

Cheju, KOREA, 1-4, June 1999

Agenda Item:**Source: SAMSUNG Electronics Co.****Title: Proposal for an interleaver for Serially Concatenated
Convolutional Codes (Revised)****Document for: Discussion**

1. Introduction

SCCC has been considered for data service requiring BER 10^{-6} or less, as it is known to show no error floor typically shown in PCCC, a working assumption for data service requiring BER 10^{-6} or above. The characteristic of SCCC appears to stem from its inherent structure such that the component encoders of SCCC are serially concatenated with an internal interleaver among them, thus inner coder encodes outer code in stead of information itself. Nevertheless, the performance of SCCC still largely depends on internal interleaver, given fixed component encoders, just like PCCC (Turbo code), performance of which is more sensitive to the characteristics of its internal interleavers. Thus, we propose SCCC internal interleavers for 4-state SCCC encoder in this paper.

2. Background

Figure-1 shows the 4-state SCCC encoder. Two components recursive systematic convolutional coder (RSC) are concatenated serially and the inner encoder encodes outer code from outer coder. As the free distance of outer code is 5, weight of the input sequence to the inner encoder is at least 5, as the weight 5 error pattern should return to zero state trellis path for minimum free distance of inner coder. Therefore, code constrained free distance of inner code is more than 5. Therefore, unlike PCCC interleaver design, minimum distance consideration resulting from the weight-two information sequence is much relieved and more flexibility of design can be obtained. However, there exist some critical patterns that result in small free distance, Thus, in encoder's perspective, as in PCCC, one of the interleaver design criteria would be to find such critical error patterns and maximize free distance of the error patterns.

In decoder's perspective, one of the roles of SCCC interleaver is to scatter burst error patterns not corrected in one component decoder so that the following component decoder can correct. Another useful criterion would be the randomness of interleaver to maintain random code structure for each encoder. Non-random interleaver, for example, such as simple block interleaver is known to result in performance loss of more than 1 dB comparing with random interleaver.

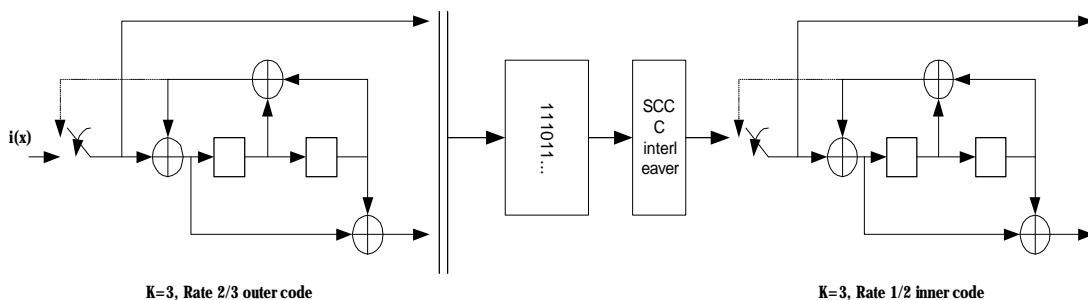


Figure-1 UMTS SCCC encoder structure

3. 2-D PN interleaver for SCCC

Previously mentioned two criteria are used to design non-optimized initial 2-D SCCC interleaver. Then, we optimize the interleaver by searching for optimal parameters that lead to a good free distance. Although there can be many different methods, the simplest one is to use PN sequence generator over $[R \times C]$ data array. Figure-2 depicts the general concept of 2-D PN interleaver, where $[R \times C]$ matrix are filled with data in left-to-right and top-to-bottom order. For each row, there is an interleaving rule $P(r, k)$.

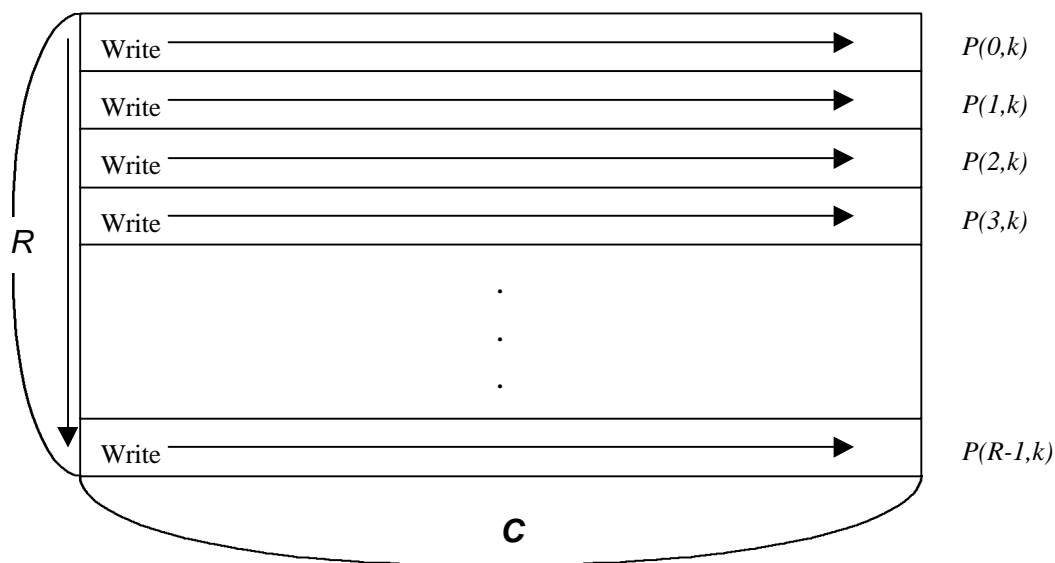


Figure-2 General idea of 2-D SCCC interleaver

Let us denote the size of information sequence to SCCC encoder in Figure-1 by L . Then, SCCC interleaver size $N = (L + 2) * 3 / 2$. Let us also denote $row_mux()$ by row permutation rule, $\%$ by mod operator, and $P(r, k)$ by an interleaving rule within row r . Thus, $R \times C \geq N$ and $R \times C - N$ invalid addresses are deleted. Interleaving is done as in the following procedure.

2-D SCCC interleaving Algorithm

1. Let $k = 0$
2. Choose a row $r = row_mux(k \% R)$
3. $Address = P(r, k) + r * C$
4. If $Address < N$, then output address $Address$
5. $k = k + 1$
6. If $k < R \times C$, go to 2
7. Stop

The above algorithm simply means that, at each clock k , the algorithm selects a row r first and then applies interleaving rule $P(r, k)$ to read/write a data within the row. For every R clocks, rows from 0 to $R-1$ is selected only once in the order $row_mux(k \% R)$. Address puncturing is necessary to eliminate invalid addresses just like PCCC interleavers, but no two consecutive addresses are punctured, which is also important to maintain constant clock speed in H/W implementation.

For 2-D PN interleavers, PN shifter registers (PNSR) with selected primitive polynomials are used to obtain intra-row interleaving rule $P(r, k)$. In doing that, there are several different implementation options according to the choice of **primitive polynomials, number of different PNSRs, R , C , and row multiplexing rule $row_mux()$** . For example, R for a given interleaver size N is one of the important design parameters that affect the performance of SCCC significantly. Generally, R should be chosen to be greater than free distance of code constrained inner code such that the SCCC interleaver decouples the output of the outer encoder from the input of the inner encoder. In addition, given R rows, one can use only one PNSR with one primitive polynomial, two PNSRs with two different primitive polynomials, or R PNSRs with R primitive polynomials. In the following example, we illustrate the principles of 2-D SCCC PN interleavers. We take only 2 PNSRs for simplicity of design and small H/W complexity.

Example-1

Let $L = 640$. Then, $N = (640 + 2) \times 3 / 2 = 963$. Let 2-D array be $[R \times C] = [16 \times 63]$. Suppose that we use two PNSR such that $P(r, k) = P1(k)$ for $r = 0, 1, 2, \dots, 7$ and $P(r, k) = P0(k)$ for $r = 8, \dots, 15$.

9, 10, ..., 15. The order of primitive polynomial $p0(x)$ and $p1(x)$ is 6 so that the period of $p0(x)$ and $p1(x)$ equal to $2^6-1 = C = 63$. $row_mux(a)$ is the bit reversal operation, namely, for $R = 16$, row order {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15} maps to new row selection order {0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15}.

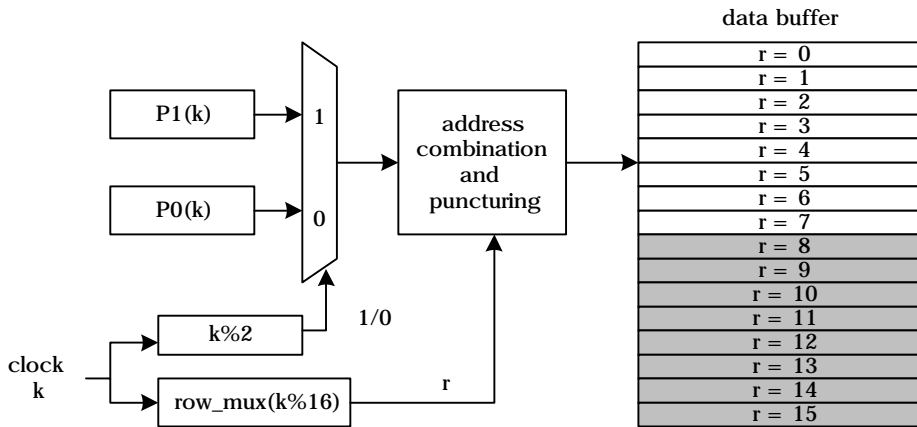


Figure-3 Example of 2-D SCCC PN interleavers with 2 PNSRs, $P0(k)$ and $P1(k)$

Figure-3 illustrates how to generate entire interleaving addresses with only two PNSRs $P0(k)$ and $P1(k)$, which generate two different random address patterns. Main idea of the Example-1 is that, with proper combination of $row_mux()$ function, number of PNSRs, and PNSR selection rule, one can obtain many different interleaving rules with very simple hardware complexity. The operation is as follows: At each clock k , $row_mux(k\%16)$ selects a row r alternatively. In the example r is the high 4 bits, as we have 16 rows in total. Then, the selected PNSR by $k \% 2$ selection rule outputs low 6 bits of data address to be generated. Finally, low 6 bits and high 4 bits are combined to generate a data address. If the generated address is invalid ($\geq N$), then it is punctured.

To generalize the 2-D PN interleaver structure shown in Figure-3, we first define some of the parameters as follows.

- $F =$ maximum number of different PNSRs, not necessarily equal to R
- $f =$ number of different PNSRs to be used in address generation ($\leq F$)
- $f0(k) =$ PNSR selecting function of clock counter (e.g. mode operation such as $k \% F$)
- $f1(k) =$ row multiplexing function of clock counter k (e.g. bit reversal of $k\%R$)

One can use f different PNSRs out of total F PNSRs hoping for more random property by using f different random sequence patterns for R rows. This will provide more design flexibility in terms of parameter optimization but with increased hardware complexity.

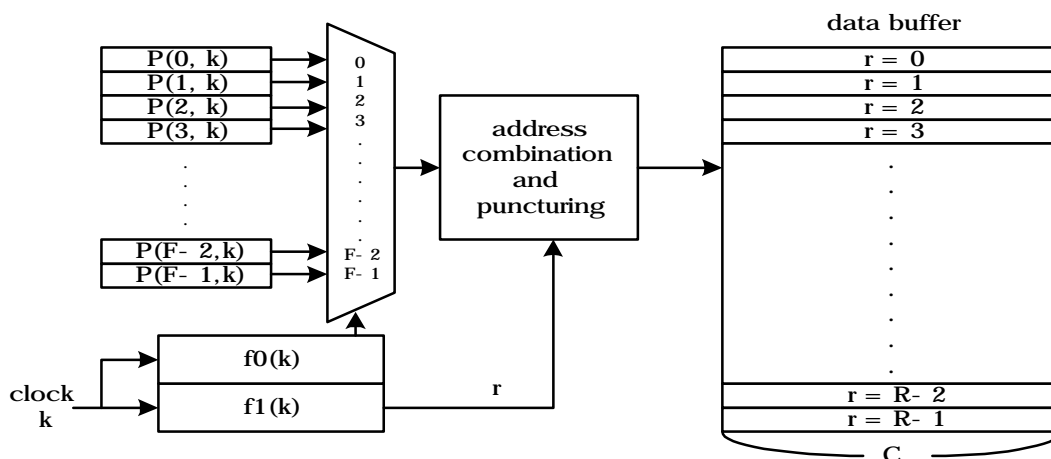


Figure-4 General 2-D SCCC PN interleaver structure

Figure-4 depicts the general structure of 2-D SCCC interleaver, where $P(i, k)$ for $i = 0, 1, 2, \dots, F-1$ generate F different random address patterns for the selected row $row_mux(k\%R)$. The operation is similar to that of Figure-3. The only difference now is the $f0(k)$ and $f1(k)$ is not just simple mod operation and bit-reversal operation, respectively. They can be arbitrary functions of k or some ROM tables.

4. H/W Complexity of 2-D PN Interleaver

Table-2 shows the H/W complexity of 2-D SCCC PN interleavers in terms of gate counts with assumption that we use $0.35\mu\text{m}$ technologies and 3.3V standard Samsung library, STD90. In addition, 6.18 gates per 1 flip/flop is assumed to estimate gate counts of PNSR and other part using flip/flops. For the functions $f0(k)$ and $f1(k)$, we use the same operations in Figure-3. Maximum interleaver size supported is assumed to be 8192.

Table-1 Estimated gate counts

Component	Estimated gate Counts (gates)	Detail	Comments
PNSRs	$52 \times F$	8 bit flip/flop* 6.18 * F	

<i>MUX_4x1</i>	12	Maximum 4 x 1 mux	Maximum F = 4
Address combination And pruning circuitry	200	14 bits SFR x 2 + control logics	
<i>F0(k)</i>	10	Clock divider, need 1 flip/flop	$k \% 2$ operation
<i>F1(k)</i>	87	6 bit flip/flop x 6.16 + 60 combinational logics	$k \% R$ operation, Maximum $R = 32 = 6$ bits
<i>Rom for parameters</i>	7 bitsx0.5gatesxF		<i>Primitive polynomials</i>
Total	309 + 55.5 x F		

- Note that the gate count includes pruning circuitry which, in our proposed interleaver, amounts to almost half of the total.

Therefore, our estimated gate counts of the 2-D PN SCCC interleaver amounts to **364.5 gates for F=1 to 531 gates for F = 4.**

Table-2 Comparison of H/W complexity with PIL

P I L	S A M S U N G	%	F
2 7 0 0	1 6 4 . 5	6 %	1
2 7 0 0	2 2 0	8 %	2
2 7 0 0	3 3 1	1 2 %	4

- Note that the gate count does not include pruning circuitry for fair comparison with PIL.

In Table-2, we compared our proposed SCCC interleaver with PIL in terms of H/W complexity I gate counts. As can be seen, the H/W complexity of the SCCC interleaver is negligible. For F=1 case, it is about 6% of that of PIL.

5. Simulation Results

In this section, we provide some preliminary simulation results of our proposed SCCC interleavers. The channel we use is a simple AWGN channel and we tested two different interleaver sizes, 963 and 7683 corresponding to information size 640 and 5120, respectively. Following is the summary of parameters of the interleaver sizes.

- AWGN channel
- Interleaver size 963[640], 7683[5120]
- Component decoder is SISO algorithm by Lucent
- SISO algorithm by Lucent
- Maximum 10 iterative decoding
- Code rate = $1/3 = 2/3_{outer} \times 1/2_{inner}$
- $[R \times C] = [16 \times 63]$ for 963 and $[R \times C] = [32 \times 255]$ for 7673
- Row multiplexing is bit-reversal order for both sizes.

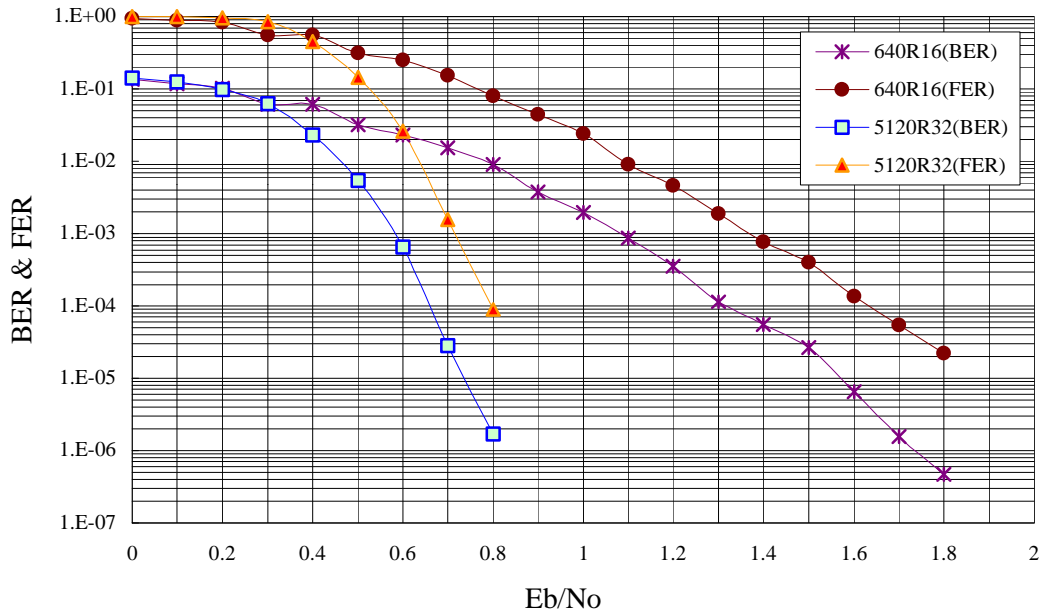


Figure-5 Simulation results of 2-D PN interleaver for SCCC over an AWGN

As can be seen in the Figure-5, BER= 1.7×10^{-6} is obtained at SNR=0.8 dB for information size 5120, and BER= 4.7×10^{-7} at SNR=1.8dB for information size 640. Those results are almost identical to the results reported in [1]. Currently, further simulations have been being taken place for BER down to 10^{-8} and for information sizes other than 640 and 5120. In addition, we are searching for optimal parameters to maximize free distance of a given interleaver and improve the performance.

6. Conclusions

In this paper, we presented a 2-D PN interleaving schemes using PNSR considering low hardware complexity and good performance. Following conclusions can be derived.

- *H/W complexity of 2-D PN interleaver for SCCC is as small as 364.5 gates for F=1 and 531 gates including address pruning circuitry for F=4, which is a design parameters to be optimized by trading-off system complexity and performances.*
- *Preliminary simulation results over an AWGN of our proposed interleaver size 963 and 7683 with F=2 are almost identical to the results reported in [1] by Lucent Technologies.*

- *For further performance improvement, we are now searching for optimal parameters such as R , row multiplexing, and F to obtain large free distance.*

7. Reference

[1] *A unifying code proposal for all data rates, block sizes and Quality of Service: Performance /Complexity trade-off by Lucent Technologies, TSGR1#3(99)189*

Contact info: bjkim@telecom.samsung.co.kr

Mgkim@telecom.samsung.co.kr

Yh1212@telecom.samsung.co.kr

