November 2019

Fast Adaptive Voltage and Boost Frequencies for Central Processing Units

Derek Basehore

Nick Sanders

Stephen Boyd

Sonny Rao

Alex Levin

Fast Adaptive Voltage and Boost Frequencies for Central Processing Units

Abstract:

This publication describes methods, techniques, and apparatuses that enable a user equipment (UE) to quickly increase or lower the supply voltage and/or the clock frequency to handle changes in load operating conditions of the components of a system on chip (SoC). The UE uses a dynamic voltage and frequency scaling (DVFS) to handle changes in load operating conditions. During the DVFS, an application processor (AP) writes the supply voltage and the clock frequency settings to shared memory between the SoC, the AP, and a microcontroller unit (MCU). The MCU, then, can change the supply voltage using a voltage controller and/or change the clock frequency using a clock controller, which includes multiple phase-locked loops (PLLs). The utilization of a clock controller with multiple PLLs enables the MCU to trigger a switch between preset clock frequencies much faster than when using a clock controller with a single PLL. Further, the MCU can anticipate the load operating conditions of the components of the SoC and can quickly adjust the supply voltage and the clock frequency settings to run the anticipated load, enabling the UE to save power and increase performance.

Keywords:

Microcontroller, microcontroller unit, MCU, integrated circuit, IC, clock rate, clock frequency, boost frequencies, supply voltage, adaptive voltage, central processing unit, CPU, graphics processing unit, GPU, dynamic voltage and frequency scaling, DVFS, system on chip, SoC, application processor, AP, multi-core processor, variable load, phase-locked loop, PLL.

Background:

User equipment, such as smartphones, notebooks, tablets, and the like, may include a first central processing unit (CPU), a second CPU, a graphics processing unit (GPU), and other computing components, that may be embedded in and/or on a system on chip (SoC). A CPU may be a multi-core CPU that may operate one, two, or all cores depending on the type of operation. The components of the SoC run code under various load operating conditions. To increase the reliability of the CPU, the UE sets different clock frequencies and supply voltages to run light, medium, or heavy loads. As such, during a light load operating condition, the UE can increase the clock frequency to increase the performance of the CPU or can decrease the supply voltage to save power. The change in clock frequency and supply voltage may be referred to as adaptive voltage scaling (AVS).

Formula 1 illustrates the relation between the power consumed, the supply voltage, and the clock frequency:

$$P = K \cdot C \cdot V^2 \cdot f \qquad (1)$$

where *P* stands for the power consumed by an integrated circuit (IC), *C* stands for the capacitance of the IC, *V* stands for the supply voltage, *f* stands for the clock frequency, and *K* is a constant. The precise architecture of the IC helps determine the values of *C* and *K*. As is illustrated by Formula 1, regardless of the precise architecture of the IC, the power consumed by the IC is directly (instead of inversely) proportional to the square (a quadratic relation) of the supply voltage *V*, and is directly proportional (a linear relation) to the clock frequency *f*.

For a given supply voltage and a given load operating condition, it may be possible to increase the clock frequency to increase the performance of the CPU. When the load operating conditions change, however, the UE needs to change the supply voltage or the clock frequency to optimize its performance under the new load operating condition. Supplying a higher than necessary voltage results in an unnecessary increase in the power consumed by the CPU. On the other hand, supplying a lower than necessary voltage results in inferior performance of the CPU. Similarly, generating a higher or lower than necessary clock frequency results in an unnecessary increase in the power consumed by the CPU and/or inferior performance of the CPU. An inferior performance of the CPU may be lower computational power, a partial brownout (unintentional voltage drop), a complete failure of the CPU, some or all the transistors of the CPU may be in an unsafe (undetermined) state, or other CPU operational malfunctions.

Therefore, it is desirable to have a technological solution that enables a UE to quickly increase or lower the supply voltage and/or the clock frequency to handle changes in load operating conditions. By doing so, the UE can save power and increase the performance of the CPU or other computing components.

Description:

This publication describes methods, techniques, and apparatuses that enable a UE to quickly increase or lower the supply voltage and/or the clock frequency to handle changes in load operating conditions. An SoC and its embedded components, such as a first CPU, a second CPU, a GPU, a digital signal processor (DSP), a multimedia processor, memory (*e.g.*, volatile and/or non-volatile), or other IC components, require different supply voltage and/or clock frequency settings to handle varying load operating conditions. How fast a UE can change the supply voltage and/or the clock frequency settings to accommodate a change in the load operating condition can impact the power consumed and the performance of the SoC and its components. For example, during a light load operating condition, the first CPU may be running code using only one core, while the rest of the CPU cores may be in an idle state. In that case, the UE may increase the clock frequency to increase the performance of the non-idle core of the first CPU, may decrease the supply voltage to save power, or may increase the clock frequency and decrease the supply voltage.

Figure 1 helps illustrate how the UE can quickly change the supply voltage and/or the clock frequency settings depending on the load operating conditions.

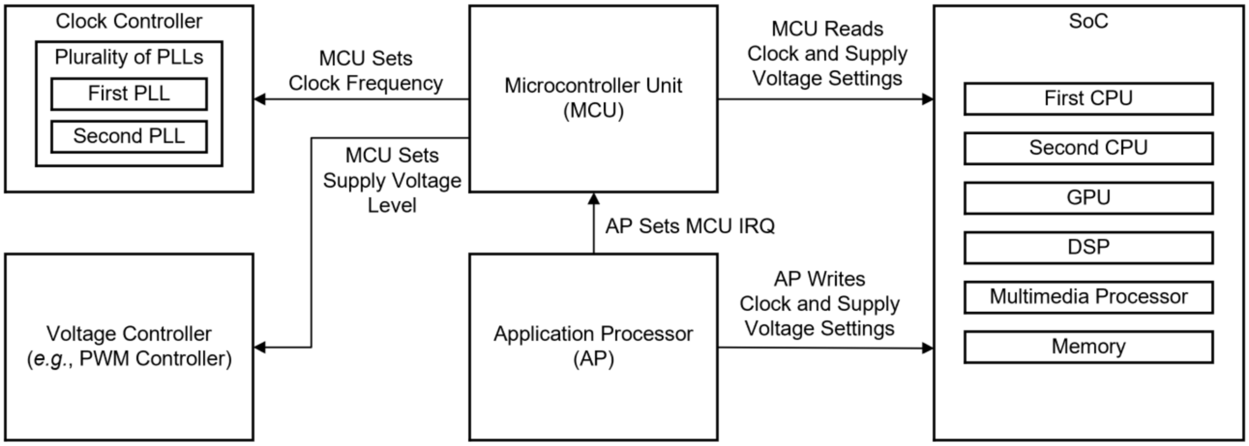
Clock Controller
Plurality of PLLs
First PLL
Second PLL
MCU Sets Clock Frequency
Microcontroller Unit (MCU)
MCU Reads Clock and Supply Voltage Settings
SoC
First CPU
Second CPU
GPU
DSP
Multimedia Processor
Memory
MCU Sets Supply Voltage Level
AP Sets MCU IRQ
Voltage Controller (e.g., PWM Controller)
Application Processor (AP)
AP Writes Clock and Supply Voltage Settings

Figure 1

Figure 1 illustrates how the UE can achieve a dynamic voltage and frequency scaling (DVFS) when the components of the SoC run code under different load operating conditions. During the DVFS, an application processor (AP) determines the supply voltage and the clock frequency settings based on the computational needs of various application software. The AP writes the supply voltage and the clock frequency settings to the memory of the SoC. The memory of the SoC can be volatile (*e.g.*, dynamic random-access memory (DRAM), static RAM (SRAM), synchronous DRAM, double data rate SDRAM (DDR SDRAM)), can be non-volatile (*e.g.*, NOR flash memory, NAND flash memory), or a combination of volatile and non-volatile memory. The memory can be embedded in the SoC (as is illustrated in Figure 1) and/or can be a stand-alone module. Regardless of the type and the location of the memory, the memory can be accessed by the SoC, the AP, and a microcontroller unit (MCU). Once the AP determines and writes the supply voltage and the clock frequency settings to the memory of the SoC, the AP sends an interrupt request (MCU IRQ) to the MCU, signaling the MCU to temporarily stop changing the supply voltage or the clock frequency settings and start monitoring the supply voltage and/or the clock frequency settings set by the AP. The MCU has access to the memory and can monitor the total usage of the memory (*e.g.*, 20%, 50%, 99%, or 100%) and which computing component is using the memory (*e.g.*, the first CPU, the second CPU, the GPU, the DSP, or the multimedia processor). Also, the MCU can read the supply voltage and the clock frequency settings that the AP writes to the shared memory. The MCU, then, can change the supply voltage and the clock frequency using a voltage controller and a clock controller, respectively.

The voltage controller can be any voltage controller, such as a pulse width modulation (PWM) controller, a shunt voltage regulator, a series voltage regulator, and other types of voltage controllers or regulators. The MCU can trigger an increase or decrease of the supply voltage to match the demands of a computing component. For example, assume the first CPU is a multi-core CPU with four cores. The voltage margin (the maximum voltage minus the voltage minimum or the difference between “peaks” and “valleys”) is tighter when running code using one CPU core compared to running code using four CPU cores. Theoretically, the voltage margin is four times tighter in the CPU running code using one core compared to the CPU running code using four cores. Thus, when the CPU runs code using one core while the rest of the cores are in an idle state, the MCU can lower the overall supply voltage, because the tighter voltage margin results in less supply voltage fluctuations. A decrease in the supply voltage leads to a decrease in the power consumed by the CPU. Also, when the CPU runs code using only one core, it is possible to increase the clock frequency to increase the performance of the CPU. An increase in the clock frequency, however, leads to an increase in the power consumed by the CPU. Nevertheless, recall from Formula 1 that the power consumed by an IC (*e.g.*, a CPU) is directly (instead of inversely) proportional to the square (a quadratic relation) of the supplied voltage, and is directly proportional (a linear relation) to the value of the clock frequency. Therefore, in certain circumstances, the CPU may consume less power while running at a faster speed (better performance).

To increase the speed that the MCU can trigger a clock frequency switch from a first clock frequency to a second clock frequency, the illustrated clock controller includes a plurality of phase-locked loops (PLLs), such as a first PLL, a second PLL, and so forth. Each PLL of the clock controller can generate a preset clock frequency that is different from the preset clock frequencies of the other PLL(s) of the clock controller. Also, similar to a conventional clock controller that includes a single PLL, each PLL of the illustrated clock controller can increase or decrease the clock frequency. Continuing with the example of the first CPU, assume the first CPU can run clock frequencies between 2.4 gigahertz (GHz) and 2.8 GHz. Using a conventional clock controller with a single PLL it may take a few hundred microseconds (an order of 10^{-6} seconds) to increase the clock frequency from 2.4 GHz to 2.8 GHz. During the time it takes a single PLL to increase the clock frequency (f), the CPU is underperforming. Likewise, it may take single PLL a few hundreds of microseconds to decrease the clock frequency from 2.8 GHz to 2.4 GHz. During the time it takes the PLL to decrease the clock frequency, the CPU is consuming more power than is necessary. The clock controller illustrated in Figure 1, however, utilizes multiple PLLs and can switch between preset clock frequencies considerably faster. Ideally, the MCU can trigger a clock frequency switch, from the first preset clock frequency to the second preset clock frequency, after one clock cycle by choosing between the first PLL or the second PLL, which can happen within a few nanoseconds (an order of 10^{-9} seconds). If the first PLL is configured to generate a clock frequency of 2.8 GHz, and the second PLL is configured to generate a clock frequency of 2.4 GHz, the MCU can trigger the illustrated clock controller in Figure 1 to change the clock frequency from

2.4 GHz to 2.8 GHz or from 2.8 GHz to 2.4 GHz approximately 1000 times faster compared to a conventional clock controller with a single PLL. If the AP requests a clock frequency that is different from a preset frequency (*e.g.*, 2.5 GHz), the MCU requests the clock generator to select the PLL with the closest preset clock frequency and adjusts the clock frequency to match the clock frequency requested by the AP. Although to reach this non-preset clock frequency takes longer than one clock cycle, it still takes a shorter time than using a conventional clock controller with a single PLL.

Further, the MCU also controls CPU core idle operations. The MCU controlling the CPU core idle operations enables the MCU to predict the supply voltage and the clock frequency settings required by the CPU before the CPU can add or drop cores from running code. The MCU reads the voltage level and clock frequency settings set by the AP, anticipates the load operating conditions of the components embedded in or on the SoC (*e.g.*, first CPU, second CPU, GPU), sets the clock frequency of the clock controller, and sets the supply voltage of the voltage controller, before it can allow a CPU core to run code or it can idle a CPU core. Using the MCU to control when the supply voltage is changed, when the clock frequency is changed, and when a core of a CPU can run code, enables the MCU to quickly set the optimum supply voltage and clock frequency settings according to the load operating conditions.

Similar to the MCU triggering the voltage controller and the clock controller to adjust the supply voltage and the clock frequency based on the load operating conditions of the first CPU, the MCU can trigger the voltage controller and the clock controller to adjust the supply voltage and the clock frequency based on the load operating conditions of the second CPU, the GPU, the DSP, and any other computing components. Although not illustrated in Figure 1, there may be brownout detection circuits embedded in or on the SoC that can provide information to the MCU in case some ICs or parts of the ICs require more voltage and/or power. Such information enables the MCU to quickly trigger a change in the supply voltage to mitigate brownouts.

In conclusion, the described methods, techniques, and apparatuses that enable a UE to quickly increase or lower the supply voltage and/or the clock frequency to handle changes in load operating conditions, can save power and increase the performance of the UE.

References:

[1] Patent Publication: US20070210158A1. Combined radio frequency identification and optical imaging module. Priority Date: March 9, 2006.

[2] Patent Publication: US20080082854A1. Methods and systems for adjusting clock frequency. Priority Date: September 28, 2006.

[3] Patent Publication: US20060066376A1. Frequency management apparatus, systems, and methods. Filing Date: September 28, 2004.

[4] Patent Publication: US20150194969A1. Semiconductor device and control method. Priority Date: January 7, 2014.