Designing ADC-DAC System from Scratch for DE2

Tutorial



System Level Solutions, Inc. (USA) 14100 Murphy Avenue San Martin, CA 95046 (408) 852 - 0067

http://www.slscorp.com

Version : 0.1.1 Date : July 10, 2007

Copyright©2007, System Level Solutions.All rights reserved. SLS, An Embedded systems company, the stylized SLS logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of SLS in India and other countries. All other products or service names are the property of their respective holders. SLS products are protected under numerous U.S. and foreign patents and pending applications, mask working rights, and copyrights. SLS warrants performance of its semiconductor products to current specifications in accordance with SLS is standard warranty, but reserves the right to make changes to any products and services at any time without notice. SLS assumes no responsibility or liability arising out of the application or use of any information, products, or service described herein except as expressly agreed to in writing by SLS. SLS customers are advised to obtain the latest version of specifications before relying on any published information and before orders for products or services.

MPD_SO03_TT1_011

ii





Introduction

This tutorial provides all the steps for creating a system for the ADC-DAC board from the scratch with the DE2 board. Also it shows how to create, compile, debug and run a C/C++ program using the Nios II IDE.

Table below shows the Revision history of Designing ADC-DAC System from Scratch for DE2.

Version	Date	Description
0.1.1	July 2007	First Publication of Designing ADC-DAC System from Scratch for DE2 Tutorial.

How to Contact SLS

For the most up-to-date information about SLS products, go to the SLS worldwide website at http://www.slscorp.com. For additional information about SLS products, consult the source shown below.

Information Type	E-mail
Product literature services, SLS liter- ature services, Non-technical cus- tomer services, Technical support.	support@slscorp.com

Typographic Conventions

Designing ADC-DAC System from Scratch for DE2, Cyclone II Edition uses the typographic conventions shown as below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	All Headings and Sub Headings Titles in a document are dis- played in bold type with initial capital letters; Example: Introduction, Creating a Quartus II Project.
Bold Type with Italic Letters	All Definitions, Figure and Table Headings are displayed in Italics. Examples: Figure 2-1. Create a New System Dialog Box, Figure 2-2. SOPC Builder
1. 2.	Numbered steps are used in a list of items, when the sequence of items is important. such as steps listed in procedure.
•	Bullets are used in a list of items when the sequence of items is not important.
Courier type	Anything that must be typed exactly as it appears is shown in Courier type. For Example cpu
	The hand points to information that requires special attention.
CAUTION	The caution indicates required information that needs special con- sideration and understanding and should be read prior to starting or continuing with the procedure or process.
WARNING	The warning indicates information that should be read prior to starting or continuing the procedure or processes.
••••	The feet direct you to more information on a particular topic.

Contents



Ab	out this Tutorialiii
	Introductioniii
	How to Contact SLS
	Typographic Conventionsiv
Fi	guresvii
1.	Introduction
	Hardware & Software Requirements
2.	Designing & Compiling
	Start SOPC builder
	Specify Target Hardware Settings
	Adding CPU & Peripherals
	Nios II 32-bit CPU
	JTAG UART
	Timer
	Tristate Bridge
	SRAM Memory
	Parallel I/O for ADC Output Start Bit
	Parallel I/O for ADC Output Enable Bit
	Parallel I/O for ADC Output Address Latch Enable Bit
	Parallel I/O for ADC Output Address Bits
	Parallel I/O for ADC Input End Of Character Bit
	Parallel I/O for ADC Input Data Byte
	Parallel I/O for DAC Output Data Byte
	Parallel I/O for LEDs
	Parallel I/O for Pushbutton Switches
	Generating the System

	Adding the Quartus II Symbol to the BDF Compiling the Design	. 22
3.	Programming	. 42
4.	Running Nios II IDE Creating the Software Building the Project Running the Project	44 . 44 . 50 . 51



Figures

Figure 2-1.	Create New System Dialog Box	3
Figure 2-2.	SOPC Builder	4
Figure 2-3.	Nios II Configuration Wizard - Nios II Core Tab	5
Figure 2-4.	Nios II Configuration Wizard -JTAG Debug Module	6
Figure 2-5.	JTAG UART Configuration Wizard	7
Figure 2-6.	Timer Configuration Wizard	8
Figure 2-7.	Avalon Tristate Bridge Configuration Wizard	9
Figure 2-8.	SRAM Memory Configuration Wizard	10
Figure 2-9.	PIO Configuration Wizard For ADC Output Start Bit	11
Figure 2-10.	PIO Configuration Wizard For ADC Output Enable Bit	12
Figure 2-11.	PIO Configuration Wizard for ADC Output Address Latch Enable Bit	13
Figure 2-12.	PIO Configuration Wizard For ADC Output Address Bits	14
Figure 2-13.	PIO Configuration Wizard For ADC Input End Of Character Bit	15
Figure 2-14.	PIO Configuration Wizard For ADC Input Data Byte	16
Figure 2-15.	PIO Configuration Wizard For DAC Output Data Byte	17
Figure 2-16.	PIO Configuration Wizard For LEDs	18
Figure 2-17.	PIO Configuration Wizard For Pushbutton Switches-Basic Settings	19
Figure 2-18.	PIO Configuration Wizard For Pushbutton Switches-Input Options Settings	20
Figure 2-19.	SOPC Builder After Adding All Components	21
Figure 2-20.	Generating the System	22
Figure 2-21.	New Block Design File	23
Figure 2-22.	BDF System_Top Symbol	24
Figure 2-23.	Generate Pins	25
Figure 2-24.	Autogenerated Pins for Symbol	25
Figure 2-25.	Pin Properties	26
Figure 2-26.	Symbol Window	27
Figure 2-27.	MegaWizard Plug-In Manager [page 2C]	28
Figure 2-28.	MegaWizard Plug-In Manager [page 3]	29
Figure 2-29.	MegaWizard Plug-In Manager [page 4]	30
Figure 2-30.	MegaWizard Plug-In Manager [page 5]	31
Figure 2-31.	MegaWizard Plug-In Manager [page 6]	32
Figure 2-32.	MegaWizard Plug-In Manager [page 7]	33

Figure 2-33.	NOT Gate Selection	34
Figure 2-34.	NOT Gate Connection	. 35
Figure 2-35.	Counter Output	. 36
Figure 2-36.	Counter Output Connection	37
Figure 2-37.	Settings	. 38
Figure 2-38.	Device & Pin Options	. 39
Figure 2-39.	Final BDF	40
Figure 2-40.	Compilation Report Window	.41
Figure 3-1.	Programming Window	43
Figure 4-1.	New Project	44
Figure 4-2.	Creating new Nios II C/C++ Application	.45
Figure 4-3.	Listing of Your Project under Nios II C/C++ Project Tab	46
Figure 4-4.	Importing File System	46
Figure 4-5.	Import Dialog Box	47
Figure 4-6.	Asking for Details of the files to be imported	48
Figure 4-7.	Asking for Details of the files to be imported	49
Figure 4-8.	Inserted C file template in the application list	. 50
Figure 4-9.	Build Project	. 51
Figure 4-10.	Build Process	. 51
Figure 4-11.	Run Option	. 52
Figure 4-12.	Run Dialog Box	. 53
Figure 4-13.	Run Dialog- Target Connection Settings	.54
Figure 4-14.	Run As-Nios II Hardware Selection	. 55
Figure 4-15.	IDE Console Output	. 55
Figure 4-16.	Analog to Digital Selection Console Window	. 56
Figure 4-17.	Analog to Digital Output Console Window	. 57
Figure 4-18.	Digital to Analog Selection Console Window	. 57
Figure 4-19.	Digital to Analog Output Console Window	. 58
Figure 4-20.	Auto Test Selection Console Window	. 58
Figure 4-21.	Auto Test Output Console Window	. 59

1. Introduction



This tutorial walks you through the hardware & software development flow. It shows you how to use SOPC Builder and the Quartus II software to create and use your own Nios II system

This tutorial is basically for users who are new to the Nios II processor as well as users who are new to the concept of using embedded systems in FPGA. This tutorial guides you through the steps necessary to create and compile a ADC-DAC System Design, called **sc_adcdac_refdes_de2**. This simple, single-master ADC-DAC control system consists of a Nios II embedded processor and associated system peripherals as well as interconnections for use with the input & output hardware available on the DE2 board.

This tutorial is divided into the following three sections:

- 'Designing & Compiling' Teaches you how to use SOPC builder to create the ADC-DAC module in block design file (.bdf) and how to compile the ADC-DAC design using the Quartus II Compiler.
- *Programming'* Teaches you how to use the Quartus II Programmer and the USB-Blaster cable to configure the FPGA on DE2 board.
- 'Running the Software on Your Nios II System' Provides the instructions for running software on your Nios II system using the Nios II Integrated Development Environment (IDE).

The user will require following hardware & software

- A PC running with Win 2000/XP OS
- Nios II Embedded Processor
- The Quartus II Software, version 6.1
- DE2 Development Kit

Hardware & Software Requirements

1

2. Designing & Compiling



To use the instructions in this section, you need to be familiar with the Quartus II software interface-specifically tool bars. Refer to Quartus II help for more information about using the Quartus II software.

Creating a Quartus II Project

Here are the steps to create a new Quartus II project:

- 1. Open the Quartus II.
- 2. Choose File>New Project Wizard.
- 3. Click Next.
- Select Working Directory of the Project, Name of the project as 'sc_adcdac_refdes_de2' & top-level entity as 'sc_adcdac_refdes_de2'.
- 5. Click Next.
- 6. Click Next.
- 7. Select the family as 'Cyclone II'.
- We will select the FPGA (which is Cyclone II EP2C35F672C6) for DE2 board, so under Filters / Speed Grade select 6. Then under Available devices: Select 'EP2C35F672C6'. Click Next.
- 9. Click Next.
- 10. Click Finish.

Start SOPC builder

SOPC builder is a software tool that allows you to create a fully functioning, ADC-DAC system module. A complete ADC-DAC system module contains a Nios II embedded processor and its associated peripherals.

To start SOPC builder, perform the following steps:

- 1. Open the Quartus II software.
- **2.** Choose **SOPC Builder** (**Tools menu**). SOPC Builder displays the Create New System dialog box.

- **3.** Type 'System_Top'. See Figure 2-1.
- 4. Specify Verilog or VHDL in HDL Language field.

SOPC Builder generates plain text Verilog HDL or VHDL for all of its native components depending on the language you choose.

Figure 2-1. Create New System Dialog Box

哇 Create New System 🛛 🛛 🔀		
System Name: System_Top		
Target HDL		
💽 Verilog 🛛 🔿 VHDL		
Cancel		

5. Click **OK**. The Altera SOPC Builder - System_Top window appears and the System Contents tab is displayed.

You are now ready to set the speed and add the Nios II CPU and peripherals to your system. The components you will be adding are located in the module pool on the left hand side of the System Content tab. See Figure 2-2.

Specify Target Hardware Settings

The functionality of the SOPC Builder system depends on the hardware on which it will run. Thus, specifying the target board is the first step in creating a system.

- Choose a **board type** in the Target pull-down menu. As DE2 is not yet added in the menu, keep the **'Unspecified Board'**.
- Select the System Clock Frequency as 27Mhz.

Figure 2-2. SOPC Builder

Liena SOPC Builder - System File Module System View Tools System Contents System Generation	1_Тор Нер				. 7 🗙
Attera SOPC Builder Create New Component Avalan Components Nos Il Processor - A Bridges Communication Prose	Target Board: Unspecfied Board Device Family: Cydone II V HardCopy Co	Click Source ck External click to add	MHz F 27.0	Pipeline	
Bisplay Display D	Use Module Name	Description		put Ci Base E	ind
Done checking for updates.	Exit < Prev	Next > Generate			

Adding CPU & Peripherals

4

This section describes adding following modules to the SOPC Builder.

- Nios II 32-bit CPU
- JTAG UART
- Timer
- Tristate Bridge
- SRAM Memory
- Parallel I/Os for
 - ADC Output Start Bit
 - ADC Output Enable Bit
 - ADC Output Address Latch Enable Bit
 - ADC Output Address Bits
 - ADC Input End of Character Bit

- ADC Input Data Byte
- DAC Output Data Byte
- LEDs
- Pushbutton Switches

Nios II 32-bit CPU

To add the Nios II 32-bit CPU, named CPU, perform the following steps:

- 1. Under Avalon Modules, select Nios II Processor Altera Corporation.
- 2. Click Add. The Nios II configuration wizard titled Altera Nios II cpu_0 displays.
- 3. Specify the following options in the Nios II Core tab.
 - Select the **processor core:** Nios II/s as in Figure 2-3.
 - Instruction Cache Size: 4Kbytes

Figure 2-3. Nios II Configuration Wizard - Nios II Core Tab

📙 Altera Nios II - cp	ou_0			E C	
Nios II Core Caches	& Tightly Coupled Memories	dvanced Features JTAG Debug	Module Custom Instructions		
Select a Nios II core:					
	○Nios II/e	⊙Nios II/s	○Nios II/f		
Nios II Selector Guide Family: Cyclone II f _{system:} 48 MHz	RISC 32-bit	२।ऽ८ ३२-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide Barrel Shifter Data Cache Dynamic Branch Prediction		
Performance at 48 MHz	Up to 4 DMIPS	Up to 24 DMIPS	Up to 49 DMIPS		
Logic Usage	600-700 LEs	1200-1400 LEs	1400-1800 LEs		
Memory Usage	Two M4Ks	Two M4Ks + cache	Three M4Ks + cache		
Hardware Multiply: Embedded Multipliers 😪 🔲 Hardware Divide					
	Cancel < Prev Next > Finish				

4. Click the JTAG Debug Module tab and choose the selected tab shown in Figure 2-4.

Select a debugging level:				
🔾 No Debugger	⊙Level 1	⊖Level 2	O Level 3	OLevel 4
	JTAG Target Lonnection Download Software Software Breakpoints	n Ng Target Connection Download Software Software Breakpoints 2 Hardware Breakpoints 2 Data Triggers	Drawnload Software Download Software Software Breakpoints 2 Hardware Breakpoints 2 Data Triggers Instruction Trace On-Chip Trace	Drawfload Software Download Software Software Breakpoints 4 Data Triggers Instruction Trace Data Trace On-Chip Trace Off-Chip Trace
No LEs	300-400 LEs	800-900 LEs	2400-2700 LEs	3100-3700 LEs
No M4Ks	Two M4Ks	Two M4Ks	Four M4Ks	Four M4Ks
Advanced debug licenses can be purchased from FS2. http://www.fs2.com/ On-Chip Trace Buffer: 128 Frames V				

Figure 2-4. Nios II Configuration Wizard -JTAG Debug Module

- **5.** Clicking **Finish.** You return to the SOPC Builder **System Content** tab and an instance of the CPU named cpu_0 now appears in the table of available components.
- 6. Right click cpu_0 and select Rename.
- 7. Type cpu and press Enter.

It is recommended to rename the components for avoiding ambiguity in further design process.

JTAG UART

The **JTAG UART** interface component is added to reduce the number of connections necessary to 'talk' to the Nios II. To add it

 Select Communication > JTAG UART and click Add.. The JTAG UART - jtag_uart_0 wizard displays as shown in Figure 2-5.

Figure 2-5. JTAG UART Configuration Wizard

🤨 JTAG UART - jtag_uart_0	×
Configuration Simulation	
Write FIFO (data from Avalon to JTAG)	
Depth: 64 💌 IRQ Threshold: 8	
Construct using registers instead of memory blocks	
Read FIFO (data from JTAG to Avalon)	
Depth: 64 V IRQ Threshold: 8	
Construct using registers instead of memory blocks	
Cancel < Prev Next > Finish	

- 2. Accept the default options by clicking **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the JTAG UART named jtag_uart_0 now appears in the table of available components.
- 3. Right click **jtag_uart_0** and select **Rename**.
- 4. Type jtag_uart and press Enter.

Timer

The Timer is necessary for some of the default device drivers provided in the HAL system library, for example, the JTAG UART. To add the timer perform the following steps:

- 1. Choose Other > Interval Timer and click Add.
- **2.** Leave the default settings in the Avalon Timer **timer_0 window**. Figure 2-6.

Figure 2-6. Timer Configuration Wizard

Timeout Period	
Initial Period:	1 msec 💙
Input Clo	ck Frequency: 48 MHz
- Hardware Options	
Preset Configurations:	Full-featured
Registers	
Vritable period	
🔽 Readable snapsł	not
🛃 Start/Stop contr	ol bits
- Output Signals	
Timeout pulse (1	clock wide)
System reset on	timeout (Watchdog)
	amooat (natanaog)

- **3.** Click **Finish**. You return to the SOPC Builder **System Contents** tab and an instance of the Timer named timer_0 now appears in the table of available components.
- 4. Right click timer_0 and select Rename.
- 5. Type timer and press Enter.

Tristate Bridge

For the ADC-DAC system to communicate with Tristate memory external to the FPGA on the DE2, you must add a bridge between the Avalon bus and external memory controller. To add this:

8

 Select Bridges > Avalon Tri-State Bridge and click Add. The Avalon Tri-State Bridge - tri_state_bridge_0 wizard displays. See Figure 2-7.

Figure 2-7. Avalon Tristate Bridge Configuration Wizard

💶 Avalon Tristate Bridge - tri_state_bridge_0 🛛 🔀
Incoming Signals Registered Increases off-chip Fmax, but also increases latency.
Not registered Reduces latency, but also reduces off-chip Fmax. NOTE: Check the Input Setup Times analysis in the Quartus Compilation Report to be sure your bus inputs meet system-level timing requirements.
Outgoing address and control signals are always registered. Cancel < Prev

- 2. Click Finish. You return to the SOPC Builder System Content tab and an instance of the Tri_state_bridge named tri_state_bridge_0 now appears in the table of available components.
- 3. Right click tri_state_bridge_0 and select Rename.
- 4. Type tri_state_bridge and press Enter.

SRAM Memory

Depending on which hardware you are using user has to select the external SRAM or any memory. We shall be using SRAM avialable on DE2 board.

SLS has developed a generic SRAM controller which is provided in the reference design directory.

Therefore, before following the steps mentioned below, copy the contents of the folder <.../SLS_ADC-DAC_Board\ADC-DAC_Reference_Designs/ Components> to <Quartus installation path>/quartus/sopc_builder/components/ directory. Go to SOPC builder File menu and click Refresh Component List.

To add SRAM Memory Controller perform the following steps:

- 1. Select Memory > SLS_IS61LV6416L_10T_SRAM-sram_0 and click Add.
- 2. The SRAM wizard displays. See Figure 2-8.
- 3. Set Memory Size = 512 kB

Figure 2-8. SRAM Memory Configuration Wizard

LS_IS61LV6416L_10T_SRAM - sram_0
Attributes
Static RAM
This is a 10 ns 16-bit SRAM component, which can also be customized for Address span for other 16-bit 10 ns SRAMs.
Memory Size: 512 kB 🗸
18 Word Aligned Address Bits
Generic Memory Model (Simulation Only)
Include a functional memory model in the system testbench.
Cancel < Prev Next > Finish

- **4.** Click **Finish**. You return to the SOPC Builder **System Content** tab and an instance of the SRAM named sram_0 now appears in the table of available components.
- 5. Right click **sram_0** and select **Rename**.
- 6. Type sram and press Enter.

Parallel I/O for ADC Output Start Bit

To provide an interface for ADC Output Start Bit on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- **2.** Specify the Options. See Figure 2-9.

10

- Width =1 bits
- Direction = Output ports only.
- **3.** Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.
- 5. Type adc_start and press Enter.

Figure 2-9. PIO Configuration Wizard For ADC Output Start Bit

🛓 Avalon PIO - pio_0	×
Basic Settings Input Options Simulation	
└ Width	
1 bits	
PIO width must be between 1 and 32 bits	
O bioirectional (tri-state) ports	
 Input ports only 	
Both input and output ports	
o both input and output ports	
Output ports only	
Cancel < Prev Next >	FINISh

Parallel I/O for ADC Output Enable Bit

To provide an interface for ADC Output Enable Bit on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- 2. Specify the Options. See Figure 2-10.
 - Width =1 bits
 - Direction = Output ports only.

- **3.** Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.
- 5. Type adc oe and press Enter.

Figure 2-10.	PIO Confid	nuration	Wizard	For ADC	Output	Enable	Bit
		,					

👱 Avalon P	10 - pio_0	×
Basic Settings	Input Options Simulation	
Vidth		
	1 bits	
PIC) width must be between 1 and 32 bits	
O Bidire	ctional (tri-state) ports	
🔵 Input	ports only	
🔵 Both i	nput and output ports	
(O Outpu	ut ports only	
Cancel	<pre></pre>	Finish

Parallel I/O for ADC Output Address Latch Enable Bit

To provide an interface for ADC Output Address Latch Enable Bit on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- 2. Specify the Options. See Figure 2-11.
 - Width =1 bits
 - Direction = Output ports only.
- **3.** Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.

12

5. Type adc_ale and press Enter.

Figure 2-11.PIO Configuration Wizard for ADC Output Address Latch Enable Bit

🖳 Avalon P10 - pio_0	
Basic Settings Input Options Simulation	
└ Width	
1 bits	
PIO width must be between 1 and 32 bits	
Direction	
O Bidirectional (tri-state) ports	
O Input ports only	
O Both input and output ports	
Output ports only	
Cancel < Prev Next > Finish	N

Parallel I/O for ADC Output Address Bits

To provide an interface for ADC Output Address Bits on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- 2. Specify the Options. See Figure 2-12.
 - Width =3 bits
 - Direction = Output ports only.
- **3.** Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.
- 5. Type adc_add and press Enter.

🗖 Avalon PIO - pio_0	Þ
Basic Settings Input Options Simulation	
Width 3 bits	
PIO width must be between 1 and 32 bits	
Direction	
O Bidirectional (tri-state) ports	
O Input ports only	
O Both input and output ports	
Output ports only	

Figure 2-12. PIO Configuration Wizard For ADC Output Address Bits

Parallel I/O for ADC Input End Of Character Bit

To provide an interface for ADC Input End of Character Bit on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- 2. Specify the Options. See Figure 2-13.
 - Width =1 bits
 - Direction = Input ports only.
 - Accept Input Options and Simulation as default.
- **3.** Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.
- 5. Type adc_eoc and press Enter.

14

🛓 Avalon PIO - pio_0 🛛 🔀				
Basic Settings Input Options Simulation				
Width I bits PIO width must be between 1 and 32 bits				
Direction O Bidirectional (tri-state) ports				
Both input and output ports				
Output ports only				
Cancel < Prev Next > FinishN				

Figure 2-13.PIO Configuration Wizard For ADC Input End Of Character Bit

Parallel I/O for ADC Input Data Byte

To provide an interface for ADC Input Data Byte on DE2 Board, add the PIO by performing the following steps:

- 1. Select **Other > PIO** (Parallel IO) and click **Add.**
- 2. Specify the Options. See Figure 2-14.
 - Width =8 bits
 - Direction = Input ports only.
 - Keep the default settings under **Input Options** and **Simulation** as default.
 - Click Finish
- **3.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.
- 5. Type adc_data and press Enter.

Basic Settings Input Options Simulation Width B bits PIO width must be between 1 and 32 bits Direction Direction Directional (tri-state) ports Input ports only	Settings Input Options Simulation	🚊 Avalon Pl() - pio_0	X
Width 8 bits PIO width must be between 1 and 32 bits Direction Bidirectional (tri-state) ports input ports only	8 bits PIO width must be between 1 and 32 bits ection 9 Bidirectional (tri-state) ports input ports only 9 Both input and output ports 0 Output ports only	Basic Settings	Input Options Simulation	
Direction Bidirectional (tri-state) ports Input ports only	Bidirectional (tri-state) ports input ports only Both input and output ports Output ports only	PIO v	8 bits width must be between 1 and 32 bits	
) Both input and output ports) Output ports only	Direction	ional (tri-state) ports	
 Both input and output ports) Output ports only	🔿 Both ing	put and output ports	
Output ports only		🚫 Output	ports only	
		Cancel	< Prev Next > Fi	nish [

Figure 2-14. PIO Configuration Wizard For ADC Input Data Byte

Parallel I/O for DAC Output Data Byte

To provide an interface for DAC Output Data Byte on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- 2. Specify the Options. See Figure 2-15.
 - Width =8 bits
 - Direction = Output ports only.
 - Click Finish.
- **3.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.
- 5. Type dac_data and press Enter.

Basic Settings	Input Options Simulation
width	8 bits
PIO	width must be between 1 and 32 bits
Direction -	
O Bidirect	tional (tri-state) ports
🔘 Input p	ports only
🔵 Both in	put and output ports
O Dutput	t ports only

Figure 2-15. PIO Configuration Wizard For DAC Output Data Byte

Parallel I/O for LEDs

To provide an interface for led's on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- 2. Specify the Options. See Figure 2-16.
 - Width =4 bits
 - Direction = Output ports only.
 - Click Finish.
- **3.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 4. Right click **pio_0** and select **Rename**.
- 5. Type led_pio and press Enter.



🛓 Avalon PIO - pio_0	X
Basic Settings Input Options Simulation	
Width 4 bits PIO width must be between 1 and 32 bits	
Direction Direction Directional (tri-state) ports	
O Input ports only	
Both input and output ports Output ports only	
Cancel < Prev Next > Fi	nish []

Parallel I/O for Pushbutton Switches

To provide an interface for Pushbutton Switches on DE2 Board, add the PIO by performing the following steps:

- 1. Select Other > PIO (Parallel IO) and click Add.
- 2. Specify the Options. See Figure 2-17.
 - Width =4 bits
 - Direction = Input ports only.

Figure 2-17.PIO Configuration Wizard For Pushbutton Switches-Basic Settings

🖳 Avalon PIO - pio_0
Basic Settings Input Options Simulation
Width 4 bits PIO width must be between 1 and 32 bits Direction 0 Bidirectional (tri-state) ports 1 Input ports only 0 Both input and output ports
Output ports only
Cancel < Prev Next > Finish

- 3. Click Input Options Tab
 - Under Edge Capture Register, Check **Synchronously Capture**. Under Synchronously capture, select option **Either Edge**. See Figure 2-18.
 - Under Interrupt, check **Generate IRQ**. Under Generate IRQ, select **Edge option**. See Figure 2-18.
- 4. Click **Finish.** You return to the SOPC Builder **System Contents** tab and an instance of the PIO named pio_0 now appears in the table of available components.
- 5. Right click **pio_0** and select **Rename**.

6. Type key_pio and press Enter.

Figure 2-18.PIO Configuration Wizard For Pushbutton Switches-Input Options Settings



After adding all components, the complete SOPC builder system looks like Figure 2-19.



Figure 2-19.SOPC Builder After Adding All Components

Generating the System

To generate the design logic, perform the following steps.

- 1. Click the System Generation tab.
- 2. Specify the following settings from the Options window. Figure 2-20.
 - HDL: Check this box
 - Simulation: Check this box if you have Modelsim installed and would like to simulate the design.

Figure 2-20. Generating the System

🖳 Altera SOPC Builder - System_Top
File Module System View Tools Help
System Contents Nios II More "cpu" Settings System Generation
Options
Info: Quartus II Shell was successful. 0 errors. 0 warnings
Info: Allocated 40 megabytes of memory during processing
Info: Processing ended: Tue Jul 10 11:05:17 2007
Info: Elapsed time: 00:00:03
2007.07.10 11:05:17 (*) Completed generation for system: System Top.
2007.07.10 11:05:17 (*) THE FOLLOWING SYSTEM ITEMS HAVE BEEN GENERATED:
SOPC Builder database : E:/Mrkt Dev/projects/sc adcdac refdes de2/System Top.ptf
System HDL Model : E:/Mrkt Dev/projects/sc adcdac refdes de2/System Top.v
System Generation Script : E:/Mrkt_Dev/projects/sc_adcdac_refdes_de2/System_Top_generation_script
2007.07.10 11:05:17 (*) SUCCESS: SYSTEM GENERATION COMPLETED.
Press 'Exit' to exit.
cpu was generated as a time-limited OpenCore Plus module and will time-out unless compiled in Quartus II with a valid license. cpu: defaulting Reset Address, Exception Address to sram cpu: The reset address points to volatile memory. Execution of undefined code may occur upon reset. Done checking for updates.
Exit < Prev Next > Re-Generate

- 3. Click Generate. See Figure 2-20.
- 4. When generation is complete, the SYSTEM GENERATION COMPLETED message displays. "DO NOT EXIT SOPC BUILDER AT THIS POINT." We will return to this window prior to testing the system with software.

Adding the Quartus II Symbol to the BDF

During generation, SOPC Builder creates a symbol of the **System_Top**, for using in Quartus II. To **add the symbol** perform the following steps:

- 1. Select File (menu) > New.
- 2. Under Device Design Files, select Block Diagram/Schematic File. See Figure 2-21.

Figure 2-21.New Block Design File



3. Click **OK**. You are return to the **Quartus II software** and double click anywhere inside the **BDF window**. The Symbol dialog box appears. See Figure 2-22.

Figure 2-22.BDF System_Top Symbol

Symbol	
Libraries: Conter Co	System_Top clk reset_n out_port_from_the_adc_add[2.0] in_port_to_the_adc_data[7.0] in_port_to_the_adc_eoc out_port_from_the_adc_oe
	out_port_from_the_adac_start out_port_from_the_dac_data[7.0]
Name: System_Top	out_port_from_the_led_pio[3.0] read_n_to_the_sram select_n_to_the_sram tri_state_bridge_address[18.0]
Repeat-insert mode Insert symbol as block Launch MegaWizard Plug-In MegaWizard Plug-In	tri_state_bridge_byteenablen[1.0]
OK Cancel	

- **4.** From Libraries, expand the **Project directory** by clicking the + sign nearby.
- 5. Click System_Top. A large symbol will appear representing the ADCDAC system module you just created.
- 6. Click OK. The Symbol dialog box closes and an outline of the System_Top symbol is attached to the pointer.
- 7. Place the **symbol** in the Block Diagram file by clicking the left mouse button.
- You can generate the input and output node by selecting the symbol and right click on it and select Generate Pins for Symbol Ports. See Figure 2-23.

Figure 2-23. Generate Pins

100				
- IS	ystem_Top			
_				
	~			
-	reset o			
	_100_A0	ron_tre_edc_edd[2.0]		
	out	port from the add ale		
-	In_port_to_the_edc_deta(7.0)			
-	in port to the add exc			
	a start of the sta			
			diam'ne an	
ъ	out i	COR	CENTRA	
- E		Const	Children	
		2007	COPPE	
		Dada		
		Delete	Del	
-	in port to the key piol3.01			
		Locate		• • • • • • • • • • • • • • • • • • • •
				·
	04_00	MegaWizard Plug-In	Manager	
	-			
		Edit Selected Symbol		
		Course Desiders File		
		Open Design Pile		
	P1_000	Lindaha Sumbol or Block		
	bi state t	oposite symbol or below.		
	N 1	File Horizontal		
	1	Plip Vertical		
1.00	a	Notate by Degrees		
а,				
		Taxes		
		2.00m		
		Generate Dife for Sumbol		
		developer of the street		
		Properties		

9. Input, output and bidirectional pins for the symbol will be automatically generated and connected accordingly. See Figure 2-24.

Figure 2-24. Autogenerated Pins for Symbol



10. The autogenerated pin names can be changed. Right click the **pin name** and select **Properties.** The Pin Property dialog box appearsas shown in Figure 2-25.

Figure 2-25. Pin Properties

Pin Properties 🛛 🔀
General Format
To create multiple pins, enter a name in AHDL bus notation (for example, "name[30]"), or enter a comma-separated list of names.
Pin name(s): dac_data(70
OK Cancel

11. You can give all the name according to Table 2-1 . The names are given for ease of reference.

Table 2-1. Pin Names in Final BDF		
Pin Name of the Symbol	Custom Pin Names	
clk	clk	
reset_n	reset_n	
in_port_to_the_adc_data[70]	adc_data[70]	
in_port_to_the_adc_eoc	adc_eoc	
in_port_to_the_key_pio[30]	key[30]	
out_port_from_the_adc_add[20]	adc_add[20]	
out_port_from_the_adc_ale	adc_ale	
out_port_from_the_adc_oe	adc_oe	
out_port_from_the_adc_start	adc_start	
out_port_from_the_dac_data[70]	dac_data[70]	

out_port_from_the_led_pio[30]	ledg[30]
read_n_to_the_sram	sram_oe_n
select_n_to_the_sram	sram_ce_n
tri_state_bridge_address[180]	sram_add[180]
tri_state_byteenablen[10]	sram_be_n[10]
tri_state_bridge_data[150]	sram_data[150]
write_n_to_the_sram	sram_we_n

12. Add the counter by following the steps below:

- Double click anywhere inside the **BDF** window.
- It will pops up the **Symbol** window.
- Select <*dirve path>:/altera/../megafunctions/arithmatic/ Ipm_Counter*. See Figure 2-26. Click **OK**





- It will pops up the MegaWizard Plug-In Manager window. See Figure 2-27.
- Give the name counter in the Output file text box. See Figure 2-27.

Figure 2-27.MegaWizard Plug-In Manager [page 2C]

MegaWizard Plug-In Manager[page 2c]		MegaWizard Plug-In	Manager[page 2c]
Selected Megafunctions: LPM_COUNTER	Which type of output file do you want to create? AHDL VHDL Verilog HDL What name do you want for the output file? Browse E:Whik_Dev\projects\sc_adcdac_refdes_de2\typn_counter0 I Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in the global user libraries specified in the Uptions dialog box (Tools menu), or a user library specified in the Uptions dialog box (Tools menu), or a user library specified in the Uptions dialog box (Tools menu), or a user library specified in the Uptions dialog box (Tools menu), or a user library specified in the Uptions dialog box (Tools menu), or a user library specified in the Uption dialog box (Tools menu), or a user library specified in the Uption set library directories are:	Selected Megafunctions	Which type of output file do you want to create? AHDL VHDL Veriog HDL What name do you want for the output file? Browse E:Wrinkt_Dev\projects\sc_adcdac_refdes_de2\counter Note: To compile a project successfully in the global user libraries specified in the Options dialog box (Tools menu), or a user library specified in the User Libraries page of the Settings dialog box (Assignments menu). Your current user library directories are:
Don't ask me for an output file name or the output file format again. In future, name output files automatically and use the current output file format. (Note: You can turn the Block Editor's auto naming and auto format selection on and off with the Options command in the Tools menu.) Cancel < Back Next> Finish		Don't ask me for an o In future, name outpu (Note: You can turn t on and off with the D	utput file name or the output file format again. It files automatically and use the current output file format. he Block Editor's auto naming and auto format selection pitons command in the Tools menu.) Cancel < Back Next > Finish

- Click **Next**. You will go to the next page of MegaWizard.
- Select the output bus width as 6 bits. See Figure 2-28.

Figure 2-28.MegaWizard Plug-In Manager [page 3]

MegaWizard Plug-In Manager - LPM_COUNTER [page 3 of 7]					
tPM Version	_COUNTE	R	About	Documentation	
Parameter Settings General General General	n 3 Summary				
Settings Lbrary General Optional Inputs Image: setting setting of the setting					

- Click **Next**. You will see the page 4 of MegaWizard.
- Leave the default settings as shown in Figure 2-29.

Figure 2-29.MegaWizard Plug-In Manager [page 4]

MegaWizard Plug-In Manager - LPM_COUNTER [page 4 of 7]			
2 LPM	n 6.1 Qocumentation		
1 Parameter 2 Simulat Settings Library General General2	on 3 Summary Optional Inputs		
counter up counter clock q[50]	Which type of counter do you want? Plain binary Modulus, with a count modulus of		
	Do you want any optional additional ports?		
Resource Usage 6 lut + 6 reg	Cancel < Back Next > Einish		

• Click Next.

Figure 2-30.MegaWizard Plug-In Manager [page 5]

MegaWizard Plug-In Manager - LPM_COUNTER [page 5 of 7]			
2 LPM Versio	I_COUNTER n 6.1	About	Documentation
1 Parameter 2 Simulat Settings Library General General2	ion 3 Summary Optional Inputs		
counter up counte clock q[50]	Do you want any optional Synchronous inputs Caa Load Set Set to all 1's Set to 0	Asynchron Clear Load Set Set Set	to all 1's
Resource Usage 6 lut + 6 reg	Cancel	< Back N	ext >

- Under Synchronous inputs, Check the **Clear box**. See Figure 2-30.
- Click Next.
- You will see page 6 of MegaWizard. SeeFigure 2-31.

Figure 2-31.MegaWizard Plug-In Manager [page 6]

MegaWizard Plug-In Manager - LPM_COUNTER [page 6 of 7] Simulation Libraries				
2 LPM	_COUNTER	About Documentation		
1 Parameter 2 Simulation Settings Library	on 3 Summary			
Counter up counter clock q[50]	To properly simulate the generated design fi file(s) are needed File Description Ipm LPM megafunction simulation library	les, the following simulation model		

- Click Next.
- You will see page 7 of MegaWizard. See Figure 2-32.

Figure 2-32.MegaWizard Plug-In Manager [page 7]

MegaWizard Plug-In Manager	- LPM_COUNTER [p	age 7 of 7] Summa	ry	\mathbf{X}
🎲 LPM_CC	UNTER			
Version 6.1			<u>A</u> bout	<u>D</u> ocumentation
1 Parameter 2 Simulation 3 Settings Library	Summary			
counter up counter clock q[50]	Turn on the files you w automatically generate Finish to generate the subsequent MegaWiza The MegaWizard Plug- directory: E:\Mrkt_Dev	ish to generate. A gray che rd, and a red checkmark ind selected files. The state of rd Plug-In Manager session In Manager creates the sele \projects\sc_adcdac_refde	eckmark indic icates an op each checkt s. ected files in s_de2\	ates a file that is tional file. Click box is maintained in the following
	File	Description		
	Counter.tdf	Variation file		
	Counter.inc	AHDL Include file		
	counter.cmp	VHDL component declar	ation file	
	✓ counter.bsf	Quartus II symbol file		
	Counter_inst.tdf	Instantiation template file		
Resource Usage				
6 lut + 6 reg				
		Cancel	< <u>B</u> ack	Next > Einish

- Click Finish.
- You will return to BDF window
- Place the **Counter symbol** in the Block Diagram file by clicking the left mouse button
- **13.** Now the **clk** input of counter is directly connected to the **clk** input of the System_Top. The **sclr** input of the counter is connected to the **reset** input of the System_Top using **NOT gate**.
- **14.** For selection of the NOT gate double click inside the block diagram and select *<drive path>:/.../library/primitives/logic/not*. See Figure 2-33.





15. Then by using orthogonal node tool, you can connect the input of NOT gate to the reset input of the System_Top and the output of the NOT gate to the sclr input of the counter. See Figure 2-34.



Figure 2-34. NOT Gate Connection

16. Now you have to expand the line as shown in Figure 2-35. Give the name **cntr_out[5..0**].





- **17.** Now for the ADC clock output, add output symbol by selecting it from the library *<drive path>:/.../library/primitives/pin/output.*
- **18.** Give the output pin name as **adc_clk**.
- Do not connect it with the output of the counter strech its line as shown in Figure 2-36. Give the name as cntr_out[5] and press Enter.



Figure 2-36.Counter Output Connection

- **20.** After assignment of the IO pins, select **Assignment>Settings.** The Settings Dialog Box pops up as shown in Figure 2-37.
- 21. In Device dialog box click the button **Device & Pin Options**.
- 22. Device and Pin options dialog box pops up as shown in Figure 2-38.
- **23.** Click the **Unused pins** Tab. Select **As Input tri-stated** in Reserve all unused pins combo box.
- 24. Click OK. You will return to Device Settings window.
- 25. Click OK.
- **26.** Now select the **Processing** from the menu bar and click on **Start**.and select the **Start Analysis And Synthesis.**
- 27. For assignment open the Assignments from menu bar and select the Assignment Editor. Select the category as pin and then you can see the list of all pins used in the project. If you do not see the pins then go to View menu and select Show All Known Pin Names.

Figure 2-37.Settings

Settings - sc_adcdac_refdes_de2		×
Settings - sc_adcdac_refdes_de2 Category: General - Files - User Libraries (Current Project) - Device - Operating Conditions - Compilation Process Settings - EDA Tool Settings - Analysis & Synthesis Settings - Analysis & Synthesis Settings - Filter Settings - Assembler - Design Assistant - Signal Tap II Logic Analyzer - Logic Analyzer Interface - Simulator Settings - PowerPlay Power Analyzer Settings	Device Select the family and device you want to target for compilation. Eamily: Cyclone II Device & Pin Options Target device C Auto device selected by the Fitter © Specific devices selected in 'Available devices' list C Other: n/a Available devices: Name LEs Memor Embed PLL EP2C50F672C6 50528 S9432 172 4 EP2C50F672C6 S0528 594432 LE 1152000 Auto Automation	
	EP2C50F672C6 50528 594432 172 4 EP2C70F672C6 68416 1152000 300 4 Migration compatibility Companion device HardCopy II: Imit DSP & RAM to HardCopy II device resources O migration devices selected Imit DSP & RAM to HardCopy II device resources OK Cancel	

28. Now you have to go to the sc_adcdac_refdes_de2.csv file at SLS_ADC-DAC_Board/ADC-DAC_Reference_Designs folder and select the pins and copy it. Open the Quartus II software and paste it in the assignment editor.

Figure 2-38. Device & Pin Options

Device & Pin Options
Pin Placement Error Detection CRC Capacitive Loading Board Trace Model General Configuration Programming Files Unused Pins Dual-Purpose Pins Voltage Specify device-wide options for reserving all unused pins on the device. To reserve Specify device-wide options for reserving all unused pins on the device. To reserve
individual dual-purpose configuration pins, go to the Dual-Purpose Pins tab. To reserve other pins individually, use the Assignment Editor.
Reserve all <u>u</u> nused pins: As input tri-stated
Reserves all unused pins on the target device in one of 5 states: as inputs that are tri-stated, as outputs that drive ground, as outputs that drive an unspecified signal, as input tri-stated with bus-hold, or as input tri-stated with weak pull-up.
<u>R</u> eset
Cancel

- **29.** After adding Inputs/Output, **Inouts and pin assignments** the final BDF looks like Figure 2-39. Please refer the provided Reference Design.
- 30. Choose File>Save.

Figure 2-39. Final BDF



Note: to Figure 2-39.

- (1) In this tutorial we are using only one memory (SRAM). In order to disable SDRAM and FLASH memories, the pins are at Pull up status.
- (2) You can assign the pin number using the SLS_ADC-DAC_Board/ADC-DAC_Reference_Designs/ sc_adcdac_refdes_de2.csv file.

Compiling the Design

During compilation, the Compiler locates and processes all design, project files, generates messages and reports related to the current compilation, creates the **SRAM object file** (.sof) as well as any **optional programming files**.

To compile the sc_adcdac_refdes_de2 design, follow these steps:

- 1. Choose Start Compilation (Processing menu), or click the Start Compilation toolbar button. If you get a message asking if you want to save the changes you made the BDF file, choose Yes.
- 2. When compilation completes, you can view the results in the ADC-DAC Compilation Report window. See Figure 2-40.

Figure 2-40.Compilation Report Window



3. Programming



After successful compilation, the Quartus II Compiler generates one or more programming files that the Programmer uses to program or configure a device.

To program your design, perform the following steps.

- 1. Choose **Programmer** (Tools menu). The Programmer window opens.
- 2. In the Mode list of the programmer window, make sure JTAG is selected.
- **3.** Click **Hardware Setup..** to configure the programming hardware. The Hardware Setup dialog box appears.
- 4. From the Hardware column, select USB Blaster .
- 5. Click Close to exit the Hardware Setup window.
- 6. In the Programmer window, turn on **Program/Configure**. See Figure 3-1.
- 7. Click Start.

Figure 3-1. Programming Window

🍕 Quartus II - E:/Mrkt_Dev/project	s/sc_adcdac_refdes_de2/sc_adcdac_refdes_d	le2 - sc_adcdac_ref	des_de2 - [sc_	adcdac_refdes_de	2_time_limited	. 💶 🗗 🗙
File Edit View Project Assignments	Processing Tools Window Help					_ 8 ×
🛛 🗅 🖨 🗶 😓 😓 🖉	🗠 🔽 sc_adcdac_refdes_de2 💽 💥 🖌	🥝 🦁 🗇 🔍 🕨	🔊 👦 🏷 🔇	🐌 😓 🧶	20	
Project Navigator	sc_adcdac_refdes_de2.bdf	😔 Compilation Report - F	Flow Summary	📔 sc_adco	lac_refdes_de2_tim	e_limited
Entity Logic	Hardware Setup USB-Blaster [USB-0]		Mode: ITAG	-	rogrees:	
Cyclone II. EF2L30F672L6			mode. Janka	·	iogress.	0 ~
BDF W_GGGGGGG_GGG_GGG_	Enable real-time ISP to allow background programmin	ng (for MAX II devices)				
	Start File D	levice Chi	ecksum Usero	code Program/ Configure	Verify Blank- Check B	Examine Secu Bit
	sc_adcdac_refdes_de2 E	P2C35F672 005	52AC1E FFFFF	FFF 🔽		
	Auto Detect					
	× Delete					
	C ¹ AND I					
Hierarchy E Files Posign Units	Add File					
Status	Change File					
Full Compilation	🔛 Save File					
Analysis & Synthesis 100 %	Add Device					
Partition Merge 100 %	- Min Line					
Assembler 100%						
Classic Timing Analyzer 100 % 🛩	Down					
	<					>
× 🚯 Info: Started Full Com	pilation at Tue Jul 10 11:07:35 2007					
🔹 🔅 Info: Ended Full Compi	lation at Tue Jul 10 11:14:17 2007					
Info: Started Full Com	pilation at Tue Jul 10 11:14:57 2007					
Info: Ended Full Compi	lation at Tue Jul 10 11:21:40 2007					
 Info: Started Full Compi Info: Ended Full Compi 	pliation at fue Jul 10 11:33:52 2007					
 Info: SRAM Object File 	sc adcdac refdes de2 time limited.sof c	ontains time-limit	ted megafuncti	ion that supports	OpenCore Plus f	feature N
System (Processing) Extra Info	λ Info λ Warning λ Critical Warning λ Error λ Supp	pressed /				>
🖁 Message: 0 of 7 👘 🐁	Location:					Locate
Eor Help, press E1	4.7 					NUM
rornop, prosini				19 0 - 10 - 1	iue	PROPT



4. Running the Software in Nios II IDE

We will be using Nios II Integrated Development Environment (IDE) to run our software on top of Nios II System. To start the Nios II IDE from the Quartus II software, perform the following steps:

1. Click the **System Generation** tab of the SOPC Builder, then click the **Run Nios II IDE** button.

2. From the opening window, choose File>New>Nios II C/C++ Application. See Figure 4-1.

3. You will see the New Project Wizard Dialog Box as shown in Figure 4-2.

Figure 4-1. New Project

Creating the

Project

le	Edit	Refactor	Navigate	Search	Project	Tools	Rur	n Windo	w	Help		
	New				Alt+S	Shift+N	•	📑 Proje	ect			
	Open F	ile						🔊 Mios	пс	IC++ Applicati	00	1.
	Close				Ctrl+	W		Nios	IIS	ystem Library	011	1
	Close A	.II			Ctrl+	Shift+W		💦 Nios	ΠU	iser-Makefile C	C++	Application
IJ	Save				Ctrl+	s		💣 Nios	II C	:/C++ Library		
2	Save A	s						💕 Sour	ce F	older		
ù	Save A	1			Ctrl+	Shift+S		💣 Sour	ce F	ile		
	Revert							📝 Head	der F	File		
	Move							🞯 Clas:	5			
	Renam	e			F2			📑 Othe	er			
	Refrect	·			E2		L					

- **4.** The user has been asked to select Blank Project, since SLS has provided a reference C file to access the ADC-DAC on DE2. To import this C file into this project do the following steps.
 - Select the "Blank Project"
 - Give a Name to the project as ADCDAC_DE2_Board.
 - Select the SOPC builder system by clicking on **Browse** and selecting the **System_Top.ptf** file in the reference design. See Figure 4-2.

Figure 4-2. Creating new Nios II C/C++ Application

New Project		
Nios II C/C++ Application Click Finish to create application with E:\Mrkt_Dev\projects\sc_adcdac_refo	a default system library as les_de2\software\ADCDAC_DE2_Board	G
Name: ADCDAC_DE2_Board		
Location: E:\Mrkt_Dev\projects\s	_adcdac_refdes_de2\software	Browse
Select Target Hardware.		
SOPC Builder System: E:\Mrkt_De	v\projects\sc_adcdac_refdes_de2\System_Top.ptf	Browse
CPU: cpu		~
Select Project Template Blank Project Board Diagnostics Count Binary Custom Instruction Tutorial Dhrystone Hello Freestanding Hello NeroC/OS-II Hello World Hello World Hello World Hello World Hello World Hello World Memory Test MicroC/OS-II Message Box	Description Creates a blank project Details Blank Project creates an empty project to which you can add your code. This software example runs on the following Nios II hardware designs: - Standard - Full Featured - Fast - Small	
0	< Back Next > Finish	Cancel

- 5. Click Next
- 6. Click Finish.
- 7. You will observe your project created under the Nios C/C++ Projects Tab. See Figure 4-3.

Figure 4-3. Listing of Your Project under Nios II C/C++ Project Tab



8. Select the **Project (ADCDAC_DE2_Board).** Right Click and hit **Import.** See Figure 4-4.

Figure 4-4. Importing File System

Nios II	C/C++ Projects 🗙 📃 🗖	😰 Import	X
	↓ ↓ @ □	Select Import resources from the local file system into an existing project.	Ľ
⊞-∰ AI	New r Go Into	Select an import source: type filter text General Archive File Sreakpoints Existing Projects into Workspace Selectors	
	Run As Debug As Build Project Clean Project Copy	Altera Altera C/C++ CVS Team	
	Paste Poste Move Rename Post Front	(?) < Back Next > Finish	Cancel

9. You will now be asked for the details of the file to be imported.

• You have to select the file system as shown in Figure 4-4.

• Click **Next**. You will see the import dialog box as shown in Figure 4-5.

Figure 4-5. Import Dialog Box

🔊 Import	×
File system Source must not be empty.	
From directory:	Browse
Filter Types) Select All Deselect All	
Into folder: ADCDAC_DE2_Board	Browse
Options Overwrite existing resources without warning Create complete folder structure Create selected folders only	
2 Sark Next >	Finish
<pre>Sack Next ></pre>	

- Browse to *SLS_ADC-DAC_BoardADCDAC_Reference_Designs/ software*. See Figure 4-6.
- Click **OK.** You will return back to Import Wizard.

Import from directory	? 🗙
Select a directory to import from.	
🖃 🚞 SLS_ADC-DAC_Board	~
ADC-DAC_Demonstrations	
ADC-DAC_Documents	-
🖃 🚞 ADC-DAC_Reference_Designs	
🗉 🛅 sc_adcdac_refdes_de1	
🗉 🛅 sc_adcdac_refdes_de2	
🗁 Software	~
	>
Folder: Software	
Make New Folder	ncel

Figure 4-6. Asking for Details of the files to be imported

- **10.** Select the **sc_adcdac_app.c** file in the template by checking the box. See Figure 4-7.
- 11. Click Finish.

🛿 Import 🔀
File system Import resources from the local file system.
From directory: \\SIs_mobile1\SLS_Intercom\SLS_ADC-DAC_Board\ADC-DAC_Refe 💟 Browse
Software
Filter Types Select All Deselect All
Into folder: ADCDAC_DE2_Board Browse
Options Overwrite existing resources without warning Create complete folder structure Create selected folders only
Cancel

Figure 4-7. Asking for Details of the files to be imported

12. Observe the inserted **sc_adcdac_app.c** file template in the application list. See Figure 4-8.



Figure 4-8. Inserted C file template in the application list

R

You can browse through the contents of the file by double clicking the file Name in the list.

Building the Project

After successful creating yourproject, its now time to build your project. To build your project follow steps below:

1. Right click the **Name of the Project** and select **"Build Project"** to build your project. See Figure 4-9.

Figure 4-9. Build Project

Nios II C/C+	+ Projects 🗙 📃 🗖
🕀 🔁 ADCD <u>AC</u>	DE2 Board
🗄 😂 ADCD	New
	Go Into
	Open in New Window
	Rebuild Index
	Active Build Configuration
	Run As
	Debug As
	Build Project
	Clean Project 😽

2. You will observe the window. See Figure 4-10.

Figure 4-10. Build Process

Build Project
Operation in progress
Building ADCDAC_DE2_Board
И
Run in Background Cancel Details >>

Running the Project

After building the project successfully, its now time to run the project on the hardware. There are two ways to run your project.

1. Select **Run** > **Run.** as shown in Figure 4-11.

Figure 4-11. Run Option



2. It will pops up Run Dialog Box as shown in Figure 4-12.

Figure 4-12.Run Dialog Box

Run			
Create, manage, and run con	figurations		
Image: System of the system Image: System <td< th=""><th>Name: blank_project_0 Nios II H</th><th>W configuration ion 🏂 Debugger 🤯 Source 🔲 <u>C</u>ommon</th><th>Нер</th></td<>	Name: blank_project_0 Nios II H	W configuration ion 🏂 Debugger 🤯 Source 🔲 <u>C</u> ommon	Нер
C Debug hello_world_S.e C Debug hello_world_S.e C Debug hello_world_S.e Nios II Hardware blank_project_0 Nios II	ADCDAC_DE2_Board Nios II ELF Executable: Debug/ADCDAC_DE2_Board.elf	5	Browse
The helo_world_2 Nios II H helo_world_2 Nios II H helo_world_2 Nios II H helo_world_3 Nios II H helo_world_4 Nios II H helo_world_5 Nios II H	 Target Hardware SOPC Builder: System: CPU: Additional nios2-download argu 	E:\Mrkt_Dev\projects\sc_adcdac_refdes_de2\Syste	Browse
igar Nios II HW configu igar Nios II HW configu Nios II Instruction Set Simu Nios II ModelSim Nios II Multiprocessor Colle	Validate Nios II system ID bef	fore software download	
< <u>></u>		Apply	Revert
0		Run	Close

- **3.** Under Project, select the **Nios II C/C++ Porject** by browsing your ADC-DAC Project.
- 4. Click on Target Connection Tab. You will see the Figure 4-13.



Run		\mathbf{X}
Create, manage, and run cor	nfigurations	
Ype filter text C/C++ Local Application C Debug hello_world_5.e Nos II Hardware blank_project_0 Nios II hello_world_1 Nios II H hello_world_3 Nios II H hello_world_4 Nios II H hello_world_5 Nios II H hello_world_5 Nios II H Mios II Instruction Set Simu Nios II Multiprocessor Colle	Name: blank_project_0 Nios II HW configuration Main Target Connection The Debugger Common Help JTAG cable: Page Source Common USB-Blaster [USB-0] Image: Refresh Refresh JTAG device: Image: Refresh Refresh automatic <the device="" has="" processor="" the="" which=""> Image: Refresh Nios II Terminal communication device: Image: Refresh Image: Refresh Host COM port: Image: Refresh Image: Refresh Additional nios2-terminal arguments: Image: Refresh Image: Refresh</the>	
?	Apply Revert	

- 5. Select the target Hardware USB-Blaster [USB-0] under JTAG Cable. Click Refresh.
- 6. Click Apply.
- 7. Click Run.
- **8.** Alternatively you can run the project by Right click the **Project** and select **Run As > Nios II Hardware.** Figure 4-14.

Nios II C/C	🖙 🕂 🕂 🕂 🕂	
	(+ +) @ 🖻 🞸 ▽	
🗉 🖅 🔂 ADCD	AC DE2 Board	
🗄 😂 Adce	New	>
	Go Into	
	Open in New Window	
	Rebuild Index	
	Active Build Configuration	<u> </u>
	Run As	🕨 📷 Nios II Hardware 🔪
	Debug As	Nios II Instruction Set Simulator
	Build Project	III Nios II ModelSim
	Clean Project	

Figure 4-14. Run As-Nios II Hardware Selection

- 9. You will observe the console window as shown in Figure 4-15.
- **10.** The Console Window asks for the selection of choice from the following options:
 - 1. Analog To Digital Converter
 - 2. Digital To Analog Converter
 - 3. Auto Test
 - 4. Exit
 - Please enter your choice :

Figure 4-15.IDE Console Output



11. If you select **Option 1** and press enter, following message will be displayed. See Figure 4-16.

Figure 4-16. Analog to Digital Selection Console Window



- Give the analog input at pin IN0 and connect GND_ADC to ground.
- Enter analog input sample number and press **Enter**. You shall see the sample values displayed.
- Following figure shows the output window for 5V DC analog input at IN0 and 200 samples as analog input sample number. See Figure 4-17.

Here each reading is displayed after 10 samples. It shows the average value of 10 sample. Therefore only 20 readings are displayed here.

Figure 4-17. Analog to Digital Output Console Window

🖻 Console 🗙	×	*	-		1	• [] •	- 0
blank_project_0 Nios II HW configuration (1) [Nios II Hardware] Nios II Terminal Window (7/23/07 6:53 PM)							
, moer i co recurn co recvious menu.							^
Please enter the input sample numbers:200							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000				ĩ			
Volt : 5.000000				r			-
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
Volt : 5.000000							
							~
8							>

12. If you select **Option 2** and press enter, it will display the following message. See Figure 4-18.

Figure 4-18.Digital to Analog Selection Console Window

E Console ×	🔳 🗶 🍇 📑 🛃 🖃 🖛 🗗	•	
blank_project_0 Nios II HW configuration (1) [Nios II Hardware] N	os II Terminal Window (7/23/07 6:53 PM)		
** Digital To Analog Conve	rter **		~
**	**		-
*****	******		
******	*******		
Please read the instructions before driv	ing DAC.		
-> Innut range is in between 0 (o 255.		-
-> Enter -1 to return to Previou	s menu.		
Please enter Input value: 1]			~
		1	

Give the digital input signal range between 0 to 255 and press Enter

The analog output can be measured at IOUT_n pin and you can see the display as shown in Figure 4-19.

Figure 4-19. Digital to Analog Output Console Window



13. If you select **Option 3** and press enter, following message will be displayed. See Figure 4-20.

Figure 4-20.Auto Test Selection Console Window



This is an **AUTO TEST** that requires a loop back connection. Therefore, make the connections as mentioned below.

- Connect IOUT_n i.e. output of the DAC to the IN0 i.e. input of the ADC. forming a loop.
- Connect the GND_ADC to ground

• After successful connection, you will be able to see the following window. See Figure 4-21.

Figure 4-21.Auto Test Output Console Window

 * Analog Input 		Digital	Output	*	-
******	******	*******	*******	* * *	
0.294118	T	0			
0.196078	1	30			
0.274510		60			
0.509804		90			
0.588235		120			
0.549020		150			
0.588235		120			
0.509804		90			
0.274510		60			
0.196078		30			
0.294118		0			

14. If you select **Option 4**, it will exit form the procedure of conversion.



For details on the hardware for ADC-DAC Board refer to the **ADC-DAC Board Reference Manual.**