

# Worst-Case Energy Consumption: A New Challenge for Battery-Powered Critical Devices

David Trilla, Carles Hernandez, Jaume Abella, Francisco J. Cazorla

**Abstract**—The number of (edge) devices connected to the IoT is on the rise, reaching hundreds of billions in the next years. Many devices will implement some type of critical functionality, for instance in the medical market this includes infusion pumps and implantable defibrillators. Energy awareness is mandatory in the design of IoT devices given their huge impact on worldwide energy consumption and the fact that many of them are battery powered. *Critical* IoT devices further require addressing new energy-related challenges. On the one hand, factoring in the impact of energy-solutions on device's performance, providing evidence of adherence to domain-specific safety standards. On the other hand, deriving safe worst-case energy consumption (WCEC) estimates is fundamental to ensure the system can continuously operate under a pre-established set of power/energy caps, safely delivering its critical functionality. In this line, we analyze for the first time the impact that different hardware physical parameters have on both model-based and measurement-based WCEC modeling, for which we also show the main challenges they face compared to chip manufacturers' current practice for energy modeling and validation. Under the set of constraints that emanate from how certain physical parameters can be actually modeled, we show that measurement-based WCEC is a promising way forward for WCEC estimation.

**Index Terms**—Real Time, Worst-Case Energy Consumption, Energy

## 1 INTRODUCTION

The proliferation of battery-powered IoT and power-constrained devices controlling increasingly critical aspects of human life is relentless in domains such as health, smart cities, and intelligent transportation systems. The complexity of software running on those devices increases every generation to cover the demands for more autonomous operation, implementing decision making and data analysis techniques (among others). Handheld devices, which will govern part of the critical-applications functionality, will also inherit part of application criticality.

In battery-powered devices, energy is one of the most important resources as battery life is a key element for products' competitive edge. Analogously, power-constrained devices cannot exceed specific energy thresholds in short timeframes due to limited power sources (e.g. solar cells in Space). This has resulted in a vast set of academic and industrial works on low-power techniques at different levels: from hardware design to system/application software. When the device implements some type of critical functionality, a new set of energy-related requirements arise. This emanates from the fact that critical functionality – and in particular the hardware and software implementing it – has to undergo a stringent validation and verification (V&V) process to show adherence to the prospects in domain-specific standards [33]. V&V provide evidence of correct functional behavior, while also cover non-functional aspects like ensuring that the timing (duration) of software fits its allocated budget. Software timing verification is achieved by deriving worst-case execution time (WCET) estimates for tasks and deriving a feasible schedule for all tasks.

The rise of battery-powered and power-constrained critical

devices makes energy a first-class citizen, as relevant as functional and timing requirements. At the V&V level, evidence must be provided that power-control techniques do not jeopardize the safe operation of the device [23]. This relates to assessing the effect of those techniques on the timing of the software to prevent any overruns and providing evidence that they are triggered/deactivated in a controlled manner [10]. Evidence is required that with a given energy budget the device can effectively run all critical activities (tasks) due to battery or power source related constraints. This calls for methods and tools for worst-case energy consumption (WCEC) estimation. In battery-powered devices evidence is needed to show that task runs (jobs) can execute adhering to their WCEC bound, so that the total energy consumed during operation is proven not to exceed battery capacity. Meanwhile, in power-constrained devices similar evidence is needed within smaller timeframes to prove that energy consumed does not exceed power supply capabilities.

Intuitively, the properties required on WCEC estimates are comparable to those for WCET estimates, namely providing tight upper bounds to actual energy consumptions and evidence for certification. However, as we show in this paper, despite the similarities in the concept, WCET and WCEC estimation are different processes subject to fundamentally different sets of requirements coming from the hardware. The latter shapes the set of assumptions that can be made on the hardware information required for tight energy measuring and modeling.

In this paper, we make a taxonomy of the factors affecting dynamic and static energy consumption and hence, WCEC estimation. We describe the difficulties in deriving tight WCEC estimates using model- and measurement-based approaches. The contributions of this paper can be summarized as follows: ① We analyze how physical effects cause power/energy variations (Section III). ② We describe chip manufacturers' current practice for average and maximum energy estimation (Section IV). ③ We show that (static) model-based WCEC estimation builds on pa-

- D. Trilla is with the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.
- D. Trilla and C. Hernandez and J. Abella and F.C. Cazorla are with the Barcelona Supercomputing Center (BSC), Barcelona, Spain.
- C. Hernandez is with the Universitat Politècnica de València, València, Spain.

rameters usually not made public by chip manufacturers, which makes it resorting to worst-case predictions potentially delivering pessimistic WCEC estimates (Section V). ④ We describe the challenges for measurement-based WCEC estimation that relate to dealing with PAVT variations and software effects (Section VI). ⑤ We conclude (Section VII) that model-based WCEC estimation will find difficulties to effectively derive WCEC estimates for software running on real complex processors being useful in early-design stages to derive tasks’ early WCEC estimates. Instead, measurement-based approaches are promising to obtain tighter task-level WCEC estimates on complex processors on late-design phases and further provide evidence for certification.

Overall, we settle the ground on the challenges for practical and reliable WCEC estimation and aim at becoming a reference for future works on WCEC estimation.

## 2 BACKGROUND

### 2.1 Validation and Verification

*Criticality* derives from functional safety and safety standards, e.g. IEC 62304 for medical devices and IEC61508 for industry). Interestingly, safety standards do not aim at removing the appearance of failures, which is arguably impossible in a real system. Instead, they aim at making their likelihood of occurrence to be quantified and assessed against reference values, asserting with sufficiently high confidence that the residual risk of violation falls below tolerable rates. In this line, despite common wisdom, systems are designed such that a task overrun never lead to an unsafe state of the system, which would mean a bad-designed safety solution. A safety process is defined (according to the corresponding standard) covering the definition of safety goals and requirements, and a safety strategy in general, to mitigate the risk that hardware or software misbehavior causes a system failure. As the criticality of the software component under analysis increases, more mechanisms are put in place (replication, online monitoring, watchdog) to detect and react to undesired situations.

### 2.2 Power estimation

Following common practice, in this section we build on power formulation (rather than energy), although power and energy can be used interchangeably given a fixed execution time  $t$ . The relation between power ( $P$ ) and energy ( $E$ ) is given by  $E = P \cdot t$ . Dissipated power can be classified into two complementary terms:  $P_{total} = P_{stat} + P_{dyn}$ .

**Static power** ( $P_{stat}$ ) dissipates when maintaining a circuit powered up. It covers the power dissipated through leakages, free carriers (electrons and holes) that are able to scape the isolation layers of the silicon. There are several models for deriving static energy consumption but a widely accepted formulation is  $P_{static} = V_{cc} \cdot N \cdot k_{design} \cdot I_{leakage} \cdot V_{cc}$  is the nominal voltage for the circuit;  $N$  the number of transistors;  $k_{design}$  an implementation dependent constant; and  $I_{leakage}$  the leakage current that depends on the technology used for the chip implementation [9].

Interestingly,  $N$  and  $k_{design}$  are truly constant parameters, while  $V_{cc}$  and  $I_{leakage}$  are theoretically assumed constant, but they can actually suffer some fluctuation.  $V_{cc}$  may vary due to techniques for power saving such as Dynamic Voltage and Frequency Scaling (DVFS) and drowsy operation modes [20].  $V_{cc}$  also depends on the quality of the voltage supply source and the chip package. It further suffers from significant fluctuations

at operation time, especially in multicore setups [4], since the likelihood of abrupt power dissipation variations increases due to, for instance, several cores having high energy consumption requirements at the same time. This creates current glitches and thus, voltage droops.  $I_{leakage}$  highly depends on the thermal status, so that high temperatures increase the leakage current, thus increasing the dissipated static power.

While  $k_{design}$  is constant, it is an approximation to abstract the internal complexities of processor designs and also depends on the individual chip fabricated since process variations lead to variations across chip units.

Despite those sources of variation,  $P_{stat}$  is often assumed constant due to it being highly stable over time, which makes nominal  $P_{stat}$  estimates be very precise w.r.t. average behavior. However, this does not necessarily hold for maximum  $P_{stat}$  estimates, which are the ones of interest in this paper.

**Dynamic power** ( $P_{dyn}$ ) dissipates due to the charging and discharging of transistor’s gate capacitance and can be expressed as  $P_{dyn} = A \cdot V_{cc}^2 \cdot C_{eq} \cdot f$ , where  $A$  is the switching activity or activity factor, representing the percentage of transistors’ capacitance flipping value,  $V_{cc}$  is the nominal voltage,  $C_{eq}$  is the equivalent capacitance of the transistor inputs and  $f$  is the operating frequency of the device.

In general, all those parameters are subject to variations, and so it is  $P_{dyn}$ .  $A$  strongly depends on the input changes of the components. Those inputs include data and control signals of the circuit. As an example,  $A$  for an adder depends on the input data variation as well as on the control signals to add/subtract, etc. Deriving approximations to  $A$  has been the subject of intense research [29]. It has been observed that  $A$  decreases exponentially across gate levels when moving from inputs to outputs [29]. However, this cannot be proven in general and the exponential factor can only be approximated for specific circuit types. Thus, to the best of our knowledge, reliable and tight upper-bounds to the activity factor usable for any type of circuit do not exist.

$V_{cc}$  suffers from the same variation effects explained before. In the case of DVFS, both  $V_{cc}$  and  $f$  vary coordinately. In that case, we regard  $f$  as constant w.r.t.  $V_{cc}$ , so that given a nominal  $V_{cc}$  value, a given nominal  $f$  is set. In practice,  $f$  may change when the clock source is subject to some form of variation, such as, for instance, temperature variations, which may slow down or speed up the clock slightly given a fixed  $V_{cc}$  value.  $C_{eq}$  is a nominal value that depends on the size of the transistors and it is also subject to process variations introduced during manufacturing.

## 3 SOURCES OF POWER VARIABILITY

Two are the main physical factors that particularly complicate power estimation at the hardware component level.

- 1) The power dissipation of any hardware component (e.g. the whole processors or a floating point unit) varies across units<sup>1</sup>. Further, power dissipation figures differ from their (theoretical) nominal value. This relates to physical limitations for hardware manufacturing.
- 2) The power dissipation of a given component varies over time in each unit due to several sources of variation.

Operation-time (fabrication) process, aging, temperature, and voltage (PAVT) variations cause that, even if hardware designers

1. A unit is a physical implementation of a given component, e.g., a processor may have two floating-point units

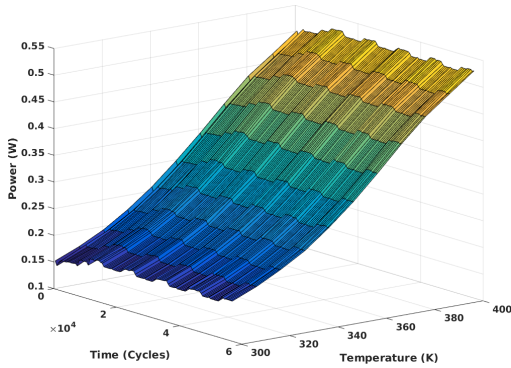


Fig. 1. Average power dissipation of a program through execution time for different temperatures. The binary alternates execution of memory and FPU instructions on an in-order 4 stage processor with separate instruction and data level 1 caches, and a unified level 2 cache.

could model circuits at the lowest (most-accurate) level, designers would still miss the actual variations experienced by each individual processor unit. This seriously complicates – in fact makes it de facto impossible – predicting exactly power consumption a priori. Furthermore, specific processor unit(s) under study are used to derive power estimates for all of them.

**Process Variations.** Limitations in the manufacturing process cause device (e.g. wires, vias and transistor components) parameters (e.g. geometry, thickness and number of dopants) to differ from their nominal values. Taking as an example the lithographic process, variations have a systematic and a random component. The former manifests in spatial correlation so that variations affect in a similar manner neighboring devices; while the random component refers to individual devices suffering independent variations. Variations make delay and power dissipation of each individual device differ from nominal values and at a coarser granularity, variations lead to delay and power variability of processor components. For instance, 3X power variations with 90nm technology [11] and 20X leakage (static) power variations [5] with 180nm technology across different processor units (between the most power efficient and the most power hungry units).

**Aging Variations** like electromigration [3], bias temperature instability [35], and hot carrier injection [12], affect the resistance of wires and threshold voltage ( $V_t$ ) of transistors. They also change processor energy consumption over time and affect physical characteristics of the devices by displacing molecules and dopants from their original locations. Hence, power dissipation for a unit slowly changes over time.

**Temperature and Voltage Variations.** Processors operate within a given temperature and supply voltage range. Both of them vary due to the activity of the whole processor, ambient temperature and physical characteristics of the supply source, package, processor pins, etc. For instance, if some cores in a multicore move from idle to active, they will increase switching activity, thus consuming more power. This will produce higher temperature, that will propagate to the neighbor cores, and will reduce the amount of current available for other cores, which will perceive a  $V_{cc}$  decrease. This, ultimately, affects power dissipation dynamically at very fine grain (e.g. voltage variations may occur at the scale of few nanoseconds). As an illustrative example, Figure 1 shows average power measurements of 500-cycle intervals for a program execution in a relevant temperature range for many embedded microcontrollers [24]. A temperature increase of 100

degrees leads to a power increase of up to 3.5x.

#### 4 CURRENT PRACTICE ON PROCESSOR-LEVEL TYPICAL AND MAXIMUM POWER ESTIMATION

As an initial step to define a method to derive reliable WCEC bounds, we describe current practice for low-level processor energy modeling. Arguably, chip vendors have the most advanced techniques and tools for that end. Hence, understanding the limitations of those models is fundamental to understand the limits of WCEC estimation. Note that chip vendors are interested in determining suitable cooling solutions, so their focus is on sustained power estimation under highly stressful scenarios.

Power models and measurements are used to estimate power during processor design [1], [2]. They help iteratively modifying the design until there is enough evidence that target peak power values are not exceeded, see Figure 2. During the process, chip vendors also use techniques such as adaptive body bias<sup>2</sup> [39] to trade off between maximum operating frequency and power dissipation of the processor. Due to the known inaccuracy of the models at the different abstraction levels, safety margins are applied to account for the unknown, such as deviations in the actual switching activity estimated, the impact of PAVT variations or the effectiveness of the cooling solutions [6].

**Models.** Model-based techniques are known for being slow, limiting the *window of analysis* to few thousands of cycles at most. For instance, in electrical-level SPICE models, characterizing a memory macrocell with synthetic stimuli can take days of simulation, with a single BSIM4 CMOS transistor model accounting for more than 40 parameters [43]. On the one hand, the huge time requirements of models are handled by abstracting physical behavior keeping the model usable but reducing its accuracy. On the other hand, despite the complexity of the models, their accuracy w.r.t. reality may not be sufficiently high and, moreover, it is also hard to be estimated. This emanates from the limitations of the model to capture all physical effects and its inability to model *exactly* PAVT variations, often accounted for statistically [7].

Power models are used in chip industry for pre-silicon validation and design refinement (Figure 2), for instance for determining whether the power supply is enough, the appearance of power hot-spots and the efficiency of cooling solutions. Models comprise an analytical part and a wide set of parameters obtained from measurements on ‘prototype implementations’ such as macrocells, small prototype chips, etc. or technology projections derived from previous implementations on similar technology (feature size) [31]. The model is evaluated on small hand-made kernels (power viruses) to derive extreme behavior. However, power viruses do not guarantee that the worst power is captured. This relates to the difficulties to produce those inputs leading to the worst switching across the full chip, under the worst PAVT variations conditions. Identifying the sequence of inputs needed for each Functional Unit Block (FUB) of the processor is simply unaffordable. Then, producing those inputs *simultaneously* in all FUBs is more challenging requiring controllability to produce the worst combined inputs and preventing to control PAVT variations.

**Measurements.** Measuring actual power consumption in real processors is limited by the availability of power monitoring units. The granularity at which power readings can be provided is coarse

2. Body bias techniques rely on modifying the voltage of the substrate to either increase threshold voltage ( $V_t$ ) so that leakage power and speed decrease; or to decrease  $V_t$  causing an increase in speed and leakage power.

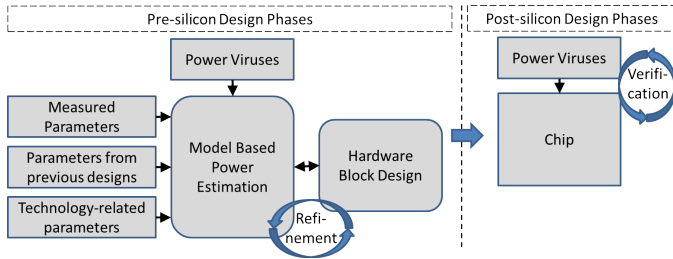


Fig. 2. Usage of (measurement/analytical) models and measurements during the hardware design process.

in time (e.g. 1 second [26]) and space, e.g. components in the pipeline can neither be isolated nor accessed physically to measure their power dissipation. As a result, engineers stick to external means to take coarse-grain power measurements. Interestingly, while some processors provide built-in power monitors for some components, those are power-proxy approaches with which power is derived as a linear model of performance monitoring counters (activities), which are weighted by constants. Those constants are derived empirically with a regression model from the execution of several reference applications. This is the case of the IBM POWER7 [21].

Measurements are used for post-silicon validation (Figure 2). Due to the complexity of achieving accurate power estimates analytically, chip vendors verify chip power using actual measurements – despite their own limitations. This allows deriving power and energy figures for the different processor components. The main challenge for deriving worst-case energy and power measurements resides on the definition of representative scenarios. For example, maximum peak power numbers for processors are obtained using benchmarks that generate the most (*expected*) stressing situations a.k.a. power viruses [22].

Despite advanced models and measurement approaches, the risk of inaccuracies is not removed. One of the most well-known failures in the prediction of the peak/typical power, is the Intel Tejas processor (a.k.a. Pentium V), which finally exceeded its power/temperature budgets due to model inaccuracy at a level that even body biasing could not correct, so its production was abandoned [14]. Although these practices are costly and not always effective, they are still affordable and used in practice by experts due to being the most accurate methods available.

**Summary.** Overall, model-based approaches build on detailed knowledge of the system. The applicability of this type of white-box approaches is challenged by the lack of details of real processors. In contrast, measurement-based approaches, a form of black-box approach, can still derive estimates through experimentation although uncertainty may remain due to the difficulties to create representative tests.

## 5 MODEL-BASED TASK-LEVEL WCEC

Intuitive solutions based on multiplying the average power consumption and the WCET estimate for a task may not lead to high-quality WCEC estimates since energy and time do not necessarily correlate [25]. Few works address the problem of WCEC estimation from an analytical point of view [25], [40]. Following the principles of static WCET analysis, model-based (static) WCEC analysis builds on deriving a cost function for each instruction, with the (obvious) observation that the latter

uses energy as cost function. Energy cost is derived at instruction level and then combined to derive energy cost of basic blocks. From that point on, standard Integer Linear Programming (ILP) formulation – or any other sound formulation – is used to derive WCEC estimates for the task.

WCEC techniques work at a high abstraction level compared to what we discussed in the previous section. Those techniques focus on pipeline effects (Fetch, Decode, etc.) and hardware components used in each stage (e.g. caches, functional units and the like). As reference figures to compare against, WCEC models use estimates provided by open-source power models. Those models are generic, i.e. not tailored to any particular processor, and can indistinctly result in over- or under-estimates. Hence, obtaining WCEC estimates above the estimates provided by the reference model does not guarantee high-quality WCEC estimates as they can be lower or far higher than the actual energy figures.

### 5.1 Granularity and Accuracy

There are several levels at which power can be modeled, such as (in increasing order of abstraction) electrical (e.g. SPICE models), gate level and register transfer level (RTL). At the highest levels, small programs are used to derive power estimates for a given hardware component. These programs are usually restricted to small power viruses [22] that aim at generating high power consumption by, for instance, increasing the activity factor.

Modeling full-program energy consumption poses many challenges. One of them relates to keeping the execution time requirements affordable, which inevitably results in simplifying the underlying power model. In particular, the number of physical details factored in is reduced, which basically plays against the accuracy of the power estimates. Model simplifications may cause inaccuracies either under- or over-estimating power. For instance, gate-level or RTL models lose some accuracy and can only be afforded to simulate small programs (e.g. simulating a full processor during several thousands of execution cycles may require several days of simulation). As the complexity of the models decreases to make the problem tractable, information such as the switching activity of the transistors is lost. A feasible approach to increase the granularity minimizing the impact on accuracy would be using measurements coupled with statistical bounding analysis at the desired granularity level as inputs for the models. Following this approach, any implementation-dependent factor is captured by the measurements and upper-bounded by statistical formulation.

### 5.2 Upper-Bounding the Activity Factor

The activity factor (aka switching activity) of a given FUB is a figure in the range 0-1 that describes the fraction of the total capacity of the FUB that switches (and hence consumes dynamic power) in a particular processor cycle. The activity factor plays a key role when estimating dynamic power, see Section 2. Deriving the activity factor for a FUB requires extensive knowledge about the particular transistors (and their geometries) whose inputs change on a FUB input change. First, many processor details are not visible at the software level. For instance, it is inconceivable devising how control signals switch (e.g. to manage queues between pipeline stages) from the abstract analysis of program instructions. Second, this information can only be obtained with transistor-level simulations, which incur huge overheads to enable modeling full programs. Note that chip vendors may not make those details public for competitive reasons. Additionally, it is

TABLE 1

Toggle coverage for different Workloads on a RTL model of the LEON3

<i>IU</i> components	EEMBC AutoBench			Mälardalen WCET	
	rspeed	canldr	ttsprk	matmult	firFn
Fetch	72.58%	72.58%	74.19%	57.26%	58.06%
Decode	70.27%	68.92%	72.30%	63.51%	60.13%
Register access	80.00%	77.88%	79.70%	73.33%	71.82%
Execute	77.78%	76.19%	77.38%	73.54%	72.22%
Memory access	62.43%	60.22%	62.15%	65.19%	74.86%
Exception	66.51%	64.22%	65.82%	76.15%	76.26%
Write back	29.19%	28.57%	28.57%	26.09%	45.96%
Data cache	57.52%	57.21%	57.52%	56.44%	57.06%
Instruction cache	41.32%	41.32%	42.36%	35.33%	41.32%
Register file	92.48%	92.48%	92.48%	87.97%	97.97%
Others	12.38%	10.6%	10.78%	15.32%	16.09%
<b>Total</b>	39.2%	37.8%	38.5%	39.8%	41.6%

simply unaffordable precomputing the energy consumption of all potential input transitions due to computational and storage cost. As an illustrative example, let us assume a particular FUB such as an adder. A 32-bit adder has, at least,  $2^{64}$  different inputs if we ignore control signals, and so there are at least  $2^{128}$  different input transitions possible, each one producing a specific capacity to switch. Identifying the worst possible transition analytically (out of the thousands of transistors) or empirically is beyond the reach of any circuit designer which, at most, can guess what the worst transition is. Hence, the complexity of obtaining and managing such detailed information is beyond the reach of static models.

An intuitive way to handle this, as done in WCET analysis, is making pessimistic assumptions. For instance, switching activity is assumed to be 1 since providing evidence that a lower value is an actual upper bound would resort to unaffordable low-level information/models. However, typical switching activity is largely below 1 due to idle blocks whose inputs do not change in specific cycles, or due to the usual bias of input values operated and stored towards specific values, which lead to very limited switching activity. To provide concrete empirical evidence on this general intuition, we show an example that builds on the so called *toggle factor* in Table 1. It represents the fraction of nodes<sup>3</sup> that have switched at least once in the processor and hence, can be regarded as an upper-bound of the switching activity of a circuit since only a subset of the transistors in the toggled nodes have effectively switched. In particular, we have computed with the QuestaSim RTL simulator the toggle activity factor for several benchmarks executed in an RTL LEON3 processor description. As shown, only around 40% of the nodes toggled, i.e. the activity factor is at most 40% (but typically much lower).

Worst-case assumptions on the activity factor result in remarkably pessimistic estimates. On the previous example, and assuming that half of the transistors switch in a toggled node, the processor could consume 20W, while we would account for 40W assuming the toggle factor, and 100W assuming switching activity 1. Hence, the estimated WCEC may implicitly lead to a power dissipation above the actual capabilities of the processor, reducing its practical use. As explained before, switching activity decreases exponentially (often quadratically) across gate levels, so activity factors of up to 5% are expected for simple circuits [36]. Lower factors are expected for more complex circuits.

3. A node in RTL represents a high number of transistors in the actual circuit.

### 5.3 PAVT Variations

As detailed in Section 3, PAVT variations can produce large power variations across units. Any static WCEC estimation model aiming at providing arguably sound energy upper-bounds – that cannot be exceeded under any circumstance – cannot afford using typical values or values obtained from statistical distributions (e.g. mean plus six sigma). The latter can be probabilistically exceeded and, even if that could occur with a negligible probability, it cannot be proven to be zero. Such a WCEC estimation approach confronts with chip vendors’ current practice: simply deriving the worst possible value is out of the reach of chip manufacturers that, instead of relying on a theoretical value, build upon measurements to determine the parameters of a Gaussian distribution matching best the observed values. Then, an upper-bound value is chosen based on  $N$ -sigma approaches. In other words, industry resorts to measurements to determine bounds to different parameters and use as upper-bound the mean ( $\mu$ ) plus  $N$  times the standard deviation ( $\sigma$ ), where  $N$  is typically in the range 3-6, depending on the exceedance rate that can be afforded for that particular component and metric [30].

Interestingly, even if we assume that the highest observed value is a true upper-bound, in practice not due to the uncertainty brought by test campaigns on specific processor units, the degree of pessimism for power estimation can be huge. For instance, process variations may produce power discrepancies of 3x across processor units [11], voltage variations can produce  $\approx 25\%$  power variations [5], and temperature variations around 3.5x power variations as shown before. Therefore, even neglecting aging variations, PAVT variations in power (and so in energy) can be as significant as 13x if the absolute worst case needs to be accounted for.

## 6 MEASUREMENT-BASED WCEC ESTIMATION

To our knowledge, no measurement-based WCEC estimation technique exists. Next, we detail the main aspects of WCEC estimation for tasks with measurement-based approaches.

### 6.1 Quality of the Measurements

Using the target platform for collecting power measurements offers the advantage of speed and removes discrepancies with reality due to modeling. Furthermore, measurement-based analysis can also handle complex scenarios by mimicking real-world workloads (i.e. multiple tasks running simultaneously) through the use of stressing tests and operation conditions (e.g. high temperature), thus accounting for interactions between tasks by merely executing them together without the need for any detailed model (i.e. a form of black-box approach). Whenever some effects cannot be properly accounted for through measurements, then disabling or enabling some features (e.g. cache partitioning) can limit the complexity of multi-task workload interference. The other side of the coin are the challenges to observe and account for PAVT variations as well as software-dependent (internal) effects.

Regarding observability, while power meters can be used, they may create some effects on the power consumption of the processor due to the coupling of the power supply lines and may have some degree of inaccuracy. Moreover, power meters measure the power of the full processor rather than the power of the task only, so deducing task energy consumption can only be done with separate experiments running and not running the task, but some non-controlled PAVT variations may interfere measurements

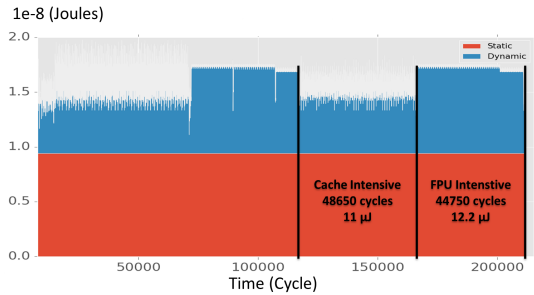


Fig. 3. Average energy consumption of a two-path program

differently across runs. Finally, synchronizing the start of the run of the task with measurement collection is a tough task, so measurements need to be collected at a coarse granularity (e.g. for 1,000 runs of the task with identical inputs) to mitigate this effect.

Regarding PAVT variations, process variations correspond to those of the actual processor being used, so how they represent other processor units can only be studied statistically using other processor units, extrapolating the effect from small-scale experiments on simulated platforms or with data provided by the manufacturer. Analogously, aging, voltage and temperature conditions observed may be representative of neither the typical case nor the worst case. Thus, it may be required to use simulations for extrapolating their typical effect for statistically relevant scenarios.

## 6.2 Input Space Coverage and Representativeness

As for timing analysis, measurement-based WCEC analysis has to deal with all challenges related to input-space coverage such as program path coverage and memory placement of objects (and its influence on cache behavior), both in single-core and multicore execution environments. However, differently to timing analysis, some of these factors have non-obvious effects on power and, moreover, a number of parameters may be innocuous for performance, but not for power.

**Cache** behavior correlates well with energy in general, with hits served faster and with lower energy than misses. The latter need to further access another cache level or memory and take longer to be served. Still, it is possible finding specific examples where hits lead to higher energy consumption than misses.

**Execution Paths.** While path coverage is equally important for both, timing and energy analysis, the challenge for energy relates to the fact that higher execution time does not imply necessarily higher energy consumption. First, there is a direct relation between execution time and energy consumption due to static energy, which is roughly proportional to execution time. Thus, in general, paths with longer execution time will likely produce higher energy consumption, but only if the instruction mix and values operated are similar enough. And second, an execution path that incurs many cache misses may take longer than a computation intensive path. However, the latter may produce much higher switching activity due to computation than the former, where the pipeline stays mostly idle with low switching activity. Figure 3 illustrates this effect by showing dynamic and static energy consumption for 50 execution cycles intervals for a multipath program with 2 paths, each executed twice. On the second iteration of the program, the first cache-intensive path takes 48650 cycles to execute and consumes  $11 \mu J$  while the second computation-intensive path

consumes more energy (i.e.  $12.2 \mu J$ ) and has a shorter duration (i.e. 44750 cycles).

Overall, assessing the relationship between energy and execution time for a given task, or simply identifying the paths leading to the highest energy consumption for a task, is an open challenge. Initial solutions can build on those derived for WCET based on using the input data used for functional testing or some type of randomization to automatically cover the design space and derive probabilistic coverage arguments [27] [4].

**Activity Factor.** The relationship between the activity factor and input data is extremely hard to establish. As indicated before, input values for FUBs may produce high or low switching activity. This often relates to the number of changing bits across operated values, since changing bits may induce some switching activity. However, other effects such as memory placement (and so cache placement), even if performance remains the same, may lead to significantly different switching activities. For instance, different addresses may produce different switching activity when operated to add an offset. Analogously, if two addresses are mapped to the same cache set, even if their accesses produce the same hit/miss sequences, may cause different switching activity in the cache decoders, in the replacement information of the cache sets, etc. Hence, determining a realistic and tight upper-bound to the switching activity of the task under analysis is difficult. Since it depends on highly distant layers (i.e. input data for the task and transistor-level implementation of the processor), no practical means can be realistically set up to get measurable confidence. Instead, only argumentation based on exhaustiveness of test campaigns can be used, whose reliability is difficult to assess.

## 6.3 PAVT Variations

Some variations, such as temperature and voltage, can be induced during analysis by placing the chip in an oven and manipulating the power source of the processor. Yet, relating those conditions with worst-case operation conditions is a complex challenge.

Other variations, such as aging, can be accounted for applying accelerated aging on a processor. This is typically done by applying overly high temperatures and voltages so that the accumulated aging occurred in several years of operation is produced in a few hours. However, whether accelerated aging produces *exactly* the same effects as aging during operation due to physical implications of using different stress conditions is unclear.

Finally, process variations change across processor units, thus making energy estimates obtained for a given chip unit be invalid for any other chip unit. Thus, the only reasonable way to account reliably for the effect of process variations is performing the analysis on the chip to be deployed. This, however, poses a serious issue for many industries: power analysis needs to be repeated for *all* processor units delivered. This is virtually unaffordable for many industries where the number of units can be in the range of millions and cost constraints are severe. Although industry performs a number of verification tests in all units deployed to detect obvious defects, the full validation and verification process followed for certification/qualification purposes is not repeated for each system unit, including all its components. Thus, process variations also bring uncertainty to WCEC estimates. Similar to the model-based approaches, and as stated in section 5, process variation effects can be accounted by analyzing a representative large enough number of processor units and obtaining its corresponding statistical and probabilistic distributions.

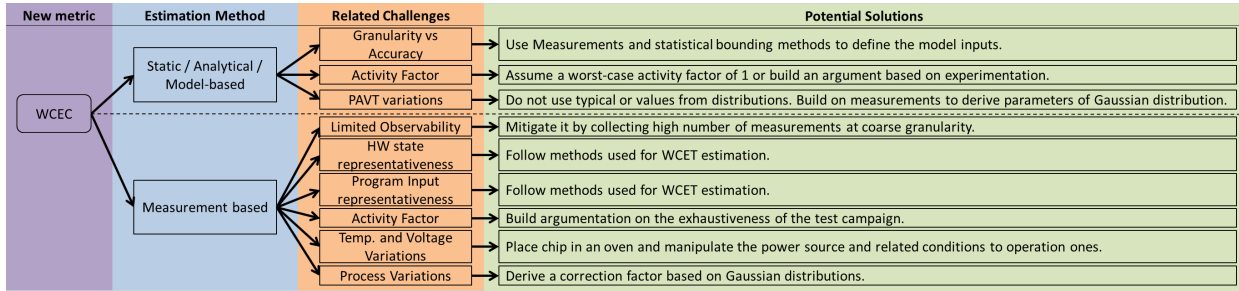


Fig. 4. Diagram of the main challenges, and potential paths to follow, addressed by model-based and measurement-based WCEC estimation

## 7 PUTTING IT ALL TOGETHER

Complex power models have been used for low-level hardware energy modeling, e.g. hardware component, transistor, and capacitor level. Extending these models to derive WCEC estimates at the task level faces the challenges of granularity and precision, see the top part of Figure 4. The former covers the infeasibility of using existing slow models to scale to the size of tasks. The latter covers the fact that abstractions are needed to reduce performance requirements, which naturally cause trading some precision of the models. This translates into making worst-case assumptions for many parameters, resulting in pessimistic estimates that limit their usability and restrict them to early design stages when the objective is to derive initial tasks’ energy/power budgets and task schedules that fit a given energy/power budget.

Measurement-based approaches offer proxies close enough to reality to be usable and to be understood by end users in their certification arguments about tasks worst-case energy consumption. Yet it is required to deal with several sources of uncertainty with qualitative reasoning and statistical methods as the only approaches available to ascertain the degree of uncertainty, see the bottom part of Figure 4. In particular, mechanisms need to be devised to mitigate uncertainty and increase confidence and representativeness of measurements collected: i) for increasing the observability of hardware and software interactions measurements have to comprise a very high number of observations first with identical inputs and later varying inputs to achieve sufficient coverage. Tests have to be intended to enable out-of-normality cases to surface to the observer; ii) for hardware-state and program-input effects, we can build on existing solutions used for WCET either based on randomization as a way to naturally explore complex interactions of software and hardware [27] or techniques based on the user’s ability to build test campaigns able to cover the worst possible situations [4]; iii) the activity factor case builds upon exhaustive tests and a necessary qualitative argumentation to reliably trust those tests; iv) temperature and voltage variations can be accounted by stressing the hardware under analysis by subjecting it to extreme cases of voltage variations or applying accelerated aging; v) finally, process variation uncertainty can be reduced by the use of a large enough test pool of processor units from which to derive statistical distributions that allow the application of a correction factor on the WCEC measurements.

## 8 PAST AND RELATED WORK

Powerful tools exist to measure power at the electrical level, such as SPICE [13], or higher granularities, such as CACTI [38], which models resistances and capacitances of memory structures, and McPAT [28] and WATTCH [8], which estimate the power of

full processors building upon CACTI. Literature on power and energy estimation mostly focuses on empirical regression models, dealing with the selection of the features that should be used to most effectively model energy for different types of platforms or processors (e.g. CPU, GPU, ARM Based, etc) building upon their performance counters [15], [17], [37]. Hybrid models, which combine analytical and empirical models, are also proposed to trade accuracy for microarchitecture independence [18]. Other works approach energy modeling from a probabilistic view by using stochastic models and random distributions [42]. The use of manufacturer-provided models (e.g. Intel’s RAPL) has also been considered and enhanced by several works as an out-of-the-shelf viable accurate alternative [19]. However, none of those tools or models is intended for WCEC estimation of full tasks, as needed in the context of critical real-time systems.

Other works [16] have also verified that variations across identical instances of the same processor are not negligible, which directly impacts empirical models and how to account for the worst-case across a processor pool. Some authors have assessed the strong dependence between WCEC and input values of different components [34]. Others [32] assess the validity of current WCEC methods, showing that WCEC cannot be estimated with mathematical proofs, instead of requiring a shift towards a more statistical framework.

For model-based WCEC techniques, [25] shows that multiplying average power by the WCET is not reliable, so they build upon model-based WCEC estimates for basic blocks extrapolated from micro-architectural level power models and use implicit path enumeration techniques (IPET) to estimate the global WCEC. This approach has been improved [40] [41] combining IPET with genetic algorithms to trade off between reliability and tightness.

We attack the WCEC estimation problem from a different angle. Starting from current industrial practice, we identify the key elements challenging industrially-viable WCEC estimation and provide the basis for a measurement-based probabilistic approach.

## 9 CONCLUSIONS

Energy is a key metric in critical battery-power and power-constrained edge devices, calling for effective means for WCEC estimation. We describe key aspects of WCEC estimation (impact of switching activity and PAVT variations) so far ignored by previous methods. To our knowledge, no previous work covers the increasing gap between WCEC estimation methods and how energy varies in real systems. We make a first step in that direction by bringing together knowledge from industrial practice on energy estimation and WCEC estimation in the embedded domain. Overall, this paper settles the ground on the grand challenges

(and directions to address them) for practical and reliable WCEC estimation and aims at becoming a reference for future works.

## ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under grant TIN2015-65316-P and the HiPEAC Network of Excellence. Jaume Abella has been partially supported by the MINECO under Ramon y Cajal postdoctoral fellowship number RYC-2013-14717. Carles Hernandez is jointly funded by the MINECO and FEDER funds through grant TIN2014-60404-JIN.

## REFERENCES

- [1] Acp - the truth about power consumption starts here. [http://www.amd.com/Documents/43761D-ACP\\_PowerConsumption.pdf](http://www.amd.com/Documents/43761D-ACP_PowerConsumption.pdf), 2011.
- [2] Measuring Processor Power: TDP vs. ACP. <https://www.intel.com/content/dam/doc/white-paper/resources-xeon-measuring-processor-power-paper.pdf>, 2011.
- [3] J. Abella et al. Electromigration for microarchitects. *ACM Comput. Surv.*, 42(2):9:1–9:18, March 2010.
- [4] R. Bertran et al. Voltage noise in multi-core processors: Empirical characterization and optimization opportunities. In *MICRO*, 2014.
- [5] S. Borkar et al. Parameter variations and impact on circuits and microarchitecture. In *DAC*, 2003.
- [6] P. Bose. Pre-silicon modeling and analysis: Impact on real design. *IEEE Micro*, 26(4):3–3, July 2006.
- [7] K. A. Bowman et al. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. *IEEE JSSC*, 37(2), 2002.
- [8] D. Brooks et al. Wattch: a framework for architectural-level power analysis and optimizations. In *ISCA*, 2000.
- [9] J.A. Butts et al. A static power model for architects. In *MICRO*, 2000.
- [10] Certification Authorities Software Team. Multi-core Processors - Position Paper. Technical report, CAST-32A, November 2016.
- [11] S. Chandra et al. Considering process variations during system-level power analysis. In *ISLPED*, 2006.
- [12] K. Chen et al. Reliability effects on mos transistors due to hot-carrier injection. *IEEE JSSC*, 20(1):306–313, Feb 1985.
- [13] EECS Department of the University of California at Berkeley. SPICE website: [www.http://bwrcs.eecs.berkeley.edu/classes/icbook/spice/](http://bwrcs.eecs.berkeley.edu/classes/icbook/spice/).
- [14] EETimes. Intel cancels tejras, moves to dual-core designs, 2004. [http://www.eetimes.com/document.asp?doc\\_id=1150169](http://www.eetimes.com/document.asp?doc_id=1150169).
- [15] Bhavishya Goel et al. A Methodology for Modeling Dynamic and Static Power Consumption for Multicore Processors. In *IPDPS*, 2016.
- [16] Jakim von Kistowski et al. Variations in cpu power consumption. In *ICPE*, 2016.
- [17] Kristoffer Robin Stokke et al. High-Precision Power Modelling of the Tegra K1 Variable SMP Processor Architecture. In *MCSOC*, 2016.
- [18] Sam van den Steen et al. Micro-Architecture Independent Analytical Processor Performance and Power Modeling. In *ISPASS*, 2015.
- [19] Spencer Desrochers et al. A validation of DRAM RAPL Power Measurements. In *MEMSYS*, 2016.
- [20] K. Flautner et al. Drowsy caches: simple techniques for reducing leakage power. In *ISCA*, 2002.
- [21] M.S. Floyd et al. Adaptive energy-management features of the IBM POWER7 chip. *IBM Journal of Research and Development*, 55(3), 2011.
- [22] K. Ganesan et al. System-level max power (sympo) - a systematic approach for escalating system-level power consumption using synthetic benchmarks. In *FACT*, 2010.
- [23] K. Grüttner et al. CONTREX: design of embedded mixed-criticality control systems under consideration of extra-functional properties. *MICPRO*, 2017.
- [24] Infineon. *TC27XDC Family Aurix Microcontrollers Data Sheet*, 2017.
- [25] R. Jayaseelan et al. Estimating the worst-case energy consumption of embedded software. In *RTAS*, 2006.
- [26] V. Jiménez et al. Power and thermal characterization of POWER6 system. In *FACT*, 2010.
- [27] L. Kosmidis et al. PUB: Path upper-bounding for measurement-based probabilistic timing analysis. In *ECRTS*, 2014.
- [28] S. Li et al. Mepat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *MICRO*, 2009.

- [29] E. Macii et al. High-level power modeling, estimation, and optimization. *IEEE TCAD*, 17(11), 1998.
- [30] T. McConaghy. Analog behavior in custom ic variation-aware design. In *ICCAD*, pages 146–148, Nov 2013.
- [31] M. Miyama et al. Pre-silicon parameter generation methodology using bsim3 for circuit performance-oriented device optimization. *IEEE T-SM*, 14(2):134–142, May 2001.
- [32] J. Morse et al. On the infeasibility of analysing worst-case dynamic energy. In *arXiv preprint arXiv:1603.02580*, 2016.
- [33] U.S. Department of Health, Human Services, Food, and Drug Administration. General principles of software validation; final guidance for industry and FDA staff, 2012.
- [34] J. Pallister et al. Data dependent energy modeling for worst case energy consumption analysis. In *arXiv preprint arXiv:1505.03374*, 2015.
- [35] D. K. Schroder. Negative bias temperature instability: What do we understand? *Microelectronics Reliability*, 47(6):841 – 852, 2007. Modelling the Negative Bias Temperature Instability.
- [36] D. Singh et al. Power conscious cad tools and methodologies: a perspective. *Proceedings of the IEEE*, 83(4):570–594, Apr 1995.
- [37] Sriram Sankaran. Predictive modeling based power estimation for embedded multicore systems. In *CF*, 2016.
- [38] S. Thoziyoor et al. Cacti 5.1 - HP labs technical report. 2008.
- [39] J. W. Tschanz et al. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. *IEEE JSSC*, 37(11), 2002.
- [40] P. Wägemann et al. Worst-case energy consumption analysis for energy-constrained embedded systems. In *ECRTS*, 2015.
- [41] P. Wägemann et al. Whole-system WCEC analysis for energy-constrained real-time systems. 2018.
- [42] Walteneus Dargie. A Stochastic Model for Estimating the Power Consumption of a Processor. In *IEEE TC*, 2015.
- [43] Xi X. et al. *BSIM4.3.0 MOSFET Model - User's Manual*. 01 2003.



**David Trilla** is a PhD. Student for the CAOS group at BSC. He obtained his M.S. degree in 2016 and graduated in Informatics Engineering in 2014, both titles obtained from the Universitat Politècnica de Catalunya. He enrolled BSC in 2014 and has participated in the European project ESA-HAIR. His current research focuses on the effects of randomized architectures on energy, security and reliability.



**Carles Hernandez** received the PhD in computer sciences from Universitat Politècnica de Valencia in 2012. He is currently senior PhD. Researcher at BSC. His area of expertise includes time-predictable and reliability-aware processor design. He participates (has participated) in NaNoC, parMERASA, PROXIMA IP7, VeTeSS ARTEMIS, and RECIPE H2020 projects. He has published more than 50 papers in top international conferences and journals.



**Jaume Abella** is a senior PhD. Researcher at BSC and HiPEAC member. He received his MS (2002) and PhD. (2005) degrees from the UPC. He worked at the Intel Barcelona Research Center (2005-2009) in microarchitectures for fault-tolerance and low power. He joined the BSC in 2009 where he has been the BSC PI for several projects with industry and academia. He has authored more than 15 patents and 100 papers in top conferences and journals. He is (has been) co-advisor of ten MS and PhD students.



**Francisco J. Cazorla** is the leader of the CAOS group at BSC. He has led research projects funded by industry (IBM and Sun Microsystems), the European Space Agency (ESA) and public-funded projects ( PROARTIS project and FP7 PROXIMA project). His research area focuses on high-performance and real-time systems. He has co-authored 3 patents and over 100 papers in international refereed conferences/journals.