

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

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

FPGA in DSP Applications



Automotive



Communications



Consumer



Industrial



**Test and
Measurement**



Military

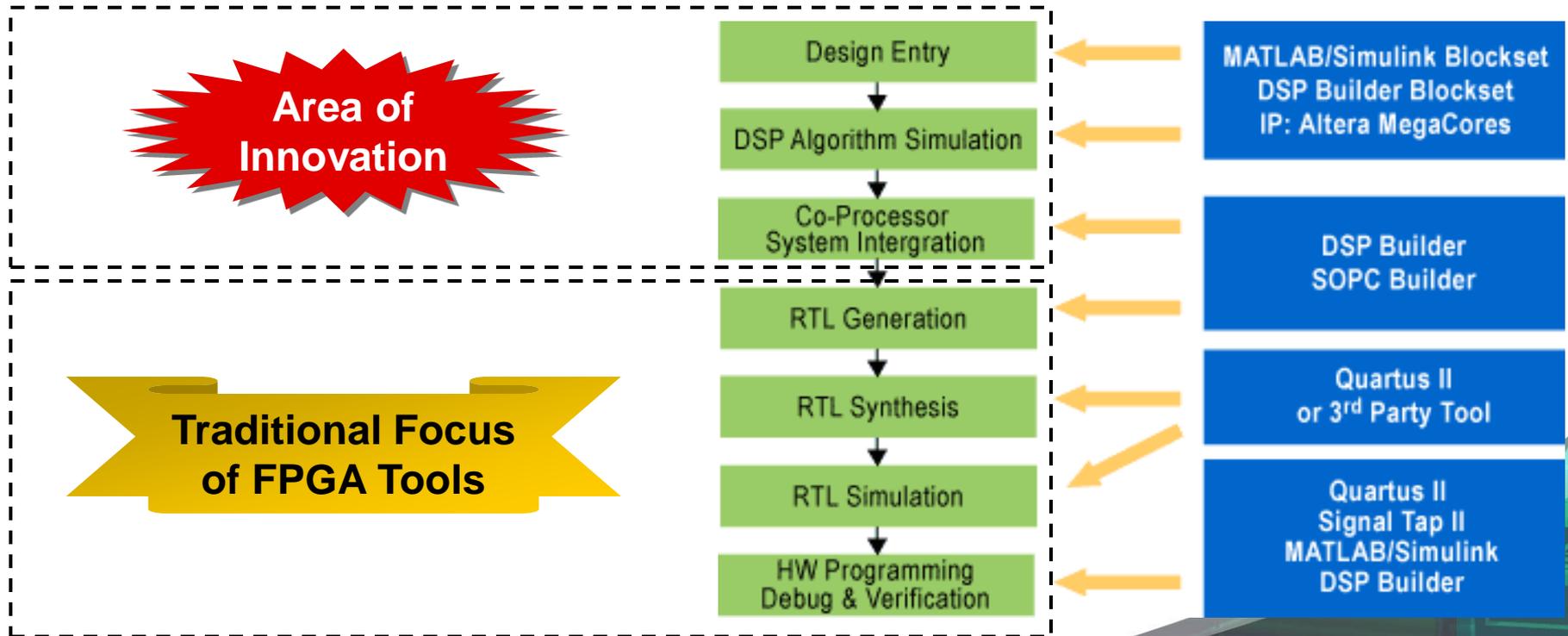


Broadcast

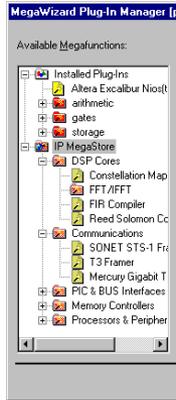


Medical

DSP Design Flow in FPGA



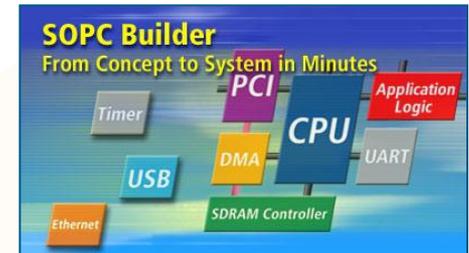
Altera System-Level Design Tools



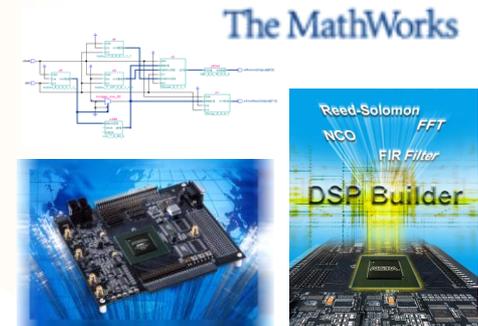
IP Integration



Software Development



System Integration

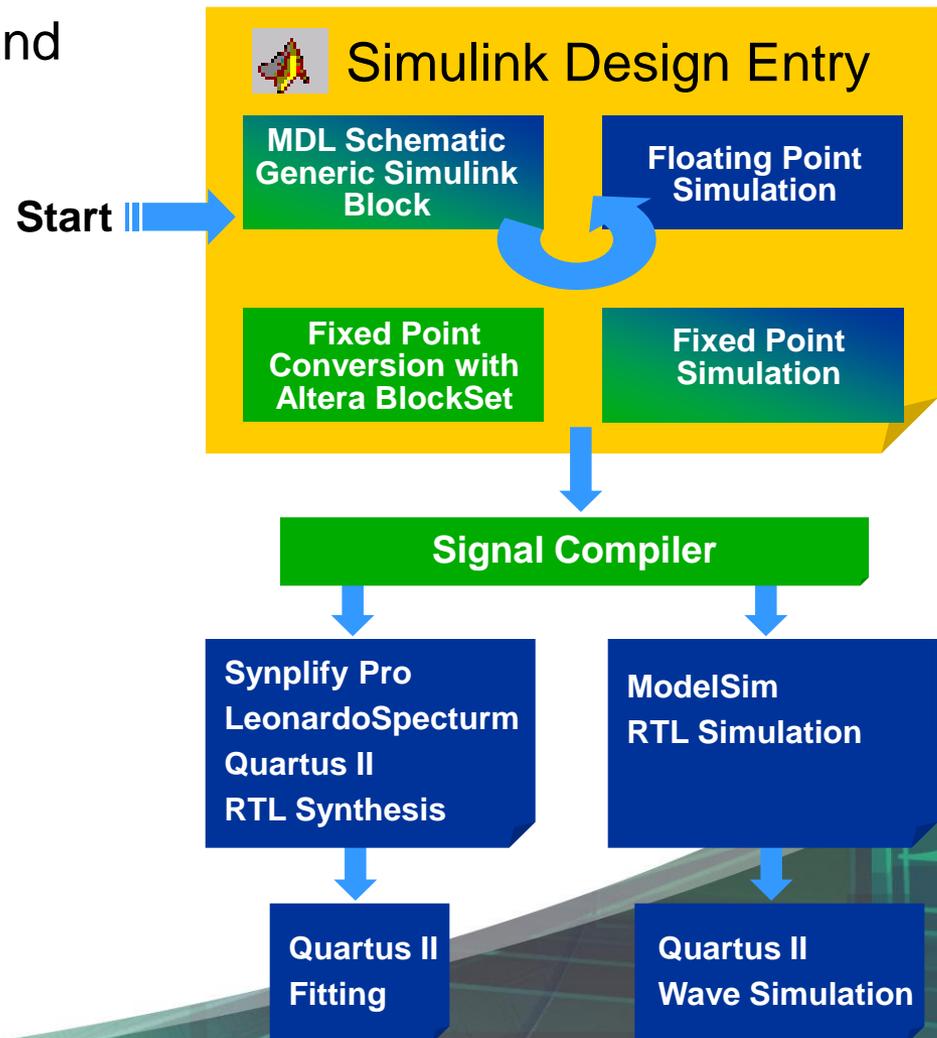


DSP Algorithm Development

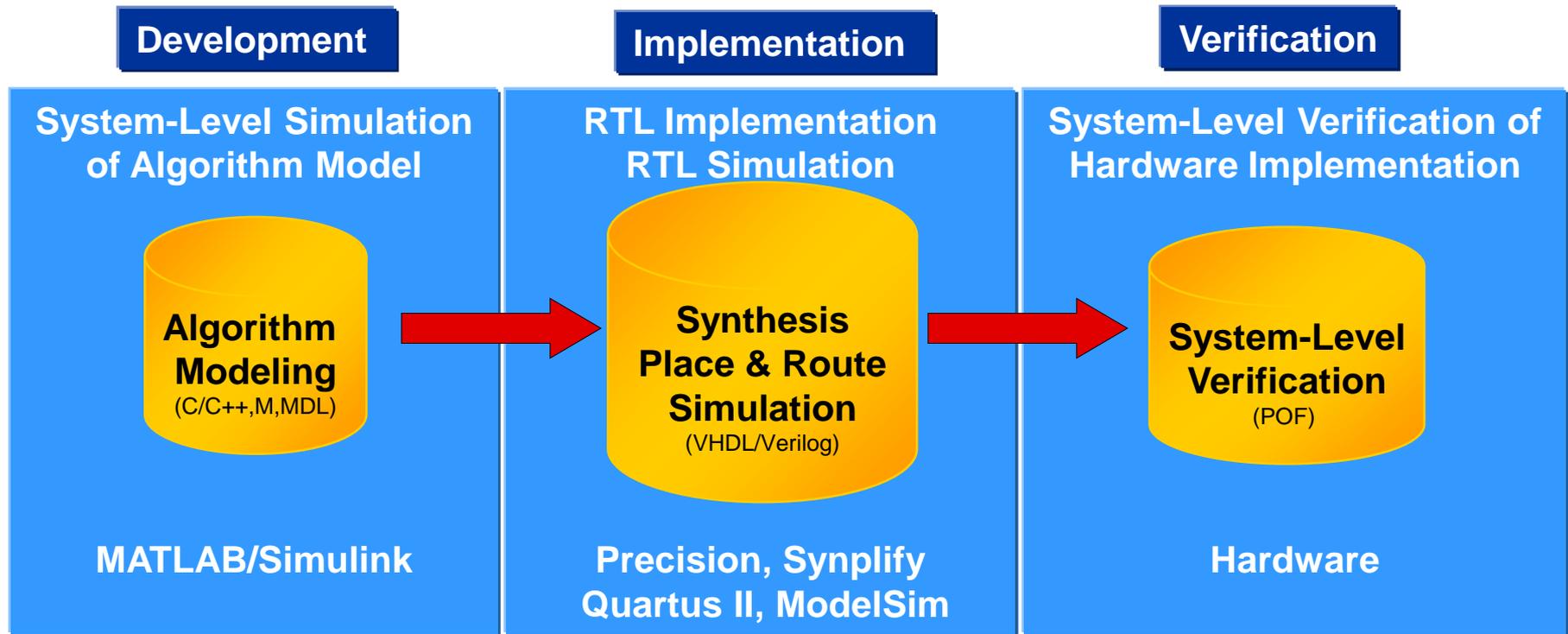
What is DSP Builder?

 DSP Builder

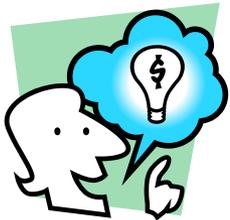
- Interface between Quartus II and MATLAB/Simulink
- Library add-on to Simulink
- Altera blockset
 - Library of fixed-point Simulink functions
 - Uses double precision
- Altera DSP IP
 - OpenCore Plus
- SignalCompiler utility
 - Converts between Simulink and Altera domain
- Hardware debug
 - Hardware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



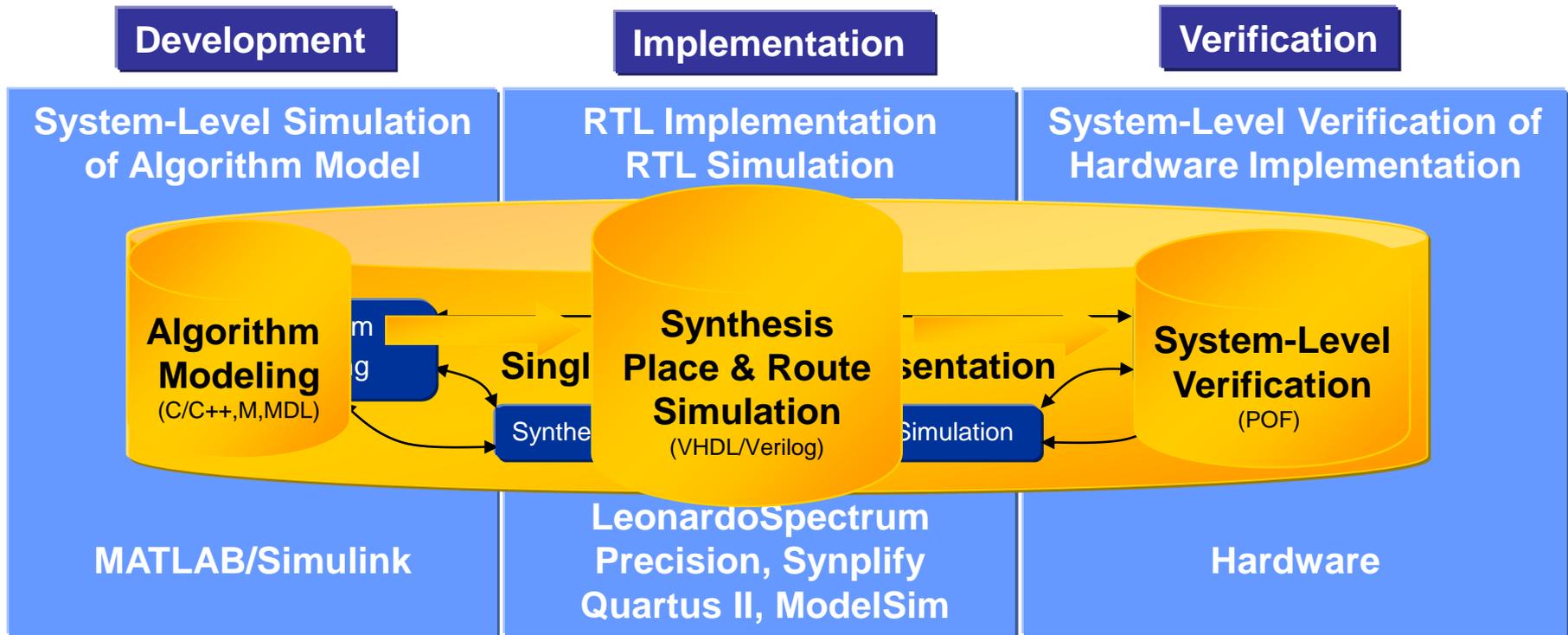
Traditional System Design Tool Flow



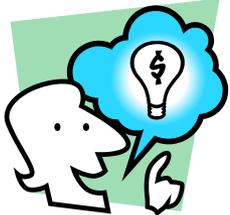
System Algorithm Design and FPGA Design Separated



DSP Builder - Simulink Design Flow



System Algorithm Design and FPGA Design Integrated



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

DSP Builder Benefits

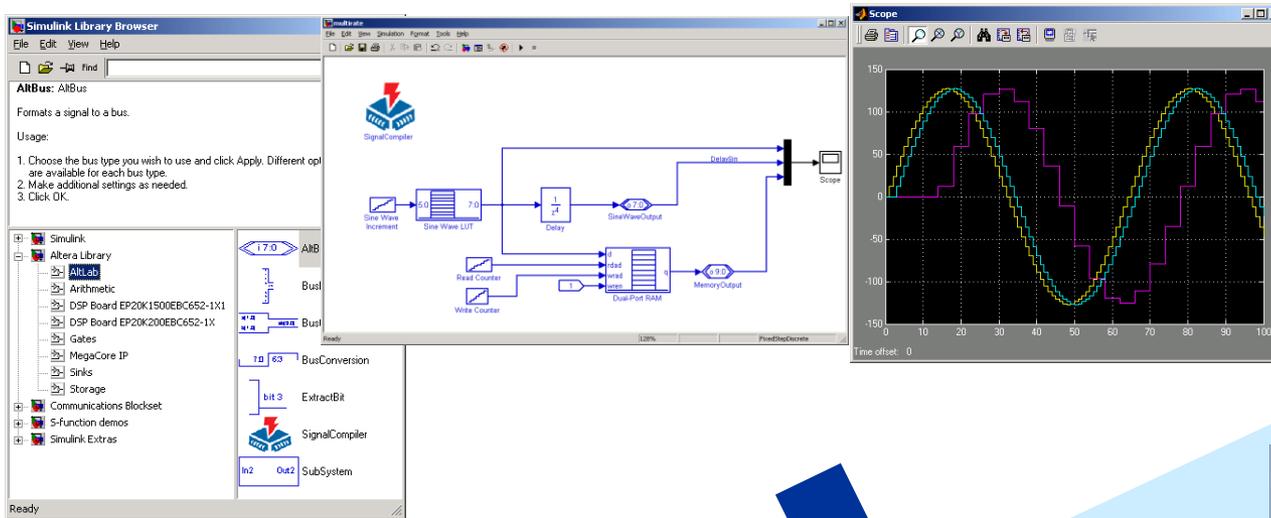
■ For hardware engineer

- 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

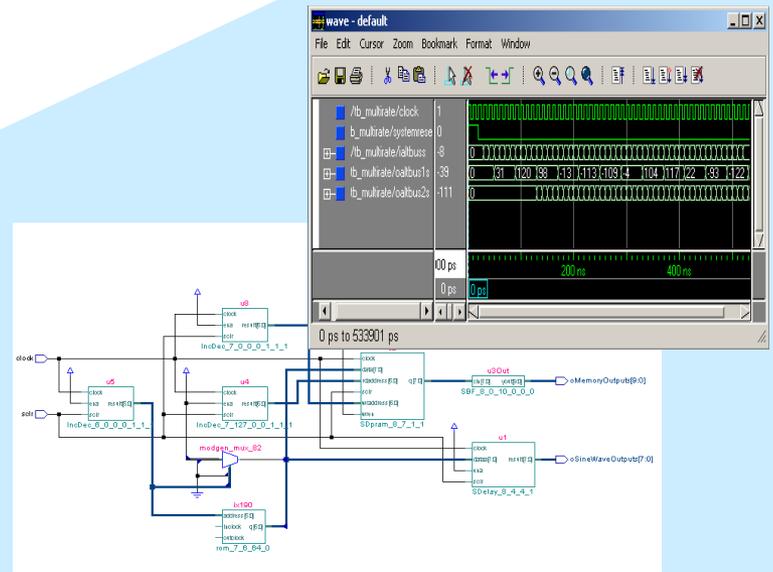
DSP Builder Design Flow



Matlab/Simulink Domain
(C+ System Analysis)



VHDL Domain
(Implementation/Simulation)

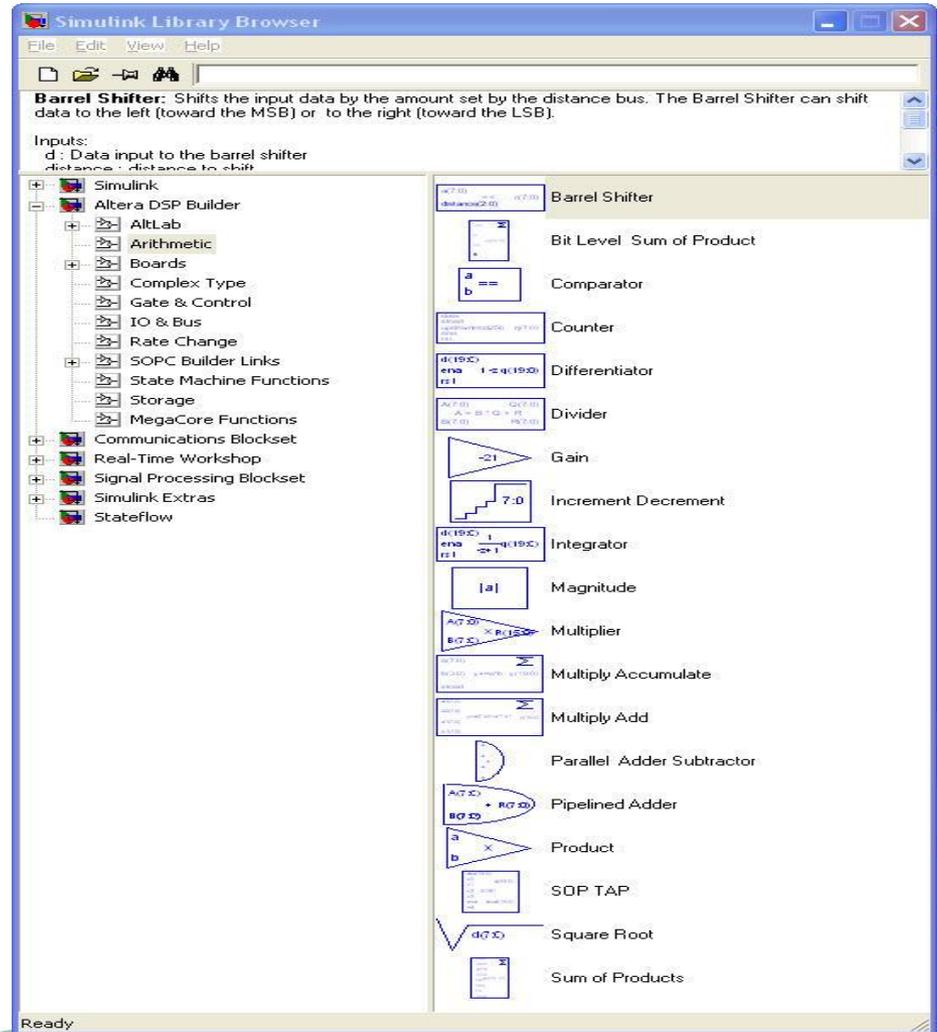


Design Flow Overview

- 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

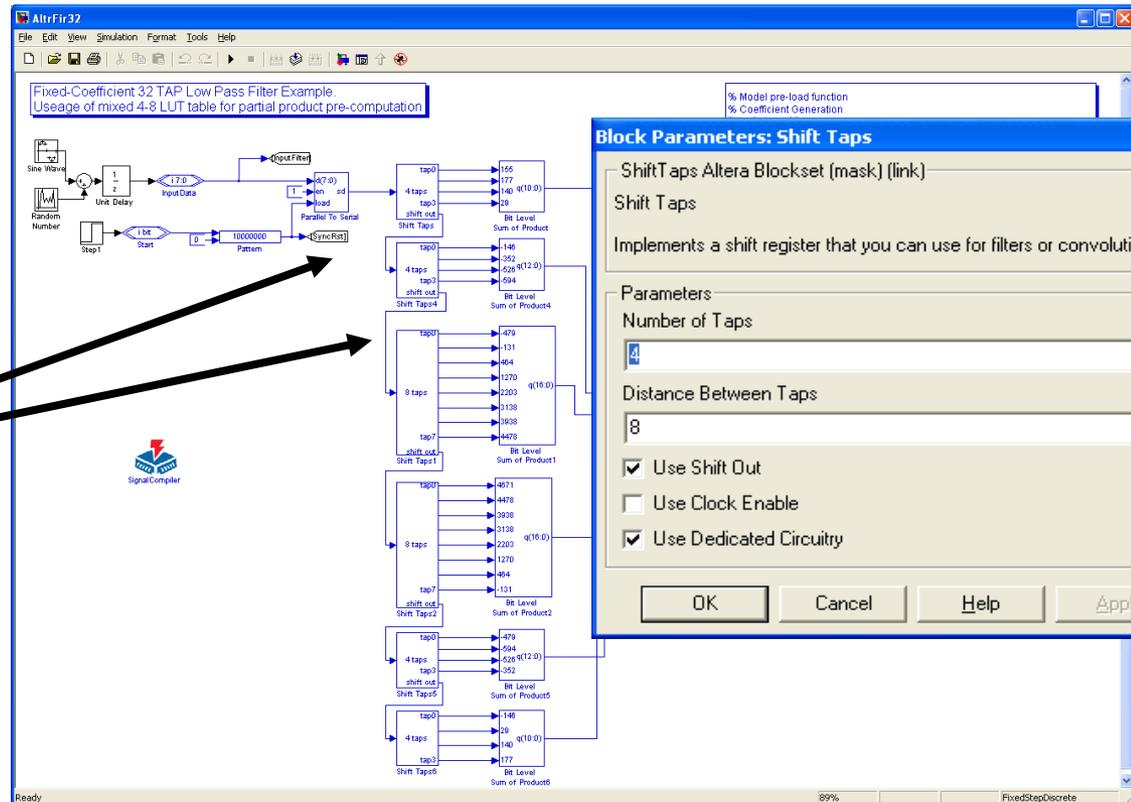
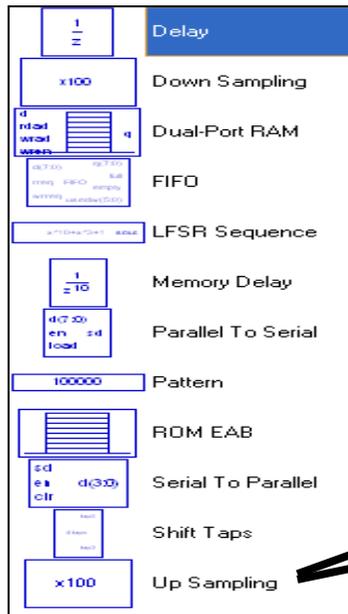
Altera DSP Builder Libraries

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

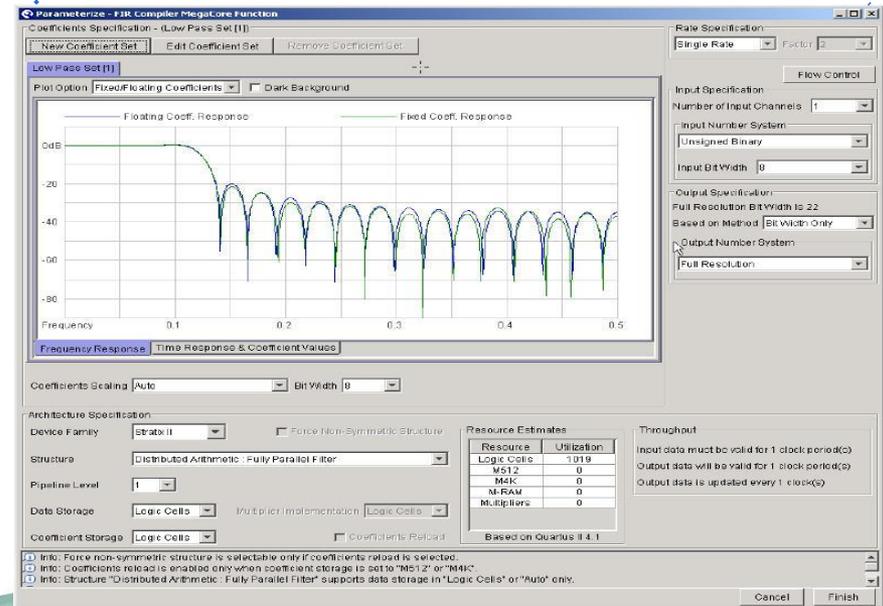
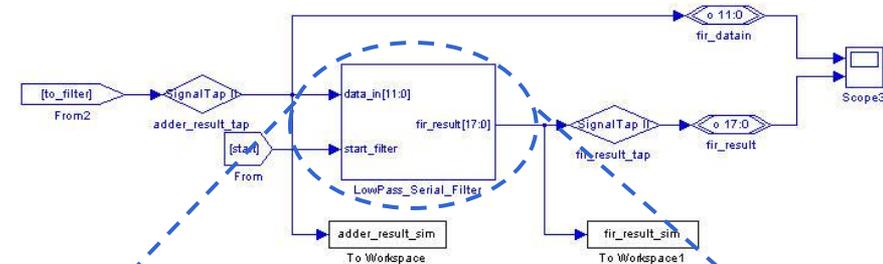
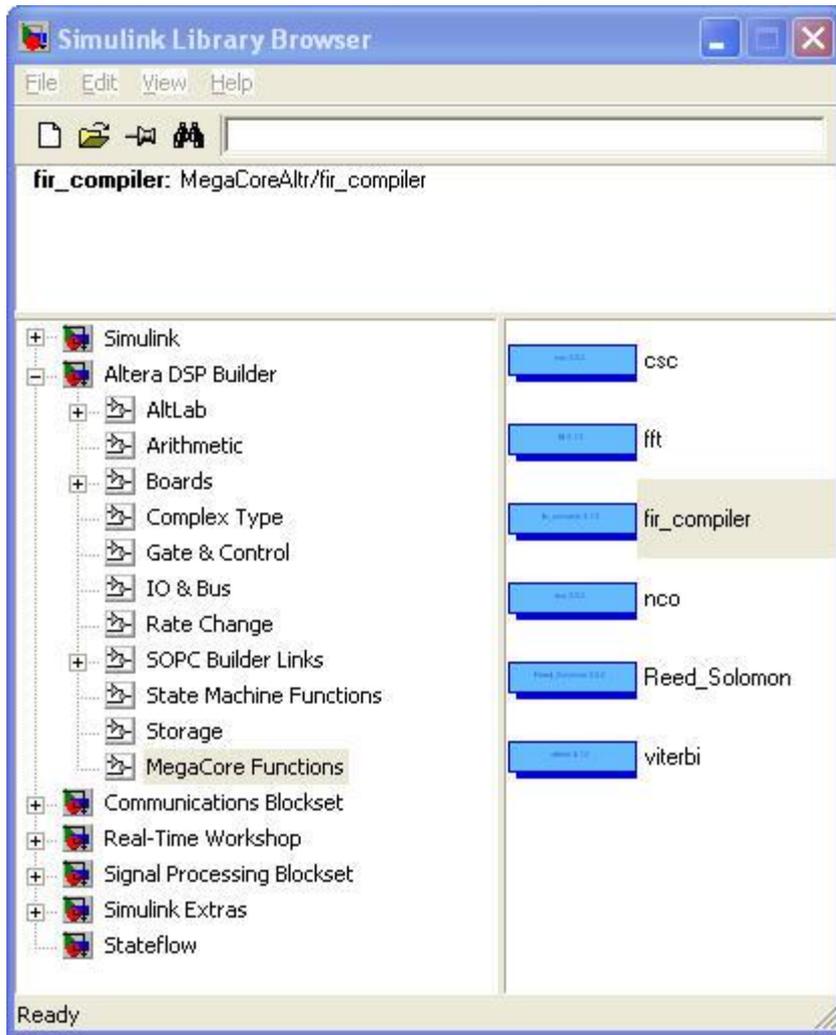


Step 1: Create Design in Simulink Using Altera Libraries

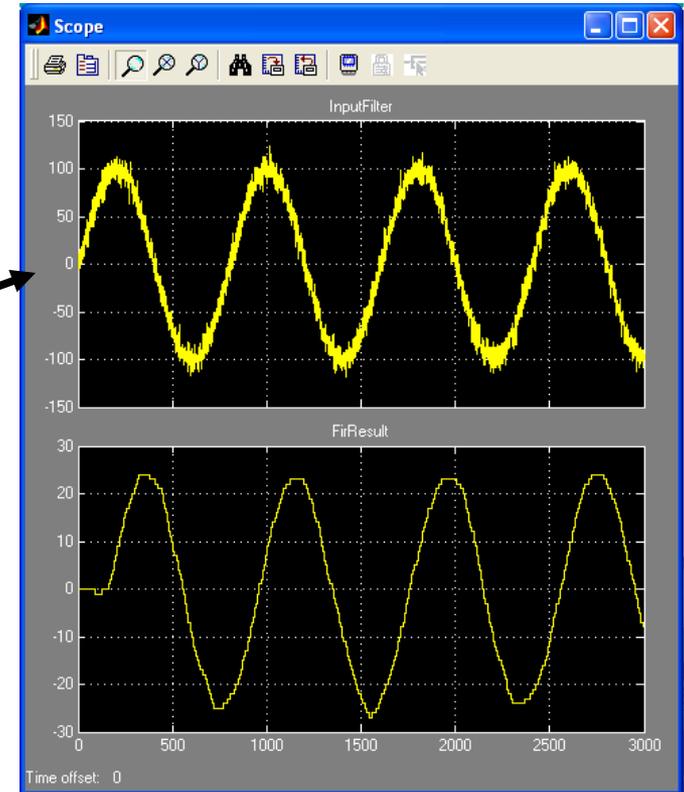
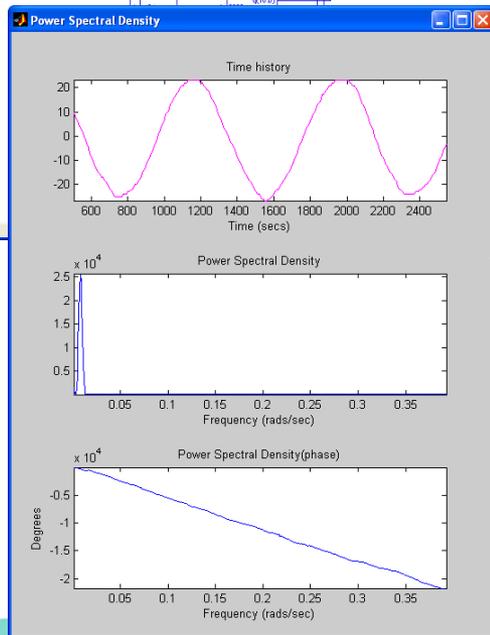
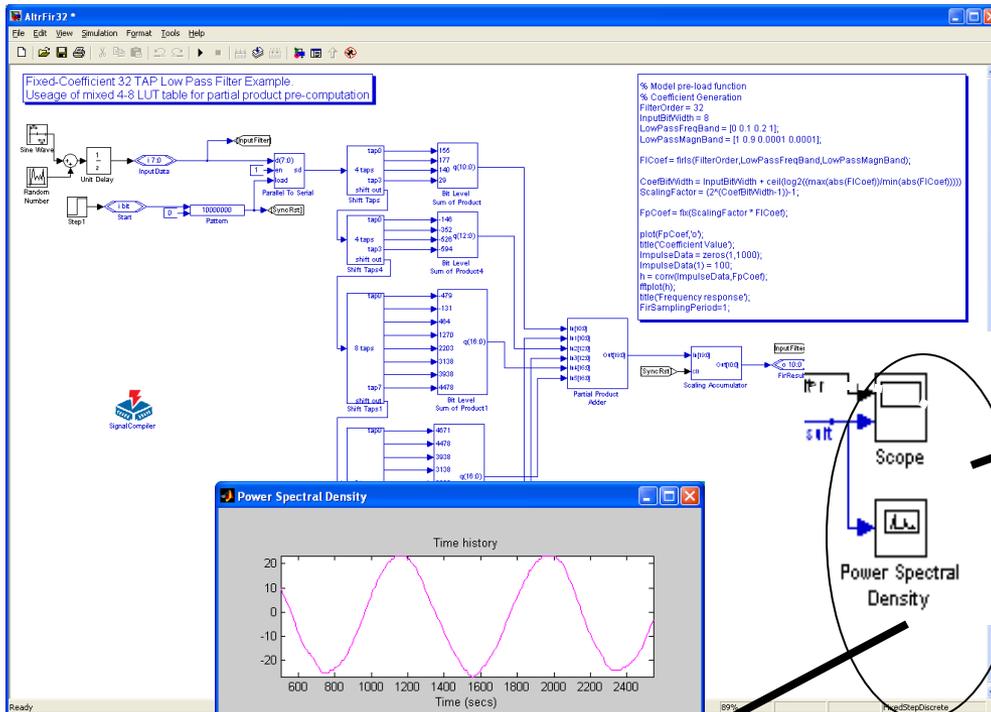
- Drag and drop library blocks into Simulink design and parameterize each block



Parameterization of IP MegaCore Functions



Step 2: Simulate in Simulink



Step 3: Add SignalCompiler to Model to Generate HDL Code

Project Setting Options

FilteringLab.mdl

Device: Stratix II

Synthesis tool: Quartus II

Optimization: Speed

Main Clock: Reset | SignalTap II | Testb |

Period: 10 ns

Hardware Compilation

Single step compilation

- 1 - Convert MDL to VHDL
- 2 - Synthesis
- 3 - Quartus II Filter
- 4 - Program Device

Execute steps 1, 2 and 3

Messages

```
> Device Family Setting has been changed to Stratix II
> Re-Run Step 1 : Convert MDL to VHDL
```

OK | Project Info | Report File | Cancel

- 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

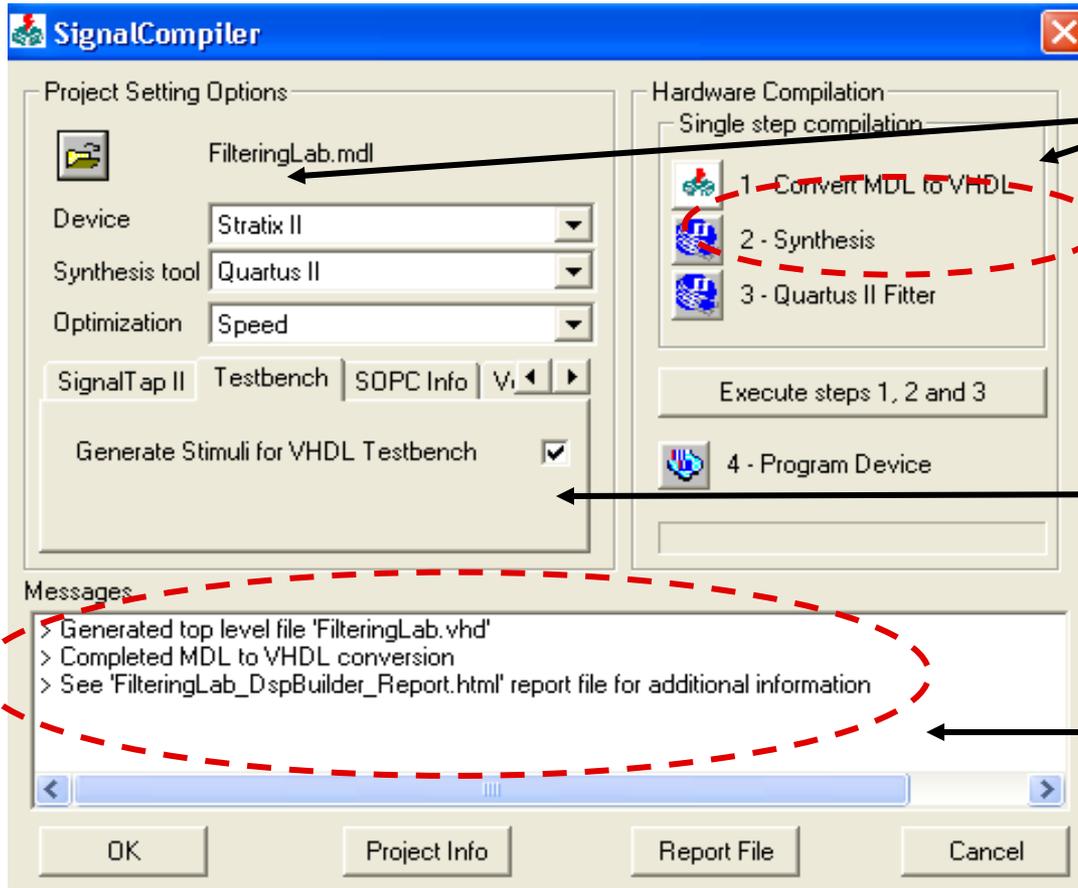
- LeonardoSpectrum™
- Synplify
- Precision
- Quartus II

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

Testbench generation

Message window

Step 4: Create HDL Code and Generate Testbench



FilteringLab.mdl

Enable "Generate Stimuli for VHDL Testbench" Button

FilteringLab.vhd

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_datain0 : 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_datain0;

-- 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;

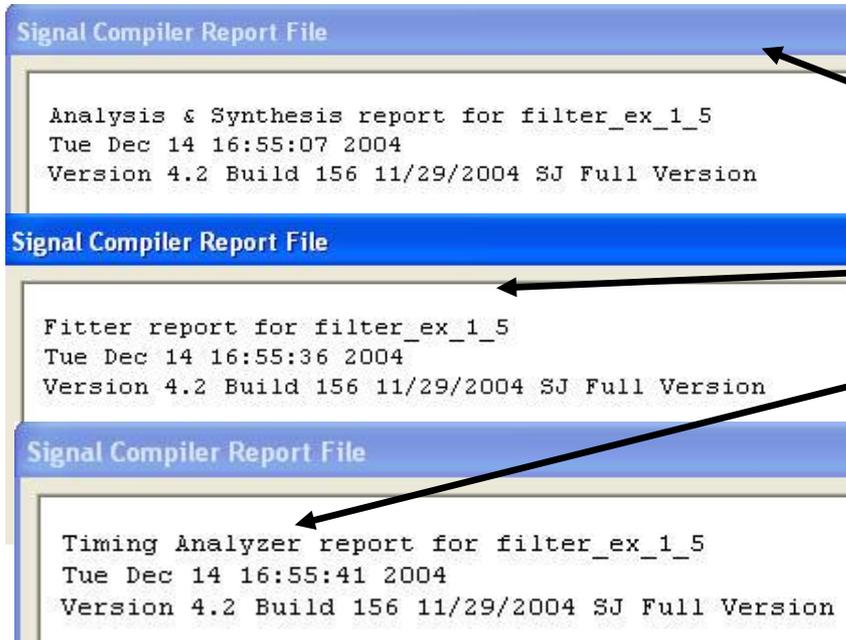
-- Simulink Block "VCC"
A15W <= '1';

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

-- Simulink Block "GND1"
```

DSP Builder Report File

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



The large window titled "Signal Compiler Report File" displays a detailed report for "filter_ex_1_5.mdl". The report is organized into several sections:

DSP Builder Report File for filter_ex_1_5.mdl

Project Setting

- Device Family: Stratix II
- Synthesis Tool: Quartus II
- Optimization: Speed
- Date: Tuesday, December 14, 2004
- Time: 16:54:55
- Version: 3.0.0 b23

Compilation

- Convert Mdl to VHDL : PASSED
- Synthesis : PASSED [filter_ex_1_5.map.rpt](#)
- Quartus II Fitter : PASSED [filter_ex_1_5.fit.rpt](#)
- Timing Analyzer report : [filter_ex_1_5.fit.tan](#)

Resource Usage Summary

Resource	Usage
Total combinational functions	105
ALUT usage by number of inputs	
-- 7 input functions	0
...	...

Step 5: Perform RTL Simulation (ModelSim)

ModelSim ALTERA 5.6a - Custom Altera Version

File Edit View Compile Simulate Tools Window Help

1) Set working directory (**File => Change Directory**)

Execute Do File

Look in: altrdemos

- altera_mf
- db
- dspbuilder
- DSPBuilder_AltrFir32
- lpm
- work
- AltrFir32_quartus.tcl
- tb_AltrFir32.tcl

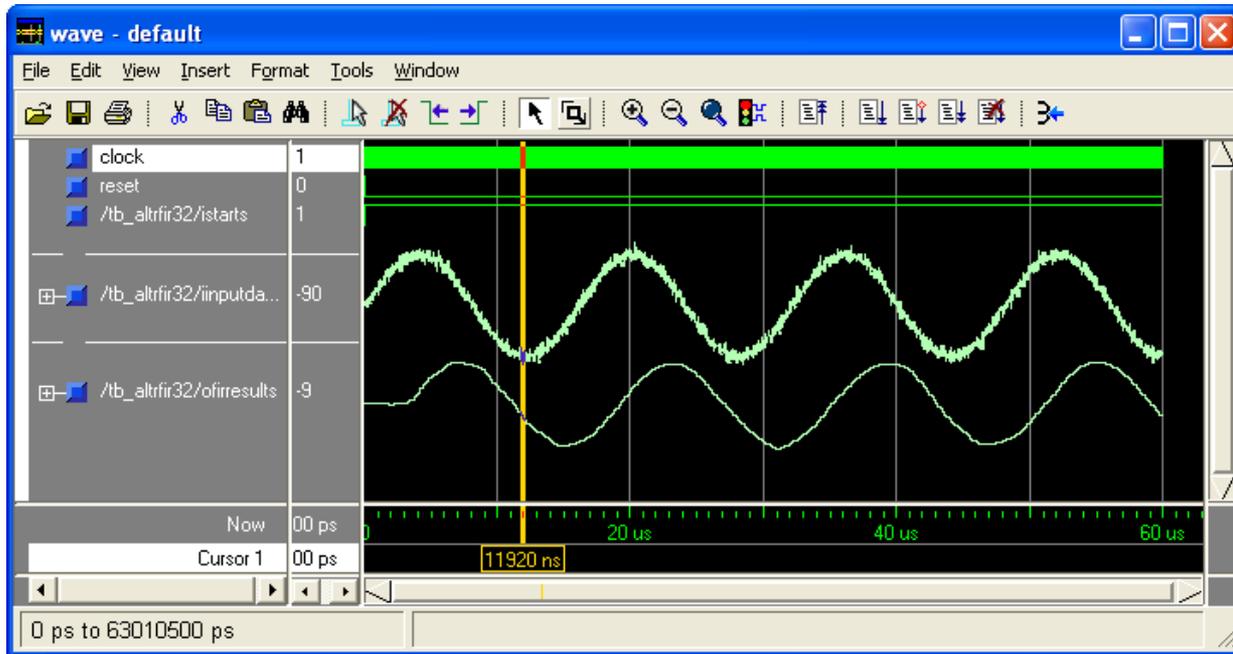
File name: tb_AltrFir32.tcl

Files of type: Macro Files (*.do,*.tcl)

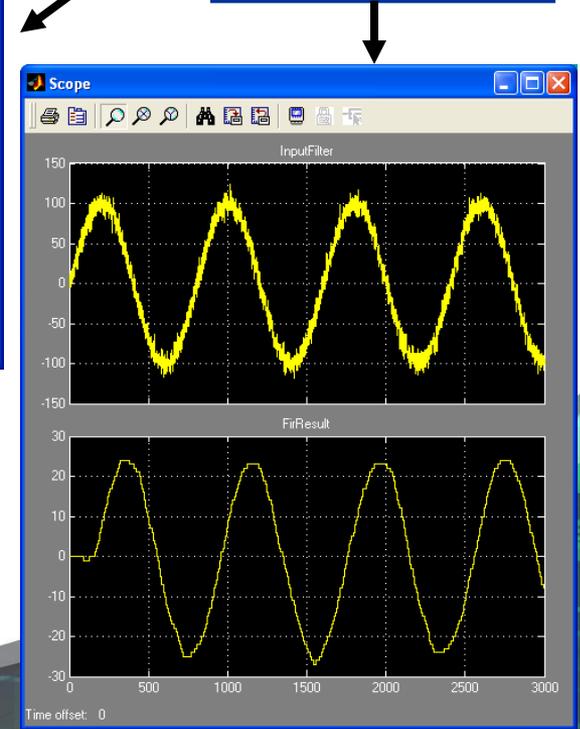
Open Cancel

<No Design Loaded>

Perform Verification

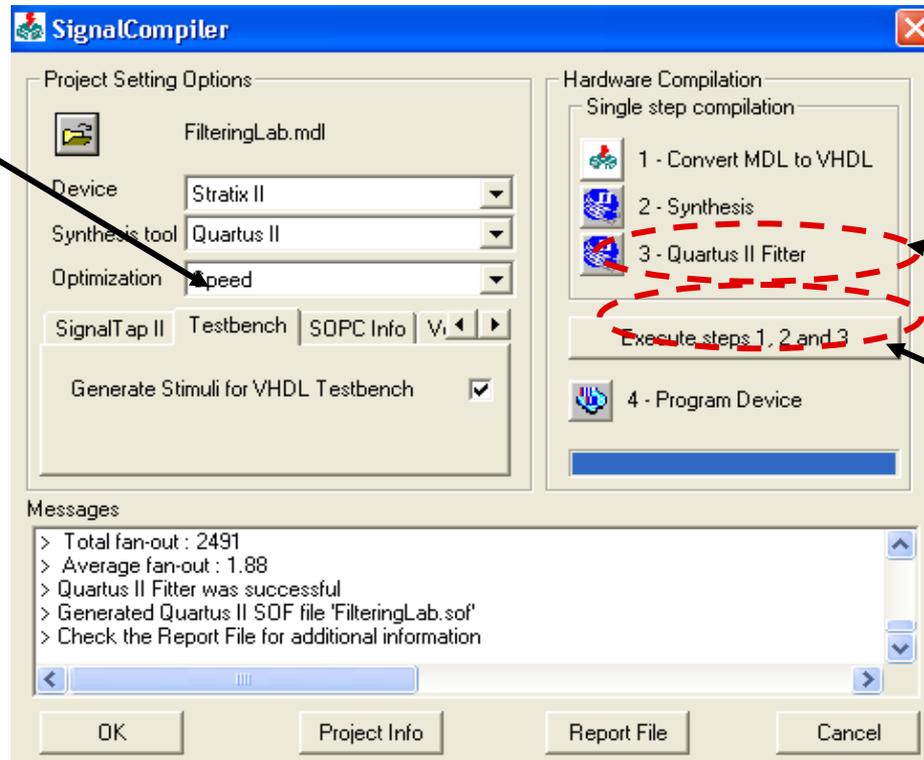


ModelSim
vs.
Simulink



Step 6: Synthesize HDL and Place and Route

- Leonardo Spectrum
- Synplify
- Quartus II



•Synthesis

•Quartus II Fitter

Step 7: Program Device

The screenshot displays the SignalCompiler software interface. The main window is titled "SignalCompiler" and shows the "Project Setting Options" for a project named "FilteringLab.mdl". The "Device" is set to "Stratix II", the "Synthesis tool" is "Quartus II", and the "Optimization" is "Speed". The "Generate Stimuli for VHDL Testbench" checkbox is checked. The "Hardware Compilation" section shows a "Single step compilation" process with four steps: 1 - Convert MDL to VHDL, 2 - Synthesis, 3 - Quartus II Fitter, and 4 - Program Device. A button labeled "Execute steps 1, 2 and 3" is visible. The "Messages" pane at the bottom shows the following output:

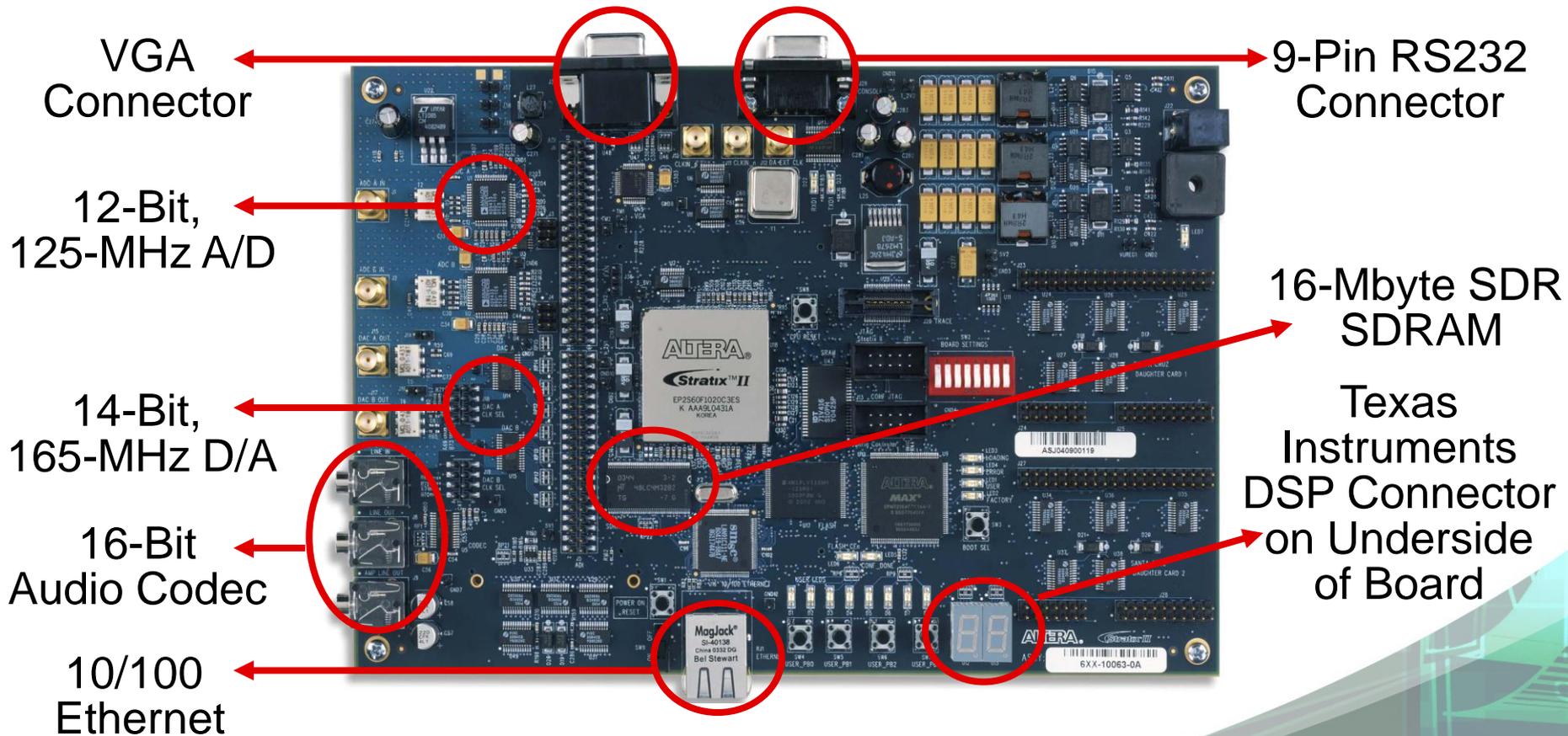
```
> Total fan-out : 2491
> Average fan-out : 1.88
> Quartus II Fitter was successful
> Generated Quartus II SOF file 'FilteringLab.sof'
> Check the Report File for additional information
```

Buttons for "OK", "Project Info", "Report File", and "Cancel" are located at the bottom of the window. In the background, a "Simulink Library Browser" window is visible, showing a tree view of components under "Stratix II DSP Board 2560 Configuration".

**Download Design
to DSP
Development Kits**

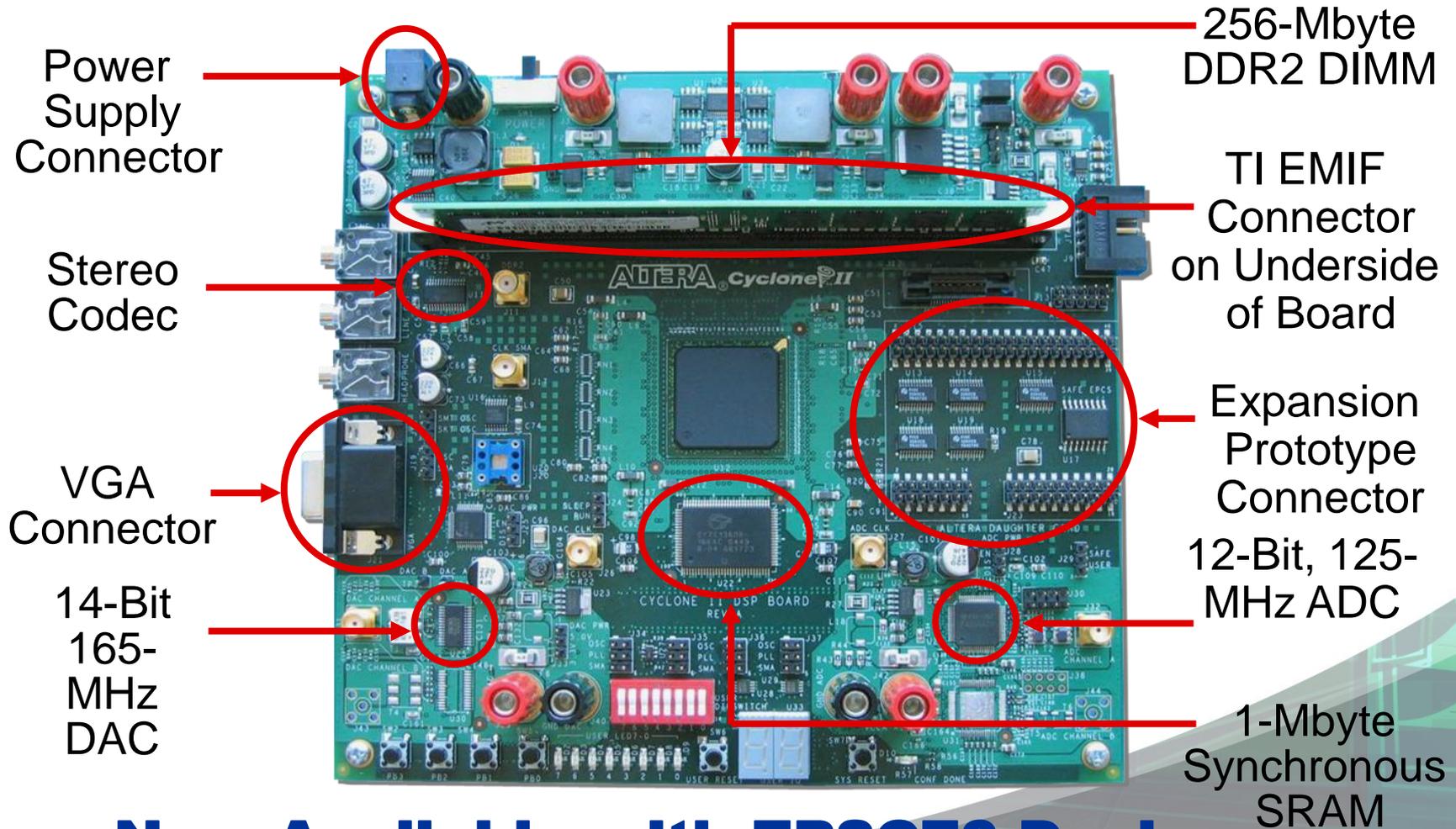


Stratix II DSP Development Board



Available with EP2S60 or EP2S180 Device

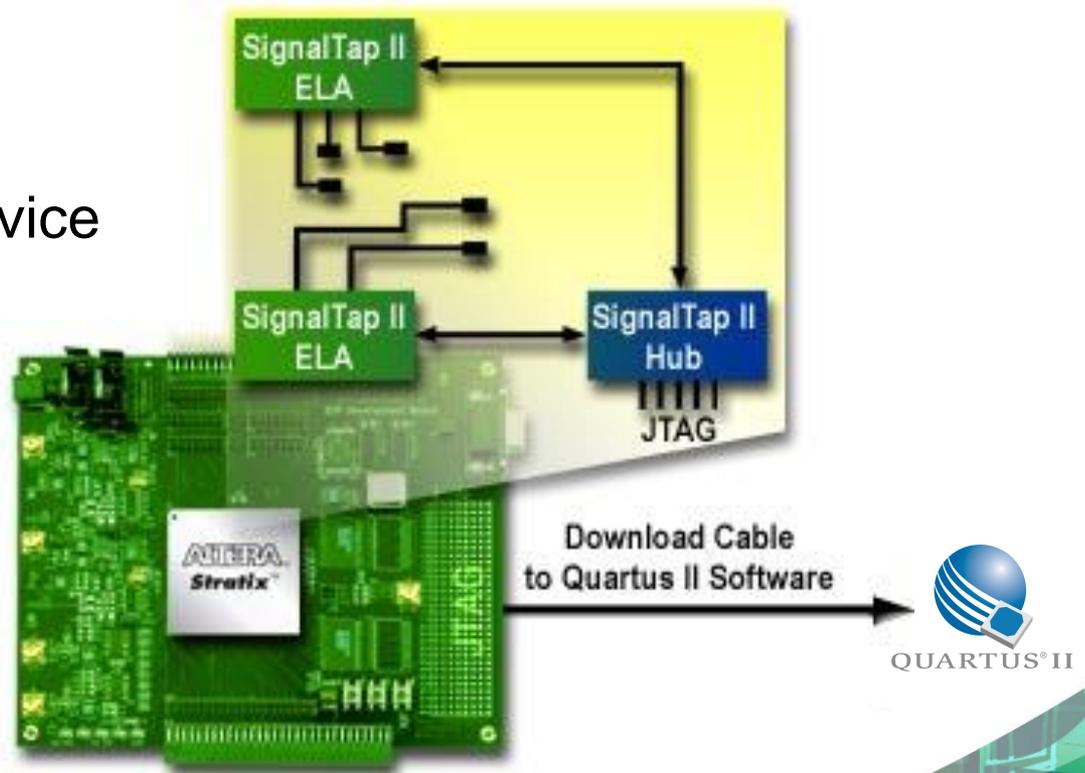
Cyclone II DSP Development Board



Now Available with EP2C70 Device

Step 8: SignalTap II Logic Analyzer

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



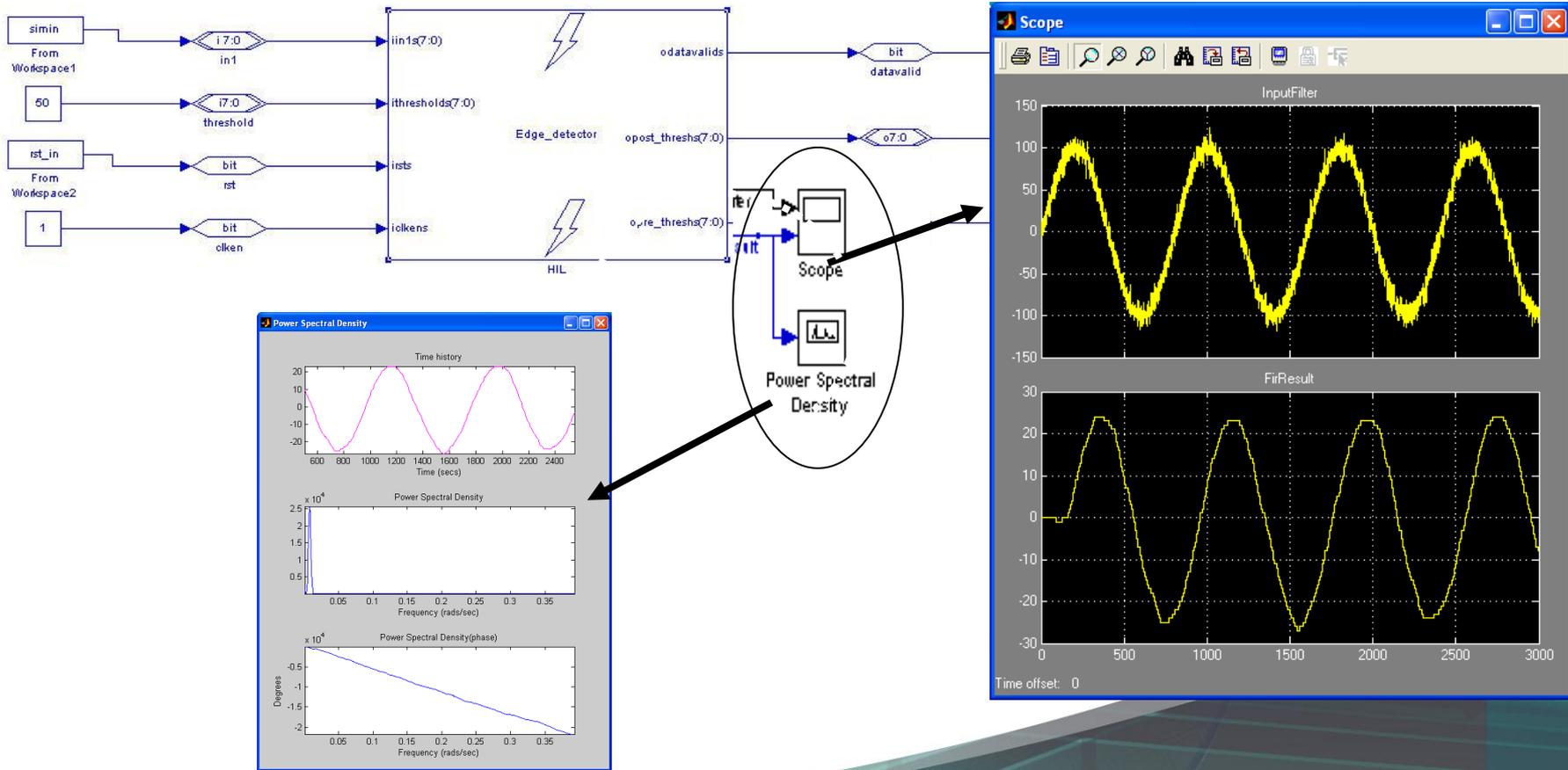
SignalTap II Logic Analyzer

The image displays the MATLAB environment with several windows open. The main workspace shows a list of variables, including phase_acc, plot_created, sample_depth, scrsz, sine_table, and sine_table_1/2. A callout box labeled "Imported Data" points to the workspace list. The Command Window shows the startup file path: C:\M... >>. A window titled "Figure No. 5" displays a plot of imported data, showing a periodic waveform with a callout box labeled "Analysis of Imported Data". A window titled "Figure No. 1: SignalTap Analysis Results" shows a multi-channel waveform plot with a callout box labeled "Imported Plot". The plot shows multiple channels of data over a sample index from 0 to 200. The x-axis is labeled "Sample Index" and the y-axis is labeled "sine_table_1".

Name	Size	Bytes	Class
phase_acc	256x1	2048	double
phase_acc_1	256x1	2048	double
phase_acc_2	256x1	2048	double
phase_acc_3			
phase_acc_4			
phase_acc_5			
plot_created	1x1	8	double
sample_depth	1x1	98	cell array
scrsz	1x4	32	double
sine_table	256x1	2048	double
sine_table_1	256x1	2048	double
sine_table_2	256x1		

Hardware in Loop (HIL)

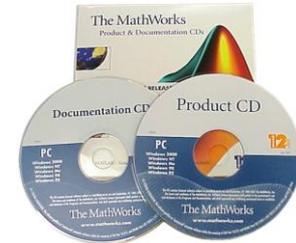
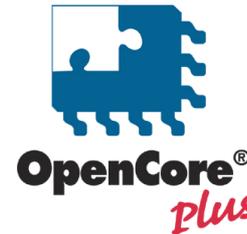
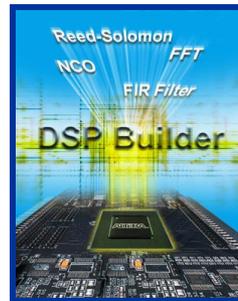
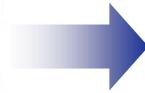
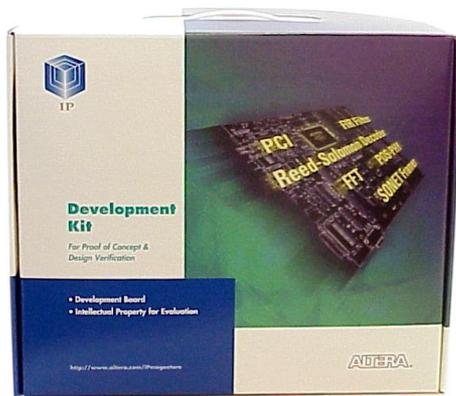
HIL Edge Detection Design



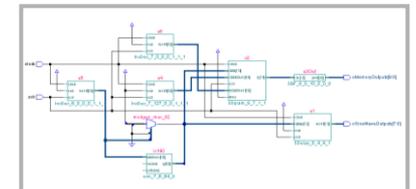
Design Flow Review

- 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

Altera DSP Development Kits



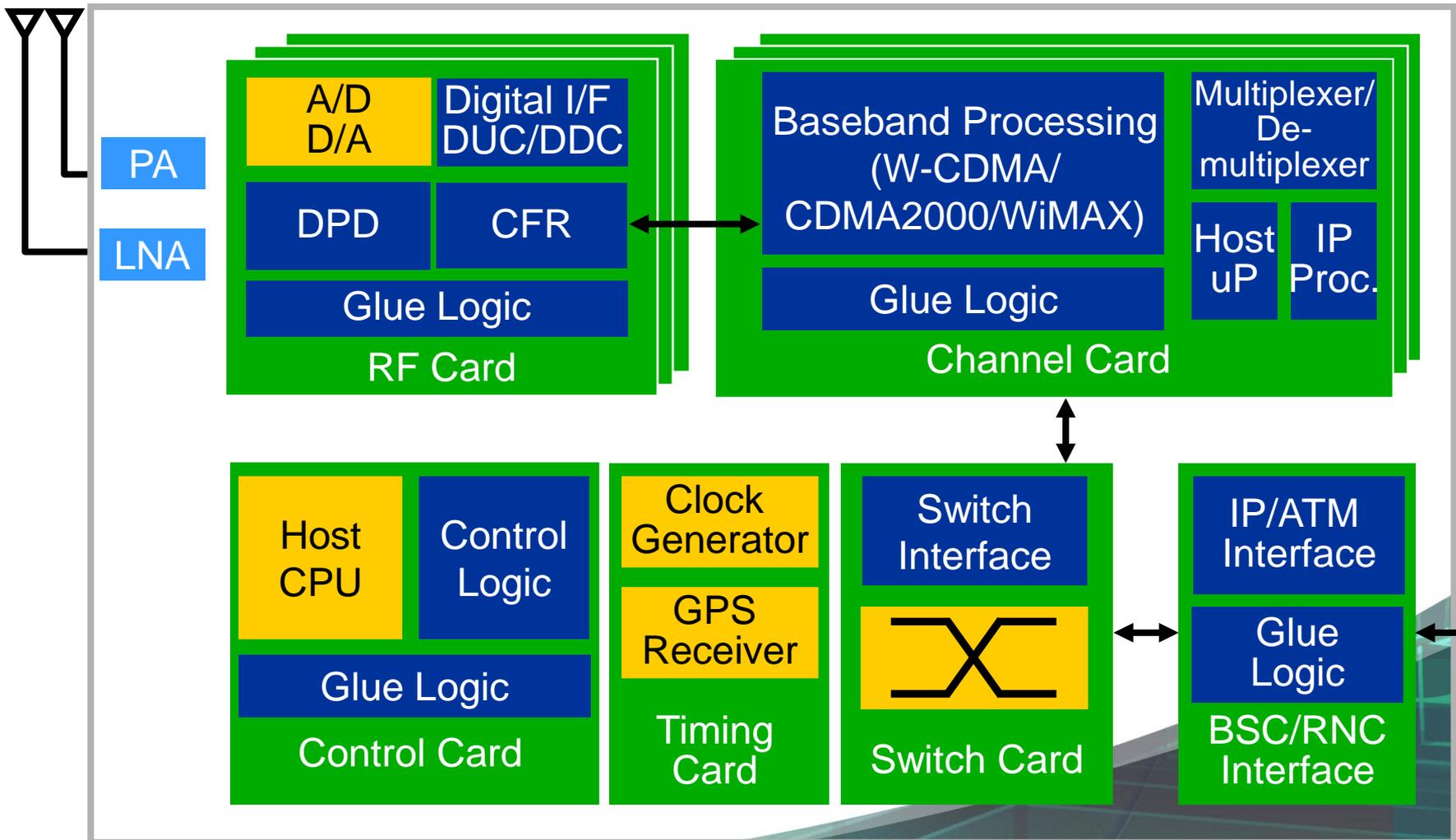
30-Day Evaluation Version



System Reference Designs

WiMAX DUC and DDC Design Case Study

Base Station Architecture Overview

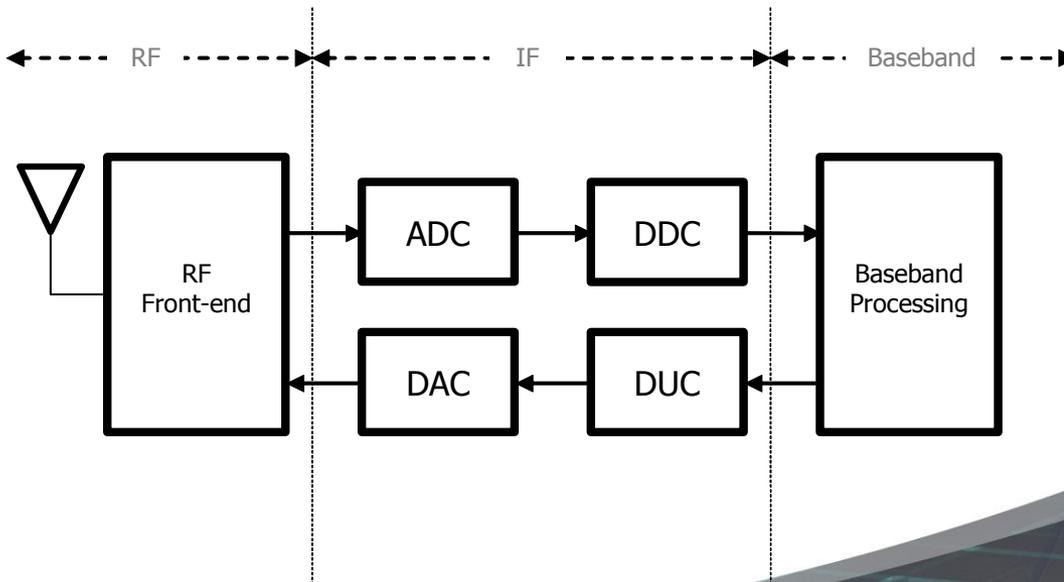


■ PLD Applications

Reference Design Overview

■ DUC/DDC

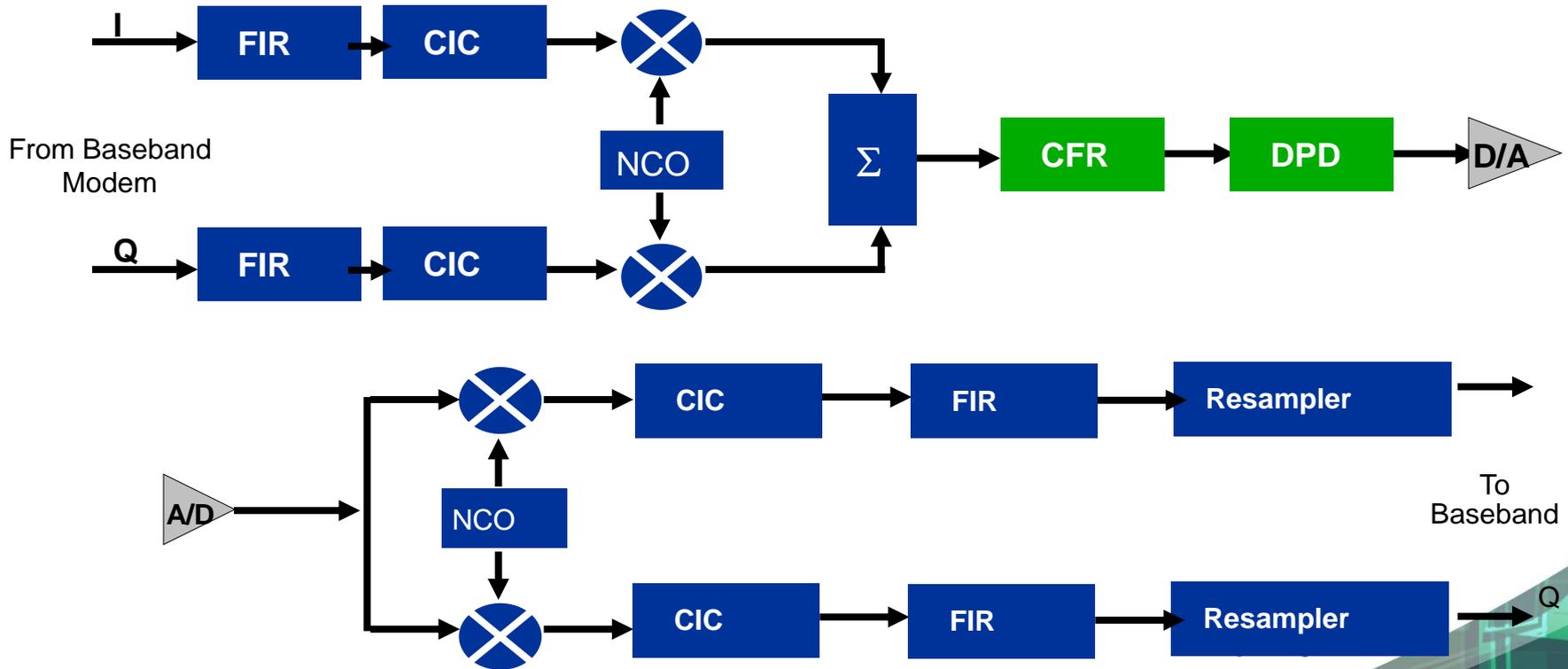
- Provides the link between digital baseband and analog RF front end of generic transceiver
- High throughput signal processing required makes FPGA ideal platform



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

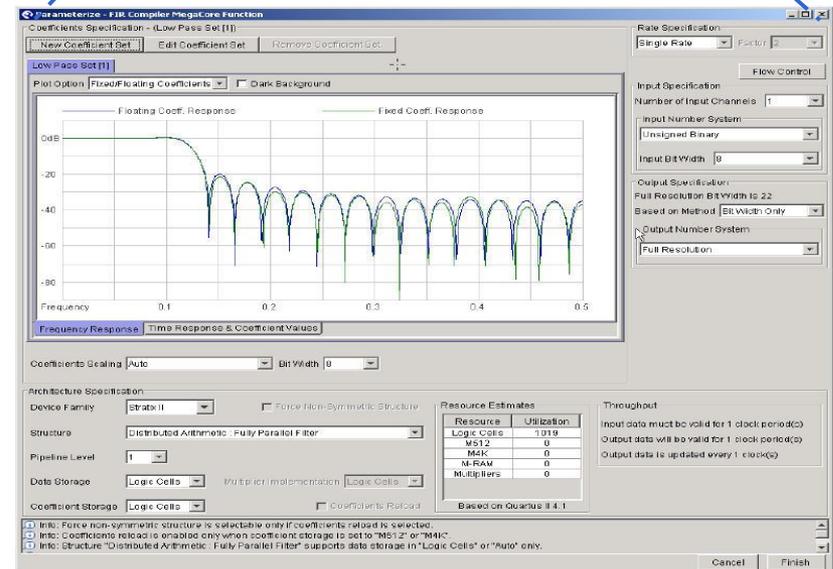
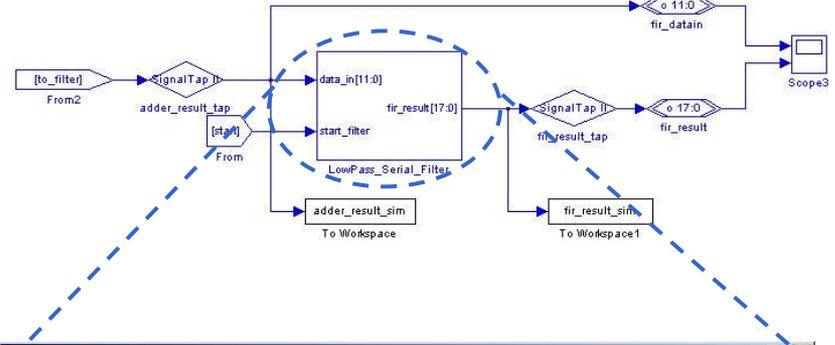
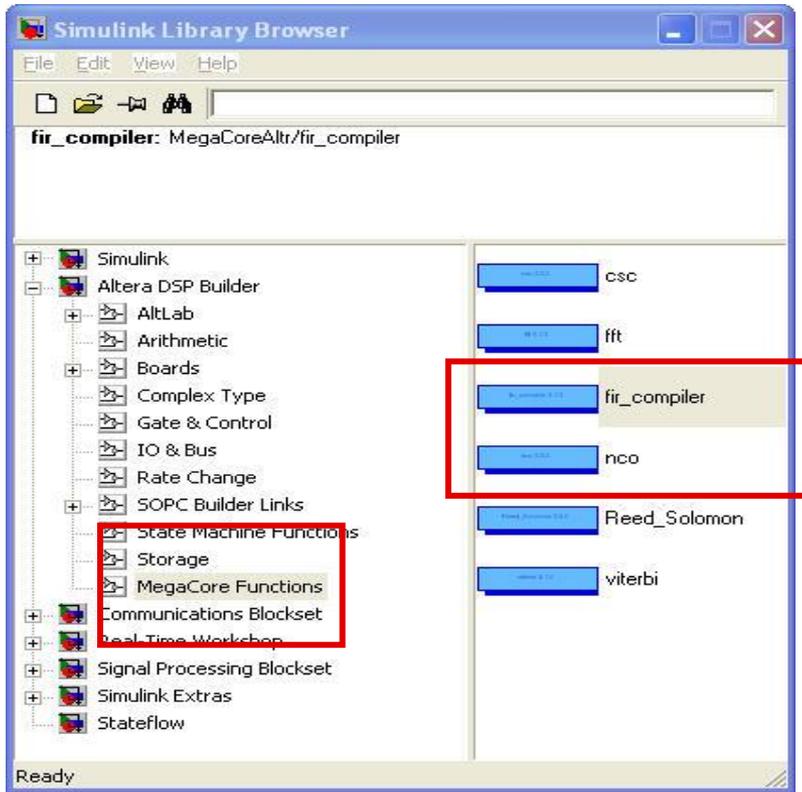
DUC and DDC High-Level Block Diagrams



■ DUC and DDC:
Wireless, Military, Medical, Broadcast

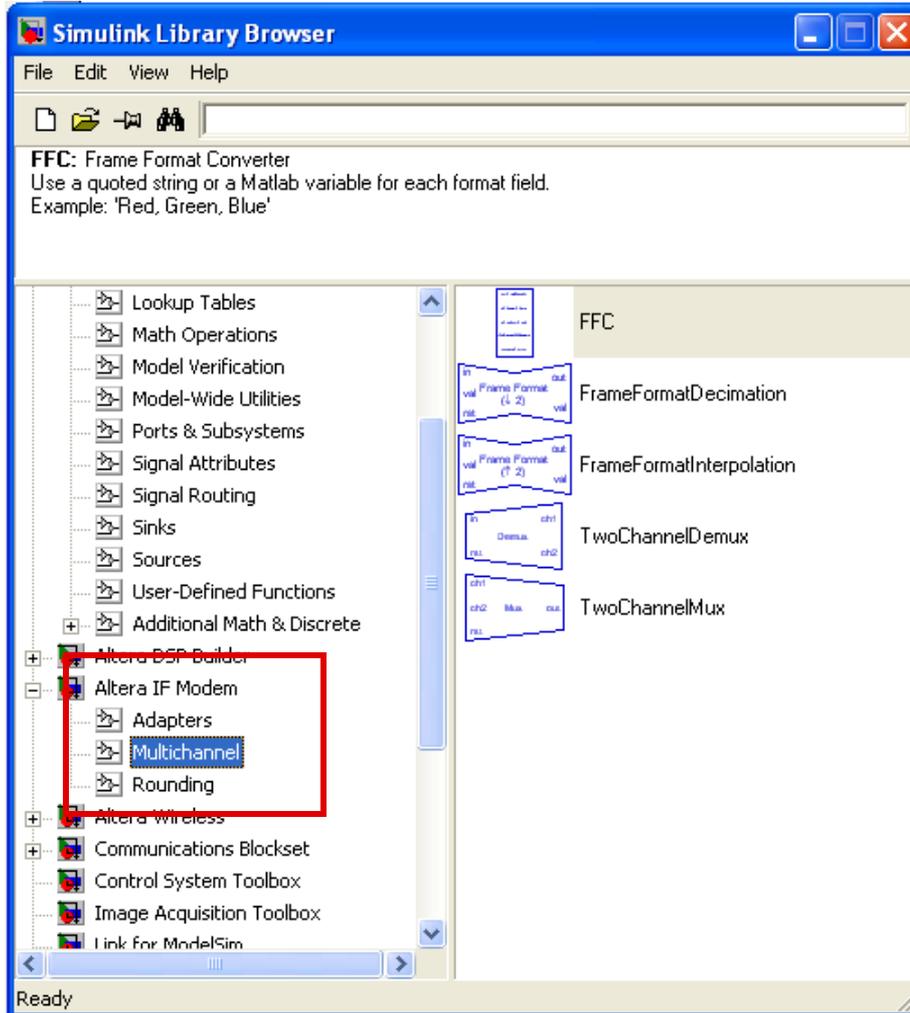
■ Crest-Factor Reduction (CFR) and
Digital Predistortion (DPD): Wireless

DSP Builder Implementation: IP MegaCore Library



IP Can Be Added to the Library Separately

DSP Builder Implementation: Digital Intermediate Frequency (IF) Library



■ Adapters

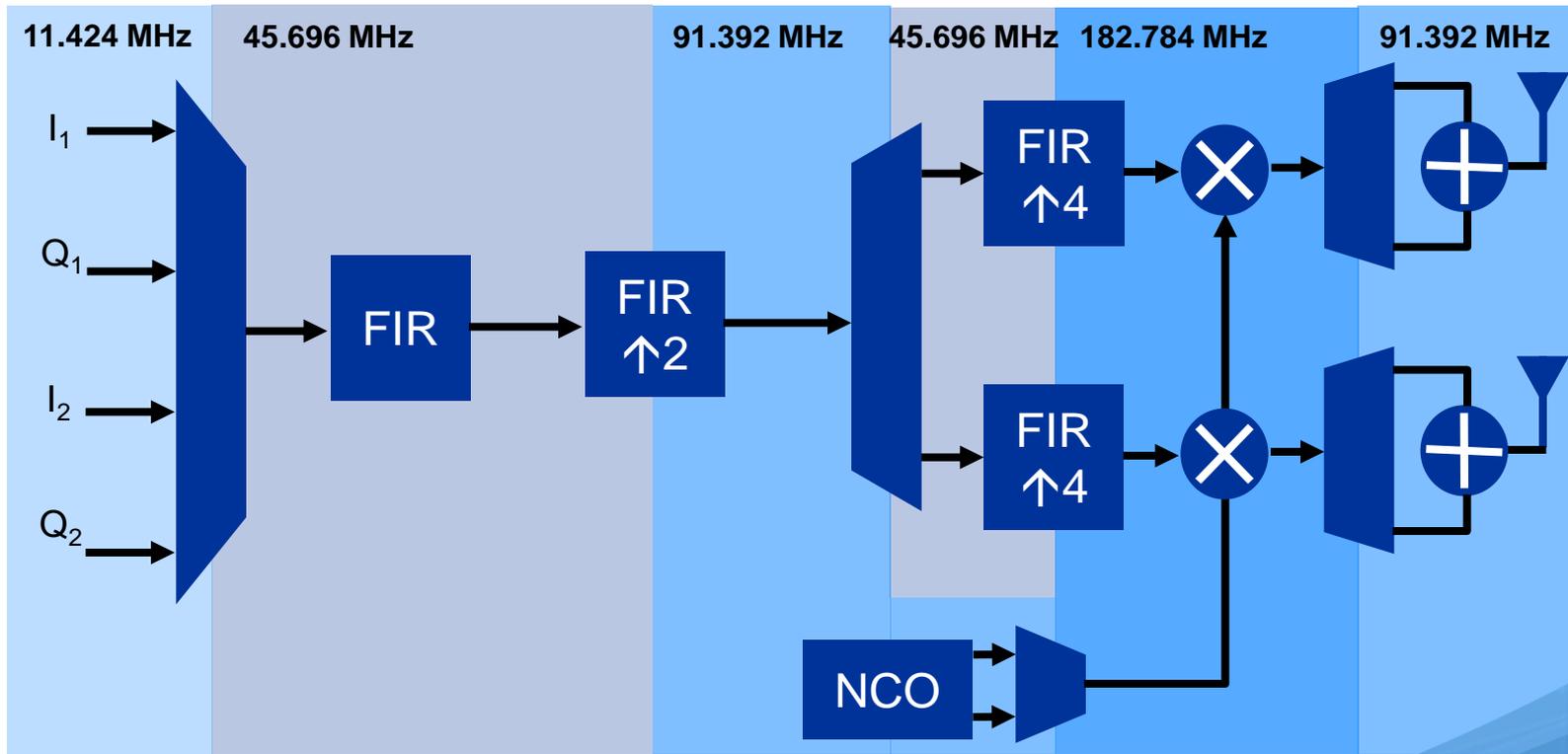
- Provide input/output interface to finite impulse response (FIR) filter

■ Multichannel

- Frame format converter
- Decimation
- Interpolation
- Multiplexer
- Demultiplexer

■ Rounding

DUC With 2 Antennas Design Architecture



Timeshare DUC Hardware Between Antennas

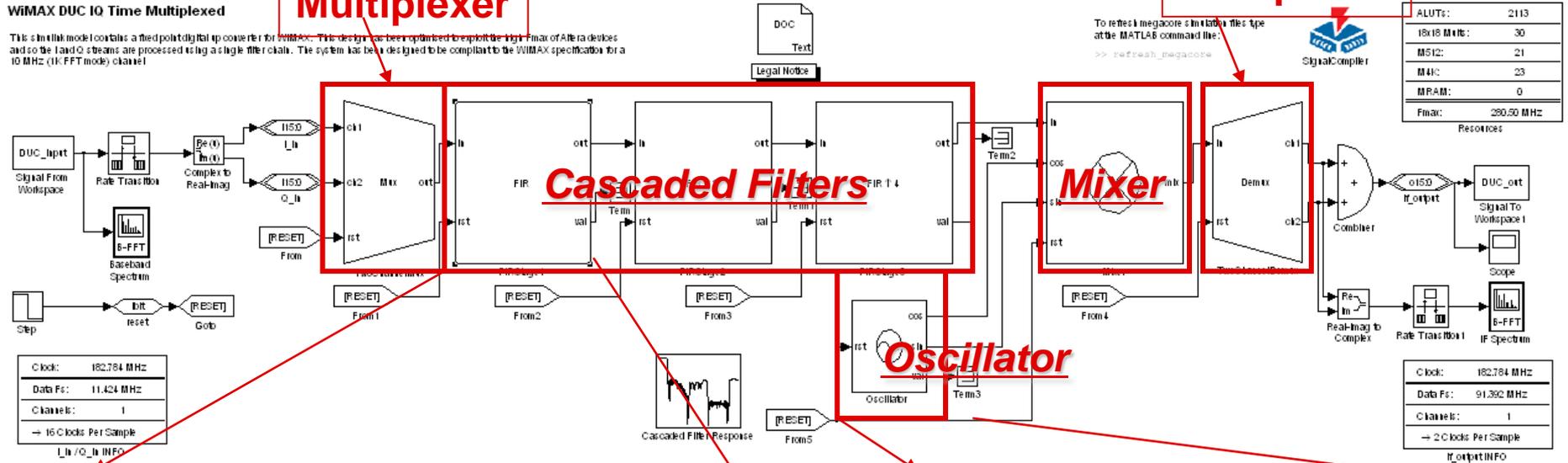
DSP Builder Implementation: DUC Example Design With 2 Antennas

Multiplexer

Multiplexer

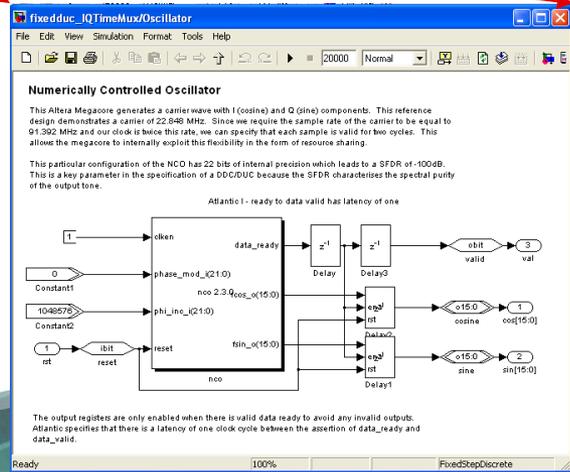
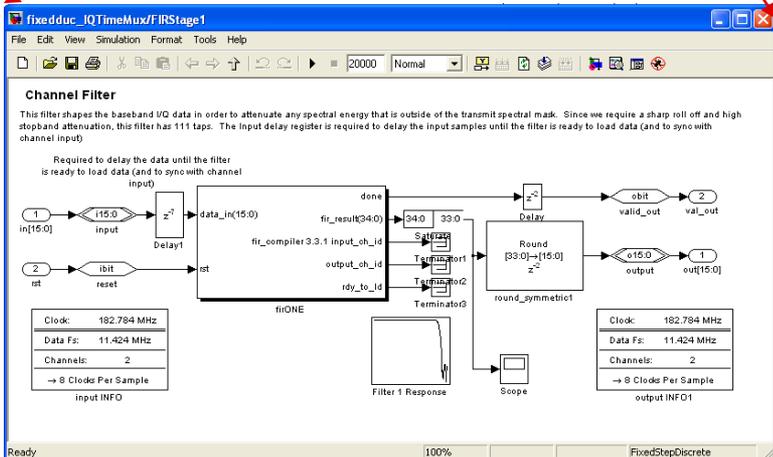
WIMAX DUC IQ Time Multiplexed

This block model contains a fixed point digital up converter for WIMAX. This design has been optimized to exploit the high performance of Altera devices and so the I and Q streams are processed using a single filter chain. The system has been designed to be compliant to the WIMAX specification for a 10 MHz (1K-FFT mode) channel.



Clock:	182.784 MHz
Data Fs:	11.424 MHz
Channels:	1
→ 16 Clocks Per Sample	
Lh/Qh INFO	

Clock:	182.784 MHz
Data Fs:	91.392 MHz
Channels:	1
→ 2 Clocks Per Sample	
If_output INFO	



Use FIR Compiler IP

Use Numerically Controlled Oscillator (NCO) Compiler IP

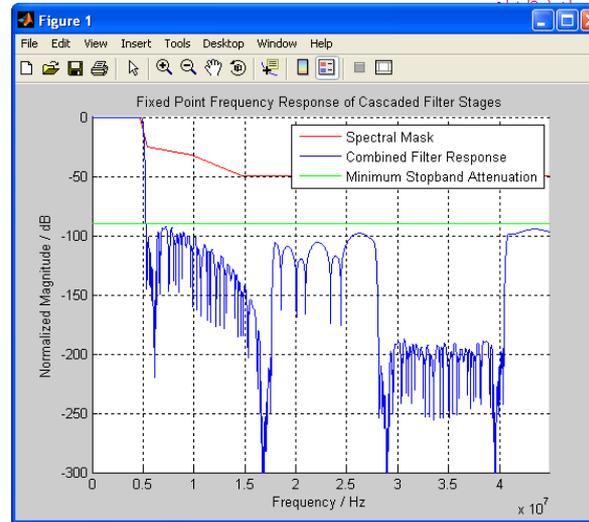
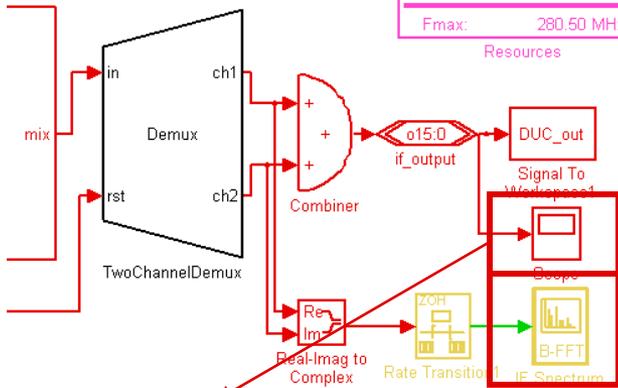


Simulation

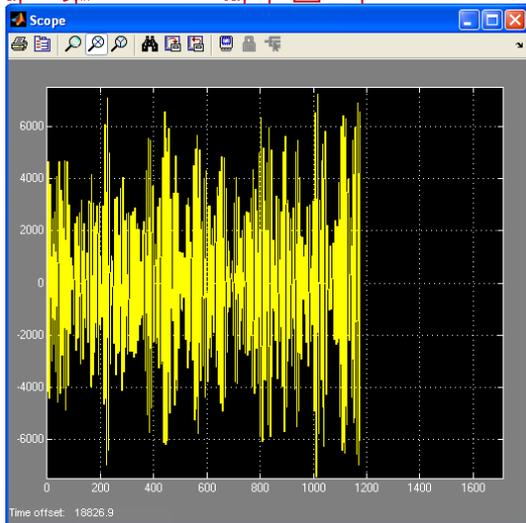
egacore simulation files type
AB command line:
h_megacore



ALUTs:	2113
18x18 Mults:	30
M512:	21
M4K:	23
MRAM:	0
Fmax:	280.50 MHz
Resources	

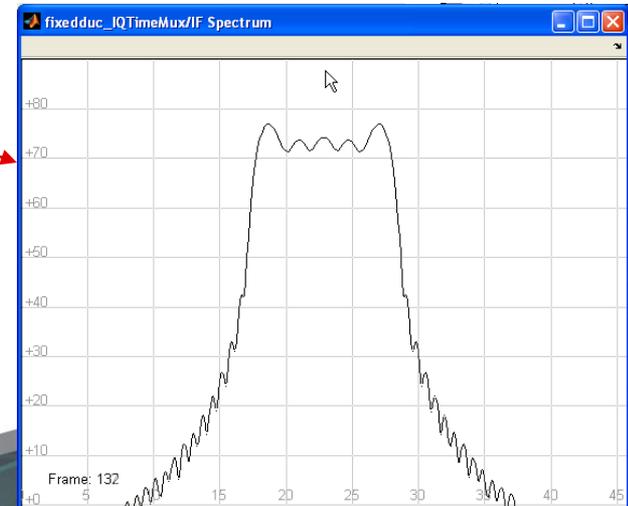


Cascaded Filter Output



182.784 MHz
91.392 MHz
1
Clocks Per Sample

After Upconversion



Convert to VHDL: SignalCompiler

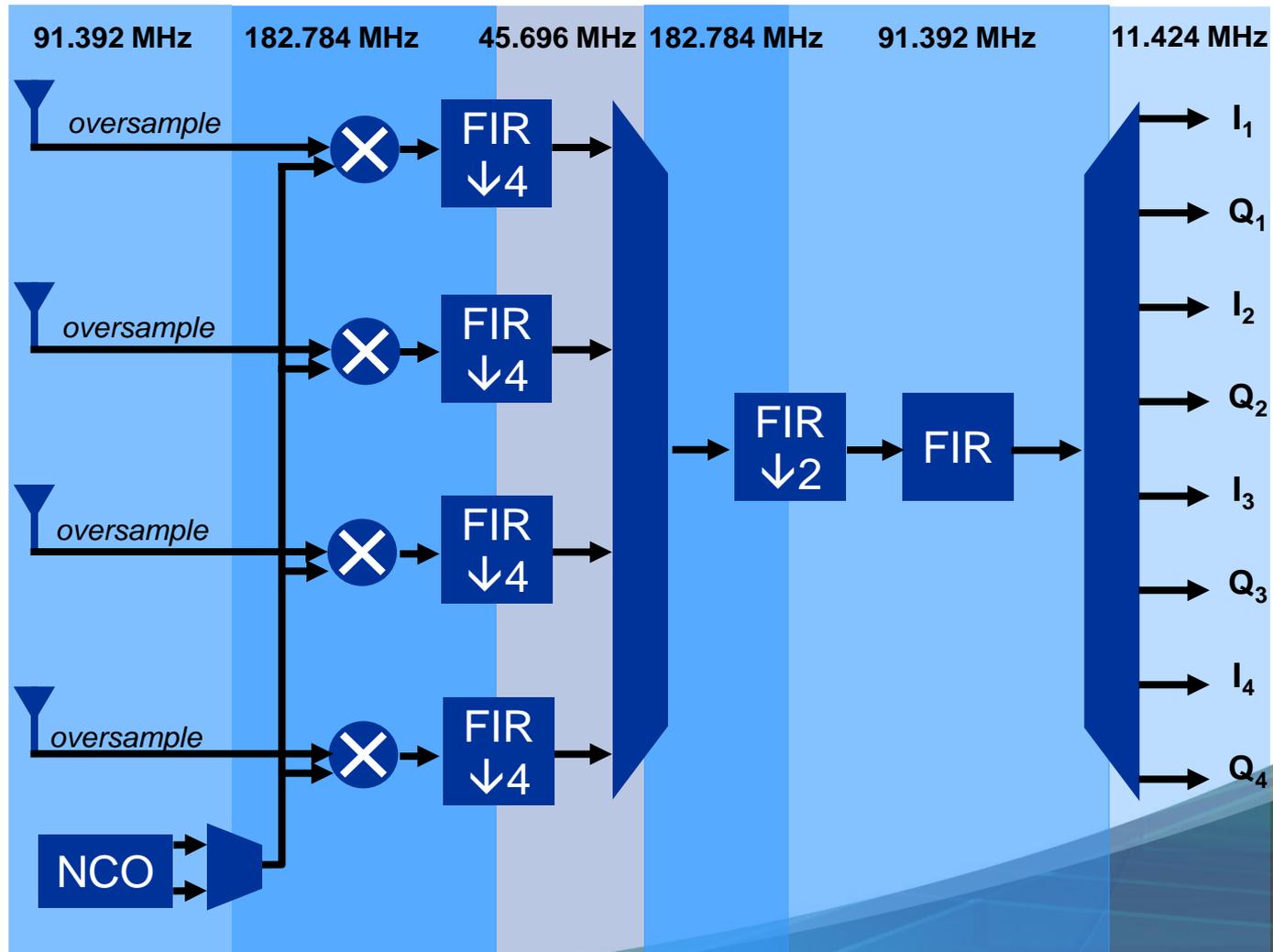
The image shows a screenshot of the SignalCompiler software interface. On the left, a portion of a circuit diagram is visible, featuring a 'TwoChannelDemux' block with inputs 'in' and 'rst', and outputs 'ch1' and 'ch2'. A 'Combiner' block is connected to 'ch1' and 'ch2', with an output 'e15.0' labeled 'if_output'. A 'Real-Imag to Complex' block is also present, with a 'Rate Trans' block below it. A red dashed circle highlights the SignalCompiler icon in the diagram, which is also shown in a larger inset window.

The SignalCompiler window has a title bar with the text 'SignalCompiler' and a close button. The main area is divided into two panes:

- Project Setting Options:**
 - Project file: example_design_data_path.mdl
 - Device: Cyclone II
 - Synthesis tool: Quartus II
 - Optimization: Speed
 - Buttons: SignalTap II, Testbench, SOPC Info, View (V), Left Arrow, Right Arrow
 - Generate SOPC Builder PTF File:
- Hardware Compilation:**
 - Single step compilation:
 - 1 - Convert MDL to VHDL
 - 2 - Synthesis
 - 3 - Quartus II Fitter
 - Execute steps 1, 2 and 3
 - 4 - Program Device

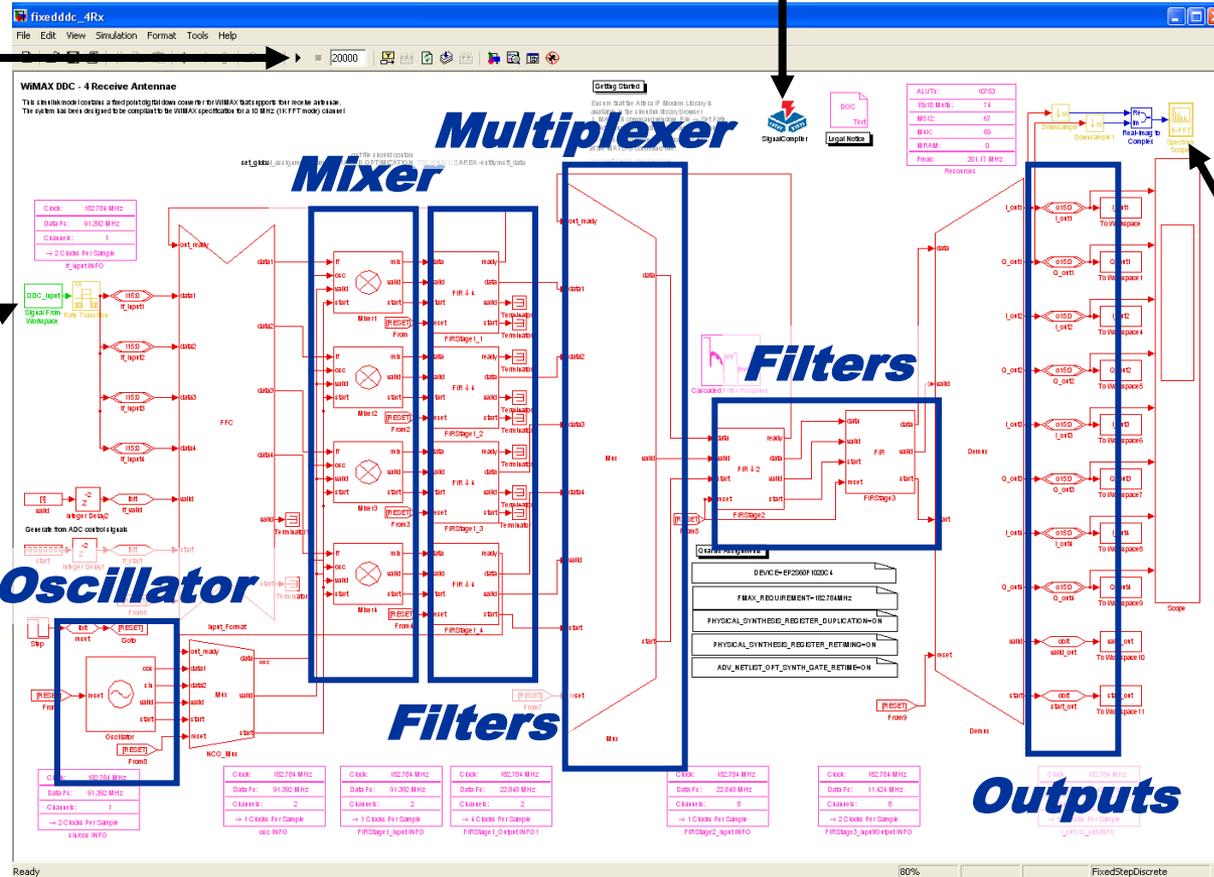
The bottom of the window contains a 'Messages' pane and four buttons: OK, Project Info, Report File, and Cancel.

DDC With 4 Antennas Design Architecture



DSP Builder Implementation: DDC Example Design With 4 Antennas

Simulate
button



Input stimulus
loaded as part
of model
Initialization
function

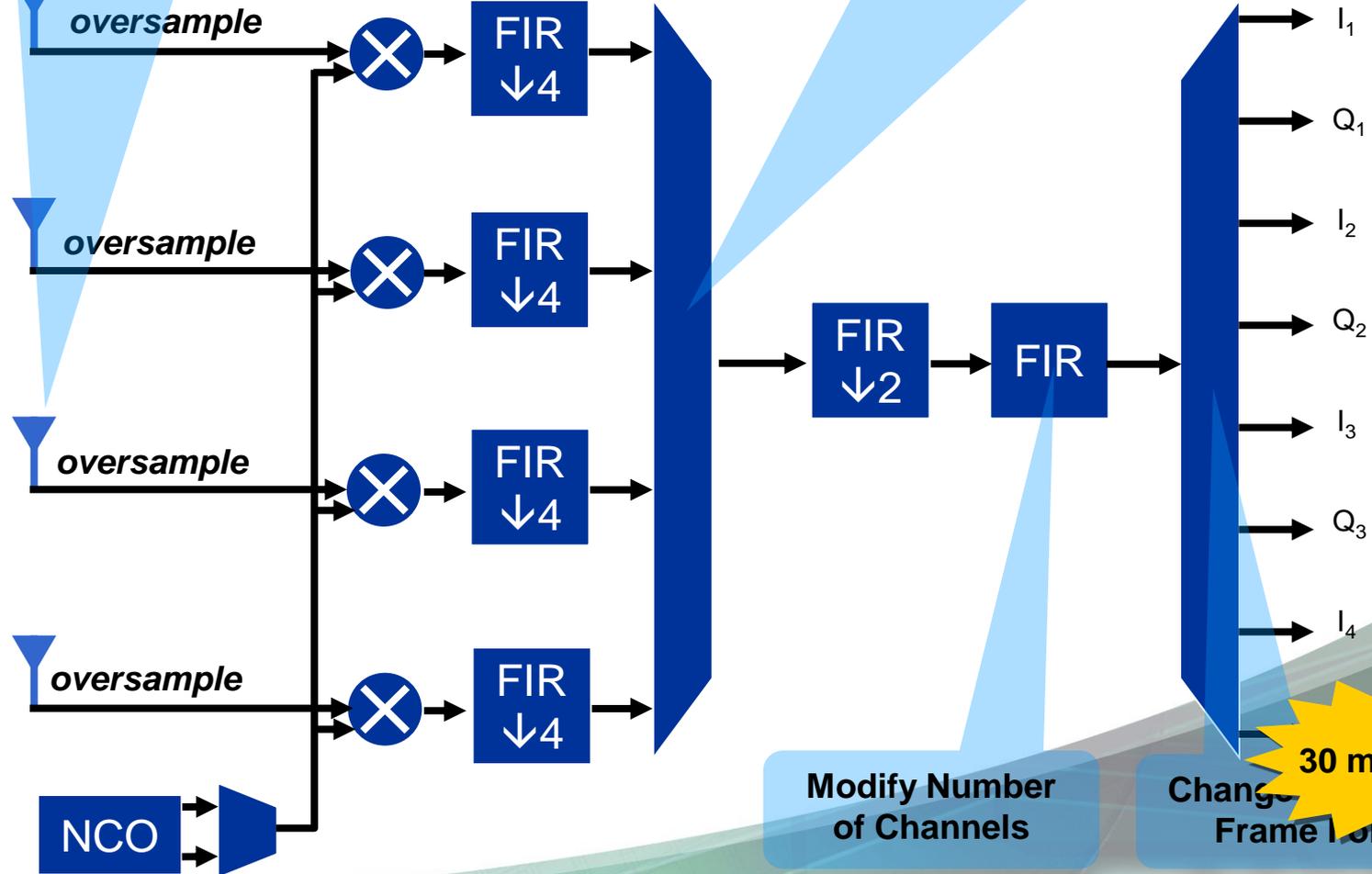
SignalCompiler
(convert to HDL)

Analyze
output data

2xRx DDC Architecture

Remove Redundant Filter Chains

Change Input/Output Frame Formats



Modify Number of Channels

Change Frame Formats

30 minutes!

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***

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

Thank You Q & A