Arria™ 10 Device Architecture

Jeffrey Tyhach, Mike Hutton, Sean Atsatt, Arifur Rahman, Brad Vest, David Lewis, Martin Langhammer, Sergey Shumarayev, Tim Hoang, Allen Chan, Dong-Myung Choi, Dan Oh, Hae-Chang Lee, Jack Chui, Ket Chiew Sia, Edwin Kok, Wei-Yee Koay, and Boon-Jin Ang

Altera Corporation, San Jose, CA 95134, USA

***Abstract* — This paper presents the architecture of Arria 10, a high-density FPGA family built on the TSMC 20SOC process. The design of the device includes an embedded dual-core 1.5 GHz ARM A9 subsystem with peripherals, more than 1M logic elements (LEs) and 1.7M user flip-flops, and 64Mb of embedded memory organized into configurable memory blocks. The Arria 10 family is also the first mainstream FPGA family to include hardened single-precision IEEE 754 floating point, with an aggregate throughput of 1.3 TFLOPs. Device I/O consists of 28G programmable transceivers with an enhanced PMA architecture hardened PCIe sub-blocks and hardened DDR external memory controllers. New methods for digitally-assisted analog calibration are used to address process variation. The fabric is optimized for an aggressive die-size reduction and power improvement over 28nm FPGAs and includes features such as time-borrowing FFs for micro-retiming, tri-stated long-lines for improved routability, programmable back-bias at LAB-cluster granularity and power-management features such as Smart-VID for balancing leakage and performance across the process distribution.**

***Index Terms* — FPGA, DSP, Routing Fabric, single precision floating point, transceiver, memory-interface, hardened processor, clocking, security, power management**

I. INTRODUCTION

As FPGAs have moved down the process curve the mainstream vendors have segmented the market into high-end (high-performance and features), mid-range (balanced performance/features against cost and power), and low-cost/power FPGAs targeting higher-volume applications. This paper overviews the architecture and circuit design of an FPGA targeted primarily to the mid-range, but with some features like high-speed transceivers with hardened IP cores and hardened floating-point that are more typical of high-performance devices.

Some of the key target applications for this FPGA include wireless remote radio-head units (RRU) [1], 4xOTU4 Optical Transport Networking (OTN) [2], cloud-server acceleration [3], and mainstream networking. Such applications drive both the design and the architectural focus of the architecture development. For example, RRU requires fixed-point DSP to 491 MHz operating speed, cloud and datacenter desires floating-point DSP for compute-acceleration (e.g. neural-networks), and OTN/networking require significant amounts of configurable on-chip and external memory, as well as high-speed serial I/O. Partial reconfiguration is becoming a mainstream desire in FPGAs in some parts of wireline OTN for muxponders, as well as a driving force for compute acceleration and datacenter and is used regularly for Configuration Memory (CRAM) scrubbing to combat single-event upset (SEU). As functions on FPGAs become more complex, the desire for integrated software processors comes from nearly all end-users and these are also becoming mainstream.

Practical considerations in the design of an FPGA family include the ability to quickly proliferate new devices through composable sub-systems, regularity of design for timing models, and the ability to adjust device parameters and operating conditions for power management. The fabric architecture of an FPGA requires a significant amount of configurability and multi-purpose re-use—for example, embedded memory needs to be configurable to different width/depth combinations, hardened DSP needs to be used with and without adder chains and with variable precision.

Fig. 1 shows a generic floorplan of an Arria 10 device, highlighting columns of I/O and logic resources (amounts vary across the family) and the Hard-Processor block and a die-shot of the A10GX1150, the largest family member. The pictured chip is 5.3B transistors, uses 11 layers of metal on TSMC 20SOC, and has 2 columns each of transceivers and GPIO.

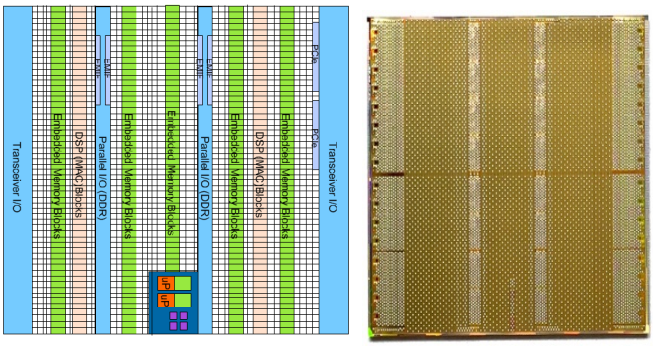


Fig. 1. Floorplan and Die-Photo.

Design challenges are increasing for all chip designers with decreasing feature-sizes: more restrictive design-rules and double patterning, increased process variation, clock-loss and other effects are commonly quoted in research. FPGAs have these issues as well, compounded by the fact that the largest devices approach reticle size, the critical paths for timing are not known at design time (they are determined the user's RTL implemented later by the software which differs for each design implemented on the FPGA), and transceivers and I/O have to implement a wide variety of different I/O standards. At 20nm we addressed these through a number of new techniques that will be detailed throughout in the paper: digitally assisted I/O calibration, increased clock network flexibility and quantity of global clocks, and a more modular floorplan. Some other items of note that are not discussed

here are that we required a significant increase in the use of statistical and many-corner analysis of timing–particularly on analog and memory sense circuits–to ensure robustness, we used a decreasing amount of custom layout over previous FPGA generations–to combat complexity, and performed additional analysis of interconnect and power-grid distribution–to attack reduced scaling of metal relative to the healthier scaling of transistors.

Section II of the paper will describe the logic (including memory and DSP) and routing architecture improvements from 28nm. Section III will describe the I/O subsystems, including the PMA/PCS design, signal integrity issues, digital-assist via embedded micro-controllers and the hardened memory controller. Section IV covers the system-level features and design decisions, including the hard-processor subsystem and security aspects of the device, configuration and SEU mitigation, power management and Smart-VID. Concluding remarks are in Section V. A high-level overview of some of this content was presented at HotChips 2014 [4].

II. FABRIC LOGIC AND ROUTING

This section describes the core FPGA blocks and routing.

A. Logic, Memory and DSP Blocks

The historical FPGA logic-block architecture based on the 4-LUT and FF [5][6] has largely been replaced by the fracturable LUT-complex introduced in [7][8] and illustrated in Fig. 2, denoted an adaptive logic module (ALM). The ALM is based on a 6-LUT (64 bits of LUT-mask) which can either implement a single 6-LUT or can be fractured into two smaller LUTs, with 4 optional FFs following the LUTs. There are 8 data-inputs to the block, and various programmable modes (e.g. register packing) allow these to be swapped and re-used to allow for a variety of uses to consume the 8 inputs. Common usages include: 6-LUT with 2 independent FFs (total of 8 inputs), 2 5-LUTs which share 2 inputs and each drive out through a FF (again 8 total inputs), and arithmetic with LUT-based pre-computation.

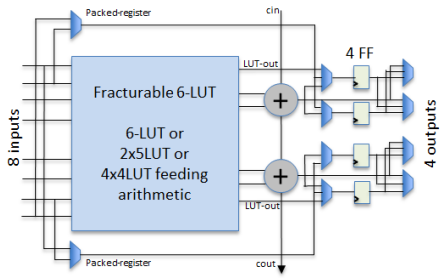


Fig. 2. Fracturable LUT with 4 FFs.

As in Stratix V [9] the FFs are implemented as time-borrowing pulse latches [10][11] rather than as full flip-flops. The logical behavior of a time-borrowing flip-flop (TBFF) is shown in Fig. 3. By programing the delay of the pulse generator, micro re-timing can be generated by the software, allowing for an improved clock speed when the design is implemented onto the FPGA – from 1% to 3% on average.

For Arria 10, TBFFs were added into the M20K ingress and egress registers, allowing for further micro re-timing. M20K blocks are designed to have high-speed performance when used in single-cycle mode (ingress registers with combinational output), but this gives them significant slack in fully registered mode. The TBFF allows the M20K block to transfer slack in a fully registered internal memory path into the egress routing paths.

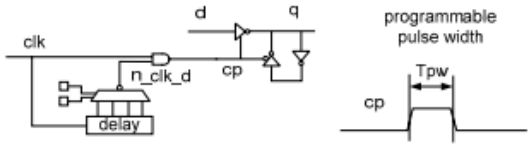


Fig. 3. Extended range time-borrowing FF (TBFF).

The motivation for 4 FFs with only 2 logic signals (ALUTs or sub-LUTs) is that FPGA designers have increasingly added pipelining and delay lines to their RTL to achieve higher clock speeds, resulting in designs having more FFs than LUTs. It is relatively inexpensive to add two additional FFs and, whenever unused inputs (of the 8) are available, this allows un-related FFs to be packed into the ALM using that routing. Fig. 4 shows that, as the FF:LUT ratio for a set of reference designs increases, the relative number of LABs (clusters of 10 ALMs) required to implement the design decreases sharply with some designs seeing a close to 25% reduction in size. Even designs with a FF:LUT ratio less than 1 see a marginal amount of benefit because they tend to have local regions of the design which have improved packing.

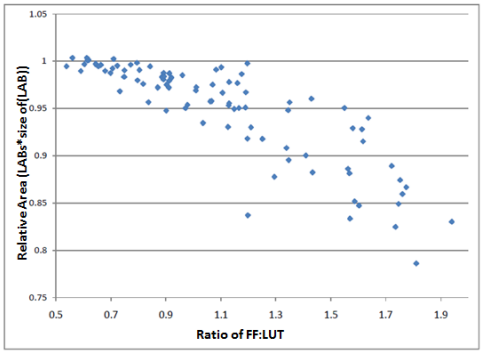


Fig. 4. Packing Improvements with Secondary FFs.

Logic blocks are arranged into clusters called LABs – 10 ALMs per LAB. This architectural hierarchy allows for improved local delays, better software performance (fewer placeable objects), and results in the row/column arrangement shown in Fig. 1. All DSP, M20K and LAB blocks are pitch-matched in y forming LAB-rows. This allows for all fabric to take advantage of logical redundancy (Section IV.C).

The M20K block (Fig. 5) contains 20Kb of data, configurable from x1 to x40 widths and implemented with native dual-port cells. We combine ECC codes targeting adjacent errors with 4b spatial interleaving [12] to ensure the array can detect any error up to 6 adjacent bits, and correct

errors that impact a subset of size 4 bits (For more detail, see [9]).

To achieve best performance/power tradeoff while achieving targeted high capacity memory bit-cell yield, the M20K is supplied by dual supply rails. The SRAM array and wordlines are supplied by a fixed memory supply rail, while the memory periphery circuits are supplied by the core fabric Smart-VID power rail (see Section IV.D). Alternatively, a fixed voltage supply can be used on both rails to reduce on-board regulator BOM cost. Power down mode is supported for unused M20K blocks. R/W assist circuitry ensures robust SRAM cell read write margin, yield and end of life DPM.

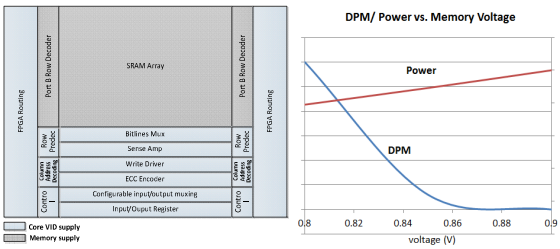


Fig. 5. M20K Block with Dual Supply Rails.

The DSP block has been enhanced to support IEEE754 [13] compliant single precision floating point multiplication and addition, while remaining complete backwards compatibility to legacy fixed point features. The floating point features are underpinned by two innovations: the implementation of the RNE (round to nearest even) rounding mode in the fixed point pipeline with negligible area and performance impact, and the seamless support of a recursive vector mode. For architectural details, see [14].

Fig. 6 illustrates the block in (legacy) fixed point mode, implementing two independent multipliers with optional accumulation chains to the next DSP block shown entering the top and leaving the bottom. Fig. 7 shows the block in floating-point mode using the re-combination described above.

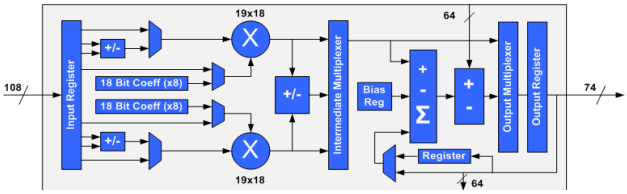


Fig. 6. DSP Block in dual 19x10 Fixed-Point Mode.

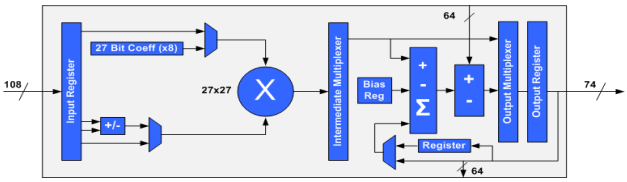


Fig. 7. DSP Block in IEEE754 Block Mode.

As well as the straightforward mapping of FIR filters and dot products to the recursive vector mode structures, newer algorithms for common matrix decompositions such as Cholesky have been introduced [15] which can take advantage of processing entire rows or columns via dot-product structures by reducing or eliminating data dependencies and direct support for these has a strong impact on the implementation.

Fig. 8 illustrates a set of DSP blocks configured in recursive vector mode: Each pair of DSP blocks implements a two-element dot product. For example, the leftmost two blocks calculate AB+CD by routing the CD result into the adder following AB. Each pair of subsequent blocks also calculates the sum of two multiplies, in this case EF+GH, IJ+KL, and so on. The adder in the right-hand block of each pair is then used to sum the tree. The latency of the tree with maximum internal pipelining is $3*(ceil(log_2(size)))+1$.

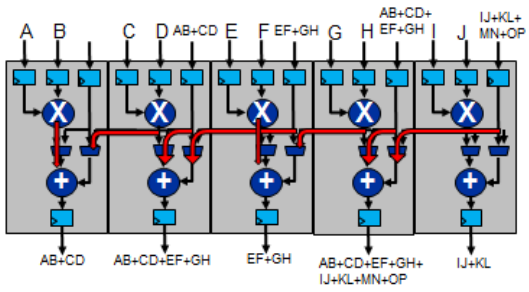


Fig. 8. DSP Recursive Vector Mode.

To illustrate the advantages of this DSP architecture, a typical single precision floating point multiplier/adder pair consumes in the range of 500-700 LUTs as well as a multiplier of 24x24 (or greater precision). The 1632 DSP Blocks on the mid-range 660K LUT A10 device triple the effective size of this device for many DSP and data center applications, as well as greatly reducing the device power consumption.

B. Programmable Routing Fabric

Fig. 9 shows the FPGA routing fabric interface common to all LAB/M20K/DSP blocks. On the input side, signals from global horizontal (H) or vertical (V) wires drive LAB input muxes (LIM) which generate local interconnect wires called LAB lines. LAB lines are sampled by logic-element input muxes (LEIMs) which drive the LUT inputs (or M20K/DSP inputs). Output signals from the block or from other H/V wires passing over the block are selected by a driver input mux (DIM) to drive an H or V wire. Wires are denoted Hn and Vn (e.g. H3,V4) to indicate their direction and logical length in blocks.

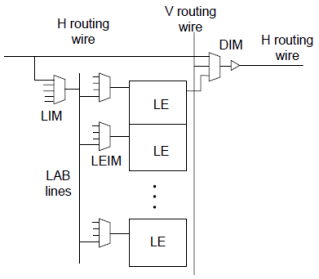


Fig. 9. Routing Fabric: LAB input, H/V Generation.

From Stratix V to Arria 10, the numbers and lengths of routing wires change slightly. Such changes arise from re-tuning the architecture to differences in the scaling of metal and transistor delays in the process shift, re-allocation of wires

across metal layers, and general improvements found during architecture experimentation. Stratix V routing fabric consists of {108xH3, 216xH6, 144xV4} for the primary interconnect and {48xH24, 32xV16} long-lines. With staggered interconnect, 108 H3 wires result from 18 wires driving each left and right from each LAB (or M20K, DSP) and re-starting every 3rd grid-point.

We made some modifications to the primary routing fabric: {72xH3, 234xH6, 208xV4}. To improve long wire utilization these are made bi-directional via tri-stated DIMs (Fig. 10), with one each driven from each LAB ({32xH32, 27xV27} total) replacing the direct-drive wires used in most preceding devices [16]. With this, long-line min-channel width required to implement the same design improves 2.6%, and overall fmax improves modestly (0.72%). Due to the shorter delays of H3/H6/V4 wires in the device, they do not win in either area or delay, and these are left as direct-drive (unidirectional).

Fig. 10. Tristated Long-Lines.

III. TRANSCEIVER, I/O AND CLOCKING

The transceiver and IO sub-systems form columns inside the FPGA. The common theme for both interfaces is the level of flexibility, in configuration and standards support, and integration due to high bandwidth requirements.

A. Transceiver

The transceiver design supports approximately 40 interface standards, ranging in data rate from 28Gbps down to 611Mbps. Standards include CEI-28G, 25G Ethernet, 10G-KR, OTN, SDI, CPRI, Interlaken, PCIe Gen 1/2/3, and SFF-8431. The hardened PCS has 8b/10b and 64b/66b codecs, FEC for 10G-KR, protocol IP for PCIe, Ethernet, and Interlaken (Fig. 11). Of course, IP for all other standards can be realized inside the FPGA core fabric as soft IP. The transceiver sub-system is made up of 6 channel blocks that are aggregated to provide 48 transceiver channels on each side of the chip. The total transceiver count in the largest family member is 96.

To counteract the statistical on-die variation that makes analog design in advance processes difficult, analog circuits are calibrated using firmware that is run on the embedded controller that exists in each column (details in Section III.B).

The transmitter is a voltage mode driver with a total of 5 taps for pre-emphasis equalization. It is well known that voltage mode drivers generate more noise into the power supply than current mode drivers. Furthermore, voltage mode drivers tend to be more sensitive to power noise. One way to solve this problem is to place the voltage mode driver under regulation. However, the challenge in such regulators is the fluctuations in voltage level that depends on the pattern being transmitted that causes jitter. A novel voltage mode driver is used in Arria 10 that introduces a transition into the power supply every bit, compensating the varying transition density in the actual data [17].

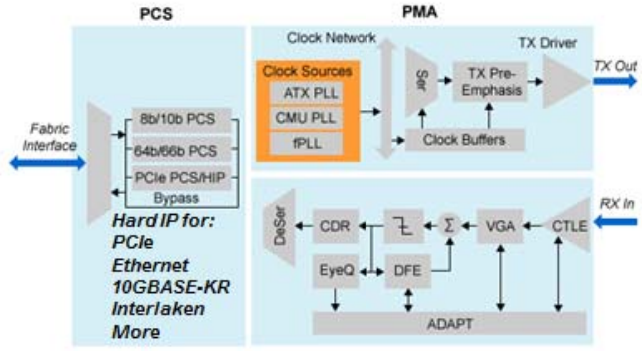


Fig. 11. Transceiver Block Diagram.

Fig. 12 shows a spectral view of the power supply noise when using this driver with jitter compensation capability. By enforcing a transition every bit, the power supply noise is shaped to the very high frequencies. As most power supply networks peak near 100MHz, the dominant power supply noise (> 1GHz) will be filtered out. The effectiveness of the jitter compensating transmitter is shown in Fig. 13 where the jitter is reduced from 25ps down to 10ps at 28Gbps. This measurement is taken with maximum aggressor activity.

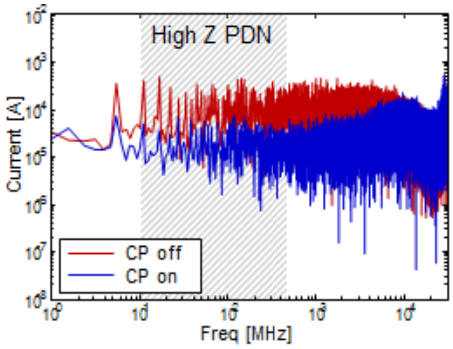
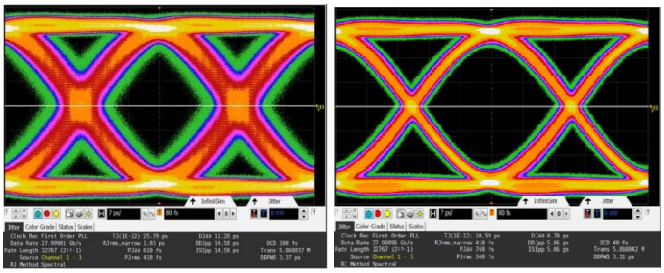


Fig. 12. Power supply noise spectrum with jitter compensation mode on/off.



(a) Compensation off (b) Compensation On

Fig. 13. Transmitter eye diagram with jitter compensation mode off and on, with adjacent transmitters transmitting PRBS31.

Due to the wide range of data rates and protocols supported, the receiver has three different analog front-ends in the RX: a high bandwidth continuous time linear equalizer (CTLE) for data rates between 16 to 28Gbps, a high gain CTLE for backplane applications up to 17.4Gbps, and a low

common mode receiver for low power chip to chip interconnects that are typically DC coupled. The three front ends provide maximum flexibility while allowing each one to be optimized in power and area to the target application.

For very difficult backplane applications where the channel loss at Nyquist can approach 30dB, a fully adaptive Decision Feedback Equalizer (DFE) is designed. The DFE has 7 fixed taps, and 4 taps that can be placed anywhere up to 28UI after the main cursor to address legacy channels with large reflections. Each block of 6 transceivers has two LC based and two ring based PLLs. All four PLLs are fractional synthesizers to allow maximum flexibility to the designer in frequency planning. The fractional synthesizers support aggressive jitter applications such as OTN and SDI (Fig. 14). This high performance fractional capability allows the replacement of expensive Digital or Voltage controlled oscillators (DCXO or VCXO) typically used in these synchronous standards.

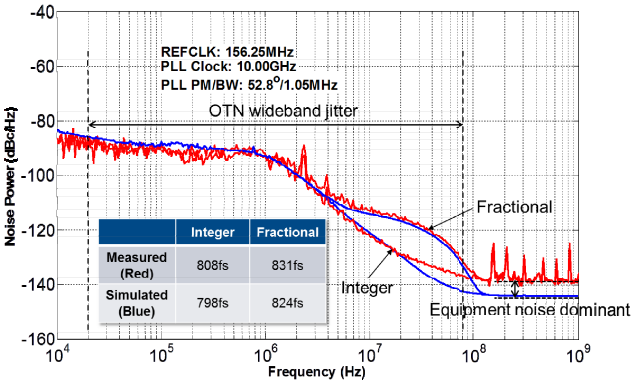


Fig. 14. LC PLL in fractional and integer mode: measured phase noise (red) vs simulation (blue) showing good correlation. Phase noise at high offset is limited by the equipment.

B. Parallel-IO and Hardened Memory Controller

The memory interface on the FPGA provides for flexible standards and bit width configurations. It is split into a series of tiles, each holding 48 IO pins (Fig. 15). A tile may be configured as a control path that drives all of the address and command pins for an interface, or as a data path that drives 32 (36) data pins for DDR (RLDRAM). Users can also bond tiles to form interfaces as wide as 144b. The memory subsystem supports industry-leading performance, including DDR4-2667, DDR3-2133, and LVDS at 1.6GHz.

Each of the tiles has its own hardened memory controller: a pipelined engine employing dynamic command and data reordering to achieve high performance and efficient bus utilization. The hardened controller supports standard timing critical commands such as activate, precharge, read, and write; and sideband commands such as refresh, ZQ calibration, and power-down. Users are free to use the memory controller or to bypass it in favor of a soft solution as needed.

A hard logic block for on-chip termination also provides impedance matching and termination support for memory signal integrity and exists in each IO tile; if a tile employs on-chip termination, it must devote one IO pin to an external 240 ohm reference resistor. The block supports both series and parallel termination, as well as center-tap and pseudo-open-drain interfaces, and users can separately calibrate command/address pins from data pins as required by DDR4. A separate dynamic phase adjuster automatically tracks input transitions to within 1/8th UI, to optimally sample received LVDS data. A phase-locked loop in each tile provides not only a phase-matched clock to the IO tile but also a clock source to the rest of the FPGA fabric.

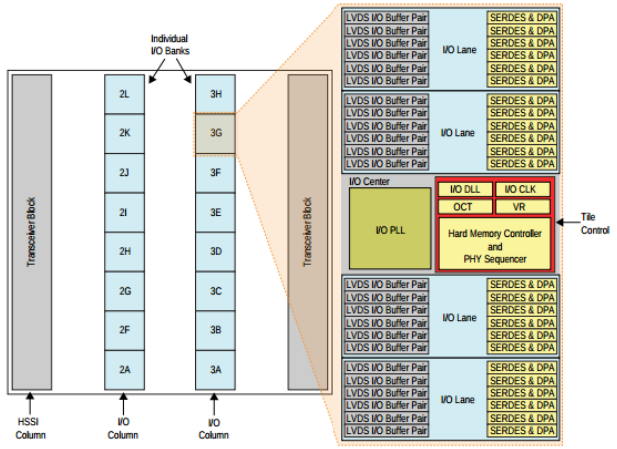


Fig. 15. IO48 Arrangement. Programmable micro-controllers (not shown) sit at the bottom of each column and drive calibration sequences up each column of IO tiles.

C. I/O Calibration

Much as the high-speed transceivers described above can be calibrated using internal sequencing engines, the memory interface also leverages an embedded controller for its own calibration. The controller supports algorithms for DDR2/3/4, QDRII/IV, RLDRAMII/III, and LPDDR2/3, and can calibrate multiple separate interfaces upon system bring-up.

Physically, the embedded controller sits at the bottom of a memory IO column on the FPGA, and it supports all the tiles in that column. It consists of both hardened logic and firmware; the former includes a processor, watchdog timer, and an embedded 56KB program memory, all tied together with an Avalon network [18]. If more compute or memory is required, the Avalon network has a port into the programmable core fabric. At system start, the FPGA control block seeds the embedded memory with bring-up and calibration code and, once the IO PLL locks, it boots the controller. It then uses hooks into the memory interface tiles to control a set of sequencing engines distributed through the memory column. These sequencers are read-write managers that can generate memory commands and memory data, enabling arbitrary patterns used for calibration and training.

Memory routines currently employed include DQS calibration, post-amble tracking, per-pin read de-skew, write leveling and per-pin write de-skew, receiver threshold voltage training, command/address calibration (especially for DDR3 and RLDRAM), and FIFO calibration. A configurable duty cycle corrector samples and adjusts internal and external DQS taps to ensure high signal fidelity. With robust memory calibration and configuration, the FPGA memory interfaces can attain reliable high-frequency performance across a broad

set of industry standards [18]. Fig. 16 shows performance measurements for DDR4.

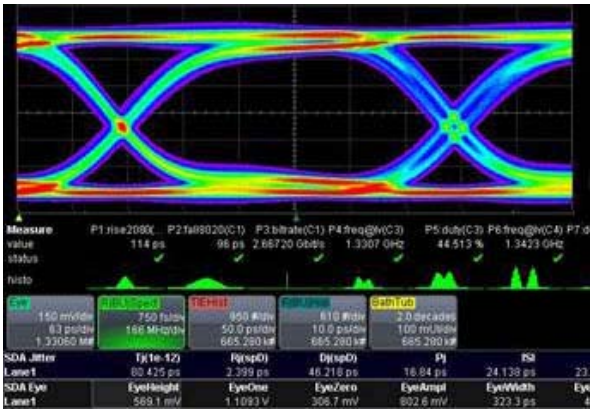


Fig. 16. Measured memory interface performance for DDR4-1333MHz (2666Mbps).

D. Clocking

FPGAs are commonly used for bridging between multi-rate protocols and a large number of flexibly routed clocks are required. The device contains global, regional and periphery clocks and the latter are further divided into SPCLK and LPCLK spines. Global clock distribution is shown in Fig. 17; GCLK can be fed by pins or recovered from data in the HSSI blocks discussed earlier. The clock hierarchy is shown in Fig. 18, resulting in row-clocks that support a fixed row of the device and drive into LAB/DSP/M20K logic. The programmable clock network provides several locations where the clock mux can be gated, either for clock slipping/balancing between domains, functional clock-enables, or for power-savings via activity gating, including high-granularity at the GCLK mux itself and at the per-LAB level. Arria 10 increases the number of GCLK to 32, from 16 in Stratix V.

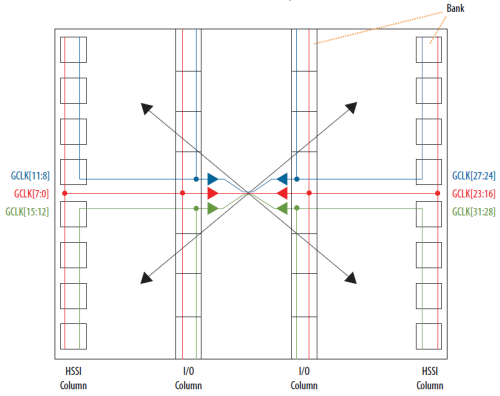


Fig. 17. Clock Generation.

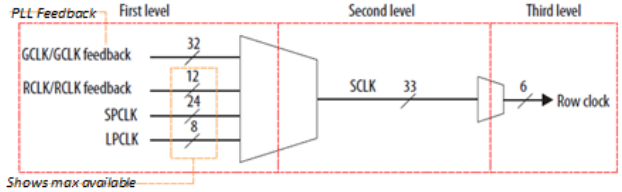


Fig. 18. Clock Generation Hierarchy.

IV. SYSTEM DESIGN

Beyond the traditional FPGA fabric and I/O design topics, Arria 10 contains an embedded processor subsystem, enhanced security features in SOC subsystem, high-speed SEU features in configuration, and new power management techniques, described here.

A. Hard Processor

The Arria 10 family integrates a dual-core ARM A9 [18] subsystem for communications processing, acceleration, offload, embedded processing, and FPGA management. The low-latency control path for FPGA and CPU and non-blocking CPU-DDR path continue from 28nm Arria V SOC™. Notable improvements include increasing the scratch RAM to 256KB, upgrading the DDR interface to DDR4 including a high-quality scheduler and new firewall control. 16b NAND allows for faster file-system access and full DMA and HW ECC allow for maximum throughput (soft-ECC is possible, but inhibits performance). A 3rd Ethernet MAC is added to allow separation of ingress, egress and control, and a direct connection from EMAC to core allows packets to be streamed directly into the core rather than through the NOC.

Fig. 19 shows the main hierarchy of masters and slaves in the SOC. The ARM processors and the FPGA both have low latency controlled paths to shared DDR. A large set of DMA masters have access to the SOC address space via the Multi-layer Network-on-Chip (NOC). This includes Debug Masters as well as USB, EMAC, SDMMC, a general purpose DMA controller as well as the FPGA core itself. The NOC also allows these Masters to write coherently into the L2 Cache for high performance interactions with the ARM processors.

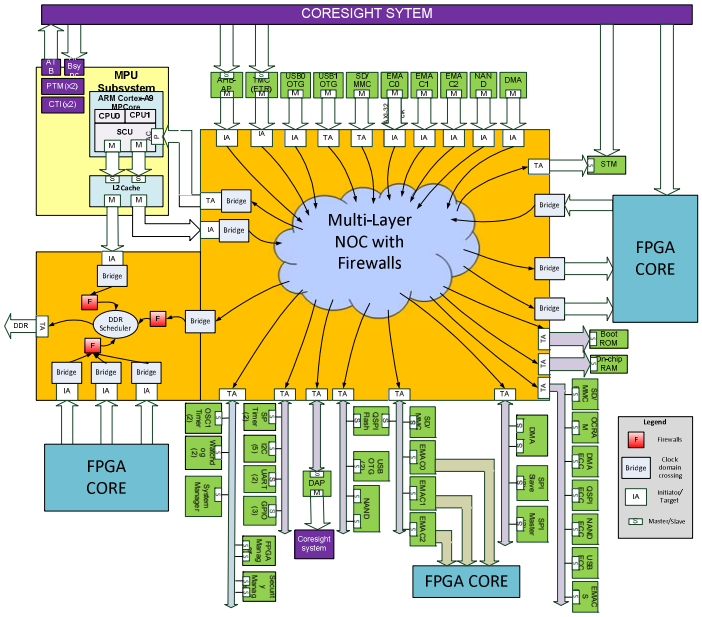


Fig. 19. Hard-Processor Subsystem Interconnect.

B. Security

Security is of key interest in the HPS, and Arria 10 adds secure-boot via a public key hierarchy. Secure-boot is the

ability for the processor to come up through reset in a known state and then validate the authenticity and integrity of the external code prior to executing it. The use of Asymmetric Public/Private key removes the fundamentally difficult issue of maintaining and inserting a secret key onto the device for authentication. Once the HPS has securely booted it can then use public key authentication on the FPGA POF prior to loading it, this can be in addition to or instead of the symmetric key authentication provided by the FPGA control block. This allows a designer to ensure that only code they have authorized is executed on their system, e.g. it prevents anyone re-flashing the system with code containing an attack. Secure-boot implements both root-of-trust and transitive support based on ECDSA public-key authentication. This allows an expected behavior which is capable of describing the current trustworthiness of the associated system. For example, JTAG access is prohibited prior to the establishment of the security state, and then only if permitted by the design settings. All internal test mechanisms are disabled or prevent specified secure optimizations per design settings to prevent attacks via the test interface. Design settings are established in option-bits by the configuration software tool, Quartus® II.

The use of Public/Private keys in a hierarchy allow for more secure and flexible management of keys following security certificate standards such as X.509 Use of these methods allow root private keys to be safely stored off device and for flexibility in signing layers and code update. For example a 3-level chain consisting of a root key signed by KAK/CSK/external: the root key is used to sign a key-authorization key allowing the root secret key to be stored in a vault. The KAK is used to sign a code-signing key stored in locked fuses. Finally, the CSK is used to sign the external code to be loaded.

The software secure hierarchy supports both 3rd party and industry-standard certification. Boot-to-OS has multiple stages in which each stage authenticates the next using either same or new public keys. And an application layer authenticated by the OS is used to authenticate the FPGA bitstream (POF) before it can be loaded on the FPGA.

The HPS system fully supports ARM Trustzone [20] and Privilege Filtering with multiple layers of firewalls within the main NOC switch and DDR switch. These firewalls allow highly controlled and partitioned access from each system master to sub regions of Peripherals, Onchip RAM, FPGA and DDR. An example use model is to partition DDR to control access so that masters that share common physical DDR cannot corrupt each other's memory space.

C. Configuration and Configuration Features

Arria 10 continues to support legacy features such as bitstream compression and authentication. Hardware redundancy is implemented by adding several architecturally redundant rows to the device that are programmably disabled by eFUSE at test time. As in memory devices, this provides a significant improvement in product yield.

Arria 10 supports partial re-configuration, in which column-frames of the device can be re-configured while the device is in operation. This enables both design-use models such as OTN muxponder with hot-swap protocols, virtualization of compute acceleration functions and also allows for internal scrubbing of the FPGA for soft-errors in the configuration RAM (CRAM) when an error is located. Though soft-errors remain a rare occurrence, secure systems require SEU mitigation strategies.

In 28nm, error-detect CRC (EDCRC) continually performed a frame-by-frame CRC on the bitstream, and corrected errors when one was found. We parallelized the computation of the CRC across 32b frame-columns, as shown in Fig. 20, and improved the detection time from 68ms (largest Stratix V device at 952K LE) to 22ms (largest Arria 10 device at 1.1M LE). The error-message-register (EMR) functions largely as in Stratix V, tying in to software hierarchy tagging to allow designers of high-reliability systems to make mitigation decisions based on the location of an error within the RTL design hierarchy. To address of multi-bit upsets, the 32b frames are interleaved to guarantee that any MBU to physically-adjacent bits is fully correctable.

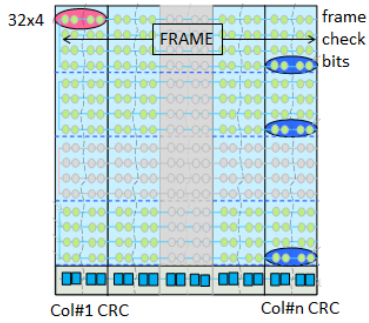


Fig. 20. SEU Error-Detection.

D. Power Management

Arria 10 FPGA retains Programmable Power Technology (PPT) originally introduced in Stratix III [18]. In PPT, logic and routing circuits in non-timing-critical paths are set to low power mode through back biasing [23]. Since a small percentage of logic and routing circuits in a typical design are timing critical, PPT is an effective way to reduce power without sacrificing performance. In addition, PPT puts unused memory, DSP blocks, and LAB into low-power mode. Quartus II software analyzes the available slack in a design and controls the transistors and logic blocks to switch between high-performance and low-power modes and it is transparent to the user. In Arria 10, PPT enables up to 20% full chip static power reduction.

Arria 10 introduces Smart Voltage ID (VID) which enables the FPGA to run at lower than nominal core voltage, V_{CC}, and periphery voltage, V_{CCP}, while retaining the same performance level and reducing static and dynamic power [24]. Fig. 21 illustrates operational aspects of VID and resulting savings in static power. The minimum voltage level required by Arria 10 devices is programmed into a fuse block during manufacturing. During operational mode, an external

host or power system controller reads these values and sets appropriate voltage levels.

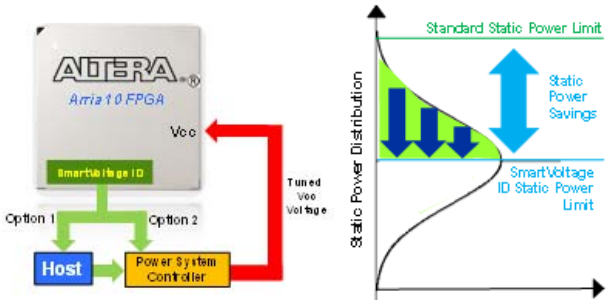


Fig. 21. Smart Voltage-ID.

The V_{CC} Power Manager feature allows the device to run at the lowest V_{CC} and V_{CCP} voltage levels by trading off performance for power savings. Both Smart VID and V_{CC} Power Manager use VID fuse data to set appropriate voltage levels. In Arria 10, Smart VID features enable up to 40% reduction of full chip static power. Each device has its own 5-bit VID. Altera supports Power Management BUS (PMBUS^{TM}) and a 5-bit parallel interface for communication between FPGA and power system controller.

V. CONCLUSION

This paper has presented a full description of the Arria 10 architecture including transceiver design and performance, GPIO design and performance, fabric and logic architecture, clocking and the integrated processor subsystem. The highlights of this family include the innovations around the configurable I/O such as digital-assist for the analog calibration, jitter and noise compensation. New blocks in the device include hardened memory controllers and floating-point enabled DSP blocks. The HPS system has been updated with an emphasis on security features. New techniques for power management (e.g. VID) represent a trend to greater integration between the workings of the on-chip PDN and power regulators.

VI. ACKNOWLEDGEMENTS

The authors would like to acknowledge some of the many additional people who contributed to the Arria 10 device fabric, I/O, HSSI and SOC architecture; and to preparation of this paper: Richard Cliff, Ron Ho, David Cashman, Tim Vanderhoek, Jeff Chromczak, Ninh Ngo, Andy Lee, Alan Chai, Hui Fang Tan, Hee Kong Phoon, and Han Wooi Lim.

VII. REFERENCES

[1] Z. Zhang, F. Heiser, J. Lerzer and H. Leuschner, "Advanced Baseband Technology in Third-Generation Radio Base Stations," Ericsson White Paper (link). 2003.
[2] Altera, "Optical Transport Networks for 100G Implementation in FPGAs," White Paper WP-01115-1.1 (link)
[3] A. Putnam et al., "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services," *41st Int'l Symp. On Computer Architecture (ISCA)*, 2014.
[4] B. Vest, S. Atsatt and M. Hutton, "Design of a High-Density SoC FPGA at 20nm," HotChips, 2014.
[5] V. Betz, J. Rose and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*, Kluwer 1999.
[6] D. Lewis et al., "The Stratix™ Routing and Logic Architecture," *Proc. Int'l Symp. On FPGAs (FPGA)*, pp. 12.20, 2003.
[7] M. Hutton et al., "Improving FPGA Performance and Area Using an Adaptive Logic Module," *Proc. Field-Programmable Logic 2004*, LNCS 3203, 2004.
[8] D. Lewis et al., "The Stratix™ II Routing and Logic Architecture," *Proc. Int'l Symp. On FPGAs (FPGA)*, 2005.
[9] D. Lewis et al., "Architectural Improvements in Stratix™ V," *Proc. Int'l Symp. On FPGAs (FPGA)*, pp. 147-156, 2013.
[10] N. Nedovic et al., "A Clock Skew Absorbing Flip-Flop," *Proc. ISSCC*, pp. 342-343, 2013.
[11] B. Teng and J. Anderson, "Latch-Based Performance Optimization for FPGAs," *Proc. ISSCC 2003.* pp. 342-343, 2003.
[12] R. Goodman and M. Sayano, "The Reliability of Semiconductor RAM Memories with On-Chip Error-Correction Coding," *IEEE Trans. Information Theory*, May, 1991, pp. 884-896.
[13] IEEE Standard for Floating-Point Arithmetic, IEEE Std. 754-2008, pp. 1-58, 2008.
[14] M. Langhammer and B. Pasca, "Floating-Point DSP Block Architecture for FPGAs," *Proc. Int'l Symp. On FPGAs (FPGA)*, pp. 117-125, 2015.
[15] S. Derirsoy and M. Langhammer, "Cholesky Decomposition using Fused Datapath Synthesis," *Proc. Int'l Symp. On FPGAs (FPGA)*, pp. 241-244, 2009.
[16] V. Betz and J. Rose, "FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density," *Proc. Int'l Symp. On FPGAs (FPGA)*, pp. 59-65, 1999.
[17] K. Oh, Y. Shim, T. Hoang, W. Ding, and S. Chandra, "Systems and Methods for Reducing Power Supply Noise or Jitter," *US Patent 8,836,384*, Sep. 16, 2014.
[18] Altera, "Arria 10 Core Fabric and General Purpose IOs Handbook," link, 2015
[19] Arm, Cortex A9 Processor, arm.com (link).
[20] Arm TrustZone Technology, arm.com (link).
[21] Altera, "Avalon Interface Specification," White Paper (link), 2015.
[22] D. Lewis et al., "Architectural Enhancements in Stratix-III and Stratix IV," *Proc. Int'l Symp. On FPGAs (FPGA)*, pp. 33-42, 2009.
[23] J. Tschanz et al., "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage," *IEEE J. Solid State Circuits*, pp. 1396-1402, vol 37, no 11, Nov 2002.
[24] Altera, "Power Reduction Features in Arria 10 Devices," Application Note (link), 2015