Efficient System-Level DSP Design Flow with WiMAX DUC and DDC Case Study

SOPC WORLD

Agenda

- FPGA in digital signal processing (DSP) applications overview
- System-level design tools for Altera® FPGAs
  - Overview
  - Introduction to DSP Builder
  - DSP Builder design flow walkthrough
- Case study: WiMAX digital upconverter (DUC) and digital downconverter (DDC) design

SOPC WORLD

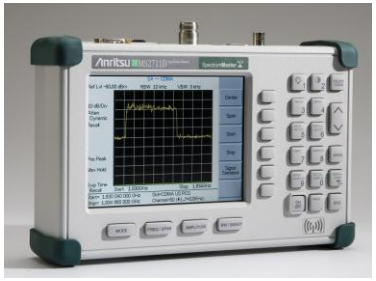
FPGA in DSP Applications

Automotive

Communications

Consumer

Industrial



Test and Measurement

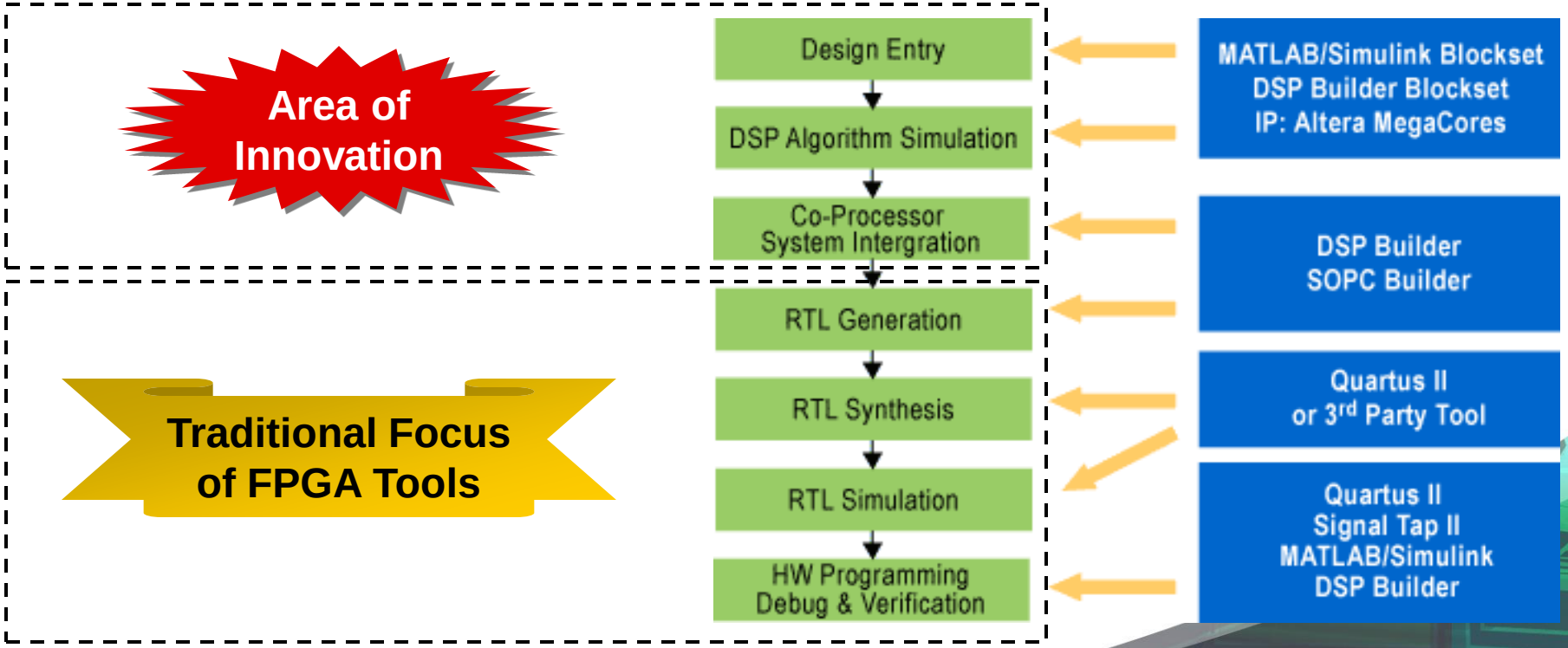
Military

Broadcast

Medical

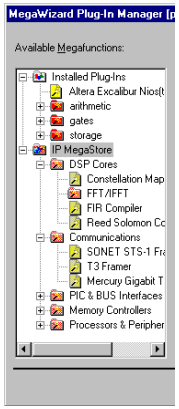
SOPC WORLD

DSP Design Flow in FPGA



SOPC WORLD

Altera System-Level Design Tools



MegaWizard® Plug-In

OpenCore™

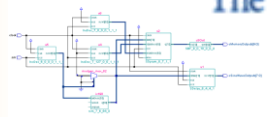
IP Integration

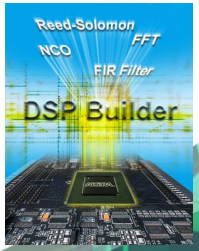
QUARTUS® II

System Integration

Software Development

The MathWorks



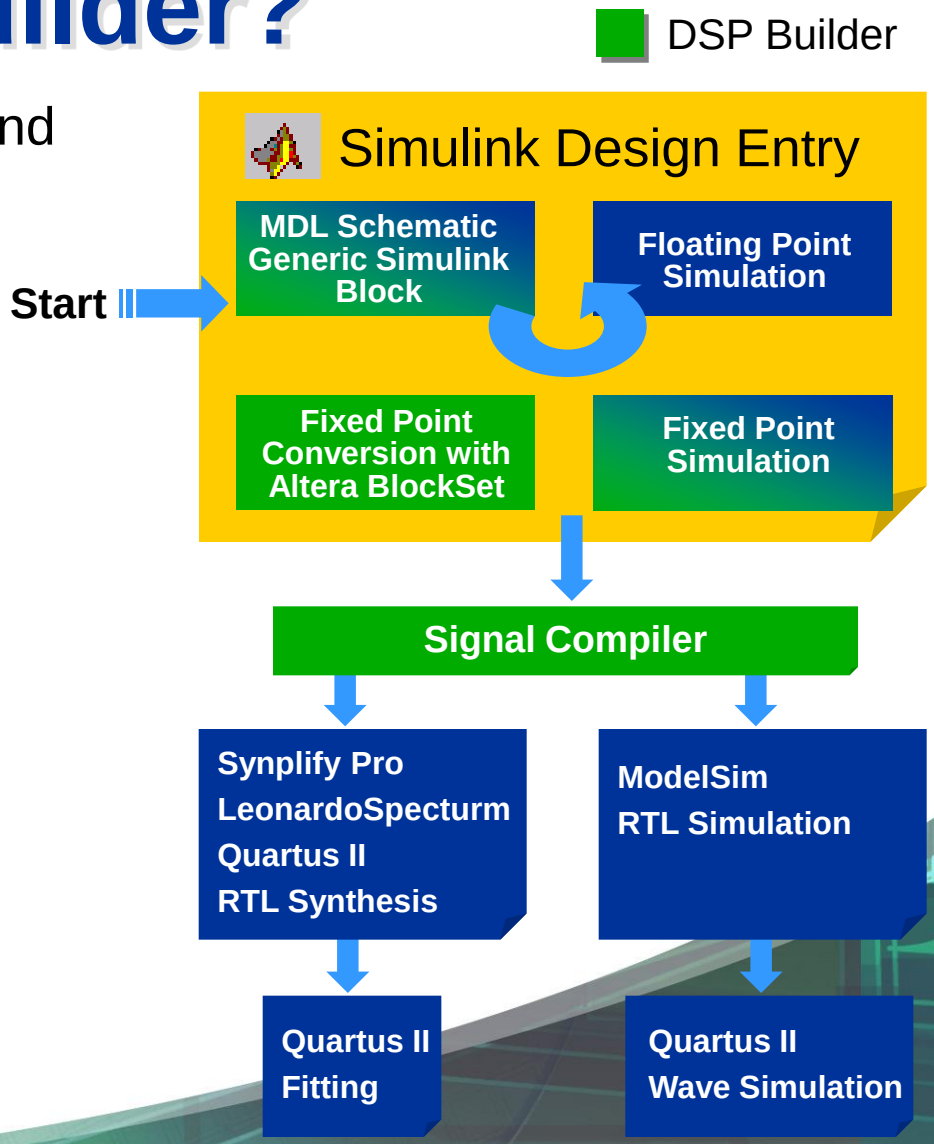


DSP Algorithm Development

SOPC WORLD

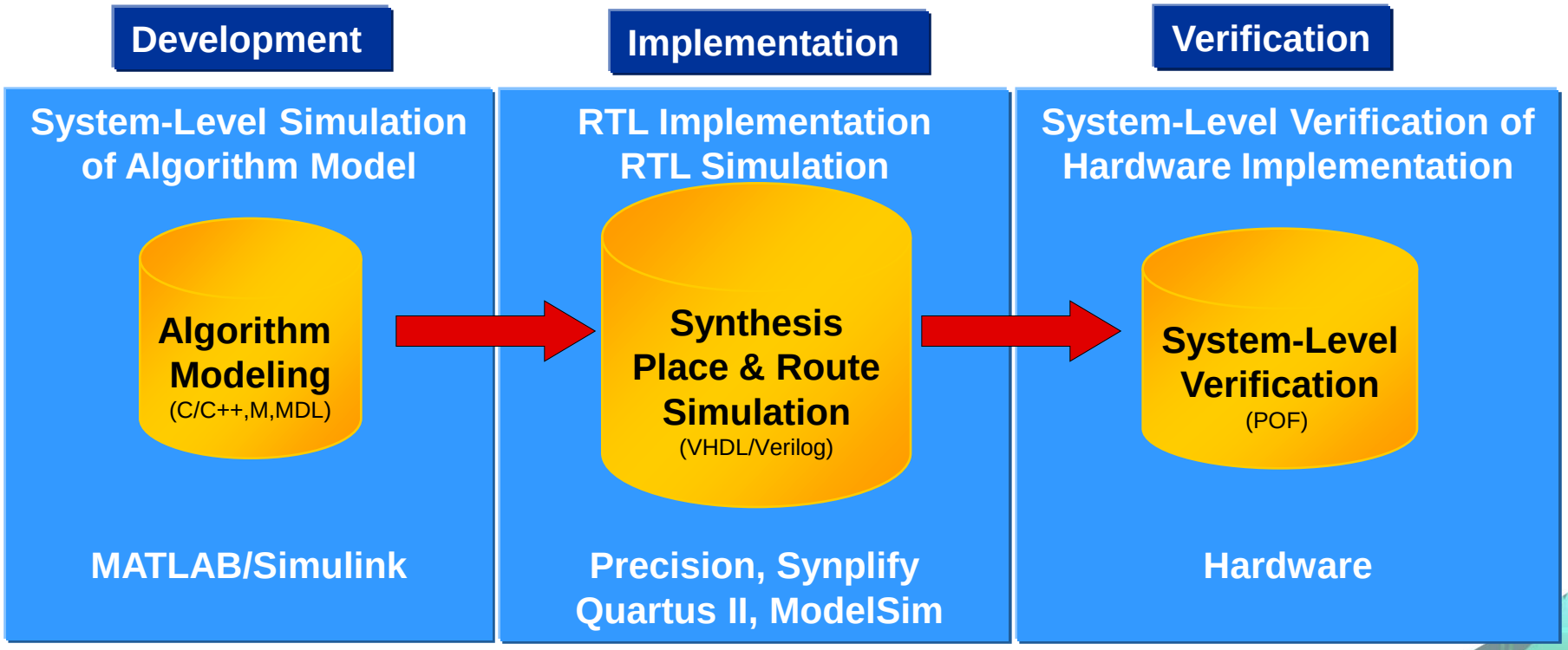
What is DSP Builder?

- Interface between Quartus II and MATLAB/Simulink
- Library add-on to Simulink
- Altera blockset
  - Library of fixed-point Simulink functions
  - Uses double precision
- Altera DSP IP
  - OpenCore Plus
- SignalCompiler utility
  - Converts between Simulink and Altera domain
- Hardware debug
  - Hardware in the loop (HIL)/ SignalTap® logic analyzer

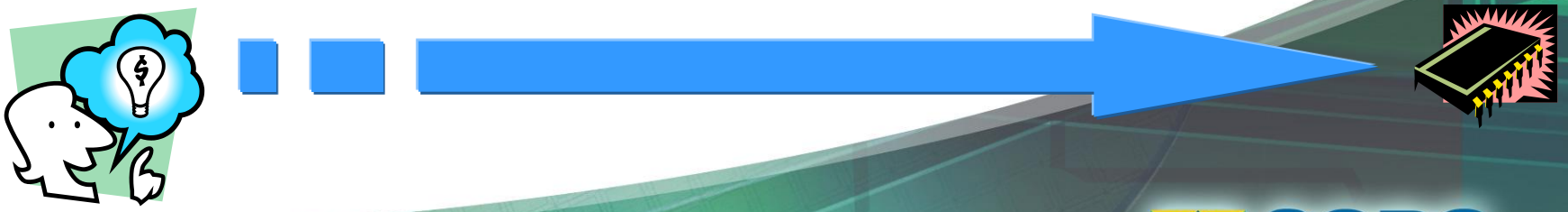


SOPC WORLD

Traditional System Design Tool Flow

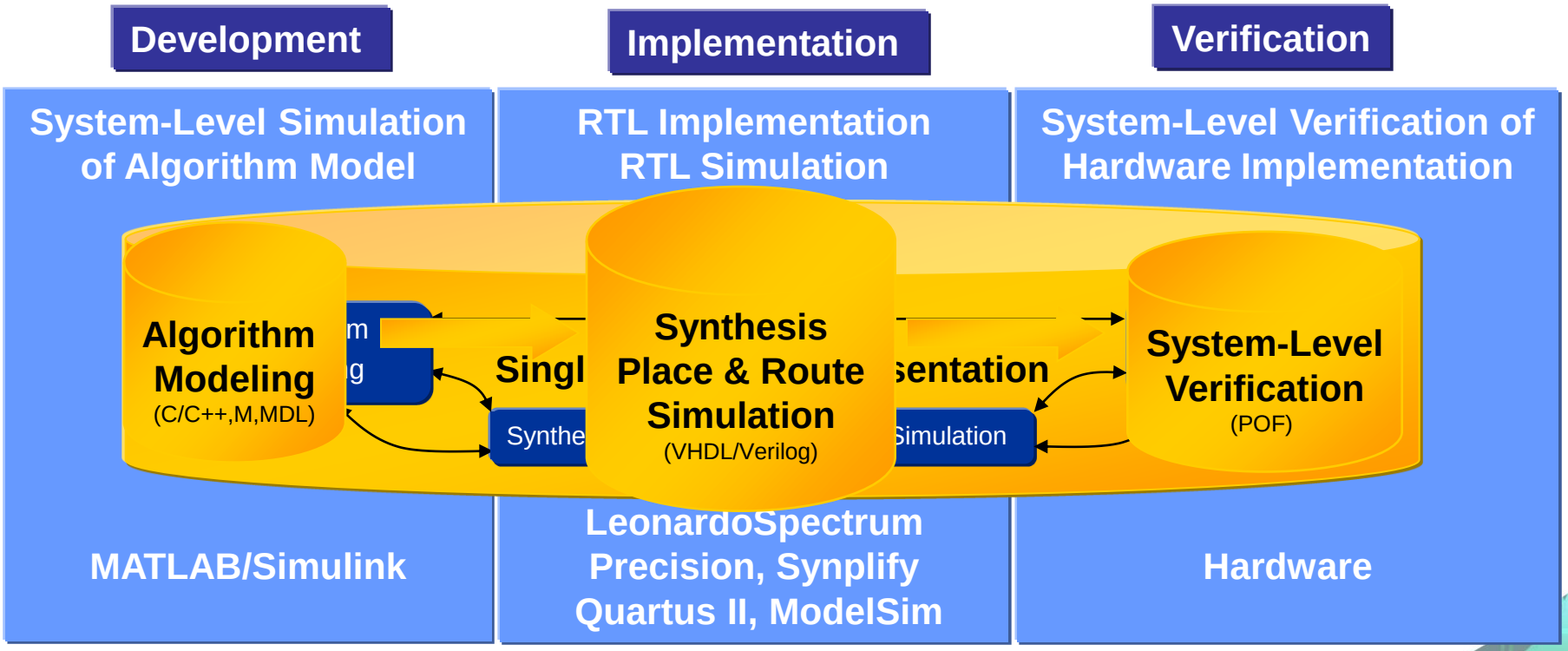


System Algorithm Design and FPGA Design Separated



SOPC WORLD

DSP Builder - Simulink Design Flow



System Algorithm Design and FPGA Design Integrated

SOPC WORLD

DSP Builder Features

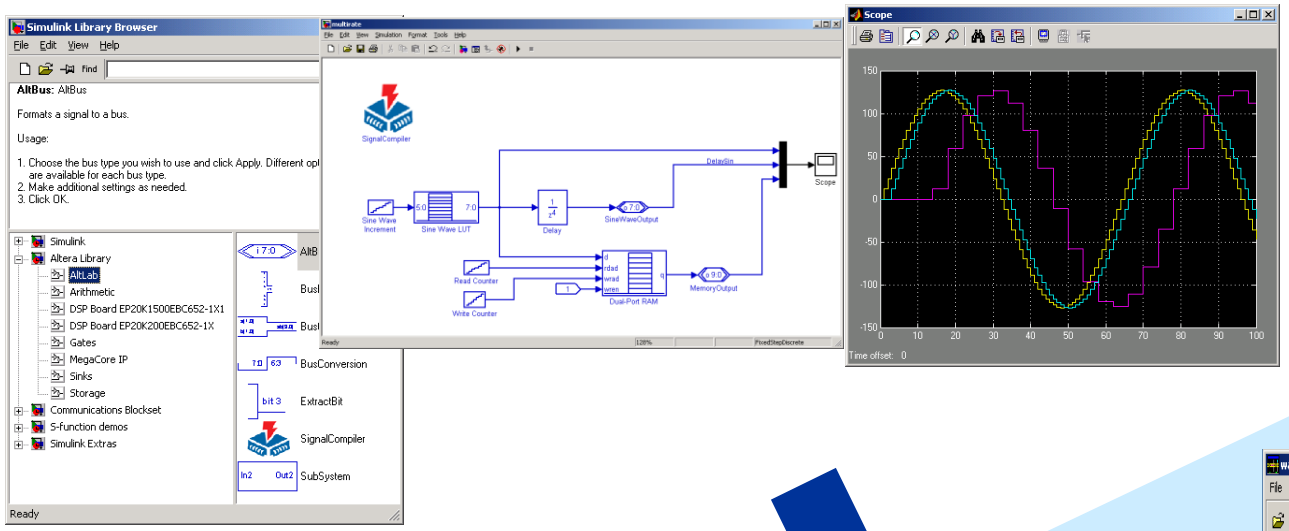
- Automatic generation of VHDL design from a MATLAB/Simulink representation
- Automatic generation of VHDL testbench
  - Captures stimulus from Simulink, writes testbench
- HDL import
  - Reads in design: Verilog or VHDL, or Quartus II project
  - Creates Simulink simulation model
- SignalTap embedded logic analyzer
  - Captures internal data and it into MATLAB
- HIL testing
  - Pass vectors to/from board

SOPC WORLD

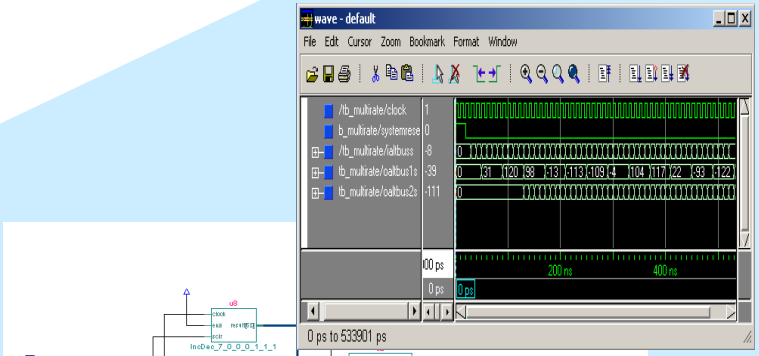
DSP Builder Benefits

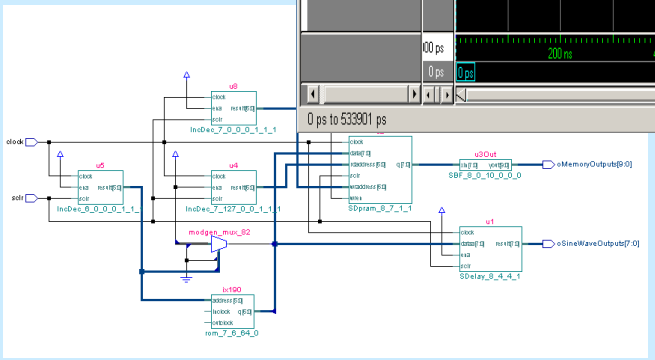
- For hardware engineer
  - Extends RTL analysis and debug capabilities to system-level tool
    - Access to MATLAB data formatting
    - Access to a large library of Simulink models
  - Speeds up simulation run time
  - Enables IP evaluation at system level
- For system-level engineer
  - Allows rapid prototyping with minimal PLD expertise
  - Provides easy access to hardware evaluation
  - Extends floating-point to fixed-point system analysis

DSP Builder Design Flow



Matlab/Simulink Domain
(C+ System Analysis)





VHDL Domain
(Implementation/Simulation)

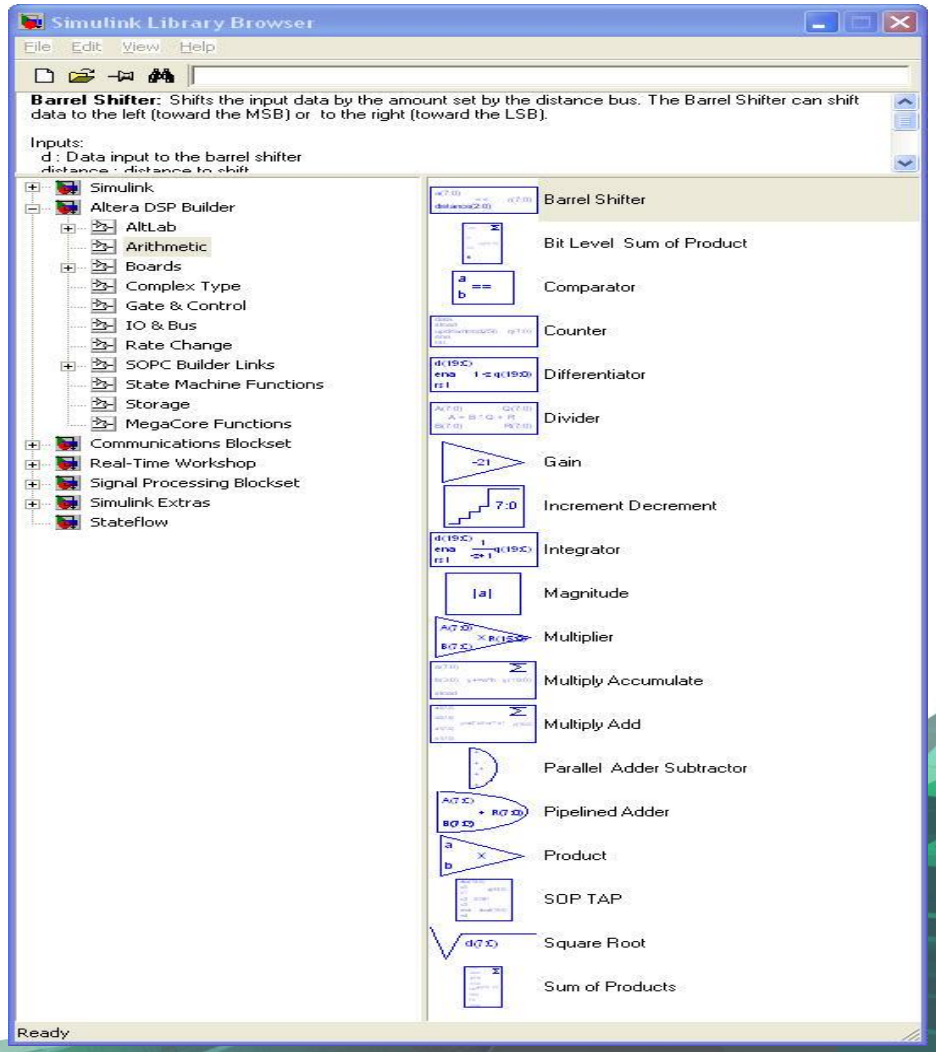
SOPC WORLD

Design Flow Overview

1) Create design in Simulink using Altera libraries
2) Simulate in Simulink
3) Add SignalCompiler to model
4) Create HDL code and generate testbench
5) Perform RTL simulation
6) Synthesize HDL code and place and route
7) Program device
8) Verify hardware: SignalTap logic analyzer/HIL

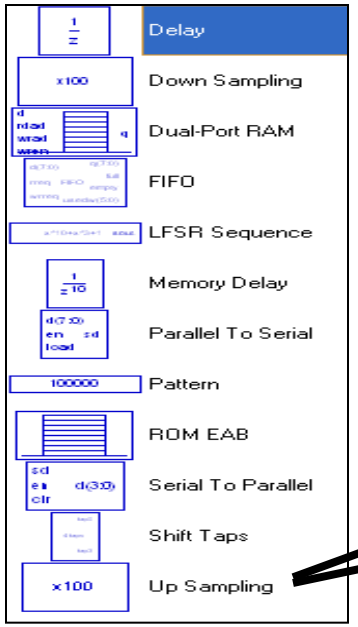
Altera DSP Builder Libraries

- AltLab
- Arithmetic
- Boards
- Complex type
- Gate and control
- I/O and bus
- Rate change
- SOPC Builder links
- State machine functions
- Storage
- MegaCore functions

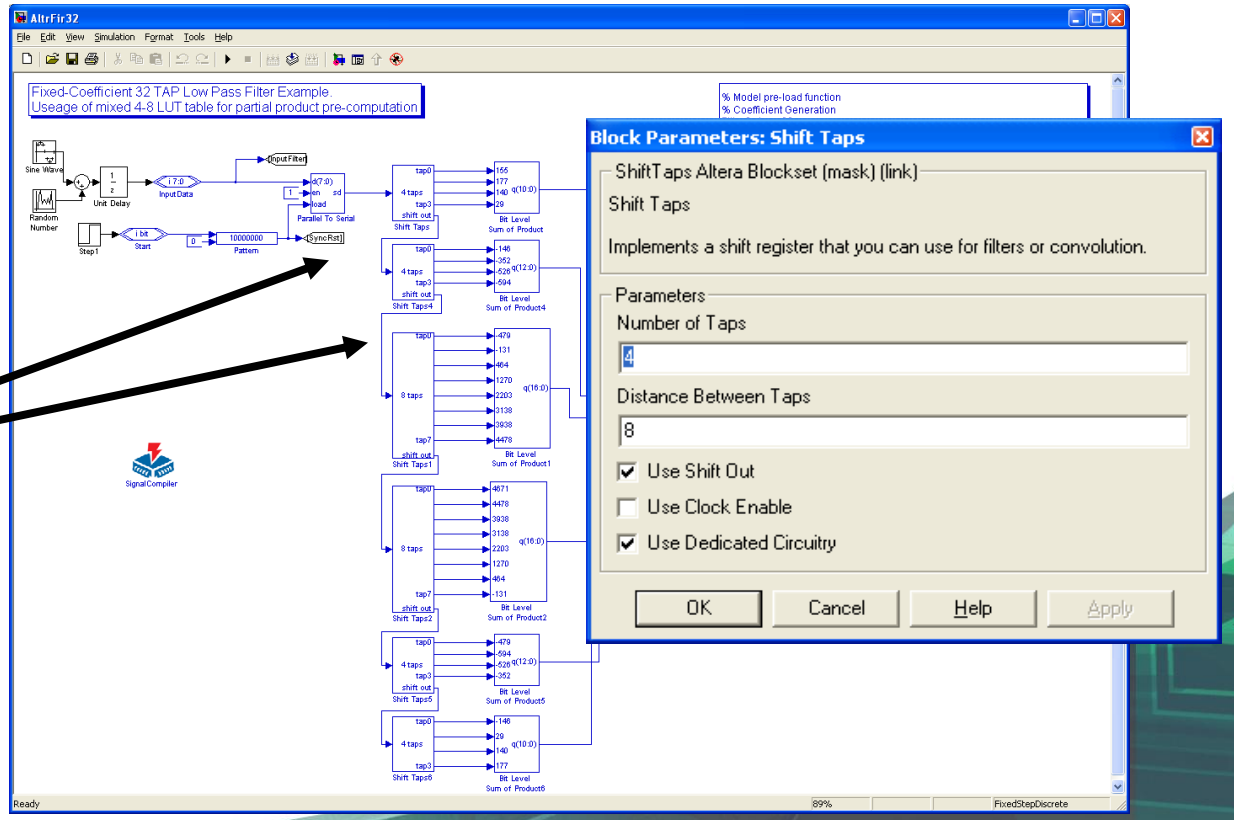


SOPC WORLD

Step 1: Create Design in Simulink Using Altera Libraries

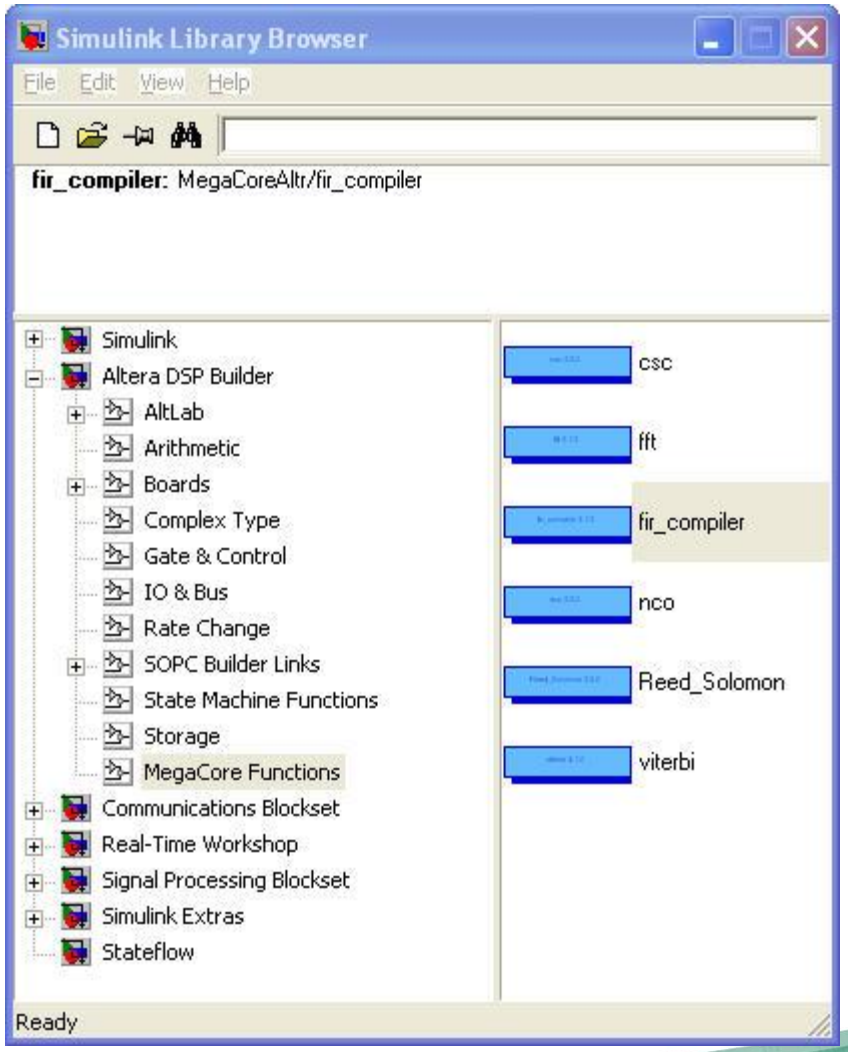


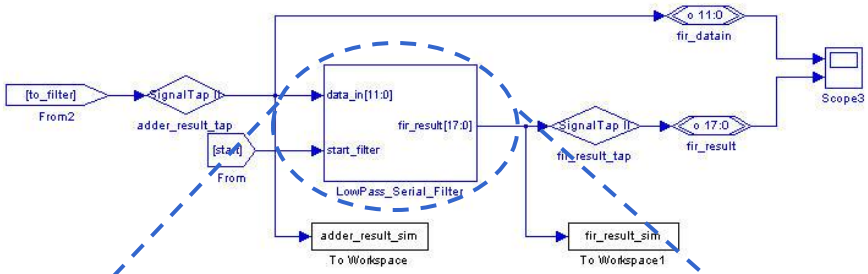
- Drag and drop library blocks into Simulink design and parameterize each block

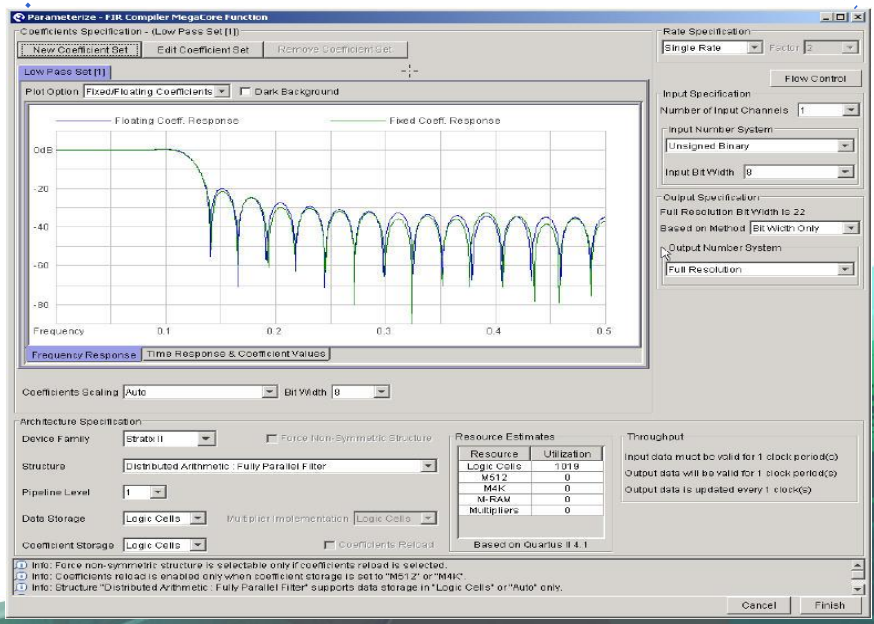


SOPC WORLD

Parameterization of IP MegaCore Functions

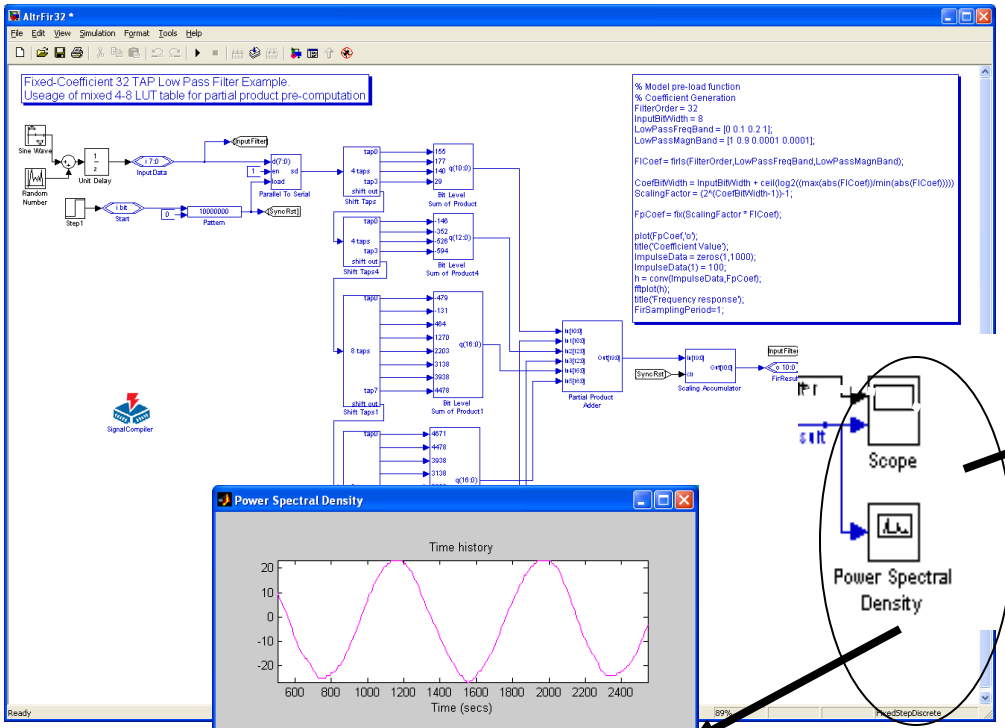


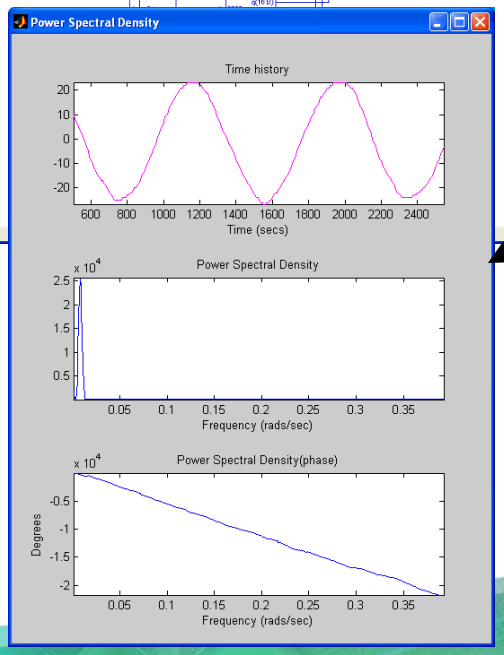


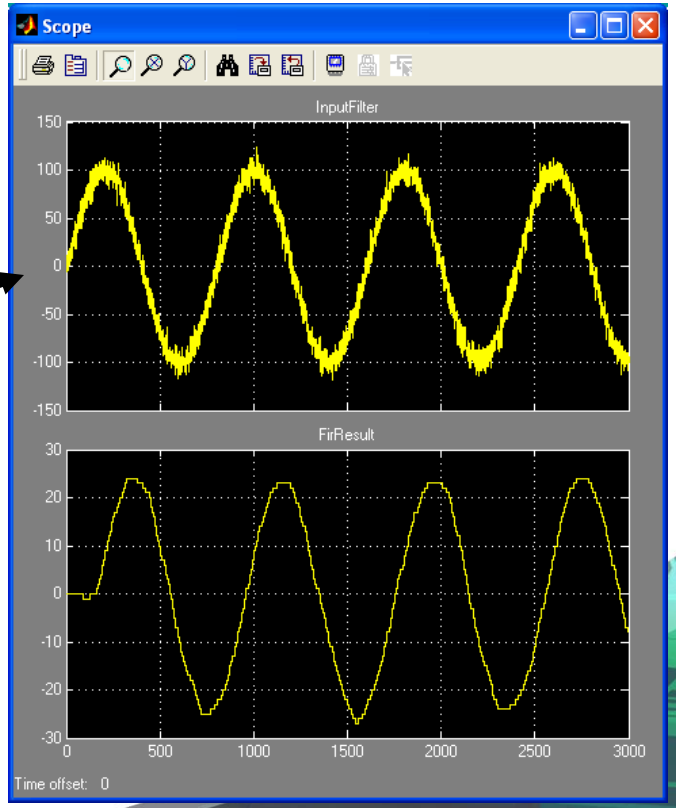


SOPC WORLD

Step 2: Simulate in Simulink

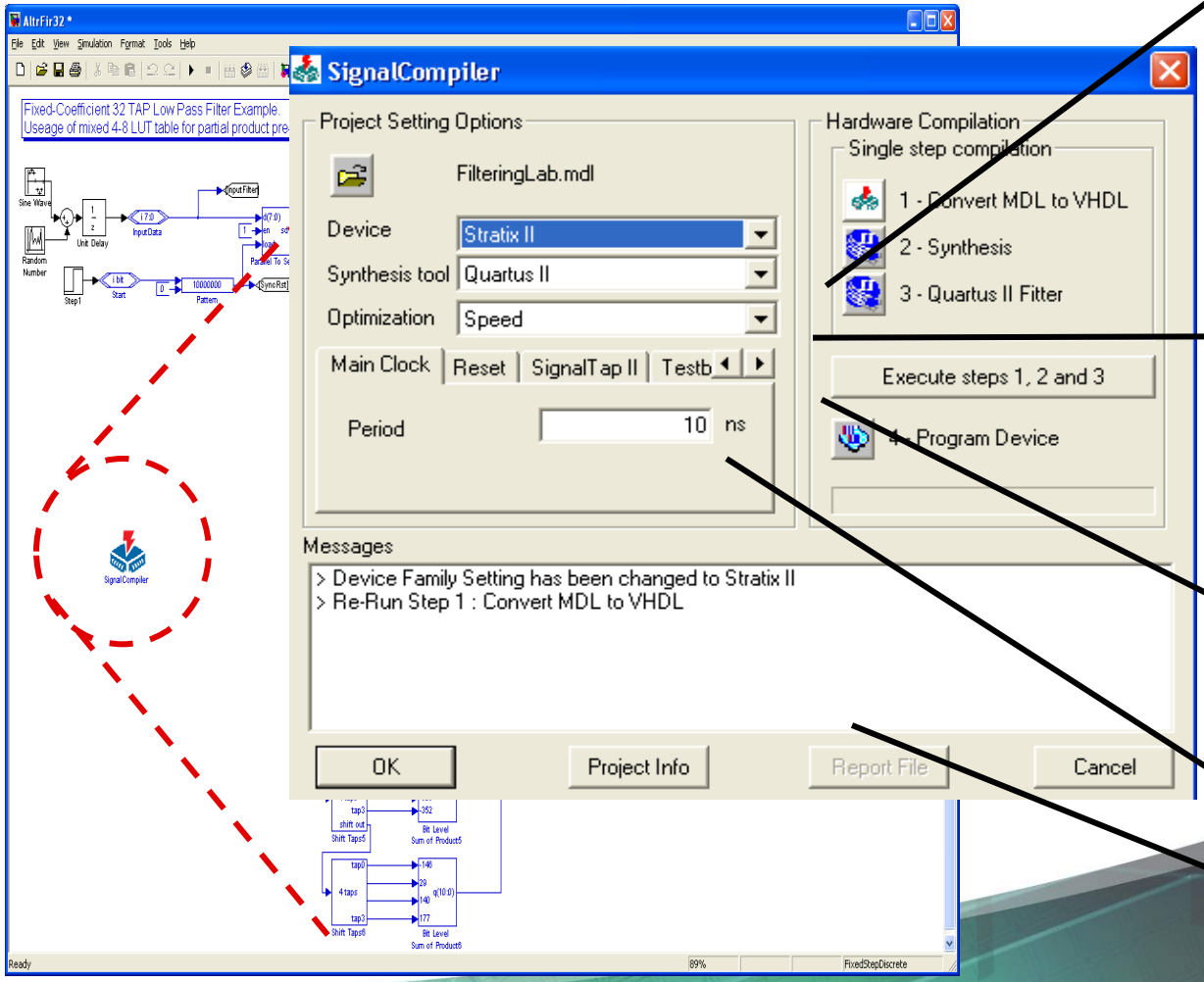






SOPC WORLD

Step 3: Add SignalCompiler to Model to Generate HDL Code



- Stratix and Stratix II
- Stratix GX
- Cyclone & Cyclone II
- ACEX® 1K
- Mercury™
- FLEX® 10K and FLEX 6000
- Development Boards
- APEX™ 20K/E/C
- APEX II

- LeonardoSpectrum™
- Synplify
- Precision
- Quartus II

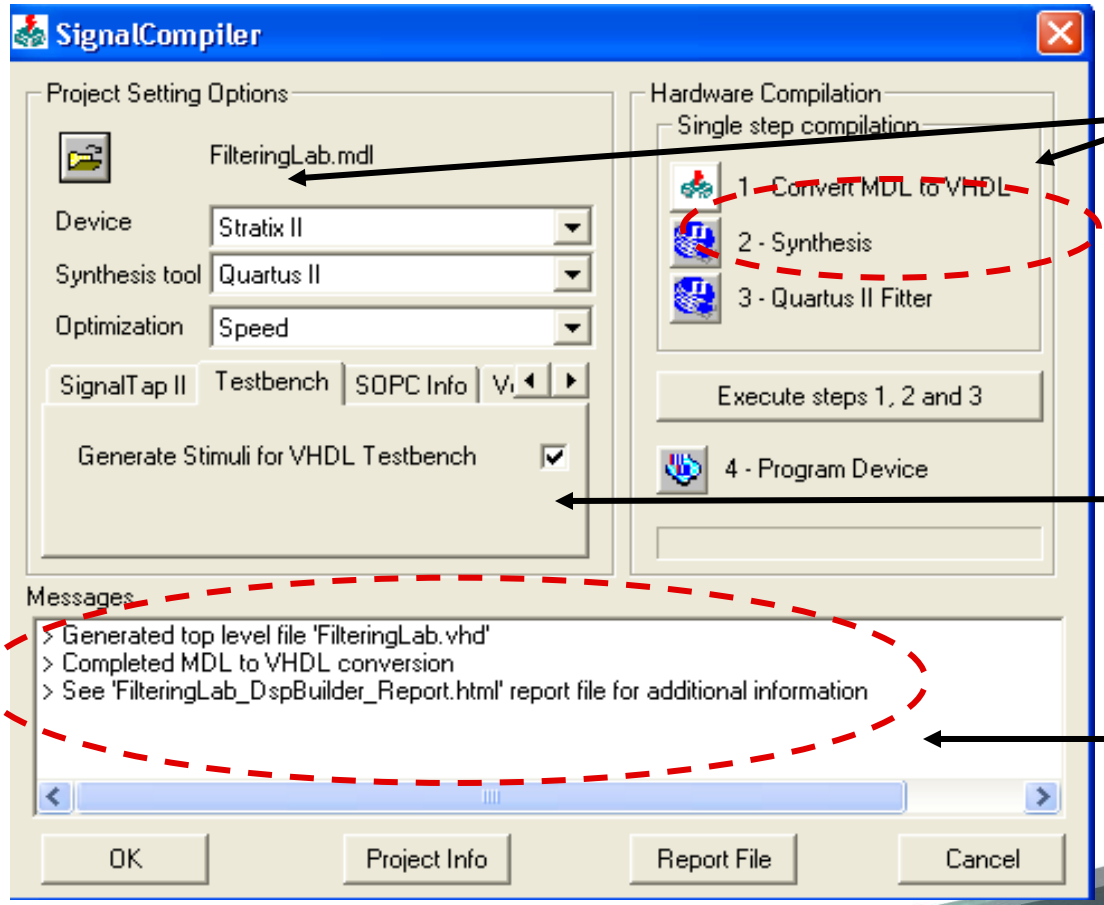
- Speed
- Area
- Balanced
- Fast fit – no timing optimization
- Use current Quartus II project

Testbench generation

Message window

SOPC WORLD

Step 4: Create HDL Code and Generate Testbench



FilteringLab.mdl

Enable "Generate Stimuli for VHDL Testbench" Button

FilteringLab.vhd

SOPC WORLD

HDL Code Generation

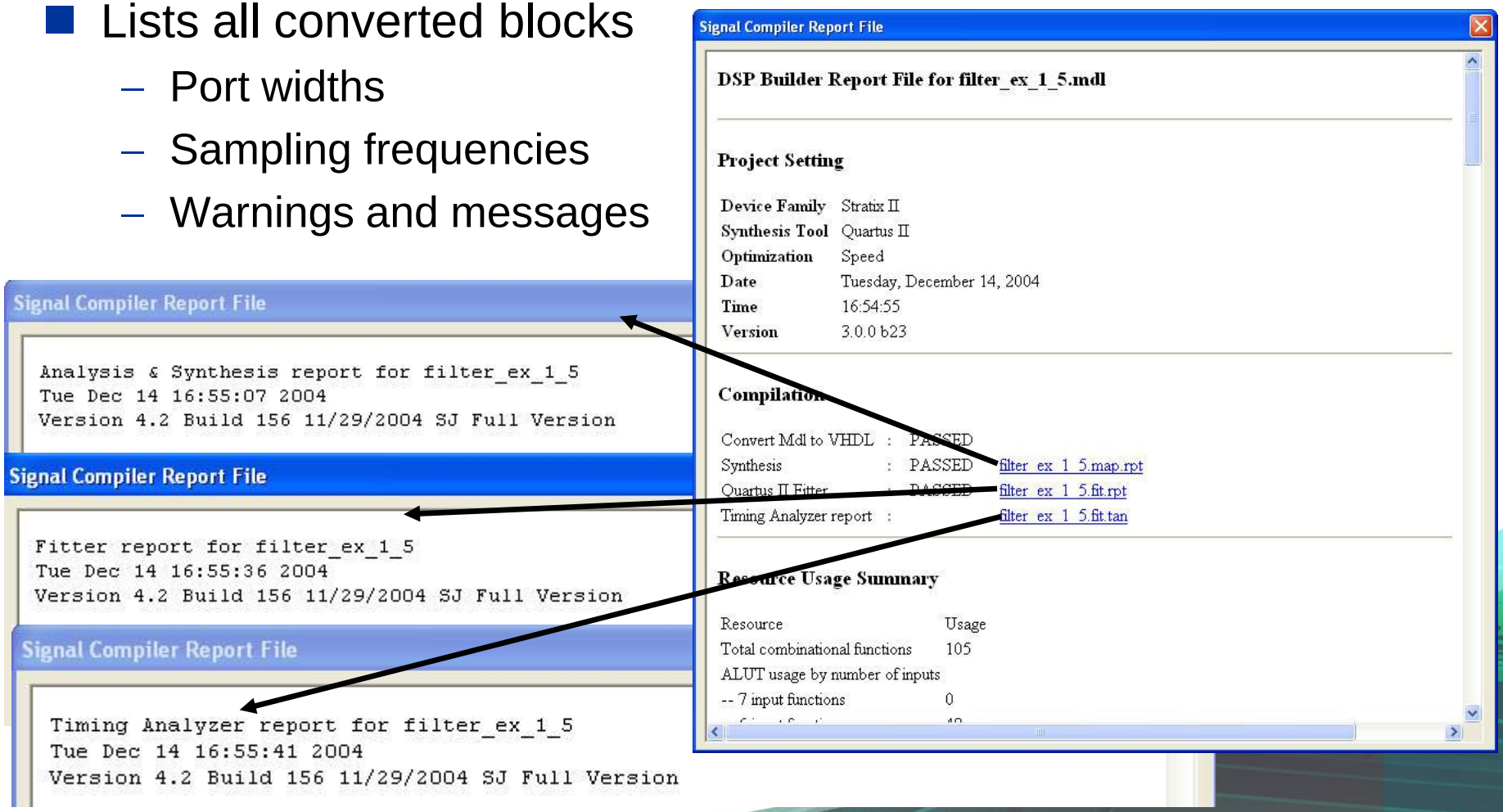
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

library dspbuilder;
use dspbuilder.dspbuilderblock.all;

library lpm;
use lpm.lpm_components.all;

Entity FilteringLab is
        Port(
                clock           :       in std_logic;
                sclrp           :       in std_logic:='0';
                iSW3s           :       in std_logic_vector(7 downto 0);
                clk_out1        :       out std_logic;
                oLED1s          :       out std_logic;
                ofir_datains    :       out std_logic_vector(11 downto 0);
                ofir_results    :       out std_logic_vector(17 downto 0)
            );
end FilteringLab;

architecture aDspBuilder of FilteringLab is


signal  SAfir_datainO   :       std_logic_vector(11 downto 0);
signal sclr     :       std_logic:='0';

--Using PLL to drive pin Y3 (DAC clock source)
component dspboard1S25_pll PORT(
                                        inclk0          :       IN STD_LOGIC ;
                                        c0              :       OUT STD_LOGIC);
end component ;

signal board_clk_out_int : std_logic;

signal  A0W             :       std_logic_vector(7 downto 0);
signal  A1W             :       std_logic;
signal  A2W             :       std_logic;
signal  A3W             :       std_logic_vector(12 downto 0);
signal  A4W             :       std_logic;
signal  A5W             :       std_logic_vector(13 downto 0);
signal  A6W             :       std_logic_vector(17 downto 0);
signal  A7W             :       std_logic_vector(12 downto 0);
signal  A8W             :       std_logic_vector(12 downto 0);
signal  A9W             :       std_logic_vector(13 downto 0);
signal  A10W            :       std_logic_vector(17 downto 0);
signal  A11W            :       std_logic_vector(12 downto 0);
signal  A12W            :       std_logic_vector(12 downto 0);
signal  A13W            :       std_logic;
signal  A14W            :       std_logic;
signal  A15W            :       std_logic;
signal  A16W            :       std_logic;
signal  A17W            :       std_logic;
signal ExtExtract4  :   std_logic_vector(7 downto 0);
```

```
-- SubSystem Hierarchy - Simulink Block "LowPass_Serial_Filter"
component LowPass_Serial_Filter
                port(
                        clock           :       in std_logic ;
                        sclr            :       in std_logic ;
                        iInputDatas     :       in std_logic_vector(11 downto 0) ;
                        iStarts         :       in std_logic ;
                        oFirResults     :       out std_logic_vector(17 downto 0)
                        );
end component ;


-- SubSystem Hierarchy - Simulink Block "SineWave_Generator"
component SineWave_Generator
                port(
                        clock           :       in std_logic ;
                        sclr            :       in std_logic ;
                        iStarts         :       in std_logic ;
                        osin_833_33kHzs  :      out std_logic_vector(12 downto 0) ;
                        osin_83_33kHzs  :       out std_logic_vector(12 downto 0)
                        );
end component ;


Begin

assert (1<0)  report altversion severity Note;


-- Output - I/O assignment from Simulink Block  "LED1"
oLED1s          <=      A14W;
ofir_datains    <=      SAfir_datainO;

-- Output - I/O assignment from Simulink Block  "fir_result"
ofir_results    <=      A10W;
sclr            <=      sclrp;

-- Input - I/O assignment from Simulink Block "iSW3s"
A0W             <=       iSW3s;

-- Bit Extraction - Simulink Block "ExtractBit"
ExtExtract4     <=      A0W;
A1W             <=      ExtExtract4(0);
sclr_u5         <=              A16W or sclr;
sclr_u6         <=              A17W or sclr;

-- Simulink Block "VCC"
A15W            <=              '1';

-- Simulink Block "GND"
A16W            <=              '0';

-- Simulink Block "GND1"
```


SOPC WORLD

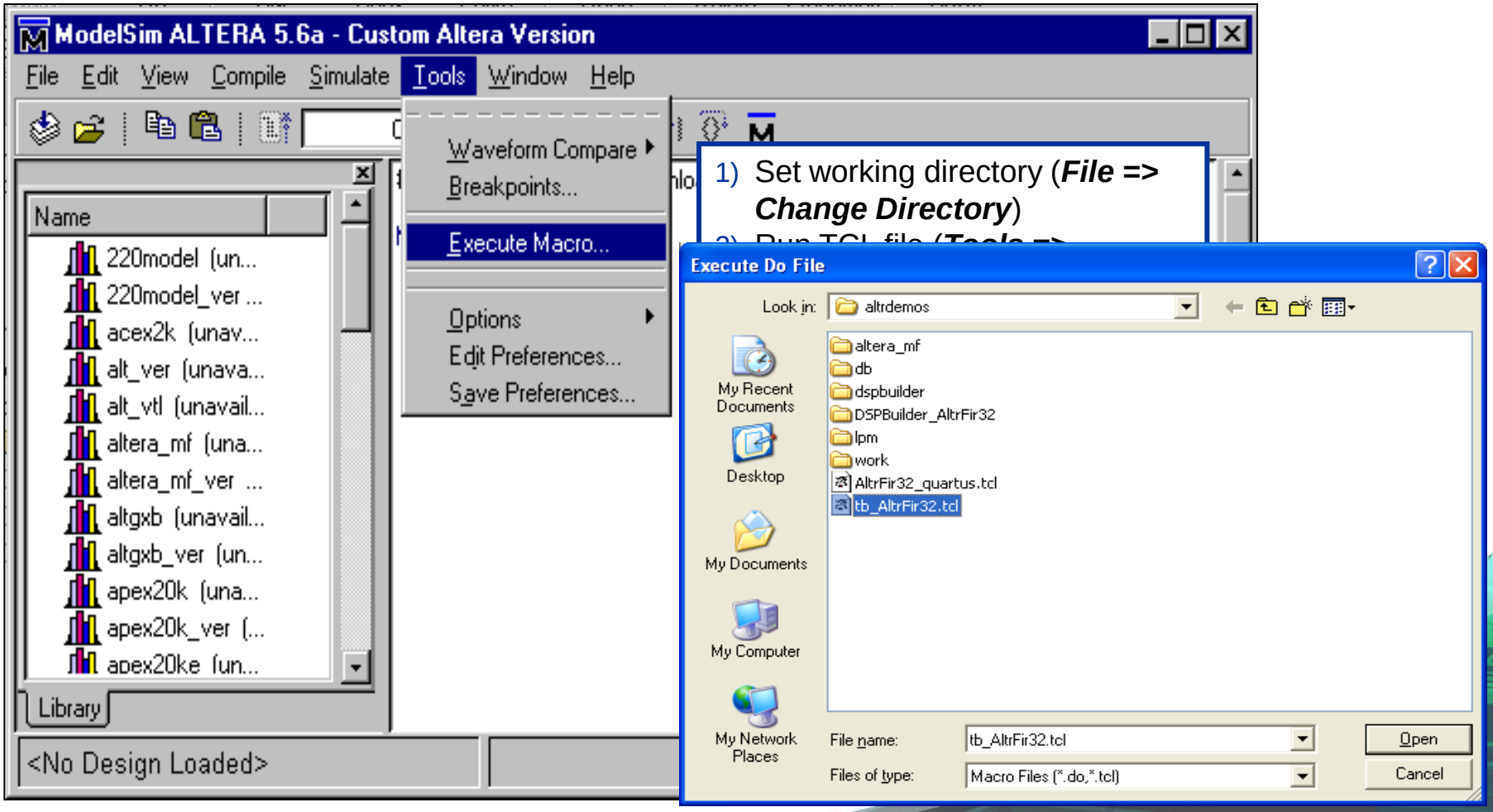
DSP Builder Report File

- Lists all converted blocks
  - Port widths
  - Sampling frequencies
  - Warnings and messages



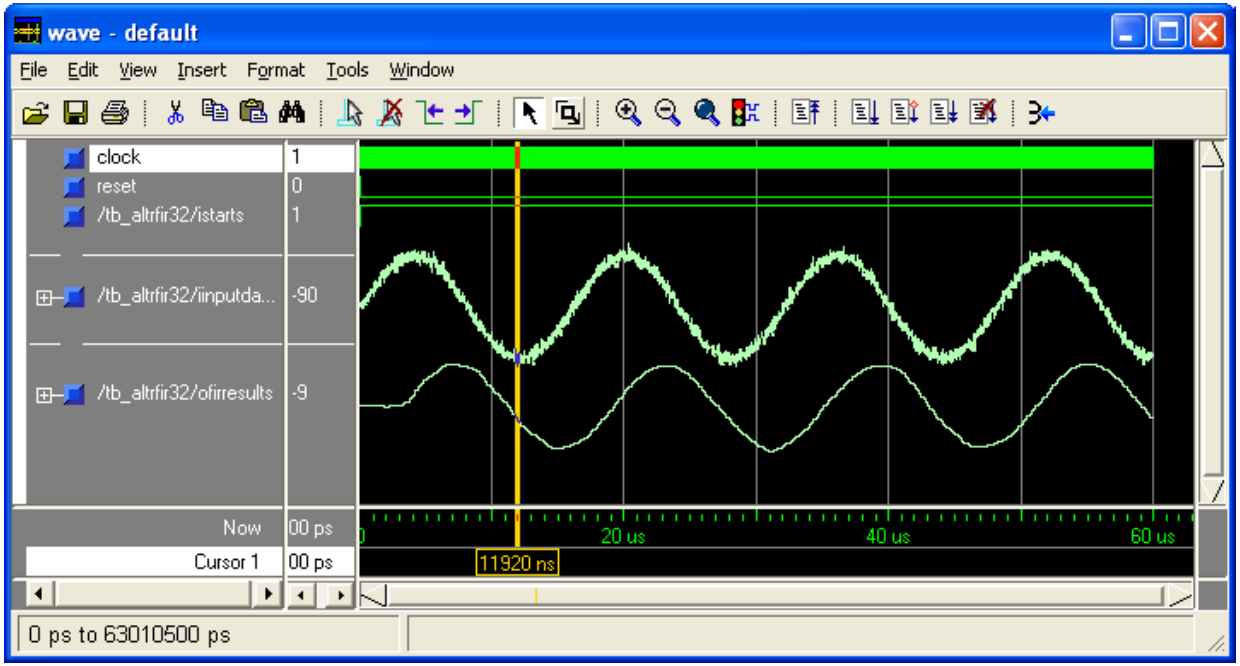
SOPC WORLD

Step 5: Perform RTL Simulation (ModelSim)

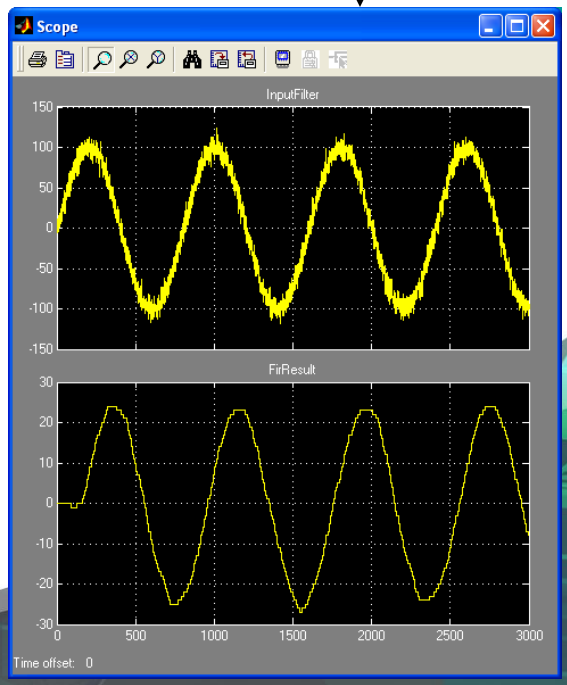


SOPC WORLD

Perform Verification



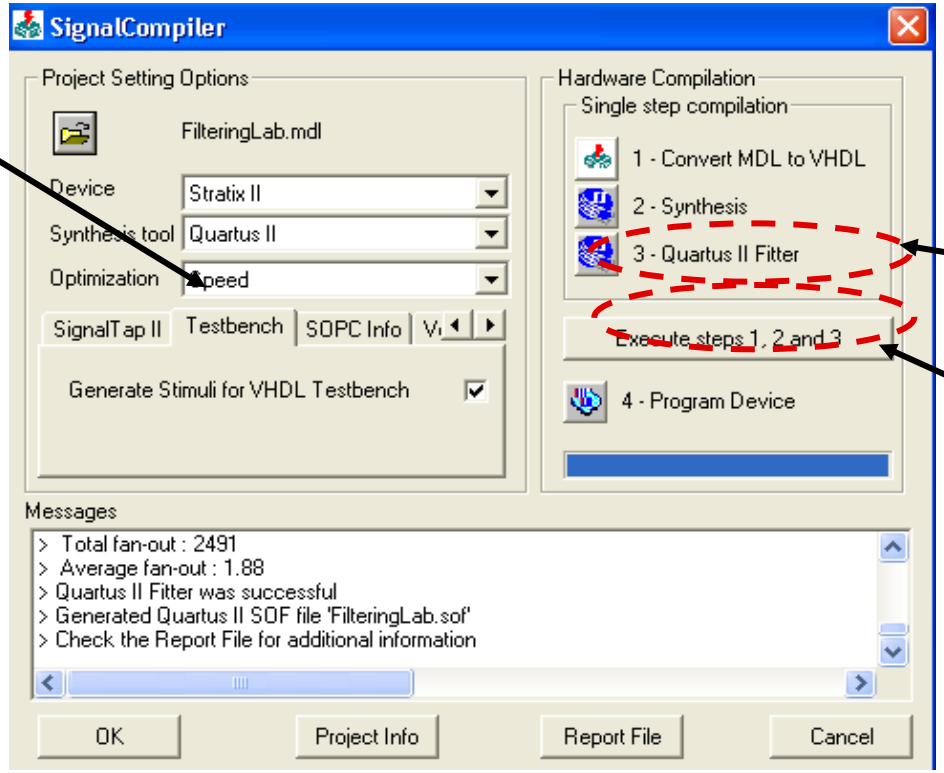
ModelSim vs. Simulink



SOPC WORLD

Step 6: Synthesize HDL and Place and Route

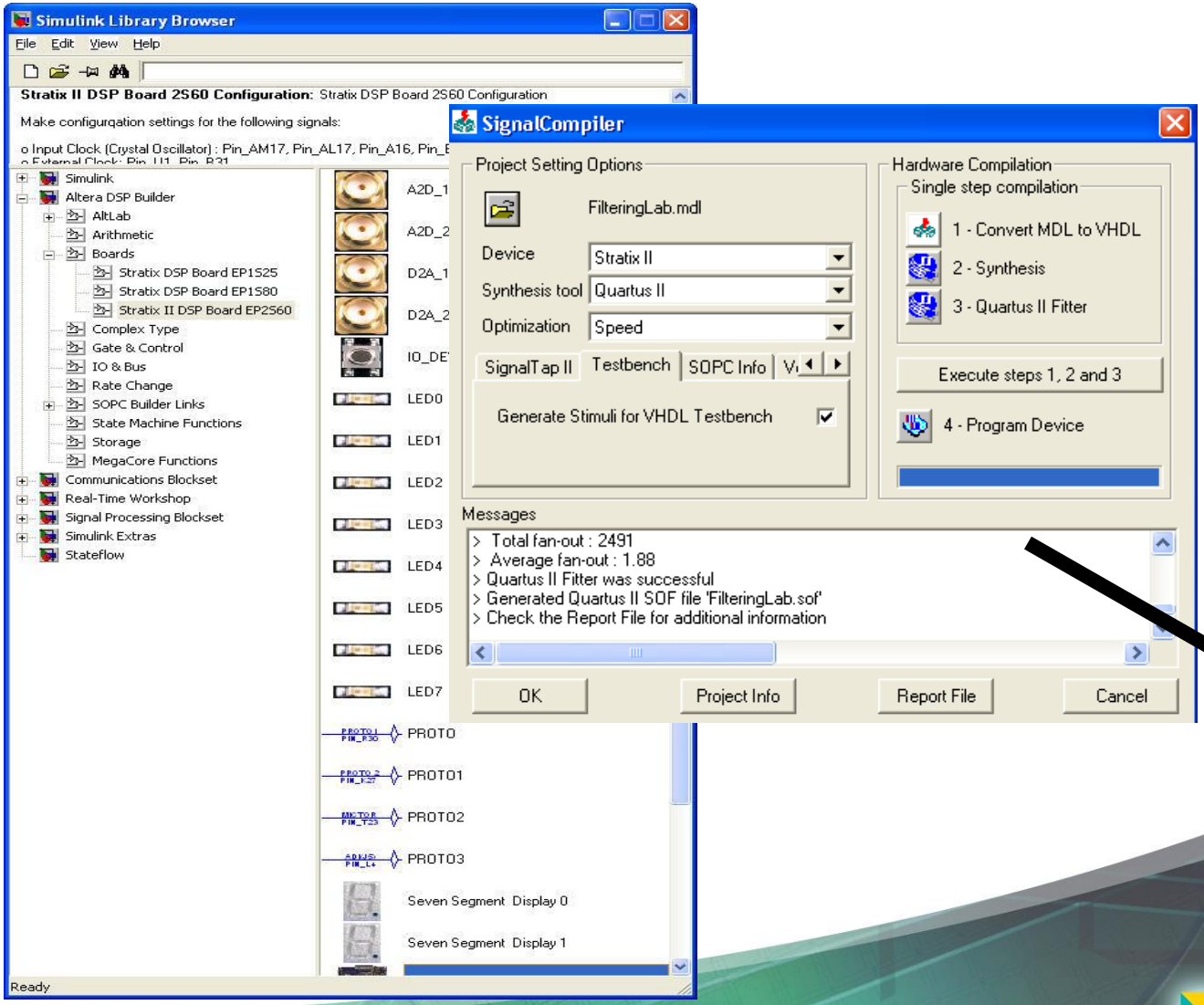
- Leonardo Spectrum
- Synplify
- Quartus II



- Synthesis
- Quartus II Fitter

SOPC WORLD

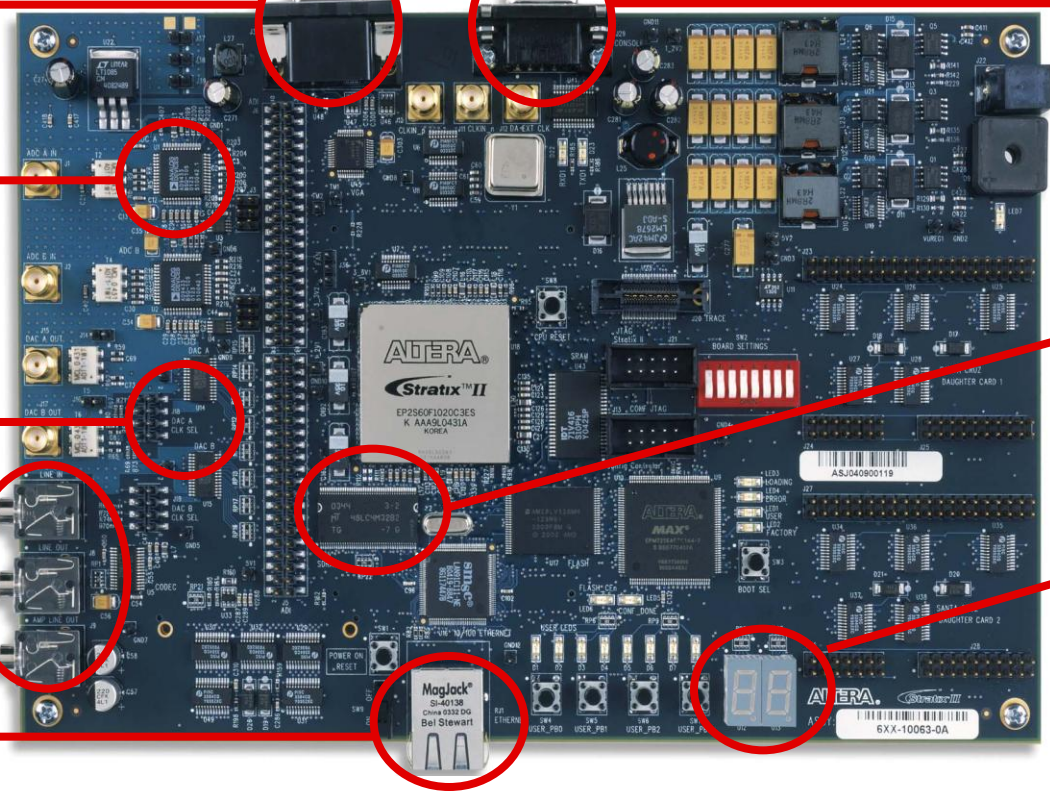
Step 7: Program Device



Download Design to DSP Development Kits

SOPC WORLD

Stratix II DSP Development Board



VGA Connector

12-Bit, 125-MHz A/D

14-Bit, 165-MHz D/A

16-Bit Audio Codec

10/100 Ethernet

9-Pin RS232 Connector

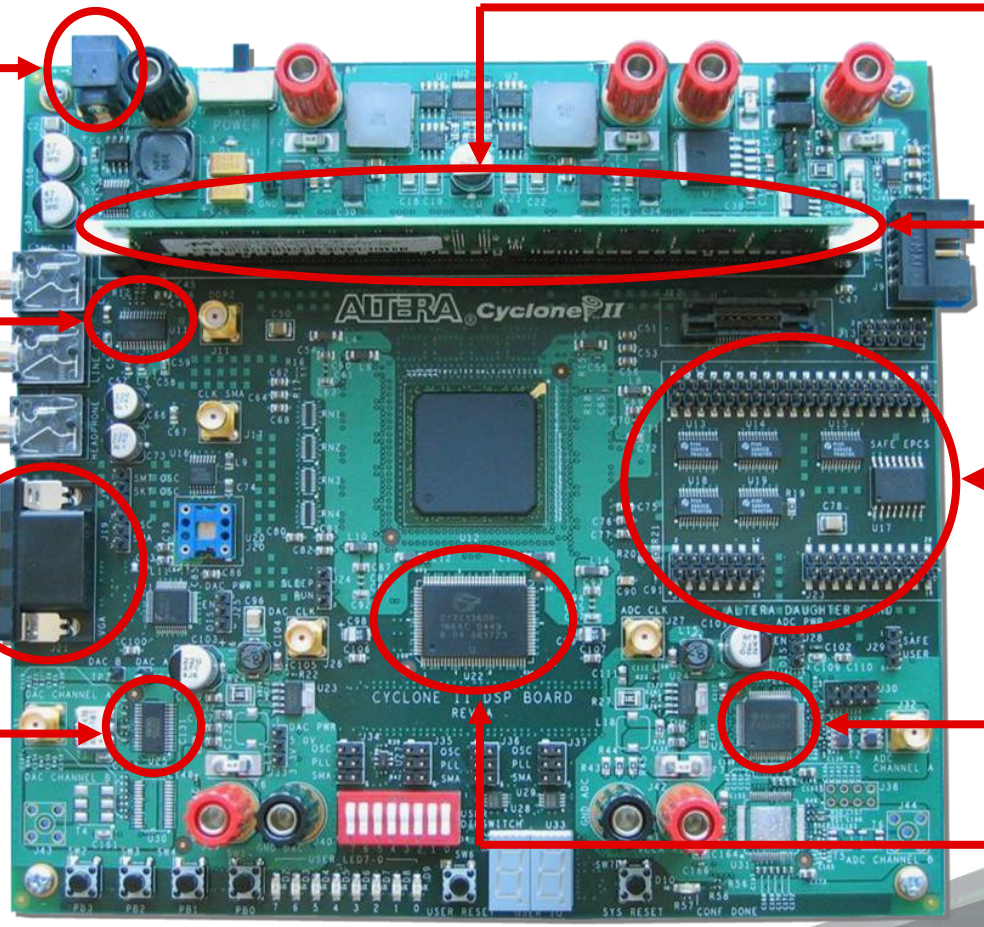
16-Mbyte SDR SDRAM

Texas Instruments DSP Connector on Underside of Board

Available with EP2S60 or EP2S180 Device

SOPC WORLD

Cyclone II DSP Development Board



Power Supply Connector

Stereo Codec

VGA Connector

14-Bit 165-MHz DAC

256-Mbyte DDR2 DIMM

TI EMIF Connector on Underside of Board

Expansion Prototype Connector

12-Bit, 125-MHz ADC

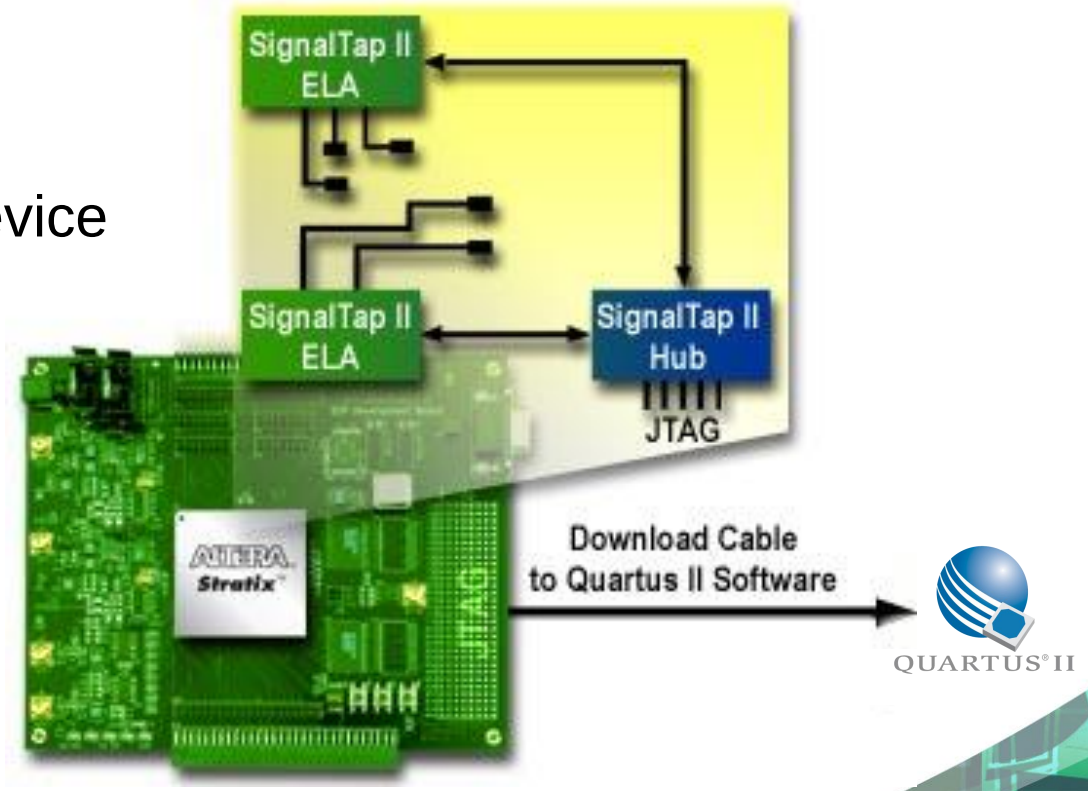
1-Mbyte Synchronous SRAM

Now Available with EP2C70 Device

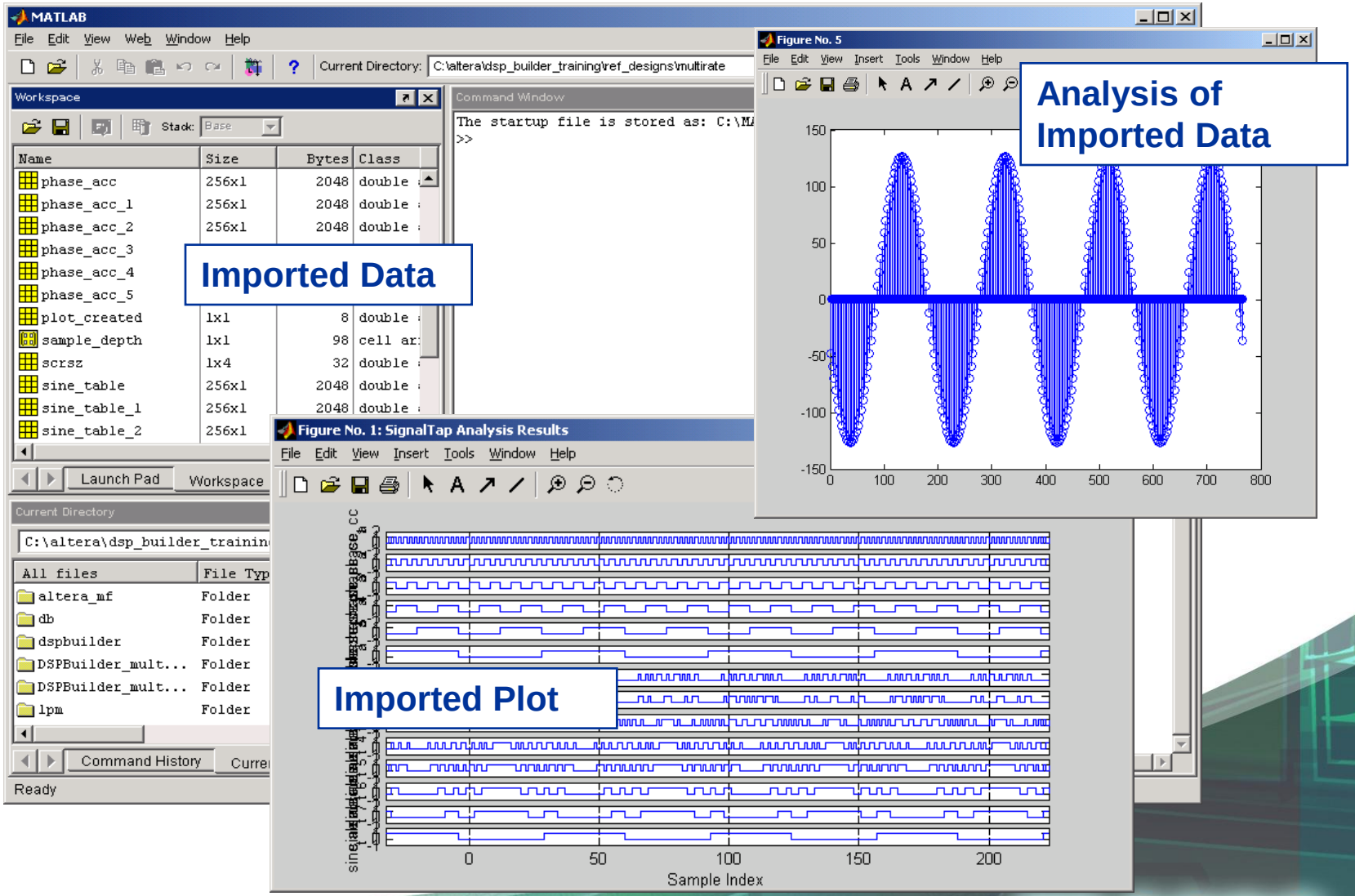
SOPC WORLD

Step 8: SignalTap II Logic Analyzer

- Embedded logic analyzer (ELA)
  - Downloads into device with design
  - Captures state of internal nodes
  - Uses JTAG for communication

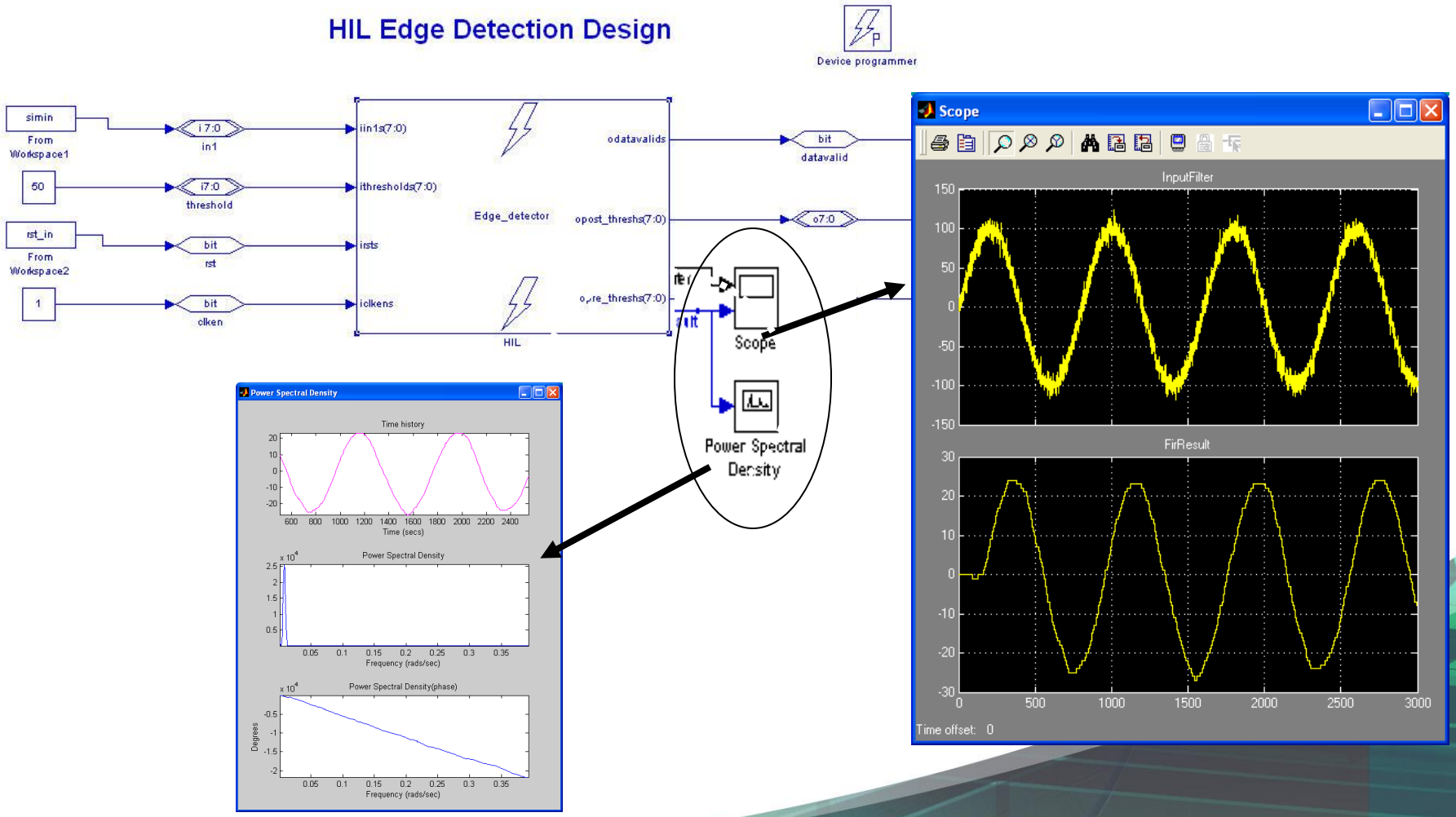


SignalTap II Logic Analyzer



SOPC WORLD

Hardware in Loop (HIL)



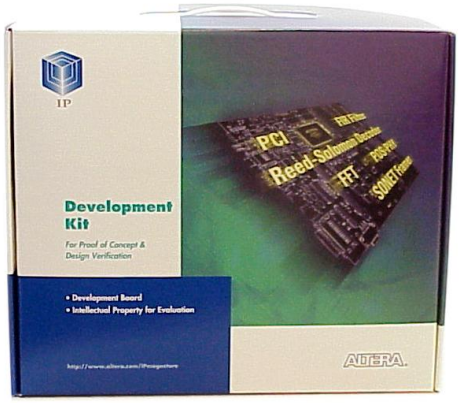
SOPC WORLD

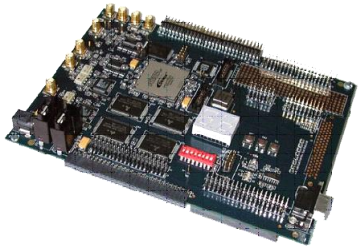
Design Flow Review

1) Create design in Simulink using Altera libraries
2) Simulate in Simulink
3) Add SignalCompiler to model
4) Create HDL code and generate testbench
5) Perform RTL simulation
6) Synthesize HDL code and place and route
7) Program device
8) Verify hardware: SignalTap logic analyzer HIL

SOPC WORLD

Altera DSP Development Kits

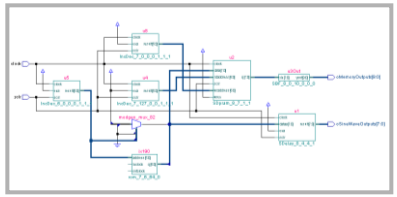




OpenCore® plus

30-Day Evaluation Version

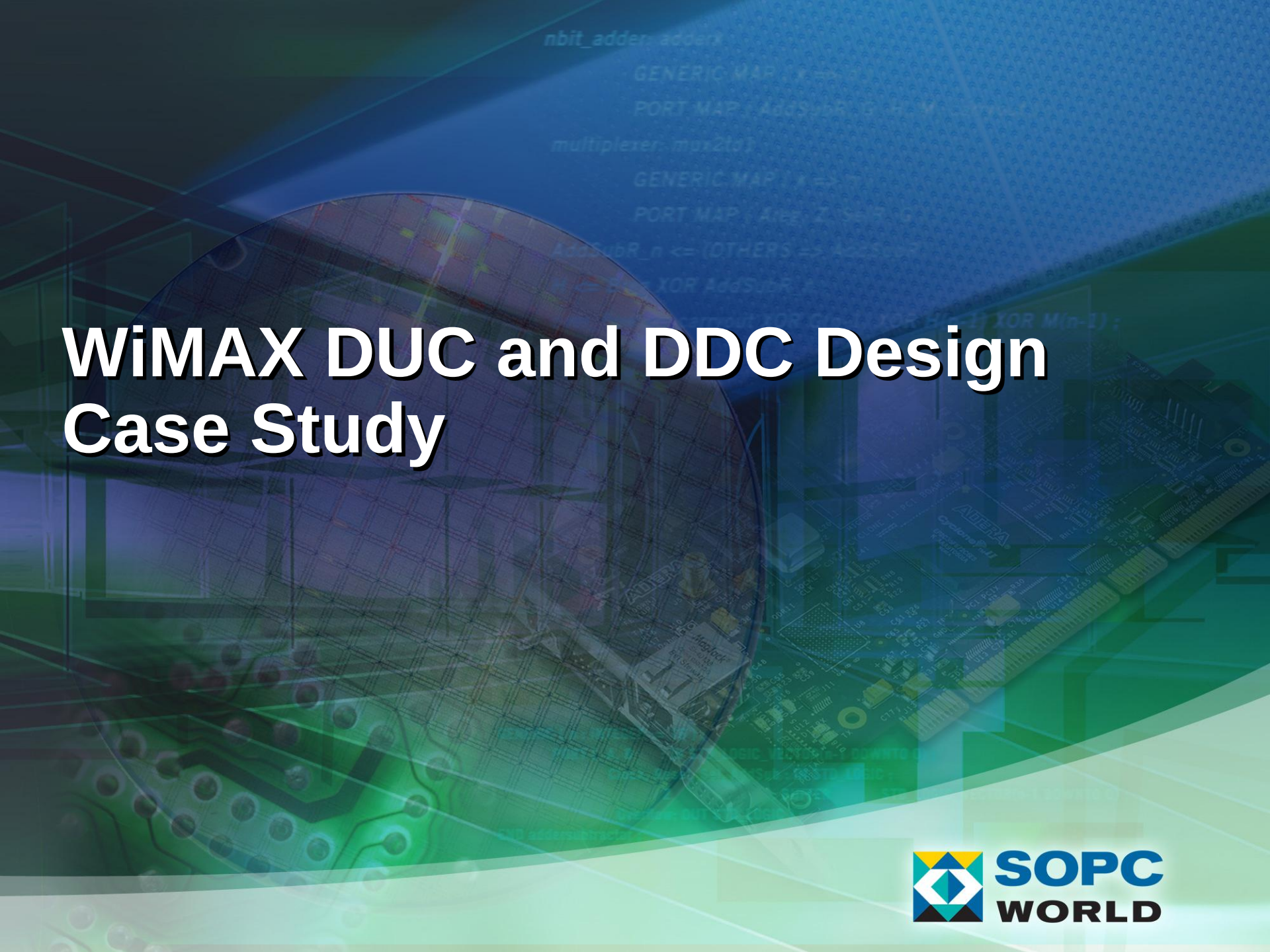
QUARTUS® II



System Reference Designs

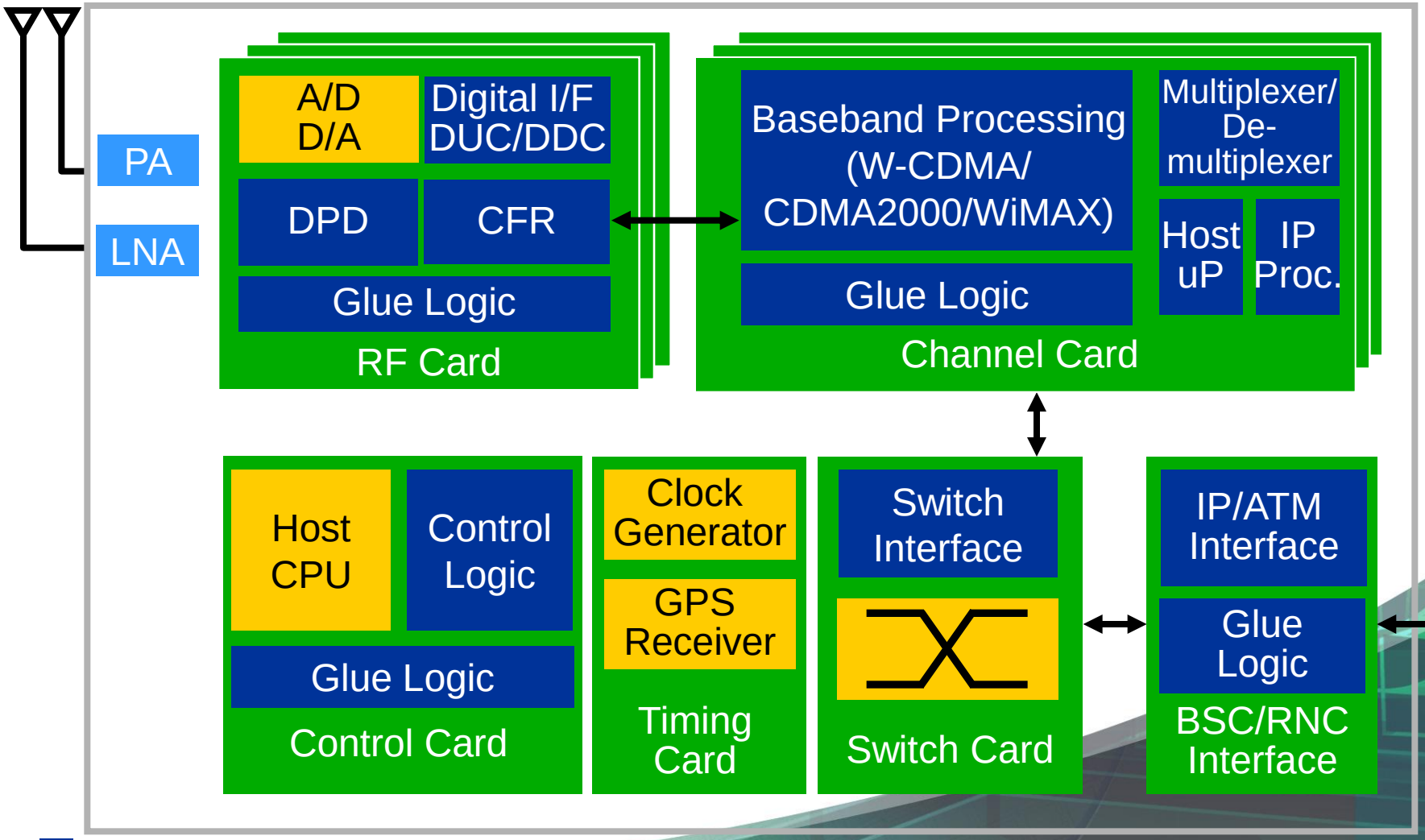
SOPC WORLD

WiMAX DUC and DDC Design Case Study



SOPC WORLD

Base Station Architecture Overview

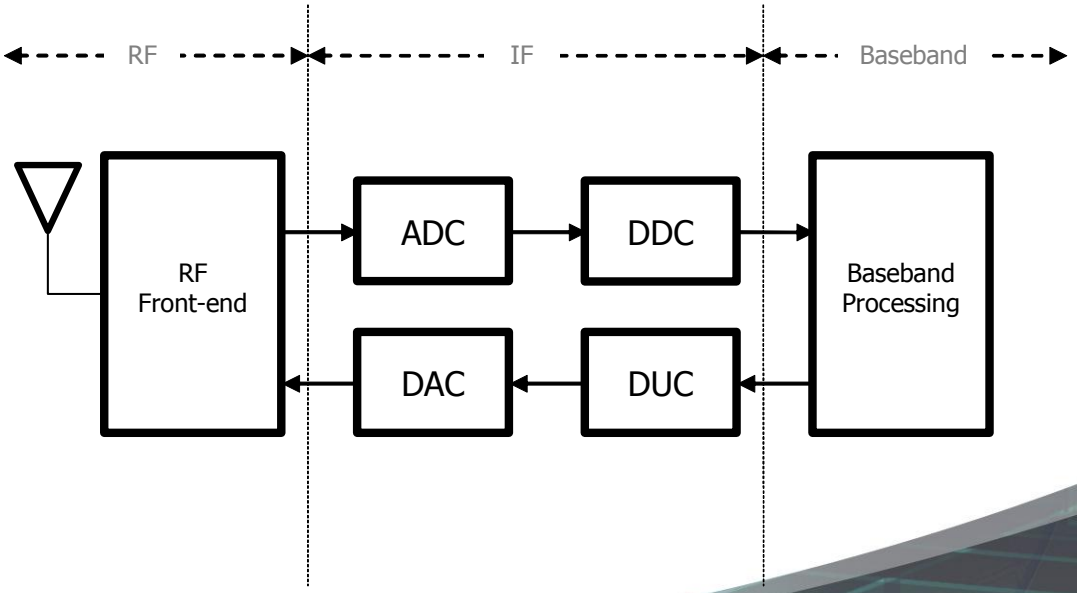


PLD Applications

SOPC WORLD

Reference Design Overview

- DUC/DDC
  - Provides the link between digital baseband and analog RF front end of generic transceiver
  - High throughput signal processing required makes FPGA ideal platform



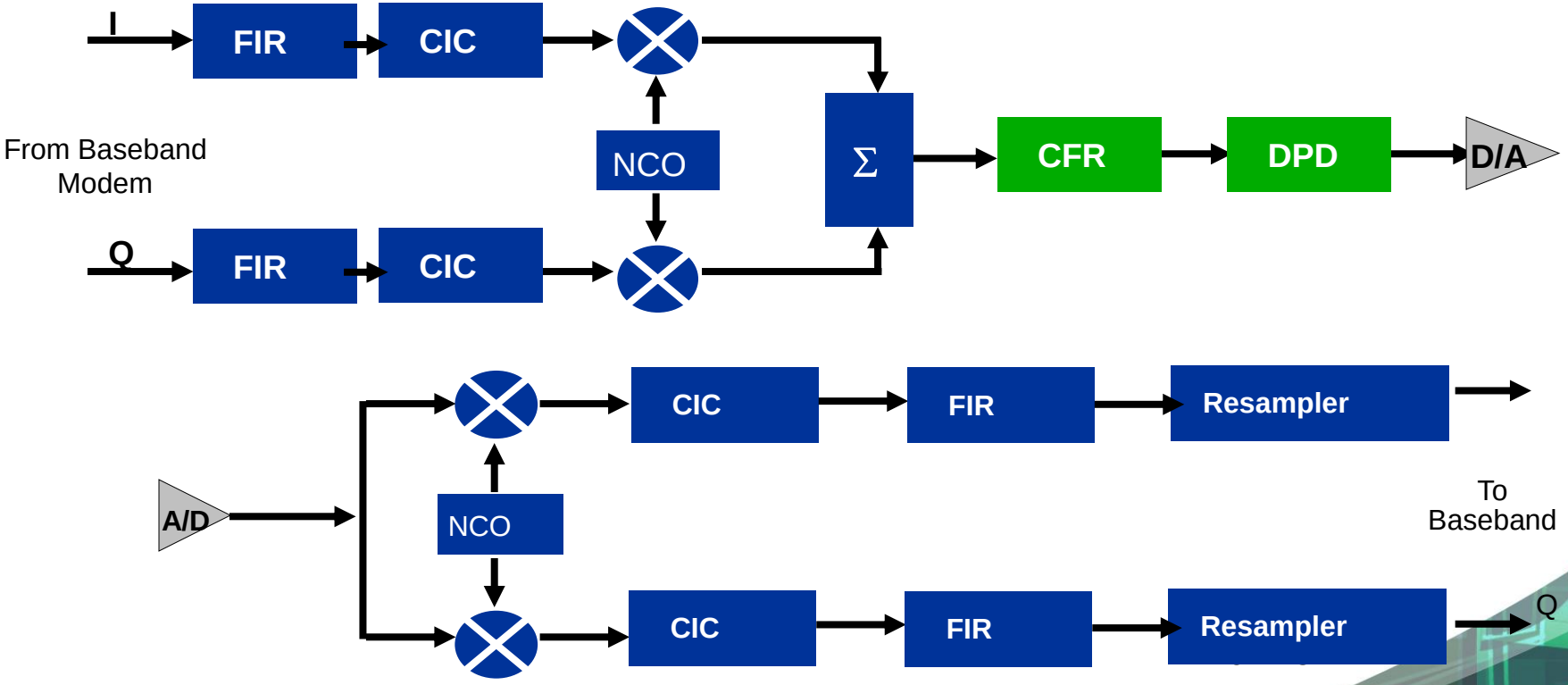
DSP Builder Ready

WiMAX DUC and DDC Designs

- Compliant to the draft WiMAX standard (IEEE 802.16)
- Multi-channel filter design for low cost
- Support for multiple transmit and receive antenna configurations
- Easily modifiable to support scalable channel bandwidths
- Uses DSP Builder methodology
- Backed up by DSP Builder-ready, highly parameterizable IP MegaCore functions

SOPC WORLD

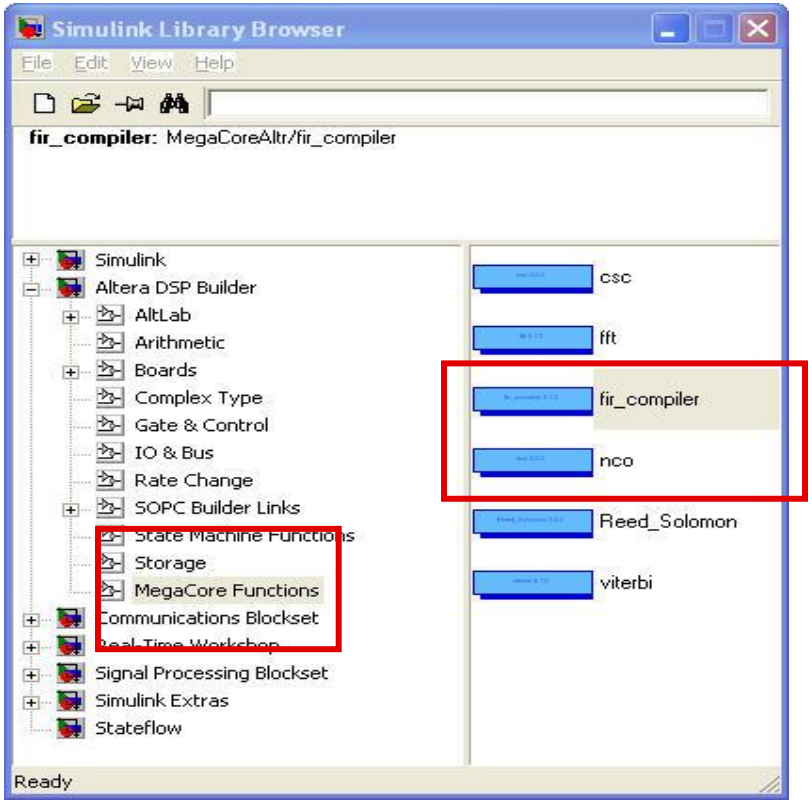
DUC and DDC High-Level Block Diagrams

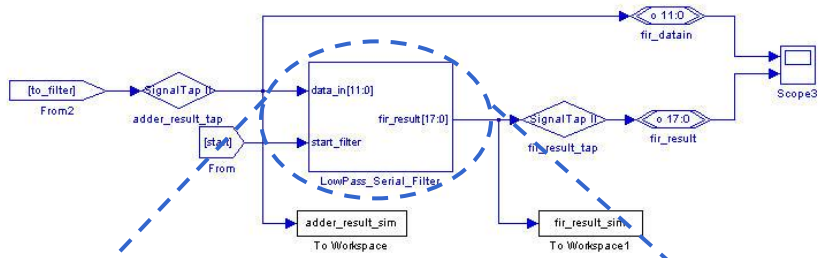


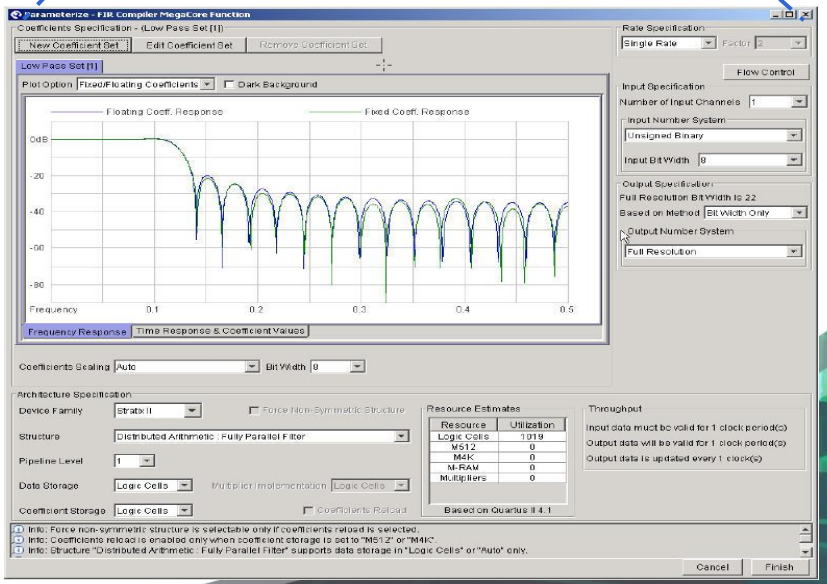
- DUC and DDC: Wireless, Military, Medical, Broadcast
- Crest-Factor Reduction (CFR) and Digital Predistortion (DPD): Wireless

SOPC WORLD

DSP Builder Implementation: IP MegaCore Library



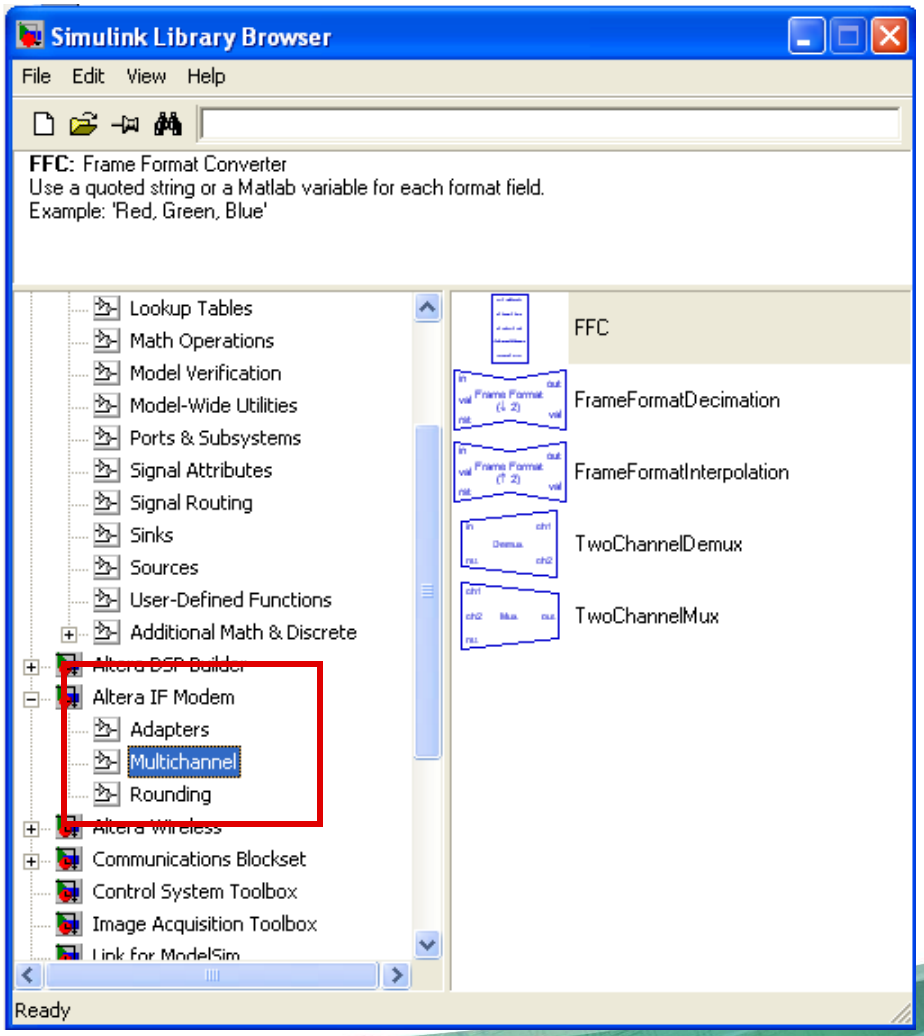




IP Can Be Added to the Library Separately

SOPC WORLD

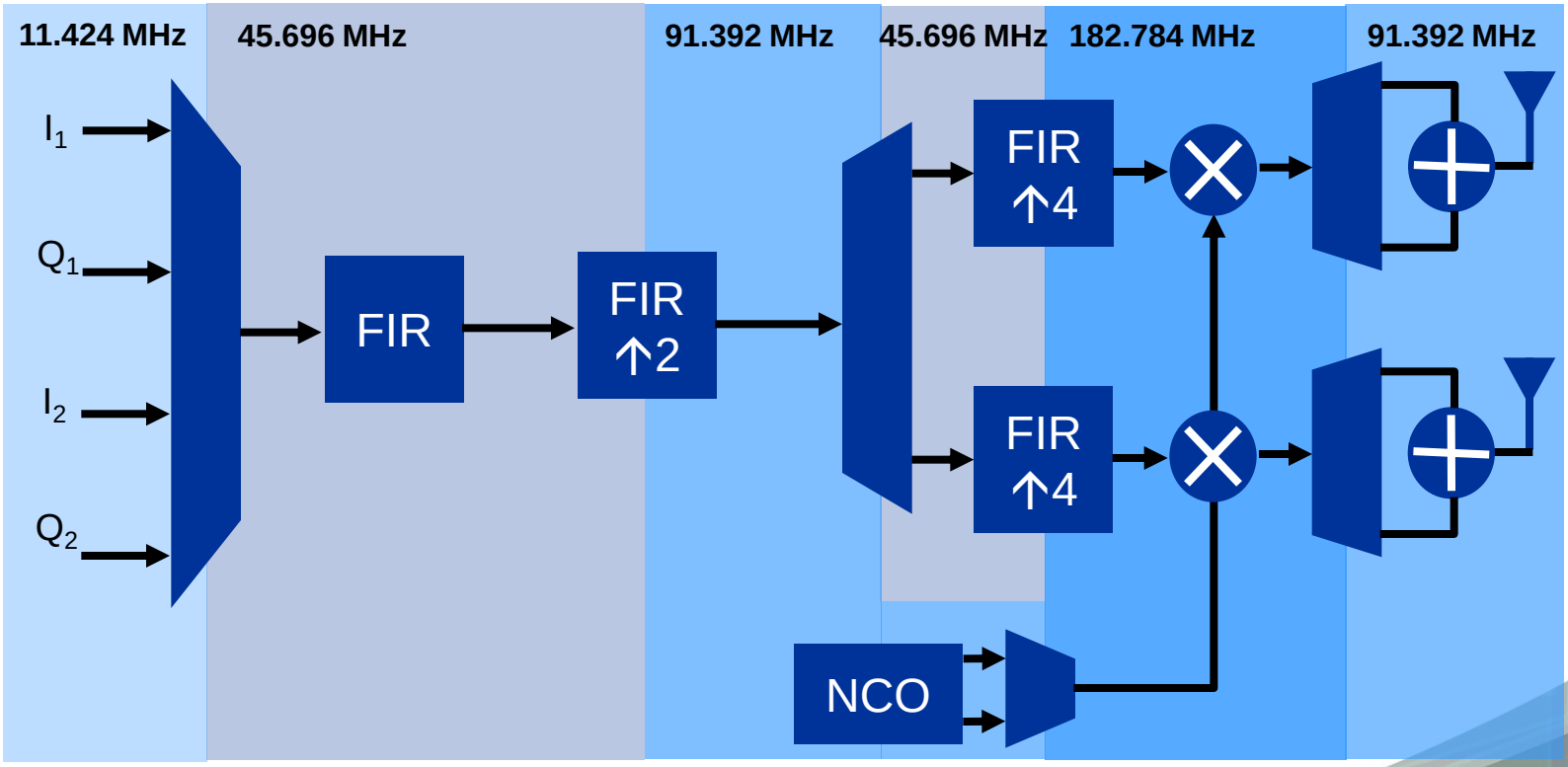
DSP Builder Implementation: Digital Intermediate Frequency (IF) Library



- Adapters
  - Provide input/output interface to finite impulse response (FIR) filter
- Multichannel
  - Frame format converter
  - Decimation
  - Interpolation
  - Multiplexer
  - Demultiplexer
- Rounding

SOPC WORLD

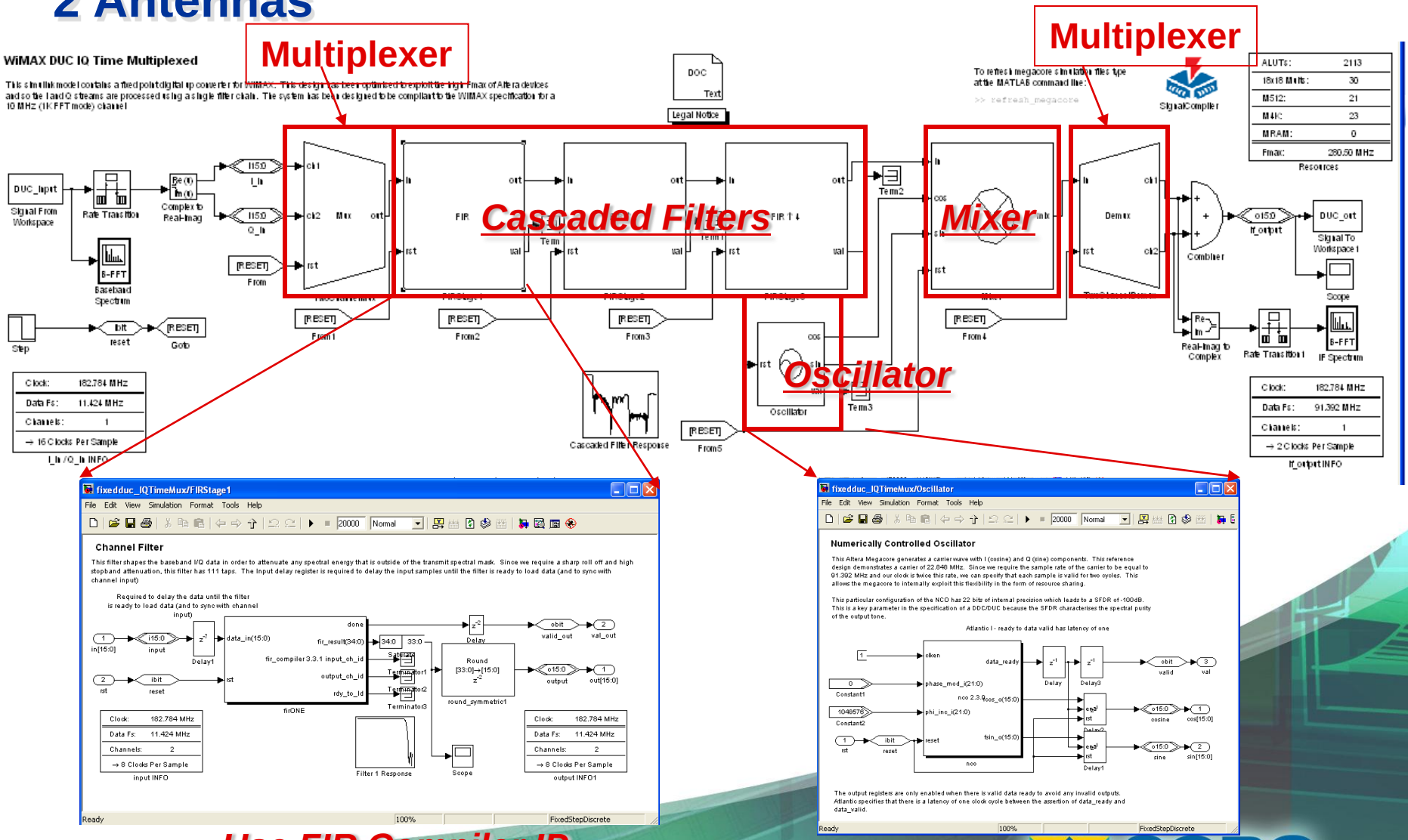
DUC With 2 Antennas Design Architecture



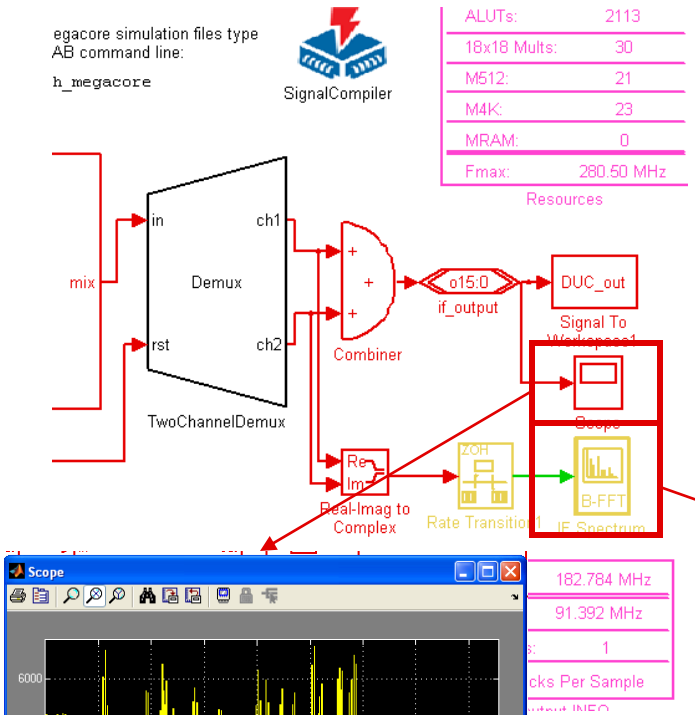
Timeshare DUC Hardware Between Antennas

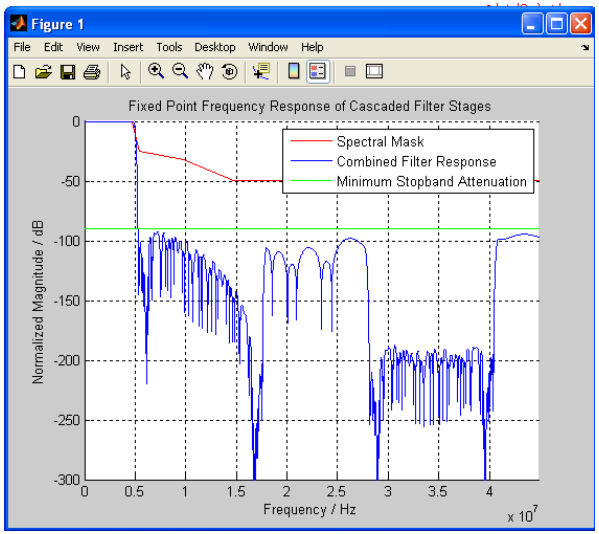
SOPC WORLD

DSP Builder Implementation: DUC Example Design With 2 Antennas

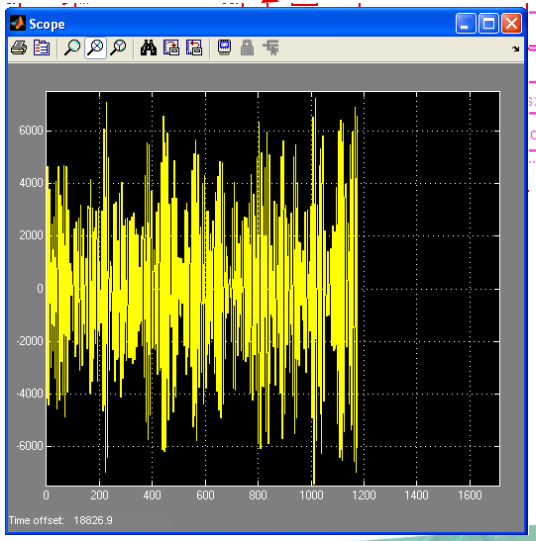


Simulation

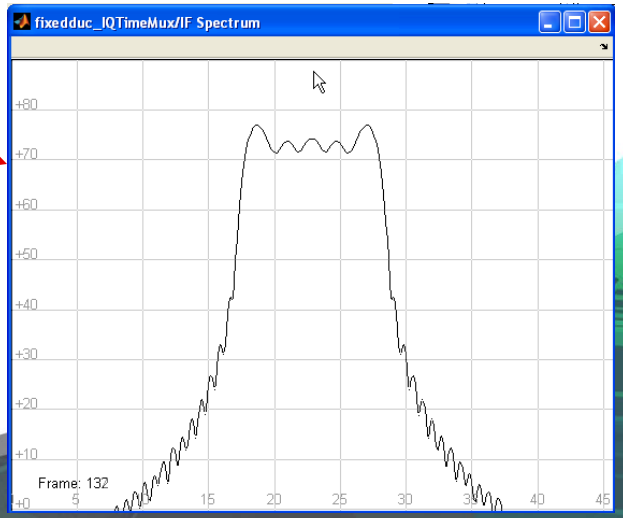




Cascaded Filter Output

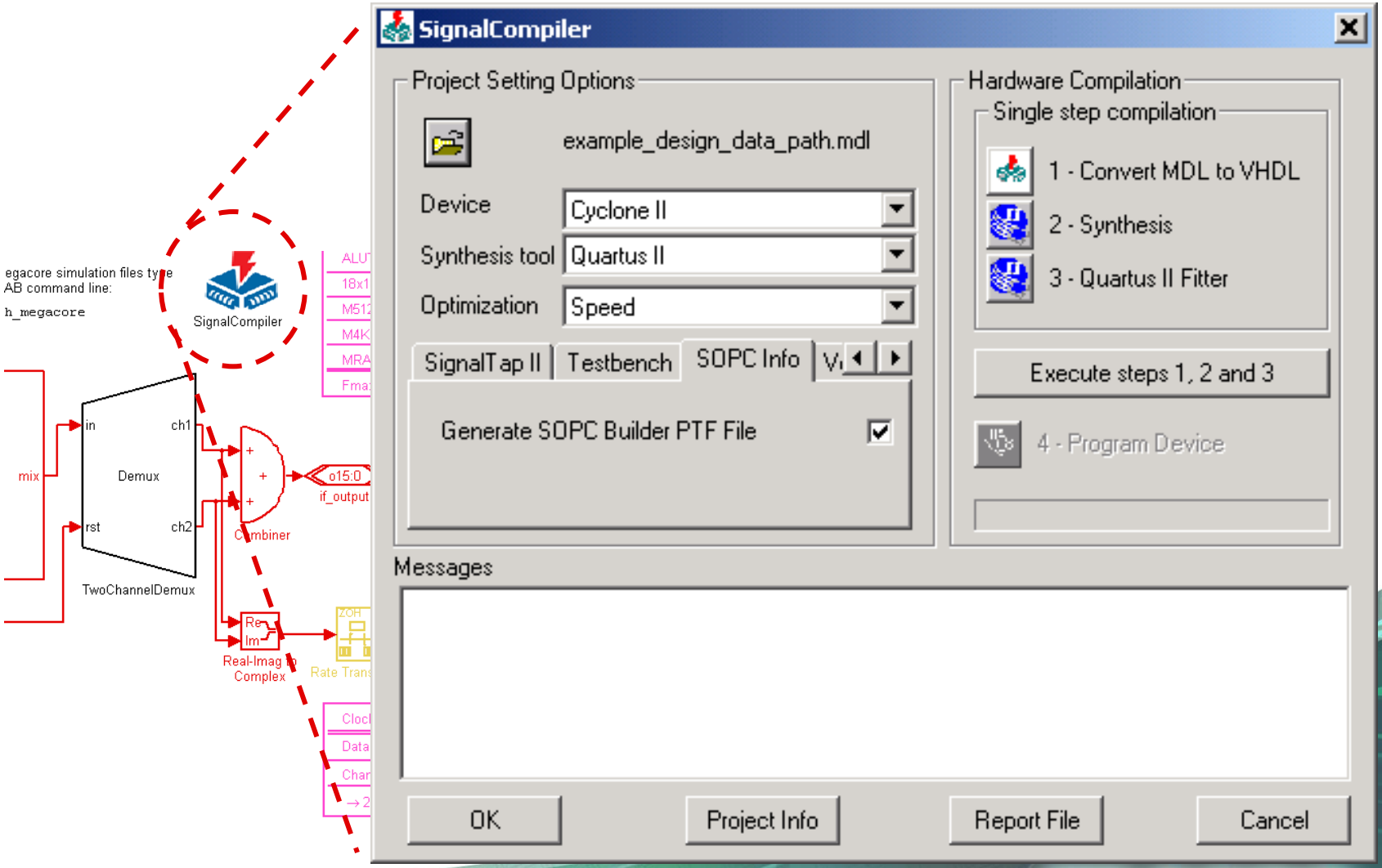


After Upconversion



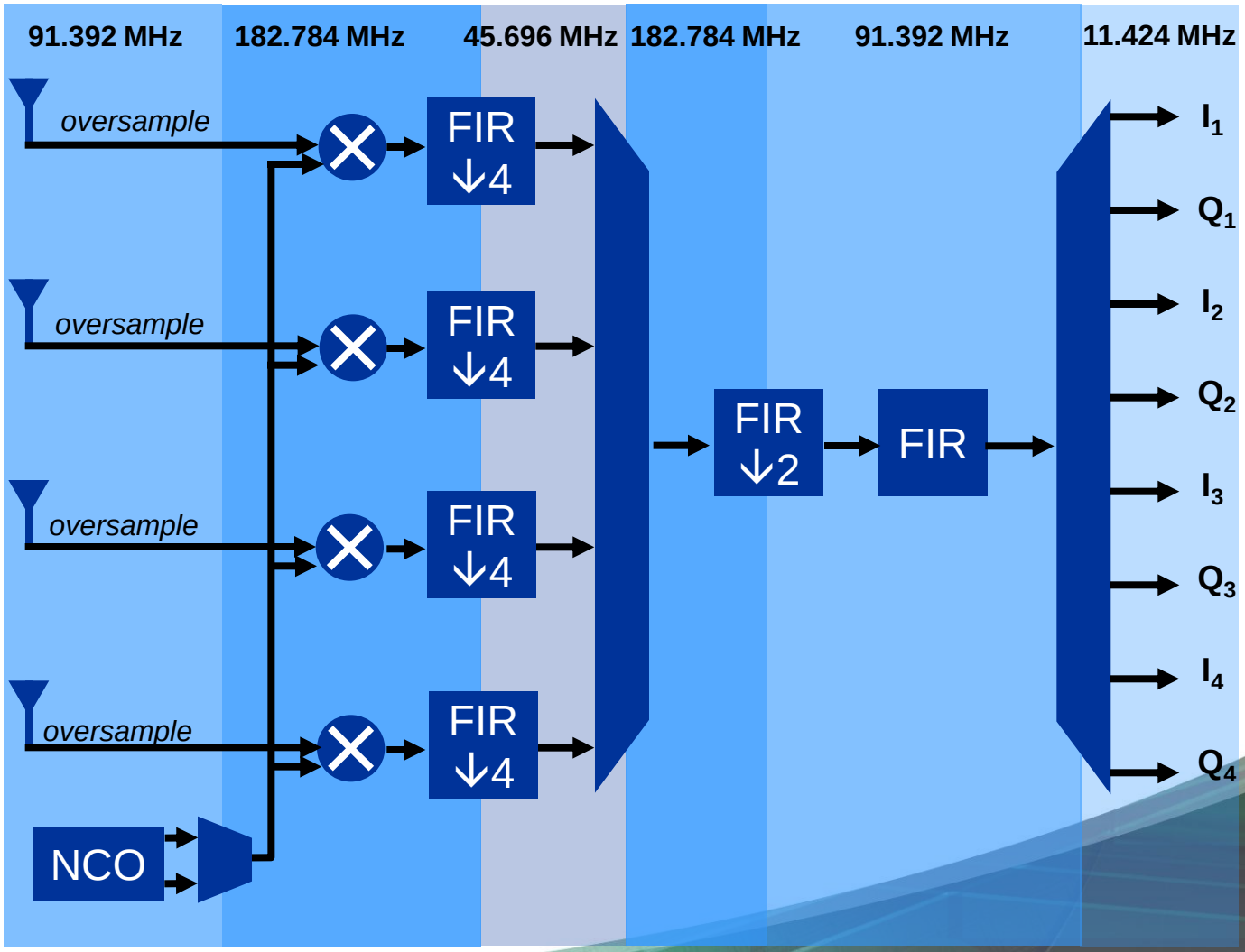
SOPC WORLD

Convert to VHDL: SignalCompiler



SOPC WORLD

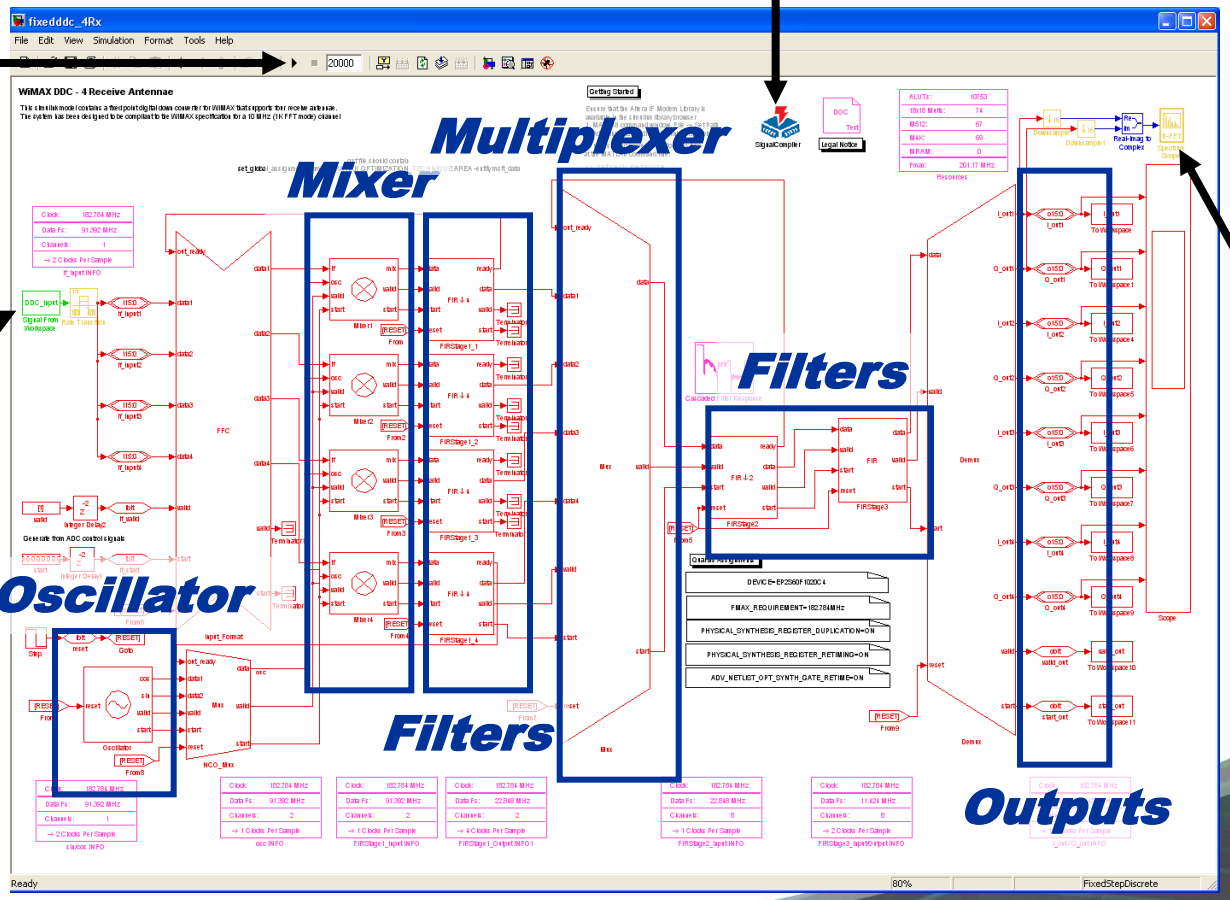
DDC With 4 Antennas Design Architecture



SOPC WORLD

DSP Builder Implementation: DDC Example Design With 4 Antennas

Simulate button



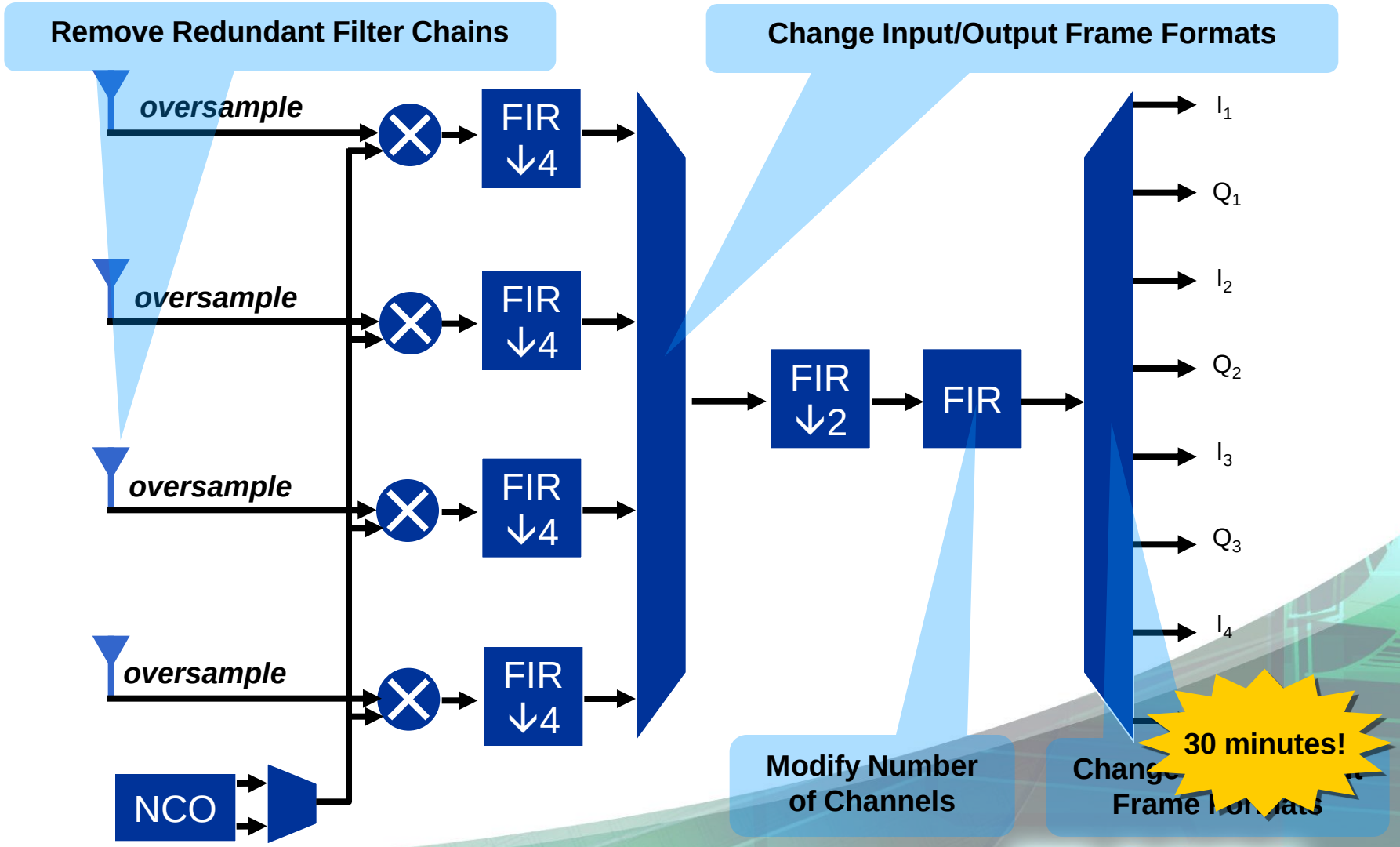
SignalCompiler (convert to HDL)

Analyze output data

Input stimulus loaded as part of model Initialization function

SOPC WORLD

2xRx DDC Architecture



SOPC WORLD

DUC and DDC Synthesis Results

ALUTs	M512	M4K	MRAM	Multipliers 18x18	f_{max} MHz
DUC Time Multiplexed IQ Design					
2,113	21	23	0	30	281
DUC 2 Antenna Design					
4,229	21	56	0	55	193
DDC Time Multiplexed IQ Design					
2,488	19	22	0	25	293
DDC 4 Antenna Design					
10,753	67	69	0	74	201

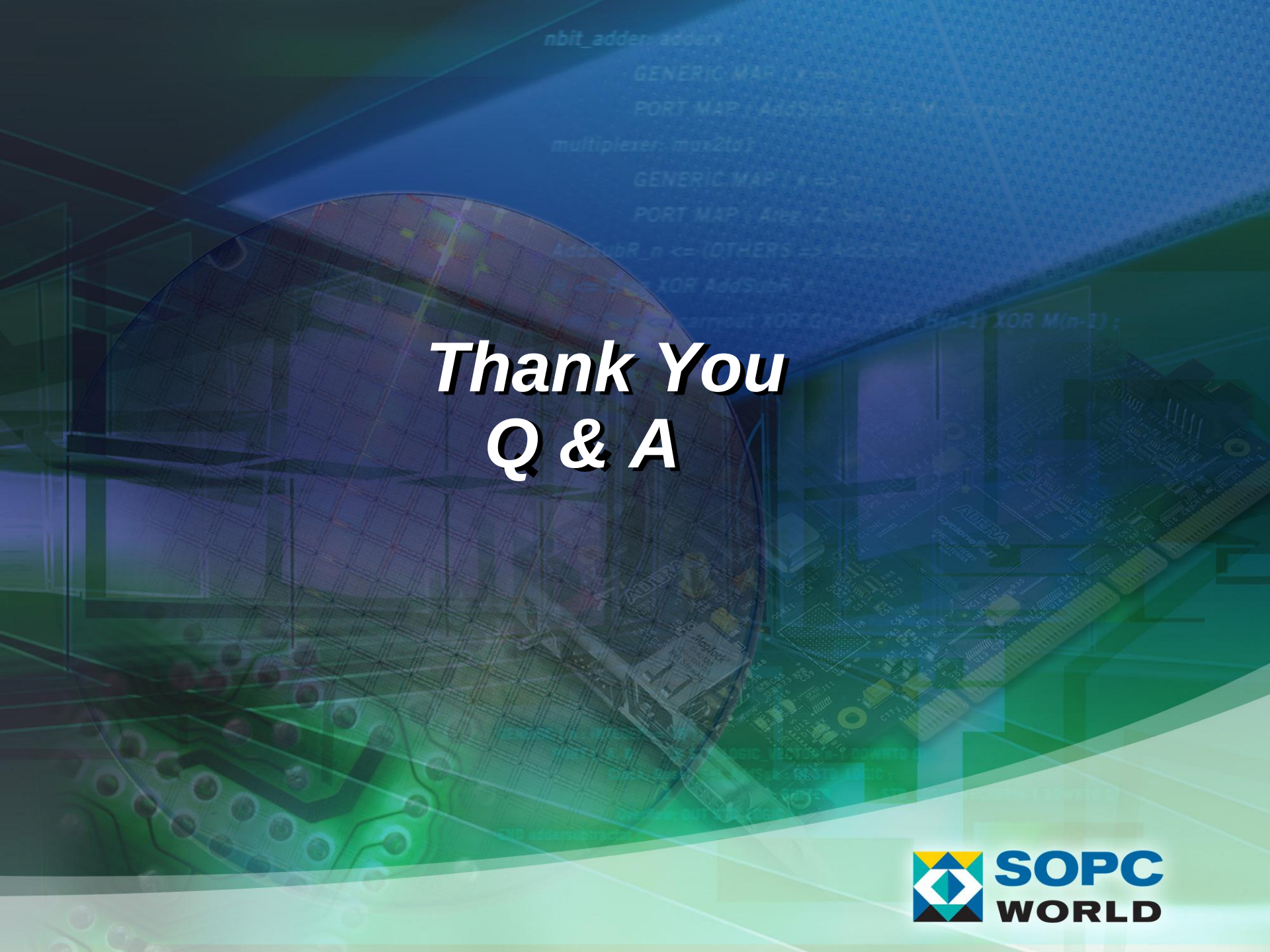
Highly Optimized and Cost Efficient Designs!
More Information at www.altera.com

SOPC WORLD

Summary

- DSP Builder tool improves productivity
  - System-level DSP design and FPGA design integrated into one platform: Simulink
- WiMAX DUC and DDC application example
  - DSP Builder-based IQ time multiplexed and multi-antenna designs
    - Use FIR compiler and NCO compiler IP
  - Design methodology significantly reduces the development time for different standards
  - Highly optimized and cost-efficient designs

SOPC WORLD


Thank You
Q & A
SOPC
WORLD