Short Papers

A Programmable Nonlinear ADC Using Optimal-Sized ROM

Dinesh K. Anvekar and B. S. Sonde

Abstract—A new programmable successive approximation ADC useful for realizing nonlinear transfer characteristics often required in instrumentation and communications is presented. This nonlinear ADC (NADC) requires a much smaller sized ROM than an NADC reported earlier.

I. INTRODUCTION

Nonlinear ADCs (NADCs) find wide application in instrumentation and communications. In the former case, they are useful for the linearization of transducer output, while in the latter case, nonlinear A/D conversion is needed for companding. In both these application areas, it is desirable that the NADC characteristic be easily programmable. With the programmability feature built in, the same NADC may be used with a number of different transducers by just programming its characteristic rather than by redesigning a separate NADC for each transducer. This is especially relevant in the development of integrated sensors which have the transducer, signal conditioning circuits, and the NADC all in the same IC package. In voice communication using PCM, either of the required compression characteristics, viz., A-law or μ -law, may be selected if the NADC is programmable. A straightforward approach to a programmable NADC is to use a linear successive approximation ADC followed by a suitable ROM look-up table. While this technique is simple and flexible, its main drawback is that the ROM size doubles with every bit increase in the output code of the linear ADC. An alternative approach is to integrate the ROM with the A/D conversion process so that the nonlinear code is produced directly. This leads to a considerable reduction in the ROM size, especially in applications involving code compression. In such an NADC, the successive approximation register (SAR) is truly not needed; moreover, its presence can increase the conversion time of the NADC. Therefore, it is often desirable to avoid the SAR in the NADC to speed up A/D conversion. In an NADC of this type presented in [1], no attempt seems to have been made to optimize the ROM size. Taking this factor into consideration, an improved NADC [2] which requires a much smaller sized ROM has been developed as described below:

II. IMPROVED NADC USING OPTIMAL-SIZED ROM

Fig. 1 gives the basic configuration for an NADC using a SAR and a ROM. Here, the SAR output is used as the address for the $2^n \times m$ bit ROM which is programmed to contain the m -bit nonlinear codes corresponding to an n -bit linear scale. This enables the input voltage to be compared with the voltage levels corresponding to the nonlinear scale while the SAR output is updated as in the

Fig. 1. NADC using SAR and ROM.

linear ADC. With $m > n$, nonlinear code compression is achieved with m -bit precision in certain portions of the n -bit linear scale at the cost of lower precision at other parts of the scale.

For the purpose of illustration, consider the example of a nonlinear characteristic with $n = 4$ and $m = 5$, as shown in Table I. Here, a 5-b code is mapped to 4-b by having nonuniform step size for the nonlinear scale. As this mapping is generally one-to-one, it is possible to use the current code-approximation in conjunction with the comparator output in Fig. 1 as the pointer to the stored value of the next code-approximation in the ROM. This avoids the SAR in the NADC [1] as shown in Fig. 2. However, an $(m + 1)$ -bit register is now required to hold the ROM address. The memory layout for the ROM in Fig. 2 for the nonlinear characteristic of Table I is shown in Table II, from which it is seen that nearly half the number of memory locations is unused, indicating that the ROM is not optimally used. A new technique for the optimization of ROM size in the NADC is now discussed.

In the NADC of Fig. 2, after each successive approximation step, the comparator output gives a bit result of the n -bit linear scale to which the m -bit nonlinear codes are mapped, and after n steps, the ROM output gives the m -bit nonlinear code corresponding to input V_i . Therefore, as a first step towards ROM size optimization, the number of bits of ROM address can be reduced from $(m + 1)$ bits to $(n + 1)$ bits as shown in Fig. 3. Here, the ROM is logically divided into two blocks: -a) block I to provide the n -bit code of the linear scale (OUTPUT1N) which is used with the comparator out-

Manuscript received April 6, 1991.

The authors are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India.
IEEE Log Number 9104905.

Fig. 3. NADC with reduced number of ROM address bits.

TABLE I
NONLINEAR CHARACTERISTIC WITH $m = 5$ AND $n = 4$

Linear Code:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Nonlinear Code:	0	4	8	12	16	20	22	23	24	25	26	27	28	29	30	31

TABLE II
MEMORY DATA LAYOUT FOR NONLINEAR CHARACTERISTIC OF TABLE I

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Data	24	—	—	—	—	—	—	—	0	4	—	—	—	—	—	—
Address	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Data	4	12	—	—	—	—	—	—	8	12	—	—	—	—	—	—
Address	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Data	8	22	—	—	—	—	—	—	16	20	—	—	20	23	22	23
Address	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Data	16	28	24	25	26	27	26	27	26	30	28	29	29	31	30	31

— unused.

TABLE III
MEMORY DATA LAYOUT FOR ROM OF FIG. 3 WITH $m = 5$ AND $n = 4$

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OUTPUT1N	8	—	0	1	1	3	2	3	2	6	4	5	5	7	6	7
OUTPUT2	24	—	0	4	4	12	8	12	8	22	16	20	20	23	22	23
Address	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
OUTPUT1N	4	12	8	9	9	11	10	11	10	14	12	13	13	15	14	15
OUTPUT2	16	28	24	25	25	27	26	27	26	30	28	29	29	31	30	31

— unused.

put as the address of the next successive approximation code, and b) block II to provide the m -bit nonlinear code corresponding to OUTPUT1N as the input to the DAC. The memory data layout for the ROM in this NADC for realizing the nonlinear characteristic of Table I is given in Table III. The ROM size reduction (r) achieved in this modified NADC as compared to that of Fig. 2 is $\{1 - (m + n)/m \cdot 2^{m-n}\} 100\%$, which for the example nonlinear characteristic is 10%. The ROM size can be further optimized by considering the fact that the LSB of OUTPUT1N in Fig. 3 is 0 for the first $(n - 1)$ steps, and is the same as the comparator output after the n th step. Therefore, only the most significant $(n - 1)$ bits of OUTPUT1N along with the comparator output in the LSB position, can be used as the pointer to the next approximation stored in the ROM, and the LSB of OUTPUT1N is not required. As only n bits (instead of $(n + 1)$ bits) of ROM address are now needed, the ROM size required gets reduced further, as compared to the

NADC of Fig. 3. The improved NADC configuration is shown in Fig. 4. In this case, the ROM is logically divided into three blocks:

- Block I for providing the most significant $(n - 1)$ bits of data required for generating the next address for the successive approximation process;
- Block II for providing an m -bit nonlinear code for the DAC; and
- Block III for producing the final m -bit nonlinear code after the n -step successive approximation process.

Table IV shows the memory data layout for the ROM in Fig. 4 for realizing the nonlinear characteristic of Table I. The operation of the NADC is now briefly described. Initially, the ROM address register is reset. This makes OUTPUT1 = 4, and OUTPUT2 = 24 which is converted into the analog value and compared with V_i . After comparison, the next address is formed by juxtaposing

Fig. 4. Improved NADC configuration equivalent to the NADC of Fig. 2.

TABLE IV
MEMORY DATA LAYOUT FOR ROM OF FIG. 4 WITH $m = 5$ AND $n = 4$

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OUTPUT1	4	—	0	1	1	3	2	3	2	6	4	5	5	7	6	1
OUTPUT2	24	—	4	12	8	22	20	23	16	28	25	27	26	30	29	31
OUTPUT3	0	4	8	12	16	20	22	23	24	25	26	27	28	29	30	31

— unused.

OUTPUT1 and the comparator output in the LSB position. Thus, the next address is either 8 or 9, and accordingly, as shown in Table IV, OUTPUT2 is either 16 or 28. In this manner, the successive approximation process is continued to determine the most significant three bits of the NADC output. After the fourth step, these three bits are contained in the three MSBs of the address, and the comparator output is loaded as the LSB of the address. Therefore, the resulting ROM address itself is the 4-b compressed code corresponding to the analog input, and OUTPUT3 gives the corresponding 5-b nonlinear code. In applications where only the compressed code corresponding to the nonlinear quantization scale is needed, block III of the ROM is not necessary. As the ROM size required for this improved NADC configuration is $2^n \times (2m + n - 1)$ bits, the ROM size reduction (r) achieved is $[1 - 2^n(2m + n - 1)/(2^{n+1}m)]100\%$ when compared to the NADC of [1]. For the nonlinear characteristic of Table I, $r = 35\%$, and for a typical codec with $m = 13$ and $n = 8$, r is 96%, clearly illustrating the saving in ROM size and hence in chip area.

III. GENERATION OF ROM DATA

The starting point for the generation of memory data for the ROM in Fig. 4 is a set of ordered pairs corresponding to the linear and nonlinear codes as shown in Table I. This determines directly the entry OUTPUT3 of Table IV. The values of OUTPUT1 are $(n - 1)$ bit codes obtained by ignoring the LSB of the corresponding n -bit code OUTPUT1N. Let the notation SYMBOL(a) represent the value of the entity SYMBOL corresponding to the ROM address a . As the ROM address register is reset at the start of A/D conversion

$$\text{OUTPUT1N}(0) = 2^{n-1}$$

which is the starting code value for the successive approximation

process. Then, as the new address is obtained by juxtaposing the $(n - 1)$ MSB's of OUTPUT1N (0) and the comparator output in the LSB position, the two possible next addresses are 2^{n-1} and $2^{n-1} + 1$. Therefore

$$\text{OUTPUTIN}(2^{n-1}) = 2^{n-1} - 2^{n-2},$$

and

$$\text{OUTPUTIN}(2^{n-1} + 1) = 2^{n-1} + 2^{n-2}.$$

Then, with $2^{n-1} - 2^{n-2}$ as the ROM output, the next two possible addresses are $2^{n-1} - 2^{n-2}$ and $2^{n-1} - 2^{n-2} + 1$, and accordingly,

$$\text{OUTPUTIN}(2^{n-1} - 2^{n-2}) = 2^{n-1} - 2^{n-2} - 2^{n-3}$$

and

$$\text{OUTPUTIN}(2^{n-1} - 2^{n-2} + 1) = 2^{n-1} - 2^{n-2} + 2^{n-3}.$$

In the same manner, all the values of OUTPUTIN may be determined by logically following the successive approximation sequence for each output value of the ROM, and the two possible values of the comparator output. An algorithm for the automated computation of OUTPUTIN is given in Fig. 5. This algorithm may be easily coded into a computer program. Once all the values of OUTPUTIN are determined, the values of OUTPUT1 may be easily found by the equation

$$\text{OUTPUT1} = \text{OUTPUTIN} \text{ div } 2$$

where 'div' is the integer division operator. The values of OUTPUT2 of Table IV may be obtained by the relation

$$\text{OUTPUT2}(a) = \text{OUTPUT3}(\text{OUTPUTIN}(a)).$$

Fig. 5. Algorithm for OUTPUTIN computation.

TABLE V
EXAMPLE NONLINEAR CHARACTERISTIC USED FOR EXPERIMENTAL
IMPLEMENTATION

Linear Code (5-b)	Nonlinear Code (7-b)	Linear Code (5-b)	Nonlinear Code (7-b)
0	0	16	27
1	1	17	31
2	2	18	35
3	3	19	39
4	4	20	43
5	5	21	47
6	6	22	51
7	7	23	55
8	9	24	63
9	11	25	71
10	13	26	79
11	15	27	87
12	17	28	95
13	19	29	103
14	21	30	111
15	23	31	119

For example, in Table IV, OUTPUT2 (5) is obtained as OUTPUT3 (OUTPUT1N (5)). As OUTPUT1N (5) = 6, it is found that OUTPUT2 (5) = OUTPUT3 (6) = 22.

IV. EXPERIMENTAL IMPLEMENTATION

The improved NADC of Fig. 4 has been experimentally implemented using a microcomputer as the controller. In the implementation, the RAM of the microcomputer was used in place of the

Fig. 6. Experimental transfer characteristic for the NADC of Fig. 4.

ROM of the NADC for programming convenience. The example nonlinear characteristic shown in Table V, with $m = 7$ and $n = 5$ has been chosen for illustration. From this Table, ROM programming data shown in Table VI is derived following the procedure outlined in Section III. The NADC was programmed using these values and an automated test was conducted. The microcomputer was used with an 8-b DAC to provide the analog input with a full scale $V_i = 3$ V. The NADC output codes for the test inputs were stored in the microcomputer memory and later graphically plotted using a personal computer, as shown in Fig. 6, which clearly validates the operation of the NADC.

TABLE VI
ROM DATA FOR THE NONLINEAR CHARACTERISTIC OF TABLE V

ADDRESS	OUTPUT1	OUTPUT2	OUTPUT3	ADDRESS	OUTPUT1	OUTPUT2	OUTPUT3
0	8	27	0	16	4	9	27
1	—	—	1	17	12	63	31
2	0	1	2	18	8	31	35
3	1	3	3	19	9	39	39
4	1	2	4	20	9	35	43
5	3	6	5	21	11	51	47
6	2	5	6	22	10	47	51
7	3	7	7	23	11	55	55
8	2	4	9	24	10	43	63
9	6	17	11	25	14	95	71
10	4	11	13	26	12	71	79
11	5	15	15	27	13	87	87
12	5	13	17	28	13	79	95
13	7	21	19	29	15	111	103
14	6	19	21	30	14	103	111
15	7	23	23	31	15	119	119

VI. CONCLUSION

An NADC using a new technique for optimization of ROM size has been described. The operation of the NADC has been experimentally validated. The new NADC is superior to the NADC of [1] as it requires a much smaller sized ROM.

REFERENCES

- [1] C. K. Yuen, "Flexible A/D conversion using a ROM in place of a SAR," *Proc. IEEE*, vol. 71, pp. 1454-1456, Dec. 1983.
- [2] D. K. Anvekar, "Programmable nonlinear ADCs: Some new techniques," Ph.D. thesis, Indian Institute of Science, 1990.