
Project X-Ray Documentation

SymbiFlow Team

Jul 03, 2020

Xilinx 7-series Architecture

1	Overview	3
2	Configuration	5
2.1	Addressing	5
2.2	Loading sequence	6
2.3	Other	7
3	Bitstream format	9
4	Glossary	11
	Index	13

Project X-Ray documents the [Xilinx](#) 7-Series FPGA architecture to enable development of open-source tools. Our goal is to provide sufficient information to develop a free and open Verilog to bitstream toolchain for these devices.

CHAPTER 1

Overview

Todo: add diagrams.

Xilinx 7-Series architecture utilizes a hierarchical design of chainable structures to scale across the Spartan, Artix, Kintex, and Virtex product lines. This documentation focuses on the Artix and Kintex devices and omits some concepts introduced in Virtex devices.

At the top-level, 7-Series devices are divided into two *halves* by a virtual horizontal line separating two sets of global clock buffers (BUFGs). While global clocks can be connected such that they span both sets of BUFGs, the two halves defined by this division are treated as separate entities as related to configuration. The halves are referred to simply as the top and bottom halves.

Each half is next divided vertically into one or more *horizontal clock rows*, numbered outward from the global clock buffer dividing line. Each horizontal clock row contains 12 clock lines that extend across the device perpendicular to the global clock spine. Similar to the global clock spine, each horizontal clock row is divided into two halves by two sets of horizontal clock buffers (BUFHs), one on each side of the global clock spine, yielding two *clock domains*. Horizontal clocks may be used within a single clock domain, connected to span both clock domains in a horizontal clock row, or connected to global clocks.

Clock domains have a fixed height of 50 interconnect tiles centered around the horizontal clock lines (25 above, 25 below). Various function tiles, such as *CLBs*, are attached to interconnect tiles.

Within an FPGA, various memories (latches, block RAMs, distributed RAMs) contain the state of signal routing, *BEL* configuration, and runtime storage. Configuration is the process of loading an initial state into all of these memories both to define the intended logic operations as well as set initial data for runtime memories. Note that the same mechanisms used for configuration are also capable of reading out the active state of these memories as well. This can be used to examine the contents of a block RAM or other memory at any point in the device's operation.

2.1 Addressing

As described in *Overview*, 7-Series FPGAs are constructed out of *tiles* organized into *clock domains*. Each tile contains a set of *BELs* and the memories used to configure them. Uniquely addressing each of these memories involves first identifying the *horizontal clock row*, then the tile within that row, and finally the specific bit within the tile.

Horizontal clock row addressing follows the hierarchical structure described in *Overview* with a single bit used to indicate top or bottom half and a 5-bit integer to encode the row number. Within the row, tiles are connected to one or more configuration busses depending on the type of tile and what configuration memories it contains. These busses are identified by a 3-bit integer:

Address	Name	Connected tile type
000	CLB, I/O, CLB	Interconnect (INT)
001	Block RAM content	Block RAM (BRAM)
010	CFG_CLB	???

Within each bus, the connected tiles are organized into columns. A column roughly corresponds to a physical vertical line of tiles perpendicular to and centered over the horizontal clock row. Each column contains varying amounts of configuration data depending on the types of tiles attached to that column. Regardless of the amount, a column's configuration data is organized into a multiple of *frames*. Each frame consists of 101 words with 100 words for the connected tiles and 1 word for the horizontal clock row. The 7-bit address used to identify a specific frame within the column is called the minor address.

Putting all these pieces together, a 32-bit frame address is constructed:

Field	Bits
Reserved	31:26
Bus	25:23
Top/Bottom Half	22
Row	21:17
Column	16:7
Minor	6:0

2.1.1 CLB, I/O, CLB

Columns on this bus are comprised of 50 directly-attached interconnect tiles with various kinds of tiles connected behind them. Frames are striped across the interconnect tiles with each tile receiving 2 words out of the frame. The number of frames in a column depends on the type of tiles connected behind the interconnect. For example, interconnect tiles always have 26 frames and a CLBL tile has an additional 12 frames so a column of CLBs will have 36 frames.

2.1.2 Block RAM content

As the name says, this bus provides access to the Block RAM contents. Block RAM configuration data is accessed via the CLB, I/O, CLB bus. The mapping of frame words to memory locations is not currently understood.

2.1.3 CFG_CLB

While mentioned in a few places, this bus type has not been seen in any bitstreams for Artix7 so far.

2.2 Loading sequence

Todo: Expand on these rough notes.

- Device is configured via a state machine controlled via a set of registers
- CRC of register writes is checked against expected values to verify data integrity during transmission.
- Before writing frame data:
 - IDCODE for configuration's target device is checked against actual device
 - Watchdog timer is disabled
 - Start-up sequence clock is selected and configured
 - Start-up signal assertion timing is configured
 - Interconnect is placed into Hi-Z state
- Data is then written by:
 - Loading a starting address
 - Selecting the write configuration command
 - Writing configuration data to data input register

- * Writes must be in multiples of the frame size
- * Multi-frame writes trigger autoincrementing of the frame address
- * Autoincrement can be disabled via bit in COR1 register.
- * At the end of a row, 2 frames of zeros must be inserted before data for the next row.
- After the write has finished, the device is restarted by:
 - Strobing a signal to activate IOB/CLB configuration flip-flops
 - Reactivate interconnect
 - Arms start-up sequence to run after desync
 - Desynchronizes the device from the configuration port
- Status register provides detail of start-up phases and which signals are asserted

2.3 Other

- ECC of frame data is contained in word 50 alongside horizontal clock row configuration
- Loading will succeed even with incorrect ECC data
- ECC is primarily used for runtime bit-flip detection

CHAPTER 3

Bitstream format

Todo: Expand on rough notes

- Specific byte pattern at beginning of file to allow hardware to determine width of bus providing configuration data.
- Rest of file is 32-bit big-endian words
- All data before 32-bit synchronization word (0xAA995566) is ignored by configuration state machine
- Packetized format used to perform register reads/writes * Three packet header types
 - Type 0 packets exist only when performing zero-fill between rows
 - Type 1 used for writes up to 4096 words
 - Type 2 expands word count field to 27 bits by omitting register address
 - Type 2 must always be preceded by Type 1 which sets register address
 - NOP packets are used for inserting required delays
 - Most registers only accept 1 word of data
 - Allowed register operations depends on interface used to send packets
 - * Writing LOUT via JTAG is treated as a bad command
 - * Single-frame FDRI writes via JTAG fail
- CRC
 - Calculated automatically from writes: register address and data written
 - Expected value is written to CRC register
 - If there is a mismatch, error is flagged in status register
 - Writes to CRC register can be safely removed from a bitstream
 - Alternatively, replace with write to command register to reset calculated CRC value

- Xilinx BIT header
 - Additional information about how bitstream was generated
 - Unofficially documented at http://www.fpga-faq.com/FAQ_Pages/0026_Tell_me_about_bit_files.htm
 - Really does require NULL-terminated Pascal strings
 - Having this header is the distinction between .bin and .bit in Vivado
 - Is ignored entirely by devices

basic element

BEL

basic logic element

BLE For example a LUT5, LUT6, CARRY4, or MUX, but not PIPs.

BELs come in two types:

- Basic BEL - A logic unit which does things.
- Routing BEL - A unit which is statically configured at the routing time.

bitstream Binary data that is directly loaded into an FPGA to perform configuration. Contains configuration *frames* as well as programming sequences and other commands required to load and activate same.

clock domain Portion of a *horizontal clock row* to one side of the global clock spine. Often refers to *tiles* that are associated with these clocks.

column Collection of *tiles* physically organized as a vertical line.

configurable logic block

CLB Basic building block of logic.

frame Fundamental unit of configuration data consisting of 101 *words*.

half Portion of a device defined by a virtual line dividing the two sets of global clock buffers present in a device. The two halves are simply referred to as the top and bottom halves.

node Collection of *wires* spanning one or more tiles.

programmable interconnect point

PIP Connection point between two wires in a tile that may be enabled or disabled by the configuration.

horizontal clock row Portion of a device including 12 horizontal clocks and the 50 interconnect and function tiles associated with them. A *half* contains one or more horizontal clock rows and each half may have a different number of rows.

site Portion of a tile where *BELs* can be placed. *Slices* in a *CLB* tile are sites.

slice Portion of a *CLB* tile that contains *BELs*.

tile Fundamental unit of physical structure containing a single type of resource or function.

wire Physical wire within a *tile*.

word 32-bits stored in big-endian order. Fundamental unit of *bitstream* format.

B

basic element, [11](#)
basic logic element, [11](#)
BEL, [11](#)
bitstream, [11](#)
BLE, [11](#)

C

CLB, [11](#)
clock domain, [11](#)
column, [11](#)
configurable logic block, [11](#)

F

frame, [11](#)

H

half, [11](#)
horizontal clock row, [11](#)

N

node, [11](#)

P

PIP, [11](#)
programmable interconnect point, [11](#)

S

site, [12](#)
slice, [12](#)

T

tile, [12](#)

W

wire, [12](#)
word, [12](#)