Stratix III DSP EDA Functional Description

Version 1.0

March 9, 2009

By

Altera Corporation

Table of Contents:

1. Overview

This document describes the overall Software support plan for the Stratix III DSP block. It will focus on changes planned for Quartus v6.1, but when possible will also list those items deferred to later releases.

The Stratix III DSP block is a dedicated hard block that implements basic DSP functions such as multipliers, adders, and shift registers. The Stratix III DSP block has more features than the Stratix-II DSP block, but it is not a complete superset of the Stratix-II DSP block. The Stratix-II features not supported in Stratix III are expected to be those that are not commonly used.

Stratix III DSPs are targeted to support up to 560 Mhz performance (see NPP or Quartus for actual numbers). In the largest device there are up to 112 DSP blocks, which implies in the best case that up to 896 18x18 multipliers can be used.

The major changes in Stratix III compared to Stratix-II are native support for 12x12 multipliers, the ability to chain adders between multiple blocks to form larger adders, lack of dynamic mode support, a specialized shift/rotate mode, 4-multipliers feeding the accumulator, shift registers on only one operand, pipeline register moved to after the adder instead of multiplier, and enhanced rounding/saturation.

This feature is derived from device support requests covered in the Stratix III DSP NPP.

2. Glossary

DSP block – Digital Signal Processing block

DSP function – See DSP slice

DSP slice – Represents a single DSP function, or group of connected mac_mults and mac_out

FFT – Fast Fourier Transform

FIR Filter – Finite Impulse Response Filter

FLED – Floorplan Editor

IIR Filter – Infinite Impulse Response Filter

MAC – Multiply Accumulator. Sometimes used as synonym for DSP block.

Q1.15 – Fixed point arithmetic notation with 15-bits of precision

3. Comparison to Stratix-II

Table 1: Stratix-II Versus Stratix III DSP Block

Feature	Stratix-II	Stratix III
Block dimensions	4 x 1 LABs	4 x 1 LABs
Native multiplier widths	9x9, 18x18, 36x36	9x9, *12x12*, 18x18, 36x36
Multipliers per mode	Depending on mode	Depending on mode, but more than Stratix-II in some modes

Input shift register	Both A & B operands and most modes	Only A operand and only sum of 4 mult modes, and 18x18 modes. Plus extra register stage when using adder chainout register.
Input shift register parallel load	Yes	No
Extra output adder with chaining	No	Yes – essentially a 3rd level adder.
Loop back multiplier sum mode	No	Yes
Dynamic mode	Yes	No
Dynamic add/sub on first level adder	Yes	No. But static configuration of both adders possible.
Shift mode (left, right, rotate)	No	Yes
4-multiplier-accumulator mode	No	Yes
1-multiplier-accumulator mode	Yes	Yes, as special case of 4-mult-accum mode
Accumulator width	Up to 54-bits	Up to 44-bits
Accumulator loading	Yes, entire 54-bits in one cycle.	No. But lower 36-bits can be loaded with output of two mults.
Register/Clocking Flexibility	Very flexible. Every input bank is separate, and can select between 4 clocks.	Slightly less flexible. Each quadrant has 3 register banks: at input, pipeline, output. Each register bank can select between 4 clocks, except the output register bank which can select different clocks for the upper and lower halves (for chainout mode).
Register bypassing	Yes, separate for every control signal and 18-bit operand	Yes, separate for every control signal in input, pipeline and output register banks; and separate for 18-bit data operand in input bank, but entire data path for pipeline and output bank. In chainout modes, the 2nd-stage register and output register can be bypassed independently as well.
Adder gating for power savings	No	Yes
Swapping 18-bit operands in 36x36 mode for power savings	No	Yes
Timing model each bit through multiplier separately (faster smaller width mults)	No	No

Rounding	Yes, only Q1.15 biased rounding. Available after multiplier and after accumulator.	Yes, but different: programmable width, biased/unbiased rounding. Post-op-block so no mult rounding in all cases. Also available after 2nd-level adder and chainout adder in chainout modes.
Saturation	Yes, only Q1.15 asymmetric saturation. Available after multiplier and after accumulator.	Yes, but different: programmable width, asymmetric/symmetric saturation. Post-op-block so no mult saturation in some cases. Also available after 2nd-level adder and chainout adder in chainout modes.
Pipeline register	Yes, after multiplier	Yes, but after *1st-stage adder.*

Table 2: Number of Functions per DSP Block

DSP Mode	# of Functions in Stratix-II	# of Function in Stratix III
Basic 9x9 multiplier	8	8
Basic 12x12 multiplier (implement with 18x18 in SII)	*4*	*6*
Basic 18x18 multiplier	4	4
Basic 36x36 multiplier	*1*	*2*
Sum of two 18x18	*2*	*4*
Sum of four 18x18	*1*	*2*
Sum of two 9x9 (implement with 18x18 in Stratix III)	4	4
Sum of four 9x9 (implement with 18x18 in Stratix III)	2	2
18x18 multiply accumulator	2	2
Shift Mode	*0*	*2*
Dynamic Mode	*1*	*0*

4. Implementation

This section describes some implementation details of the Stratix III DSP block, focusing on changes from Stratix-II.

4.1 Modes

Table 3 describes the logical modes (i.e. at function level) that can be implemented in a DSP block.

Table 3: Stratix III DSP Block Mode Details

Mode	Multiplier In Width	# of Mults	# per Block	Signed, Unsigned	RND, SAT	In Shift Register	Chainout Adder	Add, Sub
Independent Multiplier	9-bits	1	8	Both	No	No (use 18-bit)	No	N/A
	12-bits	1	6	Both	No	No (use 18-bit)	No	N/A
	18-bits	1	4	Both	Yes	Yes	No	N/A
	36-bits	1	2	Both	No	No	No	N/A
Multiply-Accumulator	18-bits	4	2	Both	Yes	Yes	Yes	Both, Acc also has both
Multiply-Adder	18-bits	4	2	Both	Yes	Yes	Yes	Both
	18-bits	2	4	Signed full precision Unsigned not full or only for smaller widths	Yes	No	No	Both
Multiply-Adder with Loopback	18-bits	2	2 of 4[1]	Same as above	Yes	No	No	Both
Shift (dynamic switch between arithmetic, logical, left, right, rotate)	36-bits (32-bit data)	1	2	Both	No	No	N/A	N/A

[1] In loopback mode, two instances in a DSP block are two-multiplier-adder-with-loopback functions and two instances are regular two-multiplier-adder functions

4.2 WYSIWYG

Refer to the Stratix III DSP WYSIWYG document for details.

Table 4 lists the WYSIWYG level details of the available DSP functions (or DSP slices) that implement each of the higher-level modes. Items in *italics* are those that are different from Stratix-II.

Table 4: DSP WYSIWYG Mode Details

High-Level Mode	Multiplier In Width	MAC_MULT In Width	Number of MAC_MULT	MAC_OUT Mode	MAC_OUT Out Width
Independent Multiplier	9-bits	9-bits	1	output_only	18-bits
	12-bits	*12-bits*	*1*	*output_only*	*24-bits*
	18-bits	18-bits	1	output_only	36-bits
	36-bits	18-bits	4	36_bit_multiply	72-bits
Multiply-Accumulator	18-bits	18-bits	*4*	accumulator	*44-bits*
	18-bits	*18-bits*	*4*	*accumulator_chain_out*	*44-bits*
Multiply-Adder with loopback	*18-bits*	*18-bits*	*2*	*loopback*	*36-bits*
Multiply-Adder	18-bits	18-bits	2	one_level_adder	*36-bits*
	18-bits	18-bits	4	two_level_adder	38-bits
	18-bits	*18-bits*	*4*	*two_level_adder_chain_out*	*44-bits*
Shift (dynamic switch between arithmetic, logical, left, right, rotate)	*36-bits (32-bit data)*	*18-bits*	*4*	*shift*	*72-bits (32-bits data)*

4.2.1 Block Diagrams

The following are block diagrams for the MAC_MULT and MAC_OUT WYSIWYGs. Refer to the Stratix III DSP WYSIWYG document for further details.

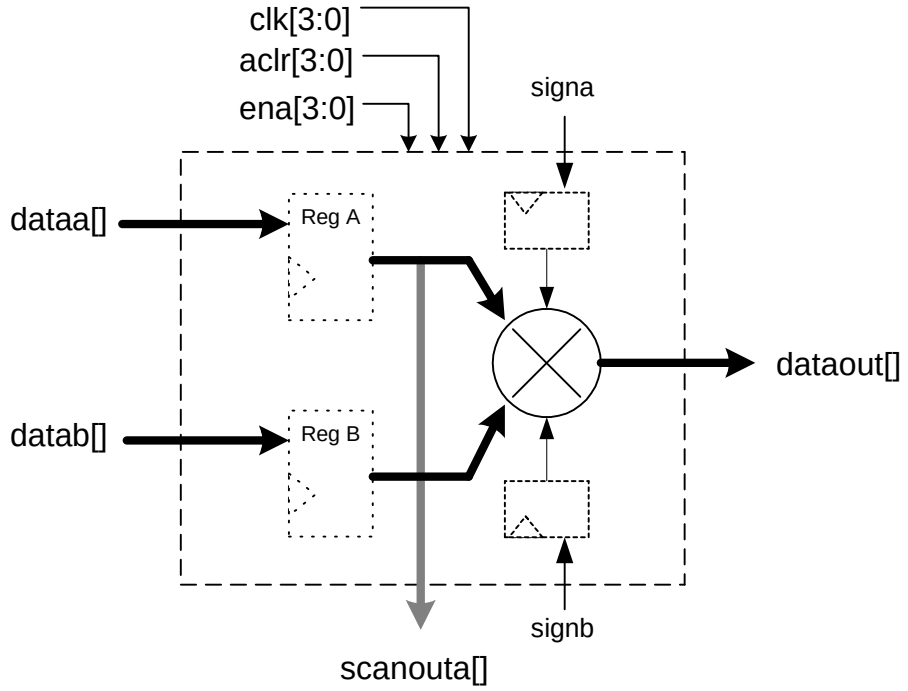

Figure 1: MAC_MULT Block Diagram

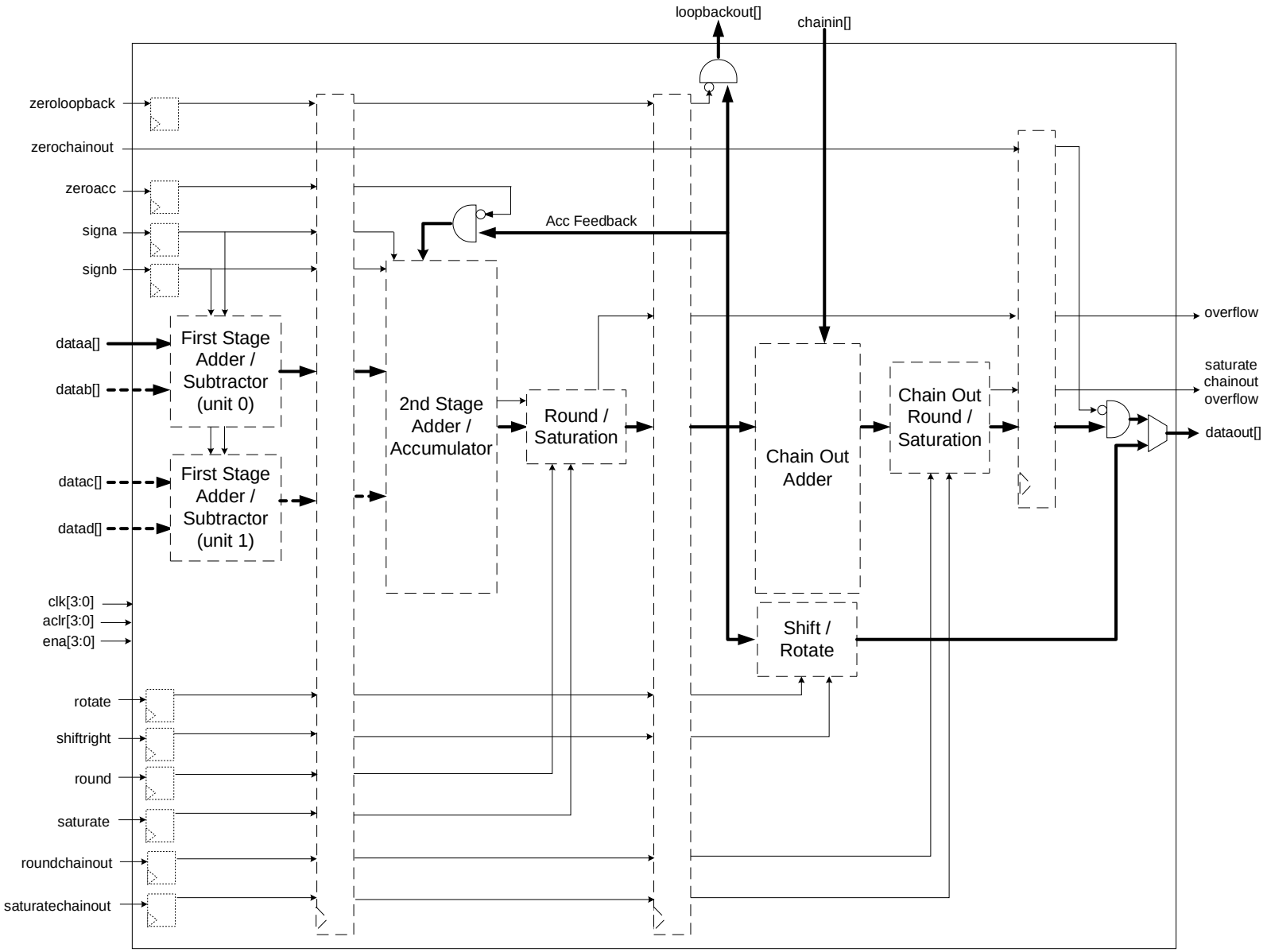

Figure 2: MAC_OUT Block Diagram

4.3 Multiplier Widths

Similar to Stratix-II, Stratix III DSPs support different native widths for the multiplier depending on the mode as outlined in Table 3: Stratix III DSP Block Mode Details.

New for Stratix III is support for native 12-bit multipliers. Additionally, adder modes do not have native support for 9-bit multipliers. However, since more 18-bit width summation modes can be implemented per DSP block, there is no loss in capability compared to Stratix-II.

As with Stratix-II, smaller widths will be implemented by padding GNDs to the LSBs during fitting.

4.4 Rounding and Saturation

The Stratix III rounding and saturation features have significantly changed compared to Stratix-II. They are more configurable (i.e. not just Q1.15 format), and the round/saturation block is located only after the second-stage adder and chainout adder (i.e. not after the multiplier).

New in Stratix III is the location of the saturation overflow signal. In modes where not enough outputs are available, the saturation overflow signal is part of the data, similar to Stratix-II, except that the overflow signal is located at the MSB bit of the data instead of the LSB. In modes where enough outputs are available, the saturation overflow signal is generated from the data – for these cases, a separate output signal is used.

There can be up to two round/saturation blocks in the datapath depending on the mode. For chainout modes, there is a round/saturation block after the 2nd-stage adder as well as after the chainout adder. For other modes, there is only one round/saturation block at the output. The two blocks can be individually enabled/disabled, however they must be configured similarly (e.g. same rounding width and same biased/unbiased selection). The atom has two sets of parameters for completeness, but both sets of parameters must match.

The saturation overflow output is shared with the accumulator overflow output in accumulator modes. When saturation is enabled, the output is always the saturation-overflow since the accumulator overflow will by definition be unasserted. In chainout modes, the chainout-saturation-overflow output is separate from the 2nd-stage overflow port.

4.5 Chainout Adder

This is a new feature in Stratix III. It is essentially an optional 3rd-level adder that can be used in the multiplier-adder modes which can be chained together between multiple DSP blocks.

The adder is fed by the 2nd-level adder of the current half-block, as well as the 3rd-level adder output of the previous half-block. This allows chaining together multiple DSP blocks to form very large output adders. The output of the 3rd-level chainout adder can also be registered.

In chainout modes, the dataout[] signal feeds both regular routing and dedicated routing to the chainin[] of the DSP below it. The zerochainout signal zero's the output of the chainout adder, which feeds the dataout[] output. In a typical configuration, the dataout[] will feed the chainin[] for multiple DSPs, with the last DSP in the chain having its dataout[] feed to the core.

The chainout adder output also feeds the saturation/rounding block and so the data can be saturated/rounded a second time after the chainout adder.

The following two diagrams show a comparison between the physical data paths of the two-level-adder mode and the two-level-adder-with-chainout mode as implemented in the hardware. As can be seen, in the chainout mode there is logically a 3rd level adder as well as two stages of output registers (the one after the 2nd-stage adder is referred to as the 2nd-stage-register in these modes). The chain-adder is in the upper quadrant, and the 2nd-level adder is in the lower-quadrant. The WYSIWYG and atoms represent this data path in a logical structure (refer to the Stratix III DSP WYSIWYG document for details).

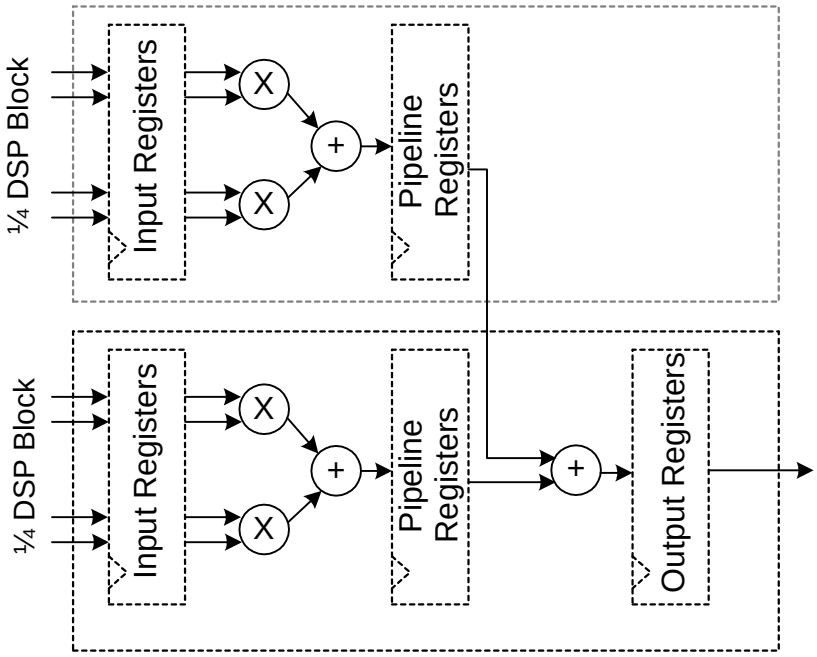

Figure 3: Two-Level-Adder Mode Data Path

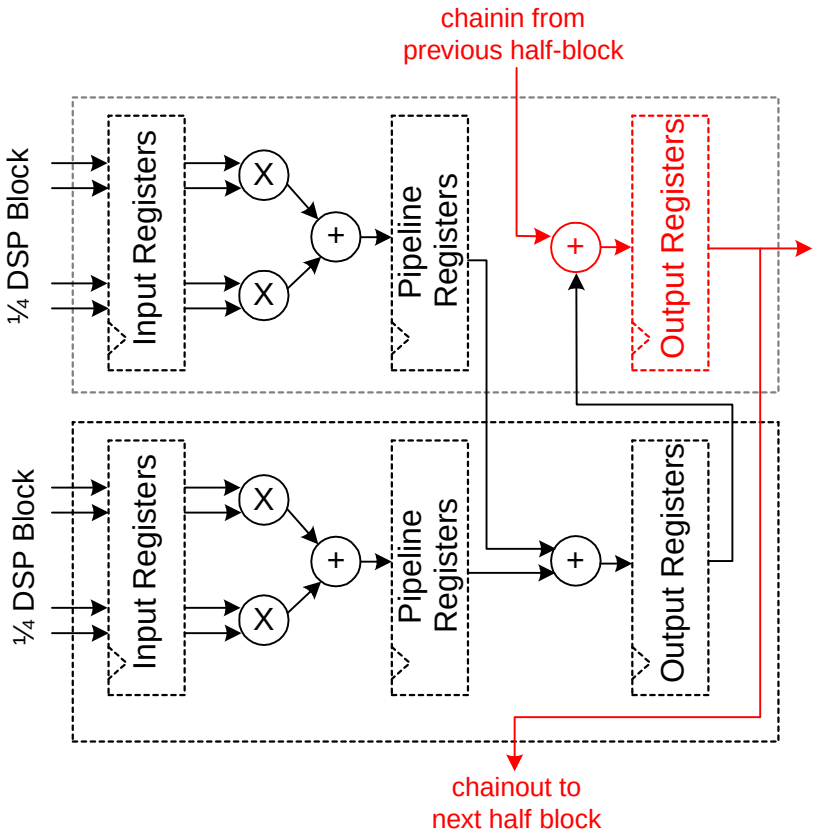

Figure 4: Two-Level-Adder-With-Chainout Mode Data Path

4.6 Input Shift Register

This is implemented similar to Stratix-II, except that it only exists on the A operand (not on the B operand). As in SII, the shift register output from the bottom multiplier in a DSP can feed regular routing in addition to dedicated routing. New in Stratix III is an extra register after the last input register which is intended for use with the chainout modes in order to maintain the correct data latency. Note that if the extra register is in use, it must feed the first MAC_MUTL in the next DSP block, no regular routing is allowed. It also implies that if output port scanouta is registered, the port bit-width has to be 18.

The input shift registers can technically be used even for designs that don't use DSPs: the multipliers can be configured to do a multiply by 1 to effectively bypass the multiply/add logic. This may be useful in designs where buses of shift registers are needed to match pipelining in other areas of the design that might otherwise be implemented in RAMs or LEs. The advantage over shift registers implemented in RAMs is that the shift registers in DSPs can be tapped at every stage. However one major caveat for implementing shift registers in DSPs is that it may adversely affect fmax and placement (e.g. registers inserted to pipeline the critical path or reduce routing congestion).

4.7 Add/Subtract Control

Unlike Stratix-II, the add/subtract control is not dynamic. Each of the 1st-level adders can statically select whether to add or subtract. The 2nd-level adder can only perform an add in most modes. However, the 2nd-level adder is used as the accumulator adder and so in the accumulator mode, the 2nd-level adder can also be statically configured to do subtraction (i.e. RESULT[n] = RESULT[n-1] – MULTA[N] – MULTB[N] – MULTC[N] – MULTD[N]). The 3rd-level adder can only do addition.

4.8 Loopback

For the case where two multipliers feed an adder, there is an optional internal loopback path from the output of the adder to the input of the top multiplier. This loopback path only exists for the top multiplier in each half block. Consequently, only two instances per block can implement this loopback, while the other two instances can be used as regular two-multiplier-adder functions.

As this loopback feature is specifically targeted for IIR applications that use fixed-point math, it is always the upper 18-bits of the adder output that are fed back to the multiplier.

This loopback feature can be implemented using regular external routing as well when the zeroloopback control signal is not used (or extra LEs are needed to implement that logic). The only advantage for using the internal loopback path is speed. As a result, the loopback mode should only be used for time critical paths.

Loopback mode will be a separate mode that the user instantiates. The altmult_add megafunction will be responsible for allowing the user to specify this special mode. Users have to manually decide which loopback functions to implement in this special mode and which to implement using slower external routing. Re-using the one-level-adder mode for loopback mode is not possible since the zeroloopback control signal is only available for the loopback case.

4.9 Clocking Scheme

The Stratix III clocking scheme is different from Stratix-II: it is slightly less flexible.

There is or a 4-1 MUX for every register bank quadrant in the DSP block (e.g. all data and control signals in a particular quadrant must share the same register configuration: same clock, clear, and clock-enable signals). However, registers can be bypassed independently for each control

signal in the input, pipeline and output register banks. The data signals can be bypassed per 18-bit operand in the input bank, but only per DSP quadrant in the pipeline and output banks. This allows specifying constant inputs correctly.

Refer to Figure 5 for more details.

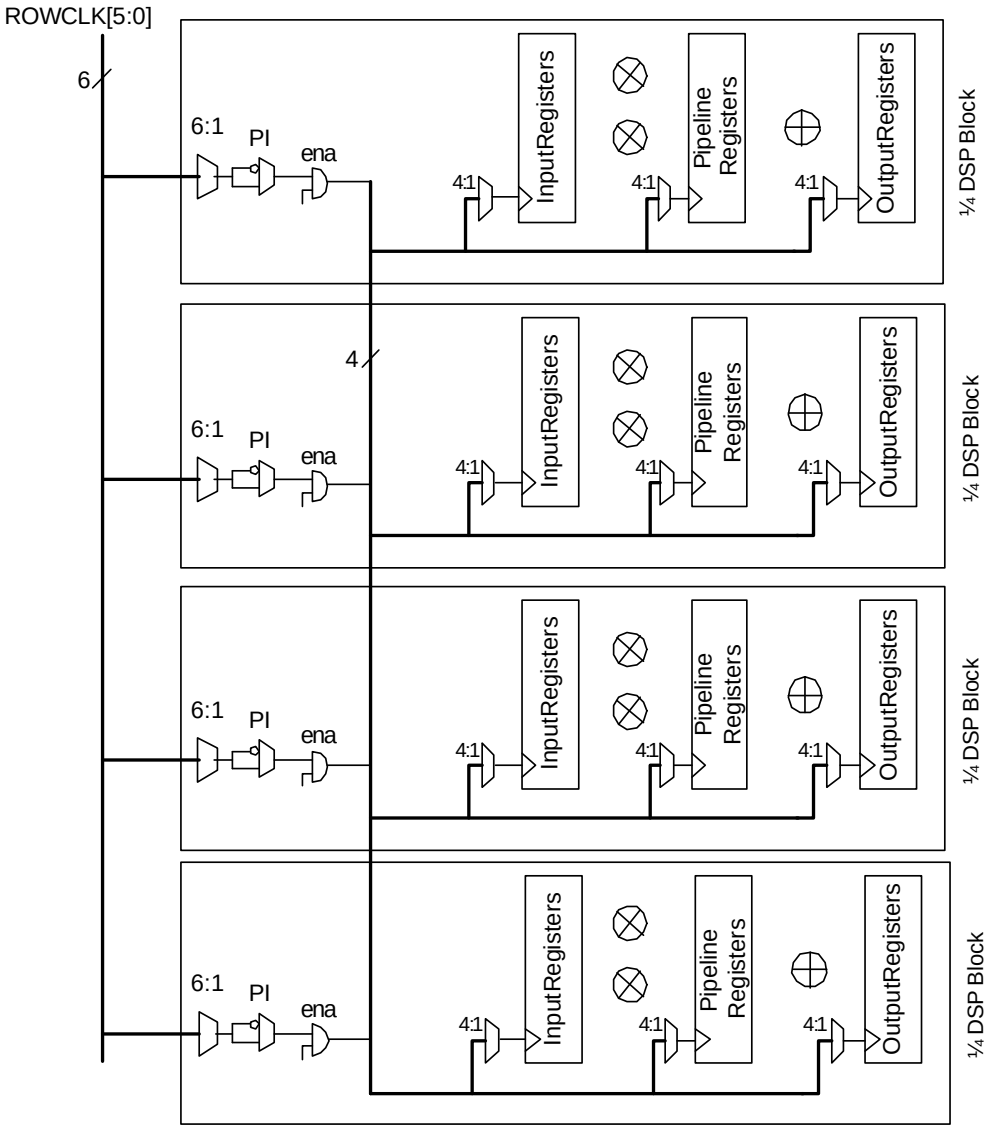

Figure 5: DSP Clocking Scheme

4.10 Location of Pipeline Register

In Stratix-II, the pipeline register was located after the multiplier. However in Stratix III, the pipeline register is located after the first-stage adder. This affects several areas including atom modeling, simulation, megafunctions, inferencing, register packing, etc.

The following diagrams show a comparison between Stratix-II and Stratix III.

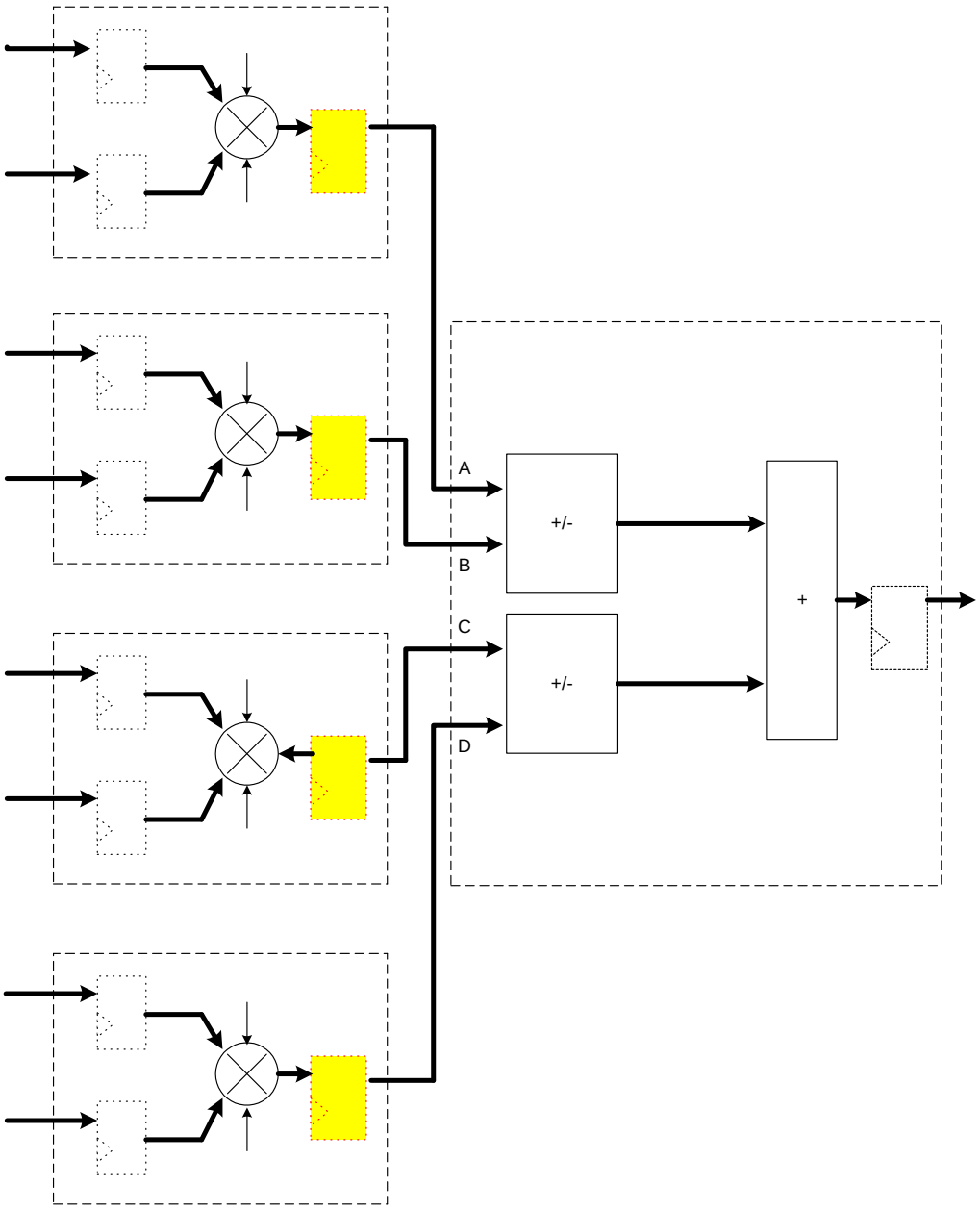

Figure 6: Stratix-II Pipeline Register Location

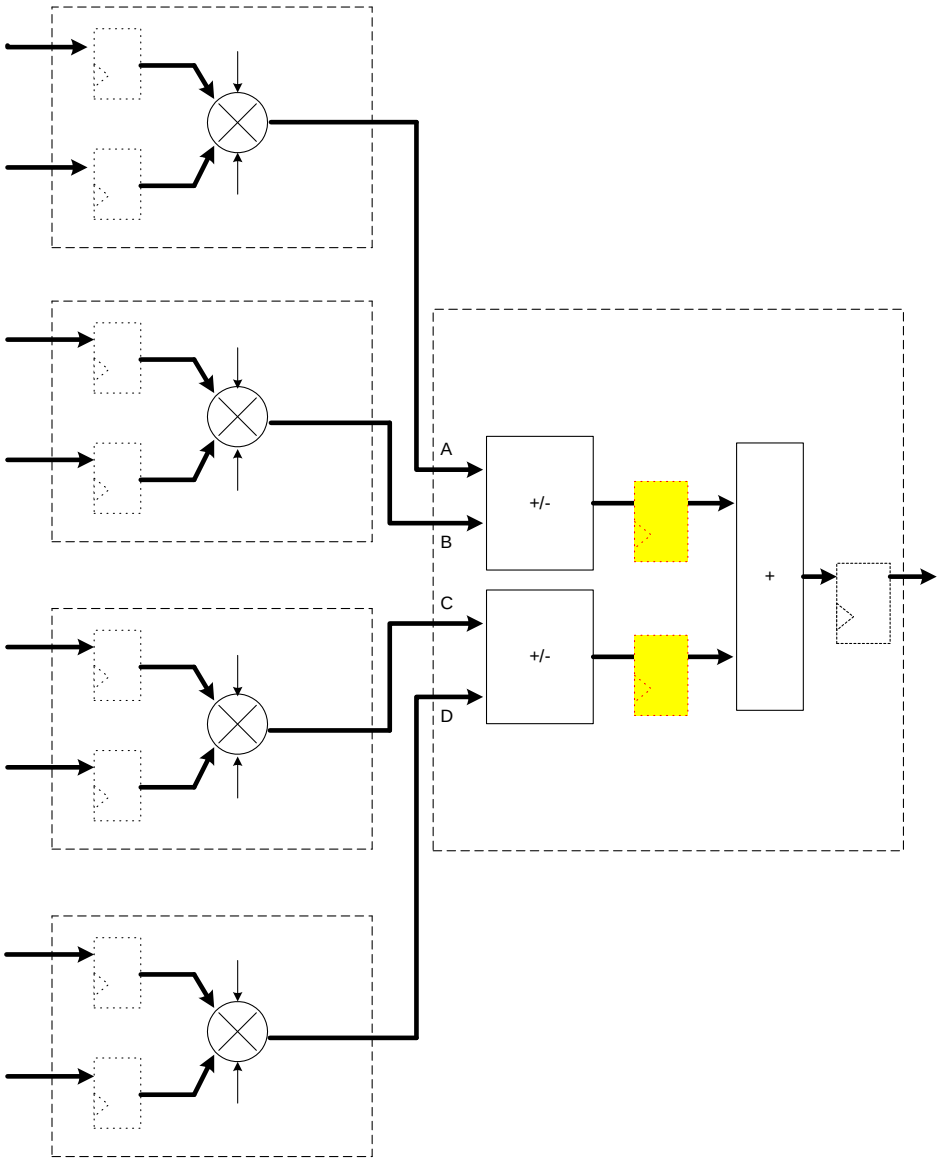

Figure 7: Stratix III Pipeline Register Location

4.11 Dynamic Mode

Unlike Stratix-II, there is no dynamic mode in Stratix III.

4.12 Shift Mode

The shift mode is intended for use by Nios in order to implement 32-bit left-shift, right-shift, and rotate operations. It is basically the same as the 36x36 multiply mode with the exception of the output mux which aligns the output to the same location. Control signals specify which operation is being performed, and the 32-bit output of the operation is always aligned to the same 32-bit location such that no additional external LE MUXes are necessary when switching between left/right/rotate cases.

The 32-bit input data must be 0-padded on its LSBs such that the MSBs are aligned (i.e. dataa[35..4] and datab[35..4] holds 32-bit data), and the 32-bit output is always at dataout[39..8]).

As with the 36-bit mode, this zero padding is the responsibility of the user (or the MegaFunction if one exists).

4.13 Swapping Operands in 36x36 Mode

The 36x36 mode is built out of four 18x18 multipliers just as in Stratix-II. However, in Stratix-II the 18-bit operands could not be swapped for a particular 18x18 multiplier since it was always fixed as to which operand was the 18 LSBs of the 36-bit operand, which implies that the operand must be unsigned (i.e. signx_internally_grounded = true). However, in Stratix III the two 18x18 multipliers that have mixed LSBs and MSBs (i.e. one is internally-grounded sign control while the other isn't) can swap which one is the internally grounded sign control.

This allows swapping larger width multipliers into the operand which consumes less power.

4.14 Sum of Two Precision

The sum-of-two-multiplier modes (i.e. one_level_adder and loopback modes) do not support full precision unsigned numbers.

Normally an 18x18 multiplier produces a 36-bit result, which when added with another 36-bit result would yield a 37-bit result. However, in Stratix III since there are not enough DSP block outputs, the sum-of-two modes only have a 36-bit wide output (the MSB is discarded and only the lower 36-bits of the 37-bit result is sent to the DSP outputs).

When doing signed multiplication, this still allows for full precision results. This is because it is not possible to get a sum-of-two signed case that will require the full 37-bits of output. Two 18-bit signed numbers multiplied together cannot produce the worst case 36-bit number, so when added with another similar 36-bit number, will not require the worst-case 37-bit result.

However for unsigned multiplication, a full precision 37-bit result is possible and so cannot be implemented in the Stratix III DSP block. Smaller width unsigned multiplication cases can be implemented, provided care is taken to align the input and outputs correctly. And of course, if the design does not use the full precision result, then unsigned multiplication can also be implemented.

5. User flow

5.1 Use and Instantiation

5.1.1 WYSIWYGs

See the Stratix III DSP WYSIWYG document for details. As with Stratix-II, two WYSIWYGs (mac_mult, mac_out) are used in various configurations to represent the different modes.

5.1.1.1 Default Values

In order to support the unified netlist specification, all ports and parameters must have a legal default value. In Stratix-II this was not always the case as in many instances, the connectivity of the port or usage of a parameter implied different conditions. As a result, some change will be necessary.

For example, the user should be able to connect all unused ports to their default value and still get the same fit as if the ports were not connected.

5.1.2 MegaFunctions and MegaWizards

Stratix III should continue to support the DSP MegaFunctions supported by Stratix-II: lpm_mult, altmult_add, altmult_accum, altfp_mult.

However, several changes will be necessary to the user interface of altmult_add and altmult_accum (though likely not lpm_mult and altfp_mult) in order to support the new Stratix III DSP features. And of course, changes to the implementation for all MegaFunctions.

Additionally, the MegaFunctions should support migrating designs from Stratix-II to Stratix III – specifically taking care of cases where the pipeline register is specified differently.

Refer to the comparison of the Stratix III DSP and Stratix-II DSP to determine the necessary changes. Highlights include four-multipliers feeding the accumulator instead of just one; changes to the accumulator loading functionality; changes to the clocking scheme and location of pipeline register; input shift registers on only one operand; a new adder chaining feature; loop-back mode.

The sum-of-two modes cannot yield full precision results in all cases, so the altmult_add megafunction is responsible for aligning the data appropriately for cases with smaller widths, or cases where the user does not want full precision results. See section 4.14 (Sum of Two Precision) for details.

The loopback mode will be implemented as a special case of the altmult_add megafunction.

Refer to the MegaFunction FD document for further details.

5.1.3 Inferencing from RTL

As with Stratix-II, DSP blocks should be inferred from RTL operators (i.e. '*' and '+' structures in Verilog and VHDL). Other new features that may be inferred include chain-out, round/saturation, loopback mode, rotate/shift mode, non-DSP shift registers, etc. Currently it is expected that the chain-out feature will be the most useful new feature to infer.

5.1.4 Backwards Compatibility

As many cases from Stratix-II (and Stratix) should compile in Stratix III as possible. But there are some cases that will not migrate without user intervention.

Megafunctions created for Stratix-II should compile in Stratix III, with the megafunction doing the appropriate implementation for Stratix III.

Stratix-II WYSIWYGs should also compile in Stratix III where possible.

5.1.5 Resource Balancing

Similar to Stratix-II, the DSP Resource Balancer for Stratix III is responsible for deciding whether to implement DSP structures in LEs versus DSP blocks, based on the availability of resources and the performance benefit.

Stratix III's clustering rules are slightly more complex than Stratix-II, which complicates the ability of the DSP Balancer to predict the number of DSP blocks that will be used in a design. As a result, the balancer may need to be aware of more than just the simplest DSP clustering restrictions, as it does currently.

The loop-back-multiplier mode is of concern since only 2 of them are available per DSP block (with 2 other regular two-mult-adder modes in the block). If implemented using the regular two-mult-adder mode, then 4 of them are available. The main difference between the two modes is that the loop-back mode has a higher performance, and the zeroloopback control signal. Consequently, the balancer should only use loop-back-mode when DSP resources are not a concern, or only for performance critical paths.

Some trade-off may also be needed for using the adder chain-out versus using external adders in LEs. In most cases it should be a benefit for using the chainout. But for very long combinatorial chains for example, it may be faster to use parallel adders in LEs. Further investigation is necessary.

5.2 Register Names

As with Stratix-II, register names are stored on observable oterm ports. The customer should be able to name these registers by naming the observable ports on their WYSIWYG. But as with Stratix-II, naming the internal register will not be possible from a MegaFunction, and cannot be named differently from the hierarchy it is in.

Register packed registers will retain their names as in Stratix-II, so that is one way for customers to name the DSP registers.

5.3 Compilation Report

As with Stratix-II, the compiler report will list a DSP summary as well as a DSP details section. The summary should list overall usage based on mode, and the details section should list individual DSP functions (or DSP slices)

5.3.1 Counting DSP Elements

In Stratix-II, the term “DSP element” is used to represent the smallest building block in a DSP block. In Stratix-II, this is the 9-bit pseudo multiplier, so there are 8 DSP elements per DSP block. Using this to count the usage of specific functions implemented in a DSP block, it is possible to compare usage between different modes as well describe total DSP usage.

For Stratix III this concept is more challenging because of the 12x12 multiplier, and because depending on the mode, some of the internal 18x18 multipliers are not usable.

We will use the same element counting methodology as Stratix-II, but each element represents the internal 18x18 hardware multiplier, which is the smallest usable element in the block. There are 8 18-bit multipliers per DSP block. To distinguish from Stratix-II which referred to these as “DSP 9-bit elements”, Stratix III can call them “DSP 18-bit elements”.

The element count per slice (i.e. per function or per atom group containing one MAC_OUT atom) can be calculated by dividing 8 by the maximum number of slices per block. Table 5 summarizes this for each of the modes.

For 12x12 multipliers a fractional value results (8 divided by 6 = 4/3). As a result, to avoid the user seeing fractional 18x18 DSP element usages in the report file, the element usage should be rounded up.

Table 5: DSP Element Counting per Mode

High-Level Mode	Multiplier In Width	MAC_OUT Mode	Max # of Slices per Block	# of DSP 18-bit Elements per Slice	Fractional Size of Slice in Block
Independent Multiplier	9-bits	output_only	8	1	1/8
	12-bits	output_only	6	4/3rd (1.333)	1/6
	18-bits	output_only	4	2	1/4
	36-bits	36_bit_multiply	2	4	1/2
Multiply-Accumulator	18-bits	accumulator	2	4	1/2
	18-bits	accumulator_chain_out	2	4	1/2
Multiply-Adder with loopback	18-bits	loopback	2 of 4	2	1/4
Multiply-Adder	18-bits	one_level_adder	4	2	1/4
	18-bits	two_level_adder	2	4	1/2
	18-bits	two_level_adder_chain_out	2	4	1/2
Shift	36-bits (32-bit data)	shift	2	4	1/2

5.4 Location Assignments

Stratix III DSP locations will be similar to Stratix-II, with the exception of the MAC_OUT locations. Since each half can operate fairly independently, instead of a single DSPOUT block for the entire DSP, there will be two.

As before, the lower left coordinate will be used for the coordinate of the entire block, and each multiplier will have a specific coordinate based on the row it is in. Additionally, as before, the user will not have to specify the sub-location for the MAC_OUT/DSPOUT since it is implied from the fact that it is MAC_OUT location.

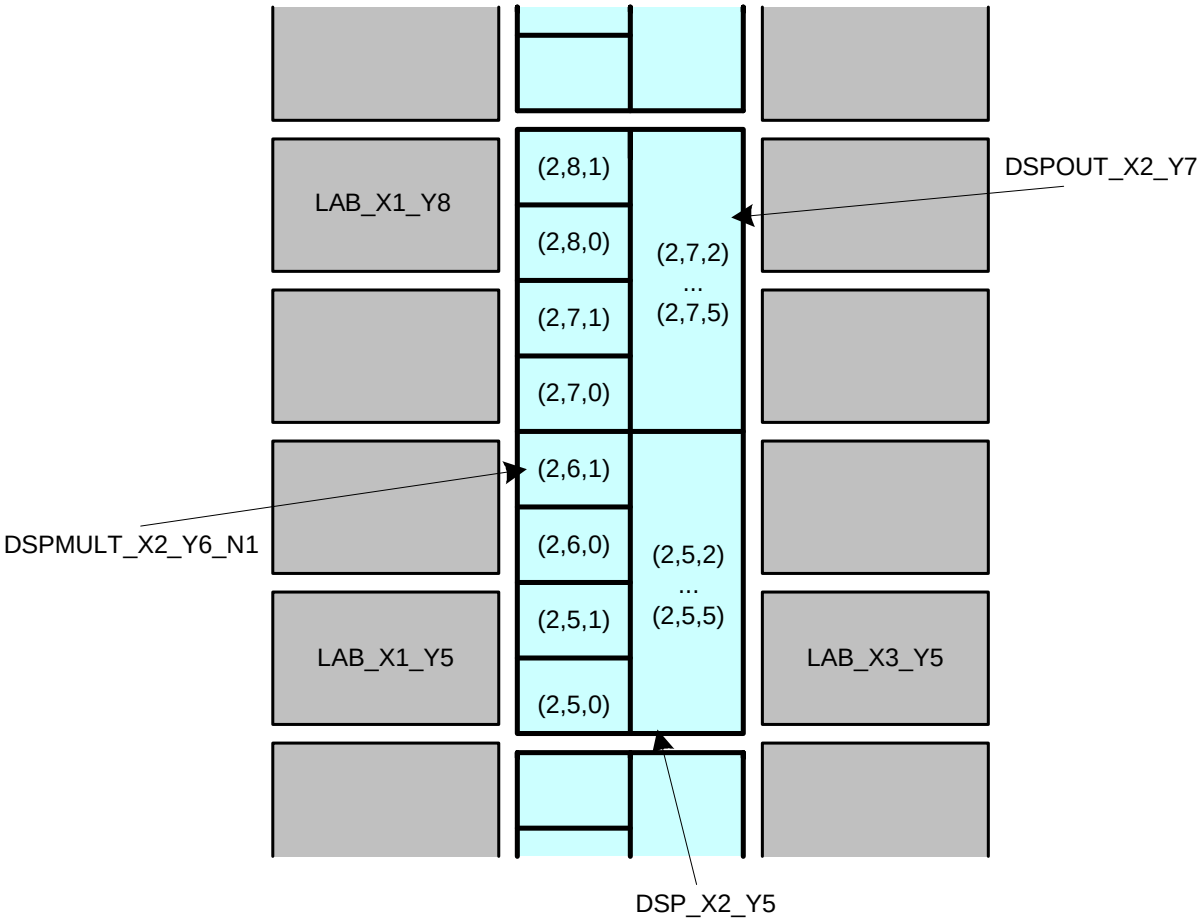

Figure 8: Stratix III DSP User Locations

As before, the internal fitter view of the locations (i.e. block list) will allow assigning each MAC_OUT atom to a different DSPOUT sub-location. The following diagram shows the generalized DSP coordinate scheme.

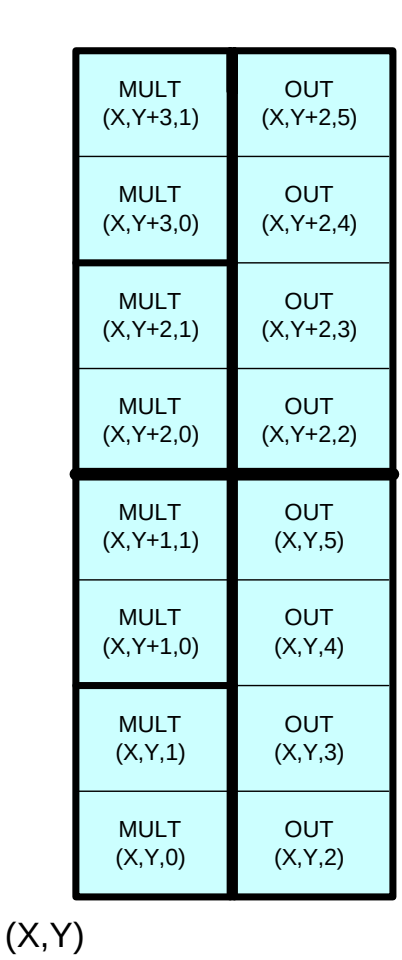

Figure 9: Stratix III DSP Internal Coordinate System

6. Fitter Processing

6.1 DSP Width Alignment

Same as in Stratix-II, this operation converts intermediate multiplier widths to natively supported multiplier widths by padding 0s to the LSBs (e.g. convert 15x15 multiplier to 18x18 by padding 3 GND bits to the LSBs and ignoring the lower 6 bits of the output).

Differences from Stratix III include:

1. Native support for 12x12 independent multipliers
2. No 9x9 summation modes (all are now 18x18 only modes)
3. Several new modes
4. Chainout mode (similar issues to input scan-chain)

6.2 DSP Register Packing

Similar to Stratix-II, the clocking scheme is more restrictive which affects what can be packed.

Input Shift register is available only on dataa and not on datab. If a register chain is seen on datab, then register packing should rotate dataa and datab in order to be able to implement the shift register in dataa. This is especially useful for RTL designs.

There is a fourth set of registers at the output when using chainout modes. There is also an extra register in the input shift-register path when using chainout modes.

6.3 DSP Power Optimizations

Similar to Stratix-II, multiplier operands should be swapped so that smaller width operands use either dataa or datab, depending on which operand consumes less power. Unlike Stratix-II, Stratix III DSPs also allow swapping 18-bit operands of the 36x36 multiplier since the signa/b_internally_grounded option is now programmable.

Unused multiplier inputs should be tied to GND to avoid unnecessary toggling.

7. Power Optimizations and Analysis

The following power optimization cases should be considered:

1. Smaller width operand on one of dataa or datab depending on which consumes less power (pre-fitter operation swaps smaller width operands as appropriate)
2. Unused adder logic can be gated, thus reducing power (Assembler must set appropriate CRAMs)
3. Rounding/Saturation logic cannot be gated
4. Multiplier operands are usually padded with 0s on the LSBs, thus less logic is toggling
5. Unused multiplier inputs should be tied to GND to ensure no toggling and correct operation (Assembler must set appropriate CRAMs)
6. scanin/chain inputs must not be selected when not used (Assembler must set appropriate CRAMs)
7. For unused registers, set clock to one that is unused and grounded (Assembler must set appropriate CRAMs)