SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2016, Vol. 59 042401:1–042401:11
doi: 10.1007/s11432-015-5371-1

Value locality based storage compression memory architecture for ECG sensor node

Chaojun ZHAO[1], Chen CHEN[2], Zhijian CHEN[1]* & Jianyi MENG[2]

[1]*Institute of VLSI Design, Zhejiang University, Hangzhou* 310027*, China;*
[2]*State Key Laboratory of ASIC & System, Fudan University, Shanghai* 201203*, China*

Received December 24, 2014; accepted April 29, 2015; published online March 1, 2016

Abstract This paper proposes a value compression memory architecture for QRS detection in ultra-low-power ECG sensor nodes. Based on the exploration of value spatial locality in the most critical preprocessing stage of the ECG algorithm, a cost efficient compression strategy, which reorganizes several adjacent sample values into a base value with several displacements, is proposed. The displacements will be half or quarter scale quantifications; as a result, the storage size is reduced. The memory architecture saves memory space by storing compressed data with value spatial locality into a compressed memory section and by using a small, uncompressed memory section as backup to store the uncompressed data when a value spatial locality miss occurs. Furthermore, a low-power accession strategy is proposed to achieve low-power accession. An embodiment of the proposed memory architecture has been evaluated using the MIT/BIH database, the proposed memory architecture and a low-power accession strategy to achieve memory space savings of 32.5% and to achieve a 68.1% power reduction with a negligible performance reduction of 0.2%.

Keywords ECG R peak detection, wavelet transform, memory compression, low power, memory architecture

Citation Zhao C J, Chen C, Chen Z J, et al. Value locality based storage compression memory architecture for ECG sensor node. Sci China Inf Sci, 2016, 59(4): 042401, doi: 10.1007/s11432-015-5371-1

1 Introduction

Wearable electrocardiogram (ECG) devices can deliver better healthcare at low medical resource costs through a wireless body sensor network (WBSN). The monitoring system continuously records the ECG signal and transmits the original signal or the alert signal to the remote healthcare center or to the users themselves. For such a wearable ECG device, the battery is the main challenge [1].

There is a clear trend that the ECG node must deliver more powerful computational ability [1]. There is much previous works targeting this topic [2–12]. ASIC design, such as [2–6], uses a function-specific ASIC to provide low power QRS detection and data compression, or a mixed-signal ASIC [7–9], which accomplishes power savings by simplifying the design of the digital processing unit. In addition, accelerator based design, such as [10–12], depends on the FFT, the FIR and various other accelerators to achieve power efficiency. However, both the ASIC design and the accelerator-based design require the co-design of the software and hardware, which makes the software highly dependent on the hardware platform.

* Corresponding author (email: chenzj@vlsi.zju.edu.cn)

Table 1 Selected algorithms for ECG system

Reference	Hardware platform	Function	Algorithm
[2], [5]	ASIC	QRS detection	DWT
[15]	MSP430	QRS detection	DWT
[6]	ASIC	QRS detection, compression	DWT
[17]	MSP430	QRS detection, compression	DWT
[3]	ASIC	QRS detection, classification	DWT
[8]	Mixed-signal ASIC	QRS detectoin	CWT
[10]	Accelerator based	QRS detectoin	CWT
[9]	Mixed-signal ASIC	QRS detection	Filter,derive
[11]	Accelerator based	QRS detection	Filter,derive
[16]	Silicon Hive PearlRay	QRS detection	Filter,derive
[7]	Mixed-signal ASIC	QRS detection	bandpower
[4]	ASIC	QRS detection,compression	QLV

The long design cycle and the platform dependency will prevent the development of ECG applications. Among the proposed solutions, the traditional general purpose processor, such as ATmega128L [13] or MSP430 [14], is still widely used in ECG systems [15–18]. Compared with the ASIC design, the general purpose processor faces serious problems for power efficiency, including power inefficient atomic execution units, massive memory loads and storage. However, its software independency ensures more power, more diversification and a more robust ECG platform. The ultra-low-power ECG processor is still the most attractive research area.

Several biomedical processors targeting low power have been proposed [19–22]; Ref. [19] used the sub-threshold design to reduce the circuit power consumption, Ref. [20] used the asynchronous circuit to avoid the clock power, and Refs. [21, 22] used carefully designed instructions sets to achieve efficient execution. As the memory power consumption takes a dominate part of the whole system [4], this paper focuses on an architectural improvement that is processor architecture independent and proposes a novel memory architecture based on the insight of the data value. These improvements can save memory space and deliver low accession power. This paper is organized in the following manner. First, the characteristic of the data value in memory is analyzed for basic ECG functions (the QRS detection algorithm), and the value spatial locality is explored in Section 2. Second, a compressed storage strategy with a compressed memory architecture is proposed in Section 3; it can dynamically compress the stored data and reduce memory space consumption. Furthermore, a load and store buffer-based low-power memory accessing strategy, which utilizes the accessing flow of the algorithm, is proposed in Section 3. Combined with the compressed store mechanism, the load and store buffer can deliver the data compression control together with the low-power accession. Finally, the results are evaluated in Section 4; compared with the original memory architecture, this memory architecture can save 32.5% memory space and achieve a 68.1% power reduction with a total performance reduction of only 0.2%.

2 Value locality in ECG application

There are mainly two aspects of detection algorithms: the time domain algorithm, which uses derivation calculation and filter bank [23], band power [24], or others [4] to extract and detect the QRS complex. And the transform domain algorithm, mainly the wavelet transform, including continuous wavelet transform (CWT) and discrete wavelet transform (DWT), which decompose the ECG signal into several frequency component, and separately detect the suspect QRS complex in each frequency component. The wavelet transform based QRS detection algorithms can deliver better detection accuracy [5]. Besides, we have investigated the recent published ECG system in Table 1, the wavelet transform based algorithm is the most popular algorithm for the ECG platform. In this paper, the DWT-based ECG QRS detection

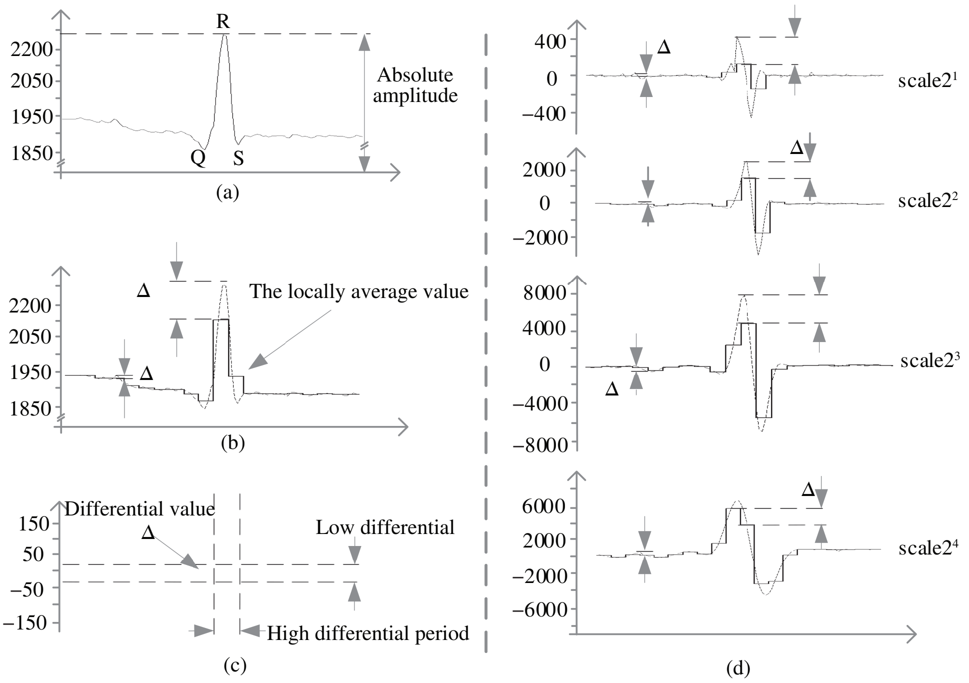


Figure 1 The locality feature of ECG signal. (a) ECG waveform; (b) locally average value; (c) the differential value; (d) the wavelet transform of ECG.

algorithm [25] and the MIT/BIH database [26] are selected as the basic specifications for the analysis and experiment.

The ECG QRS detection algorithm can be clearly divided into the preprocessing stage and the feature detection stage. The preprocessing stage of the QRS algorithm is responsible for transforming the original signal into the wavelet formulation and filtering out the inapplicable information. The feature detection stage will further check and confirm the QRS complex. The preprocessing stage is the most critical part of the algorithm because it deals with all of the original ECG signals: the experiment results show it consumes 80% of total dynamic instructions and 76% of the total memory accession operations of the algorithm. Furthermore, 90% of the total memory space is consumed by the data structure of the preprocessing stage (primarily the space for the wavelet transform). The preprocessing stage should be the key for the optimizing.

Figure 1(a) shows a typical ECG signal. Figure 1(b) shows the average value of 5 consecutive samples, named the local average value. The differential between the original value and the local average value is shown in Figure 1(c). There exists a slightly higher differential period in Figure 1(c), which corresponds to the sharpness QRS complex (within 80 ms [24]), but most of the differential values have a low amplitude. Furthermore, a similar feature can be found for the wavelet transform of the ECG in Figure 1(d). In a word, the value of the consecutive samples in the original ECG signal and the corresponding calculations (DWT) is local. Further, continuous memory space will be allocated for these data structures. Based on this insight, the data value for all of the data structures in the preprocessing stage are quantitatively investigated in Figure 2.

For the data structure in the preprocessing stage, at every 128 consecutive store operations on the given set of addresses, the value difference distribution is calculated for the average local value. Figure 2 shows, at a precision of 32 bits and for different address set sizes, 30% of the value difference from the local average is within 16 (4 bits width), and the value differences are primarily within 256 (approximately 80%). Especially, the value differences larger than 2048 (11 bits width) only take a part of 3%. The results strongly indicate that the data value at a practical set of address in the preprocessing stage is local; this is called the data value spatial locality.

The strong value spatial locality feature shows great opportunity to compress the total memory space

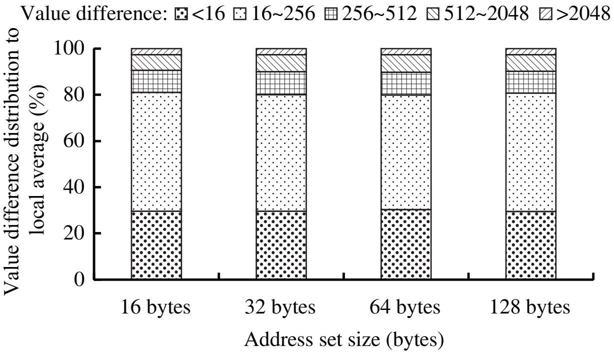


Figure 2 The local value difference distribution.

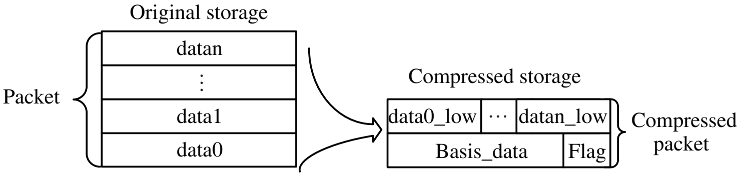


Figure 3 The compressed storage strategy.

for the ECG algorithm. In this paper, a novel memory architecture is proposed that can reduce memory space consumption and deliver low memory accessing power.

3 Value compression based memory architecture

3.1 Compressed storage strategy

For the memory space with value spatial locality, the storage can be compressed by only storing one basis datum for several consecutive addresses, and for each address only storing the displacement. The basis value in Section 2 is the average value, which is calculation inefficient. As a compromise, the high significant bits of the packet's data are selected as the basis data, the displacement will be the least significant bits for each address. The original data for each address can be attained from the basis data and the individual least significant field data with no calculation cost. Figure 3 describes the compressed storage strategy. The storage is divided into packets, which contain a set of addresses. In the compressed packet, the first word is used to store the basis data. The compression flag is also stored, which indicates whether the packet is compressed, and the following space is used to store the least significant bits for each address. The advantage of this method is no calculation cost for the compression and decompression operation.

3.2 Uncompressed backup section

The data value locality is a strong feature, but an exception may exist in that the fixed width basis data do not match the newly stored data values; this phenomenon is called a value locality miss. Additional memory space call uncompressed backup section is proposed to address this exception.

The uncompressed backup section is required to be dynamically allocated and de-allocated, as the value locality miss is dynamic. The link table mechanism is selected for managing the uncompressed backup memory section using a packet as the basic unit. The first word in the packet is used as the link pointer when the packet is not allocated. The architecture is shown in Figure 4, and an empty packet link head (EPLH) register is added to store the head of the link table. When resetting the system, the link table is created by storing the address of the next packet in the link pointer of the current packet.

When a value locality miss occurs, an uncompressed packet space can be simply allocated from the head of the link table, which is the EPLH register. Then, one additional memory load operation will

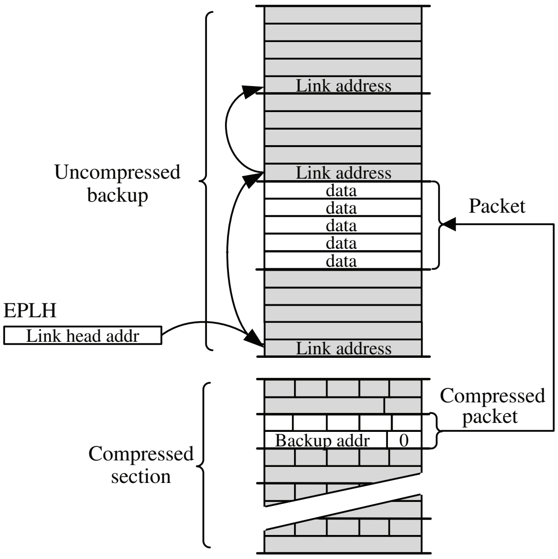


Figure 4 The uncompressed backup section handling mechanism.

be used to set the link pointer of the allocated packet as the new link head. Meanwhile, as shown in Figure 4, the basis data segment of the compressed packet will be changed with the address of the allocated uncompressed packet, and the compression flag will be set to zero. Any memory accession to this compressed packet will be directed to the corresponding uncompressed backup packet.

When de-allocating an uncompressed packet, the packet will be simply added to the link head by storing the address in the EPLH. An additional store operation will be used to store the old value of the EPLH register in the link pointer of the de-allocated packet.

3.3 Storage compression memory architecture

Figure 5 describes the storage compression memory architecture. The compressed storage strategy will be used to compress most of the space of the original memory, which is called compressed section. And for the stack spaces, there is an uncompressed section in the memory architecture. Most importantly, the uncompressed backup section is added to dynamically store the uncompressed packet when a value locality miss occurs in the compressed section, which is invisible to the software. The addresses of the two software visible memory sections are transferred from the software address via address management and can be simply calculated by simple addition or subtraction operations. Figure 6 shows the details of address management. Based on the size of the original compressed section (CSR in Figure 6), the address located in the compressed section will be calculated from the original address multiplied by the compression ratio, because the packet set size will be to the power of 2, it can be calculated by simple shift and add operations. And the address for the uncompressed section can be calculated by the original address less than the size of the total compressed section(CSR_CMP in Figure 6).

3.4 Low-power accessing strategy

The low-power accessing strategy is designed based on the processing flow of the ECG preprocessing stage. Figure 7(a) shows the software abstraction of the preprocessing stage of the DWT-based QRS detection. Through the loop flow, the data structure for the ECG and four wavelet decomposition scales are all continuously accessed. The low-power accession for the proposed memory is achieved by accessing the memory of a single packet, and the continuous loop flow will ensure that the consecutive address will be accessed in the near future. Figure 7(b) shows the idea of low-power accessing. For each new load operation, the whole packet will be loaded and decompressed for later access. For the store operation, until the processor has stored all addresses of the packet, the packet value will be compressed and stored in memory. Figure 7(c) shows the memory store power distribution of the proposed low-power accession

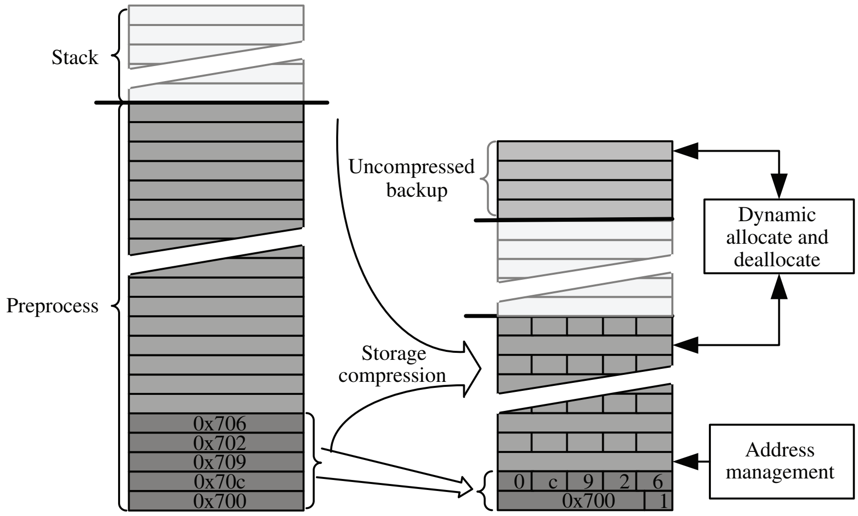


Figure 5 The storage compression memory architecture.

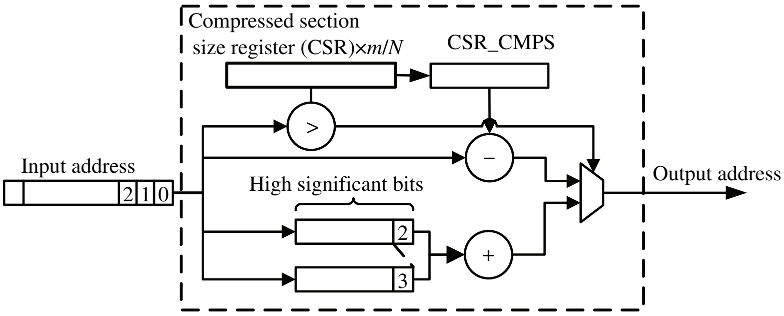
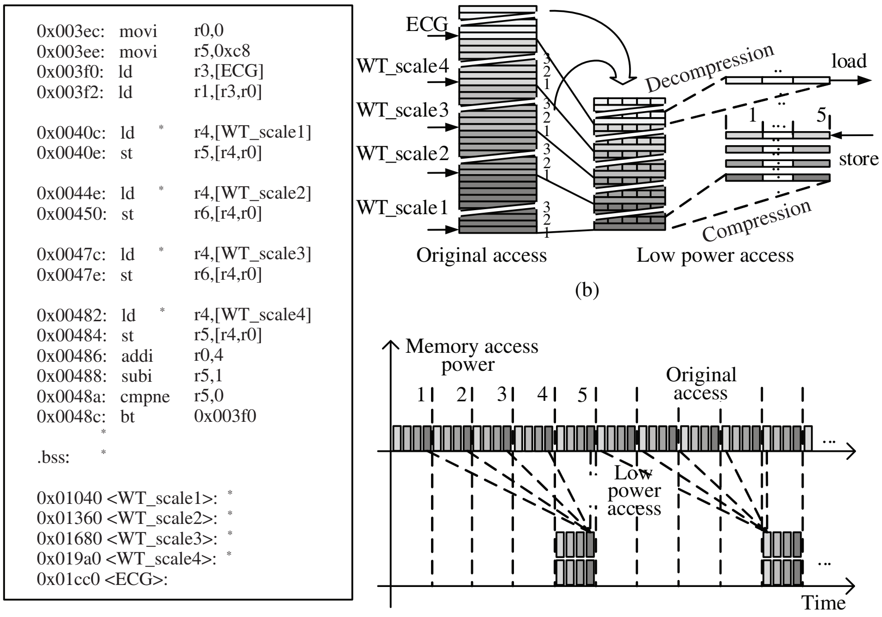


Figure 6 The address management.

strategy, and the accession is no longer averagely distributed. Compared with the sequential access of the original accessing strategy shown in Figure 7(c), the store power will be reduced through the compressed storage.

A set of load and store buffer are introduced for the low-power load and store accession. Each buffer set uses the FIFO replacement strategy when the address is missing. The load operation will be easily maintained through the FIFO strategy, but the store operation together with the compressed store management (a process that checks whether the packet meets the compressed storage strategy) will be much more complicated.

Several new mechanisms are introduced for the compressed store management. First, the high significant bits of the initially stored value will be used as the basis data of the packet, which will simplify the design of the basis data. Second, to reduce the memory accession, when a store address miss occurs, the original data will not be loaded, the packet in store buffer will be compressed store using the new basis data. Third, the store buffer will try to compress and record all of the store data in the packet, if there exist some addresses in the packet that have not been written by the processor, the packet will be stored to uncompressed backup section. When a data value locality miss occurs, meaning that the store operation whose address hits on the buffer, the data value does not hit on the basis data. An uncompressed backup packet space will be needed. The compression flag in the original compressed section will indicate if an uncompressed backup packet already exists. If no such backup packet exists, a new uncompressed backup packet is allocated, and additional loading of the data in compressed packet are used to ensure that the buffer holds all of the latest packet data. When a store address miss occurs, an old entry must be written back. The new entry compression flag will indicate where the data are going to be stored. Additionally, the old compression flag will indicate whether there exists an uncompressed backup packet for the entry. If a packet can be stored compressed, the backup packet will de-allocate.



```
0x003ec:  movi     r0,0
0x003ee:  movi     r5,0xc8
0x003f0:  ld       r3,[ECG]
0x003f2:  ld       r1,[r3,r0]

0x0040c:  ld    *  r4,[WT_scale1]
0x0040e:  st       r5,[r4,r0]

0x0044e:  ld    *  r4,[WT_scale2]
0x00450:  st       r6,[r4,r0]

0x0047c:  ld    *  r4,[WT_scale3]
0x0047e:  st       r6,[r4,r0]

0x00482:  ld    *  r4,[WT_scale4]
0x00484:  st       r5,[r4,r0]
0x00486:  addi     r0,4
0x00488:  subi     r5,1
0x0048a:  cmpne    r5,0
0x0048c:  bt       0x003f0
             *
.bss:        *

0x01040 <WT_scale1>:  *
0x01360 <WT_scale2>:  *
0x01680 <WT_scale3>:  *
0x019a0 <WT_scale4>:  *
0x01cc0 <ECG>:
```

(a)

(b)

(c)

Figure 7 The low-power accessing strategy. (a) The processing code of DWT; (b) the memory access diagram; (c) the access power distribution.

4 Experimental results

The experimental results are based on the MIT/BIH arrhythmia database and the DWT algorithm in a 32-bit data width processor, CK802. CK802 is an independent intellectual property processor implemented with a 16bits and 32bits wide combined instruction set, which is a RISC processor with two pipeline stages, focusing on ultra-low-power applications. The sample rate of the ECG signal is a critical parameters, which will directly affect the memory space and the complexity of the algorithm. The MIT/BIH database is the signal with 360 Hz, which is a typical sample rate for the ECG. The investigation begins by studying the key parameters for the potential compressibility at the proposed compressed storage strategy. Then, at the proposed low-power accessing strategy, the entry number is analyzed for the load and store buffer, which directly affects the compression ratio and energy consumption. Finally, the power estimation and performance impaction are described.

4.1 Potential memory space saving analysis

The packet set size and compressed data width (the width of the displacement) are the key parameters for memory space saving. At the mechanism in Section 3, the base data for a packet is aligned; if there exists only one address in the packet whose data cannot be compressed, the whole packet will be stored uncompressed in the backup section. The potential compression ratio of the proposed strategy will be different from the result presented in Section 2. Figure 8 describes the mean memory space saving for the MIT/BIH arrhythmia database. The bar with the negative value indicates that the memory is expanded; the majority of the packet suffers from the value locality miss.

For a given packet set size, a small compressed data bit width will significantly reduce the compression packet size; however, it will increase the value locality miss ratio. Optimal memory space saving is achieved at a compressed data width of 10 bits to 12 bits. Similarly, a large packet set size will reduce the compression packet size but increase the space penalty at every locality value miss. The results demonstrate that the optimal set size is 32 bytes to 64 bytes. The optimal memory space saving is 52% at a packet set size of 64 bytes and a compressed data width of 10 bits. Considering the resources of the load and store buffer, small packet set size of 32 bytes is chosen. And importantly, the data alignment of the compressed packet must be taken into consideration. Compressed data width of 8 bits and 12 bits,

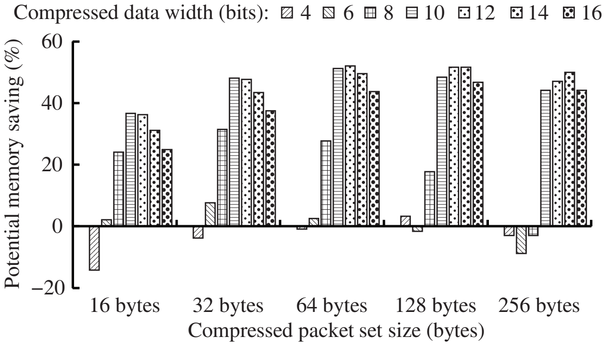


Figure 8 The potential compression ratio.

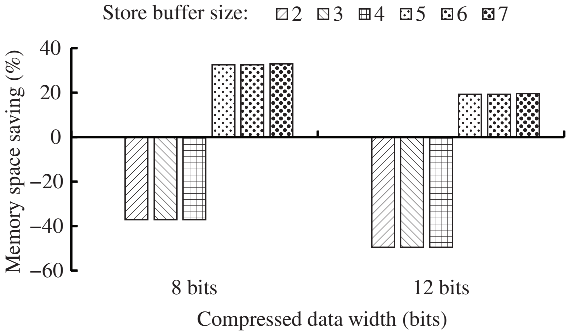


Figure 9 The memory saving at different store buffer.

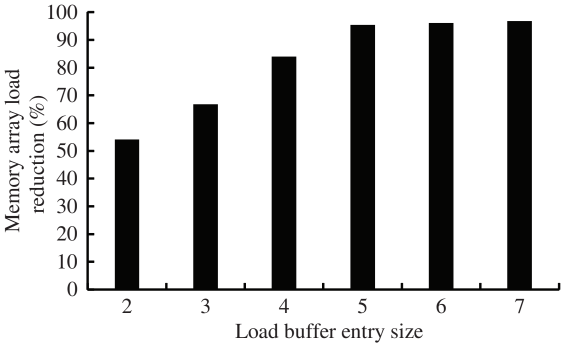


Figure 10 The load accession reduction to memory array.

which make the compressed packet size be 12 bytes and 20 bytes are selected for the candidate compression parameters for the implementation. These values can potentially save 32.7% and 47% of the memory.

4.2 Memory space saving analysis at the proposed store buffer

The compressed store mechanism binds with the store buffer described in Section 3. The entry size of the store buffer will directly affect the memory saving, which makes the real memory saving differ from the potential results shown in Figure 9.

There is an important mechanism in the proposed store buffer; the packet with some addresses not written by the processor will be uncompressedly stored. Two conditions will cause this situation: The address confliction of the store buffer and the address boundary of the continuously accessed address (the space of the wavelet transform), which is not aligned with the packet set size (32 bytes). The address confliction will occur when there is no enough store buffer entry. The processing flow in Figure 7 shows, there are four decomposition scales that are continuously stored, which means at least 4 store buffer will be needed to avoid the address confliction. Figure 9 provides the memory saving at different store buffer entries. For both of the compressed data width, there is a step change at the buffer size of 5 entries when the address confliction can be avoided. Furthermore, due to the not aligned address boundary of wavelet transform, the compressed data width of 8 bits will get a better memory space saving than 12 bits. The large compressed data bit width will cause large memory penalty. Five store buffer entries and compressed data bit width of 8 bits will be selected, which can save 32.5% of the memory space.

4.3 The power estimation

For the proposed memory architecture, the accessing power of the memory array is still dominant regarding the total memory power. The load and store buffer will directly affect the accession of the memory array.

For the store operation, address confliction of the store buffer is the main factor that causes the accession of the memory array. Based on the analysis of Figure 9, 5 store buffer entries can avoid the

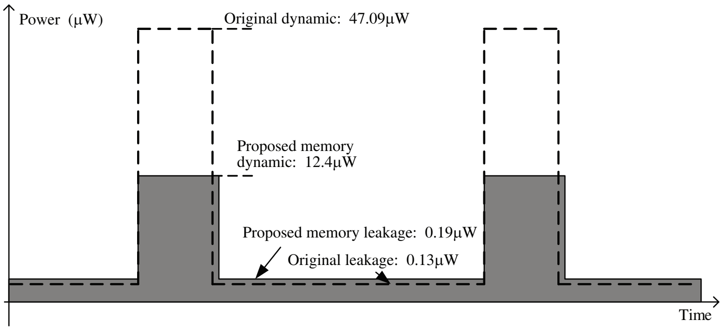


Figure 11 The power reduction of the proposed memory architecture.

Table 2 Access latency for the proposed architecture

Access	Description	Latency (cycles)	Ratio (%)
Load operation	Load hit on load and store buffer.	1	95.09
	Load miss but hit on the compressed section.	4	4.69
	Load miss but hit on the uncompressed backup section.	9	0.22
Store operation	Store address hit on load and store buffer.	1	91.57
	Store address hit but the value locality miss and the packet is in compressed section.	4	1.67
	Store miss cause a compressed packet entry write back.	4	6.51
	Store miss cause a uncompressed packet entry write back.	9	1.92

conflict; this can gain the optimal reduction of memory accession. In addition, it is worth noting that the maintenance of uncompressed backup memory will cost load and store memory accession; however, at the store buffer size of 5 entries, it only consumes a negligible 0.2% of the total memory accession.

Furthermore, similar address confliction will occur with the load buffer. When extracting the key point of the wavelet transform, 4 wavelet decomposed scale will be continuously accessed. So a 5 entries load buffer may get the optimal reduction of the memory load accession. Figure 10 studies the load access to the memory array at different load buffer sizes. Each bar represents the reduction ratio compared with the original memory load operations. As the buffer entry size increases, the load address confliction is decreased, indicating the higher memory load accession reduction in Figure 9. The highest reduction ratio of 95.4% will be achieved when the buffer size is greater than 4. Considering the buffer resources, 5 load buffer entries are chosen for implementation.

In summary, 5 store buffer entries and 5 load buffer entries are selected for the hardware implementation. Based on the CK802 low-power processor, the power of the proposed memory architecture is evaluated. The data memory array size is fixed to 4K bytes, and the working frequency is 1 MHz. The results in Figure 11 are evaluated based on the gate level simulation using the TSMC013g technology. The dynamic power of the DMEM can be reduced by 74%; however, the leakage power is slightly increased, but the proposed memory architecture still achieves an average power reduction of 68.1%. Because the proposed memory architecture can reduce the memory space consumption, the memory accession power can be further reduced by using a smaller memory array.

4.4 Performance estimation

Table 2 shows the basic latency of the proposed memory architecture, considering the most frequent operation, the load operation when it hits on the load and store buffer (95.09% of the total load operation), and the access latency caused by only one cycle. When the load miss transpires, depending on the compression flag, the latency is 4 cycles with a hit on the compressed memory section occurs (4.69% of the total load operation) and 9 cycles when a hit on the uncompressed memory section occurs (0.22% of the total load operation).

Similarly, 91.57% of the total store operation whose address is hitting on the load and store buffer have only latency of one cycle. There are two conditions that will cause more access latency: when a store value locality miss occurs and the old data need to be loaded from the compressed memory section, the latency is 4 cycles, taking a 1.67% share of the total store operation. When a store address miss occurs, one store buffer entry will be written back, and the latency will be 4 cycles when storing to the compressed memory section (only 6.51% of the total store operations). The latency will be 9 cycles when storing to the uncompressed backup memory section (only 1.92% of the total store operations).

The final performance results are evaluated using the MIT/BIH database as the basic database. The IMEM and DMEM are separately integrated with the CPU using an AHBlite bus protocol; the SOC platform is implemented in FPGA. The average negative impact on the performance is only 0.2%, and it can be concluded that the performance impact of the proposed memory architecture is negligible.

5 Related work

In previous work focusing on sub-threshold designing in low-power memory design for ultra-low-power sensor nodes, such as [27,28], up to a 60% power reduction was achieved. However, the drawbacks include suppressing the leakage and maintaining the stability of the memory access. Refs. [20, 23] employed the memory banking technology. The software will select individual banks after the memory is divided into banks; then, the software will put the other memory banks into low-power mode. However, the additional instruction will affect the program model, and frequent changes by the program to the memory pages will affect the system performance.

The feature of data value has been employed in some low-power memory architecture. Ref. [29] designed a low-power memory architecture to utilize the silent store whose newly stored data are the same as the old data. Ref. [29] employed a cache structure to check and ignore the silent store operation. However, compared with this work, the data value locality can cover more memory addresses and achieve more aggressive power reduction. Ref. [30] proposed a compressed store on the DMEM in an ultra-low-power oriented processor; a Huffman encoding code dictionary was used to encode the stored data and achieve a reduction of memory space. However, the fixed Huffman code dictionary cannot dynamically adapt the program, and it will consume additional memory space. This work utilizes the data value locality feature of the QRS detection algorithm and employs a much simpler way to accomplish the compression. Together with the low-power accession strategy, this work can achieve a significant power reduction for the ECG QRS detection algorithm with a negligible performance impact.

6 Conclusion

In this paper, a low-power storage compression memory architecture for the DWT-based QRS detection has been proposed and evaluated. Based on the most critical preprocessing stage, the feature of the value spatial locality is explored. A cost efficient compression strategy, a storage compression memory architecture and a low-power memory accession strategy are proposed. The new architecture can achieve 68.1% power reduction and save an average of 32.5% of memory space with a negligible 0.2% performance impact. The proposed memory architecture can achieve ultra-low power on a traditional, general-purpose, processor-based ECG sensor node.

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Pop V, de Francisco R, Pflug H, et al. Human++: wireless autonomous sensor technology for body area networks. In: Proceedings of the 16th Asia and South Pacific Design Automation Conference, Pacifico Yokohama, 2011. 561–566

2 Liu X, Zheng Y, Phyu M W, et al. Power and area efficient wavelet-based on-chip ECG processor for WBAN. In: Proceedings of International Conference on Body Sensor Networks (BSN), Singapore, 2010. 124–130

3 Liu X, Zheng Y, Phyu M W, et al. Multiple functional ECG signal is processing for wearable applications of long-term cardiac monitoring. IEEE Trans Biomed Eng, 2011, 58: 380–389

4 Kim H, Yazicioglu R F, Torfs T, et al. A low power ECG signal processor for ambulatory arrhythmia monitoring system. In: Proceedings of IEEE Symposium on VLSI Circuits (VLSIC), Honolulu, 2010. 19–20

5 Ieong C I, Mak P I, Lam C P, et al. A 0.83-QRS detection processor using quadratic spline wavelet transform for wireless ECG acquisition in 0.35-CMOS. IEEE Trans Biomed Circ Syst, 2012, 6: 586–595

6 Liu X, Zhou J, Yang Y, et al. A 457-nW cognitive multi-functional ECG processor. In: Proceedings of IEEE Asian Solid-State Circuits Conference (A-SSCC), Singapore, 2013. 141–144

7 Yazicioglu R F, Kim S, Torfs T, et al. A 30 W analog signal processor ASIC for portable biopotential signal monitoring. IEEE J Solid-State Circ, 2011, 46: 209–223

8 Kim H, Yazicioglu R F, Kim S, et al. A configurable and low-power mixed signal SoC for portable ECG monitoring applications. In: Proceedings of Symposium on VLSI Circuits (VLSIC), Kyoto, 2011. 142–143

9 Jeon D, Chen Y P, Lee Y, et al. An implantable 64nW ECG-monitoring mixed-signal SoC for arrhythmia diagnosis. In: Proceedings of IEEE Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, 2014. 416–417

10 Ashouei M, Hulzink J, Konijnenburg M, et al. A voltage-scalable biomedical signal processor running ECG using 13pJ/cycle at 1MHz and 0.4 V. In: Proceedings of IEEE Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, 2011. 332–334

11 Kim H, Choi S, Yoo H J. A low power 16-bit RISC with lossless compression accelerator for body sensor network system. In: Proceedings of Asian Solid-State Circuits Conference, Hangzhou, 2006. 207–210

12 Kwong J, Chandrakasan A P. An energy-efficient biomedical signal processing platform. IEEE J Solid-State Circ, 2011, 46: 1742–1753

13 ATMEL. 8-bit Atmel Microcontroller with 128Kbytes in-System Programmable Flash Rev 2467XCAVRC06/11. 2011

14 Texas Instruments. MSP430x2xx Family User's Guide. 2013

15 Boichat N, Atienza D, Khaled N. Wavelet-based ECG delineation on a wearable embedded sensor platform. In: Proceedings of 6th International Wearable and Implantable Body Sensor Networks, Berkeley, 2009. 256–261

16 Yseboodt L, De Nil M, Huisken J, et al. Design of 100 µW wireless sensor nodes for biomedical monitoring. J Signal Process Syst, 2009, 57: 107–119

17 Mamaghanian H, Khaled N, Atienza D, et al. Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes. IEEE Trans Biomed Eng, 2011, 58: 2456–2466

18 Polastre J, Szewczyk R, Culler D. Telos: enabling ultra-low power wireless research. In: Proceedings of 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, 2005. 364–369

19 Lee K H, Verma N. A 1.2C0.55 V general-purpose biomedical processor with configurable machine-learning accelerators for high-order, patient-adaptive monitoring. In: Proceedings of European Solid State Circuits Conference, Bordeaux, 2012. 285–288

20 Ekanayake V, Kelly IV C, Manohar R. An ultra low-power processor for sensor networks. ACM SIGARCH Comput Architect News, 2004, 32: 27–36

21 Duarte F, Hulzink J, Zhou J, et al. A 36µW heartbeat-detection processor for a wireless sensor node. ACM Trans Des Automat Electron Syst, 2011, 16: 51

22 Nazhandali L, Minuth M, Zhai B, et al. A second-generation sensor network processor with application-driven memory optimizations and out-of-order execution. In: Proceedings of International Conference on Compilers, Architectures and Synthesis for Embedded Systems, San Francisco, 2005. 249–256

23 Pan J, Tompkins W J. A real-time QRS detection algorithm. IEEE Trans Biomed Eng, 1985, 32: 230–236

24 Torfs T, Yazicioglu R F, Kim S, et al. Ultra low power wireless ECG system with beat detection and real time impedance measurement. In: Proceedings of Biomedical Circuits and Systems Conference (BioCAS), Paphos, 2010. 33–36

25 Martnez J P, Almeida R, Olmos S, et al. A wavelet-based ECG delineator: evaluation on standard databases. IEEE Trans Biomed Eng, 2004, 51: 570–581

26 Moody G B, Mark R G, Goldberger A L. PhysioNet: a web-based resource for the study of physiologic signals. IEEE Eng Med Biol Mag, 2001, 20: 70–75

27 Teman A, Pergament L, Cohen O, et al. A 250 mV 8 kb 40 nm ultra-low power 9T supply feedback SRAM (SF-SRAM). IEEE J Solid-State Circ, 2011, 46: 2713–2726

28 Lutkemeier S, Jungeblut T, Berge H K O, et al. A 65 nm 32 b subthreshold processor with 9T multi-Vt SRAM and adaptive supply voltage control. IEEE J Solid-State Circ, 2013, 48: 8–19

29 Kücük G, Basaran C. Reducing energy consumption of wireless sensor networks through processor optimizations. J Comput, 2007, 2: 67–74

30 Hanson S, Seok M, Lin Y S, et al. A low-voltage processor for sensing applications with picowatt standby mode. IEEE J Solid-State Circ, 2009, 44: 1145–1155