International Journal of Computer Applications (0975 – 8887)
Volume 9– No.9, November 2010

A Novel Area Efficient Folded Modified Convolutional Interleaving Architecture for MAP Decoder

S.Shiyamala
Department of ECE
SSCET
Palani, India.

Dr.V.Rajamani
Principal
IGCET
Trichy,India

ABSTRACT

This paper proposes a novel area-efficient folded modified convolutional interleaving (FMCI) architecture for MAP decoder. The end to end delay of proposed FMCI requires only 2M. In addition, the number of latches can reduce to M-2. Hence the proposed FMCI architecture has lesser end to end delay and minimizes the number of latches usage as compare with block interleaving and convolutional interleaving. In addition to this the life time analysis and forward-backward allocation has also been implemented. Therefore, FMCI architecture can reduce the memory elements about 88% as compare with block interleaving for the particular condition when M = NJ

General Terms

Performance

Key words

MAP decoder, convolutional interleaving, folded technique, latches, end to end delay, area efficient

1. INTRODUCTION

Internet traffic is increasing many folds in the recent years due to the advancement of communication technologies. Most of the communication is in digital. This is due to less noise and also faster transmission rate. In the digital communication system coding and decoding are important mechanisms in both transmitter and receiver side. Many algorithms were derived by many researchers in the digital communication systems. Some of them are Shannon, Viterbi algorithm, MAP/BCJR and Max-log-MAP. Turbo code was first introduced in 1993 by Berrou, Glavienx and Thitimajshima. It provides soft output [3] and represents a quantum leap in channel coding performance for deep space applications.

A class of convolutional codes, called TURBO codes, whose performance is in terms of BER are close to the Shannon limit. RSC codes can be better than NSC code for high code rate[11]. Turbo codes [8] [9] have been incorporated into many important wireless protocols from deep-space communication to mobile communication .They are used extensively in 3G mobile telephony standards, MediaFLO, terrestrial mobile television systems has integrated turbo codes as the error-correction algorithm. IEEE, 802.16, a wireless metropolitan network standard, also uses turbo coding; DVB-RCS, DVB-RCT, INMARSAT, EUTELSAT and BRAN [7] are other systems which use turbo codes as the channel coding algorithm. It has large decoding latency and low throughput due to iterative decoding. For high rates, turbo codes are not efficient. For this high speed decoding (MAP) schemes have to be employed.

The process of turbo code starts [5] with the formation of a posteriori probabilities (APPs) for each bit, which is followed by choosing the data bit value that corresponds to the maximum a posteriori (MAP) probability for that data bit. High throughput architecture can be classified into parallel processing, look ahead computation and algorithm reformulation approaches.The MAP algorithm provides not only the estimated sequence, but also the probabilities for each bit that has been decoded correctly. The decoding complexity of the MAP algorithm has been reduced in log-MAP algorithm [7][9] by operating it in the log- domain .The sub optimal implementation of log-MAP decoder called the max-log-MAP decoder. All log-MAP decoder are based on BCJR (Bahl-Cocke-Jelinex-raviv) algorithm.

Interleaving is a form of time diversity that mitigates the effects of error bursts. It is a way to arrange the data in a non – contiguous way to increase performance of the decoder. This enhancement comes at the cost of introducing additional time-delay, memory space requirements and system complexity. Convolutional interleaving that cuts down time delay and memory space increase considerably.

Instead of block interleaving, combine the convolutional interleaving technique with MAP decoder. Block interleaving occupies more space for the same operation. To reduce the memory requirement and end to end delay convolutional interleaving [1] technique is applied along with folded technique [10].

The rest of the paper is organized as follows: In section 2, brief definition of an interleaving, modified convolutional interleaving is discussed. In section 3, FMCI technique is deeply discussed. In section 4, based on the results, compare the memory requirement and end to end delay for convolutional interleaving with FMCI finally section 5 contains the conclusions.

2. CONVOLUTIONAL INTERLEAVING

The interleaver shuffles the code symbol over a span of several block lengths or several constraint lengths. Interleaving techniques are traditionally used to enhance the quality of digital transmission. This is usually accomplished by scrambling successive symbols of the transmitted sequence into different time slots. A channel is considered fully interleaved when consecutive symbols of the received sequence appear to be independent, i.e., not affected by the same error burst. In block interleaving, permutation of the block is accomplished by filling the columns of an M-row by N- columns (M x N).For block codes, M should be larger than the code block length, while for convolutional codes, M should be larger than constraint length.

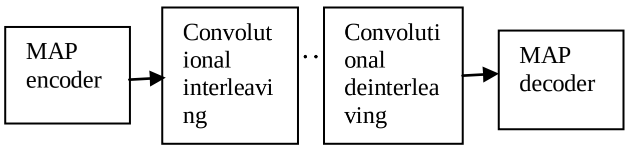


Figure 1 Modified Architecture – when M = N

Convolutional interleavers have been proposed by Ramsey and Forney. It is non block interleavers, because of less delay; it has been preferred for some application. The code symbols are shifted into N registers, [1] each successive registers provides J symbols more storage than did the proceeding one.

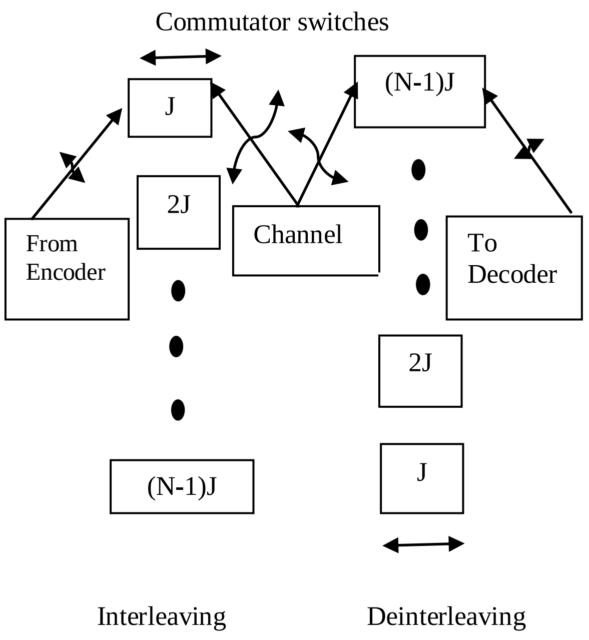
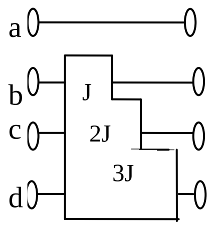


Figure 2 Convolution interleaver / deinterleaver

Some input data will remain in the interleaver memories and appear at the interleaver later. Optimized interleaver is obtained by deletion of stuff bits that has been located at the end parts. Figure (2) shows the example of a simple convolutional four–register (J=1) interleaver being loaded by a sequence of code symbols.

The performance of convolutional interleaving is similar to block interleaving. Here the end to end delay is M (N-1), symbols without propagating delay when M=NJ where (J=1). Memory requirement is also M (N-1)/2.Memory requirement and end to end delay is one half over block interleaving. Block interleaving accept [2] the coded symbols in blocks from the encoder, permutes the symbols and then feed the rearranged symbols. [2][4]M rows and N columns are taken for transmission. Any burst of less than N contiguous channel symbol errors results in isolated errors at the deinterleaver output. Here the end to end delay is 2MN-2M+2 and memory requirement is MN. Instead of transmitting the variables in a regular manner, the structure is slightly modified as shown in Figure (4) to minimize the number of memory elements.



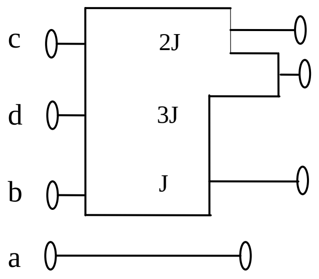
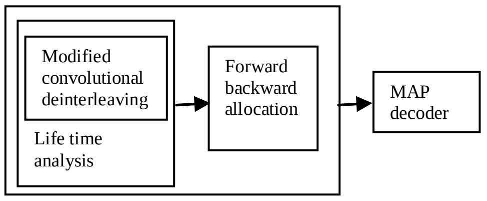


Figure 3 Original CI Figure 4 Modified CI

3. FOLDED MODIFIED CONVOLUTIONAL INTERLEAVING TECHNIQUE [FMCI]

Folding transformation provides a systematic technique for designing control circuits for hardware while several algorithm operations are time multiplexed on a single functional unit. In general CI technique needed M (N-1)/2 memory elements. To reduce the memory element, folded technique [10] is applied on it. Figure (5) shows how folded technique applied on convolutional interleaving. Register minimization technique is used to minimize the possible number of registers for designing DSP architecture. Lifetime analysis is a procedure used to compute the minimum number of registers required to implement an algorithm in hardware. A variable is live from the time it is produced through the time it is consumed, it is dead. Forward-backward register allocation is allocation schemes that can be used to allocate the data to the register. The steps used to perform FMCI are as follows


Modified convolutional deinterleaving
Life time analysis
Forward backward allocation
MAP decoder

Folded technique on FMCI

Figure 5 Proposed architecture

Step 1: Change the structure of CI to get reduced number of latches

Normally the number of latches used for CI is (N-1)J . Instead of sending the variable in a regular format as shown in Figure (3), the structure is slightly modified as in Figure (4).

Table 1 Forward backward allocation table

cycle	i/p	R1	R2	o/p
0	**c**			
1	**d**	c		
2	b	d	c	c
3	a	b	d	a,b
4		d		d

Step 2: Apply linear life time analysis technique
Let us consider four variables a, b, c, d. Variable 'c' be live during time units, n Є {1, 2},for 'd' n Є {2,3,4}, for 'b' n Є {3},no life time for variable 'a'. The minimum number of registers or latches required to implement this program is the maximum number of life variable at any time unit, which is max {1, 2, 2,1} = 2. {(N l + i)} where 'l' is any non negative integer and 'i' lies between 0 to (N-1). The time instances of this problem is 4l+3

Step 3: Apply forward-backward register allocation technique

i. determine the minimum number of registers using life time analysis technique. (R1,R2)
ii. allocate each variable in a forward manner until it is dead or it reaches the last variable. First 'c' allocates R1 register then the same variable 'c' holds by R2.
iii. For variables that reach the last register and are not yet dead, the remaining life period is calculated and allocated to a register in a backward manner on a FIFO basis. Here'd' first allocate to R1 and R2 register in a forward manner then because of non completion of life time it reallocate backward in R1.This is shown in table (1).

Step 4: Draw the folded architecture that corresponding to allocation table

The architecture for the 4x4 matrix transposer is shown in Figure (7). The backward allocation of'd' during cycle 3 from R2 to R1. All switching instances must be of the form 4l+m for $0 \leq m \leq 3$. The output of R1 is taken at the instance 4l+0, 3 and for 4l+2 output taken from R2.

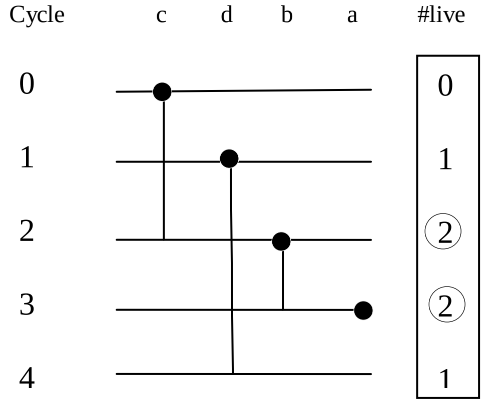


Figure 6 Linear life time chart

Here 'M' represents number of rows and 'N' represent number of columns .For eg when M = 4 and N = 4

Case: 1 For BI {a, b, c, d}, in this end to end delay is 26 and number of memory elements used are 16

Case : 2 For original CI {a, b, c, d}={0,1,2,3} ,in this end to end delay 12 and number of memory elements used are 6
Case :3 For FCI {a, b, c, d} ={0,1,2,3} ,in this end to end delay is 12 and number of memory elements used are 2

Case : 4 For FMCI {d, c, b, a} ={3,2,1,0} ,in this end to end delay is 6 and number of memory elements used are 3

Case :5 For FMCI {c, d, b, a} ={2,3,1,0} ,in this end to end delay is 8 and number of memory elements used are 2

From this case study, case 5 is fascinating for end to end delay and memory element usage.

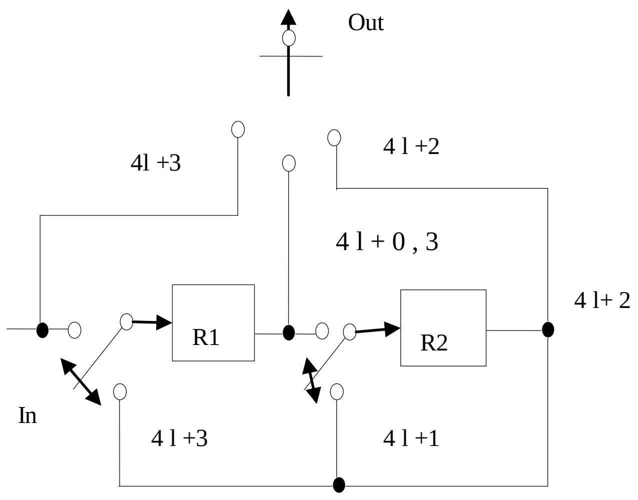


Figure 7 The architecture corresponding to the allocation

4. RESULTS AND DISCUSSION

By using FMCI technique, it is possible to minimize the number of latches usage and end to end delay at least 3 times lesser than CI .The reduction in delay and latches are achieved by folding and modified convolutional interleaving technique. The problem of this algorithm is, it is applicable only for when M =NJ. Block interleaving technique generally used in coding systems. Memory requirement and end to end delay is shown in table 2. In VLSI, register minimization takes important. With the help of convolutional interleaving technique these two parameters are minimized half.

Table 2 Comparison chart for BI, CI, FCI and FMCI

Parameter	BI	CI	FCI	**FMCI**
	(for all condition)	**(When M=NJ)**		
End to end delay	2MN-2M+2	M(N-1)	M(N-1)	**2M**
No of one bit latches	MN	M(N-1)/2	M-2	**M-2**

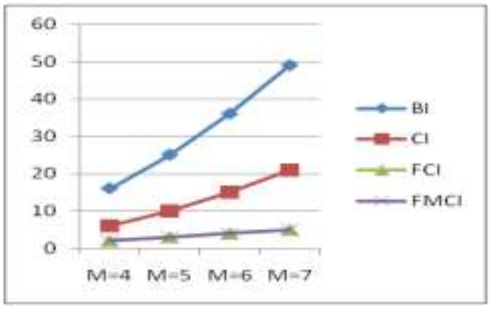


Figure : 10
Memory elements Vs number of row elements for different techniques

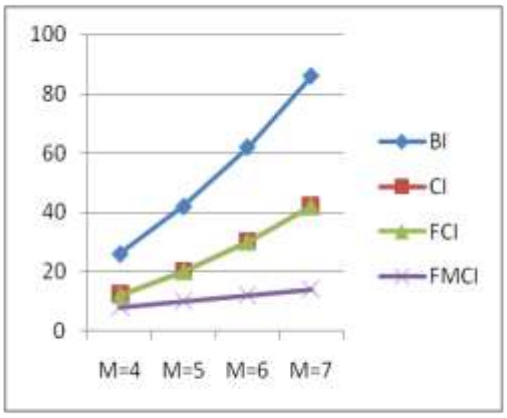


Figure : 11
End to end delay Vs number of row elements for different techniques

Register minimization is the vital criteria for any DSP algorithm. By adding interleaving the performance will be increased. Figure (10) shows the variations in number of latches according with number of rows for different techniques. In BI, for M=4, the number of latches used are 12, but the only advantage is, it can applied for any condition i.e. M ≠ N is also applicable. For the same condition, 50% latches can be saved by CI technique. Folded is a technique to minimize the number of latches usage. It is applied on CI, the memory usage is 2 i.e. three times lesser than CI.

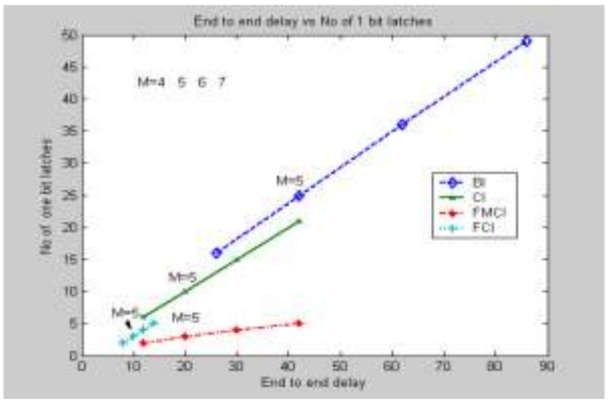


Figure12
End to end delay Vs number of one bit latches

Figure (11) shows the variation of end to end delay with number of row elements for different techniques. Here the performance of the proposed algorithm FMCI is more than the other techniques like BI, CI, and FCI. When M= 4 for BI the end to end delay (without propagation delay time) is 26, for CI it is reduced up to 54% i.e. 12, as compare with BI when M= N J (J=1). Both rows and columns are equal in nature, at that time CI is the apt technique for MAP coding as compare with BI. By applying folded technique on CI no change in end to end delay, the only change in number of latches usage only. We have to, slightly modified convolutional interleaving along with folded technique, save at least 69.2% time delay as compare with BI.
Figure(12) crisply gives the suggestion for choosing the best one for both the parameters end to end delay and number of latches. If M value increases means the end to end delay increases excessive. As compare with all the techniques FMCI is the best one for the above said parameters when M = NJ condition.

5. CONCLUSION

A folded (life time analysis and forward backward allocation) technique with MAP decoder has been introduced. The net result of all these techniques is almost an 88% decrease in memory elements. By applying folded technique and slightly alter the convolutional interleaving technique, area efficient is achieving and also end to end delay is reduced. For high rate, Turbo decoder is not efficient. MAP decoder performs well at high rates also. Future works needs to be directed towards reducing the critical path delay by introducing retiming, symbol based techniques. State metrics occupies more area. It is a promising area of research.

6. REFERENCES

[1] Bernard Sklar, (2003). Design Communication Fundamentals and applications, Person Education, 2003, 437 – 461

[2] Seok-Jun Lee,Naresh R.S Ahanbhag, Andrew C. Singer, (2005) .Area-Efficient High – Throughput MAP Decoder Architectures, IEEE trans on VLSI, Vol.13 No.8, 921 – 933, Aug2005

[3] Kazuhiko yamaguchi,Shinya Maehara, (2006).study on turbo decoding using hard decision decoder - principle of hard-in – soft-out decoding, International

Symposium on Information Theory and its Application,ISITA 2006,114-118.

[4] Sina Vafi,Tadeusz Wysocki, (2004). Modified convolutional Interleavers and their performance in turbo codes,computer and Electronics engineering faculty publications,symposium on trends in communications,2004, 54-57.

[5] Michacl Tuchler,Ralf Koetter,and Andrew C.Singer , (2002). Turbo Equalization : Principles and New Results,IEEE tran on communications,vol 50, No : 5,May 2002, 754-767

[6] Atluri,I., Arslan.T, "Low power VLSI Implementation of MAP decoder for turbo codes through forward recursive calculation of Reverse state metrics "IEEE

[7] Mustafa Taskaldiran,Richard C.S.Morling and lzzet kale, " The modified Max-Log_MAP turbo decoding Algorithm by Extrinsic Information scaling for wireless applications"

[8] Zhongfeng wang,Zhipei Chi and Keshab K.parthi, (2002). Area – Efficient High-speed decoding schemes for turbo decoders",IEEE Tran on VLSI,vol 10,no:6,Dec 2002, 902-912.

[9] Guido Masera,Marco Mazza,Gianluca Piccinni, (2002) .Architecture strategies for low-power VLSI turbo decoders IEEE Tran on VLSI ,vol 10,no:3,June 2002, 279-285.

[10] Keshab K, Parthi, "VLSI Digital Signal processing systems design and implementation", 149 – 164.

[11] C.Berrou.A.Glavieux,and P.Thitimajshima, (1993).Near Shannon limit error – correcting coding and decoding : Turbo codes ," in proc.IEEE nt.Conf.Communication,Geneva, Switzerland, May 1993, 1064 – 1070.

[12] Sina vafi and Tadeusz Wysocki, (2004).Modified Convolutional interleavers and their performance in Turbo Codes , Joint IST workshop on the Mobile Future and symposium on trends in communications,2004.SympoTic '04 ; 54 –57.