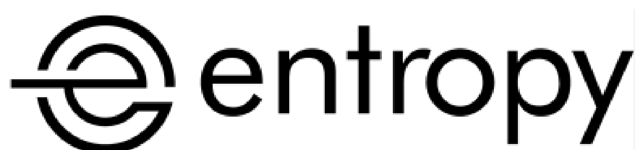
Can You Really Simulate an FPGA?



CAN YOU REALLY SIMULATE AN FPGA DEVICE?

WHAT IS AN FPGA?

HOW IS AN FPGA DIFFERENT TO A CPU?

An FPGA is a semiconductor device that is based around a matrix of programmable logic. The functionality of the device can be specifically programmed by the design engineer giving huge flexibility to FPGA integration. This flexibility comes at a cost of increased time, and skill, needed for development when compared to an off-the-shelf ASIC.

FPGA structures vary depending on the vendor, but fundamentally they follow the same structure. Basic functional logic elements are connected together through programmable interconnections between fixed wires. Figure 1 shows the functional units as grey boxes, IO elements as white boxes and black lines are wires and programmable interconnects. The strategic connection of these functional units can replicate larger scale logic units, such as processors or memory, in an interconnected network on the chip.

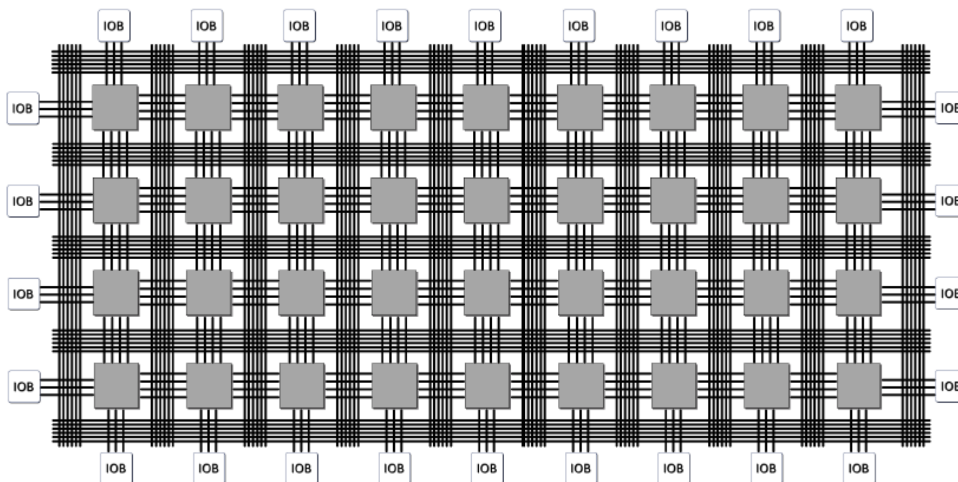


FIGURE 1 - FPGA STRUCTURE [1]

In an FPGA most of the delay in the chip comes from the interconnect. In order to connect one functional unit to another functional unit in a different part of the chip often requires a connection through many transistors and switch matrices, each of which introduces extra delay [2]. Figure 2 shows in more detail the level of switching required to determine connectivity between functional blocks.

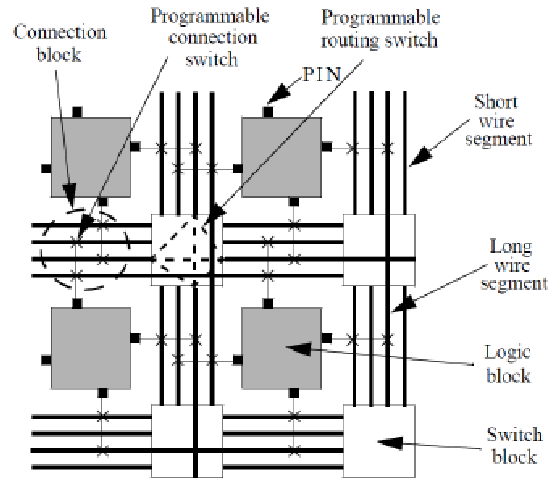


FIGURE 2 - FPGA INTERCONNECTING SWITCH METHOD

The sheer programmability of FPGAs implies that more transistors are needed to implement a given logic circuit in comparison with custom ASIC technologies. This leads to a higher power consumption per gate and increased power demand per device [3].

FPGA SHAPES + LOGIC

WHAT ARE THEY?

The configurability of an FPGA is delivered using large amounts of logic collated into fundamental building blocks called CLBs (Configurable Logic Blocks), formed of smaller components: flip-flops and look-up tables. The allocation and control of signals between these blocks drives the functionality of the FPGA and so position and density of active logic is entirely user dependent. This presents a real challenge to accurate thermal simulation of a device, and using a typical junction to case resistance can give wildly inaccurate results as operability is adjusted device to device.

Designing an FPGA architecture from scratch using only these CLBs is extremely labour intensive due to the high level of functional detail needed for modern computing. When developing logic using CLBs only, the resultant logic is referred to as “soft blocks”, so named because of their high configurability.
“Hard blocks” by contrast are embedded functionality on an FPGA that can only be used for a predetermined purpose. Examples of these could be processors, memory blocks or high-speed transceivers. These are beneficial in that they have optimised routing and increased logic density allowing for reduced timing restrictions, while consequentially they reduce the configurability of the chip [4] and significantly increase the heat flux density in these dedicated areas [5].

HOW ARE THEY DISTRIBUTED?

When a circuit is implemented, the place and route tools place critical logic close together and spread other logic as far as allowed by the circuit constraints [6]. Logic distribution is typically dependent on, and local to, pinout placement as timing restrictions on certain IO require minimised interconnection length between pin and active logic. This mirroring is not a perfect prediction however as: not all logic requires these strict timing restrictions; the die size is typically much smaller than the solder balls; communication with other control logic may need more optimal placement; and physical logic must be available and so can divert routing. Software such as Intel’s Quarts Prime or Xilinx’s Vivado Design Suite will handle the majority of this floorplanning and can also offer the user the opportunity to prioritise switching performance, thermal performance, or a balance between the two. Unfortunately from a thermal perspective, choosing this focus may significantly affect the latency of an FPGA where some high density logic is critical to functionality of the device, and this option is rarely available.

WHAT CREATES HEAT IN A CHIP?

In any transistor based switching device, some power will be lost as heat due to inefficiencies in the device or due to the nonzero resistances to current in a gate [7]. This is ubiquitous for all semiconductor architecture and creates a requirement for suitable chip and system level cooling.

Accurate thermal management of these devices is critical to maintaining the desired operating lifetime of electronic devices, which is exponentially shortened by increasing temperature [8]. Overengineering a thermal solution by contrast can have a negative impact on a product by increasing undesirable factors, such as mass and cost.

Thermal power dissipation in FPGA CMOS transistor devices can primarily be divided into dynamic and leakage, also known as static, power dissipation. Dynamic losses arise from capacitive charging and discharging of the transistors, and short circuit power, typically providing the majority of thermal dissipation in an FPGA device. In legacy FPGAs, dynamic power contributed up to 67% of power usage, with static power providing just 22%. In more recent 28nm devices, static power has increased its dissipation contribution to closer to 40% of the total thermal loss [9]. The lack of knowledge of how exactly leakage power is distributed across the chip leads to highly inaccurate power traces, and therefore unreliable thermal estimation [10].

The dynamic power varies greatly with design and is characterised in detail through vendor's power estimation tools (Intel's Powerplay Quartus or Xilinx's XPower, for example). It is a function of the known logic quantity, switching frequency and toggle rate.

$$\boldsymbol{P_{dynamic}} = \left[\frac{1}{2}\boldsymbol{CV^2} + \boldsymbol{Q_{ShortCircuit}V}\right]\boldsymbol{f \cdot activity} \tag{1}$$

Where C is the capacitance of the transistor, V is the power rail voltage, $Q_{ShortCircuit}$ is the power consumed during a change in the CMOS logic gate, f is the net frequency, and *activity*, or toggle rate, is the average number of signal transitions relative to a clock rate (%).

Leakage power is as a result of the non-infinite resistance across an inactive gate threshold, and is heavily dependent on the device temperature [11]. The following equation describes the exponential relationship between leakage power, P_{leak}, and temperature, T.

$$\boldsymbol{P_{leak}} = \boldsymbol{P_0} \times \boldsymbol{e^{-k/T}} \tag{2}$$

Where P_0 and k are process dependent constants [10].

Unlike in an ASIC, logic that is not utilized in an FPGA remains on the device and so remains powered even though not in use, creating a large power demand for a device even with a low logic load. Static power generally does not vary significantly with logic utilization, but is more greatly dependant on the amount of logic on the die [9] [12]. The impetus is therefore on the design engineer to select the smallest device for the given application.
While the design engineer should take steps to ensure that their FPGA has been sized correctly for the functionality desired, it is almost impossible to achieve full utilization in a device due the limited supply of programmable routing resources. In the writers experience, a highly utilized FPGA architecture holds around 75% utilization, which is not an unreasonable estimate given the 62% utilization identified in [13]. This report is slightly dated and it is expected that FPGA technology has developed since then. For an extreme example, Xilinx claim the Ultrascale can accommodate utilization of up to 90% [14] although this will be entirely dependent on the desired functionality.

Circuit gating is an additional option commonly utilized to reduce power [15], whereby blocks of unused logic are "turned off" from the voltage rail until needed and so have no static power draw. This technique has been shown to be used by Xilinx [14] and Intel [16] on recent devices.

A significant contributing factor to both the dynamic and static power dissipation in an FPGA is the joule heating of the interconnects. Research completed by Shang et al [17] shows as much as 50%-70% of the total power dissipated in a Xilinx Virtex-II was from the interconnection network, shown in Figure 3. While the allocation of this loss to the static or dynamic power contribution is not fully determined, it is expected that with the programmable nature of the interconnect switching the majority of this power is concentrated around active logic.

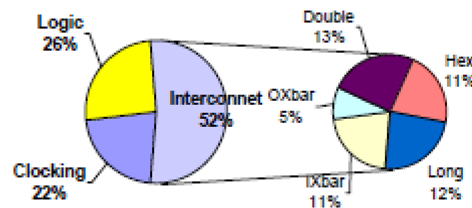


FIGURE 3 - THE POWER DISTRIBUTION IN A "REAL" FPGA CIRCUIT [17]

This high dissipation factor is a result of significantly longer interconnect lengths in FPGAs than ASICs due to the larger area consumed by the logic [3]. Observing the joule heating and electrical resistivity equations we can identify this relationship.

$$P = I^2R \tag{3}$$

$$R = \frac{\rho L}{A} \tag{4}$$

Where *P* is power consumed, *I* is electrical current, *R* is the wire resistance, *ρ* is the resistivity of the material, *L* is the length of the wire and *A* is the cross sectional area.

Equations (3) and (4) can then be combined to give a linear relationship between the length of and the power dissipated in the interconnect.

$$P = I^2 \frac{\rho L}{A} \tag{5}$$

Finally, the most power hungry input on an FPGA will usually be the power rail, often denoted as V_{cc}. This is understandable because the core power rail drives the logic, the use of which is central to any FPGA design [18].

HOW TO SIMULATE IT

When designing a rugged embedded product, either using vendor provided Power Estimator spreadsheets or Power Analyzer tools, simply calculating power output at an estimated junction temperature is not sufficient. These spreadsheets rely on a known ambient temperature, and a predetermined thermal resistance for any attached heatsink. In good rugged system design, the resistance of the heatsink will be dependent on the cooling requirements of the system, including adjacent thermally critical devices, and be both cost and mass efficient. Using a pre-determined thermal resistance will not allow for optimisation of the thermal solution.

In high ruggedization environments the enclosure temperature can be set at +85°C, meaning the junction temperature must be higher than this. Figure 4, taken from a Xilinx White Paper, describes that the leakage power estimate becomes unstable at temperatures above 85°C.

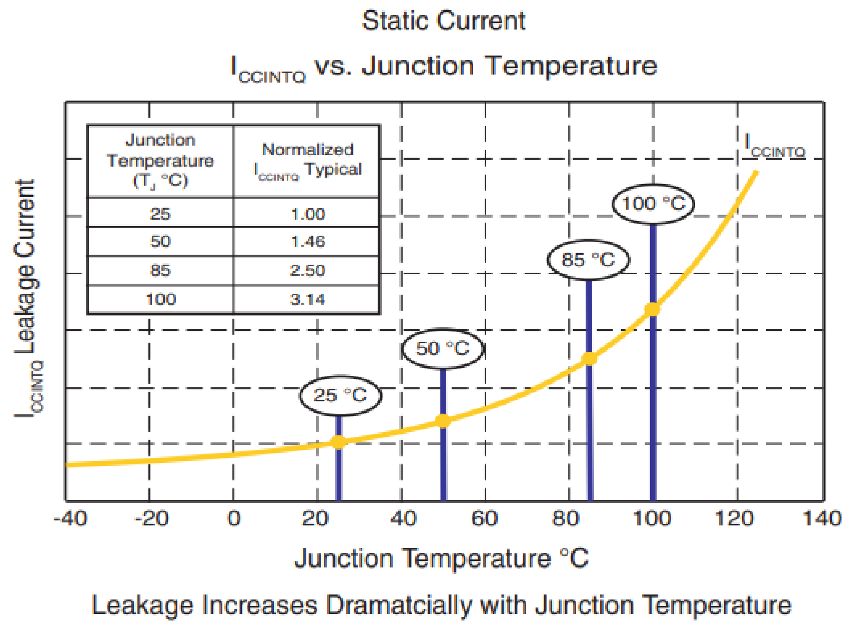


FIGURE 4 - LEAKAGE CURRENT VS TEMPERATURE [19]

To validate this curve, we tested a notional Ultrascale architecture for its power dependence with junction temperature. We were able to vary the junction temperature of the device and observe the resultant power output given by the Power Report, Figure 5 shows the results. Considering Equations (1) and (2), the dynamic power is independent of temperature and so increasing power loss with increasing temperature is as a direct result of the leakage power only.

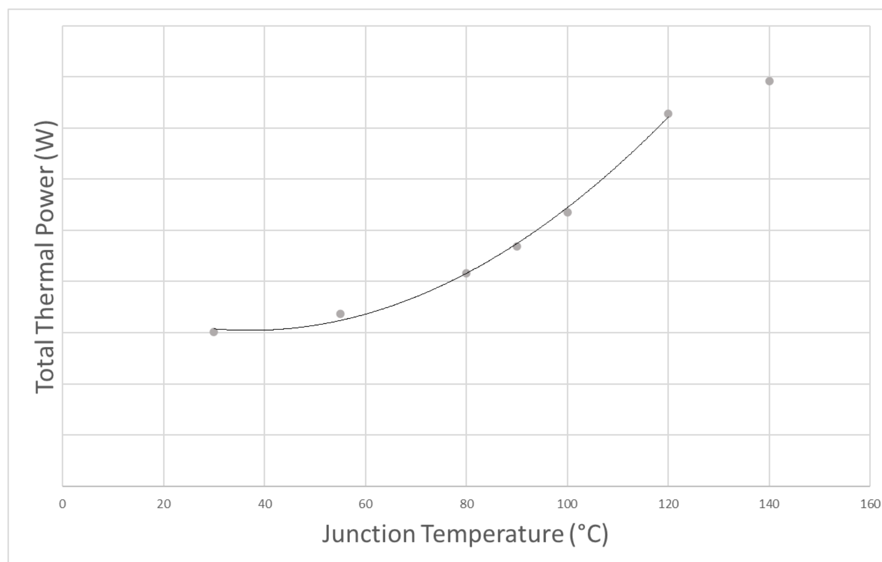


FIGURE 5 - THE RELATIONSHIP BETWEEN TOTAL POWER AND JUNCTON TEMPERATURE IN VIVADO

The power report results almost perfectly follow the exponential nature of leakage power rise in the FPGA. It is critically important therefore that the junction temperature used in the Power Estimator is reflective of the junction temperature identified in a thermal simulation. This will be an iterative process to convergence on some stability of the design and ensure accuracy, see Figure 6.

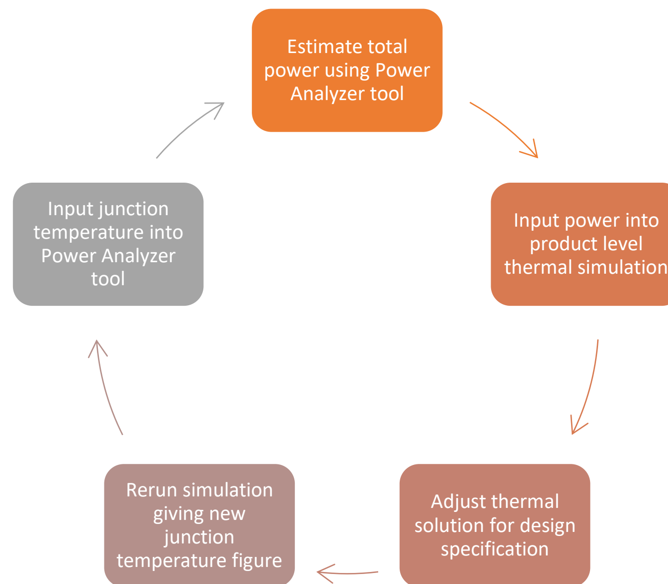


FIGURE 6 - ITERATIVE WORKFLOW TO CONVERGE ON CORRECT POWER FIGURE FOR THERMAL SOLUTION

The anomalous power at 140°C is most likely due to a predictive throttling result occurring when the device has passed maximum junction temperature. When running the test at 140°C, the Power Report function flagged a warning that the device had gone outside of its operating range and it is expected that the dynamic power of the device was significantly throttled back to reduce the impact of this.

It should be noted that this test was performed on a Xilinx device, which can have 2-3 times lower static power consumption that other competitive FPGAs [19]. Estimating this junction temperature accurately is therefore even more critical for these devices of higher current instability.

THE TYPICAL APPROACH TO JUNCTION TEMPERATURE DISTRIBUTION

Considering the unequal and varied distribution of logic within each individual FGPA architecture, there is inherent inaccuracy in assuming the temperature and therefore power in the die can be considered with a single value, or applied uniformly across the surface of the die [20]. This effect is likely to be mitigated with the onset of increased leakage power significance on small transistor dies and with the wide distribution of interconnect heating, but there will inevitably be some variance. Both Intel and Xilinx provide Delphi and detailed IC models of their FPGAs, however these are calibrated using a uniform heat flux on the die only. Intel reports that this method gives an average accuracy of only 10% for the resistance of the device [21].

THERMAL DESIGN TOOLS

Given the variability of factors described above, an accurate thermal simulation can only be truly achieved with a user guided approach bespoke to each FPGA architecture. Some research [10] [6] [22] has been completed into how to more accurately predict the temperature variation within the silicon. W. Huang et al [23] proposed a modelling methodology, HotSpot, which divides the FPGA die into discrete blocks. Each of these blocks has assigned a thermal resistance generated from the geometry of the block and material properties of the silicon die, and an assigned thermal power.

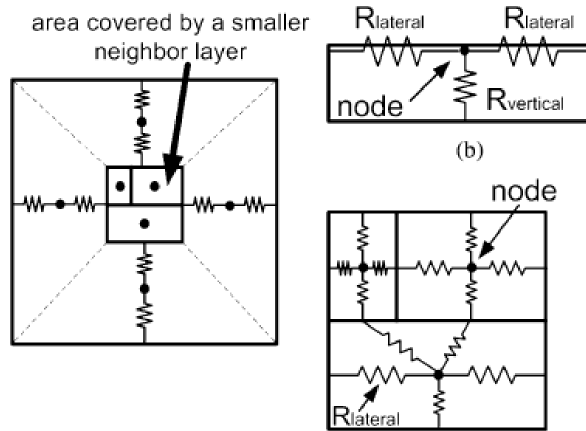


FIGURE 7 - SHOWING THE DISCRETE METHOD HOTSPOT EMPLOYS TO DERIVE LOCAL DIE TEMPERATURES

This method has shown tremendous accuracy for temperature distribution, but it does not describe the method of identifying power source and its dependency with temperature.

To extract much more applicable thermal information for embedded design, Amouri et al [10] describe a process which utilises the Hotspot methodology for estimating temperature variation, but iteratively calculates the impact of leakage power distribution across the die using the following inputs available from FPGA development tools:

- Die dimensions (taken from datasheet)
- A floorplan circuit description of the device
- A detailed power report

This method assumes initially that the junction has a uniform temperature and so leakage power is evenly distributed across the die. It then provides Hotspot with power per block information on the given design, which in turn calculates the temperature distribution, which is in turn discretized and fed into a leakage model based on Equation (2).

Figure 8 shows this process pictorially. Please note that the example from the literature utilised a Xilinx architecture to create this model, hence reference to XDL (Xilinx Design Language). Intel's Quartus tool can also output power dissipation by block [9] and so may also be applicable to this methodology.

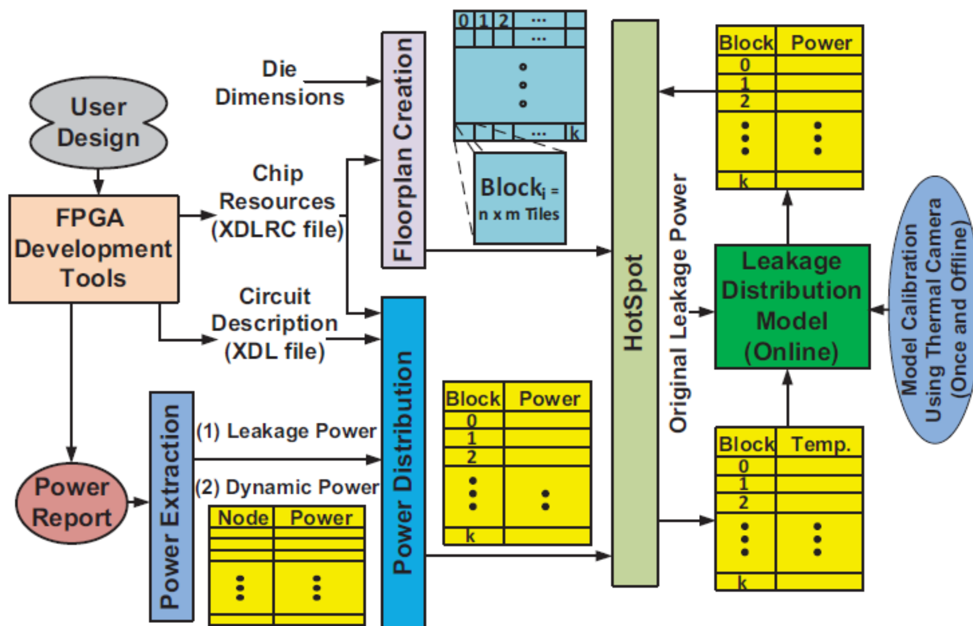


FIGURE 8 - PROPOSED TEMPERATURE INFORMATION FLOW FOR ACCURATELY REPRESENTING LEAKAGE POWER USING A XILINX BASED DEVELOPMENT TOOL [10]

This method is the most complete found in literature research and if implemented correctly will provide a designer with a highly accurate temperature distribution (average error of 1°C) across a die [10].
Using the discretized power data, this can be applied to the detailed IC package geometry within a CFD package giving genuine high confidence in thermal simulation results.

CONCLUSION

Fundamentally speaking, any operation dependent simulation of an FPGA has flaws due to the incredibly labour-intensive process needed to accurately predict the performance of the device. Any simulation will only be as good as the input data it receives and given the given Power Analyser accuracies of 10% and 20%, from Intel and Xilinx respectively, this is an upper limit to thermal simulation possibilities. While inaccuracy this is not necessarily a given, and power figures should always be validated with test, it should be treated with concern by any thermal engineer.

The 10% inaccuracy derived from uniform heat flux thermal models can however be significantly improved upon by considering the physical floorplan of each device. Using available software tools and processes, thermal simulation can be tailored not only to a device level, but to an architecture level with a high degree of accuracy. Thermal engineers should be mindful that while test data has shown these iterative simulation studies can provide an impressive 1°C of accuracy, there is no validation of power figures at the test stage. These results typically utilise an off the shelf heatsink which is applicable with the given development tools, while more complicated heatsink design should be calibrated against the power estimator results.
Until further validation can be provided and quantified for the true power consumption of an FPGA, a conservative solution should always be evaluated. For a Xilinx device, this may be as significant as simulating a device at 48W for a given 40W power consumption.

REFERENCES

[1] L. Santangelo, "Viv2XDL: a bridge between Vivado and XDL based software," University of Pisa, Pisa, 2014.

[2] B. Zeidman, "All about FPGAs," EE Times, 22 03 2006. [Online]. Available: https://www.eetimes.com/all-about-fpgas/.

[3] J. H. Anderson and F. N. Najm, "Power Estimation Techniques for FPGAs," IEEE, 2004.

[4] J. M. Weber and M. J. Chin, "Using FPGAs with Embedded Processors for Complete Hardware and Software Systems," American Institute of Physics, 2006.

[5] P. Sundararajan, A. Gayasen, N. Vijaykrishnan and T. Tuan, "Thermal Characterization and Optimization in Platform FPGAs," ICCAD, 2006.

[6] S. Velusamy, W. Huang, J. Lach, M. Stan and K. Skadron, "Monitoring Temperature in FPGA based SoCs," IEEE, San Jose, 2005.

[7] Engineering Entropy, "Engineering Entropy," February 2020. [Online]. Available: https://secureservercdn.net/160.153.138.53/nm2.751.myftpupload.com/wp-content/uploads/2020/02/3.-What-actually-is-TDP-and-why-is-it-important-3.pdf?time=1589271825.

[8] V.Lakshminarayanan and N.Sriraam, "The Effect of Temperature on the Reliability of Electronic Components," IEEE, Bangalore, 2014.

[9] Intel FPGA, "Power Analysis," Intel Corporation, 13 Nov 2018. [Online]. Available: https://www.youtube.com/watch?v=8y6M-rmz19I.

[10] A. Amouri, H. Amrouch, T. Ebi, J. Henkel and M. Tahoori, "Accurate Thermal-Profile Estimation and Validation for FPGA-Mapped Circuits," IEEE, Karlsruhe, 2013.

[11] A. Kushwaha, G. Verma and V. K. Kakar, "Thermal Analysis and Modelling of Power Consumption for FPGAs," International Conference of Advances in Computing and Communication Engineering, Paris, 2018.

[12] T. Tuan and B. Lai, "Leakage Power Analysis of a 90nm PGA," IEEE, 2003.

[13] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. Irwin and T. Tuan, "Reducing Leakage Energy in FPGAs Using Region-Constrained Placement," Monterey, 2004.

[14] Xilinx Inc., "Proven Power Reduction with Xilinx UltraScale FPGAs," Xilinx Inc., 2015.

[15] J. Lach, J. Brandon and K. Skadron, "A General Post-Processing Approach to Leakage Current Reduction in SRAM-based FPGAs," IEEE, 2004.

[16] Intel Corporation, "Intel® Stratix® 10 Power Management User Guide," Intel Corporation, 2020.

[17] L. Shang, A. Kaviani and K. Bathala, "Dynamic Power Consumption in Virtex™-II FPGA Family," 2002.

[18] Intel Corporation, "Understanding and Meeting FPGA Power Requirements," Intel Corp, 2017.

[19] Xilinx, Inc, "Static Power and the Importance of Realistic Junction Temperature Analysis," Xilinx, Inc, 2005.

[20] Intel FPGA, "Thermal Management in Intel® Stratix® 10 Devices," 23 Jan 2018. [Online]. Available: https://www.youtube.com/watch?v=IiX97BwjhyM&t=331s.

[21] Altera Corporation, "Thermal Management for FPGAs," Altera Corporation, 2012.

[22] W. Huang, K. Skadron, S. Gurumurthi, R. Ribando and M. Stan, "Differentiating the Roles of IR Measurement and Simulation for Power and Temperature-Aware Design," IEEE, Boston, 2009.

[23] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron and M. Stan, "HotSpot:ACompact Thermal Modeling Methodology for Early-Stage VLSI Design," IEEE, 2004.