

Chapter 2

Low-Power Circuit Technologies

Takashi Nakada

Abstract In this chapter, basic low power techniques are explained including DVFS and other power management such as power gating. Discussion on energy overhead of these techniques is given to understand the limitation on temporal granularity. Normally-off computing tries to apply fine-grain power management with the help of non-volatile memory. Thus, access time and read/write energy of non-volatile memory are also discussed to reveal the condition to reduce energy from the view point of temporal granularity of memory accesses.

Keywords Fine-grain power management · Power gating · DVFS · Break even time

2.1 Introduction

In this chapter, several well-known low-power technologies are explained. Energy consumption is classified into the dynamic and the static energy. The former is caused by switching activities of transistors and essential for computing. On the other hand, the latter is caused by leakage current and not essentially required for computing but consumed whenever power is supplied.

As we briefly introduced in Chap. 1, different types of low power technologies have their own characteristics. For instance, Dynamic Voltage and Frequency Scaling (DVFS) can reduce the dynamic energy by controlling trade-off between performance and energy efficiency. Meanwhile, Power Gating (PG) can eliminate the static power by turning off their power supply while idle state.

In order to realize the optimal energy management, a correlation between low power techniques should be considered carefully. An illustrative example is that of

T. Nakada (✉)
The University of Tokyo, Bunkyo, Tokyo, Japan
e-mail: nakada@hal.ipc.i.u-tokyo.ac.jp

DVFS and PG. In case that DVFS is applied, energy efficiency improves but the execution time gets longer and idle period becomes shorter. Then, the effectiveness of PG is diminished because PG suffers from the transition overhead as described in Sect. 2.3.1.

Modern computer systems consist of many kinds of components and their characteristics is different from one another. One of the most important elements is memory. Flip flops are the most basic memory element and contained in almost all logic circuits. Memory modules are obviously common memory element.

Usually, performance is the most important criteria for many computer systems. To achieve desired performance, high-speed volatile memory is widely used. However, the volatile memories lose their contents when power supply is cut off. Therefore, it is difficult to utilize low-power technologies such as PG.

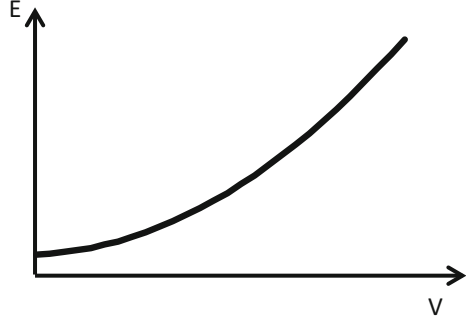
Recently, new generation non-volatile memories (NV-RAMs) are emerged and comparable performance to the volatile memories. A combination of the NV-RAM and the aggressive power managements present a promising opportunity. A major drawback of the NV-RAM is a write performance. To write information permanently, a longer latency and larger power are required. To achieve significant energy reduction while minimizing performance degradation, how to optimize write operation is critically important.

If volatile RAM is replaced with NV-RAM, stand-by power is not required to maintain its contents but write energy becomes larger. In general, write interval is a key to decide which memory consumes lower energy. Such noticeable parameter that indicates the trade-off time interval is called break even time (BET). However, the amount of write data can be optimized by system configuration such as a memory hierarchy. From a viewpoint of performance, negative impact of the longer write operation can be hidden by a memory hierarchy. Therefore, to find optimal configuration and management, it is important to understand detailed behavior and characteristics of each component and each low power technology.

The first step to realize the optimal power management is to understand the behavior of the target system. To recognize idle periods are essentially important, as the low-power technologies reduce the energy consumption during idle periods.

Idle periods have wide variety of time interval when systems are under operation. Especially, their temporal and spatial granularities are the most important characteristics. The spatial granularity corresponds to management unit in terms of area, such as power domain for PG. If the spatial granularity is finer, the number of components in one area is fewer, then opportunity to switch to low-power mode increases. If the granularity is coarser, the opportunity decreases but control overhead becomes smaller, because the area of control logic is relatively smaller.

In general, hardware implementation can minimize management overhead, but its behavior is given by predefined logic. Conversely, software implementation can manage more flexible, but control overhead is relatively larger. To realize optimal management, collaboration with each other is significantly important.

Fig. 2.1 Energy model

2.2 Basics of Low Power Techniques

In this section, we briefly introduce major power reduction techniques including clock gating, power gating, DVFS, and so on.

In general, the relation between energy, voltage and clock frequency can be modeled by following well known equation [5] and shown in Fig. 2.1.

$$E_{proc} = \alpha_1 T_1 C V^2 f + T_2 V I_{leak} \quad (2.1)$$

Here, E_{proc} represents the energy consumption of the microprocessor. α_1 , T_1 , C , V and f represent a switching activity, the execution time, the circuit capacity, the supply voltage, the operation frequency respectively. T_2 and I_{leak} represent the total time that includes idle period and the leakage current respectively.

The first and the second clauses represent the dynamic and the static energy respectively. The former is caused by switching activities of transistors and essentially consumed by computing. On the other hand, the latter is caused by leakage current and always consumed whenever power is supplied.

DVFS is a popular way to reduce the dynamic power and has been around for more than a decade [10]. DVFS allows the voltage and the clock frequency to be decreased dynamically to trade time for energy.

Clock Gating (CG) simply cuts clock delivery from a clock oscillator. In general, the oscillator keeps running for a quick restart. Since this technique has no performance penalty, when there is no ready task, processor cores should switch to clock gating mode unless other low power techniques are applicable.

Power Gating (PG) is a promising way to reduce the static power. In modern computer system, all the components need not work all the time during computation.

Dynamic Power Management (DPM) [4] manages power states of the components based on the BETs. The power states are defined in each component as a power knob. When an idle period is encountered, the BET of each component is compared with the length of the idle period. If the BET is longer than the length of the idle period, the component should be clock gated or power gated. Therefore, when the length

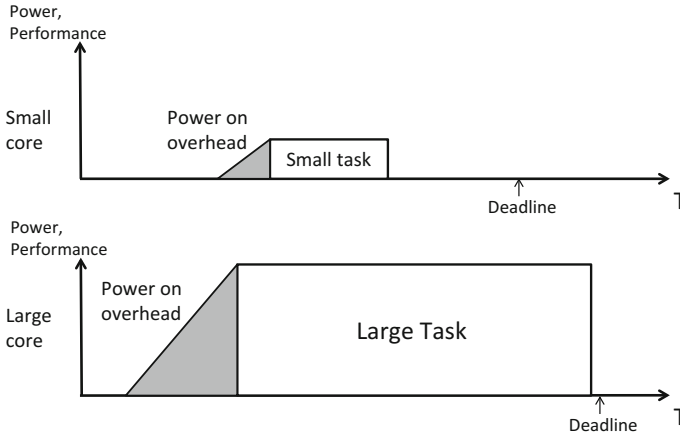


Fig. 2.2 Heterogeneous cores

of idle period becomes longer, more components can be clock gated or power gated and energy reduction is more significant.

For the combination of DPM and DVFS, tradeoffs between the two techniques should be considered [6]. When DVFS is used, the clock frequency is decreased to reduce the energy consumption during the execution of tasks, while the execution time increases and the idle time decreases.

2.2.1 Heterogeneous Hardware

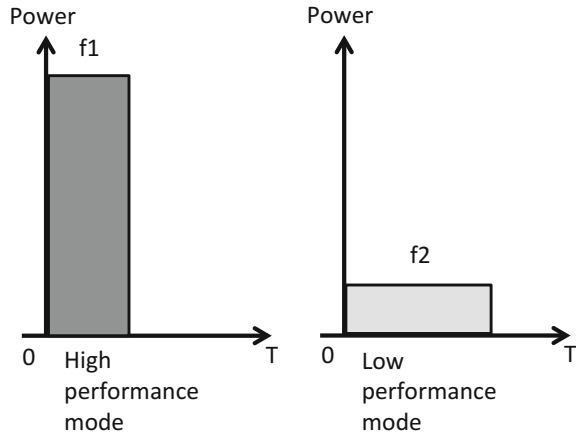
A heterogeneous hardware is a solution to adopt a variety of applications.

Modern mobile devices such as smartphones are integrated with heterogeneous processor such as big.LITTLE architecture [2]. As energy efficiency largely depends on hardware architecture, optimal core selection helps minimizing total energy consumption.

For example, when low performance is required, small core, which is energy efficient, is preferable. Large core is used only when high performance is necessary to meet their requirement. Based on this idea, usage of energy efficient core is maximized and total energy consumption is reduced. Additionally, larger core requires larger power on overhead.

Figure 2.2 shows a simple example of a heterogeneous execution. Y-axis shows relative performance and power consumption. For a small task, small core is suitable to minimize power consumption. For a large task, large core is necessary to meet its deadline even its energy efficiency is low. As a result, it is important to execute as much task as possible should be executed on small core.

Fig. 2.3 Dynamic Voltage and Frequency Scaling (DVFS)



2.2.2 DVFS

As shown in Eq. 2.1, the higher performance is realized by higher voltage, frequency and the larger circuit that causes the larger circuit capacity.

Dynamic Voltage and Frequency Scaling (DVFS) allows the voltage and the clock frequency to be decreased dynamically to trade time for energy. A lot of research is done in this area. By considering the consumed energy as a cost function, while considering deadlines as constraints, a mathematical problem can be defined and the optimal clock frequencies can be found for many kinds of real-time systems [7, 11]. For the combination of DPM and DVFS, tradeoffs between the two techniques should be considered [6]. When DVFS is used, the clock frequency is decreased to reduce the energy consumption during the execution of tasks, while the execution time increases and the idle time decreases.

Figure 2.3 shows executions of same task with two different performance modes. In high performance mode (left), the execution time is shorter but the power consumption is higher. On the other hand, low performance mode needs longer execution time but lower power consumption. According to Eq. 2.1, the total energy consumption, which is represented by the area in the figure, of high performance mode is larger than that of low performance mode. Thus, DVFS can trade between the performance and the energy consumption.

Additionally, DVFS and heterogeneous processors are complementary to each other. DVFS can adjust performance finely but adjustment range is limited. On the other hand, heterogeneous processors can adjust performance largely but possible performance mode is limited by the number of implemented cores. Their combination realizes fine and wide performance adjustment.

2.2.3 Dynamic Power Management

An overview of Dynamic Power Management (DPM) techniques is given in a survey article [4]. DPM manages both static and dynamic power dynamically. Thus DPM is important to reduce the static power when the processor core is in an idle state. This is in contrast with DVFS, which mainly reduces the dynamic power.

An example of a typical set of power modes is shown in Fig. 2.4 and Table 2.1. In active mode, all components are turned on. For a short idle interval, Clock Gating (CG) is preferable. The processor core can restart from the CG state instantly. In Core Power Gating mode, PG is applied to the processor core to obtain more energy reduction. To return from this mode, several clock cycles are required and some transition energy overhead is consumed. Vcc power gating is applied for a very long idle period. In this mode, the power supply is completely cut off. The only way to recover from this mode is to resume power supply. Then the processor core should follow almost the same as a power on reset procedure. Therefore, we can get the largest power reduction but both time and energy overhead become the most costly.

As mentioned in the previous section, there exist BETs between these power modes. To determine the appropriate power state, the length of next idle period is compared with these BETs. This strategy can be modeled with a function of cost with the length of the idle period. This function turns out to be piecewise-linear, increasing and concave [3].

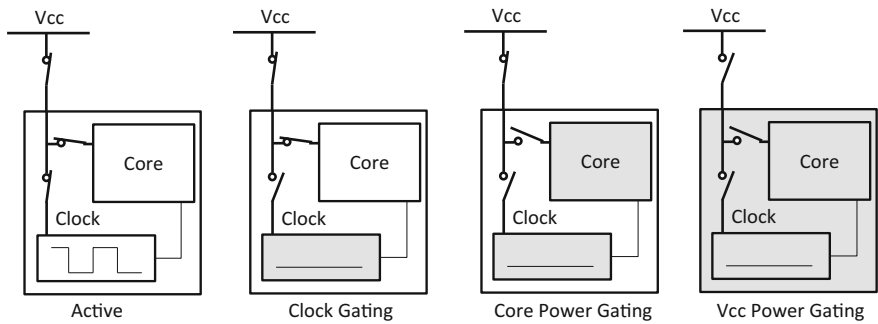


Fig. 2.4 Power modes

Table 2.1 An example of power mode

	Vcc	Core	Clock
Active	ON	ON	ON
Clock gating	ON	ON	OFF
Core power gating	ON	OFF	OFF
Vcc power gating	OFF	OFF	OFF

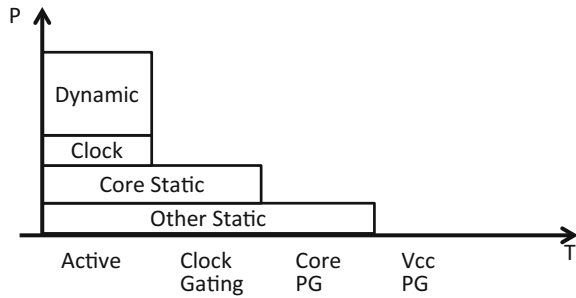
Fig. 2.5 Power breakdown

Figure 2.5 shows power breakdown of power modes. In active state, all components are on. When clock gating is applied, the processor core is stopped and no execution is possible but dynamic power of processor core and clock logic is suppressed. When core PG is applied, static power is reduced additionally. Finally, when Vcc PG is applied any power is not consumed except stand-by power of power supply which is out of scope of this book.

In embedded systems, executed tasks are fixed and periodic and their scheduling is known. So, when an idle state is encountered, the time when the next task can be invoked is definitely predictable. Then the length of the idle period is also predictable. Additionally, since the restart time is predictable, wakeup overheads are easily hidden by a pre-wakeup technique. Therefore, the optimal power management is easily determined by the strategy. Finally, DPM related parameters include hardware parameters and the length of the idle period.

2.2.4 Sleep Mode

As an extension of DPM, modern hardware has more low power modes, called sleep modes or sleep states. Sleep modes are defined for each component. For example, the Advanced Configuration and Power Interface (ACPI) specification provides an open standard for device power management. First released in December 1996, ACPI defines platform-independent interfaces for power management, monitoring and related technologies.

2.2.4.1 Global/Processor State

The ACPI [1] specification defines four Global “Gx” states as shown in Table 2.2. G1 (Sleeping) is divided into four Sleep “S1–S4” states and G2 (Soft Off) also defined as “S5” as shown in Table 2.3.

In both state definitions, the larger number corresponds to the deeper sleep mode.

Table 2.2 Global system state definition (cited from [1])

State	Definition
G0 working	A computer state where the system dispatches user mode (application) threads and they execute
G1 sleeping	A computer state where the computer consumes a small amount of power, user mode threads are not being executed, and the system “appears” to be off. Latency for returning to the Working state varies on the wake environment selected prior to entry of this state
G2 soft off	A computer state where the computer consumes a minimal amount of power. No user mode or system mode code is run. This state requires a large latency in order to return to the Working state
G3 mechanical off	A computer state that is entered and left by a mechanical means. It is implied by the entry of this off state through a mechanical means that no electrical current is running through the circuitry

The CPU power states “Cx” are defined as shown in Table 2.4. Additional states are defined by manufacturers for some processors. For example, Intel’s Haswell platform has states up to C10, which defines core states and package states.

Basically, deeper sleep state consumes less static power, but consumes more transition energy and takes longer transition time. In general, the length of each idle period is hard to predict. To realize an adaptive control, these states are managed by a timeout-based scheme. When an idle time continues more than predefined threshold times, their state will be deeper sleep mode.

If the length of an idle period is predictable, optimal sleep state can be determined and transit to the state directory to minimize overheads. For longer idle periods, deeper sleep states are preferable. Therefore the length of idle period is important to realize optimal power management. Additionally, there exists a specific length of idle period between two adjacent sleep modes. That length is definitely the BET for these sleep modes.

2.2.5 Fine-Grained Power Gating

Power gating (PG) is a promising way to reduce static power and used mainly in embedded systems. In modern computer systems, all the components need not work all the time during computation. Based on this observation, there exist many chances for PG. So far, PG is applied in a coarse manner. Especially its temporal granularity

Table 2.3 Sleeping state definition (cited from [1])

State	Definition
S1 sleeping state	The S1 is a low wake latency sleeping state. In this state, no system context is lost (CPU or chip set) and hardware maintains all system context
S2 sleeping state	The S2 is a low wake latency sleeping state. This state is similar to the S1 sleeping state except that the CPU and system cache context is lost
S3 sleeping state	The S3 is a low wake latency sleeping state where all system context is lost except system memory. CPU, cache, and chip set context are lost in this state. Hardware maintains memory context and restores some CPU and L2 configuration context
S4 sleeping state	The S4 is the lowest power, longest wake latency sleeping state supported by ACPI. In order to reduce power to a minimum, it is assumed that the hardware platform has powered off all devices. Platform context is maintained
S5 soft off state (G2 soft off)	The S5 state is similar to the S4 state except that the OS does not save any context. The system is in the “soft” off state and requires a complete boot when it wakes. Software uses a different state value to distinguish between the S5 state and the S4 state to allow for initial boot operations within the BIOS to distinguish whether or not the boot is going to wake from a saved memory image

is quite coarse. Recently, however, fine-grain PG receives much attention because finer granularity increases the chances of PG.

PG is a representative static power reduction technique, which helps cutting off the power supply to idle circuit blocks by turning off (or on) the power switches which are inserted between the GND/VDD lines and the blocks. PG has been applied to different types of circuit blocks with various granularities.

So far, power gating is applied in a coarse-grain manner. Recently, however, fine-grain power gating receives much attention because finer granularity increases the chances of PG. For example, Geyser-3 [8, 9] implements a fine-grained run-time PG. In these processors, PG is applied to function units (FUs). Each FU can be powered on or off instruction by instruction. In other words, instruction-level power gating is implemented in these processors. Based on this observation, there exist many chances for PG in a wide range of idle periods.

Table 2.4 Processor power state definition (cited from [1])

State	Definition
C0 processor power state	While the processor is in this state, it executes instructions
C1 processor power state	This processor power state has the lowest latency. The hardware latency in this state must be low enough that the operating software does not consider the latency aspect of the state when deciding whether to use it. Aside from putting the processor in a nonexecuting power state, this state has no other software-visible effects
C2 processor power state	The C2 state offers improved power savings over the C1 state. The worst-case hardware latency for this state is provided via the ACPI system firmware and the operating software can use this information to determine when the C1 state should be used instead of the C2 state. Aside from putting the processor in a non-executing power state, this state has no other software-visible effects
C3 processor power state	The C3 state offers improved power savings over the C1 and C2 states. The worstcase hardware latency for this state is provided via the ACPI system firmware and the operating software can use this information to determine when the C2 state should be used instead of the C3 state. While in the C3 state, the processor’s caches maintain state but ignore any snoops. The operating software is responsible for ensuring that the caches maintain coherency

Generally speaking, coarser-grain PG can reduce more static power but have a larger transition overhead. Thus, coarse-grain PG should be applied only for long idle period. For a short idle period, finer-grain PG is preferable. Therefore, there exists certain idle time between PGs. The boundary times are called BETs.

Granularity is a key metric to understand what the best power management is. In power gating technology granularity corresponds to a power domain. In a same power domain, power supply is controlled together by one control signal. Namely, a finer power domain controls power supply of a smaller area. The power domains can be hierarchical, namely a coarser power domain can contain several finer power domains. When the coarser power domain turns off, all of the finer power domains in it necessarily turn off.

A simple example is shown in Fig. 2.6. In this example, gray color shows power off area and others are on. In left processor, only one function unit is turned off by fine-grain PG but right processor is totally powered off by coarse-grain PG.

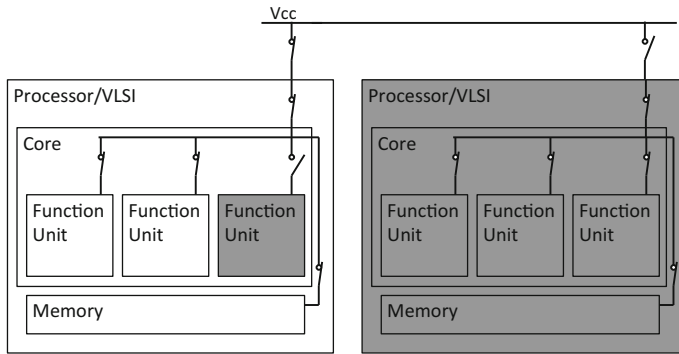


Fig. 2.6 Hierarchy of power domain

From view point of circuit, in general, to minimize control overhead it is better to use as coarse domain as possible. Because, to realize fine grain power management, a large number of power switches and control logics are required. The static power of the switched and logics cannot be turned off by themselves. On the other hand, when the power domain is larger, the opportunity of PG may be shorter in time domain. Therefore, to realize optimal power management, hierarchical power domain management, which is combined coarse and fine domain managements.

With such hierarchical management, as large as possible power domain should be turned off. Then within the remaining domain, finer power management should be applied. When a coarser power domain turns off, several finer domains in it are automatically turns off with their power switches and control logics.

As a result, it is important that as coarse as possible power domain should be powered off for as long time period as possible.

2.3 Energy/Performance Trade-Off

In this section, we take a general view of energy overheads of low power technologies. There are two type of overhead; one is an energy overhead, the other is a performance overhead. As the performance overhead can be hidden and largely depends on application and other situations, we focus on the energy overhead.

We formulate typical trade-offs and clarify important parameters to achieve energy reduction.

As we have already introduced several low-power technologies, there may exist several low-power modes for a particular control domain. There are trade-offs between these modes. Namely, lower power mode can reduce more static power but need larger transition overhead. If this is not true, it means that there exist a low-power mode that is more energy efficient and needs lower overhead than other

Fig. 2.7 Energy model and BET

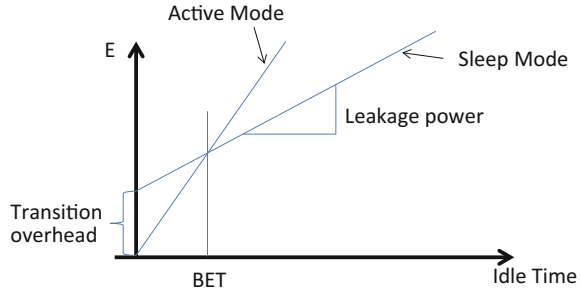
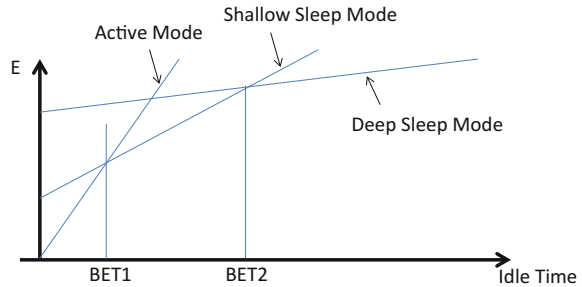


Fig. 2.8 BETs with multiple sleep modes



modes. Then, we can always use such ideal low-power mode. In the real systems, such ideal mode is not exists. Therefore, an adaptive power management is important.

2.3.1 Transition Energy and BETs

When a component has only one low power mode, to achieve energy reduction by this power mode, the minimum length of an idle period *BET* is given by follows.

$$BET = E_{OH} / (P_{active} - P_{sleep}) \quad (2.2)$$

Here, E_{OH} represents transition overhead energy. P_{active} and P_{sleep} represent power consumptions when active and low power states respectively. If the length of idle period is longer than the *BET*, that component should turn into low power mode to reduce total energy consumption.

Figure 2.7 shows energy models of active mode and sleep mode. As there is no transition overhead for active mode, the energy function of active mode starts from origin. On the other hand, as there is some transition overhead for sleep mode, the energy function of sleep mode starts from higher than origin. Meanwhile, the energy function of sleep mode has more moderate sloop than that of active mode. Therefore, there are cross point and we call it *BET*. If the idle time is longer than the *BET*, sleep mode is preferable. Otherwise, sleep mode should not be chosen.

When a component has multiple low power modes, *BET* can be extended as follows.

$$BET_i = (E_{OH_i} - E_{OH_{(i-1)}}) / (P_{(i-1)} - P_i) \quad (2.3)$$

Here, the component has N modes and transition overhead energy and power consumption of i th mode are given by E_{OH_i} and P_i respectively. The energy functions are shown in Fig. 2.8. We assume $E_{OH_i} > E_{OH_{(i-1)}}$ and $P_i < P_{(i-1)}$, namely, 1st mode is the deepest sleep mode and N th mode is the shallowest. As 0th mode corresponds to an active state, $E_{OH_0} = 0$ and $P_0 = P_{active}$.

When BET_i is longer than the length of idle mode, i th sleep mode can reduce total energy consumption. If multiple BET_i satisfy this condition, deeper sleep mode can reduce more energy.

In this example, if the length of idle period is longer than BET_2 , deep sleep mode should be chosen. If the length of idle period is shorter than BET_1 , it is better to stay active mode. In any other case, shallow sleep mode should be chosen.

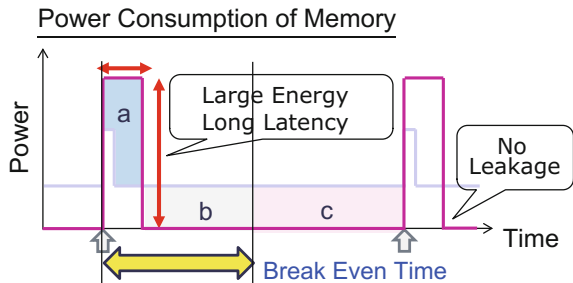
Note, this discussion assume the component directly transit to the suitable sleep mode from its active state. When the transitions occur step by step, BET_i will be longer.

2.3.2 Access Energy of NVRAM and BET

For the access energy of NVRAM, same discussion of low power mode is possible. The latest NVRAM consumes less static energy but consumes larger access energy especially for write operation as shown in Fig. 2.9. To simplify this discussion, we assume the read access energy of NVRAM is same as that of volatile memory and the static energy of NVRAM is 0. Then, to realize energy reduction, the average interval of the write accesses BET_{NV} is given by follows.

$$BET_{NV} = (WE_{NV} - WE_V) / P_V \quad (2.4)$$

Fig. 2.9 BET in NVRAM



Here, WE_{NV} and WE_V represent write energy per write access for NVRAM and volatile memory respectively. P_V represents the static power of the volatile memory. When the average write interval is longer than BET_{NV} , NVRAM can reduce total energy consumption.

2.4 Summary

Low power technologies of computer systems are indispensable for the forthcoming sustainable and sophisticated information society. Normally-off computing is one of the promising ways to achieve this goal. In this paper, we described its expectation and introduced our Normally-Off Computing project.

In this chapter, we attempt to find the best way to make full use of the new generation non-volatile memories. So far, from the view point of memory system optimization, only two properties of memory are focused on and discussed, that is latency and capacity. Now, non-volatility appears as the third property.

This property absolutely contributes further power reduction. However, it will not effectively lead to power reduction without careful considerations because of two major problems or challenges, that is, memory hierarchy and temporal granularity. In other words, once we overcome these problems, further power reduction would be achieved by using dreamy zero-leakage non-volatile memories. Cooperation and cooptimization of different design layers, including algorithm, OS, compiler, architecture, circuit and device, are definitely required.

Currently a lot of researches on new memory devices are performed. Other types of useful memory devices other than what introduced in this paper may become available in the future. However, it would not be fruitful if we discuss the device properties only. The viewpoint of “computing technology,” the technology how to make full use of the attractive property is essentially important.

References

1. Advanced configuration and power interface specification 5.0a. http://www.acpi.info/DOWNLOADS/ACPI_5_Errata%20A.pdf
2. Arm unveils its most energy efficient application processor ever; redefines traditional power and performance relationship with big.little processing (Press release). ARM Holdings (2011)
3. Augustine, J., Irani, S., Swamy, C.: Optimal power-down strategies. In: 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 530–539 (2004). doi:[10.1109/FOCS.2004.50](https://doi.org/10.1109/FOCS.2004.50)
4. Benini, L., Bogliolo, A., De Micheli, G.: A survey of design techniques for system-level dynamic power management. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **8**(3), 299–316 (2000). doi:[10.1109/92.845896](https://doi.org/10.1109/92.845896)
5. Burd, T., Brodersen, R.: Energy efficient cmos microprocessor design. In: Twenty-Eighth Hawaii International Conference on System Sciences, vol. 1, pp. 288–297 (1995). doi:[10.1109/HICSS.1995.375385](https://doi.org/10.1109/HICSS.1995.375385)

6. Gerards, M.E.T., Kuper, J.: Optimal dpm and dvfs for frame-based real-time systems. *ACM Trans. Archit. Code Optim.* **9**(4), 41:1–41:23 (2013). doi:[10.1145/2400682.2400700](https://doi.org/10.1145/2400682.2400700). <http://doi.acm.org/10.1145/2400682.2400700>
7. Huang, W., Wang, Y.: An optimal speed control scheme supported by media servers for low-power multimedia applications. *Multimed. Syst.* **15**(2), 113–124 (2009). doi:[10.1007/s00530-009-0153-5](https://doi.org/10.1007/s00530-009-0153-5)
8. Kondo, M., Kobayashi, H., Sakamoto, R., Wada, M., Tsukamoto, J., Namiki, M., Wang, W., Amano, H., Matsunaga, K., Kudo, M., Usami, K., Komoda, T., Nakamura, H.: Design and evaluation of fine-grained power-gating for embedded microprocessors. In: *Proceedings of the Conference on Design, Automation & Test in Europe, DATE 2014*, pp. 145:1–145:6. European Design and Automation Association, 3001 Leuven, Belgium (2014)
9. Usami, K., Kudo, M., Matsunaga, K., Kosaka, T., Tsurui, Y., Wang, W., Amano, H., Kobayashi, H., Sakamoto, R., Namiki, M., Kondo, M., Nakamura, H.: Design and control methodology for fine grain power gating based on energy characterization and code profiling of microprocessors. In: *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 843–848 (2014). doi:[10.1109/ASPDAC.2014.6742995](https://doi.org/10.1109/ASPDAC.2014.6742995)
10. Weiser, M., Welch, B., Demers, A., Shenker, S.: Scheduling for reduced cpu energy. In: *Proceedings of the 1st USENIX Conference on Operating Systems Design and Implementation, OSDI 1994*. USENIX Association, Berkeley, CA, USA (1994). <http://dl.acm.org/citation.cfm?id=1267638.1267640>
11. Yao, F., Demers, A., Shenker, S.: A scheduling model for reduced cpu energy. In: *36th Annual Symposium on Foundations of Computer Science*, pp. 374–382 (1995). doi:[10.1109/SFCS.1995.492493](https://doi.org/10.1109/SFCS.1995.492493)

Normally-Off Computing

Nakada, T.; Nakamura, H. (Eds.)

2017, VII, 136 p. 106 illus., 74 illus. in color., Hardcover

ISBN: 978-4-431-56503-1