PAPER A Low Power SOC Architecture for the V2.0+EDR Bluetooth Using a Unified Verification Platform

Jeonghun KIM[†], Suki KIM^{††}, Nonmembers, and Kwang-Hyun BAEK^{†††a)}, Member

This paper presents a low-power System on Chip (SOC) SUMMARY architecture for the v2.0+EDR (Enhanced Data Rate) Bluetooth and its applications. Our design includes a link controller, modem, RF transceiver. Sub-Band Codec (SBC), Expanded Instruction Set Computer (ESIC) processor, and peripherals. To decrease power consumption of the proposed SOC, we reduce data transfer using a dual-port memory, including a power management unit, and a clock gated approach. We also address some of issues and benefits of reusable and unified environment on a centralized data structure and SOC verification platform. This includes flexibility in meeting the final requirements using technology-independent tools wherever possible in various processes and for projects. The other aims of this work are to minimize design efforts by avoiding the same work done twice by different people and to reuse the similar environment and platform for different projects. This chip occupies a die size of 30 mm^2 in $0.18 \mu \text{m}$ CMOS, and the worst-case current of the total chip is 54 mA.

key words: Bluetooth V2.0, enhanced data rate (EDR), low-power architecture, wireless SOC, sub band codec, platform-based design, verification

1. Introduction

Bluetooth is widely adopted in mobile applications such as cellular phones, portable multimedia players (PMP), and wireless stereo head-sets. Bluetooth applications are also rapidly evolving from voice-only to highly featured applications [1], [2]. The Bluetooth v2.0+EDR (Enhanced Data Rate) specification supports not only multiple data rates but also several low-power modes. Although many parts of the Bluetooth standard have been commercially implemented, there are still rooms for improvements to the physical radio, such as improving the integrated circuits to reduce the cost and power consumption [2]–[12]. This paper focuses on reducing power consumption while maintaining a high data rate. To support these demands, we propose a low-power architecture which includes a double buffering scheme, a power management unit, and clock gated schemes.

In order to maximize the speed of the internal memory access with minimal hardware requirements, a dual bus system has been adopted. In order to effectively support memory access for the MCU to control data, and for the link controller to process packets, a double buffering scheme with dual-port memory is added, and redundant memory ac-

DOI: 10.1587/transinf.E93.D.2500

cesses are minimized. The low-power approaches consist of a double buffering scheme to reduce data transfer rate, a clock management unit, and a clock gated cell. This architecture incorporates an Expanded Instruction Set Computer (EISC) processor with high-speed bus architecture, a Sub Band Codec (SBC), a baseband transceiver, and an RF transceiver.

Bluetooth specification v2.0+EDR describes search modes which are named "Inquiry" and "Page", as well as power-saving modes which consist of "Hold", "Sniff", and "Park" cases [1]. In "Page" and "Inquiry" modes, a Bluetooth device searches other devices to connect them. The proposed SOC design supports both the search modes and the power-saving modes that are included in "Sleep" mode of the SOC. In addition to "Sleep" mode, the SOC also has two more power-down modes which are "Deep Sleep" and "Standby". The typical mode of operation is "Active" mode which is connected to other Bluetooth devices and exchanging data. "Active" mode can be changed into any of the power-down modes depending on the scheduled activity and the status. This paper also presents a description of each power-down mode and a procedure for how to implement them.

As SOC design complexity increases, the verification environment is turning into a critical bottleneck in the design process [13]. To overcome this bottleneck, a number of efforts have been made to improve the tools and languages [14]–[16]. In Bluetooth SOC, the reusable IP and integration methods are introduced [17]. Although there are several verification languages in the market today, simulation and the traditional verification method are still considered as the method of choice [18]-[20]. We also adopt the simulation-based verification methodology and improve a unified verification environment [19], [20]. An environment constructed for this purpose uses standardized RTL and scripts, does not rely on any specialized tools and is capable of running on all major simulators. This environment is also designed with reusable scripts and environments that are as tool-, platform-, and technology-independent as possible. This includes flexibility in meeting the final requirements using tools and independent technology wherever possible in various processes and for projects. This work also focuses on reducing design efforts by avoiding the same work done twice by different people and reusing the similar environment and platform for different projects. In this paper, we present the unified verification platform and the integrated environment.

Manuscript received December 9, 2009.

Manuscript revised April 28, 2010.

[†]The author is with the Computer Science Department, UCLA, CA, USA.

^{††}The author is with Korea University, Seoul, Korea.

^{†††}The author is with School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea.

a) E-mail: kbaek@cau.ac.kr

The organization of this paper is as follows. Section 2 presents the low-power SOC architecture and describes each component in the architecture; it also describes the overall power management for low-power operation. Section 3 presents a power management unit, its operation, and a description of each power-down mode and a procedure for its implementation. Section 4 discusses the unified design and verification platform. The experimental results and measurements are presented in Sect. 5. Conclusions are given in Sect. 6.

2. Low-Power SOC Architecture

Figure 1 shows the proposed low-power SOC architecture for the v2.0+EDR Bluetooth. The dual bus architecture consists of a system bus with a dual-port memory and Advanced Micro-controller Bus Architecture (AMBA). The AMBA has two bus layers which are Advanced High performance Bus (AHB) and Advance Peripheral Bus (APB). The dual bus structure in the system bus and double buffering scheme using dual-port memory are selected for efficiency in data transfers between the MCU and the link controller (LC). The dual bus allows MCU to access the instruction memory through the system bus and DMA to access the data memory which comes from the link controller, efficiently.

Generally, to control packet data, the MCU transfers packet data from LC buffers to the internal SRAM. And then, the MCU send the packet data to other blocks in terms of its purpose such as audio decoding or host data. These data transfers between LC buffers, internal SRAM and other blocks can be reduced using the double buffering scheme. The LC writes the data to the dual-port SRAM and shares it with the MCU. This scheme reduces the data transfer between the link controller and MCU, so we can reduce the power consumption. The arbiter of the system bus efficiently decides whether the MCU or the AHB owns the data bus. This chip also includes a power management unit (PMU), a memory controller, a sub-band codec (SBC), and



Fig. 1 Low-power architecture for Bluetooth SOC.

data interfaces such as USB v1.1, Timer, SPI, IIS, IIC and UART.

2.1 Data Transfer Reduction

The double buffering scheme using dual-port memory is used for reducing data transfers between the MCU and the LC. This scheme allows the MCU and the LC to share the dual-port SRAM, therefore additional data transfer is not necessary. The data transfers between the dual-port SRAM and other application blocks are controlled by the double buffering management block and the MCU.

In order to allow simultaneous access to packet data from both the LC as well as the MCU during both transmission and reception of packets, it is customary to implement a double buffered management block. The double buffered management block is located between the dual-port SRAM and the LC in Fig. 2. There is an 8 Kbyte dual-port SRAM to double buffer the packet data. The first 4 Kbytes are used for the transmit packets and the rest is used for the receive packets. Essentially, the number of available buffers with which packet transfers can take place from/to during transmission/reception is increased to more than two, also known as "multiple data buffering". Several registers defined for various options allow the firmware to use the multiple data buffering option. These registers store the status of the dualport SRAM (such as a busy state) and the ownership of memory. In the double buffering management block, one block reads the next packet from the receiver while the other processes the data. The time required to process the data should be shorter than the required time to read the memory.

2.2 Clock Gated Scheme

The power consumption in a digital system depends mainly on the clock frequency and on the switching of flipflops [21]. Figure 3 shows the clock-gated flip-flop approach, a design strategy to avoid wasted energy by eliminating unnecessary switching of the flip-flops. The clock gated method is based on the condition that the clock signal is obtained either through an enabling signal or through the flip-flop itself. In order to reduce the activity on the clock node, we can enable the clock only when data is valid. We can gate the clock signal using the clock gated cell, as depicted in Fig. 3 (b). In this work, the clock gated scheme was implemented using Synopsys's Power Compiler and does not affect the fault coverage. Also, the Power Compiler was



Fig. 2 Block diagram of the double buffering management block.



Fig.3 (a) Logic block diagram without clock gated scheme. (b) Logic block diagram with clock gated scheme.

invoked to perform power optimizations based on the toggle rates the simulation generated with delay and area constraints still present.

2.3 RF Transceiver and Baseband

The RF transceiver is optimally partitioned into analog and digital parts for high performance. The analog part is designed with differential structure that uses the single 1.8 V supply. The receiver uses the optimized Low-IF (1.5 MHz) architecture which is a trade-off between power and performance in CMOS technology [4], [7], [8]. This block uses the image rejection Gilbert mixer and the controllable AGC. In the transceiver the digital part consists of modulators and demodulators. And the demodulation type of digital down converter consists of an LPF, a digital mixer, a decimator, and an IO demodulator selected by the header information from the LC. We use the energy squaring method which includes a differential detector and symbol timing recovery for sync detection. The transmitter uses direct up conversion. The firmware controls the output swing level of the PGA depending on the transmission channel and keeps the modulation index constant.

The power-down control signals and the simplified block diagram of the RF transceiver are shown in Fig. 4. There are three power-down control signals; "RX_PD" for RF receiver, "PLL_PD" for PLL block, and "TX_PD" for RF transmitter. The power management unit generates these signals to control the power state and transition of each block. When each of the power-down control signals is high, each block goes to the power-off state. The power management unit drives the power-down control signals to low in order to wake up the RF transceiver from "Sleep" mode according to the wake-up sequence. The power control sequence consists of the stabile time of the PLL, the VCO tuning time, the ramp-up time in TX mode and the phase calibration time in RX mode. The detailed descriptions of



Fig. 4 Simplified block diagram of RF transceiver.

the transitions between the active and power-down modes are presented in Sect. 3.

The link controller performs the entire required data packet processing in the Bluetooth standard. The LC supports all Bluetooth data types which consist of guard time, SYNC field, forward error correction, CRC calculation, data whitening and data payload. The structure of the LC is centered on a dual-port SRAM which is shared with the MCU. By the state machine of the LC, the packet header is assembled from the link context information. This header is appended to the appropriate payload data stream for supporting 2 M bps and 3 M bps of EDR. The 2 M or 3 M packet type, the guard time, and the SYNC field are added for PSK synchronization of EDR. The transceiver changes the modulation method between GFSK, $\pi/4DOPSK$ or 8DPSK according to the transmission type. The clock scheme is changed depending on the 2 M or 3 M data rates. The LC precisely controls the modem and the radio function for the selection of the packet type.

3. Power Management Unit

The power management unit (PMU) consists of the clock generator and the power-down controller of the analog blocks. The clock generator allows the MCU and each peripheral to use various divided clocks in each of the power mode. The power-down controller is designed to generate power-down control signals according to the power down and wake up sequence. We present the mode and the sequence in detail in this section. This chip supports four modes including the three types of power-down modes shown in Table 1.

3.1 Power-Down Modes

The power-down modes consists of "Sleep", "Deep Sleep" and "Standby". "Active" mode is the normal operation mode which is connected to other Bluetooth devices and exchanging data. Depending on the Bluetooth protocol and on the host's activity, the chip autonomously decides the proper power mode to use. "Standby" mode is entered only after an explicit command from the host is issued. In "Sleep"

Mode	Descriptions			
"Active"	Accepts HCI commands. Supports full and low			
	speed of the MCU and all types of Bluetooth con-			
	nections, links and exchange data.			
"Sleep"	Accepts HCI commands. Supports low speed			
	of the MCU and all types of Bluetooth's power-			
	saving and search modes ("Sniff", "Hold",			
	"Park", "Page", and "Inquiry").			
"Deep Sleep"	Does not accept HCI commands. The MCU			
	is powered off. Allows a transition to "Sniff",			
	"Hold", "Park", "Page", or "Inquiry" modes.			
"Standby"	Most parts of the chip are powered off. Allows a			
	transition to "Sniff", "Hold", "Park", "Page", or			
	"Inquiry" modes.			

 Table 1
 Descriptions of active and power-down modes.

mode, the proposed SOC supports the search and powersaving modes defined in Bluetooth specifications, which are "Sniff", "Hold", "Park", "Page" and "Inquiry" modes. In "Deep Sleep" mode, this chip does not accept HCI commands in the Bluetooth protocol. In "Inquiry", a Bluetooth device either tries to discover a nearby device, or it is being discovered by other devices within its locality. During "Inquiry", this device does not use any of the architectural layers above the physical channel. "Page" is used only for forming connections. "Hold" mode operates only once for each invocation and exits when it is completed, returning to the previous mode. In this mode, the physical link is only active for the synchronous connection oriented (SCO) link. "Sniff" mode is used for the asynchronous link (ACL). In "Park" mode, the device cannot support any logical links but it has a physical link in a piconet. The device may engage in activity on other physical channels, or enter power-down modes.

Figure 5 shows transitions between states which consist of the "Active", "Sleep", "Deep Sleep", and "Standby". State transitions are caused by control signals and commands. The "Hard Reset" is the hardwired reset signal which comes from an external pin. After the "Hard Reset" is released, the Bluetooth device goes into "Sleep" mode and is waiting the data request command from the host. When the host sends a Bluetooth link or a connect command, the chip goes into "Active" mode and tries to discover the other device or waits to be discovered. Otherwise, the chip stays in "Sleep" mode. The "Specific PIN" is needed for detecting the wake up command of the host from "Deep Sleep" mode. This pin can be assigned to the GPIO or the serial interface pin for this purpose. The "PDOWN" command is to enter into "Standby" mode. When "Specific PIN" is asserted in "Standby" mode, the state is transferred to "Sleep" mode while keeping the previous register values. On the other hands, "Hard Reset" forces any state to be transferred to "Sleep" mode while initializing all register values. The "DSLEEP" is the command for entering into "Deep Sleep" mode. The time of "Sleep" mode is programmed by the MCU. When the time expires, the chip leaves "Deep Sleep" mode and enters into "Standby" mode.



Fig. 5 State machine for active and power-down modes.

3.2 Power Management Unit

The power management unit controls the power-off state of each block according to the slots state of the Bluetooth device. The slot can either be the transmitted slot (TX) or the received slot (RX). Each slot period is set to $625 \,\mu s$. Figure 6 shows the timing chart for the transmission and reception slots. The type of Bluetooth link and the host command decide the slot states including connected, "Inquiry", or "Page". The Bluetooth device uses a time division duplex scheme that is controlled by a counter that counts from 0 to 1249 at 1 μ s interval. The packet assembler starts transmission when the counter is zero and the packet dis-assembler starts reception when the counter hits 625. The packet assembler clock enables the TX mode while the packet disassembler clock enables the RX mode. For this reason, an independent native bit count, as well as a native Bluetooth clock must be maintained.

The native Bluetooth clock is counted by the native counter with a clock frequency of 1 MHz in "Active" mode. In "Sleep" mode, the native Bluetooth clock is maintained in the 3.2 kHz domain (low-power clock). The low-power clock was chosen from 3.2 kHz or 32 kHz to allow for a resolution of 312.5 ms. Depending upon exiting "Sleep" mode, the native bit counter is updated to the correct value by use of a continuously maintained native Bluetooth clock. In "Standby" mode, this chip does not maintain the native Bluetooth clock and awaits the host command to enter into "Sleep" or "Active" mode.

The timing diagram of the power-down control for the transition between sleep and normal modes is shown in Fig. 7. The "PLL_PD" signal is the power-down signal for the PLL block, the "TX_PD" is the power-down signal for the RF transmitter and modem, and the "TX_DATAEN" is power-down signal for the link controller. The PMU drives the "PLL_PD" signal to low in order to wake up the radio



Fig. 6 Timing chart of transmission and reception slots.



Fig. 7 Timing diagram of power-down control at TX slot.



Fig. 8 Shutdown and wakeup logic for "Standby" mode.

from "Sleep" mode. When the "PLL_PD" is low, the wakeup timer directly controls this signal due to the required stable time for the PLL block.

After waiting for the stable state of PLL, the MCU writes the hop channel value to the hop channel register of RF block and the VCO is tuned to the Bluetooth radio frequency during the PLL setup state. When the link controller encounters the start time of the slot, the MCU sends the TX command. In the ramp updown state, the modulation block slowly increases or decreases the slope of the baseband signal to prevent abrupt transitions of the signal. After that, the MCU sets the "TX_DATAEN" to low and this chip transfers the data to the other Bluetooth device. When the data transfer is completed, the "TX_DATAEN" goes to high, the RF transmitter enters the sleep state through the ramp down state, and then the "PLL_PD" signal goes to low and PLL blocks goes to "Sleep" mode through the stable state. The MCU also enters either "Sleep" mode or "Standby" mode. In "Standby" mode, the PMU shuts down the 12 MHz oscillator and it is used to clock the baseband, the link controller and the 32 kHz wake-up timer. Figure 8 describes the shut down logic of the oscillator and the wakeup logic for "Standby" mode. "Standby" mode is decided the PDOWN command of the host (in Fig. 5). The "Wake" signal in Fig. 8 comes from the "Specific PIN" which indicates the external data request or the wakeup command (in Fig. 5).

4. Design and Verification Platform

We present the unified design and verification platform that includes the tools environment, technology parameters, the synthesis and the static timing analysis (STA) scripts. We developed the unified platform based on the previous works [19], [20] using the Makefile, Perl, Tcl and a version control program. The important thing of the unified platform is the centralized data structure of designs, constraints and characterization information [18]. The Makefile utility is the best utilities for maintaining a group of programs. The purpose of the Makefile utility is to automatically determine programs that are needed to be recompiled, to re-make itself as others commit their changes to the repository, and to manage targets and dependences which consist of the execution, the compile and regressions. This management of the make script allows to avoid issues related to stale the make script which consists of a common-part including tools, parameters on the base project, and a core-part including a module, a test-specific configuration, components lists, clock characteristics, and constraints. Figure 9 shows the scripts structure of the unified verification platform. The make script generates the data paths and the variables which are a module name, a sub-module, and a specific configuration. The data paths are used for the library and module scan in the RTL simulator and for the run script of RTL simulation. The priority of the search path can be controlled by changing the configuration. The other part is to develop the environment of tools and technology parameters. The make script assembles the executable target and commands for each design flow.

4.1 Database Structure

A database structure consists of a concurrent version system repository, common data, and user data. Figure 10 shows the data structure. During the development of the project, it is important to have a "snapshot" of the sources used for the build. In this way, we do not need to depend on the source files being stable over any period of time. A project data structure consists of the project environment, the full-chip and its intellectual property (IP). Because the full-chip and IP structure have the same sub-directories, it is easy for all of the users to manage and access the database. The subdata structure consists of Makefile scripts, RTL, test-bench, synthesis scripts, and a test driver which includes a bus functional model (BFM) and a core functional model. The project environment has common scripts, Tcl and a tools environment that includes technology parameters and is used across the components of the entire project.

4.2 SOC Verification Platform

The reusability of verification environments helps to reduce the SOC design time [17]. With this approach, we can enhance the reusability of verification components that are as-



Fig. 9 Script structure of verification platform.



Fig. 10 Structure of database.

sociated with design under verification (DUV) in the SOC level and across the entire SOC. In addition, it is also important that the block verifications are easily ported to system level environment as well. The main purpose of doing this is to perform sanity checks on the module. The test cases such as module register access, interrupts triggering, and its basic functionality can be ported depending on the complexity. If the IP verification platform can be reused partly or completely in the SOC verification platform, the design cost can be reduced and verification can be improved. By implementing the steps above, the verification engineer can focus more on SOC level checks instead of recreating test cases for checking the integration of the module into the system. We designed the SOC verification platform similar to the IP verification platform for reusability, and we can verify the IP function with a C-program that is converted into a ROM file.

The overall synthesis, compilation and optimization strategies may be relatively constant, but there are various choices to be made to manage the tradeoffs in a power, timing, an area, and a run time. The execution of the synthesis process is controlled by the make script. The each synthesis step is controlled by the target options. The synthesis script consists of common project data, the clock, and the constraints script in each block. This script shares these con-

straints with the synthesis process and allows for the annotation of post layout extracted parasitic. Therefore, we expect to make appropriate choices without having to reverseengineer the details in the scripts and without risking introduction of bugs into the scripts. In the compilation and execution step, the options can be controlled and selected in the flow of the overall synthesis script by the design intents and specifications. Therefore, we should have the benefits of reusable and reconfigurable synthesis scripts. If an engineer needs to modify his own synthesis script, he can modify the part of whole script which has the variable of each IP without risking the introduction of a bug into such scripts and without understanding the overall synthesis process. Each block is expected to represent a unique set of challenges corresponding to the added flexibility inherent in its form.

4.3 Implementation

We implement this unified verification platform on a Bluetooth SOC. For verification, we build a test bench including functional models of a packet processor, RF modeling, BFM, and a Protocol Monitor. This verification environment is excellent in reusing components that verify each block and the full-chip. For a synthesis process environment, we build synthesis, clock, and constraint scripts. We need to reconstruct the proposed environment form from the previous project, but we get more advantage and reusability in the third project. All the IP block environments are changed into the sub-data structure form. The blockdependent environment and the sub-block are reused in the system level. The simulation test environment is written in HDL, and optional components are selected by generic names and test-specific configurations. This unified platform is helpful in reducing the design time and improving its reusability. A test vector and various scripts that are used in each block are reused again on the full-chip level. Then the IP, the full-chip, and its interconnection are concurrently designed and verified. When the project started, the development IP is completed and the full-chip designer co-works with one designer for each block. The design time is the combination of the time for SOC integration, the time for verification and the backend time.

5. Measurements

The proposed Bluetooth SOC is measured at 1.8 V/24 MHz. Comparisons between the previous work and the proposed architecture are given in Table 2, which is showing both the simulation results (normal and power compile cases) and measurement data. We measure the power consumption at the worst-case condition. The proposed architecture uses the dual-port memory with the double-buffering management block and the clock gated scheme, which are different from the previous work [2]. The previous work used the internal single-port memory for MCU, and TX/RX buffers are used only for the link controller. Therefore, when MCU

Measured currents for different modes.

	This Work					
	Simulation	(HV1 mode)		Previous		
Section	Normal Compile	Power Compile	Measured (HV1 mode)	Work (Measured) [2]		
Gate Count	14023345	13204287	13204287	14344436		
Worst-case Total Current (Digital)	72 mA (45 mA)	59 mA (32 mA)	54 mA (26 mA)	65 mA (32 mA)		

Table 2Performance comparison.

needs to transfer the data from host to another Bluetooth device, it has to save data in the internal memory and copy them to TX buffer again. On the other hands, the proposed design with dual-port memory can reduce transfer data rate by half, because the link controller and MCU share the dualport memory. The clock gated scheme also reduces the both dynamic and static powers. This scheme achieves 5.8% reduction in the number of gates (including sequential logics such as F/Fs) and it eliminates unnecessary clock switching of the flip-flops. When we run power analysis tool, the power compile case shows 29% reduction in the power dissipation compared to the normal compile case. By using aforementioned methods, the total digital current reduction of 6 mA is achieved in the proposed design: the current reductions of 1.2 mA and 4.8 mA come from data transfer reduction and clock gated scheme, respectively. Therefore, 18.7% power reduction is achieved in the digital block compared to the power dissipation of the digital block in the previous work [2], which contributes to 12.4% reduction out of 17% total power-saving. Note that 4.6% power reduction comes from the improved RF block. Because this paper is focused on improvements in the digital block, detailed description about the RF block is not presented in this paper.

The High-quality Voice 1 (HV1) mode sends the voice data at 1.25 ms intervals. The transfer cycle of High-quality Voice 2 (HV2) and High-quality Voice 3 (HV3) are 2.5 ms and 3.75 ms, respectively. The internal SRAM (40 K) of the previous work consumes about 1.5 mA at HV1 TX slot. But the dual-port SRAM (8 K) consumes only 0.3 mA at the proposed method without using the internal SRAM.

Table 3 summarizes the measured currents for different modes. We measure currents of the SCO and ACL connections in TX operation, because the worst-case current occurs in TX timing slot. Note that more components generally operate in TX than in RX, and the power amp (PA) which is one of power hungry blocks is disabled in RX operation. The RF block includes the transmitter, receiver, and PLL block, and the digital block includes the link controller, modem, MCU, and peripherals. The SCO packet is divided into three types; HV1, HV2 and HV3. In Table 3, currents of the SCO packet are measured at minimum and maximum intervals, which are HV1 and HV3, respectively. ACL packet is divided into three types; auxiliary 1 (AUX1), data-medium rate (DM) and data-high rate (DH). In Table 3, currents of

	Mod	9	DE	Digital	Total
widde			KI [*]	Digitai	Totai
		HV1	28 mA	26 mA	54 mA
TX SCO		HV3	12 mA	10 mA	22 mA
Operation		3 DH5	25 mA	20 mA	45 mA
	ACL	DH5	12 mA	13 mA	25 mA
Power Down		"Sleep" ("Park" case)	4.5 mA	3 mA	7.5 mA
		"Deep Sleep"	76 µA	1 mA	1.076 mA
		"Standby"	76 µA	0.5 µA	76.5 μA

Table 3

the ACL packet are measured at low and high speeds with 5 slots per packet, which are 3DH5 and DH5, respectively. Note that the first number of ACL packet name means the data rate and the last number of the name means the number of slots per packet. For example, 3DH5 means a data-high rate which has 3 Mbps data rate and 5 slots per packet. Currents in power-down modes are also reported in Table 3 to show current differences between active and power-saving modes. In "Sleep" mode which is one of three power-down modes, there are search and power-saving modes as mentioned in Sect. 3.1; "Sniff", "Hold", "Park", "Page", and "Inquiry". Among these five modes, we report the current of "Park" case in "Sleep" mode, because "Park" case has only physical link without data exchange, thus causing the lowest power consumption. In "Deep Sleep" and "Standby" modes, the oscillator is disabled and a 32 kHz low-power clock is used. The difference between the two modes is that the serial interface block is partially active in "Deep Sleep" mode.

Figure 11 shows the die micrograph of the Bluetooth SOC. The total chip area is 30 mm^2 in $0.18 \,\mu\text{m}$ CMOS. The RF transceiver is located in the lower-right corner and the transmitter part consists of a PA, PGA, and up-conversion mixer. The receiver front-end consists of an LNA, a down conversion mixer, and an AGC. The digital part covers roughly $\frac{1}{2}$ of the area, the memory takes up $\frac{1}{4}$ of the area, and the RF-part takes up another $\frac{1}{4}$ of the entire chip area.

6. Conclusion

A low-power architecture for a v2.0+EDR Bluetooth SOC has been presented. The measurement results meet the Bluetooth specification v2.0+EDR. This paper presents the simulation and measurement results of power consumption. The proposed architecture fulfills the main design goals of reducing the data transfer and its power consumption. This chip supports various low-power modes. We achieved the 17% power reduction compared to the total power of the previous works by improving both digital and RF blocks. We also show that our unified verification platform is able to increase reusability, modularity, and productivity by reusing the environment and the common database structure. SOC design flow has repetitive works when its functions are implemented and its performances are verified. To increase efficiency of these works, it is important to have reusability and modularity of source code in SOC design environ-



Fig. 11 Die micrograph of the Bluetooth SOC.

ment. The proposed unified verification platform is designed with considering these factors. The reusability is how much source codes can be used again with slight or no changes for other similar projects or even new one. We use a reusable module and classes in environment script and test vectors with a common interface. To increase reusability, we use modular programming technique (modularity) which is composed of modules and common sub-routines. This reusability reduces implementation time, so that we can increase productivity. Therefore, this paper shows that the proposed design and verification platform allow the SOC design to be low-power operation and to be reduced design time so that meet the time-to-market goal.

Acknowledgments

This work is supported by "System IC 2010" project of Korea Ministry of Knowledge Economy. The authors would like to thank K.T Moon, J.W. Jeong and T.S. Bang for the RF, SBC and the link controller design. A preliminary version of portions of this paper has been presented at 2008 Digest of Technical Papers International Conference on Consumer Electronics (ICCE2008) [22].

References

- Specification of the Bluetooth System, Core Specifications, version 2.0+EDR, v1.2, v1.1, https://www.Bluetooth.org/spec/
- [2] J. Kim, J. Jeong, S. Lee, and S. Kim, "The v2.0+EDR bluetooth SOC architecture for multimedia," IEEE Trans. Consum. Electron., vol.52, no.2, pp.436–444, May 2006.
- [3] B.A. Georgescu, J. Nakaska, R. Randall, and J.W. Haslett, "A 0.18 μm CMOS bluetooth frequency synthesizer for integration with a bluetooth SOC reference platform," IEEE Int. Workshop on System-on-Chip for Real-Time Applications, pp.258–263, Calgary, Alberta, Canada, July 2003.

- [4] H. Ishikuro, M. Hamada, K. Agawa, S. Kousai, H. Kobayashi, D.M. Nguyen, and F. Hatori, "A single-chip CMOS bluetooth transceiver with 1.5 MHz IF and direct modulation transmitter," Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Paper, pp.272–273, San Francisco, CA, USA, Feb. 2003.
- [5] R.B. Staszewski, C.-M. Hung, K. Maggio1, J. Wallberg, D. Leipold1, and P.T. Balsara, "All-digital phase-domain TX frequency synthesizer for Bluetooth radios in 0.13 μm CMOS," Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Paper, pp.272–273, San Francisco, CA, USA, Feb. 2004.
- [6] C. Durdodt, H. Friedrich, C. Grewing, M. Hammes, A. Hanke, S. Heinen, J. Oehm, D. Pham-Stabner, D. Seippel, D. Theil, and S. van Waasen, "The first near zero-IF RX, 2-point modulation TX CMOS SOC bluetooth solution," IEEE Conf. on Custom Integration Circuit, pp.429–432, San Diego, CA, USA, April 2001.
- [7] F.O Eynde, J.-J. Schmit, V. Charlier, R. Alexandre, C. Sturman, K. Coffin, B. Mollekens, J. Craninckx, S. Terrijn, A. Monterastelli, S. Beerens, P. Goetschalckx, M. Ingels, D. Joos, S. Guncer, and A. Pontioglu, "A fully-integrated single-chip SOC for bluetooth," Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Paper, pp.196–197, San Francisco, CA, USA, Feb. 2001.
- [8] W.W. Si, D. Weber, S. Abdollahi-Alibeik, M. Lee, R. Chang, H. Dogan, G. Haitao, Y. Rajavi, S. Luschas, S. Ozgur, P. Husted, and M. Zargari, "A single-chip CMOS bluetooth v2.1 radio SOC," IEEE J. Solid-State Circuits, vol.43, no.12, pp.2896–2904, Dec. 2008.
- [9] J. Cheah, E.-H. Kwek, E.C. Low, C.K. Quek, C. Yong, R. Enright, J. Hirbawi, A. Lee, H. Xie, L. Wei, L. Luong, J. Pan, S.-T. Yang, W.F.A. Lau, and W.-L. Ngai, "Design of a low-cost integrated 0.25 μm CMOS bluetooth SOC in 16.5 mm² silicon area," Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, pp.90–91, San Francisco, CA, USA, Feb. 2001.
- [10] H. Darabi, S. Khorram, E. Chien, M. Pan, S. Wu, S. Moloudi, J.C. Leete, J.J. Rael, M. Syed, R. Lee, B. Ibrahim, M. Rofougaran, and A. Rofougaran, "A 2.4 GHz CMOS transceiver for bluetooth," Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, pp.200–201, San Francisco, CA, USA, Feb. 2001.
- [11] K. Muhammad, D. Leipold, B. Staszewski, Y.-D. Ho, C.M. Hung, K. Maggio, C. Fernando, T. Jung, J. Wallberg, J.-S. Koh, S. John, I. Deng, O. Moreira, R. Staszewski, R. Katz, and O. Friedman, "A discrete-time Bluetooth receiver in a 0.13 μm digital CMOS process," Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, pp.268–269, San Francisco, CA, USA, Feb. 2004.
- [12] S.-B. Hyun, G.-Y. Tak, S.-H. Kim, B.-J. Kim, J. Ko, and S.-S. Park, "A dual-mode 2.4 GHz CMOS transceiver for high-rate bluetooth systems," ETRI J., vol.26, pp.229–240, June 2004.
- [13] J.M. Pendleton and C. Burns, "An integrated CAD environment for system design," Proc. Int. Conf. on System Sciences, vol.1, pp.39– 48, Kailua-Kona, HI, USA, Jan. 1989.
- [14] G. Falconeri, W. Naifer, and N. Romdhane, "Common reusable verification environment for BCA and RTL models," Proc. Design, Automation and Test in Europe, vol.3, pp.272–277, Messe Munich, Germany, March 2005.
- [15] M. Bocchi, C. Brunelli, C. De Bartolomeis, L. Magagni, and F. Campi, "A system level IP integration methodology for fast SOC design," Proc. Inter. Symp. on System on Chip, pp.127–130, Tampere, Finland, Nov. 2003.
- [16] A. Sagahyroon, G. Lakkaraju, and M. Karunaratne, "A functional verification environment," 48th Midwest Symposium on Circuits and Systems, pp.108–111, Kobe, Japan, May 2005.
- [17] T. Grahm and B. Clark, "SOC integration of reusable baseband bluetooth IP," Proc. Design Automation Conf., pp.256–261, Las Vegas, NV, USA, June 2001.
- [18] R. Wang, W. Zhan, G. Jiang, M. Gao, and S. Zhang, "Reuse issues in SOC verification platform," The 8th international conference on computer supported cooperative work, pp.685–688, Helsinki, Finland, Sept. 2003.
- [19] J. Kim, K. Moon, J. Jeong, and S. Kim, "A platform based SOC

design environment," IP Based SOC Design Conference and Exhibition, pp.426–471, Grenoble, France, Dec. 2006.

- [20] J. Kim, J. Jeong, T. Bang, and S. Kim, "Integrating makefile, tcl and perl into SOC-based design environments," 3rd IEEE Inter. Conf. on Circuits and Systems for Communications, pp.193–198, Bucharest, Romania, July 2006.
- [21] T. Lang, F. Musoll, and J. Cortadella, "Individual flip-flops with gated clocks for low power data paths," IEEE Trans. Circuits Syst. II, vol.44, no.6, pp.507–516, June 1997.
- [22] J. Kim, K. Moon, J. Jeong, Y. Kim, and S. Kim, "A low power SOC architecture for the V2.0+EDR bluetooth," Int. Conf. on Consumer Electronics Dig. Tech. Paper, pp.1–2, Las Vegas, NV, USA, Jan. 2008.



Kwang-Hyun Baek received his B.Eng. degree in electronics and computer engineering in 1990, M.Eng. degree in electronics engineering in 1998 both from Korea University, Seoul, Korea, and his Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign (UIUC), Illinois, USA in 2002. From 1990 to 1996, he was with Samsung Electronics and developed several microcontrollers including a 32-bit RISC-type embedded system. From 2000 to 2006, he was with the

Department of High-Speed Mixed-Signal ICs as a senior scientist at Rockwell Scientific, formerly Rockwell Science Center (RSC), CA, USA. Since 2006 he has been with the School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea where he is a faculty member. His research interests include high-performance analog and digital integratedcircuit design, modeling, and synthesis.



Jeonghun Kim received his Ph.D degree in electronics and computer engineering from Korea University, Seoul, Korea, in 2008. Currently, he is with the VLSI CAD laboratory at the Computer Science Department of UCLA, CA, USA as an assistant researcher. He is working on SOC design/verification automation for low-power designs and power-aware verification for high-level synthesis. From 2007 to 2008, he worked in the ISD SOC verification team at Magnachip semiconductor. From 2005

to 2006, he worked in Wireless SOC R&D center, Mewtel technology. From 1993 to 2004, he worked for Samsung Electronics where he did research on the optical storage servo system and SOC design and methodology. His research interests are in image signal processing, system-on-chip design methodology, power aware verification, and automation.



Suki Kim received his PhD degree in electrical engineering from University of Minnesota in 1980. He was with AT&T Bell Lab from 1980 to 1985. He was with Hughes, USA from 1988 to 1990. In 1990, he was Vice President of Samsung Electronics, Inc., Korea. From 1990 to 1995, he was an Adjunct Professor at the University of Minnesota in Minnesota, USA. Since 1995 he has been with the Department of Electronics Engineering, Korea University. His research interests are in mixed mode IC design

and the virtual prototyping of micro systems for system-on-chip applications.