Stratix III DSP Megafunction EDA Functional Description

Version 1.1

Nov 19 2008

by

Altera Corporation

Table of Contents:

1. Overview

This documentation includes the function definition of two megafunctions: ALTMULT_ADD and ALTMULT_ACCUM. For stratixIII, altmult_add and altmult_accum share the same ports and parameters.

2. Glossary

Cbx – clearbox.

DSP block – Digital Signal processing block.

sum of 2 mode - two multipliers followed by an adder.

output_only – simply multiplier.

one_level_adder – same as sum of 2 mode.

two_level_adder – four multipliers followed two level adder.

accumulator – multipliers followed by adder then by accumulator.

accum_chainout – output of accumulator is added with chainin data.

loopback – one input of a one_level_adder is connected to the output of this block.

two_level_adder_chainout – the output of two_level_adder is added with chainin data.

36_bit – an independent multiplier with 36 bit inputs.

Shift – multiplier is used as a shift. Inputs and output are 32 bits.

Double – special mode used to deal complex multiplication.

3. Changes in the Mega functions:

3.1 altmult_add Changes

This is the multiply-add megafunction, it also include multiply-add-accumulate megafunction. From a behavioral perspective, it consists of 1 or more multipliers feeding a parallel adder. The user specifies widths and registering options. If the widths exceed those that can be supported by Stratixiii hardware, the megafunction will add logic as needed. The main exception is that if the rounding or saturation features are used, support is limited to that available in native DSP blocks.

The result of parallel adder can also be fed into an accumulator. If this feature is selected, support is also limited to implementation in native DSP blocks for Stratixiii Device family.

The Stratixiii DSP block has more features than the Stratix-II DSP block, but it is not a complete superset of the Stratix-II DSP block.

StratixIV and ArriaII device family have the same DSP Block as Stratixiii device family; the altmult_add megafunction for them is identical as Stratixiii.

New features related to altmult_add include:

1. loopback input for sum of 2 mode,
2. enhanced saturation and rounding mode,
3. accumulator followed the adder
4. the ability to chain adders/accumlator between multiple blocks to form larger adders and dynamic shift mode.
5. Shift mode that can dynamically change between left, right, rotation, logic and arithmetic.

Some of the features that were supported in Stratix II but no longer supported in Stratixiii are:

1. scanin input mode changed from two input ports support to one input port support and only for sum of 4 in 18 by 18 modes.
2. Input shift register parallel load is removed.
3. dynamic input source selection is removed
4. dynamic mode support only available in shift mode
5. pipeline register position is moved
6. each quadrant has 4 register banks: at input, pipeline, output, chainout and each register can select between 4 clock and 4 aclr signals. The corresponding register of input signal in each register bank should be identical or skipped.
7. Dynamic add/sub at the first level adder is removed.

3.1.1 Parameter and port list

```
PARAMETERS
(
    NUMBER_OF_MULTIPLIERS,
    WIDTH_A,
    WIDTH_B,
    WIDTH_RESULT,
    WIDTH_CHAININ,                                  -- Stratixiii-only!

    -- A MULTIPLIER INPUTS
    -- DATAA and DATAB should use the same CLOCK and ACLR resource for Stratixiii
    -- Multipliers in each quadrant should share the same clock and aclr signals.
    INPUT_ACLR_A0,
    INPUT_SOURCE_A0 = "DATAA",                      -- need update in Stratixiii
    INPUT_REGISTER_A1,
    INPUT_ACLR_A1,
    INPUT_SOURCE_A1 = "DATAA",                      -- need update in Stratixiii
    INPUT_REGISTER_A2,
    INPUT_ACLR_A2,
    INPUT_SOURCE_A2 = "DATAA",                      -- need update in Stratixiii
    INPUT_REGISTER_A3,
    INPUT_ACLR_A3,
    INPUT_SOURCE_A3 = "DATAA",                      -- need update in Stratixiii
    REPRESENTATION_A= "UNUSED",
    SIGNED_REGISTER_A,
    SIGNED_ACLR_A,
    SIGNED_PIPELINE_REGISTER_A,
    SIGNED_PIPELINE_ACLR_A,
    SCANOUTA_REGISTER=" UNREGISTERED",              -- Stratixiii-only!
    SCANOUTA_ACLR="NONE",                           -- Stratixiii-only!
```

```
-- B MULTIPLIER INPUTS
INPUT_REGISTER_B0,
INPUT_ACLR_B0,
INPUT_SOURCE_B0 = “DATAB”,                    -- “DATAB” only for Stratixiii
INPUT_REGISTER_B1,
INPUT_ACLR_B1,
INPUT_SOURCE_B1 = “DATAB”,                    -- “DATAB” only for Stratixiii
INPUT_REGISTER_B2,
INPUT_ACLR_B2,
INPUT_SOURCE_B2 = “DATAB”,                    -- “DATAB” only for Stratixiii
INPUT_REGISTER_B3,
INPUT_ACLR_B3,
INPUT_SOURCE_B3 = “DATAB”,                    -- “DATAB” only for Stratixiii
REPRESENTATION_B = “UNUSED”,
SIGNED_REGISTER_B,
SIGNED_ACLR_B,
SIGNED_PIPELINE_REGISTER_B,
SIGNED_PIPELINE_ACLR_B,

-- MULTIPLIER OUTPUTS: Used as pipeline registers. Each quadrant should share the
same clock and aclr signals
MULTIPLIER_REGISTER0=” UNREGISTERED”,
MULTIPLIER_ACLR0=“NONE”,
MULTIPLIER_REGISTER1=“CLOCK0”,
MULTIPLIER_ACLR1=“NONE”,
MULTIPLIER_REGISTER2=“CLOCK0”,
MULTIPLIER_ACLR2=“NONE”,
MULTIPLIER_REGISTER3=“CLOCK0”,
MULTIPLIER_ACLR3=“NONE”,
MULTIPLIER1_DIRECTION=”ADD”,
MULTIPLIER3_DIRECTION=”ADD”,
ACCUM_DIRECTION=”ADD”,                                  -- Stratixiii-only!

-- OUTPUT
OUTPUT_REGISTER=“CLOCK0”,
OUTPUT_ACLR=“NONE”,
CHAINOUT_REGISTER=“CLOCK0”,                   -- Stratixiii-only!
CHAINOUT _ACLR=“NONE”,                                  -- Stratixiii-only!

-- ROUND SATURATION CONTROL SIGNAL IN Stratixiii is different to Stratix II.
OUTPUT_ROUNDING = “NO”,                                 --Stratixiii-only!
OUTPUT_ROUND_TYPE = "NEAREST_INTEGER",                  --Stratixiii-only!
WIDTH_MSB = 10,                               --Stratixiii-only!
OUTPUT_ROUND_REGISTER=“CLOCK0”,               --Stratixiii-only!
OUTPUT_ROUND_ACLR=“NONE”,                               --Stratixiii-only!
OUTPUT_ROUND_PIPELINE_REGISTER=“CLOCK0”,      --Stratixiii-only!
OUTPUT_ROUND_PIPELINE_ACLR=“NONE”,                      --Stratixiii-only!

OUTPUT_SATURATION = “NO”,                               --Stratixiii-only!
OUTPUT_SATURATE_TYPE = "ASYMMETRIC",                    --Stratixiii-only!
WIDTH_SATURATE_SIGN = 1,                      --Stratixiii-only!
OUTPUT_SATURATE_REGISTER=“CLOCK0”,            --Stratixiii-only!
OUTPUT_SATURATE_ACLR=“NONE”,                            --Stratixiii-only!
OUTPUT_SATURATE_PIPELINE_REGISTER=“CLOCK0”,   --Stratixiii-only!
OUTPUT_SATURATE_PIPELINE_ACLR=“NONE”,                   --Stratixiii-only!
```

```
CHAINOUT_ROUNDING = “NO”,                                   --Stratixiii-only!
CHAINOUT_ROUND_REGISTER=“CLOCK0”,                  --Stratixiii-only!
CHAINOUT_ROUND_ACLR=“NONE”,                                 --Stratixiii-only!
CHAINOUT_ROUND_PIPELINE_REGISTER=“CLOCK0”,    --Stratixiii-only!
CHAINOUT_ROUND_PIPELINE_ACLR=“NONE”,                        --Stratixiii-only!
CHAINOUT_ROUND_OUTPUT_REGISTER=“CLOCK0”,      --Stratixiii-only!
CHAINOUT_ROUND_OUTPUT_ACLR=“NONE”,                          --Stratixiii-only!
CHAINOUT_SATURATION = “NO”,                                 --Stratixiii-only!
CHAINOUT_SATURATE_REGISTER=“CLOCK0”,               --Stratixiii-only!
CHAINOUT_SATURATE_ACLR=“NONE”,                              --Stratixiii-only!
CHAINOUT_SATURATE_PIPELINE_REGISTER=“CLOCK0”,--Stratixiii-only!
CHAINOUT_SATURATE_PIPELINE_ACLR=“NONE”,                      --Stratixiii-only!
CHAINOUT_SATURATE_OUTPUT_REGISTER=“CLOCK0”,--Stratixiii-only!
CHAINOUT_SATURATE_OUTPUT_ACLR=“NONE”,                       --Stratixiii-only!

ACCUMMULATOR="NO",                                          --Stratixiii-only!
CHAINOUT_ADDER="NO",                                        --Stratixiii-only!
ZERO_CHAINOUT_OUTPUT_REGISTER=“CLOCK0”,        --Stratixiii-only!
ZERO_CHAINOUT_OUTPUT_ACLR=“NONE”,                           --Stratixiii-only!
ZERO_LOOPBACK_REGISTER=“CLOCK0”,                   --Stratixiii-only!
ZERO_LOOPBACK_ACLR=“NONE”,                                  --Stratixiii-only!
ZERO_LOOPBACK_PIPELINE_REGISTER=“CLOCK0”,      --Stratixiii-only!
ZERO_LOOPBACK_PIPELINE_ACLR=“NONE”,                         --Stratixiii-only!
ZERO_LOOPBACK_OUTPUT_REGISTER=“CLOCK0”,        --Stratixiii-only!
ZERO_LOOPBACK_OUTPUT_ACLR=“NONE”,                           --Stratixiii-only!

ACCUM_SLOAD_REGISTER=“CLOCK0”,                     --Stratixiii-only!
ACCUM_SLOAD_ACLR=“NONE”,                                    --Stratixiii-only!
ACCUM_SLOAD_PIPELINE_REGISTER=“CLOCK0”,        --Stratixiii-only!
ACCUM_SLOAD_PIPELINE_ACLR=“NONE”,                           --Stratixiii-only!

SHIFT_MODE ="NO",                                           --Stratixiii-only!
ROTATE_REGISTER=“CLOCK0”,                          --Stratixiii-only!
ROTATE_ACLR=“NONE”,                                         --Stratixiii-only!
ROTATE_PIPELINE_REGISTER=“CLOCK0”,                 --Stratixiii-only!
ROTATE_PIPELINE_ACLR=“NONE”,                                --Stratixiii-only!
ROTATE_OUTPUT_REGISTER=“CLOCK0”,                   --Stratixiii-only!
ROTATE_OUTPUT_ACLR=“NONE”,                                  --Stratixiii-only!
SHIFT_RIGHT_REGISTER=“CLOCK0”,                     --Stratixiii-only!
SHIFT_RIGHT_ACLR=“NONE”,                                    --Stratixiii-only!
SHIFT_RIGHT_PIPELINE_REGISTER=“CLOCK0”,  --Stratixiii-only!
SHIFT_RIGHT_PIPELINE_ACLR=“NONE”,                           --Stratixiii-only!
SHIFT_RIGHT_OUTPUT_REGISTER=“CLOCK0”,              --Stratixiii-only!
SHIFT_RIGHT_OUTPUT_ACLR=“NONE”,                             --Stratixiii-only!
PORT_OUTPUT_IS_OVERFLOW= “PORT_UNUSED”,                     --Stratixiii-only!
PORT_CHAINOUT_SAT_IS_OVERFLOW = “PORT_UNUSED”,              --
Stratixiii-only!


EXTRA_LATENCY,
DEDICATED_MULTIPLIER_CIRCUITRY = “AUTO”  -- no behavioral impact

DEVICE_FAMILY = “Stratix”,                          -- default to Stratix/Stratix II behavior

PORT_SIGNA,
PORT_SIGNB,
```

```
    -- Parameters keep for backward compatability
    MULTIPLIER01_SATURATION = “NO”,
    MULT01_SATURATION_REGISTER =“CLOCK0”,
    MULT01_SATURATION_ACLR =”NONE”,
    MULTIPLIER23_SATURATION = “NO”,
    MULT23_SATURATION_REGISTER=“CLOCK0”,
    MULT23_SATURATION_ACLR=”NONE”,
    ADDER1_ROUNDING = “NO”,
    ADDNSUB1_ROUND_REGISTER=“CLOCK0”,
    ADDNSUB1_ROUND_ACLR=”NONE”,
    ADDNSUB1_ROUND_PIPELINE_REGISTER=“CLOCK0”,
    ADDNSUB1_ROUND_PIPELINE_ACLR=”NONE”,
    ADDER3_ROUNDING = “NO”,
    ADDNSUB3_ROUND_REGISTER=“CLOCK0”,
    ADDNSUB3_ROUND_ACLR=”NONE”,
    ADDNSUB3_ROUND_PIPELINE_REGISTER=“CLOCK0”,
    ADDNSUB3_ROUND_PIPELINE_ACLR=”NONE”,
    MULTIPLIER01_ROUNDING = “NO”,
    MULT01_ROUND_REGISTER=“CLOCK0”,
    MULT01_ROUND_ACLR=”NONE”,
    MULTIPLIER23_ROUNDING = “NO”,
    MULT23_ROUND_REGISTER=“CLOCK0”,
    MULT23_ROUND_ACLR=”NONE”,

    -- Parameters not valid in Stratixiii (default value will disable the function in Stratixiii)
    PORT_ADDNSUB1= “UNUSED”,
    PORT_ADDNSUB3= “UNUSED”,
    PORT_MULT0_IS_SATURATED = “UNUSED”,
    PORT_MULT1_IS_SATURATED = “UNUSED”,
    PORT_MULT2_IS_SATURATED = “UNUSED”,
    PORT_MULT3_IS_SATURATED = “UNUSED”,
    ADDNSUB_MULTIPLIER_REGISTER1 =“CLOCK0”,
    ADDNSUB_MULTIPLIER_ACLR1 =”NONE”,
    ADDNSUB_MULTIPLIER_PIPELINE_REGISTER1=“CLOCK0”,
    ADDNSUB_MULTIPLIER_PIPELINE_ACLR1=”NONE”,
    ADDNSUB_MULTIPLIER_REGISTER3 =“CLOCK0”,
    ADDNSUB_MULTIPLIER_ACLR3 =”NONE”,
    ADDNSUB_MULTIPLIER_PIPELINE_REGISTER3=“CLOCK0”,
    ADDNSUB_MULTIPLIER_PIPELINE_ACLR3=”NONE”,
)
SUBDESIGN ALTMULT_ADD
(
    dataa[NUMBER_OF_MULTIPLIERS * WIDTH_A – 1..0]      : INPUT;
    datab[NUMBER_OF_MULTIPLIERS * WIDTH_B – 1..0]      : INPUT;

    clock3, clock2, clock1, clock0                     : INPUT = VCC;
    aclr3, aclr2, aclr1, aclr0                         : INPUT = GND;
    ena3, ena2, ena1, ena0                             : INPUT = VCC;

    signa, signb                                       : INPUT = GND;

    -- input ports new in Stratixiii
    output_saturate                                    : INPUT = GND;      -- Stratixiii-only!
    output_round                                       : INPUT = GND;      -- Stratixiii-only!
```

```
    chainout_saturate                       : INPUT = GND;      -- Stratixiii-only!
    chainout_round                          : INPUT = GND;      -- Stratixiii-only!
    zero_chainout                           : INPUT = GND;      -- Stratixiii-only!
    zero_loopback                           : INPUT = GND;      -- Stratixiii-only!
    accum_sload                             : INPUT = GND;      -- Stratixiii-only!
    chainin[WIDTH_CHAININ - 1..0]           : INPUT = GND;      -- Stratixiii-only!
    rotate                                  : INPUT = GND;      -- Stratixiii-only!
    shift_right                             : INPUT = GND;      -- Stratixiii-only!

    --Input port reserved for stratixii backward compatible (It will convert the parameters to new
    parameters. Just keep the old port name)
    mult01_round                            : INPUT = GND;
    mult23_round                            : INPUT = GND;
    addnsub1_round                          : INPUT = GND;
    addnsub3_round                          : INPUT = GND;
    mult01_saturation, mult23_saturation    : INPUT = GND;

    -- input ports not valid in Stratixiii (default value will disable the function in Stratixiii)
    addnsub1, addnsub3                      : INPUT = GND;
    sourcea[NUMBER_OF_MULTIPLIERS-1..0]     : INPUT = GND;
    sourceb[NUMBER_OF_MULTIPLIERS-1..0]     : INPUT = GND;

    scanina and scaninb         can not be used simultaneously;


    result[WIDTH_RESULT – 1..0]             : OUTPUT;

    -- output ports new in Stratixiii
    overflow                                : OUTPUT;       -- Stratixiii-only!
    Chainout_sat_overflow                   : OUTPUT;       -- Stratixiii-only!

    -- output ports not valid in Stratixiii
    scanina and scaninb         can not be used simultaneously;

    mult0_is_saturated                      : OUTPUT;
    mult1_is_saturated                      : OUTPUT;
    mult2_is_saturated                      : OUTPUT;
    mult3_is_saturated                      : OUTPUT;
)
```

3.1.2 Parameter definitions

NUMBER_OF_MULTIPLIERS: The number of multiplier which is to be added together. This value must be 1 to 4.

WIDTH_A: The width of the dataa input busses.

WIDTH_B: The width of the datab input busses.

WIDTH_RESULT: Width of the result output bus includes all bits before rounding and saturation.

WIDTH_CHAININ: Width of the CHAININ bus. It's a new parameter in Stratixiii. WIDTH_CHAININ = WIDTH_RESULT if port chainin is used. The value of width_chainin should be 44. Default value is 1 for simulation model when port chainin is not used.

Each quadrant of DSP block has 4 register banks: input, pipeline, output, chainout and each register can select between 4 clocks and 4 aclr signals. All the *INPUT_REGISTER* in the same

register bank should have the same setting or you can skip any of them. All the *INPUT_ACLR* should have the same setting and flow the usage of corresponding register.

INPUT_ACLR default value is "ACLR3" if the corresponding register is been used and the "aclr" value is not been set. If the corresponding register is not registered, "aclr" signal should be set to "NONE" and it won't be passed to wysiwyg.

INPUT_REGISTER_A0: Clock source for the A inputs of the first multiplier. Legal values are "UNREGISTERED", "CLOCK0", "CLOCK1", "CLOCK2", and "CLOCK3". When input dataa forms scan chain, this register and followed input_register can not be skipped.

INPUT_ACLR_A0: Asynchronous clear source for the A inputs of the first multiplier. Legal values are "ACLR0", "ACLR1", "ACLR2", and "ACLR3". Default value is "ACLR3" if the corresponding register is been used and the aclr value is not been set. If the corresponding register is not registered, aclr signal should be set to "NONE" and it won't be passed to wysiwyg.

INPUT_SOURCE_A0: Specifies the source of the A inputs to the first multiplier. Legal values are "DATAA", "SCANA" and "VARIABLE". The value of "VARIABLE" is supported in Stratix II devices only.

INPUT_REGISTER_A1: Clock source for the A inputs to the second multiplier. Legal values are "UNREGISTERED", "CLOCK0", "CLOCK1", "CLOCK2", and "CLOCK3". The value should follow *INPUT_REGISTER_A0* or set as "UNREGISTERED".

INPUT_ACLR_A1: Asynchronous clear source for the A inputs to the second multiplier. Legal values are "ACLR0", "ACLR1", "ACLR2", and "ACLR3". Default value is "ACLR3" if the corresponding register is been used and the aclr value is not been set. If the corresponding register is not registered, aclr signal should be set to "NONE" and it won't be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

INPUT_SOURCE_A1: Specifies the source for the A inputs to the second multiplier. Legal values are "DATAA", "SCANA", and "VARIABLE". The value of "VARIABLE" is supported in Stratix II devices only.

INPUT_REGISTER_A2: Clock source for the A inputs to the third multiplier. Legal values are "UNREGISTERED", "CLOCK0", "CLOCK1", "CLOCK2", and "CLOCK3". The value should follow *INPUT_REGISTER_A0* or set as "UNREGISTERED".

INPUT_ACLR_A2: Asynchronous clear source the A inputs to the third multiplier. Legal values are "ACLR0", "ACLR1", "ACLR2", and "ACLR3". Default value is "ACLR3" if the corresponding register is been used and the aclr value is not been set. If the corresponding register is not registered, aclr signal should be set to "NONE" and it won't be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

INPUT_SOURCE_A2: Specifies the source for the A inputs to the third multiplier. Legal values are "DATAA", "SCANA", and "VARIABLE". The value of "VARIABLE" is supported in Stratix II devices only.

INPUT_REGISTER_A3: Clock source for the fourth and all subsequent multipliers. Legal values are "UNREGISTERED", "CLOCK0", "CLOCK1", "CLOCK2", and "CLOCK3". The value should follow *INPUT_REGISTER_A0* or set as "UNREGISTERED".

INPUT_ACLR_A3: Asynchronous clear source for the fourth and all subsequent multipliers. Legal values are "ACLR0", "ACLR1", "ACLR2", and "ACLR3". Default value is "ACLR3" if the corresponding register is been used and the aclr value is not been set. If the corresponding register is not registered, aclr signal should be set to "NONE" and it won't be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

INPUT_SOURCE_A3: Specifies the source of the A inputs to the fourth multiplier. Legal values are "DATAA", "SCANA", and "VARIABLE". The value of "VARIABLE" is supported in Stratix II devices only.

REPRESENTATION_A: For specifying the number representation of the A multiplier inputs. Legal values are “UNSIGNED”, “SIGNED”, and “UNUSED”. A value of “UNSIGNED” causes the A inputs to be interpreted as unsigned numbers. A value of “SIGNED” causes the A inputs to be interpreted as signed two’s complement numbers. A value of “UNUSED” allows for dynamic control of the representation through the signa port. In shift mode, REPRESENTATION_A = REPRESENTATION_B = “SIGNED” will make shift type as “ARITHMETIC”. REPRESENTATION_A = REPRESENTATION_B = “UNSIGNED” will make shift type as “LOGICAL”. REPRESENTATION_A = REPRESENTATION_B = “UNUSED” will allow dynamic shift type selection.

SIGNED_REGISTER_A: The clock source for the first signa register. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

SIGNED_ACLR_A: Asynchronous clear source for the first signa register. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used and the aclr value is not been set. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

SIGNED_PIPELINE_REGISTER_A: The clock source for the second signa register. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

SIGNED_PIPELINE_ACLR_A: Asynchronous clear source for the second signa register. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used and the aclr value is not been set. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set.
SCANOUTA_REGISTER: The clock source for the scanouta data bus registers. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii.

SCANOUTA_ACLR: Asynchronous clear source for the scanouta data bus registers Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. It’s a new parameter in Stratixiii.

INPUT_REGISTER_B0: Clock source for the B inputs of the first multiplier. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

INPUT_ACLR_B0: Asynchronous clear source for the B inputs of the first multiplier. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

INPUT_SOURCE_B0: Specifies the source of the B inputs to the first multiplier. Legal values are “DATAB”, “SCANB”, “LOOPBACK” and “VARIABLE”. The value “VARIABLE” is supported in Stratix II devices only. The value of “LOOPBACK” is supported in Stratixiii devices only (and only available for sum2 mode).

INPUT_REGISTER_B1: Clock source for the B inputs to the second multiplier. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

INPUT_ACLR_B1: Asynchronous clear source for the B inputs to the second multiplier. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal

should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

INPUT_SOURCE_B1: Specifies the source for the B inputs to the second multiplier. Legal values are “DATAB”, “SCANB”, and “VARIABLE”. The value of “VARIABLE” is supported in Stratix II devices only.

INPUT_REGISTER_B2: Clock source for the B inputs to the third multiplier. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

INPUT_ACLR_B2: Asynchronous clear source for the B inputs to the third multiplier Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

INPUT_SOURCE_B2: Specifies the source for the B inputs to the third multiplier. Legal values are “DATAB”, “SCANB”, and “VARIABLE”. The value of “VARIABLE” is supported in Stratix II devices only.

INPUT_REGISTER_B3: Clock source for the fourth and all subsequent multipliers. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

INPUT_ACLR_B3: Asynchronous clear source for the fourth and all subsequest multipliers. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

INPUT_SOURCE_B3: Specifies the source of the B inputs to the fourth multiplier. Legal values are “DATAB”, “SCANB”, and “VARIABLE”. The value of “VARIABLE” is supported in Stratix II devices only.

Stratixiii support one input scan chain. It can be scana or scanb. You can not use both.

REPRESENTATION_B: For specifying the number representation of the B multiplier inputs. Legal values are “USNIGNED”, “SIGNED”, and “UNUSED”. A value of “UNSIGNED” causes the B inputs to be interpreted as unsigned numbers. A value of “SIGNED” causes the B inputs to be interpreted as signed two’s complement numbers. A value of “UNUSED” allows for dynamic control of the representation through the signb port. In shift mode, REPRESENTATION_A = REPRESENTATION_B = “SIGNED” will make shift type as “ARITHMETIC”. REPRESENTATION_A = REPRESENTATION_B = “UNSIGNED” will make shift type as “LOGICAL”. REPRESENTATION_A = REPRESENTATION_B = “UNUSED” will allow dynamic shift type selection.

SIGNED_REGISTER_B: The clock source for the first signb register. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

SIGNED_ACLR_B: Asynchronous clear source for the first signb register. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set.

SIGNED_PIPELINE_REGISTER_B: The clock source for the second signb register. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

SIGNED_PIPELINE_ACLR_B: Asynchronous clear source for the second signb register. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set.

All the *MULTIPLIER_REGISTER* should have the same setting or “unregistered”. All the *MULTIPLIER_ACLR* should have the same setting or follow the corresponding register.

MULTIPLIER_REGISTER0: Clock source for the register immediately after the first adder (it shares the same name as in StratixII but been moved to the end of first adder stage). Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. Used as PIPELINE Register in Stratixiii.

MULTIPLIER_ACLR0: Asynchronous clear source for the register immediately after the first adder (it shares the same name as in Stratix II but been moved to the end of first adder stage). Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg.

MULTIPLIER_REGISTER1: Clock source for the register immediately after the first adder (it shares the same name as in StratixII but been moved to the end of first adder stage). Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. Used as PIPELINE Register in Stratixiii. Legal value should be the same as MULTIPLIER_REGISTER0 in Stratixiii.

MULTIPLIER_ACLR1: Asynchronous clear source for the register immediately after the first adder (it shares the same name as in StratixII but been moved to the end of first adder stage). Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set.

MULTIPLIER_REGISTER2: Clock source for the register immediately after the first adder (it shares the same name as in StratixII but been moved to the end of first adder stage). Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. Used as PIPELINE Register in Stratixiii. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

MULTIPLIER_ACLR2: Asynchronous clear source for the register immediately after the first adder (it shares the same name as in Stratix II but been moved to the end of first adder stage). Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set.

MULTIPLIER_REGISTER3: Clock source for the register immediately after the first adder (it shares the same name as in StratixII but been moved to the end of first adder stage). Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. Used as PIPELINE Register in Stratixiii. Legal value should be the same as MULTIPLIER_REGISTER2 in Stratixiii.

MULTIPLIER_ACLR3: Asynchronous clear source for the register immediately after the first adder (it shares the same name as in Stratix II but been moved to the end of first adder stage).

Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set.

In Stratixiii MULTIPLIER_REGISTER0/ MULTIPLIER_REGISTER1 is used as first_adder_register0,

MULTIPLIER_REGISTER2/ MULTIPLIER_REGISTER3 is used as first_adder_register1.

MULTIPLIER1_DIRECTION: Specifies whether the second multiplier will add or subtract its value from the sum. Legal values are “ADD” and “SUB”. Default value is “ADD”. StratixII and Stratix device family support dynamic add/sub when port addnsub1 is used. Stratixiii does not support dynamic add/sub.

MULTIPLIER3_DIRECTION: Specifies whether the second multiplier will add or subtract its value from the sum. Legal values are “ADD” and “SUB”. Default value is “ADD”. StratixII and Stratix device family support dynamic add/sub when port addnsub3 is used. Stratixiii does not support dynamic add/sub.

ACCUMULATOR: Controls accumulator mode of the final adder stage. Legal values are “YES” and “NO”. Default value is “NO” for backward compatibility for Stratix II. It’s a new parameter in Stratixiii.

ACCUM_DIRECTION: Specifies whether the accumulator will add or subtract its value from the sum. Legal values are “ADD” and “SUB”. Default value is “ADD”. It’s a new parameter for altmult_add. When input data are unsigned, “SUB” is not valid.

OUTPUT_REGISTER: Clock source for the output register. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”.

OUTPUT_ACLR: Asynchronous clear source for the output register. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg.

CHAINOUT_REGISTER: Clock source for the chainout mode result register which is an additional stage after the second adder. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii.

CHAINOUT_ACLR: Asynchronous clear for the chainout mode result register which is an additional stage after the second adder. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. . It’s a new parameter in Stratixiii.

EXTRA_LATENCY: Specifies latency to add to the circuit in addition to any registers which were already specified. The register clock and aclr follows the last stage register before the extra_latency applied. The parameter only available in stratixiii device family and only when implementation can be put in one DSP block.

DEDICATED_MULTIPLIER_CIRCUITRY: Specifies whether or not to use the DSP block to implement the circuit. Legal values are “YES”, “NO”, and “AUTO”. A value of “YES” will cause an implementation in DSP block. Default value is ”AUTO”.

DEVICE_FAMILY: Defaults to “Stratix”. Normally, this parameter would only be used for behavioral models, since megafunctions look at the Quartus-defined device family. In this case, it is used to select Stratix-compatible assumptions about port and parameter priority resolution. In the original Stratix version of altmult_add, dynamic control inputs such as signa, when connected, took precedence over the value of conflicting parameters, in this case REPRESENTATION_A. Going forward, we prefer that all behavioral information be taken from parameter and port values, and not from connectivity information, since this requires non-standard Verilog and VHDL instantiations. The MegaWizard plug-in for altmult_add will always write a value for this

parameter starting in Quartus II 3.1. The megafunction will check to see if a value is assigned, and if none is assigned, it will revert to pre-Quartus II 3.1 port/parameter priority rules.

The following 2 parameters are used for formal verification support of dynamic control input ports

Operation mode	Rounding /saturation Support	Rounding/ saturation Mode	Width limitation (width_result > width_MSB) When width_result is not the one mentioned, refer to the width_saturate_sign Table and WIDTH_MSB Table	Comments : refer to symmetric saturation table
output_only	output_rounding /output_saturation	non_sym/sym Nearest_integer /Nearest_even	Width_a<=18, width_b<=18 14<Width_MSB <31, 1< width_saturate_sign <9. Base on width_result= width_a + width_b	symmetric saturation has limitation
one_level_adder	output_rounding /output_saturation	non_sym/sym Nearest_integer /Nearest_even	Width_a<=18, width_b<=18 14<Width_MSB <31, 1< width_saturate_sign <9. Base on width_result= width_a + width_b	hardware support is limited to 36 bit as output width. Symmetric saturation has limitation
two_level_adder	output_rounding /output_saturation	non_sym/sym Nearest_integer /Nearest_even	Width_a<=18, width_b<=18 16<Width_MSB <33, 1< width_saturate_sign <11. Base on width_result= width_a + width_b +2	symmetric saturation has limitation
Accumulator	output_saturation	non_sym/sym	Width_a<=18, width_b<=18 1< width_saturate_sign <17. Base on width_result=44	output_rounding will feed garbage data back into add loop. Use chainout_rounding if possible.
accum_chainout	chainout_rounding/ chainout_saturation /output_saturation	non_sym/sym Nearest_integer /Nearest_even	Width_a<=18, width_b<=18 22<Width_MSB <39, 1< width_saturate_sign <17. Base on width_result=44	output_rounding will feed garbage data back into add loop. Use chaiout_rounding for accumulator output. Chaiout_rounding can only be applied at the last stage of the chain
Loopback	output_rounding	Nearest_integer/ Nearest_even	Width_a=18, width_b=18, width_MSB=18. Base on width_result=36	Only upper 18 bit will be feedback. When saturation is turned on, MSB is saturation_overflow, it ruined the sign bit
two_level_adder _chainout	chainout_rounding/chainout_saturation /output_saturation	non_sym/sym Nearest_integer /Nearest_even	Width_a<=18, width_b<=18 22<Width_MSB <39, 1< width_saturate_sign <17. Base on width_result=44	Chaiout_rounding can only be applied at the last stage.
36_bit	NA			
shift	NA			
double	NA			

like signa, signb, to write out a fully connected netlist :

PORT_SIGNA: Legal values are “PORT_USED”, “PORT_UNUSED” and “PORT_CONNECTIVITY” (default). If the value is “PORT_CONNECTIVITY”, then the megafunction checks if the signa port is not connected, and if true, uses the parameter value to determine the port value. If value is “PORT_USED” then the port is assumed to be connected; if “PORT_UNUSED” then the port is ignored and only the parameter value is looked at.

PORT_SIGNB: Legal values are “PORT_USED”, “PORT_UNUSED” and “PORT_CONNECTIVITY” (default). If the value is “PORT_CONNECTIVITY”, then the megafunction checks if the signb port is not connected, and if true, uses the parameter value to determine the port value. If value is “PORT_USED” then the port is assumed to be connected; if “PORT_UNUSED” then the port is ignored and only the parameter value is looked at.

--Stratixiii only

WIDTH_MSB	min	max	condition	round_width of MAC_OUT	Definination
			WR >= RW apply to all rounding		WA = width_a
Chaiout_adder =YES without_sign_extend	15+WRS	Min (30+WRS , WR)	WR>14	8 - WRS + RW	WB = width_b
chaiout_adder =YES with_sign_extend**	Max(WR-21, 1)	WR-6	WR>6	44 - WR + RW	WR = width_result
multiplier = 1 (NO_ACC,NO_CHAINOUT)	Max(15 + WRS,1)	Min(30+WRS, WR)	WA+WB > 14	RW - WRS	WRS = WR - WA - WB
multiplier = 2 (NO_ACC,NO_CHAINOUT, NO_LOOPBACK) *	Max(14 + WRS,1)	Min(29+WRS, WR)	WA+WB > 13	RW - WRS + 1	RW = width_msb
multiplier = 2 (NO_ACC,NO_CHAINOUT, NO_LOOPBACK)	Max(15+ WRS,1)	Min(30+WRS, WR)	WA+WB > 14	RW - WRS	
LOOPBACK	WR-WA	WR-WA	WR>WA	RW - WRS	
multiplier > 2 (NO_ACC,NO_CHAINOUT)	Max(15 + WRS,1)	Min(30+WRS, WR)	WA+WB > 14	RW - WRS +2	

example: WA + WB = 24, one mult, WR = 24: Max RW = 24, Min RW = 15

example: WA + WB = 36, one mult, WR = 25: Max RW = 19, Min RW = 4

example: WA + WB = 24, one mult, WR = 36: Max RW = 36, Min RW = 27

example: WA + WB = 15, one mult, WR = 15: Max RW = 15, Min RW =15

example: WA + WB = 24, two mult, WR = 24: Max RW = 24, Min RW = 15

example: WA + WB = 36, two mult, WR = 25: Max RW = 19, Min RW = 4

example: WA + WB = 24, two mult, WR = 36: Max RW = 36, Min RW = 26

example: WA + WB = 14, two mult, WR = 15: Max RW = 15, Min RW =15

example: WA + WB = 34, four mult, WR = 36: Max RW = 32, Min RW = 17

example: WA + WB = 34, four mult, WR = 34: Max RW = 30, Min RW = 15

example: WA + WB = 15, four mult, WR = 15: Max RW = 15, Min RW = 15

* this case sign extended one bit of dataa or dataab : WR >WA + WB and WA+WB <36

** In accumulator or chainout_adder, if saturation is used or width_result>width_a+width_b+8, it always use sign extend.

Summary of Stratixiii rounding and saturation support:

OUTPUT_ROUNDING: For Stratixiii, enable rounding handling at the final output stage. It's different from Stratix II settings. In some cases, it can be derived from Stratix II rounding settings

if original design is Stratix II. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the saturation handling feature to permanently on or off, and “VARIABLE” allow the saturation handling feature to be controlled dynamically input. It’s a new parameter in Stratixiii.

OUTPUT_ROUND_TYPE: Control nearest_integer/nearest_even rounding mode. Legal values are “NEAREST_EVEN” and “NEAREST_INTEGER”. “NEAREST_INTEGER” means round-to-nearest-integer. “NEAREST_EVEN” means round-to-nearest-even. Default value is “NEAREST_INTEGER” which is compatible with Stratix II. It’s a new parameter in Stratixiii.

WIDTH_MSB: Control the effective data bit width. It includes sign bits(width_saturate_siign) and data bits. Default value is 17 to compatible with Q1.15 mode in Stratix II. It’s a new parameter in Stratixiii. The value counts bits from the MSB (before saturation) to the LSB (after rounding). The valid value should be calculated according to mode, WIDTH_A, WIDTH_B and WIDTH_RESULT. The value should be unsigned integer. If you can not find a positive number for it, that means no rounding allowed in your bit width and mode setting. WIDTH_MSB must smaller than WIDTH_RESULT.

In loop_back mode: WIDTH_MSB can only be 18.

OUTPUT_ROUND_REGISTER: Clock source for the first register on the output_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

OUTPUT_ROUND _ACLR: Asynchronous clear source for the first register on the output_round input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

OUTPUT_ROUND_PIPELINE_REGISTER: Clock source for the second register on the output_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

OUTPUT_ROUND_PIPELINE_ACLR: Asynchronous clear source for the second register on the output_round input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

OUTPUT_SATURATION: For Stratixiii, enable saturation handling at the final output stage. It’s different from Stratix II settings. In some cases, it can be derived from Stratix II rounding settings if original design is Stratix II. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the saturation handling feature to permanently on or off, and “VARIABLE” allow the saturation handling feature to be controlled dynamically input. It’s a new parameter in Stratixiii.

OUTPUT_SATURATE_TYPE: Control SYMMETRIC/ASYMMETRIC saturation mode. Legal values are “SYMMETRIC” and “ASYMMETRIC”. “SYMMETRIC” means the max negative number has the same absolute value as the max positive number. “ASYMMETRIC” means the max negative number is bigger the max positive number. Default value is “SYMMETRIC” which is compatible with StratixII. It’s a new parameter in Stratixiii.

WIDTH_SATURATE_SIGN: Control saturation position. Default value is 1 to be compatible with Q1.15 mode in Stratix II. It’s a new parameter in Stratixiii. The value counts bits from the MSB to

all the sign bits including the sign bit of valid result. The valid value should be calculated according to mode, WIDTH_A, WIDTH_B and WIDTH_RESULT. The value should be unsigned integer. If you can not find a positive number for it, that means no saturation allowed in your input/output width and mode setting. Loopback mode won't support saturation.

width_saturate_sign	min	max	condition	saturate_width of MAC_OUT	Definination
accumulator=YES || chaiout_adder =YES with_sign_extend	1	WR -28	45>WR>28	44 - WR + SW	WA = width_a
multiplier = 1 (NO_ACC, NO_CHAINOUT)	Max of (WRS+2, 1)	Min of (8+WRS, WR)		SW + WRS + 8	WB = width_b
multiplier = 2 (NO_ACC, NO_CHAINOUT NO_LOOPBACK) *	Max of (WRS+1, 1)	Min of (7+WRS, WR)		SW + WRS + 8 + 1	WR = width_result
multiplier = 2 (NO_ACC, NO_CHAINOUT NO_LOOPBACK)	Max of (WRS+2, 1)	Min of (8+WRS, WR)		SW + WRS + 8	WRS = WR - WA - WB
multiplier > 2 (NO_ACC, NO_CHAINOUT)	Max of (WRS, 1)	Min of (8+WRS, WR)		SW + WRS + 6	SW = width_saturate_sign
LOOPBACK	NA	NA		NA	NA
example: WA + WB = 24, one mult, WR = 23: Min RW = 1, Max RW = 7					
example: WA + WB = 24, one mult, WR = 24: Min RW =2 , Max RW = 8					
example: WA + WB = 21, two mult, WR = 21: Min RW = 2, Max RW = 8					
example: WA + WB = 21, two mult, WR = 18: Min RW = 1, Max RW = 5					
example: *WA + WB = 21, two mult, WR = 22: Min RW = 2, Max RW = 8					
example: WA + WB = 24, accumulator, WR = 29: Max SW = 1, Min SW = 1					
example: WA + WB = 24, accumulator, WR = 35: Max SW = 7, Min SW = 1					
example: *WA + WB = 21, four mult, WR = 21: Min RW = 1, Max RW = 8					
* this case sign extended one bit of dataa or dataab : WR >WA + WB and WA+WB <36					
** In accumulator or chainout_adder, if saturation is used or width_result>width_a+width_b+8, it always use sign extend.					

Symmetric saturation has more limitation listed in the following table:

When chainout_adder=NO:

Operation mode			Valid case	Additional Width_limitation
output_only one_level_adder two_level_adder	output_rounding ON	non_sym	Yes	no
		sym	Yes	no
	output_rounding OFF	non_sym	Yes	no
		sym	Yes	WA + WB=36 or 15<WA + WB <30
Accumulator	output_rounding	non_sym	Yes	no

	OFF	**sym**	**Yes**	**no**
	output_rounding ON	**non_sym**	**NA**	**NA**
		sym	**NA**	**NA**

When chainout_adder=YES:

				Valid case	**Additional Width_limitation**
output_rounding OFF	**chainout_rounding ON**	**output_saturation ON**	**non_sym**	**Yes**	**no**
			sym	**No**	**no**
		output_saturation OFF	**non_sym**	**Yes**	**no**
			sym	**Yes**	**no**
	chainout_rounding OFF	**output_saturation ON**	**non_sym**	**Yes**	**no**
			sym	**Yes**	**no**
		output_saturation OFF	**non_sym**	**Yes**	**no**
			sym	**Yes**	**no**
output_rounding ON	**NA**				

OUTPUT_SATURATE_REGISTER: Clock source for the first register on the output_saturate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

OUTPUT_SATURATE _ACLR: Asynchronous clear source for the first register on the output_saturate input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

OUTPUT_SATURATE_PIPELINE_REGISTER: Clock source for the second register on the output_saturate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

OUTPUT_SATURATE_PIPELINE_ACLR: Asynchronous clear source for the second register on the output_saturate input. . Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

CHAINOUT_ROUNDING: For Stratixiii, enable rounding handling at the chainout stage. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the saturation handling feature to permanently on or off, and “VARIABLE” allow the saturation handling feature to be controlled dynamically input. It’s a new parameter in Stratixiii.

CHAINOUT_ROUND_REGISTER: Clock source for the first register on the chainout_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. . It’s a new parameter in Stratixiii. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

CHAINOUT_ROUND _ACLR: Asynchronous clear source for the first register on the chainout_round input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

CHAINOUT_ROUND_PIPELINE_REGISTER: Clock source for the second register on the chainout_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. . It’s a new parameter in Stratixiii. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

CHAINOUT_ROUND_PIPELINE_ACLR: Asynchronous clear source for the second register on the chainout_round input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

CHAINOUT_ROUND_OUTPUT_REGISTER: Clock source for the third register on the chainout_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. . It’s a new parameter in Stratixiii. The value should follow *output_register* or set as “UNREGISTERED”.

CHAINOUT_ROUND_OUTPUT_ACLR: Asynchronous clear source for the third register on the chainout_round input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3. . It’s a new parameter in Stratixiii.

CHAINOUT _SATURATION: For Stratixiii, enable saturation handling at the chainout stage. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the saturation handling feature to permanently on or off, and “VARIABLE” allow the saturation handling feature to be controlled dynamically input. It’s a new parameter in Stratixiii.

CHAINOUT_SATURATE_REGISTER: Clock source for the first register on the chainout_saturate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

CHAINOUT_SATURATE _ACLR: Asynchronous clear source for the first register on the chainout_saturate input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

CHAINOUT_SATURATE_PIPELINE_REGISTER: Clock source for the second register on the chainout_saturate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

CHAINOUT_SATURATE_PIPELINE_ACLR: Asynchronous clear source for the second register on the chainout_saturate input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

CHAINOUT_SATURATE_OUTPUT_REGISTER: Clock source for the third register on the chainout_saturate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *output_register* or set as “UNREGISTERED”.

CHAINOUT_SATURATE_OUTPUT_ACLR: Asynchronous clear source for the third register on the chainout_saturate input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should be the same as *OUTPUT_ACLR* if it needs to be set. It’s a new parameter in Stratixiii.

ACCUMULATOR: Controls accumulator mode of the final adder stage. Legal values are “YES” and “NO”. Default value is “NO” for backward compatibility for Stratix II. It’s a new parameter in altmult_accum in Stratixiii.

CHAINOUT_ADDER: Controls chainout mode of the final adder stage. Legal values are “YES” and “NO”. Default value is “NO” for backward compatibility for Stratix II. It’s a new parameter in Stratixiii.

ZERO_CHAINOUT_OUTPUT_REGISTER: Clock source for the first register on the zero_chainout input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *chainout_register* or set as “UNREGISTERED”.

ZERO_CHAINOUT _OUTPUT_ACLR: Asynchronous clear source for the first register on the zero_chainout input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should be the same as *OUTPUT_ACLR* if it needs to be set. It’s a new parameter in Stratixiii.

ZERO_LOOPBACK_REGISTER: Clock source for the first register on the zero_loopback input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

ZERO_LOOPBACK _ACLR: Asynchronous clear source for the first register on the zero_loopback input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

ZERO_LOOPBACK_PIPELINE_REGISTER: Clock source for the second register on the zero_loopback input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

ZERO_LOOPBACK_PIPELINE_ACLR: Asynchronous clear source for the second register on the zero_loopback input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

ZERO_LOOPBACK_OUTPUT_REGISTER: Clock source for the third register on the zero_loopback input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *chainout_register* or set as “UNREGISTERED”.

ZERO_LOOPBACK_OUTPUT_ACLR: Asynchronous clear source for the third register on the zero_loopback input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should be the same as *OUTPUT_ACLR* if it needs to be set. It’s a new parameter in Stratixiii.

ACCUM_SLOAD_REGISTER: Clock source for the first register on the accum_sload input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in altmult_add. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

ACCUM_SLOAD _ACLR: Asynchronous clear source for the first register on the accum_sload input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

ACCUM_SLOAD _PIPELINE_REGISTER: Clock source for the second register on the accum_sload input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in altmult_add. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

ACCUM_SLOAD_PIPELINE_ACLR: Asynchronous clear source for the second register on the accum_sload input. . Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

SHIFT_MODE: Legal value is “NO”, “LEFT”, “RIGHT”, “ROTATION” and “VARIABLE”. If “VARIABLE” is selected, rotate, shift_right ports are used to selected shift left, shift right or rotation. It’s a new parameter in Stratixiii. Default value is “NO”.

ROTATE_REGISTER: Clock source for the first register on the rotate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *INPUT_REGISTER_A0* or set as “UNREGISTERED”.

ROTATE _ACLR: Asynchronous clear source for the first register on the rotate input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

ROTATE_PIPELINE_REGISTER: Clock source for the second register on the rotate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *multiplier_register0* or set as “UNREGISTERED”.

ROTATE_PIPELINE_ACLR: Asynchronous clear source for the second register on the rotate input. Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It’s a new parameter in Stratixiii.

ROTATE_OUTPUT_REGISTER: Clock source for the third register on the rotate input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new parameter in Stratixiii. The value should follow *chainout_register* or set as “UNREGISTERED”.

ROTATE_OUTPUT_ACLR: Asynchronous clear source for the third register on the rotate input. . Legal values are “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. Default value is “ACLR3” if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to “NONE” and it won’t be passed to wysiwyg. The value should be the same as *OUTPUT_ACLR* if it needs to be set. It’s a new parameter in Stratixiii.

SHIFT_RIGHT_REGISTER: Clock source for the first register on the shift_right input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It’s a new

parameter in Stratixiii. The value should follow *INPUT_REGISTER_A0* or set as "UNREGISTERED".

SHIFT_RIGHT _ACLR: Asynchronous clear source for the first register on the shift_right input. . Legal values are "ACLR0", "ACLR1", "ACLR2", and "ACLR3". Default value is "ACLR3" if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to "NONE" and it won't be passed to wysiwyg. The value should follow *INPUT_ACLR_A0* in Stratixiii if it needs to be set. It's a new parameter in Stratixiii.

SHIFT_RIGHT_PIPELINE_REGISTER: Clock source for the second register on the shift_right input. Legal values are "UNREGISTERED", "CLOCK0", "CLOCK1", "CLOCK2", and "CLOCK3". It's a new parameter in Stratixiii. The value should follow *multiplier_register0* or set as "UNREGISTERED".

SHIFT_RIGHT_PIPELINE_ACLR: Asynchronous clear source for the second register on the shift_right input. Legal values are "ACLR0", "ACLR1", "ACLR2", and "ACLR3". Default value is "ACLR3" if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to "NONE" and it won't be passed to wysiwyg. Legal value should be the same as MULTIPLIER_ ACLR0 in Stratixiii if it needs to be set. It's a new parameter in Stratixiii.

SHIFT_RIGHT_OUTPUT_REGISTER: Clock source for the third register on the shift_right input. Legal values are "UNREGISTERED", "CLOCK0", "CLOCK1", "CLOCK2", and "CLOCK3". It's a new parameter in Stratixiii. The value should follow *chainout_register* or set as "UNREGISTERED".

SHIFT_RIGHT_OUTPUT_ACLR: Asynchronous clear source for the third register on the shift_right input. Legal values are "ACLR0", "ACLR1", "ACLR2", and "ACLR3". Default value is "ACLR3" if the corresponding register is been used. If the corresponding register is not registered, aclr signal should be set to "NONE" and it won't be passed to wysiwyg. The value should be the same as *OUTPUT_ACLR* if it needs to be set. It's a new parameter in Stratixiii.

PORT_OUTPUT_IS_OVERFLOW: Legal values is "PORT_UNUSED", "PORT_USED". New in Stratixiii. If the value is "USED", an output pin overflow is added at the interface. It's a new parameter in Stratixiii.

PORT_CHAINOUT_SAT_IS_OVERFLOW: Legal values is "PORT_UNUSED", "PORT_USED". New in Stratixiii. If the value is "USED", an output pin chainout_sat_overflow is added at the interface. It's a new parameter in Stratixiii.

--Stratix II only

MULTIPLIER01_SATURATION: For StratixII, enable saturation handling at the output of the multiplier stage, for multipliers 0 and 1. Saturation at the output of the multiplier can only occur if dataa and datab are both the maximum negative number. In this case, the product is a positive that is too large to be represented in the given number of bits. Turning on saturation handling will give a maximum positive result for this case. For StatixIII, saturation is at the end of the second adder. Legal values are "NO", "YES", and "VARIABLE". "NO" and "YES" set the saturation handling feature to permanently on or off, and "VARIABLE" allow the saturation handling feature to be controlled dynamically input. Default is "NO" for Stratix compatibility. For Stratixiii, the value is "NO".

MULT01_SATURATION_REGISTER: Clock source for the register on the mult01_saturation input. Legal values are "UNREGISTERED", "CLOCK0", "CLOCK1", "CLOCK2", and "CLOCK3". For Stratixiii, the value is "UNREGISTERED".

MULT01_SATURATION_ACLR: Asynchronous clear source for the register on the mult01_saturation input. Legal values are "NONE", "ACLR0", "ACLR1", "ACLR2", and "ACLR3". For Stratixiii, the value is "NONE".

MULTIPLIER23_SATURATION: For StratixII, enable saturation handling at the output of the multiplier stage, for multipliers 0 and 1. Saturation at the output of the multiplier can only occur if dataa and datab are both the maximum negative number. In this case, the product is a positive that is too large to be represented in the given number of bits. Turning on saturation handling will give a maximum positive result for this case. For StatixIII, saturation is at the end of the second adder. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the saturation handling feature to permanently on or off, and “VARIABLE” allow the saturation handling feature to be controlled dynamically input. Default is “NO” for Stratix compatibility. For Stratixiii, the value is “NO”.

MULT23_SATURATION_REGISTER: Clock source for the register on the mult23_saturation input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”.. For Stratixiii, the value is “UNREGISTERED”.

MULT23_SATURATION_ACLR: Asynchronous clear source for the register on the mult23_saturation input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

ADDER1_ROUNDING: Enable rounding in the adder stage, for the adder used with multiplier 1. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the rounding feature to permanently on or off, and “VARIABLE” allow the rounding feature to be controlled dynamically using the *addnsub1_round* input. Default is “NO” for Stratix compatibility. It can also be the rounding of the independent multiplier. For Stratixiii, the value is “NO”.

ADDNSUB1_ROUND_REGISTER: Clock source for the first register on the addnsub1_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB1_ROUND_ACLR: Asynchronous clear source for the first register on the addnsub1_round input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

ADDNSUB1_ROUND_PIPELINE_REGISTER: Clock source for the second register on the addnsub1_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB1_ROUND_PIPELINE_ACLR: Asynchronous clear source for the second register on the addnsub1_round input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3. For Stratixiii, the value is “NONE”.

ADDER3_ROUNDING: Enable rounding in the adder stage, for the adder used with multiplier 3. Rounding should only be used with input that conforms to signed 1.15 fractional number representation. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the rounding feature to permanently on or off, and “VARIABLE” allow the rounding feature to be controlled dynamically using the *addnsub3_round* input. Default is “NO” for Stratix compatibility. For Stratixiii, the value is “NO”.

ADDNSUB3_ROUND_REGISTER: Clock source for the first register on the addnsub3_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It should be the same as DATAA register. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB3_ROUND_ACLR: Asynchronous clear source for the first register on the addnsub3_round input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

ADDNSUB3_ROUND_PIPELINE_REGISTER: Clock source for the second register on the addnsub3_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. It should be the same as pipeline register. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB3_ROUND_PIPELINE_ACLR: Asynchronous clear source for the second register on the addnsub3_round input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

ADDNSUB_MULTIPLIER_REGISTER1: Clock source for the first register on the addnsub1 input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB_MULTIPLIER_ACLR1: Asynchronous clear source for the first register on the addnsub1 input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

ADDNSUB_MULTIPLIER_PIPELINE_REGISTER1: Clock source for the second register on the addnsub1 input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB_MULTIPLIER_PIPELINE_ACLR1: Asynchronous clear source for the second register on the addnsub1 input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

ADDNSUB_MULTIPLIER_REGISTER3: Clock source for the first register on the addnsub3 input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB_MULTIPLIER_ACLR3: Asynchronous clear source for the first register on the addnsub3 input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

ADDNSUB_MULTIPLIER_PIPELINE_REGISTER3: Clock source for the second register on the addnsub3 input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

ADDNSUB_MULTIPLIER_PIPELINE_ACLR3: Asynchronous clear source for the second register on the addnsub3 input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

MULTIPLIER01_ROUNDING: Enable rounding at the output of the multiplier stage, for multipliers 0 and 1. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the rounding feature to permanently on or off, and “VARIABLE” allow the rounding feature to be controlled dynamically using the *mult01_round* input. Default is “NO” for Stratix compatibility. For Stratixiii, the value is “NO”.

MULT01_ROUND_REGISTER: Clock source for the register on the mult01_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

MULT01_ROUND_ACLR: Asynchronous clear source for the register on the mult01_round input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

MULTIPLIER23_ROUNDING: Enable rounding at the output of the multiplier stage, for multipliers 2 and 3. Rounding should only be used with input that conforms to signed 1.15 fractional number representation. Legal values are “NO”, “YES”, and “VARIABLE”. “NO” and “YES” set the rounding feature to permanently on or off, and “VARIABLE” allow the rounding feature to be controlled dynamically using the *mult23_round* input. Default is “NO” for Stratix compatibility. For Stratixiii, the value is “NO”.

MULT23_ROUND_REGISTER: Clock source for the register on the mult23_round input. Legal values are “UNREGISTERED”, “CLOCK0”, “CLOCK1”, “CLOCK2”, and “CLOCK3”. For Stratixiii, the value is “UNREGISTERED”.

MULT23_ROUND_ACLR: Asynchronous clear source for the register on the mult23_round input. Legal values are “NONE”, “ACLR0”, “ACLR1”, “ACLR2”, and “ACLR3”. For Stratixiii, the value is “NONE”.

PORT_MULT0_IS_SATURATED: Indicates that the *mult0_is_saturated* output should be used or unused. Legal values are “UNUSED” (default), and “USED”. If the value is “UNUSED”, the *mult0_is_saturated* output will be stuck at GND. For Stratixiii, the value is “UNUSED”.

PORT_MULT1_IS_SATURATED: Indicates that the *mult1_is_saturated* output should be used or unused. Legal values are “UNUSED” (default), and “USED”. If the value is “UNUSED”, the *mult1_is_saturated* output will be stuck at GND. For Stratixiii, the value is “UNUSED”.

PORT_MULT2_IS_SATURATED: Indicates that the *mult2_is_saturated* output should be used or unused. Legal values are “UNUSED” (default), and “USED”. If the value is “UNUSED”, the *mult2_is_saturated* output will be stuck at GND. For Stratixiii, the value is “UNUSED”.

PORT_MULT3_IS_SATURATED: Indicates that the *mult3_is_saturated* output should be used or unused. Legal values are “UNUSED” (default), and “USED”. If the value is “UNUSED”, the *mult3_is_saturated* output will be stuck at GND. For Stratixiii, the value is “UNUSED”.

PORT_ADDNSUB1: Legal values are “PORT_USED”, “PORT_UNUSED” and “PORT_CONNECTIVITY” (default). If the value is “PORT_CONNECTIVITY”, then the megafunction checks if the addnsub1 port is not connected, and if true, uses the parameter value to determine the port value. If value is “PORT_USED” then the port is assumed to be connected; if “PORT_UNUSED” then the port is ignored and only the parameter value is looked at. For Stratixiii, the value is “PORT_UNUSED”.

PORT_ADDNSUB3: Legal values are “PORT_USED”, “PORT_UNUSED” and “PORT_CONNECTIVITY” (default). If the value is “PORT_CONNECTIVITY”, then the megafunction checks if the addnsub3 port is not connected, and if true, uses the parameter value to determine the port value. If value is “PORT_USED” then the port is assumed to be connected; if “PORT_UNUSED” then the port is ignored and only the parameter value is looked at. For Stratixiii, the value is “PORT_UNUSED”.

3.1.3 Port definitions

dataa: These are the A inputs to the multipliers.

datab: These are the B inputs to the multipliers.

clock3, clock2, clock1, clock0: These are the four clock inputs.

aclr3, aclr2, aclr1, aclr0: These are the four asynchronous clear inputs.

ena3, ena2, ena1, ena0: These are the four clock enables. Ena3 corresponds to clock3; ena2 corresponds to clock2, etc.

signa, signb: These are for dynamically controlling the representation of the a and b inputs. A high value on signa/b causes the A/B inputs to be interpreted as signed two’s complement numbers. A low value on signa/b causes the A/B inputs to be interpreted as unsigned numbers.

scanina: The control pin to the A scan chain when INPUT_SOURCE_A is “SCANA” or “VARIABLE”. NA in stratixiii.

output_round: dynamic control port. It enables output rounding when active, It works when OUTPUR_ROUNDING = “VARIABLE”.

output_saturate: dynamic control port. It enables output saturation when active, It works when OUTPUT_SATURATION = “VARIABLE”.

chainout_round: dynamic control port. It enables chainout rounding when active, It works when OUTPUR_ROUNDING = “VARIABLE”.

chainout_saturate: dynamic control port. It enables chainout saturation when active, It works when OUTPUT_SATURATION = “VARIABLE”.

zero_chainout: dynamic control port. It sets chainout value to zero when active, It works when CHAINOUT_ADDER= “YES”.

zero_loopback: dynamic control port. It sets loop back value to when active, It works when INPUT_SOURCE_B0 = “LOOPBACK”. Customer need to toggle it correctly to make the loop back mode work correctly.

accum_sload: : dynamic control port. It sets accumulator value to zero when active, It works when accumulator is used, user need to toggle it correctly to make the accumulator mode work correctly.

chainin: input bus of previous stage adder’s result when CHAINOUT_ADDER= “YES”. In stratixiii, the port is hardwired. It can only be connected to output of previous DSP block. It can not be a regular port.

rotate: dynamic control port. It makes rotation when active in shift mode.

shift_right: dynamic control port. It makes shift to right when active in shift mode.

mult01_round: Enables multiplier stage rounding for multiplier 0 and 1 when MULTIPLIER01_ROUNDING = “VARIABLE”. Input port reserved for stratixii backward compatible (The related stratixii parameters will be converted to Stratixiii parameters. Just keep the old port name)

mult23_round: Enables multiplier stage rounding for multiplier 2 and 3 when MULTIPLIER23_ROUNDING = “VARIABLE”. Input port reserved for stratixii backward compatible (The related stratixii parameters will be converted to Stratixiii parameters. Just keep the old port name)

mult01_saturate: Enables multiplier stage saturation handling for multiplier 0 and 1 when MULTIPLIER01_SATURATION = “VARIABLE”. Input port reserved for stratixii backward compatible (The related stratixii parameters will be converted to Stratixiii parameters. Just keep the old port name)

mult23_saturate: Enables multiplier stage saturation handling for multiplier 2 and 3 when MULTIPLIER23_SATURATION = “VARIABLE”. Input port reserved for stratixii backward compatible (The related stratixii parameters will be converted to Stratixiii parameters. Just keep the old port name).

adder1_round: Enables adder stage rounding for the adder used with multiplier 1 when ADDER1_ROUNDING = “VARIABLE”. Input port reserved for stratixii backward compatible (The related stratixii parameters will be converted to Stratixiii parameters. Just keep the old port name).

adder3_round: Enables adder stage rounding for the adder used with multiplier 3 when ADDER3_ROUNDING = “VARIABLE”. Input port reserved for stratixii backward compatible (The related stratixii parameters will be converted to Stratixiii parameters. Just keep the old port name).

addnsub1, addnsub3: These are for dynamically controlling whether to do an add or subtract of the corresponding multiplier. A high value on addnsub1/3 causes an addition to be performed of

the second/(fourth and subsequent odd) multipliers. A low value causes a subtraction. NA in Stratixiii.

scanina: The optional input to the A scan chain when INPUT_SOURCE_A is "SCANA" or "VARIABLE". Not needed in Stratixiii because variable mode is removed.

scaninb: The optional input to the B scan chain when INPUT_SOURCE_B is "SCANB" or "VARIABLE". NA in Stratixiii.

result: The result of the multiply and addition.

scanouta: These are the outputs of the A scan chains. In stratixiii, the port is hardwired. It can only be connected to the data oport of the next multiplier. It can not be a regular port.

overflow: Overflow flag when saturation or accumulator is applied.

In Stratixiii, if the output_saturation is turned on, it gives out the overflow of output_saturation. In sum of two mode, if output_saturation is dynamic mode, overflow will take dual porpose. It will be the MSB of output when output_saturation is disabled, and it will work as overflow when output_saturation is enabled.

Chainout_sat_overflow: Overflow flag for the chainout saturation. It's new in Stratixiii.

scanoutb: These are the outputs of the B scan chains. NA in Stratixiii.

mult0_is_saturated: Indicates that saturation handling has occured for multiplier 0. Must be explicitly enabled with the parameter PORT_MULT0_IS_SATURATED = "USED". NA in Stratixiii.

mult1_is_saturated: Indicates that saturation handling has occured for multiplier 1. Must be explicitly enabled with the parameter PORT_MULT1_IS_SATURATED = "USED". NA in Stratixiii.

mult2_is_saturated: Indicates that saturation handling has occured for multiplier 2. Must be explicitly enabled with the parameter PORT_MULT2_IS_SATURATED = "USED". NA in Stratixiii.

mult3_is_saturated: Indicates that saturation handling has occured for multiplier 3. Must be explicitly enabled with the parameter PORT_MULT3_IS_SATURATED = "USED". NA in Stratixiii.

4. User Interface

MegaWizard plug-in updates will be needed to support new Stratix III features and some feature selection is limited conditionally.

Megawizard interface for the altmult_add and alt_mult_accum megawizards will be updated to support new Stratix III features and some feature selection is limited conditionally.

The following megawizards will be updated with the mentioned changes.

4.1 ALTMULT_ADD interface change when device family is Stratix III

1. For Stratix III, "all the multipliers have similar configurations" is forcibly selected.
2. Saturation and rounding have more selection choices, and should be done in Rounding and Saturation page. A GUI calculator will help user to decide the position or rounding and saturation.
3. Disable selection for shiftout output of input B.
4. Adder operation for the first adder can only be "ADD", "SUB".
5. Add Shift mode and related parameter selection.

6. Input A of the multiplier can only connect to “Multiplier input”, “Shiftin input”. Input B of the multiplier can only connect to “Multiplier input”.
7. Add output chainout mode and related parameter selection.
8. Add selection of Multiply-Adder with loopback in mode selection.
9. Output bit width is limited to 44 when saturation or rounding is selected.
10. Add accumulation and related parameter selection.

4.2 ALTMULT_ACCUM interface change when device family is Stratix III

1. Disable “creat a shiftoutb prot” selection.
2. If support hardware support round and saturation, the output bit width should be limited to 44.
3. Saturation and rounding have more selection choices, and should be provided in Rounding and Saturation page. A GUI calculator will help user to decide the position or rounding and saturation.
4. Port A input source are limited to “Dataa”, “Shiftina”. Port B input source are limited to “Datab”.
5. Accumulate directory can only be “ADD”, “SUB”.
6. Disable all the accum_sload_upper_data relation selection.

4.3 Rounding and saturation backward compatible support for stratixii

4.3.1 Rounding and saturation backward compatible support for altmult_add:

To support the backward compatible rounding and saturation function in stratixii design, the ports of rounding and saturation are kept in autmult_add and altmult_accumi, parameters are converted to new rounding and saturation parameters. A warning message of these converting will be given to user.

4.3.2 Rounding and saturation backward compatible support for altmult_accum:

In altmult_accum, accum_rounding and accum_satuation are kept. mult_round and mult_saturate will be obseleted because no compatible hardware in Stratix III dsp block.