A Hardware Architecture for MAP Decoding Based on Nibble Alignment

Seungkwon Cho, Sok-Kyu Lee
Short Range Radio Transmission Research Team
Electronics and Telecommunication Research Institute
Daejeon, Korea
{skcho, sk-lee}@etri.re.kr

Youngnam Han
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon, Korea
ynhan@kaist.ac.kr

Abstract—**In IEEE 802.16-2009 standard, a Subscriber Station (SS) has to meet timing constraints imposed by uplink and downlink MAP relevance. Thus, the standard adopts nibble alignment in MAP Information Elements (IEs) in order to support fast MAP decoding. However, some MAP IEs are not nibble-aligned with increased implementation complexity of a MAP decoder. In this paper, we present a hardware architecture designed to efficiently decode MAP message on a nibble basis while suppressing the increase in implementation complexity. The feasibility of the proposed architecture is verified by FPGA synthesis and its performance is presented in terms of MAP decoding time.**

Keywords-IEEE 802.16-2009; MAP relevance; nibble alignment; MAP decoding

I. INTRODUCTION

Recently, the IEEE 802.16 Working Group on Broadband Wireless Access (BWA) Standards released IEEE 802.16-2009 [1] as a new base standard that supersedes and makes obsolete IEEE standard 802.16-2004 and all of its subsequent amendments and corrigenda. The new standard is expected to provide high data rate communications in metropolitan area networks (MANs). In order to support such a high speed transmission, careful considerations should be given to the implementation architecture of both physical (PHY) layer and medium access control (MAC) layer.

From the traditional hardware and software partitioning methodologies point of view, most of MAC functions are implemented by software because it enables a system to have more flexibility and reduced time-to-market. However, as transmission speed increases, more and more timing critical MAC functions have been migrated from software to hardware due to the inherent latency of software processing such as interrupt latency and relatively low performance compared with the dedicated hardware processing [2]. One of such examples is MAP decoding, where MAP is a MAC message that defines frame structure with the information about the scheduled access both downlink MAP (DL-MAP) and uplink MAP (UL-MAP) sub-frames. Each subscriber station (SS) can access the frame only after correctly receiving and decoding the MAP message. Since all the MAP processing should be completed to meet the MAP relevance, MAP decoding is regarded as one of timing critical MAC functions in SS. As a result, the IEEE 802.16-2009 standard took the fast map decoding into consideration and adopted nibble alignment as an underlying concept when each MAP IE is designed. It is because nibble alignment reduces the implementation complexity of MAP decoder [3] and thus motivates hardware implementation.

The standard explicitly requests that both hybrid automatic repeat request (HARQ) MAP IE and subburst IE should be nibble-aligned. However, most of subburst IEs, in IEEE 802.16-2009, are not nibble-aligned. Even though there was an effort to fix this inconsistency in the standard [4], the standardization body rejected this opinion due to the compatibility issue with the existing legacy SS. The inconsistency is less likely to be resolved even in the future and it is an obstacle to implementing the MAP decoder by hardware. Therefore, in order to achieve fast MAP decoding with low implementation complexity, it is necessary for a MAP decoder to deal with the MAP in nibble-by-nibble fashion whether the MAP IE is nibble-aligned or not. In this paper, we propose a hardware acceleration architecture for MAP decoding that maintains nibble processing whether the MAP IE is nibble-aligned or not. The proposed architecture is fully implemented by the hardware and its implementation results are provided in this paper.

The rest of this paper is structured as follows. Section II provides background and description of MAP decoding focused on nibble alignment. The definition of nibble alignment is also introduced. The proposed hardware acceleration architecture for MAP decoding and its detail operations are presented in Section III, and FPGA implementation results are covered in Section IV. Section V shows the performance of the implemented MAP decoder in terms of MAP decoding time. Section VI concludes this paper.

II. BACKGROUND AND DESCRIPTION OF MAP DECODING IN IEEE 802.16-2009

MAP itself is simply a MAC management message, which is carried in a payload of MAC Protocol Data Unit (PDU). The inband signaling nature of MAP message and its nontrivial signaling overhead have made the original MAP message obsolete in most of implementations [5]. The compressed MAP is a simplified version of the original MAP adopted to reduce the size of original MAP message. Fig. 1 shows the hierarchical structure of the compressed MAP in IEEE 802.16-2009 focusing on the subburst allocations by HARQ DL/UL

 ISBN: 978-1-61208-164-9

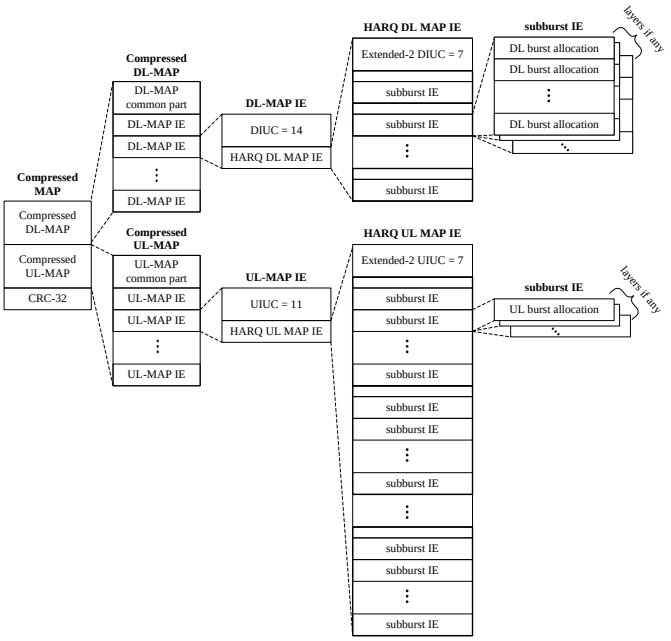

Figure 1. Hierarchical structure of the compressed MAP in IEEE 802.16-2009.

MAP IEs. Contrary to the original MAP message, the compressed map has no generic MAC header (GMH) and the compressed DL-MAP appears first with the first three bits set to 110. The compressed DL-MAP indicates the presence of the compressed UL-MAP by a single bit in the compressed DL-MAP data structure. The compressed DL-MAP includes several DL-MAP IEs identified by a downlink usage interval code (DIUC). The DIUC defines the type of DL access and the DL burst profile associated with that access. A DL-MAP IE entry with a DIUC = 14 indicates that it carries special information and conforms to the DL-MAP Extended-2 IE format [1]. Among the DL-MAP Extended-2 IEs, HARQ DL MAP IE and Persistent HARQ DL MAP IE are extraordinary MAP IEs in that they have a HARQ region for burst allocations with different modulation and coding scheme. Each burst inside the HARQ region is separately encoded and referred to as a subburst. These subburst allocations are specified by subburst IEs included in either HARQ DL MAP IE or persistent HARQ DL MAP IE. The compressed UL-MAP has similar hierarchical structure to the compressed DL-MAP with differences on how to allocate multiple subbursts, as is shown in Fig. 1. In case of DL subburst allocations, multiple DL subbursts with the same HARQ and coding scheme can be allocated by a single subburst IE. On the contrary, every single UL subburst allocation requires its own subburst IE.

At every frame, the compressed MAP is broadcast to all SS in a sector and each SS begins MAP decoding by checking the CRC appended at the end of the compressed MAP. The MAP is discarded and the SS does not access the relevant DL subframe and UL subframe when a CRC error is detected. Otherwise, the SS parses the MAP and extracts all the control information included in MAP IEs. This MAP decoding process is timing-critical due to MAP relevance. Fig. 2 shows MAP relevance with orthogonal frequency-division multiple access

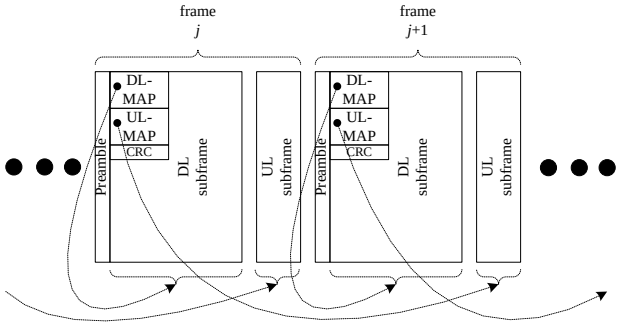

Figure 2. TDD MAP relevance for OFDMA PHY

(OFDMA) PHY in Time Division Duplex (TDD) system. According to the standard, information in DL-MAP always pertains to the same frame in which the MAP was received. In case of UL-MAP, all the information in the UL-MAP applies to UL subframe starting at the Allocation Start Time measured from the beginning of the current frame. TDD system has the ability to control the transition time between DL subframe and UL subframe by adjusting the Allocation Start Time included in the UL-MAP [1]. However, the maximum value of Allocation Start Time is bounded by twice the frame duration and the minimum value is equal to or greater than one frame duration. Referring to Fig. 2, this means that the UL subframe relative to UL-MAP in frame *j* always exists in the next frame (*j*+1). Therefore, all the uplink processing should be completed no later than the beginning of the UL subframe in the next frame in IEEE 802.16-based TDD-OFDMA system.

UL-MAP decoding must be regarded as one of the timing-critical uplink operations because of the upper bound for the UL-MAP relevance. A successful MAP reception without CRC error initiates the decoding process. After decoding the received MAP, the SS can identify the allocated uplink resources. Then the SS constructs MAC PDU by carrying out packing, fragmentation, GMH generation, payload encryption, and CRC calculation. The constructed MAC PDUs are finally concatenated to build an uplink burst and handed over PHY for transmission. It should be noted that all of these uplink processing cannot be done simultaneously but sequentially. Moreover, it is not until MAP decoding is completed that all of these uplink processing can be initiated. Therefore, fast UL-MAP decoding is very important, because the SS cannot access the scheduled uplink medium when it fails to keep the UL-MAP relevance even if SS succeeds in MAP decoding.

Fast MAP decoding is also necessary for downlink. The SS should not ignore any data following MAP message because it has no information about whether there are its own bursts or not. The downlink PHY of SS should buffer all the received data until it receives the result of DL-MAP decoding that indicates the detail allocation information about DL bursts. Thus, fast DL-MAP decoding has a beneficial effect on downlink PHY of the SS in the reduction of the required size of buffer memory. Besides the advantage in memory size, fast DL-MAP decoding enables the SS to send ACK/NAK of DL HARQ burst using UL ACK region in the next uplink subframe. The ACK/NAK signal of DL HARQ burst is sent by

 ISBN: 978-1-61208-164-9

the SS after fixed delay specified by HARQ DL ACK delay for DL burst field in Uplink Channel Descriptor (UCD) message. The standard allows three values for this delay, one, two, or three frame offset. Most implementation adopts the minimum delay of one frame because the fast HARQ retransmission is preferred. In this case, before the beginning of next UL subframe, the SS should complete the DL HARQ processing such as the checking of 16 bits CRC appended at each DL HARQ burst and figure out the offset in the UL ACK region in order to transmit ACK/NACK signals. The offset in the UL ACK region is determined by the order of HARQ-enabled DL burst in the DL-MAP. Besides, UL ACK region can be defined by not only UCD message but also HARQ ACK Region Allocation IE included in UL-MAP. Thus, when the UL ACK region is allocated by UL MAP IE, entire MAP decoding is needed to initiate DL HARQ processing. As a result, fast MAP decoding is also helpful for DL HARQ operations because the minimum HARQ DL ACK delay can be adopted.

III. DESCRIPTION OF THE PROPOSED NIBBLE ALIGNMENT SCHEME

As we mentioned earlier, fast MAP decoding is strongly required and the standard supports it by adopting nibble alignment in MAP IE. The nibble alignment in MAP IE can be classified into two broad categories according to nibble alignment, either at the end of MAP IE and or inside MAP IE. Nibble alignment at the end of MAP IE means that the length of entire MAP IE should be multiples of 4 bits. In the standard, there are three kinds of nibble alignment inside MAP IE. One is nibble alignment at the end of an if-else clause inside MAP IE regardless of whether the if-clause or the else clause is executed. The others are nibble alignment before a loop and inside the loop. Nibble alignment before a loop means that the first bit of a loop should be the MSB (Most Significant Bit) of the nibble and nibble alignment inside the loop means that the length of a loop should be multiples of 4 bits. UL Sounding Command IE is the best example that shows how to accomplish the nibble alignment inside MAP IE. UL Sounding Command IE appends padding bits for byte alignment at the end of MAP IE and utilizes reserved bits 11 times for the nibble alignment inside MAP IE.

Even though all the DL/UL MAP IEs in IEEE 802.16 are byte-aligned or nibble-aligned at the end of MAP IE, nibble alignment inside MAP IE is rarely kept by most of MAP IEs. It is because the nibble alignment inside MAP IE is not mandatory in general MAP IEs. However, the standard explicitly mandates subburst IEs to be nibble-aligned [1]. In spite of this requirement for subburst IEs, unfortunately, most of subburst IEs are not nibble-aligned. Therefore, the current IEEE 802.16-2009 loses consistency throughout the standard with respect to the nibble alignment inside subburst IEs. As a result, the fact that not all the MAP IEs are nibble-aligned should be taken into consideration especially when the MAP decoder is implemented by hardware for the sake of fast MAP decoding. The contribution of our paper is to present a hardware architecture designed for fast MAP decoding on a nibble basis.

Fig. 3 shows the proposed MAP decoder architecture for nibble alignment. According to the control of nibble alignment

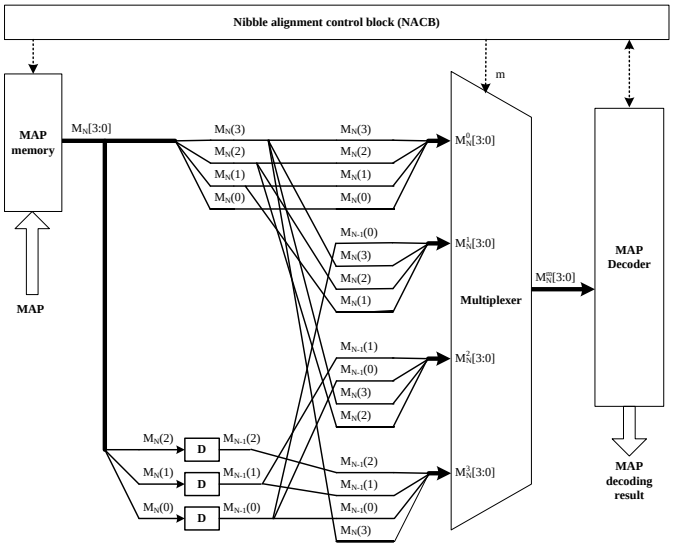

Figure 3. Proposed MAP decoder architecture for nibble alignment

control block, the MAP memory stores valid MAP data with no CRC error and outputs the MAP by the unit of nibble, $M_N[3:0]$, to both multiplexer and D flip-flop, where N denotes time index (N = 0, 1, 2, …). Let $M_0[3:0]$ be the arbitrary value before the MAP memory is filled with received MAP data. Thus $M_1[3:0]$ is the first nibble at the beginning of the MAP. D flip-flop takes a single bit $M_N(B)$ as an input and produces delayed version of it by one time index, $M_{N-1}(B)$, where B is bit index (B = 0, 1, 2, 3) and $M_N(3)$ corresponds to MSB of the nibble $M_N[3:0]$. The initial value of each D flip-flop is set to $M_0(2)$, $M_0(1)$, and $M_0(0)$, respectively, and they are allowed to be arbitrary value. The multiplexer is ordered to select one of the following four inputs according to the alignment mode m (m = 0, 1, 2, 3) specified by nibble alignment control block (NACB). Therefore, the output of the multiplexer is one of the following values:

$$M_N^0[3:0] = M_N[3:0]$$

$$M_N^1[3:0] = M_{N-1}(0) \,\&\, M_N[3:1]$$

$$M_N^2[3:0] = M_{N-1}[1:0] \,\&\, M_N[3:2]$$

$$M_N^3[3:0] = M_{N-1}[2:0] \,\&\, M_N(3)$$

where the operator & stands for bit concatenation. The multiplexer generates the nibbled-aligned MAP, $M_N^m[3:0]$, that is an input to MAP decoder. Finally, MAP decoder reads $M_N^m[3:0]$ and produces MAP decoding result.

More detail operation of nibble alignment operations are illustrated in Fig. 4. Assume that the MAP decoder is in state K and the current mode is m. Let the current output of MAP memory be $M_{n+1}[3:0]$ ($n \in N$). At this time, suppose that the MAP decoder has found that a loop in the MAP IE begins at the bit $M_{n+1}^m(b)$ where $b \in B$ and $b \neq 3$. Notice that if b=3, the current input of MAP decoder, $M_{n+1}^m[3:0]$, is nibble-aligned before the loop and we do not need any further operations for nibble alignment. Since the first bit corresponding to the beginning of a loop is not the MSB of $M_{n+1}^m[3:0]$, the MAP

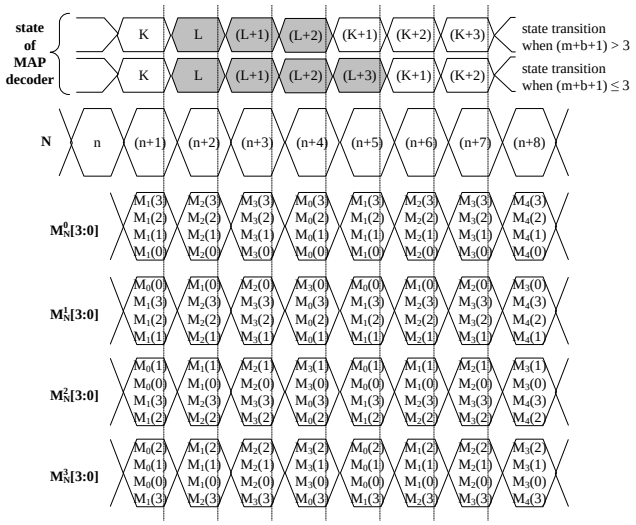

Figure 4. State transition of MAP decoder for nibble alignment

decoder makes a request for nibble alignment at the bit $M_{n+1}^{m}(b)$ to NACB. Thus only (3-*b*) bits of the current nibble, $M_{n+1}^{m}[3:(b+1)]$, is used for map decoding in state *K* and (*b*+1) bits are not used. The MAP decoder informs NACB of the number of bits that is not used for decoding in the current state and it transits to the overhead state *L*. The states from *L* to (*L*+3) are overhead states because no actual MAP decoding takes place in these states and they are used only for nibble alignment. In state *L*, NACB changes the read address for MAP memory so that the output of MAP memory can be $M_0[3:0]$ in state (*L*+2). At the same time, it carries out the addition of (*m*+*b*+1). In state (*L*+1), NACB finds a new mode, *m*', by performing following modulo operations with (*m*+*b*+1) obtained from the previous state:

$$m' = (m+b+1) \bmod 4.$$

where mod is a modulo operator. In state (*L*+2), NACB compares (*m*+*b*+1) with 3 and it transits to state (*K*+1) if (*m*+*b*+1) is greater than 3. Otherwise, it transits to state (*L*+3) which is simply a temporary state before state (*K*+1). The output of multiplexer in state (*K*+1) is now $M_{n+5}^{m'}[3:0]$ if (*m*+*b*+1) > 3 and $M_{n+6}^{m'}[3:0]$, otherwise. In either case, the MSB of the input to the MAP decoder is the start of the loop in state (*K*+1). The MAP decoder resumes MAP decoding process from the state (*K*+1) with the nibbled-aligned MAP data. Therefore, the presented alignment scheme makes it possible for the MAP decoder to operate on a nibble basis.

IV. IMPLEMENTATIONS

The feasibility of the proposed architecture is investigated by implementing the compressed MAP decoder using Xilinx Virtex-4 XC4VLX200 FPGA. The implemented MAP decoder is capable of decoding the MAP including 10 DL MAP IEs and 17 UL MAP IEs. For the commercial Mobile WiMAX system [6], only sub-MAP and its relevant MAP IEs are excluded from the implementations. The implementation reveals that the nibble alignment inside the loop and at the end of if-else clause is much more important than before a loop. If the if-else clause is not nibbled alignment, the logic size doubles from the end of if-else clause. Moreover, if the loop is not multiples of 4 bits, the logic size of the loop increases by two times when the remainder of the length of the loop modulo 4 equals to 2, or increases by four times when the remainder of the length of the loop modulo 4 equals to 1 or 3. Since the proposed architecture supports nibble alignment for any MAP IEs, the MAP decoder can operate on a nibble basis while avoiding the dramatic increase in the size of required logic when the MAP IE is not nibble-aligned. Table I indicates the implementation results in term of FPGA resource utilization. Only a small fraction of FPGA resources are utilized. According to the synthesis report, the implemented MAP Decoder can operate up to 140 MHz clock frequency.

TABLE I. FPGA IMPLEMENTATION RESULTS

Resources	Available (a)	Used (b)	Utilization (b/a)
Number of Slice Filp Flops	178,176	5,649	3%
Number of 4 input LUTs	178,176	6,492	3%
Number of occupied slices	89,088	5,090	5%
Number of FIFO16/RAMB16s	336	7	2%
Number of DSP48s	96	3	3%

V. PERFORMANCE

The performance of the implemented MAP decoder is investigated by register transfer level simulation with 80MHz clock speed. Performance evaluation is carried out in terms of MAP decoding time, which is defined as the time elapsed between the moment the MAP decoder reads the first nibble and the moment it reads the last nibble of the MAP. The MAP decoding time does not include the time required to verify CRC-32 appended at the end of MAP because the MAP memory stores valid MAP data with no CRC error. As we mentioned previously, most of subburst IEs are not nibble-aligned. Thus, in this section, we concentrate on investigating the MAP decoding time of a MAP comprised of subburst IEs.

Two different kinds of MAP constructions are considered to study the performance of the implemented MAP decoder. One is a MAP construction for DL 2x2 Spatial Multiplexing

TABLE II. MAP IES FOR a)DL 2X2 SM AND b)UL 2 LAYER CSM

	MAP IE
a)	Compressed DL MAP
	CID Switch IE
	Space-Time Coding (STC)/DL Zone switch IE
	HARQ DL MAP IE
	MIMO DL Chase HARQ subburst IE
	Dedicated MIMO DL Control IE
	Compressed UL MAP
b)	Compressed DL MAP
	Compressed UL MAP
	CDMA allocation IE
	CDMA allocation IE
	HARQ ACKCH Region Allocation IE
	FASTFEEDBACK allocation IE
	PAPR Reduction/Safety Zone/Sounding Zone Allocation IE
	UL Zone Switch IE
	HARQ UL MAP IE
	MIMO UL Chase HARQ subburst IE

 ISBN: 978-1-61208-164-9

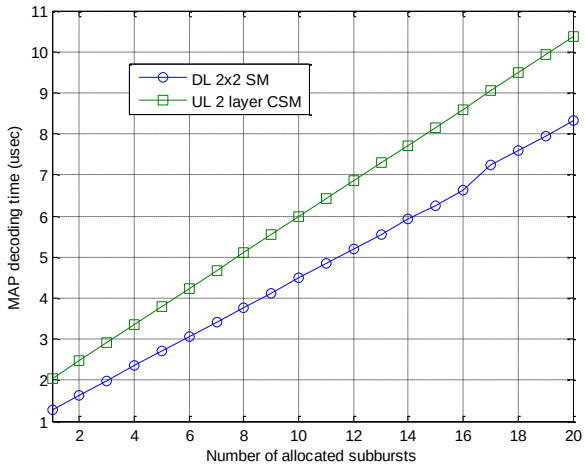

Figure 5. MAP decoding time versus the number of alloated subbursts

(SM) subburst allocations with no UL allocations. The other is a MAP construction for UL 2 Layer Collaborative Spatial Multiplexing (CSM) subburst allocations with no DL allocations. Table II shows the details of both MAP constructions. Each row of Table II is carefully indented in order to show MAP hierarchy. Notice that DL 2x2 SM subbursts and UL 2 Layer CSM subbursts are allocated by MIMO DL Chase HARQ subburst IE and MIMO UL Chase HARQ subburst IE, respectively. In case of DL 2x2 SM, the for() loop inside MIMO DL Chase HARQ subburst IE iterates as the number of subburst allocations increases. On the contrary, in case of UL 2 Layer CSM, the iteration is caused by for() loop surrounding MIMO UL Chase HARQ subburst IE. In either case, the size of MAP increases as the number of subburst allocations increases.

Fig. 5 shows the MAP decoding time of the implemented MAP decoder with respect to the number of allocated subbursts. Regardless of the number of subburst allocations, MAP decoding time of UL 2 layer CSM is larger than that of DL 2x2 SM . It is because the MAP size of UL 2 layer CSM is much bigger than that of DL 2x2 SM due to the overhead imposed by UL control region allocation. CDMA allocation IEs, HARQ ACKCH Region Allocation IE, FASTFEEDBACK allocation IE, and PAPR Reduction/Safety Zone/Sounding Zone Allocation IE is needed to allocate UL control regions such as ranging regions, HARQ ACK region, CQI region, and Sounding channel, respectively. Even though MAP coding time of each scheme is different, their MAP decoding time increase linearly for the observed number of allocated subbursts. However, in case of UL 2 layer CSM, MAP decoding time abruptly increases when the number of allocated subbursts changes from 16 to 17. Since MIMO UL Chase HARQ subburst IE can be used to allocate 16 subbursts at most, one more MIMO UL Chase HARQ subburst IE is needed to allocated 17 subbursts. Thus, the overhead arisen from one additional subburst IE makes a rapid increase at 17 subburst allocations.

Obviously, MAP decoding may be processed by software. In such a case, the initiation of MAP decoding will rely on an interrupt driven by channel decoder. It is interesting that the mean value of interrupt latency is approximately 10 μsec in a PC operating at 2GHz CPU core [2]. From Fig. 5, notice that the implemented MAP decoder operating at 80MHz can decode a MAP with 19 subburst allocations within 10 μsec. In other words, even before a software MAP decoder is invoked, the presented hardware MAP decoder can complete MAP decoding in most cases in Fig. 5. Thus, fast MAP decoding is the benefit of the proposed hardware acceleration architecture for map decoding that maintains nibble processing whether the map IE is nibble-aligned or not.

VI. CONCLUSIONS

MAP decoding by hardware is required for the fast MAP decoding that is needed to achieve the broadband access provided by the system based on IEEE 802.16-2009. The nibble alignment is adopted by the standard to support the fast MAP decoding while suppressing the increase in implementation complexity. However, some of the MAP IEs are not nibbled-aligned, which causes difficulties when the MAP decoder is implemented by hardware. In this paper, we propose a hardware architecture for MAP decoding that maintains nibble processing with the help of nibble alignment. From the synthesis result and the investigated MAP decoding time, we can conclude that the proposed architecture facilitates implementing a realistic fast MAP decoder with low hardware complexity.

ACKNOWLEDGMENT

This work was supported by the IT R&D program of MKE/KEIT. [KI002108, Research on Radio Transmission Technology for IEEE 802.11 VHT WLAN].

REFERENCES

[1] IEEE Standard for Local and Metropolitan Netoworks, Part 16: Air Interface for Broadband Wireless Access Systems, IEEE Std. 802.16-2009, May. 2009.

[2] Sunkyung Shin, Seungkwon Cho, Jaewoo Park, Yeong-Gon Lee, and Sok-Kyu Lee, “MAC HW/SW partitioning for aggregation in IEEE 802.11n based on a interrupt latency measurement”, ICCE 2010, pp. 291-292, Jan. 2010.

[3] Jaejoon Park, Seokheon Cho, Youngil Kim, Chulsik Yoon, Mihyun Lee, Inseok Hwang, and Panyuh Joo, “Corrections for Nibble Alignment in MAP_IEs”, IEEE 802.16’s 802.16 Task Group e, June. 2005.; http://www.ieee802.org/16/tge/contrib/C80216e-05_271r2.pdf

[4] Seungkwon Cho, Seokhun Cho, Jaesun Cha, and Young-il Kim, “Corrections for nibble alignment in MIMO HARQ subburst IEs”, IEEE 802.16’s 802.16 Task Group Maintence, Jun. 2008, http://www.ieee802.org/16/maint/contrib/C80216maint-08_251.doc.

[5] Fan Wang, A. Ghosh, C. Sankaran, P. Fleming, F. Hsieh, and S. Benes, “Mobile WiMAX Systems: Performance and Evolution”, IEEE Commun. Mag., vol. 46, pp. 41-49, Oct. 2008.

[6] WiMAX ForumTM Mobile Protocol Implementation Conformance Statement (PICS) Proforma, DRAFT-T24-001-R010v06-A, WiMAX Forum, Jun. 2009.

 ISBN: 978-1-61208-164-9