Cyclone III Family Functional Description

Version 2.0

March 9, 2009

by

Altera Corporation

Table of Contents:

1. Overview

The purpose of this document is to give a general overview of the Cyclone III family with a focus on specific differences between Cyclone II and Cyclone 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 Cyclone II architecture is essential in order to understand the concepts in this document.

2. Family Overview

The Cyclone III family is the next low cost FPGA family after Cyclone II. It is backwards compatible with both Cyclone and Cyclone II. Any design that is compiled for these existing families can be recompiled for Cyclone III without changing the functionality of the design. In rare cases, a few special features of advanced I/O blocks will not transfer from Cyclone II to Cyclone 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 Cyclone III family, both in terms of functionality and how Quartus will model them.

Table 1: Block Type Highlights for the Cyclone III family

Block Type	Functional Highlights	Modeling Differences
LAB	- 16 LEs per LAB (16 in Cyclone II) - LAB and LE are identical to Cyclone II	- Same as Cyclone II
I/O	- Added hardware DDIO output circuitry and hardware DDIO OE circuitry	- All sub-components of the I/O (including the PAD) are being modeled as separate atoms.
MEAB	- Cyclone III uses M9K blocks, which can support 9KB of data (plus parity bits) (Cyclone II uses M4K blocks) - Added new features for read-during-write - Added aclr for output latch and address inputs - Added new behavior for byte-enable masks - Changed the RAM to write on the positive edge and not the negative edge	- MEAB will use the same RAM atom like Cyclone II. This atom will include the RAM registers.
DSP	- Identical to Cyclone II. Just a multiplier. Added programmable invert	- Will be modeled like Cyclone II

	on the sign bit	
PLL	Similar PLL to Titan. Reworked PLL. Functionally equivalent to Cyclone II but re-architected and new features added	
LVDS	- More hard buffers added compared to Cyclone II but there is no dedicated SERDES. Everything is soft.	
OCT	- Has calibrated OCT. It is different that Stratix II. It is similar to Titan but not identical	- Refer to FD for details
Clocking	Same style as Cyclone II but different number. Basically, 1 global clock type	

3. General Coordinate System and Floorplan

The coordinate system used for Cyclone III, which is very similar to that of Cyclone 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 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 Cyclone III is the JTAG & Config block which is several LABs in dimension. For Cyclone 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 Cyclone 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, 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. 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

Figure 1 shows an example floorplan of a Cuda device.

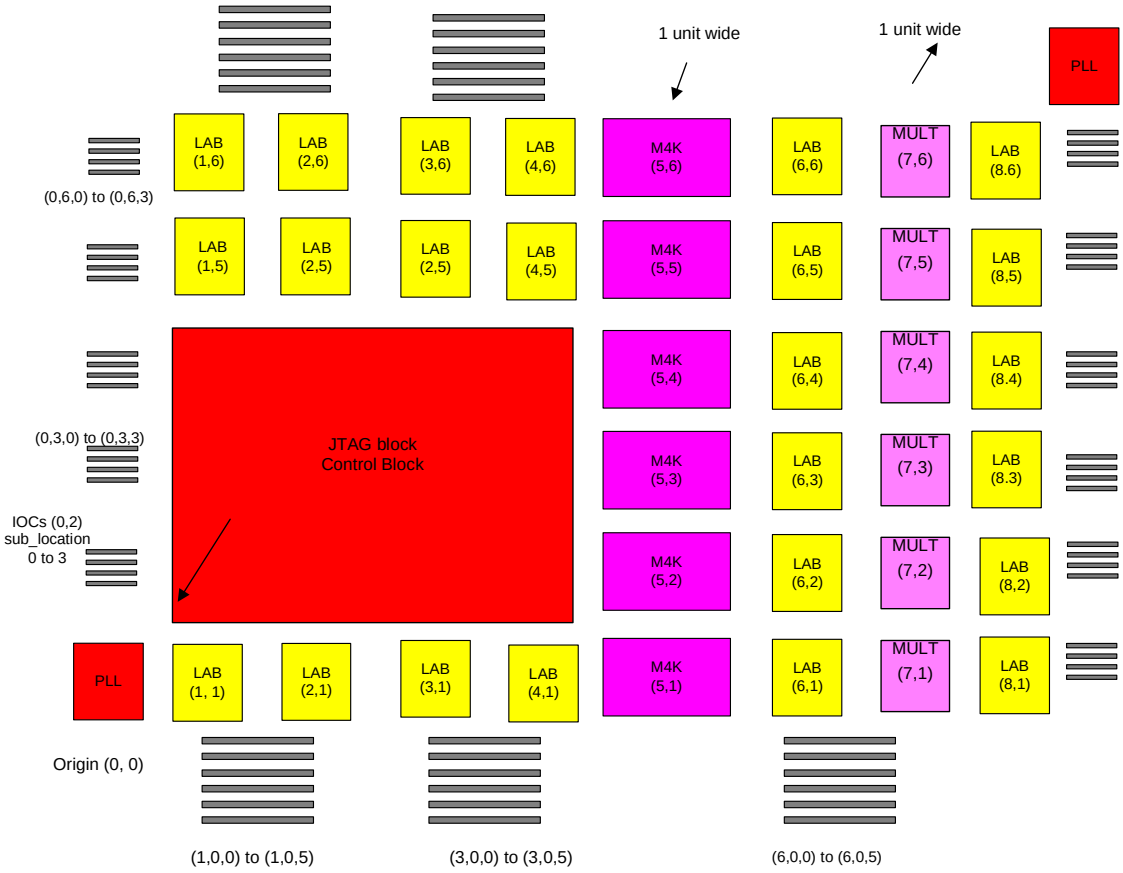


Figure 1: Floorplan and Coordinate System

3.1 LAB Details

The LAB in Cyclone III devices is identical to the LAB in Cyclone II devices. It consists of 16 LE_COMB blocks and 16 LE_FF blocks. LE_COMB blocks are 1 combinational element each while LE_FF blocks are 1 flipflop (register) element each. Each group of 1 LE_COMBs and 1 LE_FFs are organized into 1 LE (Cyclone III LE). Thus, each LAB has 16 LEs. Even numbered sub-locations (indices) in the LAB are combinational elements (LE_COMBs) while odd numbered sub-locations are registered (LE_FFs) elements.

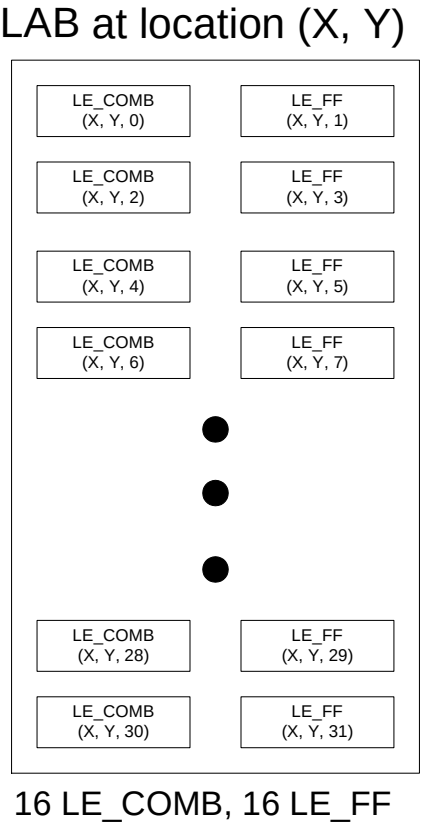


Figure 2 LAB Coordinate System

3.2 I/O Details

The Cyclone III I/O structure is very similar to the Cyclone II I/O with a few enhancements to improve the DDIO 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,1 optional register and some 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. Cyclone II only had 1 output register. Cyclone III was enhanced to add hardware DDIO circuitry. And the output-enable path handles the OE signal from core to the output tri-state buffer. It has 2 registers and delay elements. Cyclone II only had 1 optional register. Cyclone III has 2 registers for enhance DDIO circuitry.

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

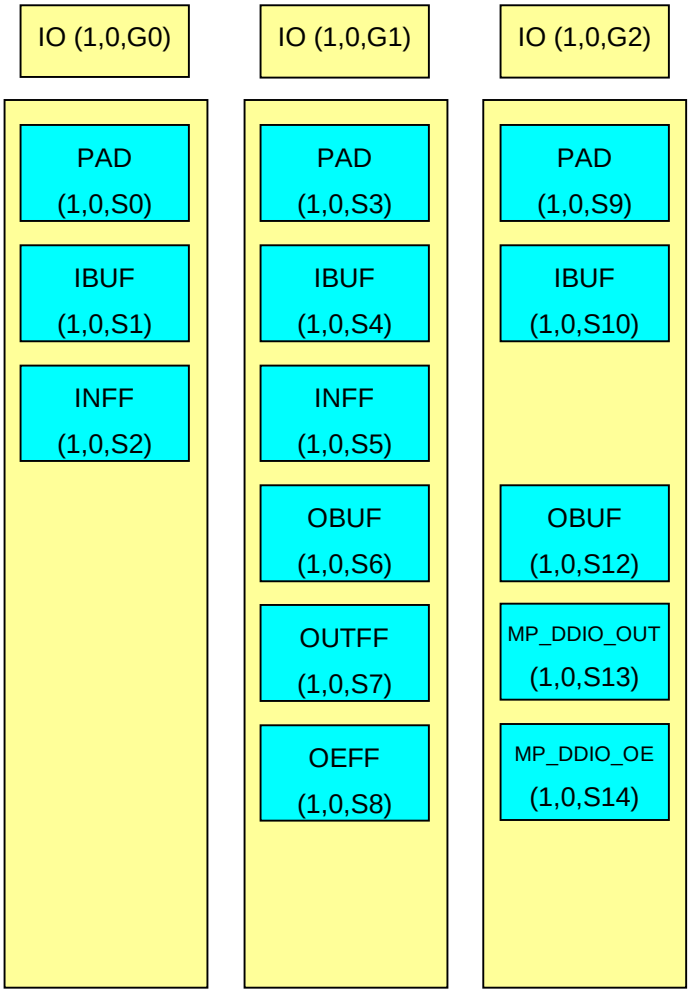
- Past experience of problems in current single-atom model
    - Netlist change overhead with register packing
    - Name usability issue with buried registers
    - Hierarchy issues with single-atom model – can't separate hierarchy for components within I/O, which impacts incremental compile and formal 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 Cyclone III I/O will be called a group and consists of 11 sub-components as follows:

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

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 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

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 Mult Block Details

This section is identical to Cyclone II since the Multiplier block was not changed when it was moved to Cyclone III

A MULT block is 1 LABs wide by 1 LABs long. Each MULT block consists of 2 embedded multiplier blocks and 1 MULT output block. The (X,Y) coordinates of the embedded multipliers follow the LAB numbering scheme and an index to decide if it is the first or the second in that LAB row. For details of the contents of each block please refer to the MULT block functional description. Figure 3 shows the coordinates for a MULT block.

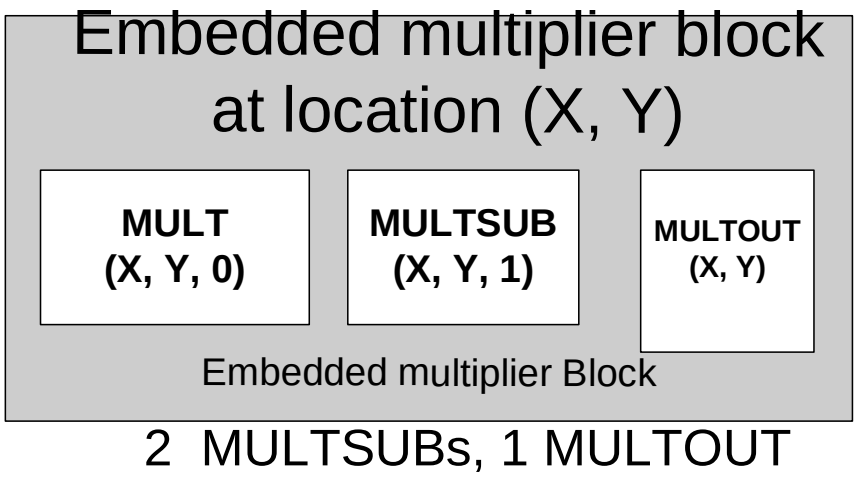


Figure 4. Embedded Multiplier Block Coordinate Details

3.4 RAM Blocks

Cyclone III has 1 RAM block called the M9K block. It is double the size of the M4K block. Everything else about it is identical to Cyclone II from a coordinate system point of view. It has the dimension of 1x1. From a software support point of view, it is the same as Cyclone II. The functionality of the block has been enhanced as described in Table 1

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 Titan 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 Cyclone III floorplan:

- LAB_X1_Y4 refers to the LAB block at location (1,4). Note: LAB_X1_Y4_N0.
- 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.
- 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 Cyclone III Concepts

This section gives more detail about several important concepts that are being introduced in the Cyclone III and Titan 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 DDIO_OUTPUT atom and the basic FF atom (when used as an output I/O FF) can be placed in the same exact physical location on the floorplan in Quartus. 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) Cyclone III Family FD
2) Lcell Wysiwyg Description for Cyclone III
3) Cyclone III EDA Functional Description

6. Conclusion

This is a very high-level document to get people started on Cyclone 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.