Thermal Intelligent Control DVFS for Cyber-physical Systems

Pablo Hernández Almudi[1,2], Darío Suárez Gracia[1], Eduardo Montijano[2]

Afiliación: [1] Grupo de Arquitectura de Computadores, [2] Grupo de Robótica, Percepción y Tiempo Real
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: pabloha@unizar.es

Summary

This work explores the use of well-known control techniques to tackle the heat dissipation problem in embedded devices. These types of devices lack of good solutions for managing heat and the existing ones may lead to unpredictable performance variations. We will show a way of solving the problem with classic control theory and workload identification.

Introduction

The improvements in the performance of embedded system have been one of the major milestones on cyberphysical systems and internet of things (IoT). Examples like autonomous driving, robots, medical monitoring, or vigilance systems have seen their processing power increased. These have also led to improvements in the capabilities and algorithms that are now more complex than ever and require deterministic execution times. Because of that, better heat dissipation techniques are required to manage to extract all the thermal energy generated.

There are many different solutions to this problem, such as bigger heat sinks, fans to increase the flow of cold air, low power water cooling (Yueh 2015), … that work perfectly fine without space restrictions. On the other hand, with size constrained systems or those solutions are not enough, we have software solutions. (Park 2018) proposes to prioritize tasks in the foreground and even turn off background tasks. CoScale (Deng 2012) improves the consumption of memory intensive tasks by coordinating the CPU frequency with the memory.

We propose to use control theory to keep a stable temperature in the CPU by using the dynamic voltage frequency scaling (DVFS). The higher the frequency, the higher the voltage must be, and therefore increasing the performance but also the heat generated. By using control techniques, we can keep a stable temperature and frequency that will give a deterministic performance.

Some proposals that use control theory already exist, like SPECTR (Rahmani 2018) with a multiple-input multiple-output (MIMO) control based on a state machine to keep the quality of service (QoS) of computer vision workloads. (Rahmani 2015) also proposes a feedback loop to protect multicore systems against energy spikes. Arm's IPA (Wang 2017) uses a proportional-integral-derivative (PID) controller to limit the maximum frequency a processor can reach. The objective of our approach, keeping a stable temperature in the presence of different workloads, is something that none of them have considered.

Control strategy

In order to design a control, we first need a model of the response so that we can tune the control to better fit it. To model the thermal response, we use a transfer function in the form,

$$G(s) = \frac{\Delta T(s)}{\Delta F(s)} = \frac{k_p}{1 + T_p s}$$

where kp represents the heat generated because of the DVFS, and T_p describes the dissipation of the heat.

We identify 3 types of workloads that have a very distinctive behaviour, memory intensive, floating point and integer. By adapting the control to each of them we get an improvement of the response, and we can keep a better control of the temperature.To do so, we propose the scheme in Figure~1, where we have a PID controller that gets a change in temperature as input and outputs a frequency. Since the frequency-temperature response is not linear we choose a point for linearisation, and that is why we also add to our scheme around the PID a gain. Before the frequency is given to the processor, we need to quantify it since the processor cannot use any arbitrary frequency. Above all of that, we have a supervisor to collect statistics of the execution in order to identify the type of workload we are running. Depending on the type, the supervisor can modify all the different control parameters.

Revista "Jornada de Jóvenes Investigadores del I3A", vol. 10 (Actas de la XI Jornada de Jóvenes Investigadores del I3A – 16 de junio de 2022). ISSN 2341-4790.

Methodology

The evaluation and experiments have been performed using the development board HiKey 960. It has a system-on-chip (SoC) with 8 cores with big.LITTLE Arm architecture, which means it has 4 powerful cores and 4 very efficient. In our experiments we have used the big cores as those are the ones that generate the most heat. The controller has been implemented as a kernel module in the operative system, which allows it to have full access to the hardware and better control over the timing.

To model the thermal behaviour, we have used 3 different suites each focusing on a different type of workload. Memcpy for memory intensive operations, Stress-ng for floating point, and Dhrystone and Stress-ng in its integer form for more general workloads. To perform the evaluation, we use the suite Geekbench that executes a diverse set of benchmarks and operations.

Results

We start by performing the system identification fitting the real behaviour to our formula. In order to do so, we use frequency steps that we maintain in time, and we record the response of the temperature. Doing this for each type of workload gives us the behaviour for each of them. In Figure~2 we can see the fitting of the model with the temperature response of the CPU running a floating-point benchmark. This is good enough to test different PID parameters in a simulator before testing on the real world.

Once we have the model for each of the workloads, we train the supervisor to identify them. In order to do so, we use performance counters, which are an architectural feature that allows to count events in the processor. We use counters for floating point instructions, memory accesses, integer instructions and total instructions. Instead of using absolute values we use the percentage over total amount of instructions, avoiding this way variations in the rating if the processor changes the frequency. After performing the training, we find out that using only memory and integer counters we achieve 79.6% precision in training dataset and 58.62% for tests, with a simple multinomial logistic regression.

Now that we have all our parameters and model, we can start doing tests with the real system. Figure 3 shows the results of the control for each class. First row shows the evolution of the temperature in time (black continuous line) and the changes of the objective temperature (dotted red line). Second row shows the frequencies applied each sampling time (grey), and because the control is quantified it needs to change many times the frequency, that is why we show in black the 5 second average window of the frequency to understand the response.

Finally, we put everything together and in Figure~4 we have the temperature comparison of our proposal with the default control in Android. We can see that at the beginning of the execution, our proposal tries to get to the assigned temperature allowing higher frequencies. Closer to the end the high variability of the type of benchmarks, make it more difficult to keep a stable temperature, but our proposal keeps it at a lower level. In Figure~5 we can see the frequency chosen by each control and we can see a more stable and less changed frequency from our proposal.

Conclusions

This work presents a feedback control scheme that manages the temperature of a SoC with DVFS in embedded systems. To adapt the control, we propose a supervisor that checks the type of load and changes the parameters to obtain a better response from the system. Comparing it with standard systems, it shows a better behaviour and improved performance stability avoiding huge drops of frequency.

Bibliography

[1]. YUEH, Wen, et al. Experimental characterization of in-package microfluidic cooling on a system-on-chip. In 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). IEEE, 2015. p. 43-48.

[2]. PARK, Jihoon; LEE, Seokjun; CHA, Hojung. App-oriented thermal management of mobile devices. In Proceedings of the International Symposium on Low EssdfsdfasdfPower Electronics and Design. 2018. p. 1-6

[3]. DENG, Qingyuan, et al. Coscale: Coordinating cpu and memory system dvfs in server systems. In 2012 45th annual IEEE/ACM international symposium on microarchitecture. IEEE, 2012. p. 143-154.

[4]. RAHMANI, Amir M., et al. Spectr: Formal supervisory control and coordination for many-core systems resource management. In Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems. 2018. p. 169-183.

[5]. RAHMANI, Amir-Mohammad, et al. Dynamic power management for many-core platforms in the dark silicon era: A multi-objective control approach. In 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). IEEE, 2015. p. 219-224.

[6]. WANG, Xin. Intelligent Power Allocator. In 2017 Arm Technical Report.

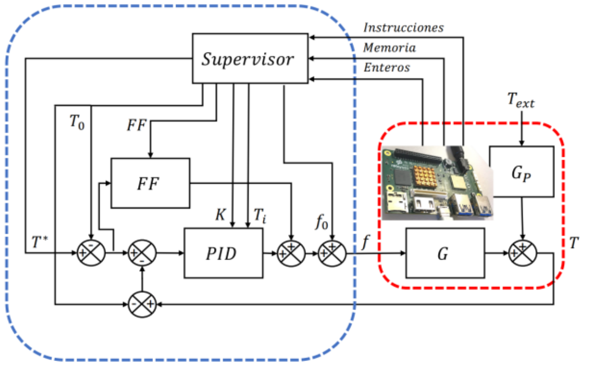

Figure 1

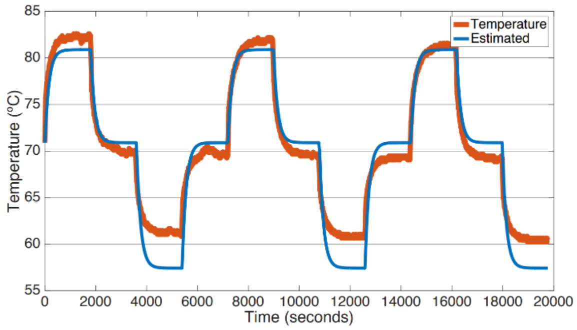

Figure 2

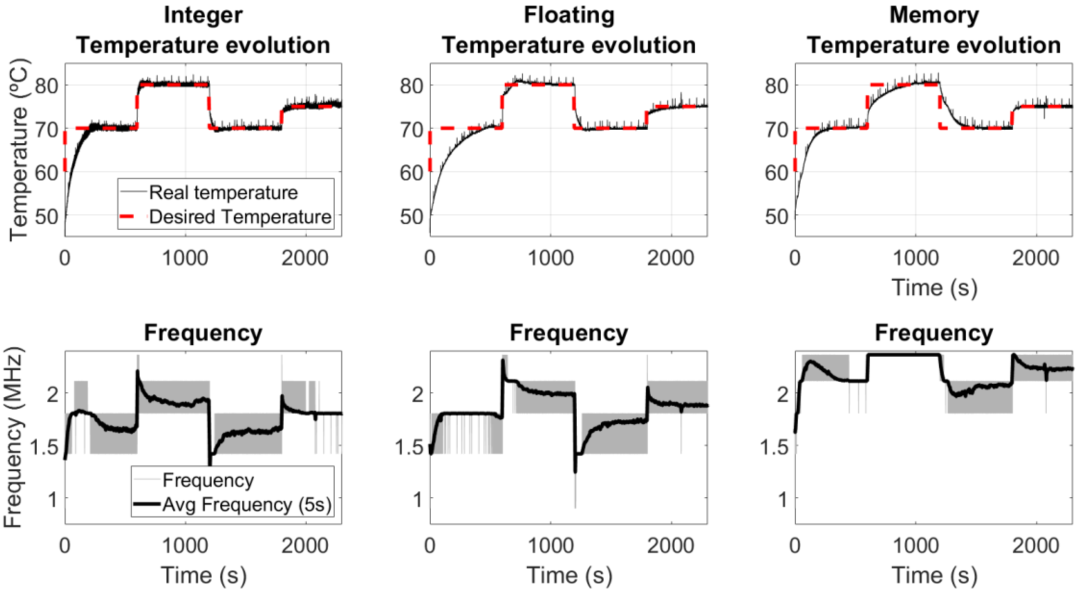

Figure 3

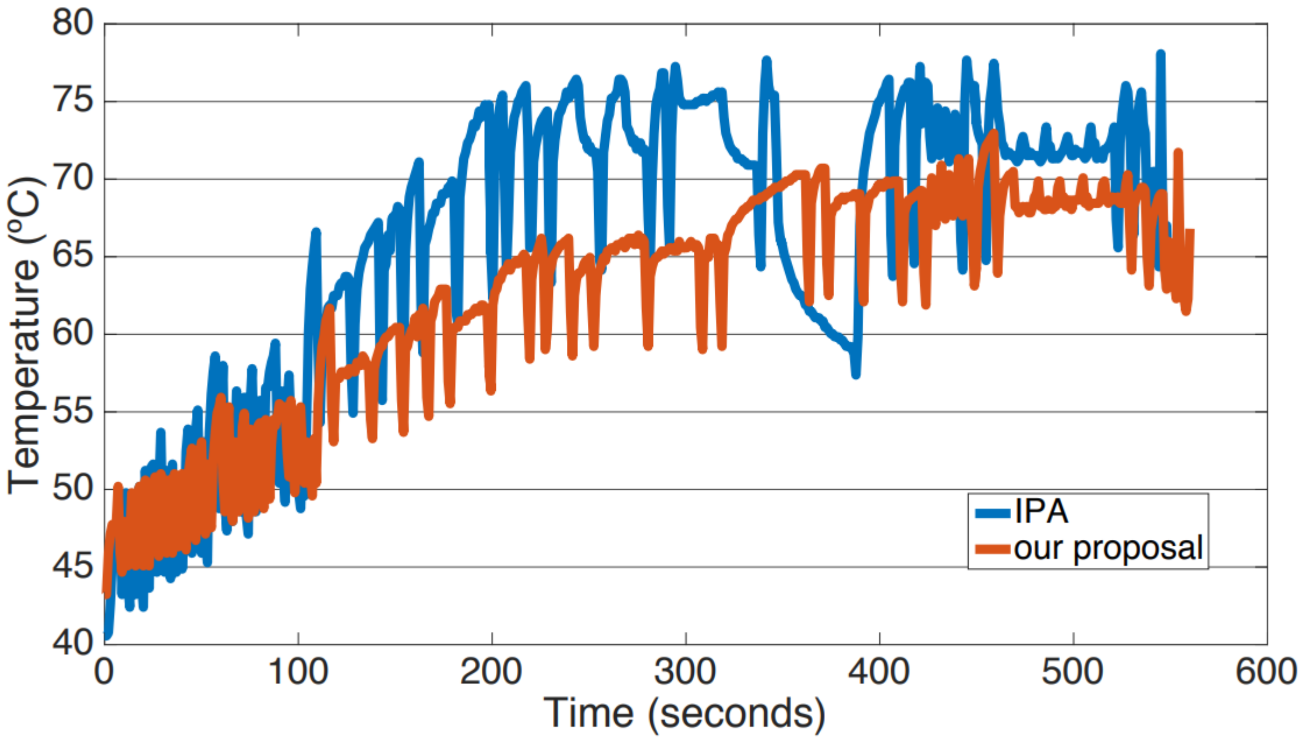

Figure 4

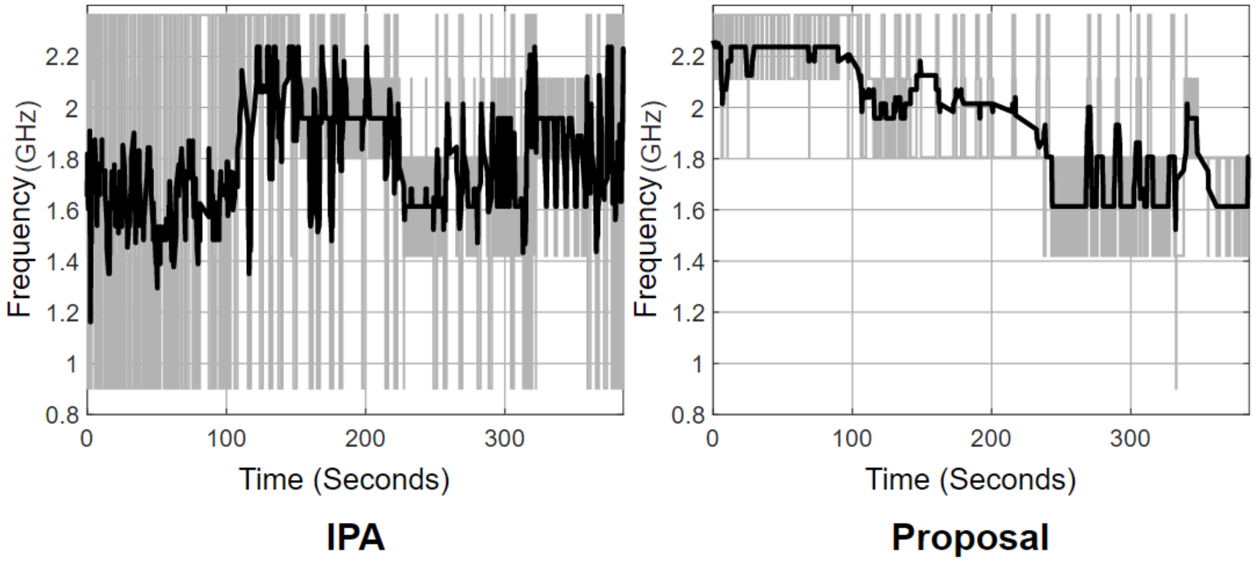

Figure 5

ANEXO: INSTRUCCIONES COMPLEMENTARIAS PARA LA PRESENTACIÓN DEL ABSTRACT

- **Título de la presentación**: El título deberá ir obligatoriamente en minúsculas (si el título es en inglés, irá en mayúsculas la letra inicial de cada palabra, salvo las *palabras vacías* –artículos, preposiciones y pronombres-).
- **Nombre del autor(es)**: Irán en orden natural -Nombre Apellido1 (Apellido2)- y en minúscula, separando cada autor por coma [,]. Se recomienda no utilizar iniciales para los nombres, sino la forma completa.
- **Afiliación**:
    - Deberá figurar el nombre del grupo de investigación del autor(es) en la primera línea desarrollado y con el acrónimo al final entre paréntesis, aunque todos los autores pertenezcan al mismo grupo.
    - Si todos los autores son del mismo grupo de investigación, no hará falta poner la nota en superíndice [1], ya que no es necesario aclarar qué afiliación le corresponde cada uno.
    - En caso de que haya autores de dos o más grupos del I3A, el segundo (y siguientes) se pondrá justo debajo del primero, antes de los datos de contacto.
    - El resto de información de contacto (dirección del I3A, etc.) se mantendrá tal cual aparece en la plantilla, salvo la dirección de email, que se pondrá únicamente la del autor que vaya a presentar la ponencia.
    - Para autores externos solo se citará el nombre de la institución.
- **Listado de referencias bibliográficas**: Deberá seguir obligatoriamente el estilo de la Norma UNE-ISO 690:2013. Se recomienda el uso de gestores bibliográficos como Refworks o Endnote (se desaconseja la utilidad de citas integrada en Google Scholar por su inconsistencia). Ante cualquier duda, consultar con i3a@unizar.es .

Todos los trabajos deberán presentarse obligatoriamente en formato Word y sin la inclusión de este anexo a través del sistema de depósito de trabajos en línea de la revista disponible en la siguiente dirección: https://papiro.unizar.es/ojs/index.php/jji3a/about/submissions . A través del sistema de depósito se indicará la sección a la que se adscribe el trabajo: “Artículos (Ingeniería Biomédica)”, “Artículos (Procesos y Reciclado)”, “Artículos (Tecnologías de la Información y las Comunicaciones)” o “Artículos (Tecnologías Industriales)”**.**

Cualquier duda respecto a la forma de presentación podrá consultarse en la dirección de correo electrónico i3a@unizar.es. Los trabajos que no sigan estas indicaciones serán devueltos a los autores para su corrección.