SISTEMI EMBEDDED AA 2012/2013

Building a Nios II processor from scratch

Introduction

- Problem:
 - Build a (NIOS II) system tailored to application needs
- Solutions:
 - Use library cores and custom Verilog code
 - Use specific design tools (SOPC Builder/Qsys) with library of CPUs and peripherals that can be connected through the Avalon System Fabric
- DE2_basic and DE2_media are examples
 - Pre-built Nios_systems
 - Different requirements (basic system, media system)
 - Different peripherals

Tutorial: First Nios System

• Handles switches and leds through PIO peripherals



First Nios System units

- CPU (simplest, *i.e.*, economy variant) with JTAG Debug Module
- 2 PIOs
 - Input for switches (8 bit)
 - Output for driving LEDs (8 bit)
- On-chip memory for program and data (8 kB)
- System ID Peripheral for CPU identification (must be named *sysid*)

Nios II Hardware Flow



SoPC Builder Flow



Guided example (1)

- Create a new project in Quartus II
- Launch SoPC Builder
- Define the Nios_system components
 - CPU
 - Choose the simplest (Nios_II_e with debug) among the 3 options; keep the Level 1 JTAG Debug Module
 - PIO_0
 - Output for driving LEDS
 - PIO_1
 - Input for reading switches status
 - System ID Peripheral (ID = 1!)

SOPC builder library

😃 Altera SOPC Builder					
File Edit Module System View To	ioola Help				
System Contents System Generation					
Component Librory	Target	Clock Settings			
Project	Device Family. Cyclone I 🔽	Name	Source	MHz	Add
- 🛄 New component		cik_0	External	50.0	Remove
Eibrary Avsion Verification Sulle					
Bridges and Adapters					
Interface Protocols		Description	Charle.	Dates	L Fed
Herrorian and Memory Controllers	Dise G Module	Description	CIDOK	Dese	End
🖶 Merin Components					
Peripherals					
Processor Additions					
Processors					
B-SLS					
He wided and mage Processing					
< 11 >					
					~
					>
New Edit Add	Remove Edt	🔺 💌 🗶 🔺 Addh	ess Map Fitters	Filter, Default	
	Exit Help	🖣 Prev 🛛 Next 🕨	Generate		

CPU choice

- Choose the processor core most suited
- 3 variants:
 - Economy
 - Standard
 - Fast
- Different features
 - Draw-back performancecost

Thos in Processor					
Magacana NIOS	II Processor				About Documentation
arameter iettings					
Core Nos II 🔪 Cach	nes and Memory Interfaces	🔪 Advanced Features >	MMU and MPU Settings > 🗄	ITAG Debug Module	> Custom Instructions
ore Nias II					
elect a Nios II core:					
	● Nios II/e	ONios II/s	ONios II/f		
Nios II Selector Guide Family: Cyclone II f _{system} : 50.0 MHz opuid: 0	RISC 32-bit	FI9C 32-kit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide	RSC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide Barrel Shifter Data Cache Dynamic Branch Predicti	on	
erformance at 50.0 MH	tz Up to 5 DMIPS	Lip to 25 DMIPS	Up to 51 DMPS		
.ogic Usage	600-700 LEs	1200-1400 LEs	1400-1800 LEs		
Memory Usage	Two M4Ka (prieguivi)	Two M4Ks + cache	Three M4Ks + cache		
fardware Multiply: Emb	edded Multipliers	Hardware Divide			
Exception Mertor: Merr	000				
Dinclude MMU Only include the MMU w Fest TLB Miss Exception	han using an operating system Vector: Memory:	n that explicitly supports an Mh	1J Offset 0x0		
include MPU					
🚹 Warning Reset vecto	or and Exception vector canni	of be set until memory devices :	are connected to the Nos II proc	e330f	
				Cance	< Back Next > Fin

Additional peripherals

PIO (Parallel I/O) -	pio_0 🔀
PIO (Pa Mogeters) atters_avalur	rallel I/O)
* Block Diagram	
clock reset avaion conduit	pio_0 cik reset st external_connection
* Basic Settings	
Width (1-32 bits):	в
Direction	O Bidir
	💿 lingvil
	🔿 inOut
	Output
Output Port Reset Value	D×000000000000000000000000000000000000
Output Register	
Enable individual bit	setting/clearing
🕆 Edge capture registe	ir 🗌
Synchronously capt	ure
Edge Type:	RISING 👻
Enable bit-clearing fi	or edge capture register
 Interrupt 	
🔄 Generate RQ	
IRG Type:	LEVBL V
Level: Interrupt CPU who Edge: Interrupt CPU who register is logic true, Ave	en any unwesked IO pin is logic bue at any unwesked bit in the edge-capture sligkie when synchronous capture is anabled
* Test bench wiring	
Hardwire PIO inputs	in test bench
Drive inputs to:	0x00000000000000
🕕 Inte: pio_0: PIO inputs	are not hardwired in test bench. Undefined values
< U	>
	Cancel Finish

Block Diagram	1	▼ Details
Show signals Sysid Cik reset control_slave avalon attera_avalon_sysid		System ID: 1 Time stamp: 1365584360 A unique ID is assigned every time the system is generated.

On-chip memory

- Define the organization of the on chip-memory
 - Type (ROM, RAM)
 - Size
 - Word length

On-Chip Memory (RAM)	or ROM) - onchip_memory	2_0					
MageCare On-Chip M atera_avsion_ond	On-Chip Memory (RAM or ROM) atera_avaion_onchip_memory2						
* Block Diagram							
	onchip_memory2_t clock == dk1 avalon == s1 reset == reset1	0					
Memory type							
Туре:	RAM (Writable)						
Duskport access							
Read During Write Mode:	DONT_CARE V						
Block type:	Auto 💙						
* Size							
Data width:	32 💌						
Total memory size:	4096	bytes					
Minimize memory block us	sage (may impact fmax)						
Read latency							
Slave st Latency:	1 😒						
Slave s2 Latency:	1 💌						
Memory initialization							
🔽 Initialize memory content							
Enable non-default initial:	zation file						
User created initialization file	onchip_memory2_0						
Enskie In-System Memory	r Content Editor featura						
Instance ID:	NONE						
		Cancel Finish					

Guided example (2)

- Generate the processor system (core + peripherals)
 - The Nios_system is now an usual hardware block to be used in Quartus II

Altera SOPC Builder - nios_system.sopc* (E:\Wo	rk\Teac	hing\SE\	Lab\Home\My_DE2	_First_Co	omputer\Verilog\nios_system.sopc)	.040	- 2 × 2-	5			X
<u>File Edit Module System View Tools Nios II Help</u>											
System Contents System Generation											
Component Library	Targe	et		Clock S	ettings						
	Device Family: Cyclone II 👻		Name Sc		ource	MHz	MHz		A	dd	
Project New Component Library				clk_0	Ext	ternal	50,0			Rer	move
RGBtogreyscaleconvertor Bridges	Use	Conn	Name		Description	Clock	Base	End			Tags
Uninguration & Programming	V		cpu_0 instruction_ma data_master jtag_debug_ma	ister odule	Nios II Processor Avalon Memory Mapped Master Avalon Memory Mapped Master Avalon Memory Mapped Slave	[clk] clk_0 [clk] [clk]	IR(≓ 0x4800	0 0x4fff	IRQ 31	←×	
Microcontroller Peripherals Peripherals Debug and Performance Avalor.ST Data Pattern Checke	V		 onchip_memory s1 SLIDER_SWITC s1 	ory2_0 :HES	On-Chip Memory (RAM or ROM) Avalon Memory Mapped Slave PIO (Parallel VO) Avalon Memory Mapped Slave	[clk1] clk_0 [clk] clk_0	 ox2000 ox5010 	0x3fff 0x501f			
Avalon-ST Data Pattern Genera Avalon-ST Test Pattern Checke Avalon-ST Test Pattern Checke Avalon-ST Test Pattern Genera	V V		GREEN_LEDS s1 sysid		PIO (Parallel VO) Avalon Memory Mapped Slave System ID Peripheral	[clk] clk_0 [clk]	₽° 0x5000	0x500f			
System ID Peripheral Display Microcontroller Peripherals Multiprocessor Coordination PLL Processors Nios II Processor SLS Illisionarity Drogenee Ill P	1										•
New Edit Edit Edit Edit Edit Edit Edit Filters									ŗ		
 Info: onchip_memory2_0: Memory will be initialized from onchip_memory2_0.hex Info: SLIDER_SWITCHES: PIO inputs are not hardwired in test bench. Undefined values will be read from PIO inputs during simulation. 											
Exit Help I Prev Next D Generate											

Guided example (3)

- Back to Quartus II
 - Create the root module of the project
 - Include the Nios_system module as hierarchical block (Verilog or schematic)
 - Import pin assignment from de2.qsf
 - Compile the project to make the hardware ready

Testing First Nios System

- Write a program that makes the GREEN LEDS to be controlled by the SLIDERS SWITCHES
- If successful, go back to SoPC Builder, add the JTAG-UART peripheral, regenerate the Nios system and compile the design again (top level entry does not need to be changed)
- Write a program that say Hello to the host telling him the sistem ID and timestamp
 - The file .sopcinfo needs to be updated as the Nios system has been regenerated!
- That's all folks!!