*Int J Software Informatics,* Volume 8, Issue 2 (2014), pp. 193–205
International Journal of Software and Informatics, ISSN 1673-7288

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661040

# Hardware Security in the Dark

Aaron Carpenter and Yu Chen

(Dept. of Electrical & Computer Engineering, Binghamton University, USA)

**Abstract**

The increasing transistor count on a single chip provides an unprecedented amount of resources for chip designers. Unfortunately, the power consumed by each transistor does not shrink similarly, decreasing the amount of transistors that can be on simultaneously. This *utilization wall* leaves a growing percentage of transistors *dark*, or powered-off, as the chip cannot (a) provide the necessary current or (b) maintain a low operating temperature. To account for *dark silicon*, the computer architecture community has begun taking advantage of the wealth of available transistors to design efficient, time-sharing systems, often through specialized architectures.

Meanwhile, security is quickly becoming a first-tier design constraint, increasing the need for hardware security mechanisms, in order to maintain high levels of availability and to detect and protect from intrusion. As we move into the many-core environment, many of these security mechanisms will need to be integrated on-chip. In a chip-multiprocessor environment, security will be necessary as multiple programs or users are sharing resources, thus facilitating attacks. In both a single-user and multiple-user environment, designers can build specialized hardware to provide support for security functions, such as authenticity, cryptography, and intrusion detection.

In this paper, we survey current hardware security trends and provide insight on how future chip designs can leverage dark silicon for more secure designs. We provide preliminary designs and discuss future challenges and opportunities in dark silicon security. The merging of hardware security and dark silicon will facilitate efficient, fast, and secure designs.
**Key words:** hardware security; dark silicon; energy-efficiency

**Carpenter A, Chen Y. Hardware security in the dark.** *Int J Software Informatics*, **Vol.8, No.2 (2014): 193–205. http://www.ijsi.org/1673-7288/8/i189.htm**

## 1 Introduction

As multi-core processors become omnipresent in our digital lives, it becomes increasingly important to provide security mechanisms for the on-chip environment. Typically, security is viewed as a necessity only in specialized hardware and is often disregarded in favor of increased system speed or energy-efficiency.

Meanwhile, as transistors continue to shrink in physical dimensions, chip designers benefit from quadratic growth in transistor count. Until recently, computer architects have used these transistors to integrate more cores and better use the thread level parallelism available to the system. However, as the voltage and power of the transistor no longer scales significantly, the power density of the entire chip grows and becomes prohibitive, both in terms of power distribution and

Corresponding author: Aaron Carpenter, Email: carpente@binghamton.edu
Received 2014-06-27; Revised 2014-09-04; Accepted 2014-09-07.

thermal management. To account for this, chip designers will be forced to keep a large percentage of the system powered off, resulting in *dark silicon*, or transistors that are not powered on at any given time[10,34,35]. This underutilization of available resources provides significant challenges and opportunities to the microprocessor design community.

In particular, these resources can be at least in part devoted to creating or incorporating hardware security mechanisms, without sacrificing what was considered valuable silicon area. By leveraging dark silicon to provide architectural security, we will make future designs to be efficient, fast, and secure.

The work described in this paper describes the foundation for the use of dark silicon for hardware security by designing:

- Moving target defense (MTD) mechanisms for isolating, migrating, and obfuscating victim operation in a multi-user environment.
- Physical unclonable functions (PUFs) to provide large arrays of random numbers, mainly for cryptographic functions and authentication.
- Coprocessor-based heterogeneous chip multiprocessors, for encryption, decryption, tamper/intrusion detection, and other security functions.

The rest of the paper is organized as follows: Section 2 discusses background for and the work related to dark silicon and hardware security. Section 3 introduces the hardware security mechanisms as designed for the dark silicon environment. Section 4 concludes the paper.

## 2 Background & Related Work

This paper discusses the use of power-constrained dark silicon resources to provide hardware security mechanisms. Before presenting this interaction, it is necessary to discuss relevant background and existing research of each topic.

**Table 1 In the post-Dennard scaling era, voltage and power no longer scale. S represents the scaling factor[34,35].**

| Attribute | Dennard | Post-Dennard |
|---|---|---|
| Transistor size | $S$ | $S$ |
| Number of transistors | $S^2$ | $S^2$ |
| Capacitance | $1/S$ | $1/S$ |
| Supply Voltage | $1/S$ | 1 |
| Power | 1 | $S^2$ |

### *2.1 Dark silicon*

Each new transistor generation increases the number of transistors available on a single die. Dennard scaling[7,10,34,35] predicted that as we scaled transistor dimensions, voltage and power would shrink proportionately. As Table 1 shows, as transistors shrink, in order to maintain a constant power consumption, it is necessary to scale

the supply voltage as the reciprocal of the transistor scaling factor. However, device-level complications have slowed voltage scaling, resulting in an exponential increase in power.

Recent studies have approximated that even though the number of cores on a chip could grow to be over 400 for the 8nm transistor node, only a small percentage of these cores can be on at any given time[10, 35]. As interest in dark silicon grows, researchers will (a) try to avoid dark silicon by decreasing power consumption, and (b) accept that not all components can be on and take advantage of the wealth of resources now available, likely through specialized architectures.

Taylor explains the possible directions architects and circuit designers will likely take to prevent dark silicon or to take advantage of the additional resources without causing further power issues[34, 35]. Two of these directions, shrinking chip dimensions with transistor dimensions and creating alternatives to silicon static CMOS devices, are not relevant in the scope of our proposed hardware security mechanisms and are left to the reader to further explore. The remaining two categories, dim components and specialized hardware, will be discussed in more detail.

As illustrated in Figure 1, dim silicon means that some transistors/components on-chip are selectively run at a lower supply voltage and a slower clock, utilizing the benefits of near-threshold computing techniques[3,15,24,25,38]. This will lower the power consumption of individual components, but will allow more components to be active simultaneously. This technique comes at the cost of computational speed. Recent work has argued that using this technique effectively can eliminate dark silicon completely, at least in the short term[33-35,38]. We can similarly use these dim silicon approaches to run non-time-critical security hardware on components (either specialized or homogeneous) without significantly effecting the power consumption of the entire system.

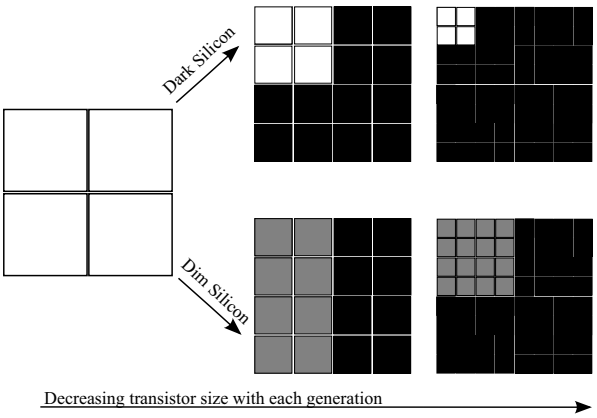


Figure 1. If some cores are under-clocked and powered by a lower $V_{DD}$, performance suffers, but more cores can be on simultaneously. The top path assumes homogeneous power and clocking, and the bottom shows when dim silicon is implemented.

Converse to using lower power components, *computational sprinting* uses nominal voltage circuits, but only for short bursts of time, creating temporarily higher temperatures[22]. This can be useful to get higher performance for small periods of time[22, 35, 34, 39].

The last technique presented is that of *specialization*. Specialized hardware provides energy efficient hardware solutions to specific computational problems. These components would be powered off when not in use. When the system determines the accelerator would be more efficient for a specific task, computation is offloaded from the general purpose core to the specialized core. These components are special-purpose, and therefore are not useful for all computation. However, for specific algorithms, they provide higher efficiency. We will leverage this in Section 3 to provide specialized hardware security components[5,12,17,21,34,35,37,40].

Dark silicon is a relatively new focus in the computer architecture community. Nonetheless, there has been significant research done in dark silicon, mainly focusing on improving energy efficiency and performance through the previously discussed dim silicon and specialized systems. The work presented here is orthogonal to previous research as it explores dark silicon specifically for hardware security purposes.

### *2.2 Hardware & architectural security*

#### 2.2.1 Multiple on-chip users

Out of 400 cores, it is possible many programs will not be able to use more than 32 cores[10]. To effectively use the available resources, even in a dark silicon environment, it will be necessary to populate the cores with multiple programs and/or users. These cores would then be vulnerable to architectural level attacks, as resources in a chip multiprocessor (*e.g.* caches, on-chip networks, memory ports) are shared. This provides attackers with opportunities to snoop on or disrupt functionality of on-chip victims. Therefore, it is necessary to provide security measures to prevent side-channel attacks and denial-of-service (DoS) attacks. While this is already a pressing issue in standard chip multiprocessors (CMPs), when significantly more cores are integrated and dark silicon is accounted for, security is a first-tier concern. Fortunately, dark silicon also provides significant opportunities to prevent attacks in the shared CMP environment.

#### 2.2.2 Security-specific hardware architecture

Security-specific hardware architecture has been studied intensively in past decades. Security-oriented hardware architecture aims at confidentiality and integrity of software and data storage, process isolation, and confidential interrupt handling, while improving performance of the functions[1]. Secure hardware ranges from small processors costing relatively little to serious industrial strength cards with extensive physical tamper protection, integrated high speed FPGAs, and hardware random number generators such as on the IBM 4758[9].

The proposed hardware solutions can be divided into two major categories. Secure processor architectures try to provide a hardware *root of trust*, which cannot be compromised by malicious software or users' mis-operation. For example, the Trusted Platform Module (TPM) suggested by the Trusted Computing Group (TCG) specifies a hierarchal level of functions which can extend/limit the trust boundary[13]; the Execute-only Memory (XOM) architecture focuses on hardware exclusively executing instructions without manipulation[18,19]; and AEGIS, an architecture that allowed users to operate within a tamper-evident and

tamper-resistant environment[31].

The other category includes trusted and secure co-processors, a processor that works with the central processing unit (CPU) to handle computational heavy security protocols. The secure coprocessor is specifically designed to be tamper resistant, reliable facing physical damages, computationally powerful for security protocols, secure in communications, portable, and within a reasonable budget[28,29]. For example, cryptographic processors for security are used to ensure the integrity of financial transactions[23,30].

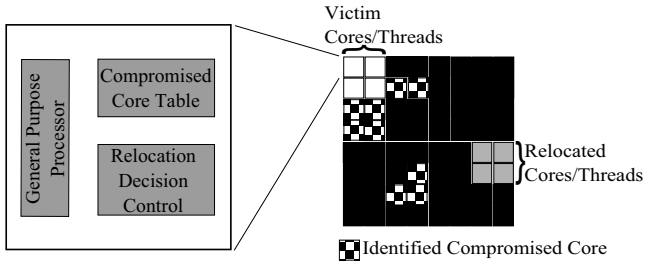


Figure 2. If an attack is detected, the victim cores are relocated to previously dark cores, which can be assumed as safe, as long as these cores are not identified as compromised.

## 3 Hardware Security Mechanisms in Dark Silicon

In the era of dark silicon, transistors are considered abundant resources. This power-constrained environment can be refocused to provide embedded on-chip security mechanisms, including isolation, moving target defenses, physical unclonable functions, and security-specific coprocessors.

### *3.1 Isolation*

Most current programs cannot use more than a few cores effectively, necessitating the incorporation of multiple independent users onto a single chip, in order to effectively use available resources. In the dark silicon era, 50–90% of the chip will remain off. This highlights thread-core mapping algorithms (*e.g.* how to map programs to the available cores) as a central issue[35,10,34], for interference avoidance, thermal management, and obfuscation of program operation. By separating the computation of multiple independent threads, we can limit thermal hotspots, which, is in its own right an important goal. However, we focus here on hardware security, and provide a level of isolation to limit accidental or malicious interference. Threads can be spatially isolated using dark cores as a physical pseudo-barrier to limit direct contact with a victim thread. This type of isolation has been explored in FPGAs to isolate logic blocks[16] and is well suited to the dark silicon environment. These secure sub-systems can be self-contained, pending careful memory and I/O mapping. This is a naïve method of security, but provides the foundation for movable multi-threaded computation clusters.

### *3.2 On-chip moving target defenses*

Isolation alone does not necessarily prevent interference from an intelligent attacker. Moving target defenses (MTDs) migrate victim threads among available

resources to alleviate availability attacks and protect victims from snooping. An on-chip version of MTDs for dark silicon chip multiprocessors is shown in Fig. 2. While MTDs are typically implemented in large distributed computing systems, they have yet to be incorporated into a chip multiprocessor environment. Dark silicon provides a wealth of under-utilized components, as most are turned off. These dark regions can be considered "safe" for new threads. We can use these resources as the destination for a moving target, both under active and passive placement/movement algorithms.

The MTD for dark silicon can be achieved both logically and physically. The "target" that has been exploited by attackers will disappear by migrating the user job to a newly launched core, reloading a different OS, or reconfiguring the on-chip network topology. This ensures that the previously identified vulnerabilities will be hidden or simply be turned off.

Active defense mechanisms detect intrusions or DoS attacks and locate safe resources for relocation[6,8,11,26,41], while passive mechanisms randomly redistribute workloads to avoid potential attacks. An area of dark silicon may be considered a safe area to conduct secure processing, thus a suitable location to move to. The proposed design may be applied to threads running on a single core, or on clusters of cores, depending on the architecture and program needs. In determining the next processing location, we have considered both centralized control as well as distributed control. Distributed control is more robust without the vulnerability of a single point of failure. Each core in the processor contains a general-purpose core and specialized security hardware, including a list of potentially compromised cores and relocation decision hardware.

Upon startup, a secure processor initializes a pseudo random number generator with a seed. This starts the generation of a sequence of random numbers, which is used to determine potential core locations to move processing to. The random numbers correspond to the indices of cores in the core array. The pseudo random number generator will include a 128-bit state register which is updated at a specified interval. Each core also maintains a table of suspicious cores. Intrusion detection can be done using security coprocessors, which will be discussed in Section 3.4. These suspicious cores are ones suspected of being an attacker, but may still reside on-chip. This allows the thread-mapping algorithm to avoid cores in the vicinity of potential attackers upon relocation or initial placement. Updates to this list of cores are broadcast on the on-chip network.

Dark silicon CMPs provide an abundance of available cores. As the victim is migrated, the previously inhabited cores are powered off, maintaining constant power consumption and erasing any trail of the relocated threads. During the transition, a minor increase in power would be necessary, but the short amount of time would not significantly impact the thermal characteristics. This is similar to computational sprinting[22, 39], which offers the additional benefit of allowing short bursts of security-critical computation which once finished will power off, as per the requirements of dark silicon. These short-lived threads would be harder for attackers to locate and disrupt, particularly if sprinting threads are integrated with the moving target defense. While cores would otherwise remain largely underutilized to maintain thermal integrity, the available resources would be re-purposed for MTDs in order to provide secure on-chip

co-habitation of multiple users and/or programs.

**On-going and future work:** Moving target defenses are known techniques in the off-chip environment, but the on-chip constraints change significantly. Existing MTD algorithms need to be evaluated for chip multiprocessors to determine feasibility, then will be adapted to be more suitable for an integrated environment. Evaluation metrics also should be created to quantifiably compare algorithms.

### *3.3 Physical unclonable functions*

Essentially, physical unclonable functions (PUFs) identify digital components of any scale (*e.g.* a device or a computer) using intrinsic, unique, and random physical features[20]. Consisting of circuit blocks, PUFs are embedded in an integrated circuit (IC). The intrinsic randomness is often inherent and is generated by the fabrication procedure, which is normally uncontrollable even to manufacturers[2]. There are a number of ways to implement PUFs, including SRAM-generated random numbers, ring-oscillator PUFs, and arbiter-based PUFs[14,27,32], as shown in Fig. 3. These three examples are representative of a large design space, including many more PUF circuit designs. Each of these PUFs gets its random numbers from the random behavior which originates from the silicon manufacturing variability.

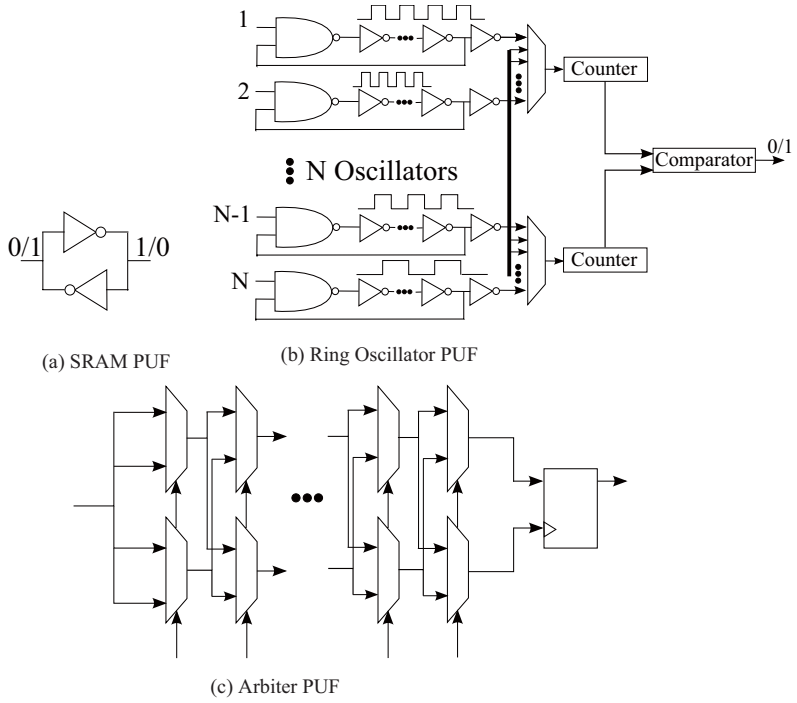


Figure 3. Three implementations of a physical unclonable functions. Variation in the circuits from fabrication and the unique environment create pseudo-random functions.

While the robustness of PUFs-based security solution is widely recognized, users are concerned more with its inflexibility. The intrinsic uniqueness also makes it infeasible to revise or revoke the signatures or master keys generated using PUFs.

Once the intrinsic pattern has been compromised by attackers, the only option for device owner is to replace the chip.

The dark silicon paradigm is very promising to eliminate that concern. The abundance of transistors enables multiple unique PUFs existing in a single chip. The redundancy allows more flexible management of the life cycle of PUFs as they do not always have to be on. By allowing the patterns to retire as they are "worn-out," the victims will disappear from the attackers' purview. That also implies each individual device can possess multiple unique master keys or signatures for root of trust, which can remain dark when not in use.

While excess silicon in chips can be re-purposed to PUFs for secret key generation, there are things to take into account. PUFs are susceptible to temperature and silicon wear which will affect the delay characteristics of the PUF. If a system designer were to implement a PUF where key generation can be replicated, such variables need to be taken into account. PUFs themselves are also susceptible to attacks. Such attacks as outlined by Ruhrmair et al.[27] should be accounted for. PUFs can also be used to create a hardware-implemented, unclonable time stamp, that would not only guarantee what chip the data had been created from, but also what time-frame it was generated.

In the security coprocessor system, there are the general-purpose cores, which would be similar to cores on conventional multicore systems. There would also be an array of PUFs (regardless of type), which would be used to generate the PUFs. Lastly, there is the controller for the PUF array, which would implement the logic to extract the PUF and control the re-write of the array.

The PUF array is separated into quadrants that can be individually configured by the PUF controller. This array could fill any unused portions of the chip with minimal impact on power. All of the PUFs contained in this array would be off whenever it was not being used, which would be the vast majority of the time.

The PUF array is logically isolated from the rest of the machine, and can only be activated and accessed through the PUF controller through a very specific pair of ports: challenge and response. These ports are typical for PUF systems. Data that are to be processed by the PUF are sent into the challenge port, and the results from the PUF are returned through the response port. The PUF controller also has ports to and from the array, including the data ports to extract the PUF, and the ports controlling the rewrites of the PUF. The controller rewrites the PUF by selecting quadrants in the array to destroy, which should irreversibly change the PUF. The PUF controller includes logic that extracts the PUF and creates the response from the PUF and the challenge. Next, there is a time and date tracker that determines when to perform a scheduled rewrite. An independent time and date tracker is integrated instead of using the system's time and date as a precaution against a malicious system preventing updates or causing too many updates for some reason. Last is a pseudo-random number generator, which is used to determine which quadrant(s) to discard in the array.

There are some additional options. One is to create a hybrid permanent/ rewritable PUF. This would dedicate part of the array to be permanent, so that it would never be rewritten. It is used to generate a PUF that would unchangingly represent the system hardware. There are also parts of the array that are dedicated

as changeable, which would be rewritten regularly, and would be used to represent the time-frame for time stamping. This design would have the advantages of both a permanent PUF and a changeable PUF.

**On-going and future work:** One area that is currently being investigated is to guarantee all modifications to the PUF are irreversible. Another area is investigating whether or not the PUF controller's time and date could be fooled, and what vulnerabilities exist if it was fooled. Also, discussion in this section assumed a physically secure system. Further investigations must be done to see how it could be exploited in a physically insecure system, and what measures should be taken to reduce any vulnerabilities. The exact method of updating the challenge-response pairing database that stores the PUFs needs to be determined. Lastly, the fabrication effects of each PUF design must be studied for each manufacturing node to determine randomness, lifetime, used area, and power consumption, all of which will inform the designer on which PUF is ideal for individual systems.

### *3.4 Specialized security co-processor & computational spring*

A popular direction in dark silicon is that of specialization[34,35,40]. By providing specialized hardware, specific algorithms can be computed more efficiently. The same principle can be applied to dark silicon. Security co-processors can be provided to improve the speed and efficiency of cryptographic and intrusion detection algorithms. General-purpose cores would selectively turn on the specialized hardware and off-load computation. Because these components are only on for specific times and functions, they can better utilize the otherwise homogeneous dark silicon.

This specialization can take various forms, as shown in Fig. 4. First, each core/tile of a chip multiprocessor can include small, private specialized security hardware for lightweight security applications. Each processor can off-load computation to these special function coprocessors. Another possibility is to provide large shared co-processors. Finally, as we will describe here, we can have a combination of the shared and private, where the shared coprocessors are higher frequency and are significantly more powerful.

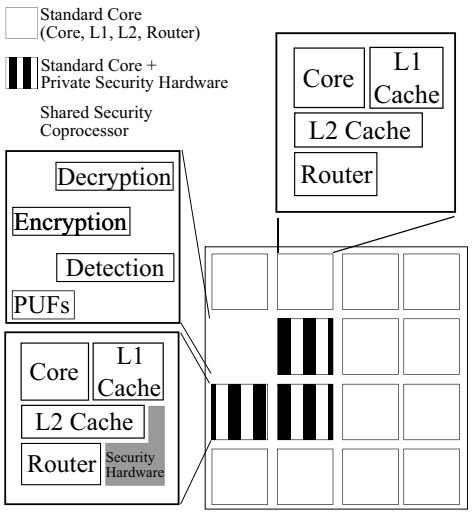

Figure 4. A specialized security CMP would contain standard cores, cores enhanced with light-weight security provisions, and specialized, high-performance security cores.

A coprocessor-based system relies on three types of cores: 1) a standard core, much like those found in conventional multicore systems; 2) a standard core enhanced with small-scale security hardware; and 3) a faster, more powerful core that is optimized for operation at a higher frequency. When this faster core is not in use, it would be power-gated to save energy/power. This design has multiple standard cores and one high-frequency core. The high-frequency core would only be used in specific situations and for short periods of time.

Some of this design is motivated by the needs of programs that are too sensitive to run simultaneously with other programs. Therefore, there must be times when the system is locked down so that the secure program can run. If that secure program has parts that cannot be run in parallel, then the most efficient solution is to complete that part of the program as fast as possible. Since it cannot take advantage of multiple cores, the only other option to speed up its execution is to use a higher frequency core.

There are three modes of operation: normal multicore, secure multicore, and secure single-core. Normal multicore operation is similar to how normal multicore systems work: multiple programs timesharing multiple cores. The enhanced standard cores in this case could be used for limited secure processing purposes. Secure multicore mode locks down the system to ensure only the secure program has access to its resources and information. Lastly, there is secure single-core, in which all off the normal cores are shut off, and the secure program uses the high-frequency core. As many of these resources would be powered-off if unneeded, it is well suited for the dark silicon environment, where these specialized coprocessors do not take up otherwise useful resources.

Additionally, a resource manager must be able to kick out a misbehaving program so that it does not control the resources for too long. This resource manager would not be unique to the coprocessor design, as any core with a rogue program should be removed from the system.

Furthermore, in a many-core dark silicon chip multiprocessor, it is necessary to understand the real-time power consumption of each component and the thermal implications of its use, in order to properly schedule threads on the required hardware. Computational sprinting[22,39] shows that high-power cores can operate high priority, time-sensitive operations/threads for a short period of time. Even though the components heat up, brief and intermittent use of the secure co-processor will have minimal effect on the overall temperature. Further studies will explore the impact of security hardware on the overall power consumption of the entire many-core, dark silicon environment.

#### 3.4.1 Near- and sub-threshold coprocessors

Dim silicon uses lower voltage and frequency cores to lower the power consumption while better using available resources and increasing throughput. Threads that are not timing-critical can be placed onto these near- and sub-threshold cores to save energy. Near- and sub-threshold computing are common

techniques for saving energy, usually at the cost of performance[3,15,24,25,38]. Specific security functions can be identified as low-priority for timing and run on these low-voltage coprocessor cores to have less impact on the overall system, allowing more computation to be completed simultaneously.

The dim silicon can benefit applications with limited power budget and a relaxed constraint on "real-time." For example, in self-powered wireless sensor network-based environment monitoring[4,36], the requirement in delay is a low priority, which can be at the level of minutes or even hours. Instead, they are expected to record the data with minimal data miss rate. The near- and sub- threshold coprocessors are ideal candidates as they are able to maintain power levels and throughput and tolerate the fluctuations in energy harvesting.

**On-going and future work:** The key decisions pertaining to security coprocessor-based design center on determining what functionality should be included. Preliminary systems include cryptographic functions and intrusion detection, but those are certainly not the only security mechanisms necessary to create a secure system. As new components are proposed for the coprocessor design, each will be designed at the logic level to determine power, area, and performance, each of which will need to be weighed at run-time to maintain correct functionality within the power and area constraints.

## 4 Summary

The development of billion-transistor chips has created dark silicon, in which large portions of the chip are powered off to remain within the given power envelope. To adapt to the increased transistor count, increasing power density, and power budget, designers have begun to specialize hardware for particular functions, providing more efficient computation which is used selectively, maintaining manageable power consumption. Here we leverage the available silicon to integrate hardware security mechanisms into a single chip.

On-chip moving target defense mechanisms provide a chip multiprocessor environment which can protect multiple users simultaneously co-habitating the chip. These mechanisms move threads among the many available, previously powered-off cores to avoid intrusion or denial-of-service attacks and hide victim operations from potential attackers.

The available logic can also be used to implement security-specific hardware. Large arrays of physical unclonable functions provide unique keys for encryption and authentication. Co-processors can be integrated for security functions. These co-processors are not powered unless in use, keeping the thermal and power impact minimal, while providing fast and efficient security features.

Overall, the reallocation of otherwise dark silicon provides a powerful tool in the effort to create secure hardware for an increasingly dangerous digital landscape. To date, there are few reported efforts in this direction, and we hope this work inspires more attention to the field of dark silicon-releated security in the computer architecture community.

**Acknowledgement**

The authors would like to acknowledge Daniel Butash, Kenneth Donovan, Anthony Padua, and Kevin Lopes for their work during the initial phases of this project.

## References


[1] Anderson R, Bond M, Clulow J, Skorobogatov S. Cryptographic processors-a survey. Proc. of the IEEE, 2006, 94(2): 357–369.

[2] Armknecht F, Maes R, Sadeghi A, Standaert O, Wachsmann C. A formalization of the security features of physical functions. 2011 IEEE Symp. on Security and Privacy (SP). IEEE. 2011. 397–412.

[3] Calhoun B, Khanna S, Mann R, Wang J. Sub-threshold circuit design with shrinking CMOS devices. IEEE Int'l Symp. on Circuits and Systems, May 2009. 2541–2544.

[4] Chen Y, Twigg C, Sadik O, Tong S. A self-powered adaptive wireless sensor network for wastewater treatment plants. 2011 IEEE Int'l Conf. on Pervasive Computing and Communications Workshops (PERCOM Workshops). IEEE. 2011. 356–359.

[5] Cota E, Mantovani P, Petracca M, Casu M, Carloni L. Accelerator memory reuse in the dark silicon era. IEEE Computer Architecture Letters. 2012.

[6] Crouse M, Fulp E, Canas D. Improving the diversity defense of genetic algorithm-based moving target approaches. National Symp. on Moving Target Research. 2012.

[7] Dennard R, Gaensslen F, Yu H, Rideout V, Bassous E, LeBlanc A. Design of ion-implanted MOSFET's with very small physical dimensions. IEEE Journal of Solid-State Circuits, 1974, SC-9(5): 256–268.

[8] Dunlop M, Groat S, Marchany R, Trout J. Implementing an ipv6 moving target defense on a live network. National Symp. on Moving Target Research. Jun, 2012.

[9] Dyer J, Lindemann M, Perez R, Sailer R, Van Doorn L, Smith S. Building the ibm 4758 secure coprocessor. Computer, 2001, 34(10): 57–66.

[10] Esmaeilzadeh H, Blem E, St. Amant R, Sankaralingam K, Burger D. Dark silicon and the end of multicore scaling. Int'l Symp. on Computer Architecture. Jun, 2011. 265–376.

[11] Frederick E, Rowe N, Wong A. Testing deception tactics in response to cyperattacks. National Symp. on Moving Target Research. June 2012.

[12] Goulding-Hotta N, Sampson J, Zheng Q, Bhatt V, Auricchio J, Swanson S, Taylor M. GreenDroid: An architecture for the dark silicon age. Asia and South Pacific Design Automation Conf. Jan, 2012. 100–105.

[13] TCG Group et al. Tcg specification architecture overview. TCG Specification Revision, 2007, 1: 1–24.

[14] Guajardo J, Kumar S, Schrijen G, Tuyls P. Fpga intrinsic pufs and their use for ip protection. Cryptographic Hardware and Embedded Systems-CHES 2007. Springer. 2007. 63–80.

[15] Hanson S, Zhai B, Seok M, Cline B, Zhou K, Singhal M, Minuth M, Olson J, Nazhandali L, Austin T, Sylvester D, Blaauw D. Performance and variability optimization strategies in a sub-200mV, 3.5pJ/inst, 11nW subthreshold processor. Proc. of the IEEE Symp. on VLSI Circuits. June 2007. pages 152–153.

[16] Huffmire T, Brotherton B, Wang G, Sherwood T, Kastner R, Levin T, Nguyen T, Irving C. Moats and drawbridges: An isolation primitive for reconfigurable hardware based systems. IEEE Symp. on Security and Privacy. 2007. 281–295.

[17] Karpuzcu U, Greskamp B, Torrellas J. The bubblewrap many-core: Popping cores for sequential acceleration. Int'l Symp. on Microarchitecture. Dec. 2009. 447–458.

[18] Lie D, Thekkath C, Horowitz M. Implementing an untrusted operating system on trusted hardware. ACM SIGOPS Operating Systems Review. ACM. 2003, 37. 178–192.

[19] Lie D, Thekkath C, Mitchell M, Lincoln P, Boneh D, Mitchell J, Horowitz M. Architectural support for copy and tamper resistant software. ACM SIGPLAN Notices, 2000, 35(11): 168–177.

[20] Maes R, Verbauwhede I. Physically unclonable functions: A study on the state of the art and future research directions. Towards Hardware-Intrinsic Security. Springer. 2010. 3–37.

[21] Patsilaras G, Choudhary N, Tuck J. Efficiently exploiting memory level parallelism on asymmetric coupled cores in the dark silicon era. ACM Trans. on Architecture and Code Optimization, Jan. 2012, 8(4).

[22] Raghavan A, Luo Y, Chandawalla A, Papaefthymiou M, Pipe K, Wenisch T, Martin M.

Computational sprinting. Int'l Symp. on High Performance Computer Architecture. Feb. 2012. 1–12.

[23] Ravi S, Raghunathan A,. Kocher P, Hattangady S. Security in embedded systems: Design challenges. ACM Trans. on Embedded Computing Systems (TECS), 2004, 3(3): 461–491.

[24] Raychowdhury A, Paul B, Bhunia S, Roy K. Ultra low power computing with sub-threshold leakage: A comparative study of bulk and SOI technologies. Proc. of the Conf. on Design, Automation, and Test in Europe. 2006. 856–861.

[25] Raychowdhury A, Chandra Paul B, Bhunia S, Roy K. Computing with subthreshold leakage: device/circuit/architecture co-design for ultralow-power subthreshold operation. IEEE Trans. VLSI Syst., 2005, 13(11): 1213–1224.

[26] Rowe J, Levitt K, Demir T, Erbacher R. Artificial diversity as maneuvers in a control theoretic moving target defense. National Symp. on Moving Target Research. June 2012.

[27] Ruhrmair U, Sehnke F, Solter J, Dror G, Devadas S, Schmidhuber J. Modeling attacks on phyical unclonable functions. ACM Conf. on Computer and Communications Security. Oct. 2010. 237–249.

[28] Smith S. Secure Coprocessing Applications and Research Issues, 1996.

[29] Smith S, Safford D. Practical server privacy with secure coprocessors. IBM Systems Journal, 2001, 40(3): 683–695.

[30] Smith S, Weingart S. Building a high-performance, programmable secure coprocessor. Computer Networks, 1999, 31(8): 831–860.

[31] Suh G, Clarke D, Gassend B, Van Dijk M, Devadas S. Aegis: architecture for tamper-evident and tamper-resistant processing. Proc. of the Annual Int'l Conf. on Supercomputing. ACM. 2003. 160–171.

[32] Suh G, Devadas S. Physical unclonable functions for device authentication and secret key generation. Design Automation Conf. Jun. 2007. 9–14.

[33] Swaminathan K, Kultursay E, Saripalli V, Narayanan V, Kandemir M, Datta S. Steep-slope devices: From dark to dim silicon. IEEE Micro, Sept.-Oct. 2013, 33(5): 50–59.

[34] Taylor M. Is dark silicon useful?: Harnessing the four horsemen of the coming dark silicon apocalypse. Design Automation Conf. 2012. 1131–1136.

[35] Taylor M. A landscape of the new dark silicon design regime. IEEE Micro, Sept.-Oct. 2013, 33(5): 8–19.

[36] Tong S, Tong M, Twigg C, Chen Y, Sadik O. Energy efficient control for smart wastewater treatment plants using self-powered wireless sensor networks. *Sensor Letters*, 2013, 11(9): 1689–1694.

[37] Venkatesh G, Sampson J, Goulding-Hotta N, Venkata S, Taylor M, Swanson S. QsCores: Trading dark silcon for scalable energy efficiency with quasi-specific cores. Int'l Symp. on Microarchitecture. Dec. 2011.

[38] Wang L, Skadron K. Dark vs. dim silicon and near-threshold computing. Technical Report UVA-CS-13–01. 2013.

[39] Zhan J, Xie Y, Sun G. NoC-sprinting: Interconnect for fine-grained sprinting in the dark silicon era. Design Automation Conf. 2014. 1–6.

[40] Zheng Q, Goulding-Hotta N, Ricketts S, Swanson S, Taylor M, Sampson J. Exploring energy scalability in coprocessor-dominated architectures for dark silicon. ACM Trans. on Embedded Computing Systems, Apr. 2014, 13(4s).

[41] Zhuang R, Zhang S, DeLoach S, Ou X, Singhal A. Simulation-based approaches to studying effectiveness of moving-target network defense. National Symp. on Moving Target Research. June 2012.