Stratix III Family Functional Description

Version 1.0

April 21, 2006

by

Altera Corporation

Table of Contents:

1. OVERVIEW 3

2. FAMILY OVERVIEW 3

3. GENERAL COORDINATE SYSTEM AND FLOORPLAN 4

3.1 LAB Details 6
3.2 I/O Details 7
3.3 DSP Block Details 10
3.4 RAM Blocks 11
3.5 Assignment Strings 11

4. IMPORTANT STRATIX III CONCEPTS 12

4.1 Multi-Purpose Locations 12

5. EXTRA TERMINOLOGY AND READING ORDER 12

6. CONCLUSION 13

1. Overview

The purpose of this document is to give a general overview of the Stratix III family with a focus on specific differences between Stratix II and Stratix III. It is a brief description and it is not intended to have a detailed description of any of the features. This is intended as a starter document before reading all the other documents. Each feature will have its specific Functional Description in the future. Familiarity with the Stratix II architecture is essential in order to understand the concepts in this document.

2. Family Overview

The Stratix III family is the next high-density FPGA family after Stratix II. It is backwards compatible with both Stratix and Stratix II. Any design that is compiled for these existing families can be recompiled for Stratix III without changing the functionality of the design. In rare cases, a few special features of advanced I/O blocks will not transfer from Stratix II to Stratix III. These cases should be clearly spelled out in the FD of the specific I/O blocks. Table 1 outlines the major changes to the different block types present in the Stratix III family, both in terms of functionality and how Quartus will model them.

Table 1: Block Type Highlights for the Stratix III family

Block Type	Functional Highlights	Modeling Differences
LAB	- 10 ALMs per LAB (8 in Stratix II) - Combinational resources can support a reduced feature FF atom - Entire LAB can be flipped into a simple-dual port 32dx16w or 64dx8w RAM (only applies to ½ the LABs)	- 4 “slots” per ALM (like Stratix II) - Different atom types can be placed at LE_COMB locations. - A new atom type, LUTRAM, will model logical RAMs that are implemented in LABs - LUTRAM atoms will not have implicit input and/or output registers
I/O	- Ubiquitous memory interface support. Different speed on top vs side - Migratable I/O banks	- All sub-components of the I/O (including the PAD) are being modeled as separate atoms.
SEAB	- This block types does not exist in the Stratix III family.	- LABs should be used to implement small logical RAMs (1 LAB = 640 bits). They are called LUT-RAMs. LUTRAMs will be shown to the users as RAM to replace SEABs
MEAB	- Stratix III uses M8K blocks, which can support 8KB of data (plus parity bits)	- MEAB & MRAM will use the same RAM atom like Stratix II. This atom will include the RAM registers.
M-RAM	- M-RAM capacity is reduced to 128KB - M-RAM has hardware error correction	- The floorplan will arrange the M-RAMs in columns (as opposed to holes)

	(ECC) built in in x72 mode - Supports ROM mode and pre-initialized RAM	
DSP	- Up to 8 18x18 multipliers when using cascaded output adder - Refer to FD for details	- DSP will be modeled like Stratix II with MAC_MULT & MAC_OUT atoms
PLL	A single PLL type (rather than 2) similar to the Stratix II GPLL. Please refer to FD for details	
LVDS	- Forward the phase of the DPA clock to the core - Please refer to the FD for details	
OCT	- Changed a lot from Stratix II. Available on all sides. - Please refer to the FD for details	
Clocking	New flexible clocking scheme replaces global & regional clock partition. Please refer to FD for details.	

3. General Coordinate System and Floorplan

The coordinate system used for Stratix III, which is very similar to that of Stratix II, is a flat two-dimensional grid of locations. It is a Cartesian based grid system where the coordinate (0,0) represents the location in the lower-left hand corner of the device. The size of the grid is directly based on the number of routing channels available in the device. In any documentation describing the coordinate system, the terms *nx* and *ny* represent the number of columns and rows, respectively, in the device.

A given (X,Y) location can contain one or more blocks. This situation occurs when multiple blocks have access to the same routing interface. A typical example of this is the periphery of the device where a handful of I/Os (between 3 and 5) share the same routing interface. The coordinate system in Quartus uses a 3rd dimension to differentiate between the available slots at the same (X,Y) location. We arbitrarily define the lower-left hand corner of a block to represent the origin of that block. A block that is *x* units wide and *y* units high that has its lower-left hand corner placed at (7, 3) will occupy the coordinate space (7, 3) → (7 + *x* – 1, 3 + *y* – 1). An example of a "large" block in Stratix III is the DSP block, which is 1 unit wide and 4 units high. For Stratix III, the 3rd coordinate dimension is denoted Z and it must be non-negative. Therefore, callers can uniquely identify a block, of a given type, by the (X, Y, Z) coordinate triple.

Since the set of block types used by the Stratix III family represent different levels of abstraction for the elements present in the device, it is useful to define a new concept, called "hierarchy level" that indicates the relative level of abstraction of a given block type. This concept of hierarchy level will then be used to create a rule that defines the legality of a device floorplan. All block types that have a 1 to 1 correspondence with atoms that can appear in the atom netlist (e.g. PLL, LVDS_TX, FF) are on the lowest hierarchy level, since they cannot be decomposed into a set of constituent blocks. We arbitrarily specify that level 0 represents the lowest hierarchy level. Block types that are constructed out of atoms (e.g. LABs, DSPs, I/Os) are on the second hierarchy level, which we define as level 1. Finally, the GXB (a.k.a. QUAD) block in the Stratix III

devices that have Gigabit Transceiver blocks is at the 3rd hierarchy level (level = 2), since it contains I/O blocks, which are at the 2nd hierarchy level. Formally, we define the level for a composite block type (i.e. not an atom) is: *max hierarchy level of a subblock* + 1.

For any floorplan, the following rule must be observed: *"Two blocks at the same floorplan hierarchy level cannot share the same (X,Y,Z) coordinate triple."* This rule must hold even when considering blocks larger than 1 unit wide by 1 unit high. Therefore, we observe that a block on the floorplan can be uniquely identified by a (level, X, Y, Z) quadruple. The device database will provide queries to return the global ID of a placement element by both (type, X, Y, Z) and (level, X, Y, Z). Here are some examples to illustrate the overlapping principle:

- A 1x1 LAB block at (X, Y, 0) does not overlap with a COMB block at (X, Y, 0), since they are at different hierarchy levels (LAB is level 1, COMB is level 0).
- A 2x1 I/O block (X, Y, 0) does not overlap with another I/O block at (X+2, Y, 0), since the (X,Y) coordinates used by each of the I/Os are distinct.
- A 2x1 I/O block (X, Y, 0) does not overlap with another I/O block at (X, Y, 1), since their 3rd dimension is different
- A 2x6 MRAM block (X, Y, 0) **does** overlap with another 2x6 MRAM block at (X+1, Y+5, 0), since the coordinate (X+1, Y+5, 0) is used by both blocks.

Figure 1 shows an example floorplan of a Stratix III device.

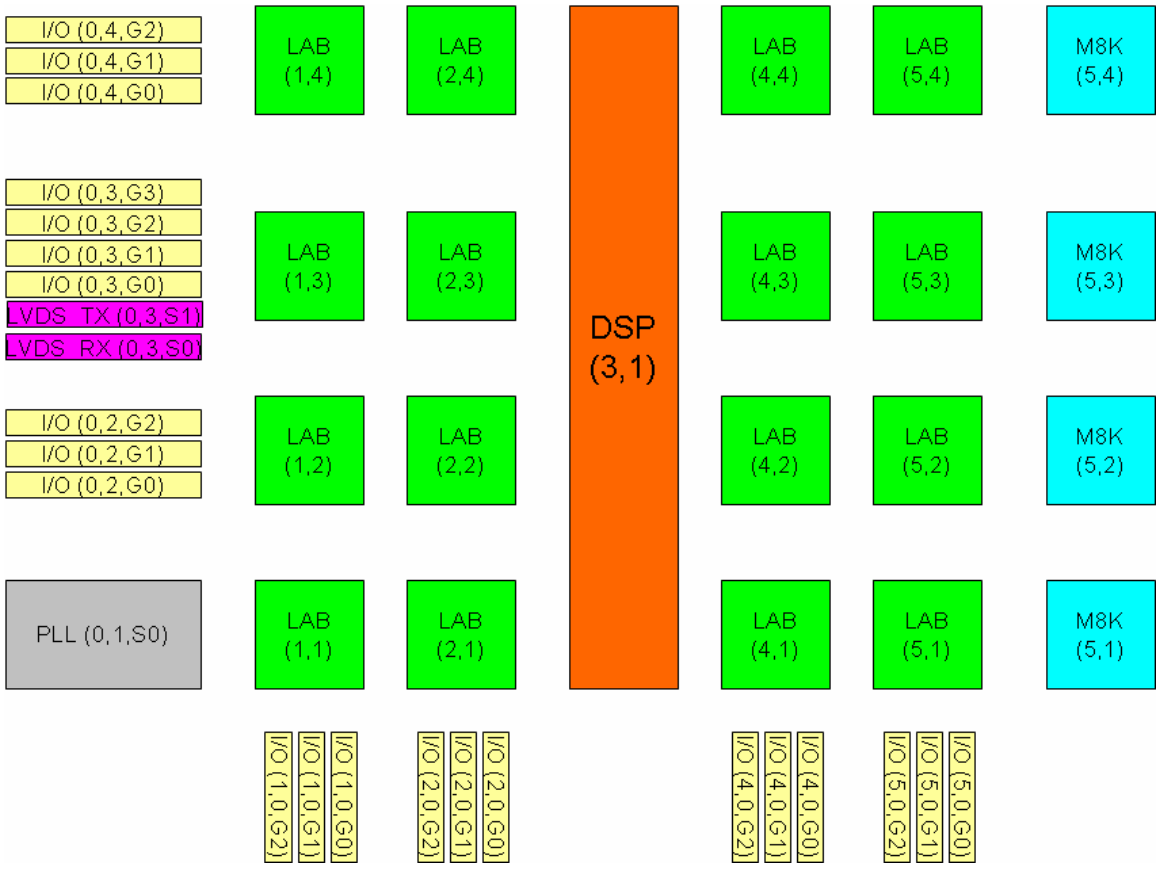


Figure 1: Floorplan and Coordinate System

3.1 LAB Details

Stratix III LABs consist of 10 Adaptive Logic Elements (ALE) that implement a superset of the ALE functionality present in the Stratix II architecture. There are two main features that were not contained in Stratix II:

- All the combinational resources in the LAB can be converted to function as a 32x16 or 64x8 simple dual-port RAM. The netlist will contain LUTRAM atoms to model this mode of the LAB. This ability is only present on ½ the LABs on the device.
- The combinational resources in ALEs (i.e. the 64 CRAM bits and the MUX tree) can implement a master-slave flip-flop. The netlist will contain a "regular" LCELL_FF atom when the ALE is used in this mode. When the "LUT reg" created does not use a SCLR signal, a 2-input LUT function that feeds the register, and does not fan out elsewhere, can also be implemented in the ALE.

Due to the fact that we want to use different atom types to model different uses of the same hardware, we need to use a new concept called multi-purpose locations (see section 4.1 for more details). Essentially, within the Stratix III ALE model will that there are four "slots" in the ALE – two of them will be multi-purpose locations to represent the different atom types (LCELL_COMB, LUTRAM, LCELL_FF) that can be implemented using the combinational resources of the ALE, and the other two for the dedicated FF elements in the ALE.

The Stratix III LAB consists of 20 MP_COMB blocks and 20 LE_FF blocks. The MP_COMB blocks refer to the generic combinational elements in the LAB that can support LUTRAM, FF, and LCELL_COMB atoms. The LE_FF blocks refer to the dedicated flip-flop elements in the LAB. It was arbitrarily decided that even numbered sub-locations in the LAB are combinational elements and odd numbered sub-locations are registered elements. The LABs in the device that support LUTRAM will contain MP_MLAB_COMB blocks, since the NPP refers to these LABs as "MLABs". The LABs in the device that do not support LUTRAM will contain MP_LAB_COMB blocks. Please refer to the Stratix III WYSIWYG documents for more details regarding the content of the COMB, FF, and LUTRAM elements. Figure 2 shows the coordinate system for the cells inside a LAB. A Stratix III LAB is a 1x1 size block in the (x,y) coordinate space of the device floorplan.

When an ALE implements a LUT-reg, the MP_COMB with the higher Z coordinate should contain the FF atom, and a 2-LUT that feeds this FF and meets the ALE legality constraints can be placed in the top (lower Z coordinate) MP_COMB.

The concept of multi-purpose locations (blocks) should not be made visible to the user. Instead of using the string LABCOMB_X1_Y1_N0 (or MLABCOMB_X2_Y1_N0) to refer to the multi-purpose locations in the device, it is preferable to use one of the suitable atom types to refer to the location. For example, LCCOMB_X1_Y1_N0 and FF_X1_Y1_N0 are both valid strings to refer to a MP_LAB_COMB location.

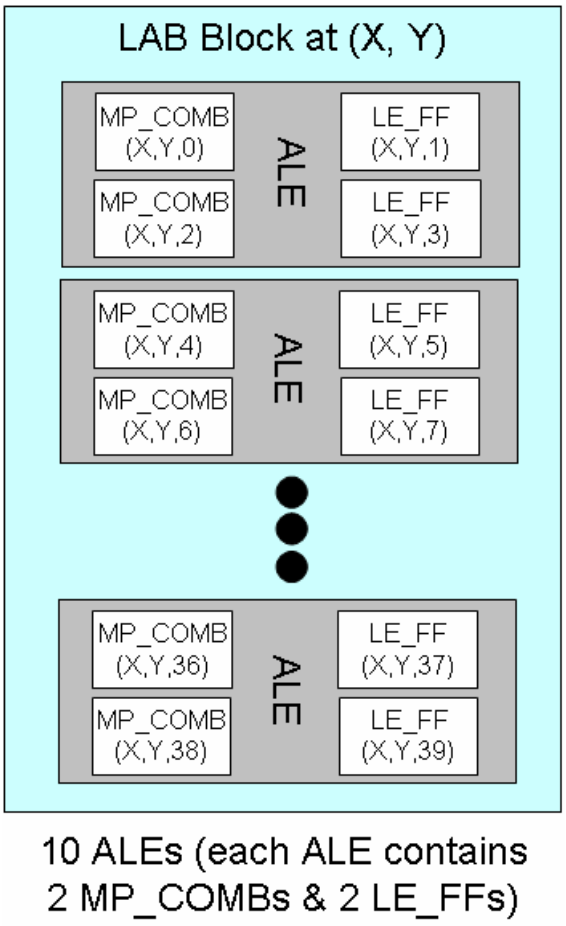


Figure 2: LAB Block Details

3.2 I/O Details

Stratix III I/O structure is very similar to the Stratix II I/O with a few enhancements to improve the DDR support. It has 3 components – input path, output path and output-enable path. The input path handles the processing of data from the pin to the core. It contains an input buffer, 2 registers and delay elements. The output path handles the data from core to the pin. It contains the tri-state output buffer, 2 registers, ddio mux and delay elements. And the output-enable path handles the OE signal from core to the output tri-state buffer. It has 2 registers and delay elements.

The main difference in Quartus is we will model the Stratix III I/O structure as multiple separate smaller components as opposed to the single component used in Stratix II. The motivations for this modeling change are listed as follows:

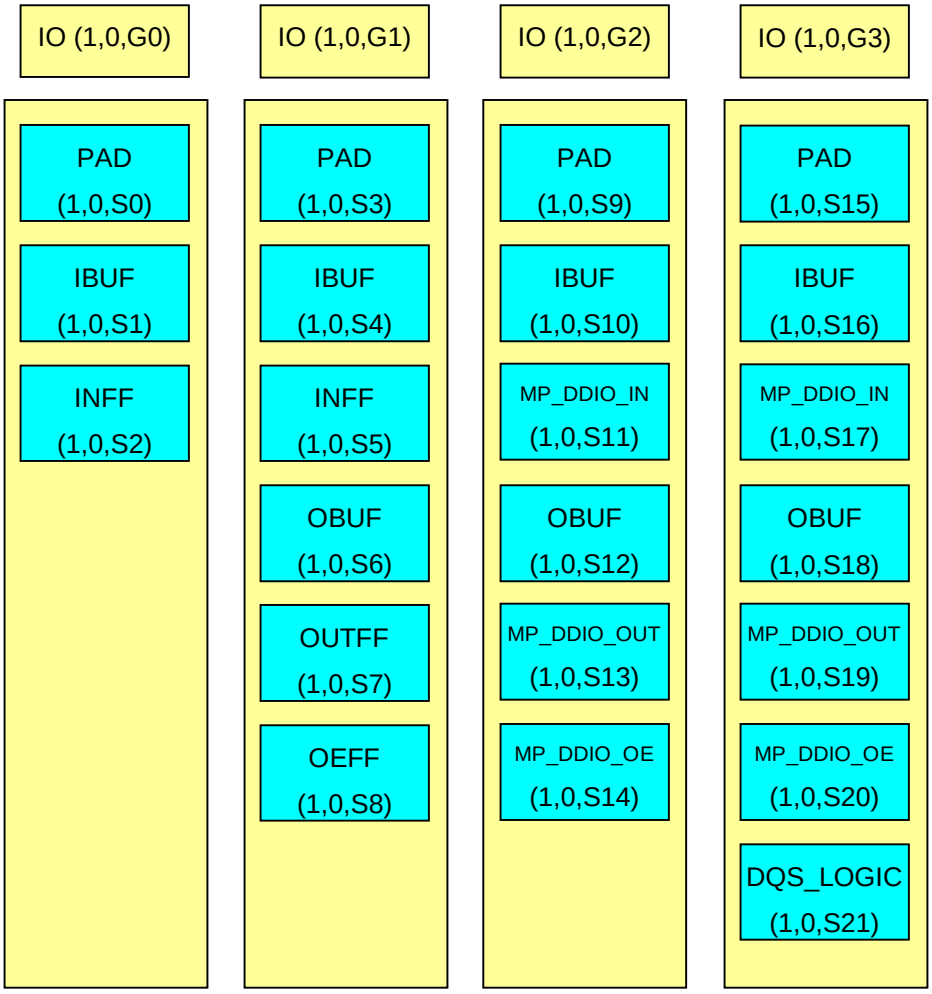
- Past experience of problems in current single-atom model
    - Netlist change overhead with register packing
    - Name usability issue with buried registers
    - Hierarchy issues with single-atom model – can't separate hierarchy for components within I/O, which impacts incremental compile and formal verification
    - Unfriendly for I/O timing optimization – register packing is a heavy-weight operation that require netlist updates.

    - Auto-delay chain: many flags to represent delay chains
    - Desire for an atom that will represent the physical pad (port) to get on and off the chip
- Increased complexity calls for changes.

Due to the modeling change, I/O will now be on the second level hierarchy just like LAB. The underlined components will be on the lowest hierarchy level. Each Stratix III I/O will be called a group and consists of 10 sub-components as follows:

- PAD
- Input buffer
- Output buffer (include the tri-state)
- Input FF
- Output FF
- OE FF
- DDIO Input Logic (DDIO_IN)
- DDIO Output Logic (DDIO_OUT)
- DDIO OE Logic (DDIO_OE)
- DQS Logic

An important thing to note here is the DDIO logics are a superset of the normal FFs functionality. So we will use the multi-purpose location scheme to model the DDIO components. An I/O with DDIO functionality will only contain the multi-purpose DDIO locations that can accommodate both the normal FF and DDIO logic atoms. An I/O without DDIO functionality but with normal FFs will contain only the normal FF location. Figure 3 shows the contents of various types of I/Os. The sizes of the I/O blocks are expected to be 1x1 for HIOs, and either 1x1 or 2x1 for VIOs.



Note: IO at (1,0,G0) only has input capability

IO at (1,0,G1) has bidir capability, but no DDIO

IO at (1,0,G2) can be bidir or DDIO, but no DQS

IO at (1,0,G3) can be bidir, DDIO or DQS

MP_DDIO_IN can support both normal input FF and DDIO_IN logic

MP_DDIO_OUT can support both normal output FF and DDIO_OUT logic

MP_DDIO_OE can support both normal OE FF and DDIO_OE logic

Figure 3: I/O Details

3.3 DSP Block Details

A DSP block has a width of 1 LAB and a height of 4 LABs. Each DSP block consists of 8 DSP Multiplier blocks and 8 DSP Output blocks. The DSP multipliers are evenly distributed across the 4 rows of the DSP block and are represented by sub-locations 0 & 1. Likewise, the DSP Output blocks are evenly distributed across the 4 rows of the DSP block and are represented by sub-locations 2, 3, 4, and 5. Unlike the DSP Multiplier blocks, the DSP output blocks have their base coordinate corresponding to the bottom-left coordinate of the DSP *half block* they are in. This is because each DSP half block can operate mostly independently and because there are some modes which occupy the entire half of the DSP block. It is necessary to have 16 distinct locations inside the DSP block since, in one of the DSP modes, up to 8 Multiplier and 8 Output blocks can be placed inside a single DSP block. Figure 4 shows the contents of a Stratix III DSP block.

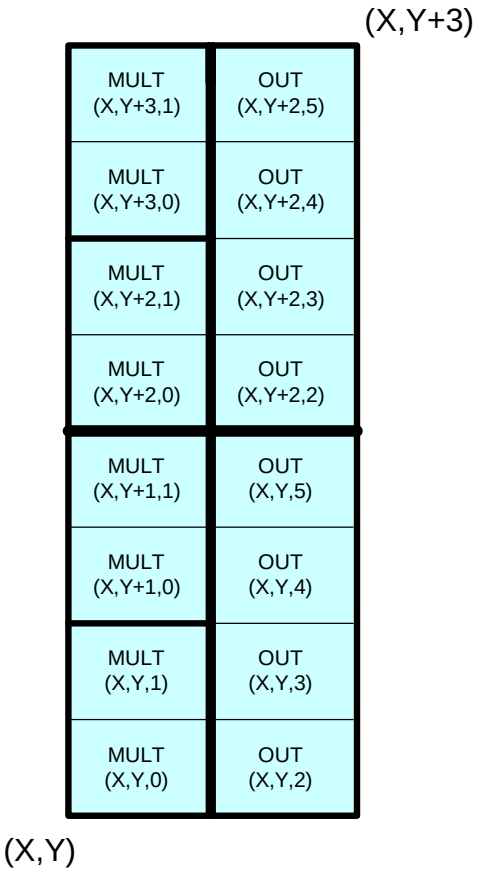


Figure 4: DSP Block Details

As with Stratix-II, the user locations for the DSP block vary slightly from the internal representation described above. Specifically, the DSP Output block sub-locations are not exposed to the user and users are only allowed to assign to each DSP Output block half rather than down to the sub-location of that half. This distinction is made because the sub-locations of the DSP output block have no meaning in most modes where the output block in each half is combined to implement the mode. As an example see Figure 5: DSP Block User Locations.

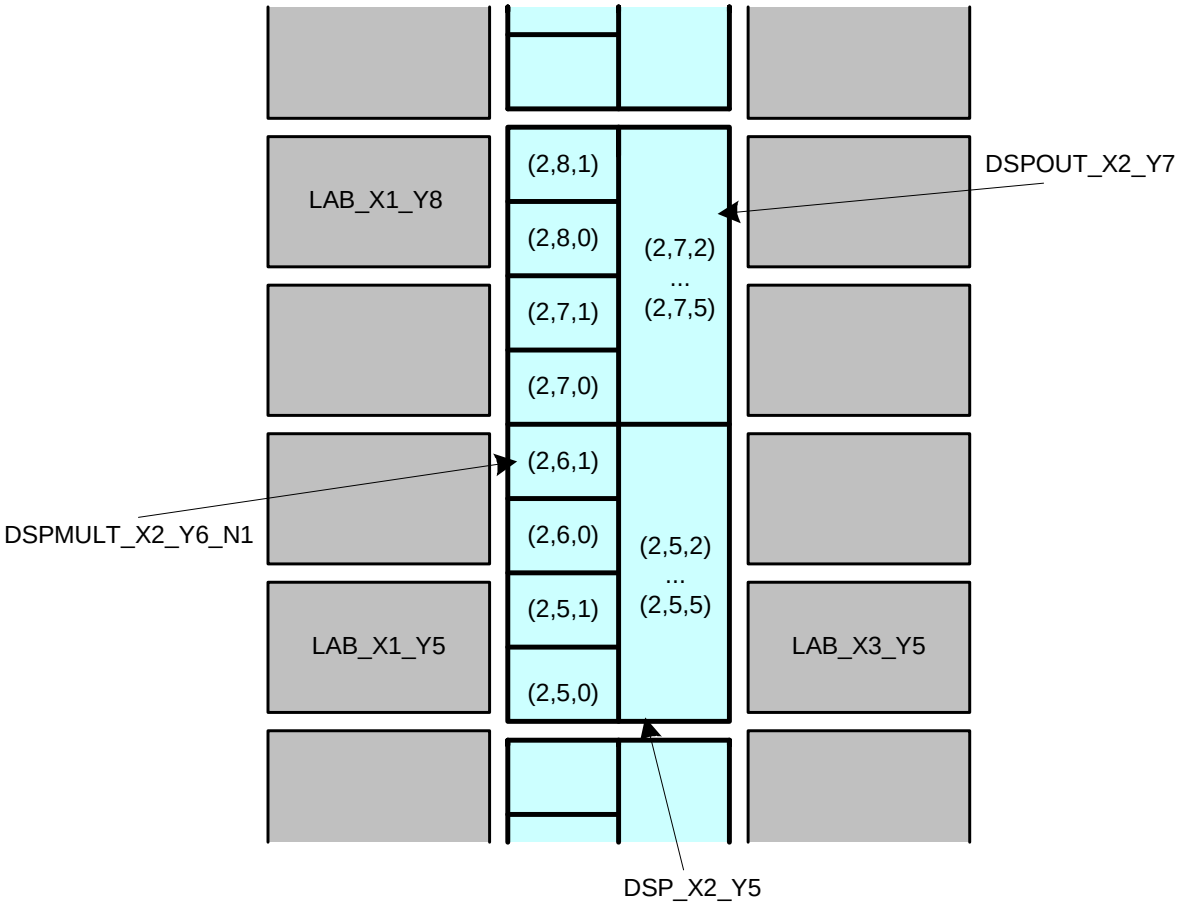


Figure 5: DSP Block User Locations

3.4 RAM Blocks

There are two sizes of RAM blocks (other than LUT RAM) in Stratix III devices. The M8K RAM contains 8kbit + parity (9216 bits), while the M128K RAM contains 128 kbit + parity (147456 bits). The M8K block 1x1 in device coordinates, while the M128K RAM is 1x8 in device coordinates.

3.5 Assignment Strings

The user should be shielded from the concepts of the Z dimension and hierarchy levels. Therefore, to maintain some consistency with previous device families, all location strings that are shown to the user should have the following structure: <type>_X#_Y#_N#, where N represents the user "string" (character, actually) for the Z dimension. To be compatible with user expectations, Quartus must also be able to accept location strings of the form *<type>_X#_Y#* for composite block types whose Z coordinate can be inferred from the type, X coordinate, and Y coordinate. However, Quartus will always internally represent coordinates and location constraints using all three dimensions.

To correctly identify a composite block (such as the LAB, DSP, RAM), one needs the coordinate of the bottom left hand corner of the block. To identify a specific component inside the block, a three dimensional coordinate (within the region of the block) is also required. The format of a location in Stratix III must always match one of the following templates:

<*Location Name*>_**X**<*X coordinate*>_**Y**<*Y coordinate*> or

<*Location Name*>_**X**<*X coordinate*>_**Y**<*Y coordinate*>_**N**<*Z coordinate*>

The following list contains examples of location strings for objects in a Stratix III floorplan:

- LAB_X1_Y4 refers to the LAB block at location (1,4). Note: LAB_X1_Y4_N0.
- DSP_MULT_X24_Y2_N1 refers to the second DSP Multiplier block at location (24,2). One cannot tell what the coordinates of the "parent" DSP block is simply based on the location of one of its constituent elements.
- IOC_X0_Y3_N1 refers to the second I/O block whose bottom left corner is location (0,3).
- FF_X0_Y3_N1 refers to one of the FFs in the I/O located at (0, 3). One cannot tell the exact IO block that is the container object of the I/O FF from inspection. To obtain this information, one would need to query the device database for additional data.
- QUAD_X74_Y30 refers to the QUAD block at (74, 30). Note: QUAD_X74_Y30_N0 is also valid.
- PLL_4 refers to the PLL on the device that was given the index 4 by IC-design. Only PLLs can be accessed by an index number

4. Important Stratix III Concepts

This section gives more detail about several important concepts that are being introduced in the Stratix III device family.

4.1 Multi-Purpose Locations

Multi-purpose locations are used to model the fact that different logical atoms can use the same hardware (i.e. physical resources). As an example, multi-purpose locations are used to model the fact that the combinational portion of the ALM (i.e. the 64 CRAM bits & 6-level MUX tree) can be used to implement 1) a single 6-input lookup table (and some 7-input functions), 2) two lookup tables with some amount of input and/or logical compatibility, 2) a master-slave flip-flop, or 3) a 32x2 or 64x1 simple dual-port RAM. The Stratix III LAB contains 20 multi-purpose locations (2 for each ALM) to model the capability of the ALM to support different kinds of logical atoms. Specifically, those 20 locations will contain a "multi-purpose combinational block" (denoted MP_MLAB_COMB or MP_LAB_COMB). The Stratix III device database will also indicate that if someone tries to put a FF atom into a MP_COMB location, it will behave as a LUT-FF (rather than an I/O output enable FF or a dedicated ALM flip-flop). All Quartus modules will need to understand that the timing, power, and (potentially) low-level functionality of an atom will depend on the exact (X, Y, Z) coordinate where it is placed.

5. Extra Terminology and Reading order

We have made every effort to get a consistent naming convention between all the documents. It should be noted that LE_COMB is a DEV enum that corresponds to a location that supports a combinational atom, whereas LCELL_COMB refers to the enum of the combinational

atom. Similarly, the relationship between LE_FF and LCELL_FF is the same as the relationship between LE_COMB and LCELL_COMB.

The documents should be read in the following order:

1) Stratix III Family FD
2) Lcell Wysiwyg Description for Stratix III
3) Stratix III EDA Functional Description

6. Conclusion

This is a very high-level document to get people started on Stratix III. This is the first document to read and then you can pick which block to learn in detail by reading the specific Functional Description of each block.