Architecture of a Power-Gated Wireless Sensor Node

Goran Panić, Daniel Dietterle and Zoran Stamenković
IHP GmbH, 15236 Frankfurt (Oder), Germany
{panic, dietterle, stamenko}@ihp-microelectronics.com

Abstract

In this paper we investigate the benefits of power gating in wireless sensor nodes operating at a very low duty cycle. It is shown that the static power loss in such a node is a significant portion of the total power consumption. Therefore, we have defined power-gated wireless sensor node architecture and developed a power gating mechanism to reduce the static power loss of its functional blocks. The new node architecture, its components and power gating mechanism are described in detail.

1. Introduction

Wireless sensor networks (WSN) consist of a large number of wireless sensor nodes deployed randomly in the area. The nodes collect the environmental data and send them through the network towards the sink node. The sensor networks are constructed to be operational for a long time without replacing the batteries. Therefore, power is considered to be the most important constraint in all aspects of WSN design.

Minimizing the total power needed to transfer the data from a leaf node to the sink is the primary task of WSN design. The most of questions arise on network level. Depending on a specific application, different network topologies and corresponding MAC (media access control) [1] and routing protocols have been investigated [2]. The problems of data dissemination and data aggregation are strongly power related and therefore receive a lot of attention [3]. The proposed hardware solutions include novel node architectures that utilize optimized radio and digital parts [4]. The energy harvesting is also considered as a feasible solution to extend the battery life [5]. The two most promising non-standard approaches to power saving within a sensor node are the dynamic voltage/frequency scaling [6] and power gating [7]. While the first is considered as a promising solution for high performance WSN nodes, the second is mostly effective in WSN nodes operating at a very low duty cycle where cutting the power of inactive functional blocks leads to significant power savings.

Power gating technique suggests the implementation of power gates in order to switch off the leakage currents of inactive functional blocks. This technique is primarily seen as a low power option for the emerging multi threshold CMOS (MTCMOS) technology. It uses the high-threshold-voltage (high-$V_{th}$) transistors to isolate the low-threshold-voltage (low-$V_{th}$) circuits from the power and ground lines. Therefore, we have decided for power gating as the main power saving leverage in wireless sensor nodes and propose a novel architecture of the sensor node, which includes a power gating controller and a sophisticated functional-block power gating mechanism.

The paper starts with discussion on dynamic and static power contributions to the total power consumption of a sensor node (Section 2). Section 3 describes details of the power gating mechanism. Section 4 presents the proposed sensor node architecture. Section 5 concludes the paper.

2. Dynamic Vs Static Power in Sensor Node

The design requirements of a sensor node are typical to all portable and battery powered devices. The main target is to reach maximum performance at minimum power. Since the leakage contribution increases with every new generation of CMOS technology [8], the power optimization has to consider both the static and the dynamic power.

2.1. Dynamic Power

The dynamic power is consumed during the active mode of operation. It depends on the supply voltage and applied clock frequency and is linear to the activity factor of a circuit. Reducing one of those parameters will reduce the dynamic power. The common practice
to reduce the dynamic power is to apply the clock gating of inactive blocks. Clock gating turns clocks off when they are not needed and thus prevents undesired toggling in the distribution network of the clock.

2.2. Static Power

The static power is the power loss due to leakage currents in the circuit. The most important leakage sources are the subthreshold leakage and leakage through the oxide. The latter depends on the oxide thickness and the supply voltage. Once the high-K dielectrics reach the mainstream production, this source of leakage will be significantly reduced. The subthreshold leakage depends on the threshold voltage \( V_{th} \) and the supply voltage \( V_{dd} \). Increase of \( V_{th} \) or reduction of \( V_{dd} \) will reduce the leakage. The leakage is present in the circuit whenever the power supply is on. The contribution of leakage depends on the applied CMOS technology and the duty cycle of a node. The more time a circuit spends in idle mode, the higher is the leakage contribution. The straightforward method to reduce the leakage during idle periods is to switch off the idle circuit from the power supply.

2.3. Total Power

The total power of a sensor node can be considered as the sum of powers consumed by its components:

\[
P(t) = \sum_i P_i(t),
\]

where \( P_i \) is the power of the \( i \)-th component of a sensor node at the time \( t \). We assume that the component is either in active (on) or inactive (off) mode. Thus, the total energy consumed by a sensor node before the battery gets empty is given by:

\[
E = \int \sum_i P_i(t) dt = \sum_i P_i^{on} t_i^{on} + \sum_i P_i^{off} t_i^{off} = E_d + E_s,
\]

where \( P_i^{on} \) and \( P_i^{off} \) represent dynamic and static power of the \( i \)-th component, \( E_d \) and \( E_s \) are dynamic and static energy consumed by a sensor node; \( t = t_i^{on} + t_i^{off} \) is the sensor node lifetime. Most applications see sensor nodes as low duty cycle systems, e.g. \( t_i^{off} \ll t_i^{on} \). On the other hand, one predicts \( P_i^{off} = 0.5 P_i^{on} \) for CMOS technologies below 90 nm. In this scenario, the most of the sensor node’s energy is consumed during idle periods (\( E_s \ll E_d \)). Therefore, the power gating (cutting the power lines) of some functional blocks is only feasible solution for power saving in the case of low duty cycle sensor nodes.

3. Power Gating Mechanism

Power gating is a design technique used to reduce the overall static power loss of a chip [9]. Compared to the clock gating used to reduce dynamic power, the power gating is more invasive since it affects inter-block communication. It adds significant delays and needs a block isolation technique that secures the proper system functionality.

3.1. Component Activity Profiles

The efficiency of power gating depends on the activity profile of a component. Generally, the component’s activity consists of subsequent active and idle periods. Figure 1 shows an example of the activity profile when only the clock gating is applied. The active and idle periods are initiated by the corresponding WAKE and SLEEP signals.

![Figure 1. Activity profile without power gating](image)

Figure 2 shows an example of the activity profile when the basic power gating is applied. There is a certain delay after the WAKE signal is applied and before the system enters the active state. The static power savings are not instantaneous but need some time until the target levels are reached (dotted lines).

![Figure 2. Activity profile with power gating](image)

3.2. Principles of Power Gating

Power gating may be applied externally (off chip) or internally (on chip). In the first case, a certain block is supplied by a dedicated power supply that can be shut down. In the other, the power switches are
implemented on chip. Internal power gating may be a better solution when there are many blocks to be gated. External gating might be useful to switch off the power of pads. Internal gates are usually distributed around a power-gated block. They create a power gating network that disconnects either $V_{dd}$ or $V_{ss}$ from the block. The outputs of a power-gated block must be isolated to prevent crowbar currents in an always powered-up block (Figure 3).

Figure 3. Block diagram of a SOC with power gating

It is sometimes desirable to retain the internal state of a certain block in power-down mode, and restore this state in power-up mode. One way of doing it is use of retention registers instead of ordinary flip-flops. A retention register typically includes an auxiliary, always powered-up, shadow register that stores the data while the power is down. This register is slower than the main register but consumes less static power. The other possibility, technology dependent, is to use non-volatile shadow registers that retain their state even when the power is off. It is also possible to use scan-chains to move the register data into external memory. The scan chains are loaded into memory while the power is up.

The power gating mechanism is controlled by a power gating controller. The controller ensures that each block is properly powered up and down without disturbing the system functionality. The presence of retention registers increases complexity of the controller since the control of retention sequences must be provided.

4. Power-Gated Sensor Node

The basic architecture of a sensor node includes three main parts: processing part, radio part, and sensing part. Nowadays sensor nodes are mainly built from the off-the-shelf components assembled on a battery-powered printed circuit board. The choice of components and optional hardware strongly depends on the application.

4.1. Sensor-Node-on-Chip

The architectural impact on the power consumption in a PCB-based sensor node is not strong, since it basically depends on the choice of used components and their characteristics. On the other hand, there is a tendency to integrate as many as possible node components in a single chip (Sensor-Node-on-Chip). SNOC allows designer to optimize processing and save the power. Depending on the requirements, the SNOC designer may add optional application-specific cores or hardware accelerators to improve the node performance. The architectural tweaks may be used to allow efficient implementation of different power saving mechanisms (the clock and power gating).

4.2. Components of a Sensor Node

The implementation of power gating in a node need a comprehensive architectural consideration. The architecture must be carefully partitioned to efficiently implement the power gating mechanism. Most power-hungry components should stay shut-down for long intervals and the radio communication should be relaxed. To achieve these goals, we propose an architecture that includes a pre-processing unit on the interface to sensors, communication accelerators and a low-power wake-up radio mechanism. Additionally, we introduce a power gating controller with an integrated timer. The architecture is shown in Figure 4 and the power gating partitions are identified. The optional components (I/O, timer) are power-gated separately or coupled with some other components. For example, the I/O ports and system timer in Figure 4 could share the same power island with the processor.

Figure 4. Sensor node architecture partitioned for power gating
4.2.1. Processor. In a power-gated architecture, the processor is active only during the data transmission process, otherwise it sleeps. Its task is to control the communication process and the application. The complete software (firmware) is stored in ROM (e.g. Flash). The firmware may be loaded via debug port or via radio. The radio load option assumes the existence of dedicated hardware that recognizes incoming update command and controls the data flow to the Flash. When the application requires saving of the processor state, the retention registers must be implemented.

4.2.2. Memory. The memory sub-system consists of program and data memories. The program memory is a flash that shares the same power rail with the processor. The data memory is used to store measured data and for random data access. Often, it may be handful to physically divide the storage memory from the operational. Since there is a certain dynamics in data collection process, the storage RAM is always active (not power gated). If the incoming data frequency is low enough, it is possible to use a flash for data storage. The operational RAM is power-gated separately or together with the processor.

4.2.3. Communication Accelerator. The communication protocol typically involves a number of time-critical and computation intensive tasks, such as check sum generation and acknowledgment frame handling. It is an established design approach to introduce protocol accelerators, i.e. dedicated logic that performs critical protocol functions. This protocol processor achieves a higher computational efficiency than the general-purpose microcontroller by exploiting parallelism in the design and implementing protocol-specific operations. Consequently, the operating frequency and power consumption can be reduced keeping the same throughput.

The architecture of a hardware accelerator for the MAC protocol is shown in Figure 5. Our objective is to implement receive and transmit data paths completely in hardware. This is achieved by the CRC components (for frame check sum generation), encryption and decryption blocks, and by direct memory access. The microcontroller is responsible for the management-related parts of the protocol.

Additionally, the protocol accelerator includes a synchronization unit. Synchronization frames are analyzed and information on allocated time slots or contention access periods is extracted. With the help of timers, this information is used to access the channel, for instance after performing a random backoff procedure or when the start of an allocated time slot for frame transmission has been reached.

The synchronization information can also be used to control the power state of the wireless transceiver. In contrast to a software implementation of the power control algorithm, the hardware solution can trigger a power-down immediately after the reception of a frame and does not waste energy in the radio by interrupt latency or processing delay.

Figure 5. MAC hardware accelerator in sensor node system

4.2.4. Preprocessor. The function of the preprocessor is to filter, compress and store the sampled data into the storage memory without disturbing the processor. The filtering operations may be implemented as programmable or hard-coded and may include the engines for data smoothing, rounding, event detection, etc. Which engines will be implemented depends on the user application and the specific nature of collected data.

In-node data compression reduces the communication overhead at the cost of computation. When the parameters change mostly linearly and relatively slow compared to the applied sampling rate (e.g. environmental monitoring), it is possible to implement a very simple and effective data compression algorithm based on the repetition counting. The idea is to count the number of continuous repetitions of sampled values. For example, if the data format is 8-bit, one could add a 4-bit header that will indicate the number of continuous repetition of the same data. In this particular example, it is theoretically possible to represent 16 bytes with a 12-bit value. Other compression algorithms that efficiently exploit the data linearity could be implemented as well.

The preprocessor is implementing the mechanism to store sampled data directly to the storage memory (e.g. DMA capability). It keeps the track of the recorded data block to be used later to identify the data to be transmitted over the radio.

4.2.5. Power Gating Controller. It controls the power gating mechanism in the node. The architecture of the power gating controller is shown in Figure 6. It consists of three main units: register file, control unit and timer.

The register file keeps the control values in the controller. The control values are set by the processor. The processor selects which power partitions will be gated and specifies the time stamps for the start of
wake up sequence. A 1-bit register is used to indicate the start of power-down procedure.

![Figure 6](image)

**Figure 6.** Architecture of the power gating controller

The control unit runs as a finite-state-machine. Once the power-down sequence is started, the control unit performs steps to shut down selected blocks and starts the timer. The timer outputs are compared with the specified time stamps and when they meet the controller starts the wake-up sequence. The wake-up sequence may also be triggered with external interrupt signal. Upon finish of all the sequences, the control unit clears the register file and resets the timer. The sequences are shown in Figure 7.

![Figure 7](image)

**Figure 7.** Power gating mechanism

The size of the timer is application dependent and specified by the user. The best practice is to design the power gating controller as a reusable core with selectable bit-size of the registers and timer.

### 4.2.6. Radio

The radio consumes significant amount of energy whenever transmitting or receiving. Some network protocols use the TDMA-like power-management schemes based on wake-up/sleep scheduling to save the power. The wake-up/sleep scheduling approach is not optimal due to a complicated synchronization mechanism that wakes-up the nodes on regular basis. It introduces an overhead because frequently there is no data to transfer but only the synchronization beacons are received.

In the case of a sensor node, a solution for this problem can be switching off the node while no activity in the network and switching it on again when the wake-up event occurs. To support such behavior, one has to implement a low-power radio circuitry that will detect the wake-up signal and initiate the power-up of the node. The wake-up circuit could be a simple energy detector that would sense the energy in the wake-up message [10] or a more complicated stand-by receiver [11].

### 4.3. Implementation Issues

Implementation of a power-gated sensor node requires resolving of certain architectural issues. The first of all, one needs to decide if the power or the ground rail should be switched. It is possible to switch both, but it increases the IR drop. If the external gating exists, the $V_{dd}$ switching is natural. On the other hand, ‘footers’ used for $V_{ss}$ switching have smaller size and leakage currents.

The next decision is about the power networks: to implement one or more separate power networks. For example, the power rail for always-on logic and the power rail for CPU subsystem could be separated. Also, the PLL supply and the supply for pads are usually connected to a separate rail, etc. The number of power rails should not be too high since it makes the floorplan and layout more complex. If the external power gating is required, the control logic must be designed to deal with significant wake-up delays.

The insertion of isolation cells is required on the interface between power-gated and always-on block. The isolation cells clamp the outputs of a power-gated block to logical 0 or 1. Some tools support automatic insertion of the isolation cells but it is also possible to insert the isolation cells manually. The designer may use AND gates and OR gates to set the isolated signals to high or low, respectively.

A critical decision is to use the fine or coarse grain power gating. In fine grain power gating the switch is placed locally inside each standard cell in the library. The switching transistor is quite large and the area overhead is significant. The key advantage is low timing impact of the IR drop. In coarse grain power gating, a block of gates has its power switched by a collection of switch cells. The sizing of a coarse grain switch network is more difficult than a fine grain but the area penalty is significantly less. The designer may estimate the peak power of a certain block and calculate the requirements for the related switching network. The given constraint is the minimum IR drop at the estimated maximum current.

Coarse grain power gating can be implemented in either a ring or a grid style power network. Ring based power gating is less complex and does not impact
significantly placement and routing but it does not support retention registers and requires more area than the grid based.

One of the key challenges is managing in-rush current when the power is reconnected. To reduce the produced noise in a switch fabric, it is recommended not to switch all the switch transistors at once. One solution is to buffer the control power-down signal so that different groups of switches are selected successively.

4.4. How a Sensor Node Functions

The sensor node is supposed to run communication and application software. The data acquisition can be application driven or event driven.

In the case of an application driven network, after the network is setup and the nodes are synchronized, the application starts to collect environmental data and stores it into memory. The application defines the frequency of data sampling and, for practical reasons, usually stops the sampling during the data transfer to other nodes. In this period of time, the power rail to sensors and preprocessor is gated. After the data transfer is completed, the processor clears the data storage, moves the relevant register file data to retention registers and sets the power gating controller. Then the power gating controller initiates the power-down mode of the radio and processor.

In the event driven scenario, the node sleeps until an external event occurs (the sensors, preprocessor and wake-up radio are active). An external event is either the communication request from a node or the specific value of sensed data that triggers an interrupt, and consequently, starts the wake-up procedure. After that the communication with other nodes is established and the data is transferred to the next node.

5. Conclusion

The power gating is a promising power saving technique for future MTCMOS technologies. When applied to the design of a sensor node, it can significantly reduce the total power consumption and extend the lifetime of the node battery. As shown in the paper, an efficient implementation of power gating depends on various architectural issues. We have presented the sensor node architecture that combines embedded hardware components with tweaks in the data-flow path. The presented approach improves the overall in-node data processing and implements the efficient power gating mechanism.

Currently, the implementation of power gates is only partly supported by library providers and tool vendors. However, one can expect in near future the full support of EDA industry in the power gating design.

6. References


