Summer Training Report

On

Unlocking the Potential of FPGAs

Submitted in partial fulfillment of the requirements for the completion of six weeks' summer internship/training [ ]

Name: Harsh Kumar Sharma

Enrollment Number: 07019011721

Under The Supervision Of

Mr. Vaibhav Saini

(Scientist F, IRDE DRDO Dehradun)

UNIVERSITY SCHOOL OF AUTOMATION AND ROBOTICS
GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY
EAST DELHI CAMPUS, SURAJMAL VIHAR, DELHI- 110032

DECLARATION

I hereby declare that the Summer Training Report entitled(Unlocking the Potential of FPGAs) is an authentic record of work completed as requirements of Summer Training ( ) during the period from 21st June'24 to 30th Aug'24 in Instruments Research & Development Establishment DRDO, under the supervision of **Mr. Vaibhav Saini (Scientist F, IRDE Dehradun)**.

Harsh Kumar Sharma

B.Tech IIoT (USAR)

07019011721

Date:

Mr. Vaibhav Saini

(Scientist F, IRDE Dehradun)

Date:

Acknowledgement

I would like to express my special thanks to my mentor **Mr. Vaibhav Saini (Scientist F, IRDE Dehradun)** for his time and efforts he provided throughout the duration of summer internship [ ] for completion of the project titled 'Unlocking the Potential of FPGAs'. His useful advice and suggestions were helpful to me during the project's completion. I'd also like to thank my friends and parents for their support and encouragement as I worked on this assignment. I would like to acknowledge that this project was completed entirely by me and not by someone else.

Harsh Kumar Sharma

07019011721

IIoT (USAR)

INDEX

Table of figures

ABSTRACT

This research investigated the potential of Trion T8 and T120 FPGAs for accelerating critical applications in the realm of edge computing. As part of this research conducted, a comprehensive exploration of FPGA architecture, development tools, and design methodologies was undertaken. Practical experimentation was conducted to assess power efficiency, speed, and resource utilization. Our findings underscore the FPGA's potential to significantly accelerate computationally intensive tasks such as image and signal processing, . However, design complexities and resource constraints remain significant challenges. To fully harness the FPGA's capabilities, a concerted effort is required to develop efficient design tools, libraries, and methodologies . Additionally, exploring FPGA-based hardware acceleration for emerging technologies like artificial intelligence and machine learning is critical. This research provides a foundational understanding to propel future advancements in FPGA-based systems. Specific attention to low-power design, radiation hardening techniques, and real-time performance optimization, coupled with rigorous testing and validation, will be crucial for deploying FPGA-based solutions in harsh environments.

1. Introduction

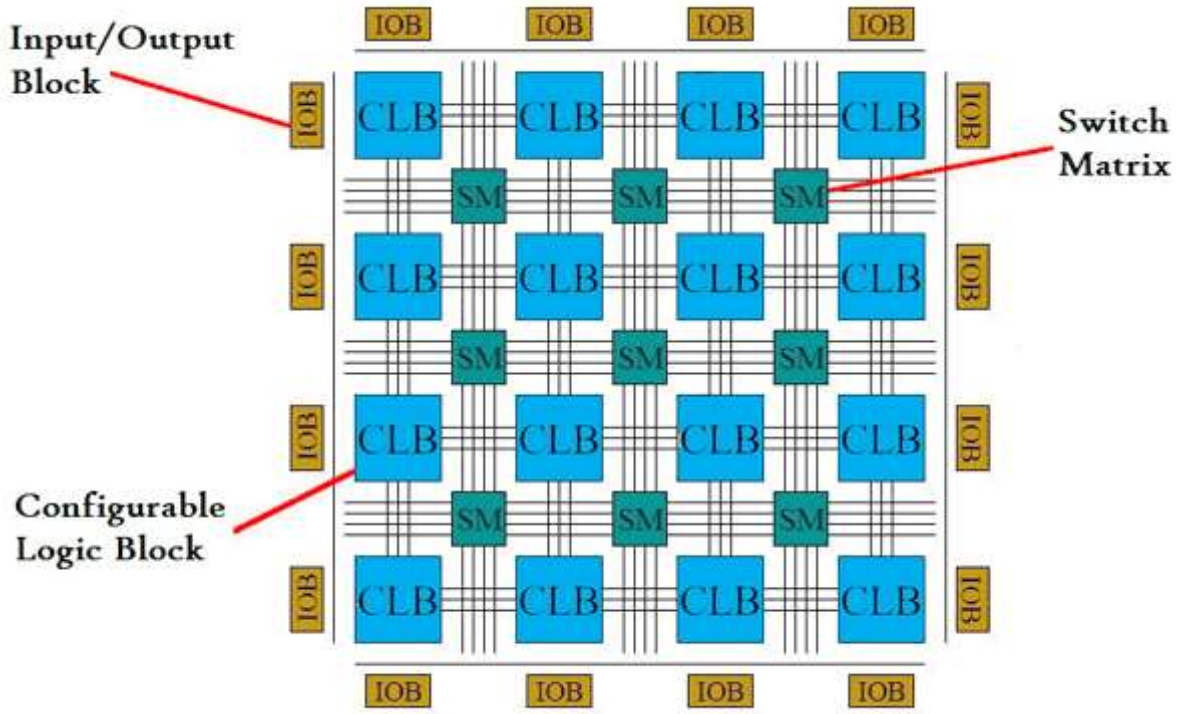
Field Programmable Gate Arrays (FPGAs) are integrated circuits often sold off-the-shelf. They're referred to as 'field programmable' because they provide customers with the ability to reconfigure the hardware to meet specific use case requirements after the manufacturing process. This allows for feature upgrades and bug fixes to be performed in situ, which is especially useful for remote deployments.

FPGAs contain configurable logic blocks (CLBs) and a set of programmable interconnects that allow the designer to connect blocks and configure them to perform everything from simple logic gates to complex functions. Full SoC designs containing multiple processes can be put onto a single FPGA device.

2.1. FPGA Architecture

An FPGA has a regular structure of logic cells or modules and interlinks which is under the developers and designers' complete control. The FPGA is built with mainly three major blocks such as **Configurable Logic Block (CLB), I/O Blocks or Pads and Switch Matrix/ Interconnection Wires**. Each block will be discussed below in brief.

- **CLB (Configurable Logic Block):** These are the basic cells of FPGA. It consists of one 8-bit function generator, two 16-bit function generators, two registers (flip-flops or latches), and reprogrammable routing controls (multiplexers). CLBs are applied to implement other designed functions and macros. Each CLBs have inputs on each side which makes them flexile for the mapping and partitioning of logic.
- **I/O Pads or Blocks:** The Input/Output pads are used for the outside peripherals to access the functions of FPGA and using the I/O pads it can also communicate with FPGA for different applications using different peripherals.
- **Switch Matrix/ Interconnection Wires:** Switch Matrix is used in FPGA to connect the long and short interconnection wires together in flexible combination. It also contains the transistors to turn on/off connections between different lines.

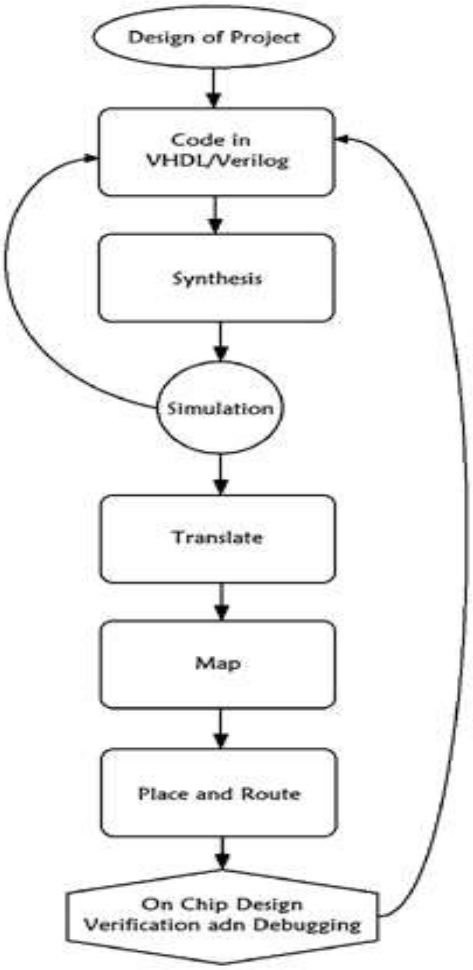


1.FPGA Architecture

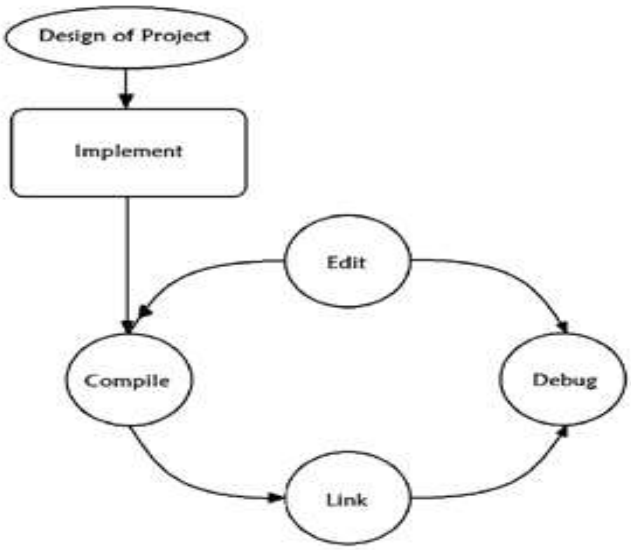
2.2. <u>FPGA programming</u> involves creating a digital circuit using Hardware Description Languages (HDLs) like VHDL or Verilog. Unlike software, which is sequential, FPGA code defines circuit behavior concurrently.

<u>Key steps in FPGA programming:</u>

- Design entry: Creating the circuit's behavior using HDL.
- Synthesis: Translating HDL code into a netlist (circuit description).
- Simulation: Verifying circuit functionality using simulation tools.
- Implementation: Mapping the design to the FPGA's hardware resources (place and route).
- Bitstream generation: Creating a configuration file to program the FPGA.
- Verification and debugging: Ensuring the design work as intended on the FPGA.
- FPGA design flow differs significantly from software development, with longer compile times and a focus on hardware architecture.



(a) Hardware Design Flow



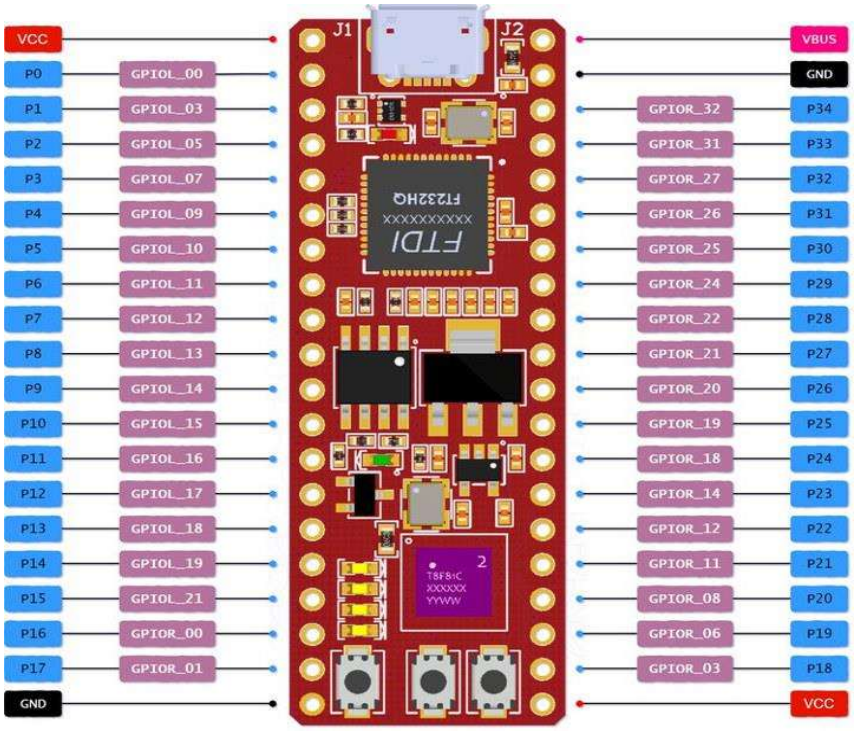
(b) Software Design Flow

2. Design flow in FPGA programming

3. TRION T8

The T8 FPGA features the high-density, low-power Efinix® Quantum® architecture wrapped with an I/O interface in a small footprint package for easy integration. T8 FPGAs support mobile, consumer, and IoT edge markets that need low power, low cost, and a small form factor. With ultra-low power T8 FPGAs, designers can build products that are always on, providing enhanced capabilities for applications such as embedded vision, voice and gesture recognition, intelligent sensor hubs, and power management

3.1 Features



3.Trion T8 Board

• High-density, low-power Quantum®

architecture

• Built on SMIC 40 nm process

• Less than 150 µA typical core leakage current at 1.1 V

• Ultra-small footprint package options

• FPGA interface blocks

— GPIO

— PLL

— LVDS 600 Mbps per lane with up to 6 TX pairs and 6 RX pairs

— Oscillator

• Programmable high-performance I/O

— Supports 1.8, 2.5, and 3.3 V single-ended I/O standards and interfaces

• Flexible on-chip clocking

— 12 low-skew global clock signals can be driven from off-chip external clock signals or PLL synthesized clock signals

— PLL support

• Flexible device configuration

— Standard SPI interface (active, passive, and daisy chain)

— JTAG interface

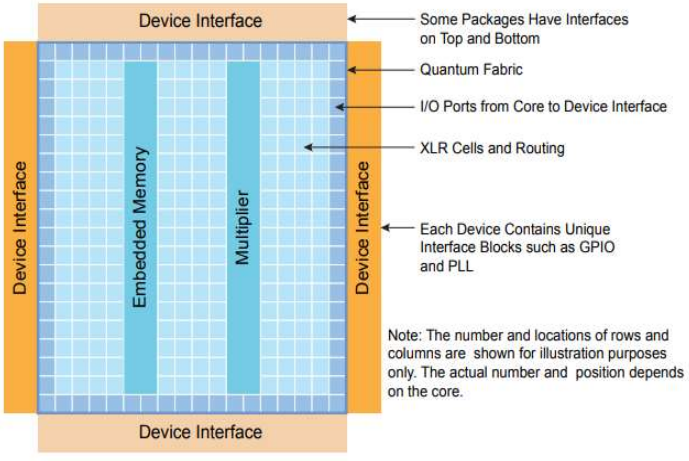
— Optional Mask Programmable Memory (MPM) capability

• Fully supported by the Efinity®

software, an RTL-to-bitstream compiler

3.2. Device Core Functional Description

T8 FPGAs feature an exchangeable Logic and Routing (XLR) cell that Efinix has optimized for a variety of applications. Trion® FPGAs contain three building blocks constructed from XLR cells: logic elements, embedded memory blocks, and multipliers. Each FPGA in the Trion® family has a custom number of building blocks to fit specific application needs. As shown in the following figure, the FPGA includes I/O ports on all four sides, as well as columns of XLR cells, memory, and multipliers. A control block within the FPGA handles configuration.

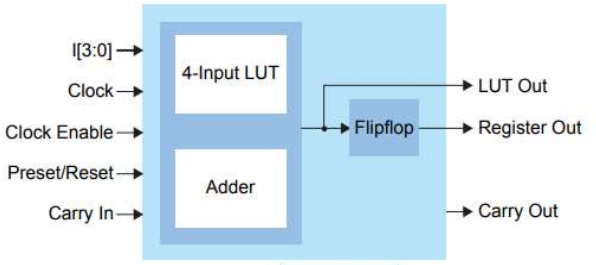


4. T8 FPGA Block Diagram

3.2.1. **The XLR cell** is the fundamental building block of Efinix's Quantum architecture. This unique cell can function as both logic and routing, providing flexibility and efficiency. By combining these functions, Efinix can reduce the number of transistors needed, leading to smaller and more power-efficient FPGAs

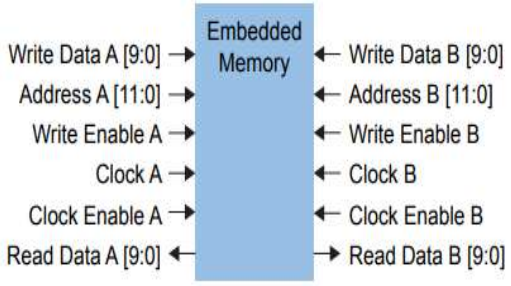
3.2.2. **A logic cell** is the fundamental building block within an FPGA. It typically consists of a Look-Up Table (LUT) and a flip-flop. The LUT can perform any

Boolean function on its inputs, while the flip-flop stores data. By combining multiple logic cells, complex digital circuits, including arithmetic units and counters, can be constructed.



5. Logic Cell Block Diagram

3.2.3. **Embedded Memory FPGAs** contain built-in memory blocks of 5kb each. These blocks are versatile, serving as RAM, dual-port RAM, FIFOs, or ROM. They can be combined to create larger memory arrays. Initial data can be loaded into the memory during the FPGA configuration process.



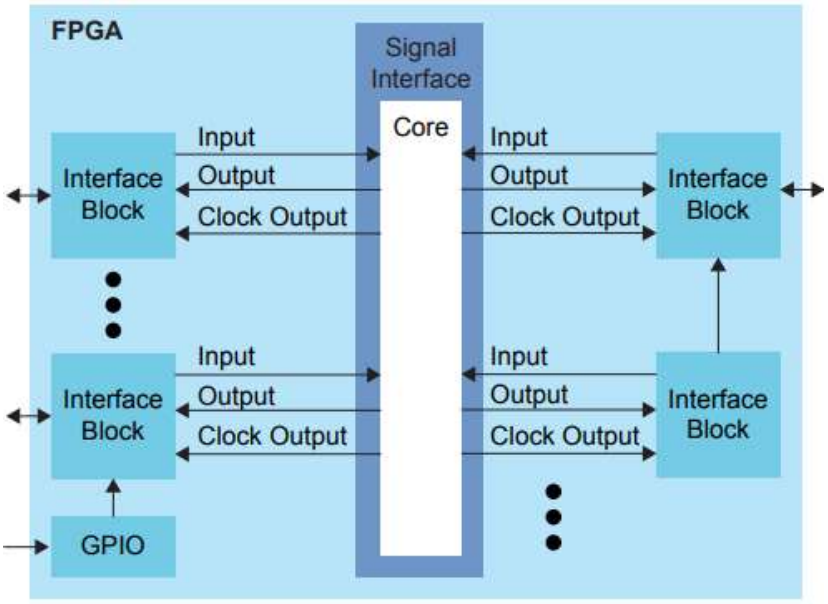
6. Embedded Memory Block Diagram (True Dual-Port Mode)

3.3 Device Interface Functional Description

The device interface wraps the core and routes signals between the core and the device I/O pads through a signal interface. Because they use the flexible Quantum® architecture, devices in the Trion® family support a variety of interfaces to meet the needs of different applications.

3.3.1 Interface Block Connectivity

The FPGA core fabric connects to the interface blocks through a signal interface. The interface blocks then connect to the package pins. The core connects to the interface blocks using three types of signals: • Input—Input data or clock to the FPGA core • Output—Output from the FPGA core • Clock output—Clock signal from the core clock tree.



7. Interface Block and Core Connectivity

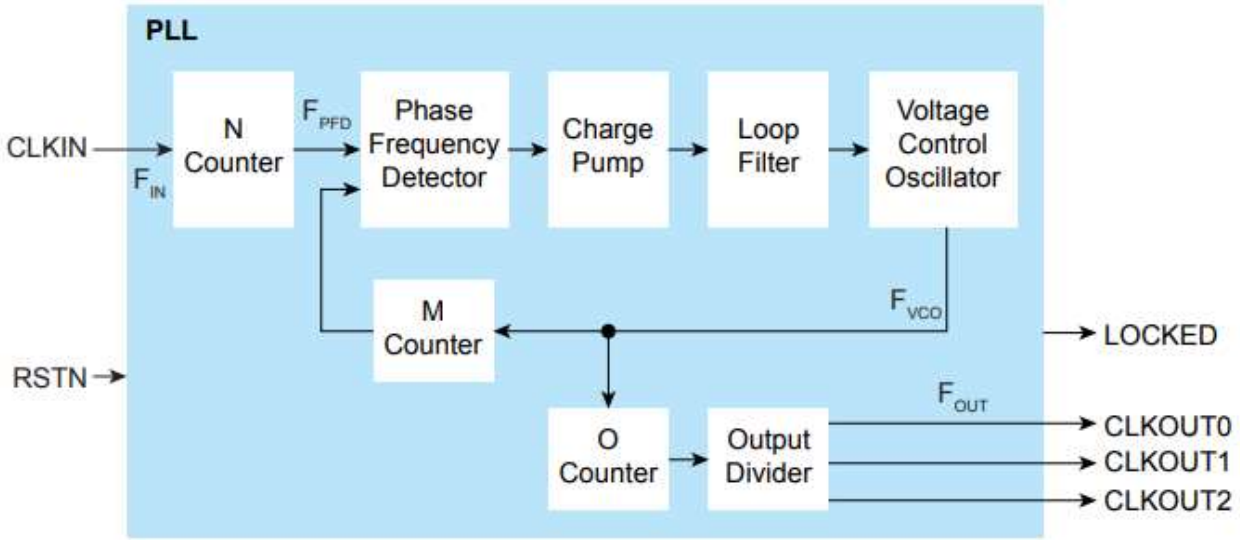
GPIO blocks in Trion FPGAs are highly flexible, offering multiple operational modes. They can be configured as inputs, outputs, or even bypass signals directly to other interface blocks. This versatility is crucial for efficient design.

Importantly, GPIOs used for specific functions like reference clocks or clock outputs are managed within the interface block configuration rather than the core design itself. This distinction is essential for understanding the FPGA's overall architecture and design flow.

GPIO Mode	Description
Input	Only the input path is enabled; optionally registered. If registered, the input path uses the input clock to control the registers (positively or negatively triggered). Select the alternate input path to drive the alternate function of the GPIO. The alternate path cannot be registered. **QFP144 packages:** In DDIO mode, two registers sample the data on the positive and negative edges of the input clock, creating two data streams.
Output	Only the output path is enabled; optionally registered. If registered, the output path uses the output clock to control the registers (positively or negatively triggered). The output register can be inverted. **QFP144 packages:** In DDIO mode, two registers capture the data on the positive and negative edges of the output clock, multiplexing them into one data stream.
Bidirectional	The input, output, and OE paths are enabled; optionally registered. If registered, the input clock controls the input register, the output clock controls the output and OE registers. All registers can be positively or negatively triggered. Additionally, the input and output paths can be registered independently. The output register can be inverted.
Clock output	Clock output path is enabled.

3.4 Simple PLL

T8 FPGAs in BGA49 and BGA81 packages have a simple PLL. The T8 has 1 PLL to synthesize clock frequencies. The PLL's reference clock input comes from a dedicated GPIO's alternate input pin. The PLL consists of a pre-divider counter (N counter), a feedback multiplier counter (M counter), post-divider counter (O counter), and an output divider per clock output.



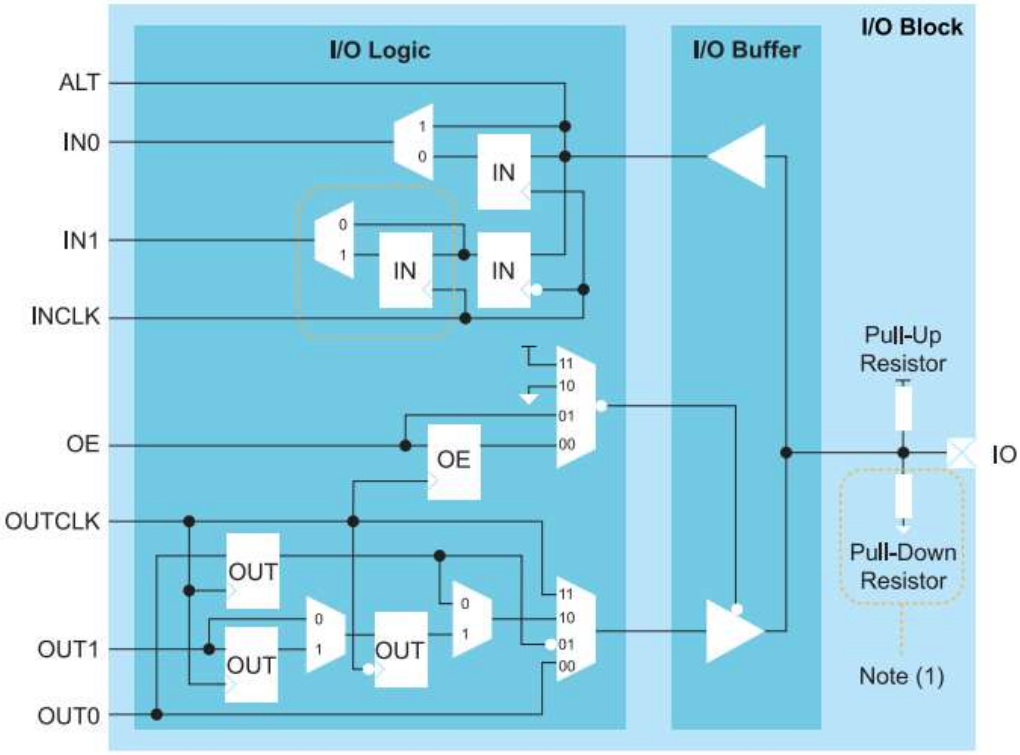
8. T8 PLL Block Diagram

Port	Direction	Description
CLKIN	Input	Reference clock. This port is also a GPIO pin; the GPIO pins' alternate function is configured as a reference clock.
RSTN	Input	Active-low PLL reset signal. When asserted, this signal resets the PLL; when de-asserted, it enables the PLL. Connect this signal in your design to power up or reset the PLL. Assert the RSTN pin for a minimum pulse of 10 ns to reset the PLL.
CLKOUT0 CLKOUT1 CLKOUT2	Output	PLL output. The designer can route these signals as input clocks to the core's GCLK network.
LOCKED	Output	Goes high when PLL achieves lock; goes low when a loss of lock is detected. Connect this signal in your design to monitor the lock status. This signal is analog asynchronous.

9. PLL Pins

T8 QFP144 Interface Description

T8 FPGAs in QFP144 packages have advanced general-purpose I/O logic and buffers, I/O banks, advanced PLLs, and supports LVDS interface. Complex I/O Buffer



10. I/O Interface Block

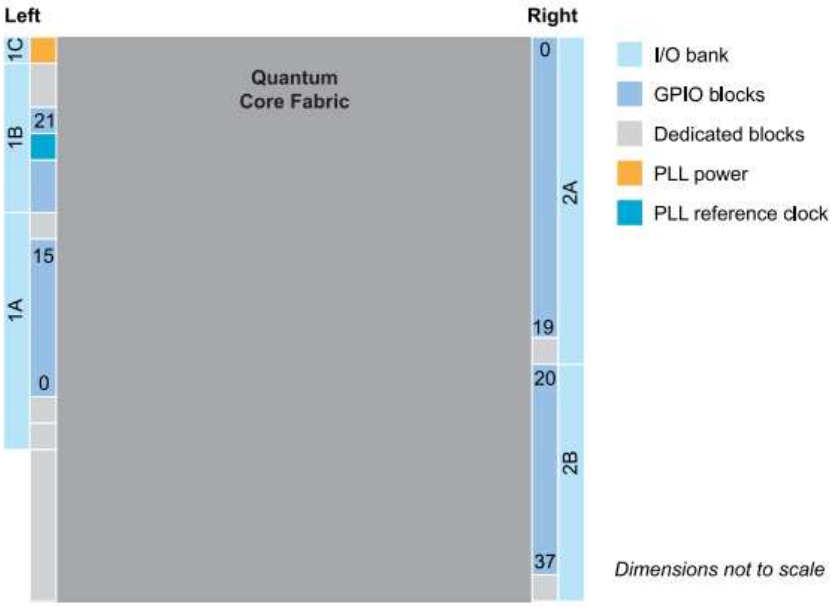
3.5 T8 Interface Floorplan

Note: The numbers in the floorplan figures indicate the GPIO and LVDS number ranges. Some packages may not have all GPIO or LVDS pins in the range bonded out. Refer to the T8 pinout for information on which pins are available in each package.

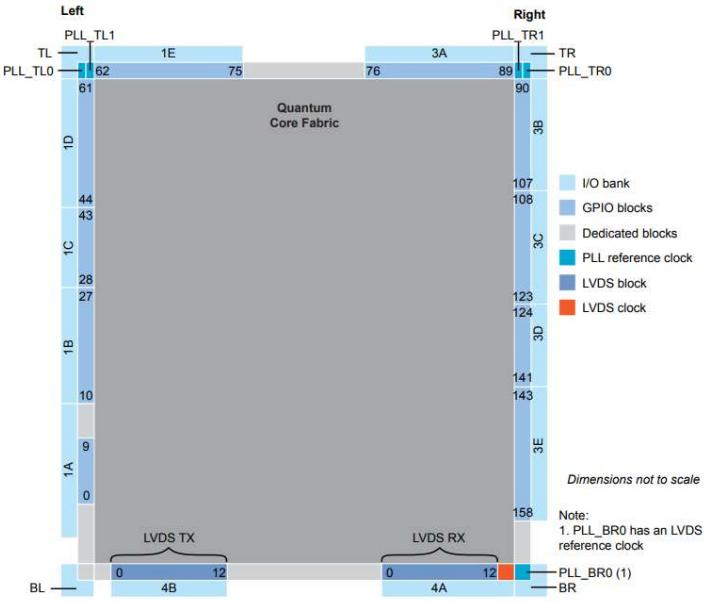
Key Points:

- GPIO and LVDS Number Ranges: The floorplan uses numbers to denote specific ranges of GPIO and LVDS pins. This is a common method to group related pins for efficient routing and design.
- Not All Pins Bonded Out: While a range of GPIO or LVDS pins might be specified, not all pins within that range are necessarily available on a particular package. The actual pinout for a specific package will determine which pins are accessible.

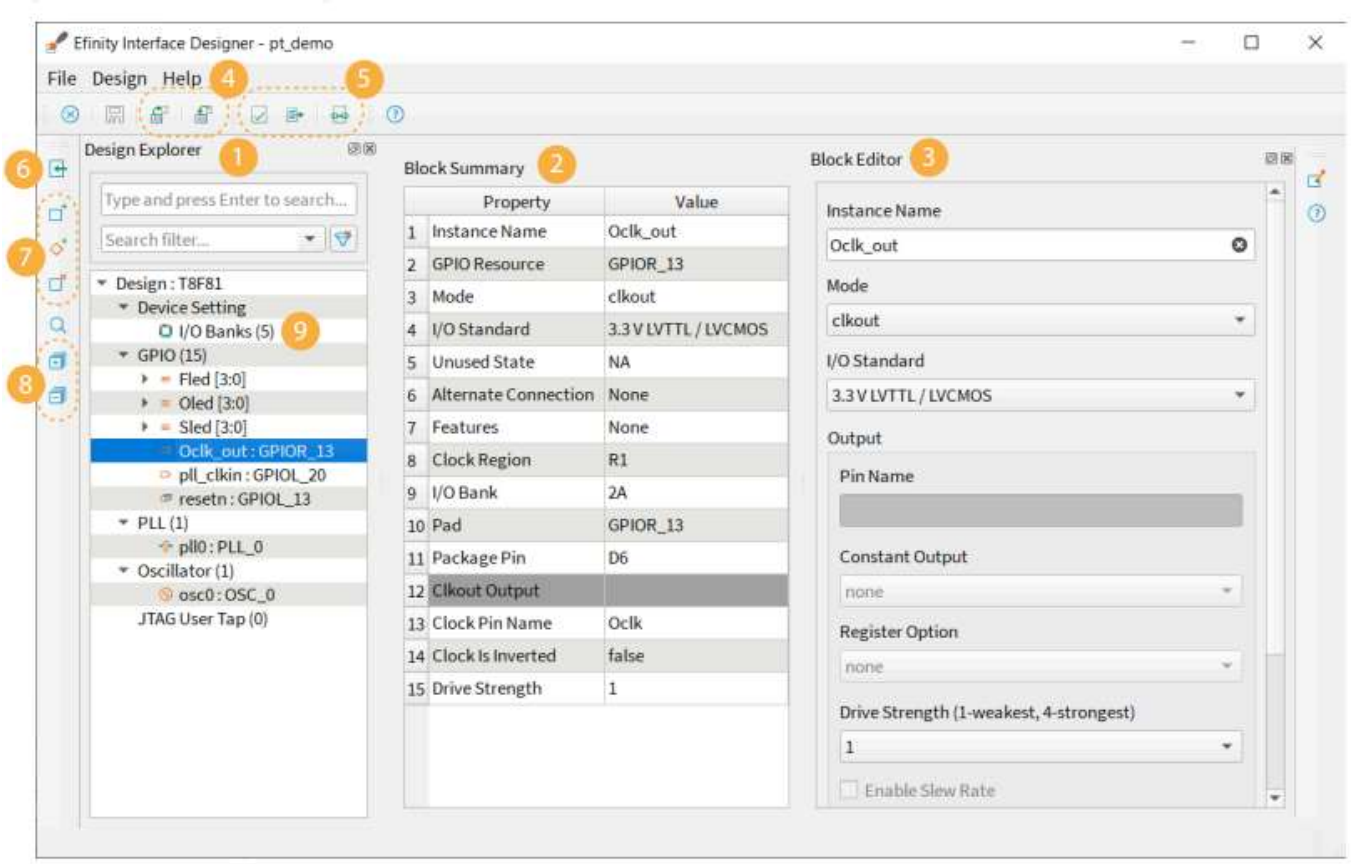
- T8 Pinout Reference: To accurately identify which GPIO or LVDS pins are available on a specific T8 package, it's essential to consult the detailed pinout documentation for that package.
-



11. Floorplan Diagram for BGA49 and BGA81 Packages



12. Floorplan Diagram for QFP144 Packages



Notes: 1. The Design Explorer shows the interface blocks in your design. They are organized by block type. 2. The block summary shows the settings for the block selected in the Design Explorer. 3. Use the Block Editor to add or change settings for the interface block. 4. You can import or export GPIO resource assignments using a .csv or .isf file. 5. Use the project management tools to perform design checks, view reports, generate constraints, etc. 6. Click Show/Hide Resource Assigner to toggle a tabular view of assignments. 7. Use the block tools to add or delete blocks and buses. 8. Expand or collapse the Design Explorer folders. 9. The number in parentheses shows the number of used blocks.

4. TRION T120

The T120 FPGA features the high-density, low-power Efinix® Quantum® architecture wrapped with an I/O interface for easy integration. With a high I/O to logic ratio and differential I/O support, T120 FPGAs support a variety of applications that need wide I/O connectivity. The T120 also includes a MIPI D-PHY with a built-in, royalty-free CSI-2 controller, which is the most popular camera interface used in the mobile industry. Additionally, T120 FPGAs support a DDR3, LPDDR3, LPDDR2 PHY with memory controller hard IP that provides faster access to data stored in memory. The carefully tailored combination of core resources and I/O provides enhanced capability for applications such as embedded vision, voice and gesture recognition, intelligent sensor hubs, power management, and LED drivers.

4.1 Features

- High-density, low-power Quantum® architecture
- Built on SMIC 40 nm process
- FPGA interface blocks

— GPIO

— PLL

— LVDS 800 Mbps per lane with up to 52 TX pairs and 52 RX pairs

— MIPI DPHY with CSI-2 controller hard IP, 1.5 Gbps per lane

— DDR3, DDR3L, LPDDR3, LPDDR2 x32 PHY (supporting x16 or x32 DQ widths) with memory controller hard IP, up to 1066 Mbps

- Programmable high-performance I/O

— Supports 1.8, 2.5, and 3.3 V single-ended I/O standards and interfaces

- Flexible on-chip clocking

— 16 low-skew global clock signals can be driven from off-chip external clock signals or PLL synthesized clock signals

— PLL support

• Flexible device configuration

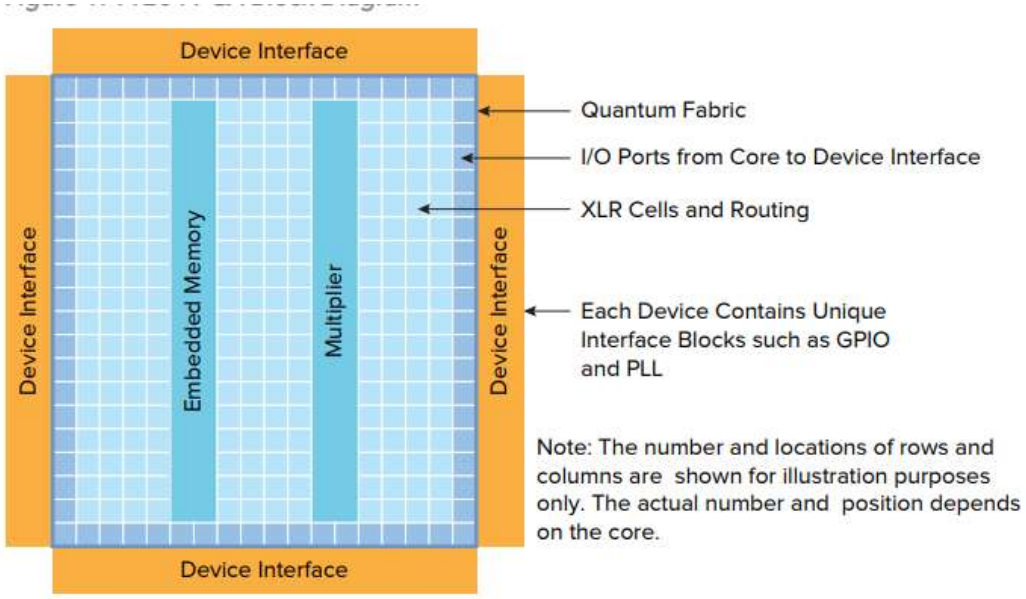
— Standard SPI interface (active, passive, and daisy chain)

— JTAG interface

• Fully supported by the Efinity® software, an RTL-to-bitstream compiler

4.2 Device Core Functional Description

T120 FPGAs utilize a flexible architecture built around eXchangeable Logic and Routing (XLR) cells, which can be configured for logic or routing functions. These cells combine to form logic elements, memory, and multipliers, with the number of each block varying per FPGA model to accommodate diverse application requirements. The FPGA features I/O ports on all sides and internal columns housing these building blocks, while a dedicated control block manages configuration.



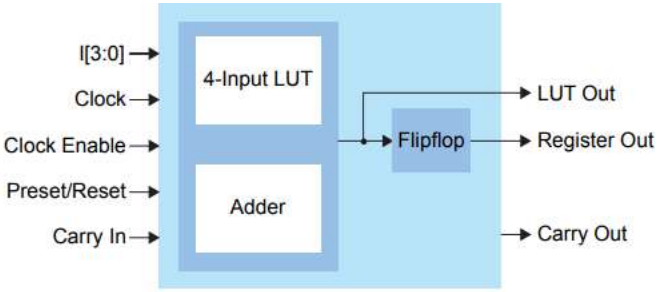
13.T120 FPGA Block Diagram

4.2.1 XLR Cell

The eXchangeable Logic and Routing (XLR) cell is the basic building block of the Quantum® architecture. The Efinix XLR cell combines logic and routing and supports both functions interchangeably. This unique innovation greatly enhances the transistor flexibility and utilization rate, thereby reducing transistor counts and silicon area significantly.

4.2.2 Logic Cell

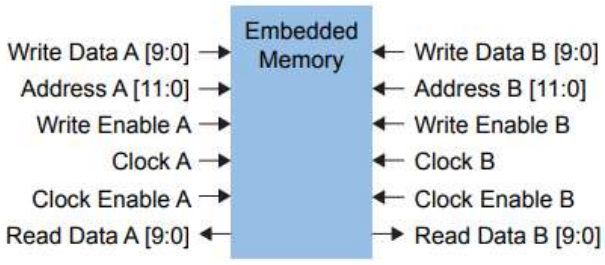
The logic cell comprises a 4-input LUT or a full adder plus a register (flipflop). You can program each LUT as any combinational logic function with four inputs. You can configure multiple logic cells to implement arithmetic functions such as adders, subtractors, and counters.



14. Logic Cell Block Diagram

4.2.3 Embedded Memory

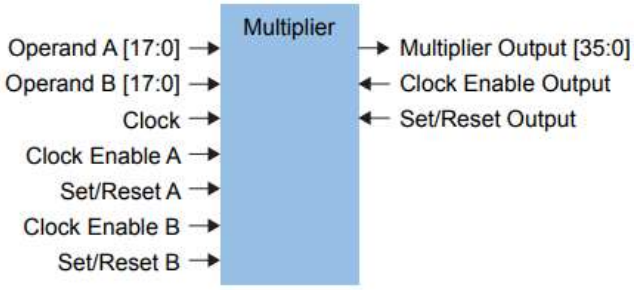
The T120 FPGA incorporates 5-kbit high-speed SRAM blocks. These versatile memory blocks can be configured as single or dual-port RAM, FIFOs, or ROM, with initial data loadable during configuration. The Efinity software allows for cascading multiple blocks into larger memory arrays, enabling customization of memory size and depth.



15. Embedded Memory Block Diagram (True Dual-Port Mode)

4.2.4 Multipliers

The FPGA has high-performance multipliers that support 18 x 18 fixed-point multiplication. Each multiplier takes two signed 18-bit input operands and generates a signed 36-bit output product. The multiplier has optional registers on the input and output ports.

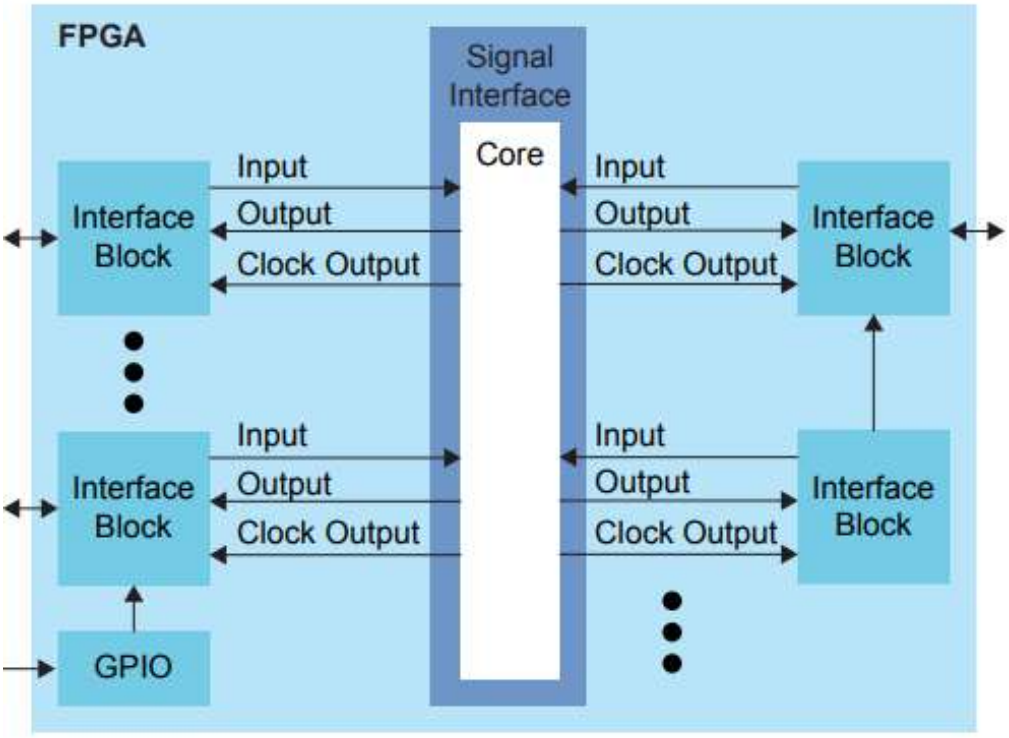


4.3 Device Interface Functional Description

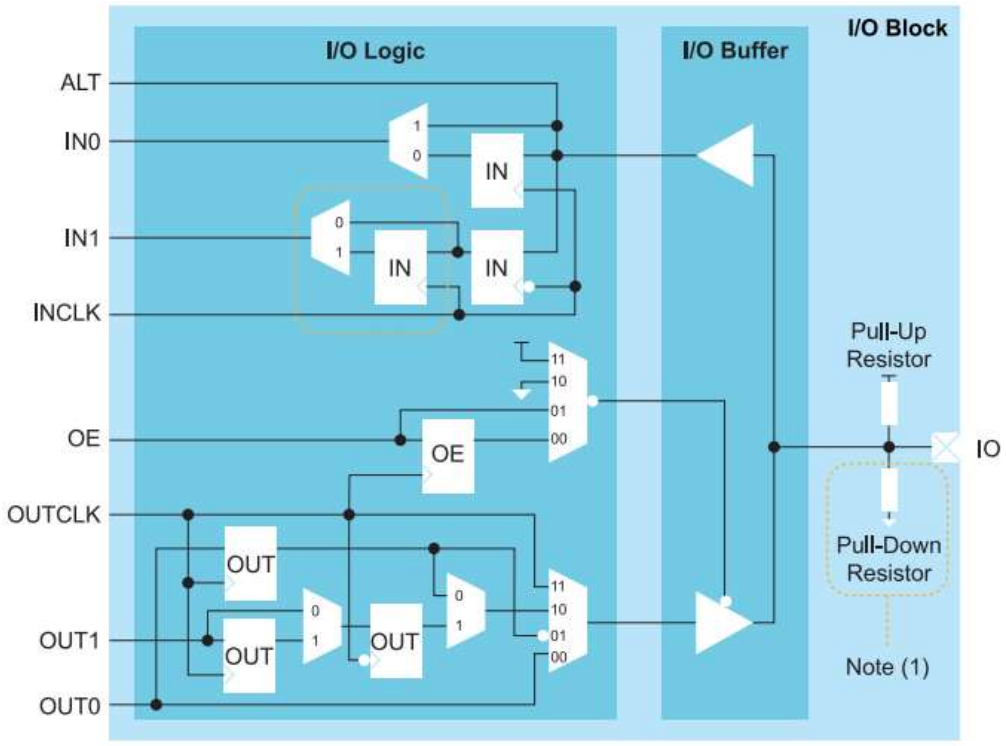
The device interface wraps the core and routes signals between the core and the device I/O pads through a signal interface. Because they use the flexible Quantum® architecture, devices in the Trion® family support a variety of interfaces to meet the needs of different applications.

Interface Block Connectivity The FPGA core fabric connects to the interface blocks through a signal interface. The interface blocks then connect to the package pins. The core connects to the interface blocks using three types of signals:

- Input—Input data or clock to the FPGA core
- Output—Output from the FPGA core
- Clock output—Clock signal from the core clock tree



16.Interface Block and Core Connectivity



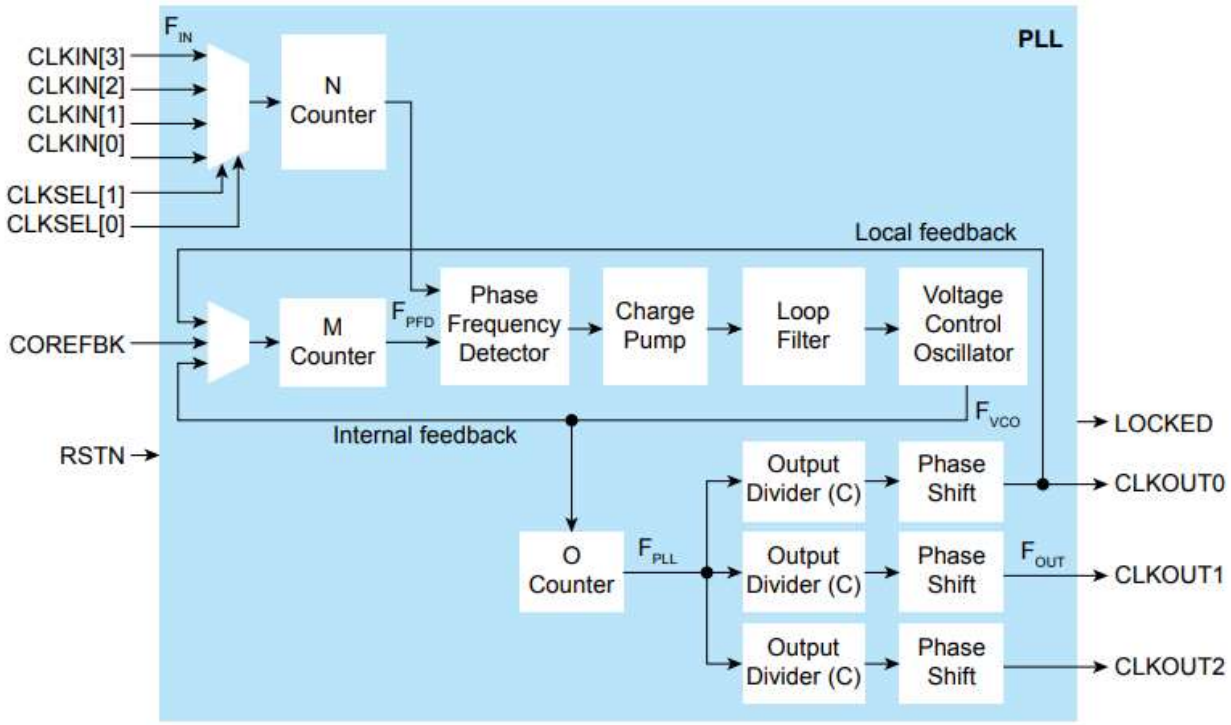
17.I/O Interface Block

4.4 PLL

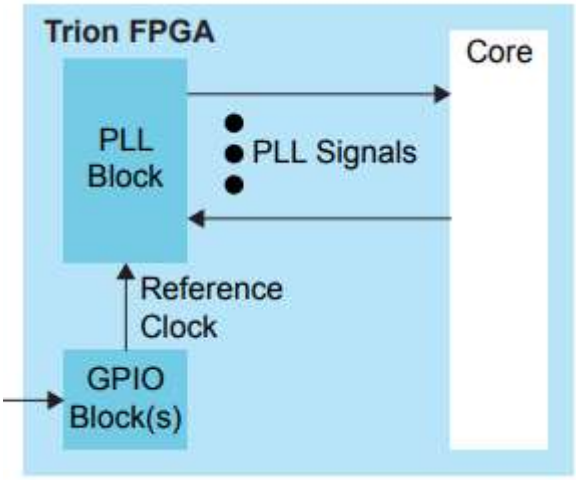
The T120 FPGA offers clock flexibility with **7 or 8 PLLs (depending on package)** for generating custom clock frequencies. These PLLs can adjust for timing delays (skew) using internal or external feedback.

Here's a breakdown of the key features:

- **Multiple Sources:** The PLL reference clock can come from up to four different sources.
- **Dynamic Selection:** You can switch the reference clock source on-the-fly using the CLKSEL port (but hold the PLL in reset during the switch).
- **LVDS Compatibility:** Some PLLs can accept an LVDS receiver buffer as the reference clock input.
- **Detailed Control:** Each PLL has internal counters for pre-division (N), multiplication (M), post-division (O), and final output division, allowing for precise clock frequency generation.



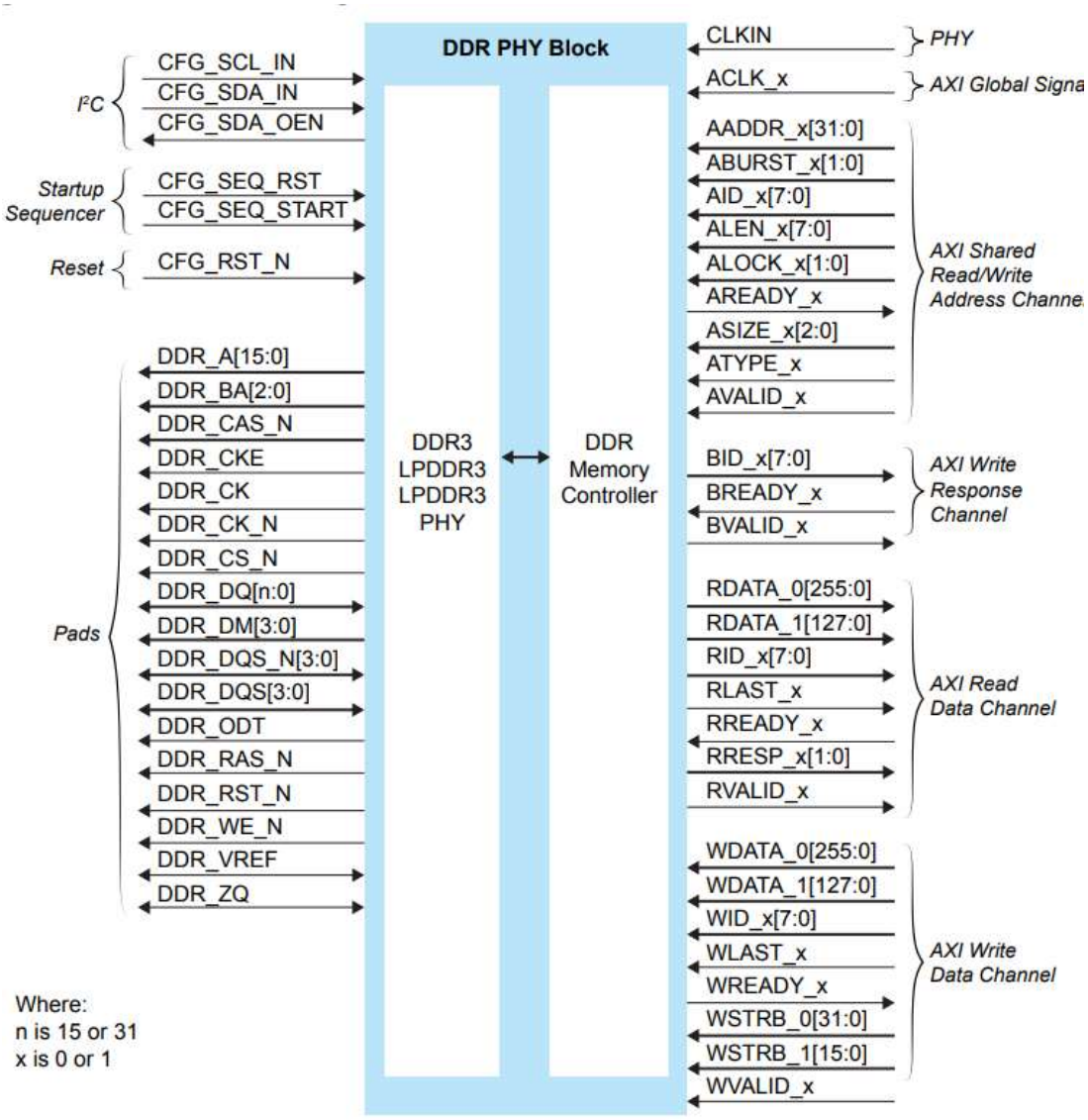
18.PLL Block Diagram



19.PLL Interface Block Diagram

4.5 DDR DRAM

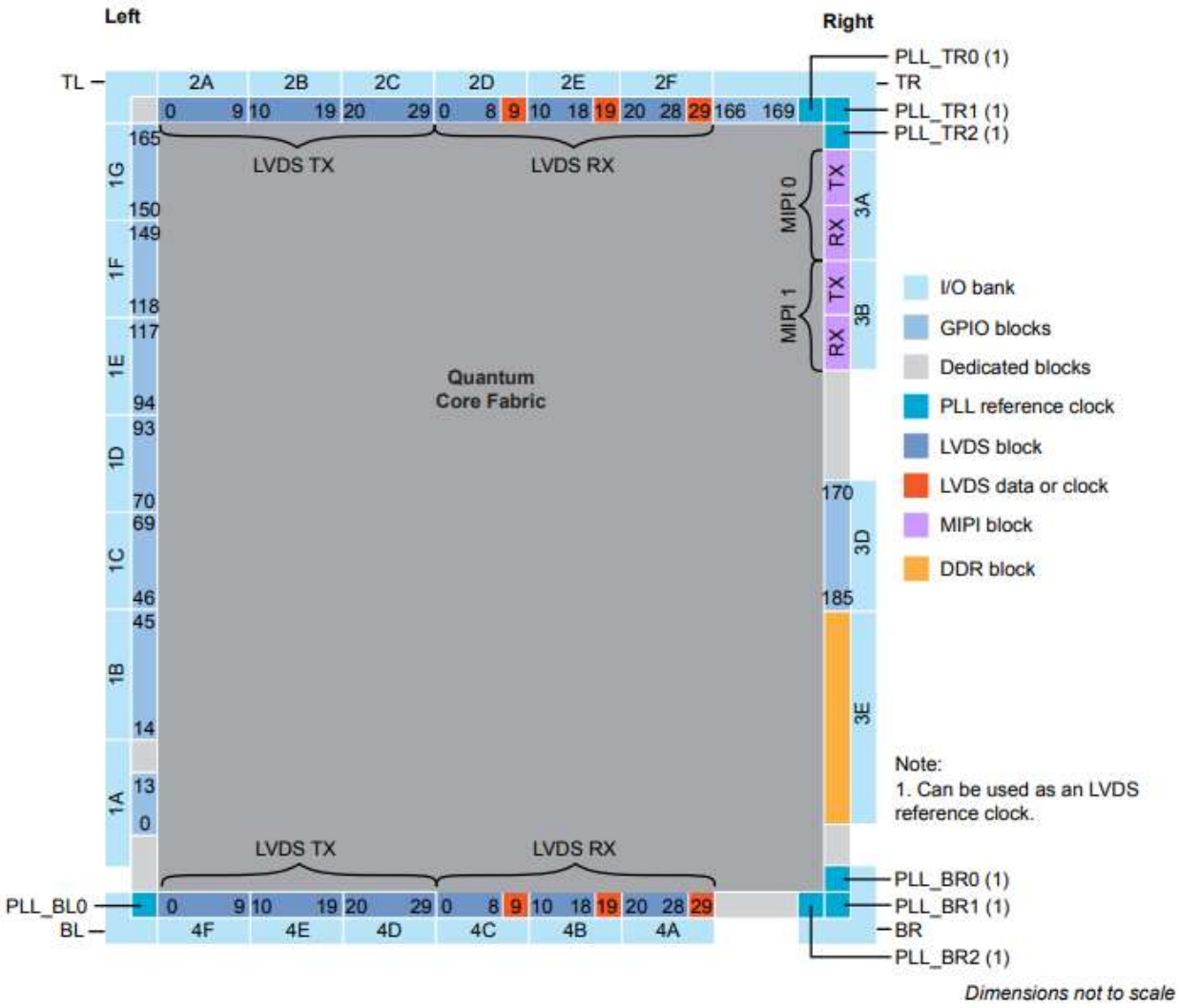
T120 FPGAs have a x32 DDR PHY interface supporting DDR3, DDR3L, LPDDR3, and LPDDR2 as well as a memory controller hard IP block. The DDR DRAM interface supports x16 or x32 DQ widths, depending on the package. The DDR PHY supports data rates up to 1066 Mbps per lane. The memory controller provides one 128 bit AXI bus and one 256 bit AXI bus to communicate with the FPGA core



The DDR DRAM block supports an I²C calibration bus that can read/write the DDR configuration registers. You can use this bus to fine tune the DDR PHY for high performance.

4.6 T120 Interface Floorplan

Note: The numbers in the floorplan figures indicate the GPIO and LVDS number ranges. Some packages may not have all GPIO or LVDS pins in the range bonded out. Refer to the T120 pinout for information on which pins are available in each package.



20. Floorplan Diagram for BGA324 Packages (with DDR and MIPI)

CONCLUSION

This research has demonstrated the potential of Trion T8 and T120 FPGAs as powerful tools for accelerating critical applications in edge computing. Through a comprehensive exploration of the FPGA architecture, development tools, and design methodologies , we have gained valuable insights into their capabilities.

Key Findings:

- FPGAs offer significant performance improvements for computationally intensive tasks like image and signal processing
- Practical experiments have confirmed their ability to achieve high speed and efficiency for these tasks.

Challenges and Future Directions:

- Design complexities and resource constraints still pose significant challenges.
- Development of efficient design tools, libraries, and methodologies is crucial to fully exploit FPGA capabilities.
- Investigating FPGA-based hardware acceleration for emerging technologies like AI and machine learning holds significant promise for future systems.

Recommendations for Future Research:

- To successfully deploy FPGA-based solutions in harsh environments, focus on:

- Low-power design techniques
- Radiation hardening strategies
- Real-time performance optimization
- Rigorous testing and validation procedures

By addressing these challenges and pursuing further research along the identified directions, this study paves the way for substantial advancements in FPGA-based systems, ultimately enhancing operational capabilities.

BIBLIOGRAPHY

1. **Efinix software**
   https://www.efinixinc.com/
2. **Trion T8 Data Sheet**
   https://www.efinixinc.com/docs/trion8-ds-v5.1.pdf
3. **Risc V software**
   https://www.efinixinc.com/products-efinity-riscv-ide.html
4. **Trion T120 Data Sheet**
   https://www.efinixinc.com/docs/trion120-ds-v3.7.pdf
5. **Sapphire High-Performance RISC-V SoC Data Sheet**
   https://www.efinixinc.com/support/docsdl.php?s=ef&pn=SAPPHIREHPBDS
6. **Sapphire High-Performance RISC-V SoC Hardware and Software User Guide**
   https://www.efinixinc.com/support/docsdl.php?s=ef&pn=SAPPHIREHPBUG

7. **Generating Pulse Width Modulation of Variable Duty Cycle Example**
   https://www.efinixinc.com/support/ed/generating-pulse-width-modulation-various-duty-cycles.php
8. **PLL with Multiple Output Frequencies Example**
   https://www.efinixinc.com/support/ed/pll-generating-multi-frequency-t8.php
9. **Configuring Trion FPGAs with Multiple Images Example**
   https://www.efinixinc.com/support/ed/configuring-fpgas-with-multiple-images.php
10. **Opal SoC Example (for Trion T8 BGA81 Development Board)**
    https://www.efinixinc.com/support/ed/t8-riscv-opal.php
11. **Trion T8 BGA81 Tutorial**
    https://www.efinixinc.com/support/ed/t8f81-tutorial.php

12. **Trion T120 BGA324 Ruby SoC with Frame Buffer Example**
    https://www.efinixinc.com/support/ed/t120f324-r5-framebuffer.php
13. **Ruby SoC with Triple-Speed Ethernet Example**
    https://www.efinixinc.com/support/ed/t120f324-tsemac-ruby.php

14. **T8 IBIS Models**
    https://www.efinixinc.com/support/docsdl.php?s=ef&pn=T8IBIS
15. **T8 FBGA81 BSDL Files**
    https://www.efinixinc.com/support/docsdl.php?s=ef&pn=T8_FBGA81