

FPGA Co-Processors in SDR Signal Processing

Agenda

- Applications of FPGA-Based Co-Processors SDR
- Considerations for Co-Processor Design
- Development Tools & Methodologies Available from Altera to Build Co-Processors
 - SOPC Builder
 - DSP Builder
- FPGA Co-Processor Development Examples
 - QAM Modulator
 - FIR Filter

Techniques of Implementing of SDR System Reconfiguration

- Using parameterized radio (and protocol) modules
- Exchange of (a) single component(s) within a module
- Exchange of complete radio modules or protocol layers

Using parameterized radio (and protocol) modules

- Applicable within a standard
 - CDMA2000 – Spreading factors, Viterbi parameterization
 - 3GPP – spreading factor, Viterbi parameterization
- Not Applicable across widely varying standards
 - GSM, 3GPP – Need to replace entire physical layer
 - GSM, CDMA2000 – Need to replace entire network layer - GSM-MAP and ANSI-41

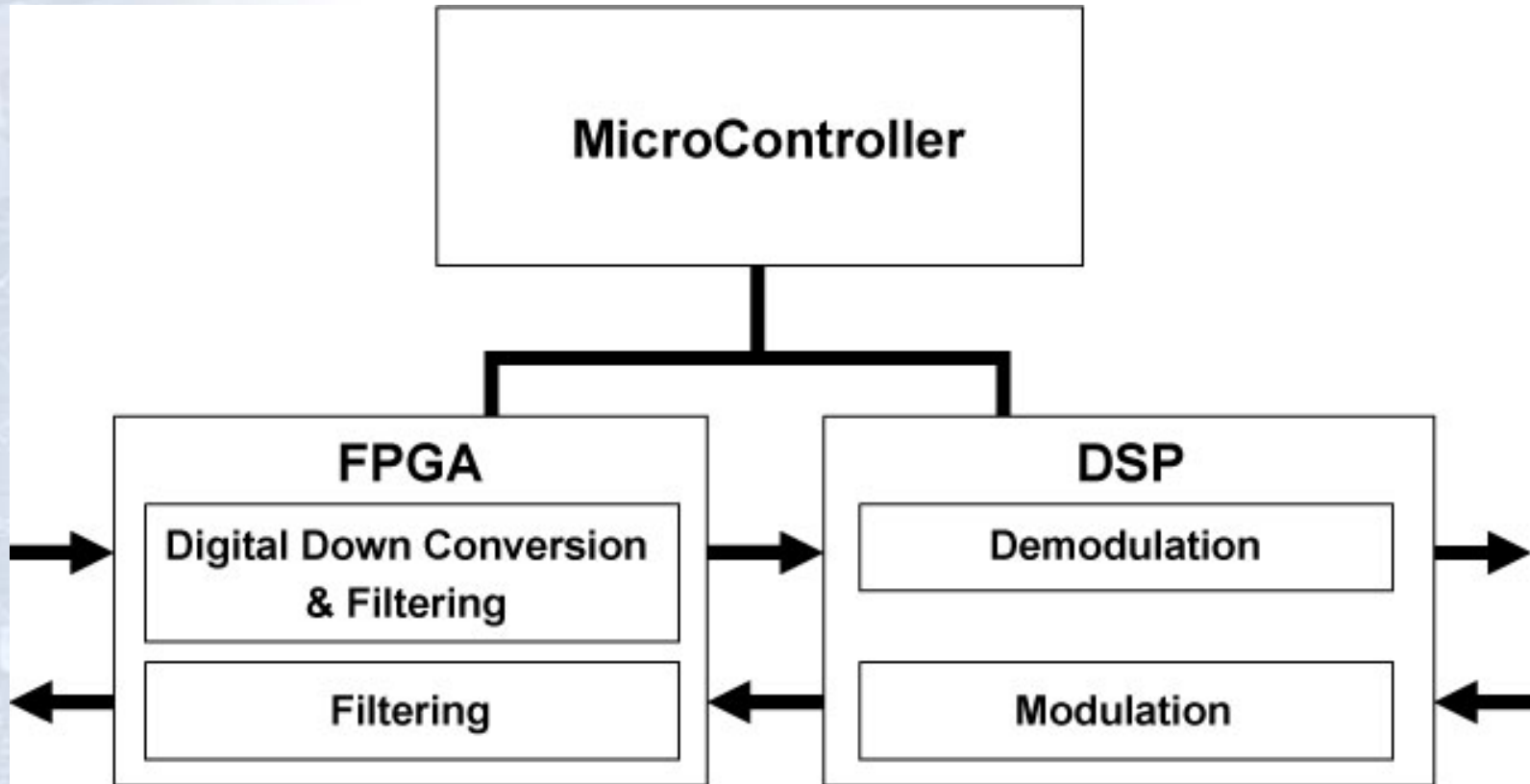
Exchange of (a) single component(s) within a module

- Applicable within a standard
 - 3GPP – Turbo, Viterbi decoding
 - GSM/EDGE – Transceiver chain
- Not Applicable across widely varying standards
 - GSM, 3GPP – Need to replace entire physical layer
 - GSM, CDMA2000 – Need to replace entire network layer - GSM-MAP and ANSI-41

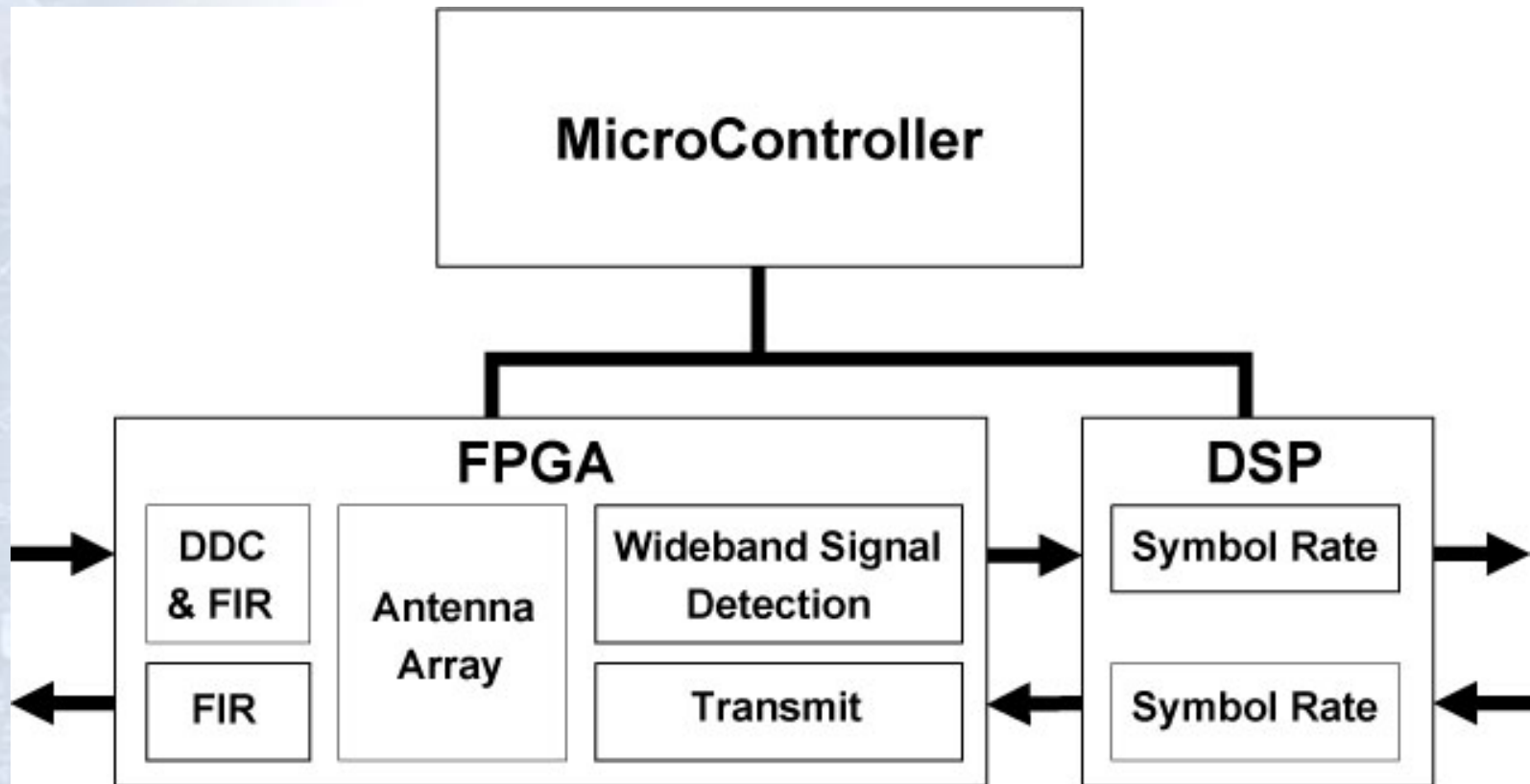
Exchange of complete radio modules or protocol layers

- Applicable across radio standards
 - GSM to EDGE to 3GPP to CDMA2000 to WIFI to
 - GSM-MAP to ANSI-41

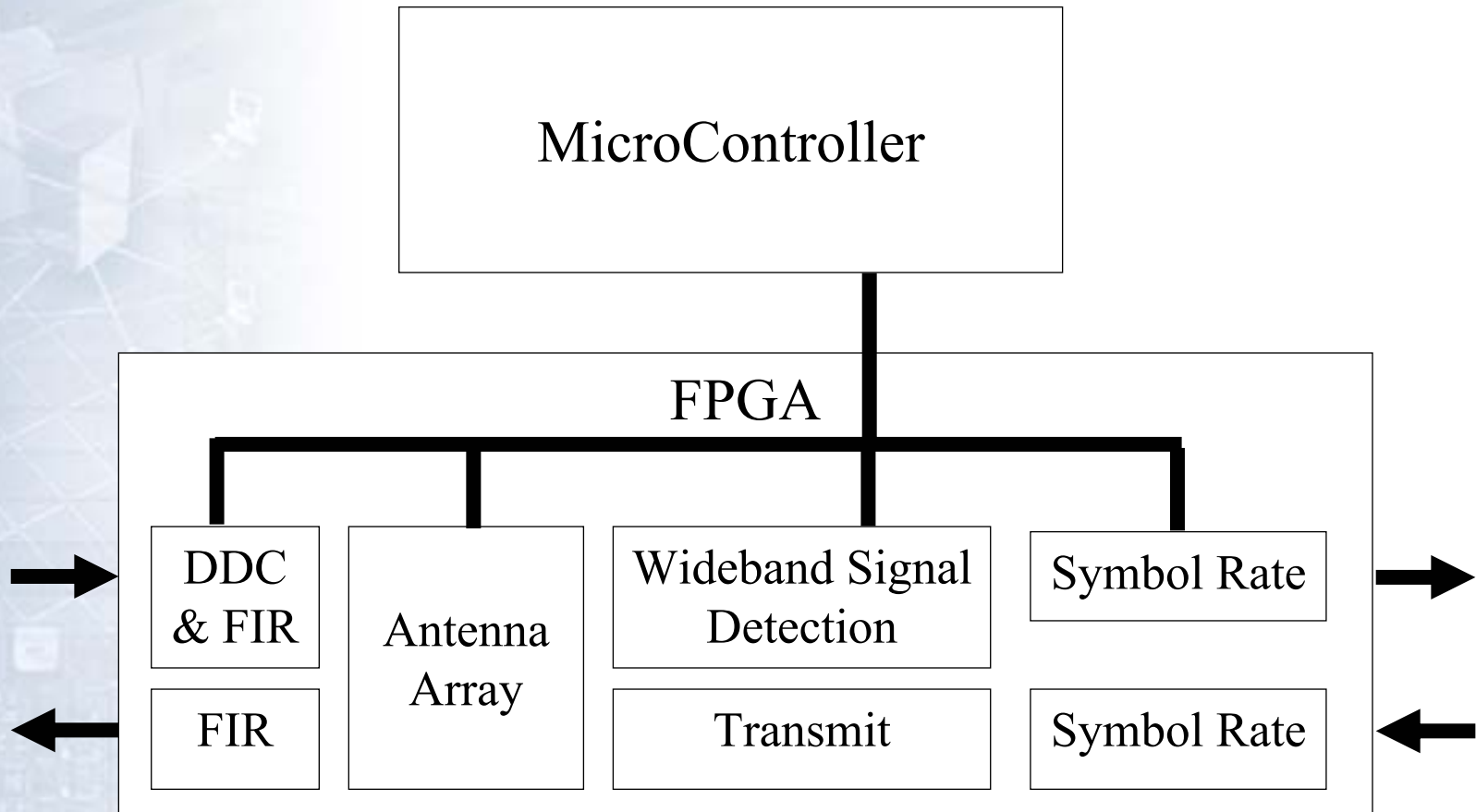
Narrowband System Partitioning



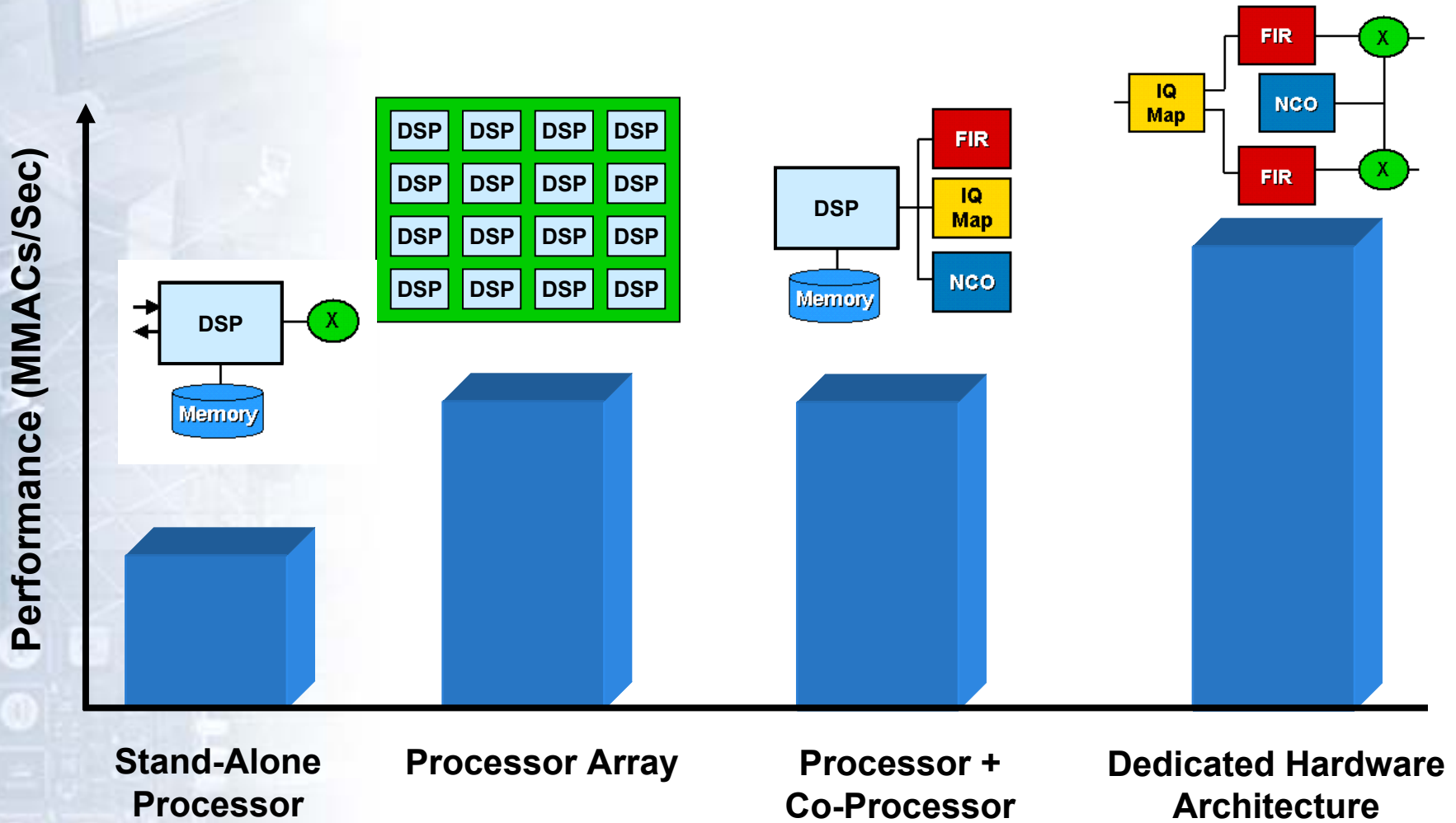
Wideband System Partitioning



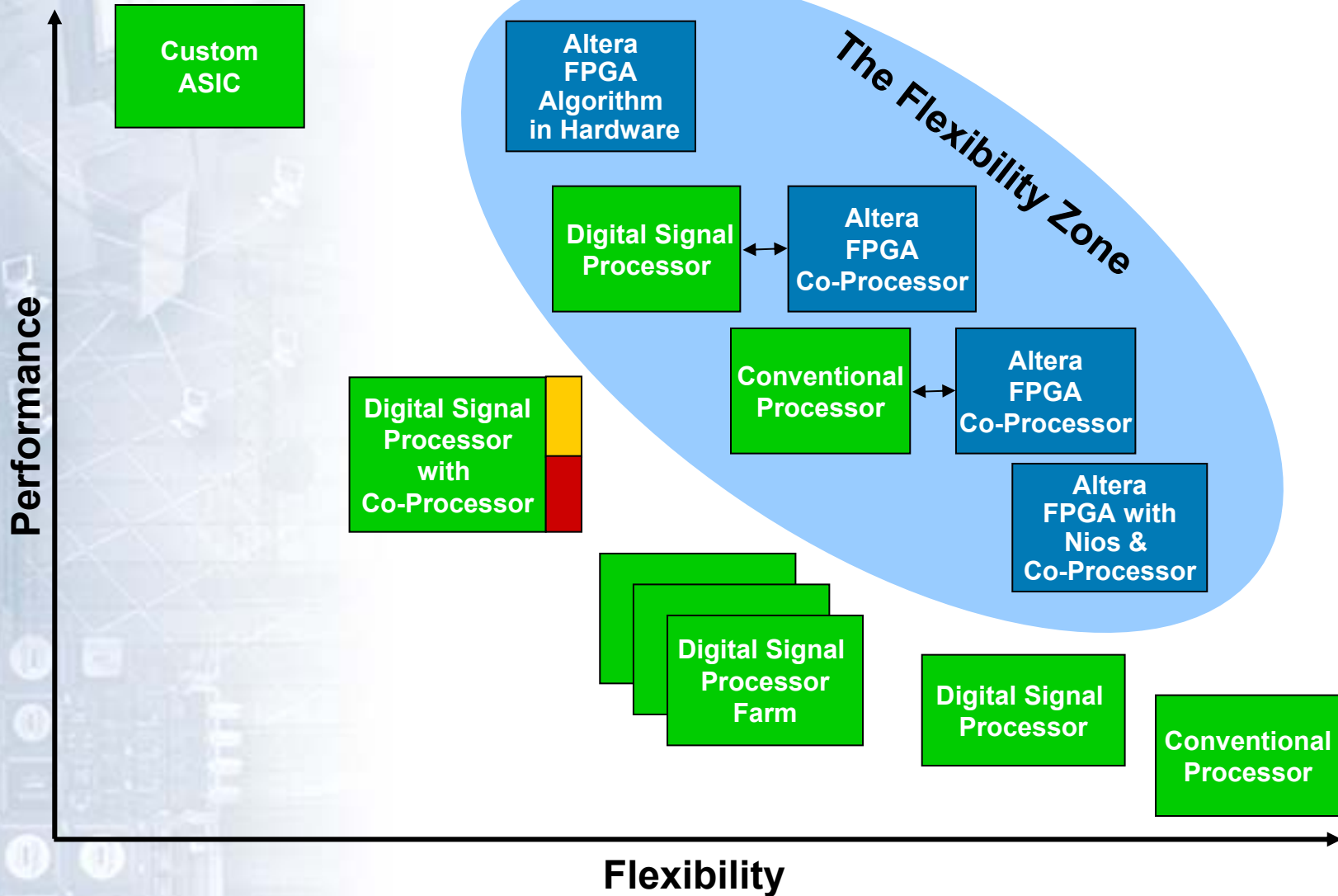
FPGA Based SDR Radio



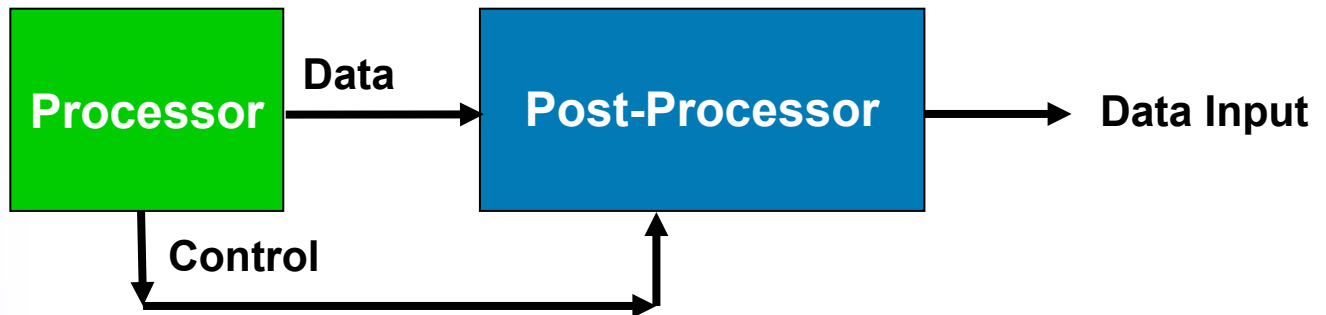
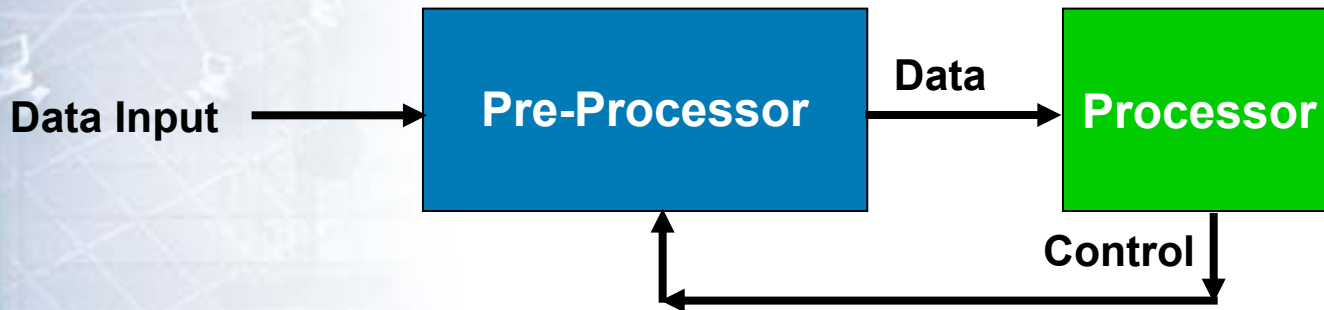
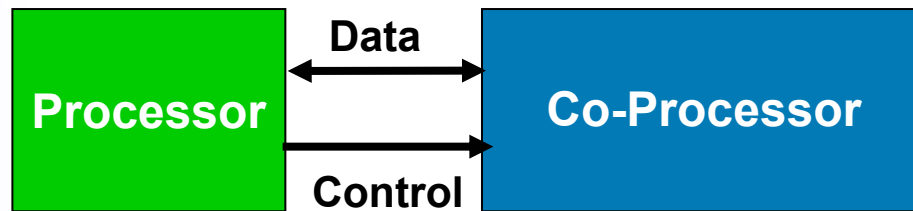
DSP System Architecture Options



Exploring the DSP Design Space

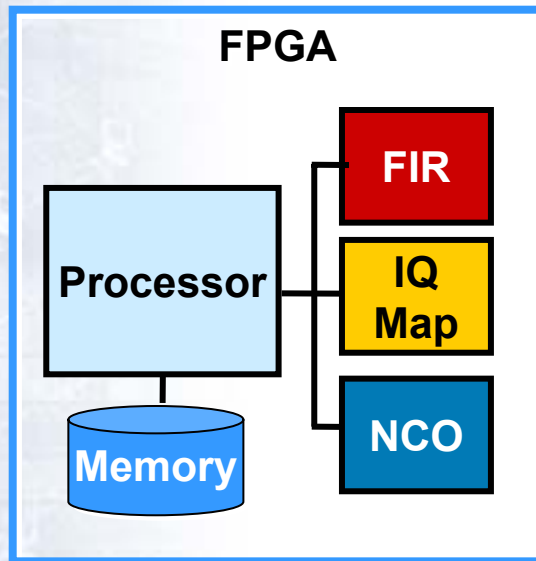


Co-Processing, Pre-Processing and Post-Processing

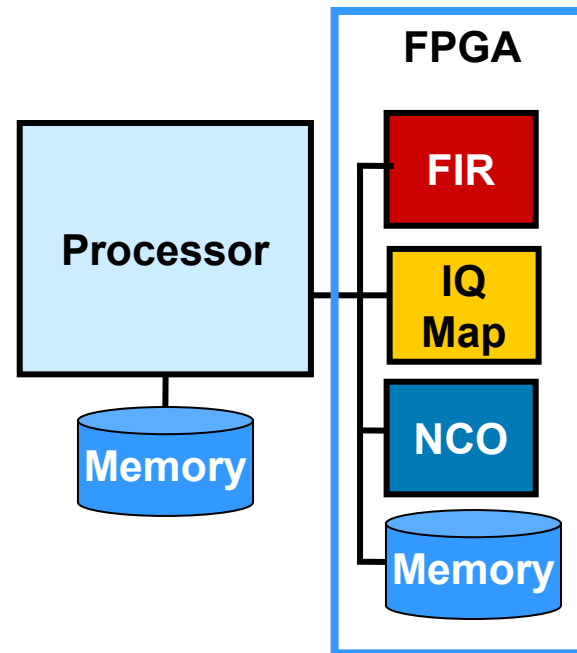


Co-Processing on FPGAs

Processor on FPGA

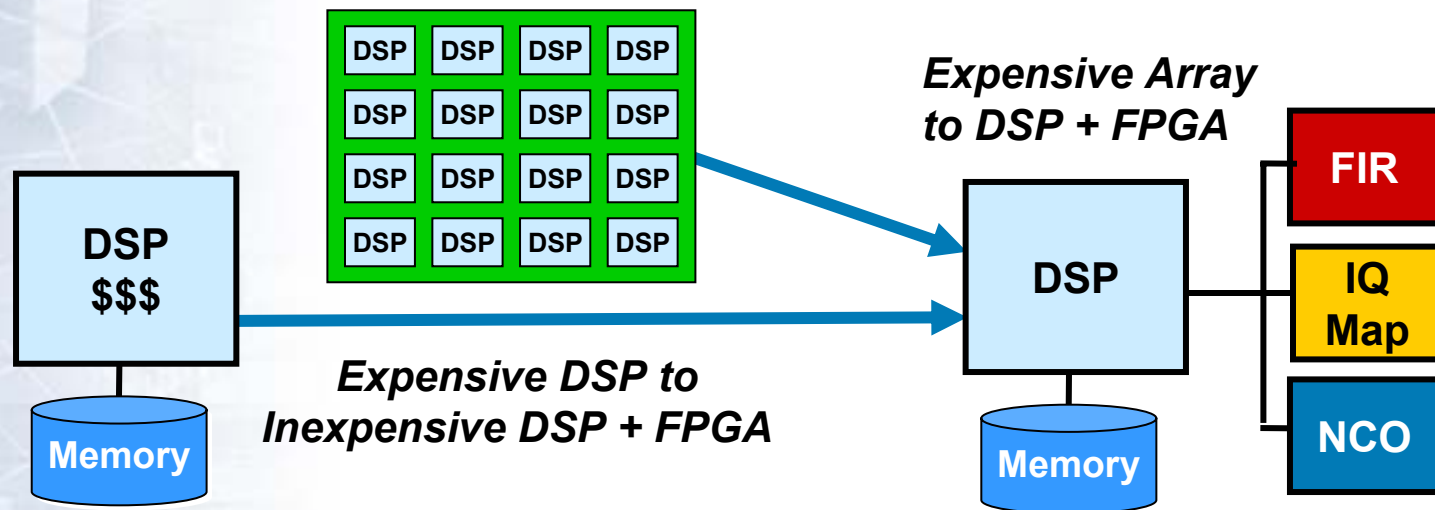


Processor External to FPGA




When Do FPGA Co-Processors Reduce System Cost?

- Off-Loading Algorithms to Co-Processor Reduces Number or Cost of Digital Signal Processors



- Applications
 - Algorithms with Large Amount of Digital Signal Processing & Small Amount of Control Processing

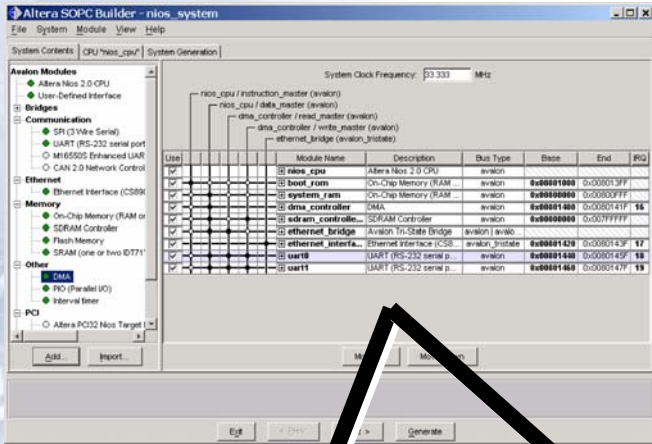


ALTERA®

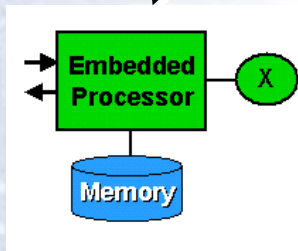
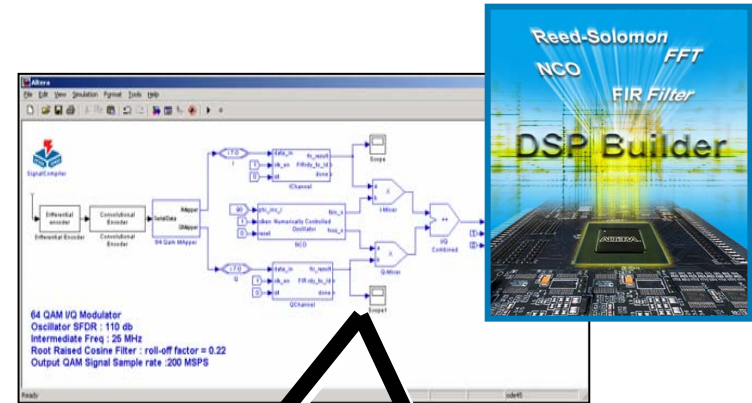
FPGA Co-processor Design Tools and IP

FPGA Co-Processor Design Tools

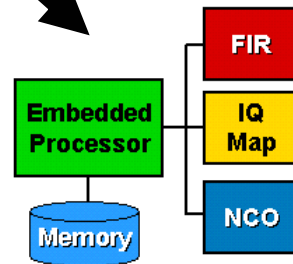
SOPC Builder



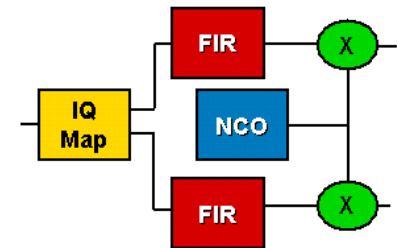
DSP Builder




Stand-alone Processor



Processor + Co-Processor



Dedicated Hardware Architecture

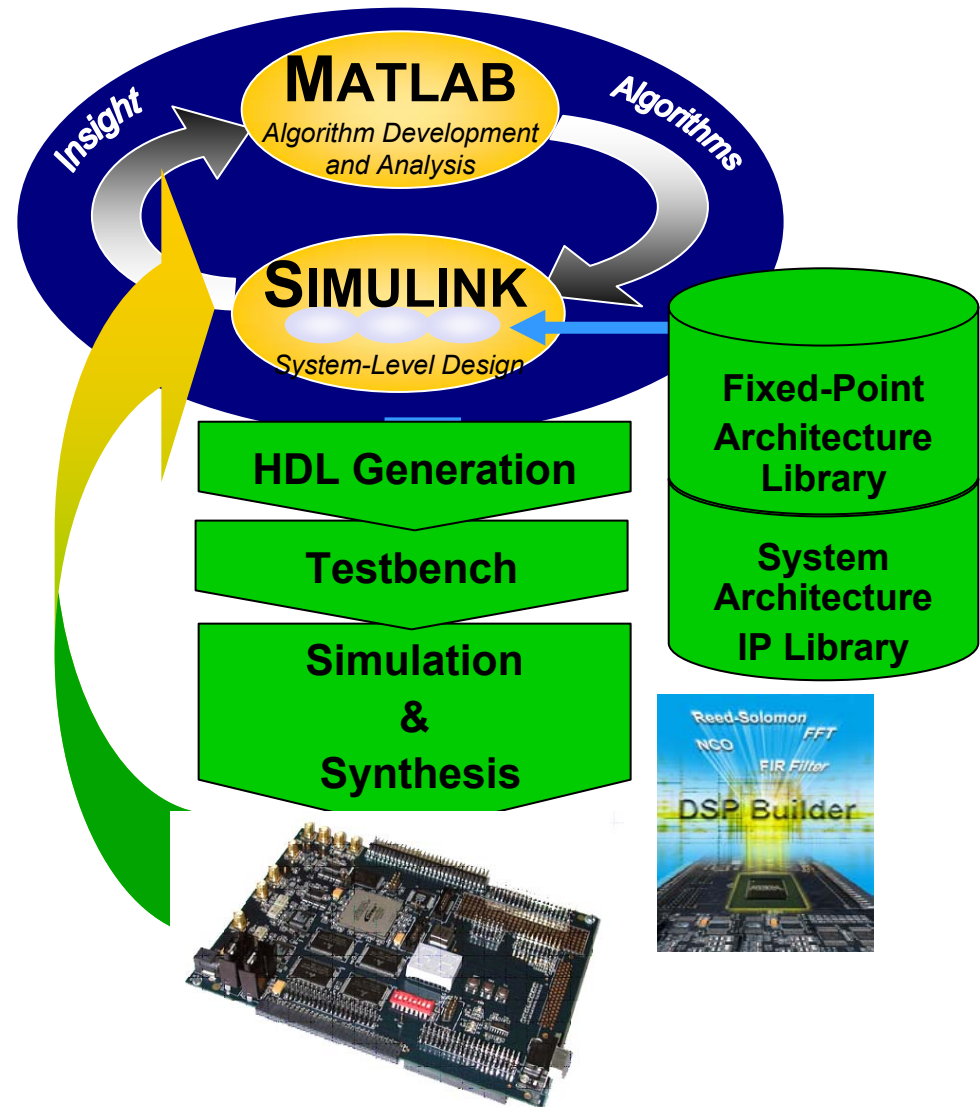


ALTERA®

DSP Builder

Altera® DSP Builder

- Fixed-Point Hardware Architecture Library
- System Architecture IP Library
- Automatic HDL Generation
- Automatic Testbench Generation
- Integrated Link to Altera Development Boards
- Real-Time Link from Development Board Back to Simulink
- Link to SOPC Builder and Nios Custom Instruction



Model Technology
A MENTOR GRAPHICS COMPANY

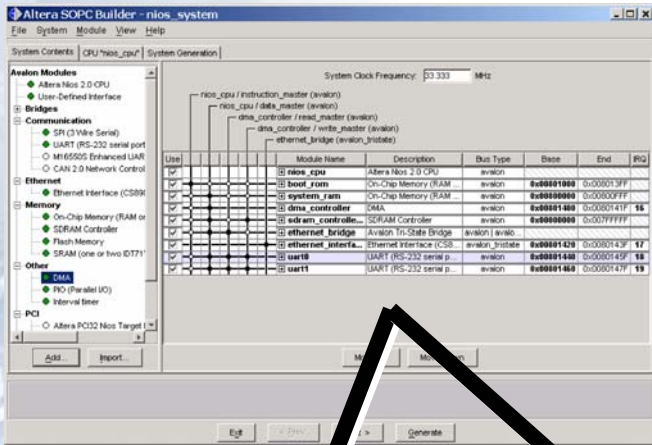

QUARTUS® II


Synplicity

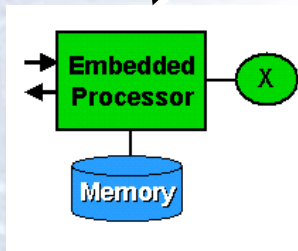
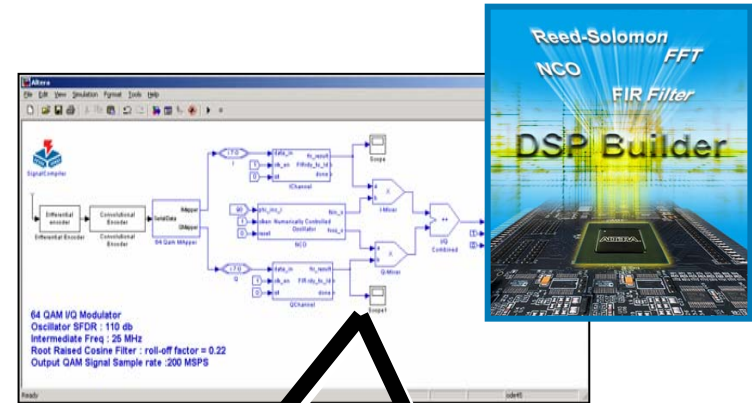

exemplar
A MENTOR GRAPHICS COMPANY

FPGA Co-Processor Design Tools

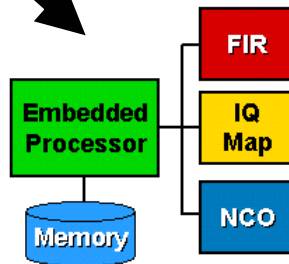
SOPC Builder



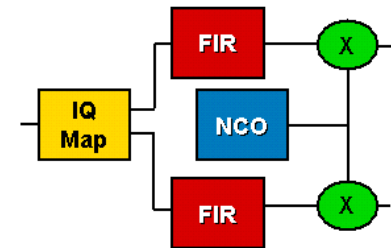
DSP Builder



Stand-alone Processor



Processor + Co-Processor



Dedicated Hardware Architecture

Datapath/Co-processor Design using DSP Builder

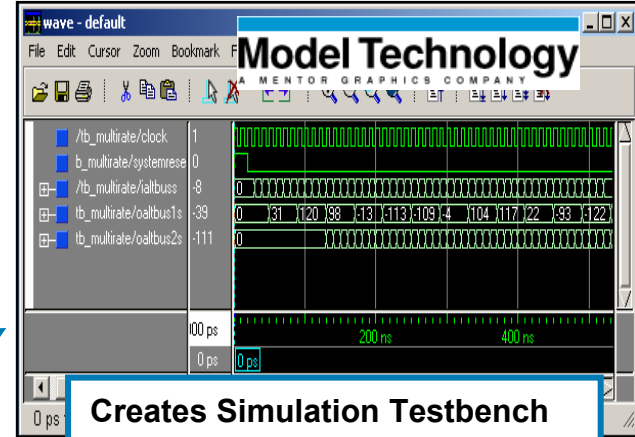
```

25
26 library ieee;
27 use ieee.std_logic_1164.all;
28 use ieee.std_logic_arith.all;
29
30 library altlink;
31 use altlink.AltFpga.all;
32
33 library lpm;
34 use lpm.lpm_components.all;
35
36 Entity multirate is
37 Port(
38   clock      :in std_logic;
39   sclr       :in std_logic='0';
40   iAltBuss   :in std_logic_vector(7 downto 0);
41   oAltBuss1  :out std_logic_vector(9 downto 0);
42   oAltBuss2s :out std_logic_vector(7 downto 0);
43 end multirate;
44
45 Architecture a of multirate is
46
47 signal SAAltBus10 : std_logic_vector(9 downto 0);
48 signal SAAltBus20 : std_logic_vector(7 downto 0);
49 signal A0W : std_logic_vector(7 downto 0);
50 signal A1W : std_logic_vector(7 downto 0);
51 signal A2W : std_logic_vector(7 downto 0);
52 signal A3W : std_logic_vector(7 downto 0);
53 signal A4W : std_logic_vector(7 downto 0);
54 signal A5W : std_logic;
55 signal A6W : std_logic;
56 signal A7W : std_logic;
57 signal sclr_u9 : std_logic;
58
59 Begin
60
61
62

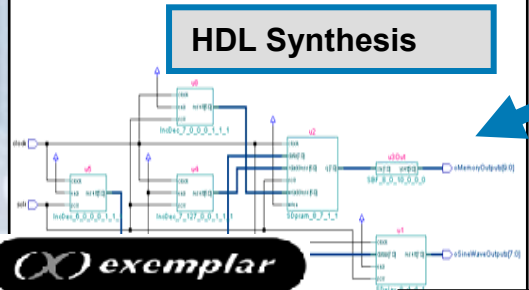
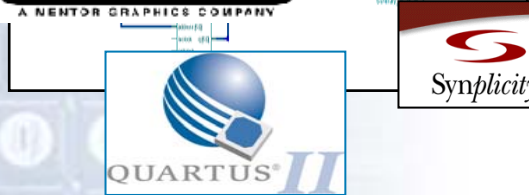
```

Creates HDL Code

Place and Route

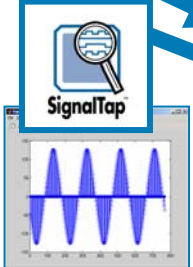




HDL Synthesis

SOPC Builder

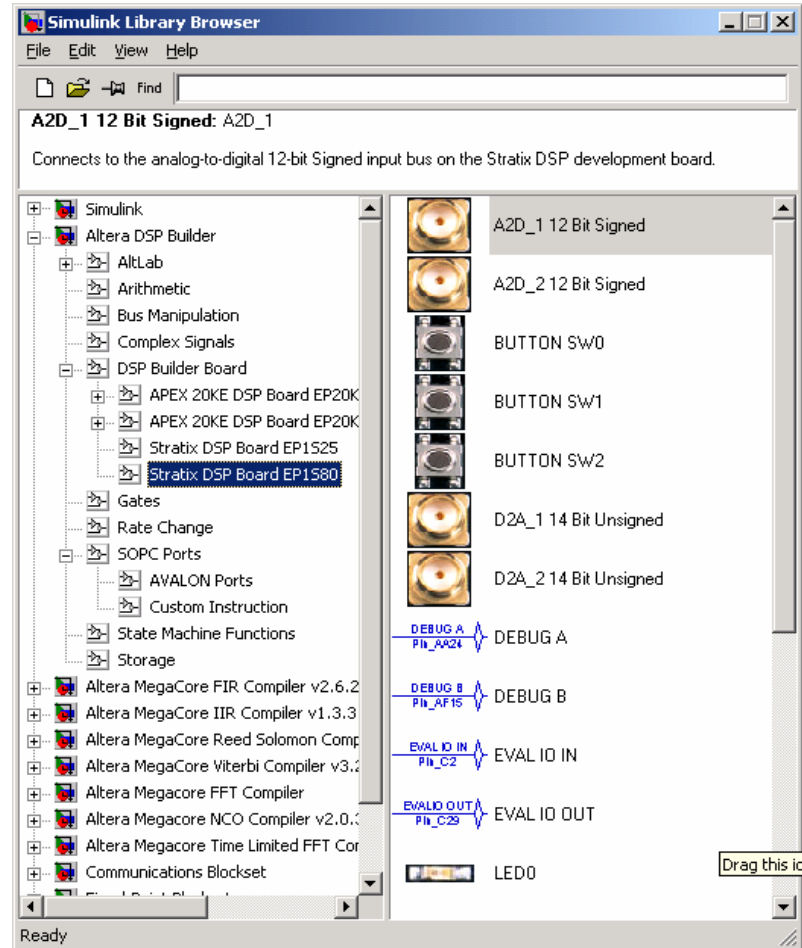
Integrates Co-processor into System

Download Design to DSP Development Kits

Library Components

- Altera Library
- Arithmetic
- Bus Manipulation
- Complex Signals
- DSP Board
- Logical Components
- MegaCore IP
- Rate Change
- SOPC Ports
- State Machine
- Storage



IP MegaCore Functions

Simulink Library Browser

File Edit View Help

Find

NCO: Numerically Controlled Oscillator

Implements the NCO Compiler MegaCore function. You can set the parameters of the function via the wizard interface.

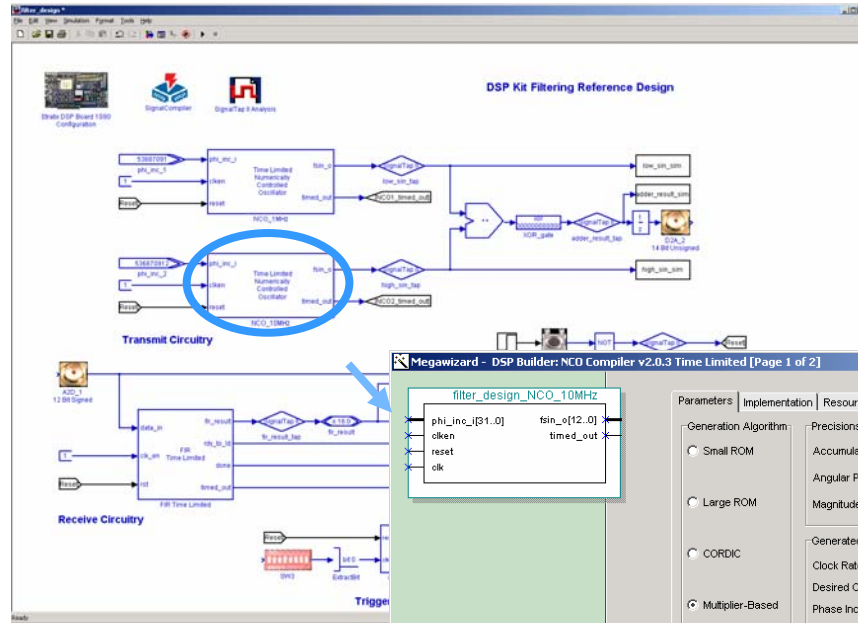
To configure the NCO Compiler:

- 1 - The system must contain the "Signal Compiler" block with the correct Working directory setting.
- 2 - Double click on the NCO Compiler block to invoke the MegaWizard.

- Simulink
- Altera DSP Builder
- Altera MegaCore FIR Compiler v2.6.2
- Altera MegaCore IIR Compiler v1.3.3
- Altera MegaCore Reed Solomon Comp
- Altera MegaCore Viterbi Compiler v3.0
- Altera Megacore FFT Compiler
- Altera Megacore NCO Compiler v2.0.3
- Altera Megacore Time Limited FFT Co
- Communications Blockset
- S-function demos

NCO

NCO Time Limited



Megawizard - DSP Builder: NCO Compiler v2.0.3 Time Limited [Page 1 of 2]

filter_design_NCO_10MHz

phi_inc_[31.0] fsin_o_[12.0]
clock reset timed_out
clk

Parameters Implementation Resource Estimate Simulation Output

Generation Algorithm: Small ROM Large ROM CORDIC Multiplier-Based

Precisions: Accumulator Precision [32] Angular Precision [12] Magnitude Precision [13]

Phase Dithering: Implement Phase Dithering Dither Level [Min Max]

Generated Output Frequency Parameters: Clock Rate [80] MHz Desired Output Frequency [10] MHz Phase Increment Value [536870912]

Frequency Domain Response Time Domain Response

Magnitude(dB)

Frequency x10⁷ Hz

Cancel < Prev Next > Finish

DSP Builder MegaCores

- FIR
- Reed Solomon
- Viterbi
- FFT
- NCO

FIR MegaCore Function

FIR Compiler 2.3.1 : Coefficient Analysis

97 Coefficient(s) in this set

No Scaling

Auto Scaling Scale By Power of 2

Manual Scaling

Signed Binary Fractional

Bits Left of Decimal Point: 0

Bits Right of Decimal Point: 8

Positive Symmetry

Coefficient Bit Width: 10

Scaling Factor: 10

Coefficient Values

32:	0
33:	-24
34:	-38
35:	-29
36:	0
37:	37
38:	58
39:	47
40:	0

RECREATE THIS Coefficient Set

RELOAD THIS Coef. Set from File

Current Coef. Set: 1

Make NEW Coefficient Set

DELETE THIS Coefficient Set

Load NEW Coef. Set from File

Single Rate

Factor: 2

dB

Frequency

Blue: Freq. Response of Fixed Point Coefficients

Green: Freq. Response of Floating Point Coefficients

< Back Cancel Finish Next >

NCO MegaCore Function

Megawizard - DSP Builder: NCO Compiler v2.0.3 Time Limited [Page 1 of 2]

filter_design_NCO_10MHz

- phi_inc_i[31..0]
- fsin_o[12..0]
- clken
- timed_out
- reset
- clk

Parameters | Implementation | Resource Estimate | Simulation Output

Generation Algorithm:

- Small ROM
- Large ROM
- CORDIC
- Multiplier-Based

Precisions:

- Accumulator Precision: 32
- Angular Precision: 12
- Magnitude Precision: 13

Phase Dithering:

- Implement Phase Dithering
- Dither Level: Min to Max

Generated Output Frequency Parameters:

- Clock Rate: 80 MHz
- Desired Output Frequency: 10 MHz
- Phase Increment Value: 536870912

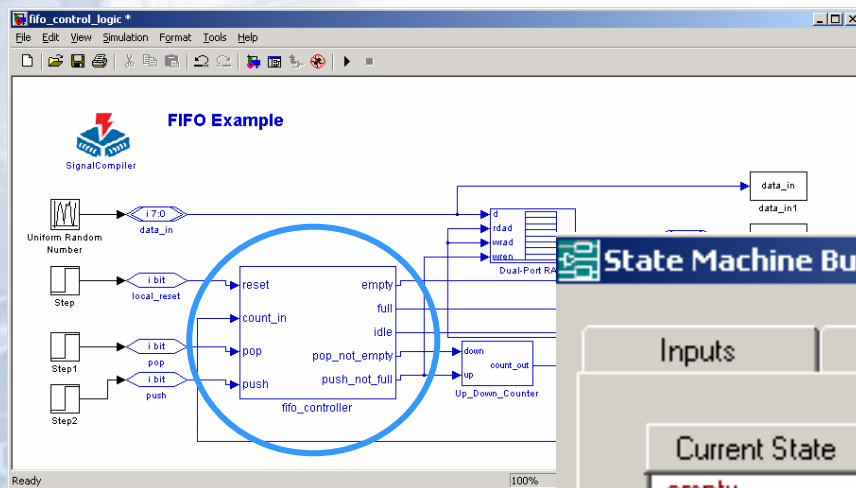
Frequency Domain Response | Time Domain Response

Magnitude (dB)

Frequency $\times 10^7$ Hz

Cancel < Prev Next > Finish

State Machine Builder



State Machine Builder v2.1.0

Inputs States Conditional Statements Design Rule Check

Current State	Condition	Next State	
empty	(push=1)&(count_in!=250)	push_not_full	▲
empty	(push=0)&(pop=0)	idle	
full	(push=0)&(pop=0)	idle	
full	(pop=1)	pop_not_emp	
idle	(pop=1)&(count_in=0)	empty	
idle	push=1	push_not_full	
idle	(pop=1)&(count_in!=0)	pop_not_emp	
idle	(push=1)&(count_in=250)	full	▼

SignalCompiler

- Generates VHDL Design Files
- Generates Tool Command Language (TCL) Scripts
- Generates Testbench
- Enables Parameterization of IP Blocks
- Launch Hardware Compilation from the Simulink Cockpit

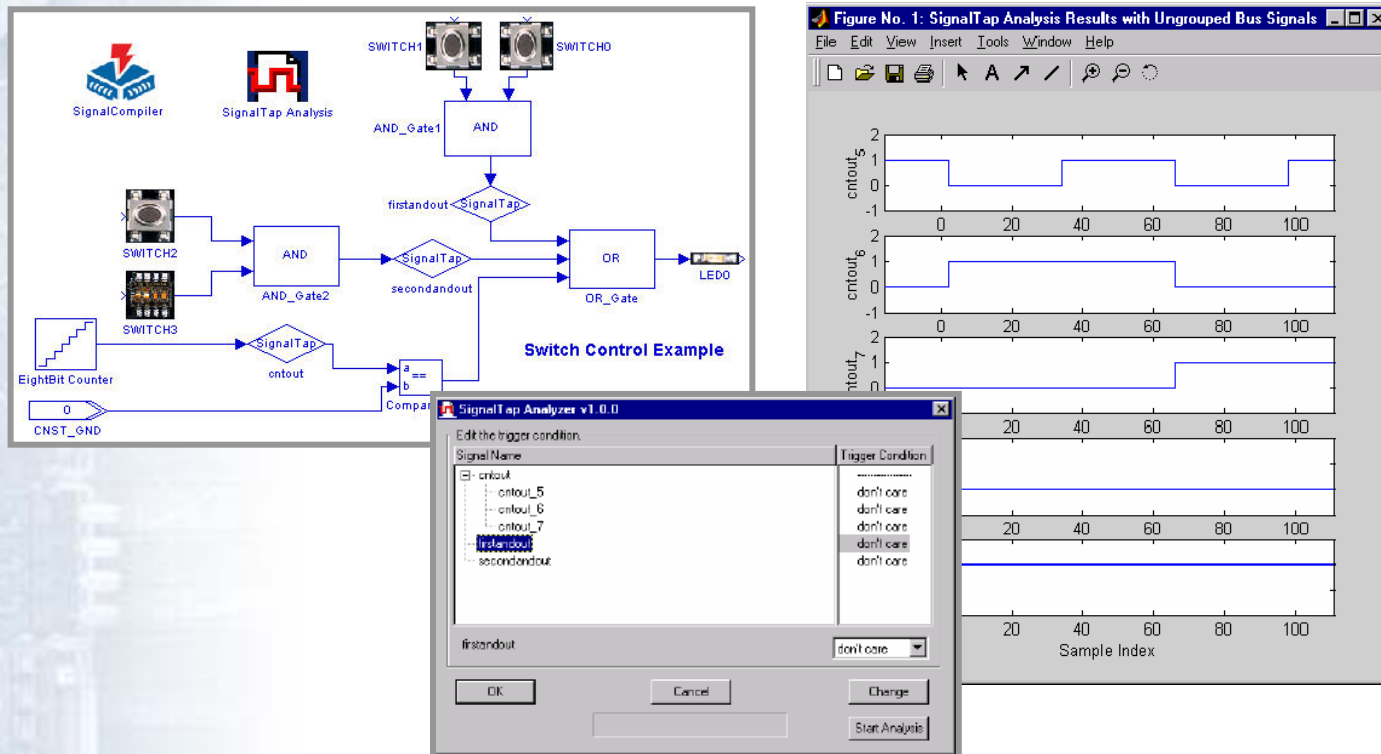
The screenshot displays the Signal Compiler Version 2.1.2 interface. The main window is titled "filter_design.mdl" and shows the following settings:

- Project Setting Options:
 - Device: DSP Board
 - Synthesis Tool: Quartus II
 - Optimization: Speed
 - Reset: SignalTap II | Testbench | SDF
 - Connect to Ground:
 - Global Reset: Active High
- Hardware Compilation:
 - 1 - Convert MDL to VHDL
 - 2 - Synthesis
 - 3 - Quartus II Fitter
 - 4 - Program DSP Board
- Messages:
 - > Resource utilization by entity filter_design :
 - > 4443 Logic Cells 4203 Registers 21 Pins
 - > 4992 Memory Bits 8 DSP Elements
 - > Analysis and synthesis completed

The interface also shows a Simulink block diagram with various components like "Transmit Circuitry", "Receive Circuitry", "Reset Circuitry", "Time Out Control Circuitry", and "Trigger Control Circuitry". A red arrow points from the "SignalCompiler" icon in the Simulink cockpit to the Signal Compiler window.

SignalTap II Logic Analyzer

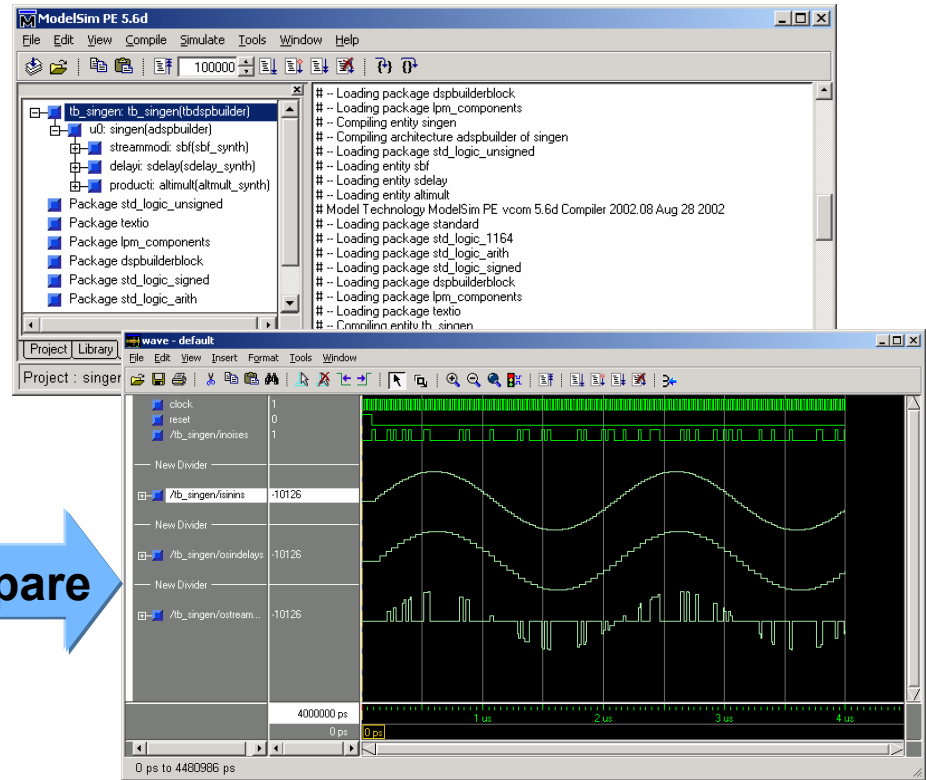
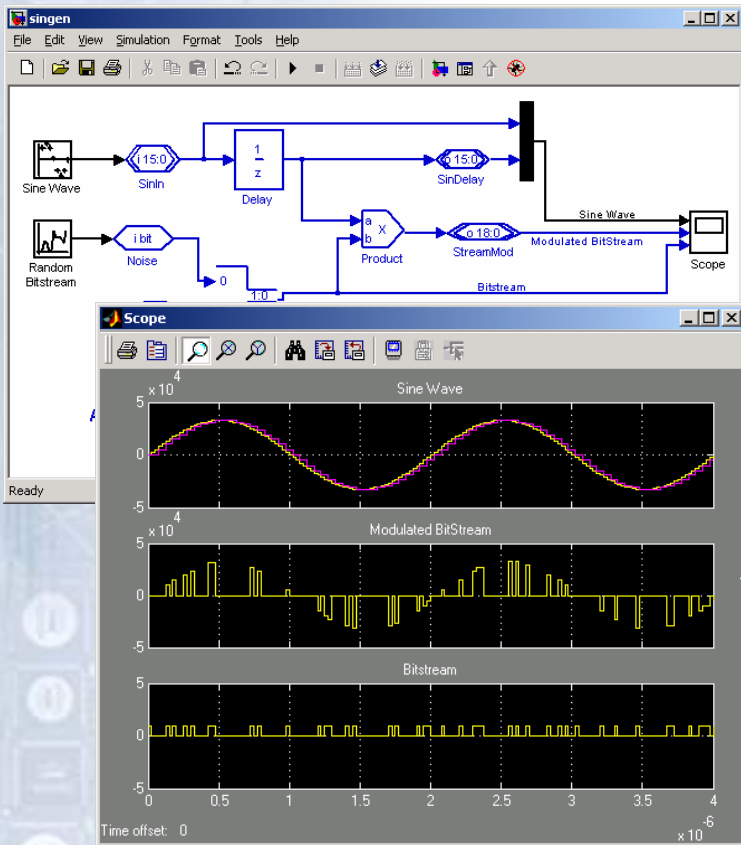
- Built-In SignalTap II Interface to DSP Builder
- Captures Signal Activity from Internal Device Nodes
- Displays Captured Data in MATLAB/Simulink



Automated Link to RTL Simulator

SIMULINK Domain

ModelSim Domain



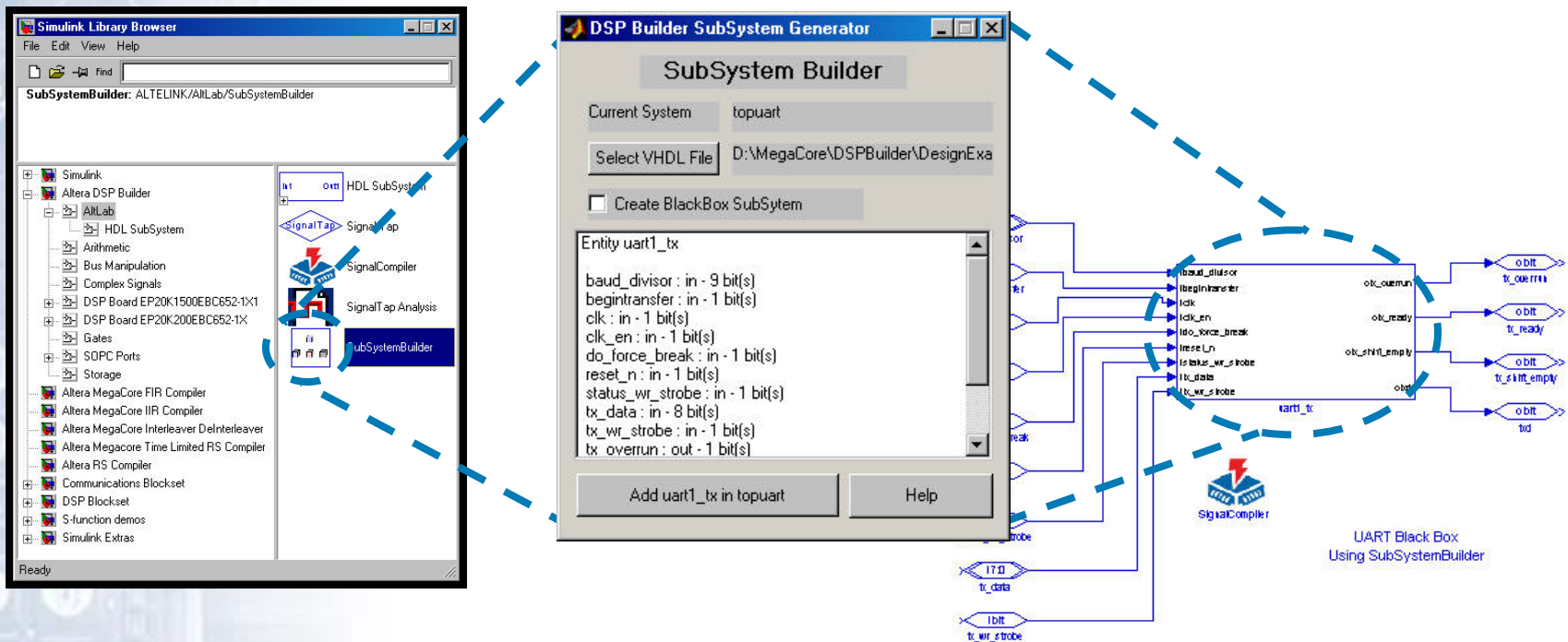
Compare



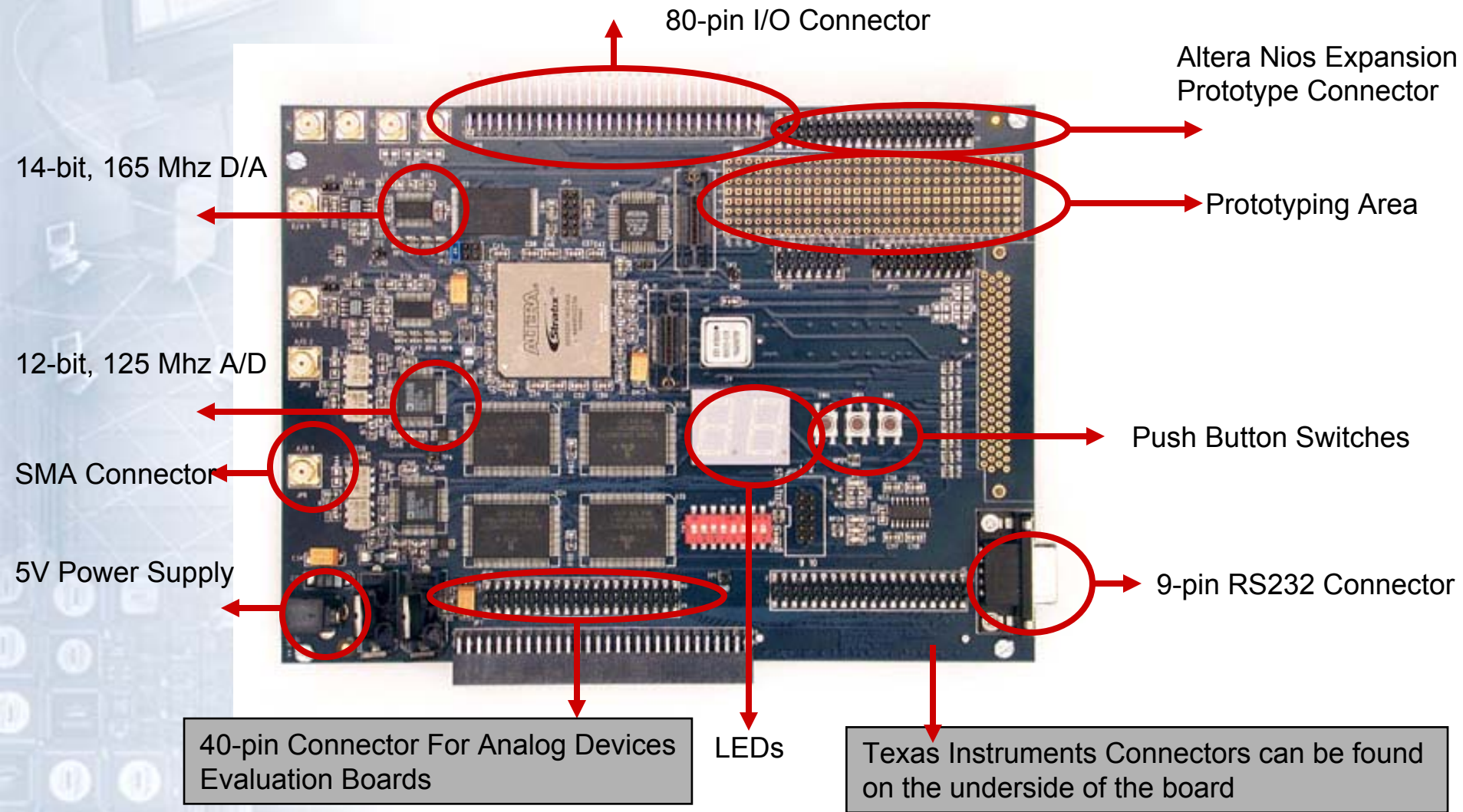
SignalCompiler

Sub-System Builder

- Import Existing VHDL Design into Simulink
- Simulink Simulation Options
 - Convert into DSP Builder Blocks or MATLAB Functions
 - Treat VHDL Design as Black Box



Stratix 1S25/1S80 DSP Development Board





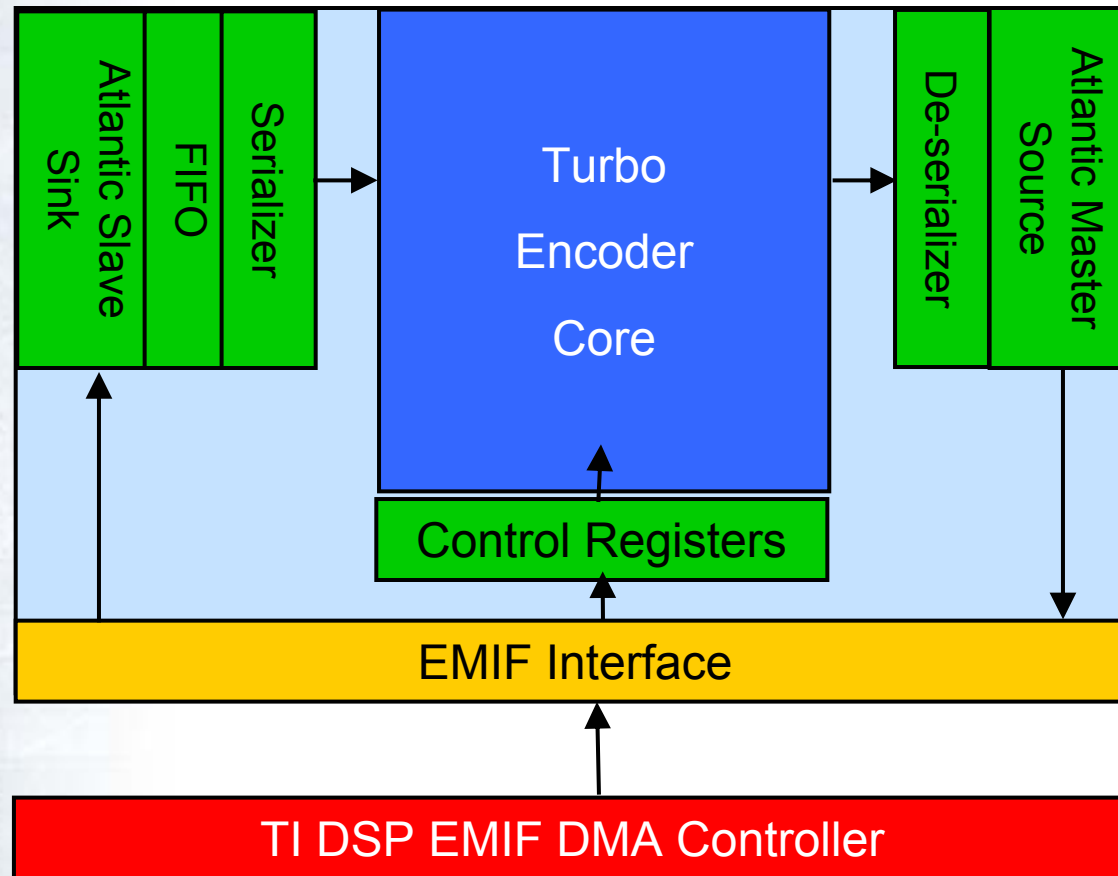
FPGA Co-processor Reference Design

Turbo Encoder for HSDPA

Turbo Encoder Co-processor Reference Design (TI Solution)

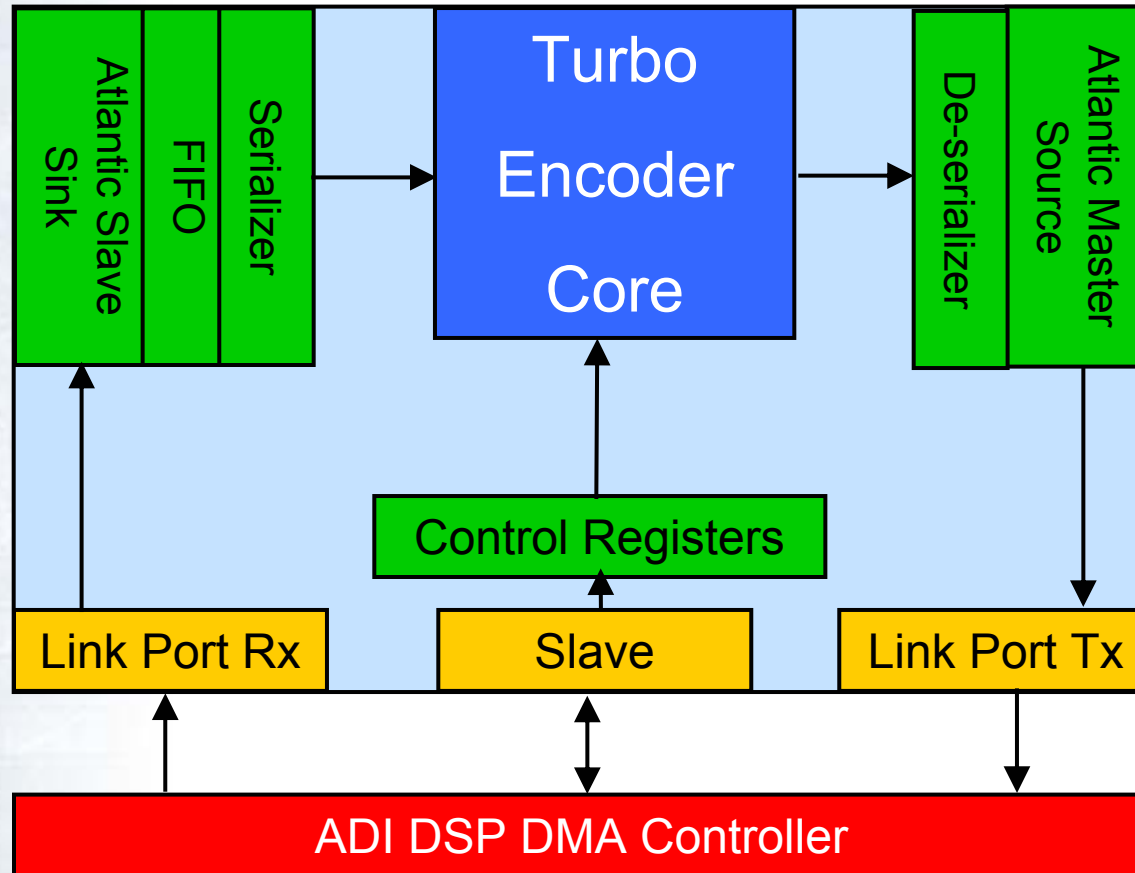
- Interface to TI DSP EMIF (External Memory Interface)
 - Other processor interfaces in development
- Uses DSP Processor's on-chip DMA
- Wrapper for Turbo Encoder MegaCore
 - Can select a different configuration (e.g. block size) for each data packet/block
- Software Libraries

Turbo Co-processor Block Diagram (TI Solution)



 Turbo Core Wrapper

Turbo Co-processor Block Diagram (ADI Solution)



 Turbo Core Wrapper

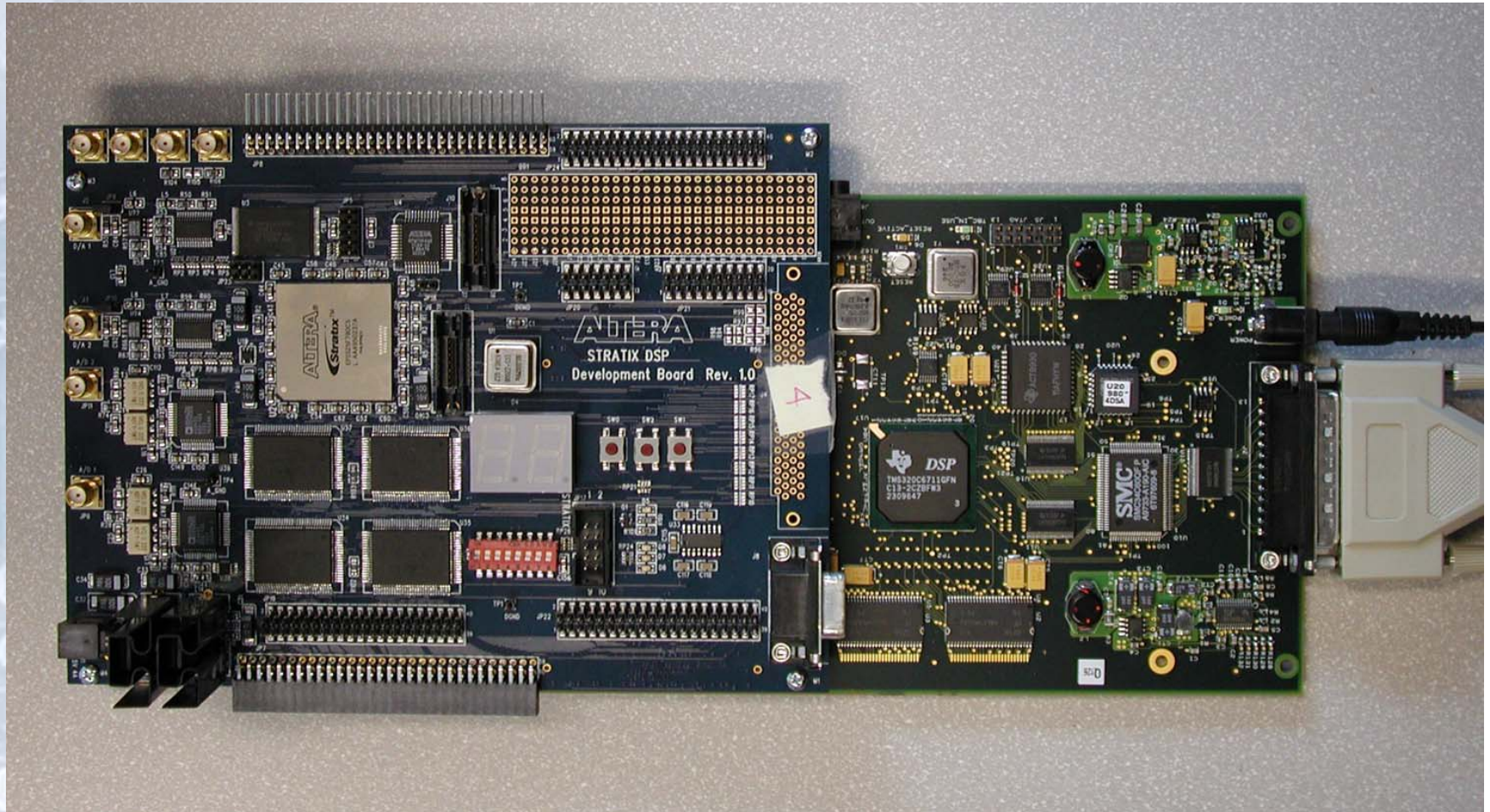
Data Buffering

- FIFOs in Atlantic slave sink store EMIF 64 byte bursts whilst being serialised into core
- Slave sink in receive converter has similar FIFO
- FIFOs introduce latency but
 - throughput is maintained
 - System bus not tied up during serialization/deserialization

TI Development Environment

- Stratix DSP Development Board plugs into EMIF slots of TI 6713 DSK (DSP Starter Kit)
- Limited to Asynchronous EMIF interface due to design of TI card
 - Customer designs may use synchronous Interface

TI Development Hardware



Software Environment

- Running on TI DSP
- Uses EDMA controller to transfer data
 - To and from accelerator
- Asynchronous, streaming, interface
 - Callback when packet is complete
 - Can submit 2nd packet before 1st is done
 - Callbacks occur in order

Header file

- `typedef void (* TXCALLBACK) (void * handle);`
- `typedef void (* RXCALLBACK) (uchar * data, void * handle);`
- `void turbo_encode(const uchar * data, uchar * output, uint bits, TXCALLBACK txcallback, RXCALLBACK rxcallback, void * handle);`

Using the accelerator

- Call function for each packet
 - Packets can have any block size
- Data is sent to encoder
 - Queued if accelerator is already busy
- txcallback called when data has been sent
- rxcallback is called once results available
- Interface can be changed if desired

Cost Analysis Example (1)

14.4Mbits/s Turbo Encoder

Cyclone: \$10.75

- 2600 LE equivalent cost Based on EP1C3T100C8 1k units (July 2003)

TI C64 DSP: \$33.00

- 136/600 MHz based on 9.7cycles/bit (Source: TI)
- C6416/600MHz @ \$145 1kunit pricing (Source: TI)

Cost Analysis Example (2)

58Mbits/s Turbo Encoder

Cyclone: \$21.58

- 2600 LE equivalent cost Based on EP1C3T100C6 1k units (July 2003)

TI C64 DSP: \$136.00

- 563/600 MHz based on 9.7cycles/bit (Source: TI)
- C6416/600MHz @ \$145 1kunit pricing (Source: TI)

TI Reference Design Summary

■ Contents

- Turbo Encoder MegaCore (OpenCore)
- TI EMIF Interface
- Wrapper
- DSP Software Libraries

■ Expected Availability

- September 2003

ADI Reference Design Summary

- Contents

- Turbo Encoder MegaCore (OpenCore)
- ADI Link Port Interface
- Wrapper
- DSP Software Libraries

- Expected Availability

- Q1 2004

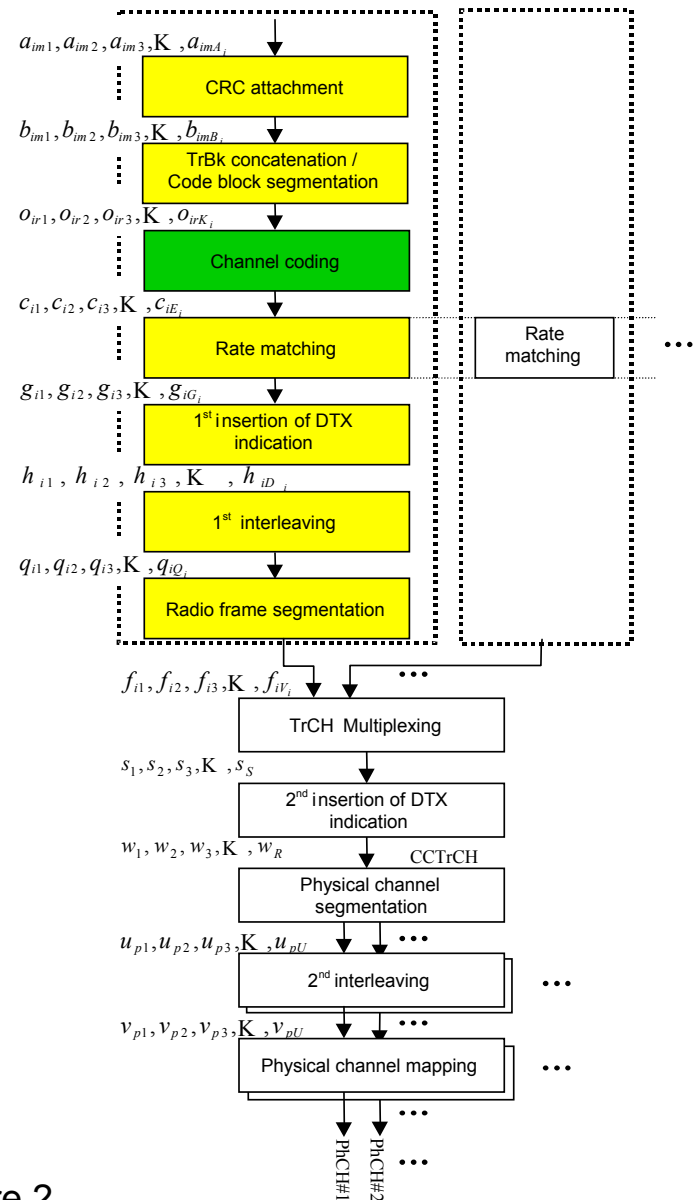
Future HSDPA Functionality

Currently implemented:

- Channel coding

Future additions:

- CRC attachment
- Rate Matching
- DTX generation
- Interleaving



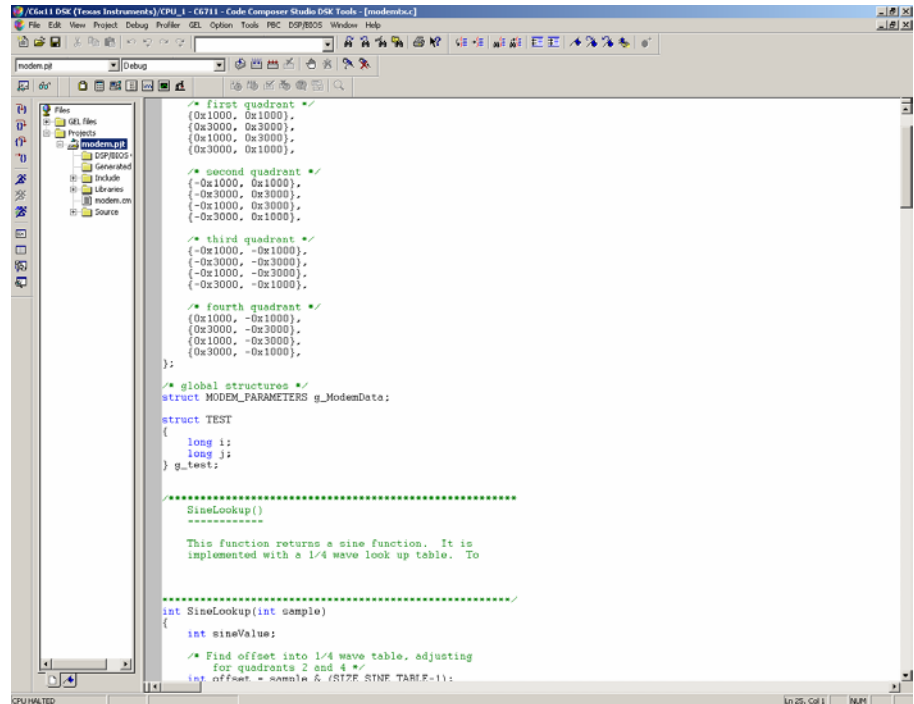
Source: 3GPP TS 25.212 V5.3.0 (2002-12), page 12, figure 2



QAM Modulator Co-Processor Design Example

Modem Reference Design

- Installed with Code Composer Studio As a Tutorial
 - C:\ti\tutorial\ds6711\modem
- Used to Demonstrate CCS Functionality
- 16 QAM TX Modem



```
/* first quadrant */
{0x1000, 0x1000},
{0x3000, 0x3000},
{0x1000, 0x3000},
{0x3000, 0x1000},

/* second quadrant */
{-0x1000, 0x1000},
{-0x3000, 0x3000},
{-0x1000, 0x3000},
{-0x3000, 0x1000},

/* third quadrant */
{-0x1000, -0x1000},
{-0x3000, -0x3000},
{-0x1000, -0x3000},
{-0x3000, -0x1000},

/* fourth quadrant */
{0x1000, -0x1000},
{0x3000, -0x3000},
{0x1000, -0x3000},
{0x3000, -0x1000},
};

/* global structures */
struct MODEM_PARAMETERS g_ModemData;

struct TEST
{
    long i;
    long j;
} g_test;

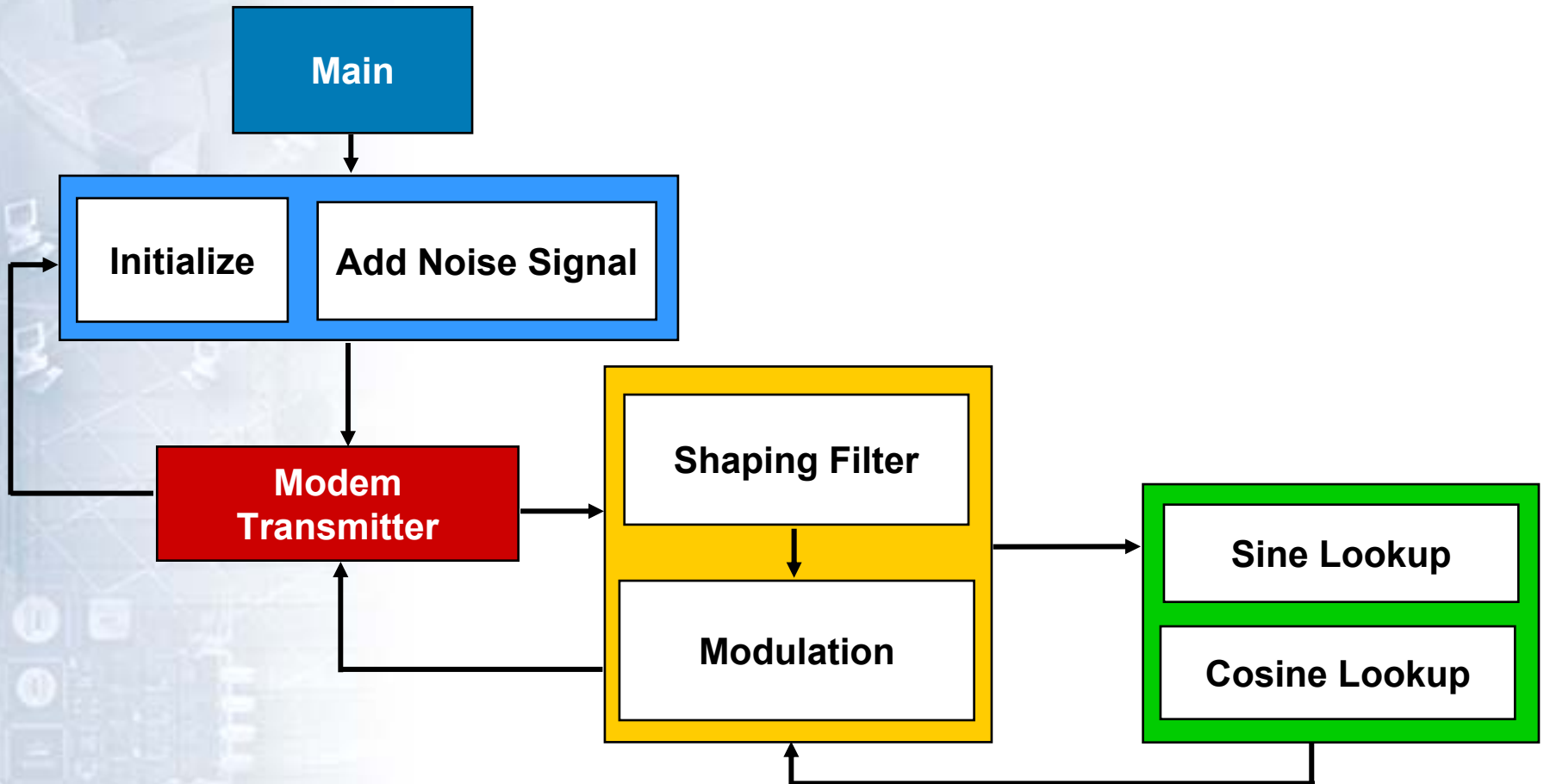
.....
SineLookup()
.....

This function returns a sine function. It is
implemented with a 1/4 wave look up table. To

.....
int SineLookup(int sample)
{
    int sineValue;

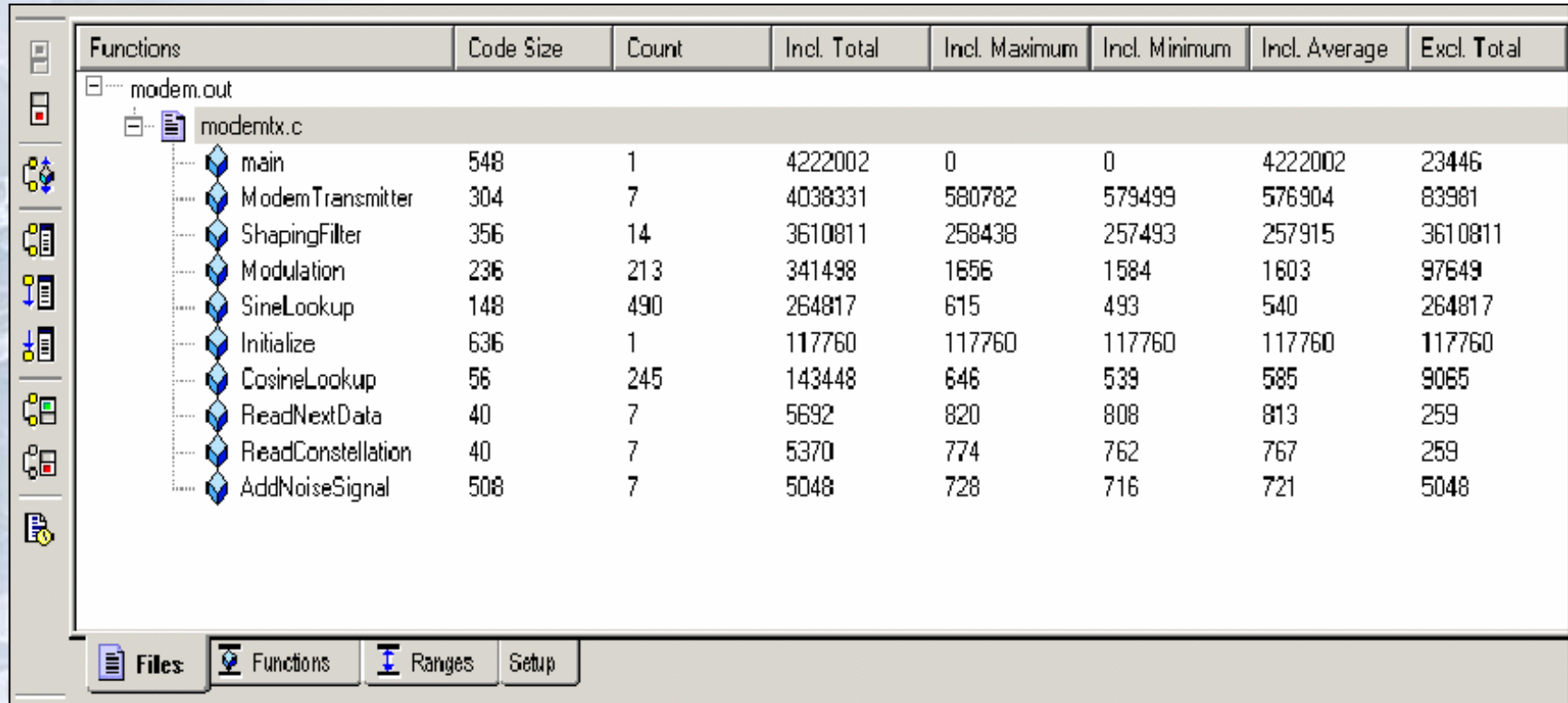
    /* Find offset into 1/4 wave table, adjusting
    for quadrants 2 and 4 */
    int offset = sample & (RTPF_SINE_TABLE-1);
```

Modem Design Overview



Modem Design Code Profile

TI Code Composer Studio Profiling Output

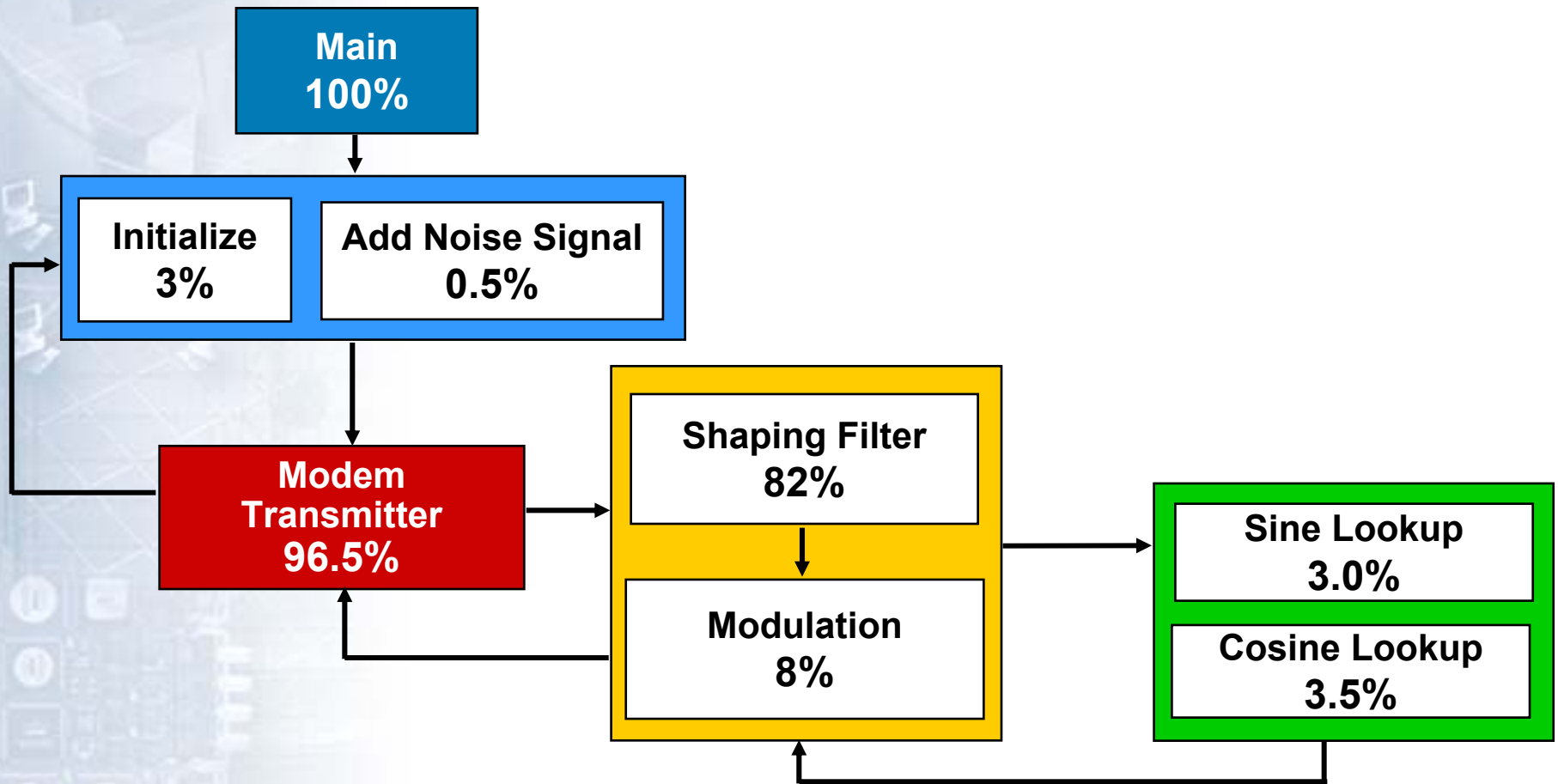


The screenshot shows the TI Code Composer Studio Profiling Output window. The window has a sidebar on the left with various icons. The main area displays a tree view of the project structure, with 'modem.out' expanded to show 'modemtx.c'. Below the tree view is a table with the following columns: Functions, Code Size, Count, Incl. Total, Incl. Maximum, Incl. Minimum, Incl. Average, and Excl. Total. The table lists the following functions and their statistics:

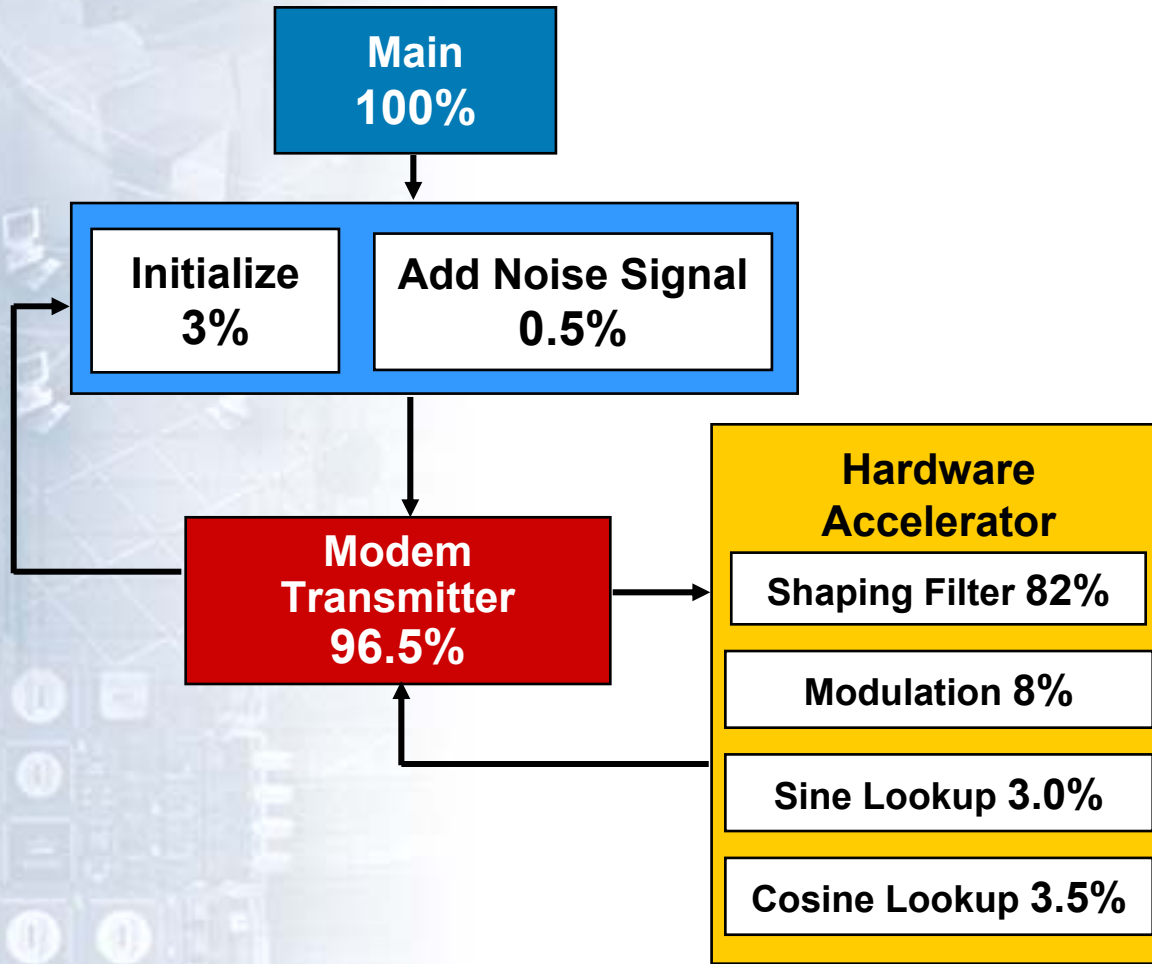
Functions	Code Size	Count	Incl. Total	Incl. Maximum	Incl. Minimum	Incl. Average	Excl. Total
modem.out							
modemtx.c							
main	548	1	4222002	0	0	4222002	23446
ModemTransmitter	304	7	4038331	580782	579499	576904	83981
ShapingFilter	356	14	3610811	258438	257493	257915	3610811
Modulation	236	213	341498	1656	1584	1603	97649
SineLookup	148	490	264817	615	493	540	264817
Initialize	636	1	117760	117760	117760	117760	117760
CosineLookup	56	245	143448	646	539	585	9065
ReadNextData	40	7	5692	820	808	813	259
ReadConstellation	40	7	5370	774	762	767	259
AddNoiseSignal	508	7	5048	728	716	721	5048

At the bottom of the window, there are four tabs: Files, Functions, Ranges, and Setup. The 'Functions' tab is currently selected.

Modem Design Code Profile

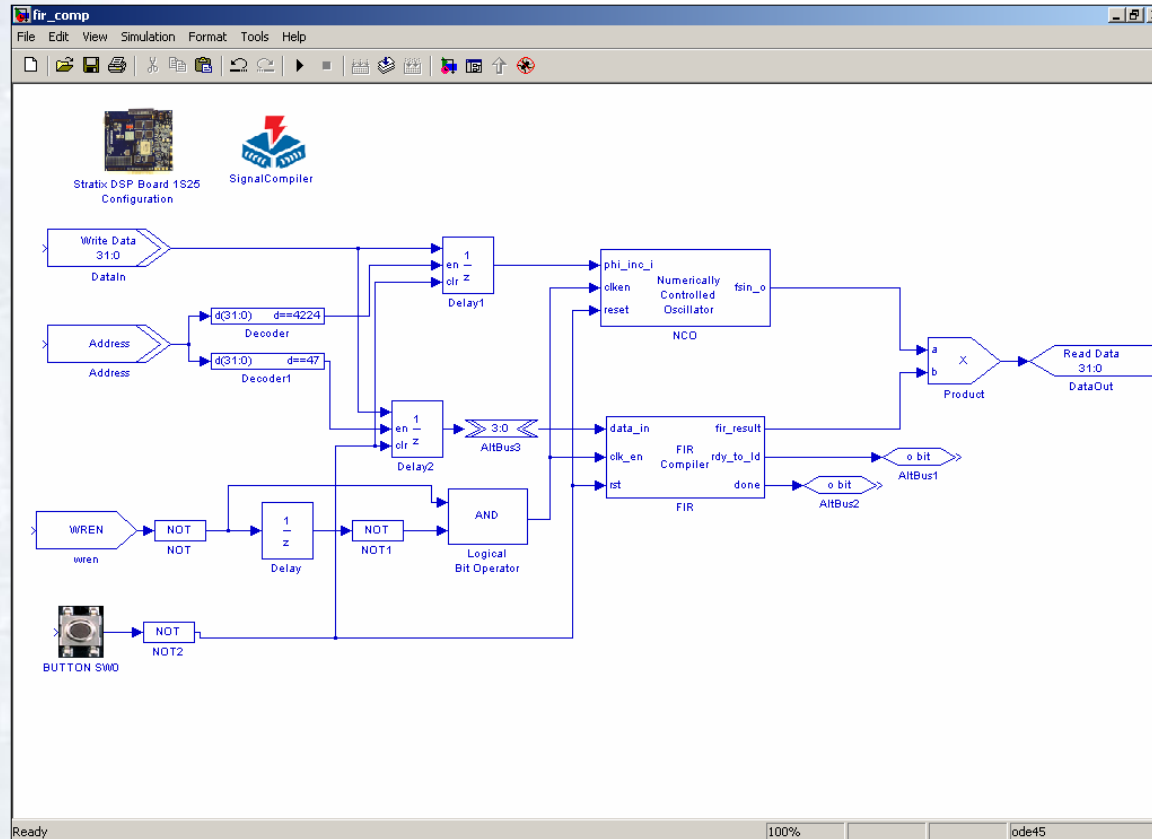


Modem Design Hardware/ Software Petition



Modulator Co-Processor

DSP Builder Used to Build Hardware DSP Data Path



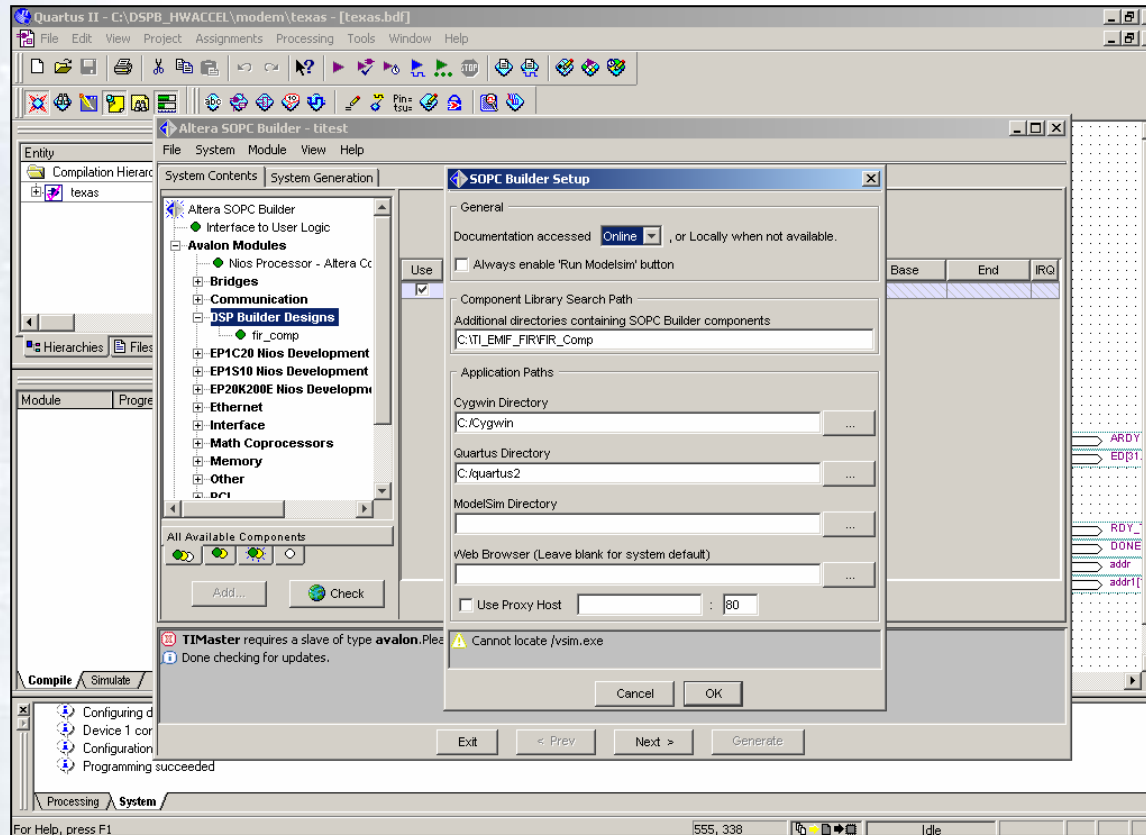
Avalon™ Interface in SOPC Builder

The image displays three overlapping windows from the Altera design suite:

- Simulink Library Browser:** Shows a tree view of Simulink blocks. The 'AVALON Ports' category is highlighted. A list of Avalon modules is shown on the right, including 'Address', 'Chip Select', 'Write Data 31.0', 'Read Data 31.0', 'Data Available', 'End Of Packet', 'IRQ', 'Ready for Data', 'WREN', and 'WRITE'.
- Altera SOPC Builder - DSP Builder:** Shows the system configuration. The 'Avalon Modules' list includes 'Altera Nios 2.0-CPU' and 'User-Defined Interface'. The system clock frequency is set to 33.333 MHz.
- fir_comp:** A block diagram showing the integration of the Simulink blocks into the SOPC architecture. The diagram includes:
 - Write Data 31.0:** A Simulink block that outputs data to the 'DataIn' input of the 'FIR' block.
 - Address:** A Simulink block that outputs an address to the 'Address' input of the 'FIR' block.
 - WREN:** A Simulink block that outputs a write enable signal to the 'wren' input of the 'FIR' block.
 - Read Data 31.0:** A Simulink block that outputs data from the 'DataOut' output of the 'FIR' block.
 - Internal Logic:** The diagram shows the internal connections between the Simulink blocks and the hardware components of the 'FIR' block, including a 'Numerically Controlled Oscillator (NCO)', a 'FIR Compiler', and various buses (AIRBus3, AIRBus1, AIRBus2).

DSP Builder— SOPC Builder Import

Import DSP Builder Generated Co-Processor into SOPC Builder



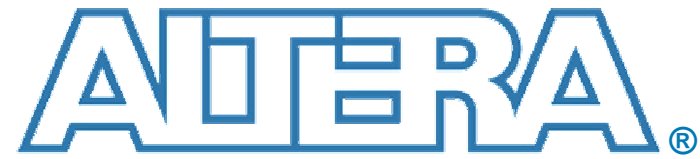
SOPC Builder Integration

Modem Co-Processor (fir_comp) Integrated with TI EMIF I/F (TIMaster)

The screenshot shows the Altera SOPC Builder interface. The title bar reads "Altera SOPC Builder - titest". The menu bar includes "File", "System", "Module", "View", and "Help". The "System Contents" tab is active, showing a tree view of the system components. The "Avalon Modules" section is expanded, showing the "fir_comp" module selected. The "System Clock Frequency" is set to 80 MHz. A table lists the modules in the system:

Use	Module Name	Description	Bus Type	Base	End	IRQ
<input checked="" type="checkbox"/>	TIMaster	Texas	avalon			
<input checked="" type="checkbox"/>	fir_comp	fir_comp	avalon	0x00000020	0x00000027	

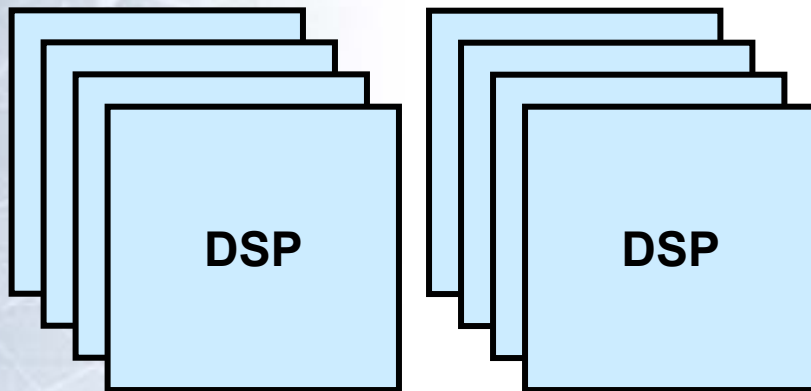
Buttons for "Move Up" and "Move Down" are visible below the table. At the bottom of the window, there are buttons for "Exit", "< Prev", "Next >", and "Generate". A status bar at the bottom indicates "Done checking for updates."



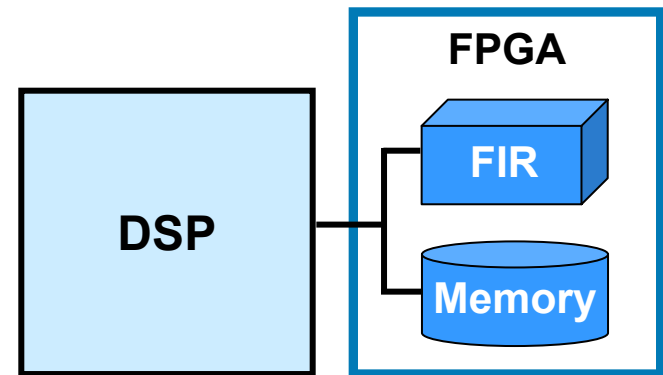
FIR Filter Co-Processor Design Example

Driving Down System Costs

Multi-Processing DSP



Digital Signal Processor + FPGA Co-Processor



FIR Co-Processor Design Example

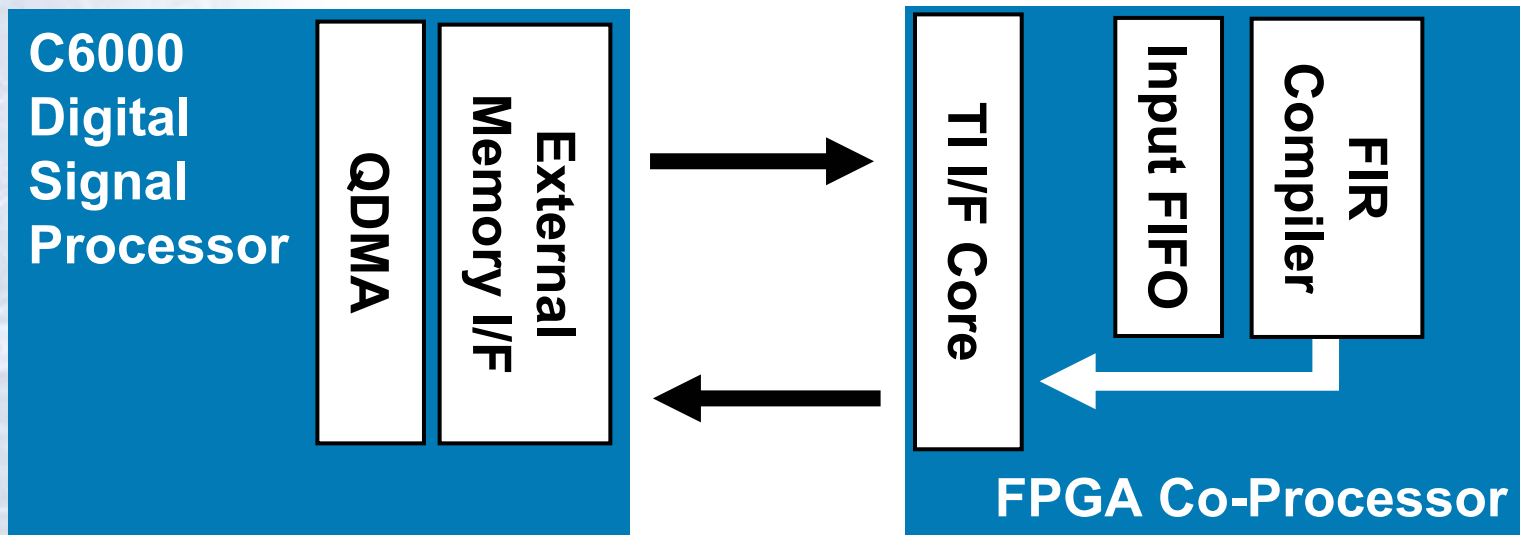
- FIR Parameters
 - 128-Tap
 - 16-Bit Data, 14-Bit Coefficients
- Four FIR Implementations for Comparison
 - TI C6711-Optimized TI DSPLib Function
 - TI C6416-Optimized TI DSPLib Function
 - Altera Eight-Cycle FIR Co-Processor
 - Altera One-Cycle FIR Co-Processor

TI Filtering Library (DSPLib)

- C-Callable Optimized Assembly Routines
- TI C67x DSPLib: FIR Filter (Radix 8)
 - Formula: $N_h * N_r / 2 + 13$
 - N_h = Number of Coefficients
 - N_r = Number of Samples
 - ~1 Sample/ 64 Cycles (128 Tap Filter)
- TI C64x DSPLib: FIR Filter (Radix 8)
 - Formula: $N_h * N_r / 4 + 17$
 - ~ 1 Sample/ 32 Cycles (128 Tap Filter)

Filter Co-Processor Design Example

- Current Implementation
 - 100 MHz, 32-Bit, Asynchronous EMIF on DSK
 - TI Writes 300 Samples to Co-Processor (Input Data)
 - Filter & Send Output to TI



FIR Filter Example* – 16X Cost/Performance Improvement

Device	Solution	FIR Performance (MHz)	Device Cost****	Cost per FIR MHz
TI C6713-200	64-Cycles** at 200 MHz	3.125	\$24.59	\$7.87
TI C6416-600	32-Cycles** at 600 MHz	18.75	\$160	\$8.53
Altera EP1C3-8	7-Cycles*** at 197 MHz	28	\$14	\$0.50
Altera EP1C12-8	1-Cycle*** at 170 MHz	170	\$84	\$0.49

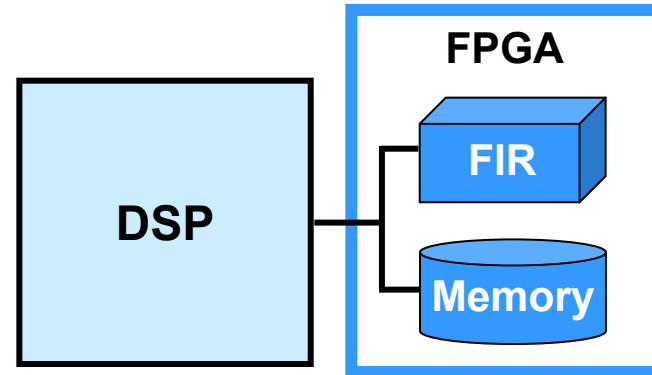
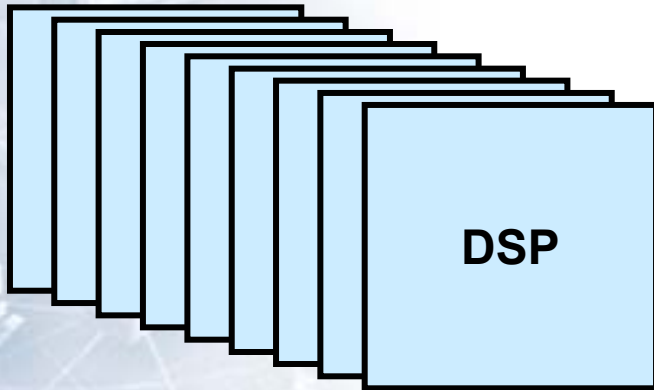
* FIR 128 Tap, 16-Bit Data, 14-Bit Coefficients

** DSPLib Optimized Assembly Libraries from Texas Instruments

*** Optimized MegaCore FIR Compiler from Altera

**** Pricing in Quantity of 100 at Arrow 6/25/03

14X Reduction in System Costs



Architecture	FIR Performance (MHz)	Total FIR Performance (MHz)	Device Costs	Total Cost
9 * TI C6416-600	9 * 18.75	167	9 * \$160	\$1,440
Altera EP1C12-8 + 1 TI C6713-200	170 + 3	173	\$84 + \$25	\$110

Summary

- FPGA Co-Processors for DSP Offer Many Advantages
 - 10X Performance Boost
 - More Channels
 - More Complex Algorithms
 - Increased System Throughput
 - 10X Cost Reduction
 - Fewer Components
 - Complementary to DSP-Based Systems
 - Offloads Existing DSP
 - Integrates Into Existing DSP IDE
 - Evolution Not Revolution

Altera Code: DSP Solutions

- FPGAs
 - Stratix, Stratix GX, Cyclone
- Development Tools
 - DSP Builder, SOPC Builder
- Intellectual Property
 - FIR, FFT, Viterbi, Turbo, Reed Solomon, NCO
 - AMPP Third-Party Partners
- Development Kits
 - Altera
 - Third-Parties
- Design Services
 - ACAP Third-Party Partners
- Training