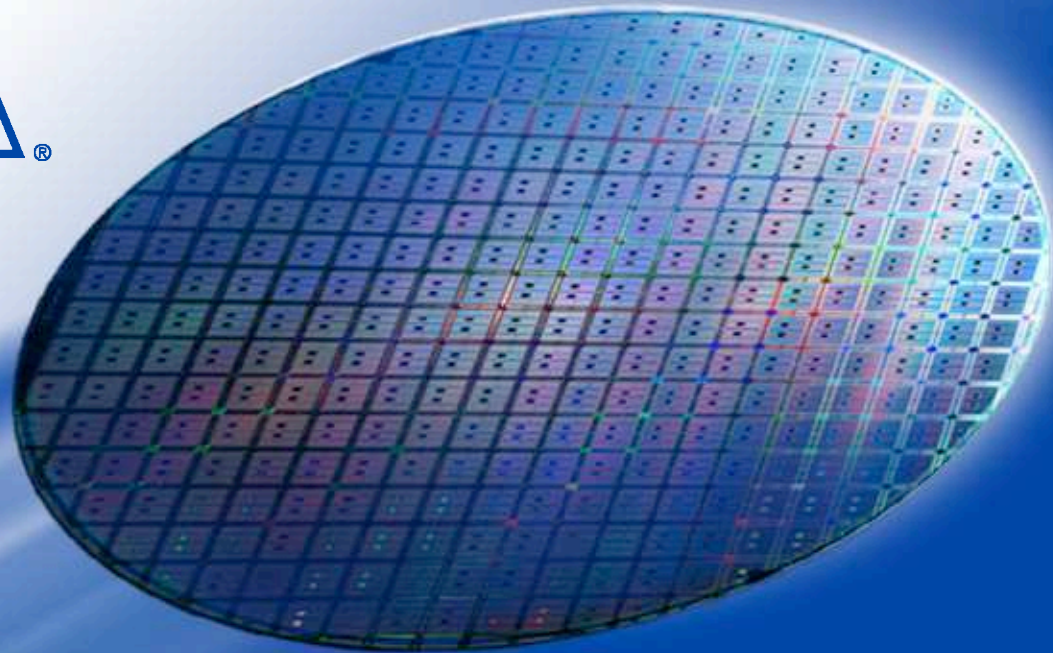


**ALTERA**<sup>®</sup>

**Nios**<sup>®</sup> **II**



# The Nios II Family of Configurable Soft-core Processors

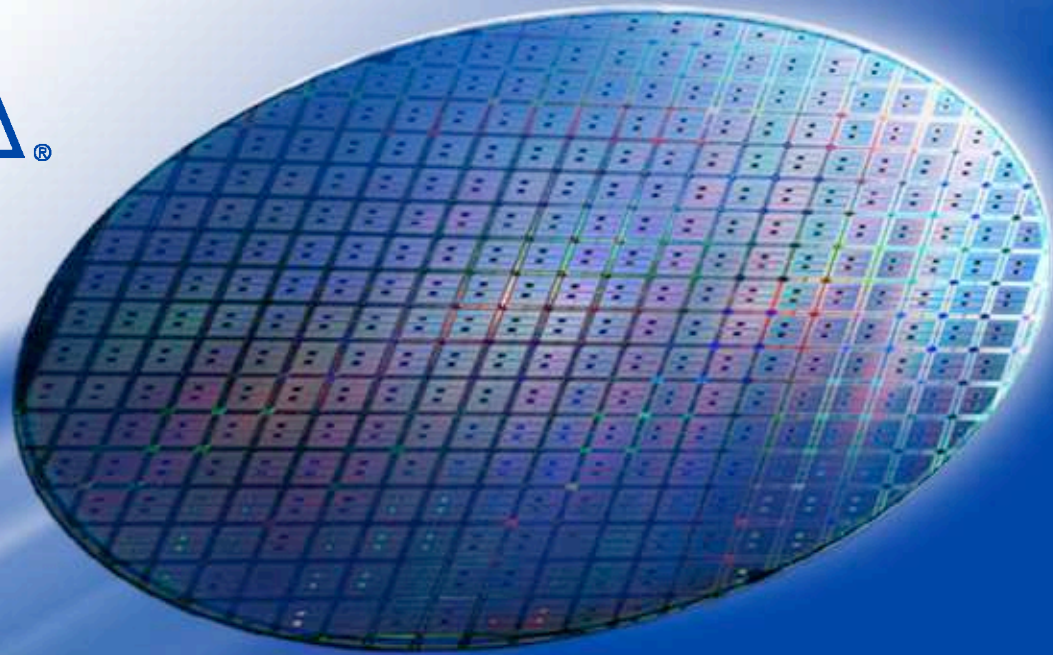
*James Ball*

*August 16, 2005*

# Agenda

- Nios II Introduction
  - Configuring your CPU
- FPGA vs. ASIC CPU Design
  - Instruction Set Architecture
  - CPU Micro-architecture
- Nios II/f CPU Description
  - Pipeline details
- Nios II Embedded Systems
  - Taking advantage of FPGA configurability

**ALTERA**®



# Nios II Introduction

# Nios II Overview

- Nios II is Altera's soft-core configurable CPU
  - Introduced summer/2004
  - New 32-bit RISC Instruction Set Architecture (ISA)
  - Replaces original 16-bit Nios
- Over 4500 active licenses
  - Most licensed embedded CPU in the world
- Designed for embedded FPGA-based systems
  - Strong performance (up to 225 Dhrystone MIPS)
  - Support for many operating systems
  - Available in all current Altera FPGAs

# Why a New Instruction Set?

## ■ Primary Issue

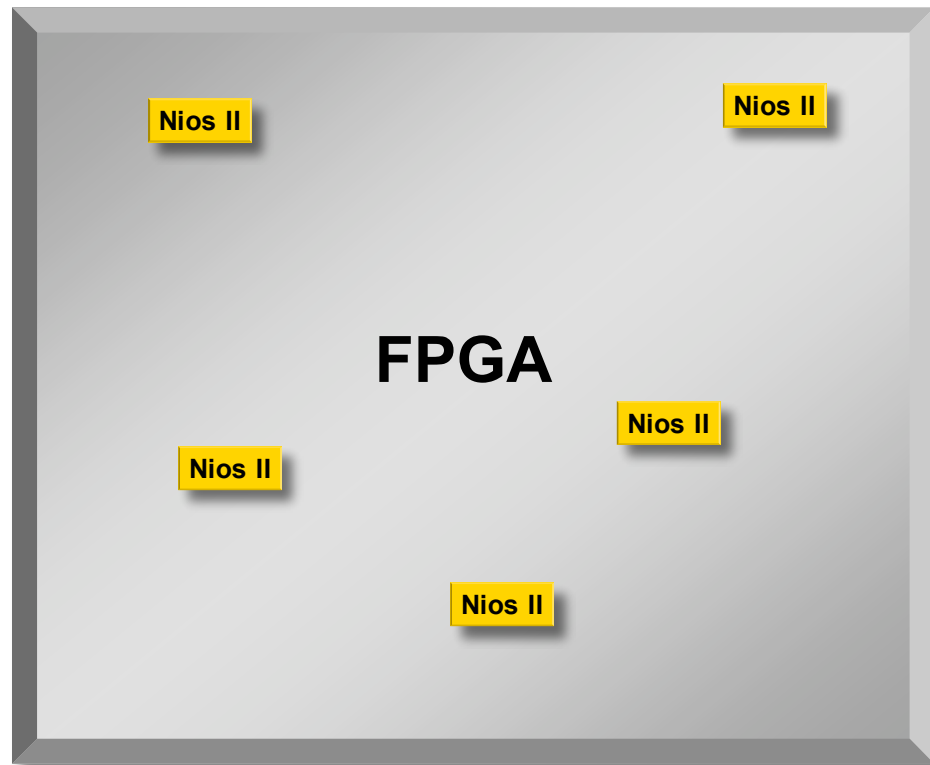
- Existing instruction sets optimized for ASIC
- Inefficient in FPGA

## ■ Secondary Issue

- Existing instruction sets have licensing restrictions

# Nios II Size

**Largest 90nm FPGA  
180,000 LUTs**



**Smallest 90nm FPGA  
4600 LUTs**



**13% of FPGA  
Nios II/e "economy"**

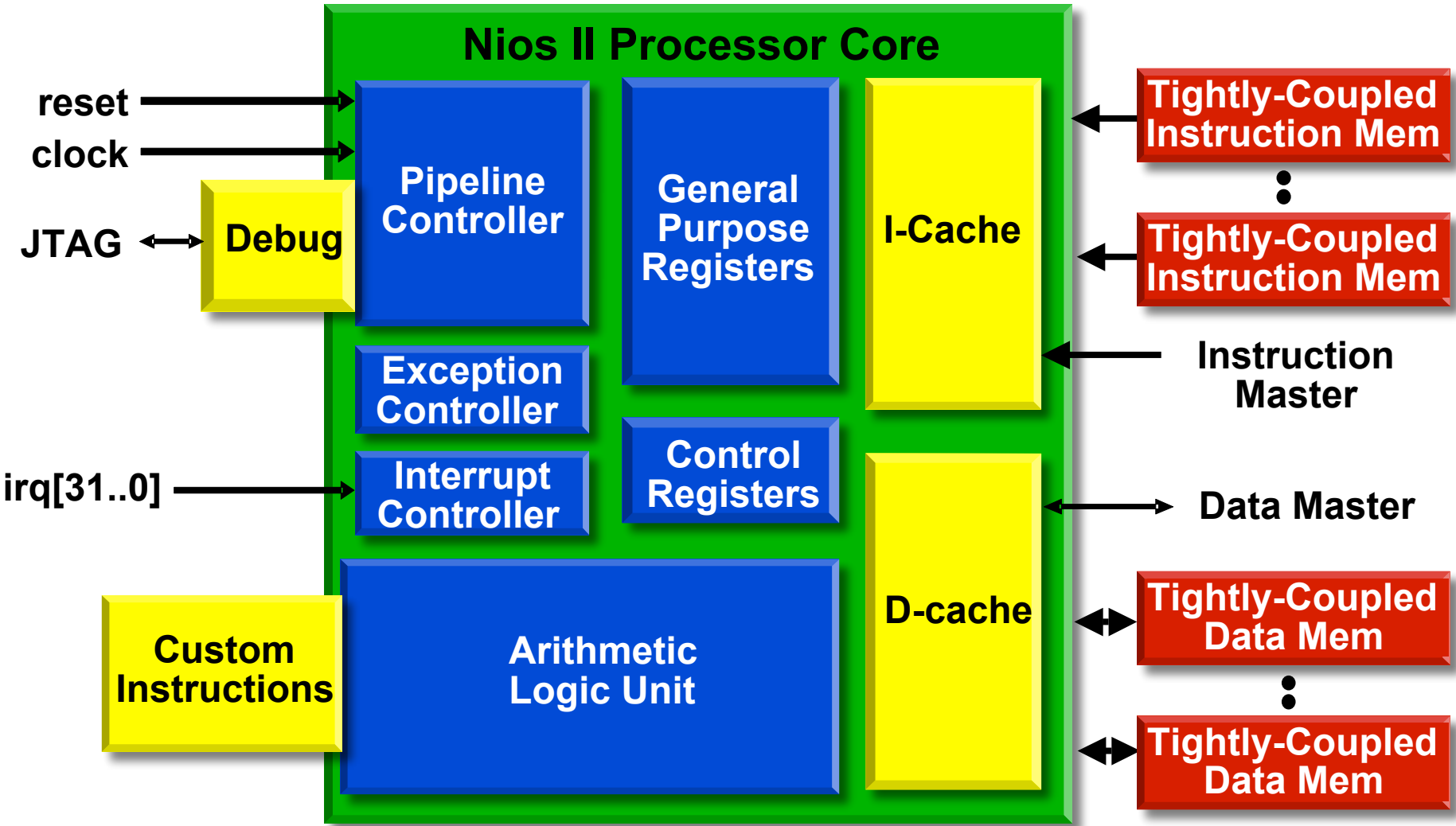
**35¢ in lowest  
cost FPGA**

**1% of FPGA  
Nios II/f "fast"**

# Nios II is Classic RISC

- 32-Bit Instruction Set
- 32-Bit Data path
- 32 General-Purpose Registers
- 3 Instruction Formats
- 82 Instructions
  - Instruction set is not configurable
  - Provides code compatibility for all implementations
- Up to 256 Custom Instructions
- 3 Operand Instructions (2 source, 1 destination)
- Optional Multiply and Divide

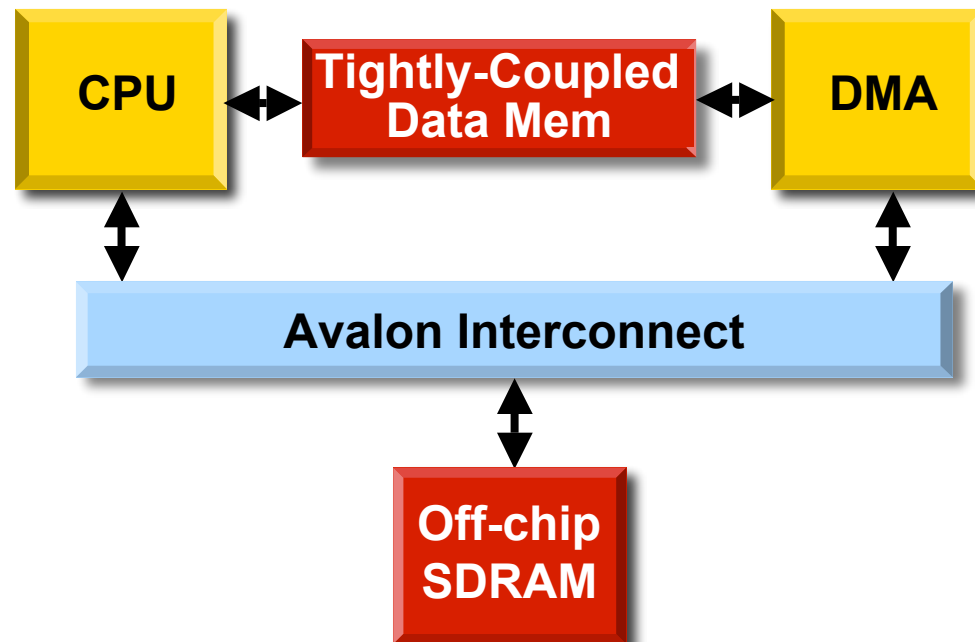
# Nios II Processor Block Diagram





# Configurable Tightly Coupled Memories

- Map on-chip RAMs into CPU address space
  - Behave like caches that never miss
  - One access every cycle without stalling
- FPGA RAMs are already dual-ported
  - One port for Nios II connection
  - Second port available for other uses



# Configurable CPU Implementation

- Choose your pipeline

<b>Nios® II</b>	<i>Nios II/f</i> “Fast”	<i>Nios II/s</i> “Standard”	<i>Nios II/e</i> “Economy”
Pipeline	<b>6-stage</b>	<b>5-stage</b>	<b>none</b>
Max Frequency <sub>1</sub>	<b>200 MHz</b>	<b>180 MHz</b>	<b>210 MHz</b>
Max D-MIPS <sub>1</sub>	<b>225</b>	<b>130</b>	<b>30</b>
Size (4-input LUTs)	<b>1800</b>	<b>1200</b>	<b>600</b>
Branch Prediction	<b>Dynamic</b>	<b>Static</b>	<b>no</b>
I-Cache	<b>Up to 64K</b>	<b>Up to 64K</b>	<b>no</b>
D-Cache	<b>Up to 64K</b>	<b>no</b>	<b>no</b>

1. Characteristics in Stratix II 90nm FPGA

# Configurable Pipeline Options

## ■ Cache options

- Size
- Line size

## ■ Multiply instruction options

- Fully pipelined using built-in FPGA multipliers
- Un-pipelined using normal LUT logic
- Trap (software emulated)

## ■ Divide instruction options

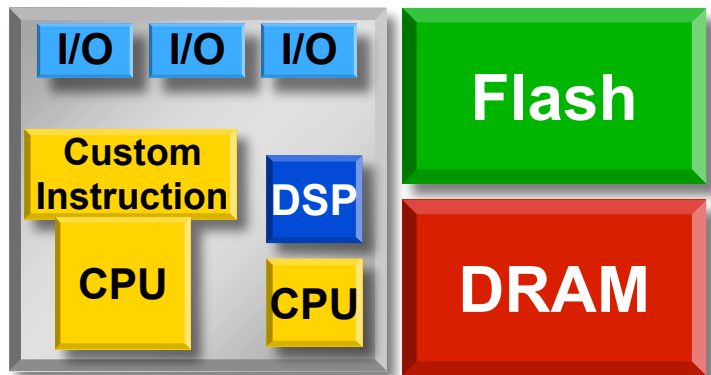
- Un-pipelined using normal LUT logic
- Trap (software emulated)

# Configurable Custom Instructions

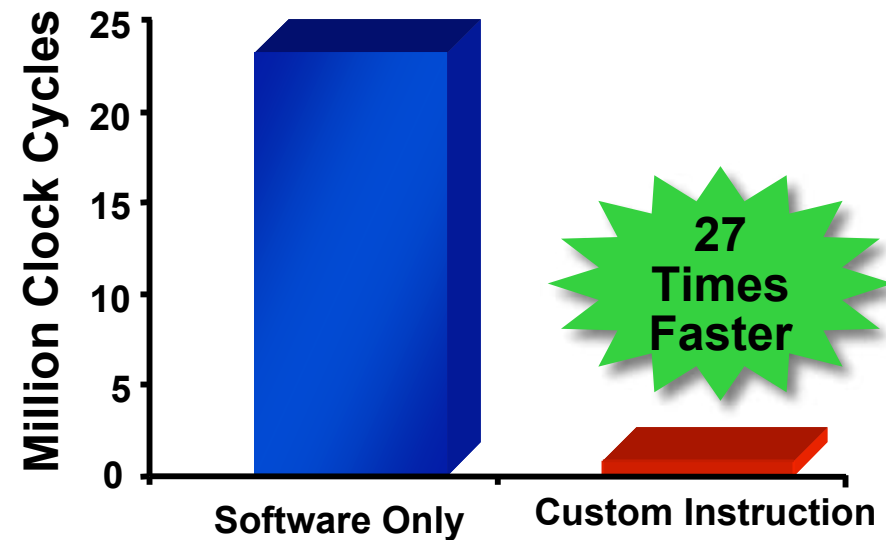
- Users write Verilog/VHDL for custom instructions
  - Added to CPU with automatic configuration tool
  - Callable from C-code or assembly language
- Pipeline independent
- 2 source operands and 1 destination operand
  - Access CPU register file
  - Access custom instruction register file
- Combinatorial custom instructions
  - Execute in parallel with ALU
- Multi-cycle custom instructions
  - Stall CPU pipeline until complete

# Configuring for Higher Performance

## *Add Custom Instructions*

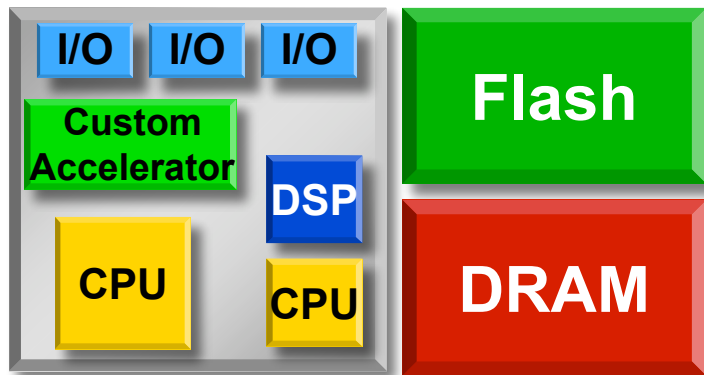


Example:  
64 Kbyte CRC

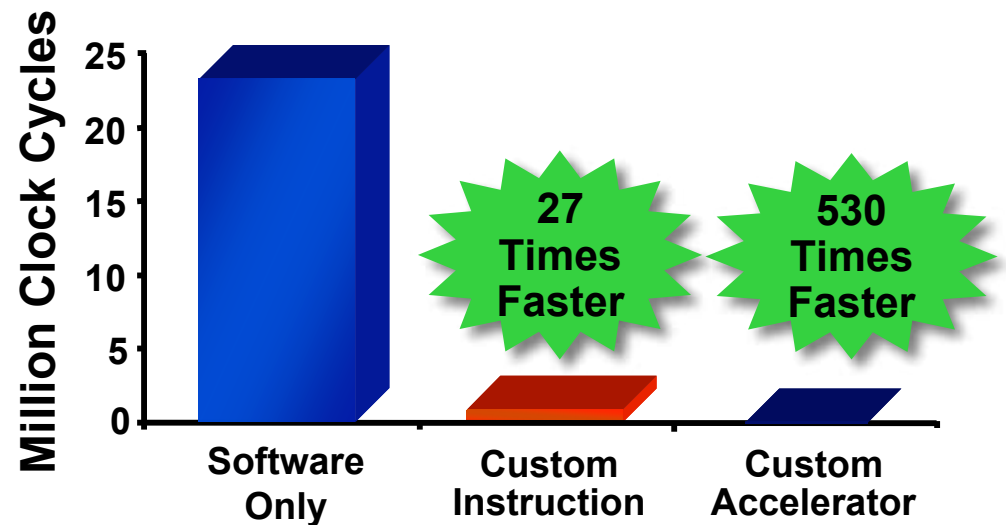


# Configuring for Higher Performance

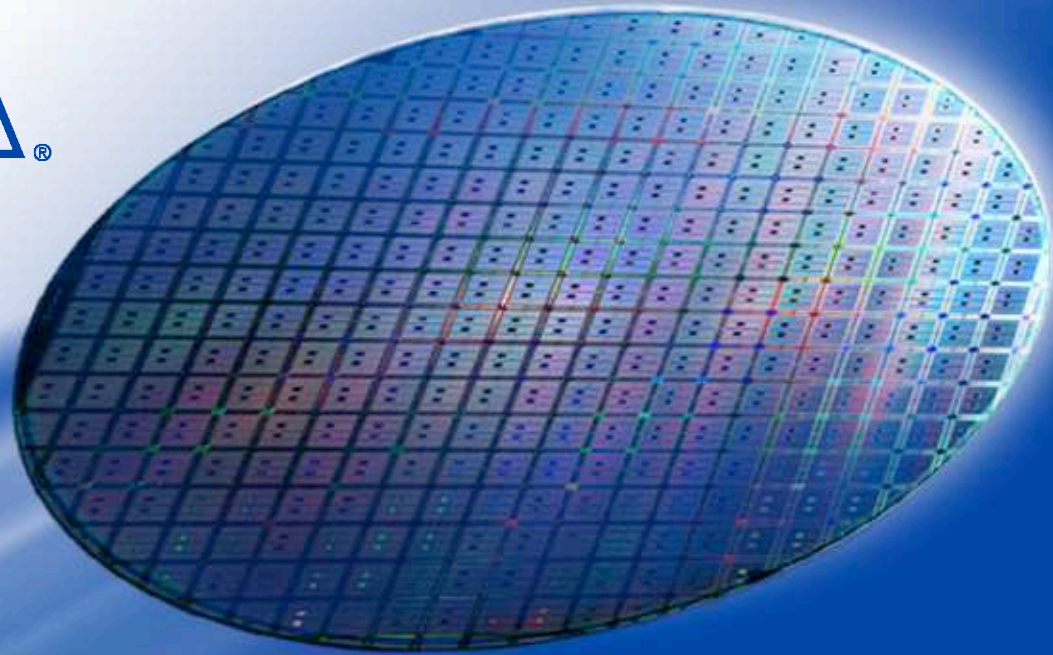
## *Add Custom Accelerator*



Example:  
64 Kbyte CRC



**ALTERA**®



# FPGA vs. ASIC CPU Design

# Efficient FPGA Design Guidelines

- RAMs, adders, registers, and multipliers
  - Relatively fast and plentiful
  - RAMs are already dual-ported
- Muxing and control logic
  - Relatively slow and expensive
- Wire delays
  - Relatively long
- Take advantage of FPGA configurability
  - Minimize run-time control registers
  - Rely on configuration-time options



# Existing ISAs are Inefficient in FPGAs

- Variable-length instructions or 16-bit instructions
  - Higher code density not worth extra control logic
- Register windows
  - Lower memory bandwidth not worth extra control logic
  - Can create difficult real-time requirements
- Barrel shifts combined with other arithmetic operations
  - Barrel shifts are relatively slow on FPGAs due to muxing
- Delay slots
  - Decreased branch penalty not worth extra control logic
  - Unnatural for some pipelines

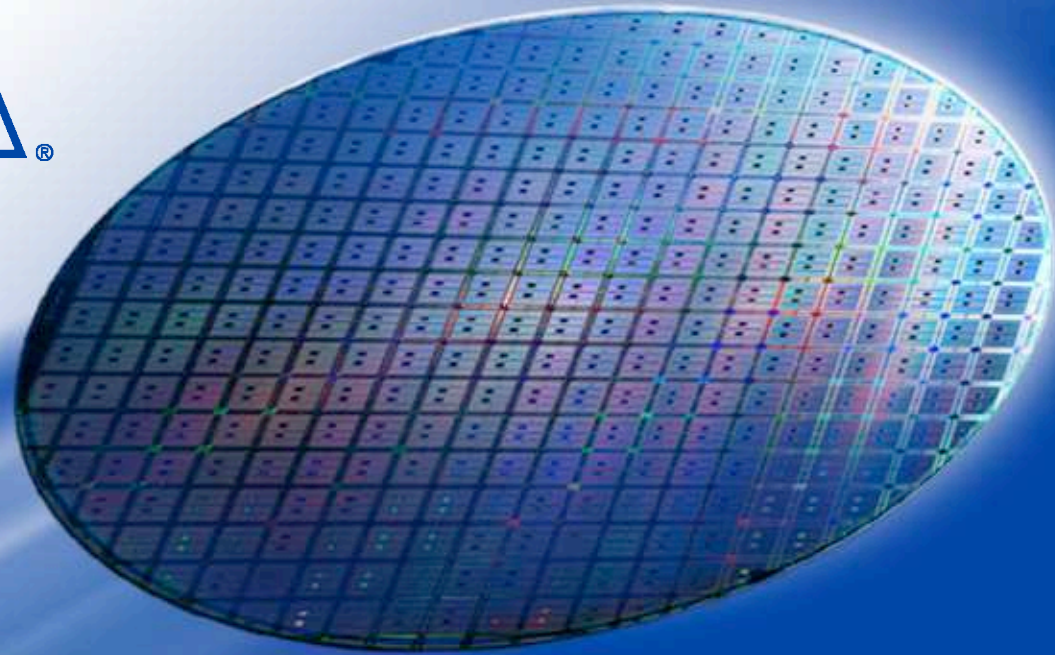
# Existing ISAs are Inefficient in FPGAs

- Condition code register
  - Complicates pipeline control and increases muxing
- Multiply/divide 64-bit operand registers
  - All 64-bits rarely used in C language and increases muxing
- Many run-time control registers
  - Extra logic not required in a configurable FPGA CPU
- Complex cache management
  - State machines to initialize on reset not worth extra logic
  - Many instruction options for flushing not worth extra logic
- Vectored interrupts
  - Not required for most designs
  - Use custom instruction to reduce interrupt latency

# Getting Back to RISC Roots

- CPU is an engine to run C code
  - Benchmarking shows Nios II has comparable performance to established embedded CPUs
- To increase CPU performance in an FPGA
  - Increase the Nios II cache size
  - Add Nios II custom instructions
  - Add custom accelerators
  - Add multiple Nios II CPUs
  - Add tightly-coupled memories

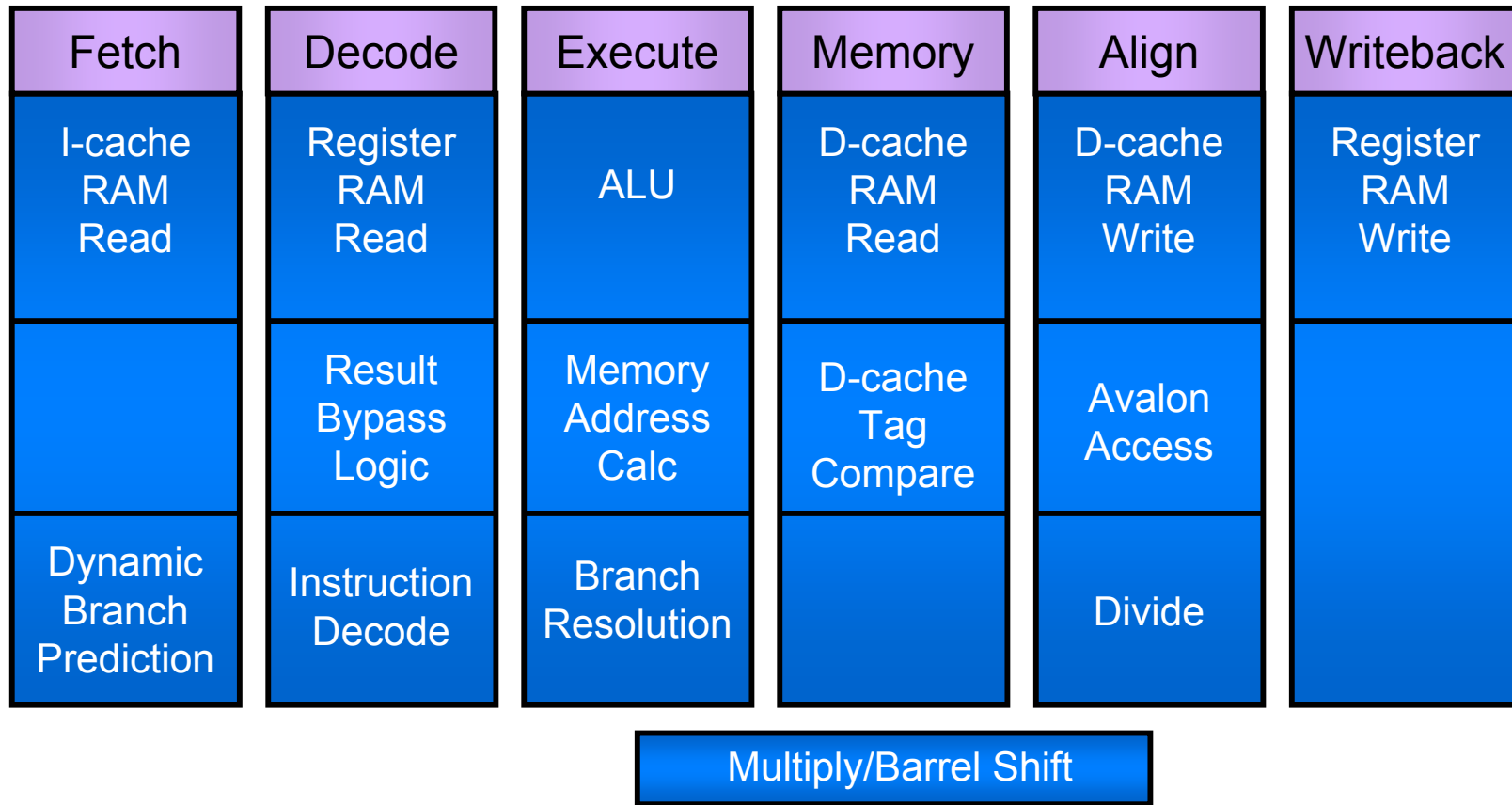
**ALTERA**®



# Nios II/f CPU Description

*“Fast”*

# Nios II/f Pipeline



# Caches

- Direct-mapped
  - Set-associative caches inefficient in FPGA
- I-cache
  - 32-byte line
  - Critical word first
- D-cache
  - 4/16/32-byte line
  - Writeback with write allocate
  - One entry writeback buffer

# Dynamic Branch Prediction

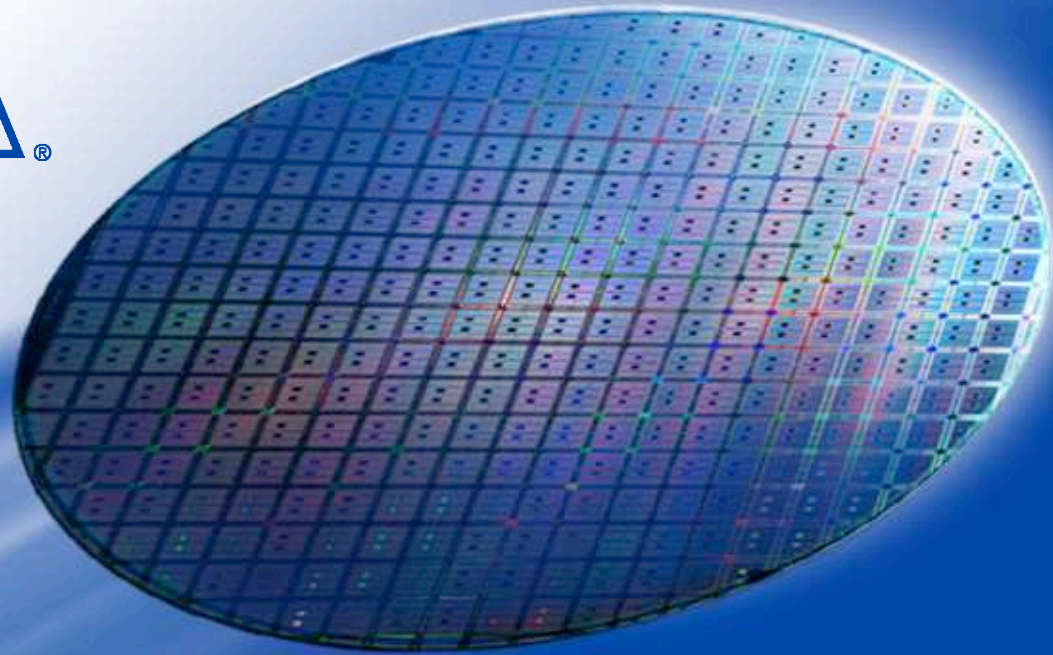
- 2-bit branch prediction (g-Share algorithm)
  - Branch History Table RAM (256x2 bits)
- No Branch Target Buffer
  - Simple ISA allows fast branch target calculation
- Performance
  - Taken branch is 2 cycles
  - Not taken branch is 1 cycle
  - Mispredicted branch penalty is 4 cycles

# Arithmetic Instructions

- 32-bit Multiply
  - 1 cycle throughput (fully pipelined)
- 32-bit Divide
  - 4-67 cycle throughput (not pipelined)
- Barrel shift/rotate
  - Uses multiplier with  $2^n$  calculation
  - Better performance and lower cost than using LUTs



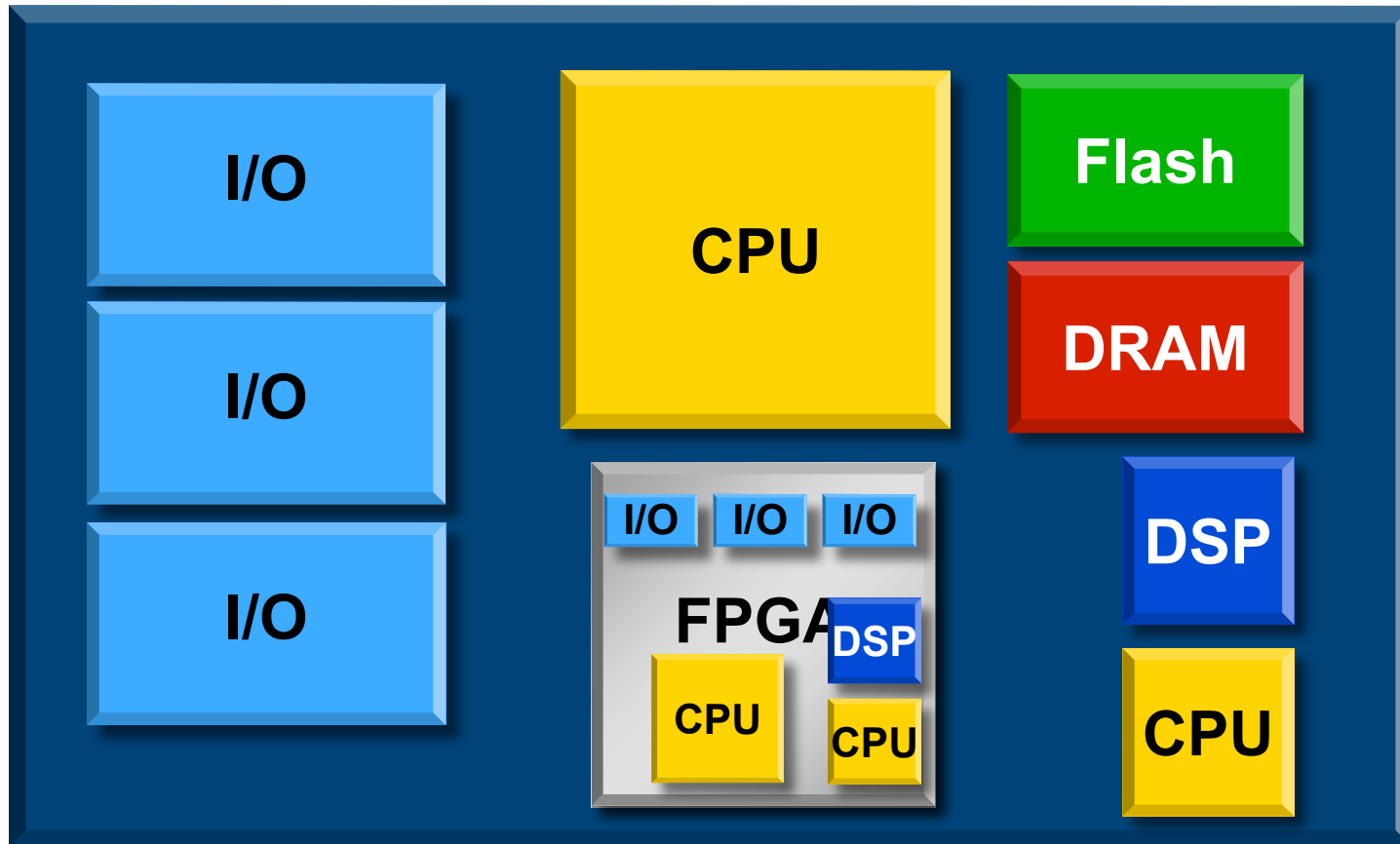
**ALTERA**®



# Nios II Embedded Systems

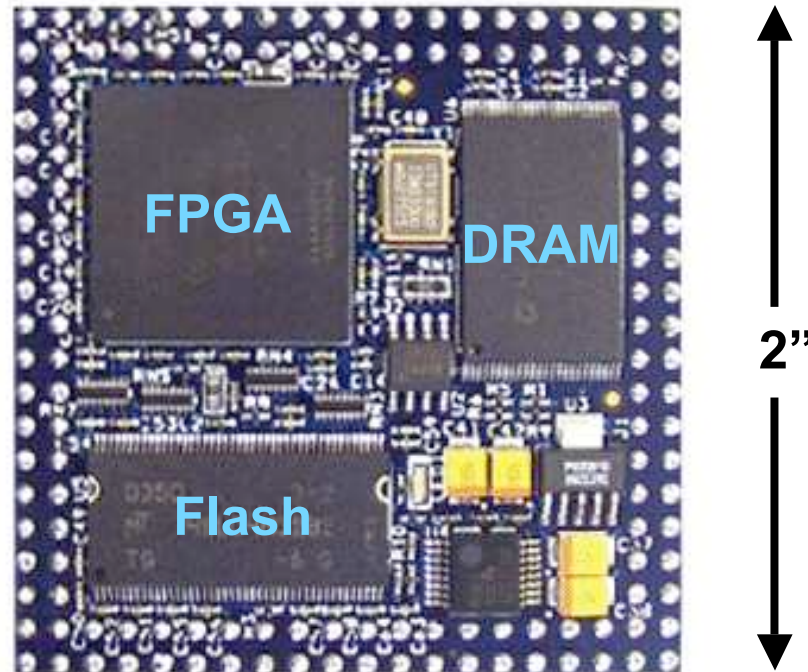
# Board-based Embedded System

## FPGA-based Embedded System



*Move board components into FPGA*

# Nios II Evaluation Board



**Preconfigured with a web server running under  $\mu$ Clinux**

# FPGA-based Systems

- It's all configurable
  - Configurable CPUs
  - Configurable Memories (on-chip and off-chip)
  - Configurable Peripherals
  - Configurable I/O
  - Configurable System Interconnect
  - Custom Accelerators
- and we provide the tools to make it easy ...

# System Configuration Tool

Altera SOPC Builder - std\_2s60

File Module System View Debug Tools Help

System Contents Board Settings **Nios II** More "cpu" Settings System Generation

Target: Board: Nios Development Board, Stratix II (EP2S60) Device Family: Stratix II  HardCopy Compatible

Clock (MHz): clk 100.0

Use	Module Name	Description	Base	End	IRQ
<input checked="" type="checkbox"/>	<b>cpu</b>	Nios II Processor - Altera Corporation			
	instruction_master	Master port			
	data_master	Master port			
	jtag_debug_module	Slave port	0x02120000	0x021207FF	IRQ 0 - IRQ 31
<input checked="" type="checkbox"/>	<b>ext_ram_bus</b>	Avalon Tri-State Bridge			
	avalon_slave	Slave port			
	tristate_master	Master port			
<input checked="" type="checkbox"/>	<b>ext_flash</b>	Flash Memory (Common Flash Interface)	0x000000...	0x00FFFFFF	
<input checked="" type="checkbox"/>	<b>ext_ram</b>	IDT71V416 SRAM	0x020000...	0x020FFFFFF	
<input checked="" type="checkbox"/>	<b>onchip_ram_64_kbytes</b>	On-Chip Memory (RAM or ROM)	0x021000...	0x0210FFFF	
<input checked="" type="checkbox"/>	<b>lan91c111</b>	LAN91c111 Interface (Ethernet)	0x02110000	0x0211FFFF	6
<input checked="" type="checkbox"/>	<b>sys_clk_timer</b>	Interval timer	0x02120800	0x0212081F	0
<input checked="" type="checkbox"/>	<b>jtag_uart</b>	JTAG UART	0x021208B0	0x021208B7	1
<input checked="" type="checkbox"/>	<b>button_pio</b>	PIO (Parallel I/O)	0x02120860	0x0212086F	2
<input checked="" type="checkbox"/>	<b>led_pio</b>	PIO (Parallel I/O)	0x02120870	0x0212087F	
<input checked="" type="checkbox"/>	<b>lcd_display</b>	Character LCD (16x2, Optrex 16207)	0x02120880	0x0212088F	
<input checked="" type="checkbox"/>	<b>high_res_timer</b>	Interval timer	0x02120820	0x0212083F	3
<input checked="" type="checkbox"/>	<b>seven_seg_pio</b>	PIO (Parallel I/O)	0x02120890	0x0212089F	
<input checked="" type="checkbox"/>	<b>reconfig_request_pio</b>	PIO (Parallel I/O)	0x021208A0	0x021208AF	
<input checked="" type="checkbox"/>	<b>uart1</b>	UART (RS-232 serial port)	0x02120840	0x0212085F	4
<input checked="" type="checkbox"/>	<b>sysid</b>	System ID Peripheral	0x021208B8	0x021208BF	
<input checked="" type="checkbox"/>	<b>sdram</b>	SDRAM Controller	0x010000...	0x01FFFFFF	

Done checking for updates.

Exit < Prev Next > Generate

# CPU Configuration Tool

The screenshot shows the 'Altera Nios II - cpu' configuration window. The 'Caches & Tightly Coupled Memories' tab is active. The 'Instructions' section has 'Instruction Cache' set to '64 Kbytes' and 'Include tightly coupled instruction master port(s)' checked with 'Number of ports' set to '1'. The 'Data' section has 'Data Cache' set to '64 Kbytes', 'Data Cache Line Size' set to '32 Bytes', and 'Include tightly coupled data master port(s)' checked with 'Number of ports' set to '1'. There is an unchecked checkbox for 'Omit data master port'. Navigation buttons at the bottom include 'Cancel', '< Prev', 'Next >', and 'Finish'.

Altera Nios II - cpu

Nios II Core Caches & Tightly Coupled Memories JTAG Debug Module Custom Instructions

Instructions

Instruction Cache: 64 Kbytes

Include tightly coupled instruction master port(s)

Number of ports: 1

You must connect each port to exactly one memory in the SOPC Builder connection panel.

Data

Data Cache: 64 Kbytes  Omit data master port

Data Cache Line Size: 32 Bytes

Include tightly coupled data master port(s)

Number of ports: 1

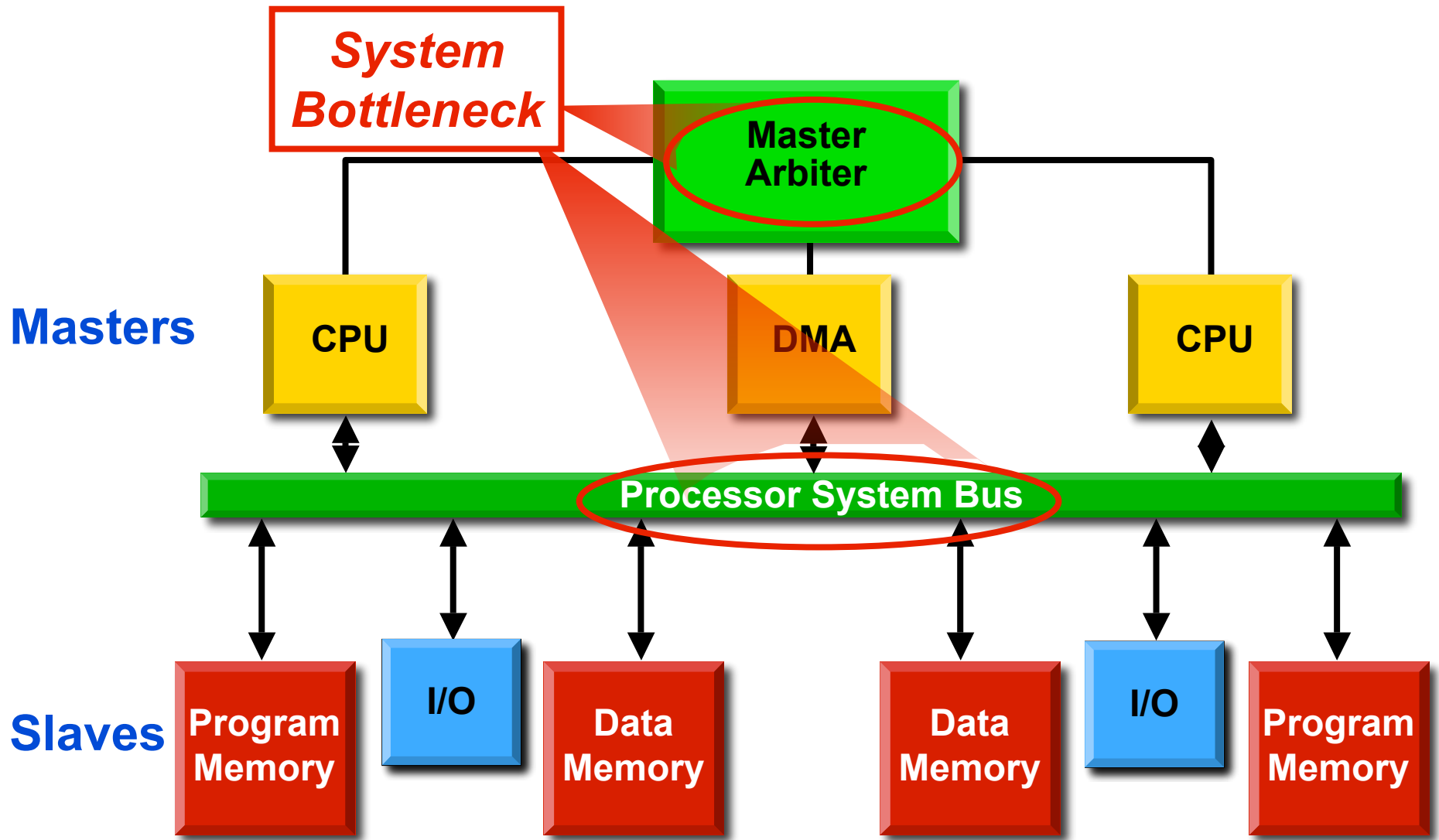
You must connect each port to exactly one memory in the SOPC Builder connection panel.

Cancel < Prev Next > Finish

# Avalon System Interconnect

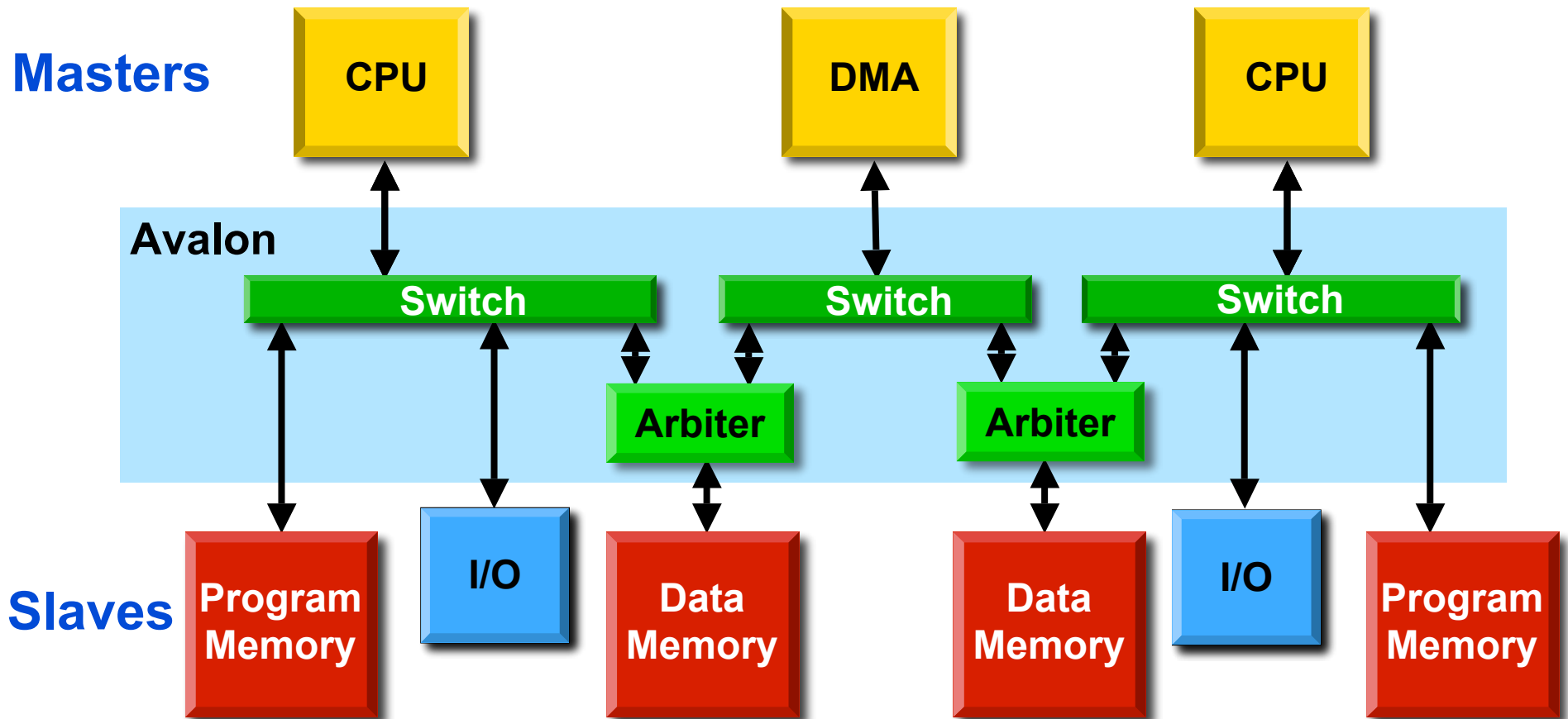
- Automatically generated for your system
- Switches connect components – not a bus
- Slave side arbitration
  - Enables concurrent accesses
- Avalon Functions
  - Arbitration
  - Multiplexing
  - Address Decoding
  - Wait-State Generation
  - Dynamic Bus Sizing

# Traditional Bus Interconnect





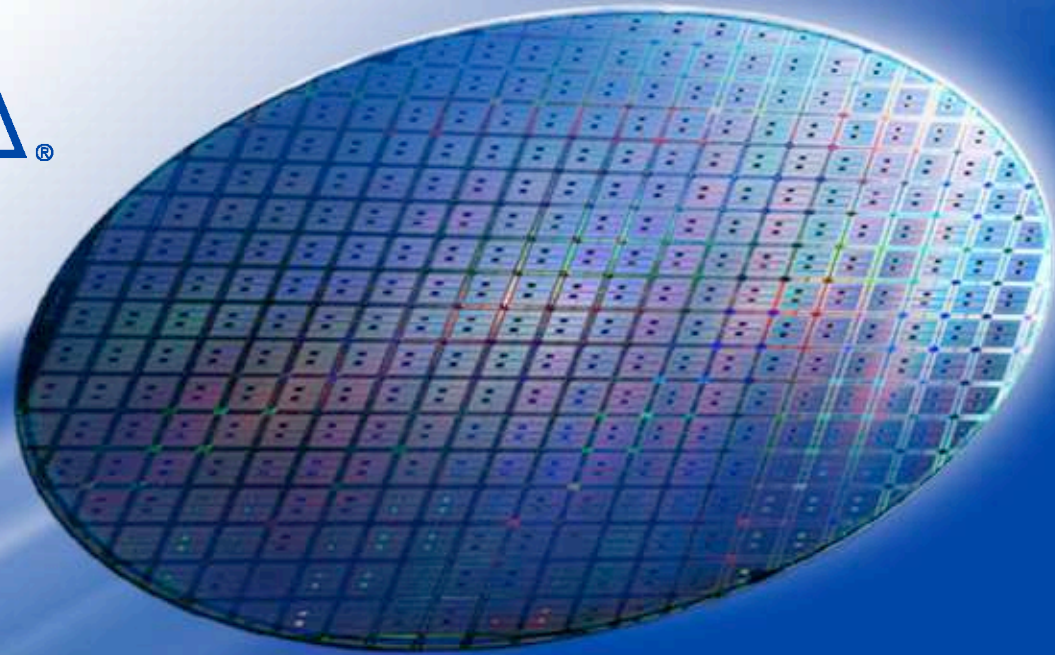
# Avalon Switch Interconnect



# Conclusions

- Efficient FPGA design takes advantage of configurable CPUs and systems
- Nios II is optimized for FPGA-based systems
- Established CPUs based on ISAs optimized for ASICs are less efficient in FPGAs

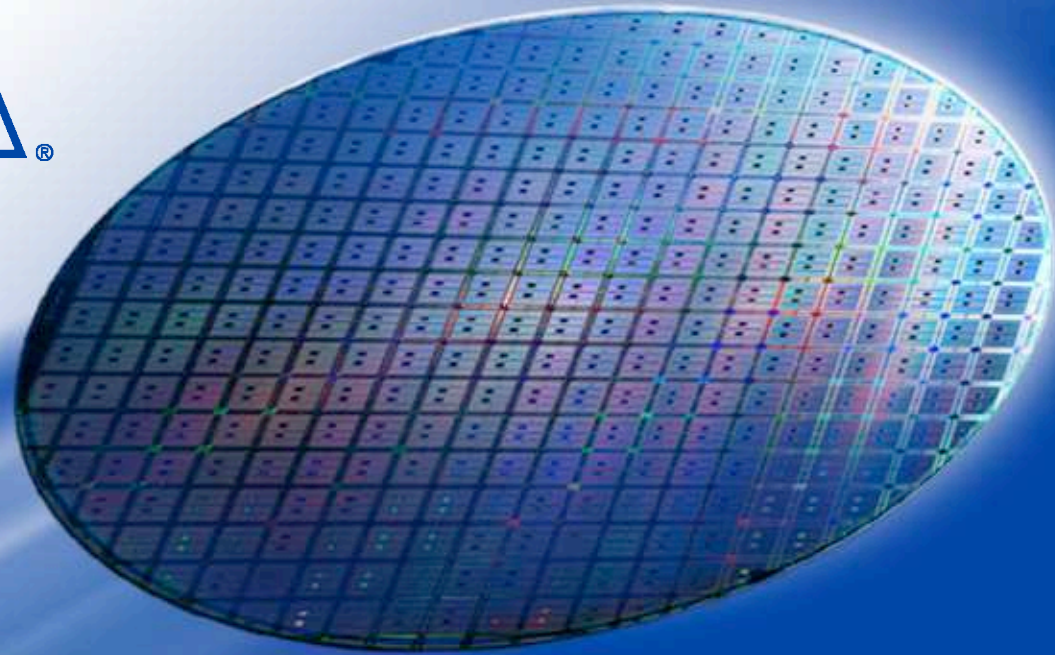
**ALTERA**<sup>®</sup>



**The End**

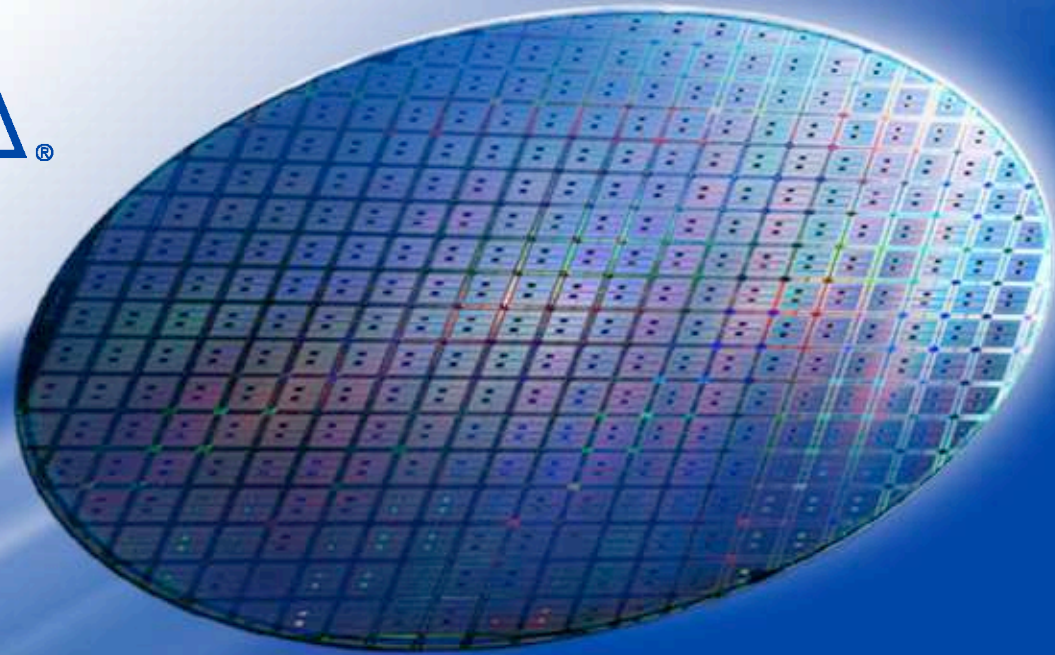
*Questions?*

**ALTERA**<sup>®</sup>



**Backup Slides**

**ALTERA**®



# Why a Soft-Core FPGA CPU?

# FPGA Soft-Core CPU Advantages

- **Flexibility**
  - Utilize existing silicon resources
- **Scalability**
  - Number of CPUs, CPU types, cache sizes, etc.
- **Configurability**
  - Generation-time configuration instead of run-time
  - Eliminates logic required to control CPU options
- **Ubiquity**
  - Available in all FPGA families

# FPGA Soft-Core CPU Advantages

- Relatively small compared to FPGA capacities
  - Largest Altera FPGA fits 300 Nios II/e CPUs
  - May have spare capacity so CPU is free
- Lifecycle
  - No obsolescence
  - New releases of CPU improve your design
  - Improved efficiency with latest silicon technologies

# Altera's Latest FPGA Devices

	<i>Stratix II</i>	<i>Cyclone II</i>
Technology	90 nm	90 nm
4-input LUTs	180,000	70,000
18-bit Multipliers	384	180
On-chip RAM	1.2 Mbytes	144 Kbytes