



Hardware/Software Co-Design in Programmable Logic



Embedded System Development

Challenges at Every Stage

- Specification of Hardware & Software Components
- Integration of Hardware Modules
- Integration of Software with Hardware
- Verification of Hardware Functionality
- Verification of Software
 Running on Hardware
- Meeting System
 Performance Requirements

System Definition CPU **Peripherals** Memory **User Logic** Integration Hardware Blocks **Software with Hardware** Verification Hardware Design **Software Applications System Level** Optimization **Performance**

Resource Utilization





Specification Challenges

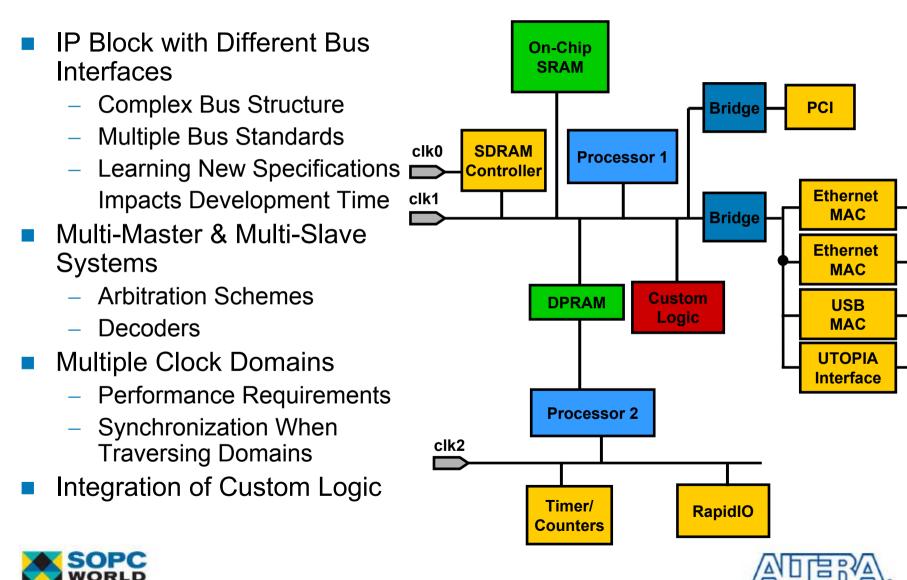
System Component Selection

- Finding Microcontroller with Exactly the Right Peripherals for Your Application
- Changing System Specification
 - Supporting Leading-Edge Emerging Standards
 Means High Potential for Changes
 - Feature Creep by Marketing Introduces New Challenges





Hardware Integration Challenges



Software Development Challenges

- Accelerating Software Development Cycle
 - Software Development Stalled Waiting for Silicon Prototype
 - Need Hardware or Model to Serve As Virtual Prototype
- Keeping Software Blocks in Sync with Hardware
 - Header Files
 - Peripheral Drivers
 - Software Libraries
 - OS/RTOS Kernels





Hardware/Software Coherency

Hardware Changes & Software Breaks

- Hardware Updates
- Bug Fixes
- Memory Map Changes
- Updating Header Files & Drivers
 - Error-Prone
 - Tedious
 - Yet Critical to Successful Design





System Verification Challenges

- Increased System Complexity
 - System Level Designs on Single Chip
- Larger Software Content
 - Software IP
 - RTOS



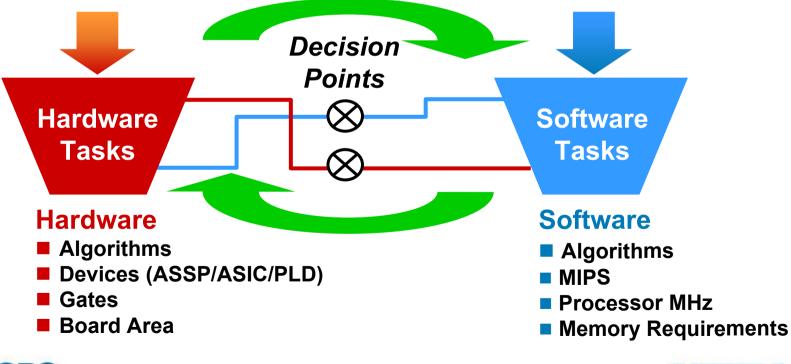
- Need Fast, Effective Way to Validate Complete System
- RTL Models for Hardware
 - Processor Hardware Models Are Slow
 - No Link to Software Debugger
- Test Bench Creation
 - Difficult to Create Hardware & Software Test Bench for IP
 - IP Expertise Lies with Vendor





Meeting System Performance

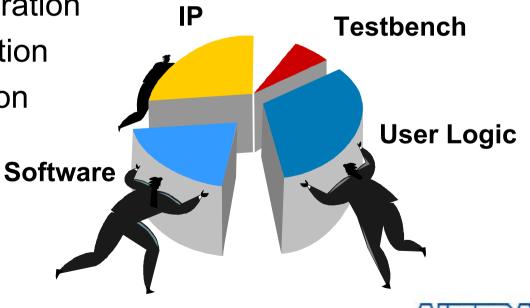
- Function Too Slow in Software?
 - Implement in Hardware Logic
- Need More Hardware Resources?
 - Implement Non-Time-Critical Functions in Software





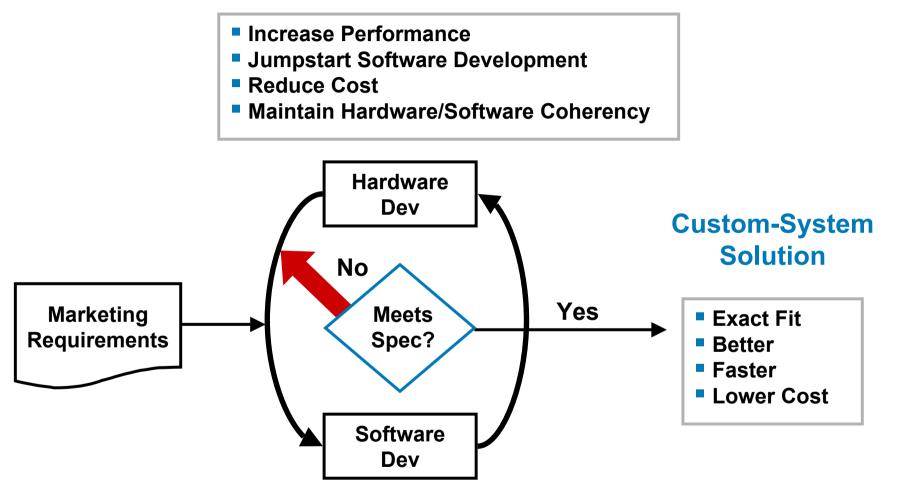
Addressing the Challenges

- Programmable Logic for Rapid Prototyping
- Excalibur[™] Embedded Processor Solutions
- SOPC Builder
 - System Customization
 - Component Integration
 - Software Generation
 - System Verification





Hardware/Software Co-Design







Embedded System Development

Challenges at Every Stage

- Specification of Hardware & Software Components
- Integration of Hardware Modules
- Integration of Software with Hardware
- Verification of Hardware Functionality
- Verification of Software
 Running on Hardware
- Meeting System
 Performance Requirements

System Definition CPU Peripherals Memory User Logic

Integration Hardware Blocks Software with Hardware

Verification Hardware Design Software Applications System Level

Optimization Performance Resource Utilization





SOPC Builder

Exact-Fit Custom Solution

- Select From Library of Peripherals
- Integrate Custom Logic

- System Changes
 - Adding New Features
 - Bug Fixes
 - Memory Map Changes

 Altera Nios 2.0 CPU User-Defined Interface Bridges Communication SPI (a Wre Serial) UART (RS-232 serial port Mito Controller / read_master (avalon) attract (avalon) attra	- 0 :							/stem		Altera SOPC Builder - nie File System Module View He		
Altera Nios 2.0 CPU User-Defined Interface Bridges Communication O SPI (3 Wire Serial) UART (RS-232 serial port O M16550S Enhanced UAR O CAN 2.0 Network Control O M16550S Enhanced UAR O CAN 2.0 Network Control O CONTON O CAN 2.0 Network Control O CONTON								neration	/stem	System Contents CPU "nios_cpu" System		
O CAN 2.0 Network Control Imits of the control interface (CS89) Imits of the				MHz	on) (avalon)	on_master (avalon) a_master (avalon) troller / read_master (aval a_controller / write_master	cpu / data dma_conti dma_	nios_cr		User-Defined Interface Bridges Communication SPI (3 Wire Serial)		
Ethernet Ethernet <li< td=""><td>IRG</td><td>End</td><td>Base</td><td>Bus Type</td><td>Description</td><td>Module Name</td><td></td><td></td><td>Us</td><td></td></li<>	IRG	End	Base	Bus Type	Description	Module Name			Us			
Ethernet Interface (CS89(Memory On-Chip Memory (RAM or SDRAM Controller SDRAM Controller SRAM (one or two IDT71: Other Other Other Other DMA PIO (Parallel I/O) Interval timer PCI O Altera PCI32 Nios Target [0000	anna	1111111	avaion	Altera Nios 2.0 CPU	🛨 nios_cpu						
Memory On-Chip Memory (RAM or SDRAM Controller SRAM (one or two IDT71 Other Other PIO (Parallel I/O) Interval timer PCI OAdtera PCI32 Nios Target [*	3FF	0×008013FF	0x00801000	avalon	On-Chip Memory (RAM	boot_rom		\bullet		-		
On-Chip Memory (RAM or SDRAM Controller Flash Memory SRAM (one or two IDT71' Other PCI PCI PCI PCI C Altera PCI32 Nios Target [*	FFF	0×00800FFF	0x00800000	avalon	On-Chip Memory (RAM	system_ram		<u> </u>				
SDRAM Controller SDRAM Controller SDRAM Controller Status	41F 16	0×0080141F	0x00801400	avalon	DMA	dma_controller	- -	↓ ↓ 				
Flash Memory SRAM (one or two IDT71') Other DMA PIO (Parallel I/O) Interval timer PCI CAtera PCI32 Nios Target [*	FFF	0×007FFFFF	0x00000000	avalon	SDRAM Controller	• sdram_controlle	+	ð é ó é				
SRAM (one or two IDT71 Cother SRAM (one or two IDT71 Cother SRAM (one or two IDT71 SRAM SRA	000	anna	and a second	avalon avalo	Avalon Tri-State Bridge	ethernet_bridge		} + + -<				
Other O	43F 17	0×0080143F	0x00801420	avalon_tristate	Ethernet Interface (CS8	ethernet_interfa						
P PIO (Parallel I/O) Interval timer PC Altera PCI32 Nios Target I	45F 18	0x0080145F	0x00801440	avalon	UART (RS-232 serial p	- 🕀 uart0				SRAM (one or two IDT71'		
PIO (Parallel I/O) Interval timer PCI Attera PCI32 Nios Target I	47F 19	0×0080147F	0x00801460	avaion	UART (RS-232 serial p	🛨 uart1		} + + +				
Add Import Move Up Move Down				n	ve Up Move Down	Mo				PIO (Parallel I/O) Interval timer PC(Altera PCI32 Nios Target I		







- Include Custom Logic
 - User-Defined Peripherals

- Hardware Acceleration
 - Custom Instruction

Ontional FIEO Memory Other Logic

Interface to User Logic - user_defined_interface_0	optional i ii o, meniory, other Logic
Ports Instantiation Timing Publish	Nios Processor
Bus Interface Type: Avalon Slave	
✓ Import Verilog, VHDL, EDIF, or Quartus Schematic File ▲dd	B Custom
Delete Billion Contraction Con	
Port Information Port Name ARM_Stripe_ads_sdk ARM_Stripe_sdk db excal_system_sim ahb_avalon_bridge_0.v	A Nios ALU
ARM_Stripe.v File name: Populat Files of type: Accepted HDL files Abort file chooser dialog	
AHB Slave's Addressable Space Address span: 0x100000000 Bits: 32 Use Device as Memory	
Cancel < Prev Next > Add to System Add to Library	





J

Embedded System Development

Challenges at Every Stage

- Specification of Hardware & Software Components
- Integration of Hardware Modules
- Integration of Software with Hardware
- Verification of Hardware Functionality
- Verification of Software
 Running on Hardware
- Meeting System
 Performance Requirements

System Definition CPU Peripherals Memory User Logic

Integration Hardware Blocks Software with Hardware

Verification Hardware Design Software Applications System Level

Optimization Performance Resource Utilization





Complex Bus Architecture

- AHB Master Accessing Non-AHB Peripheral
 - ARM[®] AHB Master
 - Avalon Slave
- Bridge Required



- Interfacing between Bus Standard Requires
 In-depth Knowledge of Both
- Less than Optimal Bridge Logic Results in Extra Latency
- Maintain Support for Both Standards
 Transactions





SOPC Builder Demo

Bridging Bus Standards

	tings	System	Generation	u <u> </u>		
Interface to User Logic	Í				System Clock Frequency: 3	3.333
valon Modules			M_Stripe(AHB) alon_bridge_0(a		
O Altera Nios 2.0 CPU Bridges		. . [-	
Avalon To AHB Bridge	Use			lodule Name	Description	
Avalon Tri-State Bridge				_Stripe	ARM-based Excalibur St	
AHB To Avalon Bridge		- - +	— <u></u>		SPI (3 Wire Serial)	
		┝╋╌┝┟	— 🕀 ahb_	avalon_bridge	e AHB To Avalon Bridge	A
SPI (3 Wire Serial)						
 UART (RS-232 serial port) 						
 O M16550S Enhanced UART - 						
CAN 2.0 Network Controller	1					
O I2C Bus Interface - Mentor In						
-Ethernet						
0 10/100 Ethernet MAC - Alcal						
○ Nios Ethernet Development k						
-Memory						
On-Chip Memory (RAM or R						
SDRAM Controller						
 SSRAM (Micron MT58L256) 						
Flash Memory						
● SRAM (one or two IDT71V0						
Other						

- Bus Connection Patch Panel
- Multi-Master Bus

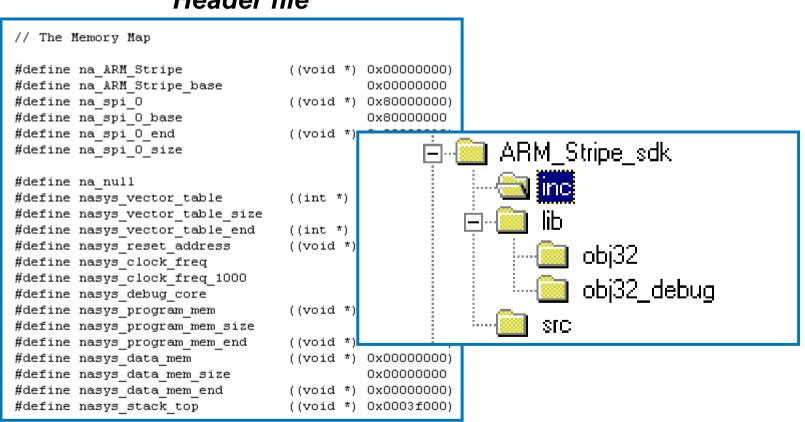
- Slave-Side Arbitration
- Optimized for Throughput
- Bus Bridging
 - AMBA[™] Advanced High-Performance Bus (AHB)
 - Avalon Bus
 - Atlantic[™] Interface
 - PCI
 - More to Follow . . .





Automatic Software Generation

Software Always Matches Target System



Header file





Embedded System Development

Challenges at Every Stage

- Specification of Hardware & Software Components
- Integration of Hardware Modules
- Integration of Software with Hardware
- Verification of Hardware Functionality
- Verification of Software
 Running on Hardware
- Meeting System
 Performance Requirements

System Definition CPU Peripherals Memory User Logic

Integration Hardware Blocks Software with Hardware

Verification Hardware Design Software Applications System Level

Optimization Performance Resource Utilization





Simulation Test-Bench Creation

- Automatically Creates Complete Testbench
- Initializes System
- Sets up Simulation Environment

Itera SOPC Builder - excal_system						
<u>F</u> ile System <u>M</u> odule ⊻iew <u>H</u> elp						
System Contents More "ARM_Stripe" Settings System Generation						
Options						
SDK. Generate header and library files for CPU(s) and peripherals in your system.						
HDL. Generate bus and system logic in Verilog for the Excalibur verilog device family.						
Synthesis. Run Leonardo Spectrum(tm) to generate an EDF file optimized for area 💌 .						
Simulation. Create ModelSim(tm) project files. Run ModelSim						
Run ended On Tue Aug 15 15.15.22 Pacific Daylight Hint 2017						
LeonardoSpectrum Level 1 run successfully completed. Goodbye !						
# 2002.08.13 13:13:23 (*) Spectrum Done. # 2002.08.13 13:13:23 (*) Starting generation for system: excal_system.						
# 2002.08.13 13:13:23 (*) THE FOLLOWING SYSTEM ITEMS HAVE BEEN GENERATED:						
ARM Stripe include files such as memory maps : d:/sopc_seminars_2002/demo/ARM_Stripe_ads_sdk/inc/						
ARM_Stripe library files : d:/sopc_seminars_2002/demo/ARM_Stripe_ads_sdk/lib/						
ARM_Stripe example programs : d:/sopc_seminars_2002/demo/ARM_Stripe_ads_sdk/src/						
ARM_Stripe include files such as memory maps : d:/sopc_seminars_2002/demo/ARM_Stripe_sdk/inc/						
ARM_Stripe library files : d:/sopc_seminars_2002/demo/ARM_Stripe_sdk/lib/						
ARM_Stripe example programs : d:/sopc_seminars_2002/demo/ARM_Stripe_sdk/src/						
Synthesis Output : d:/sopc_seminars_2002/demo/excal_system.edf						
SOPC Builder database : d:/sopc_seminars_2002/demo/excal_system.ptf System HDL Model : d:/sopc_seminars_2002/demo/excal_system.v						
System HDL Model: d./sopc_seminars_2002/demo/excal_system.v						
Synthesis Command File : d:/sopc_seminars_2002/demo/excal_system_generation_script						
Synthesis Command File : dt/sopc_seminars_2002/demo/excal_system_leonardo_commands.cmd ModelSim Simulation Directory : dt/sopc_seminars_2002/demo/excal_system_sim						
# 2002.08.13 13:13:23 (*) SUCCESS: SYSTEM GENERATION COMPLETED.						
Press 'Exit' to exit.						





Debugging

- Correlating Software & Hardware Debug Is Challenging
- Software Returns Incorrect Result after Reading Peripheral Register
- Error Could Be in...
 - C Code
 - Memory Map
 - Hardware Address Decoding
 - Hardware Peripheral Logic
 - Read/Write Access Timing for Peripheral





Co-Simulation

Gain Visibility Into Hardware at Software Breakpoint
 Enables You to Evaluate Required Changes in Hardware

or Software C dhrystone.c - Source Window _ 🗆 × 🖶 wave - default - 🗆 × File Run View Control Preferences Help Cursor Zoom Format Window Edit 🦄 🕐 🕐 🗘 😗 🐨 👗 🔍 🚍 🔗 🛔 📲 📟 0x3a0 278 0000 - EL EÌ EL 🕅 ≣Ŧ 267 for (i = 0; i < LOOPS; ++i): 268 times(&tms); /retrustave_test/cll 269 nulltime = tms.tms utime - starttime; /* Computes overheady of looping * vslave_test/finished 270 #endif ave test/resetn 271 272 PtrGlbNext = (RecordPtr) malloc(sizeof(RecordType)); est/HCLK Software **Stripe Simulation** Logic Simulator Debugger Model ս սսբ 281 -- Start Timer -e test/HSELRetry ave test/HREADY 283 #ifdef TIME slave test/HRDATA (00000000 Οτοοοοοία 284 starttime = time(0); **-**-285 #endif /retryslave_test/HRESP 286 #ifdef TIMES /retruslave_test/HBUSREQ 287 times(&tms); starttime = tms.tms utime; 288 #endif 289 for (i = 0; i < LOOPS; ++i)100 1 270 ps Program • View Waveforms Full Software Debug Control dhru Opsito 650 ps Access to Stripe Registers





Embedded System Development

Challenges at Every Stage

- Specification of Hardware & Software Components
- Integration of Hardware Modules
- Integration of Software with Hardware
- Verification of Hardware Functionality
- Verification of Software
 Running on Hardware
- Meeting System
 Performance Requirements

System Definition CPU Peripherals Memory User Logic

Integration Hardware Blocks Software with Hardware

Verification Hardware Design Software Applications System Level

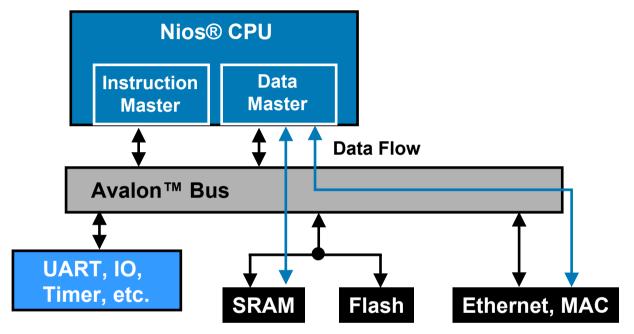
Optimization Performance Resource Utilization





Performance Optimization Example

Web Server Application Using Embedded Processor



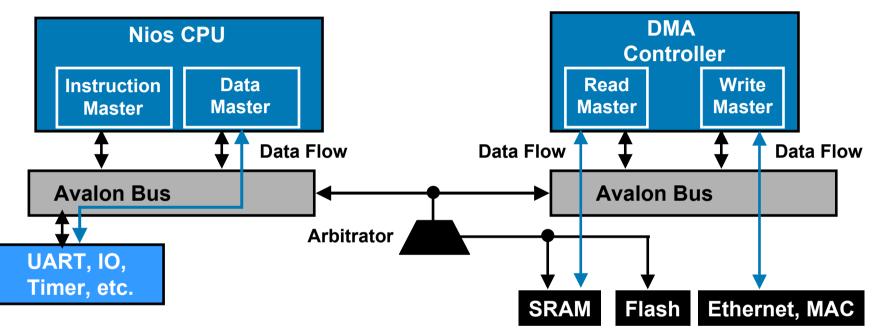
File Size (bytes)	Transmission Latency (ms)	Transmission Throughput (Mbps)	HTTP Server Latency (Ms)	HTTP Server Throughput (Mbps)
14,650	38	3.08	57	2.06
37,448	96	3.12	142	2.10
60,036	153	3.14	226	2.12





Optimization 1: DMA Controller

Adding DMA Controller in FPGA Boosts Throughput 250%



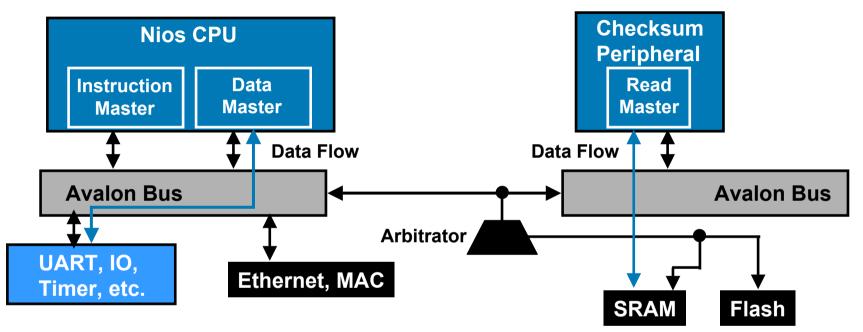
File Size (bytes)	Transmission Latency (ms)	Transmission Throughput (Mbps)	HTTP Server Latency (ms)	HTTP Server Throughput (Mbps)
14,650	18	6.50	24	4.88
37,448	44	6.81	58	5.16
60,036	71	6.76	94	5.10





Optimization 2: Checksum Peripheral

Moving Checksum Calculation to Hardware Boosts Throughput 25%



File Size (bytes)	Transmission Latency (ms)	Transmission Throughput (Mbps)	HTTP server Latency (ms)	HTTP Server Throughput (Mbps)
14,650	27	4.37	45	2.60
37,448	68	4.40	114	2.62
60,036	109	4.40	181	2.65





Combined Optimization Improvement

- Addition of Both DMA Controller & Checksum Peripheral Increase Server Throughput 280%
- FPGA Logic Element Usage Increased only 65%
- No Board Re-Spin Required

(bytes) I	Latency (ms)	Transmission Throughput (Mbps)	HTTP server Latency (ms)	HTTP Server Throughput (Mbps)
14,650	12	9.76	20	5.86
37,448	31	9.66	49	6.11
60,036	51	9.42	79	6.07





Celoxica C-to-Hardware Tools

- Facilitates Hardware Acceleration in Embedded Processor FPGAs
- Supplies High-level Design Tools that Bridge the Gap between Software & Hardware Design
- ANSI-C Based Handel-C Language Used to Describe Hardware Behavior
- For More Information, Go to www.celoxica.com





Summary

- Programmable Logic Provides
 - Effective Platform for Hardware/Software Co-Design
 - System-Level Integration
 - Flexibility for System Customization
 - Methods for Fast, Effective Verification



