Design of (2, 1, N) Parallel Convolutional encodes for VLSI

Maoqiang Duan¹, a, Xiaoli Huang², b

¹ Lab of Industrial Control Network and System Shenyang Institute of Automation, Chinese Academy of Sciences, Liaoning Shenyang, P. R. China
² Competitive Intelligence Center Institute of Scientific & Technical Information of Liaoning Province, Liaoning Shenyang, P. R. China

a duanmaoqiang@sia.cn, b sophiaya@yeah.net

Keywords: convolutional encodes; parallel computing; low power dissipation

Abstract. The characters of more high speed computing and much less low power dissipation are needed to settle for convolutional encodes. In this paper, we present a parallel method for convolutional encodes with SMIC 0.35µm CMOS technology; hardware design and VLSI implementation of this algorithm are also presented. Use this method, parallel circuits structure can be easily designed, which take on excellent characters of more high speed computing and low power dissipation compared with traditional serial shift register structure for convolutional encodes.

Introduction

Convolutional codes, first introduced by P. Elias[1] in 1955, is one technique within the general class of channel codes. Because of the channel added noise, introduced bit errors and distorted signal, information sequence become unreliable over a channel. Those make convolutional codes be focused on by more and more researchers. In these period, convolutional codes is proved very good channel codes due to its outperforming block codes of the same order of complexity. Based on those, Claude Berrou and Alain Glavieux put forward Turbo-codes [2] together with their students in 1993. Turbo-codes encoder is completed using a parallel concatenation of two recursive systematic convolutional codes and the associated decoder, using a feedback decoding rule, whose performances in terms of bit error rate (BER) are close to the shannon limit. In recent years, BCH-convolutional codes [3], a class of convolutional codes using a new parity-check matrix, is presented, which have many characteristics of BCH block codes and excellent free distance. RS-convolutional codes[4], employing concatenated reed solomon codes and convolutional codes, which employs the information available from the channel state estimator to vary the amount of redundancy and its error correcting capability according the state of the channel. LDPC-convolutional codes [5], Low-density parity-check (LDPC) convolutional codes, are capable of achieving excellent performance with low encoding and decoding complexity. On the other hand, lots of researches are investigated on convolutional decodes. The algebraic decoders and probabilistic decoders have been at odds for a considerably longer period. The convolutional code dichotomy owes its origins to the development of sequential (probabilistic) decoding by Wozencraft [6] and of threshold (feedback, algebraic) decoding by Massey. And, Viterbi introduces other decoding technique, viterbi algorithm [7], in 1967, which is soon thereafter shown to yield maximum likelihood decisions.

Nowadays, convolutional codes have applied in many applications, including deep-space communications and voiceband modems. And continue to play an important role in low-latency applications such as speech transmission, wireless communication and so on. Furthermore, the characters of more high speed computing and much less low power dissipation are needed for convolutional codes. Although many papers focus on parallel cyclic redundancy check (CRC) algorithm [8], these are not completely adapted to the convolutional encodes. So in this paper, we pay attentions to implementation of parallel convolutional encodes, which is proved to be a available method to approach more high speed computing and much less low power dissipation. In section II, we review the theories of convolutional encodes, and given a traditional serial shift register structure of (2, 1, 5). Based on this, we derive a general parallel convolutional encodes formula in section III. In
section IV, power dissipation of convolutional codes (2, 1, N) with 8-bits degree of parallelism information sequence in serial structure and parallel structure respectively is discussed, where $3 \leq N \leq 9$. Meanwhile, power dissipation of convolutional codes (2, 1, 5) with different degree of parallelism information sequence is also discussed in this section. Each algorithm is implemented with 3.3V power supply using SMIC 0.35µm CMOS technology. Section V draws the conclusion.

Preliminaries

Convolutional codes are very suitable to protect the digital data transmission from random errors due to any noise source. It achieves error free transmission by adding sufficient redundancy to the source symbols. Convolutional codes are usually described using two parameters: the code rate and the constraint length. The code rate is expressed as a ratio of number of input symbols ($k$) into the channel encoder to the number of output symbols ($n$) by the channel encoder in a given cycle. Then, the code rate is expressed as

$$R = \frac{k}{n} \text{ bits / symbol} \quad (1)$$

The constraint length $v=m+1$ denotes the length of the convolutional encoder, $m$ is depth of registers. Convolutional encoder increases the length of the message sequence by adding redundant bits in order to increase the likelihood of detecting the transmitted sequence even if errors have occurred during transmission. A $(n, k, v)$ convolutional code can be described by applying matrix method; polynomial method; shift register method and so on.

A rate $1/n$ convolutional encoder is characterized by $g$ generator sequences [9]

$$g^{(j)} = (g_0^{(j)}, g_1^{(j)}, \cdots, g_v^{(j)}) \quad j = 1, 2, \cdots, n$$

Let the information sequence is $u = u_{i-1}u_i u_{i+1} \cdots$ and $u_i = (u_i^1, u_i^2, \cdots, u_i^k)$. Let the $n$ encoded output sequences be

$$v(j) = (v_0^{(j)}, v_1^{(j)}, \cdots, v_n^{(j)}) \quad j = 1, 2, \cdots, n$$

corresponding encoding of the information sequence is given by $v = u G$.

$$v = (v_0^{(1)}, v_0^{(2)} \cdots v_0^{(n)}, v_1^{(1)}, v_1^{(2)} \cdots v_1^{(n)}, \cdots)$$

Where $v$ is the composite encoded sequence, also called a codeword, obtained by multiplexing the $n$ encoded sequences and $G$ is the semi-infinite generator matrix

$$G = \begin{pmatrix}
G_0 & G_1 & G_2 & \cdots & G_v \\
G_0 & G_1 & \cdots & G_{v-1} & G_v \\
G_0 & \cdots & G_{v-2} & G_{v-1} & G_v \\
\vdots & \ddots & \ddots & \ddots & \ddots
\end{pmatrix} \quad (2)$$

Where blanks indicate zeros and $G_l = [g_l^{(1)} g_l^{(2)} \cdots g_l^{(n)}]$.

Also, for example, convolutional codes (2, 1, 5) can be described by shift register in fig.1.
Figure 1. Serial register structure of (2, 1, 5) convolutional encodes

Consider a binary \( R=1/2 \) convolutional encodes, \( n=2, k=1, m=4 \). The input to the encodes at time \( k \) is a bit \( u_k \) and the corresponding binary couple \( (v_{1k}, v_{2k}) \), is equal to

\[
v_{1k} = \sum_{i=0}^{m} g_{1i} u_{k-i} \quad g_{1i} = 0,1 \tag{3a}\]

\[
v_{2k} = \sum_{i=0}^{m} g_{2i} u_{k-i} \quad g_{2i} = 0,1 \tag{3b}\]

\( G_{1i}: \{g_{1i}\}=(23)_{oct} \) \( G_{2i}: \{g_{2i}\}=(35)_{oct} \) are the two encodes generators.

**Parallel convolutional encodes**

Papers present lots of research for computation and implementation of cyclic redundancy check (CRC) parallel algorithm [9]. But the method of those is not appropriate to convolutional encodes. We are going to derive the formula of parallel convolutional encodes, using \( (2, 1, 5) \) with 8-bits degree of parallelism information sequence.

As shown in figure 1, encoded sequence \( v \) is determined by information sequence \( u \); the value of shift registers \( D \) and generator \( G \) and can be expressed as:

\[
v_{1i} = \left[ D' u_i \right]^{T} G_1 \tag{4a}\]

\[
v_{2i} = \left[ D' u_i \right]^{T} G_2 \tag{4b}\]

Where \( D'=[D_{m-1}' D_{m-2}' \cdots D_0'] \), \( i \) and \( m \) are \( ith \) the times of series input and depth of memory, respectively. Register state value will changed by shift right, when new \( u_i \) input. So, Eq. (4), the key point is how to solve \( D' \) state value.

As \( u_0 \) input, the register state value \( D' \) can be expressed as \( D' = [D_3' D_2' D_1' D_0']^T \)

\[
\begin{bmatrix}
D_3' \\
D_2' \\
D_1' \\
D_0'
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
D_3^0 \\
D_2^0 \\
D_1^0 \\
D_0^0
\end{bmatrix} + u_0 \tag{5}
\]

Let
So,

\[ D^1 = FD^0 + Bu_0 \]  \hspace{1cm} (6)

As \( u_i \) input, the register state value \( D^2 \) can be expressed as,

\[ D^2 = FD^1 + Bu_1 \]  \hspace{1cm} (7)

Bring Eq. (6) into (7)

\[ D^2 = F^2D^0 + FBu_0 + Bu_1 \]  \hspace{1cm} (8)

Step by step, the register state value \( D^8 \) can be described as,

\[ D^8 = F^8D^0 + F^7Bu_0 + F^6Bu_1 + \cdots + FBu_6 + Bu_7 \]  \hspace{1cm} (9)

Generally considered, when \( u_{i-1} \) input,

\[ D^i = F^iD^0 + F^{i-1}Bu_0 + F^{i-2}Bu_1 + \cdots + FBu_{i-2} + Bu_{i-1} \]

So, the register state value can be finally expressed:

\[ D^i = F^iD^0 + \sum_{j=0}^{i-1} F^jBu_{i-j} \]  \hspace{1cm} (10)

In table 1, \( \oplus \) is EXOR operation. Therefore, a parallel circuit structure is achieved for \( (2, 1, 5) \) convolutional encodes. Encodes sequence can be expressed using EXOR between 8-bits degree of parallelism information sequence and initial value of register, then the register value will be replaced by MSB information sequence, before new parallel information input.

<table>
<thead>
<tr>
<th>Table 1. Parallel circuit expression for encodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{v}_1 )</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>( v^0_1 )</td>
</tr>
<tr>
<td>( v^1_1 )</td>
</tr>
<tr>
<td>( v^2_1 )</td>
</tr>
<tr>
<td>( v^3_1 )</td>
</tr>
<tr>
<td>( v^4_1 )</td>
</tr>
<tr>
<td>( v^5_1 )</td>
</tr>
<tr>
<td>( v^6_1 )</td>
</tr>
<tr>
<td>( v^7_1 )</td>
</tr>
</tbody>
</table>

**Discussion**

In section III, we have already derived formulas of the parallel convolutional encodes, and parallel circuit structures can be easily to get according to Eq. (4) and (10). In this section, we will discuss power dissipation characters of traditional serial shift register structure and parallel structure under different constraint length (N), respectively. We have implemented convolutional encodes \( (2, 1, N) \) using hardware for serial structure and parallel structure, where \( 3 \leq N \leq 9 \).

Meanwhile, power dissipation of convolutional codes \( (2, 1, 5) \) with different degree of parallelism information sequence is also discussed in this section.
The main work is listed:
- Described circuit structures with Verilog HDL
- Stimulation functional with Synopsys VCS
- Synthesis gates with Synopsys DC
- Analysis power dissipation with Synopsys PTPX

Implementation conditional:
- Each algorithms is implemented with 3.3\,V power supply using SMIC 0.35\,\mu m CMOS technology
- All of the algorithms have the same encode radio
- There is 8-bits degree of parallelsim information sequence for parallel structure with different constraint length in figure 2 and figure 4, 3.125\,MHz work frequency for parallel structure and 25\,MHz work frequency for serial structure
- With (2,1,5) convolutional encode, different degree of parallelism information sequence for parallel structure are shown in figure 5 and figure 6

As Fig. 2 shown, power dissipation for parallel structure is slightly enhanced, whereas power dissipation for serial structure is lineal enhanced. And the distance between them increased with adding constraint length. When constraint length is 9, the value of power dissipation of parallel structure is 20 percent less than that of serial structure (Fig. 3). Fig. 4 shows power dissipation for parallel structure with different degree of parallelism information sequence. We can see that the power dissipation sharply decreased with width parallel input for parallel structure.

![Figure 2. Power dissipation: serial vs. parallel](image1)

![Figure 3. Power saving percent: compare parallel to serial](image2)

![Figure 4. Power dissipation for parallel structure with different degree of parallelsim information sequence](image3)

**Conclusions**

In this paper, we review traditional theories of convolutional encodes and derive a parallel formula. We have implemented all algorithms in VLSI with SMIC 0.35\,\mu m CMOS technology, and discussed power dissipation with different constraint length between parallel structure and serial structure. To the character of low power dissipation, parallel structure is better than serial structure. The value of
power dissipation of parallel structure is 20 percent less than that of serial structure, when constraint length is 9. If less-considered resource consumption, we can achieve higher error-correction ability applied long constraint length convolution codes, with less power dissipation or faster computing speed. If more low power dissipation is needed, increasing degree of parallelism information sequence is performance method.

Acknowledgements
This work was financially supported by the National High Technology Research and Development Program of China (863 Program) under grant 2011AA040102.

References
Design of (2, 1, N) Parallel Convolutional Encodes for VLSI
10.4028/www.scientific.net/AMM.321-324.2822