

PAPER

Divided Static Random Access Memory for Data Aggregation in Wireless Sensor Nodes

Takashi MATSUDA^{†a)}, Shintaro IZUMI[†], *Members*, Yasuharu SAKAI[†], *Nonmember*, Takashi TAKEUCHI[†], Hidehiro FUJIWARA[†], Hiroshi KAWAGUCHI[†], Chikara OHTA[†], and Masahiko YOSHIMOTO[†], *Members*

SUMMARY One of the most challenging issues in wireless sensor networks is extension of the overall network lifetime. Data aggregation is one promising solution because it reduces the amount of network traffic by eliminating redundant data. In order to aggregate data, each sensor node must temporarily store received data, which requires a specific amount of memory. Most sensor nodes use static random access memory (SRAM) or flash memory for storage. SRAM can be implemented in a one-chip sensor node at low cost; however, SRAM requires standby energy, which consumes a lot of power, especially because the sensor node spends most of its time sleeping, i.e. its radio circuits are quiescent. This study proposes two types of divided SRAM: equal-size divided SRAM and equal-ratio divided SRAM. Simulations show that both proposed SRAM types offer reduced power consumption in various situations.

key words: wireless sensor network, data aggregation, data storage management, SRAM

1. Introduction

Recent advances in wireless communication technology and electronics have enabled the development of low-power wireless sensor networks. These networks are attracting much attention because they can easily collect a wide range of information over wide spatial extents. Most sensor nodes are battery powered and distributed over such wide areas that it is difficult to replace their batteries. Battery exchange frequency must, therefore, be reduced in order to improve the efficiency of wireless sensor networks and extend system lifetime.

Developing a one-chip LSI for a wireless sensor node can lower the manufacturing cost and node size, and reduce battery exchange frequency. In order to create a one-chip sensor node, upper layer network protocols and lower layer hardware must be designed concurrently.

Data-centric routing is a promising paradigm for sensor network routing [1]. In this paradigm, a sensor node can buffer several packets temporarily, eliminate redundant information, generate a new packet, and then send it to the next hop. This style of operation, termed data aggregation, is expected to reduce the amount of transmitted data, and so yield remarkable power savings. To the best of our knowledge, however the power needed to buffer the packets in the node has not been considered in past studies even though it is not negligible as shown later.

Conventional sensor nodes primarily use either SRAM or flash memory for data storage. Both types of memory have their advantages and disadvantages. SRAM, for example, is easily implemented as one chip, it has low cost, and its active power consumption is small. However, SRAM must be continually driven to retain data, and, as a result, its leak current is significant. Flash memory, in contrast, consumes no leak current while the sensor node is sleeping, but the cost of a one-chip sensor node is higher with flash memory than with SRAM. In addition, the active power consumption of flash memory is high, and it cannot provide high-speed memory access. Moreover, flash memory has a finite number of erase-write cycles which does not suit its application to the packet buffer, which must be written and read many times. Therefore, this paper considers an energy-efficient SRAM architecture that can be implemented in a one-chip sensor node at low cost.

The leak current of SRAM increases in proportion to SRAM capacity, while data aggregation becomes more efficient as SRAM capacity is increased. This leads to a trade-off between data aggregation efficiency and power consumption reduction. In general, however, it is difficult to determine the SRAM capacity in advance since the optimum SRAM capacity varies with the type of application. To increase SRAM support for various applications, we propose two types of divided SRAM: equal-size divided SRAM and equal-ratio divided SRAM. The divided SRAM effectively supports the various applications that utilize data aggregation. In what follows, however, we assume a data collection application to evaluate the effectiveness of the divided SRAM architectures since it is typical of how sensor networks will be used.

The remainder of the paper is organized as follows. Section 2 examines the types of memory used in sensor nodes. Section 3 investigates the relationship between data aggregation and SRAM capacity from the viewpoint of power consumption. In Sect. 4, we propose a divided SRAM that operates only when needed to record data packets. Our proposed approach is compared to the conventional scheme in Sect. 5. Section 6 summarizes our findings.

2. Relationship between Data Aggregation and SRAM Capacity

We expect that many sensor nodes will aggregate data packets by temporarily buffering them and then generate a new

Manuscript received March 17, 2011.

Manuscript revised September 3, 2011.

[†]The authors are with the Graduate School of Science and Technology, Kobe University, Kobe-shi, 657-8501 Japan.

a) E-mail: matsuda@cs28.cs.kobe-u.ac.jp

DOI: 10.1587/transcom.E95.B.178

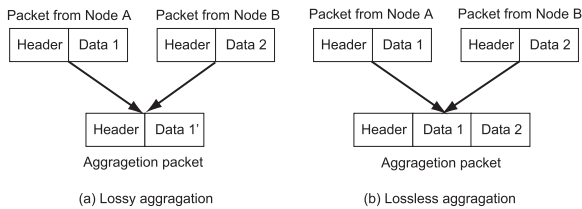


Fig. 1 Data aggregation type: Lossy aggregation and lossless aggregation.

packet that is sent to the next hop. This is expected to reduce the amount of transmitted data, resulting in remarkable power savings.

Data aggregation can be categorized into two classes: lossy and lossless [2].

Perfect aggregation [3] and beam-forming [4] are lossy aggregation (Fig. 1(a)). With perfect aggregation, a sensor node aggregates received data into one unit of data and then sends it to the next hop, where average, maximum, and count operations are examples of perfect aggregation functions [5]. Such an operation can remarkably reduce the amount of transmitted data. Perfect aggregation is quite efficient in this sense, whereas available applications are limited. In [4], beamforming is proposed as a data aggregation scheme to collect acoustic data. Beamforming itself is an algorithm to combine acoustic signals from multiple sensors for minimizing mean square error, maximizing signal-to-noise ratio, or minimizing variance. Data from neighboring sensors tend to be highly correlated. In [4], a sensor network forms several clusters which consists of two or more sensor nodes. Acoustic signal observed by each sensor node is transmitted to its corresponding cluster-head. The cluster-head chooses weights of FIR (Finite Impulse Response) filter for signals sensed by sensor nodes by means of beamforming algorithm. Then the resultant weighted linear sum of the signals is transmitted from the cluster-head to the sink node. Consequently, the amount of transmitted data, i.e. the energy to transmit data is reduced although operation of beamforming algorithm consumes power. In [4], it is shown that beamforming algorithm is energy-efficient in total. Although lossy aggregation schemes has a high compressibility, some information can be lost.

Linear aggregation [3] and coding by ordering [6] (Fig. 1(b)) are examples of lossless aggregations. Linear aggregation performs a simple operation: header elimination. A sensor node concatenates the payloads of buffered packets whose next-hops are equal and then puts it into one packet. Coding by ordering is an improvement of linear aggregation. In linear aggregation, the sequence of concatenated data (payload) is meaningless. On the other hand, coding by ordering gives numerical meaning to the sequence of concatenated data. For instance, there are six ways to sequence three different data. Therefore, the sequence can have $\log_2 6$ bits of information. This implies that properly sequencing some of data represents the content of the others, which need not be transmitted. In [6], the relationship between the

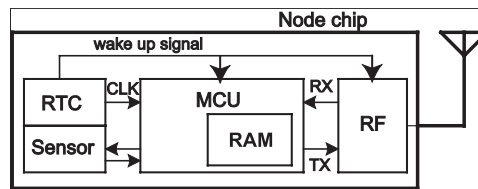


Fig. 2 Functional block diagram of typical sensor node.

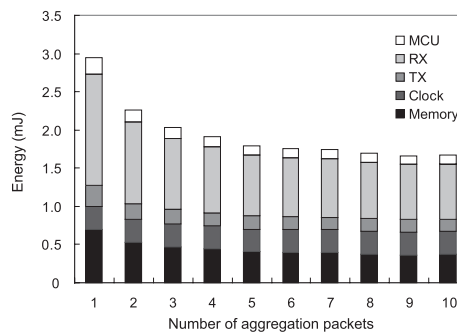


Fig. 3 Energy consumption, which includes MCU, RX and TX (RF), clock, and memory, when the maximum number of aggregated packets is changed (SRAM capacity = 1,024 bytes).

number of data and that of suppressed data is discussed. In lossless aggregation schemes, all raw data is transmitted to sink node although the compressibility is low.

In this paper, we assume linear aggregation (header elimination). Linear aggregation has lower compressibility than other aggregation schemes. It, however, supports all applications and requires no complex calculations. Linear aggregation is simple but the most energy-consuming to transmit data among all lossless aggregation variants. Therefore, the energy consumption to store data will become relatively lower compared to that to transmit data. Under such the assumption, we will investigate whether the proposed divided-SRAMs are effective or not in total.

Next, power consumption of the sensor node module was analyzed with respect to data aggregation and SRAM capacity.

Figure 2 shows a typical sensor node consisting of five modules: radio frequency (RF), random access memory (RAM), micro controller unit (MCU), real-time clock (RTC), and sensors. The power consumption of a sensor strongly depends on the type of sensor, and the type of sensor in turn depends on the application. The power consumed by the sensor is, therefore, not considered in this paper.

Figure 3 shows the maximum number of aggregated packets and the sending and receiving energy requirements; data packet size is 8 bytes and the header size is 4 bytes including one byte trailer (simulation conditions are described in Sect. 5). (We assume that each unit of sensing data consists of a pair of 4 byte attribute and 4 byte value. The header has 1 byte destination node ID, 1 byte source node ID, 1 byte control field, and 1 byte frame check sequence.) The active time of each module is shown in Table 1. These values are estimated by the simulation shown in Sect. 5, and they are

Table 1 Active time of MCU, RX and TX (RF), clock, and memory (SRAM capacity = 1,024 bytes, maximum number of aggregation packets = 5).

Module	Active time
MCU	241.5 ms
RX	219.0 ms
TX	22.5 ms
Clock	600.0 s
RAM (Leak)	33.3 s
RAM (Read/Write)	2.1 μ s

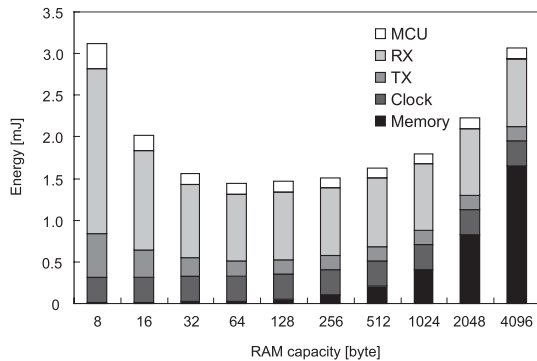


Fig. 4 Energy consumption of MCU, RF (RX and TX), clock, and memory when SRAM capacity is changed (maximum number of aggregation packets = 5).

calculated as average active times over sensor nodes that are one-hop away from the base station. The simulation time is 600 s and the clock is always powered on during simulation. MCU is only operated when TX or RX is powered on. This is because memory access can be finished within RF communication time. With linear aggregation, we assume that the sensor node performs aggregation when as many packets as a predefined number called “the maximum aggregation packets” are buffered or SRAM becomes full. For instance, when SRAM capacity is 32 bytes, the sensor node will aggregate four data units.

TX and RX energy and memory energy consumption are reduced following data aggregation and the subsequent reduction of traffic. This effect saturates when more than five data packets are aggregated. If the ratio of the header to data changes, the effectiveness of data aggregation also changes; larger headers make the header aggregation method more effective. Unfortunately, transmission errors occur more readily if long aggregated packets are used.

Figure 4 shows the energy consumption of the sensor node versus SRAM capacity. The effect of data aggregation is large (a dramatic reduction in energy consumption) at SRAM capacities between 8 and 64 bytes; this is because the increased data capacity of the relay node shortens the data hold times, which allows smoother relay. When SRAM capacity exceeds 128 bytes, its leak current rises, increasing overall energy consumption. Each read and write operation consumes a significant amount of power; however, they have very short durations. As a result, they consume, overall, much less energy than memory maintenance. Since

advances in process technology are further reducing the power consumed by read/write operations, the importance of tackling the memory maintenance problem will continue to strengthen. Thus, there are significant trade-offs between power consumption and SRAM capacity when using data aggregation.

3. Divided SRAM

The previous section revealed that there is an optimal SRAM capacity for data aggregation. The optimal SRAM capacity varies with the sensor network application, e.g. the data size. To cope flexibly with the diversity of sensor network applications, we propose two types of divided SRAM: equal-size divided SRAM and equal-ratio divided SRAM. The underlying principle of divided SRAM readily supports both. In both types, the divided SRAM consists of several smaller blocks, and only the blocks necessary to hold data (called “active blocks”) are turned on while the other blocks (called “non-active blocks”) are switched off. This strategy suppresses leak current in the unused cells and so enables large SRAM capacities. Note that the memory controller necessary for divided SRAM operation is activated only when the memory is accessed; therefore, the power overhead of the memory controller is quite small. Even in divided SRAM, however, a partially used memory block suffers some leak current wastage since not all memory cells are used. To mitigate this problem, smaller block size is preferred. Divided SRAM has the ability to lower the power consumption more than the use of high-compression rate data aggregation schemes such as perfect aggregation, because the number of non-active blocks increases with the compression rate.

3.1 Equal-Size Divided SRAM

Figure 5 illustrates equal-size divided SRAM, which consists of blocks of equal size. In the case where S -bit equal-size divided SRAM consists of N blocks, the size of each block is S/N bits.

Recall that to reduce the leak current wastage, we should reduce the block size. However, the number of power lines doubles when the block size is halved for the same total memory size, which complicates the circuit design of the memory and memory controller. On the other hand, since each block has the same size, implementing the memory blocks is simple.

3.2 Equal-Ratio Divided SRAM

To overcome the tradeoff inherent in equal-size divided SRAM, we propose equal-ratio divided SRAM which has blocks of various sizes from small to large. Conceptually, an S -bit equal-ratio divided SRAM is constructed by applying the following operation n times to an S -bit SRAM: one of the smallest blocks is divided into m equal-size blocks. Here, m is the splitting ratio, and the typical value of m is

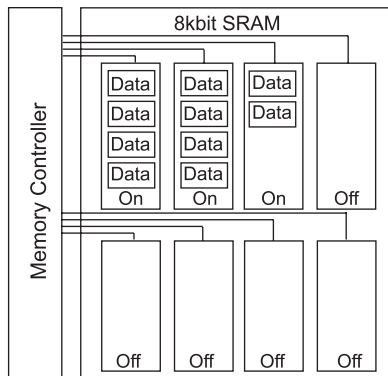


Fig. 5 Divided SRAM with same block size (equal-size divided SRAM).

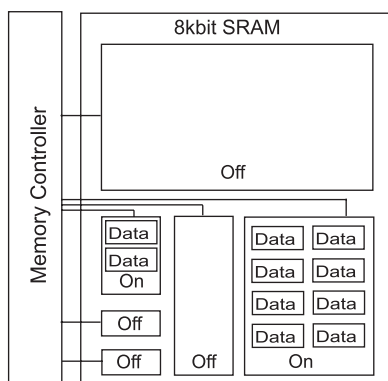


Fig. 6 Divided SRAM with stepped block size (equal-ratio divided SRAM (splitting ratio $m = 2$, the splitting number $n = 5$, the number of blocks $N = 6$)).

two. The logical result is, there are $m - 1$ blocks of each size except the smallest size for which there are m blocks. As a result, the total number N of the blocks is given by $N = (m - 1)(n - 1) + m$, so that $n = (N - 1)/(m - 1)$. See Fig. 6 for the case of $m = 2$, $n = 5$, and $N = 6$. For convenience, let S_k denote the k th smallest block-size; it is calculated as

$$S_1 = \frac{S}{m^n}, \text{ and } S_k = S_1 m^{k-1} \text{ for } k > 1. \quad (1)$$

Thus, an S -bit equal-ratio divided SRAM is designed to have $(m - 1)$ blocks of S_k -bit size for $k > 1$, and m blocks of S_1 -bit size. Using these notations, the total size S of the SRAM can be expressed as

$$\begin{aligned} S &= \sum_{k=1}^n (m - 1) S_k + S_1, \\ &= \left\{ \sum_{k=1}^n (m - 1) m^{k-1} + 1 \right\} S_1. \end{aligned} \quad (2)$$

As mentioned later, the MCU can manage the blocks so as to reduce the unused capacity in the active blocks to less than the size of the smallest block. The S -bit equal-size divided SRAM and the S -bit equal-ratio divided SRAM have at least S/N -bit blocks and S/m^n -bit blocks, respectively, if each SRAM consist of N blocks. Thus, the equal-ratio divided SRAM can has $N/m^{(N-1)/(m-1)}$ times larger block size

than the equal-size divided SRAM.

In equal-ratio divided SRAM, the number of power lines is increased by just one when one block is halved. Furthermore, when the splitting ratio is a multiple of two, the start address of each block is easily determined. That is, equal-ratio divided SRAM offers simpler memory controllers than equal-size divided SRAM, but makes it more complex to implement the memory blocks.

3.3 Data Store Scheme

In what follows, for convenience, we number all N blocks from 0 to $N - 1$, in ascending order of block-size. As a result, the blocks of the smallest size are labeled from block-number 0 to $m - 1$, and for $k > 1$, those of the k th smallest size are labeled from block-number $(k - 1)(m - 1) + 1$ to $k(m - 1)$.

The basic strategy to leverage divided SRAM is to minimize the total size of active blocks to store data, which implicitly minimize the unused capacity in the active blocks. We call such a situation the “ideal state.” In order to achieve the ideal state, when data is newly stored in or removed from divided-SRAM, data are transferred among blocks if necessary.

We assume that the MCU holds the amount of stored data, so it can determine how many (and of which size, in the case of equal-ratio divided SRAM) blocks should be turned on to realize the ideal state. Suppose that the total size of data to be stored is D bits. In the case of equal-size divided SRAM, $\lceil D/(S/N) \rceil$ blocks should be active. On the other hand, in the case of equal-ratio divided SRAM, how many and of which size blocks should be active is determined based on the quotient of D and S_1 and whether or not some remainder exists. D divided by S_1 can be expressed as

$$\frac{D}{S_1} = \sum_{k=1}^n s_k m^{k-1} + r, \quad (3)$$

where s_k and r are a non-negative integer less than m and a non-negative real number less than or equal to 1, respectively. Thus the quotient of D and S_1 is represented as “ $s_n s_{n-1} \dots s_1$ ” in the positional system with base m . The following is directly derived from (1) and (3):

$$D = \sum_{k=1}^n s_k S_k + r S_1. \quad (4)$$

Therefore, in order to store data of D bits, it is necessary that only s_k blocks of S_k -bit size, say, the block-number $k(m - 1)$ through $k(m - 1) - s_k - 1$, are active for $k = 1, \dots, n$ and the block with block-number 0 is additionally active if some remainder exists, otherwise it is inactive. In such a case, it is recognized that the unused space of the active blocks is $(1 - r)S_1$ bits. As an example, if the quotient is represented as “1021” in the positional number with base 3 and some remainder exists, one S_4 -bit block, zero S_3 blocks, two S_2 -bit blocks and two S_1 blocks need to be turned on.

3.3.1 Case of Adding Data

First, we consider the case where data are newly stored in a divided-SRAM.

In the case of equal-size divided SRAM, the MCU tries to append newly sensed data or a received packet to unused space, if any, of the active blocks, e.g., in ascending order of block-number. Each time the current active blocks become full, a non-active block is switched on to store the remaining data. The non-active block turned on is determined as the one with the smallest block-number. Note that no data are transferred among blocks when appended. This data storage scheme is simpler than that of the equal-ratio divided SRAM.

Equal-ratio divided SRAM operates like equal-size divided SRAM, but data can be transferred among blocks if necessary. Like equal-size divided SRAM, the MCU tries to append new data to unused space, if any, in descending order of block-number. This is because an active block with larger block-number is expected to remain active in the immediate ideal state. Then, every time the current active blocks become full, the block with the smallest block-number among the non-active blocks is additionally switched on to store the remaining data. This is because the block is expected to be active in the immediate ideal state. The state yielded by the above procedure is usually temporary, so data are transferred among the blocks to realize the ideal state if necessary. The exception is the case in which data is removed from the SRAM, e.g., a newly received packet invokes the immediate transmission of an aggregated packet. Data transfer among blocks is performed realize the ideal state as follows: First which blocks should be active, for the ideal state, is determined based on the amount of currently stored data. All data stored in the blocks that are to be non-active in the ideal state are moved to the unused space of the active blocks, e.g., from large to lower block-number.

3.3.2 Case of Removing Data

Next, we consider the case where data are removed from divided-SRAM.

In the case of equal-size divided SRAM, if the number of the blocks storing data is the same as the number of active blocks in the ideal state, i.e., $\lceil D/(S/N) \rceil$, no data is transferred among blocks. Otherwise, the following operation is repeated until the number of the blocks storing data decreases to the number of active blocks in the ideal state: The data in the block storing the least data is moved to the block with the smallest but sufficient unused space. An example of the above operation is illustrated in Figs. 7(a) and (b).

In the case of equal-ratio divided SRAM, all the data left in the Blocks that are to become inactive are moved to the active blocks in the ideal state. (Recall that how to determine which block is active is described in the beginning of Sect. 3.3.) Although there is no set policy, as one example,

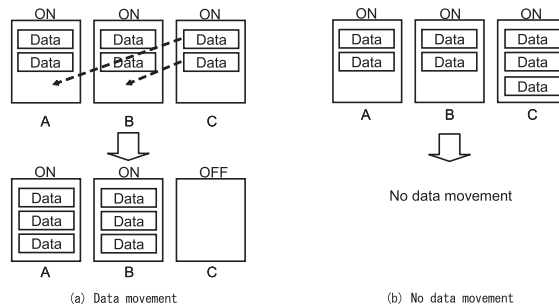


Fig. 7 Example of data store scheme in equal-size divided SRAM.

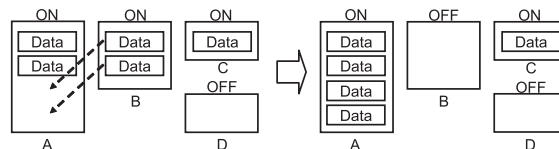


Fig. 8 Example of data store scheme in equal-ratio divided SRAM.

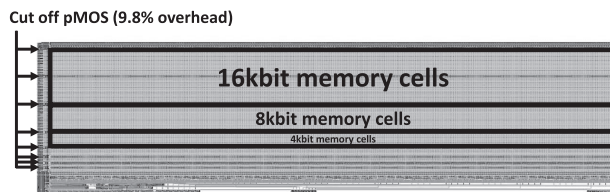


Fig. 9 Layout plot image of 4 kbytes equal-ratio divided SRAM.

the data left in a non-active block with larger block-number are moved to an active block with larger block-number. An example of the above operation is illustrated in Fig. 8.

3.4 Estimation of Overhead

To implement divided SRAM, several registers must be added to memorize the states of blocks. The area overhead and power overhead for the registers are, however, negligible because the number of the registers is much smaller than that of SRAM cells. Other than the registers, one pMOS switch is needed for each block to power on and off the block. Power supply management using the p-MOS switch is reported in [7], [8].

To evaluate the overheads of proposal scheme, we implemented the divided SRAM in a 180-nm CMOS process. Figure 9 shows the layout plot image of the 4 kbytes equal-ratio divided SRAM. As a result, the divided SRAM has 9.8% area overhead for the p-MOS switch compared to the 4 kbytes memory cells (Fig. 10.). In terms of the overall sensor node, the area overhead is not likely to exceed 1%.

Next, we made a simulation of the power-on sequence for the 8 kbits memory cells, which is the maximum size of the power domain in our implementation (Fig. 9). In the simulation, we use the H-SPICE circuit simulator under the condition shown in Table 2. Figure 11 shows the simulation results. As shown in Fig. 11(b), the required time of the VDD_MC increase to 99% is 16 ns. Furthermore, from

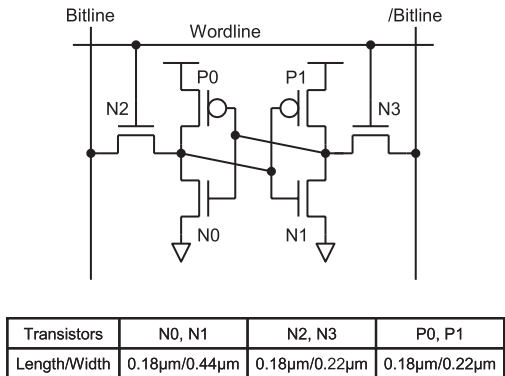


Fig. 10 Memory cell.

Table 2 Simulation conditions to estimate power.

Description	Type/Value
Technology	CMOS 180-nm process
Supply voltage	1.8 V
Process corner	Typical

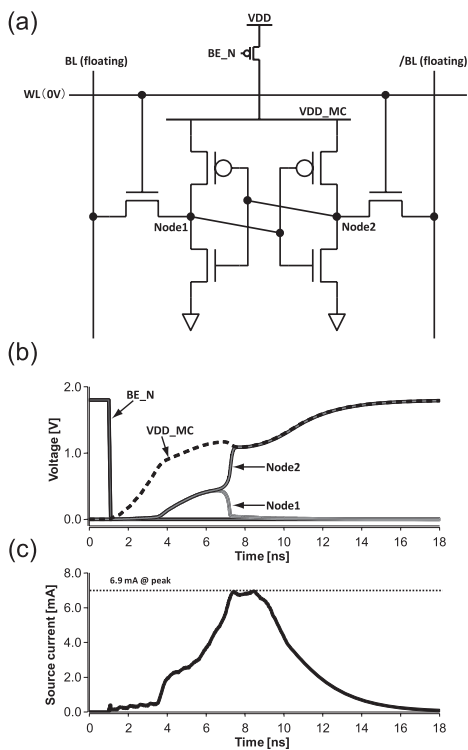


Fig. 11 Power-on sequence simulation: (a) simulation setup, (b) waveform, and (c) source current.

Fig. 11(c), the peak current is 6.9 mA, and the energy consumption is 70.95 pJ.

It is observed that a sensor node relays at most 50 packets in each data gathering in our simulation, see Sect. 4. Even when 16 kbits memory cells block is switched on every time a packet is relayed, this energy consumption is 7.1 nJ. In our simulation shown in Sect. 4, the whole block of the normal SRAM is assumed to be turned off when there is

Table 3 Simulation conditions.

Parameter	Value
Transmission range	20 m
Header size	4 bytes
Data size	8 bytes
ACK size	4 bytes
RTS/CTS size	4 bytes
Clock power	0.5 μW
MCU power	0.5 mW
Memory controller power	2 μW
Transmission power	8.5 mW
Reception power	3.6 mW
Leak power	1.5 nW/bit
Write power	1.3 mW
Read power	1.3 mW

no data to be stored. Therefore, the normal SRAM consumes energy to switch the block and one p-MOS switch is needed. (Note that this assumption is favorable to the normal SRAM.) If power supply management is not used, the normal SRAM always consumes leak power. For example, when 4 kbytes SRAM is turned on for 600 s and leak power is 1.5 nW/bit as shown in Table 3, leak energy consumption in the normal SRAM is 29.5 mJ (= 1.5 nW/bit × 600 s × 4,096 bytes × 8 bits/byte). Consequently, the power overhead of the p-MOS switch due to power supply management is negligible.

4. Performance Evaluation

In this section, we used the QualNet simulator to evaluate our proposed schemes [9].

4.1 Simulation Conditions

The following were assumed in this simulation. The simulation area is set to 100 m × 100 m. Sensor nodes are uniformly deployed in the sensing area. The base station is at the center of the sensing area. The simulation time is the time needed to collect the sensing data at the base station. Sensor nodes collect data for the base station in accordance with Tiny Diffusion, which is a simplified Directed Diffusion method [10], [11]. We used I-MAC as the MAC protocol [12]. I-MAC is equipped with ARQ (Automatic Repeat reQuest), where the retry limit is set to 10. MCU uses a register but no dedicated SRAM. Other parameters are shown in Table 3. The power values of RF, RTC, and MCU are taken from [13], [14], and [15], respectively. Data packet size is 8 bytes unless otherwise noted, and the header size is 4 bytes including one byte trailer. The maximum number of aggregated packet is five and the number of memory block is eight unless otherwise noted. (See Appendix.) The power of MCU is assumed that MCU always operates at 4 MHz. Finally, the memory power is estimated with an H-SPICE circuit simulator [16]. Figure 10 illustrates a memory cell and the transistor parameters used with H-SPICE. The power-supply voltage is 1.8 V, and the threshold voltage is 0.45 V.

The operation frequency of the memory is assumed to be 4 MHz, and memory capacity is assumed to be 4 kbytes (it is determined by reference to [17] and can be applied for sound sensing applications.); 8 bit writing takes one cycle and 8 bit reading takes two cycles; The leak current of memory is assumed to be linear against memory capacity; We use the values of read and write power estimated for the case of 4 kbyte memory capacity, in all simulations. Although they actually depend on memory capacity, the read and write power is, however, negligible since leak current is dominant. For example, if the memory reads 4 kbytes data, read operation time is about 2 ms, and the energy consumption of memory read is $2.6 \mu\text{J}$. On the other hand, if the memory hold 4 kbytes data for 100 s, the leak power is 4.9 mJ. For equal-ratio Divided SRAM, the splitting ratio m is 2 and the splitting time n is 7. Thirty simulation trials with different random seeds were carried out for each set of simulation parameters, and the simulation result was taken as their average.

4.2 Simulation Results

Figure 12 shows the energy consumption of MCU, RF (RX and TX), clock, and memory using normal SRAM, equal-size divided SRAM, and equal-ratio divided SRAM when the SRAM capacity is changed. Note that the minimum block size is 0.5 bit in the case of equal-ratio divided SRAM with 8 byte RAM capacity. This value is not unrealistic, but it is used as it is in order to investigate the tendency of energy consumption. Memory energy consumption of equal-size divided is smaller than that of normal SRAM if the SRAM exceeds 128 bytes; most of the power is consumed by memory maintenance. Therefore, the lower power consumption of this memory is due to the reduction in leak current offered by the divided SRAM. For example, the energy consumption of equal-size divided SRAM is 20% that of the conventional scheme under the condition that SRAM totals 1,024 bytes. In the case of equal-ratio divided SRAM, power consumption is almost the same from 64 bytes to 4,096 bytes. This is because, for the same number of blocks, equal-ratio divided SRAM provides a wide variety of block

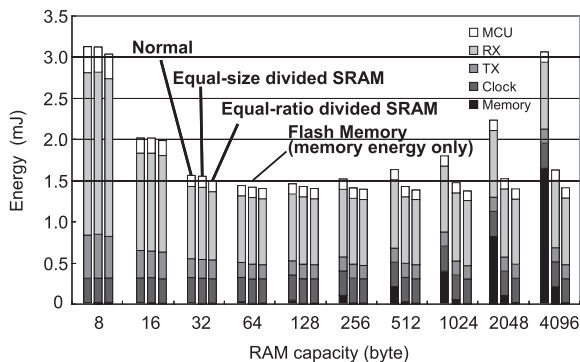


Fig. 12 Energy consumption of MCU, RF (RX and TX), clock, and memory using equal-ratio divided SRAM (number of memory blocks = 8, maximum number of aggregated packets = 5).

sizes.

The power consumption of equal-ratio divided SRAM is less than 1% that of the sensor node. On the other hand, flash memory has no leak current but the power consumed by deletion and writing is large. With regard to the parameters of the flash memory of MICA2, the operation voltage is 3.6 V, the current consumption of deletion and writing is 12 mA, and the deletion and writing times are 14 ms [18]. Data is written an average of 2.5 times to realize one data collection in this simulation. Flash memory, therefore, consumes 1.5 mJ to acquire one data point. In contrast, equal-ratio divided SRAM offers lower power consumption than flash memory (see energy consumption of memory in Fig. 12).

Figure 13 shows the standby energy of SRAM using normal SRAM, equal-size divided SRAM, and equal-ratio divided SRAM. In equal-size divided SRAM, when the memory capacity becomes large, power supply management worsens, yielding the excessive power consumption noted. In equal-ratio divided SRAM, conversely, leak current is small even at large memory capacities, because the power supply management efficiency is high. When the memory capacity is 1,024 bytes, the energy consumption of equal-size divided SRAM is 13% that of normal SRAM and the energy consumption of equal-ratio divided SRAM is 1.3% that of normal SRAM. In other words, that equal-size and equal-ratio divided SRAM offer power consumption reductions of 87% and 98.7%, respectively, from normal SRAM.

Figure 14 shows the data collection time at various SRAM capacities when the maximum number of aggregated packets is five. Note that the results are same regardless to the SRAM structure. As the data size increases, small-capacity SRAM becomes saturated by relay packets. As a result, the transmission node may not be able to transmit packets. This is because the data collection time is large when RAM capacity is small. The size of the payload depends on the sensor and application. To expand the versatility of sensor nodes from delay viewpoints, the large-capacity SRAM is preferable.

Figure 15 shows the total energy consumed when us-

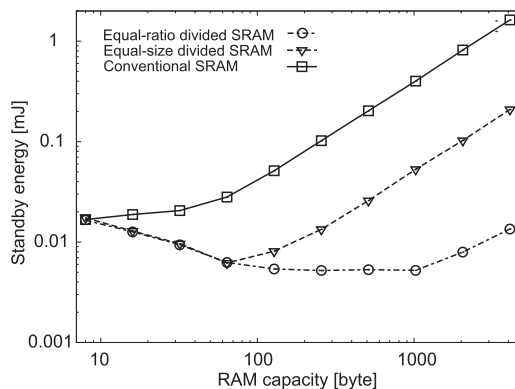


Fig. 13 Standby energy consumption of SRAM using equal-size divided SRAM and equal-ratio divided SRAM (number of memory blocks = 8, maximum number of aggregated packets = 5).

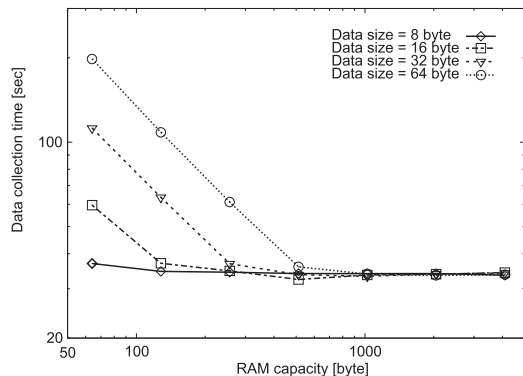


Fig. 14 Data collection time when RAM capacity is changed (number of memory blocks = 8, maximum number of aggregated packets = 5).

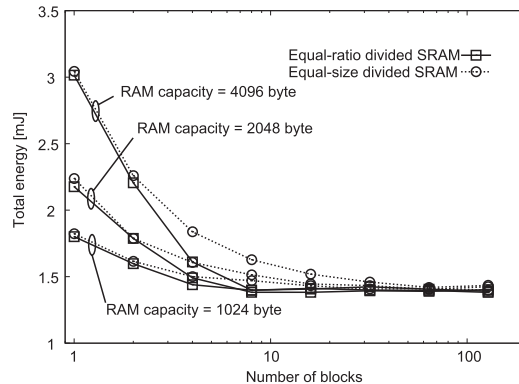


Fig. 16 Total energy consumption of sensor nodes when the number of memory blocks is changed (maximum number of aggregated packets = 5, data size = 8 byte).

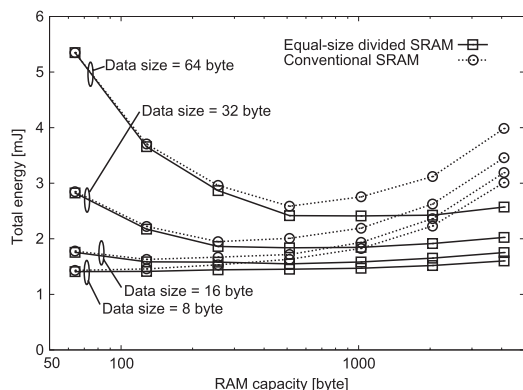


Fig. 15 Total energy using normal SRAM and equal-size divided SRAM when the payload size is changed (number of memory blocks = 8, maximum number of aggregated packets = 5).

ing normal SRAM and equal-size divided SRAM at different data sizes. For example, optimum capacity is 128 bytes when the data size is 16 bytes in normal SRAM. However, if data size changes to 32 bytes, delay and energy consumption become large. On the other hand, large capacity divided SRAMs can reduce the energy consumption and delay, even if the data size changes. Therefore, it can be concluded that divided SRAMs are more effective than conventional SRAM.

Finally, we estimate the number of memory blocks of equal-size divided SRAM with three capacities (1,024, 2,048, 4,096 bytes) Note that conventional SRAM is the case where the number of block is one. From Fig. 16, it is shown that both the divided-SRAMs are effective enough compared to conventional SRAM. Next, let us compare the both divided-SRAMs. In equal-ratio divided SRAM, when the number of memory blocks exceeds eight, the effectiveness of power reduction saturates at all three memory capacities. On the other hand, In equal-size divided SRAM, the total energy is still reducing as the number of blocks exceeds eight. In case of several tens blocks, the difference between both the divided-SRAMs is negligible since the block-size is enough small. In case of four through 16 blocks, there seems to be the difference between both the

divided-SRAM. In case of smaller RAM capacity, say 1,024 bytes, the difference is slight. The difference, however, increases with the RAM capacity. As mentioned above, to expand the versatility of sensor nodes from delay viewpoints, large-capacity SRAM is preferable. On the other hand, considering manufacturing cost, complexity of design, and management of memory blocks, it is preferable that memory blocks are fewer. In Fig. 16, eight-block equal-ratio divided SRAM has almost the same total energy as 64-block equal-size SRAM. In this sense, equal-ratio SRAM is more effective than equal-size SRAM.

5. Conclusion

The weakness of volatile RAM is that it consumes power just to hold data. In order to realize low-cost and long-life one-chip sensor nodes, we proposed two SRAM architectures for packet buffers that perform data aggregation: equal-size and equal-ratio divided SRAM. Simulations results showed that equal-size and equal-ratio divided SRAM offer power consumption reductions of 87% and 98.7%, respectively, from normal SRAM when the memory capacity is 1,024 bytes. Eight-block equal-ratio SRAM has 10% area overhead (only SRAM area) compared to conventional SRAM in the case where RAM capacity is 8 kbytes. Therefore both the divided SRAM are energy-effective enough compared to normal SRAM at the cost of small area overhead. In the future, there will be the need to develop sensor networks to handle large data like image. For this sake, equal-ratio divided-SRAM is especially effective since it can reduce power consumption with fewer blocks compared to equal-size divided-SRAM.

Acknowledgment

This research work was partially supported by the Strategic Information and Communications R&D Promotion Program (SCOPE), Ministry of Internal Affairs and Communications, Japan, by a Ministry of Education, Culture, Sports, Science and Technology (MEXT) Grant-in-Aid for Scien-

tific Research (C), No. 21500074.

References

- [1] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," IEEE INFOCOM, June 2002.
- [2] T.F. Abdelzaher, T. He, and J.A. Stankovic, "Feedback control of data aggregation in sensor networks," IEEE Conference on Decision and Control, Dec. 2004.
- [3] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of density on data aggregation in wireless sensor networks," Proc. 22nd International Conference on Distributed Computing Systems, Nov. 2001.
- [4] A. Wang, W.B. Heinzelman, A. Sinha, and A.P. Chandrakasan, "Energy-scalable protocols for battery-operated MicroSensor networks," Kluwer J. VLSI Signal Processing, vol.29, pp.223-237, Nov. 2001.
- [5] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," Proc. IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.
- [6] D. Petrovic, C. Shah, K. Ramchandran, and J. Rabaey, "Data funneling: Routing with aggregation and compression for wireless sensor networks," Proc. IEEE Sensor Network Protocols Applications, Anchorage, May 2003.
- [7] F. Hamzaoglu, K. Zhang, Yih Wang, H.J. Ahn, U. Bhattacharya, Z. Chen, Y.-G. Ng, A. Pavlov, K. Smits, and M. Bohr, "A 153 Mb-SRAM design with dynamic stability enhancement and leakage reduction in 45 nm high-K metal-gate CMOS technology," IEEE J. Solid-State Circuits, vol.44, no.1, pp.148-154, Jan. 2009.
- [8] G. Gerosa, S. Curtis, M. D'Addeo, Bo Jiang, B. Kuttanna, F. Merchant, B. Patel, M.H. Taufique, and H. Samarchi, "A sub-2W low power IA processor for mobile Internet devices in 45 nm high-k metal gate CMOS," IEEE J. Solid-State Circuits, vol.44, no.1, pp.73-82, Jan. 2009.
- [9] QualNet simulator, <http://www.scalable-networks.com>
- [10] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," Proc. IEEE/ACM Transaction on Networking, vol.11, pp.2-16, Feb. 2003.
- [11] J. Heidemann, F. Silva, and D. Estrin, "Matching data dissemination algorithms to application requirements," Proc. ACM SenSys Conference, pp.218-229, Nov. 2003.
- [12] M. Ichien, T. Takeuchi, S. Mikami, H. Kawaguchi, C. Ohta, and M. Yoshimoto, "Isochronous MAC using long-wave standard time code for wireless sensor networks," Proc. International Conference on Communications and Electronics (ICCE), pp.172-177, Oct. 2006.
- [13] B.P. Otis, Y.H. Chee, R. Lu, N.M. Pletcher, and J.M. Rabaey, "An ultra-low power MEMS-based two-channel transceiver for wireless sensor networks," Proc. Symposium on VLSI Circuits, pp.20-23, June 2004.
- [14] Real Time Clock IC, <http://www.oki.com/en/press/2007/z07003e.html>
- [15] M. Sheets, F. Burghardt, T. Karalar, J. Ammer, Y. Chee, and J. Rabaey, "A power-managed protocol processor for wireless sensor networks," Proc. Symposium on VLSI Circuits, pp.212-213, Sept. 2006.
- [16] Hspice, synopsys, <http://www.synopsys.com/products/mixedsignal/hspice/hspice.html>
- [17] N. Yamauchi, I. Urushibara, A. Aizawa, H. Sato, H. Hosaka, K. Sasaki and K. Itao, "Nature interfacer version 3 (Ni3): A wearable wireless sensor module with flexible protocol configurability for ubiquitous sensor networks," Proc. 1st International Workshop on Networked Sensing Systems (INSS), 2004.
- [18] MICA2 Mote, xrossbow, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-04_A_MICA2.pdf

Appendix: Consideration of Number of Aggregation Packets and Partitions

Based on the following consideration, we set the maximum number of aggregation packets to five and the number of blocks of divided SRAM to eight.

Figures A·1 and A·2 show the energy consumption of the sensor node when the maximum number of aggregated packets is changed. It can be understood from these figures that total energy consumption can be reduced significantly if five or more data packets are aggregated regardless of RAM capacity. This is because the energy to transmit the packet header is inversely proportional to the number of aggregated packets in the case of header aggregation. Therefore we set the maximum number of aggregated packet to five in performance evaluation.

Next, we investigated the impact of equal-size divided SRAM and equal-ratio divided SRAM on energy consumption as a function of payload size. (See Figs. A·3 and A·4.) In order to simplify the logic of the memory controller and memory itself, the number of blocks should be a power of

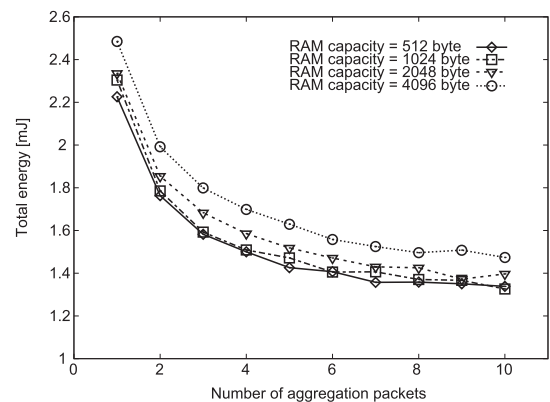


Fig. A·1 Total energy consumption of sensor nodes when equal-size divided SRAM is used and the maximum number of aggregated packets is changed. (number of memory blocks = 8, data size = 8 bytes).

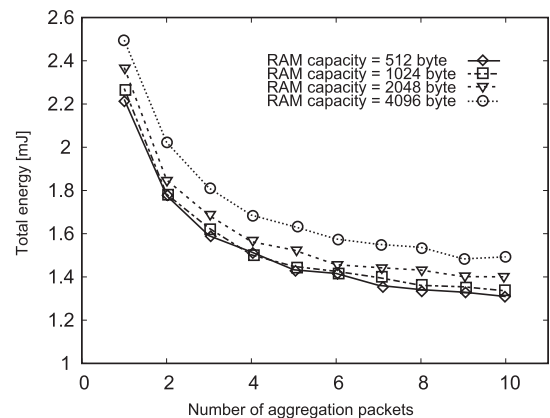


Fig. A·2 Total energy consumption of sensor nodes when equal-ratio divided SRAM is used and the maximum number of aggregated packets is changed. (number of memory blocks = 8, data size = 8 bytes).

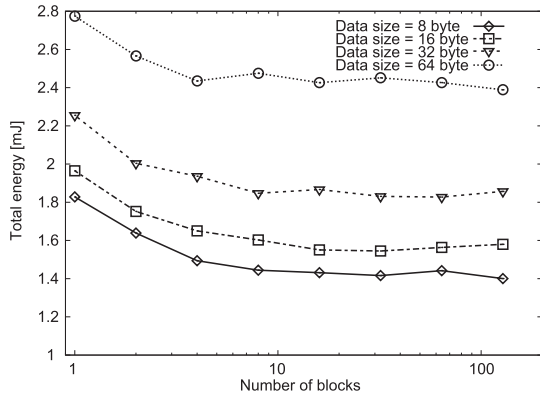


Fig. A-3 Total energy consumption when equal-size divided SRAM is used and the payload size is changed. (maximum number of aggregated packets = 5, SRAM capacity = 1,024 bytes).

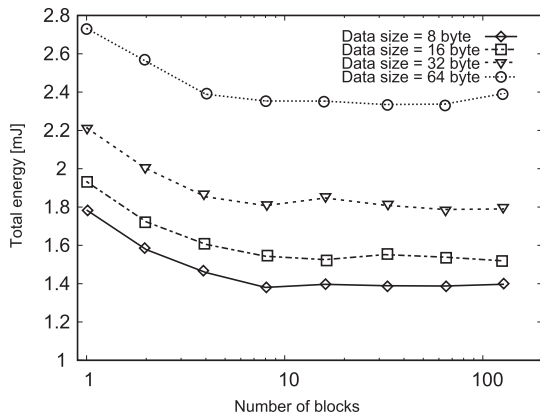
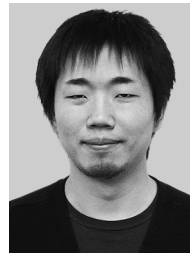


Fig. A-4 Total energy consumption of sensor nodes when equal-ratio divided SRAM is used and the maximum number of aggregated packets is changed. (number of memory blocks = 8, data size = 8 bytes).

two. When the number of blocks is eight or more, the effectiveness in terms of power consumption reduction is basically saturated regardless of packet payload size. Increasing the number of blocks increases the number of power lines. Therefore we set the number of blocks to eight in performance evaluation.



Shintaro Izumi respectively received his B.E. and M.E. degrees in Computer Science and Systems Engineering from Kobe University, Kobe, Japan, in 2007 and 2008. Currently, he is a Ph.D. course student and a JSPS research fellow at Kobe University. His current research interests include communication protocols, low-power VLSI design, and wireless sensor networks. He is a student member of the IEEE.



Yasuharu Sakai received his B.E. degree in Computer and Systems Engineering and his M.E. degree in Science and Technology both from Kobe University, Kobe, Japan in 2008 and 2010, respectively. His interests include media access controls for sensor networks.



Takashi Takeuchi received his B.E. degree in Electrical and Electronic Engineering from Kanazawa University in 2005, his M.E. degree from Kobe University in 2007. He received his D.Eng. degree in Science and Technology from Kobe University, Kobe, Japan in 2010. His interests include low-power analog circuit designs. He is a student member of IEEE.



Hidehiro Fujiwara respectively received B.E. and M.E. degrees in Computer and Systems Engineering in 2005 and 2006 from Kobe University, Hyogo, Japan. He received his D.Eng. degree in Science and Technology from Kobe University, Kobe, Japan in 2009. His current research is related to high-performance and low-power SRAM designs. He is a student member of IEEE.



Takashi Matsuda received his B.E. degree in Computer Science and Systems Engineering and his M.E. degree in Science and Technology both from Kobe University, Kobe, Japan in 2005 and 2006, respectively. He received his D.Eng. degree in Science and Technology from Kobe University, Kobe, Japan in 2010. He has been an Expert Researcher with the Medical ICT Group at the National Institute of Information and Communications Technology. His current research interests include sensor network,

two-dimensional network and contact-less power supply.



Hiroshi Kawaguchi received his B.E. and M.E. degrees in Electronic Engineering from Chiba University in Chiba, Japan, in 1991 and 1993, respectively, and his Doctorate in Engineering from the University of Tokyo, Tokyo, Japan, in 2006. In 1993 he joined Konami Corporation in Kobe, Japan, where he worked on the development of arcade entertainment systems. He joined the Institute of Industrial Science at the University of Tokyo as a Technical Associate in 1996, and was appointed a Research Associate there in 2003.

In 2005, he moved to the Department of Computer and Systems Engineering at Kobe University, where he also worked as a Research Associate. Since 2007, he has been working as an Associate Professor with the Department of Computer Science and Systems Engineering at Kobe University. He is also a Collaborative Researcher with the Institute of Industrial Science at the University of Tokyo. His current research interests include low-power VLSI designs, hardware designs for wireless sensor networks, and recognition processors. Dr. Kawaguchi was a recipient of the IEEE ISSCC 2004 Takuo Sugano Outstanding Paper Award and the IEEE Kansai Section 2006 Gold Award. He has served as a Program Committee Member for the IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips), and as a Guest Associate Editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. He is a member of both the IEEE and ACM.



Chikara Ohta was born in Osaka, Japan, on July 25, 1967. He received his B.E., M.E. and Ph.D. (Engineering) degrees in Communication Engineering from Osaka University, Osaka, Japan, in 1990, 1992 and 1995, respectively. From April 1995, he worked as an Assistant Professor at the Department of Computer Science at the Faculty of Engineering of Gunma University in Gunma, Japan. In October 1996, he joined the Department of Information Science and Intelligent Systems at the Faculty of

Engineering, University of Tokushima, Tokushima, Japan, as a Lecturer. He became an Associate Professor at the same facility in March 2001. Since November 2002, he has worked as an Associate Professor of the Department of Computer and Systems Engineering in the Faculty of Engineering at Kobe University, Kobe, Japan. From March 2003 to February 2004, he was a visiting scholar at the University of Massachusetts at Amherst, USA. His current research interests include performance evaluation of communication networks. He is a member of IPSJ, IEEE and SIGCOMM.



Masahiko Yoshimoto received his B.S. degree in Electronic Engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1975, his M.S. degree in Electronic Engineering from Nagoya University, Nagoya, Japan, in 1977 and his Ph.D. degree in Electrical Engineering, also from Nagoya University, in 1998. He joined the LSI Laboratory of Mitsubishi Electric Corp. in Itami, Japan, in April 1977. From 1978 to 1983 he engaged in the design of NMOS and CMOS static RAM, including a 64 K full CMOS RAM,

with the world's first divided word-line structure. From 1984, he engaged in the research and development of multimedia ULSI systems for digital broadcasting and digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology. In 2000, he became a Professor of the Dept. of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. In 2004, he moved to Kobe City where he became a Professor of the Dept. of Computer and Systems Engineering at Kobe University. His current activities are focused on the research and development of multimedia and ubiquitous media VLSI systems, including an ultra-low-power image compression processor and a low power wireless interface circuit. He holds 70 registered patents. He served on the Program Committee of the IEEE International Solid State Circuit Conference from 1991 to 1993. In addition, he has worked as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received two R&D100 awards from R&D Magazine for the development of the DISP and for the development of a real-time MPEG2 video encoder chipset in 1990 and 1996, respectively.