Energy Optimization in Microcontrollers

Mihai-Bogdan Stoica

National University of Science and Technology POLITEHNICA Bucharest, Bucharest, Romania

Abstract: Power conservation is a significant priority in microcontroller development, and many solutions focus on reducing consumption rates. However, since energy is the integral of power over time, both power and time must be taken into account for the best energy efficiency. I examine popular low-power technologies and suggest methods to increase their effectiveness. I developed test sets to validate my hypotheses, and the results showed that time has an equal impact on energy use as power. The notion that increasing the processor clock frequency saves energy is questioned by certain peripherals, which consume constant power but reduce their consumption when operating in low-power modes. In this article, the smallest amount of sleep time needed to save energy is examined using models for the STM32F103C8 processors.

Index Terms: energy consumption, embedded systems, energy efficiency

1. Introduction

As green computing becomes more important [1], there is a growing focus on improving energy efficiency in computing systems [2]. Embedded systems, which have become widespread, are integral to modern smart living. These systems, in contrast to general-purpose computing devices, are frequently subject to strict constraints, one of which is low energy usage. Energy consumption is a good performance indicator for embedded systems, as it has a substantial impact on battery life, thermal management, device stability, and security. As the central component of embedded processing, microcontrollers must therefore comply with strict energy regulations. Adegbija *et al.* [3] highlighted that to achieve good performance, IoT processors must be energy efficient. Many strategies have been proposed to reduce power consumption, and low-power technologies have been the subject of much research. In addition to reviewing popular low-power technologies and techniques, this study makes two energy consumption assumptions. First, even when instantaneous power consumption rises, energy savings are achievable because energy is the integral of power over time. Second, when dynamic voltage and frequency scaling (DVFS) and power modes are employed, we must consider the overhead associated with processor clock frequency transitions and power mode changes.

2. Related Studies

After reviewing two commonly used low-power technologies and approaches, I present my hypotheses for improving energy efficiency in this section. These theories are based on how power consumption and execution time affect system performance.

2.1. Dynamic Voltage and Frequency Scaling (DVFS)

DVFS is a widely used technology, in the modern embedded world, that controls power consumption by altering the CPU's clock speed and voltage. Hua et al. [4] and Choi et al. [5] researched energy optimization while retaining the performance of the hardware. These studies provide a number of strategies for achieving the optimal energy-performance trade-off based on processor load and memory access patterns.

2.2. Dynamic Power Management (DPM)

Another method for improving energy use is Dynamic Power Management (DPM), which modifies the system's power status based on its activity level. When the system is idle or under low load, it can transition to lower power states, thus reducing energy consumption without sacrificing overall performance. Studies conducted by Bhatti et al. [6] and Li et al. [7] have demonstrated that DPM, particularly when combined with DVFS, can lower energy usage by a large margin. Through the manipulation of workload demands and processor power states, DPM offers an adaptable method for maximizing energy efficiency in embedded systems.

2.3. Our Assumptions for Improving Energy Efficiency

As previously stated, numerous low-power technologies have been put out to lower microcontroller power consumption. Nevertheless, as Equation 1 below shows, energy is understood to be the integral of power over time, meaning that both power consumption and execution time affect the total energy consumption:

$$E = \int_{t=0}^{t_{\text{execution}}} P(t)\, dt \tag{1}$$

Reducing execution time and power consumption is part of energy conservation. For computationally intensive operations, increasing CPU frequency may reduce execution time and not always increase energy consumption, despite higher power usage. The energy consumption of an embedded system can be divided into two categories: peripheral energy and CPU energy. In order to verify our hypotheses and evaluate low-power technologies, we need to develop models and examine their energy impact while taking peripherals into account.

3. Testing and Results

The study utilizes a custom board using a STM32F103C8 core, designed by STMicroelectronics. To verify the theories about power management, a number of test sets were created to investigate the connection between execution time and energy use. The STM32F103C8 microcontroller, part of STMicroelectronics' STM32F1 series, is based on the ARM® Cortex®-M3 architecture. It operates at a maximum frequency of 72 MHz and has low power usage. According to STMicroelectronics' official documentation, its working current is approximately 230 μA/MHz [8].

3.1. Energy usage and Clock Frequencies

The target hardware, a STM32F103C8 microcontroller, has a programmable clock frequency of up to 72 MHz. The processor clock frequency is the only variable in this subsection. Every test case is run at a different processor clock frequency and compiled using the exact same compiler optimization options. Figure 5 shows how the microcontroller's working current increases with the processor clock frequency in a linear fashion. The following provides the approximate fitting line:

$$I(\text{mA}) = 0.20006 \times f(\text{MHz}) + 0.8642$$

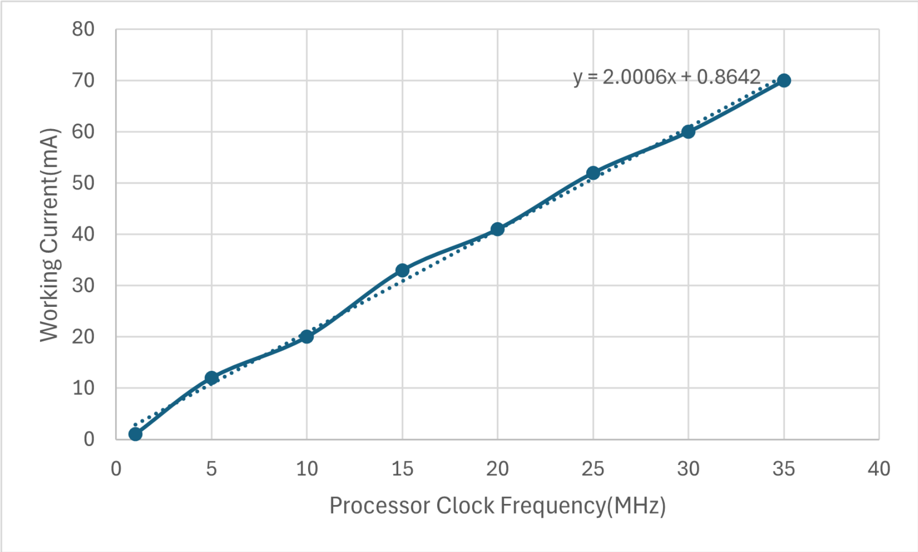

Fig. 1. Working current of the STM32F103C8 processor as a function of processor clock frequency.

By looking at Figure 1 and noting that execution time decreases with higher clock frequencies, the microcontroller's energy consumption can be determined by considering both execution time and operating current.

3.2. Energy and Time Overhead in DVFS and DPM

To examine the processor's time and energy usage during clock frequency shifts, a some test cases were created. The results are presented in Figure 2.

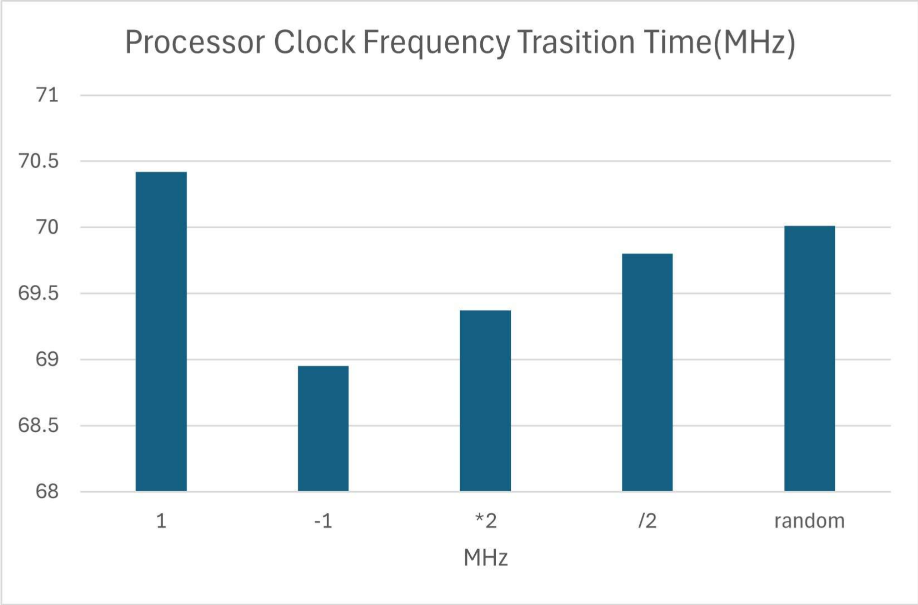

Fig. 2. Time overhead of the processor clock frequency transitions for STM32F103C8.

Examining the graph, the following conclusions can be drawn:

1) Regardless of the frequency difference between the old and new processor frequencies, the time overhead during transitions stays largely constant.
2) The STM32F103C8 processor may experience a time overhead of up to approximately 70 ms during frequency transitions.
3) Because the STM32F103C8's peripheral clock is obtained by splitting the HCLK, changing the processor frequency creates a delay overhead and temporarily disables the peripherals.The reinitialization of APB1 and APB2, which must be done to use the peripherals again, makes the situation even worse.

When the clock frequency changes, a working current of about 12.051 mA is seen. About 4.050 mA of this is used by the processor, and the rest 8.001 mA is used by the peripheral components. Therefore, an energy consumption of roughly 4.215 mJ is linked to each clock frequency switch.

Three separate power-saving modes are available on the STM32F103C8 microcontroller: sleep mode, stop mode, and standby mode. The regulator options in each of these modes differ, and the energy savings that occur depend on which power domains are turned off. The power states follow, in order, the equation:

$$y = -27.74 \ln(x) + 60.091$$

y represents the energy consumption, and x is the mode parameter corresponding to the different power-saving modes.

The results of the experiment show that peripherals power usage is independent of CPU clock rates but varies significantly based on the power-saving mode. Standby mode uses the least amount of energy, while sleep mode uses the most. The quantity of data lost and the time required to recover to normal state vary depending on which power domains are disabled. According to the STM32F103C8 datasheet, stop mode keeps data but needs clock reconfiguration at wake-up, whereas standby mode loses data and can only be recovered using external interrupts.

3.3. Future Directions

The impact of Dynamic Voltage and Frequency Scaling (DVFS) and Dynamic Power Management (DPM) on energy consumption were examined, in this paper, using a STM32F103C8 microcontroller. While the obtained results were consistent with the theoretical predictions, the tests were conducted under stable environmental conditions. The temperature of the STM32F103C8 during testing did not exceed 34℃. In order to increase energy efficiency, future research can examine how DVFS and DPM affect power and energy consumption over a range of temperatures and STM32 microcontrollers.

4. Conclusion

This study demonstrated how Dynamic Voltage and Frequency Scaling (DVFS) and Dynamic Power Management (DPM) affect energy consumption in the STM32F103C8 microcontroller. The two technologies were examined through a series of test cases. The results showed that increasing clock frequency raises power demand but reduces overall energy use by shortening system power-on time, especially during computationally intensive tasks.

The data showcased in the above sections highlights the impact of clock reconfiguration overhead and the role of peripheral components in overall energy utilization. The results emphasize that hardware manufacturers should try their best to allow independent power control of peripherals. In addition to this, software developers should deactivate unused peripherals in order to optimize energy economy. Increasing CPU frequency does not always translate into higher energy consumption. In some cases it may actually increase efficiency.

Future studies should test different operating conditions and a wider range of microcontrollers. This scientific paper may be inaccurate because it uses only the STM32F103C8 core.

References

[1] Dar, K.S.; Asif, S.; Islam, A., ”Power Management and Green Computing: An Operating System Prospective”, *Can. Int. J. Soc. Sci. Educ.*, 2015, vol. 2, pp. 164–183. [Google Scholar]

[2] Thakkar, A.; Chaudhari, K.; Shah, M., ”A Comprehensive Survey on Energy-Efficient Power Management Techniques”, *Procedia Comput. Sci.*, 2020, vol. 167, pp. 1189–1199. [Google Scholar]

[3] Adegbija, T.; Rogacs, A.; Patel, C.; Gordon-Ross, A., ”Microprocessor Optimizations for the Internet of Things: A Survey”, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 2018, vol. 37, pp. 7–20. [Google Scholar]

[4] Hua, X.; et al., ”Optimal voltage levels for energy efficiency”, *Journal of Embedded Systems*, 2020.

[5] Choi, Y.; et al., ”A DVFS technique for energy-performance tradeoffs”, *IEEE Transactions on Embedded Systems*, 2019.

[6] Bhatti, M.; et al., ”Dynamic Power Management in Embedded Systems: A Survey”, *IEEE Transactions on Embedded Systems*, 2020, vol. 19, no. 4, pp. 1234-1247.

[7] Li, Y.; et al., ”Energy-Efficient Dynamic Power Management for Real-Time Embedded Systems”, *Journal of Embedded Systems*, 2021, vol. 25, no. 3, pp. 567-576.

[8] STMicroelectronics, ”STM32F103C8 Datasheet”, available at: https://www.st.com/en/microcontrollers-microprocessors/stm32f1-series.html, accessed December 20, 2024.

Note on References: *References that are indicated with ”[Google Scholar]” can be found on Google Scholar (https://scholar.google.com). To preserve the integrity of the citation format, no direct link is given. Please use Google Scholar to look up the article title directly.*