Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

# Vigour and Vitality A 25-core manycore open source processor's specifications

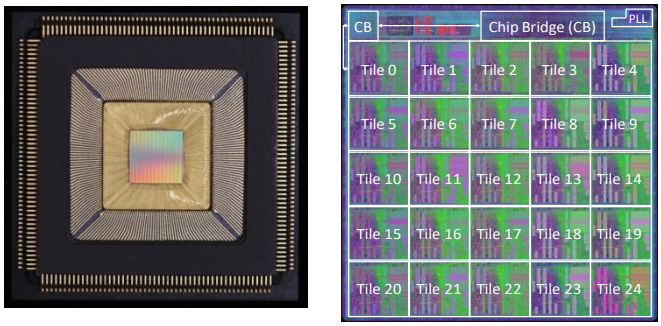
*Dr.Sachinandan Mohanty[1]*, Ms. Sushree Sangeeta Chinda[2]*
*[1]* Professor Dept. Of Computer Science and Engineering, NIT , BBSR*
*[2]Assistant Professor,Dept. Of Computer Science and Engineering, NIT , BBSR*
*sachinandanmohanty@thenalanda.com* susrisangita@thenalanda.Com*

***Abstract*— Power and energy efficiency are the main design objectives for contemporary processors as a result of Dennard's scaling's end and the impending power wall. Additionally, new applications like cloud computing and the Internet of Things (IoT) still demand higher performance and energy efficiency. Manycore CPUs have promise for resolving some of these problems. For manycore processors, there is, however, a paucity of precise power and energy statistics. We carefully examine Piton's 25-core modern open source academic processor's detailed power and energy characteristics in this work, including voltage versus frequency scaling, energy per instruction (EPI), memory system energy, network-on-chip (NoC) energy, thermal characteristics, and application performance and power consumption.**

**an open source manycore architecture that was put into silicon. The processor's open source nature adds value by providing thorough simulation-verified characterization and the ability to link outcomes with the design and register transfer level (RTL) model. This makes it possible for other researchers to construct new power models, come up with fresh research ideas, and conduct precise power and energy research utilising the open source processor using the results of this work. The parameterization data reveals several intriguing insights, such as the fact that floating - point values have a significant impact on EPI, that recomputing relevant information can be more energy efficient than loading it from memory, that on-chip data transmission (NoC) energy is low, and insights on energy efficient multithreaded core design. At http://www.openpiton.org, you may download the hardware infrastructure used and all of the data that was gathered. Processor, manycore, power, energy, thermal, and characterisation are some related keywords.**

## INTRODUCTION

Power and energy have become increasingly important metrics in designing modern processors. The power savings resulting from newer process technologies have diminished due to increased leakage and the end of Dennard's scaling [1]. Transistor power dissipation no longer scales with channel length, leading to higher energy densities in modern chips. Thus, economically cooling processors has become a challenge. This has led researchers to a number of possible solutions, including Dark Silicon [2]–[4]. Due to increased power density, design decisions have become increasingly motivated by power as opposed to performance. Energy efficiency has also become a major research focus [5]–[11]. Moreover, emerging applications continue to demand more energy efficient compute. Cloud computing and datacenters, where power is a first class citizen with direct impacton total cost of ownership (TCO) [14], [15], are growing more pervasive. On the other end of the spectrum, mobile and Internet of Things (IoT) devices demand higher performance in a relatively constant power and energy budget.




(a) (b)

Figure 1. Piton die, wirebonds, and package without epoxy encapsulation (a) and annotated CAD tool layout screenshot (b). Figure credit: [12], [13].

Manycore processors have shown promise in curbing energy efficiency issues. Single-threaded performance scaling has come to a halt with the end of clock frequency scaling largely due to the power wall. Manycore processors provide an alternative, enabling more efficient computation for parallelizeable applications through the use of many simple cores. Intel's Xeon Phi processor [16], Cavium's ThunderX processor [17], Phytium Technology's Mars processor [18], Qualcomm's Centriq 2400 processor [19], and the SW26010 processor in the Sunway TaihuLight supercomputer [20] are

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

Table I. PITON PARAMETER SUMMARY

| | | | |
|---|---|---|---|
| Process | IBM 32nm SOI | L1 Instruction Cache Size | 16KB |
| Transistor Count | > 460 million | L1 Instruction Cache Line Size | 32B |
| Package | 208-pin QFP (Kyocera CERQUAD®) | L1 Data Cache Size | 8KB |
| | | L1 Data Cache Associativity | 4-way |
| Nominal Core Volt. (VDD) | 1.0V | L1 Data Cache Line Size | 16B |
| Nominal SRAM Volt. (VCS) | 1.05V | L1.5 Data Cache Size | 8KB |
| Nominal I/O Volt. (VIO) | 1.8V | L1.5 Data Cache Associativity | 4-way |
| Off-chip Interface Width | 32-bit (each direction) | L1.5 Data Cache Line Size | 16B |
| Tile Count | 25 (5x5) | L2 Cache Slice Size | 64KB |
| NoC Count | 3 | L2 Cache Associativity | 4-way |
| NoC Width | 64-bit (each direction) | L2 Cache Line Size | 64B |
| Cores per Tile | 1 | L2 Cache Size per Chip | 1.6MB |
| Threads per Core | 2 | Coherence Protocol | Directory-based MESI |
| Total Thread Count | 50 | | |
| Core ISA | SPARC V9 | Coherence Point | L2 Cache |
| Core Pipeline Depth | 6 stages | | |

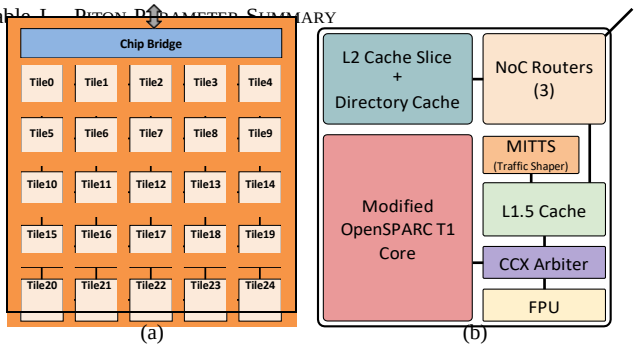



Figure 2. Piton chip-level architecture diagram (a) and tile-level diagram (b). Figure credit: [12], [13].

In this paper, we perform detailed power characterization of Piton [12], [13]. Piton is a 25-core manycore research processor, shown in Figure 1. It utilizes a tile-based design with a 5x5 2D mesh topology interconnected with three 64-bit networks-on-chip (NoCs). Cache coherence is maintained at the shared, distributed L2 cache and the NoCs along with the coherence protocol extend off-chip to support inter-chip shared memory in multi-socket systems. Piton leverages the multithreaded OpenSPARC T1 core [27]. Piton was taped-out on IBM's 32nm silicon-on-insulator (SOI) process, and fits in a 36mm$^2$ die with over 460 million transistors.

The characterization reveals a number of insights. We show energy per instruction (EPI) is highly dependent on operand value and that recomputing data can be more energy efficient than loading it from memory. Our NoC energy results contradict a popular belief that NoCs are a dominant fraction of a manycore's power [28]–[32]. This matches results from other real system characterizations [33]–[35] and motivates a reassessment of existing NoC power models [29], [36]. Additionally, our study of the energy efficiency of multithreading versus multicore provides some design guidelines for multithreaded core design.

Piton was open sourced as OpenPiton [37], including the RTL, simulation infrastructure, validation suite, FPGA synthesis scripts, and ASIC synthesis and back-end scripts. Thus, all design details and the RTL model are publicly available, enabling meaningful, detailed power characterization and correlation with the design. Most projects taped-out in silicon do not publicly release the source [38], [39], and those that do [27], [40], [41] have not published detailed silicon data for the architecture.

The power characterization of an open source manycore processor provides several benefits, including insights for future research directions in improving energy efficiency and power density in manycore processors. It also enables researchers to build detailed and accurate power models for an openly accessible design. This is particularly useful when researchers use the open design in their research, as the results directly apply. All data collected and all hardware infrastructure is open source for use by the research community and is available at http://www.openpiton.org.

The contributions of this work are as follows:

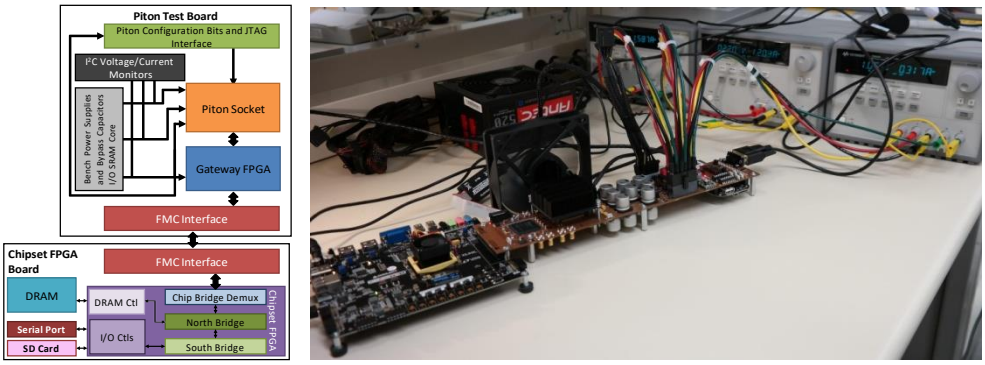
- The first detailed power characterization of an open source manycore processor taped-out in silicon, including characterization of voltage versus frequency scaling, EPI, memory system energy, NoC energy, thermal properties, and application performance and power.
- To the best of our knowledge, the most detailed area breakdown of an open source manycore.
- A number of insights derived from the characterization, such as the impact of operand values on EPI.
- An open source printed circuit board (PCB) for manycore processors with power characterization features.
- Open source power characterization data enabling researchers to build intuition, formulate research directions, and derive power models.

## I. PITON ARCHITECTURE BACKGROUND

Piton [12], [13] is a 25-core manycore academic-built research processor. It was taped-out on IBM's 32nm SOI process on a 36mm$^2$, 6mm x 6mm, die with over 460 million transistors. Piton has three supply voltages: core (VDD), SRAM arrays (VCS), and I/O (VIO). The nominal supply voltages are 1.0V, 1.05V, and 1.8V, respectively. The die has 331 pads positioned around the periphery and is packaged in a wire-bonded 208-pin ceramic quad flat package (QFP) with an epoxy encapsulation. The Piton die, wirebonds, and package, without the encapsulation, are shown in Figure 1a.

The high-level architecture of Piton and the implemented layout are shown in Figure 2a and Figure 1b, respectively. Piton contains 25 tiles arranged in a 5x5 2D mesh topology, interconnected by three physical 64-bit (in each direction) NoCs. The tiles maintain cache coherence, utilizing a directory-based MESI protocol implemented over the three

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022




(a) (b)

Figure 3. Piton experimental system block diagram (a) and photo (b). Figure (a) credit: [12], [13].

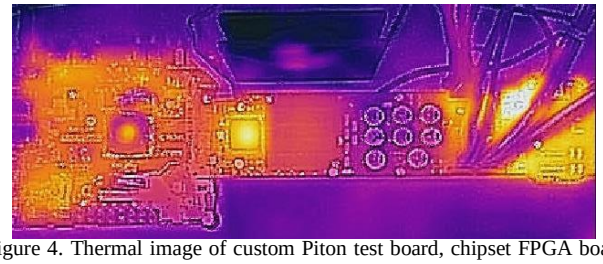


Figure 4. Thermal image of custom Piton test board, chipset FPGA board, and cooling solution running an application.

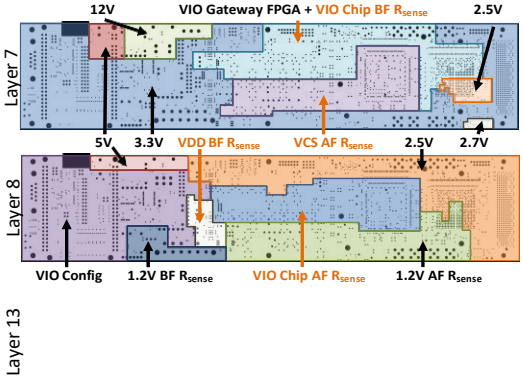



Figure 5. Split power plane layers of the Piton test board. Planes delivering power to Piton are labeled in orange.

shown in Figure 2b. The Piton core is a single-issue, 6-stage, in-order SPARC V9 core. It contains two-way fine-grained multithreading and implements Execution Drafting [5] for energy efficiency when executing similar code on the two threads. The use of a standard instruction set architecture (ISA) enables the leveraging of existing software (compilers, operating systems, etc.). The L2 cache slice contains an integrated directory cache for the MESI cache coherence protocol and implements Coherence Domain Restriction (CDR) [42], enabling shared memory among arbitrary cores in large core-count multi-socket systems. The L2 cache in aggregate provides 1.6MB of cache per chip. The three NoC routers implement dimension-ordered, wormhole routing for the three physical networks with a one-cycle-per-hop latency and an additional cycle for turns.

The L1.5 cache is an 8KB write-back private data cache that encapsulates the core's write-through L1 data cache to reduce the bandwidth requirement to the distributed L2 cache. It also transduces between the core interface (CCX) and the Piton NoC protocol. Last, a memory traffic shaper, known as the Memory Inter-arrival Time Traffic Shaper (MITTS) [43], fits memory traffic from the core into a particular inter-arrival time distribution to facilitate memory bandwidth sharing in multi-tenant systems. The architectural parameters for Piton are summarized in Table I.

## II. Experimental Setup

Figu the ex-
perimen sists of
three m circuit
board (PCB), a chipset FPGA board, and a cooling solution. Figure 4 shows a thermal image of all three components.

### A. Piton Test PCB

A custom PCB was designed to test Piton. The board design is derived from the BaseJump Double Troubleboard [44]. The Piton test board contains a 208-pin QFP socket to hold the Piton chip. The Piton chip bridge interface signals connect to a Xilinx Spartan-6 FPGA (XC6SLX150-3FGG676C), the gateway FPGA. The gateway FPGA acts as a simple passthrough for the chip bridge interface, convert-ing single-ended signals to and from Piton into differential signals for transmission over a FPGA Mezzanine Connector

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

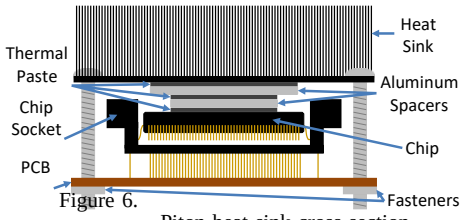
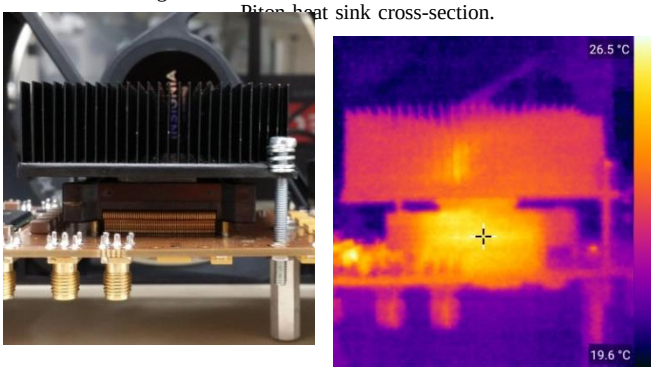



Figure 6. Piton heat sink cross-section.




(a) (b)

Figure 7. Piton heat sink photo (a) and thermal image (b).

on the Piton board has remote voltage sense (when powered from bench power supplies, remote sense is used). For these reasons, the bench power supplies were used for all studies.

Three layers of the 14-layer PCB are dedicated to split power planes, shown in Figure 5. The planes delivering power to Piton are labeled in orange. Sense resistors bridge split planes delivering current to the three chip power rails. Care was taken to ensure that only the current delivered to Piton is measured by the sense resistors.

Voltage monitors controlled via $I^2C$ track the voltage at the chip socket pins and on either side of sense resistors. All reported measurements are taken from the on-board voltage monitors. The monitors are polled at approximately 17Hz, a limitation of the monitor devices and the processing speed of the host device. Unless otherwise specified, all experiments in this work record 128 voltage and current samples (about a 7.5 second time window) after the system reaches a steady state. We report the average power calculated from the 128 samples. Unless otherwise specified, error is reported as the standard deviation of the samples from the average. Note, the recorded voltages are measured at the socket pins and do not account for current induced voltage drop (IR drop) across the socket, wirebonds, or die. The board design is open source for other researchers to leverage when building systems. It can serve as a good reference or starting point. The board files are available at http://www.openpiton.org.

Table II. Experimental System Frequencies

| | |
|---|---|
| Gateway FPGA ↔ Piton | 180 MHz |
| Gateway FPGA ↔ FMC ↔ Chipset FPGA | 180 MHz |
| Chipset FPGA Logic | 280 MHz |
| DRAM DDR3 PHY | 800 MHz (1600 MT/s) |
| DDR3 DRAM Controller | 200 MHz |
| SD Card SPI | 20 MHz |
| UART Serial Port | 115,200 bps |

Table III. Default Piton Measurement Parameters

| | |
|---|---|
| Core Voltage (VDD) | 1.00V |
| SRAM Voltage (VCS) | 1.05V |
| I/O Voltage (VIO) | 1.80V |
| Core Clock Frequency | 500.05MHz |

### B. Chipset FPGA Board

The chipset FPGA board, a Digilent Genesys2 board with a Xilinx Kintex-7 FPGA, connects to the Piton test board via a FMC connector. The FPGA logic implements a chip bridge demultiplexer which converts the 32-bit logical channel interface back into three 64-bit physical networks, a north bridge, and south bridge. The north bridge connects to a DDR3 DRAM controller implemented in the FPGA, while the south bridge connects to I/O controllers for various devices, including an SD card, network card, and a serial port. The chipset FPGA board also includes 1GB of DDR3 DRAM with a 32-bit data interface and the I/O devices.

### C. Cooling Solution

The Piton test system uses a cooling solution consisting of a heat sink and fan. Figure 6 shows a cross-sectional diagram of the heat sink setup and Figure 7 shows a photo and thermal image. A stock heat sink is used with aluminum spacers to fill the gap between the top of the chip and the top of the socket walls. Thermal paste is used at each thermal interface between the top of the chip and the heat sink. A PC case fan with airflow volume of 44cfm is used to circulate hot air out of the system and introduce cool air. The fan direction is parallel to the heat sink fins (out of the plane in Figure 6). Note, the thermal capacity of this cooling solution was purposely over-engineered and is likely excessive.

### D. System Summary

This system allows us to boot full-stack Debian Linux from an image on the SD card (also hosts the file system) or load assembly tests over the serial port into DRAM. The system is used as described for all studies in this work, unless otherwise specified. The frequencies at which interfaces operate are listed in Table II.

## III. Results

This section presents the results from careful characterization of Piton. Unless otherwise stated, all results are taken at

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

Table IV. PITON TESTING STATISTICS

| Status | Symptom | Possible Cause | Chip Count | Chip Percentage |
|---|---|---|---|---|
| Good | Stable operation | N/A | 19 | 59.4 |
| Unstable* | Consistently fails deterministically | Bad SRAM cells | 7 | 21.9 |
| Bad | High VCS current draw | Short | 4 | 12.5 |
| Bad | High VDD current draw | Short | 1 | 3.1 |
| Unstable* | Consistenly fails nondeterministically | Unstable SRAM cells | 1 | 3.1 |

* Possibly fixable with SRAM repair

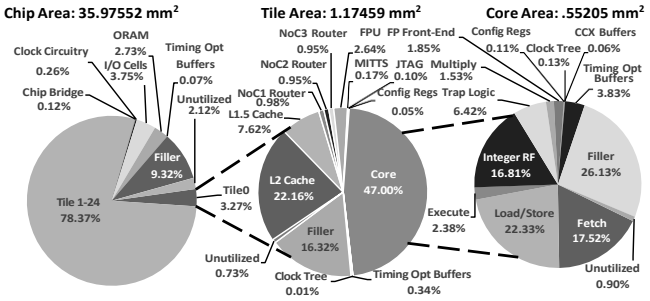



Figure 8. Detailed area breakdown of Piton at chip, tile, and core levels.

repair flow is still in development. 15.6% of the chips have abnormally high current draw on VCS or VDD, indicating a possible short circuit. These results provide understanding on what yield looks like in an academic setting, but the number of wafers and tested die are small thus it is difficult to draw strong conclusions. During our characterization, only fully-working, stable chips are used.

### A. Area Breakdown

Figure 8 shows a detailed area breakdown of Piton at the chip, tile, and core levels. These results were calculated directly from the place and route tool. The area of standard cells and SRAM macros corresponding to major blocks in the design were summed together to determine the area. Filler cells, clock tree buffers, and timing optimization buffers are categorized separately, as it is difficult to correlate exactly which design block they belong to. The total area of the core, tile, and chip are calculated based on the floorplanned dimensions and unutilized area is reported as the difference of the floorplanned area and the sum of the area of cells within them.

The area data is provided to give context to the power and energy characterization to follow. The characterization is of course predicated on the design and thus does not represent all designs. Thus, it is useful to note the relative size of blocks in order to provide context within the greater chip design space. For example, stating the NoC energy is small is only useful given the relative size of the NoC routers and the tile (indicative of how far NoCs need to route). Designs with larger NoC routers or larger tiles (longer routing distance) may not have identical characteristics. This

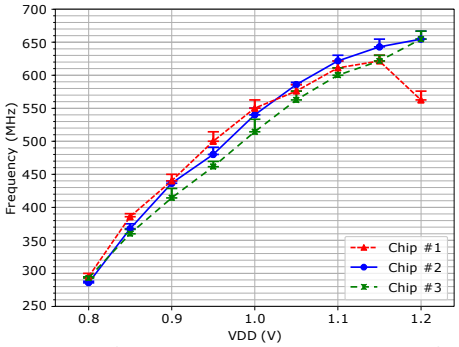



Figure 9. Maximum frequency at which Linux successfully boots at different voltages for three different Piton chips. $VCS = VDD + 0.05V$ for all VDD values. Error bars indicate noise due to quantization.

$VCS = VDD + 0.05V$. Since the PLL reference clock is discussed more in Section IV-K.

### B. Voltage Versus Frequency

Figure 9 shows the maximum frequency that Debian Linux successfully boots on Piton at different VDD voltages for three different Piton chips. For all values of VDD,

frequency that the gateway FPGA drives into the chip is discretized, the resulting core clock frequency has quantiza- tion error. This is represented by the error bars in the figure indicating the next discrete frequency step that the chip was tested at and failed. However, the chip may be functional at frequencies between the plotted value and the next step.

The maximum frequency at the high-end of the voltagespectrum is thermally limited. This is evident from thedecrease in maximum frequency at higher voltages for Chip#1. Chip #1 consumes more power than other chips andtherefore runs hotter. At lower voltages it actually hasthe highest maximum frequency of the three chips as theheat generated is easily dissipated by the cooling solution.However, after 1.0V it starts to drop below the other chipsuntil 1.2V where the maximum frequency severely drops.This is because the chip approaches the maximum amountof heat the cooling solution can dissipate, thus requiring adecrease in frequency to reduce the power and temperature.We believe the thermal limitation is largely due to packag- ing restrictions, including the die being packaged cavity up, the insulating properties of the epoxy encapsulation, and thechip being housed in a socket. This results in a higher dietemperature than expected, reducing the speed at which thechip can operate at greater voltages and frequencies. This isnot an issue of the cooling solution thermal capacity, but anissue of the rate of thermal transfer between the die and theheat sink. Results for the default measurement parameters in Table III are measured at non-thermally limited points.

Another limitation of Piton's current packaging solution is the wire bonding. As previously stated, the voltages presented in this paper are measured at the socket pins. However, IR drop across the socket, pins, wirebonds, and die results in a decreased voltage at the pads and across the die. This reduces the speed at which transistors can switch. A flip-chip packaging solution would have allowed us to supply power to the center of the die, reducing IR drop issues, and for the die to be positioned cavity down, enabling more ideal heat sinking from the back of the die. However, flip-chip packaging is expensive and increases design risk.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

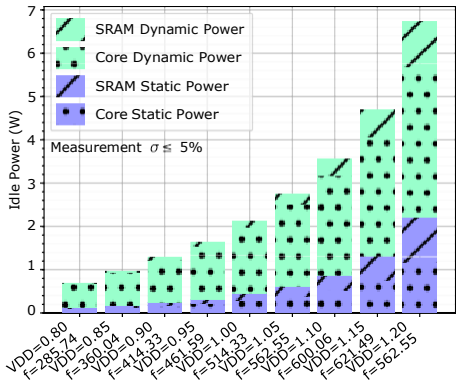



Figure 10. Static and idle power averaged across three different Piton chips for different voltages and frequency pairs. Separates contributions due to VDD and VCS supplies.

Table V. DEFAULT POWER PARAMETERS (CHIP #2)

| | |
|---|---|
| Static Power @ Room Temperature | 389.3±1.5 mW |
| Idle Power @ 500.05MHz | 2015.3±1.5 mW |

### C. Static and Idle Power

Figure 10 shows static and idle power at different voltages averaged across three Piton chips. The frequency was chosen as the minimum of the maximum frequencies for the three chips at the given voltage. The contribution from VDD and VCS supplies is indicated in the figure. Again, for all values of VDD, $VCS = VDD + 0.05V$. The static power was measured with all inputs, including clocks, grounded. The idle power was measured with all inputs grounded, but driving clocks and releasing resets. Thus, the idle power represents mostly clock tree power, with the exception of a small number of state machines and counters that run even when the chip is idle. The power follows an exponential relationship with voltage and frequency.

Chip #2 as labeled in Figure 9 will be used throughout the remainder of the results, unless otherwise stated. The static and idle power for Chip #2 at the default measurement parameters in Table III are listed in Table V.

### D. Energy Per Instruction (EPI)

In this section, we measure the energy per instruction (EPI) of different classes of instructions. This was accomplished by creating an assembly test for each instruction with the target instruction in an infinite loop unrolled by a factor of 20. We verified through simulation that the assembly test fits in the L1 caches of each core and no extraneous activity occurred, such as off-chip memory requests.

The assembly test was run on all 25 cores until a steady state average power was achieved, at which point it was recorded as $P_{inst}$, summing the contributions from VDD

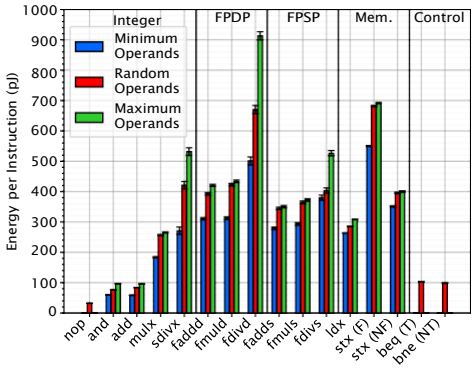



Figure 11. EPI for different classes of instructions with minimum, random, and maximum operand values for instructions with input operands.

Table VI. INSTRUCTION LATENCIES USED IN EPI CALCULATIONS

| Instruction | Latency (cycles) | Instruction | Latency (cycles) |
|---|---|---|---|
| Integer (64-bit) | | FP Single Precision | |
| nop | 1 | fadds | 22 |
| and | 1 | fmuls | 25 |
| add | 1 | fdivs | 50 |
| mulx | 11 | Memory (64-bit) L1/L1.5 Hit | |
| sdivx | 72 | ldx | 3 |
| FP Double Precision | | stx stb full | 10 |
| faddd | 22 | stx stb space | 10 |
| fmuld | 25 | Control | |
| fdivd | 79 | beq taken | 3 |
| | | bne nottaken | 3 |

from $P_{inst}$ to get the power contributed by the instruction and VCS. We measure the power while executing the test on 25 cores in order to average out inter-tile power variation. The latency in clock cycles, $L$, of the instruction was measured through simulation, ensuring pipeline stalls and instruction scheduling was as expected. In order to calculate EPI, we subtract the idle power presented in Table V, $P_{idle}$,

execution. Dividing this by the clock frequency results in the average energy per clock cycle for that instruction. The average energy per cycle is multiplied by $L$ to get the total energy to execute the instruction on 25 cores. Dividing this value by 25 gives us the EPI. The full EPI equation is:

$$EPI = \frac{1}{25} \times \frac{P_{inst} - P_{idle}}{Frequency} \times L$$

This EPI calculation is similar to that used in [24].

The store extended (64-bit) instruction, `stx`, requires special attention. Store instructions write into the eight entry store buffer in the core, allowing other instructions to bypass them while the updated value is written back to the cache. The latency of stores is 10 cycles. Thus, an unrolled infinite loop of `stx` instructions quickly fills the store buffer. The core speculatively issues stores assuming that the store buffer is not full. Since the core is multithreaded, it rolls-back and re-executes the store instruction and any subsequent instructions from that thread when it finds the store buffer is full. This roll-back mechanism consumes extra energy and pollutes our measurement result.

To accurately measure the `stx` EPI, we inserted nine `nop` instructions following each store in the unrolled loop. The `nop` instructions cover enough latency that the store buffer always has space. We subtract the energy of nine `nop` instructions from the calculated energy for the `stx` assembly test, resulting in the EPI for a single `stx` instruction. We present results using this method of ensuring the store buffer is not full (`stx` (NF)) and for the case the store buffer is full and a roll-back is incurred (`stx` (F)). This special case

Table VII. MEMORY SYSTEM ENERGY FOR DIFFERENT CACHE HIT/MISS SCENARIOS

| Cache Hit/Miss Scenario | Latency (cycles) | Mean LDX Energy (nJ) |
|---|---|---|
| L1 Hit | 3 | 0.28646±0.00089 |
| L1 Miss, Local L2 Hit | 34 | 1.54±0.25 |
| L1 Miss, Remote L2 Hit (4 hops) | 42 | 1.87±0.32 |
| L1 Miss, Remote L2 Hit (8 hops) | 52 | 1.97±0.39 |
| L1 Miss, Local L2 Miss | 424 | 308.7±3.3 |

highlights the importance of having access to the detailed microarchitecture and RTL when doing characterization.

Figure 11 shows the EPI results and Table VI lists latencies for each instruction characterized. We use 64-bit integer and memory instructions. In this study, `ldx` instructions all hit in the L1 cache and `stx` instructions all hit in the L1.5 cache (the L1 cache is write-through). Each of the 25 cores store to different L2 cache lines in the `stx` assembly test to avoid invoking cache coherence.

The longest latency instructions consume the most energy. The instruction source operand values affect the energy consumption. Thus, we plot data for minimum, maximum, and random operand values. This shows that the input operand values have a significant effect on the EPI. This data can be useful in improving the accuracy of power models.

Another useful insight for low power compiler developers and memoization researchers is the relationship between computation and memory accesses. For example, three `add` instructions can be executed with the same amount of energy and latency as a `ldx` that hits in the L1 cache. Thus, it can be more efficient to recompute a result than load it from the cache if it can be done using less than three `add` instructions.

*E. Memory System Energy*

Table VII shows the EPI for `ldx` instructions that access different levels of the cache hierarchy. The assembly tests used in this study are similar to those in Section IV-E. They consist of an unrolled infinite loop (unroll factor of 20) of `ldx` instructions, however consecutive loads access different addresses that alias to the same cache set in the L1 or L2 caches, depending on the hit/miss scenario. We control which L2 is accessed (local versus remote) by carefully selecting data addresses and modifying the line to L2 slice mapping, which is configurable to the low, middle, or high order address bits through software.

Latencies are verified through simulation for L1 and L2 hit scenarios. The L2 miss latency was profiled using performance counters since real memory access times are not reflected in simulation. We use an average latency for L2 miss, since memory access latency varies.

Similar to the caveats for the `stx` instruction discussed in Section IV-E, the core thread scheduler speculates that `ldx` instructions hit in the L1 cache. In the case the loadmisses, the core will roll-back subsequent instructions and stall the thread until the load returns from the memory system. This roll-back mechanism does not pollute the energy measurements for `ldx` instructions, as a roll-back will always occur for a load that misses in the L1 cache.

The energy to access a local L2 is noticeably larger than an L1 hit. This mostly comes from L2 access energy, however the request will first go to the L1.5 cache. The L1.5 cache is basically a replica of the L1 cache, but is write-back. Thus, a miss in the L1 will also miss in the L1.5. However, it is important to note that the energy to access the L1.5 is included in all results in Table VII except for L1 hit.

The difference between accessing a local L2 and remote L2 is relatively small, highlighting the minimal impact NoCs have on power consumption. We study this in more detail in Section IV-G. The energy for an L2 cache miss is dramatically larger than an L2 hit. This is a result of the additional latency to access memory causing the chip to stall and consume energy until the memory request returns. Note that this result does not include DRAM memory energy.

Comparing the results in Table VII to the energy required for instructions that perform computation in Figure 11, many computation instructions can be executed in the same energy budget required to load data from the L2 cache or off-chip memory. Similar to loading data from the L1 cache, it may be worthwhile to recompute data rather than load it from the L2 cache or main memory.

### *F. NoC Energy*

NoC energy is measured by modifying the chipset logic to continually send dummy packets into Piton destined for different cores depending on the hop count. We use an invalidation packet type that is received by the L1.5 cache. The dummy packet consists of a routing header flit followed by 6 payload flits. The payload flits reflect different bit switching patterns to study how the NoC activity factor affects energy. The center-to-center distance between tiles is 1.14452 mm in the X direction and 1.053 mm in the Y direction, indicative of the routing distance.

For a baseline, we measure the steady state average power when sending to tile0, $P_{base}$. We then measure the steady state average power when sending to different tiles based on the desired hop count, $P_{hop}$. For example, sending to tile1 represents one hop, tile2 represents two hops, and tile9 represents five hops. The measurement is taken for one to eight hops, the maximum hop count for a 5x5 mesh.

Due to the bandwidth mismatch between the chip bridge and NoCs, there are idle cycles between valid flits. However, the NoC traffic exhibits a repeating pattern which allows us to calculate the energy per flit (EPF). Through simulation, we verified that for every 47 cycles there are seven valid NoC flits. In order to calculate the EPF, we first calculate the average energy per cycle over the zero hop case by subtracting $P_{base}$ from $P_{hop}$ and dividing by the clock frequency. To account for idle cycles we multiply the average energy per cycle by 47 cycles to achieve the energy consumed for one repeating traffic pattern. Dividing by 7 valid flits results

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

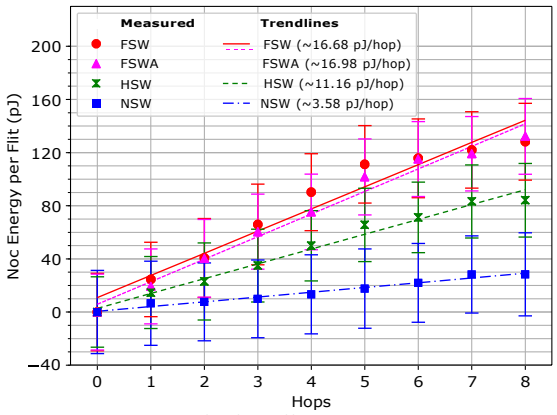



Figure 12. NoC energy per flit for different hop counts. Flit size is 64-bits. Results for different flit bit switching patterns are presented: no switching (NSW), half switching (HSW), and full switching (FSW and FSWA).

in the average EPF. The final EPF equation is:

$$EPF = \frac{47}{7} \times \frac{P_{hop} - P_{base}}{Frequency}$$

The EPF results for hop counts of zero to eight are shown in Figure 12 for different consecutive flit switching patterns. No switching (NSW) refers to all payload flit bits set to zero. Half switching (HSW) flips half of the bits between consecutive payload flits, i.e. consecutive flits alternate between 0x3333...3333 and all zeros. Full switching (FSW) flips all bits on consecutive flits, i.e. consecutive flits alternate between all ones and all zeros. Full switching alternate (FSWA) refers to alternating flits of 0xAAAA...AAAA and 0x5555...5555, which represents the effect of coupling and aggressor bits over FSW.

The standard deviation of measurement samples from the average for this experiment is relatively large, but there is a general trend from which we can draw conclusions. The energy to send a single flit scales linearly with the number of hops, with approximately 11.16 pJ/hop for the HSW case, as each additional hop is additive and all links are identical. Note that these results are for a single flit sent over one physical network in one direction. It is interesting to note the effect of different NoC switching patterns. The NoC routers consume a relatively small amount of energy (NSW case) in comparison to charging and discharging the NoC data lines. Comparing FSW and HSW, the energy scales roughly linearly with the NoC activity factor. The FSWA case consumes slightly more energy, but is within the measurement error. Thus, it is difficult to draw any conclusions on the affect of coupling and aggressor bits.

While data transmission consumes more energy than the in Figure 12 to the EPI data in Figure 11, sending a flit across the entire chip (8 hops) consumes a relatively small amount of energy, around the same as an `add` instruction. Other instructions consume substantially more energy. This shows that computation dominates the chip's power consumption, not on-chip data movement.

We can also calculate the NoC energy from the memory system energy results in Table VII by taking the difference of a local L2 cache hit and a remote L2 cache hit. A remote L2 cache hit results in a three flit request sent from the L1.5 cache to L2 cache and a three flit response. The memory system energy results indicate this consumes 330 pJ for four hops and 430 pJ for eight hops. This result is consistent with NoC router computation, our data contradicts a popular belief that on-chip data transmission is becoming a dominantportion of a manycore's power consumption [28]–[32]. Notethat our results match those from other real system charac- terizations [33]–[35] and motivate a reassessment of current NoC power models [29], [36]. Comparing the NoC EPF data

the EPF data for HSW (the memory system energy results are based on random data), 268 pJ for four hops and 536 pJ for eight hops. The difference can be explained by different NoC activity factors and measurement error.

*G. Microbenchmark Studies*

We developed a few microbenchmarks in order to study power scaling with core count and multithreading versus multicore power and energy efficiency. These microbench- marks include Integer (Int), High Power (HP), and His- togram (Hist). Int consists of a tight loop of integer instruc- tions which maximize switching activity. HP contains two distinct sets of threads. One set of threads performs integer computation in a tight loop, while the other set executes a loop consisting of a mix of loads, stores, and integer instructions with a ratio of 5:1 computation to memory. This application consumes about 3.5W when run on all 50 threads with each core executing one of each of the two different types of threads. Note, HP exhibits the highest power we have observed on Piton.

Hist is a parallel shared memory histogram computation implementation. Each thread computes a histogram over part of a shared array. Threads contend for a lock before updating the shared histogram buckets. We wrap the application in an infinite loop so it runs long enough to achieve a steady state power for measurement. Thus, each thread continually recomputes the same portion of the histogram.

Note, Hist differs from Int and HP in the way it scales withthread count. While HP and Int maintain work per thread,increasing the total amount of work as thread count scales,Hist keeps the total work constant (input array size) anddecreases the work per thread (per thread portion of array).Chip #3 as labeled in Figure 9 is used for all microbench- mark studies. Static power for this chip at room temperatureis 364.8 1.9 mW and idle power is ± ± 1906.2 2.0 mW for the default measurement parameters listed in Table III.

*1) Core Count Power Scaling:* In this section we study how power scales with core count. Each of the three mi- crobenchmarks are executed on one to 25 cores with both one thread per core (1 T/C) and two threads per core (2 T/C) configurations. HP has two distinct types of threads,

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

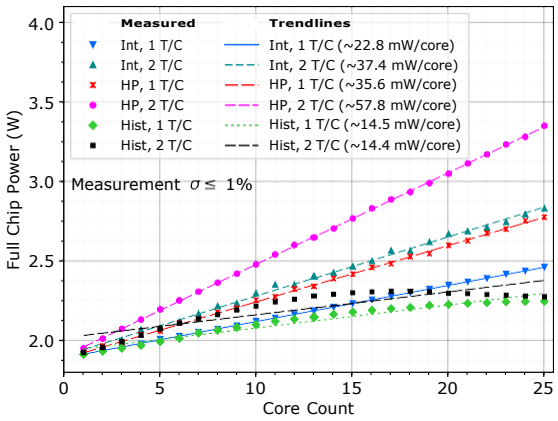


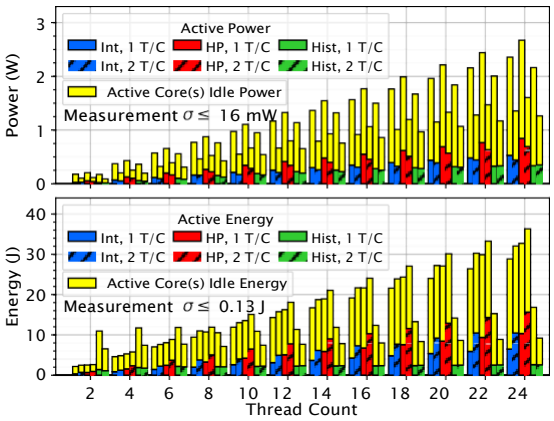

Figure 13. Power scaling with core count for the three microbenchmarks with 1 T/C and 2 T/C configurations.

so thread mapping must be taken into account. For 1 T/C, the two types of threads are executed on alternating cores. For 2 T/C, each core executes one thread from each of the two different types of threads. Figure 13 shows how the full chip power consumption scales from one to 25 cores for each application and T/C configuration. Power scales linearly with core count. Additionally, 2 T/C scales faster than 1 T/C since each core consumes more power.

Comparing across applications, Hist consumes the lowest power for both T/C configurations. This is because each thread performs both compute, memory, and control flow, causing the core to stall more frequently, and because of contention for the shared lock. This is in contrast to Int, where each thread always has work to do, and HP, where at least one of the threads always has work.

HP (High Power) consumes the most power for both T/C configurations since it exercises both the memory system and core logic due to the two distinct types of threads. This is especially true in the 2 T/C configuration since the core will always have work to do from the integer loop thread, even when the mixed thread (executes integer and memory instructions) is waiting on memory.

Hist has a unique trend where power begins to drop with increasing core counts beyond 17 cores for the 2 T/C configuration. This is a result of how Hist scales with thread counts, decreasing the work per thread. With large thread counts, the work per thread becomes small and the ratio of thread overhead to useful computation becomes larger. Additionally, threads spend less time computing their portion of the histogram and more time contending for locks to update buckets, which consumes less power due to spin waiting. This is exacerbated by increased thread counts since there is more contention for locks.




*2) Multithreading Versus Multicore:* To compare the power and energy of multithreading and multicore, we compare running the microbenchmarks with equal thread counts for both 1 T/C and 2 T/C configurations. For example, for a thread count of 16, each microbenchmark is run on 16 cores with 1 T/C versus 8 cores with 2 T/C. 2 T/C represents

Figure 14. Power and energy of multithreading versus multicore. Each microbenchmark is run with equal thread counts for both 1 T/C (multicore) and 2 T/C (multithreading) configurations.

multithreading and 1 T/C represents multicore. The same thread mappings are used for HP as in Section IV-H1. Energy is derived from the power and execution time. The microbenchmarks are modified for a fixed number of iterations instead of infinite, as was the case for power measurement, to measure execution time. The power and energy results for thread counts of two to 24 threads are shown in Figure 14. Note, we break power and energy down into active and idle portions. Idle power does not represent the full chip idle power, but the idle power for the number of active cores. This is calculated by dividing the full chip idle power by the total number of cores and multiplying by the number of active cores. Effectively, multicore is charged double the idle power of multithreading.

Interestingly, for Int and HP multithreading consumes more energy and less power than multicore. For Int, each core, independent of the T/C configuration, executes an integer computation instruction on each cycle. However, there are half the number of active cores for multithreading. Comparing active power, multithreading consumes similar power to multicore, indicating the hardware thread switching overheads are comparable to the active power of an extra core. This indicates, that a two-way fine-grained multi- threaded core may not be the optimal configuration from an energy efficiency perspective. Increasing the number of threads per core amortizes the thread switching overheads over more threads and the active power will likely be less than that for the corresponding extra cores for multicore. While switching overheads will increase with increased threads per core, we think the per thread switching overhead may decrease beyond two threads.

However, multithreading is charged idle power for half the cores compared to multicore. Thus, the total power for multicore is much higher than multithreading, but not double since the active power is similar. Translating this into energy, since the multithreading/multicore execution time ratio for Int is two, as no instruction overlapping occurs for multithreading, the total energy is higher for

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

Table VIII. SUN FIRE T2000 AND PITON SYSTEM SPECIFICATIONS

| System Parameter | Sun Fire T2000 | Piton System |
|---|---|---|
| Operating System | Debian Sid Linux | Debian Sid Linux |
| Kernel Version | 4.8 | 4.9 |
| Memory Device Type | DDR2-533 | DDR3-1866 |
| Rated Memory Clock Frequency | 266.67MHz (533MT/s) | 933MHz (1866MT/s) |
| Actual Memory Clock Frequency | 266.67MHz (533MT/s) | 800MHz (1600MT/s) |
| Rated Memory Timings (cycles) | 4-4-4 | 13-13-13 |
| Rated Memory Timings (ns) | 15-15-15 | 13.91-13.91-13.91 |
| Actual Memory Timings (cycles) | 4-4-4 | 12-12-12 |
| Actual Memory Timings (ns) | 15-15-15 | 15-15-15 |
| Memory Data Width | 64bits + 8bits ECC | 32bits |
| Memory Size | 16GB | 1GB |
| Memory Access Latency (Average) | 108ns | 848ns |
| Persistent Storage Type | HDD | SD Card |
| Processor | UltraSPARC T1 | Piton |
| Processor Frequency | 1Ghz | 500.05MHz |
| Processor Cores | 8 | 25 |
| Processor Thread Per Core | 4 | 2 |
| Processor L2 Cache Size | 3MB | 1.6MB aggregate |
| Processor L2 Cache Access Latency | 20-24ns | 68-108ns |

Table IX. SPECINT 2006 PERFORMANCE, POWER, AND ENERGY

| Benchmark/Input | Execution Time (mins) | | Piton Slowdown | Piton Avg. Power (W) | Piton Energy (kJ) |
|---|---|---|---|---|---|
| | UltraSPARC T1 | Piton | | | |
| bzip2-chicken | 11.74 | 57.36 | 4.89 | 2.199 | 7.566 |
| bzip2-source | 23.62 | 129.02 | 5.46 | 2.119 | 16.404 |
| gcc-166 | 5.72 | 38.28 | 6.70 | 2.094 | 4.809 |
| gcc-200 | 9.21 | 70.67 | 7.67 | 2.156 | 9.139 |
| gobmk-13x13 | 16.67 | 77.51 | 4.65 | 2.127 | 9.889 |
| h264ref-foreman-baseline | 22.76 | 71.08 | 3.12 | 2.149 | 9.162 |
| hmmer-nph3 | 48.38 | 164.94 | 3.41 | 2.400 | 23.750 |
| libquantum | 201.61 | 1175.70 | 5.83 | 2.287 | 161.363 |
| omnetpp | 72.94 | 727.04 | 9.97 | 2.096 | 91.431 |
| perlbench-checkspam | 11.57 | 92.56 | 8.00 | 2.137 | 11.863 |
| perlbench-diffmail | 23.13 | 184.37 | 7.97 | 2.141 | 22.320 |
| sjeng | 122.07 | 569.22 | 4.66 | 2.080 | 71.043 |
| xalancbmk | 102.99 | 730.03 | 7.09 | 2.148 | 94.077 |

multithreading. Note, we charge multicore for the idle power of a multithreaded core. While this is not quite accurate, the idle power for a single-threaded core will be lower, only making multicore power and energy even lower.

The results for HP have similar characteristics to Int. HP exercises the memory system in addition to performing integer computation, thus the overall power is higher. Since the mixed thread (executes integer and memory instructions) presents opportunities for overlapping memory and compute for multithreading, the multithreading/multicore execution time ratio is less than two. However, the percentage of cycles that present instruction overlapping opportunities is low because memory instructions hit in the L1 cache, which has a 3 cycle latency. Thus, the multithreading/multicore execution time ratio is close to two. Consequently, the trends for Int and HP are similar.

In contrast, multithreading is more energy efficient than multicore for Hist. Hist presents a large number of opportunities for overlapping memory and compute, causing multithreaded cores to stall less frequently while accessing the memory system, increasing power. Multithreading also has thread switching overheads. Multicore includes the active power for twice the number of cores, each of which stall more frequently while accessing the memory system. In total, active power for both configurations is nearly identical. The performance of multicore and multithreading is also similar due to the large number of overlapping opportunities.

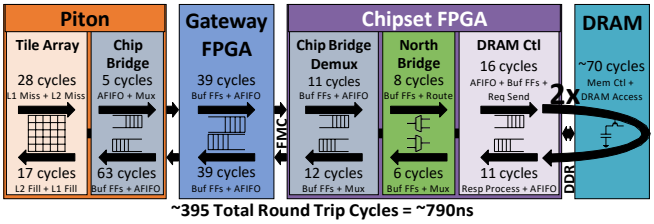
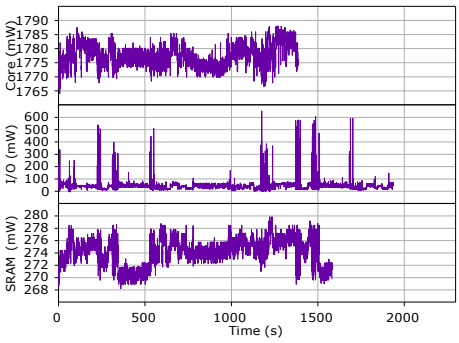



Figure 15. Piton system memory latency breakdown for a `ldx` instruction from tile0. All cycle counts are normalized to the Piton core clock frequency, 500.05MHz




This translates to similar active energy and double the idle energy for multicore, as it is charged double the idle power. This makes multithreading more energy efficient overall.

The increased overlapping opportunities presented by Hist makes multithreading more energy efficient than multicore. As a result, from an energy efficiency perspective, multi-

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

Figure 16. Time series of power broken down into each Piton supply over entire execution of `gcc-166`

threading favors applications with a mix of long and short latency instructions. Multicore performs better for integer compute heavy applications.

Last, Hist exhibits a different energy scaling trend than Int and HP. Hist energy remains relatively constant as threadcount increases, while Int and HP energy scales with thread count. This results from how the work scales with thread count. As previously stated, Hist maintains the same amount of total work and decreases work per thread as the thread count increases. Thus, the total energy should be the same for performing the same total amount of work. Int and HP increase the total work and maintain the work per thread with increasing thread counts, thus total energy increases. Hist energy, however, is not perfectly constant as thread count scales for 1 T/C. The data shows that 8 threads is the optimal configuration for 1 T/C from an energy efficiency perspective. This is because 8 threads is the point at which the thread working set size fits in the L1 caches. Beyond that, energy increases since useful work per thread decreases and lock contention increases, resulting in threads spending more energy contending for locks.

These results only apply to fine-grained multithreading. Results for simultaneous multithreading may differ.

*H. Benchmark Study*

In order to evaluate performance, power, and energy of real applications we use ten benchmarks from the SPECint 2006 benchmark suite [45]. We ran the benchmarks on both the Piton system and a Sun Fire T2000 system [46] with an UltraSPARC T1 processor [27]. A comparison of the system specifications is listed in Table VIII. The UltraSPARC T1 processor has the same core and L1 caches as Piton, except

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

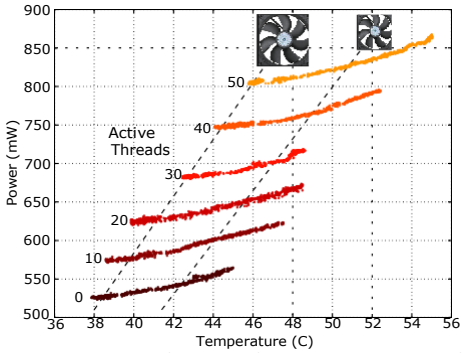



Figure 17. Chip power as a function of package temperature for different number of active threads. Cooling is varied to sweep temperature.

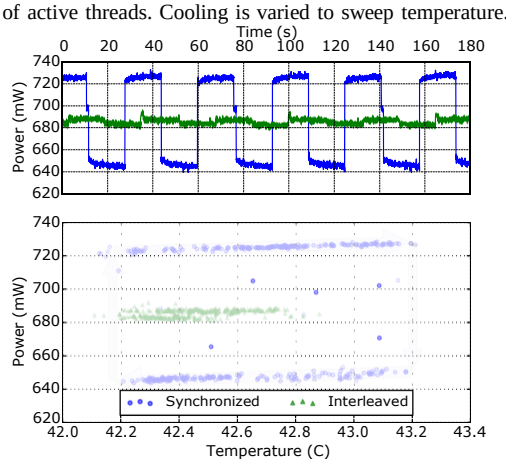



Figure 18. Power variations and power/temperature dependence for synchronized (blue) and interleaved (green) scheduling of the two-phase test application.

with four threads per core instead of two. However, the Piton uncore is completely different.

Table IX shows the execution time for the UltraSPARC T1 and Piton and the average power and energy for Piton. Power measurements are taken throughout the application runtime, not just for 128 samples as in previous studies. There are a number of reasons for the difference in performance. The UltraSPARC T1 processor runs at two times the clock frequency of Piton. The Sun Fire T2000 has much more main memory and the UltraSPARC T1 has almost two times the amount of L2 cache as Piton. Additionally, Piton's L2 access latency is larger (but the cache is more scalable) and there is an 8x discrepancy in memory latency.

There are a couple reasons for the large discrepancy in memory latency. One is the latency of the memory devices. While the Piton system uses DDR3 DRAM and the SunFire T2000 uses DDR2 DRAM, the latency is the same, as indicated by the memory parameters in Table VIII. The DDR3 DRAM in the Piton system is rated for a 933MHz actual memory timings than the memory devices are capable of supporting. Additionally, the Piton system DRAM has a 32bit interface, while the SunFire T2000 DRAM has a 64bit interface. This requires the Piton system to make two DRAM accesses for each memory request.

Another reason for the discrepancy is the latency to get to memory and back in the Piton system. Figure 15 shows a breakdown of where cycles are spent in the journey of a `ldx` instruction from tile0 to memory and back. All cycle counts are normalized to the Piton core clock clock frequency, but due to a limitation in the Xilinx memory controller IP, we are only able to clock it at 800MHz. Since the memory controller must honor the rated memory timings in nanoseconds, the actual memory timing in cycles differs from what the devices are rated for. Due to the quantization of memory timings by clock cycles, this results in longer

frequency, 500.05MHz. Almost 80 cycles are spent in the gateway FPGA and a number of cycles are wasted in off-chip buffering and multiplexing. We believe with additional optimization, some of this overhead can be reduced. Further,if the Piton PCB were to be redesigned, it could include DRAM and I/O directly on the board and eliminate the need for the gateway FPGA. For optimal performance, the DRAM controller should be integrated on chip, as in the UltraSPARC T1.

The average power for SPECint benchmarks is marginally larger than idle power, as only one core is active, and is similar across benchmarks. `hmmer` and `libquantum` are exceptions, largely due to high I/O activity (verified by analyzing VIO power logs). This is likely due to high ratios of memory instructions and high sensitivity to data cache sizes among the SPECint applications [47].

Energy results correlate closely with execution times, as the average power is similar across applications. Figure 16 shows a power breakdown time series for `gcc-166`. More breakdowns can be found at http://www.openpiton.org.

### *I. Thermal Analysis*

Researchers have studied thermal aspects of multicore andmanycore architectures to deliver lower power consumption,better power capping, and improved reliability [48], [49]. Inthis section, we characterize some thermal aspects of Piton toquantify the relationship between temperature and power andhow application scheduling affects power and temperature. We conducted our experiment without the heat sink to have access to the surface of the package. To ensure that wedon't harm the chip, core frequency, VDD, and VCS weredecreased to 100.01MHz, 0.9V, and 0.95V, respectively. Additionally, we use a different chip for this experimentwhich has not been presented in this paper thus far. Weused the FLIR ONE [50] thermal camera to measure surfacetemperature. We fixed the fan position with respect to thechip to guarantee similar airflow for all tests. The roomtemperature was $20.0 \pm 0.2^{\circ}C$ and the chip temperature under reset was around $60^{\circ}C$.

Figure 17 shows the total power consumption as a function of the package hotspot temperature for different numbers of active threads running the High Power (HP) application. The temperature is actively adjusted by changing the fan's angle. The exponential relationship between

power and temperature has been shown to be caused by leakage power in CMOS transistors [51] and is part of a larger trend which can be explored under a wider range of room temperatures and cooling solutions.

Many scholars have studied manycore scheduling strategies and power budgeting with respect to thermal design power (TDP), power capping, and lifetime concerns [52], [53]. In order to analyze such impacts, we developed a test application with two distinct phases: a compute heavy phase (arithmetic loop) and an idle phase (`nop` loop). We ran this application on all fifty threads with two different scheduling strategies, synchronized and interleaved. In synchronized scheduling all threads alternate between phases simultaneously, executing the compute and idle phases together. Interleaved schedules 26 threads in one phase and 24 in the opposite phase, thus half the threads execute the compute phase while the other half execute the idle phase.

Figure 18 shows the total power consumption with respect to time and package temperature for synchronized (blue) and interleaved (green) scheduling. The changes in power consumption resulting from application phase changes causes corresponding changes in temperature. Changes in temperature feedback and cause corresponding changes in power. The hysteresis caused by this feedback loop can be observed in the power/temperature plot for both scheduling strategies. However, synchronized exhibits a much larger hysteresis, indicated by blue arrows in the figure. The average temperature for interleaved scheduling is $0.22^{\circ}C$ lower than synchronized, highlighting the impact different scheduling strategies for the same application can have. This shows how a balanced schedule can not only limit the peak power but also decrease the average CPU temperature.

### *J. Applicability of Results*

It is important to note in which cases the results and insights in this work are applicable, as a single characterization cannot make generalizations about all chips. Ideally, researchers who would like to make use of the data should use OpenPiton in their research, as the characterization data would directly apply. However, this is not the case for designs dissimilar to Piton. We believe ISA has less of an impact on characterization results, so our results are likely applicable to somewhat similar ISAs. Microarchitecture will likely impact results to a greater extent. For example, our results for a fine-grained multithreaded core will not apply to a simultaneous multithreaded core. Thus, it is difficult to apply the results to very different microarchitectures, and this should not be expected. However, researchers studying similar microarchitectures or microarchitectural mechanisms can make use of the results.

## IV. Related Work

Manycore chips have been built in both industry and academia to explore power-efficient, high-performance com-

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 04, April : 2022

Table X. COMPARISON OF INDUSTRY AND ACADEMIC PROCESSORS

| Processor | Academic/ Industry | Manycore/ Multicore | Open Source | Published Detailed Power/ Energy Characterization |
|---|---|---|---|---|
| Intel Xeon Phi Knights Corner [59] | Industry | Manycore | ✗ | ✓ [23], [24] |
| Intel Xeon Phi Knights Landing [16] | Industry | Manycore | ✗ | ✗ |
| Intel Xeon E5-2670 [60] | Industry | Multicore | ✗ | ✓ [26] |
| Marvell MV78460 [61] (ARM Cortex-A9) | Industry | Multicore | ✗ | ✓ [26] |
| TI 66AK2E05 [62] (ARM Cortex-A15) | Industry | Multicore | ✗ | ✓ [26] |
| Cavium ThunderX [17] | Industry | Manycore | ✗ | ✗ |
| Phytium Technology Mars [18] | Industry | Manycore | ✗ | ✗ |
| Qualcomm Centriq 2400 Processor [19] | Industry | Manycore | ✗ | ✗ |
| Tilera Tile-64 [57] | Industry | Manycore | ✗ | ✗ |
| Tilera TILE-Gx100 [58] | Industry | Manycore | ✗ | ✗ |
| Sun UltraSPARC T1/T2 [27], [63] | Industry | Multicore | ✓ | ✗ |
| IBM POWER7 [64] | Industry | Multicore | ✗ | ✓ [65] |
| MIT Raw [38] | Academic | Manycore | ✗ | ✓ [33] |
| UT Austin TRIPS [39] | Academic | Multicore | ✗ | ✗ |
| UC Berkeley 45nm RISC-V [41] | Academic | Unicore | ✓ | ✗ [41][1] |
| UC Berkeley 28nm RISC-V [40] | Academic | Multicore | ✓ | ✗ [40][2] |
| MIT SCORPIO [32], [55] | Academic | Manycore | ✗ | ✗ |
| U. Michigan Centip3De [54] | Academic | Manycore | ✗ | ✓ [54] |
| NCSU AnyCore [66] | Academic | Unicore | ✓ | ✗ [66][1] |
| NCSU H3 [67] | Academic | Multicore | ✓ | ✗ |
| Celerity [56] | Academic | Manycore | ✓ | ✗ |
| Princeton Piton [12], [13] | Academic | Manycore | ✓ | ✓ |

[1] Minor power numbers provided, no detailed characterization
[2] Performed power/energy characterization of on-chip DC-DC converters, not processor architecture

puting. Many of these chips, such as Cavium's ThunderX processor [17], Phytium Technology's Mars processor [18], Qualcomm's Centriq 2400 Processor [19], University of Michigan's Centip3De processor [54], and the SW26010 processor in the Sunway TaihuLight supercomputer [20] are designed to be used in supercomputers or high-performance servers for cloud computing and other applications. Like Piton, manycore chips including MIT's Raw processor [38], Intel's Knights Landing Xeon Phi processor [16], MIT's SCORPIO processor [32], [55], the Celerity processor [56], and Tilera's Tile-64 [57] and TILE-Gx100 [58] processors utilize a tile-based architecture with NoCs.

Unfortunately, little detailed power and energy data has been publicly released for these chips, evident from Table X which compares academic and industry chips taped-out in silicon. Academics have characterized some manycore chips [23], [24], [29], [33], however the source for these chips was never released publicly. Thus, researchers are unable to correlate the results to design details and RTL models. Further, academic characterizations of proprietary designs are not able to verify, through simulation, that the design behaves as expected during measurement. Piton has been open sourced and this work presents detailed power and energy characterization which verifies expected behavior through simulation. Moreover, the test setup was specifically designed with power characterization in mind, which can be a limiting factor for some of the mentioned characterizations. This work represents the first detailed power and energy characterization of an open source manycore processor, enabling correlation of the data to the RTL and design.

Of course, other work has characterized and/or modeled power, energy, and/or thermal aspects of processors [21], [22], [25], [26], [65], [68]–[71]. This paper performs characterization in the context of a tiled manycore architecture.

## V. CONCLUSION

In this work, we present the first detailed power and energy characterization of an open source manycore research processor taped-out in silicon. Specifically, we studied volt-

age versus frequency scaling, energy per instruction (EPI), memory system energy, NoC energy, and application power and energy. The characterization revealed a number of insights, including the impact of operand values on EPI, that recomputing data can be more energy efficient than loading it from memory, on-chip data transmission energy is low, and energy efficient multithreaded design insights. All hardware infrastructure along with all data collected has been open sourced and is available at http://www.openpiton.org. We hope that this characterization, including the data and the insights derived, enables researchers to develop future research directions, build accurate power models for manycore processors, and make better use of OpenPiton for accurate power and energy research.

## ACKNOWLEDGMENT

We thank Samuel Payne for his work on Piton's off-chip interface and the chipset memory controller, Xiaohua Liang for his work on chipset routing and transducing, and Ang Li and Matthew Matl for their work on the chipset SD controller. This work was partially supported by the NSF under Grants No. CCF-1217553, CCF-1453112, and CCF-1438980, AFOSR under Grant No. FA9550-14-1-0148, and DARPA under Grant No. N66001-14-1-4040, HR0011-13-2-0005, and HR0011-12-2-0019. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our sponsors.

## REFERENCES

[1] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[2] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proc. of the 38th Annual International Symposium on Computer Architecture*, ser. ISCA '11, 2011, pp. 365–376.

[3] M. B. Taylor, "A landscape of the new dark silicon design regime," *IEEE Micro*, vol. 33, no. 5, pp. 8–19, 2013.

[4] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki, "Toward dark silicon in servers," *IEEE Micro*, vol. 31, no. 4, pp. 6–15, 2011.

[5] M. McKeown, J. Balkind, and D. Wentzlaff, "Execution drafting: Energy efficiency through computation deduplication," in *Proc. of 47th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 47, 2014, pp. 432–444.

[6] M. Laurenzano, Y. Zhang, J. Chen, L. Tang, and J. Mars, "PowerChop: Identifying and managing non-critical units in hybrid processor architectures," in *Proc. of the 43rd Annual International Symposium on Computer Architecture*, ser. ISCA '16, 2016, pp. 140–152.

[7] H. Cherupalli, R. Kumar, and J. Sartori, "Exploiting dynamic timing slack for energy efficiency in ultra-low-power embedded systems," in *Proc. of the 43rd Annual International Symposium on Computer Architecture*, ser. ISCA '16, 2016, pp. 671–681.

[8] S. Das, T. M. Aamodt, and W. J. Dally, "SLIP: Reducing wire energy in the memory hierarchy," in *Proc. of the 42nd Annual International Symposium on Computer Architecture*, ser. ISCA '15, 2015, pp. 349–361.

[9] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott, "Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling," in *Proc. of the 8th International Symposium on High-Performance Computer Architecture*, ser. HPCA '02, 2002, pp. 29–.

[10] A. Lukefahr *et al.*, "Composite cores: Pushing heterogeneity into a core," in *Proc. of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-45, 2012, pp. 317–328.

[11] G. Venkatesh *et al.*, "Conservation cores: Reducing the energy of ma- ture computations," in *Proc. of the Fifteenth International Conference on Architectural Support for Programming Languages and OperatingSystems*, ser. ASPLOS '10, 2010, pp. 205–218.

[12] M. McKeown *et al.*, "Piton: A 25-core academic manycore processor,"in *Hot Chips: A Symposium on High Performance Chips (HC28(2016))*, 2016.

[13] ——, "Piton: A manycore processor for multitenant clouds," *IEEE Micro*, vol. 37, no. 2, pp. 70–80, 2017.

[14] V. Kontorinis *et al.*, "Managing distributed UPS energy for effective power capping in data centers," in *Proc. of the 39th Annual Interna- tional Symposium on Computer Architecture*, 2012, pp. 488–499.

[15] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, 2007.

[16] A. Sodani, "Knights Landing (KNL): 2nd generation Intel Xeon Phi processor," in *Hot Chips: A Symposium on High Performance Chips (HC27 (2015))*, 2015.

[17] L. Gwennap, "Thunderx rattles server market," *Microprocessor Report*, vol. 29, no. 6, pp. 1–4, 2014.

[18] C. Zhang, "Mars: A 64-core ARMv8 processor," in *Hot Chips: A Symposium on High Performance Chips (HC27 (2015))*, 2015.

[19] T. Speier and B. Wolford, "Qualcomm Centriq 2400 processor," in *Hot Chips: A Symposium on High Performance Chips (HC29 (2017))*,2017.

[20] H. Fu *et al.*, "The Sunway TaihuLight supercomputer: system and applications," *Science China Information Sciences*, vol. 59, no. 7, p. 072001, 2016.

[21] N. Julien, J. Laurent, E. Senn, and E. Martin, "Power consumption modeling and characterization of the TI c6201," *IEEE Micro*, vol. 23,no. 5, pp. 40–49, 2003.

[22] D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, "An energy efficiency feature survey of the Intel Haswell processor," in *Proc. of IEEE International Parallel and Distributed Processing Symposium Workshop*, ser. IPDSW '15, 2015, pp. 896–904.

[23] J. Wood, Z. Zong, Q. Gu, and R. Ge, "Energy and power characterization of parallel programs running on Intel Xeon Phi," in *Proc. of 43rd International Conference on Parallel Processing Workshops*, ser.ICCPW, 2014, pp. 265–272.

[24] Y. S. Shao and D. Brooks, "Energy characterization and instruction-level energy model of Intel's Xeon Phi processor," in *Proc. of the 2013 International Symposium on Low Power Electronics and Design*, ser. ISLPED '13, 2013, pp. 389–394.

[25] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proc. of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 36, 2003, pp. 93–.

[26] M. A. Laurenzano *et al.*, "Characterizing the performance-energy tradeoff of small ARM cores in HPC computation," in *European Conference on Parallel Processing*. Springer, 2014, pp. 124–137.

[27] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: a 32-way multithreaded Sparc processor," *Micro, IEEE*, vol. 25, no. 2, pp. 21– 29, 2005.

[28] J. Zhan, N. Stoimenov, J. Ouyang, L. Thiele, V. Narayanan, and Y. Xie, "Designing energy-efficient NoC for real-time embedded systems through slack optimization," in *Proc. of the 50th Annual Design Automation Conference*, ser. DAC '13, 2013, pp. 37:1–37:6.

[29] L. Shang, L.-S. Peh, A. Kumar, and N. K. Jha, "Thermal modeling, characterization and management of on-chip networks," in *Proc. of the 37th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 37, 2004, pp. 67–78.

[30] R. Hesse and N. E. Jerger, "Improving DVFS in NoCs with coherence prediction," in *Proc. of the 9th International Symposium on Networks-on-Chip*, ser. NOCS '15, 2015, pp. 24:1–24:8.

[31] L. Chen, D. Zhu, M. Pedram, and T. M. Pinkston, "Power punch: Towards non-blocking power-gating of NoC routers," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture*, ser. HPCA '15, 2015, pp. 378–389.

[32] B. K. Daya *et al.*, "SCORPIO: A 36-core research chip demonstrating snoopy coherence on a scalable mesh NoC with in-network ordering," in *Proceeding of the 41st Annual International Symposium on Computer Architecuture*, ser. ISCA '14, 2014, pp. 25–36.

[33] J. S. Kim, M. B. Taylor, J. Miller, and D. Wentzlaff, "Energy characterization of a tiled architecture processor with on-chip networks," in *Proc. of the 2003 International Symposium on Low Power Electronics and Design*, ser. ISLPED '03, 2003, pp. 424–427.

[34] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a Teraflops Processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, 2007.

[35] J. Howard *et al.*, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, 2011.

[36] A. B. Kahng, B. Li, L. S. Peh, and K. Samadi, "Orion 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," in *2009 Design, Automation Test in Europe Conference Exhibition*, 2009, pp. 423–428.

[37] J. Balkind *et al.*, "OpenPiton: An open source manycore research framework," in *Proc. of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '16, 2016, pp. 217–232.

[38] M. B. Taylor *et al.*, "The Raw microprocessor: a computational fabric for software circuits and general-purpose programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, 2002.

[39] K. Sankaralingam *et al.*, "Distributed microarchitectural protocols in the TRIPS prototype processor," in *Proc. of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 39, 2006, pp. 480–491.

[40] B. Zimmer *et al.*, "A RISC-V vector processor with simultaneous-switching switched-capacitor DC-DC converters in 28 nm FDSOI," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 930–942, 2016.

[41] Y. Lee *et al.*, "A 45nm 1.3GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators," in *40th European Solid State Circuits Conference*, ser. ESSCIRC '14, 2014, pp. 199–202.

[42] Y. Fu, T. M. Nguyen, and D. Wentzlaff, "Coherence domain restriction on large scale systems," in *Proc. of 48th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 48, 2015, pp. 686–698.

[43] Y. Zhou and D. Wentzlaff, "MITTS: Memory inter-arrival time traffic shaping," in *Proc. of 43rd Annual International Symposium on Computer Architecture*, ser. ISCA '16, 2016, pp. 532–544.

[44] "BaseJump: Open source components for ASIC prototypes," http://bjump.org, accessed: 2016-11-17.

[45] "Standard Performance Evaluation Corporation," http://www.spec.org/, 2017, accessed: 2017-12-11.

[46] "Sun Fire T2000 server," http://www.oracle.com/us/products/servers-storage/servers/sparc-enterprise/t-series/034348.pdf, 2009, accessed: 2017-4-4.

[47] A. Phansalkar, A. Joshi, and L. K. John, "Analysis of redundancy and application balance in the SPEC CPU2006 benchmark suite," in *Proc. of the 34th Annual International Symposium on Computer Architecture*, ser. ISCA '07, 2007, pp. 412–423.

[48] T. Ebi, M. A. A. Faruque, and J. Henkel, "TAPE: Thermal-aware agent-based power economy multi/many-core architectures," in *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, 2009, pp. 302–309.

[49] Y. Ge, P. Malani, and Q. Qiu, "Distributed task migration for thermal management in many-core systems," in *Proc. of the 47th Design Automation Conference*, ser. DAC '10, 2010, pp. 579–584.

[50] "FLIR ONE," http://www.flir.com/flirone/ios-android/, accessed: 2016-04-04.

[51] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leak- age current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.

[52] K. Ma, X. Li, M. Chen, and X. Wang, "Scalable power controlfor many-core architectures running multi-threaded applications," in

*Proc. of the 38th Annual International Symposium on Computer Architecture*, ser. ISCA '11, 2011, pp. 449–460.

[53] S. Pagani *et al.*, "TSP: Thermal safe power: Efficient power budgeting for many-core systems in dark silicon," in *Proc. of the 2014 International Conference on Hardware/Software Codesign and SystemSynthesis*, ser. CODES '14, 2014, pp. 10:1–10:10.

[54] R. G. Dreslinski *et al.*, "Centip3De: A 64-core, 3D stacked near-threshold system," *IEEE Micro*, vol. 33, pp. 8–16, 2013.

[55] C.-H. O. Chen *et al.*, "Scorpio: 36-core shared memory processor demonstrating snoopy coherence on a mesh interconnect," in *Hot Chips: A Symposium on High Performance Chips (HC26 (2014))*, 2014.

[56] S. Davidson, K. Al-Hawaj, and A. Rovinski, "Celerity: An open source RISC-V tiered accelerator fabric," in *Hot Chips: A Symposiumon High Performance Chips (HC29 (2017))*, 2017.

[57] S. Bell *et al.*, "Tile64-processor: A 64-core SoC with mesh interconnect," in *Proceedings of the International Solid-State Circuits Conference*, ser. ISSCC '08. IEEE, 2008, pp. 88–598.

[58] C. Ramey, "TILE-Gx100 manycore processor: Acceleration interfaces and architecture," in *Hot Chips: A Symposium on High Performance Chips (HC23 (2011))*, 2011.

[59] G. Chrysos, "Intel Xeon Phi coprocessor (codename Knights Corner)," in *Hot Chips: A Symposium on High Performance Chips (HC24(2012))*, 2012.

[60] "Intel Xeon processor E5-2670," https://ark.intel.com/products/64595/Intel-Xeon-Processor-E5-2670-20M-Cache-2 60-GHz-8 00-GTs-Intel-QPI, accessed: 2017-4-3.

[61] "Mv78460 ARMADA XP highly integrated multi-core ARMv7 based system-on-chip processors," http://www.marvell.com/embedded-processors/armada-xp/assets/HW MV78460 OS.PDF, 2014, accessed: 2017-4-3.

[62] *66AK2E0x Multicore DSP+ARM KeyStoneII System-on-Chip(SoC)*, Texas Instruments, Dallas, Texas, March 2015.

[63] M. Shah *et al.*, "UltraSPARC T2: A highly-threaded, power-efficient, SPARC SOC," in *2007 IEEE Asian Solid-State Circuits Conference*, 2007, pp. 22–25.

[64] W. J. Starke, "POWER7: IBM's next generation, balanced POWER server chip," in *Hot Chips: A Symposium on High Performance Chips(HC21 (2009))*, 2009.

[65] R. Bertran, A. Buyuktosunoglu, M. S. Gupta, M. Gonzalez, and P. Bose, "Systematic energy characterization of CMP/SMT processor systems via automated micro-benchmarks," in *Proc. of the 45th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-45, 2012, pp. 199–211.

[66] R. B. R. Chowdhury, A. K. Kannepalli, S. Ku, and E. Rotenberg, "Anycore: A synthesizable RTL model for exploring and fabricating adaptive superscalar cores," in *Proc. of the International Symposium on Performance Analysis of Systems and Software*, ser. ISPASS '16, 2016, pp. 214–224.

[67] E. Rotenberg *et al.*, "Rationale for a 3D heterogeneous multi-core processor," in *Proc of the 31st International Conference on ComputerDesign*, ser. ICCD '13, 2013, pp. 154–168.

[68] D. Kim *et al.*, "Strober: Fast and accurate sample-based energy simulation for arbitrary RTL," in *Proc. of the 43rd Annual International Symposium on Computer Architecture*, ser. ISCA '16, 2016, pp. 128– 139.

[69] W. Huang *et al.*, "Accurate fine-grained processor power proxies," in *Proc. of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-45, 2012, pp. 224–234.

[70] A. Varma, E. Debes, I. Kozintsev, and B. Jacob, "Instruction-level power dissipation in the Intel XScale embedded microprocessor," in *Electronic Imaging 2005*. International Society for Optics and Photonics, 2005, pp. 1–8.

[71] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, 2006.