

# Thermally-Aware Composite Run-Time CPU Power Models

Matthew J. Walker\*, Stephan Diestelhorst†, Andreas Hansson†, Domenico Balsamo\*, Geoff V. Merrett\* and Bashir M. Al-Hashimi\*

\*University of Southampton  
Southampton, UK

{mw9g09, db2a12, gvm, bmah}@ecs.soton.ac.uk

†ARM Ltd.  
Cambridge, UK

firstname.surname@arm.com

**Abstract**—Accurate and stable CPU power modelling is fundamental in modern system-on-chips (SoCs) for two main reasons: 1) they enable significant online energy savings by providing a run-time manager with reliable power consumption data for controlling CPU energy-saving techniques; 2) they can be used as accurate and trusted reference models for system design and exploration. We begin by showing the limitations in typical performance monitoring counter (PMC) based power modelling approaches and illustrate how an improved model formulation results in a more stable model that efficiently captures relationships between the input variables and the power consumption. Using this as a solid foundation, we present a methodology for adding thermal-awareness and analytically decomposing the power into its constituting parts. We develop and validate our methodology using data recorded from a quad-core ARM Cortex-A15 mobile CPU and we achieve an average prediction error of 3.7% across 39 diverse workloads, 8 Dynamic Voltage-Frequency Scaling (DVFS) levels and with a CPU temperature ranging from 31°C to 91°C. Moreover, we measure the effect of switching cores offline and decompose the existing power model to estimate the static power of each CPU and L2 cache, the dynamic power due to constant background (BG) switching, and the dynamic power caused by the activity of each CPU individually. Finally, we provide our model equations and software tools for implementing in a run-time manager or for using with an architectural simulator, such as gem5.

## I. INTRODUCTION

While mobile devices are required to be ever-more energy efficient, mobile CPU designs are becoming more complex in order to achieve the ever-increasing performance demand of new applications. Key to saving energy is the system’s run-time manager (RTM), which controls CPU energy-saving techniques (such as dynamic voltage frequency scaling [DVFS] or heterogeneous cores [e.g. ARM’s big.LITTLE technology]) in order to trade off power and performance based on the current conditions and requirements. However, in order to make these trade-offs effectively, run-time knowledge of how power is being consumed is essential. A fast and accurate power model, that is able to predict the current power consumption at run-time, can therefore be used with a run-time manager to make significant energy savings [1].

Previously, top-down regression-based models using performance monitoring counters (PMCs) as inputs have been widely shown to be effective in estimating CPU power [2]–[13]. Top-down approaches use power measurements from real devices and predict this measured power using metrics. Therefore they

can potentially be very accurate and this accuracy can be validated. However, they are only valid for the specific device implementation they were developed on.

Saving energy in modern CPUs is a key current research topic, much of which is conducted using an architectural simulator, such as gem5 [14], in conjunction with a *bottom-up* power simulator, such as McPAT [15], which uses theoretical knowledge to estimate the power of each component. Bottom-up power models are adaptable to any design specification but, as a result, suffer from large errors, largely due to abstraction and specification errors [16]–[18]. Obtaining trustworthy simulation results is a key challenge in research and using tools without understanding their limitations can lead to incorrect research conclusions. While the flexibility of bottom-up tools is sometimes required, there are many cases where a top-down model (built and validated on a real hardware platform) can be used to provide an accurate and trusted reference, as highlighted in [13].

There is a lack of existing work in *top-down* power modelling that utilises measured PMC-data combined with power measurements collected on real mobile devices, largely due to technical challenges involved in doing so. A recent work [13] overcomes this and significantly improves on the state-of-the-art by presenting a statistically-rigorous methodology that ensures stability, deals with heteroscedasticity and achieves a low error on two mobile CPUs of differing microarchitecture. However, the effects of temperature on power consumption are not considered and the model could be improved by further decomposing the estimated power consumption, and therefore providing more information on *how* the power is being consumed.

We use the methodology in [13] as a starting point and model the effects of temperature on the CPU power consumption and add thermal compensation to the models. We then measure the effects of switching cores offline and then analytically decompose the power consumption.

Our proposed model significantly improves the accuracy across wide temperature ranges, results in a more stable model with statistically significant input components, and gives a breakdown of the power consumption. This provides higher quality online data to a run-time manager. Moreover, in research and design-space exploration, the power breakdown

allows this accurate and validated top-down model to be combined with bottom-up techniques, allowing some flexibility in model specification. Using a combination of the two, as opposed to relying fully on a bottom-up approach, means that more confidence can be placed in the resulting estimations.

This work focusses on mobile CPUs due to the significant importance of energy-efficiency in mobile applications and because the diversity of mobile workloads makes power modelling particularly difficult. However, the presented methodology is generic and applicable to other systems (e.g. desktop and server CPUs). The resulting models themselves accurately model the specific CPU implementation they were built on.

The key contributions of this paper are:

- Temperature compensated accurate and stable run-time CPU power models, built and validated using data measured from a real device;
- Decomposition of the state-of-the-art PMC-based power models to give static and dynamic power estimations for individual components.
- An analysis of the effect of temperature on the static power of a real-world mobile CPU;
- Evaluation of the effects of switching cores offline.

We describe our experimental setup in Section II. We then look closely at the typical modelling approaches presented in related work, explain the problems with these approaches using an example, and describe the benefits of our approach in Section III. Section IV details our methodology of analysing how thermal behaviour affects the power consumption and how we make our models thermally-aware. We then describe how we are able to decompose the estimated power consumption accurately into its constituting parts in Section V.

## II. EXPERIMENTAL SETUP

We use the ODROID-XU3 development board by Hardkernel, which contains an Exynos-5422 SoC (System-on-Chip) featuring a quad-core ARM Cortex-A7 and quad-core ARM Cortex-A15 CPU. These two clusters utilise the same ISA (instruction set architecture) but have differing microarchitectures to achieve different energy-performance trade-offs. The ODROID-XU3 board contains built-in power sensors to give power measurements of each of these two clusters, as well as the DRAM memory and the GPU. It also contains four CPU temperature sensors and one GPU temperature sensor.

In this work we use the higher-performance Cortex-A15 cluster to illustrate our approach. The Cortex-A15 CPUs each have 32 KB instruction and data caches and they share a 2 MB L2 cache (Fig. 1). Each CPU contains a NEON SIMD processing unit, which is accounted for in the power models. Unlike the older ODROID-XU board, the ODROID-XU3 board supports global task scheduling (GTS), also known as big.LITTLE MP, which allows all of the 8 cores, whether ‘big’ Cortex-A15s or ‘LITTLE’ Cortex-A7s, to be available to the operating system’s scheduler simultaneously. The maximum clock rate of the Cortex-A15 cluster is 2 GHz and the device is implemented in a 32 nm Low-Power High-K Metal Gate (HKMG) technology. The SoC also contains an ARM Mali-T626 MP6 GPU and 2 GB LPDDR3 RAM that has a maximum bandwidth of 14.9 GB/s.

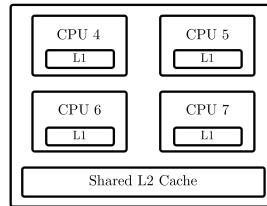


Fig. 1. Cortex-A15 quad-core cluster

This is the same platform as used by [13] and we use their provided experimental platform software and tools (available at [19]) to collect experimental data. We run workloads on the board while varying the clock frequency and recording the PMCs, power, voltage and temperature.

We use 39 diverse workloads from several benchmarking suites: MiBench [20], which is a suite of representative embedded workloads; LMBench [21], which contains microbenchmarks for activating and testing specific microarchitectural behaviours, such as memory reads at a specific level of cache; Roy Longbottom [22], which contains many multi-threaded workloads that make heavy use of the NEON SIMD processing unit and OpenMP; ParMiBench [23], which is a multi-threaded version of MiBench; and, ALPBench [24], which contains complex, parallel multimedia workloads.

Once the experimental data has been collected, we use the model building software tools (also available at [19]) to implement their automated methodology. We use the resulting models as a starting point to our work.

## III. REGRESSION-BASED MODELLING METHODOLOGY

We create a linear regression model, using an ordinary least squares (OLS) estimator to estimate its coefficients. In existing works on PMC-based run-time power models, the independent variables (i.e. PMC events and sometimes the CPU voltage ( $V_{DD}$ ), the clock frequency ( $f_{clk}$  and/or the temperature,  $T$ ) are inserted directly into a regression tool [9], [10], [25]–[27]; the relationships between the variables and power consumption are not considered. A recent work [27], which evaluates the state-of-the-art and typifies the general approach, proposes the model equation shown in Equation 1. There are several problems with this formulation, e.g. the  $V_{DD}$  is included but the dynamic power consumption is known to be proportional to  $V_{DD}^2 f_{clk}$ . Moreover, this model attempts to include temperature ( $T$ ) compensation, but the actual relationship between power and temperature has not been established; simply inserting a variable in the model does not mean that the regression analysis will be able to resolve the relationship between it and power. Later in this paper we will show, for example, that there needs to be terms in the equation proportional to  $T^2$  (note that other terms related to  $V_{DD}$ ,  $f_{clk}$  and  $T$  are not included in Equation 1; the terms not shown include PMC events and operating system statistics). This method results in poor models as the relationships between power and the available input variables have not been correctly captured. Additionally, as in the example shown in Equation 1, there is a tendency to specify many independent variables

(often providing similar information) in an attempt to improve the accuracy of the mis-specified model, resulting in instability due to large errors in the model coefficients [13].

$$P = const. + \beta_1 V_{DD} + \beta_2 f_{clk} + \beta_3 T + \beta_4 IPC + \beta_5 \frac{INT}{N} + \beta_6 \frac{VPF}{n} + \dots + \beta_{15} SoftIRQ \quad (1)$$

We use the model formulation presented in [13] (including their selection of PMC events) as our starting point (Equation 2), where  $N$  is the total number of PMC events in the model;  $n$  is the index of each event;  $E$  is the cluster-wide PMC event rate (events-per-second) after being divided by the operating frequency in MHz,  $f_{clk}$ , and averaged across all cores; and  $V_{DD}$  is the cluster operating voltage.  $P_{cluster}$  is the power for the overall quad-core Cortex-A15 cluster. This model formula breaks down the power consumed by the dynamic CPU activity and the idle power (which includes the static power and background (BG) switching power, and hence includes the  $f_{clk}$  term).

$$P_{cluster} = \underbrace{\left( \sum_{n=0}^{N-1} \beta_n E_n V_{DD}^2 f_{clk} \right)}_{\text{dynamic activity}} + \underbrace{f(V_{DD}, f_{clk})}_{\text{static and BG dynamic}} \quad (2)$$

Such a model formulation is built by understanding the relationships between the variables. For example, a known component of CMOS power consumption is the dynamic power caused by switching activity, which is proportional to  $V_{DD}^2 f_{clk}$ . Therefore, the PMC events, which indicate dynamic activity, should only be included in the dynamic power calculation and be inserted into the model after being multiplied by  $V_{DD} f_{clk}$  (dynamic activity, Equation 2).

This approach has several benefits: it allows the power consumption to be broken down; results in a more stable model with more accurate coefficients; results in a model with more physical meaning; and allows relationships and power contributions to be deduced.

For example, we can show that there is a constant dynamic power component (BG switching) present by building the model with and without the constant  $V_{DD}^2 f_{clk}$  (BG dynamic, Equation 3). When this component is added, the accuracy of the model increases significantly and the corresponding p-value of its coefficient is very low ( $p < 0.0001$ ), indicating strong statistical significance. We can therefore infer that there is a constant dynamic power component (i.e. present even when there is no activity on the cluster) and its magnitude can be estimated. The equation can therefore be rewritten as shown in Equation 3. Later in this paper we show how we can use this to decompose the power model further and understand the L2 cache behaviour when all cores are switched offline. This is not possible with the typical approaches.

$$P_{cluster} = \underbrace{\left( \sum_{n=0}^{N-1} \beta_n E_n V_{DD}^2 f_{clk} \right)}_{\text{dynamic activity}} + \underbrace{\beta_b V_{DD}^2 f_{clk}}_{\text{BG dynamic}} + \underbrace{f(V_{DD})}_{\text{static}} \quad (3)$$

#### IV. THERMAL MODELLING

This section describes how we extend the power models to make use of the instantaneous temperature readings from the on-board thermal sensors to significantly improve the power prediction accuracy.

The static power consumption of CMOS devices is known to be highly dependent on the temperature. The static power of a CMOS device can be calculated using Equation 4 [28]. The two key components of leakage current are sub-threshold and gate-oxide leakage (Equation 5). The thermal voltage ( $V_\theta$ ) increases linearly with temperature and the sub-threshold current therefore has a strong temperature dependence (Equation 6). There is a potential for thermal runaway because the device will heat up if the sub-threshold current increases. The gate-oxide has a smaller, dependence on temperature [28].

$$V_{static} = I_{leak} V \quad (4)$$

$$I_{leak} = I_{sub} + I_{ox} \quad (5)$$

$$I_{sub} = K_1 W e^{-\frac{V_{th}}{n V_\theta}} \left( 1 - e^{-\frac{V}{V_\theta}} \right) \quad (6)$$

The models built in [13] do not include any temperature compensation, despite its large effect on the static power consumption. The temperature of the device is mainly influenced by the CPU voltage ( $V_{DD}$ ), the activity/load of the CPU, and the ambient temperature conditions. A relatively accurate model is achieved in this previous work by including higher order terms related to  $V$  in the static power equation (e.g.  $V^3$ ) to absorb the error in the static power estimation caused by the temperature changing due to the DVFS level. It does not absorb the error caused by temperature changes due to the ambient temperature or CPU load. In this section we use recorded thermal data to build a fully thermal-aware model that has superior accuracy, even with extreme ambient temperature differences. As well as the accuracy being significantly improved, this temperature-dependent model is better-suited for coming with bottom-up approaches and applying theoretical equations to further decompose the model.

We run three experiments: one with the default fan settings on the ODROID board; one with the fan switched to its maximum speed; and one with the fan switched off completely. All experiments are conducted under the same ambient conditions and run 39 different workloads at clock frequencies from 200 MHz to 1600 MHz in steps of 200 MHz.

We then build two models (labelled *a* and *b*, Table I) using the methodology and software tools presented in [13]; the first one using data collected at normal ambient conditions and built using model equation 2, and the second one using data from all three experiments (varying temperature,  $V$ ,  $T$ ) and model equation 3.

When observations taken with different fan settings are used with a model without thermal compensation (model *b*), the average error increases to 6%. This average value is optimistic as many of the observations have similar temperatures, particularly at lower frequencies. However, for observations

TABLE I  
COMPARISON OF THREE MODELS USING DIFFERENT MODEL EQUATIONS  
AND UNDER DIFFERENT THERMAL CONDITIONS.

	Model Eqn.	V, T	Avg. Error (%)	Adj. $R^2$	SER (W)
a	2	N	3.34223	0.997691	0.0423673
b	3	Y	5.97732	0.994498	0.0604720
c	8	Y	3.67805	0.997418	0.0414211

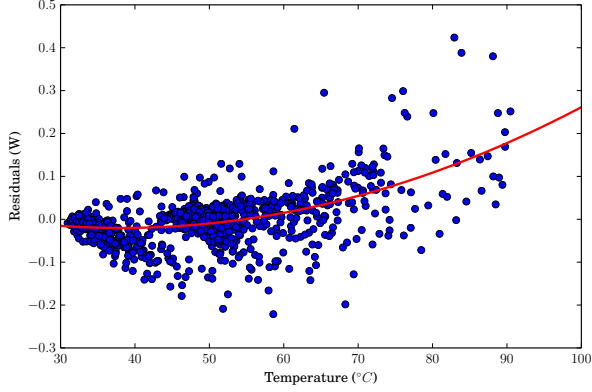


Fig. 2. Residuals of model *b* plotted against temperature. Red line shows an equation predicting this trend.

with significantly varying temperature, the errors can be large. For example, when idle at 1600 MHz, the three experiments (automatic fan setting, fan on, fan off), which have average temperatures of 43.5°, 40.2° and 73.5°, respectively, achieve average errors of 1.6%, 2.7% and 25%; despite the relatively low average error, individual observations with significant temperature deviations can have large errors. Our thermally-aware power model, which we present later in this section achieves errors of 1.2%, 0.96%, and 5.0% for these observations, respectively, therefore improving the accuracy by as much as 20%.

This dependence on temperature needs to be approximated in the regression model and the static power model therefore needs to take temperature,  $T$ , as an input (Equation 8). We plot the residuals (the difference between the observed value and the estimated value) over temperature and identify that there is a clear relationship that is not being captured in the current model (Fig. 2). We use regression analysis to predict the residuals from the temperature and find that Equation 7 fits optimally. A  $T$  and  $T^2$  term therefore should to be added to the power model.

$$Residuals = \alpha_a T^2 + \alpha_b T + c \quad (7)$$

$$P_{cluster} = \underbrace{\left( \sum_{n=0}^{N-1} \beta_n E_n V_{DD}^2 f_{clk} \right)}_{\text{dynamic activity}} + \underbrace{\beta_b V_{DD}^2 f_{clk}}_{\text{BG dynamic}} + \underbrace{f(V_{DD}, T)}_{\text{static}} \quad (8)$$

As previously stated, the model created in [13] does not use temperature as an input but *does* indirectly compensate

TABLE II  
CORTEX-A15 MODEL COEFFICIENTS AND P-VALUES, GROUPED INTO  
THREE COMPONENTS: DYNAMIC ACTIVITY (DYN. ACT.) POWER;  
CONSTANT BACKGROUND DYNAMIC POWER (BG DYN.); AND STATIC  
POWER.

Comp.	Coefficient	Weight	p-value
Dyn. act.	$0x11 \times V^2 f$	5.966e-10	$p < 0.0001$
Dyn. act.	$0x1b-0x73 \times V^2 f$	8.885e-10	$p < 0.0001$
Dyn. act.	$0x50 \times V^2 f$	1.091e-8	$p < 0.0001$
Dyn. act.	$0x6a \times V^2 f$	1.622e-8	$p < 0.0001$
Dyn. act.	$0x73 \times V^2 f$	3.445e-10	$p < 0.0001$
Dyn. act.	$0x14 \times V^2 f$	4.413e-10	$p < 0.0001$
Dyn. act.	$0x19 \times V^2 f$	2.900e-9	$p < 0.0001$
BG Dyn.	$V^2 f$	1.508e-4	$p < 0.0001$
Static	Intercept	-1.785e+0	$p < 0.001$
Static	$VT^2$	8.894e-4	$p < 0.0002$
Static	$VT$	-6.877e-2	$p < 0.003$
Static	$V$	1.850e+0	$p < 0.0009$
Static	$T^2$	-8.592e-4	$p < 0.0003$
Static	$T$	6.807e-2	$p < 0.004$

partially for temperature. As  $V_{DD}$  increases, the temperature,  $T$  also increases. The model in [13] includes  $V^2$  and  $V^3$  terms in the static power equation to absorb the temperature effects due to the cluster voltage. An important step is to remove all the terms that absorb these temperature effects before adding the new temperature compensation. Not doing so would result in a model with too many inputs and the same effect being captured by a combination of several inputs. This would result in a less stable model.

We remove all of the  $V^2$  and  $V^3$  components from the existing model and add our temperature compensation. The static power equation now only contains a  $V$  component,  $T^2$  component, a  $T$  component, and an intercept (Table II). A lower apparent average error can be achieved by including every combination of these three components into the model. However, doing so actually reduced the quality of the model and it over-fits mechanisms within the model. We carefully select six terms (including the intercept) to predict the static power consumption, all of which have an associated p-value of less than 0.003 indicating that they are all statistically significant.

Our resulting temperature-compensated model equation (Model *c*, Table I) achieves a 10-fold cross-validated mean absolute percentage error of 3.7% with observation temperatures as low as 30.9°C and as high as 90.6°C.

We re-draw the same graph showing the residuals against temperature and observe that there is no longer a trend between the two, confirming that the model has successfully captured the effect of temperature on the power consumption (Fig. 3). The cone shape of the residuals indicates the presence of heteroscedasticity (non-constance variance), which is inherent to PMC-based power modelling. We address this problem by using a heteroscedasticity-consistent standard error (HCSE) estimator as described in [13].

Figure 4 shows how the modelled static power varies with

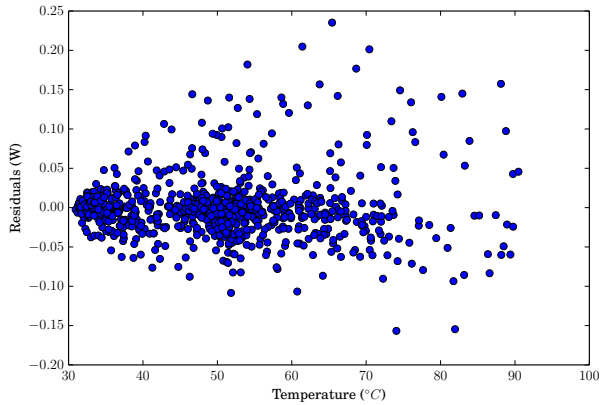


Fig. 3. Residuals of model  $c$  (temperature-compensated) plotted against temperature. Red line shows an equation predicting this trend.

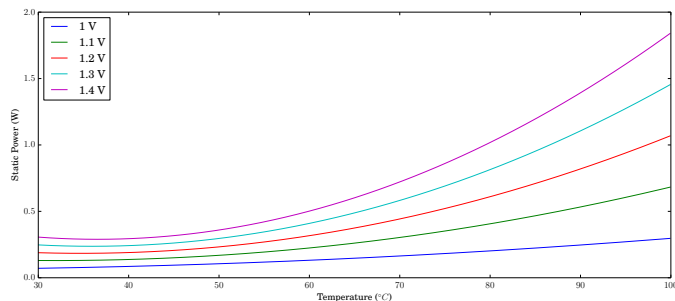


Fig. 4. Model output response when varying the temperature input at various cluster voltage points.

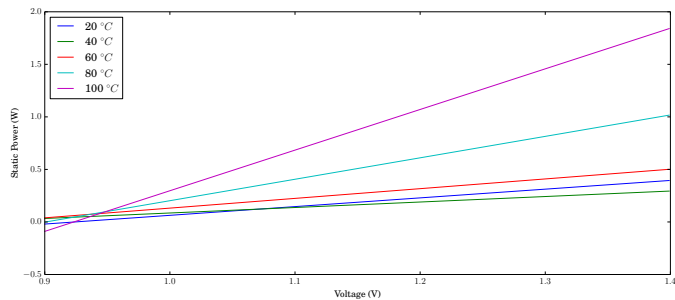


Fig. 5. Model output response when varying the voltage input at various temperature points.

temperature at several fixed voltages, while Figure 5 shows how it varies with voltage at several fixed temperatures. Later in this paper we will show how temperature impacts the static power for measured observations.

## V. MODEL DECOMPOSITION AND OFFLINE CORES

In the previous section we have created a thermally-aware model that estimates the overall power of a quad-core ARM Cortex-A15 cluster. In this section, we decompose the model, allowing it to give accurate estimation of the static and dynamic power consumed by each core and to predict the power consumption when cores are switched offline.

In [13] the PMC events used to estimate the dynamic activity power consumption are averaged across all four cores in the cluster. Therefore, the assumption is that the dynamic power impact of a PMC event is the same, whether, say, 95% of the cluster-wide event come from one core and 5% from the rest, or each core contributes to 25%. We confirm this assumption is correct by estimating the dynamic activity power of each core individually and comparing this directly with the measured power consumption. The modelling methodology provided by [13] calculates the arithmetic mean of the PMC events across all four cores and uses this data to calculate the model coefficients. We therefore take the counters from each core and divide them by four, before multiplying each event by the calculated coefficient. This per-CPU estimated dynamic activity power consumption includes the dynamic activity of the CPU itself and the dynamic power of the L2 cache caused by that CPU.

We measure the quad-core Cortex-A15 power when it is idle and switch off each core one-by-one to observe how static power is consumed (Fig. 6). We conduct this experiment at every DVFS level to see how it varies with voltage (plotted in red) and frequency (x-axis). The blue line shows the measured cluster power consumption when all four cores are online but idle. The purple line shows the power when only one core is online and the brown line shows the power when all cores are switched offline.

The power decreases significantly when the last core is switched off. At first glance one may assume that this may be because the shared 2 MB L2 cache is switched off at this point. However, we observe that, even when the voltage is constant (e.g. 200 MHz-800 MHz), the power (when more than 0 cores are online) increases with frequency, showing that there is a component of dynamic power. As shown in Section III, a constant dynamic component (BG dynamic) is present. This component occurs even when there is no activity (e.g. the system is idle) and it is therefore present in Fig. 6. We calculate this component using  $V_{DD}^2$  and  $f_{clk}$  as well as the static power of the final online core (by subtracting the average power between  $n=4$ ,  $n=3$ ,  $n=2$  and  $n=1$  from  $n=1$ ). We find that the magnitude of these two components at each frequency fit the gap between the measured power when all four cores are off and  $n=1$ . The difference between  $n=1$  and  $n=0$  is therefore due to the BG dynamic power. We can therefore infer that the drop in power when the last core is switched off is due to a constant dynamic power (BG dynamic) component disappearing (possibly due to clock gating in the L2 cache and bus) and that the L2 cache remains powered but inactive, as there is a significant static power component remaining.

We can therefore break this idle power consumption into the following:

- static power of the L2 cache and cluster-wide logic;
- constant background (BG) dynamic power which is present when one or more cores is online;
- static power of each of the four cores and their respective L1 caches.

Unlike with the dynamic power consumption, the proportion of static power consumed by each component remains

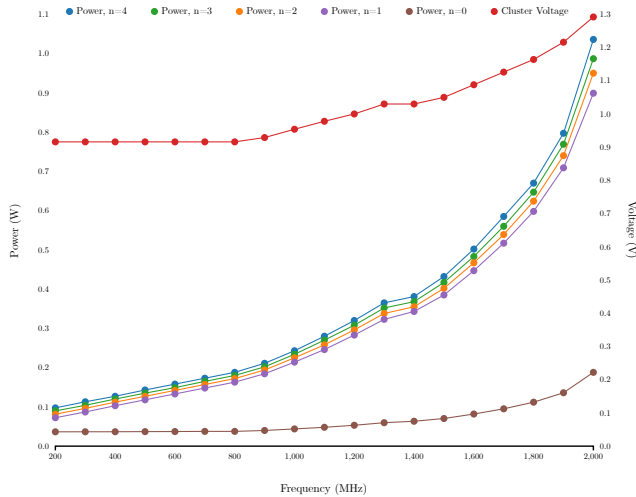


Fig. 6. The measured idle quad-core Cortex-A15 cluster power at various clock frequencies ( $f_{clk}$ ) when all four cores are online ( $n=4$ ) and switching each off in turn until no core is online ( $n=0$ ). The cluster voltage is also shown.

constant, with respect to the total static power, across each DVFS level. Each core and its L1 cache consumes 11.8% and the remaining static power consumption, consisting largely of the L2 cache, makes up 52.6%.

We can therefore break down our static power estimation, which takes both voltage and temperature into account. The base static power (mostly consisting of the L2 cache static power) is calculated using the average temperature value recorded from all four thermal sensors, while the static power for each CPU is calculated using its respective thermal sensor.

With this idle power breakdown added to our model, we can see the static power breakdown across different DVFS points when the fan was manually switched on (Fig. 7) and when the fan was permanently off (Fig. 8).

Note how the static power remains almost constant at 400 MHz, 600 MHz and 800 MHz whereas BG dynamic power increases as it is proportional to the clock frequency. There is a small increase in static power from 400 MHz to 800 MHz due to the temperature increasing with voltage.

At 1600 MHz with the fan on, the static power consumption is less than 0.14 W (temperature: 40.2 °C) and with the fan off, the static power consumption is over 0.33 W (temperature: 73.5 °C). This illustrates the importance of considering temperature within a model, particularly if the model is providing per-component static and dynamic power predictions. When summing the individual components of static power and the BG dynamic component, our model achieves an average idle power prediction error of less than 7% across the full range of temperatures considered and every DVFS level.

We calculate each component of power and sum them to obtain the overall cluster power and compare it to the measured cluster power and, as previously reported, obtain an average error of 3.7%. Fig. 9 shows the error of each individual workload (aggregated over DVFS levels and fan settings). We also plot the power consumption and its breakdown of each workload, again aggregated over every DVFS level and

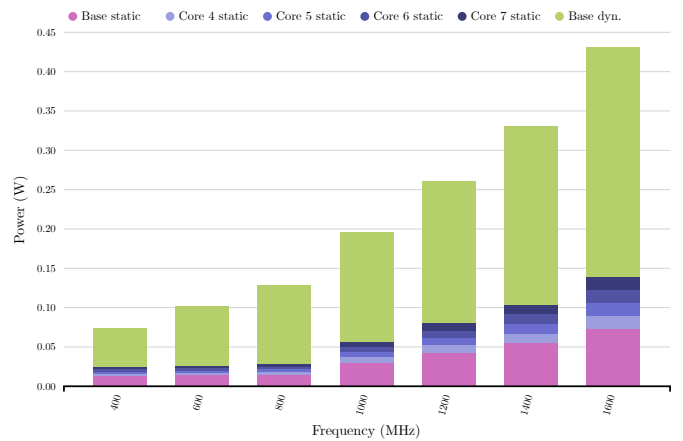


Fig. 7. Idle power consumption across different clock frequencies, showing the contribution from the static power and BG dynamic power. Fan switched on. Average temperature is 35.5 °C.

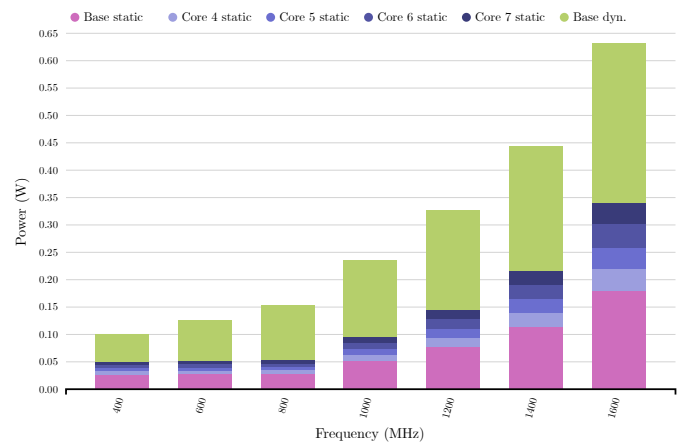


Fig. 8. Idle power consumption across different clock frequencies, showing the contribution from the static power and BG dynamic power. Fan switched off. Average temperature is 60.1 °C.

fan setting (Fig. 10). The static power varies across different workloads because the intensity of the workloads affect the temperature. It also varies because the voltage provided by the non-ideal voltage regulator changes with the dynamic CPU current draw. This effect was identified and addressed in [13].

## VI. CONCLUSION

We present our methodology of developing accurate and decomposable thermally-aware run-time power models which have uses in both online energy management and design-space exploration. We illustrate our approach using measured data from a mobile device utilising an ARM Cortex-A15 quad-core CPU. We demonstrate how we are able to build stable models that can efficiently capture the relationships between the model inputs and power, allowing us to decompose the power model into smaller components, by significantly improving on the model formulation presented in existing works. We analyse the effect of temperature on the static power consumption and add thermal compensation to our power models and show how it improves the accuracy by as much as 20%.

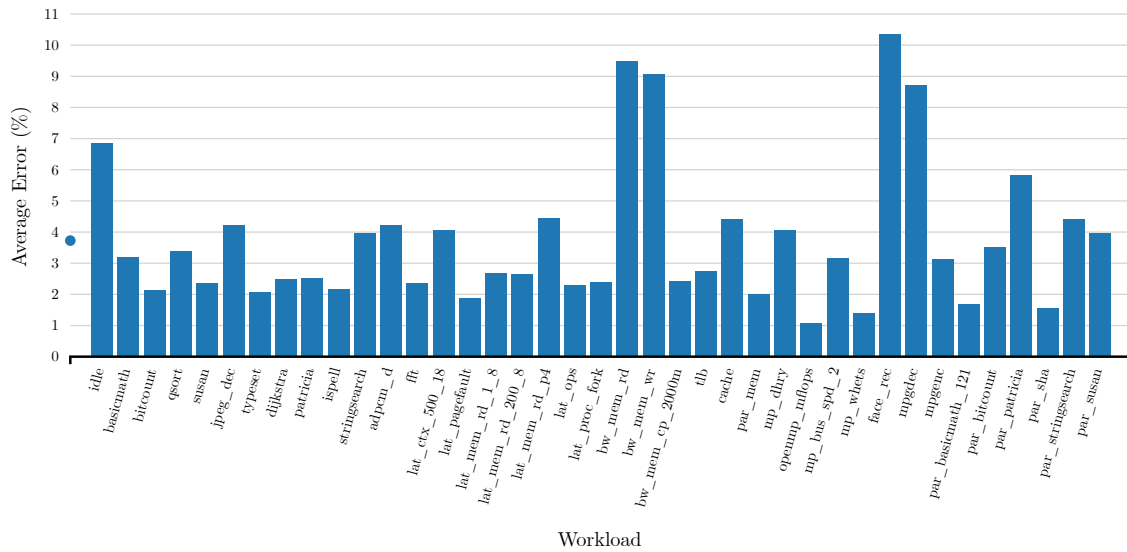


Fig. 9. Average error (mean absolute percentage error [MAPE]) for all 39 workloads, aggregated across each DVFS level and three fan settings

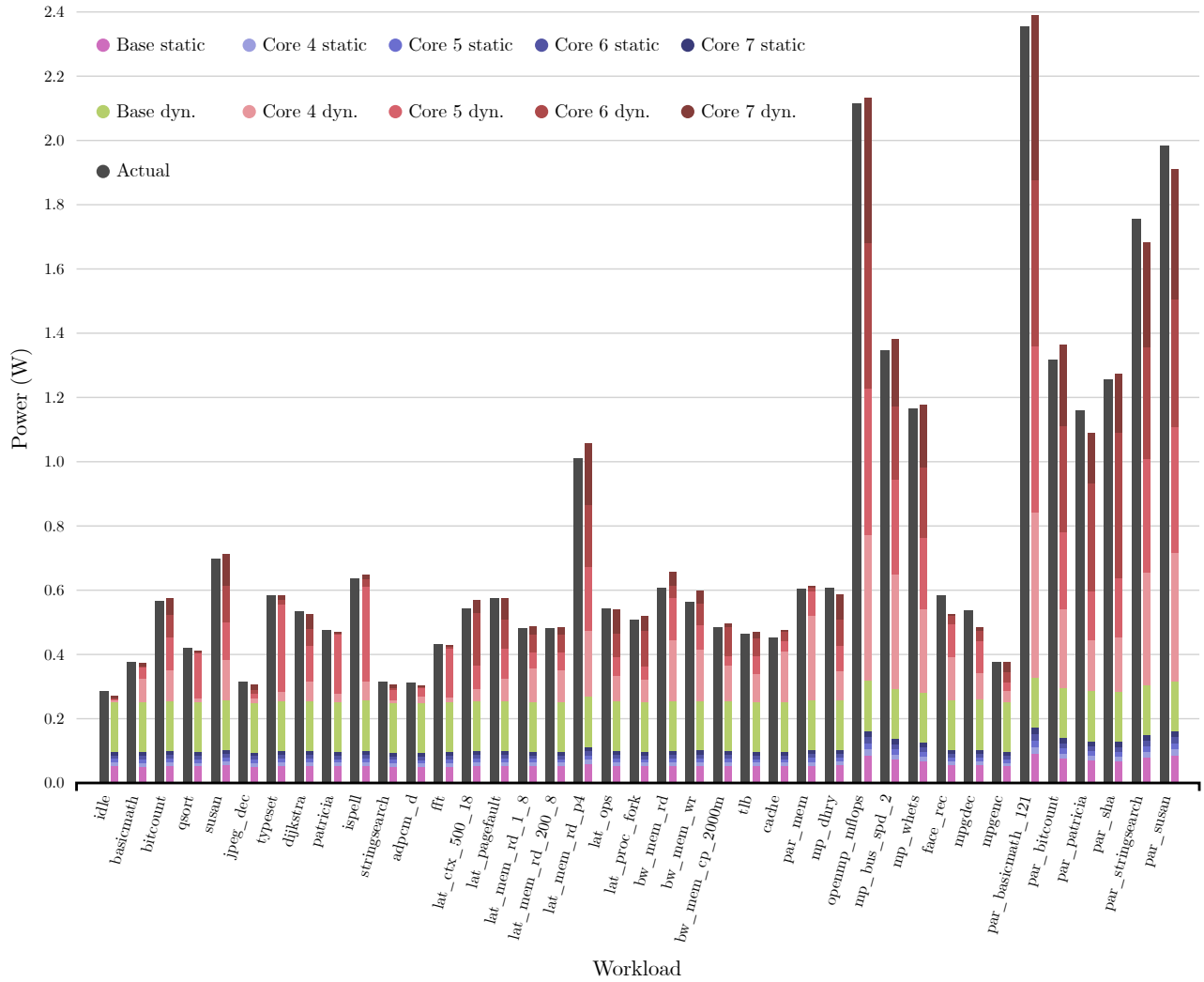


Fig. 10. Power breakdown for all 39 workloads, aggregated over every DVFS level and three different fan settings

We analyse the power consumption as CPUs are switched offline and combine this data with analytically derived power components to infer internal CPU behaviour and break down the power consumption. We are able to accurately predict the temperature-compensated static power of each CPU and the L2 cache, the background dynamic switching power, and the dynamic power caused by the workload of each CPU. We then test the validity of our analytical methods and assumptions by testing our model using measured data with large temperature deviations, a diverse selection of workloads, eight different voltage and frequency (DVFS) levels and with different numbers of CPUs utilised. Each component of the model is calculated individually and summed together to form the cluster power estimate, which was found to have an average error of 3.7%. We demonstrate the importance of applying statistical rigour to the modelling process and how our model captures the relationship between each input variable and the power consumption. We provide equations and software tools for implementing our power model (available at [19]).

#### ACKNOWLEDGMENT

This work was supported by ARM Ltd. and EPSRC Grant EP/K034448/1 (the PRiME Programme [www.prime-project.org](http://www.prime-project.org)).

Experimental data used in this paper can be found at DOI: 10.5258/SOTON/398554 (<http://dx.doi.org/10.5258/SOTON/398554>).

#### REFERENCES

- [1] A. K. Das, M. Walker, A. Hansson, B. Al-Hashimi, and G. V. Merrett, "Hardware-software interaction for run-time power optimization: a case study of embedded linux on multicore smartphones," in *Int. Symp. on Low Power Electronics and Design (ISLPED)*. IEEE, 2015.
- [2] F. Bellosa, "The benefits of event: Driven energy accounting in power-sensitive systems," in *Proc. 9th Workshop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System*, ser. EW 9. New York, NY, USA: ACM, 2000, pp. 37–42.
- [3] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: methodology and empirical data," in *Microarchitecture, 2003. MICRO-36. Proc. 36th Annu. IEEE/ACM Int. Symp.*, Dec 2003, pp. 93–104.
- [4] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade, "Decomposable and responsive power models for multicore processors using performance counters," in *Proc. 24th ACM Int. Conf. Supercomputing*, ser. ICS '10. New York, NY, USA: ACM, 2010, pp. 147–158.
- [5] W. Bircher and L. John, "Complete system power estimation using processor performance events," *Computers, IEEE Transactions on*, vol. 61, no. 4, pp. 563–577, April 2012.
- [6] S. Sankaran and R. Sridhar, "Energy modeling for mobile devices using performance counters," in *Circuits and Systems (MWSCAS), 2013 IEEE 56th Int. Midwest Symp.*, Aug 2013, pp. 441–444.
- [7] G. Da Costa and H. Hlavacs, "Methodology of measurement for energy consumption of applications," in *Grid Computing (GRID), 2010 11th IEEE/ACM Int. Conf.*, Oct 2010, pp. 290–297.
- [8] K. Singh, M. Bhadauria, and S. A. McKee, "Real time power estimation and thread scheduling via performance counters," *SIGARCH Comput. Archit. News*, vol. 37, no. 2, pp. 46–55, Jul. 2009.
- [9] W. Bircher and L. John, "Complete system power estimation: A trickle-down approach based on performance events," in *Performance Analysis of Systems Software, 2007. ISPASS 2007. IEEE Int. Symp.*, April 2007, pp. 158–168.
- [10] R. Rodrigues, A. Annamalai, I. Koren, and S. Kundu, "A study on the use of performance counters to estimate power in microprocessors," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 60, no. 12, pp. 882–886, Dec 2013.
- [11] S. Wang, H. Chen, and W. Shi, "Span: A software power analyzer for multicore computer systems," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 23 – 34, 2011.
- [12] B. Su, J. Gu, L. Shen, W. Huang, J. L. Greathouse, and Z. Wang, "Ppep: Online performance, power, and energy prediction framework and dvfs space exploration," in *Proceedings of the 47th Annual IEEE/ACM Int. Symp. Microarchitecture*, ser. MICRO-47. Washington, DC, USA: IEEE Computer Society, 2014, pp. 445–457.
- [13] M. J. Walker, S. Diestelhorst, A. Hansson, A. K. Das, S. Yang, B. M. Al-Hashimi, and G. V. Merrett, "Accurate and stable run-time power modeling for mobile and embedded cpus," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, January 2016.
- [14] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.
- [15] F. A. Endo, D. Couroussé, and H.-P. Charles, "Micro-architectural simulation of embedded core heterogeneity with gem5 and mcpat," in *Proceedings of the 2015 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, ser. RAPIDO '15. New York, NY, USA: ACM, 2015, pp. 7:1–7:6. [Online]. Available: <http://doi.acm.org/10.1145/2693433.2693440>
- [16] S. L. Xi, H. Jacobson, P. Bose, G.-Y. Wei, and D. Brooks, "Quantifying sources of error in mcpat and potential impacts on architectural studies," in *High Performance Computer Architecture (HPCA), 2015 IEEE 21st Int. Symp. on*, Feb 2015, pp. 577–589.
- [17] T. Nowatzki, J. Menon, C.-H. Ho, and K. Sankaralingam, "Gem5, gpgpusim, mcpat, gpuwattch, "your favorite simulator here" considered harmful," in *11TH ANNUAL WORKSHOP ON DUPLICATING, DECONSTRUCTING AND DEBUNKING*, 2014.
- [18] W. Lee, Y. Kim, J. H. Ryoo, D. Sunwoo, A. Gerstlauer, and L. K. John, "Powertrain: A learning-based calibration of mcpat power models," in *The IEEE Int. Symp. on Low Power Electronics and Design (ISLPED)*, July 2015.
- [19] M. J. Walker, S. Diestelhorst, A. Hansson, D. Balsamo, B. M. Al-Hashimi, and G. V. Merrett, "POWMON: Run-Time CPU Power Modelling," <http://www.powmon.ecs.soton.ac.uk/powermodeling>, Dec 2015, [Online; accessed 21-May-2016].
- [20] M. Guthaus, J. Ringenberg, D. Ernst, T. Austin, T. Mudge, and R. Brown, "Mibench: A free, commercially representative embedded benchmark suite," in *Workload Characterization, 2001. WWC-4. 2001 IEEE Int. Workshop on*, Dec 2001, pp. 3–14.
- [21] L. McVoy and C. Staelin, "Lmbench: Portable tools for performance analysis," in *Proc. of the 1996 Annu. Conf. on USENIX Annual Technical Conference*, ser. ATEC '96. Berkeley, CA, USA: USENIX Association, 1996, pp. 23–23.
- [22] R. Longbottom, "Roy longbottom's pc benchmark collection," <http://www.roylongbottom.org.uk>, September 2014, [Online; accessed 2-June-2015].
- [23] S. M. Z. Iqbal, Y. Liang, and H. Grahn, "Parmibench - an open-source benchmark for embedded multiprocessor systems," *IEEE Comput. Archit. Lett.*, vol. 9, no. 2, pp. 45–48, Jul. 2010. [Online]. Available: <http://dx.doi.org/10.1109/L-CA.2010.14>
- [24] M.-L. Li, R. Sasanka, S. V. Adve, Y.-K. Chen, and E. Debes, "The alpbench benchmark suite for complex multimedia applications," in *Workload Characterization Symposium, 2005. Proceedings of the IEEE International*, Oct 2005, pp. 34–45.
- [25] S. K. Rethinagiri, O. Palomar, R. Ben Atallah, S. Niar, O. Unsal, and A. C. Kestelman, "System-level power estimation tool for embedded processor based platforms," in *Proc. 6th Workshop Rapid Simulation and Performance Evaluation: Methods and Tools*, ser. RAPIDO '14. New York, NY, USA: ACM, 2014, pp. 5:1–5:8.
- [26] M. J. Walker, A. K. Das, G. V. Merrett, and B. Hashimi, "Run-time power estimation for mobile and embedded asymmetric multi-core cpus," in *HIPEAC Workshop on Energy Efficiency with Heterogeneous Computing*. HiPEAC, January 2015. [Online]. Available: <http://eprints.soton.ac.uk/372827/>
- [27] K. Nikov, J. L. Nunez-Yanez, and M. Horsnell, "Evaluation of hybrid run-time power models for the arm big.little architecture," in *Embedded and Ubiquitous Computing (EUC), 2015 IEEE 13th International Conference on*, Oct 2015, pp. 205–210.
- [28] N. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. Hu, M. Irwin, M. Kandemir, and V. Narayanan, "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68–75, Dec 2003.