ABSTRACT

Most low-power PC designs were either focused on power efficiency improvement or software power management in the working states until now. Our research aims to reduce the standby power in the off state. To reduce it by redesigning the power circuit, we cut off the power supply for the unnecessary chips, except that power which is necessary for the chip needed to wake up the system. We also turn off the power supply of the power controller chip which is used to control the system power status, and we use another low-power chip instead. We redesign the power sequence of the system, to maintain the system power state while the system power state controller is turned off. Finally, we use a low-power chip to control the power supply separately by means of the remote wake up devices.

Index Terms—Personal Computer, Power Saving, Power Consumption, Standby Power.

1. INTRODUCTION

The personal computer (PC) has become popular in recent years. Some new digital electronic home appliances have also been redesigned from PCs and are becoming visible everywhere in our daily life. As those digital electronic home appliances are equipped with several remote wake up functions, and extra standby power is needed when we are using these specific appliances. As these electronic home appliances both remain mostly in the power off state, and operate for just a few hours a day, standby power consumption has become an important energy waste source [1].

Methods used to reduce the system’s dynamic power consumption were either hardware to reduce the clock-swing of the processor or software to monitor and predict the power use of the consumer [2], [3]. If the system is in the idle state or does not require high performance, dynamic scaling can lower the operation voltage and the clock rate, thus saving dynamic power consumption [4], [5]. On the other hand, system static power consumption reduction both increases the efficiency of the power supply module and cuts off the power supply if the system is in the idle state [6], [7]. Some experts have even been considered estimating and minimizing energy leakage in an IC design at the component level by inserting control points [8]. Some systems can also be controlled remotely to resume the power supply [9], [10]. Based on these concepts, we turn off the system standby power by redesigning its power circuit and sequence until the users wake up the system via a PS/2 interface, a Universal Serial Bus (USB), a Local Area Network (LAN) and a power button to resume the standby power.

We should aim to reduce the environmental impact of products everywhere, including energy consumption. The international environmental protection agencies which define the rules for energy consumption are the U.S. Environmental Protection Agency in their “ENERGY STAR Program Requirements for Computers” and the European Commission in their “Eco-Design Requirements for Energy Using Products Directive” [11].

Table 1 shows that by the rules of ENERGY Program Requirements for Computers the power consumption of desktop PCs must be less than 1 Watt in the power off state.

<table>
<thead>
<tr>
<th>TABLE 1</th>
<th>ENERGY STAR PROGRAM REQUIREMENTS FOR COMPUTERS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Type</strong></td>
<td><strong>Requirements</strong></td>
</tr>
<tr>
<td>Desktop,</td>
<td>Off Mode: $\leq$ 1.0 Watt</td>
</tr>
<tr>
<td>Integrated Desktop,</td>
<td>Sleep Mode: $\leq$ 1.7 Watts</td>
</tr>
<tr>
<td>Notebook Computers</td>
<td>Idle State: $\leq$ 10.0 Watts</td>
</tr>
<tr>
<td>Workstations</td>
<td>Off Mode: $\leq$ 2.0 Watts</td>
</tr>
<tr>
<td></td>
<td>Sleep Mode: $\leq$ 4.0 Watts</td>
</tr>
<tr>
<td></td>
<td>Idle State: $\leq$ 80.0 Watts</td>
</tr>
<tr>
<td></td>
<td>Max. Power: $\leq$ 180.0 Watts</td>
</tr>
</tbody>
</table>

Since January 6, 2010, the EuP Lot 6 also defines that for electronic home appliances with standby power supply, which include desktop PCs, the standby power consumption must less than 1 Watt in the power off state [12]. The standby power consumption should be less than 2 Watts if the electronic home appliances support display, remote or wake up from LAN functions. In addition, the standby power consumption must be reduced to 0.5 Watts after 6th Jan. 2013 and to 1 Watt for electronic home appliances supporting display remote, or wake up from LAN functions.
Our measurements show that standby power consumption is related to the number of wake up methods. Our design is equipped with optional methods to wake up the system, but also allows the power button, which consumes the lowest standby power consumption of 0.0196 Watts from 0.6166 Watts to wake it up, thus meeting most authorized rules. In addition, we measure and compare our design with the Deep Sleep Well (DSW) power saving method [13], which is proposed by Intel new platform that will be ready for sale in 2012, this new platform consumes 0.0144 Watts.

In Section II we analyze major chips that consume standby power, and measure the power consumption of each. In Section III we shut off the unnecessary power supply of the chips, and redesign the power sequence to make sure that the system can boot normally. The final Section draws the conclusion and provides a look at future possibilities.

2. ANALYSIS OF SYSTEM STANDBY POWER

According to the ACPI (Advanced Configuration and Power Interface) specifications, there are five states in the standby mode, S0/S1/S3/S4/S5, which may be described as follows [14]:

S0: "System on" is the power-on state for the normal working state of the computer, which means that the Operating System (OS) and whatever other applications in use are running. In this state the power consumption is at its highest.

S5: "System off" is the system shutdown state, when the power consumption is at its minimum. Only the devices required to power-on the system consume power.

S1: "Power on Suspend" is the first state of a standby mode. In this state the system turns off some unnecessary peripheral devices like the monitor or the hard disk to save power. But as the CPU, the system memory and the interface card are still working, there is only a limited amount of power saving.

S3: "Suspend to RAM" is the third state of a standby mode. In this state the main memory is still powered, although it is almost the only component in use. In this state, as the OS, all applications and the opened documents are stored in the main memory; the main memory needs to consume power to maintain these stored data.

S4: "Suspend to Disk" is the fourth state of a standby mode. In this state all content of the main memory is saved to the hard drive, thus preserving the state of the OS and all applications and opened documents. The power consumption is equal to shutdown as there is no extra power needed to maintain the machine state. The system sleep state depends on the ACPI controller and the OS. When the user sets the standby mode in the OS and the BIOS, the OS requests a standby command from the ACPI controller, and the ACPI controller generates the S3#, S5# and POK (Power OK) signals to control the system sleep state.

Fig. 1 reveals the sleep states transition in the ACPI and indicates when the system initially starts up in G3. When the standby power (5VSB) is ready and resume reset (RSMRST#) goes high, the system switches into S5 state. When S3#, S5# and POK are logic high the system turns the power on and switches into the S0 state. When the system goes into the S1 state, S3#, S5# and POK don't change. When S3# becomes low but the S5# and the POK stay high in the S0 state, the system switches and stays in S3 state until the S3# changes to high. When both S3# and S5# change to low and POK stays high in the S0 state, the system switches and stays in either the S4 or the S5 state until both the S3# and the S5# change to high. The S0 and S1 states are controlled by the OS; the sleep state is also controlled by the ACPI controller, while the S4 and S5 states of the sleep states are controlled by the OS.

Under the current ATX (Advanced Technology eXpanding) structure, the power supply of a PC can be divided into two categories: general power and standby power [15], [16]. Except for 5VSB, which supports the standby power under both "System on" and "System off", all other power supplies are general power and are only
supplied while the system is on and not at shutdown. 5VSB is supplied to the resume system: keyboard, mouse, network, modem and so on. Apart from devices used for resuming, the ACPI controller also consumes standby power to manage and regulate the power in the power on, shutdown and standby modes of all states. Legacy power on sequence is as shown in Fig. 2.

The system power status controller, standby power control unit and remote wake up controllers consume standby power to wake up PCs from the sleep state. At first, we measure the power consumption of every chip in the sleep state. The power status of the system is defined by the South Bridge chip, and it provides both the S3# and S5# signals to the standby power control unit to control the power state.

![Fig. 3 System block diagram and power consumption map](image)

Fig. 3 shows the power consumption map in the sleep state of a previous typical design. “SW” has the MOSFET which is dominated by the standby power control unit, which consists of some TTL logics that are used to switch the power source between VCC5 and 5VSB. “VR” has the voltage regulator which is used to regulate the 5VSB to 3VSB for standby devices. First, the LAN consumes 50.82mA, which is the highest consumption in the whole standby system. Second, the power consumption of the South Bridge chip is 48.6mA, which is one of the consuming chips in the sleep state. The South Bridge chip integrates many I/O functions, such as the USB, the LAN and the power status controller which manages the system power status. Third, is the remote wake up implemented by the peripheral devices, the PS/2 keyboard and the mouse, consuming 7.8mA, and the USB consuming 7.1mA in a typical earlier design. The standby power control unit consumes less than other chips, usually only 2.5mA.

3. DESIGN AND IMPLEMENTATION

Based on the previous measurement results, we use an ACPI controller to regulate the system power instead of standby power control unit, and we also use MOSFET as a switch to turn off the system standby power to the South Bridge chip, the LAN, the PS/2 keyboard and the mouse as well, except for the ACPI controller. The VR modification is shown by the light grey rectangles in Fig. 4.

![Fig. 4 System block diagram for Deep S5](image)

After turning off the system standby power, the power state turns into a new power state which is similar to the G3 state. But this new state is not exactly the G3 state; due to the ACPI controller it still requires standby power to wake up the system. We define the new power state as Deep S5, which is not defined in the ACPI standard. Fig. 5 shows the transition of sleep states with Deep S5. All we want to do is cut off all standby power in order to reduce the power consumption in the Deep S5 state. Therefore, the unused standby devices, even the power status controller of the South Bridge chip, can be turned off. Hence the standby power in the whole system is greatly reduced.

![Fig. 5 Transition of sleep states with Deep S5](image)

As we know, that system power status is dominated by the South Bridge chip by sending S3# and S5# signals to the ACPI controller. The system power status controller cannot work anymore in the Deep S5 state because the power for
the South Bridge chip has been turned off. We use the ACPI controller to replace the South Bridge chip and to record the system power state before cutting off the standby power, and the ACPI controller maintains the power status until the next occurrence of wake up events.

As the Deep S5 power state which we defined is not ruled by the ACPI standard, we have to re-design the system power sequence to make the Deep S5 work as a normal system power sequence. The original power on sequence is shown in Fig. 2. When the battery of the real time clock (RTC) has been installed, the RTC reset (RTCRST#) becomes high, and the system goes into the initial G3 state. When the system standby power (SYS5VSB) is ready, the resume reset (RSMRST#) becomes high, and then the system goes into S5 mode. When the user presses the power button, the power switch-in signal (PSIN#) becomes low, and after its de-bounce the ACPI controller then turns on the power to wake up the system. We modified the power sequence as shown in Fig. 6. The sequence is the same as the normal power-on when the system goes to S5 from G3 until the system goes to S5 from S0. Then, when the time is up, the counter in the ACPI controller starts to count and turn off the system standby power. The power remains only in the ACPI controller; and the system turns into the Deep S5 which we have previously defined.

We also modify the wake up sequence as shown in Fig. 7. When the ACPI receives the wake up event (PSIN#), the controller first activates 5VSB and as a result the RSMRST# is ready. Meanwhile the system goes into the S5 state. The ACPI controller then sends PSOUT# to wake up the South Bridge chip, and the system resumes with S0 after a delay of 200ms by RSMRST#. The multiple remote wake up functions have been disabled in the Deep S5 state, because the modified standby power scheme supports only the push button for power-on whilst neither the PS/2 keyboard, the mouse, the LAN nor the South Bridge chip are not able to wake up the system in the Deep S5 state. We re-design the standby power scheme and control the PS/2, LAN and South Bridge chip power supplies respectively. Thus we can turn off the power of the South Bridge chip which consumes the most standby power.

A modified multiple wake up function is shown in Fig. 8. We modify the circuit to remove the MOSFET of the 5VSB, and we use the ACPI controller to manage the voltage regulators (VR) of the LAN and the South Bridge chip. Besides, the ACPI controller independently controls the power for the PS/2 (keyboard, mouse). Continuing the system power for the ACPI controller and the LAN allows the user to wake up the system remotely and allows both the PS/2 keyboard, the mouse infra-red (IR) and the radio frequency (RF) to wake up remotely while simultaneously supporting the PS/2 keyboard and the mouse wake up.

![Fig. 6 Modified power sequence with Deep S5 mode](image)

![Fig. 7 Modified wake up sequence starting from Deep S5](image)

![Fig. 8 Block diagram of multiple wake up functions](image)
Fig. 9 shows the Intel DSW power saving method. It uses a MOSFET as a switch to cut off the SYS5VSB while in the DSW state. The new South Bridge chip separated the 3VSB power planes into two (3VSB, 3VA). The 3VA supplies the standby power control unit and part of the standby power control logic of the South Bridge chip. Before the system turns into the DSW state, the South Bridge chip asserts the SUSWRN# signal to give notice that the system will go to the DSW state. When the standby power control unit ready to go to the DSW state, it will answer the SUSACK# signal to the South Bridge chip. Then the South Bridge chip turns off the MOSFET switch of the SYS5VSB. Thus the standby power of the system turns off, except the 3VA which is needed to wake up the system in the future. The system then goes into DSW state from this point on.

Fig. 9 Block diagram of DSW power saving method

4. RESULTS OF MEASUREMENT

With this new implementation, the users can choose any method to wake up and also determine the signal to control the device power. We have measured the standby power consumption in terms of all wake up functions, and the test results are shown in Table 2 which summarizes the power consumption using different wake up methods.

**Table 2**

<table>
<thead>
<tr>
<th>Power Saving Method</th>
<th>PS/2 Wake up</th>
<th>USB Wake up</th>
<th>LAN Wake up</th>
<th>Button Wake up</th>
<th>Current (mA)</th>
<th>Consumption (Watt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enable</td>
<td>Enable</td>
<td>Enable</td>
<td>Enable</td>
<td>123.32</td>
<td>0.6166</td>
</tr>
<tr>
<td>2</td>
<td>Disable</td>
<td>Enable</td>
<td>Enable</td>
<td>Enable</td>
<td>116.62</td>
<td>0.5831</td>
</tr>
<tr>
<td>3</td>
<td>Disable</td>
<td>Disable</td>
<td>Enable</td>
<td>Enable</td>
<td>56.68</td>
<td>0.2834</td>
</tr>
<tr>
<td>4</td>
<td>Disable</td>
<td>Disable</td>
<td>Disable</td>
<td>Enable</td>
<td>3.92</td>
<td>0.0196</td>
</tr>
</tbody>
</table>

Besides these we have measured another reference board with a South Bridge chip like ours and which is equipped with a new power saving solution of Intel DSW. The test results are shown in Table 3. We found that this reference board consumes more standby power in item 1 to item 3; moreover, only item 4 consumes 0.0144 Watts, which is less than our design.

**Table 3**

<table>
<thead>
<tr>
<th>Power Saving Method</th>
<th>PS/2 Wake up</th>
<th>USB Wake up</th>
<th>LAN Wake up</th>
<th>Button Wake up</th>
<th>Current (mA)</th>
<th>Consumption (Watt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No support</td>
<td>Enable</td>
<td>Enable</td>
<td>Enable</td>
<td>132.65</td>
<td>0.66325</td>
</tr>
<tr>
<td>2</td>
<td>No support</td>
<td>Enable</td>
<td>Enable</td>
<td>Enable</td>
<td>132.65</td>
<td>0.66325</td>
</tr>
<tr>
<td>3</td>
<td>No support</td>
<td>Disable</td>
<td>Disable</td>
<td>Enable</td>
<td>103.51</td>
<td>0.51755</td>
</tr>
<tr>
<td>4</td>
<td>No support</td>
<td>Disable</td>
<td>Disable</td>
<td>Enable</td>
<td>2.88</td>
<td>0.0144</td>
</tr>
</tbody>
</table>

We found that, this system will not cut off the standby power even if the wake up functions is disabled until the system turns into a DSW state. But in our design, we can not only individually disable the wake up function, but also turn off each standby power while the wake up function is disabled. In the meantime, we can enable anyone of the wake up functions individually in our design. The power consumption will increase 0.0335 Watts for PS/2 devices, 0.2997 Watts for USB devices and 0.2638 Watts for LAN.

5. CONCLUSION

In our design, we cut off the system standby power via MOSFET, we disable all wake up functions but the power button is used to power on, and we obtain the lowest power consumption in the sleep state. We also redesign the system’s power sequence to ensure that the system allows power-on from the Deep S5 as a normal S5. We, moreover, redesign the circuit to allow the ACPI controller to dominate the power of the PS/2 and the LAN. The fewer wake up
functions are in use; the more the power consumption is reduced.

Our design can be applied to most x86 based systems such as AMD, ATI and nVidia platforms, another Intel DSW enabled system. PC based digital electronic home appliances currently are using a legacy standby power design. Those home appliances can use our design to reduce the standby power consumption easily.

The improved system wakes up by power button, IR or RF remotely controller or LAN. The advantages of our design are that the user can use it easily with simple home appliances, and that in the standby state our design consumes only 0.0196 Watts. Moreover, our design complies with the 2010 EuP Lot6 standard which stipulates that home appliances in standby state should have a consumption of less than 1 Watt and also complies with the even stricter 2013 EuP Lot6 that requires a power consumption of less than 0.5 Watts in both standby and power off states.

6. REFERENCES


