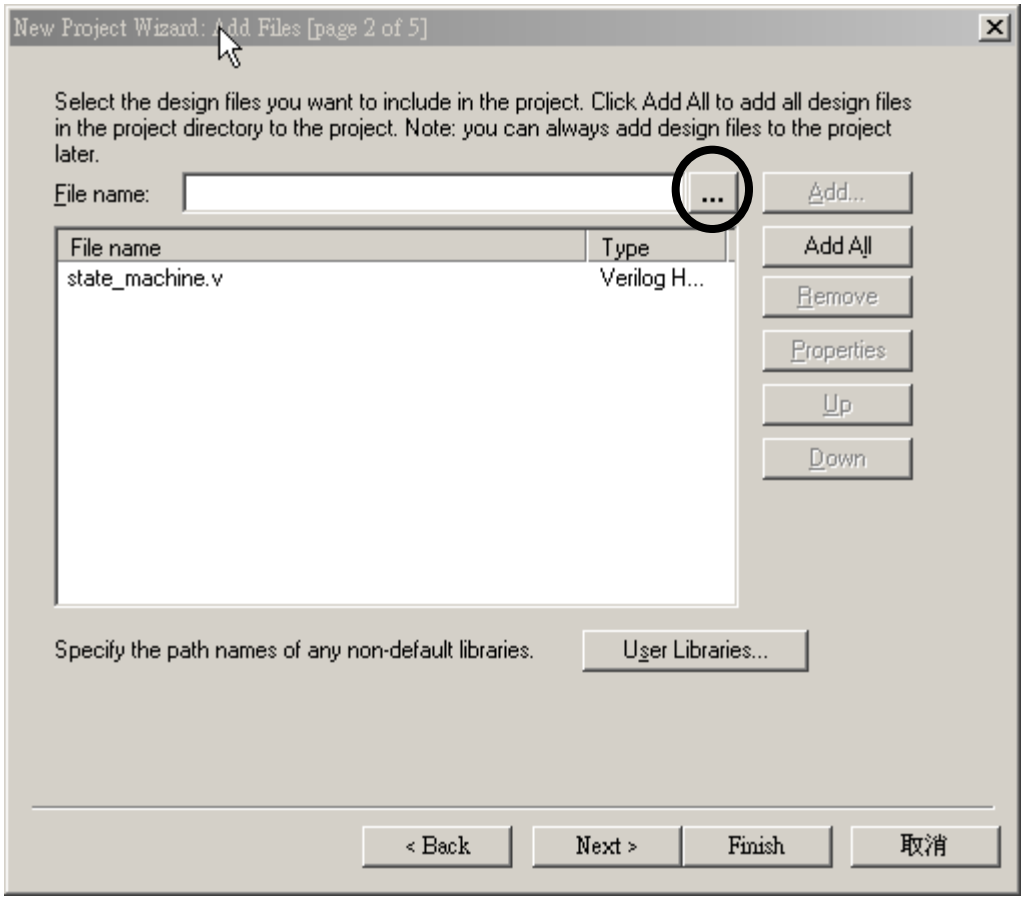


Copy "state_machine.v" and past in Dsp_7_segment

***Click Next** to advance to the **Project Wizard: Add Files [page 2 of 5]**.*

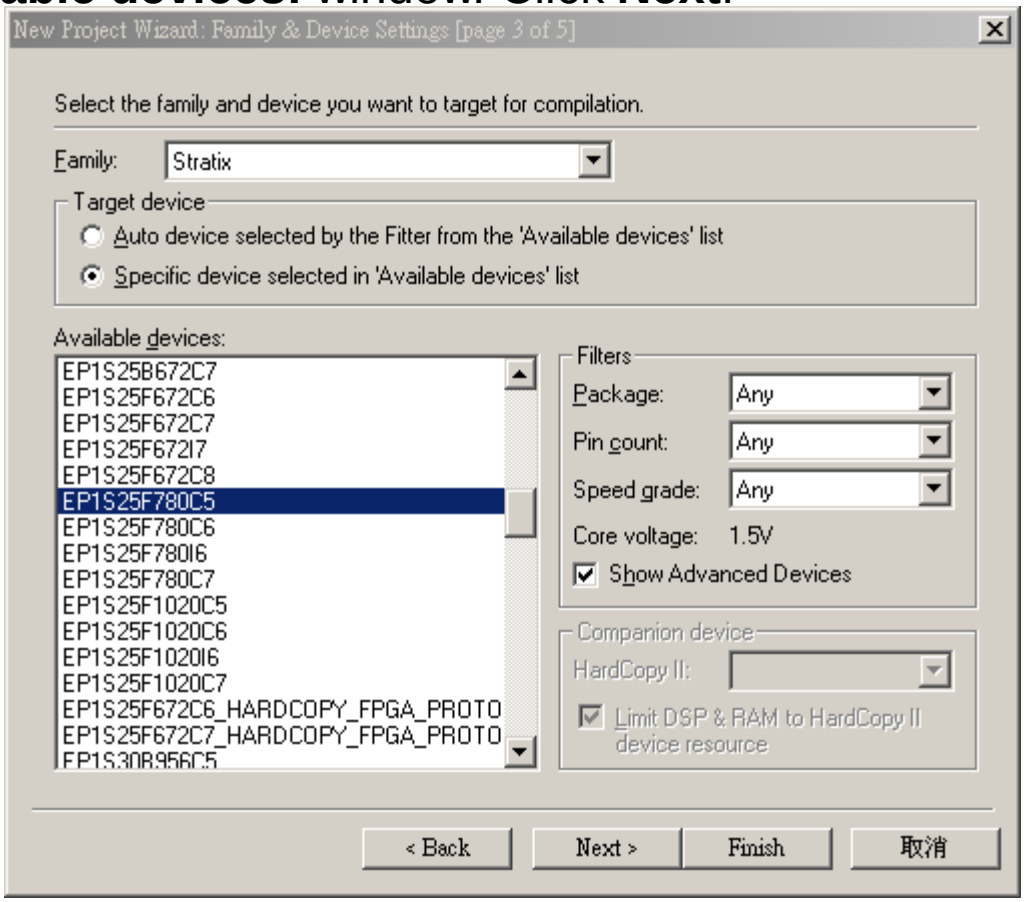
Step 3 (Setup Project for QII5_1)

Using the browse button, select **state_machine.v**
Add to the project. Click **Next**.



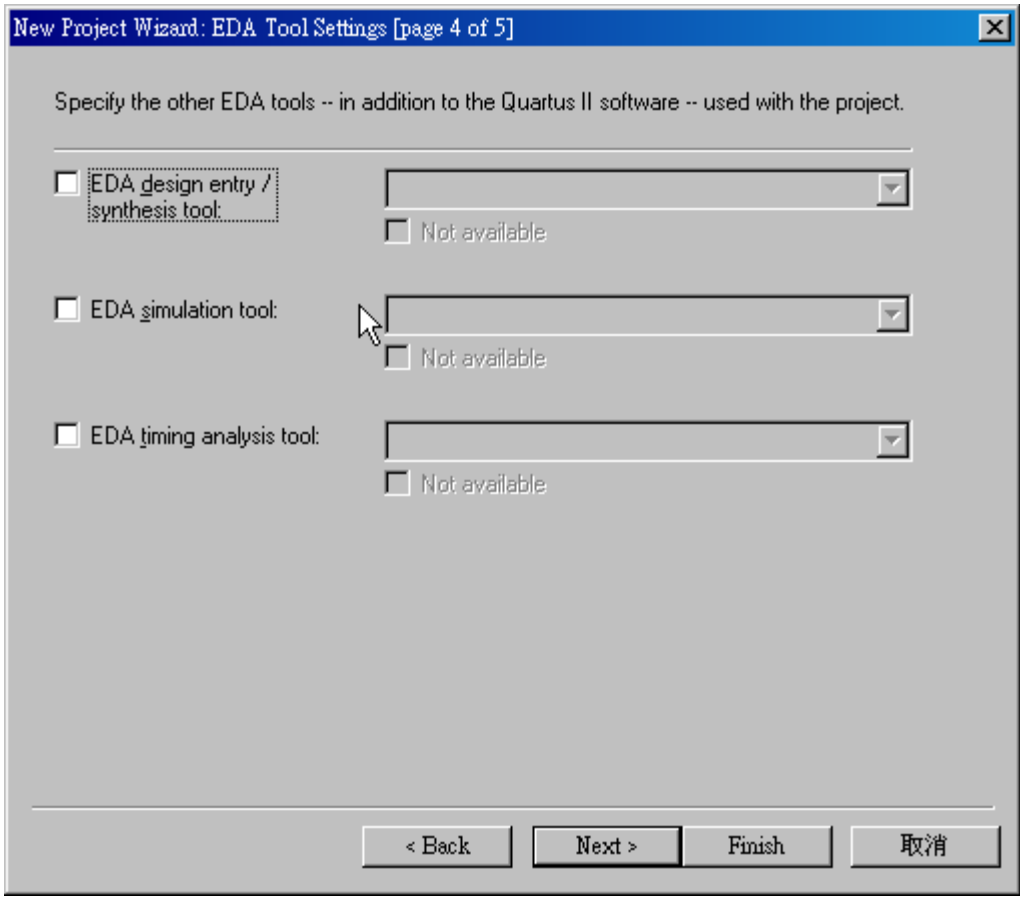
Step 4 (Setup Project for QII5_1)

On **page 3**, select **Stratix** as the **Family**. Also, in the **Filters** section, set **Package** to **FBFA**, **Pin count** to **780**, and **Speed grade** to **5**. Select the **EP1S25F780C5** device from the **Available devices:** window. Click **Next**.



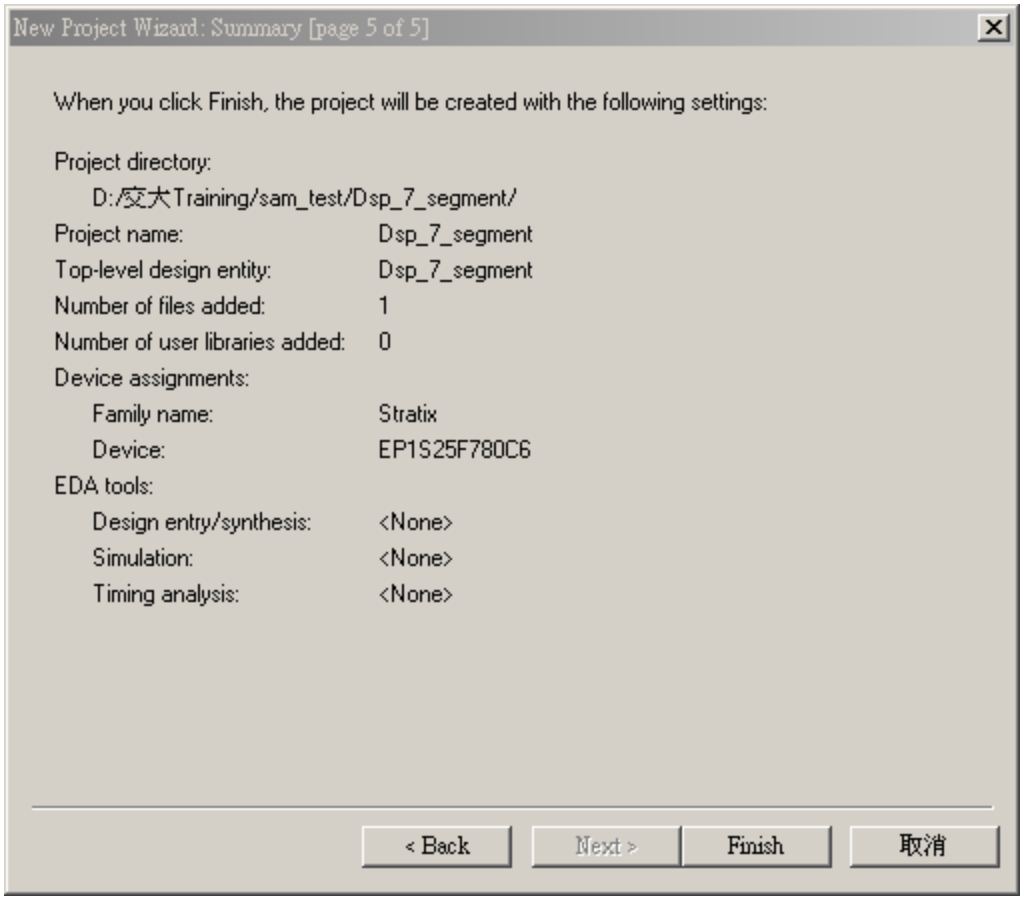
Step 5 (Setup Project for QII5_1)

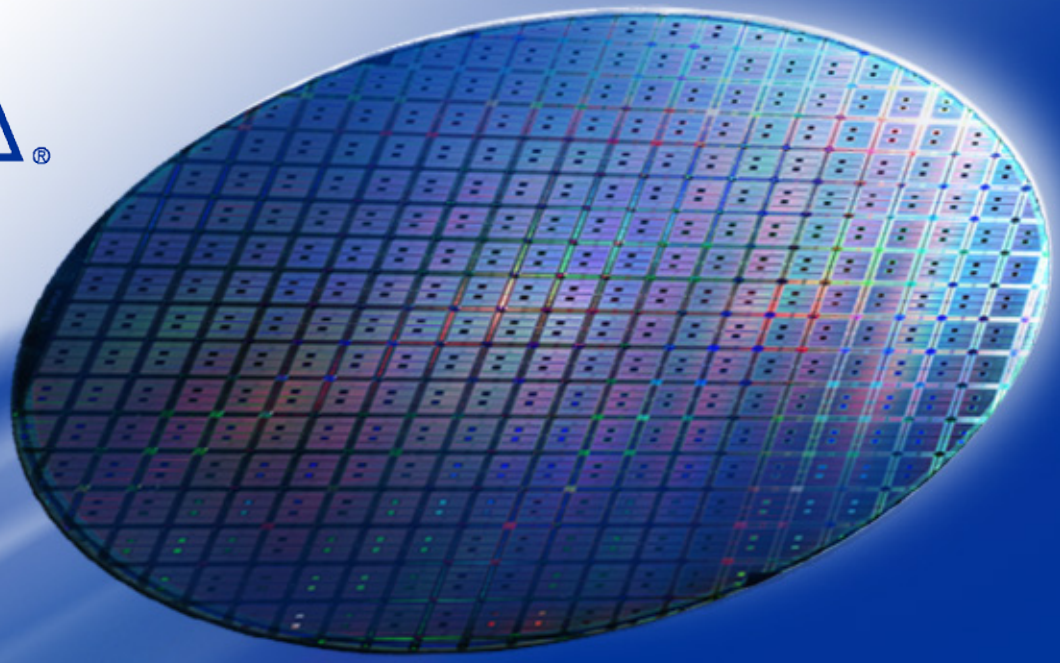
On page 4 , you can specify any third party EDA tools you may be using along with Quartus II. Since these exercises will be done entirely within Quartus II, click **Next.**



Step 6 (Setup Project for QII5_1)

The summary screen appears as shown. Click **Finish.**
The project is now created.





Quartus II Basic Training

Quartus II Quick Start

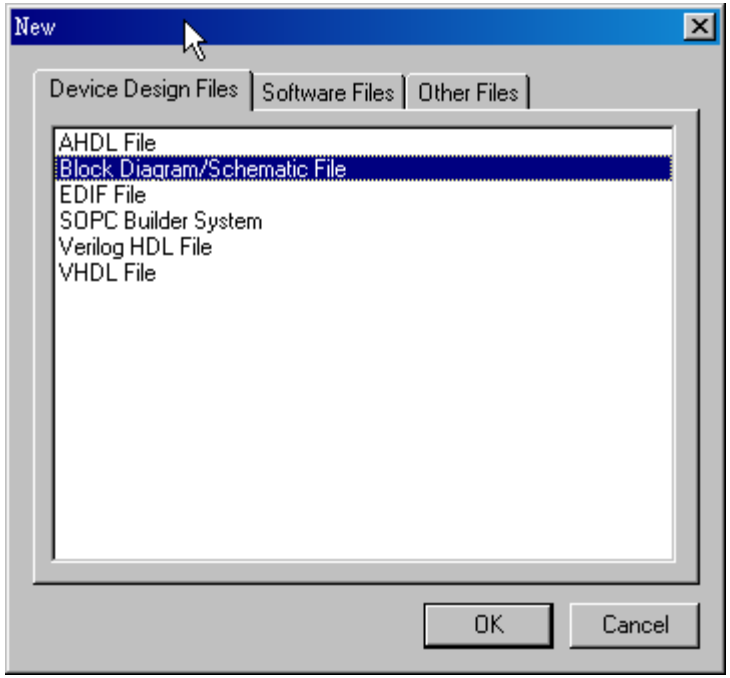
LAB2

Objectives

- *Create a counter using the MegaWizard Plug-in Manager*
- *Build a design using the schematic editor*
- *Analyze and elaborate the design to check for errors*

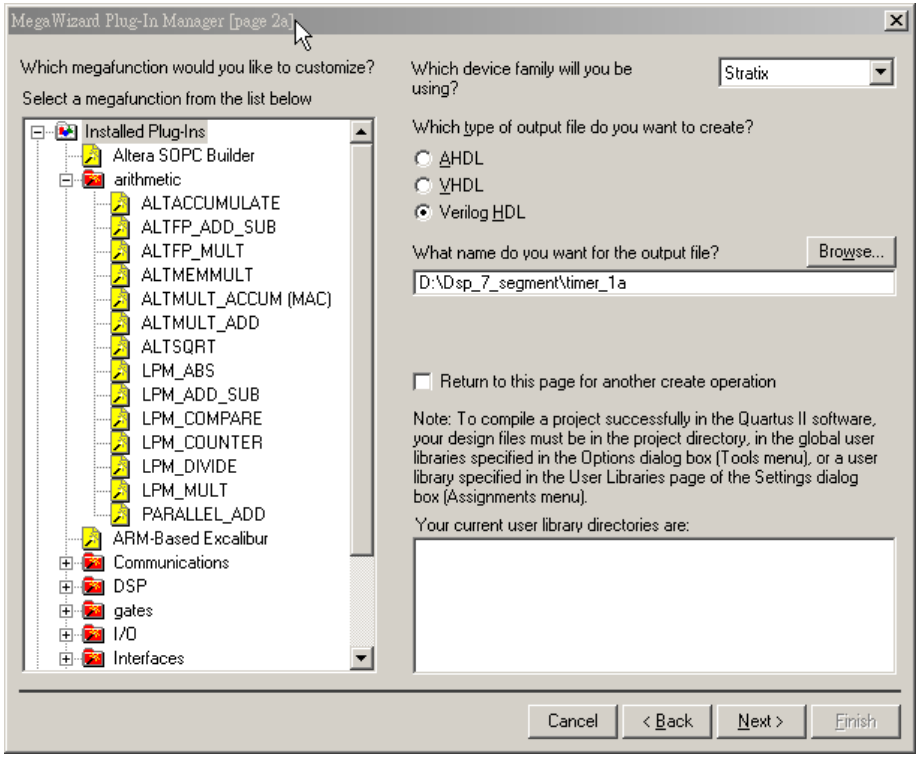
Step 1 Create schematic file

Select **File** ⇒ **New** and select **Block Diagram/Schematic File**. Click **OK**.
Select **File** ⇒ **Save As** and save the file as
<lab_install_directory> \Dsp_7_segment\ Dsp_7_segment.bdf



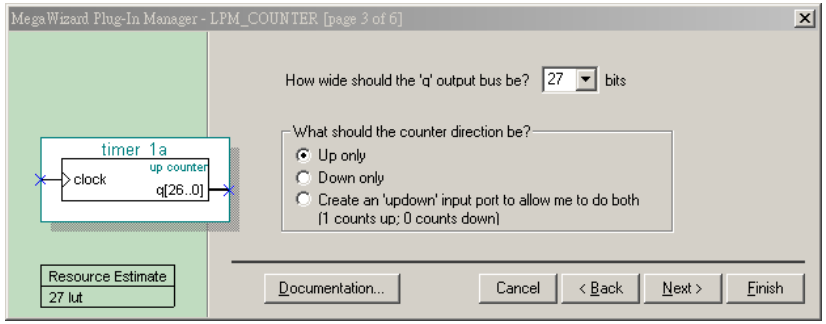
Step 2 Build an 23 bits counter using the MegaWizard Plug-in Manager

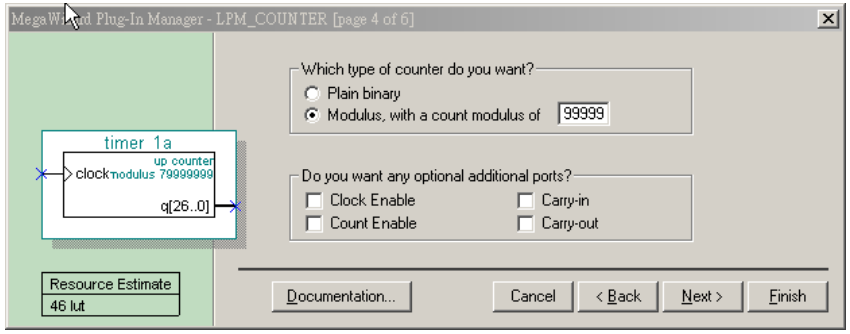
1. Choose **Tools ⇒ MegaWizard Plug-In Manager**. In the window that appears, select **Create a new custom megafunction variation**. Click on **Next**.
2. On **page 2a** of the **MegaWizard** expand the **arithmetic** folder and select **LPM_COUNTER**.
3. Choose **Verilog HDL** output For the name of the **output file,** type **timer_1s**. Click on **Next**

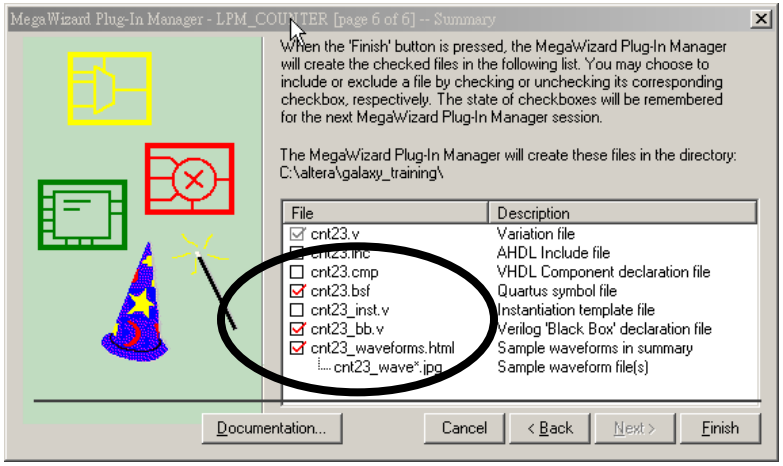


Step 3

1. Set the output bus to **27** bits. For the remaining settings in this window, use the defaults that appear .. Select **next**
2. .Turn on "**Modulus , with a count modulus of** "and key in **79999999**
3. Select **finish**

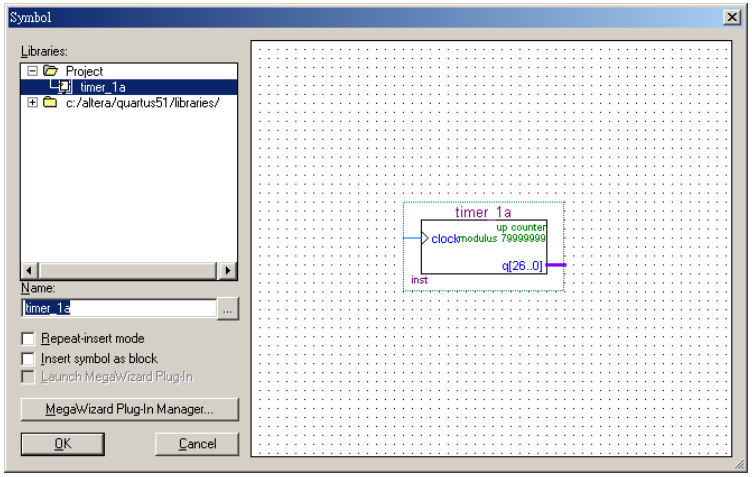


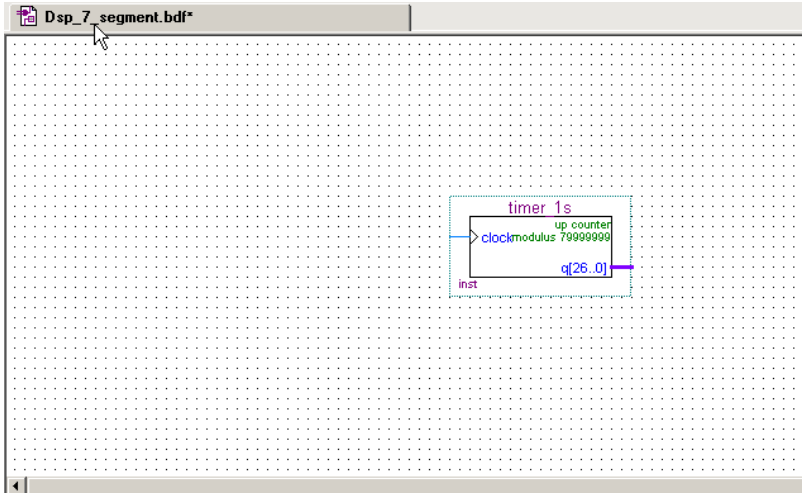




Step 4

In the **Graphic Editor**, **double-click** in the screen so that the **Symbol** Window appears. Inside the symbol window, **click** on + to expand the symbols defined in the **Project** folder. **Double-click** on **timer_1s. Click** the **left mouse button** to put down the symbol inside the schematic file.. *The symbol for* **"timer_1s"** *now appears in the schematic.*





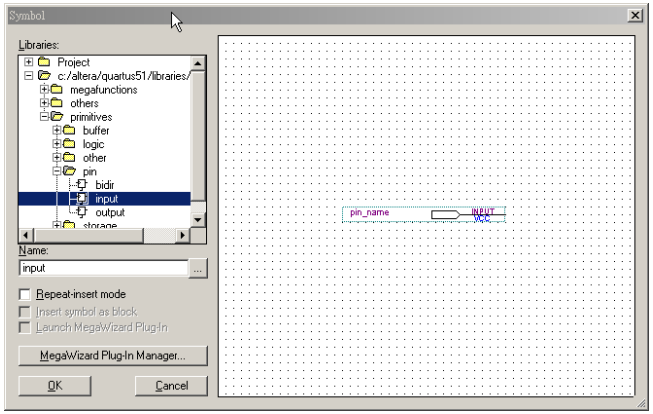
Step 5

1. From the **File** menu, **open** the file **state_machine.v**
2. From the **File** menu, go the **Create/Update** menu option and select **Create Symbol Files for Current File**. Click **Yes** to save changes to **Dsp_7_segment.bdf**.
3. Once **Quartus II** is finished creating the symbol, click **OK**. Close the **state_machine.v** file
4. In the **Graphic Editor**, **double-click** in the screen so that the **Symbol** Window appears again. **Double-click** on **state_machine** in the **Project** folder. **Click OK.**.. *The symbol for* **state_machine** *now appears in the schematic.*

Step 6 Add Pins to the Design

Input	Output
sys_clk	7_out[6..0]
reset	Dig1

For each of the pins listed in left Table , you must insert a pin and change its name

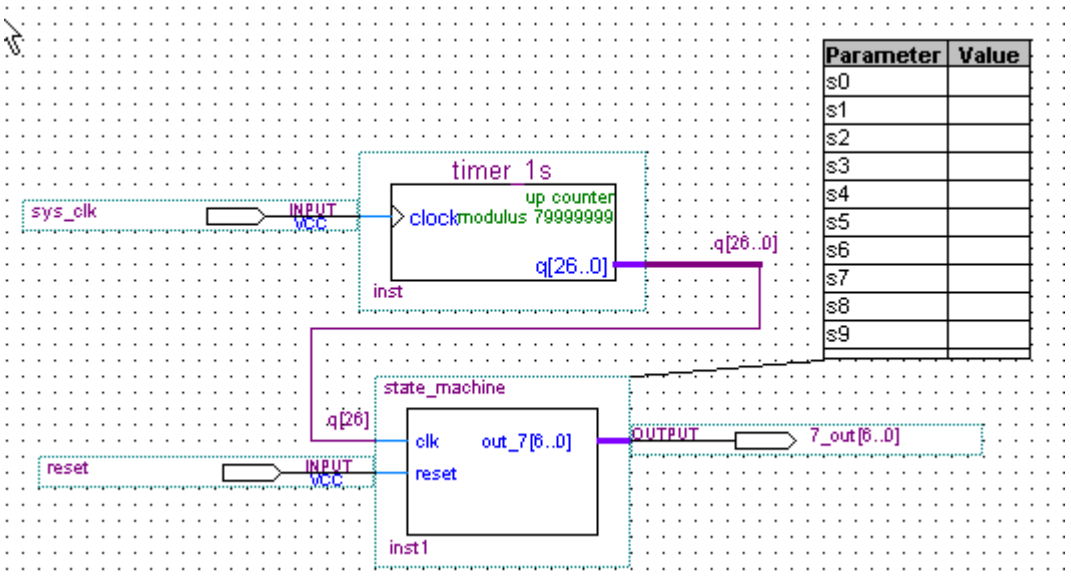


1. To place pins in the schematic file, go to **Edit ⇒ Insert ⇒ Symbol** OR **double-click** in any empty location of the **Graphic Editor**.
2. Browse to **libraries ⇒ primitives ⇒ pin** folder. **Double-click** on **input** or **output** *Hint: To insert multiple pins select* ***Repeat Insert Mode****.*
3. To rename the pins **double-click on the pin name after it has been inserted.**
4. Type the name in the Pin name(s) field and Click OK

.

Step 7 Connect the Pins and Blocks in the Schematic

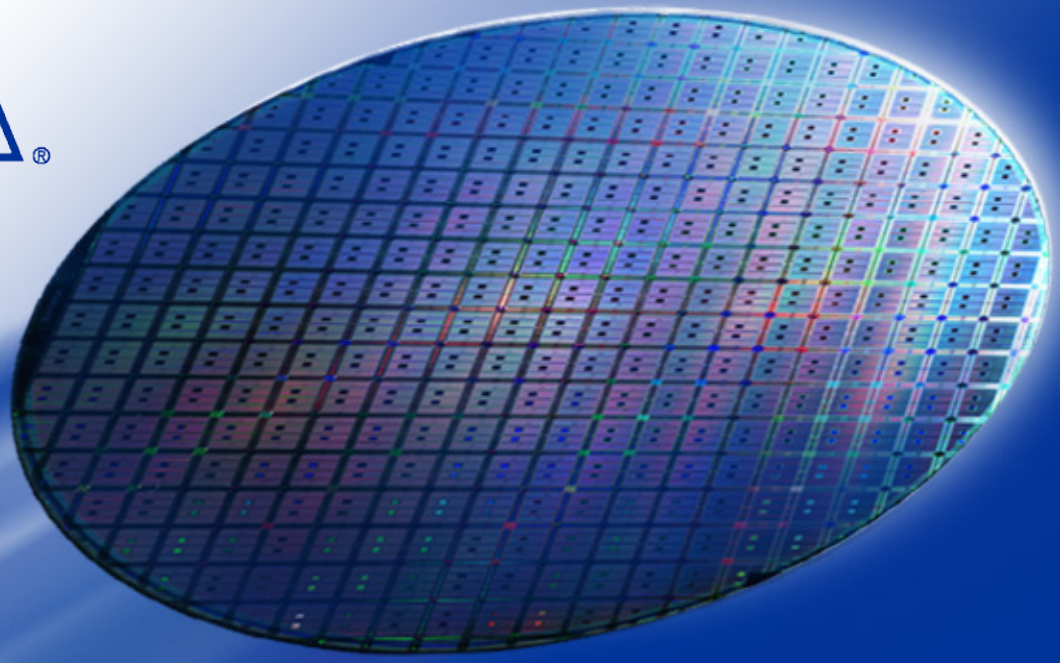
1. In the left hand tool bar click on button to draw a wire and button to draw a bus. Another way to draw wires and busses is to place the cursor next to the port of any symbol. When you do this, the wire or bus tool will automatically appear.
2. Connect all of the pins and blocks as shown in the **figure below**



Step 8 Save and check the schematic

1. Click on the **Save** button in the toolbar  to save the schematic.
2. From the Project menu, select Add/Remove Files in Project.
   Click on the browse button to make sure the **Dsp_7_segment.bdf**, **timer_1s** and **state_machine** are added to the project.
3. From the **Processing** menu, select **Start ⇒ Start Analysis & Elaboration**.

   Analysis and elaboration checks that all the design files are present and connections have been made correctly.
4. *Click **OK** when analysis and elaboration is completed*



Quartus II Basic Training

Quartus II Quick Start

LAB3

Objectives

- *Pin assignment*
- *Perform full compilation Build a design using the schematic editor*
- *How to Download programming file*

Step 1

1. Choose **Assignments** ⇒ Assignment editor.
2. From the **View** menu, select **Show All Know Pin Names.**
3. Please click **Pin** in **Category**

	To	Location	I/O Bank	I/O Standard	General Function	Sp
1	7_out			LVTTL		
2	7_out[0]			LVTTL		
3	7_out[1]			LVTTL		
4	7_out[2]			LVTTL		
5	7_out[3]			LVTTL		
6	7_out[4]			LVTTL		
7	7_out[5]			LVTTL		
8	7_out[6]			LVTTL		
9	reset			LVTTL		
10	sys_clk			LVTTL		
11	<<new>>	<<new>>				

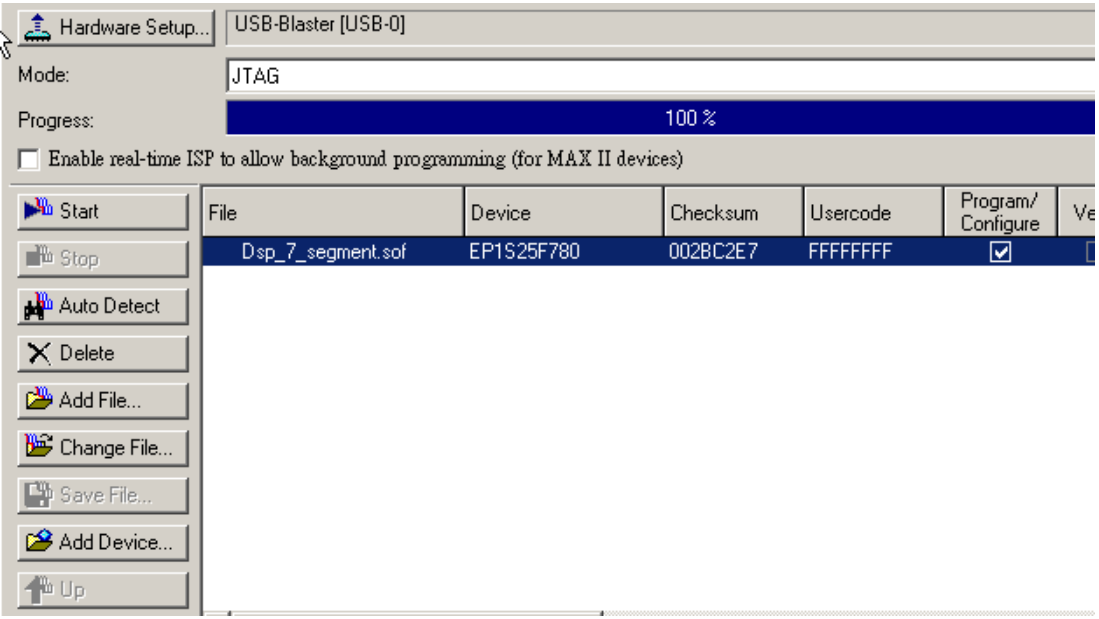
Step 2

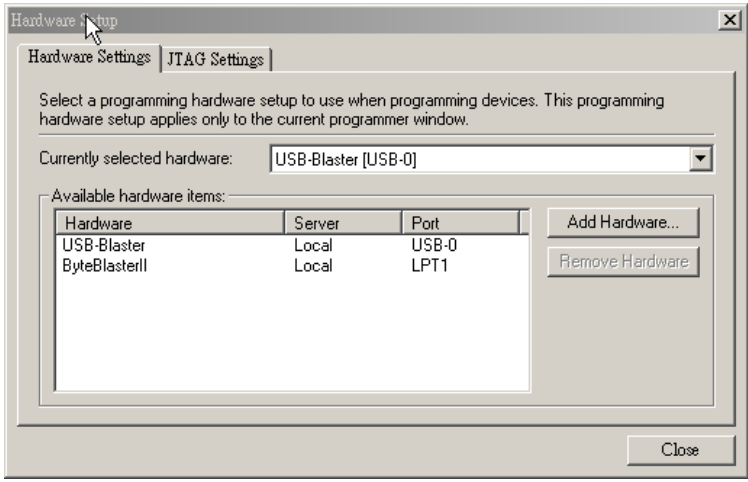
1. Pls install DSP Development Kit Stratix edtion CD
2. Open ds_stratix_dsp_bd.pdf from C:\megacore\stratix_dsp_kit-v1.1.0\Doc
3. Check clk , pushbotton and seven segment display pin location from ds_stratix_dsp_bd.pdf
4. Key your pin number in location
5. Click on the **Save** button in the toolbar 
6. From **Assignments**, select **Device**. Click **Device & Pin options**. Click **Unused pins** .Select **As input tri-stated** from **Reserve all unused pins**
7. From the **Processing** menu, select **Start Compilation**

	To	Location
1	7_out[0]	PIN_L18
2	7_out[1]	PIN_D24
3	7_out[2]	PIN_L23
4	7_out[3]	PIN_L24
5	7_out[4]	PIN_L22
6	7_out[5]	PIN_L20
7	7_out[6]	PIN_L19
8	reset	PIN_F24
9	sys_clk	PIN_K17
10	7_out	

Step 3

1. From the **Tools** menu, select **programmer**
2. Click on **Add File**. Select Dsp_7_segment.sof.
3. Check **Hardware Setup.** Select your download cable **on Currently selected hardware(ByteBlasterII)**
4. Select **JTAG** from **Mode**





Step 4

1. Turn on **Program/configure. Or see figure below**
2. **Click Start**
3. **See 7-segment status**