Creating projects with Nios II for Altera De2i-150

By Trace Stewart CPE 409

CONTENTS

Chapter 1	Hardware Design	1
	1.1 Required Features1.2 Creation of Hardware Design	1 1
Chapter 2	Programming the FPGA	33
Chapter 3	NIOS II Software Build Tools for Eclipse	42
	3.1 Creating a simple Hello World Project	42
	3.2 DCT and Quantization	47

Chapter 1 Hardware Design

This tutorial provides comprehensive information that will help you understand how to create a FPGA based SOPC system with the Nios II processor on your FPGA development board and run software upon it.

1.1 Required Features

The Nios II processor core is a soft-core central processing unit that you could program onto an Altera field programmable gate array (FPGA). This tutorial illustrates you to the basic flow covering hardware creation and software building. You are assumed to have the latest Quartus II and NIOS II EDS software installed and quite familiar with the operation of Windows OS. If you use a different Quartus II and NIOS II EDS version, there will have some small difference during the operation. You are also assumed to possess a DE2i-150 development board (other kinds of dev. Board based on Altera FPGA chip also supported).

The example NIOS II standard hardware system provides the following necessary components:

- Nios II processor core, that's where the software will be executed
- On-chip memory to store and run the software
- Sdram to use other than On-chip memory to store and run software
- JTAG link for communication between the host computer and target
- Hardware (typically using a USB-Blaster cable)
- PIO registers will be used to send data between FPGA and Nios Processor

1.2 Creation of Hardware Design

This section describes how to create the hardware system used for this project including the SOPC feature.

 First step is to open the Quartus program (We will be using Quartus 17.1 in this tutorial but other Quartus versions will work will some minor tweaking). You will then select File→New Project Wizard. See Figures 1-1 and 1-2.



Figure 1-1 Opening New project Wizard

🕥 New Project Wizard					×
Directory, Name, Top-Level Entity					
What is the working directory for this project?					
C:/Nios_Tutorial					
What is the name of this project?					
Nios_Tutorial					
What is the name of the top-level design entity for this project? This design file.	name is case se	ensitive and mu	st exactly match	the entity nan	ne in the
Nios_Tutorial					
Use Existing Project Settings					
	< Back	Next >	Finish	Cancel	Help

Figure 1-2 Input the working directory, the project name, and top-level design entity.

 Click Next through the windows until you get to the Family, Device, & Board Settings window. You will then choose your Device Family and the device name as in Figure 1-3. We are using the De2i-150 board, so we choose Cyclone IV GX with name EP4CGX150DF31C7.

Device Board									
elect the family and d ou can install additior	evice you want to tar al device support wi	rget for com ith the Instal	pilation. l Devices comma	nd on the Too	ls menu.				
o determine the version	on of the Quartus Pri	me software	e in which your ta	rget device is	supported, refe	er to the <u>Device Suppor</u>	<u>t List</u> webpage.		
Device family						Show in 'Available d	levices' list		
Family: Cyclone IV (БХ				•	Package:	Any		•
Device: All					Ψ.	Pin count	Anv		•
Tanant darian						Core speed grade:	Anv		•
						core spece grade.	niy		
O Auto device selec	ted by the Fitter					Name filter:			
Specific device se	lected in 'Available d	levices' list				Show advanced	devices		
Other: n/a									
vailable devices:									
vailable devices: Name	Core Voltage	LEs	Total I/Os	GPIOs	GXB Tra	nsmitter Channel PMA	GXB Receiver Channel PMA	PCIe Hard IP Blocks	Me
vailable devices: Name P4CGX150DF27C7	Core Voltage	LEs 149760	Total I/Os 426	GPIOs 382	GXB Trai	nsmitter Channel PMA	GXB Receiver Channel PMA	PCIe Hard IP Blocks	Me 66355
Name P4CGX150DF27C7 P4CGX150DF27C8	Core Voltage 1.2V 1.2V	LEs 149760 149760	Total I/Os 426 426	GPIOs 382 382	GXB Trai 8	nsmitter Channel PMA	GXB Receiver Channel PMA 8 8	PCIe Hard IP Blocks 1 1	Me 66355 66355
Name P4CGX150DF27C7 P4CGX150DF27C8 P4CGX150DF27I7	Core Voltage 1.2V 1.2V 1.2V	LEs 149760 149760 149760	Total I/Os 426 426 426	GPIOs 382 382 382	GXB Trai 8 8 8	nsmitter Channel PMA	GXB Receiver Channel PMA 8 8 8	PCIe Hard IP Blocks 1 1 1 1	Me 66355 66355
Name P4CGX150DF27C7 P4CGX150DF27C8 P4CGX150DF27I7 P4CGX150DF27I7AF	Core Voltage 1.2V 1.2V 1.2V 1.2V	LEs 149760 149760 149760 149760	Total I/Os 426 426 426 426	GPIOs 382 382 382 382 382	GXB Trai 8 8 8 8 8 8	nsmitter Channel PMA	GXB Receiver Channel PMA 8 8 8 8 8	PCIe Hard IP Blocks 1 1 1 1 1	Me 66355 66355 66355 66355
vailable devices: Name P4CGX150DF27C7 P4CGX150DF27C8 P4CGX150DF27I7 P4CGX150DF27I7AF P4CGX150DF31C7	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 149760 149760 149760 149760 149760	Total I/Os 426 426 426 426 426 508	GPIOs 382 382 382 382 382 382 464	GXB Trai 8 8 8 8 8 8 8 8 8	nsmitter Channel PMA	GXB Receiver Channel PMA 8 8 8 8 8 8 8 8 8 8	PCIe Hard IP Blocks 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Me 66355 66355 66355 66355 66355
vailable devices: Name P4GGX150DF27C7 P4GGX150DF27C8 P4GGX150DF2717 P4GGX150DF2717AF P4GGX150DF31C7	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 149760 149760 149760 149760 149760 149760	Total I/Os 426 426 426 426 508	GPIOs 382 382 382 382 382 382 464 464	GXB Trai 8 8 8 8 8 8 8 8 8 8 8	nsmitter Channel PMA	GXB Receiver Channel PMA 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PCIe Hard IP Blocks 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Me 66355 66355 66355 66355 66355
Name P4CGX150DF27C7 P4CGX150DF27C8 P4CGX150DF27C8 P4CGX150DF2717AF P4CGX150DF217AF P4CGX150DF217AF P4CGX150DF217AF P4CGX150DF217AF P4CGX150DF217AF P4CGX150DF217AF P4CGX150DF217AF P4CGX150DF217AF	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 149760 149760 149760 149760 149760 149760 149760	Total I/Os 426 426 426 426 508 508 508	GPIOs 382 382 382 382 382 382 464 464 464	GXB Trai 8 8 8 8 8 8 8 8 8 8 8	nsmitter Channel PMA	GXB Receiver Channel PMA 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PCIe Hard IP Blocks 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Me 66355 66355 66355 66355 66355 66355 66355
Name PAGGX1500F27C7 EP4CGX1500F27C7 EP4CGX1500F27C7 EP4CGX1500F27C7 EP4CGX1500F27C7 EP4CGX1500F27C7 EP4CGX1500F31C7 EP4CGX1500F31C7 EP4CGX1500F31C8 EP4CGX1500F31C7 EP4CGX1500F31C7 EP4CGX1500F31C7	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 149760 149760 149760 149760 149760 149760 149760 149760	Total I/Os 426 426 426 508 508 508	GPIOs 382 382 382 382 382 464 464 464 464	GXB Trai 8 8 8 8 8 8 8 8 8 8 8 8	nsmitter Channel PMA	GXB Receiver Channel PMA 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PCIe Hard IP Blocks 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Me 66355 66355 66355 66355 66355 66355 66355 66355 66355 66355 66355

Figure 1-3 New projects wizard Family, Device, & Board Settings

3.) Hit Next \rightarrow finish and it will create a new project as in Figure 1-4.



Figure 1-4 The new project has been created

4.) Now we will create our system with the Platform Designer tool (Formally known as Qsys). To do this choose Tools → Platform Designer as shown in figure 1-5. Platform Designer already creates a new system for you, but you can start a new system manually if you want. Choose File → New System to do that.



Figure 1-5 Opening Platform Designer for creating system.

5.) After opening the new system, we want to save it under whatever name we choose as in Figure 1-6 and 1-7.

Platform Designer - unsaved.qsys* (C:\Nios_Tu	torial\unsa\	/ed.qsys)				-	0 ×
File Edit System Generate View Tools Help							
New System	Ctrl+N	nts 💠 Address Map 🖄	Interconnect Requirements				- d' 0
New Component		System: unsaved					
Open	Ctrl+O	n Name	Description	Export	Clock	Base	
Save	Ctrl+S	⊡ clk_0	Clock Source				
Save As		D- dk_in	Clock Input	clk	exported		
Refresh System	F5	D- dk_in_reset	Reset Input Clock Output	reset	dk 0		
Export System as Platform Designer script (.tc)	dk reset	Reset Output	Double-click to export	CK_0		
Export System as hw.tcl Component							
Browse Project Directory							
Recent Projects	>						
Exit	Alt+F4						
New Edt Add Her. Device F ⊠ _ d ⊂ → Ck → Ck 	< ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰	Current filter: ath					- 5 0
0 Errors, 0 Warnings						Generate HDL	Finish

Figure 1-6 Saving new system under our preferred name.

5	Platform Designer -	unsaved.qsys*	(C:\Nios_	Tutorial\unsaved.	qsys)
---	---------------------	---------------	-----------	-------------------	-------

File Edit System Generate View Tools Help

📑 IP Catalog 🛛 🗕 🗗 🗖	System Cont	ents 🛛 Addr	ress Map 🛛	Interconnect Requirements	8				-	- d' 🗆
	X A U	System: unsa	ved							
Project	+ Use Cor	nn Name		Description		Export		Clock	Base	
New Component	5	□ clk_0		Clock Source						
Library	×	⊡– dk_in		Clock Input		clk		exported		
				Darach Tarach			×			
	in Save: unsa	ved					^ orl	clk_0		
Low Power Memory Interfaces and Controllers	Save	in: 📙 Nios_Tuto	orial		🗸 🤌 🖻	• 🔝 🏓	ort			
Processors and Peripherals		.qsys_edi	t							
🗄 - Qsys Interconnect	C .	db								
Iniversity Program	Recent Items									
New Edit + Add	Desktop									
🧏 Hier; 🛛 Device F 🖾 💶 🗗 🗖										
	Documents									
im → ck										
i eset										
	This PC									
				-		_				>
		File name:	Nios_System	qsys			Save			
	L Network	Files of type:	Platform Des	igner System Files (*.qsys)		~	Cancel			
									-	
	Type F	Path Me	ssage							
0 Errora 0 Warajaa	L								Conorata HDI	Finish
o cirors, o warnings									Generate hours	THIST

Figure 1-7 Save new system under "Nios_System.qsys"

6.) Now we will start adding different components to our system to get everything connected. As you can see, the system already adds a clock for us to start out with. The first thing we are going to add is our Nios II processor. To do this we will go to Library →Processors and Peripherals→Embedded Processor→Nios II Processor as shown in Figure 1-8. Alternatively, you can search for different components.

Platform Designer - Nios_System.qsys (C:\N	Nios_Tutorial\	Nios_System.qs	sys)				_	
📑 IP Catalog 🛞 🗕 📑 🗖	System	Contents 🛛	Address Map	Interconnect Requirements				- 5 0
	Ξ.	System	1: Nios_System					
Project	+ Use	Conn Nam	e	Description	Export	Clock	Base	End
New Component	1 5		lk_0	Clock Source				
	$ \times $	D-	dk_in	Clock Input	clk	exported		
Library Basic Functions			clk_in_reset	Reset Input	reset			
H-DSP	·	×	clk	Clock Output	Double-click to export	clk_0		
		×<	clk_reset	Reset Output	Double-click to export			
-Low Power								
Memory Interfaces and Controllers								
Processors and Peripherals	≖							
E Embedded Processors								
Nios II (Classic) Processor								
Nios II Processor								
Hard Processor Components								
Hard Processor Systems								
Inter-Process Communication Perinberals								
Osvs Interconnect								
. University Program								
New Edit + Add								
🕫 Hierar 🛞 Device Far 🛞 🔔 🗗 🗖) <							>
Nios System [Nios System.qsys]	nw f	t 🔫 🛒 Cur	rent filter:					
erenter en	X= Messar	105 X						
teset teset	0= 1103305							
	Туре	Path	Message					
0 Errors, 0 Warnings							Generate HDL	Finish



	locumentation
	Nocumentation
Show simple Main Vectors Caches and Memory Interfaces Arithmetic Instructions MMU and MPU Settings JTAG Debug Advanced Features	
Show sinnals Vectors Caches and Memory Interfaces Arithmetic Instructions MMU and MPU settings JTAG Debug Advanced reatures	
pige2_nan2_0 Vois II Core : O Mass Tig	
INOS2_genta_0 Otexatis	
ck data r	
reset avator instruction_rr Nios II/e Nios II/f	
rg ntempt rest debug_rest_re Summary Resource-optimized 32-bit RISC Performance-optimized 32-bit RISC	
debug_mem_slave svake niss_sustem_instruction_rt Features JTAG Debug JTAG Debug JTAG Debug	
alter_nixt ECC KAM Protection Hardware Putiply/Uvide Instruction/Data Caches	
Tightly-Coupled Masters ECC RAM Protection	
External Interrupt Controller Shadow Register Sets	
RAM Usage 2 + Dottons 2 + Options	
- I	
W Pror: mios2_gen2_0. Instruction Cache is larger than the Instruction Address. Please reduce the Instruction Cache Size. Current Tag Size is 0 M Pror: mios2_gen2_0. Please tables is not specified. Please sheet that reset save	
C Error: nios2_gen2_0: Exception slave is not specified. Please select the exception slave	
Can	E Finish

Figure 1-9 We choose Nios II/F and finish to add processor to system.

7.) You can rename system components if you would like, but that won't have any impact on the system or its function. Now we will connect the Nios processor to the clk. We will **connect clk and reset** as shown in figure 1-10. These are connected by clicking the hollow dots. When the dots become solid, it means there is a connection.

Platform Designer - Nios_System.qsys* (C:\ File Edit System Generate View Tools Help	Nios_T	utorial\N	Nios_System.qsys)						_	\times
💾 IP Catalog 🐹 🗕 🗗 🗖	1	System C	Contents 🖾 Add	lress Map	8 Interconnect Re	quirements 🛛			-	- d' 🗆
		X	System: Nios	_System I	Path: nios2_gen2.rese	:				
Project	+	Use	Connections	Name		Description	Export	Clock	Base	
Mew Component System Library B-Basic Functions G-DSP	× •		~		k_0 dk_in dk_in_reset dk	Clock Source Clock Input Reset Input Clock Output	cik reset Double-click to export	<i>exported</i>	,	
-Interface Protocols -Low Power -Memory Interfaces and Controllers -Processors and Peripherals	▲ ▼ ▼				incs2_gen2 lk reset	Nios II Processor Clock Input Reset Input	Double-click to export Double-click to export Double-click to export	clk_0 [dk]		
Co-Processors Co-Processors Co-Processors Nos II (Classic) Processor Components Hard Processor Components Hard Processor Components Hard Processor Communication Peripherals Okys Interconnect Tri-State Components University Program			,		data_master nstruction_master rq debug_reset_request debug_mem_slave custom_instruction_m	Avalon Memory Mapped Master Avalon Memory Mapped Master Interrupt Receiver Reset Output Avalon Memory Mapped Slave Custom Instruction Master	Double-click to export Double-click to export Double-click to export Double-click to export Double-click to export Double-click to export	[ck] [ck] [ck] [ck] [ck]	⊮ 0x0800	
Hierar ⊠ Device Far ⊠ _ □ □		< ተ	🔻 🝸 Current f	filter:						>
in - set in - s∎ ck 0		Message	s 🖾						-	- d' =
	Тур	e	Path 3 Errors		Message					
		3	Nios_System.nio	s2_gen2	Instruction Cache is la	arger than the Instruction Address. P	ease reduce the Instruction Cache	Size. Curren	t Tag Size is 0	
G → debug_reset_request debug_reset_request debug_reset_request instruction_master debug_reset reset	6	3	Nios_System.nio	s2_gen2 s2_gen2	Exception slave is not	specified. Please select the exception	n slave			
< >>	<						-			>
3 Errors, 0 Warnings									Generate HDL	Finish

Figure 1-10 Connecting clk and reset

8.) Next thing we will add to our system is the JTAG UART. To do this go to Library → Interface Protocols → Serial → JTAG UART as shown in Figure 1-11 and 1-12.

1.0000		System (Contents 🛛 Address	Map 🙁 Interconnect Re	equirements 🙁			-
XX		lice	Connections	Nama	Description	Evport	Clark	Pace
New Component		⊡ ⊡	Connections		Clock Source	Export	Clock	Dase
⊞-System	X		D-	dk_in	Clock Input	clk	exported	
Ibrary	1 😨		○ D	dk_in_reset	Reset Input	reset		
H-DSP	11°-		<	dk	Clock Output	Double-click to export	dk_0	
Interface Protocols				dk_reset	Reset Output	Double-click to export		
Audio & Video			í l	🗆 🖳 nios2_gen2	Nios II Processor			
Ethernet			$ \bullet \rightarrow$	dk	Clock Input	Double-click to export	clk_0	
iji - JESD	I I			reset	Reset Input	Double-click to export	[dk]	
PCI Express				data_master	Avalon Memory Mapped Master	Double-click to export	[dk]	
RapidIO				instruction_master	Avaion Memory Mapped Master	Double-click to export	[dk]	
- Serial			$ \times \longrightarrow$	irq	Interrupt Receiver	Double-click to export	[dk]	
Altera Avalon I2C (Master)				debug_reset_request	Reset Output	Double-click to export	[dk]	
Avalon-ST Serial Perioheral			∣ ♦♦>	debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[dk]	⊕ 0x0800
SerialLite Transceiver PHY Transceiver PHL Low Power Memory Interfaces and Controllers								
ew Edit								
-Processors and Perinherals w Edit Hierar Device Far		<						
Processors and Peripherals ex Edit Herar Device Par C C C C C C C C C C C C C C C C C C C		< ft wn	🕂 🍸 🍸 Current filter	r.		_		
Processors and Pernherals > ew Edt • Add Herar 20 Device Far 21 - 10* • dk • • • ck • • • ck • •		≺ n¦~ fit Message	S ⊠ Current filter	n				
Pronessors and Perchange internation		< n ~ f t Message e	· ▼ 🗑 Current filter s 🛛 Path	r: Message				-
Princessors and Perchanks ▲ term: Edit ▲ Herar ID Device Far ID - dk - C - mode - C		<mark>د اسم بالار الم</mark>	es and a second	r: Message		-		
Processors and Perchanals iew Edit Processors and Perchanals iew Edit Processors Add Processors Add Processors Add Processors Add Add Processors Add Add Add Add		< ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ Message e 3	Current filter	r: Message gen2 Instruction Cache is	larger than the Instruction Address. PA	ase reduce the Instruction Cache	Size, Current Ti	– ag Size is 0
Processors and Perchange → texm. Edit ▲ Add Herers IS Device For IS _ dk - C _ dck - C _ dck - C _ more scalar _ C _ → ctatom_instruction_matter _ → → debug_men_slave _ → → debug_men_slave _ →		< n ~ f t Message e 3	Current filter Current filter Souther Path Gerors Nios_System.nios2_ Nios_System.nios2_	r: Message gen2 Instruction Cache is gen2 Reset slave is not sp	arger than the Instruction Address. PA eofied, Please select the reset slave	ase reduce the Instruction Cache	Size. Current Ta	ag Size is 0
Processors and Perioherals New Edit Hierar SS Device Far SS → ck → reset ⊉ dis_0 ∰ most_gen2 ⊕ → ck		< ۲۰۰٫۰۰۰ sft Message e	Current filter s Path 3Errors	r: Message		-		



💑 JTAG UART - jtag_uart_0	×
JTAG UART altera_avalon_itag_uart	Documentation
Block Diagram	
Show signals	
jtag_uart_0	
Construct using registers instead of memory blocks	
clock interrupt Read FIFO (Data from JTAG to Avalon)	
reset Buffer depth (bytes): 64 🗸	
avalon_itag_slave avalon IRQ threshold: 8	
altera_avalon_jtag_uart	
Warning: itag wart 0: ITAG HART IP input clock need to be at least double (2x) the operating frequency of ITAG TCK op board	
Terring, jeug_eurc_s, sine once an input dock need to be belease double (2x) are opending inequality of STAG Tex of Dobio	
	Cancel Finish

Figure 1-12 Adding JTAG UART.

9.) We will now connect the clk, reset, and data master as shown in Figure 1-13.



Figure 1-13 Connecting JTAG to system

10.) Next, we will add the system memory through the on chip memory. Later we will change this to SDRAM. To add the on chip memory, choose Library → Basic Functions → On Chip Memory → On-Chip Memory (RAM or ROM) as shown in Figure 1-14 and 1-15.



Figure 1-14 Adding On-Chip Memory

On-Chip Memory (RAM or ROM) - onchip_memory2_0	
--	--

On-Chip Mem altera_avalon_onchip_	ory (RAM or ROM) memory2	Documentation
Block Diagram		^
Show signals	Memory type Type: RAM (Writable) ~ Dual-port access Single dock operation Read During Write Mode: DONT_CARE ~ Block type: AUTO ~	
	Minimize memory block usage (may impact finex) Read latency Slave s1 Latency: 1 Slave s2 Latency: 1 Rom/RAM Memory Protection Reset Request:	
	ECC Parameter Extend the data width to support ECC bits: Disabled	
	Memory initialization Initialize memory content Enable non-default initialization file Type the filename (e.g: my_ram.hex) or select the hex file using the file browser button. User created initialization file: Onchip_mem.hex Enable Partial Reconfiguration Initialization Mode	

Figure 1-15 Change memory size to 204800 and select finish

11.) Now we will connect the memory to the rest of the system by selecting the clk, reset, data master, and instruction master hollow dots as shown in Figure 1-16.

 \times

Platform Designer - Nios_System.qsys* (C:\Nios_Tutorial\Nios_System.qsys)

- 🗆 🗙

File Edit System Generate View Tools Help



Figure 1-16 Connecting memory to system

12.) We will now connect the Nios CPU to the memory. To do this we will click on the Nios2_gen2 component to open its settings. Then we will go to Vectors and change the Reset Vector and Exception Vector to the On-Chip memory as shown in Figure 1-17 and 1-18.

Platform Designer - Nios_System.qsys* (C:\Nios_Tutorial\Nios_System.qsys)

File Edit System Generate View Tools Help



Figure 1-17 Changing processor to use on-chip memory.

– 🗆 🗙



Figure 1-18 Changed Reset and Exception Vector

13.) Next up is to add the System ID to the system. Choose Library → Basic Functions → Simulation;
 Debug and Verification → Debug and Performance → System ID Peripheral as shown in Figure 1-19 and 1-20.

Platform Designer - Nios_System.qsys* (C:\	Nios_Tutorial\I	Nios_System.qsys)	- 0	\times
File Edit System Generate View Tools Help P CatalogCC Basic Functions Antimetic Antimetic Antimetic Conday PLLs and Resets Conday and Asptors Conday and Asptors Conday rules and Adaptors Conday and Asptors Conday rules and Asptors Alters Synstem Source State McControler State		Addre: Si Intera Si - C - Connections Name Connections Name Connections Name Connections Name Connections Name Connections Name Connections Construction Connections Construction Connections Construction	Parameters System: Nios_System Path:: nios2_gen2 Nios II Processor adtera_nios2_gen2 Main Vectors Caches and Memory Interfaces Arithmetic Instructions MMU and MPU Settings JTAG Debu Reset Vector Reset Vector Reset Vector offset: Dx00000000 Exception Vector Exception Vector Exception vector offset: Dx0000020 Fast TLB Miss Exception Vector Fast TLB Miss Exception vector offset: Dx0000000 Fast TLB Miss Exception vector offset: Dx0000000 Fast TLB Miss Exception vector Fast TLB Miss Exception vector: Dx0000000 Fast TLB Miss Exception vector: Dx00000000 Fast TLB Miss Exception vector: Dx00000000	
Image: Hierar ≥ Device Far ≥ Image: Hierar	< nh th	💙 📡 Current filter:	٢	> [×]
b ck custom_instruction_master data_master debug_mem_slave debug_reset_request debug_reset_request instruction_master	Type	Path S Errors Nios_System.nios2_gen2.data_maste Nios_System.nios2_gen2.instruction_ Nios_System.nios2_gen2.instruction	Message onchip_memory2.s1 (0x00x3fff) overlaps jtag_uart.avalon_itag_slave (0x00x7) rr nios2_gen2.debug_mem_slave (0x8000xff) overlaps onchip_memory2.s1 (0x00x3fff) master jtag_uart.avalon_itag_slave (0x00x7) overlaps onchip_memory2.s1 (0x00x7) master onchip_memory2.s1 (0x00x7) overlaps itag_uart.avalon_itag_slave (0x00x7)	
5 Errors, 1 Warning	<		Generate HDL	> Y

Figure 1-19 Adding system ID

BIOCK Diadram	Parameters	
Show signals	32 bit System ID: 0x0000000	
aurit neur 0	▼ Description	
ck olock reset reset control_slave avalon altera_avalon_systd_c	yy	

Figure 1-20

14.) Connect the clk, reset, and data master dots as shown in figure 1-21.



Figure 1-21 Connecting System ID connections

15.) Now we will add the SDRAM to the system which will be used instead of the on-chip memory. To do this choose Library → Memory Interfaces and ControleIrs → SDRAM → Controller as shown in Figure 1-22 and 1-23. Platform Designer - Nios_System.qsys* (C:\Nios_Tutorial\Nios_System.qsys)

– 🗆 🗙

File Edit System Generate View Tools Help 💶 🗗 🗖 📜 System Contents 🙁 Address Map 🙁 Interconnect Requirements 🙁 📂 IP Catalog 🛛 🕅 - d 🗆 🛎 🔺 🌉 System: Nios_System Path: sysid_qsys.control_slave × 🔯 0 Project Wew Component... P-System ٠ Use Connections Name Description Export Clock Base 1 _____ Reset Input dk_in_reset reset × Clock Output clk 0 dk Library -Basic Functions -DSP . dk_reset Reset Output \checkmark 🗆 🛄 nios2_gen2 Nios II Processor × Clock Input Double-click to exp clk clk_0 • Low Power
 Memory Interfaces and Controllers reset Reset Input [clk] Ŧ Avalon Memory Mapped Master Double-click to export [clk] data_master ---- Arria 10 × instruction_master Avalon Memory Mapped Master Double-click to export [clk] . Evi irg Interrupt Receiver [clk] debug_reset_request Reset Output Double-click to export [dk] + Flash debug_mem_slave Avalon Memory Mapped Slave [dk] 0x0000_0800 Memory Interfaces with ALTMEMPHY
 Memory Interfaces with UniPHY
 SDRAM . Double-click to expor Custom Instruction Master custom_instruction_m \checkmark 🖃 jtag_uart JTAG UART Altera SDS SDRAM Tri-State Cont dk Clock Input clk_0 ntroller Reset Input [clk] reset Processors and Peripherals avalon_jtag_slave Avalon Memory Mapped Slave [clk] 0x0000_0000 Qsys Interconnect
 Tri-State Components
 University Program Interrupt Sender [clk] irg \checkmark 🗆 onchip ry2 On-Chip Memory (RAM or ROM) clk 0 dk1 Clock Input Double-click to expor Double-click to export [clk1] **s**1 Avalon Memory Mapped Slave 0x0000_0000 reset1 Reset Input Double-click to export [dk1] < \checkmark 🗆 sysid_qsys System ID Peripheral Clock Input clk_0 dk New... Edit... 🕂 Add... reset Reset Input [clk] 👫 Hierar 🛛 Device Far 🕮 - d' 🗆 🗛 🏦 🔫 🛒 Current filter: 🗄 🚛 clock_bridge 🗄 🚛 cpu Connections Xalanda Messages ∞ - d 🗆 Path 8 Туре Message endenp_nen
endenp_n 7 Errors ^ 8 Nios_System.nios2_gen2.data_master onchip_memory2.s1 (0x0..0x3ffff) overlaps jtag_uart.avalon_jtag_slave (0x0..0x7) 8 Nios_System.nios2_gen2.data_master sysid_qsys.control_slave (0x0..0x7) overlaps onchip_memory2.s1 (0x0..0x3ffff) ⇒ sysid_qsys → → ck → → control_ → → reset 8 Nios_System.nios2_gen2.data_master onchip_memory2.s1 (0x0..0x3ffff) overlaps sysid_qsys.control_slave (0x0..0x7) 8 Nios_System.nios2_gen2.data_master iios2_gen2.debug_mem_slave (0x800..0xfff) overlaps onchip_memory2.s1 (0x0..0x3ffff < > 7 Errors, 1 Warning Generate HDL... Finish

Figure 1-22 Adding SDRAM to System

SDRAM Controller - new_sdram_c	ontroller_0	×
SDRAM Controlle altera_avalon_new_sdram_c	r ontroller	Documentation
Block Diagram Show signals new_sdram_controller_0 clk elock reset rese	Memory Profile Timing • Data Width Bits: 32 v • Architecture Chip select: 1 v Banks: 4 v • Address Width Row: 12 Column: 8 • Generic Memory model (simulation only) Include a functional memory model in the system testbench Memory Size 16 MBytes 128 MBits 128 MBits	Presets Project Cick New to create a preset. Ubrary Four SDR 100 8MByte x16 chips Micron MT8LSDT1664HG module single Aliance AS4LC2MISD 10 chip single Micron MT48LC2M3282 7 chip single Micron MT48LC4M3282 7 chip single Micron MT48LC4M3282 7 chip single NEC D4564163 A80 chip 64Mb x 16 Apply Update Delete New in the future release.
		Cancel Finish

Figure 1-23

16.) Now we will connect the SDRAM to the system by connecting the clk, reset, and data master wires as shown in Figure 1-24.

IP Catalog 🛛 _ 🗗 🗖	Sys 🗉	tem Cont	ents 🛛	Address Map 🛛 Interconnect R	Requirements 🛛			- 1
vroject	+ Us	se Cor	nections	Name	Description	Export	Clock	Base
New Component	1 18		↓	♦ reset	Reset Input	Double-click to export	[dk]	
. System	×			data_master	Avalon Memory Mapped Master	Double-click to export	[dk]	
ibrary	1 😨			instruction master	Avalon Memory Mapped Master	Double-click to export	[dk]	
+-sasc Functions				- ira	Interrunt Receiver	Double-click to export	[dk]	
- USP	II 🔺			debug reset reg	est Reset Output	Double-click to export	[dk]	
- Interface Protocols	A			debug mem daur	Auplee Memory Manned Slave	Double click to export	Calu	- 0-0000
-Low Power			ITT	debug_inem_slave	Avaion Menory Mapped Slave	Double-click to export	[Cik]	= 0x0000
Arria 10 External Memory Interface	-			custom_instruction	n_m Custom Instruction Master	Double-click to export		
Evternal Memory Interfaces Debug	- E			🗄 jtag_uart	JIAGUARI			
 Strativ 10 External Memory Interfa 		•		dk dk	Clock Input	Double-click to export	clk_0	
B-Flash			• • • •	♦ reset	Reset Input	Double-click to export	[dk]	
R. Memory Interfaces with ALTMEMPHY			++	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x0000
Memory Interfaces with UnPHY				irq	Interrupt Sender	Double-click to export	[dk]	
B-SDRAM	F	A		onchip_memory	2 On-Chip Memory (RAM or ROM)			
Altera SDRAM Tri-State Contro		- +		dk1	Clock Input	Double-click to export	clk 0	
SDRAM Controller				e1	Avalop Memory Mapped Slave	Double-click to export	[dk1]	- 0×0000
Processors and Peripherals				repet1	Paret Innut	Double-click to export	[clk1]	- 040000
Osys Interconnect			T		Custom ID Deviational	Double click to export	[OR A]	
Tri-State Components				E sysia_dsys	System 10 Periprieral			
University Program		_ I † _		dk dk	Clock Input	Double-click to export	clk_0	
				♦ reset	Reset Input	Double-click to export	[dk]	
			1 † ?	control_slave	Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x0000
				new_sdram_con	troll SDRAM Controller			
>		- I+-		dk dk	Clock Input	Double-click to export	clk_0	
			•		Reset Input	Double-click to export	[dk]	
lew Edit 🛉 Add				s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	
				· wire	Conduit	Double-click to export		
	1						1	1
new_sdram_controller	M^	* #1t - *	Curr 🗮	ent filter:				
a 🕨 dk								
i∋- 🕨 reset	6E Mes	ssages	23					-
₽-∎- s1	T.000		a the		Masanas			
Ð- ► wire	Type		aui		Message			
L nios2_gen2	=83	10	Errors					
⊕-∎-dk	8	Ni	os Systen	n.nios2 gen2.data master	onchip memory2.s1 (0x00x3ffff) ov	erlaps itag uart.avalon itag s	slave (0x0	.0x7)
custom_instruction_master	l ă		or Surton		cursid acurs control, clause (0x0, 0x7)	undans enchin memory3 c1 //		•
u − data_master			us_system	imosz_genz.uatd_Mdster	sysia_qsysicontrol_slave (0x00x7) (overlaps onchip_memory2.51 (5x00x51111	
debug_mem_slave	8	Ni	os_Systen	n.nios2_gen2.data_master	new_sdram_controller.s1 (0x00xfff	fff) overlaps sysid_qsys.control	_slave (0x)	00x7)
ebug_reset_request	8	Ni	os_Systen	1.nios2_gen2.data_master	onchip_memory2.s1 (0x00x3ffff) ov	erlaps new_sdram_controller.s	1 (0x00xf	TTTTT)
m - insulucion_master v							100	

Figure 1-24 Connecting SDRAM

17.) To change the Nios system to use the SDRAM instead of the on-chip memory, we will connect the SDRAM to the instruction master and disconnect the On-Chip memory from the instruction master as shown in Figure 1-25. We will then go to the CPU setting by clicking on the Nios component and changing the Reset and Exception Vectors to the SDRAM shown in Figure 1-26.



Figure 1-25 Connecting SDRAM to instruction master



Figure 1-26 Changing CPU to use SDRAM

18.) We will now start connecting our PIO's to send data to and from the CPU to the FPGA. Choose Library → Processors and Peripherals → PIO (Parallel I/O) as shown in figure 1-27 and 1-28. We will be using 32 bit In Out PIO's for what we will be doing. Choose accordingly depending on your use of the PIO.

IP Catalog 🛛 🗕 🗗 🗖	100	System C	Contents 🛛 Address N	Nap 🖾 Interconnect Re	equirements 🛛			- 0
X 🕸		A	System: Nios_System:	em Path: nios2_gen2				
oject ^		Use	Connections	Name	Description	Export	Clock	Base
New Component			+	→ reset	Reset Input	Double-click to export	[dk]	
System	$ \times $			✓ data_master	Avalon Memory Mapped Master	Double-click to export	[dk]	
rary				→ instruction master	Avalon Memory Mapped Master	Double-click to export	[dk]	
Dep				→ ira	Interrupt Receiver	Double-click to export	[dk]	
Interface Bratacele	II 🔺 I			✓ debug reset reque	st Reset Output	Double-click to export	[clk]	
and Protocols				debug mem dave	Avalon Memory Manned Slave	Double-click to export	C-IV1	- 0~0000
Low Power	- I			debug_itett_stave	Avalor Mentory Mapped Slave	Double-click to export	LON	= 0x0000_
Processors and Paripherals			^	custom_insedication_	The custom instruction master	Double-click to export		
-Co-Processors	-			🖃 jtag_uart	JIAGUARI			
E-Embedded Processors			•	→ dk	Clock Input	Double-click to export	clk_0	
Hard Processor Components				→ reset	Reset Input	Double-click to export	[clk]	
Hard Processor Systems				→ avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x0000_
				≺ irg	Interrupt Sender	Double-click to export	[dk]	
Peripherals				onchip_memory2	On-Chip Memory (RAM or ROM)			
Altera Avalon LCD 16207		_	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	→ dk1	Clock Input	Double-click to export	dk 0	
Altera I2C Slave To Avalon				→ e1	Avalop Memory Manped Slave	Double-click to export	Telk 11	- 0*0000
 Altera Modular ADC core 				- sacoti	Boost Input	Double click to export	[elk:1]	_ 0x0000
• Altera Modular Dual ADC co				/ resett	Custom ID Devictored	Double-click to export	[UK1]	
 Interval Timer 				🖻 sysia_qsys	System ID Peripheral			
Lauterbach Trace Interface			• · · · · · · · · · · · · · · · · · · ·	→ clk	Clock Input	Double-click to export	clk_0	
PIO (Parallel I/O)				→ reset	Reset Input	Double-click to export	[dk]	
 Pixel Converter (BGR0> E 				→ control_slave	Avalon Memory Mapped Slave	Double-click to export	[dk]	
 SPI Slave to Avalon Master 		\checkmark		new_sdram_continues	roll SDRAM Controller			
>			•	→ dk	Clock Input	Double-click to export	clk_0	
			• • • •	→ reset	Reset Input	Double-click to export	[dk]	
w Edit 📥 Add				→ e1	Avalon Memory Manned Slave	Double-click to export	[cite]	- 0×0000
				· 31	Creduit	Double click to export	Low	- 040000
				wre	Conduit	Double-click to export		
ierar 💥 Device Far 💥 🔔 📑 🗖		<						
nios2_gen2		r⊷ ft	🔫 🛒 Current filter:					
⊢∎– clk ⊢−∎ custom instruction master	X	Messages	s 🛛					- 1
- data_master			D-#		N			
debug_mem_slave	Туре	e	Paul		message			
-debug_reset_request			9 Errors					
instruction_master	E	3	Nios System.nios2 g	en2.data master	onchip memory2.s1 (0x00x3ffff) over	alaps itag uart.avalon itag s	lave (0x0	0x7)
⊢ −∎ irq			Nios System.nios2 a	en2.data_master	sysid asys control slave (0v0_0v7)	verlans onchin memory2 <1 (0x00x3ffff	
					-,,			
- E cou		5	Nios_System.nios2_g	en2.data_master	new_sdram_controller.s1 (0x00xfff	ft) overlaps sysid_qsys.control	_slave (0x0	J0x7)
- Er opu	6	3	Nios_System.nios2_g	en2.data_master	nios2_gen2.debug_mem_slave (0x80	00xfff) overlaps new_sdram_	controller.	s1 (0x00xffffff)
v	<	•	1					



PIO (Parallel I/O) - pio_0	
PIO (Parallel I/O) altera_avalon_pio	Documentation
Block Diagram Block Diagram Cate Cate Cate Cate Cate Cate Cate Cate	
	Sign Yorkronously capture Sidge Type: Sign York Consult Capture register Tatempt Generate IRQ IRQ Type: EVELow EVELow EVELow Evel Interrupt CPU when any unmasked b/to in its logic true Edge: Interrupt CPU when any unmasked b/to in the edge-capture register is logic true. Available when synchronous capture is enabled
	Test bench wiring Test bench Store PIO inputs in test bench Drive inputs to field:
Info: pio_0: PIO inputs are not hardwired in test bench. Undefine	Ld values will be read from PIO inputs during smulation.
	Cancel Finish

Figure 1-28 Adding 32 bit inout PIO

19.) We will connect the PIO clk, reset, and data master as shown in Figure 1-29. Also we will export the PIO shown in Figure 1-30.



Figure 1-29 Connecting PIO connections

Platform Designer - Nios_System.qsys* (C:\Nios_Tutorial\Nios_System.qsys)

– 🗆 🗙

File Edit System Generate View Tools Help

💾 IP Catalog 🛞 🗕 🖬 🗖		System (Contents 🛛 Address Ma	p 🛛 Interconnect Re	quirements 🛛			-	-
		∞ ▲	System: Nios_System	Path: Line 1. external_c	onnection				
Project	+	Use	Connections	Name	Description	Export	Clock	Base	
New Component	🛸		$ \downarrow \downarrow \downarrow \downarrow \downarrow \rightarrow$	debug mem slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	- 0x00	00 0: ^
System	×			custom instruction	m Custom Instruction Master	Double-click to export			-
Library	🛃			☐ jtag_uart	JTAG UART				
Basic Functions			\bullet	dk	Clock Input	Double-click to export	clk 0		
Interface Protocols	🌥		$ \downarrow \downarrow$	reset	Reset Input	Double-click to export	[clk]		
+ Low Power	🔺		$ \bullet \bullet \to \to$	avalon itag slave	Avalon Memory Mapped Slave	Double-click to export	[clk]		00 01
Memory Interfaces and Controllers	▼			irg	Interrupt Sender	Double-click to export	[clk]		-
Processors and Peripherals	I			onchip memory2	On-Chip Memory (RAM or ROM)				
Co-Processors	11		\bullet	ck1	Clock Input	Double-click to export	cik 0		
Embedded Processors	11			s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	- 0x00	00 01
Hard Processor Components	11			reset1	Reset Input	Double-click to export	[clk1]		
Hard Processor Systems	11			E sysid asys	System ID Peripheral		[
Decipherals	11		\bullet	dk	Clock Input	Double-click to export	clk 0		
Altera Avalon I CD 16207	11			reset	Reset Input	Double-click to export	[clk]		
Altera I2C Slave To Avalon	11			control slave	Avalon Memory Manned Slave	Double-click to export	[clk]	- 0×00	00.0
Altera Modular ADC core	11			E new sdram contro	oll SDRAM Controller	Double click to export	fend	- 0400	
····· • Altera Modular Dual ADC co	11		\bullet	dk	Clock Input	Double-click to export	clk 0		
 Interval Timer 	11			reset	Reset Input	Double-click to export	[clk]		
Lauterbach Trace Interface	11			st	Avalon Memory Manned Slave	Double-click to export	[clk]	- 0×00	00.0
PIO (Parallel I/O)	11			wire	Conduit	Double-click to export	Lend	- 0400	
Pixel Converter (BGR0> E	11			E line1	PIO (Parallel I/O)	Double click to export			
SPI Slave to Avaion Master	11			dk	Clock Input	Double-click to export	clk 0		
	11			recet	Recet Input	Double-click to export	CIR_0		
New Edit + Add	11			el	Avalon Memory Manned Slave	Double-click to export	(Calk)	- 0~00	
	11			external connection	Conduit	inel external connection		= 0x00	JO_0
12 Martin Car Sec. 2	il -	<		external_connection)		· ·
All Hieran 23 Device Far 23 _ = = "							r -		-
E-= Line1	11	n~ ft	👻 📉 Current filter:						
in → dk	×=	Message	• 53						
B- reset	0=	message	3 ~ L						
m ■ s1	📗 тур	be .	Path		Message				8
⊕- ⊕ ck 0	llag		11 Errors						^
⊟∎ jtag_uart			No. Coston de la cost	Didate and the		dense data an esta a constante da da da da		070	
		~	Nios_System.niosz_ger	iz.data_master	onchip_memory2.s1 (0x00x3fff) ove	haps jtag_uarcavaion_jtag_si	ave (uxu.	.0x7)	
i ∎− ∎− ck		×	Nios_System.nios2_ger	12.data_master	sysid_qsys.control_slave (0x00x7) o	verlaps onchip_memory2.s1 (0>	00x3ffff	9	
🗈 🖿 irq	(×	Nios_System.nios2_ger	n2.data_master	new_sdram_controller.s1 (0x00xffff	f) overlaps sysid_qsys.control_	slave (0x	00x7)	
		×	Nios_System.nios2_ger	12.data_master	Line1.s1 (0x00xf) overlaps new_sdran	_controller.s1 (0x00xffffff)			
riew_sdram_controller		-		-					
< >>									>
11 Errors, 2 Warnings							Ge	enerate HDL	Finish

Figure 1-30 We will export the PIO

20.) Now we need to assign base addresses to all these components. To do this **choose System** → **Assign Base Addresses** as shown in Figure 1-31.

Platform Designer - Nios_System.qsys* (C:\	Nios_Tutorial\Nios	_System.qsys)					- 🗆 ×
File Edit System Generate View Tools Help	b						
IP C Upgrade IP Cores		s 🛛 Address Ma	ap 💠 Interconnect Rec	quirements 🛛			- d 🗆
Assign Base Addresses		ystem: Nios_System	m Path: Line 1. external_co	onnection			
Projec Assign Interrupt Numbers		tions	Name	Description	Export	Clock	Base
Assign Custom Instruction Opc Sys Create Global Reset Network	odes		debug_mem_slave custom_instruction_r	Avalon Memory Mapped Slave	Double-click to export Double-click to export	[clk]	
Libran Bas Show System With Platform De	signer Interconnect	-	□ jtag_uart	JTAG UART	Double-click to export		
DSP Remove Dangling Connections	-		→ ck → reset	Clock Input Reset Input	Double-click to export Double-click to export	clk_0 [clk]	
Low Import Interface Requirements.			avalon_jtag_slave irg	Avalon Memory Mapped Slave Interrupt Sender	Double-click to export Double-click to export	[clk] [clk]	©_00000_0
Processors and Penpherais	 II ≚ I M I I I		onchip memory2	On-Chip Memory (RAM or ROM)		6G	
Co-Processors			clk1	Clock Input	Double-click to export	clk_0	
Embedded Processors			→ s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	= 0x0000_0
Hand Processor Components		• • ;	reset1	Reset Input	Double-click to export	[clk1]	
Inter-Process Communication			🖻 sysid_qsys	System ID Peripheral			
- Peripherals			→ clk	Clock Input	Double-click to export	clk_0	
 Altera Avalon LCD 16207 			reset	Reset Input	Double-click to export	[clk]	
 Altera I2C Slave To Avalon 		• •	control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
Altera Modular ADC core			new_sdram_control	oll SDRAM Controller			
Altera Modular Dual ADC co			→ clk	Clock Input	Double-click to export	clk_0	
Interval Inter			reset	Reset Input	Double-click to export	[clk]	
PIO (Parallel I/O)			→ s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	i 0x0000_0
 Pixel Converter (BGR0> E 		Ŷ	wire	Conduit	Double-click to export		
SPI Slave to Avalon Master ¥			E Line1	PIO (Parallel I/O) Clock Input	Double-click to export	clk 0	
			reset	Reset Input	Double-click to export	[clk]	
New Edit 🕂 Add		•	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	■ 0x0000_0
			external_connection	Conduit	line1		×
🧏 Hierar 🛛 Device Far 🕮 💶 🗗 🗖	<						>
ie-∎ Line1 ^	🔽 - Jt 🗤	🗑 🗑 Current filter:					
⊕-	X= Messages	x					- 6 0
	Type P	ath		Message			8
i∰ E ck_0	11	Errors					^
⊖⊶ ⊡ ⊧ jtag_uart	🖸 🔀 Nic	s_System.nios2_ge	n2.data_master d	onchip_memory2.s1 (0x00x3ffff) over	erlaps jtag_uart.avalon_jtag_s	lave (0x00)	ĸ7)
avalon_jtag_slave		s System.nios2 ne	n2.data master	sysid asys.control slave (0x00x7)	verlaps onchip memory2.s1 (x00x3ffff	<u> </u>
		s System nice? ce	n2 data master	new sdram controller st (0+0 0+fff	fff) overlage sysid, days control	slave (0v0	0x7)
🗈 🖿 reset		/s_system.mosz_ge	a2 data master	light at (0.0, 0.0 and an a	m, overlaps sysid_qsys.colluro	_siave (0x0	
🖶 🖶 new_sdram_controller 🗸 🗸		vs_system.nios2_ge	nz.uata_master l	Line1.51 (UXUUXT) overlaps new_sdrai	m_concroller.s1 (UXUUXTTTTT)		
< >	<						>
11 Errors, 2 Warnings						Gen	erate HDL Finish

Figure 1-31 Assigning base addresses

21.) Now you see that we only have a couple of warning shown in Figure 1-32. To get rid of those warnings, we will assign Interrupt Numbers shown in Figure 1-33 and 1-34.

Porticipant elements elements elements System Contents Address Map Address Addres	
Image: Control of the second secon	- 6
Project Base Description Export Clock Base Binder Component Base Description Description Double-click to export [k] # Binder Component Base Conduction Conduction Double-click to export [k] # Binder Forbocols Extern Instructions Binder Forbocols Binder Forbocols Double-click to export (k] # Binder Forbocols Binder Forbocols Binder Forbocols Double-click to export (k] # Binder Forbocols Binder Forbocols Binder Forbocols Double-click to export (k] # Binder Forbocols Binder Forbocols Binder Forbocols Double-click to export (k] # Binder Forbocols Binder Forbocols Binder Forbocols Double-click to export (k] # Binder Forbocols Binder Forbocols Binder Forbocols Double-click to export (k] # Binder Forbocols Binder Forbocols Binder Forbocols Double-click to export (k] # Binder Forbocols Binder Forbocols Binder Forbocols Double-	
Image: More Component Image: System Avalan Memory Mapped Slave Double-click to export Clkl Image: Clklk Image: Clkl Image: Clklk Image: Clkl Image: Clklk	
 B) System B)	x0204
Ubbray # Bask Functions # Jtag_uart JTAG UART Double-click to export dk_0 © DP # Deriver dk Cokinput Double-click to export dk_0 © DP New # Memory Interfaces and Controlers First Ack Protocols Double-click to export dk_0 © OP New # Or Processors Boother click to export dk_0 Double-click to export dk_0 @ Co-Processors # Or Chip Memory Mapped Slave Double-click to export dk_0 Double-click to export dk_0 @ Hard Processors # Internot Scoressors Boother click to export dk_0 Double-click to export dk_0 @ Hard Processors # Inter Processor System Double-click to export dk_0 Double-click to export dk_0 @ Hard Processor System Peripheral Gok Input Double-click to export dk_0 Double-click to export dk_0 @ Hard Processor System Double-click to export dk_0 Double-click to export dk_0 @ Hard Processor System Double-click to export dk_0 Double-click to export dk_0 @ Hard Processor System Outbe-click to export dk_0 <	-
Bill Base Functions Bill	
 Bos Bos	
i interface Protocols i interface Protocols i owno ruse i i owno ruse i owno i	
i: ow Prover j: ow Prover processors and Perphensis j: Or Processors j: Brinkedded Processor j: Brinkedded Processors j: Brinkedded Processors	
i Memory Interfaces and Controllers i Contr	¢0204_
Processors and Perphenals Go C+Processors Go C+Processor Go C	
Confracessors Head Processors Head Processor Head Proceshall Head Processor Head Proceshall Head Proceshall	
i) Hord Processors i) Hord Proc	
Bit Area Processor Components Bit Area Processor Components Bit Area Processor Systems Bit Area Processor Bit Area	
ight Hard Processor System 1D Peripheral Deuther-click to export (dk, 0) Perpherals Altera Xusion LCD 15/277 Altera Xusion LCD 15/277 Altera XUSiave To Avalon Altera Modular ADC core Altera Modular ADC core Alter	10200
i) inter-Process Communication i) emphasis i) emphasis	
Construction Altern Avalant (LD 1507) Altern 12C Save To Avalan Altern Avalant (LD 1507) Altern 12C Save To Avalan Altern Avalant (LD 1507) Altern 12C Save To Avalan Altern Modular (Ld AC) con Altern Avalant (LD 1507) Altern 12C Save To Avalan Altern Modular (Ld AC) con Altern Modular (Ld AC	
Altera Avalon LOD 16307 Altera Avalon Memory Mapped Slave Avalon Memory Mapped Slave Double-click to export (dk] (dk) (dk	
Attra 12C Stare To Avaion Attra 12C Stare 12C Stare 12C Stare 12C Stare 12C Stare Avaion Attra 12C Stare 12C Stare 12C Stare 12C Stare 12C Stare 12C Stare Avaion Attra 12C Stare 12C	
Auror Meddar ADA Ander Auror Meddar ADA	-0204
Attern Moduler Deal AGC con- Attern Moduler Deal AGC con- Littervia Timer: Attern Moduler Deal AGC con- Littervia Timer: Add Add Add Edt Add	10204
Internal Timer Low the Color Deputy	
tunic tail in a land tail in the second of the secon	
Edit. Add. Herar 20 Device Far 20	
	×0100
PixeL Conference (GLRU -> 1 PixeL Conference (GLRU ->	
SPI Skive In Junion Metter Ski Skive In Junion Metter Skive In Junion Metter Ski Skive In Junion Metter	
ex Edit OK Cool.nput Double-citic to export Citic ex Edit Add Facet Trput Double-citic to export Citic Herar Device Far external_connection Coskit Inel	
ew Edit Add Ferrar 20 Device Far 20 _ 0"	
Eetr Add Avaion Memory Mapped Save Double-click to export (chi) external_connection Conduct Ime1 	
Herar 🛛 Device Far 🖄 _ 🗂 🗖	¢0204
🕒 Line 1 💦 👘 🖓 👘 🐨 🐨 Current filter:	
bis ⊨ dk	
external_connection	- 0
- Preset	
Type Path Message	
B dk 0	
Nios_System.new_sdram_controller new_sdram_controller.wire must be exported, or connected to a matching conduit.	
The book of the sender jtag_uart Interrupt sender jtag_uart.irq is not connected to an interrupt receiver	
The person stram controller	bard
> (_
rors. 2 Warnings	

Figure 1-32

Platform Designer - Nios_System.qsys* (Edit System Generate View Tools H	C:\Nios_Tutorial	\Nios_System.qsy	s)						— c	- X
🖥 IP Catalog 🛛 🗕 📑	🗖 🎦 System	Contents 🛛	Address Map	Interconnect R	equirements	83				- d c
<x< th=""><th></th><th>🔺 🏢 System: I</th><th>Nios_System P</th><th>ath: Line 1. external</th><th>_connection</th><th></th><th></th><th></th><th></th><th></th></x<>		🔺 🏢 System: I	Nios_System P	ath: Line 1. external	_connection					
Project	+ sport		Clock	Base		End	IRQ	Tags	Opcode Name	
New Component										
System	X		exported							
Library			caporico							
-Basic Functions	En per									
DSP			CIK_U							
Interface Protocols	Doub	le-click to export	E							
Low Power										
Memory Interfaces and Controllers	Doub	le-click to export	clk_0							
Processors and Peripherals	Doub	le-click to export	[clk]							
+ Co-Processors	Doub	le-click to export	[dk]							
Embedded Processors	Doub	le-click to evnor	Edda							
Hard Processor Components	Daut	la click to export	[ell]		TRC C		700 21			
Hard Processor Systems	Doub	e-ciick to export	[CIK]		180 0		1RQ 31	1		
Inter-Process Communication	Doub	le-click to export	[cik]							
Peripherals	Doub	le-click to export	[clk]	© 0x0204_08	00	0x0204_0fff				
 Altera Avalon LCD 16207 	Doub	le-click to export	£							
 Altera I2C Slave To Avalon 										
Altera Modular ADC core	Doub		clk 0							
 Altera Modular Dual ADC co 	Doub		[all]							
Interval Timer	0000		Curd Curd							
Lauterbach Trace Interface	Doub		[CIK]	= 0x0204_10	18	0x0204_101±				
PIO (Parallel I/O)	Doub	le-click to export	[clk]					0		
Pixel Converter (BCR0> F										
 SPI Slave to Avalon Master 	Doub	le-click to export	clk_0							
>	Doub	le-click to export	[dk1]		00	0x0203_1fff				
	Doub	le-click to export	[dk1]	_		_				
Edit + Add										
	Doub									
			CIK_0							
🛛 🕺 Device Far 🕮 🔔 📑										>
1	^ n ~ f	t 🍸 🛒 Curre	nt filter:							
dk	X= Messac	ies 🐰								
reset	0_				1					
► s1	Туре	Path			Message					2
dk_0		2 Warnings								
jtag_uart		Nios System	new sdram (ontroller	new sdrar	n controller wi	re must be ever	orted or connected	d to a matching conduit	
 avalon_jtag_slave 		mos_system	inch_sarani_c	oncroner	nen_sara		e muse be expe	or connected		
► dk		Nios_System.	jtag_uart		Interrupt ser	nder jtag_uart.i	rq is not connec	cted to an interrup	treceiver	
🕨 irg		5 Info Message	s							
 reset 		Nice System	itan uart		TAGUART	IP input clock poor	to be at least .	double (2v) the co-	arating fraguency of ITAC TCY on he	ard
w_sdram_controller	×	nuos_system.	juag_uart		STAG OART I	ar input dock need	a to be at least (uousie (zx) uie opi	crowing mequency of 51AG TCK on bo	and v
>	<									>
a 2.Warnings									Constate HDI	Finial
ns, z wanings									Generate HDL.	·· rinis

Figure 1-33 Assigning interrupt numbers

Platform Designer - Nios_System.qsys* (Caller Strategy Caller Strategy Call	Nios_7	[utorial\	Nios_System.qs	ys)					- 🗆	\times
The Edit System Generate View 100is Hei The Decision of the Construction of the Cons	р • 🛱	System	Contents 🛛	Address Ma	ap 🛛 Interconnect R	equirements 🛛			- [-f 🗆
		X A	🕷 System:	Nios_System	n Path: new_sdram_co	ntroller.wire				
Project	 +	Use	Connections		Name	Description	Export	Clock	Base	
- E New Component - System	×		+ + -	×	debug_mem_slave custom_instruction	Avalon Memory Mapped Slave Custom Instruction Master	Double-click to export Double-click to export	[dk]	= 0x0204_	01 ^
Library Basic Functions DSP DInterface Protocols D-Interface Protocols D-Memory Interfaces and Controllers			•	•	☐ jtag_uart clk reset avalon_jtag_slave	JTAG UART Clock Input Reset Input Avalon Memory Mapped Slave Interrunt Sender	Double-click to export Double-click to export Double-click to export Double-click to export	clk_0 [dk] [dk]	■ 0x0204_	_1)
Handry J where a state of the other o			•		<pre>dk1 reset1</pre>	On-Chip Memory (RAM or ROM) Clock Input Avalon Memory Mapped Slave Reset Input	Double-click to export Double-click to export Double-click to export Double-click to export	clk_0 [dk1] [dk1]	₽ 0x0200_	_01
Inter-Process Communication Peripherals Altera Avalon LCD 16207 Altera I2C Slave To Avalon				↓ →	clk clk reset control_slave	System ID Peripheral Clock Input Reset Input Avalon Memory Mapped Slave	Double-click to export Double-click to export Double-click to export	clk_0 [dk] [dk]		11
Altera Modular ADC core Altera Modular Dual ADC co Interval Timer Lauterbach Trace Interface BIO (Pacellel 10)			•	↓ →	new_sdram_cont dk reset s1	roll SDRAM Controller Clock Input Reset Input Avalon Memory Mapped Slave	Double-click to export Double-click to export Double-click to export	clk_0 [dk] [dk]		01
Pixel Converter (BGR0> E				- po	wire	Conduit	sdram			
New Edit			•		Line1 ck reset s1	PIO (Parallel I/O) Clock Input Reset Input Avalon Memory Mapped Slave	Double-click to export Double-click to export Double-click to export	clk_0 [dk] [dk]	■ 0x0204_	_11
🔅 Herar 💠 Device Far 🛠 🚽	51	<		~~~	external_connection	n Conduit	line1			>
		जीन की	👻 🗑 Curr	ent filter						
œ-∎ dk_0	×=	Message	es 🛛							-
⊡-∎_ jtag_uart ⊕-∎ avalon_jtag_slave										
ter elk	I VF	be	Path			Message				**
Imerica) 	5 Into Message	25		The upper to be a deduced to be added	and the first of the second second second	6 77	C TOK hd	-î
= -= new_sdram_controller		9	wos_system	i.jtag_uart		DIAG OAKT IP INput clock need to be at le	ast upuble (2x) the operating frequ	ency or JIA	IG ICK on board	-
i → → − ck		9	Nios_System	i.sysid_qsy	/s_u	System ID is not assigned automatically.	cont the System 10 parameter to pro	vide a uniqi	un ar	
ter p= reset		9	Nios_System	i.sysid_qsy	/s_u	Time stamp will be automatically updated	when this component is generated.			- 1
wire v		<u> </u>	Nios_System	n.new_sdra	m_controller	SDRAM Controller will only be supported i	n Quartus Prime Standard Edition in	the future i	elease.	_ ~
< >	<									>
0 Errors, 0 Warnings								G	enerate HDL	Finish

Figure 1-34 Select the Hollow dot to connect the Interrupt Numbers

22.) Now we have no more warning and we can Generate our system. To do that select the **Generate** tab at the top of the screen as shown in figures 1-35 and 1-36.

	Generate HDI]						
IP Catalog	Generate Torthonch Su	rtom	em O	Contents 🛛 Addr	ess Map 🛛 Interconnec	t Requirements 🛛			-	- đ
_	Generate Example Dec	ian	, 🔺	System: Nios	System Path: new_sdram_	controller.wire				
roject	Show Instantiation Ter	nnlate		Connections	Name	Description	Export	Clock	Base	
New Componer	1	11 -0			debug mem sla	Avalon Memory Manned Slave	Double-click to export	[dk]	- 0×02	04.0
System		X			custom instructi	on m Custom Instruction Master	Double-click to export	[on]	- 0401	
ibrary					custom_instruct	UT_III Custoin Inst dealer Master	Double-click to export		_	
Basic Functions					i jtag_uart	JIAGUARI				
DSP		X				Clock Input	Double-click to export	clk_0		
Interface Protocols					→ reset	Reset Input	Double-click to export	[dk]		
Low Power		II -			→ avalon_jtag_slav	ve Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x02	04_
Memory Interfaces	and Controllers			+	irg	Interrupt Sender	Double-click to export	[dk]		
Processors and Per	pherals	T			🗉 onchin memor	v2 On-Chip Memory (RAM or ROM)				
Co-Processors					·	Clock Innut	Double-click to export	clk 0		
Embedded Proc	essors				1	Auglas Manage Managed Claus	Double allale and the surgest	Call of	0.0.00	
Hard Processor	Components				51	Avaion Memory Mapped Slave	Double-click to export	[OK1]	- UXU2	.00_
Hard Processor	Systems					Reset Input	Double-click to export	[dk1]		
Inter-Process C	ommunication				sysid_qsys	System ID Peripheral				
Peripherals				+ + + + + + + + + + + + + + + + + + + +	→ dk	Clock Input	Double-click to export	clk_0		
 Altera / 	Avalon LCD 16207					Reset Input	Double-click to export	[dk]		
• Altera I	2C Slave To Avalon				control slave	Avalon Memory Mapped Slave	Double-click to export	[dk]	0x02	04
 Altera I 	Modular ADC core					ntroll SDRAM Controller				-
• Altera I	Modular Dual ADC co				E new_surani_co		a 11 b 1 i			
 Interva 	Timer			T		Clock Input	Double-click to export	CIK_U		
 Lauteri 	ach Trace Interface				→ reset	Reset Input	Double-click to export	[dk]		
PIO (P	rallal T/O)				→ s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x01	.00_
 Pito (Fe Divol Co 	numerter (PCD0				🐦 wire	Conduit	sdram			1
 FIXELCU CDT Class 	viverter (boko>t				🕀 Line1	PIO (Parallel I/O)				
						Clark Innut	Double click to ownert	ally 0		
					UK .	Clock Input		CIK_U		
Edit.	and the second s				reset	Reset Input	Double-click to export	[CK]		
ew	- Auu			••		Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x02	:04_
		1			ଂଙ୍କ external_connec	tion Conduit	line1			
Hierar 💠 Devio	e Far 🕴 🔄 🖬 🗖	11	<							
			il. da							
🐵 🖿 s1	^		de alc	🗸 🔻 🕷 Current fi	ter:					
💷 dk_0		X-								
📑 jtag_uart		6E	Message	25 23					-	- 0
🝺 🖿 avalon_jtag_	slave		-	D-th		M				_
🔅 🖿 dk		р тур	e	Path		Message				
💩 🖿 irg				5 Info Messages						
🗄 🖿 reset		6		Nior System itse	uset	TTAC LIART IR input clock need to be at k	and double (2v) the operation from	ency of TAG	TCK on board	
new sdram cont	roller)	mos_system.jtag	_uarc	stind ower in input door need to be at it	east double (2x) the operating frequ	iency of 51MG	Tex of board	·
in ⊫ dk				Nios_System.sys	d_qsys_0	System ID is not assigned automatically.	Edit the System ID parameter to pro	vide a unique	ID	
- reset		(Nios System.sys	d asys 0	Time stamp will be automatically updated	when this component is generated.			
s1			-		and the second s			A . C		
. wire	~		Ð	nios_System.new	_soram_controller	SURAM Controller will only be supported i	n Quartus Prime Standard Edition in	the future rel	zase.	
		III -		1			and the second			

Figure 1-35 Generate HDL tab

- Generation		×
Synthesis		
Synthesis files are used to comp	e the system in a Ouartus project.	
Create HDL design files for synth	Iesis: Verlog	
	verling v	
	esumates for uniru-party EDA synthesis tools.	
Create block symbol file (.bs	f)	
 Simulation 		
The simulation model contains ge	nerated HDL files for the simulator, and may include simulation-only features.	
Simulation scripts for this compo	pent will be generated in a vendor-specific sub-directory in the specified output directory.	
	ich win beigener bedin in divendor specific bab directory in the specific dispect directory.	
Follow the guidance in the gener ip-make-simscript command-line	ated simulation scripts about how to structure your design's simulation scripts and how to use the <i>ip-setup-simulation</i> utilities to compile all of the files needed for simulating all of the IP in your design.	and
Create simulation model:	None	
 Output Directory 		
Path:	C:/Nios_Tutorial/Nios_System	
	Generate	e Cancel

Figure 1-36 Generate the system

• Generate completed	×
All 🖸 🛆 🕕	
Info: rsp_demux_002: "mm_interconnect_0" instantiated alter	era_merlin_demultiple> ^
Info: rsp_mux: "mm_interconnect_0" instantiated altera_me	rlin_multiplexer "rsp_m
Info: Reusing file C:/Nios_Tutorial/Nios_System/synthesis/su	ıbmodules/altera_merl
Info: rsp_mux_001: "mm_interconnect_0" instantiated altera	_merlin_multiplexer "r
Info: Reusing file C:/Nios_Tutorial/Nios_System/synthesis/su	ıbmodules/altera_merl
Info: avalon_st_adapter: "mm_interconnect_0" instantiated a	altera_avalon_st_adap
Info: error_adapter_0: "avalon_st_adapter" instantiated erro	r adapter "error adap
	"_dddptcr crivi_dddp
Info: Nios_System: Done "Nios_System" with 31 modules, 51 file	s
Info: Nios_System: Done "Nios_System" with 31 modules, 51 file Info: qsys-generate succeeded.	s
 Info: Nios_System: Done "Nios_System" with 31 modules, 51 file Info: qsys-generate succeeded. Info: Finished: Create HDL design files for synthesis 	s
 Info: Nios_System: Done "Nios_System" with 31 modules, 51 file Info: qsys-generate succeeded. Info: Finished: Create HDL design files for synthesis 	s
 Info: Nios_System: Done "Nios_System" with 31 modules, 51 file Info: qsys-generate succeeded. Info: Finished: Create HDL design files for synthesis Generate: completed successfully. 	s

Figure 1-37 Generation is successful

23.) We will be sending an 8x8 matrix of 8-bit numbers to and from the Nios processor so we will need a lot more PIO's. To do this we will do it exactly how it was shown previously. For our purposes we used 16 PIO's to make up the 8x8 matrix. The final product of this is shown in Figure 1-38 and 1-39.

lse	Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
\square		Line1_1	PIO (Parallel I/O)							
		dk	Clock Input	Double-click to export	clk_50					
		reset	Augles Memory Managed Slaves	Double-click to export	[CIK]	- 0-0000 0050	0.0000 0044			
		external connection	Conduit	line1 1	LON	= 0x0000_0010	0x0000_0011			
		Line1_2	PIO (Parallel I/O)							
	$ \downarrow \downarrow$	dk	Clock Input	Double-click to export	clk_50					
	$ + + + \rightarrow$	reset	Reset Input	Double-click to export	[dk]					
	•	s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	© 0x0000_00e0	0x0000_00ef			
	00	external_connection	Conduit	line1_2						
		□ Line2_1	PIO (Parallel I/O)							
		CK	Clock Input	Double-click to export	CIK_5U					
		st	Avalon Memory Manned Slave	Double-click to export	[cik]	- 0x0000 0080	0,0000 008f			
		external connection	Conduit	line2 1	Cond.					
		Line2_2	PIO (Parallel I/O)	-						
	$ \downarrow \downarrow$	dk	Clock Input	Double-click to export	clk_50					
	$ + + + \rightarrow$	reset	Reset Input	Double-click to export	[dk]					
	\diamond	s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	0b00_000x0 ¹	0x0000_00df			
	00	external_connection	Conduit	line2_2						
		□ Line3_1	PIO (Parallel I/O)							
		CK	Clock Input	Double-click to export	CIK_5U					
		st	Avalon Memory Manned Slave	Double-click to export	[cik]	- 0x0000 0090	0x0000 009f			
		external connection	Conduit	line3 1	Cand .					
		□ Line3_2	PIO (Parallel I/O)	-						
	$ \downarrow \downarrow$	dk	Clock Input	Double-click to export	clk_50					
		reset	Reset Input	Double-click to export	[dk]					
	• • • • • • • • • • • • • • • • • • • 	s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x0000_00a0	0x0000_00af			
	~~	external_connection	Conduit	line3_2						
		Line4_1	PIO (Parallel I/O) Clask Tanut	Daubla alials ta aumant	-11- 50					
		recet	Reset Toput	Double-click to export	CIK_30					
	\bullet	si	Avalon Memory Mapped Slave	Double-click to export	[dk]	CX0000 00b0	0x0000 00bf			
		external_connection	Conduit	line4_1						
		□ Line4_2	PIO (Parallel I/O)							
	$ + \rightarrow$	dk	Clock Input	Double-click to export	clk_50					
	$ + + + \rightarrow$	reset	Reset Input	Double-click to export	[dk]					
	• • • • • • • • • • • • • • • • • • •	s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x0000_00c0	0x0000_00cf			
	· ~	external_connection	Conduit	line4_2						
		Lines_1	PIO (Parallel I/O) Clask Tanut	Daubla eliels ta aunant	-11- 50					
		reset	Reset Toput	Double-click to export	CIK_30					
	\bullet \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow	si	Avalon Memory Mapped Slave	Double-click to export	[dk]		0x0000 007f			
		external_connection	Conduit	line5_1	· ·					
		Line5_2	PIO (Parallel I/O)							
	$ + \rightarrow$	dk	Clock Input	Double-click to export	clk_50					
		reset	Reset Input	Double-click to export	[dk]					
	• • • • • • • • • • • • • • • • • • •	s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	= 0x0000_0060	0x0000_006f			
	00	external_connection	Conduit	line5_2						
		🗆 Lineb_1	PIO (Parallel I/O)							

Figure 1-37

\checkmark	□ Line6_1	PIO (Parallel I/O)	_			
	dk dk	Clock Input	Double-click to export	clk_50		
	reset I	Reset Input	Double-click to export	[dk]		
	♦ ↔ s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	© 0x0000_0050	0x0000_005f
		Conduit	line6_1			
\checkmark	□ Line6_2	<u>PIO (Paral</u> lel I/O)				
	led.external_connection (start)	t	Double-click to export	clk_50		
	Conduit [conduit_end 17.1]	μt	Double-click to export	[dk]		
	Associated clock: None (asynch	mory Mapped Slave	Double-click to export	[dk]	© 0x0000_0040	0x0000_004f
	Line 1 outernal connection (a	ad	line6_2			
\checkmark	Conduit [conduit end 17.1]	nd) lel I/O)				
	Associated clock: None (asynch	t	Double-click to export	clk_50		
			Double-click to export	[dk]		
	◆ s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	© 0x0000_0030	0x0000_003f
_		Conduit	line7_1			
\checkmark	□ Line7_2	PIO (Parallel I/O)				
		Clock Input	Double-click to export	clk_50		
	reset i	Reset Input	Double-click to export	[dk]		
	◆ s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	© 0x0000_0020	0x0000_002f
_	C- external_connection	Conduit	line7_2			
\checkmark	□ Line8_1	PIO (Parallel I/O)		-		
		Clock Input	Double-click to export	clk_50		
	reset	Reset Input	Double-click to export	[clk]		
	◆ → s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	©x0000_0010	0x0000_001f
	external_connection	Conduit	line8_1			
\square	□ □ Line8_2	PIO (Parallel I/O)				
		Clock Input	Double-click to export	clk_50		
	● ● → reset	Reset Input	Double-click to export	[clk]		
	◆ → s1	Avalon Memory Mapped Slave	Double-click to export	[dk]	© 0x0000_0000	0x0000_000f
	ାଦ୍ୟ external_connection (Conduit	line8_2			

Figure 1-38

Chapter 2 Programming the FPGA

Now we will create a Verilog file that will instantiate and run everything that we have created above. It will be what connects all the components to the FPGA.

First thing we want to do is create our Verilog File. To do this, choose File → New → Verilog
 HDL File as shown in Figure 2-1 and 2-2. You will get a blank Verilog file as shown in Figure 2-3.



Figure 2-1 Creating New Verilog file



Figure 2-2 Creating new Verilog File

		Tutonat						
Project Navigator	A Hierarchy	▼ Q 📮 🗗 ×	*		Verilog1.v			
	Entity:Instance		圈 💏 📅 🚟 🗄	E 🖪 🗗 🔁 🖉 🏅	N 🖄 🧱 🗉			
A Cyclone IV GX: E	P4CGX150DF31C7							^
Nios_Tutorial	1.00							
'asks	Compilation	• ≡ ₽5×						
'asks	Compilation	▼ ≡ Q & ×						
′asks ✓ ► Comp	Compilation Task oile Design	▼ ≡ Q S ×						
asks	Compilation Task olle Design nalysis & Synthesis	▼ ≡ ₽ <i>6</i> ×						
asks ✓ ► Comp → ► A	Compilation Task oile Design nalysis & Synthesis tter (Place & Route)	▼ ≡ 0, 6 ×						
asks ✓ Comp > Ai > Fi > A	Compilation Task Jolie Design nalysis & Synthesis Itter (Place & Route) sembler (Generate prog	▼ ≡ ,						
asks ✓ Comp > A > A > T	Compilation Task Jile Design nalysis & Synthesis Itter (Place & Route) ssembler (Generate prog	▼ ≡ ♀ ♂ ×						
asks ✓ Comp > A > Fi > A > T > Fi > Comp	Compilation Task Jele Design Jalysis & Synthesis Itter (Place & Route) ssembler (Generate prog meQuest Timing Analya Da Nelist Witer	▼ ■ Q S × ramming files) is						
asks ✓ Comp > A > Fi > A > Ti > Ti > Coff Cof	Compilation Task Dile Design nalysis & Synthesis ther (Place & Route) ssembler (Generate prog mcQuest Timing Analys DA Netlisk Writer ettimes	▼ ≡ ♀ ♂ × ramming files) is						

Figure 2-3 Blank Verilog File

3.) We will need to write code for input and outputs as well as instantiating all the modules inside of the program. The code is shown in Figure 2-4 and 2-5. The input and output names can be found inside the Nios Verilog file that was created with the system we created. This is shown in Figure 2-6.

1	⊟module NiosII(
2	CLOCK_50,
3	LED,
4	SDRAM_CLK,
5	SDRAM_CKE,
6	SDRAM_ADDR,
6	SDRAM_BA,
	SDRAM_CS_N
10	SDRAM_CAS_N
11	SDRAM WE N
12	SDRAM_WE_N,
13	SDRAM DOM
14):
15	
16	input CLOCK_50;
17	output [7:0] LED;
18	output [12:0] SDRAM_ADDR;
19	output [1:0] SDRAM_BA;
20	output SDRAM_CAS_N, SDRAM_RAS_N;
21	output SDRAM_CKE, SDRAM_CS_N, SDRAM_WE_N, SDRAM_CLK;
22	OUTPUT [3:0] SDRAM_DOM;
23	inout wire [15:0] SDRAM_DQ;
24	
25	reg [31:0] Line1 1 TN:
20	reg [31:0] Line1 2 TN:
28	reg [31:0] Line2 1 TN:
29	reg [31:0] Line2 2 TN:
30	reg [31:0] Line3 1 TN:
31	reg [31:0] Line3 2 TN:
32	reg [31:0] Line4 1 IN:
33	reg [31:0] Line4_2_IN;
34	reg [31:0] Line5_1_IN;
35	reg [31:0] Line5_2_IN;
36	reg [31:0] Line6_1_IN;
37	reg [31:0] Line6_2_IN;
38	reg [31:0] Line7_1_IN;
39	reg [31:0] Line7_2_IN;
40	reg [31:0] Line8_1_IN;
41	reg [31:0] Line8_2_IN;
43 re	eg [31:0] Line1_1_OUT;
44 re	eg [31:0] Line1_2_OUT;
45 re	eg [31:0] Line2_1_OUT;
46 re	eg [31:0] Line2_2_OUT;
47 re	eg [31:0] Line3_1_OUT;
48 re	ag [31:0] Line3_2_OUT;
49 re	eg [31:0] Line4_1_OUT;
50 re	2g [31:0] Line4_2_OUT;
SL re	2g [31:0] Lines_1_001;
52 re	2g [31:0] Line5_2_001;
54	29 [51.0] Line6_1_001;
55	a [31:0] Line7 1 OUT
56	[31:0] Line7 2 OUT
57	a [31:0] Line& 1 OUT
58 14	eq [31:0] Line8 2 OUT
59	ig [biro] Enneo_2_oor,
60 w	ire [7:0] matrix_8 [64:1];
61	
52	
63 ⊡N1	ios u0(
64	.clk_clk(CLOCK_50),
65	.reset_reset_n(1'b1),
67	.ied_export(LED),
60	cdnam addr (CDRAM ADDR) //Address
60	.sdram_addr(SDRAM_ADDR), //Address
70	. Suram_Da(SURAM_BA), //Bank Address
70	.suram_cas_n(SDRAM_CAS_N), //COlumn Address Strobe
71	.Suram_CKE(SDRAM_CKE), //Clock Enable
72	.suram_cs_n(SDRAM_cs_N), //Chip Select
73	.sdram_ddm(SDRAM_DQ), //Data sdram_ddm(SDRAM_DOM) //Data_Wask
74	.suram_uqm(SDKAM_DQM), //Data Mask sdram ras n(SDRAM_DAS_N) //Dow Addross Strobo
76	.suram_ras_n(spram_ras_N), //row Address Strope sdram we n(spram_we N) //write Enable
77	. Son ant_we_n(SDRAm_we_w), //write enable

Figure 2-4 Verilog Code for Nios processor



Figure 2-5 SdramPLL is used for SDRAM Clock offset

	70 () : ## ## D D TD U D 20 20 =	
1	// Nios.v	,
3	// Generated using ACDS version 17.1 590	
5	`timescale 1 ps / 1 ps	
6	Emodule Nios (
8	output wire [7:0] led export // CIR.CIR	
ğ	input wire [31:0] line1 1 in port. // line1 1 in port	
10	output wire [31:0] line1_1_out_port, // .out_port	
11	input wire [31:0] line1_2_in_port, // line1_2.in_port	
12	output wire [31:0] line1_2_out_port, // .out_port	
14	autout wire [31:0] line2_1_m_port, // line2_1.m_port	
15	input wire [31:0] line2 in port, // line2 2 in port	
16	output wire [31:0] line2_2_out_port, // .out_port	
17	input wire [31:0] line3_1_in_port, // line3_1.in_port	
18	output wire [31:0] line3_1_out_port, // .out_port	
20	autout wire [31:0] Thes_2_nt port, // Thes_2. In port	
21	input wire [31:0] line4_1_in_port, // line4_1.in_port	
22	output wire [31:0] line4_1_out_port, // .out_port	
23	input wire [31:0] line4_2_in_port, // line4_2.in_port	
24	output wire [31:0] line4_2_out_port, // .out_port	
26	output wire [31:0] line5 1 out port. // .out port	
27	input wire [31:0] line5_2_in_port, // line5_2.in_port	
28	output wire [31:0] line5_2_out_port, // .out_port	
29	input wire [31:0] line6_1_in_port, // line6_1.in_port	
30	input wire [31:0] line6_1_out_port, // .out_port	
32	output wire [31:0] line6_2 out port. //out port	
33	input wire [31:0] line7_1_in_port, // line7_1.in_port	
34	output wire [31:0] line7_1_out_port, //out_port	
35	input wire [31:0] line7_2_in_port, // line7_2.in_port	
37	input wire [31:0] lines 1 in port // lines 1 in port	
38	output wire [31:0] line8 1 out port. // .out port	
39	input wire [31:0] line8_2_in_port, // line8_2.in_port	
40	output wire [31:0] line8_2_out_port, // .out_port	
41	input wire reset_reset_n, // reset.reset_n	
43	output wire [1:0] sdram ba. //	
44	output wire sdram_cas_n, // .cas_n	
45	output wire sdram_cke, // .cke	
46	output wire sdram_cs_n, // .cs_n	
47	nutu wire [15:0] soram_od, // .od	
49	output wire sdram ras n. // .ras n	
50	output wire sdram_we_n // .we_n	
51);	
52	uning [21:0] migs2 gen2 data master peoddata:	// mm interconnect Ounier2 and date meter readents
53	wire	<pre>// mm_interconnect_0:nios2_gen2_data_master_feaddata -> nios2_gen2:0_feaddat // mm_interconnect_0:nios2_gen2_data_master_waitrequest_> nios2_gen2:d wait</pre>
55	wine nies2 gen2_date meter debuggereet	// nige? applieduig mom elave debugarcare ta nome set intercorport Ornige'
C 1		>

Figure 2-6 Verilog module for Nios

4.) We will now need to compile the program. Choose Processing → Start Compilation as shown in Figures 2-7 and 2-8.



Quartus Prime Standard Edition - C:/Users/trace/OneDrive/Documents/CpE_405/Nios_17/Nios - Nios

Figure 2-7 Compiling Project



Figure 2-8 Compiling is complete

5.) You will need to assign pins to Clocks and any external input/output pins like LED's or Switches. You can do this through Assignments → Pin Planner as shown in Figures 2-9 and 2-10. We will then re-compile the project as shown above.

🕥 Quartus Prime Standard Edit	tion -	C:/Users/trace/OneDrive/Docu	ments/CpE_405	5/Nios	17/Nios - Nios		
File Edit View Project	Ass	ignments Processing Tools	s Window	Help			
🗋 🗖 🖶 🗠 🗂 💼 🗠	2	Device			🗲 🔅 💷 🕨 🖌 🤘 🖉	s 👋 🚮	
Project Navigator 💧	1	Settings	Ctrl+Shift	(+E	NiosII.v	Compilation Report - Nios	
Entity:	¢	Assignment Editor	Ctrl+Shift	t+A	- n 🗗 🛍 0 🛸 🕺 🎆 🛛		
A Cyclone IV GX: EP4CGX15	Ş	Pin Planner	Ctrl+Shift	:+N	v		
Y 🔤 Niosll 👛		Remove Assignments			ated using ACDS version 17	7.1 590	
abc sld_hub:auto_hub	5	Back-Annotate Assignments			le 1 ps / 1 ps ios (
abe pzdyqx:nabboc		Import Assignments			put wire clk_clk,	// clk.clk	
> 📥 Nios:u0		Export Assignments			put wire [31:0] line1_1_i	in_port, // line1_1.in_port	
> 🕴 sdramPLL:u1		Assignment Groups			put wire [31:0] line1_1_0 put wire [31:0] line1_2_i	in_port, // .out_port in_port, // line1_2.in_port	
	6	Logic Lock Regions Window	Alt+L		tput wire [31:0] line1_2_c put wire [31:0] line2_1_i	out_port, // .out_port in_port, // line2_1.in_port	
	å	Design Partitions Window	Alt+D		put wire [31:0] line2_1_0 put wire [31:0] line2_2_i	<pre>in_port, // .out_port in_port, // line2_2.in_port</pre>	
<			17 17 18 20 21 22 23 23 24 25 26 27 26 27 28 29	ir ou ir ou ir ou ir ou ir ou ir ou ir	tput wire [31:0] ine2_2_2 put wire [31:0] ine3_1_ tput wire [31:0] ine3_1_ tput wire [31:0] ine3_2_ tput wire [31:0] ine3_2_ tput wire [31:0] ine4_1_ tput wire [31:0] ine4_1_ tput wire [31:0] ine4_2_ tput wire [31:0] ine5_1_ tput wire [31:0] ine5_1_ tput wire [31:0] ine5_2_ tput wire [31:0] ine5_2_ tput wire [31:0] ine6_1_	<pre>sut_port, // inegl.in_port in_port, // inegl.in_port sut_port, // inegl.in_port</pre>	

Figure 2-9

	>								1777 - 22 - 222 - 222		
Named: * 🗸 🐇	Edit: 🔀 🗹 Out	put									
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
- CLOCK_50	Input	PIN_AJ16	4	B4_N2	PIN_AJ16	2.5 V		16mA (default)			
LED[0]	Output	PIN_AA25	5	B5_N2	PIN_AA25	2.5 V		16mA (default)	2 (default)		
LED[1]	Output	PIN_AB25	5	B5_N2	PIN_AB25	2.5 V		16mA (default)	2 (default)		
LED[2]	Output	PIN_F27	6	B6_N0	PIN_F27	2.5 V		16mA (default)	2 (default)		
LED[3]	Output	PIN_F26	6	B6_N0	PIN_F26	2.5 V		16mA (default)	2 (default)		
LED[4]	Output	PIN_W26	5	B5_N0	PIN_W26	2.5 V		16mA (default)	2 (default)		
LED[5]	Output	PIN_Y22	5	B5_N1	PIN_Y22	2.5 V		16mA (default)	2 (default)		
LED[6]	Output	PIN_Y25	5	B5_N2	PIN_Y25	2.5 V		16mA (default)	2 (default)		
LED[7]	Output	PIN_AA22	5	B5_N1	PIN_AA22	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[0]	Output	PIN_AG7	3	B3_N1	PIN_AG7	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[1]	Output	PIN_AJ7	3	B3_N1	PIN_AJ7	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[2]	Output	PIN_AG8	3	B3_N1	PIN_AG8	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[3]	Output	PIN_AH8	3	B3_N1	PIN_AH8	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[4]	Output	PIN_AE16	4	B4_N2	PIN_AE16	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[5]	Output	PIN_AF16	4	B4_N2	PIN_AF16	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[6]	Output	PIN_AE14	3	B3_N0	PIN_AE14	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[7]	Output	PIN_AE15	3	B3_N0	PIN_AE15	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[8]	Output	PIN_AE13	3	B3_N1	PIN_AE13	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[9]	Output	PIN_AE12	3	B3_N1	PIN_AE12	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[10]	Output	PIN_AH6	3	B3_N2	PIN_AH6	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[11]	Output	PIN_AE11	3	B3_N2	PIN_AE11	2.5 V		16mA (default)	2 (default)		
SDRAM_ADDR[12]	Output	PIN_AE10	3	B3_N2	PIN_AE10	2.5 V		16mA (default)	2 (default)		
SDRAM_BA[0]	Output	PIN_AH5	3	B3_N2	PIN_AH5	2.5 V		16mA (default)	2 (default)		
SDRAM_BA[1]	Output	PIN_AG6	3	B3_N2	PIN_AG6	2.5 V		16mA (default)	2 (default)		
SDRAM_CAS_N	Output	PIN_AJ4	3	B3_N2	PIN_AJ4	2.5 V		16mA (default)	2 (default)		
SDRAM_CKE	Output	PIN_AD6	3	B3_N2	PIN_AD6	2.5 V		16mA (default)	2 (default)		
SDRAM_CLK	Output	PIN_AE6	3	B3_N2	PIN_AE6	2.5 V		16mA (default)	2 (default)		
SDRAM_CS_N	Output	PIN_AG5	3	B3_N2	PIN_AG5	2.5 V		16mA (default)	2 (default)		
SDRAM_DQM[0]	Output	PIN_AF10	3	B3_N2	PIN_AF10	2.5 V		16mA (default)	2 (default)		
SDRAM_DQM[1]	Output	PIN_AB14	3	B3_N0	PIN_AB14	2.5 V		16mA (default)	2 (default)		
SDRAM_DQM[2]	Output	PIN_AH15	3	B3_N0	PIN_AH15	2.5 V		16mA (default)	2 (default)		
SDRAM_DQM[3]	Output	PIN_AH10	3	B3_N1	PIN_AH10	2.5 V		16mA (default)	2 (default)		
SDRAM_DQ[0]	Bidir	PIN_AD10	3	B3_N2	PIN_AD10	2.5 V		16mA (default)	2 (default)		
SDRAM_DQ[1]	Bidir	PIN_AD9	3	B3_N2	PIN_AD9	2.5 V		16mA (default)	2 (default)		
10 CORAM DO[2]	Ridie	DIN AEO		B2 ND	DIN AEO	2.5.1		16m A (dofault)	2 (dofault)		

Figure 2-10 Assigning pins to nodes

6.) Now we are ready to download the program to the board. To do this we will choose **Tools** → **Programmer** as shown in Figure 2-11.

File	Edit View Project Assignments	Processing	Tools	Window Help
	🚡 日 🤟 🗅 💼 ಶ 🖿 Nios			Run Simulation Tool 🔹 🕨
Project	Navigator 🔥 Hierarchy	▼ Q ₽ ₽	<u>کر</u>	Launch Simulation Library Compiler
	Entity:Instance		_	
🔥 Cy	clone IV GX: EP4CGX150DF31C7		.	TimeQuest Timing Analyzer
V abc	NiosII 🖆			Advisors •
	abc UHD sld_hub:auto_hub			Chip Planner
	pzdyqx:nabboc			Design Partition Planner
~	A Nios:u0			Netlist Viewers
>	sdramPLL:u1			Signal Tap Logic Applyment
			2	Signal Tap Logic Analyzer
				In-System Memory Content Editor
				Logic Analyzer Interface Editor
				Cincel Disks Disc
			XX	Signal Probe Pins
				Programmer
			×	
				Fault Injection Debugger
<			-	System Debugging Tools
Tasks	Compilation	▼ ≡ ₽ ₽	54	IP Catalog
	Task		1	Nios II Software Build Tools for Eclipse
6	V Compile Design		Å 1	Platform Designer
- 	> Analysis & Synthesis		1	Tcl Scripts
 Image: A second s	> Fitter (Place & Route)		()	Customize
 Image: A second s	> 🕨 Assembler (Generate progra	mming files)	(Options
 Image: A second s	> TimeQuest Timing Analysis		(I	License Setup
	> EDA Netlist Writer		S	Install Devices

Figure 2-11 Quartus Programmer

7.) Now we will select our file and device we want to program and then program the board as shown in Figure 2-12. Make sure your board is connected to your computer and it is turned on.

Hardware Setup Enable real-time ISP	USB-Blaster [USB-0]	Mode:	THE					
Enable real-time ISF	o to allow background pro		JIAG	•	Progress:			
Mu Start		ogramming when av	ailable					
	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Stop o	output_files/Nios_time	EP4CGX150DF31	00ACCBD0	00ACCBD0	\checkmark			
Auto Detect								
X Delete								
Mo Add File	د							
Change File								
Add Device 가泣 Up 가泣 Down	TDI EP4CGX1500	DF31						

Figure 2-12 Quartus Programmer

8.) Once programmed, the progress meter should be at 100% shown in Figure 2-13. The FPGA is now configured with the Nios System. Now we need to write our C program to execute.

Programmer - C:/Users/trace/OneDrive/Documents/CpE_405/Nios_17/Nios - Nios - [Nios_time_limited.cdf]								
File Edit View	Processing Tools Wind	dow Help				Sear	ch altera.c	om 🖤
Hardware Setup.	USB-Blaster [USB-0]	Mode:	JTAG	v	Progress:	10	00% (Succ	essful)
Enable real-time i	SP to allow background pro	igramming when ava	inable					
▶ ⁹ b Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
[■] Stop	output_files/Nios_time	EP4CGX150DF31	00ACCBD0	00ACCBD0	\checkmark			
Auto Detect								
🗙 Delete								
Add File								
Change File	<							>
Save File								
Add Device		-						
1 ¹⁰ Up								
LW Down								
Down	EP4CGX150	F31						
	TDO							

Figure 2-13 FPGA is programmed with Nios System

Chapter 3 NIOS II Software Build Tools for Eclipse

This Chapter covers build flow of Nios II C coded software program.

The Nios II Software Build Tools (SBT) for Eclipse is an easy-to-use graphical user interface (GUI) that automates build and makefile management. The Nios II SBT for Eclipse integrates a text editor, debugger, the BSP editor, the Nios II flash programmer and the Quartus II Programmer. The included example software application templates make it easy for new software programmers to get started quickly. In this section you will use the Nios II SBT for Eclipse to compile a simple C language example software program to run on the Nios II standard system configured onto the FPGA on your development board. You will create a new software project, build it, and run it on the target hardware. You will also edit the project, re-build it, and set up a debug session.

3.1 Creating a simple Hello World Example Project

In this section we will be creating a simple C project that will print Hello World! Onto the console.

1.) Choose Tools \rightarrow NIOS II Software Build Tools for Eclipse shown in Figure 3-1.



Figure 3-1

2.) You will then select your workspace you want to work in as shown in Figure 3-2.

See Workspace Launcher							
Select a workspace							
Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.							
Workspace:	C:\Nios_Tutorial		~	Browse			
Use this a	s the default and do not ask again	ОК		Cancel			

Figure 3-2

 Next will be creating the Hello World Example Project. Choose File → New → Nios II Application and BSP from Template as shown in Figure 3-3.

	Nios II - Eclipse				
File	Edit Navigate Search	Project Run Nios	II V	Nindow Help	
	New	Alt+Shift+N >	C++	Nios II Application and BSP from Template	
	Open File		C++	Nios II Application	
	Close	Ctrl+W	C++	Nios II Board Support Package	
	Close All	Ctrl+Shift+W	C++	Nios II Library	
	Save	Ctrl+S		Project	
	Save As		D	Other	Ctrl+N
B	Save All	Ctrl+Shift+S	Г		
	Revert		L .		
	Move		1		
2	Rename	F2	L .		
\$	Refresh	F5	L .		
	Convert Line Delimiters To	> >	L .		
Ð	Print	Ctrl+P			
	Switch Workspace	>			
	Restart				
2	Import				
\geq	Export				
	Properties	Alt+Enter			
	Exit				
_			1		



4.) Now we will choose your SOPC File which will be located in the directory that you created your Qsys in. We will then create our name and select the Hello World Template and select Finish.

Nios II Application and BSP from Template					
Nios II Software Examples Create a new application and boa template	rd support package based on a software example				
Target hardware information	C:\Users\trace\OneDrive\Documents\CpE 405\Nios 17\Nios.sopc	i			
CPU name:	nios2_gen2 v				
Application project Project name: Hello_Templa	te				
Use default location					
Project location: C:\User	<pre>\\trace\UneDrive\Documents\Cp2_403\Nilos_17\software\Hello_1er</pre>	Υ			
Templates	Template description				
Blank Project Board Diagnostics Count Binary Float2 Functionality Float2 GCC Float2 Performance Hello Freestanding Hello MicroC/OS-II	Hello World prints 'Hello from Nios II' to STDOUT. This example runs with or without the MicroC/OS-II RTOS and requires an STDOUT device in your system's hardware. For details, click Finish to create the project and refer to the readme.bt file in the project directory.	^			
Hello World Hello World Small Memory Test Memory Test Small	 The BSP for this template is based on the Altera HAL operating system. For information about how this software example relates to 				
?	< Back Next > Finish	Cancel			

Figure 3-4 Creating Hello World Template

5.) The system will create everything you will need. You can click on the hello_world.c file to see the C code that was created.

Nios II - Hello_Template/hello_world.c - Eclipse

File Edit Source Refactor Navigate Search Project Run Nios II Window Help 📑 = 🖩 🐘 🔯 = 🕸 = 🖻 = 🔗 = 🔯 = 🕐 = 🗨 = 🕐 = 🖓 = 🖉 = 🖉 = 🖓 = 🖓 = 👘 Project Explorer 🛛 🗈 hello_world.c 🛛 🖻 🔄 🐌 🗢 2⊕ * "Hello World" example... 16 🗸 😂 Hello_Template 17 #include <stdio.h> >
 Includes
 Includes
 Includes 18 19⊖ int main() create-this-app 20 { 🚡 Makefile 21 printf("Hello from Nios II!\n"); readme.txt 22 ✓ ➡ Hello_Template_bsp [Nios] 23 return 0; > 🔊 Includes 24 } > 👝 drivers 25 > 👝 HAL ic alt_sys_init.c
 in linker.h > 🔥 system.h create-this-bsp linker.x 🚡 Makefile 🚡 mem_init.mk memory.gdb 🚡 public.mk 📄 settings.bsp summary.html 🖹 Problems 🧔 Tasks 📃 Console 🔀 🔲 Properties CDT Build Console [Hello_Template] [Hello_Template clean complete] 17:40:49 Build Finished (took 210ms) < Writable Smart Insert 2:1

Figure 3-5 C code for Hello World

6.) Now we will need to build and run this program. Right click on the Hello_Template → Build Project as shown in Figure 3-6. Next we will run this program by right clicking again on Hello_Template → Run As → Nios II Hardware as shown in Figure 3-7.



Figure 3-6 Building Project

🖨 Nios II - Hello_T	emplate/hello_world.c - Eclipse					
File Edit Source	Refactor Navigate Search	Project Run Nios II Win	dow Help			
	New Go Into	>	⋪∙i⊿≎ia - a - a			
i Project Expl	Open in New Window		ample.			
✓ 😂 Hello_T 🗎	Сору	Ctrl+C				
> 🐇 Bina 💼	Paste	Ctrl+V				
> 🗊 Incl 🗙	Delete	Delete				
> 🦳 obj	Remove from Context	Ctrl+Alt+Shift+Down				
> 🐝 Hell	Source	>	m Nios II!\n");			
📄 crea	Move					
🗎 Hell	Rename	F2				
📄 Hell	Import					
	Export					
✓ [™] Hello_T						
> 🔊 Incl	Build Project					
> 🗁 driv	Clean Project					
> 🔁 HAL 🐉	Refresh	15				
> h link	Close Project					
> h syst	Close Unrelated Projects					
Crea	Make Targets	>				
📄 link	Index	>				
a mer	Build Configurations	>				
📄 mer	Show in Remote Systems view					
opub 🗋	Profiling Tools	>				
i sett	Run As	>	🔺 1 Lauterbach ISS 🕹 🗘 😫			
	Debug As	>	C 2 Local C/C++ Application			
	Profile As	>	3 Nios II Hardware			
	Restore from Local History		🔯 4 Nios II ModelSim			
	Nios II	>	Run Configurations			
	Update Linked Resources					
🛠 🕺	Run C/C++ Code Analysis					

Figure 3-7 Running Nios Project

7.) You should now see in the console a message displaying "Hello From Nios II!".



Figure 3-8

3.2 DCT and Quantization with Nios Processor

1.) We are now going to edit the project we just created for hello world to do DCT on an 8x8 matrix. First, we will need to change the code. The DCT code is shown below.

```
#include <stdio.h>
#include <math.h>
#include "system.h"
#include "io.h"
#include "altera_avalon_pio_regs.h"
#define pi 3.142857
const int m = 8, n = 8;
// Function to find discrete cosine transform and print it
int dctTransform(int matrix[][n], int Qmatrix[][n])
{
    int i, j, k, l;
```

```
// dct will store the discrete cosine transform
  float dct[m][n];
  float Qdct[m][n];
  float ci, cj, dct1, sum;
  for (i = 0; i < m; i++) {
     for (j = 0; j < n; j++) {
        // ci and cj depends on frequency as well as
        // number of row and columns of specified matrix
        if (i == 0)
           ci = 1 / sqrt(m);
        else
           ci = sqrt(2) / sqrt(m);
        if (j == 0)
           cj = 1 / sqrt(n);
        else
           cj = sqrt(2) / sqrt(n);
        // sum will temporarily store the sum of
        // cosine signals
        sum = 0;
        for (k = 0; k < m; k++) {
           for (1 = 0; 1 < n; 1++) {
            matrix[k][l] = matrix[k][l] - 128;
              dct1 = matrix[k][1] *
                   cos((2 * k + 1) * i * pi / (2 * m)) *
                   cos((2 * l + 1) * j * pi / (2 * n));
              sum = sum + dct1;
           }
        }
        Qdct[i][j] = ci * cj * sum;
        dct[i][j] = Qdct[i][j] / Qmatrix[i][j];
     }
  }
  for (i = 0; i < m; i++) {</pre>
     for (j = 0; j < n; j++) {
        printf("%f\t", dct[i][j]);
     }
     printf("\n");
  }
  return 0;
// Driver code
int main()
{
```

}

```
int Qmatrix[8][8] = { { 16, 11, 10, 16, 24, 40, 51, 61 },
                     { 12, 12, 14, 19, 26, 58, 60, 55 },
                     { 14, 13, 16, 24, 40, 57, 69, 56 },
                     { 14, 17, 22, 29, 51, 87, 80, 62 },
                     { 18, 22, 37, 56, 68, 109, 103, 77 },
                     { 24, 35, 55, 64, 81, 104, 113, 92 },
                     { 49, 64, 78, 87, 103, 121, 120, 101 },
                     { 72, 92, 95, 98, 112, 100, 103, 99 } };
int temp;
int t, r;
  for(t = 0; t < 8; t++)
  {
        for(r = 0; r < 8; r++)
              if(t == 0 && r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE1 1 BASE, r);
              if(t == 0 \&\& r >= 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE1 2 BASE, r-4);
              if(t == 1 \&\& r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE2 1 BASE, r);
              if(t == 1 && r >= 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE2 2 BASE, r-4);
              }
              if(t == 2 && r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE3 1 BASE, r);
              if(t == 2 && r >= 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE3 2 BASE, r-4);
              if(t == 3 && r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE4 1 BASE, r);
              }
              if(t == 3 && r >= 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE4 2 BASE, r-4);
              if(t == 4 && r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE5 1 BASE, r);
              if(t == 4 && r >= 4)
              {
```

```
matrix[t][r] = IORD 8DIRECT(LINE5 2 BASE, r-4);
              }
              if(t == 5 \&\& r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE6 1 BASE, r);
              }
              if(t == 5 \&\& r >= 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE6 2 BASE, r-4);
              if(t == 6 && r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE7 1 BASE, r);
              }
              if(t == 6 && r >= 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE7 2 BASE, r-4);
              if(t == 7 && r < 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE8 1 BASE, r);
              }
              if(t == 7 && r >= 4)
              {
                    matrix[t][r] = IORD 8DIRECT(LINE8 2 BASE, r-4);
              }
        }
  }
    dctTransform(matrix, Qmatrix);
return 0;
```

}

2.) Now we will go back to the Verilog file and change that to send data to the Nios Processor. Our new code is shown below. We generate a 8x8 matrix with all values of 255 for testing and put those values into our PIO's to be sent into the Nios processor. We will need to recompile this program as well.

module NiosII(CLOCK_50, LED, SDRAM_CLK, SDRAM_CKE, SDRAM_ADDR, SDRAM_BA, SDRAM_BA, SDRAM_CAS_N, SDRAM_CAS_N, SDRAM_RAS_N, SDRAM_WE_N,

```
SDRAM_DQ,
SDRAM_DQM
);
```

input CLOCK_50; output [7:0] LED; output [12:0] SDRAM_ADDR; output [1:0] SDRAM_BA; output SDRAM_CAS_N, SDRAM_RAS_N; output SDRAM_CKE, SDRAM_CS_N, SDRAM_WE_N, SDRAM_CLK; output [3:0] SDRAM_DQM; inout wire [15:0] SDRAM_DQ;

```
reg [31:0] Line1 1 IN;
reg [31:0] Line1_2_IN;
reg [31:0] Line2_1_IN;
reg [31:0] Line2 2 IN;
reg [31:0] Line3_1_IN;
reg [31:0] Line3_2_IN;
reg [31:0] Line4_1_IN;
reg [31:0] Line4_2_IN;
reg [31:0] Line5_1_IN;
reg [31:0] Line5_2_IN;
reg [31:0] Line6_1_IN;
reg [31:0] Line6_2_IN;
reg [31:0] Line7_1_IN;
reg [31:0] Line7_2_IN;
reg [31:0] Line8_1_IN;
reg [31:0] Line8_2_IN;
reg [31:0] Line1_1_OUT;
reg [31:0] Line1_2_OUT;
reg [31:0] Line2_1_OUT;
reg [31:0] Line2_2_OUT;
reg [31:0] Line3_1_OUT;
reg [31:0] Line3 2 OUT;
reg [31:0] Line4_1_OUT;
reg [31:0] Line4_2_OUT;
reg [31:0] Line5_1_OUT;
reg [31:0] Line5_2_OUT;
reg [31:0] Line6_1_OUT;
reg [31:0] Line6_2_OUT;
reg [31:0] Line7_1_OUT;
```

reg [31:0] Line7_2_OUT; reg [31:0] Line8_1_OUT; reg [31:0] Line8_2_OUT;

wire [7:0] matrix_8 [64:1];

Nios u0(

.clk_clk(CLOCK_50), .reset_reset_n(1'b1), .led_export(LED),

.sdram_addr(SDRAM_ADDR), //Address .sdram_ba(SDRAM_BA), //Bank Address .sdram_cas_n(SDRAM_CAS_N), //Column Address Strobe .sdram_cke(SDRAM_CKE), //Clock Enable .sdram_cs_n(SDRAM_CS_N), //Chip Select .sdram_dq(SDRAM_DQ), //Data .sdram_dqm(SDRAM_DQM), //Data Mask .sdram_ras_n(SDRAM_RAS_N), //Row Address Strobe .sdram_we_n(SDRAM_WE_N), //Write Enable

.line1_1_in_port(Line1_1_IN), // line1_1.in_port .line1_1_out_port(Line1_1_OUT), // .out_port .line1_2_in_port(Line1_2_IN), // line1_2.in_port .line1_2_out_port(Line1_2_OUT), // .out_port .line2 1 in port(Line2 1 IN), // line2 1.in port .line2_1_out_port(Line2_1_OUT), // .out port .line2 2 in port(Line2 2 IN), // line2 2.in port .line2_2_out_port(Line2_2_OUT), // .out_port .line3 1 in port(Line3 1 IN), // line3 1.in port .line3_1_out_port(Line3_1_OUT), // .out_port .line3_2_in_port(Line3_2_IN), // line3_2.in_port .line3 2 out port(Line3 2 OUT), // .out_port .line4_1_in_port(Line4_1_IN), // line4_1.in_port .line4_1_out_port(Line4_1_OUT), // .out_port .line4 2 in port(Line4 2 IN), // line4 2.in port .line4_2_out_port(Line4_2_OUT), // .out_port .line5_1_in_port(Line5_1_IN), // line5_1.in_port .line5 1 out port(Line5 1 OUT), // .out port .line5_2_in_port(Line5_2_IN), // line5_2.in_port .line5_2_out_port(Line5_2_OUT), // .out_port .line6_1_in_port(Line6_1_IN), // line6_1.in_port .line6_1_out_port(Line6_1_OUT), // .out_port .line6_2_in_port(Line6_2_IN), // line6_2.in_port .line6_2_out_port(Line6_2_OUT), // .out_port .line7_1_in_port(Line7_1_IN), // line7_1.in_port .line7_1_out_port(Line7_1_OUT), // .out_port .line7_2_in_port(Line7_2_IN), // line7_2.in_port .line7_2_out_port(Line7_2_OUT), // .out_port .line8_1_in_port(Line8_1_IN), // line8_1.in_port .line8_1_out_port(Line8_1_OUT), // .out_port .line8_2_in_port(Line8_2_IN), // line8_2.in_port .line8_2_out_port(Line8_2_OUT), // .out_port

);

sdramPLL u1(.inclk0(CLOCK_50), .c0(SDRAM_CLK)

);

```
genvar f;
generate
for(f=1; f<65; f=f+1) begin: kForl
begin
assign matrix_8[f] = 255;
end
end
endgenerate
```

always @(posedge SDRAM_CLK) begin

Line1_1_IN <= {matrix_8[1], matrix_8[2], matrix_8[3], matrix_8[4]}; Line1_2_IN <= {matrix_8[5], matrix_8[6], matrix_8[7], matrix_8[8]};

Line2_1_IN <= {matrix_8[9], matrix_8[10], matrix_8[11], matrix_8[12]}; Line2_2_IN <= {matrix_8[13], matrix_8[14], matrix_8[15], matrix_8[16]};

Line3_1_IN <= {matrix_8[17], matrix_8[18], matrix_8[19], matrix_8[20]}; Line3_2_IN <= {matrix_8[21], matrix_8[22], matrix_8[23], matrix_8[24]};

Line4_1_IN <= {matrix_8[25], matrix_8[26], matrix_8[27], matrix_8[28]};

Line4_2_IN <= {matrix_8[29], matrix_8[30], matrix_8[31], matrix_8[32]};

Line5_1_IN <= {matrix_8[33], matrix_8[34], matrix_8[35], matrix_8[36]}; Line5_2_IN <= {matrix_8[37], matrix_8[38], matrix_8[39], matrix_8[40]};

Line6_1_IN <= {matrix_8[41], matrix_8[42], matrix_8[43], matrix_8[44]}; Line6_2_IN <= {matrix_8[45], matrix_8[46], matrix_8[47], matrix_8[48]};

Line7_1_IN <= {matrix_8[49], matrix_8[50], matrix_8[51], matrix_8[52]}; Line7_2_IN <= {matrix_8[53], matrix_8[54], matrix_8[55], matrix_8[56]};

Line8_1_IN <= {matrix_8[57], matrix_8[58], matrix_8[59], matrix_8[60]}; Line8_2_IN <= {matrix_8[61], matrix_8[62], matrix_8[63], matrix_8[64]};

end

endmodule

3.) Now that we have recompiled the program, we will reopen the Nios software builder where we had just updated our program. We will now run that program and we should see a 8x8 matrix with the values after the DCT and Quantization as shown in Figure 3-9.

🖹 Problems	s 🔊 Tasks	🔄 Console	📩 Nios II Cor	nsole 🛛 🗌	Properties			
Final_DCT Nios	-inal_DCT Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 name: jtaguart_0							
63.499996	0.000416	-0.060249	0.077513	-0.081104	0.068931	-0.072960	0.080430	
0.342246	-0.000224	0.000227	-0.000190	0.000157	-0.000081	0.000092	-0.000120	
-0.640788	0.000425	-0.000370	0.000272	-0.000179	0.000141	-0.000133	0.000186	
1.015214	-0.000499	0.000411	-0.000339	0.000206	-0.000134	0.000163	-0.000240	
-1.114728	0.000530	-0.000338	0.000235	-0.000211	0.000145	-0.000169	0.000254	
1.118093	-0.000457	0.000293	-0.000274	0.000231	-0.000197	0.000201	-0.000274	
-0.712827	0.000330	-0.000272	0.000262	-0.000234	0.000213	-0.000243	0.000318	
0.623905	-0.000296	0.000287	-0.000290	0.000275	-0.000328	0.000352	-0.000407	

Figure 3-9 The Values after DCT and Quantization

You can use this program and incorporate it into multiple different types of projects. The next step after this will be doing full JPEG compression after sending the values that we had gotten in Figure 3-9 back to the FPGA.