A 1 GS/s 12-bit pipelined folding ADC with a novel encoding algorithm

Huasen Liu¹,², Danyu Wu², Lei Zhou², Jian Luan², Xuan Guo², Dong Wang¹,², Jin Wu², and Xinyu Liu¹

¹ Institute of Microelectronics of Chinese Academy of Sciences, Beijing 10029, China
² University of Chinese Academy of Sciences, Beijing 10029, China

a) xyliu@ime.ac.cn

Abstract: In this paper, a 1-GS/s 12-bit pipelined folding analog-to-digital converter (ADC) fabricated in 40 nm CMOS technology is presented. A new encoding algorithm based on distributed quantization is proposed to simplify the quantization process of the structure with odd folding factor and reduce the hardware consumption of the circuit. The ADC achieves spurious free dynamic range (SFDR) > 72 dB and signal-to-noise and distortion ratio (SNDR) > 57 dB in low input frequencies.

Keywords: analog-to-digital converter, folding and interpolating, odd folding factor, encoding algorithm

Classification: Integrated circuits

References

[8] J. Han, et al.: “A 10-b 500 MS/s CMOS cascaded folding A/D converter with


1 Introduction

In recent years, the study of analog-to-digital converters (ADC) with resolutions higher than 8 b and sample rates over gigahertz has enabled the use of software-defined radios to be applied in more applications [1, 2]. In high-speed applications, the flash ADCs enjoy exceptional advantages. However, the number of comparators will increase exponentially as the resolution increases, resulting in a sharp increase in the area and power consumption of the chip. In this condition, the folding-interpolating ADCs are continuously studied to solve these problems. The folding-interpolating ADCs reduce the number of comparators by introducing folding amplifiers to fold the quantization interval, and reduce the number of folding amplifiers by interpolation technology. The folding-interpolating ADCs can reach a high speed comparable to the flash ADCs for the reason that the entire quantization process does not have any feedback loop.

To the folding-interpolating ADCs, the implementation of high resolution requires high folding order to fold the quantized interval into more small intervals. Hence, multi-stage cascaded structures are commonly adopted to realize high resolution [3, 4, 5], because excessive folding factor in one stage will lead to insufficient voltage margin of the folding amplifiers and the reduction of bandwidth. In this paper, an ADC satisfying 12-bit resolution and 1-GS/s conversion speed is proposed, which adopts a folding-pipelined structure with the folding factor of 3. Due to the use of odd folding factor, a problem encountered in encoding is that the comparators’ outputs corresponding to the folding quantization curves cannot be directly converted into the form of $2^n$. A complicated digital coding unit is usually required [5]. A novel encoding algorithm is proposed to simplify the encoding process and reduce the consumption of the circuit, which is also able to be used in the structures with other odd folding factors.

2 Proposed ADC architecture

The block diagram of the proposed ADC is shown in Fig. 1. It consists of an input
A multiplexer (MUX), an input buffer, a track-and-hold circuit (THA), a pre-amp stage (stage 0), and six folding-interpolating quantizer stages (stage 1–6). The pre-amp array consisted of 27 pre-amplifiers at stage 0 amplifies the difference between the input signal and the 27 reference voltages separately to obtain 27 small quantization intervals. Then folded by six stages with folding factor of 3 and interpolating factor of 3, the entire quantization interval is divided into $3^6$ intervals. Nine comparators are set at stage 6 to compare the minimum intervals, resulting in $6561 \left(\frac{3^6 \times 9}{9}\right)$ zero-crossings to meet 12-bit resolution requirement. Due to the influence of the boundary effect, the zero-crossings at both ends of the quantization interval will be unevenly distributed, so the middle 4096 zero-crossings are taken as the actual quantization interval. Two comparators of stage 0 and fifteen comparators of stage 1 to stage 5 are used for the distributed quantization. All the outputs of the comparators are sent into the encoder module to complete the 12-bit data conversion, which is detailed in section 3.

In addition, the folding ADCs are very sensitive to device mismatch due to the open-loop nature of the folding amplifier, especially when the process size is gradually reduced. Therefore, the ADC in this paper is calibrated with its non-linearity by a foreground self-calibration method in order to achieve the required accuracy. The input mux is used for strobing calibration mode and normal working mode. When entering the calibration mode, the input mux will turn on the cal-vector module which is implemented by a digital-to-analog converter (DAC) with 14-bit precision. The cal-vector will sequentially send the ideal input voltages corresponding to the zero-crossings which are desired to be calibrated under the control of the digital calibration logic. Then the outputs of the comparators are sent back to the digital logic to be calculated and accumulated. The feedback calibration signal is given by the DAC array consisted of 27 current DACs in each stage. In this paper, the zero-crossings from stage 1 to stage 4 are calibrated. In order to achieve high precision, the 14-bit high precision DAC adopts the structure of current-steering.
The layout design of the current source arrays uses the algorithm of $Q^2$ random work to randomly arrange the transistors of the current source, reducing the effects of random errors and gradient errors effectively [6]. To ensure that the output range of the cal-vector matches the quantization range of the ADC, a negative feedback should be added between the reference ladder and the 14-bit DAC.

3 The proposed encoding algorithm

In the conventional folding-interpolating structure, the quantization method of coarse quantization combined with fine quantization is commonly adopted [7, 8]. However, the addition of the coarse quantization unit which is usually implemented by a flash ADC, will increase the load of THA. In addition, due to the inconsistent data delay between the coarse quantization path and the fine quantization path, it is necessary to solve the synchronization problem between the least significant bits (LSBs) and the most significant bits (MSBs) [9]. This paper proposed an encoding algorithm based on distributed quantization, which can save a coarse quantizer to increase the input bandwidth. Moreover, it can be applied to other odd folding factor structures and solve the problem of encoding complexity and synchronization effectively. To facilitate understanding, the algorithm is derived step by step under the condition that folding factor is 3.

In this structure, the 27 output folding-interpolating curves of each stage are labeled from 0 to 26 respectively. Stage 0 uses two comparators ($C_{1_{st0}}$, $C_{2_{st0}}$) to compare the 8th and 17th curves of its outputs, dividing the entire quantified interval into three parts; stage 1–5 add an additional comparator ($C_{3_{stN}}$, $N = 1, 2, 3, 4, 5$) connected to the 26th curve respectively to obtain redundant information for coding error correction [5]; stage 6 uses nine comparators ($C_{1_{st6}}$–$C_{9_{st6}}$) to compare the 2nd, 5th, 8th, 11th, 14th, 17th, 23th, and 26th output curves in order to meet the requirements of 12-bit resolution. The differential input curves of the comparators from stage 0 to stage 2 are shown in Fig. 2. The comparators of stage 0 have 3 output states from 0 to 2, the comparators of stage 1 have 9 output states from 0 to 8 and the comparators of stage 2 have 27 output states from 0 to 26. Combining the outputs of all the comparators of the three stages, the corresponding value from 0 to 26 can be obtained by using an encoder built by logic gate circuits. However, if the quantization process is performed in this way, it is finally necessary to logically combine 26 comparators to obtain 6561 output states of 0 to 6560, which is too complicated to build the logic coding circuit.

In order to simplify the quantization process, another method based on weight factor is adopted. As shown in Fig. 2, each folding quantified interval is divided into 3 small quantified intervals in the next stage since the folding factor of each folding stage is 3. And because the number of comparators from stage 1 to stage 5 is also 3, the output state of stage $N-1$ ($N = 1, 2, 3, 4, 5$) changes once when that of stage $N$ changes three times. Hence, each adjacent two stages can be regarded as a ternary relation. In this case, a quantization parameter $i$ of which all the possible values are 0, 1, 2 is introduced. For the convenience of the logic calculation, a two-bit binary number $(A_1, A_0)$ is used to represent $i$. For Stage 0, the logical expressions corresponding to the values of $i$ is as follows:
As shown in Table I, the $i$ of stage 0 can be expressed as:

$$A_{1}^{0} = C_{1}^{0} / C_{2}^{0} \left( 1 \right)$$

$$A_{0}^{0} = C_{2}^{0} \left( 2 \right)$$

For stage 1 to stage 5, it can be seen from Fig. 2 that there will be a 6-state digital code loop since the outputs of the comparators adopt the output method of the thermometer cyclic code. The specific correspondence is shown in Table II:

Therefore, each value of $i$ needs to correspond to two output states in order to achieve ternary. State 1 and state 4 are treated as 0, state 2 and state 5 are treated as 1, state 3 and state 6 are treated as 2. Corresponding logical expressions of $i$ are obtained in Table III:

Then the $i$ of stage 1–5 can be expressed as:

$$A_{1}^{N} = (C_{1}^{N} \oplus C_{2}^{N}) \cdot (C_{1}^{N} \oplus C_{3}^{N}), \quad (3)$$

$$A_{0}^{N} = (C_{1}^{N} \oplus C_{2}^{N}) \cdot (C_{1}^{N} \oplus C_{3}^{N}), \quad (4)$$
The digital encoding unit is realized by simple digital logical circuit, which is shown in Fig. 3.

Since nine comparators are used in stage 6, the output state of stage 5 changes by one time when the output state of stage 6 changes nine times. The relation between stage 5 and stage 6 is novenary. The same way is employed to get the values of $i$ from 0 to 8 in stage 6, which is shown in Table IV:

The $i$ of stage 6 needs to be represented by a 4-bit binary number $(A_3 A_2 A_1 A_0)$. As shown in Table II, the encoding of $i$ is given by

<table>
<thead>
<tr>
<th>Stage N (C3 C2 C1)</th>
<th>Stage N-1 (C3 C2 C1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>State1 $\rightarrow$ 3</td>
<td>000 $\rightarrow$ 001 $\rightarrow$ 011</td>
</tr>
<tr>
<td>State4 $\rightarrow$ 6</td>
<td>111 $\rightarrow$ 110 $\rightarrow$ 100</td>
</tr>
<tr>
<td>State1 $\rightarrow$ 3</td>
<td>000 $\rightarrow$ 001 $\rightarrow$ 011</td>
</tr>
<tr>
<td>State4 $\rightarrow$ 6</td>
<td>111 $\rightarrow$ 110 $\rightarrow$ 100</td>
</tr>
<tr>
<td>State1 $\rightarrow$ 3</td>
<td>000 $\rightarrow$ 001 $\rightarrow$ 011</td>
</tr>
<tr>
<td>State4 $\rightarrow$ 6</td>
<td>111 $\rightarrow$ 110 $\rightarrow$ 100</td>
</tr>
</tbody>
</table>

### Table III. Logical expression of $i$ in stage 1–5

<table>
<thead>
<tr>
<th>expression</th>
<th>$i$ (binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{1_{N}} \cdot C_{2_{N}} \cdot C_{3_{N}} + C_{1_{N}} \cdot C_{2_{N}} \cdot C_{3_{N}}$</td>
<td>00</td>
</tr>
<tr>
<td>$(C_{1_{N}} \oplus C_{2_{N}}) \cdot (C_{1_{N}} \oplus C_{3_{N}})$</td>
<td>01</td>
</tr>
<tr>
<td>$(C_{1_{N}} \oplus C_{2_{N}}) \cdot (C_{1_{N}} \oplus C_{3_{N}})$</td>
<td>10</td>
</tr>
</tbody>
</table>

### Table IV. Logical expression of $i$ in stage 6

<table>
<thead>
<tr>
<th>expression</th>
<th>$i$ (binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1 \oplus C_2$</td>
<td>0000</td>
</tr>
<tr>
<td>$C_2 \oplus C_3$</td>
<td>0001</td>
</tr>
<tr>
<td>$C_3 \oplus C_4$</td>
<td>0010</td>
</tr>
<tr>
<td>$C_4 \oplus C_5$</td>
<td>0011</td>
</tr>
<tr>
<td>$C_5 \oplus C_6$</td>
<td>0100</td>
</tr>
<tr>
<td>$C_6 \oplus C_7$</td>
<td>0101</td>
</tr>
<tr>
<td>$C_7 \oplus C_8$</td>
<td>0110</td>
</tr>
<tr>
<td>$C_8 \oplus C_9$</td>
<td>0111</td>
</tr>
<tr>
<td>$C_9 \oplus C_1$</td>
<td>1000</td>
</tr>
</tbody>
</table>

Fig. 3. The digital encoding unit of stage 1–5.
\[ A_{3\text{st}6} = C9_6 \odot C1_6, \]  
\[ A_{2\text{nd}6} = C8_6 \oplus C9_6 + C7_6 \oplus C8_6 + C6_6 \oplus C7_6 + C5_6 \oplus C6_6, \]  
\[ A_{1\text{st}6} = C8_6 \oplus C9_6 + C7_6 \oplus C8_6 + C4_6 \oplus C5_6 + C3_6 \oplus C4_6, \]  
\[ A_{0\text{th}6} = C8_6 \oplus C9_6 + C6_6 \oplus C7_6 + C4_6 \oplus C5_6 + C2_6 \oplus C3_6. \]  

In summary, the values of \( i \) from stage 0–6 are obtained respectively by using 7 groups of digital encoders with simple logic. In the case that the weight of the last stage is determined to be 1, the weight of each stage can be expressed by \( 3^n \) (\( n = 7, 6, 5, 4, 3, 2, 0 \)). As shown in Fig. 4(a), the final quantized value can be obtained mathematically by adding the product of \( i \) and the corresponding weight in each stage. However, the implementation of large numbers of multipliers in the circuit is too complicated, so the calculation method adopted is shown in Fig. 4(b). The operation of multiplying the weight is resolved into the form of multiplying 3 step by step, taking advantage of the pipelined nature of the architecture. In binary coding, multiplying 3 can be converted to the operation that shifting the digital code one bit to the left and adding it to itself. In this condition, the entire quantization process can be done with only adders and the adders used in this algorithm are much simpler than that used for parallel processing shown in Fig. 4(a). To the other odd folding factors which are less than 10, they can use the similar shift operation to simplify the calculation. And data synchronization can be achieved by just adding different number of latches in each stage since the digital encoders are pipelined.

![Fig. 4.](image-url) (a) The mathematical process of encoding. (b) The simplified process of encoding.
4 Measurement result

The proposed ADC was fabricated in 40 nm CMOS process with a core chip area 1.5 × 0.75 mm². The core ADC consumes 250 mW from the supplies of 1.8 V, 1.3 V and 0.9 V. The die micrograph is shown in Fig. 5. As shown in Fig. 6, the measured static performance shows that the DNL is within +0.7/−0.4 LSB and the INL is within +2.65/−2.65 LSB. The measured output spectrum is shown in Fig. 7(a) when a 405.1 MHz input sine wave is sampled at 1 GS/s. Its signal-to-noise and distortion ratio (SNDR) is 55.44 dB and spurious-free-dynamic-range (SFDR) is 64.06 dB. Fig. 7(b) presents the measured ENOB and SFDR versus input frequency at a sampling rate of 1 GHz, SFDR achieves 72 dB and SNDR achieves 57 dB at low input frequency. In the full range of the first Nyquist zone, SFDR and SNDR stay above 62 dB and 54 dB. The standard figure-of-merits (FOM) is 614 fJ/conversion-step (FoM = P/(2^ENOBNyquist × Sample Rate)). Table V summarizes the measured performance and compares it against other ADCs.

Fig. 5. The die micrograph.

Fig. 6. DNL and INL.
Conclusion

In this paper, a 1-GS/s 12-bit ADC which adopts a novel encoding algorithm has been demonstrated. The ADC fabricated in 40 nm CMOS technology has a measured SFDR above 62 dB and SNDR above 54 dB over the full first Nyquist band. The core ADC consumes 250 mW of power and the whole chip has a die size of $1.5 \times 0.75 \text{ mm}^2$.

Acknowledgments

This work was supported by National Science and Technology Major Project (Grant No. 2016ZX03001002). The authors would like to express their thanks to the members of Microwave Device&IC Department for their guidance and assistance.