Design Techniques for RTAX-DSP in High Reliability Applications

Minh Nguyen
Corporate Applications Engineer

November 2010

Topics

- RTAX-DSP Overview
- Mathblocks Design Techniques
  - Use Actel IP CoreFIR
  - Instantiate Mathblocks Macro
  - Synplify Inference
  - Mathworks Matlab/Simulink and Synopsys Synphony Model Compiler
- Common Connectivity Errors & Mitigation
- RTAX-DSP Design FAQ

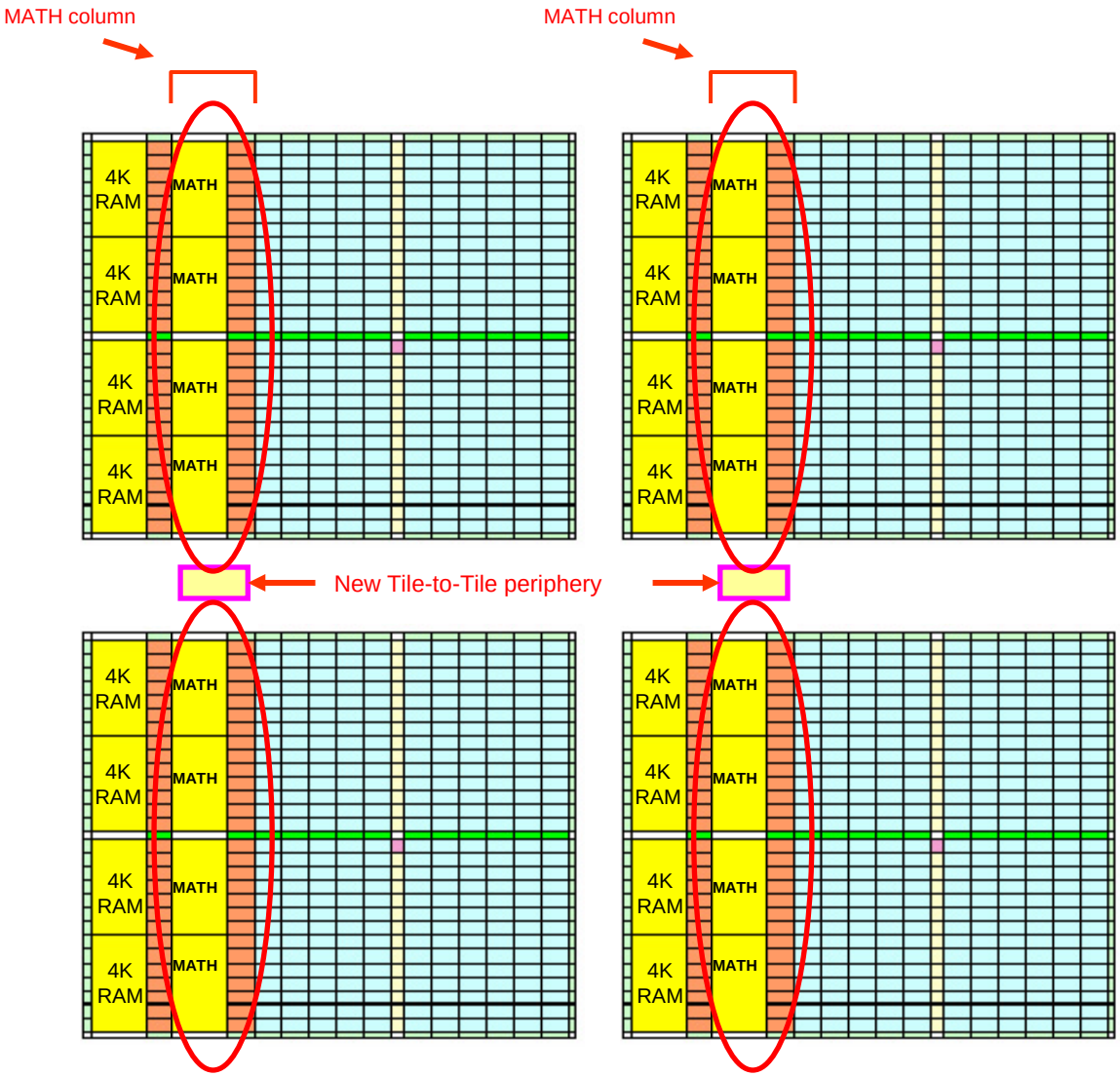
RTAX-DSP Overview

- Enhancement to Existing RTAX-S/SL Devices
  - Same 0.15 µ UMC process, same antifuse programming technology
  - DSP mathblocks run 18-bit x 18-bit multiply-accumulate at 125 MHz
    - DSP blocks protected against heavy ion radiation effects
  - RTAX-DSP and RTAX-S/SL package compatibility
    - RTAX4000D-CG1272 = RTAX2000D-CG1272
    - RTAX4000D-CQ352 = RTAX4000S/SL-CQ352
    - RTAX2000D-CQ352 = RTAX2000S/SL-CQ352
  - Routing architecture remains unchanged
    - 8.3% fewer logic modules

	RTAX2000	RTAX4000	RTAX2000D	RTAX4000D
R-cells	10,752	20,160	9856	18,480
C-cells	21,504	40,320	19,712	36,960
RAM Blocks	64	120	64	120
Math Blocks	0	0	**64**	**120**
Clocks	8	8	8	8
IO	684	840	684	840

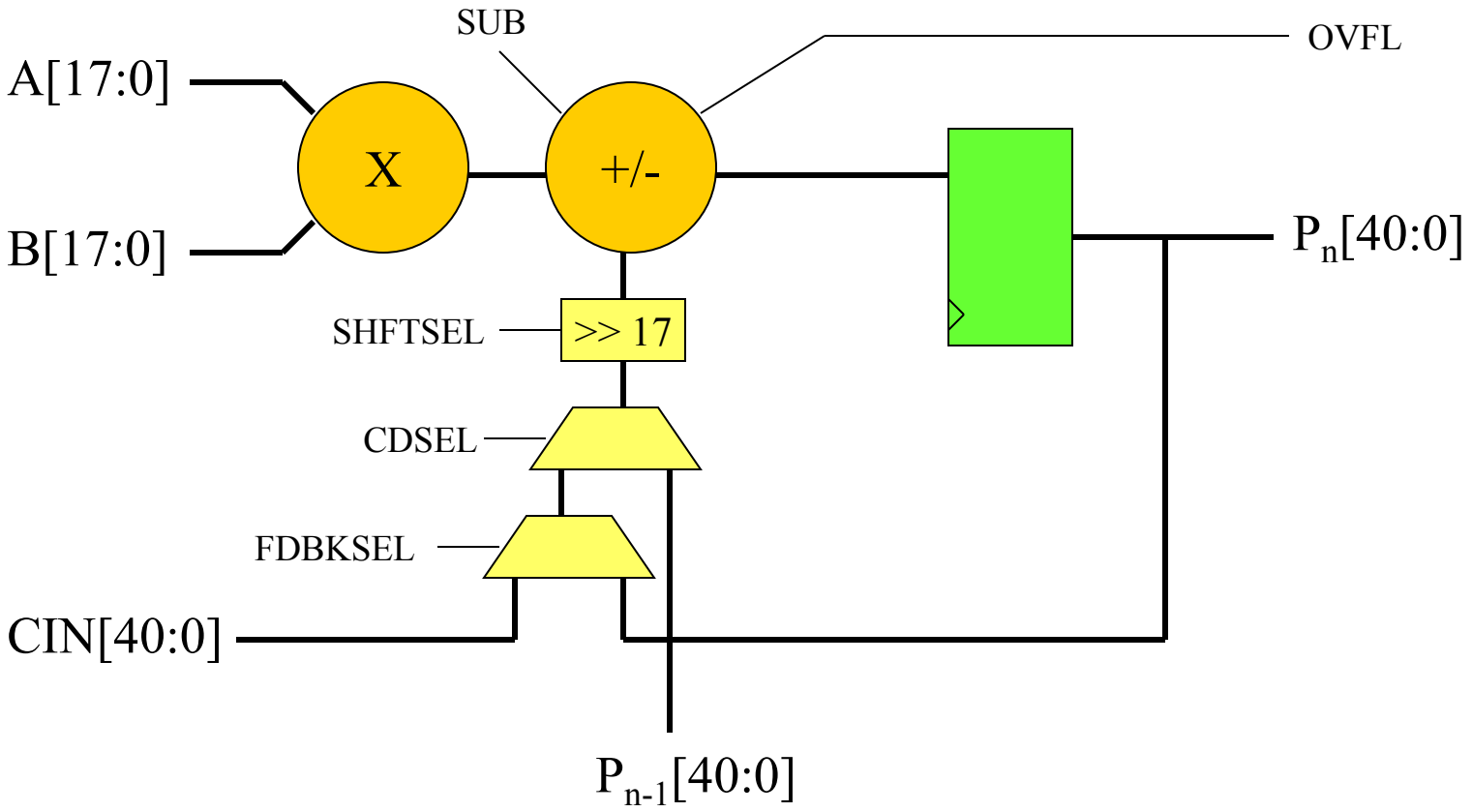
Logic cell counts decreased

Mathblocks in Core-Tile



- MATH column
  - Mathblocks plus MATH clusters
  - Placed next to the RAM column on the west side of each core-tile
- MATH clusters
  - Similar to RAM clusters
  - For interfacing to the FPGA core

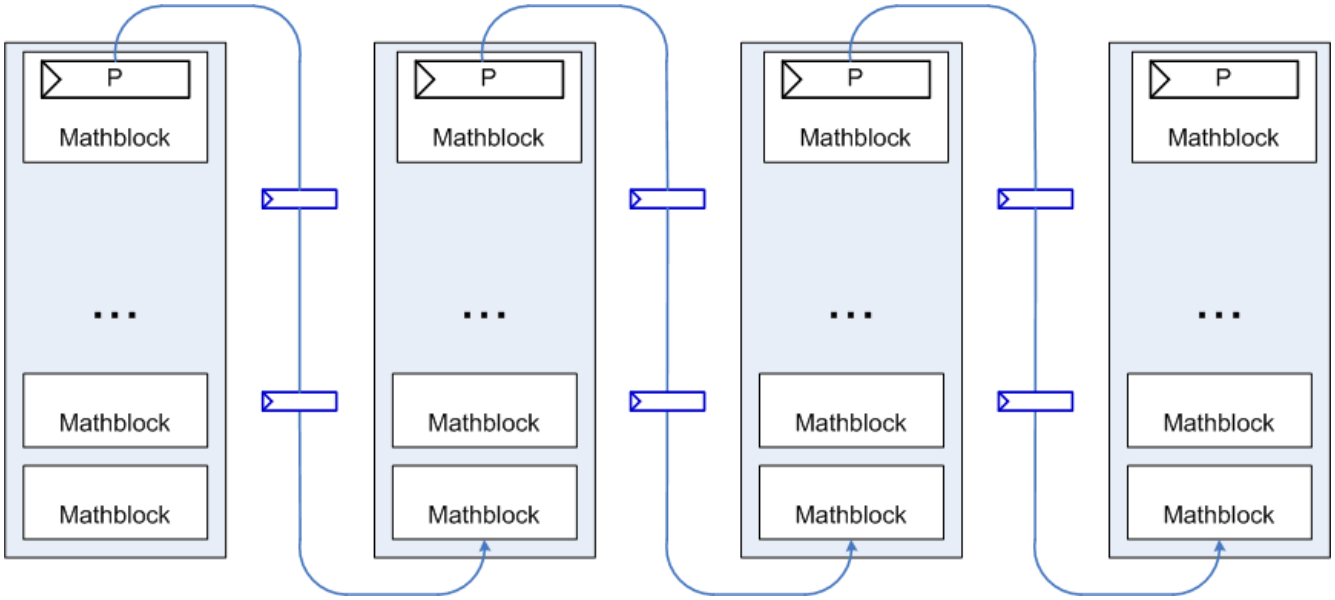
18x18 Mathblock Simplified Block Diagram



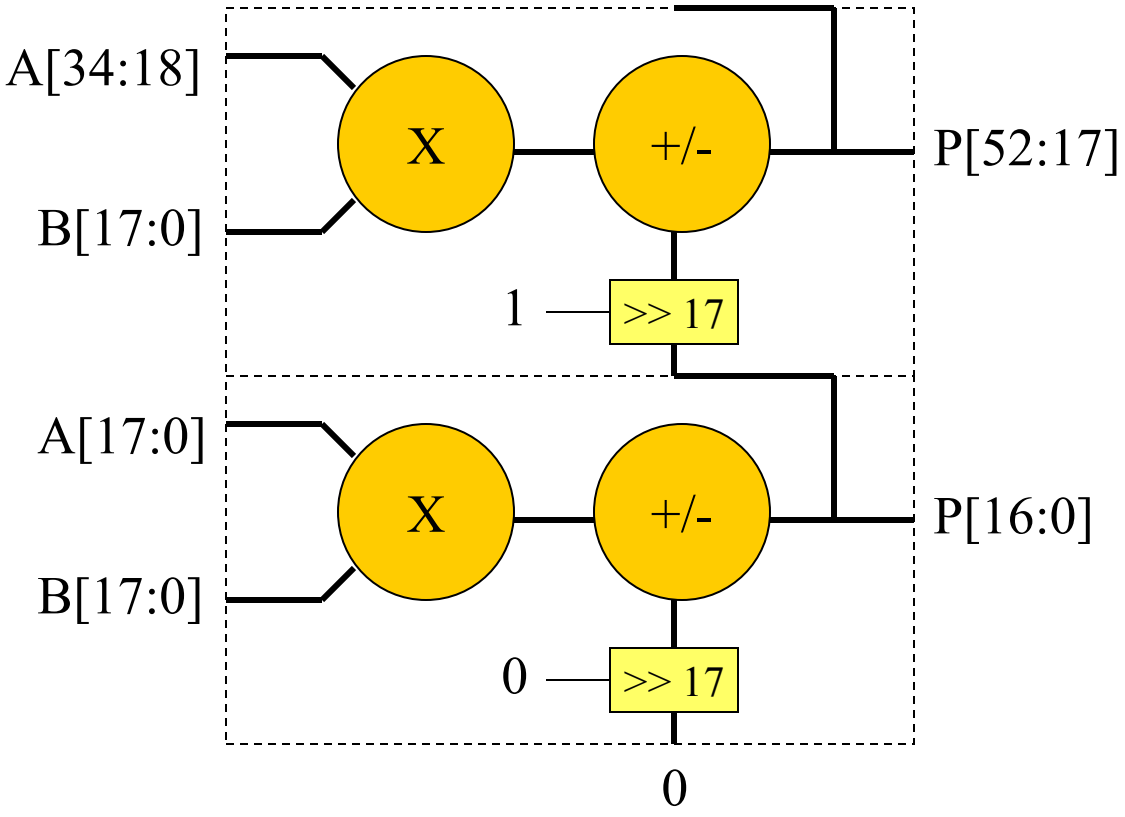
Optional input registers not shown

Mathblocks dedicated connection

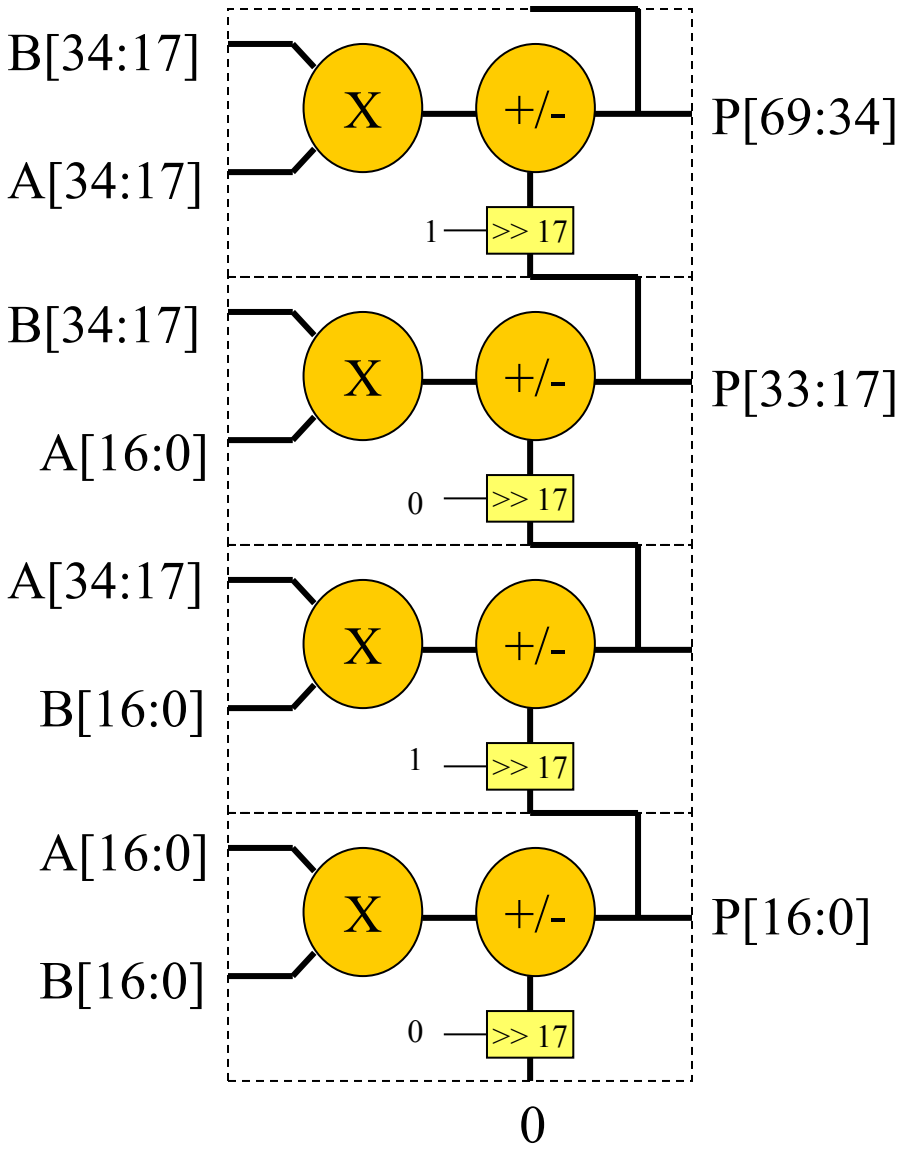
- Shift and cascade function enables creation of precise and complex functions like wide multipliers
- Dedicated connection between mathblocks columns improves performance
-



Unsigned High Precision 18x35 Multiplier



Unsigned High Precision 35x35 Multiplier



Mathblocks Design Techniques

There are 4 ways to design with mathblocks:

1. Use Actel IP such as CoreFIR and CoreFFT -- *Recommended*
   - Provides best performance with clock rate up to 125 MHz

2. Instantiate mathblocks macro

3. Use Synopsys Synplify Pro to infer mathblocks

4. Mathworks Matlab/Simulink and Synopsys Synphony Model Compiler

Use Actel IP CoreFIR

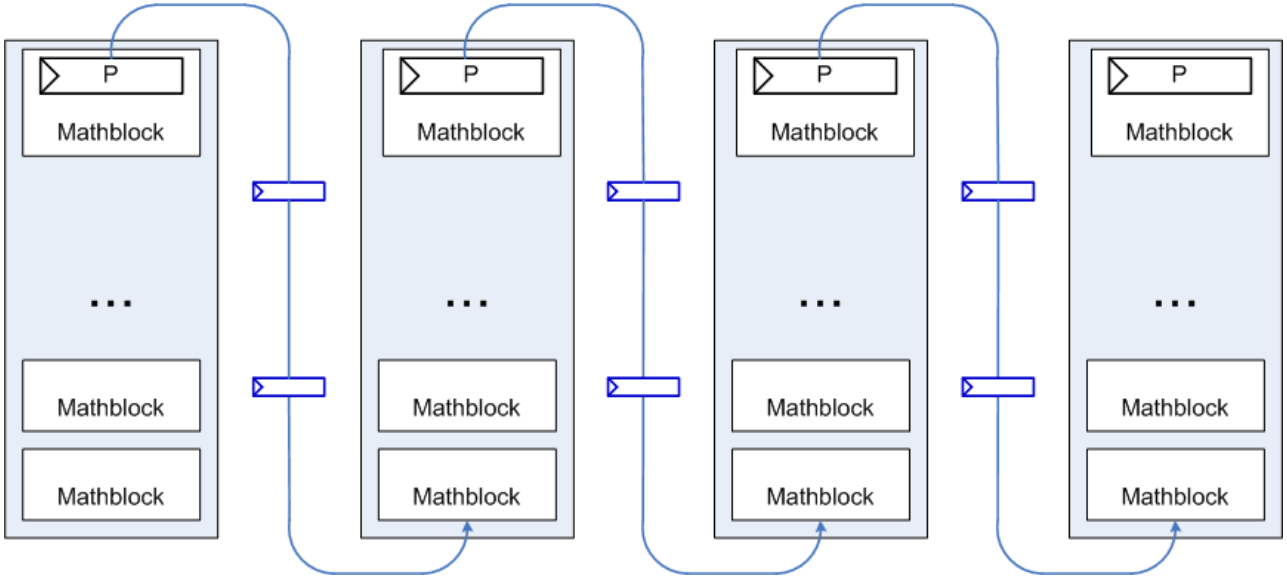
CoreFIR key features

- Highly configurable DirectCore RTL generators
- High performance single rate fully enumerated MAC Filter with clock rate up to 125 MHz
- 2 to 2N taps, where N is a number of physically available Mathblocks
- 2 to 18 bits input data and coefficient precision
- Signed or unsigned data and coefficients
- Full precision output
- Coefficient symmetry optimization
- Run-time reloadable coefficients, multiple coefficient sets, or fixed coefficients

Inter-Column Pipelines

- Dedicated resource within column provides excellent performance
- Filter utilizing more than 1 column creates extended propagation delay

  => CoreFIR automatically infers optimal number of FPGA core pipeline registers in the inter-column sections of the adder chain to reduce this delay



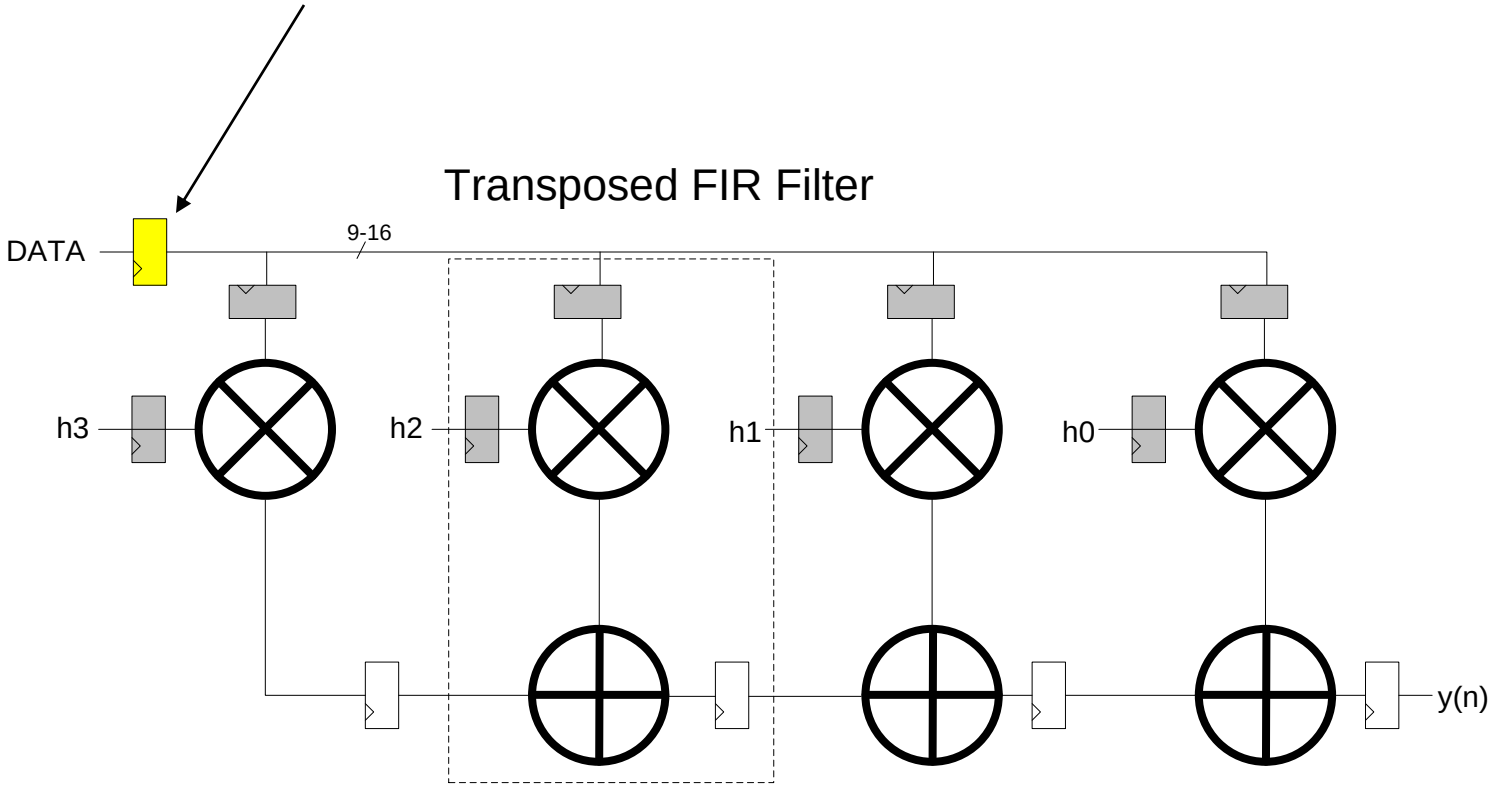
CoreFIR Utilization and Performance

Taps	Symmetry	Structure	Configuration	Bit width		Mathblocks	Cells usage (%)	Max Data Rate (Msps)
				Coefficient	Data			
16	No	Transposed	1	18	18	16	0.6	124
16	No	Transposed	2	18	18	16	0.9	124
16	No	Transposed	3	18	18	16	1.9	124
16	No	Systolic	1	18	18	16	2.7	124
16	No	Systolic	2	18	18	16	3.0	124
32	Yes	-	1	18	17	16	3.9	98
32	Yes	-	2	18	17	16	4.1	93
32	Yes	-	3	18	17	16	5.2	103
64	No	Transposed	1	18	18	64	4.5	90
64	No	Transposed	2	18	18	64	5.6	83
64	No	Transposed	3	18	18	64	10.0	83
128	Yes	-	1	18	17	64	18.1	74
128	Yes	-	2	18	17	64	19.0	69
128	Yes	-	3	18	17	64	23.4	71

- RTAX2000D Device Utilization and Performance

High Performance FIR

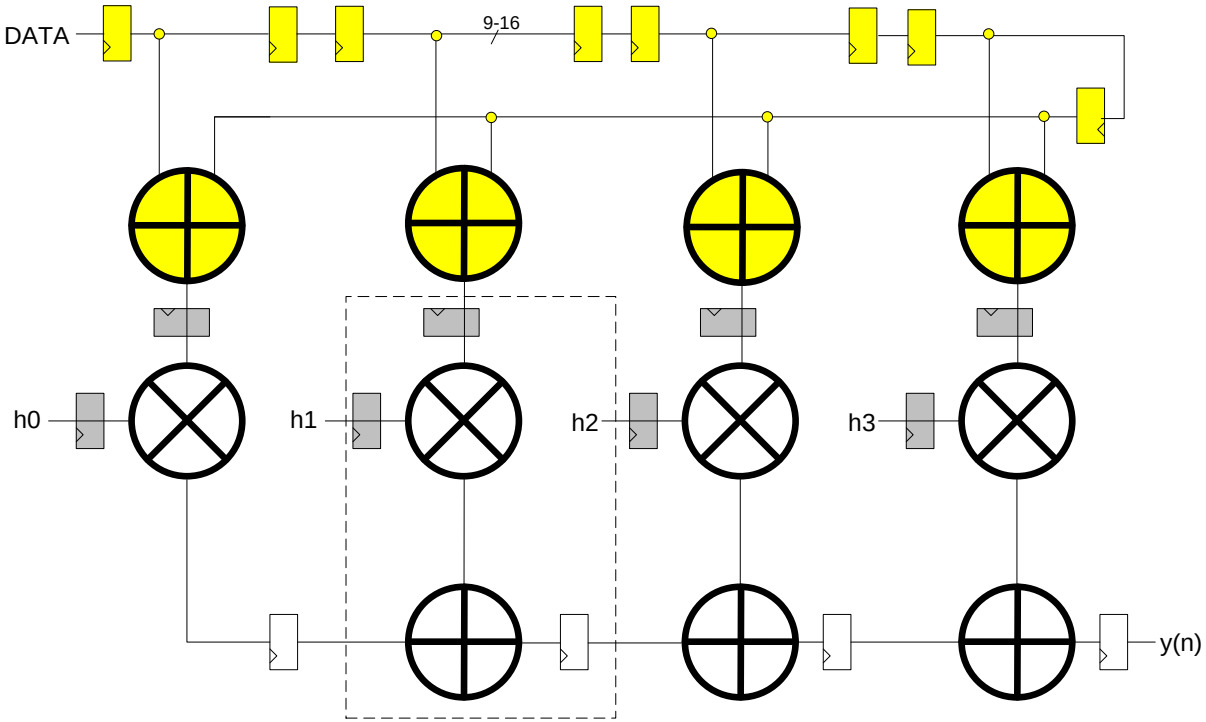
Yellow indicates a FPGA core resource



One mathblock per tap

High Performance FIR

Systolic Symmetric FIR Filter



Yellow indicates a FPGA core resource

One mathblock per two taps

CoreFIR General Info

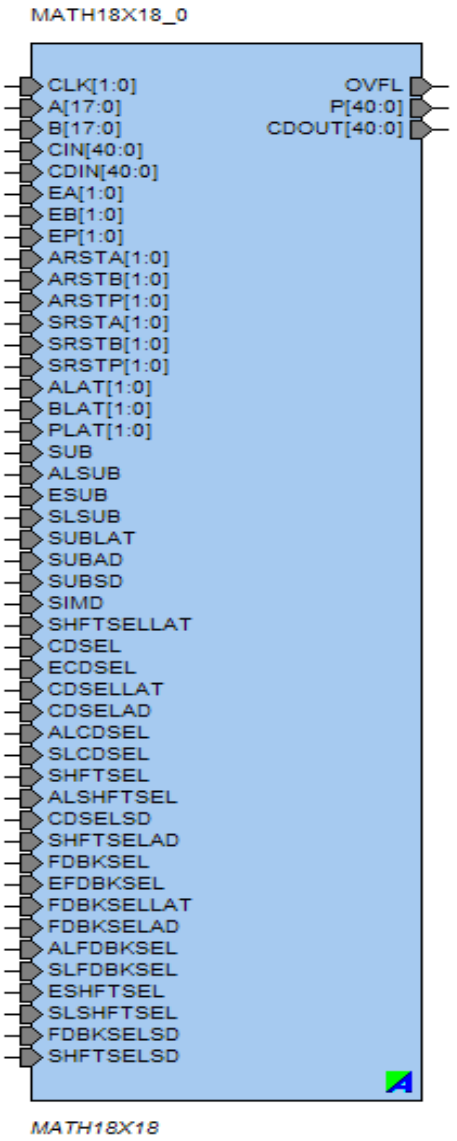
- Current version is v4.1
- Available for download from Libero IP Catalog with Libero Platinum and Libero SA license
- For more information about CoreFIR, refer to handbook
  http://www.actel.com/ipdocs/CoreFIR_HB.pdf

Instantiate Mathblocks Macro

Mathblocks Macro

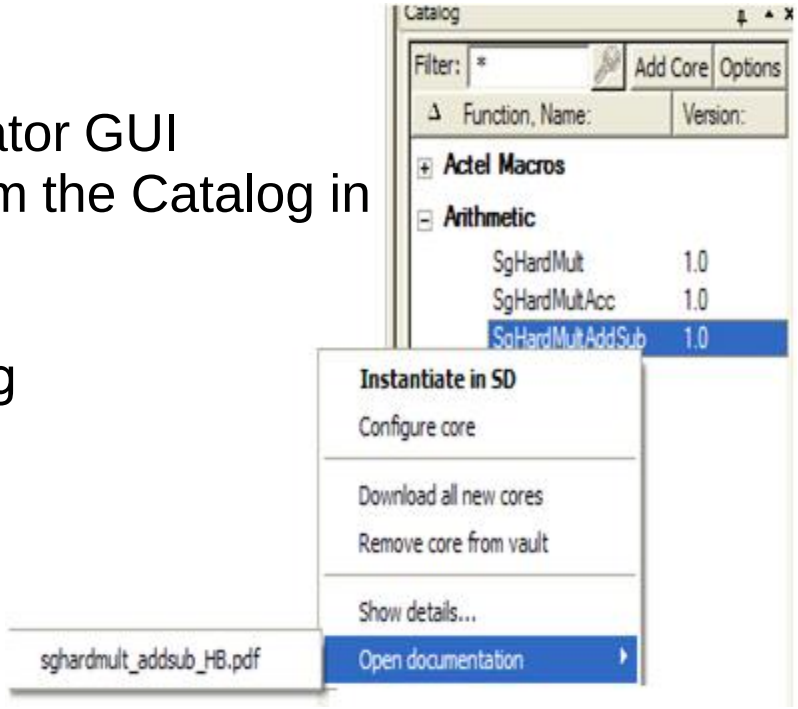
- MATH18x18 macro was released in Libero v8.5 without any configurator in Libero Catalog
- Refer to Antifuse Macro Library Guide for port descriptions and truth table of functionality
- But it is not easy to design by instantiating this macro

  => Use Libero Catalog Packaged Mathblocks core instead



Libero Catalog Core Configurator

- The macro MATH18x18 is packaged into three different cores
  - sgHardMult (Simple Multiplier)
  - SgHardMultAddSub (Multiplier with Adder/Subtractor)
  - SgHardMultAcc (Multiplier with Accumulator)
- Properties of these cores:
  - Each core has its own configurator GUI
  - Each core can be accessed from the Catalog in Libero
  - Cores are displayed under New "*Arithmetic*" node in the Catalog



Core Configurator General Info

- SgHardMult, SgHardMultAddSub, and SgHardMultAcc, must be used with Libero IDE v8.5 SP1 or later

- For more info, refer to each core's handbook:

http://www.actel.com/documents/sghardmult_HB.pdf
http://www.actel.com/documents/sghardmult_addsub_HB.pdf
http://www.actel.com/documents/sghardmult_acc_HB.pdf

Antifuse Macro Library Guide:
http://www.actel.com/documents/libguide_ug.pdf

Synplify Inference

Synplify Inference

- Support with Libero v8.6 and newer Release
- Inference of:
  - Simple Multiplier
  - Multiplier followed by Adder
  - Multiplier followed by Subtractor
- Supports any MxN multipliers
  - Any multipliers larger than 18x18 (signed) and 17x17 (unsigned) will be fractured into smaller multipliers and adder logic
- By default, all multipliers with input widths of 3 bits or greater will be mapped to DSP blocks.
  - If input width is smaller than 3 bits, multipliers will be mapped to logic
  - Default mapping behavior can be controlled through an attribute:
    - syn_multstyle = “logic” or “DSP”
- For coding style example, refer to this apps note
  - http://www.actel.com/documents/synplify_mathblocks_an.pdf

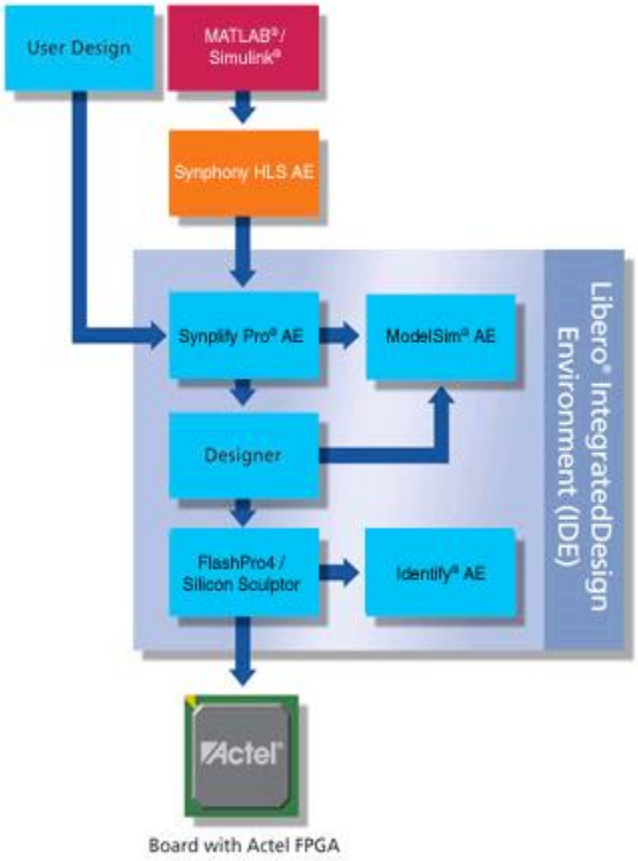
Synplify Inference (Cont.)

- Support of inferring MATH18X18 blocks spread across hierarchies, provided all Design Rules are met
- Infer pipelined multipliers to achieve max performance
- Future Support:
  - Inference of Multiplier-Accumulators (MACs)
    - Use Accumulate (Feedback) feature of MATH18x18 macro for efficient mapping
  - Inference of multi-input Mult-Add/Sub by cascading mathblocks

Mathworks Matlab/Simulink and Synopsys Synphony Model Compiler

Actel DSP Design Flow

- Synphony output is not the most efficient and optimal
  - Generic RTL, not Actel specific
  - High utilization, low performance
- Plan to release Actel libraries as an add-on to Synphony libraries:
  - Configurable core for user
  - Can be simulated and verified in Simulink
- Incoming release will support:
  - Actel CoreFIR
  - Actel CoreFFT



Common Connectivity Errors & Mitigation

Design Rule Checks (DRC)

- **CMP366: myMult_0/myMult_0/U0:CDOUT[40] must drive CDIN of another MATH18X18 instance**

  Or **CMP373:SgHardMultAddSub_0/U0:CDIN must be driven by CDOUT of another MATH18X18 instance**
  - Check for valid connection between mathblocks

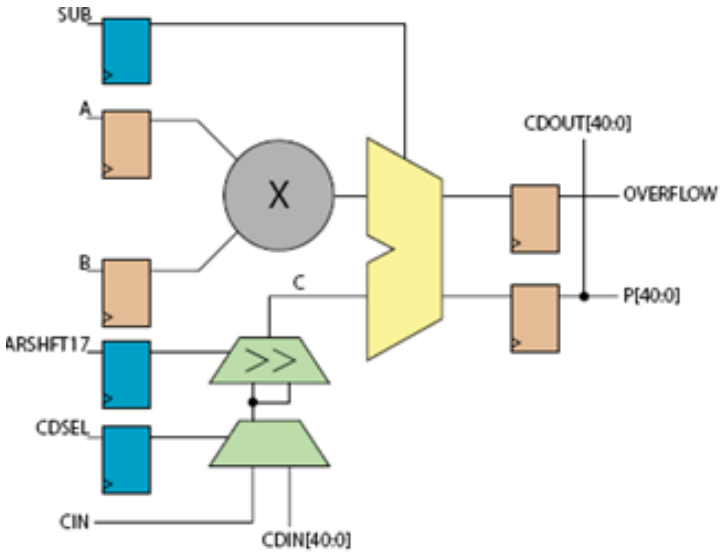
- **CMP369: MathInst1 is not using CIN, but CIN[i] is not connected to GND**
  - Make sure to connect CIN to GND if it is not used

- **CMP371: MathInst1 cascade selection is not always OFF and CDIN is not driven by CDOUT of another MATH18X18 instance**
  - CDSEL signal cannot be dynamic signal. It must be either 1 or 0. In this case, CDOUT is not driving CDIN but CDSEL could turn cascade selection on since it is dynamic signal.

Design Rule Checks (DRC) (cont)

- **CMP372: MathInst1 cascade selection is always OFF and CDIN is driven by CDOUT of another MATH18X18 instance**
  - Ensure that CDSEL = 1. Refer to Truth table below if CDSEL is registered

al	ad	lat	clk	en	sl	sd	d	q
0	x	x	x	x	x	x	x	ad
1	x	0	not rising	x	x	x	x	q
1	x	0	rising	0	x	x	x	q
1	x	0	rising	1	0	x	x	!sd
1	x	0	rising	1	1	x	x	d
1	x	1	x	0	x	x	x	q
1	x	1	x	1	0	x	x	!sd
1	x	1	x	1	1	x	x	d



Design Rule Checks (DRC) (cont)

- **ERROR: Cascaded chain of length (32) begining at fir_dsp_cells_i/dsp_cell_i0/math18x18_i exceeds the dimension of the device (16)**
  - Cascading of Mathblocks is limited to the height of the device
  - To use P(Last one of the 1st chain) -> CIN (First one of the 2nd chain) instead of COUT(Last one of the 1st chain) -> CIN (First one of the 2nd chain)
  - Note that P -> CIN is a slower connection that goes through the FPGA core. Software usually performs DRC check for the device height and converts any P -> CIN connection to COUT -> CIN connection for better performance if possible

RTAX-DSP Design FAQ

RTAX-DSP Design FAQ

1. Why is Compile invalidated on pre-v8.6 designs?

- Due to an update in the data library, the following messages will display when opening a pre-v8.6 design

Warning: The design uses outdated library data. Invalidating Compile. Compile must be re-run.

- Preserve your existing Layout by running the placer with Incremental Mode “FIX” and the router with Incremental Mode “ON”

2. When cascading mathblocks, what if the chain length is more than the mathblocks column height?

- Use P -> CIN connection instead of COUT -> CIN

RTAX-DSP Design FAQ (cont)

3. **Why using up to 4 pipeline registers did not improve my performance?**
   - When the mathblocks are in same column, the router uses dedicated hardwire connection and can generally achieve good performance.
   - When the router starts using local routing, that's when performance can drop.
   - Pipelining can help when they are placed optimally. As the number of pipeline stages increases, the register placement becomes critical and affects timing/performance more. Using 4 pipelines registers does not automatically improve performance.

RTAX-DSP Design FAQ (cont)

4. Does the latest version of Synphony AE v2009.12a SP2 work with all MATLAB/Simulink releases?

It is supported with the following versions of MATLAB/Simulink:

Release 7.6 (2008A)

Release 7.7 (2008B)

Release 7.8 (2009A)

Release 7.9 (2009B)

Release 7.9.1 (2009B SP1)

Release 7.10 (2010A) -- Recommended

RTAX-DSP Design FAQ (cont)

5. Is the Simulink "signal processing toolbox and blockset" and the "filter design toolbox" a hard requirement for Synphony?

- These libraries are optional for the Synphony/ Actel tool flow. The signal processing toolbox would be of little help in designing the FPGA portion (which has to be done using Synphony libraries exclusively), but has a lot of useful info for building testbench
- The filter design toolbox makes the design of standard filters much easier. But, if you already know what filters you need, then it's less useful

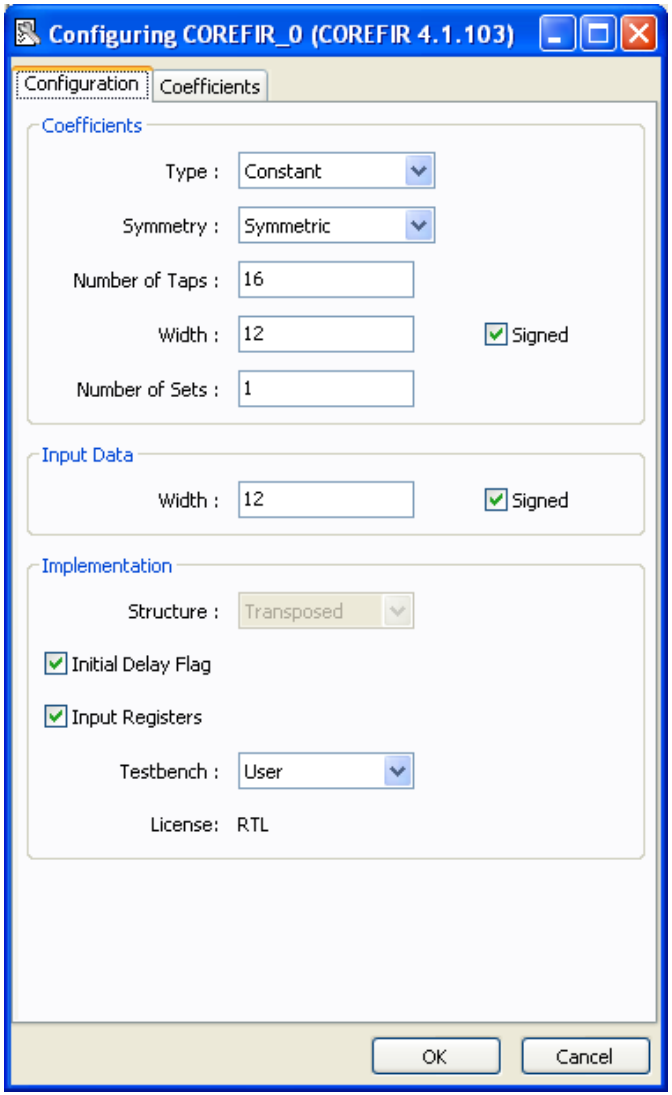
6. I haven't received a copy of Libero IDE. Can I create and run models in MATLAB/Simulink with just Synphony?

- Yes. You do not need Libero to run DSP flow up to Simulink simulation.

THANK YOU

Backup slides

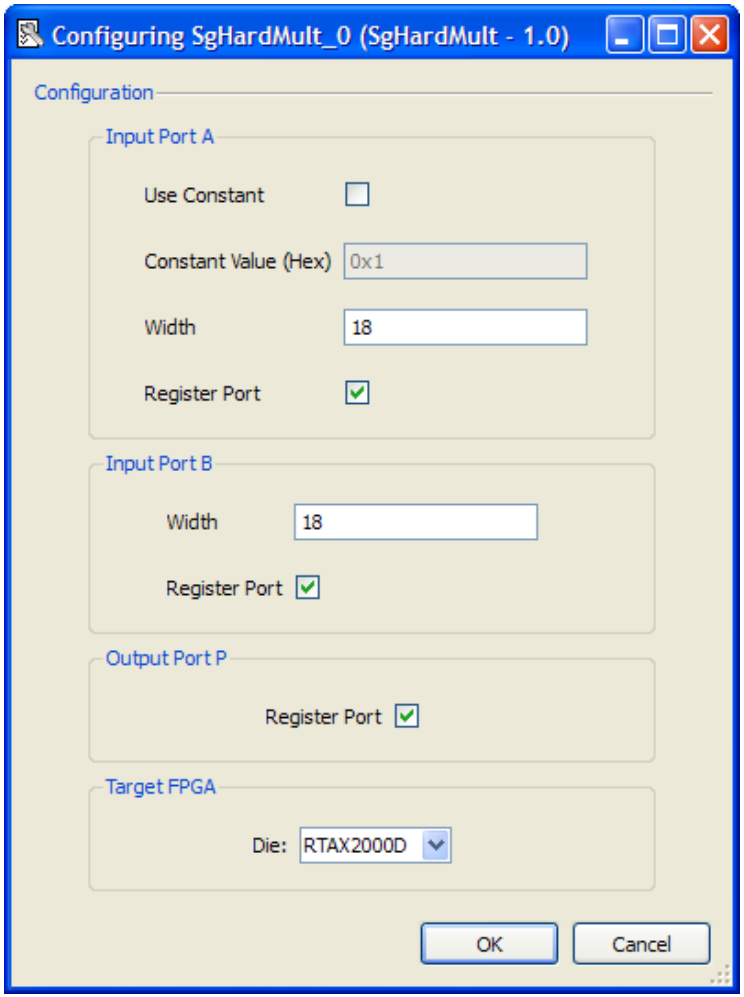
CoreFIR software implementation

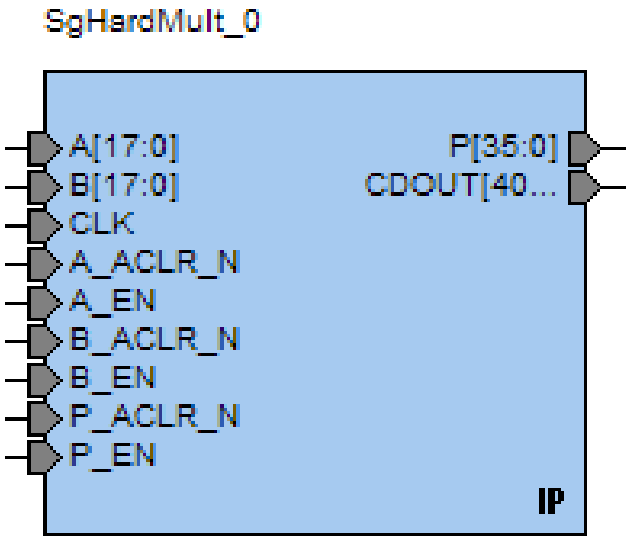


- Instantiated using SmartDesign Flow
- Include User Testbench for Simulation
- Require no special Synthesis or Layout setting, while guarantee performance up to 125 MHz

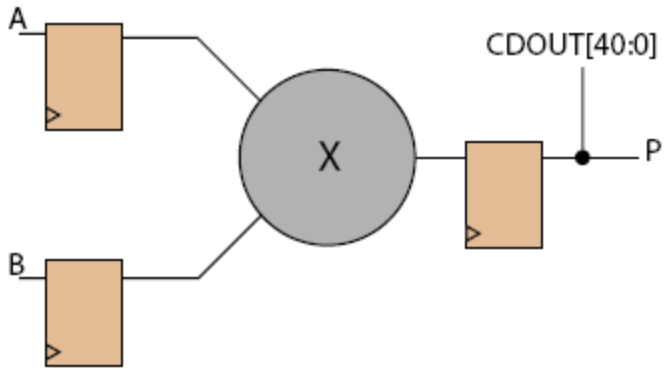
Simple Multiplier Configurator

- SgHardMult



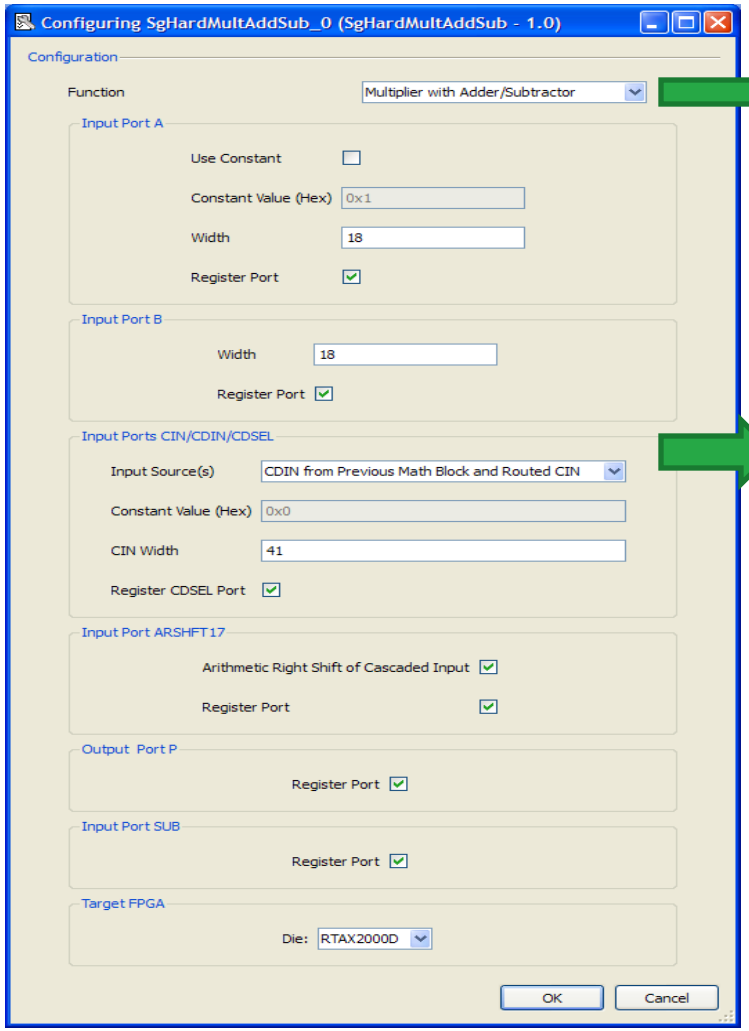


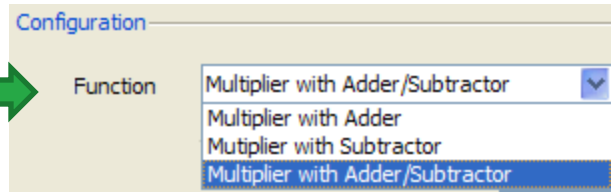
SgHardMult

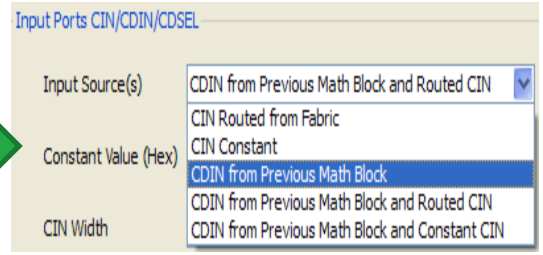


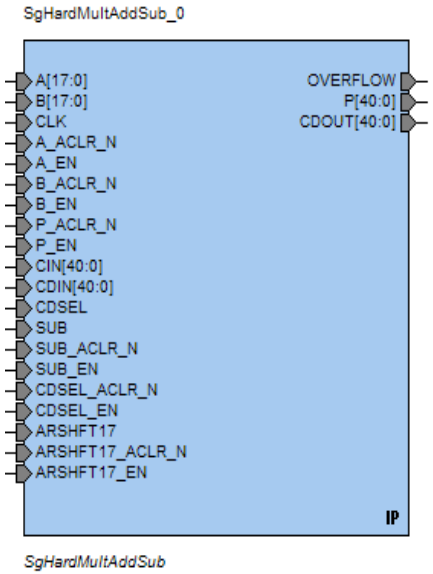
Multiplier with Adder/Subtractor Configurator

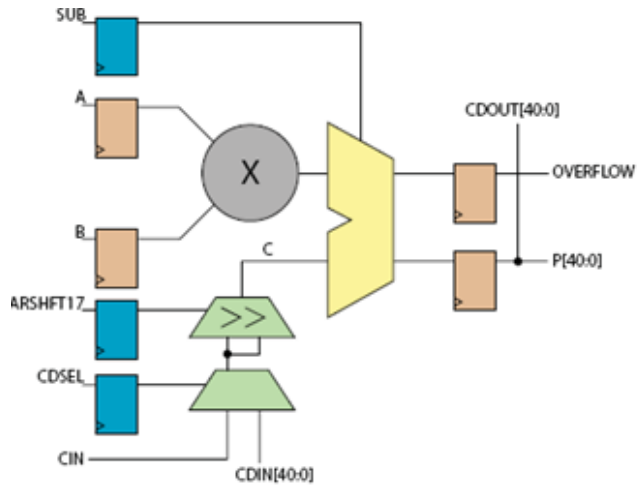
- SgHardMultAddSub











Multiplier with Accumulator Configurator

- SgHardMultAcc