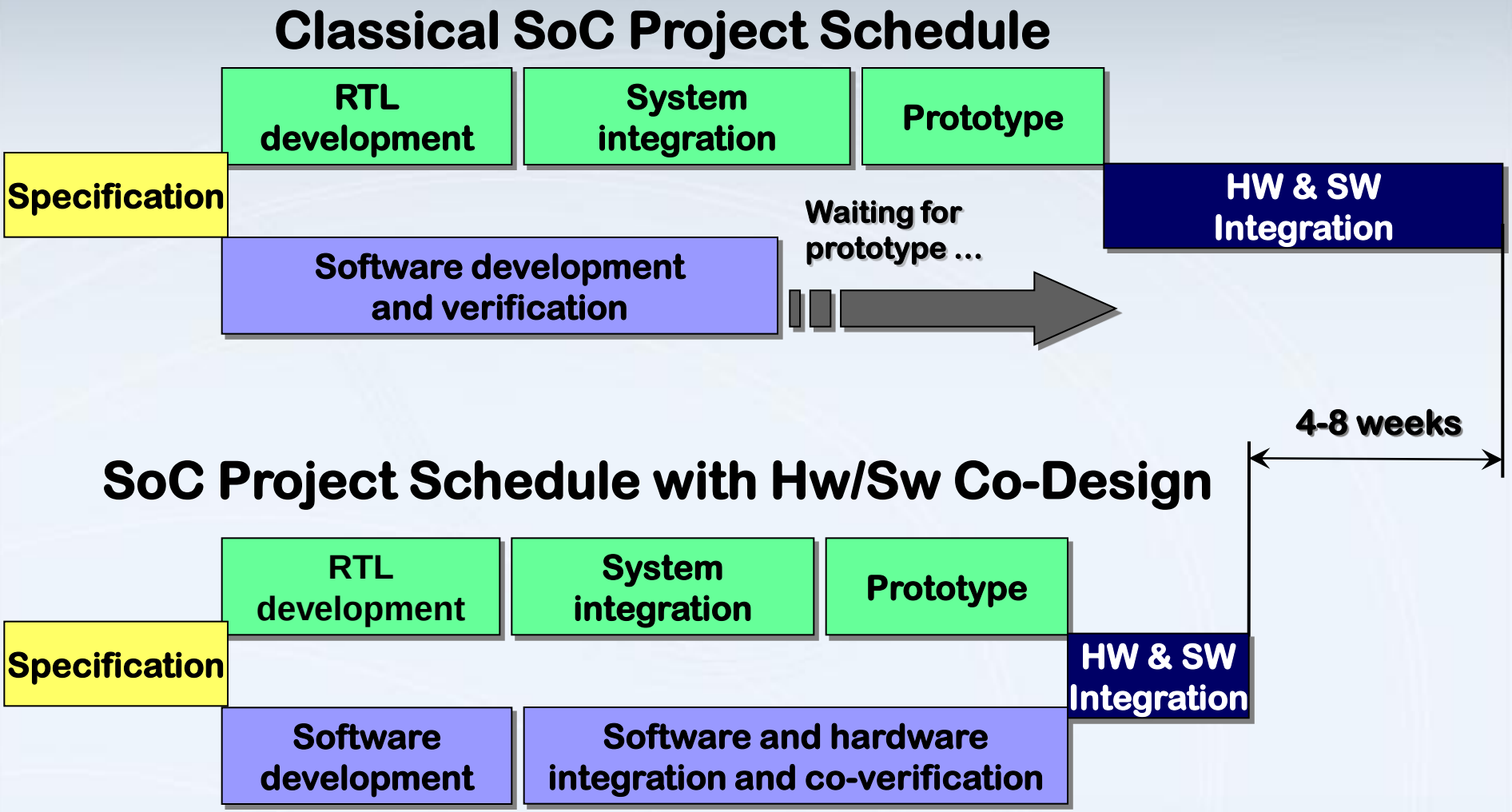
A Novel Software/Hardware Synchronization Technique for Fast Debugging of SoC

Efficient Hardware/Software Co-Design

Corporate Background

- **Focused:**
  - Development of HDL verification, hardware acceleration and prototyping technologies
- **Products:**
  - Active-HDL
  - CoVer
  - HES
  - Riviera
- **Core technology:**
  - Mixed VHDL, Verilog, SystemC and SystemVerilog simulation
  - HW/SW co-verification
  - Hardware acceleration and debugging

Arguments for hardware/software co-design

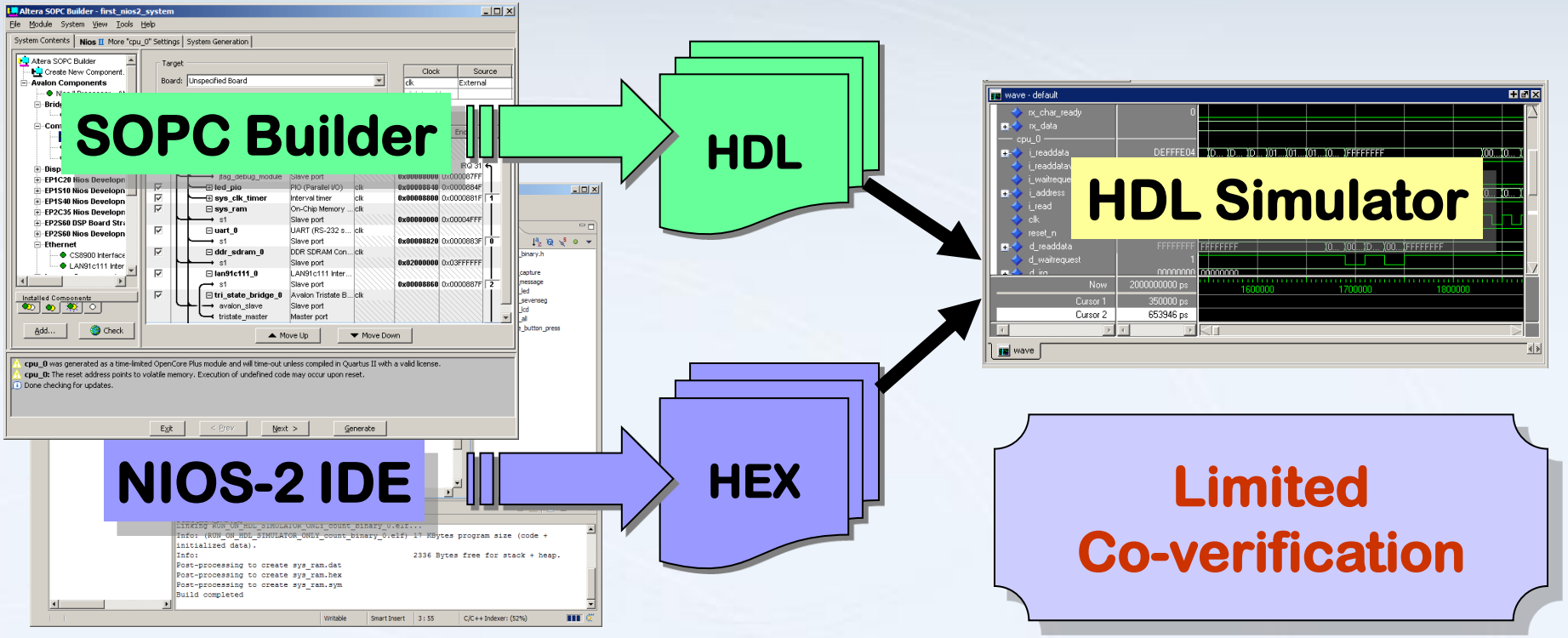


Arguments for hardware/software co-design

- **Hardware developer benefits**
  - no need for complex test harness
  - code executed by NIOS-2 is like transaction level testbench in C/C++
  - higher test coverage in less time
  - discover software triggered hardware bugs
- **Software developer benefits**
  - system available long before any prototype is ready
  - no need for “dummy peripherals” during verification
  - better understanding of hardware requirements
  - discover hardware triggered software bugs

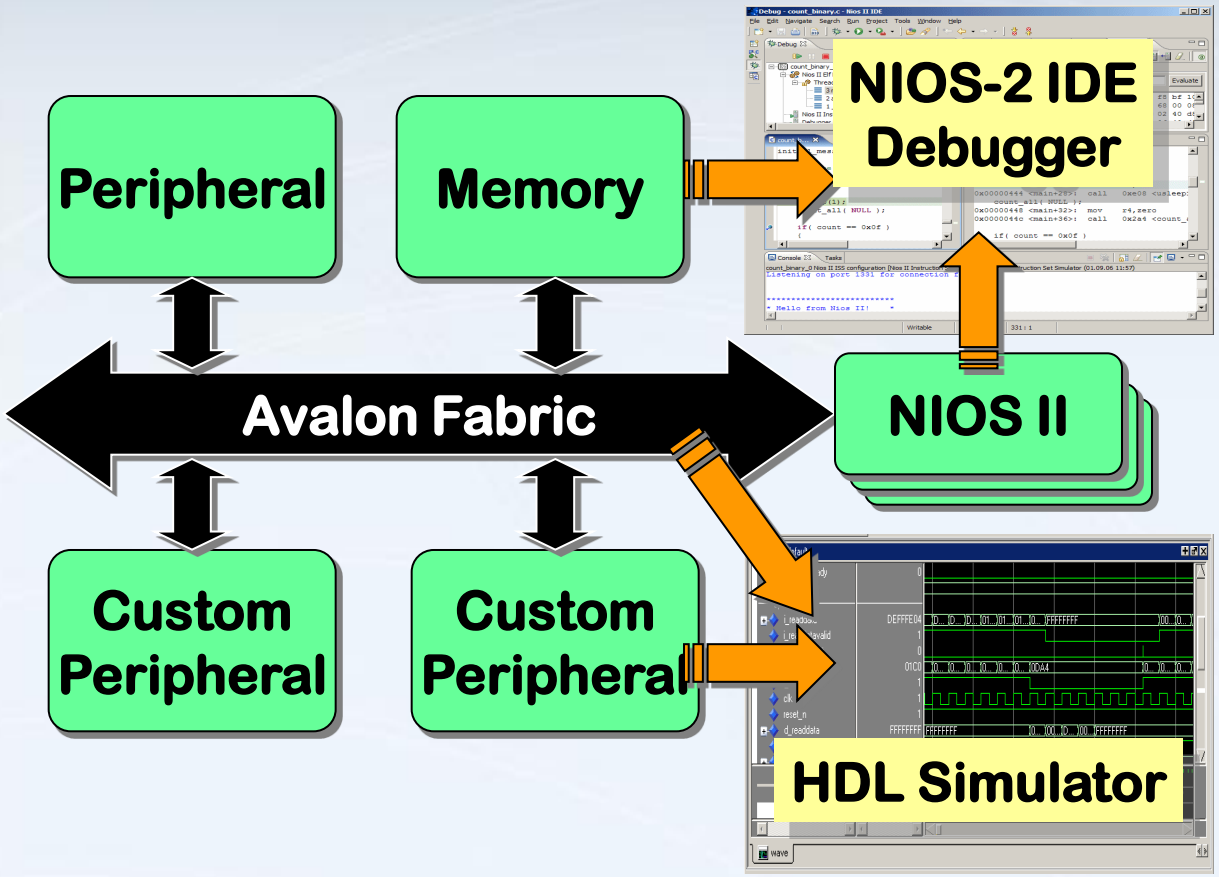
Hardware/software co-simulation drawbacks

Current solution is hardware/software co-simulation.



- **Very slow execution**
  - **1 KCycles/s max speed**
- No software debugging environment
  - co-simulation is hardware oriented

Requirements for true hardware/software co-design

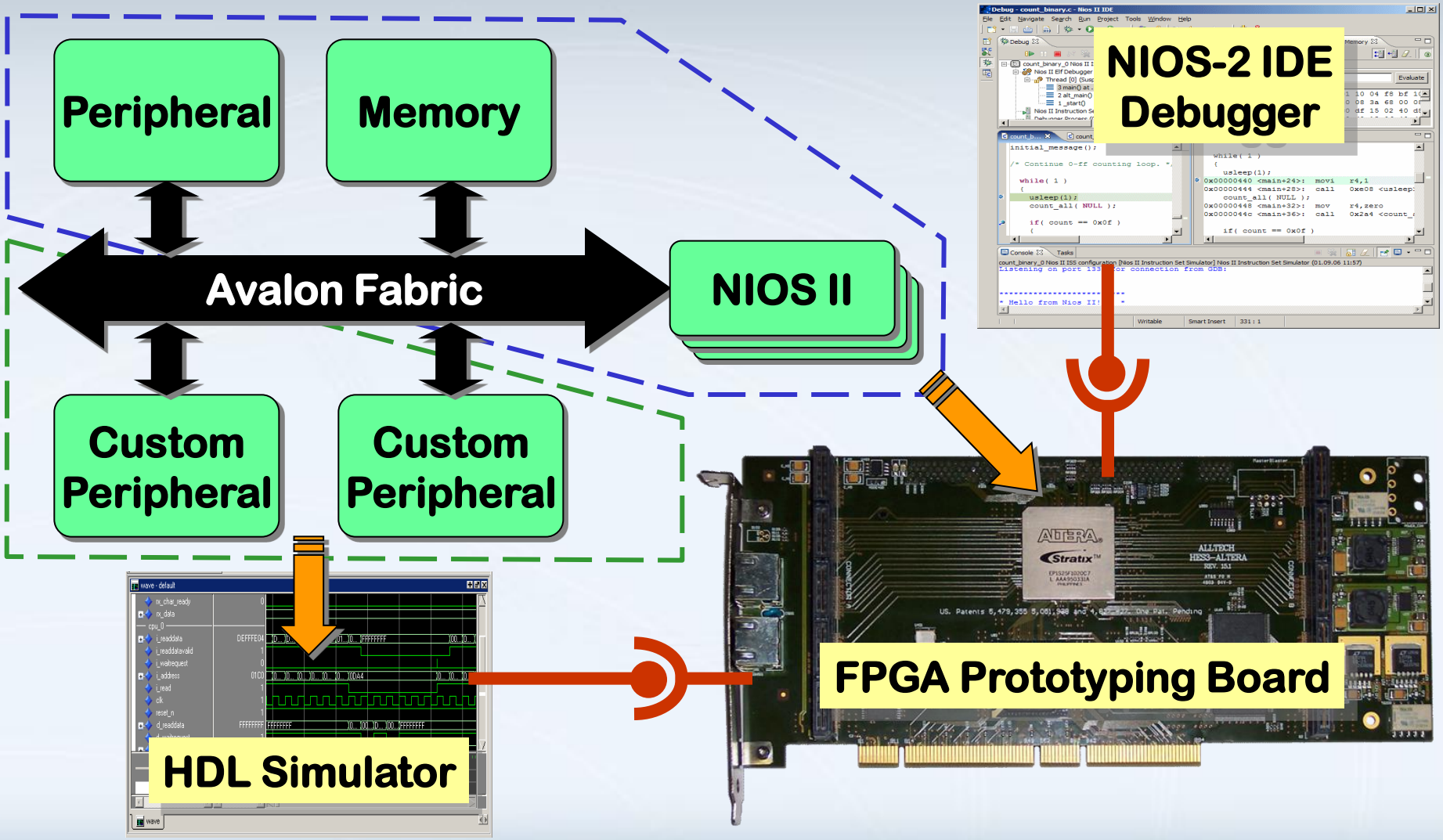


1. Software debug with C/C++ code in the processor debugger

Hardware speed freq ~ MHz

2. Selective hardware debug in HDL simulation

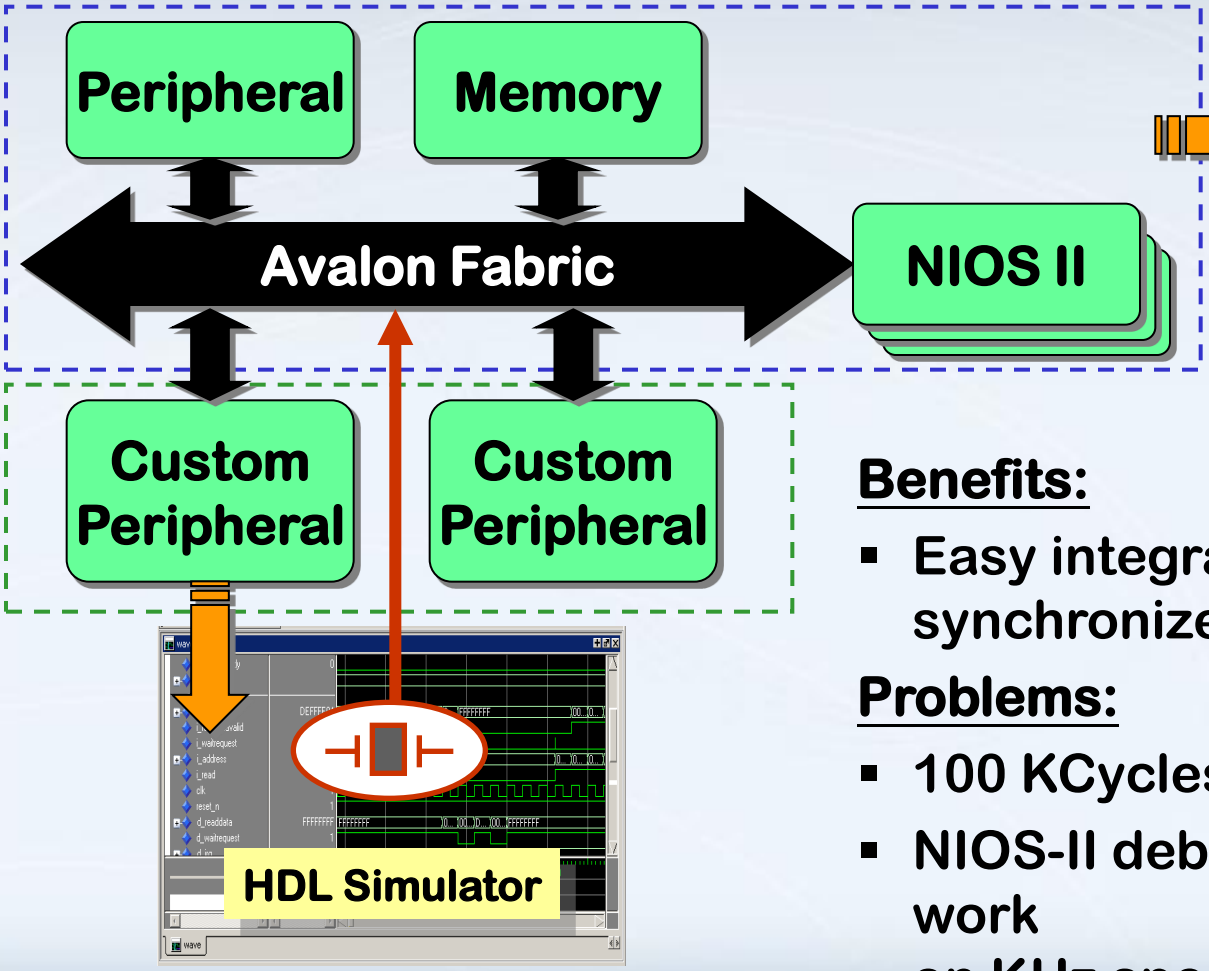
FPGA based hardware/software co-design

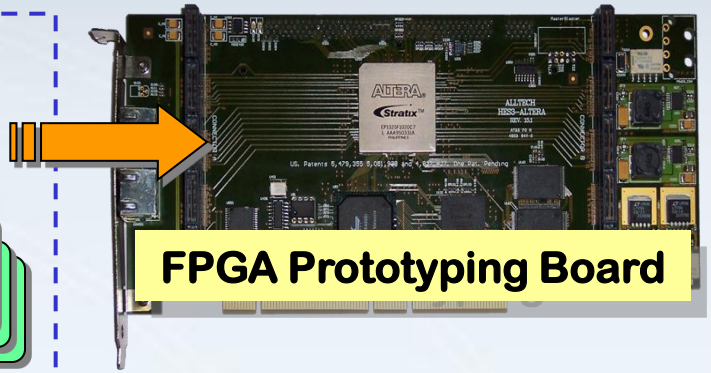


ALDEC
The Design Verification Company

Connecting FPGA board to the HDL Simulator

CASE-1: HDL simulator generates system clock





Benefits:

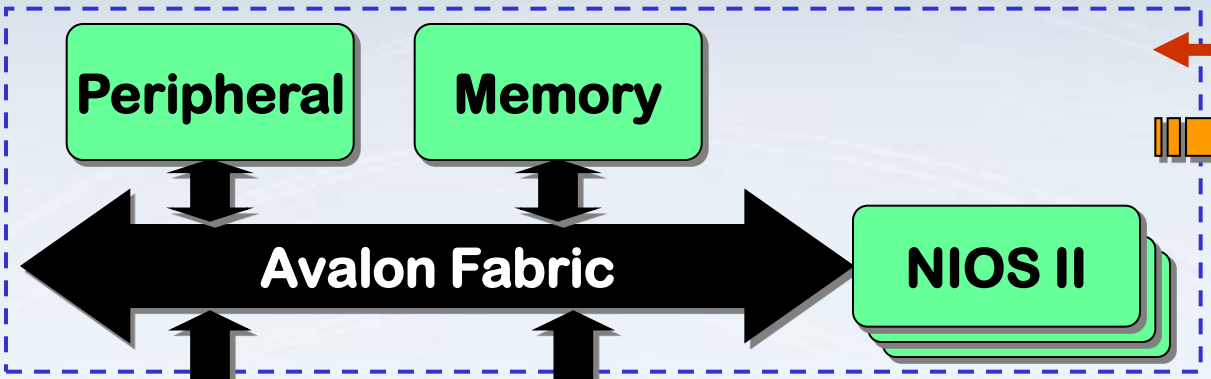
- Easy integration – all modules synchronized

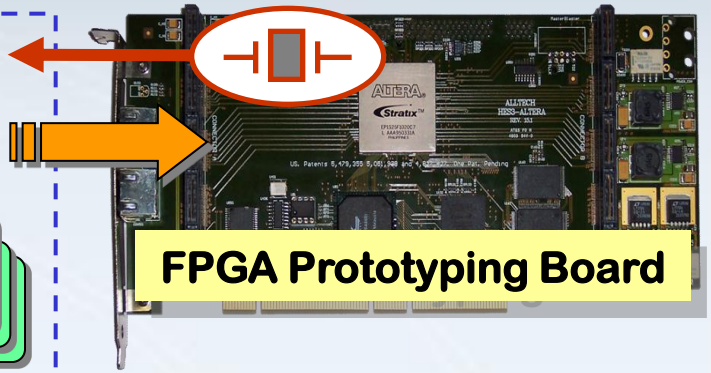
Problems:

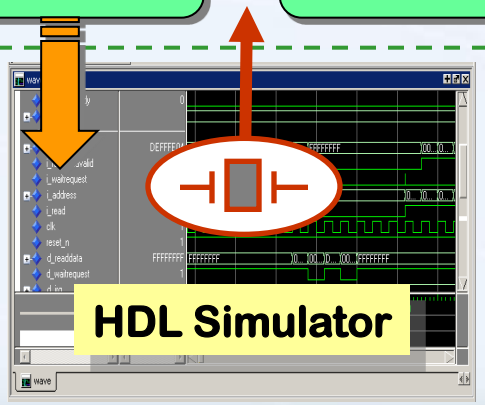
- 100 KCycles/s - max speed
- NIOS-II debug channel does NOT work
  on KHz speed !!!

Connecting FPGA board to the HDL Simulator

CASE-2: Hardware oscillator drives FPGA







- HDL Simulator generates clock for Custom Peripherals
- Rest of the system runs on hardware oscillator

Benefits:

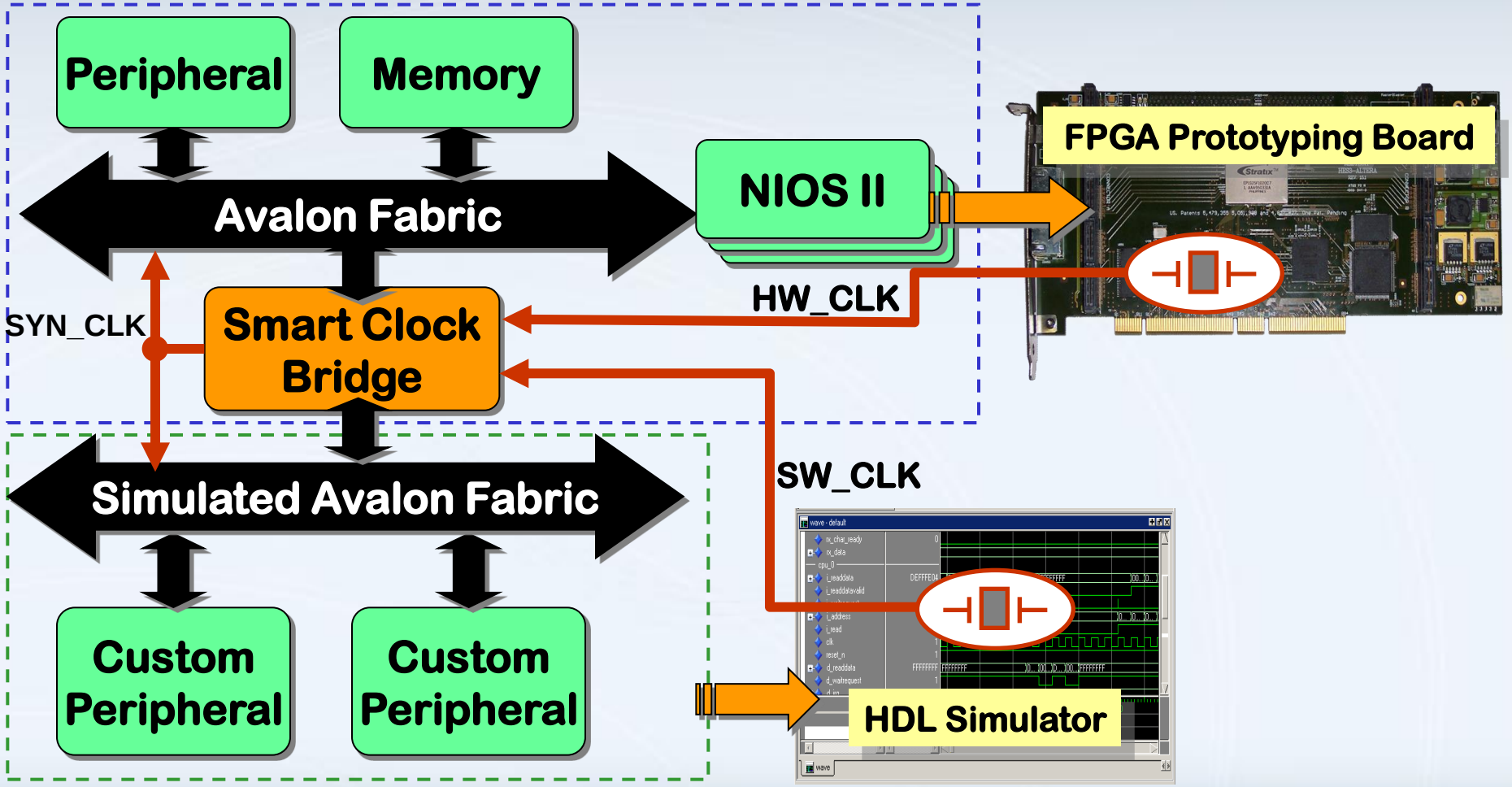
- System works with expected speed

Issues:

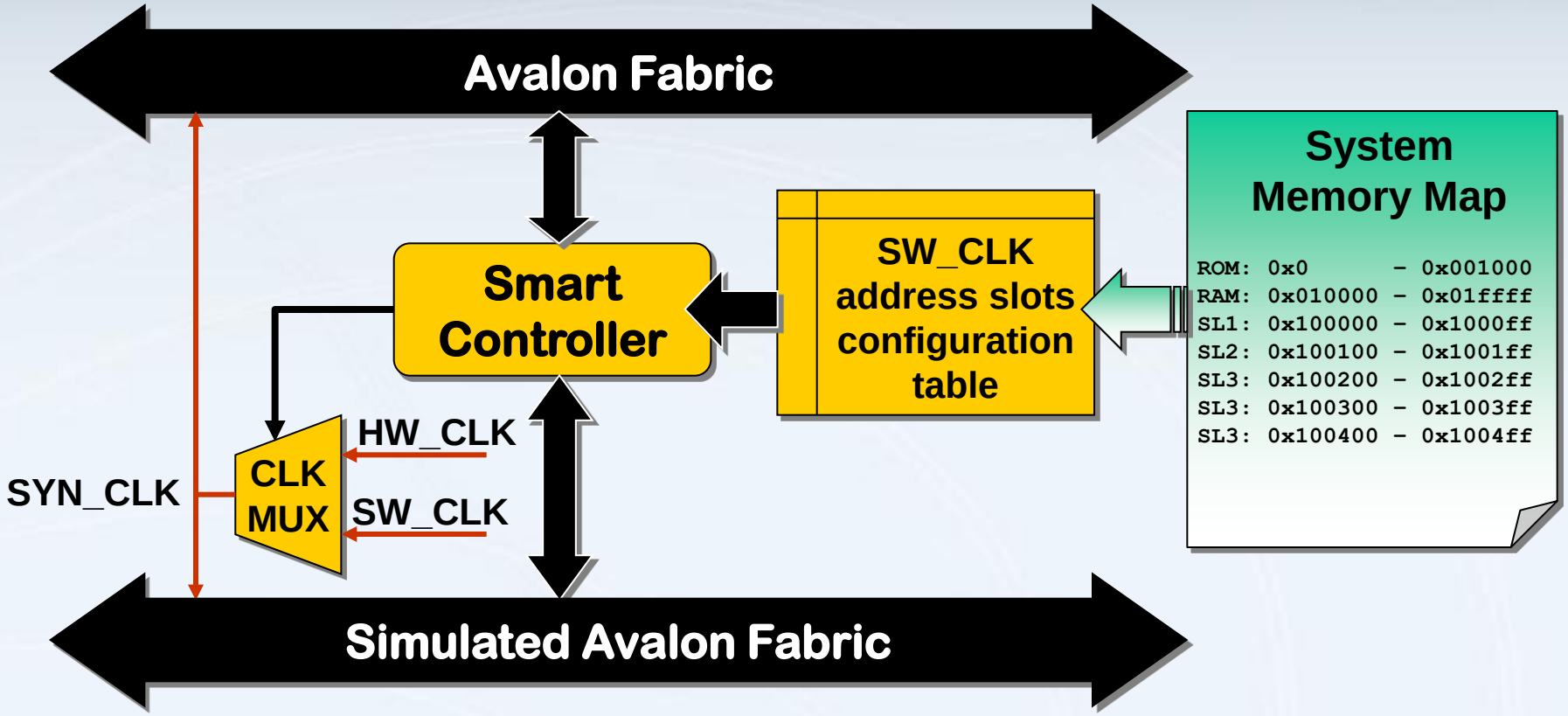
- Clocks synchronization required !

Smart Clock – the synchronization solution

CASE-2 with Smart Clock bridge synchronization module



Smart Clock Bridge



Smart Clock – Summary

- There are two clock domains
  - SW_CLK – software clock (HDL simulator)
  - HW_CLK – hardware clock (oscillator)
- Smart Clock Bridge (SCB) is used to synchronize two clock domains
- SW_CLK drives the system when Avalon bus transaction needs to be passed through HDL simulator otherwise HW_CLK is used
- SCB is programmable
  - Software Clock Window (SCW) – an address range for bus transactions executed with SW_CLK
  - SCW table is configured on the fly – no need to reconfigure FPGA
- Final performance is variable and depends on SW_CLK to HW_CLK transactions ratio.
  Typical case is:
  - 5% transactions on SW_CLK
  - 95% transactions on HW_CLK

Hypothetical Benchmark

	HW_CLK	SW_CLK
Frequency	f_H = 50 MHz	f_S = 100 kHz
Period	T_H = 20 ns	T_S = 10 000 ns
Transfer rate*	95 %	5 %
Average period	$T_{av} = T_H*0.95 + T_S*0.05 =$ **519 ns**	
Average speed	$f_{av} = 1/T_{av} \cong$ **1.9 MHz**	

Speedup ** 1900x

(*)
Transfer rate is variable and depends on design configuration. Assuming that only custom peripherals are simulated in HDL the above numbers can be considered as typical.

(**)
Assuming 1K cycles/s of HDL simulator alone

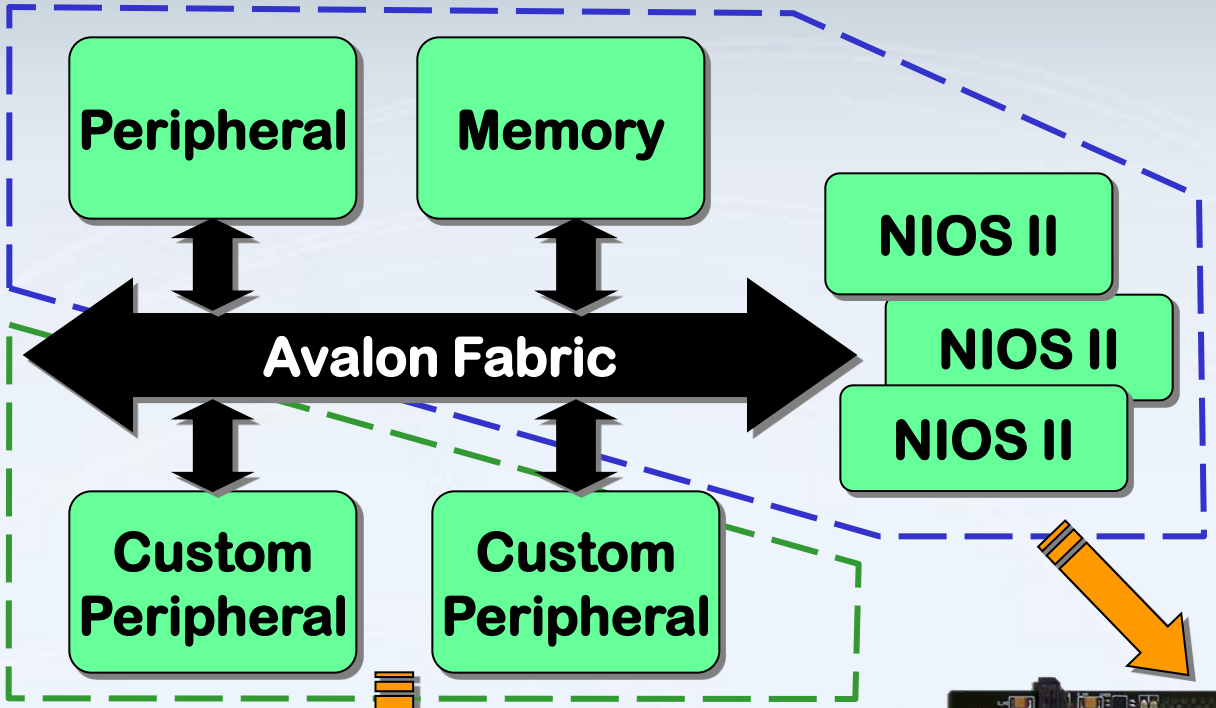
Success Stories

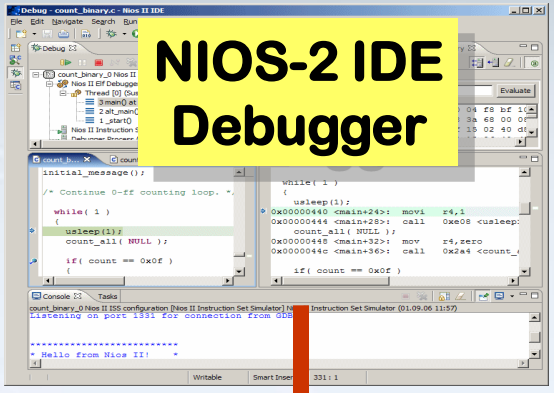
- Optimization of JPEG algorithm running on NIOS-2 processor
  - SOPC Configuration:
    NIOS-2, program RAM, data RAM, JPEG accelerator peripheral
  - HDL simulation: 52 hrs
  - CoVer: 55 s
  - Speedup: **3400x**
- uC Linux boot-strap
  - HDL simulation: 23 hrs
  - CoVer: 78 s
  - Speedup: **1000x**
- DMA channel design development and verification
  - Team-1: with HDL simulation: 40 man-hrs
  - Team-2: with CoVer: 12 man-hrs
  - Productivity improvement: **3.3x**

Multi-Processor Design Challenges

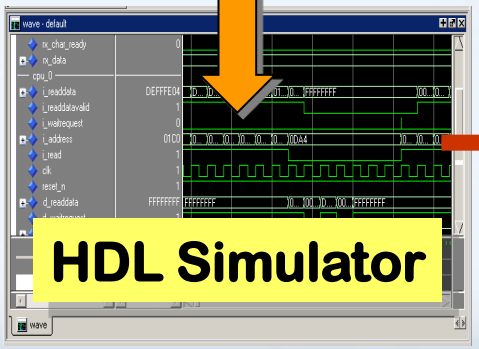
- Concurrent threads of software running on different CPUs share some resources.
- Hardware design is well supported by SOPC Builder
  - multiple NIOS-2 processors can be used
  - hardware mutex IP is available
- Software design and verification is the real challenge
  - efficient co-operation of concurrent tasks
  - lack of negative interference and access conflicts
  - avoid data corruption between threads of execution
- Debugging multi-processor design in prototype board is extremely difficult because of limited hardware visibility

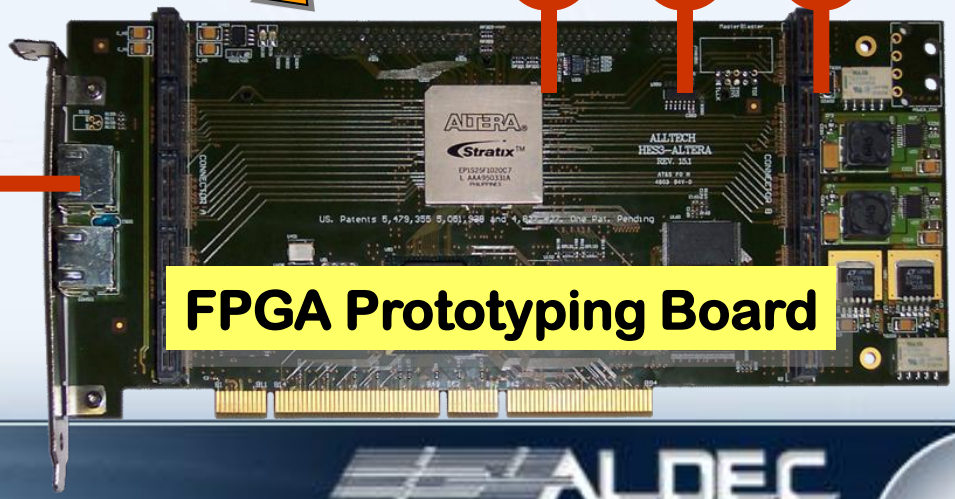
Multi-Processor Support





Multi-Channel DEBUG Controller





Summary

- Hardware/software co-design improves design quality and shortens development cycle
- True hardware/software co-design can be accomplished only in a hybrid environment with HDL Simulator and FPGA prototype board
- Smart Clock patented technology is used to synchronize HDL simulator with FPGA operating on real speeds
- Multi-processor designs are more demanding regarding software debug.
  - Each NIOS-2 processor instance should have its own independent debugging channel connected to NIOS-2 IDE debugger

Thank You For Watching

Aldec, Inc.
Corporate Headquarters
2260 Corporate Circle
Henderson, NV 89074 USA

Tel: (702) 990-4400
Toll Free: (800) 487-8743
Fax: (702) 990-4414
sales@aldec.com