

My First Nios II - Quartus II 15.0(64-bit)



Copyright © 2003-2015 Terasic Inc. All Rights Reserved.

CONTENTS

Chapter 1	Hardware Design	1
1. 1 Requirem	ents	
1. 2 Create a H	Iardware Design	
1. 3 Download	Hardware Design to MAX10 NEEK Board	
Chapter 2	NIOS II Software Build Tools for Eclipse	42
2. 1 Create the	hello_world Example Project	
2.2 Build and	Run the Program	
2.3 Edit and F	Re-Run the Program	
2.4 Why the I	ED Blinks	
2. 5 Setup Deb	bug Configuration	
2. 6 Configure	System Library	
Chapter 3	Program and Boot On-chip Flash	54
3.1 Modify Pro	oject in Qsys	
3.2 Reconfigur	re Nios II BSP Editor	
3.3 Programm	ing On-chip Flash	



Chapter 1

Hardware Design

This tutorial provides comprehensive information which will help users understand how to create a FPGA based QSYS system and implement it on MAX10 NEEK board and run software upon it.

1.1 Requirements

The Nios II processor core is a soft-core central processing unit that users program onto Altera Field Programmable Gate Array (FPGA). This tutorial illustrates the basic flow from hardware creation to software building.

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
- JTAG link for communication between the host computer and target
- hardware (typically using a USB-Blaster cable)
- LED peripheral I/O (PIO), be used as indicators

1.2 Create a Hardware Design

This section describes how to create a hardware system including QSYS feature.

1. Launch Quartus II and select **File->New** Project Wizard to create a new project, as shown in **Figure 1-1** and **Figure 1-2**.



MAX10 Neek My Niosll Manual



Figure 1-1 Select File -> New Project Wizard in Quartus II



🕼 New Project Wizard	x
Directory, Name, Top-Level Entity	
What is the working directory for this project?	
D:\altera\15.0	
What is the name of this project?	
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.	
Use Existing Project Settings	
< <u>B</u> ack <u>N</u> ext > <u>Fi</u> nish Cancel <u>H</u>	elp

Figure 1-2 First page of the New Project Wizard

2. Choose a working directory for this project. Enter the project name and top-level entity name, as shown in **Figure 1-3**. Click **Next** and a window will pop up, as shown in **Figure 1-4**.



💱 New Project Wizard	×
Directory, Name, Top-Level Entity	
What is the working directory for this project?	
D:\my_first_niosii	
What is the name of this project?	
my_first_niosii	
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.	
my_first_niosii	
Use Existing Project Settings	
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel <u>H</u> e	lp

Figure 1-3 Define the working directory, project name, and top-level design entity name



🔇 New Project Wizard	×
Add Files	
Select the design files you want to include in the project. Click Add All to add all design files in the project director	y to the project.
Note: you can always add design files to the project later.	
Eile name:	<u>A</u> dd
File Name Type Library Design Entry/Synthesis Tool HDL Version	Add All
	Remove
	Q
	Down
	Properties
Specify the path names of any non-default libraries.	
< <u>B</u> ack <u>N</u> ext >	Einish Cancel Help

Figure 1-4 Add Files

3. Click **Next** and choose the device family with device settings according to **Figure 1-5**. Click **Next** to the next window, as shown in **Figure 1-6**.



	and quantas in		, man your langer				e Support list webbade.	
Device family				Show	v in 'Available de	evices' list		
Family: MAX 10 (DA	/DF/DC/SA/SF/SC)			▼ Pack	age:	Anv	•	
					Package: An		ny •	
Devices: All				▼ Pin <u>o</u>	ount:	Any	•	
Target device				Core	Sp <u>e</u> ed grade:	Any	•	
 Auto device cela 	ate of her the Cittee			Name	e filter:			
Auto device selection	cted by the Fitter							
Specific device se	elected in 'Available d	levices' list		🗸 S	how advanced	devices		
O Other: n/a								
vailable devices:								
vailable devices:	Core Voltage	LEs	Total I/Os	GPIOs	Memory	Bits	Embedded multiplier 9-bit elements	
vailable devices:	Core Voltage	LEs 49760	Total I/Os	GPIOs 178	Memory 1677312	Bits 2	Embedded multiplier 9-bit elements	
vailable devices: Name 10M50DAF256C7G 10M50DAF256C8G	Core Voltage 1.2V 1.2V	LEs 49760 49760	Total I/Os 178 178	GPIOs 178 178	Memory 1677312 1677312	Bits 2	Embedded multiplier 9-bit elements 88 88	
vailable devices: Name 10M50DAF256C7G 10M50DAF256C8G 10M50DAF256C8GES	Core Voltage 1.2V 1.2V 1.2V	LEs 49760 49760 49760	Total I/Os 178 178 178 178	GPIOs 178 178 178	Memory 1677312 1677312 1677312	Bits 2 2 2 2	Embedded multiplier 9-bit elements 88 88 88	
vailable devices: Name 10M50DAF256C7G 10M50DAF256C8G 10M50DAF256C8GES 10M50DAF256I7G	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 49760 49760 49760 49760	Total I/Os 178 178 178 178 178	GPIOs 178 178 178 178	Memory 1677312 1677312 1677312 1677312 1677312	Bits 2 2 2 2 2	Embedded multiplier 9-bit elements 88 88 88 88	
vailable devices: Name 10M50DAF256C7G 10M50DAF256C8G 10M50DAF256C8GES 10M50DAF25617G 10M50DAF484C6GES	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 49760 49760 49760 49760 49760	Total I/Os 178 178 178 178 178 178 178 178 178 178 178 178 178	GPIOs 178 178 178 178 178 360	Memory 1677312 1677312 1677312 1677312 1677312 1677312	Bits 2 2 2 2 2 2 2 2 2	Embedded multiplier 9-bit elements 88 88 88 88 88	
Qmer: n/a vailable devices: Name 10M50DAF256C7G 10M50DAF256C8G 10M50DAF256C8G 10M50DAF25617G 10M50DAF25617G 10M50DAF484C6GES 10M50DAF484C7G 10M50DAF484C7G	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 49760 49760 49760 49760 49760 49760 49760	Total I/Os 178 178 178 178 360 360	GPIOs 178 178 178 178 178 360 360	Memory 1677312 1677312 1677312 1677312 1677312 1677312 1677312	Bits 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Embedded multiplier 9-bit elements 88 88 88 88 88 88 88 88 88 88 88 88 88	
Umer: n/a vailable devices: Name 10M50DAF256C7G 10M50DAF256C8G 10M50DAF256C8G 10M50DAF25617G 10M50DAF25617G 10M50DAF484C6GES 10M50DAF484C6GES 10M50DAF484C8G 10M50DAF484C8G 10M50DAF484C8G	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 49760 49760 49760 49760 49760 49760 49760 49760 49760	Total I/Os 178 178 178 178 360 360 360 360	GPIOs 178 178 178 178 178 360 360 360 360	Memory 1677312 1677312 1677312 1677312 1677312 1677312 1677312 1677312	Bits 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Embedded multiplier 9-bit elements 88 88 88 88 88 88 88 88 88 88 88 88 88	
Umer: nya wailable devices: Name 10M50DAF256C7G 10M50DAF256C8G 10M50DAF256C8G 10M50DAF25617G 10M50DAF25617G 10M50DAF484C6GES 10M50DAF484C6GES 10M50DAF484C8G 10M50DAF484C8GES 10M50DAF484C8GES	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 49760 49760 49760 49760 49760 49760 49760 49760 49760	Total I/Os 178 178 178 178 360 360 360 360 360 360 360 360	GPIOs 178 178 178 178 360 360 360 360 360	Memory 1677312 1677312 1677312 1677312 1677312 1677312 1677312 1677312 1677312	Bits 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Embedded multiplier 9-bit elements 88 88 88 88 88 88 88 88 88 88 88 88 88	

Figure 1-5 Family & Device Settings



and There	Teel News	E+(-)		Due Teel Automotionly
ooin ype				Run Tool Automatically to synthesize the current design
esign entry/synthesis				Pup gate-level cimulation automatically after completion
		- CNUTE>	· · ·	
	Timing	<none></none>	-	
	Symbol	<none></none>		
	Signal Integrity	<none></none>	•	<i>.</i>
	Boundary Scan	<none></none>		

Figure 1-6 EDA Tool Settings

4. Click **Next** and a summary about the project which has just been created is shown in **Figure 1-7**. Click **Finish** to complete the creation of a new project, as shown in **Figure 1-8**.



🚱 New Project Wizard	X		
Summary			
When you dick Finish, the project will be created with the following settings:			
Project directory:	D:\altera\15.0		
Project name:	my_first_niosii		
Top-level design entity:	my_first_niosii		
Number of files added:	0		
Number of user libraries added:	0		
Device assignments:			
Design template:	n/a		
Family name:	MAX 10 (DA/DF/DC/SA/SF/SC)		
Device:	10M50DAF484C6GES		
EDA tools:			
Design entry/synthesis:	<none> (<none>)</none></none>		
Simulation:	<none> (<none>)</none></none>		
Timing analysis:	0		
Operating conditions:			
Core voltage:	1.2V		
Junction temperature range:	0-85 ℃		
	< <u>Back</u> <u>N</u> ext > <u>Finish</u> Cancel <u>H</u> elp		

Figure 1-7 Summary of the new project





Figure 1-8 The new project in Quartus II

5. Choose **Tools** -> **Qsys** to open the Qsys system builder tool, as shown in **Figure 1-9**. Choose **File- > New System** to create a new Qsys system, as shown in **Figure 1-10**.



💱 Quartus II 64-Bit - D:/my_first_niosii/my_first_nio	osii - my_first_niosii	The second strength from the second strength of
File Edit View Project Assignments Processing	Tools Window Help 🗟	
Project Navigator	Run Simulation Tool	10000000000000000000000000000000000000
Entity	TimeQuest Timing Analyzer	
A MAX 10: 10M50DAF484C6GES	Advisors	
Imp_first_niosii ⁴ √a		
	Chip Planner Design Partition Planner Netlist Viewers	
	 SignalTap II Logic Analyzer In-System Memory Content Editor Logic Analyzer Interface Editor In-System Sources and Probes Editor SignalProbe Pins Programmer ITAG Chain Debugger Fault Injection Debugger System Debugging Tools 	
✓ III ▲ Hierarchy ■ Files J ^a Design Units 丫 IF	 IP Catalog Nios II Software Build Tools for Eclipse 	
Taske	👗 Qsys	
Flow: Compilation	D Td Scripts	
Task	Customize	
Compile Design	Options	
Analysis & Synthesis	License Setup	
Fitter (Place & Route)	🥡 Install Devices	
Assembler (Generate programming files)		
TimeQuest Timing Analysis		

Figure 1-9 Select Tool -> Qsys



👃 Qsys - unsaved.qsys* (D	:\my_first_niosii\unsaved	qsys)					March March
<u>File</u> <u>E</u> dit <u>System</u> <u>G</u> enerate	e <u>V</u> iew <u>T</u> ools <u>H</u> elp						
New System New Component	Ctrl+N		Syst	em Contents 🛛	Address Map unsaved	X Interconnect	t Requirements 🛛
New System New Component Open Save Save As Refresh System Export System as hw: Browse Project Direct Recent Projects Exit Browse Project Direct Recent Projects Exit Direct Browse Project Direct Recent Projects Browse Project Direct Browse Project	ce Family		Image: System Image: System <th>em Contents X</th> <th>Address Map unsaved k_0 ilk_in ilk_in_reset ilk ilk_reset</th> <th>Interconnect Description Clock Source Clock Input Reset Input Clock Output Reset Output</th> <th>t Requirements 🙁</th>	em Contents X	Address Map unsaved k_0 ilk_in ilk_in_reset ilk ilk_reset	Interconnect Description Clock Source Clock Input Reset Input Clock Output Reset Output	t Requirements 🙁
unsaved [unsaved. qs	sys*]						
treset treset treset treset treset treset							

Figure 1-10 Create a new Qsys system

6. Choose File -> Save and enter "NEEK10_QSYS.qsys", as shown in Figure 1-11 and Figure 1-12.



👃 Qsys - unsaved.qsys* (D:\my_first_nios	ii\unsaved.c	qsys)						-	-	Mg. Frank M.	-
<u>File E</u> dit <u>System G</u> enerate <u>V</u> iew <u>T</u> ools	<u>H</u> elp										
New System	Ctrl+N	- 5 🗆		System	Content	s 🖾	Address Map	83	Interconnect	Requirements	83
New Component	-	~ (M) 🕅 S1	stem:	unsaved				
Open	Ctrl+O	X 🔛	+	IIco	for a	Nome			Decerintion		
Save	Ctrl+S			V		E cl	k 0		Clock Source		
Save As			×		⊳		lk_in		Clock Input		
Refresh System	F5					c	lk_in_reset		Reset Input		
Export System as hw.tcl Component	:					c	lk Normant		Clock Output		
Browse Project Directory			-			- C	IK_reset		Reset output		
Recent Projects	Þ		Ŧ								
E <u>x</u> it	Alt+F4										
New Edit	1	🕂 Add									
Image: state											

Figure 1-11 Select File -> Save in Qsys

📥 Save: un	saved		×
Save in:	<pre>my_first_nio</pre>	sii 🔻 🌶 🖡	୭
Recen	Ì .qsys_edit Ì db		
Desktop			
My Do			
(All Computer			
	File name:	NEEK10_QSYS qsys	Save
Network	Files of type:	Qsys System Files (*.qsys) 🔹 🔻	Cancel

Figure 1-12 Enter "NEEK10_QSYS.qsys" as the file name

7. Right-click the **clk_0** component and rename it to **clk_50**. Double click **clk_50** to set the clock frequency to 50000000 Hz, as shown in **Figure 1-13**.



👃 Qsys - NEEK10_QSYS.qsys* (D:\my_first_niosii\NEEK10_QSYS.qs	ys)	Andrew Track	Marchine Root Comparison, Name
<u>File E</u> dit <u>S</u> ystem <u>G</u> enerate <u>V</u> iew <u>T</u> ools <u>H</u> elp			
🧏 Hierarchy 🛛 Device Family 🎘 🗕 🗗 🗖	🙀 Parameters 🛛 🛛 System	Contents 🛛 Address Map	🛛 Interconnect Requirements 🖾
PREKIO_QSTS [NEEK10_QSTS. qsys*]	System: NEEK10_QSYS Path:	: clk_50	
₽ = clk ₽ = reset • = clk_c50	Clock Source clock source		
	* Parameters		
	Clock frequency: 5	50000000 Hz	
	Clock frequency is know	*n	
	Reset synchronous edges:	Jone 🔻	

Figure 1-13 Rename the clock source and set it to 50000000 Hz

8. Choose Library -> Processors and Peripherals- > Embedded Processors -> Nios II Processor to add a Nios II processor into the system, as shown in Figure 1-14 and Figure 1-15.

& Qsys - NEEK10_QSYS.qsys* (D:\my_first_niosii\NEEK10_	QSYS.qsys))					
<u>File E</u> dit <u>System G</u> enerate <u>V</u> iew <u>T</u> ools <u>H</u> elp							
📑 IP Catalog 🛛	- 🗗 🗖		System	Content	s 🛛	Address Map	X
	× 🔯			S 🛄 S3	stem:	NEEK10_QSYS	Pa
E-Configuration and Programming		+	Use	Con	Name		
⊕DMA		- L			🗆 el	k 50	
⊞…On Chip Memory		X				lkin	
🗄 "Simulation; Debug and Verification						lk in reset	
Bitec		.~~				112	
				×		lle vosot	
H. Interface Protocols						IR_reset	
tow Power							
the mory Interfaces and Controllers							
Trocessors and Peripherals		—					
- Trocessors							
Bitsmon							
• Dicswap	=						
 Custom Instruction Master Iranslator 							
Custom Instruction Slave Translator							
Floating Point Hardware							
····· · Floating Point Hardware 2							
Embedded Processors							
····· · Nios II (Classic) Processor							
Ilios II Processor							
Hard Processor Components							
Hard Processor Systems							
the Inter-Process Communication	T						
New Edit	Add						
🧯 Hierarchy 🛞	- 🗗 🗖	il					
PERIO_QSTS [NEEK10_QSTS. qsys*]							
ter → clk							
🕂 🗝 reset							
E = clk_50							





MAX10 Neek My Niosll Manual

👃 Nios II Processor - nios2_gen2_0	10.000	Data Serve		x		
Nios II Processor Mesecere altera_nios2_gen2				Documentation		
Block Disgram Show signals nios2_gen2_0 clk clock avalon	Main Vectors Select an Nios II Core	Caches and Memory Interfaces Arithmet Implementation ¹ O Nios II/e ² Nios II/f	ic Instructions MMU and MPU Settings JIAC	Debug Advanced Features		
reset reset avaion- irg interrupt reset debug mem slave	Summary	Nios II/e Resource-optimized 32-bit RISC	Nios II/f Performance-optimized 32-bit RTSC	E		
avalon nics_custom_instruction	Features	JTAG Debug	JIAG Debug Hardware Waltiply/Divide Instruction/Dats Caches Tightly-Coupled Masters ECC RAM Protection External Interrupt Controller Shadow Register Sets MPU MNU			
	RAM Usage	2 + Options	2 + Options			
<pre></pre>						
				Cancel Finish		

Figure 1-15 Settings of Nios II Processor

9. Click **Finish** to return to the main window, as shown in **Figure 1-16**.

🕹 Qsys - NEEK10_QSYS.qsys* (D:\my_first_niosii\NEEK10_QSYS.qsys)									
File Edit System Generate View Tools	File Edit System Generate View Tools Help								
📑 IP Catalog 🛛 🗕 🗗 🗖		System	Contents 🛛 Addres	s Map 🛛 Interconnect	t Requirements 🙁 Device Fami	ly 🛛 Parameters 🕮		- 🗗	
🔍 🗙 🔯			System: NEEK10_0	STS Path: nios2_gen2	_0				
E Configuration and Programm -	÷	Use	Connections	Name	Description	Export	Clock	Base	
⊕DMA	B_0	V		□ clk_50	Clock Source				
+ On Chip Memory	×		P-	clk_in	Clock Input	clk	exported		
			○ D-	clk_in_reset	Reset Input	reset			
+DSP				clk	Clock Output	Double-click to	clk_50		
Interface Protocols				clk_reset	Reset Output	Double-click to			
tow Power		V		🗏 🖳 ni os2_gen2_0	Nios II Processor				
🕂 Memory Interfaces and Control 👻	<u> </u>		\diamond	clk	Clock Input	Double-click to	unconnecte	4	
< <u> </u>	×			reset	Reset Input	Double-click to	[clk]		
				data_master	Avalon Memory Mapped Master	Double-click to	[clk]		
New Edit 🕂 Add				instruction_master	Avalon Memory Mapped Master	Double-click to	[clk]		_
				ırq	Interrupt Receiver	Double-click to	[clk]		
🧏 Hierarchy 🕴 🗕 🗗 🗖				debug_reset_request	Keset Uutput	Double-click to	[CIK]	- 0-0800	
				debug_mem_slave	Avaion Memory Mapped Slave	Double-click to	[CIR]		
B NEEKIO_QSIS [NEEKIO_QSIS. qsys*			,	custom_instructi	Custom instruction Master	Double-click to	1		
i os2_gen2_0									
🗄 📲 Connections		•							F
		n _~ f	t 🝸 🐂 Current fil	ter: All Interfaces					
	× = ©=	Messag	es 🛛					- đ	
	Tw	Toma Path Massara							
		-	4.7						
		• •							- 21
۰ III ۲									
4 Errors, O Warnings							Gen	erate HDL Fini	sh
	_		-						

Figure 1-16 Complete adding Nios II processor

10. Right-click on **nios2_gen2_0** and choose **rename**, as shown in **Figure 1-17**. Rename the Nios II processor to **cpu**. Connect its clk and reset to the clk and clk_reset of **clk_50**, as shown in **Figure 1-18**.







👃 Qsys - NEEK10_QSYS.qsys* (D:\my_first_niosi	ii\NEEK10)_QSYS.qsys)						-	
File Edit System Generate View Tools Help									
🚰 IP Catalog 🛛 🗕 🗗 🗖	🖾 Syster	n Contents 🕺 Addre	ss Map 🙁]	Interconnect	t Requirements 🙁 Device Famil	y 🛛 Parameters 🕮			- d 🗆
N 🔍 📉 🕺 📗		- System. HEEKIO	18015 I 801.	cpu.reset	1				
Project	Use	Connections	Name		Description	Export	Clock	Base	End
I New Component	₩ ₩	_	□ c1k_50		Clock Source				
Library			clk_in		Clock Input	clk	exported		
• NEEK10_QSYS	-24		cik_in_r	eset	Clash Outent	reset	-11- 50		
Basic Functions	Z		cik olk roco		Reset Output	Double-click to	CIR_50		
H Arithmetic	<u> </u>		F D cnn		Nies II Processor	Double-click to			
the Bridges and Adaptors	-		clk		Clock Input	Dephlerelick to	clk 50		
Configuration and Programming	×		reset		Reset Input	Double click to	[clk]		
DMA			data_mas	ter	Avalon Memory Mapped Master	Double-click to	[clk]		
🗇 On Chip Memory			instruct	ion_master	Avalon Memory Mapped Master	Double-click to	[clk]		
Altera On-Chip Flash		$\times \longrightarrow$	irq		Interrupt Receiver	Double-click to	[clk]		IRQ O
• Altera User Flash Memor			debug_re:	set_request	Reset Output	Double-click to	[clk]		
Altera User Flash Memor		$ \bullet \bullet \longrightarrow$	debug_mer	n_slave	Avalon Memory Mapped Slave	Double-click to	[clk]	■ 0x0800	0x0fff
···· Avalon FIFO Memory		×	custom_in	nstructi	Custom Instruction Master	Double-click to			
····· @ Avalon-ST Dual Clock FI									
• Avalon-SI Multi-Channel -									
×									
Ilew Edit									
🕌 Hierarchy 🔀 🗕 🗗 🗖									
e - clk									
🕀 🕨 🕨 clk_in									
the clk_in_reset	•								4
During cik_reset	nla A								
clk	142.1	Current fi.	lter: All Into	eriaces					
🖶 🖛 custom_instruction_master 😑 📑	Messa	ges 🖾							
🛨 🖛 data_master		-							
the debug_mem_slave	Туре	Path		Message					
- uebug_reset_request instruction master	8	2 Errors							
irq	8	NEEK10_QSTS. cpu		Reset slav	e is not specified. Please select	the reset slave			
🕀 📭 reset	8	NEEK10 QSTS. cou		Exception	slave is not specified. Please se	lect the exception slave			
A Hust bridge	•			III					•
2 Errors, O Warnings								Generat	HDL Finish

Figure 1-18 Establish the connection of clk and reset

11. Choose **Library -> Interface Protocols -> Serial -> JTAG UART** to add JTAG UART and click **Finish**, as shown in **Figure 1-19** and **Figure 1-20**.





Figure 1-19 Add JTAG UART



👃 JTAG UART - jtag_uart_0	
JTAG UART MegeCore altera_avalon_jtag_uar	t
Block Diagram Show signals	Trite FIFO (Data from Avalon to JTAG) Buffer depth (bytes): 64 • IRQ threshold: 8 Construct using registers instead of memory blocks Read FIFO (Data from JTAG to Avalon) Buffer depth (bytes): 64 • IRQ threshold: 8 Construct using registers instead of memory blocks
🛦 Warning: jtag_uart_0: Jtag U	art input clock need to be at least 50Mhz to operate proper
•	4
	Cancel Finish

Figure 1-20 Settings of JTAG UART

12. Rename **jtag_uart_0** to **jtag_uart**. Connect its clk and reset to the clk and clk_reset of **clk_50**, respectively. Its avalon_jtag_slave is connected to the data_master of **cpu**. **Figure 1-21** shows the complete connections.





Figure 1-21 Rename JTAG UART

13. Choose Library -> Basic Functions -> On-Chip Memory-> On-Chip Memory (RAM or ROM) to add On-Chip memory, as shown in Figure 1-22 and Figure 1-23.



👃 Qsys - NEEK10_QSYS.qsys* (D:\my_first_niosii\NEEK10_QSYS.qsys File Edit System Generate View Tools Help 🖣 IP Catalog - 🗗 🗖 × 🔯 Project . . <u> </u> <u> </u> −System Library ···· NEEK10_QSYS Basic Functions ide ∴Arithmetic Bridges and Adaptors = Clocks; PLLs and Resets E-Configuration and Programming 🖶 On Chip Memory 🔍 🔍 Altera On-Chip Flash 🚥 🔍 Altera User Flash Memory for I2C Interface Pro Altera User Flash Memory for Parallel Interface 🚥 🛛 Altera User Flash Memory for SPI Interface Pro 📟 🔍 Avalon FIFO Memory ---- 🔍 Avalon-ST Dual Clock FIFO Avalon-ST Multi-Channel Shared Memory FIFO Avalon-ST Round Robin Scheduler Avalon-ST Single Clock FIF0 • On-Chip Memory (RAM or ROM) E. Simulation; Debug and Verification 🗄 Bitec -Interface Protocols 🗄 Audio & Video 🗄 Ethernet 🗄 Interlaken <. 111 Edit... 🕂 Add... New...

Figure 1-22 Add On-Chip Memory



& On-Chip Memory (RAM or ROM) - onchip_me	mory2_0	
0n-Chip Memory (RAM or R	OM)	
MegaCore altera_avalon_onchip_memory2		Documentation
Block Disgram		
Show signals	Nemory type	
	lype:	RAM (Writable) 🔻
onchip_memory2_0	Dual-port access	
	Single clock operation	
clk1 clock	Read During Write Mode:	DONT_CARE -
s1avalon	Block type:	AUTO -
reset1 reset		
attera avalon onchin memory?		
	Size	
	Data width:	32 •
	Total memory size:	4096 bytes
	Minimize memory block usage (may impac	t fmax)
	Read latency	
	Slave s1 Latency:	1 🔻
	Slave s2 Latency:	1 -
	PON/PAN Naman Production	
	Reset Request:	Enabled V
	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.h	nex) or select the hex file using the file browser button.
	User created initialization file	: onchip_mem.hex
	Enable In-System Memory Content Editor	feature
	Instance ID:	HONE
		VIO OSVE
	memory will be initialized from NEE	лиу-4919-0DCD1p_memory2_U. дех
J		
I		
		Cancel Finish

Figure 1-23 On-Chip Memory Box

16. Modify Total memory size to 102400 and uncheck "initialize memory content", as shown in **Figure 1-24**. Click **Finish** to return to the window as in **Figure 1-25**.



👃 On-Chip Memory (RAM or ROM) - onchip_me	mory2_0	×
0n-Chip Memory (RAM or R	OM)	
MogaCore altera_avalon_onchip_memory2		Documentation
Block Diagram		×
Show signals	Memory type	
		RAM (Writable) 🔻
onchip_memory2_0	Dual-port access	
	Single clock operation	
clock	Read During Write Mode:	DONT_CARE 👻
s1avalon	Block type:	AUTO -
reset		
altera_avalon_onchip_memory2	V C :	
	Data width:	32 🔻
	Iotal memory size:	102400 bytes
	Minimize memory block usage (may impac	t fmov)
	minimize memory brock asage (may impac	v Linaky
	Read latency	
	Slave sl Latency:	1 -
	Slave s2 Latency:	1 -
	RON/RAM Nemory Protection	
	Reset Request:	Enabled -
	ECC Parameter	
	Extend the data width to support ECC bits:	Disabled 🔻
	×	
	Initialize memory content	
	Enable non-default initialization file	
		un) en eslect the her file mine the file homen better
	lype the filename (e.g. my_ram.f	ex/ of select the nex life using the life browser button.
		onchip_mem.nex
	Enable In-System Memory Content Editor	feature
	Instance ID:	ITOITE
	This memory is not initialized duri	ng device programming.
I		
		Cancel Finish

Figure 1-24 Update the default settings of On-Chip Memory

17. Rename **onchip_memory2_0** to **onchip_memory2**. Connect its clk1 and reset1 to the clk and clk_reset of **clk_50**. Its s1 is connected to the data_master and instruction_master of **cpu**. **Figure 1-25** shows the complete connections.



& Qsys - NEEK10_QSYS.qsys* (D:\my_first_n	iosii\N	EEK10	_QSYS.qsys)						Aug. 81.	- 0 - X
<u>File E</u> dit <u>S</u> ystem <u>G</u> enerate <u>V</u> iew <u>T</u> ools <u>H</u> e	elp									
📑 IP Catalog 🛛 🗕 🗗 🗖		System	Contents 🛛	Addre	ss Map 🛛 Interconne	ct Requirements 🙁 Device Fam	ily 🛛 Parameters 🕮			- đ =
🔍 on-chip 🗙 🔯			System:	NEEK10	QSYS Path: onchip_me	emory2				
Project	+	Use	Connections		Name	Description	Export	Clock	Base	End
New Component	20	V			□ clk_50	Clock Source				
Library	×				clk_in	Clock Input	clk	exported		
Basic Functions	1			~ D -	clk_in_reset	Reset Input	reset			
Altera On-Chin Flash	1				clk	Clock Output	Double-click to	clk_50		
······ • Altera User Flash Memory 1					clk_reset	Reset Output	Double-click to			
····· Altera User Flash Memory :		V			🗆 🖳 cpu	Nios II Processor				
Altera User Flash Memory :			+	\rightarrow	clk	Clock Input	Double-click to	c1k_50		
····· Avalon FIFO Memory	X		+	$\leftrightarrow \rightarrow$	reset	Reset Input	Double-click to	[clk]		
···· · Avalon-ST Dual Clock FIFO					data_master	Avalon Memory Mapped Master	Double-click to	[clk]		
···· · Avalon-ST Multi-Channel Si					instruction_master	Avalon Memory Mapped Master	Double-click to	[clk]		
···· · Avalon-ST Round Robin Sche			$\ \ \ c$		irq	Interrupt Receiver	Double-click to	[clk]		IRQ O
Avalon-SI Single Clock FII				\succ	debug reset reques	t Reset Output	Double-click to	[clk]		
Un-Uhip Memory (RAM or RUM				\rightarrow	debug mem slave	Avalon Memory Mapped Slave	Double-click to	[clk]	0x0800 ₪	0x0fff
				×	custom instructi	. Custom Instruction Master	Double-click to			
					- itae mart	TTAG HART	PORDAL DATUR OF			
		-		\rightarrow	- ,a	Clock Input	Deel-Terre Trink Ar	c1k 50		
4				\rightarrow	racat	Reset Tanut	D 17 7 1 1	[a]]e]		
						Auglas Massau Massad Slava	Double-click to	[-1]-]	- 0~0000	0~0007
No. 2314					avaion_]tag_siave	Tetermet Cerder	Double-click to	[.1].]		0x0001
Hew		177			irq	Interrupt Sender	Double-click to	[CIR]		
)	ᆡ				onchip_memory2	Un"Chip Memory (RAM or RUM)		11.50		
🐛 Hierarchy 🛛 🛛 🗕 🗗 🗖					clkl	Clock Input	Double-click to	clk_50		
the clk	1				sl	Avalon Memory Mapped Slave	Double-click to	[clk1]	0x0000	UxUfff
🗈 🖿 clk_in			0	\rightarrow	reset1	Reset Input	Double-click to	[clk1]		
🖶 🖿 clk_in_reset										
🖶 🗝 clk_reset		•								•
• U cpu		n~ f	t 🭸 🐂 Curr	ent fi	lter: All Interfaces					
the system itse slave										
elk	8	Messag	jes 🖂							
🖶 🖿 irg										
🗄 🗭 reset	Typ	e	Fath		,	lessage				<u></u>
international i	- 8		7 Errors							
🖶 🖿 clk1	(3	NEEK10 OST	S. cpu	Re	set slave is not specified Plea	se select the reset slave			
🗎 🖿 resetl		-				and in all of the set of the set of the				
		-	BEERIU_QST	o. cpu	Ex	ception slave is not specified.	riease select the exceptio	n stave		
										•
7 Errors, O Warnings									Genera	te HDL Finish

Figure 1-25 Rename the On-Chip Memory and establish connections via nodes

18. Click **cpu** from the component list on the right to edit its settings. Click the Vectors tab to update Reset vector and Exception Vector, as shown in **Figure 1-26**, and click **Finish**.





Figure 1-26 Update the settings of Nios II processor

19. Choose Library -> Processors and Peripherals -> Peripherals -> PIO (Parallel I/O) to open add PIO component, as shown in Figure 1-27. Set the Width to 10 bits and click Finish, as shown in Figure 1-28.





Figure 1-27 Add PIO component



👃 PIO (Parallel I/O) - pio_0	
PIO (Parallel I/O) Megacore altera_avalon_pio	Documentation
Block Diagram Show signals clk reset s1 external_connection	Basic Settings Width (1-32 bits): 10 Direction: Input Input InOut 0 Output Output Port Reset Value: 0x00000000000000000000000000000000000
	 Enable individual bit setting/clearing Edge capture register Synchronously capture Edge Type: RISING Enable bit-clearing for edge capture register Interrupt Generate IRQ IRQ Type: LEVEL Level: Interrupt CPU when any unmasked I/O pin is logic true Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enabled
	Test bench wiring Hardwire PIO inputs in test bench Drive inputs to: 0x00000000000000000000000000000000000
	Cancel Finish

Figure 1-28 Set the Width to 10 bits

21. Rename **pio_0** to **pio_led** and connect its clk and reset to the clk and clk_reset of **clk_50**. Its s1 is connected to the data_master of **cpu**, reset to clk_reset of **clk_50**. Double click the Export column of external_connection to export the signal "pio_led_external_connection". **Figure 1-29** shows the complete connections and changes.





Figure 1-29 Connections and changes of PIO

22. Choose **System** -> **Assign Base Addresses**, as shown in **Figure 1-30**. The base addresses will be assigned automatically and there should be no more error or warning message, as shown in **Figure 1-31**.













23. Click **Generate HDL**... button on the bottom right corner and a window will pop up, as shown in **Figure 1-32**. Click **Generate** button and the process will begin, as shown in **Figure 1-33**. **Figure 1-34** shows there is no error message during the process.

👃 Generation	The statement	Base Same	(read)	×
Synthesis				
Synthesis files are used to compile t	he system in a Quartus II p	roject.		
Create HDL design files for synthesis	Verilog 🔻			
Create timing and resource estimat	tes for third-party EDA synt	hesis tools.		
✔ Create block symbol file (.bsf)				
Simulation				
The simulation model contains generat	ed HDL files for the simula	tor, and may include simul	ation-only features.	
Create simulation model:	None 🔻			
Allow mixed-language simulation				
Enable this if your simulator support	s mixed-language simulation			
Output Directory				
I ath.	D:/my_first_niosii/NEEK10	_QSYS		
			Generat	e Cancel

Figure 1-32 Generate HDL files in Qsys

👃 Generate 🛛 🗙
A11 🖸 🔺 🕕
Info: Saving generation log to D:/my_first_niosii/NEEK10_QSYS/NEEK10_QSYS_ger
Info: Starting: Create block symbol file (.bsf)
🕕 Info: qsys-generate D:\my_first_niosii\NEEK10_QSYS.qsysblock-symbol-file -
Progress: Loading my_first_niosii/NEEK10_QSYS. qsys
Progress: Reading input file
Progress: Adding clk_50 [clock_source 15.0]
Progress: Parameterizing module clk_50
Progress: Adding cpu [altera_nios2_gen2 15.0]
Progress: Parameterizing module cpu
🕕 Progress: Adding jtag_uart [altera_avalon_jtag_uart 15.0]
Generate
Stop

Figure 1-33 Generation process



🚣 Generate Completed 🛛 🕅
All 🙁 🛆 🕕
UInfo: rsp_demux_001: "mm_interconnect_0" instantiated altera_merlin_d
Info: rsp_mux: "mm_interconnect_0" instantiated altera_merlin_multipl
Info: Reusing file D:/my_first_niosii/NEEK10_QSYS/synthesis/submodul
Info: rsp_mux_001: "mm_interconnect_0" instantiated altera_merlin_mul
Info: Reusing file D:/my_first_niosii/NEEK10_QSYS/synthesis/submodul
Info: avalon_st_adapter: "mm_interconnect_0" instantiated altera_aval
Info: error_adapter_0: "avalon_st_adapter" instantiated error_adapter
Info: NEEK10_QSYS: Done "NEEK10_QSYS" with 29 modules, 49 files
Info: ip-generate succeeded.
Info: Finished: Create HDL design files for synthesis
۲ ۲
© Generate: completed successfully.
Stop Close

Figure 1-34 Generation is complete and successful

25. Choose **File** -> **New** to open the dialogue and be prepared to add new files, as shown in **Figure 1-35** and **Figure 1-36**.



😋 Q	uartus II 64-Bit - D:/my	_first_niosii/my_	first_nio	sii - n	ny_first_nio	sii		
<u>F</u> ile	<u>Edit V</u> iew <u>P</u> roject <u>/</u>	<u>A</u> ssignments P <u>r</u> oc	cessing	<u>T</u> ools	<u>W</u> indow	<u>H</u> elp	P	
0	<u>N</u> ew	Ctrl+N	irst_nic	osii			- 🐹	/ /
2	Open	Ctrl+O			₽₽×	:		
	Close	Ctrl+F4			×	1		
雷	New Project Wizard		ity					
1	Open Project	Ctrl+J	acy					
	Save Projec <u>t</u>							
	Clos <u>e</u> Project							
	Save	Ctrl+S						
	Save <u>A</u> s							
9	Save All	Ctrl+Shift+S						
	File Properties							
	Create / Update	+						
	Export							
	Convert Programming Files							
m	Page Setup							
ā.	Print Pre <u>v</u> iew							
	Print	Ctrl+P						
	Recent Files	•	••••		+ دار د			
	Recent Projects	•	· V IP	Comp	onents			
					₽ ₽ ×			
	E <u>x</u> it	Alt+F4		•	Customize			
	1	Task		Û				
	4 🕨 Compile Design							
	👂 🕨 Analysis & S	ynthesis						
	👂 🕨 Fitter (Place	& Route)						
	Assembler (0)	Generate programm	ing files)					
	TimeQuest T	ïming Analysis				_		
	EDA Netlist \	Writer						
	Program Device	(Open Programmer))					
1								

Figure 1-35 Select File -> New





Figure 1-36 Add new Verilog HDL File

33. Select **Verilog HDL File** and click **OK** to finish creating a blank .v file, as shown in **Figure 1-37**.





Figure 1-37 A blank Verilog HDL file

34. Type in the following codes, as shown in **Figure 1-38**. The module NEEK10_QSYS comes from NEEK10_QSYS.v, which is generated in Qsys. **Figure 1-39** shows the contents of NEEK10_QSYS.v.

```
module my_first_niosii
(
    CLOCK_50,
    LEDR
);
input
               CLOCK_50;
output [9:0]
                LEDR;
NEEK10_QSYS
                 u0
    (
                                            (CLOCK_50),
      .clk_50
      . reset_reset_n
                                            (1'b1),
      . pio_led_external_connection_export
                                            (LEDR)
      );
endmodule
```



🔇 Quartus II 64-Bit - D:/my_first_niosii/	my_first_niosii - my_first_niosii	
File Edit View Project Assignments F	rocessing Tools Window Help 🐬	Search altera.com 🔇
0 🖻 🖬 🗿 🐰 🖻 🛍 🔊 (°	my_first_niosii 🔹 😵 😰 🖌 😻 😵) 🖄 🕺 🤹 🍑 🔗
Project Navigator 🛛 🖓 🗗 🗙	♦ Verilog1.v*	
۹, ×	🛶 👪 🕼 🗂 🚎 💷 🛈 🔞 🖉 👘 🖓 👪	
Entity MAX 10: 10M50DAF484C6GES my_first_niosii	<pre>1 module my_first_niosii 2 ⊟ (3</pre>	·
▲ Hierarchy ■ Files o ² ↓	7 output [9:0] LEDR; 8 9	
Tasks 4 a ×		
Flow: Compilation Customize Task Compile Design Analysis & Synthesis	<pre>11 E NEERIO_QSYS u0(12 .clk_clk 13 .pio_led_external_connection_export 14 .reset_reset_n 15); 16 17 endmodule</pre>	(CLOCK_50), (LEDR), // pic (1'b1)
Fitter (Diare & Doute)	< III	4
All All All Type ID Message	Filter>>	
System / Processing /		4
	Ln 6 Col 38 Verilog HDL File	0% 00:00:00 dt

Figure 1-38 Input verilog Text

🚱 Quartus II 64-Bit - D:/my_first_niosii/my_first_niosii - my_first_niosi		
File Edit View Project Assignments Processing Tools Window H	elp 🗟	arch altera.com 🔇
🗋 📂 属 🥔 🐰 🗈 隆 🤟 (* 🕅 my_first_niosi		
Project Navigator 🛛 🕹 🖉 🗙	Verilog1.v* 🛛 🧄 NEEK10_QSYS.v 🛛 IP Catalog	무 🗗 🗙
Q. X	🖼 🗛 🕼 📅 🕸 🙆 🖄 🖄 🕼 🖉 🖄 📖 🛶 📔 🛸 🔍 🔍	× =
Entity	1 // NEEK10_QSYS.v A Installed IP	
A MAX 10: 10M50DAF484C6GES	2 3 // Generated using ACDS version 15 0 145	actory
Imp_first_niosii 4 [™]	4 > System	
	5 `timescale 1 ps / 1 ps 4 Library	
	6 ⊟module NEEK10_QSYS (▷ Basic Fun	ictions
	8 output wire [9:0] pio led external connection	
	9 input wire reset n ▷ DSP	
	10); > Interface	Protocols
	11 L D Low Powe	ar
	12 wire [31:0] cpu_data_master_readdata;	interfaces and Controllers
	14 wire cou data master debugaccess	rs and Peripherals
🔺 Hierarchy 📄 Files 🖓 Design Units 📉 IP Components 🕢	15 wire [13:0] cpu data master address; b sis	
Tasks P & ×	16 wire [3:0] cpu_data_master_byteenable; bilinversib	v Program
	17 wire cpu_data_master_read;	y Program
Flow: Compilation Customize	18 wire cpu_data_master_readdatavalid;	ir uier 1P
Task Ó	19 wire cpu_data_master_write;	
	20 wire [31:0] cpu_data_master_writedata;	
A Compile Design	22 wire cpu instruction master waitreguest;	
Analysis & Synthesis	23 wire [13:0] cpu instruction master address;	
Fitter (Place & Route)	24 wire cpu instruction master read;	
Assembler (Generate programming files)	25 wire cpu_instruction_master_readdatavali	
TimeQuest Timing Analysis	26 wire mm_interconnect_0_jtag_uart_avalon_	
EDA Netlist Writer	28 wire mm interconnect 0 itag wart avalon	
Program Device (Open Programmer)	29 wire [0:0] mm interconnect 0 jtag uart avalon v	
	< III.) Add	
🗗 🗚 🙆 🛆 🎿 💎 < <filter>></filter>	沿 Eind I 許 Find Next	
P		
Type ID Message		
<u>8</u>		
		4
System / Processing /		
		100% 00:02:09

Figure 1-39 The module NEEK10_QSYS generated in Qsys



35. Choose **File** -> **Save** and save the Verilog HDL file as my_first_niosii.v, as shown in **Figure** 1-40.

Save As					×
Save in:	👠 my_first_niosii		▼ ← 🗈 💣 🖩	•	
Recent Places Desktop Libraries Computer Network	Name .qsys_edit db NEEK10_QSYS	5			D. 01 01
	✓ File name: Save as type:	my_first_niosii.v Verilog HDL Files (*.v *.vlg *.ve ✓ Add file to current project	erilog)	•	Save Cancel

Figure 1-40 Save the Verilog HDL file

36. Choose **Project-> Add/Remove Files** in Project and select NEEK10_QSYS.qip, as shown in **Figure 1-40**. Click **Add** and then **OK**, as shown in **Figure 1-40**.



Select File)isk (D:)	→ my_first_niosii → NEEK10_QSYS → synthesis →	▼ 🍫 Search s	synthesis
Organize • New	folder			
🝌 Downloads	*	Name	Date modified	Туре
🐉 Recent Places		🐌 submodules	01/07/2015 16:35	File folder
🕾 Librarias		NEEK10_QSYS.qip	01/07/2015 15:07	QIP File
Libraries Documents Jubraries Music Pictures Videos Ju雷下载 Computer Local Disk (C:) √ Local Disk (D:)	III	NEEK10_QSYS.v	01/07/2015 15:07	Text Document
🛷 Local Disk (E:)	- 4	m		b
Fil	le name	e: NEEK10_QSYS.qip	 Design Files (*.tdf * Open 	.vhd *.vhc ▼ Cancel

Figure 1-41 Choose NEEK10_QSYS.qsys

General	Files						
Files Libraries IP Settings	Select the d project dire	esign file ctory to t	s you want to in he project.	clude in th	he project. Click Add All to add	all design	files in the
IP Catalog Search Locations Design Templates	File name:	NEEK10	_QSYS/synthesis	/NEEK10_	_QSYS.qip	(Add
Operating Settings and Conditic	File Name		Туре	Library	Design Entry/Synthesis Tool	HD	Add All
Temperature Compilation Process Settings	my_first	_niosii.v	Verilog HDL File		<none></none>	Def	Remove
Incremental Compilation EDA Tool Settings							Up
Simulation							Down
Board-Level							Properties
Compiler Settings VHDL Input Verilog HDL Input Default Parameters TimeQuest Timing Analyzer							
Assembler Design Assistant							
Logic Analyzer Interface PowerPlay Power Analyzer Setti SSN Analyzer							
	•			111		•	





MAX10 Neek My Niosll Manual 37. Choose **Processing > Start Compilation**, as shown in **Figure 1-43**. **Figure 1-44** shows the compilation is successful.



Figure 1-43 Start Compilation



Figure 1-44 Project is compiled successfully

39. Choose Assignments -> Pin Planner to open the pin planner, as shown in Figure 1-45. Figure



MAX10 Neek My Niosll Manual

💱 Quartus II 64-Bit - D:	/my_	first_nios	sii/	my_fir	st_i	niosii -	my_first_	niosii	
File Edit View Project	Assig	gnments	P	rocessi	ng	Tools	Window	Help	7
Project Navigator	י יצי ו	Device Settings					Ctrl+Shift+E	hift+E	
	Assignmen			Editor			Ctrl+S	hift+A hift+N	
Er Armove As MAX 10: 10M50DAF484 Max my_first_niosii Max Annot Import Ass Export Assi Assignmen				Assignments otate Assignments ssignments ssignments ent Groups					liosii
	え 赤	LogicLock Design Pa	c Ro arti	Regions Window			Alt+L Alt+D		LEDR;
 ✓ Ⅲ ▲ Hierarchy	S	•	•	10 11 12 13 14 15		NEEK	10_QSY .clk_ .pio_ .rese	S u0(clk led_e t_res	_ xternal_c et_n

Figure 1-45 Open the Pin Planner

Named: *	🕶 💨 Edit: 🗶 🕔	/					Filter: Pins:	5
Node Name	Direction	Location	I/O Bank	VREF Group	itter Locatio	[/O Standard	Reserved	1
B- CLOCK_50	Input				PIN_M8	2.5 Vfault)		1
** LEDR[9]	Output				PIN_E10	2.5 Vfault)		1
25 LEDR[8]	Output				PIN_P13	2.5 Vfault)		1
** LEDR[7]	Output				PIN_R13	2.5 Vfault)		1
45 LEDR[6]	Output				PIN_P22	2.5 Vfault)		1
LEDR[5]	Output				PIN_U22	2.5 Vfault)		1
*** LEDR[4]	Output				PIN_W14	2.5 Vfault)		1
** LEDR[3]	Output				PIN_R22	2.5 Vfault)		1
25 LEDR[2]	Output				PIN_AA14	2.5 Vfault)		1
" LEDR[1]	Output				PIN_AB14	2.5 Vfault)		1
Section 12 Section 23 Section 23 Section 23 Section 23 Section 24	Output				PIN_U21	2.5 Vfault)		1

Figure 1-46 Blank Pins

40. Input Location and IO Standard value for these plan pins, as shown in Figure 1-47.

Named: * 🔹	🖏 Edit: 🗶	I					
Node Name	Direction	Location	I/O Bank	VREF Group	itter Location	[/O Standard	Reserved
B- CLOCK_50	Input	PIN_N5	2	B2_N0	PIN_M8	3.3-V LVTTL	
25 LEDR[9]	Output	PIN_D5	8	B8_N0	PIN_E10	3.3-V LVTTL	
25 LEDR[8]	Output	PIN_C5	8	B8_N0	PIN_P13	3.3-V LVTTL	
25 LEDR[7]	Output	PIN_B5	8	B8_N0	PIN_R13	3.3-V LVTTL	
25 LEDR[6]	Output	PIN_C4	8	B8_N0	PIN_P22	3.3-V LVTTL	
25 LEDR[5]	Output	PIN_B4	8	B8_N0	PIN_U22	3.3-V LVTTL	
25 LEDR[4]	Output	PIN_A4	8	B8_N0	PIN_W14	3.3-V LVTTL	
25 LEDR[3]	Output	PIN_C3	8	B8_N0	PIN_R22	3.3-V LVTTL	
25 LEDR[2]	Output	PIN_A3	8	B8_N0	PIN_AA14	3.3-V LVTTL	
25 LEDR[1]	Output	PIN_B3	8	B8_N0	PIN_AB14	3.3-V LVTTL	
25 LEDR[0]	Output	PIN_C2	8	B8_N0	PIN_U21	3.3-V LVTTL	

Figure 1-47 Set Pins

41. Close the **pin planner** and re-compile the project.



1.3 Download Hardware Design to MAX10 NEEK Board

This section describes how to download the configuration file i.e the SRAM Object File (.sof) which contains the Nios II standard system to the board. The steps are:

1. Connect the board to the host PC via the USB download cable.

2. Connect the power adaptor to the board and turn it on.

3. Choose Tools-> Programmer in Quartus II.

4. Click **Hardware Setup** on the top left comer of the Quartus II programmer window and a Hardware Setup dialog box will appear.

5. Select NEEK10[USB-1] from the Currently selected hardware drop-down list box, as shown in **Figure 1-48**. Click Close.

Note: If the hardware NEEK10[USB-1] does not appear in the list, it is likely the driver has not been installed. Please refer to Quartus II Help on how to install the driver.

> Hardware Setup			×							
Hardware Settings JTAG	Settings									
Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.										
Currently selected hardware: NEEK10 [USB-1] Available hardware items No Hardware INEEK10 [USB-1] NEEK10 [USB-1]										
Hardware	Server	Add Hardware								
NEEK10	Local	JSB-1	Remove Hardware							
			Close							





MAX10 Neek My Niosll Manual

- 6. Click Auto Detect on the right column and the device onboard should be detected automatically.
- 7. Choose 10M50DAES and click OK, as shown in Figure 1-49.



Figure 1-49 Select Device

8. Right-click on the device and select Change File, as shown in Figure 1-50.

Programmer - F:/SVN/neek10/cd/system-cd/Demonstrations/my_first_niosii/NEEK10_golden_top - NEEK10_u											
File Edit View	Processing Tools	Windo	w He	p 🛡							Sea
Hardware S	etup NEEK10 [U	SB-1] ckground	l progra	amming wh	Mode: en available	JTAG			•	Progress	: [
Start	File	Devi	ice	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Era
Stop	<none></none>	10	Delete Select Add F	e t All file		Del Ctrl+	A				
× Delete		1	Chang Save	ge File File							
Add File			Add I Chang Delete	PS File ge IPS File e IPS File							



- 9. Browse to the project directory $\ my_{first_niosii}$.
- 10. Select the programming file i.e. myfirst_niosii.sof, as shown in Figure 1-51.



🔶 Select New P	Programming File			×
Look in:	D:\my_first_niosii\output_files	G	ə o	🧾 🗉 🔳
📕 Μy Compι	Name	Size	Туре	Date Modif
k Administra	my_first_niosii.sof	3B	sole	01/07:46
	my_first_niosii.pof	1B	pole	01/07:46
4 111 >				
File name: m	ny_first_niosii.sof			Open
Files of type: P	Programming Files (*.sof *.pof *.jam *.jbc *.ek	p *.jic)) •	Cancel

Figure 1-51 Select the programming file

12. Click the Program/Configure option, as shown in Figure 1-52.

	Processing Tool	s Window He	lp ₹							Search	n altera.com	
Hardware S	Setup NEEK10 [USB-1]		Mode:	JTAG			•	Progress	:		
Enable real-	-time ISP to allow b	ackground progr	ramming wh	en availabl	e							
Mart Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMF	
M Stop	output_files/my	. 10M50DAF4	0056DD	0056DD	V							
uto Data												
Auto Deter												
< Delete												
Add File												
nange File												
nange File Save File												
nange File Save File												
nange File Save File dd Device		484ES										
nange File Save File dd Device 1™ Up		484ES										
nange File Save File dd Device ↑™ Up		484ES										

Figure 1-52 Prepare to program myfirst_niosii.sof in Quartus II programmer



13. Click **Start** and wait for the Progress meter reaches 100%. When configuration is complete, the FPGA is configured with the Nios II system, but it does not yet have a C program in memory to execute.



Chapter 2 NIOS II Software Build Tools for Eclipse

This chapter lists the steps to build Nios II software program in C code and compile the project to run on the Nios II standard system from previous chapter on the MAX10 NEEK board. Users will also learn how to edit the project, re-build it, and setup a debug session. The Nios II Software Build Tools (SBT) for Eclipse is a graphical user interface (GUI) which 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 example software application templates included make it easy for new software programmers to get started quickly.

2.1 Create the hello_world Example Project

This section describes how to create a new NIOS II C/C++ application project based on an example which comes with the installation of Nios II SBT. Please follow the steps below in Nios II SBT for Eclipse:

1. Choose Tools->Nios II Software Build Tools for Eclipse in QuartusII software.

2. Select a workspace and click **OK**, as shown in **Figure 2-1**.



🈂 Workspace	Launcher					
Select a workspace						
Eclipse store Choose a wo	s your projects in a folder called a workspace. orkspace folder to use for this session.					
Workspace:	D:\my_first_niosii\software	▼ Browse				
_						
Use this a	is the default and do not ask again					
		OK Cancel				

Figure 2-1 Select a workspace in Nios II SBT

3. Choose **File**->**New**->**NIOS II Application and BSP from Template** from Nios II SBT to open the wizard for New Project .

- 4. There are three actions to be performed, as shown in Figure 2-2.
 - •Browse to the target hardware system NEEK10_QSYS.sopcinfo under Target hardware information.
 - •Enter the project name **my_first_niosii**.
 - •Select the Hello World project template.



Nios II Application and BSP from Template
Nios II Software Examples
Create a new application and board support package based on a software example template
Target hardware information
SOPC Information File name: D:\my_first_niosii\NEEK10_QSYS.so
CPU name: cpu 🔻
Application project
Project name: my_first_niosii
✓ Use default location
Project location: D:\my_first_niosii\software\my_first_niosii
Project template
Templates Template description
Hello Freestandin A Hello World prints 'Hello from Nios II'
Hello World
✓ ■ This example runs with or without the -
? < Back Next > Finish Cancel

Figure 2-2 Select project template to start creating a new project

5. Click **Finish** and the Nios II SBT for Eclipse will create a project named **my_first_niosii**, as shown in **Figure 2-3**.





Figure 2-3 Project named my_first_niosii is created

There are two new projects created in Nios II SBT for Eclipse and shown in Poject Explorer on the right:

my_first_niosii is the C/C++ application project. This project contains the source code and header files for users' application.

■ my_first_niosii_bsp is a board support package which encapsulates the details of the Nios II system hardware.

Note: When a system library for the first time, the Nios II SBT for Eclipse will automatically generates the following files for software development:

• Installed IP device drivers, which includes SOPC component device drivers for the NIOS II hardware system.

• Newlib C library, which is a richly featured C library for the Nios II processor.

• Nios II software packages which include NIOS II hardware abstraction layer, NicheStack TCP/IP Network stack, Nios II host file system, Nios II read-only zip file system, and Micrium's μ C/OS-II



real time operating system (RTOS).

- system.h, which is a header file that encapsulates the hardware system.
- alt_sys_init.c, which is a file that initializes the devices in the system.

•my_first_niosii.elf, which is an executable and linked format file for the application located in my_first_niosii folder under Debug.

2.2 Build and Run the Program

This section you describes how to build and run the program to execute the compiled code. The steps are:

- Right-click the project my_first_niosii in Project Explorer and choose Build Project. The Build Project dialog box will appear and the Eclipse will start compiling the project.
- 2. When the compilation is successful, a message [my_first_niosii build complete] will appear in the Console window, as shown in **Figure 2-4**. The compilation time depends on users' system.





Figure 2-4 The build of my_first_niosii is complete

3. After compilation is complete, right-click the my_first_niosii project and choose Run As -> NIOS II Hardware. The Eclipse will begin to download the program to the MAX10 NEEK board. When the target hardware executes the program, a message 'Hello from Nios II!' will appear in the Nios II Console window, as shown in Figure 2-5.



Nios II - my_first_niosii/hello_world.c - Eclipse					
rile Edit Source Refactor Navigate Search Project Nios II Run Window Help					
		Quick Acc	cess 🛛 🖻 🖳 🖳 Nios II		
Project Explorer 🛛 🗖 🗖	le hello_world.c ⊠		🗄 Outline 🛛 🗖 🗖		
<u> </u>	⊛ * "Hello World" example.[💱 🖻 📬 🖉 🖋 🖷 🗰		
4 😂 my_first_niosii	and the second		\bigtriangledown		
> 🗱 Binaries	<pre>#include <stdio.h></stdio.h></pre>		stdio.h		
> 🗊 Includes	⊖int main()		main() : int		
> 🗁 obj	{				
I in the second seco	<pre>printf("Hello from Nios II!\n");</pre>				
> 🕸 my_first_niosii.elf - [alt	noturn A:				
create-this-app	}				
Là Makefile					
my_first_niosii.map					
my_first_niosii.objdum		-			
		Þ.			
> 🗁 my_first_niosii_bsp [NEEi					
	🛿 Problems 🖉 Tasks 📮 Console 🖾 Nios II Console 🖾 🔲 Pro	operties			
	my_mrst_miosi mios il mardware configuration - cable: NEEK 10 on localhost [USB-1] device ID: 1 inst Hello from Nios II!	ance ID: 0 name	:: jtaguart_0		

Figure 2-5 Niso II Console display program output

2.3 Edit and Re-Run the Program

Users can modify the **hello_world.c** program file in the Eclipse, build it, and re-run the program to observe the changes on the target board. This section shows how to make LEDR blink.

Please follow the steps below to modify and re-run the program:

1. In the hello_world.c file, add the lines of code in blue accordingly.

#include <stdio.h>

#include "system.h"

#include "altera_avalon_pio_regs.h"

int main()

{



```
printf("Hello from Nios II!\n");
```

```
int count = 0;
```

int delay;

while(1)

{

```
IOWR_ALTERA_AVALON_PIO_DATA(PIO_LED_BASE, count & 0x01);
delay = 0;
while(delay < 2000000)
{
    delay++;
}
count++;
}
return 0;
```

}

2. Save the project and re-compile the file by right-clicking **my_first_niosii** in the NIOS II C/C++ Projects tab and choosing **Run -> Run As -> Nios II Hardware**.

Note: Users do not need to build the project manually, as the Nios II Eclipse automatically re-builds the program before downloading it to the target board.

4. Orient the MAX10 NEEK board to observe LEDR blinking.

2.4 Why the LED Blinks

The Nios II system description header file, **system.h**, contains the software definitions, name, locations, base addresses, and settings for all components in the Nios II hardware system. The **system.h** file is located in the directory **my_first_niosii_bsp**, as shown in **Figure 2-6**.





Figure 2-6 The location of system.h

The key part of **system.h** file for the Nios II project example used in this tutorial is **pio_led**. This function controls the LED and the Nios II processor controls the PIO ports, which are connected to the LEDs, by reading and writing to the register map. For the PIO, there are four registers: **data**, **direction**, **interrupt mask**, **and edge capture**. The data register of PIO needs to be written to turn on and off the LEDs.

The PIO core has an associated software file **altera_avalon_pio_regs.h**. This file defines the core's register map, providing symbolic constants to access the low-level hardware.

The altera_avalon_pio_regs.h

file is located in altera\<version number>\ip\altera\sopc_builder_ip\altera_avalon_pio.

When the **altera_avalon_pio_regs.h** file is included, several useful functions become available to access the PIO core registers, especially the function

IOWR_ALTERA_AVALON_PIO_DATA (base, data)

It can write to the PIO data register to turn the LED on and off. For more information about the PIO core and other embedded peripheral cores, please refer to Quartus II Version <version> Handbook



Volume 5: Embedded Peripherals.

When developing your own designs, you can use the software functions and resources that are provided with the Nios II HAL. Refer to the Nios II Software Developer's Handbook for extensive documentation on developing your own Nios II processor-based software applications.

2.5 Setup Debug Configuration

A debug configuration which specifies how to run the software needs to be created before users can debug a project in Nios II SBT for Eclipse. The steps to setup a debug configuration are:

1. Set a breakpoint by double clicking the front of which line to be observed in **hello_world.c**, as shown in **Figure 2-7**.



Figure 2-7 Set a breakpoint

2. Right-click the application i.e. **my_first_niosii** and choose **Debug as > Nios II Hardware** to start the process.



MAX10 Neek My Niosll Manual

3. If a **Confirm Perspective Switch** message box appears, click **Yes**.

4. The main () function should appear in the editor shortly. A blue arrow next to the first line of code indicates the execution stops at the line.

5. Choose **Run-> Resume** to continue execution.

When debugging a project in the Nios II SBT for Eclipse, users can pause, stop or single step the program. Users can also set breakpoints, examine variables, or perform other common debugging tasks.

Note: To return to the Nios II C/C++ project perspective from the debug perspective, click the icon with two arrows >> on the top right corner.

2.6 Configure System Library

This section describes how to configure some advanced options about the target memory or other things. All the available settings can be changed by following the steps below:

By performing the following steps, you can change all the available settings:

1. Right-click my_first_niosii_bsp in Nios II SBT for Eclipse and choose Nios II-> BSP

Editor.

2. The Main page contains settings related to how the program interacts with the underlying hardware. The settings have names which correspond to the targeted Nios II hardware.

3. Observe which memory has been assigned for Program memory(.text), Read-only data memory(.rodata), Read/write data memory(.rwdata), Heap memory, and Stack memory in the Linker Script box, as shown in **Figure 2-7.** These settings determine which memory is used to store the compiled executable program when running the my_first_niosii program. Users can specify which interface to be used for stdio , stdin, and stderr. Users can also add and configure a RTOS for users' application and configure build options to support C++ or reduced device drivers, etc.

4. Choose onchip_memory2 for all the memory options in the Linker Script box, as shown in **Figure 2-8**.



💩 BSP Editor - sett	ings.bsp						×
File Edit Tools Help							
Main Software Packages Driv	vers Linker Script Ena	ble File Generation	Target BSP Directory				
Linker Section Mappings							
Linker Section Name		Linker Region Name	2	Memory Device Name		Add	
her beeddir tame		land in a second	-	Inching before hand		Remove	
ontru		onchip_memory2		onchip_memory2		Restore Default	s
exceptions		onchip memory2		onchip_memory2			
. heap		onchip_memory2		onchip_memory2			
. rodata		onchip_memory2		onchip_memory2			
. rwdata		onchip_memory2		onchip_memory2			
. stack		onchip_memory2		onchip_memory2			
. text		onchip_memory2		onchip_memory2		_	
Lieber Merrer Desires							
Linker Memory Regions				5 (h. h.)	0((-++)-+)	Add	
Linker Region Name	Address Range		Memory Device Name	Size (bytes)	Offset (bytes)	Remove	
onchip_memory2	0x00420020 -	0x00438FFF	onchip_memory2	102368	32	2 Destes Defail	
Grayed out entries are autor	natically created at ge	nerate time. They a	are not editable or persisted	in the BSP settings file.		Add Memory Devi Remove Memory Devi Memory Usage Memory Map.	ce avice
Information Problems Proces	ssing						
Setting "hal.linker.interrupt_	stack_memory_region_n	ame" set to "onchip_	memory2".				
Setting "hal.linker.exception	_stack_memory_region_	name" set to "onchip	_memory2".				
Loading drivers from ensemble	ole report.						
Mapped module: "cpu" to us	e the default driver versi	on.					
Mapped module: "pio_led" to	use the default driver v	ersion.					
Mapped module: "jtag_uart"	to use the default driver	version.					
I) Finished loading drivers from	ensemble report.						
Loading BSP settings from settings	ettings file.						
Finished loading SOPC Builde	r system info file "\\N	EEK10_QSYS.sopcin	fo [relative to settings file]"				-
					[Generate	Exit

Figure 2-8 Configure BSP

5. Click **Generate** then **Exit** to close the **BSP Editor** dialog box and return to the Eclipse workbench.

Note: If users make changes to the system properties or the Qsys properties or the hardware, the project must be rebuilt by right-clicking **my_first_niosii_bsp** ->**Nios II** ->**Generate BSP** -> **ReBuild Project.**



Chapter 3

Program and Boot On-chip Flash

This chapter describes how to program and boot the on-chip Flash in MAX 10 device. The Nios II soft core process is configured to execute the code from the on-chip memory using Altera On-chip Flash IP core.

3.1 Modify Project in Qsys

1. Choose Library -> Basic Functions-> On Chip Memory ->Altera On-Chip Flash, as shown in Figure 3-1. Set Configuration Mode to Single Uncompressed Image and make sure the Initialize flash content option is left unchecked, as shown in Figure 3-2. Click Finish.

📥 Qs	ys - NEEK10_(QSYS.qsys (D:\my_firs	t_nio:
File Ed	dit System G	enerate View Tools H	Help
🛃 IP	'Catalog ¤		- 🗗
			× 💽
Pro S Lih	Viect. New Component vstem rarv asic Functio Arithmetic Bridges and Clocks: PLL Configurati DMA On Chip Men • Altera • Altera	ent ns I Adaptors s and Resets on and Programming orv On-Chip Flash User Flash Memorv User Flash Memorv User Flash Memorv	fofofo
	• Avalon • Avalon • Avalon • Avalon • Avalon • Avalon • On-Chir	F1FO Memory -ST Dual Clock FIFO -ST Multi-Channel S -ST Round Robin Sch -ST Single Clock FI D Memory (RAM or RO	ha ed F0 M) -

Figure 3-1 Add Altera On-Chip Flash in Qsys



👃 Altera On-Chip Flash - onchip_f	ash_0		×
Altera On-Chip Flash			Documentation
*Block Diagram	* Parameters		
Show signals	Data interface:	Parallel 🔻	
	Read burst mode:	Incrementing -	
onchip_flash_0	Read burst count:	8 🔻	
nreset	Configuration Mode		
data avalon	Configuration Scheme:	Internal Configuration 💌	
avalon	Configuration Mode:	Single Uncompressed Image	\supset
attera_onemp_nash	Flash Memory		
	Sector ID Access Mode	Address Mapping Type	
	1 Read and write	e 0x00000 - 0x0 UFM	=
	2 Read and write	e 0x08000 - 0x0 UFM	
	3 Read only	0x10000 - 0x6 UFM	
	NA Hidden	NA CFM	
	+ -		
	Clock Source		
	Clock frequency:	116.0 MHz	
	User is required to provide the clock fr The on-chip flash megafunction will be r	≥quency. m with 116000000.0 Hz clock frequency.	
	Flash Initialization		
	Enable non-default initializa	tion file	-
4	•	m	4
IL			Cancel Finish

Figure 3-2 Settings of Altera On-Chip Flash

2. Choose **onchip_flash_0** and rename it to **onchip_flash**. Connect its clk and nreset to the clk and clk_reset of of **clk_50**, respectively. Its data is connected to the data_master and instruction_master of cpu and its csr is connected to the data_master of the **cpu**. **Figure 3-3** shows the complete connections.





Figure 3-3 Rename onchip_flash and establish connections

3. Double click **cpu** to edit the Nios II Processor component. Change Reset vector to onchip_flash.data as shown in **Figure 3-4**.



System Contents 🛛 🖄 Parameters 🔅 Addr	ess Map 🛛 Interconnect Requirements 🌣				
System: NEEK10_QSYS Path: cpu					
Nios II Processor altera_nios2_gen2					
Main Vectors Caches and Memory Interface	s Arithmetic Instructions MMU and MPU Se				
Reset Vector					
Reset vector memory:	onchip_flash.data 🔻				
Reset vector offset:	0x0000000				
Reset vector:	0x00000000				
* Exception Vector					
Exception vector memory:	onchip_memory2.s1 💌				
Exception vector offset:	0x0000020				
Exception vector: 0x00420020					
* Fast TLB Miss Exception Vector					
Fast TLB Miss Exception vector memory:	None				
Fast TLB Miss Exception vector offset:	0x0000000				
Fast TLB Miss Exception vector:	0x0000000				

Figure 3-4 Change Reset Vector

4. Choose **System** -> **Assign Base Addresses** and click **Generate ->Generate HDL** to generate the HDL files in Qsys, as shown in **Figure 3-5**.

& Generation
* Synthesis
Synthesis files are used to compile the system in a Quartus II project.
Create HDL design files for synthesis: Verilog 🔻
Create timing and resource estimates for third-party EDA synthesis tools.
✓ Create block symbol file (.bsf)
* Simulation
The simulation model contains generated HDL files for the simulator, and may include si
Create simulation model:
Allow mixed-language simulation
Enable this if your simulator supports mixed-language simulation.
• Output Directory
Path: D:/my_first_niosii/NEEK10_QSYS
Generate Cancel

Figure 3-5 Generate HDL files in Qsys



MAX10 Neek My Niosll Manual 5. Open Quartus II and select **Assignment -> Device -> Device and Pin Options -> Configuration**. Set Configuration mode to Single Uncompressed Image (3584Kbits UFM), as shown in **Figure 3-6**. Click **OK** twice to exit the window.

General	Configuration			
Configuration Programming Files	Specify the device configuration scheme and the configuration device.			
Unused Pins Dual-Purpose Pins	Configuration scheme: Internal Configuration			
Capacitive Loading	Configuration mode: Single Uncompressed Image (3584Kbits UFM)			
Board Trace Model I/O Timing	Configuration device			
Voltage Din Discoment	Auto			
Error Detection CRC CvP Settings	Use configuration device: Device Options			
Partial Reconfiguration	Configuration device I/O voltage:			
	Force VCCIO to be compatible with configuration I/O voltage			
	Generate compressed bitstreams			
	Active serial clock source:			
	Enable input tri-state on active configuration pins in user mode			
	Description:			
	The method used to load a design into the device. Only one configuration scheme is available: Internal Configuration (use internal flash).			
	Reset			

Figure 3-6 Set the Configuration mode

6. Re-compile **my_first_niosii** project

3.2 Reconfigure Nios II BSP Editor

1. Right-click **my_first_niosii_bsp** in Nios II SBT for Eclipse and choose **Nios II ->Generate BSP**, as shown in **Figure 3-7**.



🚔 Generating BSP (my_first_niosii_bsp))		
Generating BSP (my_first_niosi	ii_bsp)		
Always run in background			
	Run in Background	Cancel	Details >>

Figure 3-7 Generate BSP

2. Right-click my_first_niosii_bsp and choose Nios II -> BSP Editor, as shown in Figure 3-8.

a BSP Editor - settings.bsp		
File Edit Tools Help		
Main Software Packages Drivers Linker Script Enable File	Generation Target BSP Directory	
SOPC Information file:\\WEEK10_QSYS.sopcinfo CPU name: cpu Operating system: Altera HAL BSP target directory: D:\my_first_niosii\software\my_first_	Version: default niosi_bsp	
Settings Common Analysis Settings Settings Sys_clk_timer Sys_clk_timer Stdin Stdout Stderr S	hal sys_clk_timer: timestamp_timer: stdin: stdin: jtag_uart ▼ stdout: stderr: itag_uart ▼ enable_small_c_library enable_gprof enable_reduced_device_drivers enable_sim_optimize hal.linker	E
	enable_exception_stack	-
Information Problems Processing Image: The second	sion.	
Mapped module: "jtag_uart" to use the default driver version Finished loading drivers from ensemble report. Loading BSP settings from settings file. Finished loading SOPC Builder system info file "\ WEEK10	n. OSYS.soocinfo Irelative to settinos file1"	Generate Exit

Figure 3-8 Choose BSP Editor

3. Go to **Settings-> Advanced ->linker** to expand the list and set the options of hal.linker according to **Figure 3-9**.





Figure 3-9 Set the options of hal.linker

4. Click on the **Linker Script** tab in Nios II BSP Editor. Set the **.text** item in the Linker Section Name to the Altera On-chip Flash in the Linker Region Name. Set the rest of the items in the Linker Section Name list to Altera On-chip RAM, as shown in **Figure 3-10**.



💩 BSP Editor - settings.bsp					- X				
File Edit Tools Help									
Main Software Packages Drivers Linker Script	t Enable File Generation Target B	SP Directory							
Linker Section Mappings									
Linker Section Name Memory Device Name Add									
. bss	onchip_memory2		onchip_memory2		Remove				
. entry	reset		onchip_flash_data		Restore Defaults				
. exceptions	onchip_memory2		onchip_memory2						
. heap	onchip_memory2		onchip_memory2						
. rodata	onchip_memory2		onchip_memory2						
.rwdata	onchip_memory2		onchip_memory2						
. stack	onchip_memory2		onchip_memory2						
, text	onchip_flash_data		onchip_flash_data						
Linker Memory Regions Linker Region Name	Address Range	Memory Device Name	Size (bytes)	Offset (bytes)	Add				
	0-00120020 - 0-00129777	lanabin nanuna?	102269	20	Remove				
onchip_memory2 BEFORE EXCEPTION	0x00120020 - 0x00138FFF	onchip_memory2	102368	32	Restore Defaults				
onchip flash data	0x00080020 - 0x000EFFFF	onchin flash data	458720	32					
reset	0x00080000 - 0x0008001F	onchip flash data	32	0	Add Memory Device				
					Add Memory Device				
					Remove Memory Device				
					Memory Usage				
					Memory Map				
]				
Graved out entries are automatically created	l at generate time. They are not e	ditable or persisted in t	he BSP settings file.						
Information Problems Processing					,				
Finished loading drivers from ensemble report.									
Loading BSP settings from settings file.									
Finished loading SOPC Builder system info file '	"\\NEEK10_QSYS.sopcinfo [relativ	e to settings file]"							
Ohanged mapped section ".text" from memory	region "onchip_memory2" to memor	y region "onchip_flash_dat	a".		+				
<u>L</u>					Generate Exit				

Figure 3-10 Setup the Linker Script

5. Click **Generate** and then **Exit**.

6. Right-click on **my_first_niosii** in Nios II SBT tool and click **Make Targets -> Build...**, as shown in **Figure 3-11**. Select **mem_init_generate** and click **Build** to generate the HEX file.



🌲 Nios II - m	ny_first_niosii/hello_world.c - Eclipse		
File Edit So	urce Refactor Navigate Search I	Project Nios II Run Wi	indow Help
11 - 11 G 2	≤ m @ ▼ @ ▼ @ ▼ ∲ ▼	• • • • • • • • • •	 I ≫ [2] ▼ ∅ ▼ ∅ ▼ ⊕ Φ ▼ ⊕ ▼ ≅
 Project Exp my my my my minimum with minim 	Image: Solution of the second state of the second stat	<pre> hello_world.c hello_world.c</pre>	<pre>id" example.[] io.h> tem.h" era_avalon_pio_regs.h" from Nios II!\n"); ; VALON_PIO_DATA(PIO_LED_BASE,</pre>
ייים איין איין איין איין איין איין איין	Import Export		2000000)
> 🧀 d > 👝 H > 🖻 al 🗞 > In lii > In sy	Build Project Clean Project Refresh Close Project Close Unrelated Projects	F5	
	Build Configurations	•	Create
l □ □ ···	Index	•	Create Build Shift+F9
l₀ n	Show in Remote Systems view		Rebuild Last Target F9

Figure 3-11 Generate the HEX file

3.3 Programming On-chip Flash

1. Click File->Convert Programming Files in Quartus II, as shown in Figure 3-12.



	Quartus II 64-Bit - D:/my_f	irst_niosii/my_fi	rst_niosii - my_first
File	Edit View Project Assig	nments Processi	ng Tools Window
□ 2	New Open Close	Ctrl+N Ctrl+O Ctrl+F4	st_niosii
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	New Project Wizard Open Project Save Project Close Project	Ctrl+J	Ma S _a († ≢ module my_ ⊟ (LE
	Save Save As Save All	Ctrl+S Ctrl+Shift+S); input output [9:
	File Properties		-
	Create / Update Export	•	B NEEK10_QS
	Convert Programming Files	•	.pio
	Page Setup Print Preview Print	Ctrl+P	.res
	Recent Files	•	endinodure
	Recent Projects	•	
	Exit	Alt+F4	
	Compile Design		1
	Analysis & Synthes	sis	
	Fitter (Place & Routher Control of the Control o	ite)	

Figure 3-12 Open the programming file to be converted

2. Set **Programming file type** to **Programmer Object File(.pof)** and **Mode** to **Internal Configuration**, as shown in **Figure 3-13**.



Convert Programmin	g File - D:/my_first_niosii/	/my_first_niosii -	my_first_niosii			×	
File Tools Window Search altera.com							
Specify the input files to o You can also import input future use. Conversion setup files	convert and the type of prog t file information from other	ramming file to g files and save the	enerate. e conversion setup information o	created here for			
Oper	n Conversion Setup Data		Save Co	nversion Setup.	•	J	
Output programming fil	e						
Programming file type:	Programmer Object File (.)	pof)			•] ≡	
Options/Boot info	Configuration device: EPC	E16	Mode:	Internal Co	nfiguration 🔹	J	
File name:	output_file.pof					J	
Advanced	Remote/Local update differ	ence file: NO	DNE		*]	
	Create Memory Map File	e (Generate outpu	ıt_file.map)				
	Create CvP files (Gener	ate output_file.pe	riph.pof and output_file.core.rb	f)			
	Create config data RPD	(Generate output	_file_auto.rpd)				
Input files to convert							
File/Data area	Properties	Start Address			Add Hex Data		
SOF Data	Page_0	<auto></auto>			Add Sof Page	-	

Figure 3-13 Settings of Convert Programming File

3. Click **Options/Boot info...**, and the dialog of MAX 10 Device Options will appear. Choose **Load memory file** for **UFM source** and browse to the generated Altera On-chip Flash HEX file (onchip_flash.hex) in the File path, as shown in **Figure 3-14.** Click **OK**.



🞬 Convert Programming File -	Max 10 Device Options	
File Tools Window	Power On Reset scheme: Instant ON	
Specify the input files to convert a You can also import input file info	Set I/O to weak pull-up prior usermode	here fi
future use.	Auto-reconfigure from secondary image when initial image fails	incre in
Conversion setup files	Use secondary image ISP data as default setting when available	<u> </u>
Open Conver	Security	n Setu
	Verify protect	
Output programming file	Allow encrypted POF only	
Programming file type: Progra	Dual Config	
Options/Boot info) Configu	Enable watchdog	ernal (
File name: output	Watch value:	
Advanced Remote	Licer Elash Memory	
Cre	IEM source: Lood memory file	
Cre		
Cre	File path: re/my_first_niosii/mem_init/onchip_flash.hex	
Input files to convert	Description:	
File/Data area	New memory file path used as UFM data	
SOF Data		
	Ok Cancel	

Figure 3-14 Set Load memory file as UFM source

4. Click **Add File** from the **Input files to convert** section and point to the generated Quartus II .sof file my_first_niosii.sof, as shown in **Figure 3-15**. Click **Generate** to create the .pof file, as shown in **Figure 3-16**.



🖼 Select Input File									
Look in:	D:\my_first_niosii\output_files	0	Ð 🖸	<u>:</u> :: =					
🔊 My Compi	Name	Size	Туре	Date Modif					
Administra	my_first_niosii.sof	ЗВ	sole	30/00:34					
File name:				Open					
Files of type: S	RAM Object Files (*.sof)		•	Cancel					

Figure 3-15 Select my_first_niosii.sof

Convert Programming	g File - D:/my_first_niosii/i	my_first_niosii - my_first_r	iiosii		- 0	X
File Tools Window				Sear	ch altera.com	٠
Output programming me						
Programming file type:	Programmer Object File (.p	oof)			•	
Options/Boot info	Configuration device: EPCE	E16	Mode:	Internal Configur	ation 🔹	
File name:	output_file.pof					
Advanced	Remote/Local update differe	ence file: NONE			.	
	Create Memory Map File Create CvP files (Genera Create config data RPD	e (Generate output_file.map) ate output_file.periph.pof and (Generate output_file_auto.r	1 output_file.core.rbf) pd)			
Input files to convert						
File/Data area	Properties	Start Address			Add Hex Data	
SOF Data my first niosii.sof	Page_0 10M50DAF484ES	<auto></auto>			Add Sof Page	
					Add File	=
					Remove	
					Up	
					Down	
					Properties	
			Generate	e Close	Help	
						at

Figure 3-16 Click Generate to create the .pof file

5. Choose Tools> Programmer in Quartus II and click Hardware Setup. Choose Neek10[USB-1]



and click Add File. Choose output_file.pof, as shown in Figure 3-17.



Figure 3-17 Choose output_file.pof file

6. Check the **Program/Configure** option in Programmer for output_file.pof .Click **Start** to begin the programming, as shown in **Figure 3-18**.



👋 Programme	er - D:/my_first_nio	osii/my_first_nios	ii - my_fin	st_niosii -	[Chain2.co	lf]*						
File Edit View Processing Tools Window Help 🕏								Search	Search altera.com			
Hardware S	Setup NEEK10 [USB-1]		Mode:	JTAG			•	Progress	: 1	.00% (S	uccessful)
Enable real	I-time ISP to allow b	ackground progra	mming wh	en available	5							
🎤 Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	isp Clamf	IPS File
Stop	output_file.pof CFM0	10M50DAF4	0213F909	0000000	✓ ✓							
Auto Dete	UFM				V							
× Delete	•			1	1							•
Add File												
Save File		×										
Add Device	10M50DAF	484ES										
1 th Up	(1 50											
J [™] Down												

Figure 3-18 Program output_file.pof

7. Turn off the board and turn it on again. The LEDs should be blinking.

