

**ALTERA**®

# Story of DSP in Altera

*Dylan Wang*



# FPGA / DSP Challenges

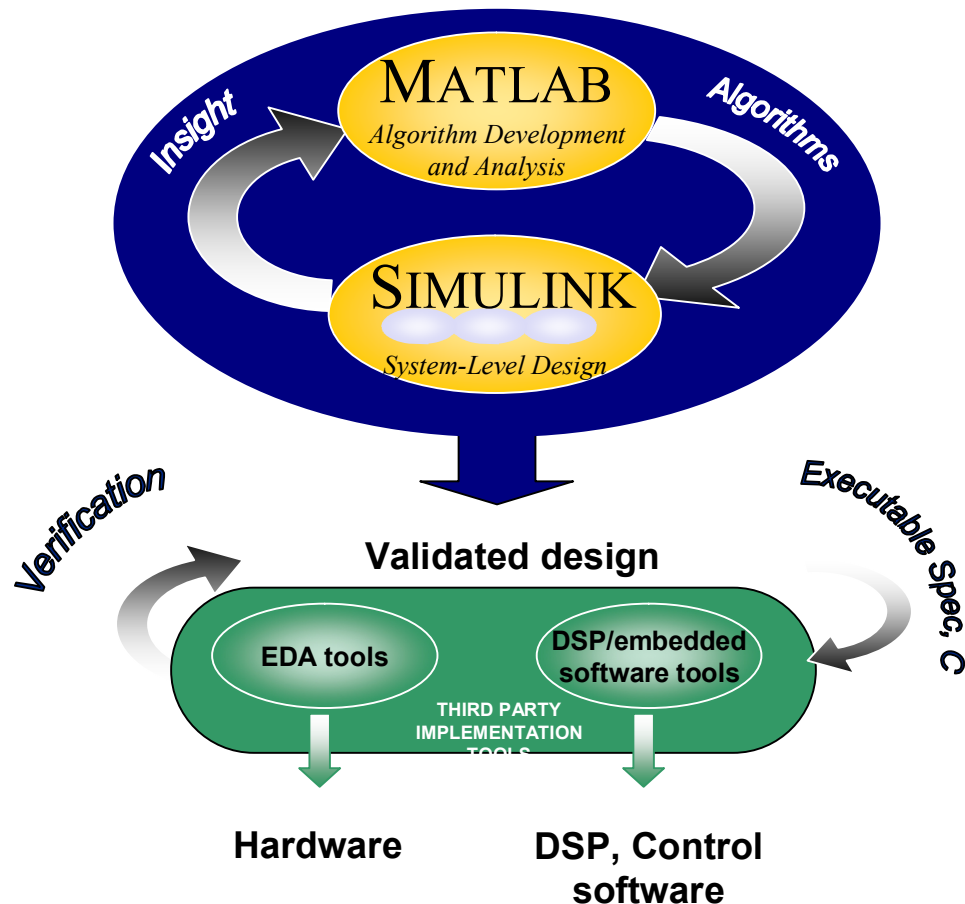
## ■ Methodology challenges

- Designing with FPGA devices is different than what DSP designers are used to
  - Different set of tools
  - C-code/model/algorithm vs. VHDL / Verilog / Schematic

## ■ Application development challenges

- System-level development & verification
- Software/hardware co-development
- Software design optimization

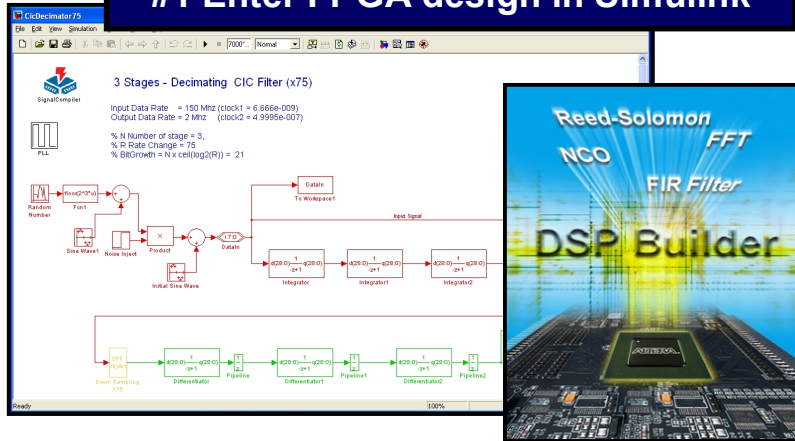
# The Mathworks Design Environment



- Top-down design
- Design & test system behavior early in the design process
  - Create validated reference design
- Detect design flaws early
- Reduced design risk & cost
- Reduced time-to-market

# DSP Builder

## #1 Enter FPGA design in Simulink



## #2 Generate RTL

```

multirate.vhd
25
26 library ieee;
27 use ieee.std_logic_1164.all;
28 use ieee.std_logic_unsigned.all;
29
30 library altera;
31 use altera.altrtl.all;
32
33 library ipm;
34 use ipm.ipm_components.all;
35
36 entity multirate is
37     port(
38         clock      : in std_logic;
39         sclk       : in std_logic='0';
40         i16tBus    : in std_logic_vector(7 downto 0);
41         o16tBus    : out std_logic_vector(9 downto 0);
42         i16tBus2   : in std_logic_vector(7 downto 0);
43     );
44 end multirate;
45
46 architecture a of multirate is
47
48     signal sA16tBus10 : std_logic_vector(9 downto 0);
49     signal sA16tBus20 : std_logic_vector(7 downto 0);
50     signal A0W : std_logic_vector(7 downto 0);
51     signal A1W : std_logic_vector(7 downto 0);
52     signal A2W : std_logic_vector(7 downto 0);
53     signal A3W : std_logic_vector(7 downto 0);
54     signal A4W : std_logic_vector(7 downto 0);
55     signal A5W : std_logic;
56     signal A6W : std_logic;
57     signal A7W : std_logic;
58     signal sclk_wv : std_logic;
59
60     Begin
61         sclk_wv <= sclk;
62
  
```

## #3 Create SOPC plug-in

## #3 Automatic FPGA compilation



- Enables FPGA design in Simulink using Altera's libraries
- DSP Builder Automate RTL Generation
- Quartus compiles the FPGA or creates SOPC Builder plug-ins
- SignalTap provides debug facilities

## #4 Program & Debug Hardware

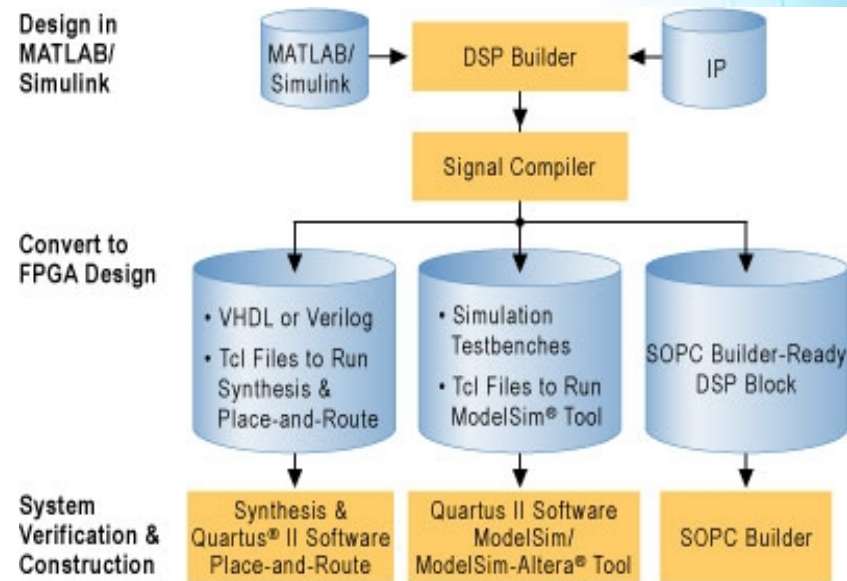


# DSP Builder Challenges

- Design looks more like the schematics
- Design the pipeline
  - Latency
  - Combinational Logic Delay
- Not so familiar with the devices
- Difficult to implement the Mult-channel design

# What is DSP Builder Advanced Blockset?

- DSPB-AB is the next generation **blockset library** for FPGA design using Mathworks Simulink
  - Built on the existing DSP Builder platform
  - Can easily integrate into existing HDL designs
  - Seamlessly integrated with Quartus II



## Timing Driven Simulink Synthesis

# Advanced Blockset - What it does..

- Automate the tedious part of FPGA implementation
  - Automates pipelining to meet required Fmax
  - Similar performance as optimized HDL
  - Easy timing closure
  - Results in fewer compile iterations
- Fast multi-channel design implementation
  - Automates control plane for TDM data paths
  - Efficiently pipelines multi-channel data paths
- Update design by editing system level parameters
- Effortless FPGA device family retargeting

# DSPB-AB Value Proposition

- Automatic timing-driven FPGA implementation from Simulink
  - Automates pipelining to meet required  $f_{MAX}$
  - Provides performance similar to optimized HDL
  - Facilitates easy timing closure resulting in fewer Quartus compiles
- *FAST* multi-channel design implementation
  - Automates control plane for time-division multiplexed data paths
  - Efficiently pipelines multi-channel data paths
- System level constraints drive design implementation
- Effortless FPGA device family retargeting

# DSPB Advanced Blockset allows

## ■ Hardware engineers to ...

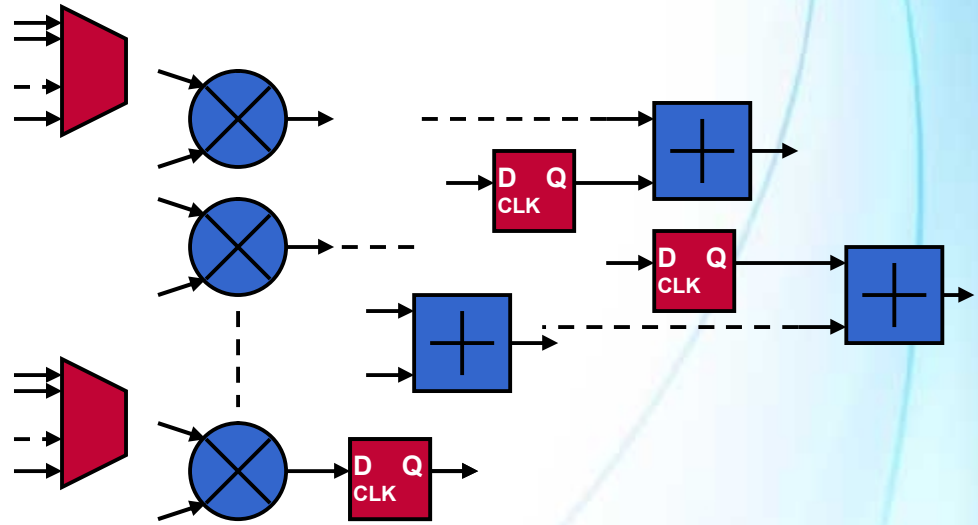
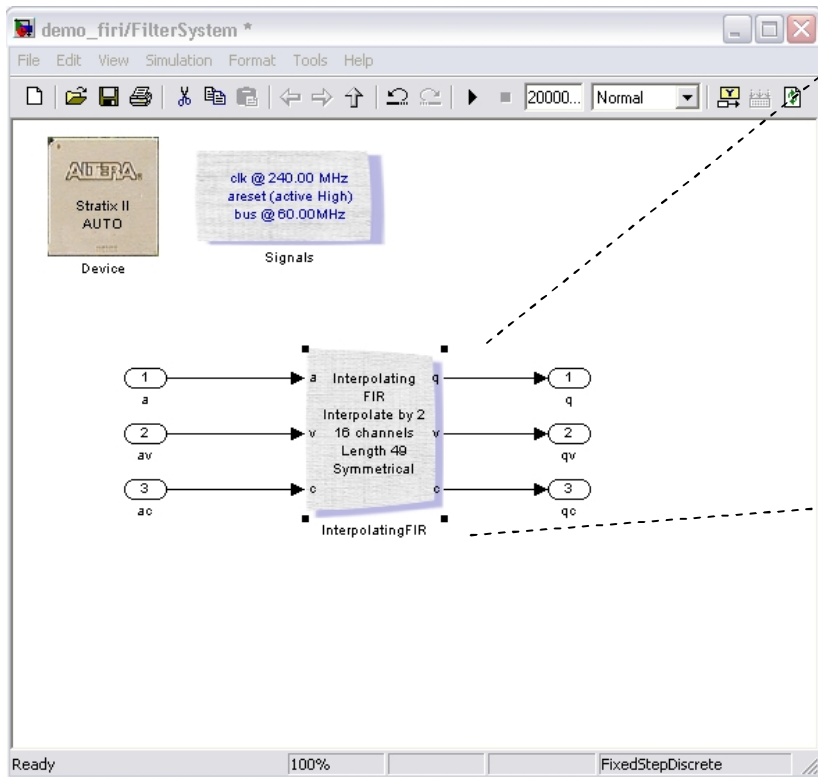
- Change the number of channels and required performance in an abstract format, eliminating the need for timing closure (auto-pipelining) and generation of any control logic
- Generate efficient RTL which in most cases outperforms hand coded HDL in terms of Fmax and development time.
- Try different devices to optimize the development

## ■ System engineers to ...

- Try the algorithm in hardware
- Create an executable spec to hardware engineers

# Automatic timing-driven FPGA implementation from Simulink

Auto pipelining  
Auto time sharing  
FAST design iterations



### DSPB Advanced Blockset based Solution:

- Removes these complexities through automation
- Focuses on optimizing key DSP design elements



# Boost Productivity

- Define a system once & thereafter just modify the system parameters

The image displays two windows from the Quartus II software. The left window, titled 'demo\_firi/FilterSystem \*', shows a block diagram of a system. It includes a 'Stratix II AUTO' device and a 'Signals' block with parameters: 'clk @ 240.00 MHz', 'areset (active High)', and 'bus @ 60.00MHz'. The main block is 'Interpolating FIR' with parameters: 'Interpolate by 2', '16 channels', 'Length 49', and 'Symmetrical'. It has three input ports labeled 'a', 'v', and 'c' (with corresponding signals 'a', 'av', and 'ac') and three output ports labeled 'q', 'qv', and 'qc' (with corresponding signals 'q', 'qv', and 'qc').

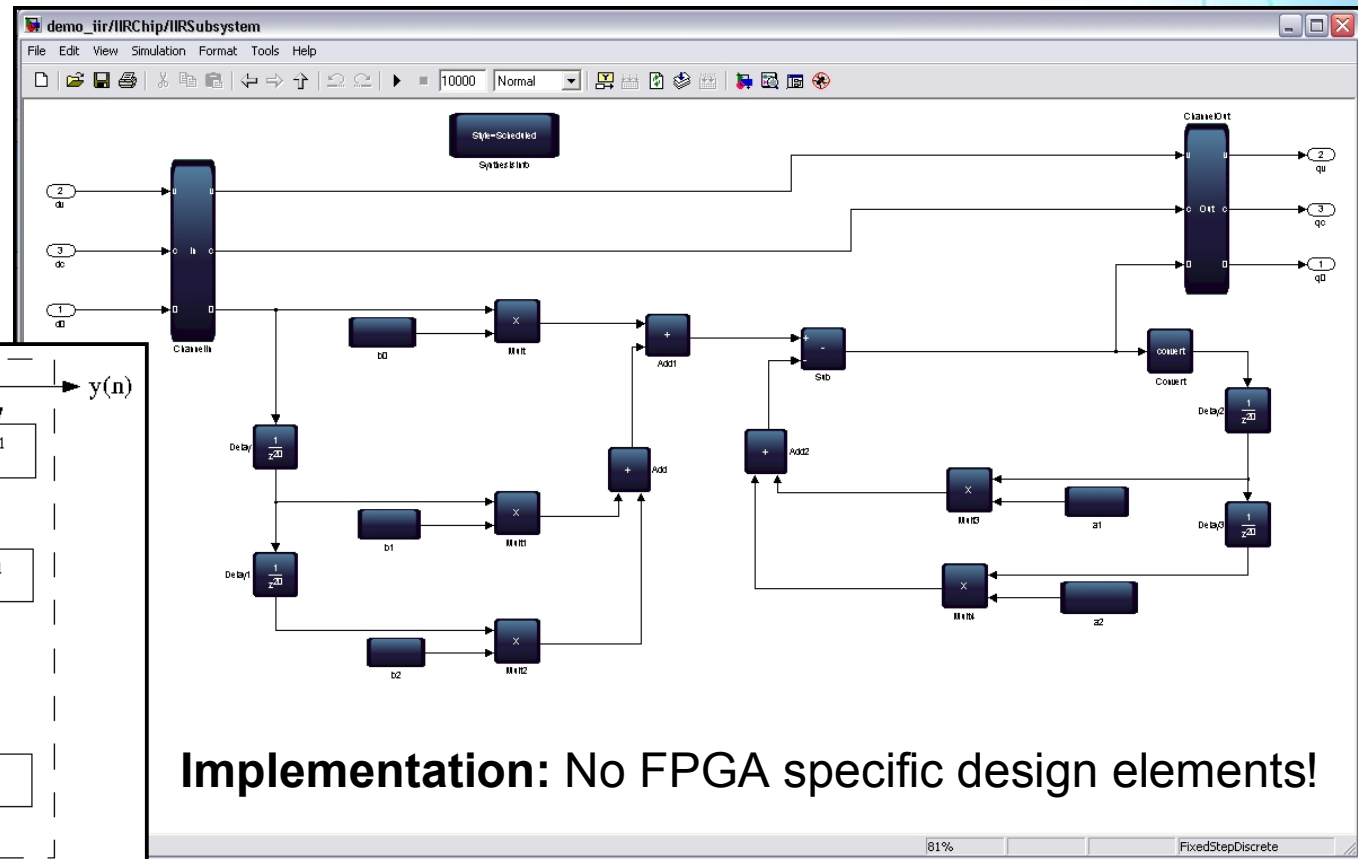
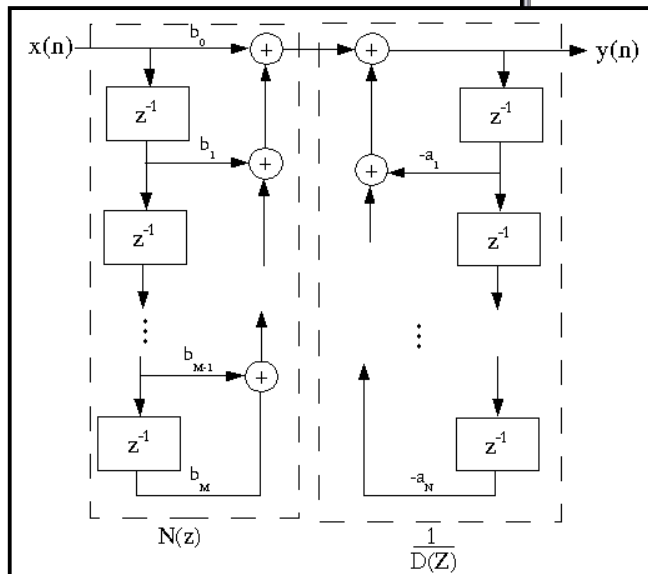
The right window, titled 'C:\AionModelIP\Examples\Filters\setup\_demo\_firi.m', shows a MATLAB script. The script defines parameters for the filter system:

```
1 % File: setup_demo_firi.m
2 % Description: Script to set variables in Matlab workspace to
3 % Revision: $Id: setup_demo_firi.m,v 1.2 2006/10/10 14:29:
4
5 ChanCount=16;
6 ClockRate=240.00;
7 ClockMargin = 0.0;
8 SampleRate=2.4;
9 Period=ClockRate / SampleRate;
10 FilterLength=49;
11 Interpolation = 2;
12 SampleTime = 1;
13 %SampleTime = 1 / (ClockRate * 1e6); % uncomment this line
14
15 disp(['Parameters set from setup_demo_firi.m: FilterLength='
16
17 % Derived Parameters
18 % WARNING - DO NOT MODIFY!!!
19 ChanWireCount=ceil(ChanCount/Period);
20 ChanCycleCount=ceil(ChanCount/ChanWireCount);
21
```

# Design Readability Benefits

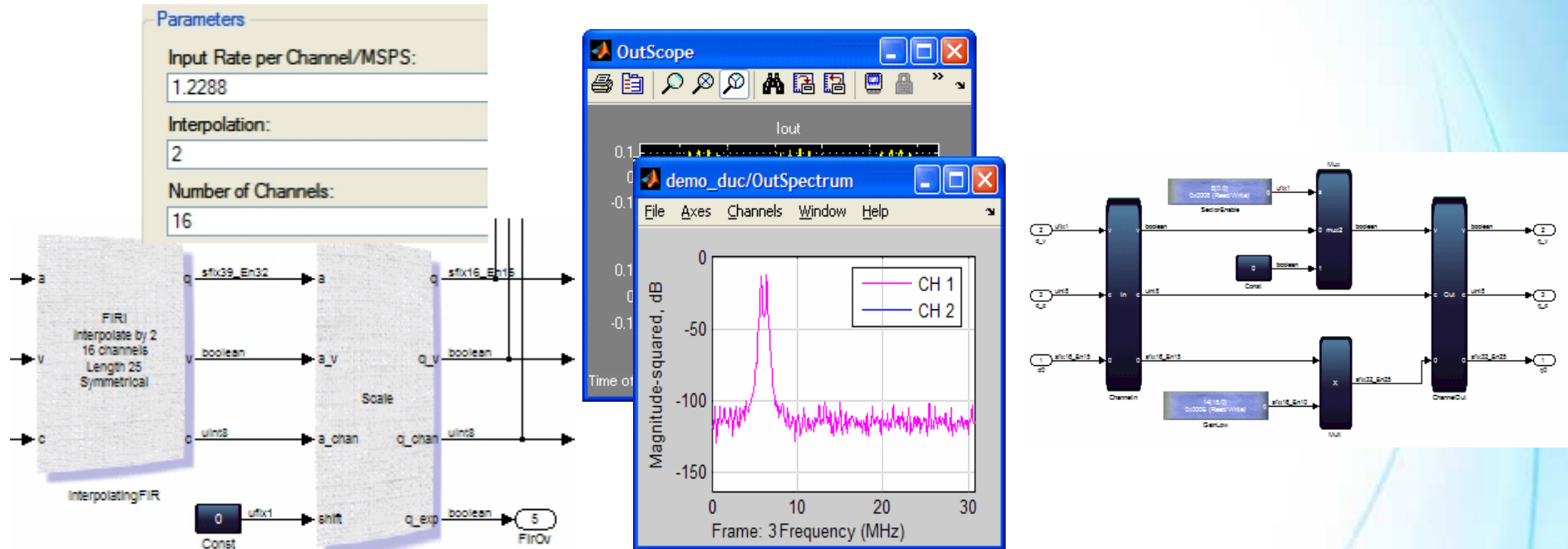
- Implement systems using 'text book' form diagrams

**Specification:** No FPGA specific design elements!



**Implementation:** No FPGA specific design elements!

# DSP Builder Advanced Blockset Advantages



## Effortless FPGA Implementation

- Automatic pipelining to meet required Fmax
  - Similar performance as optimized HDL
  - Easy timing closure
  - Fewer compile iterations

## FAST Design Space Exploration

- Fast multi-channel design implementation
- Automatic generation of control plane logic
- Efficient pipelining for multi-channel data paths
- Ability to update design by editing system level parameters
- Effortless FPGA device family retargeting