Comparative Study on Low Power FPGA Solutions

A Lattice Semiconductor White Paper.

April 2024

Learn more:

www.latticesemi.com

Contact us online:

www.latticesemi.com/contact
www.latticesemi.com/buy

TABLE OF CONTENTS

Abstract

In this comprehensive analysis, we compare a single RTL (Register Transfer Level) design implemented on three distinct FPGA (Field Programmable Gate Array) devices from different brands, each selected for its unique strengths in high-performance computing, power efficiency, and extensive I/O capabilities. The study focuses on evaluating key performance metrics, resource utilization, power consumption, and adaptability to application requirements. Significant variations were observed in the devices' performance, power efficiency, and resource demands, highlighting the importance of aligning FPGA selection with specific design and application needs. The findings emphasize the critical role of power management in FPGA design, particularly concerning static and dynamic power consumption, as well as the implications for physical space requirements and bill of materials costs. This analysis serves as a guide for optimizing FPGA selection and design strategies in a rapidly evolving technological landscape.

Introduction

This paper delves into a comparative analysis of a single RTL (Register Transfer Level) design implemented across three distinct FPGA (Field Programmable Gate Array) devices from three different brands. The objective is to understand how the same design performs when mapped to FPGAs from different manufacturers, each with its unique architecture and features. This comparison will provide insights into the nuances of FPGA selection and optimization strategies for a given design

Selection of FPGA Devices

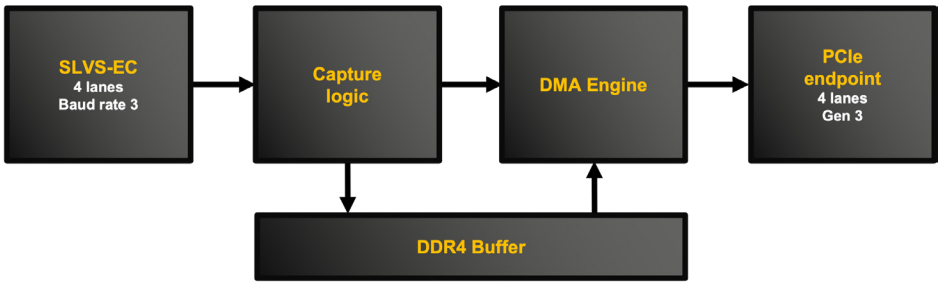
For this study, we select three FPGAs, each from a distinct brand:

- [Solution A] AMD Artix™ Ultrascale+™ AU15P
- [Solution B] Intel® Arria®10 GX160
- [CPNX] Lattice CertusPro™-NX LFCPNX-100

	[Solution A] AMD Xilinx Artix Ultrascale+ AU15P	[Solution B] Intel PSG Arria10 GX160	[CPNX] Lattice CertusPro-NX LFCPNX-100
Technology	16 nm FinFET	20 nm	28 nm FD-SOI
Logic density (LUTs)	170,000 SLCs 156,000 CLB FF 78,000 CLB LUTs	160,000 LE 61,510 ALM ~ 150,000 SLCs equiv.	96,000 LCs 79,900 LUTs ~ 130,000 SLCs equiv.
Memory	7.6 Mb block RAM	10 Mb block RAM	7.328 Mb (EBR+LRAM)
DSP	576	156	156
Transceivers	12 SERDES (GTY)	12 SERDES	8 SERDES

RTL Design Description

The RTL design chosen for this study is a multi-purpose digital signal processor (DSP) system, designed to be sufficiently complex to engage the unique features of each FPGA. This design involves elements like parallel processing cores, memory interfaces, and I/O operations, providing a comprehensive basis for comparison. Key protocols for an image sensor input is SLVS-EC and PCIe for the host interface.



SLVS-EC (Scalable Low-Voltage Signaling with Embedded Clock) is a high-speed interface standard primarily used in the field of image sensors and camera systems. It's an evolution of the traditional SLVS (Scalable Low-Voltage Signaling) standard, designed to support higher data rates and improved efficiency, which is essential in advanced imaging applications. Here are some key aspects of SLVS-EC:

1. **Embedded Clock:** Unlike many high-speed interfaces that use separate clock and data lines, SLVS-EC embeds the clock signal within the data stream. This approach simplifies the interface and reduces the number of required physical connections, making it more efficient in terms of space and design.
2. **High Data Rates:** SLVS-EC supports very high data transfer rates, which can be several Gbps (Gigabits per second) per lane. This makes it suitable for applications that require the rapid transfer of large amounts of data, such as high-resolution or high-frame-rate video imaging.
3. **Scalability:** The standard is scalable in terms of bandwidth, allowing for multiple lanes to be used. This scalability makes it adaptable to various applications, where the required data rate might differ significantly
4. **Low-Voltage Signaling:** As the name suggests, SLVS-EC operates at low voltages. This characteristic leads to reduced power consumption, which is particularly beneficial in battery-powered devices like cameras and mobile imaging systems.
5. **Applications:** The primary application of SLVS-EC is in the field of imaging, particularly in high-performance cameras used in industrial, medical, and scientific applications, as well as in consumer electronics. It's especially useful in systems where high-quality images need to be captured and transferred rapidly, like in high-speed video cameras or advanced driver-assistance systems (ADAS) in automotive applications.
6. **Compatibility and Adoption:** As a more advanced version of the SLVS standard, SLVS-EC is gradually gaining adoption in the industry, especially in areas where its high data rate capabilities and efficient signaling are critical.

Overall, SLVS-EC represents a significant advancement in high-speed digital interfaces, offering benefits in terms of data rate, efficiency, and scalability, which are crucial for modern high-resolution imaging systems. The three selected FPGA devices support SLVS-EC in its latest public revision (2.0) via dedicated SERDES. The chosen configuration is made up by 4 lanes, set at Baud Rate 3 (4.9 Gbps per lane).

The host interconnection chosen for the test is **PCI Express®**. **Implementing a PCIe® (Peripheral Component Interconnect Express)** endpoint on FPGAs is a common approach in modern digital design, especially for applications requiring high-speed data transfers between the FPGA and other components, such as CPUs or other peripheral devices. Here are key aspects of PCIe endpoint implementation in FPGAs:

1. **PCIe in FPGAs:** PCIe is a high-speed serial computer expansion bus standard. When implemented in an FPGA, it allows the FPGA to communicate with the host system (like a PC or server) at high data rates, making it ideal for applications that require rapid data processing and transfer.

2. **Endpoint Configuration:** In a PCIe system, the endpoint refers to the device that initiates or terminates a data transfer. When an FPGA is configured as a PCIe endpoint, it acts as a peripheral device that either sends data to or receives data from the PCIe host controller.
3. **IP Cores and Soft Cores:** Most FPGA vendors offer specialized Intellectual Property (IP) cores for PCIe, which can be integrated into the FPGA design. These cores include the necessary logic for handling PCIe protocol layers (physical, data link, and transaction layers). Some FPGAs also provide the flexibility to implement PCIe through soft cores, where the PCIe logic is synthesized into the FPGA fabric.
4. **Design Considerations:** Implementing PCIe on an FPGA requires careful consideration of several factors, including the choice of the right PCIe generation (e.g., PCIe Gen 1, Gen 2, Gen 3, etc.), lane configurations (X1, X4, X8, X16), and the overall architecture of the system. The design must ensure compliance with PCIe standards for signaling, data integrity, and error handling.
5. **Application:** PCIe-enabled FPGAs are widely used in a variety of applications, including data center operations, high-performance computing, networking, storage devices, and real-time signal processing. The high bandwidth and low latency of PCIe make it suitable for these demanding applications.
6. **Advantages:** The primary advantage of using PCIe in FPGAs is the ability to achieve high-speed data transfers with the host system, leveraging the FPGA's parallel processing capabilities. This setup is highly efficient for data-intensive tasks.
7. **Challenges:** Implementing PCIe in FPGAs can be challenging, particularly in terms of ensuring compliance with the PCIe standard and optimizing the design for performance and resource utilization. It requires a good understanding of both the PCIe protocol and FPGA architecture.
8. **Integration with Other Technologies:** Often, PCIe-enabled FPGAs are used in conjunction with other technologies, such as DDR memory or high-speed networking interfaces, to build complex, high-performance systems.

In summary, PCIe endpoint implementation on FPGAs is a powerful approach for creating high-speed, data-intensive applications. It combines the flexibility and parallel processing capabilities of FPGAs with the high bandwidth and industry standard interface of PCIe, enabling efficient and effective solutions in various technological fields.

Frame buffering in the design is guaranteed by DDR4 memory banks, interconnected via wide bus after NoC (256 bits @ 100 MHz).

Logic Design Comparison Methodology

Implementation Process

Each FPGA implementation follows the standard design flow:

- **RTL Coding:** The same RTL code is used for all three FPGAs.
- **Synthesis:** Using each brand's proprietary synthesis tools.
- **Place and Route:** Tailoring the design to each FPGA's specific architecture.
- **Timing Closure:** Ensuring the design meets the timing requirements.

Performance Metrics

The following metrics are used for comparison:

- **Clock Frequency:** Maximum achievable clock speed.
- **Throughput and Latency:** Measuring data processing capabilities.

- **Resource Utilization:** Logic utilization, memory blocks, and I/O ports used.
- **Power Consumption:** Analyzed under identical operational conditions.

Testing and Validation

- **Functional Simulation:** To confirm that the RTL design functions correctly on all FPGAs.
- **Timing Analysis:** To ensure reliability under different operating conditions.
- **Power Analysis:** Conducted using each brand's power analysis tools.

Design Implementation

Each FPGA component is used to implement the same design, ensuring a fair comparison. The design should be complex enough to leverage the unique features of each component.

Testing and Data Collection

- **Synthesis and Place & Route:** Analyze the results of synthesis and place & route for insights into resource utilization and potential performance bottlenecks.
- **Simulation and Timing Analysis:** Perform rigorous simulation to validate functionality and conduct timing analysis for performance metrics.
- **Power Analysis:** Use FPGA vendor tools for detailed power analysis under varied load conditions.
- **Real-world Application Testing:** Deploy each design in a controlled environment that mimics a real-world application, observing performance and adaptability.

Comparative Analysis

Solution A FPGA
- **Performance:** Expected to excel in high-speed data processing
- **Resource Utilization:** Efficient utilization of advanced logic blocks, but no dedicated memory controller or PCIe EP
- **Power Consumption:** Potentially high due to high-performance orientation, mitigated by small technology node (16 nm)

Solution B FPGA
- **Performance:** On par with Solution A, with some peculiar advantages in terms of hard macros
- **Resource Utilization:** Taking advantage of wider ALUT architecture, it generally results in more dense designs
- **Power Consumption:** Very high, proven to be the most power hungry in the test on I/Os and high speed transceivers

CPNX FPGA
- **Performance:** Adequate for most applications, with emphasis on power efficiency.
- **Resource Utilization:** Smaller IPs drive to overall smaller designs
- **Power Consumption:** Low power tech node helps in keeping energy and thermal consumptions at best grades in the round.

Here below an IP resource tabular report for Solution A/B and CPNX, extracted from their respective final P&R designs in the given configuration (4X SLVS-EC lanes Baud 3, PCIe Gen 3.0 4X EP):

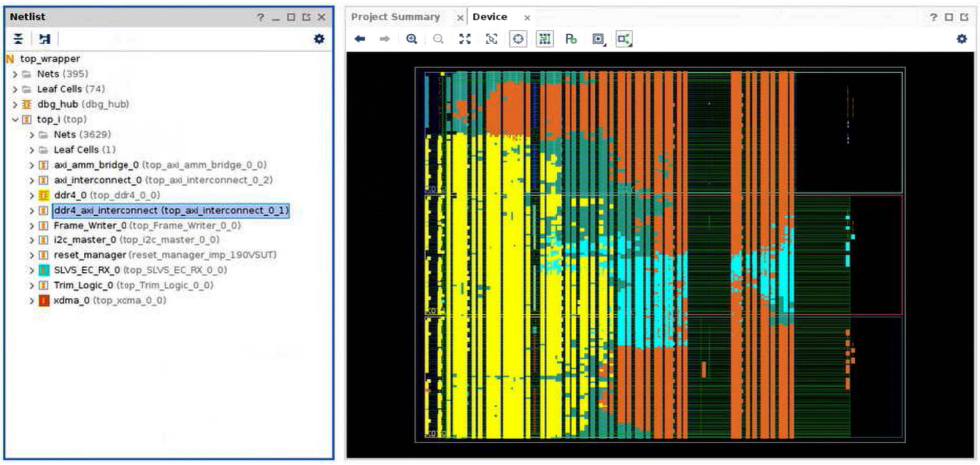
RTL Module/ Hard Macro	(A)LUTs		
	[Solution A] AMD AU15P	[Solution B] Intel 10AX16	[CPNX] Lattice LFCPNX-100
DDR4 controller	15,409	35	2,844
PCIe EP with DMA	21,600	11,110	16,008
SLVS-EC	4,652	4,607	2,708
Frame Writer	2,091	669	322
Trim Logic	219	286	340
I2C Master	254	330	216
Reset Controller	58	0	41
Interconnect Logic	12,349	5,556	1,179
RTL Module/ Hard Macro	**Dedicated Logic Registers**		
	[Solution A] AMD AU15P	[Solution B] Intel 10AX16	[CPNX] Lattice LFCPNX-100
DDR4 controller	21,192	146	4,538
PCIe EP with DMA	22,965	23,822	11,627
SLVS-EC	3,561	4,093	3,980
Frame Writer	3,172	772	590
Trim Logic	295	422	713
I2C Master	266	255	129
Reset Controller	132	3	48
Interconnect Logic	26,909	28,917	2,218
RTL Module/ Hard Macro	**Block RAM**		
	[Solution A] AMD AU15	[Solution B] Intel 10AX16	[CPNX] Lattice LFCPNX-100
DDR4 controller	0	0	0
PCIe EP with DMA	4	4	4
SLVS-EC	4	4	4
Frame Writer	0	0	0
Trim Logic	0	0	0
I^2C Master	0	0	0
Reset Controller	0	0	0
Interconect Logic	0	0	0

Notes:

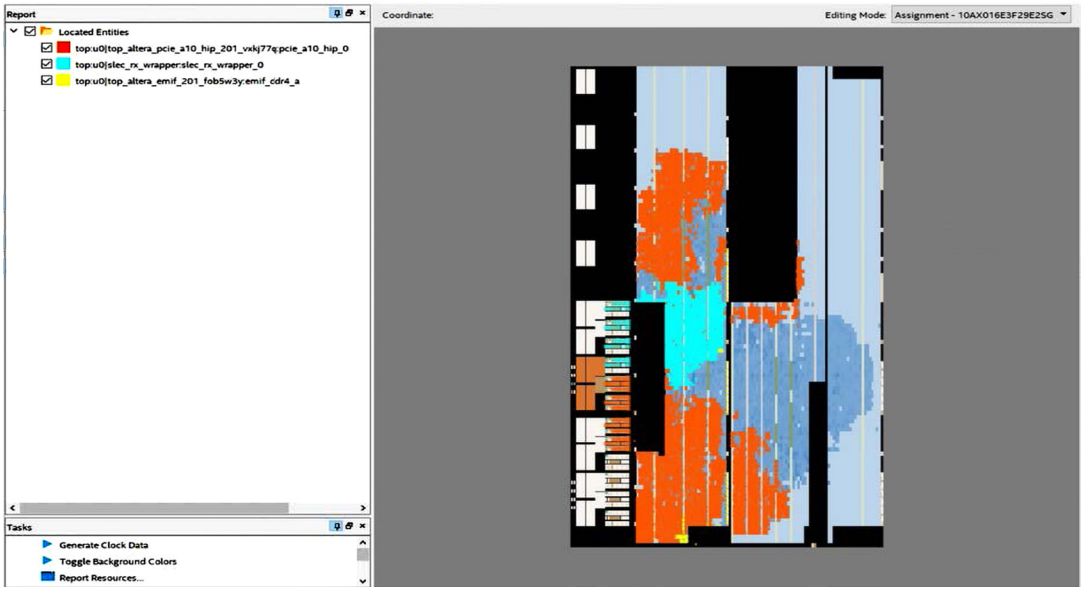
- Arria10 ALUT has 8 inputs, Artix Ultrascale+ LUT has 6 inputs
- Arria10 has hard DDR4 controller (including hard Nios for calibration)

- AMD BRAM count is scaled by a factor of 20/36 considering that BRAM size is 36 Kbit while Arria10 M20K is 20 Kbit
- Lattice CPNX mapping tool can remap Block RAMs in a mixture of EBR+Distributed RAMs when manipulating inferred RAMs from RTL. In some selected macros, like dual-clock FIFOs and line buffers, the tool has been constrained to use EBR primitives only.

The Solution A floorplan shows a significant area taken by the DDR4 controller, including its own buffers, and a relatively other large portion taken by the DMA engine:



The Solution B floorplan is similar to the one above, with the most notable difference about the memory controller, being a physical macro and thus not taking any portion of the programmable logic area:



Power Efficiency Comparison Methodology

In the realm of FPGA-based design, power consumption is a critical factor, impacting not only the operational efficiency but also the physical space requirements and cost of the bill of materials (BOM). This chapter provides a detailed comparison of the power consumption aspects of the same RTL design implemented on three distinct FPGAs from different brands.

Semiconductor devices, at their core, are susceptible to temperature-induced performance degradation and failure, necessitating effective cooling strategies to maintain optimal operation. When operating in open air without additional cooling, these devices rely solely on ambient air for heat dissipation, suitable only for low-power applications where heat generation is minimal. However, as power density and performance demands increase, passive cooling with heatsinks becomes essential. Heatsinks enhance heat dissipation through increased surface area and conductive materials, efficiently pulling heat away from the semiconductor device. For even higher power applications, active cooling systems, combining heatsinks with fans or liquid cooling solutions are required. These systems actively circulate air or liquid to more rapidly disperse heat, crucial for high-performance semiconductors where excessive heat can lead to thermal throttling, reduced lifespan, or catastrophic failure. The choice between these cooling methods hinges on the balance between the semiconductor's power consumption, heat output, and the thermal management capabilities of the device's environment.

Power Consumption Metrics

The power analysis encompasses several dimensions:

1. **Static Power:** Primarily from leakage currents in the transistors.
2. **Dynamic Power:** Due to the switching activity in the logic and interconnects.
3. **I/O Power:** Consumption related to input/output operations.
4. **SERDES Power:** Consumption related to high-speed input/output operations.

Each of these components contributes to the total power consumption and has implications for heat dissipation, cooling requirements, and overall system reliability.

In the domain of heat spreading, particularly for electronic systems, several key units of measure are used to quantify heat and its management:

1. **Watt (W):** The primary unit of power, representing the rate of energy transfer. In the context of electronics, it quantifies the amount of heat generated by a device.
2. **Degrees Celsius (°C):** The unit of temperature used to express the operational temperature of the device or the ambient temperature.
3. **Thermal Conductivity (W/m·K):** This measures a material's ability to conduct heat, crucial for understanding how effectively a heatsink can dissipate heat away from the electronic component.
4. **Thermal Resistance (°C/W):** Indicates the resistance to heat flow, which is critical for understanding how well a cooling solution can maintain a lower temperature of the electronic component compared to its surroundings.

As for the limits that dictate the need for passive or active cooling:

- **Passive Cooling:** Typically, electronic components that operate at temperatures up to about 70 °C can often be sufficiently cooled using passive methods, such as heatsinks without fans. These methods are more common in low to moderate power applications.

- **Active Cooling:** When components exceed approximately 70 °C, active cooling methods, like heatsinks with fans or liquid cooling systems, are usually required. These systems are essential for high-power applications or in environments with limited airflow, where passive cooling is insufficient to manage the heat generated.

It's important to note that these temperature thresholds are approximate and can vary depending on the specific electronic component, its use case, and the manufacturer's specifications. Moreover, long-term reliability, performance stability, and safety standards also influence the choice of cooling strategy, requiring a detailed thermal analysis for optimal heat management in electronic systems.

Apart from SERDES power, which requires a separate discussion, power consumption is typically categorized into two main types: static power and dynamic power. Understanding these two types of power consumption is crucial for efficient FPGA design and deployment, especially in applications sensitive to energy usage or heat generation.

1. **Static Power**
   - **Definition:** Static power, also known as leakage power, is the power consumed by an FPGA when it is powered on but not actively switching. It is a result of leakage currents that flow in the transistors, even when they are not actively switching.
   - **Causes:** The primary cause of static power consumption is the sub-threshold leakage in the transistors. As semiconductor technology scales down to smaller geometries (like in modern FPGAs), leakage currents increase significantly. Other sources of static power include gate leakage and junction leakage.
   - **Characteristics:** Static power consumption is relatively constant and does not vary significantly with the clock frequency or the activity of the FPGA. However, it is influenced by factors such as the manufacturing process, temperature, and the voltage supply level.

2. **Dynamic Power**
   - **Definition:** Dynamic power, on the other hand, is the power consumed when the FPGA is actively switching. This includes the power used to charge and discharge capacitors each time a signal transitions from low to high or high to low.
   - **Causes:** Dynamic power consumption is primarily caused by two factors: capacitive switching and short-circuit currents. Capacitive switching power, the more dominant of the two, occurs when transistors switch states, charging and discharging internal node capacitances. Short-circuit power occurs during the brief period whenw both the NMOS and PMOS transistors in a logic gate are partially on, allowing current to flow directly from Vdd to GND.
   - **Characteristics:** Dynamic power is highly dependent on the clock frequency, the switching activity (how often signals change states), and the voltage supply level. The relationship is given by the formula $P_{dynamic} = \alpha C V^2 f$, where α is the activity factor, C is the capacitive load, V is the supply voltage, and f is the clock frequency.

In FPGA design, both static and dynamic power consumptions are critical considerations. While dynamic power dominates at higher frequencies and heavy switching activities, static power becomes more significant in low-power applications or in scenarios where the device spends much time in idle states. Effective power management in FPGAs involves optimizing both types of power consumption to meet the requirements of the application, considering aspects like performance, heat dissipation, and energy efficiency. I/O Power is normally negligible when compared to the former two measurements, and will be ignored in this essay.

SERDES Functionalities and Power

SERDES (Serializer/Deserializer) power consumption in an FPGA device is a critical aspect, especially in applications that utilize high-speed data transmission. SERDES blocks in FPGAs are used to enable high-speed serial communication over differential signals, such as those found in standards like PCIe, USB, Ethernet, and many others. The power consumption of SERDES in an FPGA is influenced by several factors:

1. **Operating Speed:** Higher data rates typically lead to higher power consumption. The SERDES blocks need to operate at higher frequencies to serialize and deserialize data streams, which increases both dynamic and static power consumption.
2. **Signal Integrity Requirements:** Maintaining high signal integrity over longer distances or through challenging electromagnetic environments can require additional power. This might involve using more robust pre-emphasis and equalization techniques, which can increase power usage.
3. **Number of Active Lanes:** The power consumption scales with the number of active SERDES lanes. More active lanes mean more parallel serialization/deserialization processes, leading to higher overall power usage.
4. **Voltage and Process Technology:** The supply voltage and the semiconductor process technology used in the FPGA can significantly impact the power efficiency of SERDES blocks. Lower voltage operations and advanced process technologies typically reduce power consumption but may come with trade-offs in performance or cost.
5. **Activity Level:** Similar to other components in an FPGA, the power consumption of SERDES blocks is also dependent on their activity level. A constantly active SERDES block, transmitting or receiving data, consumes more power compared to a SERDES block that is idle or intermittently active.
6. **Cooling and Environmental Conditions:** External factors like temperature and the effectiveness of the cooling solution can also affect the power efficiency of SERDES blocks. Higher ambient temperatures can lead to increased leakage currents, thereby increasing static power consumption.

In summary, SERDES power consumption in FPGAs is a complex interplay of design choices, operating conditions, and environmental factors. Efficient management of SERDES power consumption is crucial, particularly in high-performance computing, telecommunications, and data center applications, where large numbers of high-speed interfaces are common. FPGA designers often need to balance the trade-offs between data rate, signal integrity, lane count, and power efficiency to optimize the overall performance and power profile of their systems.

Physical Space Requirement for Power Section

1. **Heat Dissipation and Cooling Solutions:** More power consumption necessitates larger or more efficient cooling solutions, which can increase the physical space requirement for the power section.
2. **Power Delivery Network (PDN):** The complexity and size of the PDN depend on the power requirements of the FPGA. A higher power draw might need a more robust PDN, which occupies additional space on the PCB.

Cost of Bill of Materials

1. **Cooling Components:** Higher power FPGAs require more effective cooling solutions like heat sinks, fans, or even liquid cooling systems, which add to the BOM cost.
2. **Power Regulation and Distribution:** Advanced power regulators and capacitors to handle higher currents and ensure stable power delivery contribute to the BOM.
3. **Power Efficiency Optimization Components:** Additional components such as DC-DC converters and LDOs (Low Dropout Regulators) for power efficiency optimization.

Comparative Analysis

The power analysis has been processed at various depth levels. Most important is the FPGA device power consumption, but immediately afterwards comes the discussion around the required peripherals needed by each silicon device, expressed in terms of power, thermal and cost effects.

Standing on the results coming from the design tools and verified by in-vivo measurements where possible, here below is reported a block-level power essay on the DUTs:

RTL Module/ Hard Macro	Power (mW) w/o Static	
	[Solution A] AMD AU15P	[Solution B] Intel 10AX16
DDR4 controller	977.00	1,853.11
PCIe with DMA(1)	1,677.00	2,272.31
SLVS-EC(1)	503.00	1,030.95
Frame Writer	34.00	21.65
Trim Logic	2.00	5.25
I2C Master	2.00	45.56
Reset Controller	1.00	0.03
Interconnect Logic	271.00	291.74
Total dynamic power (mW)	**3,467.00**	**5,520.59**
Static power (mW)	299.00	1,235.00
Total power (mW)	**3,766.00**	**6,755.59**

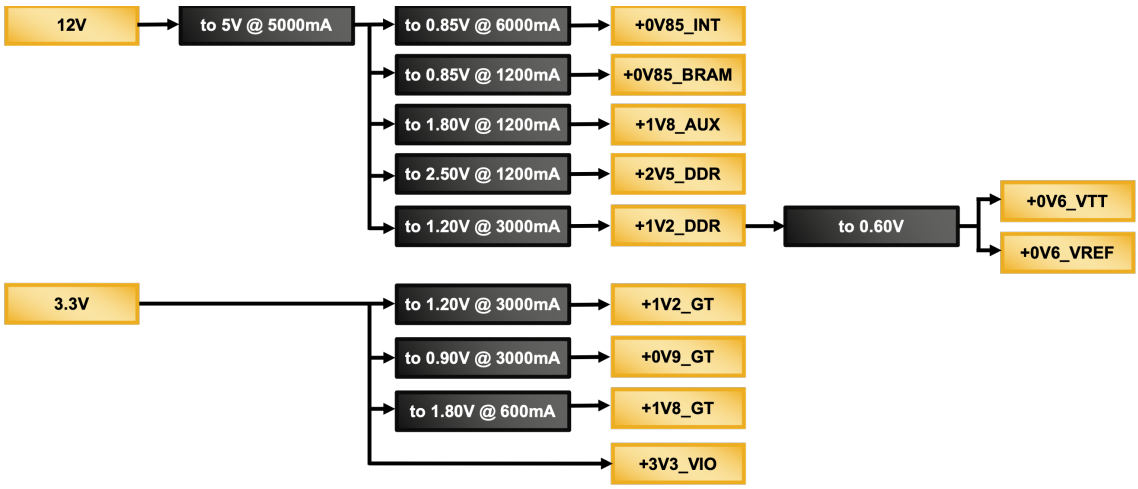
Note:
(1) Power includes HSSI/SERDES in addition to logic and registers/memory blocks

For the CPNX solution, more detailed measurements have been conducted on the CertusPro-NX PCIe Bridge Evaluation Board. With reference to the measurements sites as reported in the User Guide FPGA-EB-02056-1.0, the following rails have been probed during live operation.

Power Rail	Jumper	Voltage (V)	[CPNX] Power (mW) per Rail	
			Average	Std. Dev.
VCCAUXSDQ0	J1	1.8	0.60	0.52
VCCAUXSDQ1	J2	1.8	0.00	0.00
VCCSD	J3	1.0	163.00	1.00
VCCADC1V8	J10	1.8	0.00	0.00
VCC_CORE	J21	1.0	543.33	5.77
VCCAUX_1V8	J24	1.8	73.80	4.76
VCCIO3_1V1	J25	1.1	21.63	0.64
VCCIO5_1V8	J26	1.8	0.60	1.04
VCCIO6_3V3	J27	3.3	9.35	0.95

VCCIO2_IN	J28	3.3	0.55	0.95
VCCIO7_3V3	J29	3.3	14.30	1.91
VCCIO1_3V3	J30	3.3	2.20	1.91
VCCIO0	J34	3.3	4.40	1.91
Total power (mW)			**833.77**	

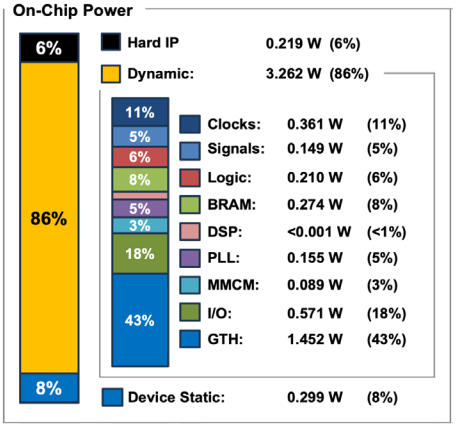
Power tree analysis for three compared solution is the starting point to evaluate the overall energy budget of the different solutions.



Power analysis from implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**3.781 W**
Design Power Budget:	**Not specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**35.3°C**
Thermal Margin:	64.7°C (23.2 W)
Effective θJA:	2.7°C/W
Power Supplied to off-chip devices:	0 W
Confidence Level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity



The first power tree reported is the one designed for the Solution A. Despite partitioning into 11 distinct rails, the reference design relies on 5 different kind of DCDC and LDO parts, due to the very strict and specific low SNRR requirements set by the silicon device. This generated a sub-optimal scaling economy in the BOM and an overall complex design to be maintained.

Solution A: AMD AU15P SoM – Power Tree Summary	
Maximum power consumption	37.2 W
FPGA power	3.781 W (incl. 1.452 W SERDES)
DDR4 power	1.824 W
AUX and other peripherals	>20 W

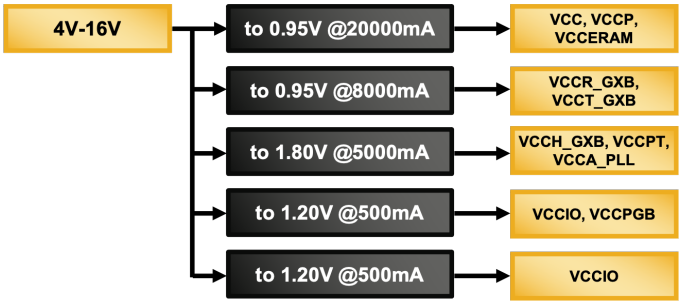
In terms of BOM for Power Management ICs (PMIC), the Solution A summarizes as:

- 4 buck converters max 3 A, 4 buck conv. at 4 A/6 A
- Integrated power modules used to save inductor space
- High power rails need expensive inductors and capacitors

Solution A PMIC	Qty	Price (USD at 1K MOQ)
Buck converters	8	$ 8.22
Power inductors	4	$ 7.32
Capacitors	28	$ 7.10

The planned BOM cost for Solution A PMIC is **$22.64 USD**.

Similar standpoint is for the Solution B. Despite showing a simpler power tree, with a few collapsed rails into 5 principal rails, the extreme power demand for the VCC (FPGA core voltage) and transceivers required a very expensive BOM. Real estate is affected as well, as it is possible to see in the tabular data.



Solution B: Intel A10 GX160 SoM – Power Tree Summary	
Maximum power consumption	>45 W
FPGA power	8.4 W (incl. 2.983 W SERDES)
DDR4 power	3.187 W
AUX and other peripherals	>20 W

In terms of BOM for PMIC, the Solution B summarizes as:

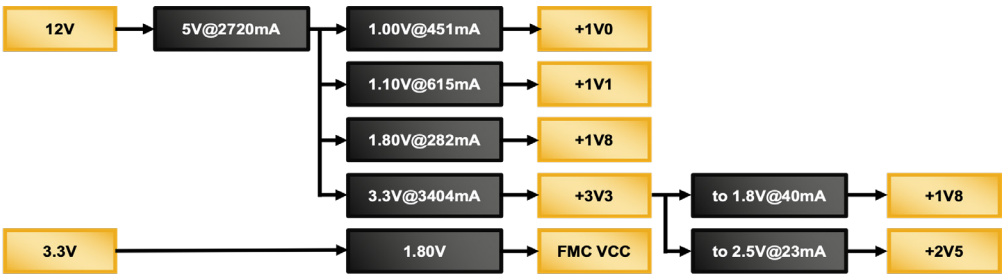
- 5 buck converters max 3 A, 3 buck conv. at 8 A/20 A
- Integrated power modules used to save inductor space
- High power rails need expensive inductors and capacitors

Solution B PMIC	Qty	Price (USD at 1K MOQ)
Buck converters	8	$ 18.38
Power inductors	5	$ 7.77
Capacitors	28	$ 6.70

The planned BOM cost for Solution B PMIC is $ 32.85 USD. As an additional note, the overall BOM contains some DCDC which are marked as NRND and show small or zero stock availability at the time

of writing. This is a crucial indicator when dealing with long-term industrial designs, which require reliable accessibility for spare parts in a 5/7/10 years time span.

The CPNX has a simpler PMIC in the considered Reference Design:



CPNX: Lattice CertusPro NX Ref. Board – Power Tree Summary	
Maximum power consumption	14.976 W
FPGA power	0.924 W (incl. 0.277 W SERDES)
DDR4 power	0.711 W
AUX and other peripherals	12 W

This solution is different than the above two, it is smaller in the overall maximum power drain and it is designed around a simplified scaling strategy (same DCDC for all the principal voltage rails). The choice of LPDDR4 for the design helped to maintain a lower power footprint even beside the FPGA, which by itself is significantly less power hungry than those in Solution A and B. It's also important to note that Solution's A and B have very strict power up and power down requirements which add to the complexity of their overall solution. The CertusPro-NX devices have no requirements for power up or power down sequencing simplifying board design, BOM cost and firmware.

These choices positively affect the BOM buying conditions:

- All DCDC with max 3 A low IQ
- 90% efficiency on most rails
- Low cost inductors and capacitors

CPNX PMIC	Qty	Price (USD at 1K MOQ)
Buck converters	6	$ 0.90
Power inductors	6	$ 3.15
Capacitors	20	$ 6.00

The planned BOM cost for CPNX PMIC is **$ 10.05 USD**

Reduced power consumption allows smaller DCDC islands on CertusPro-NX board, small VCCCORE, AUX currents enable designers to adopt HDI solutions for those power islands and SERDES power efficiency allows miniaturized LDOs usage like those used in mobile designs.

PMIC real estate measurements are therefore positively affected. Here is a table with an in-vivo measurement comparison results:

	PMIC Area
CPNX Reference design	228 mm2
Solution A – AMD AU15P	980 mm2
Solution B – Intel Arria10 GX160	> 1950 mm2

The last item of the analysis is the capability of the FPGA (and its ecosystem) to run in different thermal conditions. The CPNX is the only FPGA in the round capable of operation without any cooling device, neither active nor passive. The Solution A required at least passive cooling, which also included the DDR4 area, while Solution B has never been able to perform in a long session without an active fan (30 mm ball bearing, 12 V, 8,000 RPM, directed on the surface of a case-molded heatsink).

Design	Cooling Strategy to Safely Operate		
	Open Air	Passive Cooling	Active Cooling
Solution A – AMD AU15P	NO	YES	YES
Solution B – Intel Arria10 GX160	NO	NO	YES
CPNX Reference Design	YES	YES	YES

1. **Solution A FPGA:**
   - **Power Consumption:** Higher than expected, mostly due to its high-performance capabilities.
   - **Space Requirement:** Medium, necessitating adequate cooling and PDN.
   - **BOM Cost:** High due to robust PDN components.

2. **Solution B FPGA:**
   - **Power Consumption:** The highest in the study, it has significant impact on the design strategy.
   - **Space Requirement:** Larger space for cooling and PDN.
   - **BOM Cost:** High, with non-zero risks on procurement.

3. **CPNX FPGA:**
   - **Power Consumption:** More efficient, leading to lower overall power usage.
   - **Space Requirement:** Smaller than average, as less extensive cooling and PDN are required.
   - **BOM Cost:** Balanced, with costs driven by the need for efficient I/O power management.

Conclusion

This comparative analysis illustrates that the choice of FPGA significantly impacts the performance, power consumption, and resource utilization of the same RTL design. Each FPGA brand brings its strengths and trade-offs, emphasizing the importance of aligning the FPGA selection with the specific requirements of the design and the target application.

The power consumption profile of an FPGA has a direct and substantial impact on both the physical space requirements for the power section and the cost of the bill of materials. While high-performance FPGAs may offer superior computational capabilities, they often come with higher power demands, necessitating larger cooling solutions and more complex PDNs, which in turn increase the BOM cost. Conversely, power-efficient FPGAs can reduce both space and cost burdens. Therefore, a careful evaluation of power-related factors is essential in FPGA selection, particularly when considering the implications for the overall design footprint and cost-efficiency.

Future Directions

Further studies could explore the impact of advanced FPGA features such as embedded hard IP blocks, high-speed transceivers, and adaptive logic modules on the performance of standard RTL designs.

Additionally, the study could be expanded to include newer FPGA models and emerging technologies in the field.

Glossary

BOM (Bill of Materials)	A comprehensive list of materials, components, and assemblies required to construct, manufacture, or repair a product or service.
CPLD (Complex Programmable Logic Device)	A programmable logic device with complexity between that of PALs and FPGAs.
Degrees Celsius (°C)	A scale and unit of measurement for temperature.
DSP (Digital Signal Processor)	A specialized microprocessor designed specifically for digital signal processing, usually in real-time.
Dynamic Power	The power consumed by an electronic device due to the switching of its transistors.
Embedded Clock	A clock signal that is embedded within the data stream, often used in high-speed data transmission standards.
FPGA (Field Programmable Gate Array)	A type of digital integrated circuit that can be programmed after manufacturing to become almost any kind of digital circuit or system.
Gbps (Gigabits per second)	A unit of data transfer rate equal to 1,000 megabits per second.
Heat Sink	A passive heat exchanger that transfers heat generated by an electronic or a mechanical device to a fluid medium, often air or a liquid coolant.
IP Core (Intellectual Property Core)	A reusable unit of logic or a chip layout design that is the legally recognized property of one party.
PCIe (Peripheral Component Interconnect Express)	A high-speed serial computer expansion bus standard designed to replace older bus standards like PCI.
RTL (Register Transfer Level)	A design abstraction which models a synchronous digital circuit in terms of the flow of digital signals (data) between hardware registers and the logical operations performed on those signals.
SERDES (Serializer/Deserializer)	A pair of functional blocks commonly used in high-speed communications to compensate for limited input/output. These blocks convert data between serial data and parallel interfaces in each direction.
SLVS (Scalable Low-Voltage Signaling)	A low-voltage, high-speed interface often used in semiconductor design.
SLVS-EC (Scalable Low-Voltage Signaling with Embedded Clock)	An advanced version of SLVS that embeds the clock signal within the data stream for more efficient signaling.
SoC (System on Chip)	An integrated circuit that integrates all components of a computer or other electronic systems.

Static Power	Also known as leakage power, it is the power consumption of an electronic device when it is in a non-switching state
Thermal Conductivity (W/m·K)	A measure of a material's ability to conduct heat.
Thermal Resistance (°C/W)	A measure of a material's resistance to heat flow.
Watt (W)	The SI unit of power, equivalent to one joule per second.

Acknowledgement

This whitepaper was supported by Mario Vigliar, and we thank him for his contributions.

Mario Vigliar was born in Nocera Inferiore, Italy, in 1980. He received the M.Sc. and Ph.D. degrees in computer science from the University of Salerno, Salerno, Italy. Since 2007, he has been involved in the study and development of heterogeneous multi-core and multi-processing imaging systems. He is Chief Scientist of Spark/TTM, still holding the position of CTO of his own company, DPControl, Nocera Inferiore, Italy. He holds two international patents on innovative architectures for Visual Search, licensed by STMicroelectronics NV, and has written for several publications about video processing, image compression, and linear algebra computing systems in FPGA and ASIC.