

G. Bongiovanni
C. K. Wong

A Number Representation Converter for Magnetic Bubble String Comparators

Conventionally, there are four common ways to represent negative numbers, namely, 1's-complement, 2's-complement, signed-magnitude, and excess- 2^{m-1} . In all these representations except the last one (excess- 2^{m-1}), 1 is used as the sign bit to denote negative numbers. However, in a recently reported magnetic bubble device, using 1 as the sign bit for negative numbers may cause problems when two numbers are compared. In this paper we define a new number representation (called B1's-complement), and we propose a simple, read-free device for converting numbers represented in any of the above standard representations to either the excess- 2^{m-1} or the B1's-complement representation.

1. Introduction

Conventionally, there are four common ways to represent binary negative numbers, namely, 1's-complement, 2's-complement, signed-magnitude, and excess- 2^{m-1} (see [1, pp. 28-37]. In all these representations except the last one (excess- 2^{m-1}), 1 is used as the sign bit to denote negative numbers. (See examples in Table 1.)

However, in a recently reported magnetic bubble memory device, using 1 as the sign bit for negative numbers may cause problems when two numbers are compared. To our knowledge, the only device that does this comparison of two numbers is known as a *bubble string comparator* (BSC) [2].

It takes two binary numbers, A, B, represented in bits, as inputs and compares them bit by bit. When the first unequal bit is detected, the number with the bit 1 automatically goes to output C, while the other goes to D [3]. (See Fig. 1.)

Clearly, the sign bit 1 for negative numbers will give erroneous results, and the BSC cannot be used to compare negative numbers with such a sign bit.

For this reason, only the excess- 2^{m-1} representation can be used as is without problems in the BSC.

In this paper, we define a new number representation (called the *B1's-complement*), and we propose a simple, read-free device for converting numbers represented in any of the above standard representations to either the excess- 2^{m-1} or the B1's-complement representation. By "read-free" we mean the device need not know the number itself before doing the conversion. This is important

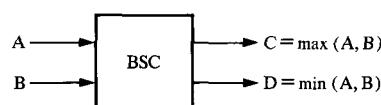


Figure 1 The bubble string comparator.

Table 1 Leftmost bit is the sign bit.

Number value	1's-complement	2's-complement	Signed-magnitude	Excess- 2^{m-1}
+5	0 1 0 1	0 1 0 1	0 1 0 1	1 1 0 1
-5	1 0 1 0	1 0 1 1	1 1 0 1	0 0 1 1

Copyright 1981 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to republish other excerpts should be obtained from the Editor.

Table 2 The different conversions performed by the GNRC.

Given number	Number representation	Number in the given representation	Conversion string $b_R b_{R-1} \dots b_0$	Resulting number	Resulting representation
+5	1's-complement	0 1 0 1	1 0 0 0	1 1 0 1	B1's-complement
+0		0 0 0 0		1 0 0 0	
-0		1 1 1 1		0 1 1 1	
-5		1 0 1 0		0 0 1 0	
+5	signed-magnitude	0 1 0 1	1 0 0 0	1 1 0 1	B1's-complement
0		0 0 0 0	1 0 0 0	1 0 0 0	
-5		1 1 0 1	1 1 1 1	0 0 1 0	
+5	2's-complement	0 1 0 1	1 0 0 0	1 1 0 1	excess- 2^{m-1} with different representation of zero
0		0 0 0 0		1 0 0 0	
-5		1 0 1 1		0 0 1 1	
+5	excess- 2^{m-1}	1 1 0 1	0 0 0 0	1 1 0 1	excess- 2^{m-1}
0		0 0 0 0		0 0 0 0	
-5		0 0 1 1		0 0 1 1	

Table 3 Leftmost bit is the sign bit.

Number value	1's-complement	B1's-complement
+5	0 1 0 1	1 1 0 1
-5	1 0 1 0	0 0 1 0

because, in the signed-magnitude representation, a different sign bit requires a different conversion scheme. (See Table 2.)

2. The B1's-complement

The B1's-complement is defined as follows: For positive numbers, we will assign sign bit 1, followed by the number itself. For negative numbers, we will use the conventional 1's-complement, but change the sign bit to 0. That is, we just change the sign bit of the conventional 1's-complement representation to its complement. See Table 3 for an example.

Clearly, comparison of positive and/or negative numbers will always produce the correct result.

3. Implementation of the number representation convertor

In general, if we change the number representation in one storage medium, it may cause problems in its communication with other parts of the system, which still use the conventional representation, unless this change can be

made completely transparent outside the storage medium. To this end, we propose a very simple device to be implemented in bubble technology, which we refer to as the *number representation convertor* (NRC). This is essentially the "exclusive-or" logic gate described in [4], but with one specific input for our purposes. (See Fig. 2.)

Suppose the number N to be converted is represented in bits as $x_R x_{R-1} \dots x_1 x_0$, where x_R is the sign bit. This string of bits is fed into the NRC (in the order of $x_R, x_{R-1}, \dots, x_0$) so that the output will be $x'_R x'_{R-1} \dots x'_0$, where x'_i is the value of the i th bit in the new representation of N .

Specifically, we have two strings of bubbles as input to the NRC: One is the number $x_R x_{R-1} \dots x_0$, another is a string $b_R, b_{R-1}, \dots, b_0$ (called *clock bubbles*), so that the two strings have exactly the same length.

If we denote the two bits of the strings currently entering the NRC by A and B , then the NRC just performs the exclusive-or logic function. The output will be $A \oplus B$. Specifically,

1. If the clock bubble $b_i = 0$, the output from the NRC will be x_i .
2. If the clock bubble $b_i = 1$, the output from the NRC will be $\bar{x}_i$. This is achieved physically as follows: If $x_i = 0$, then b_i will take its place on the output path. If $x_i = 1$, then the two bubbles will repel each other and will go out on the two vertical paths to the annihilators. Consequently, we have a zero on the output path.

The actual implementation of the NRC can be realized by the exclusive-or logic gate proposed in [4] and depicted in Fig. 3.

The exclusive-or gate works as follows. It has two incoming paths A and B, two $A \cdot B$ outgoing paths, and one $A \oplus B$ outgoing path. When bubbles on the incoming paths are in positions 1A and 1B, one of the following cases occurs:

1. If a bubble is present at 1A (or 1B), and none is present at 1B (or 1A), then at field phase 2 that bubble will move to position 2A (or 2B), since this position is nearer to 1A (or 1B) than 2A' (or 2B'). Thus, such a bubble will travel to the $A \oplus B$ output.
2. If two bubbles are simultaneously present at 1A and 1B, then the mutual repulsive force will drive them, at field phase 2, to positions 2A' and 2B'. Thus, these two bubbles will travel to the two $A \cdot B$ paths, and no bubble will travel to the $A \oplus B$ path.
3. If no bubble is present at either 1A or 1B, then obviously no bubble will move to the $A \cdot B$ paths or to the $A \oplus B$ path.

Now it is easy to see that we can, by means of the NRC, change any number representation which is inadequate for the BSC into one which is adequate. All we have to do is to tailor a fixed sequence $b_R b_{R-1} \dots b_0$ for the actual conversion we want to realize. In fact, we have the following properties:

- a. If the given number representation is the 1's-complement, then the NRC, fed with the fixed sequence

$$b_R b_{R-1} b_{R-2} \dots b_0 = 100 \dots 0,$$

will produce the number in the B1's-complement form.

- b. If the given number representation is the 2's-complement, then the NRC, fed with the fixed sequence

$$b_R b_{R-1} b_{R-2} \dots b_0 = 100 \dots 0,$$

will produce the number in the excess- 2^{m-1} representation, with a different representation of zero ($100 \dots 0$ instead of $000 \dots 0$), which preserves the right ordering anyway.

- c. If the number representation is the excess- 2^{m-1} , then the NRC, fed with the fixed sequence

$$b_R b_{R-1} b_{R-2} \dots b_0 = 000 \dots 0,$$

will leave the number unchanged, since this representation need not be changed. All these conversions are read-free, since they depend only on the length of the records to be converted (*i.e.*, fixed clock bubble sequences of appropriate length will perform the conversions).

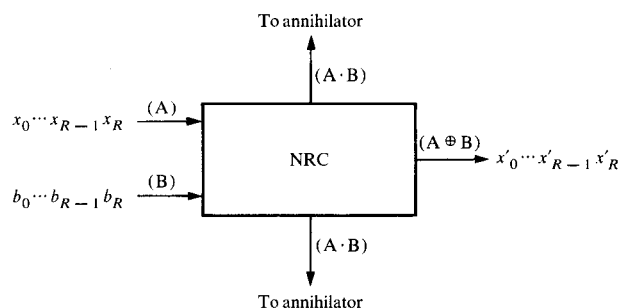


Figure 2 The number representation convertor (NRC).

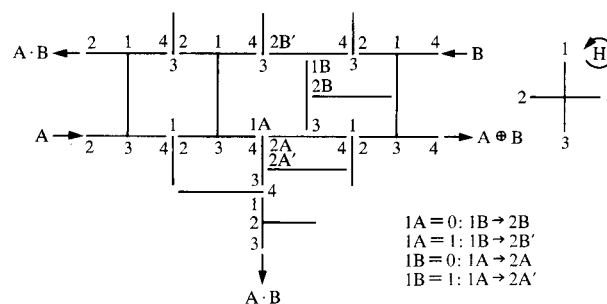


Figure 3 Implementation of the NRC in magnetic bubble technology.

- d. If the given number representation is the signed-magnitude, then we have the two following cases:

1. If the number N to be converted is positive, we have to feed the NRC with the sequence

$$b_R b_{R-1} b_{R-2} \dots b_0 = 100 \dots 0$$

in order to obtain the B1's-complement representation of N .

2. If N is negative, we have to feed the NRC with the sequence

$$b_R b_{R-1} b_{R-2} \dots b_0 = 111 \dots 1$$

in order to obtain the B1's-complement representation of N .

Thus, although the NRC is able to perform the first three conversions easily, it has to be slightly more complicated in order to perform the last one since it is data-dependent. In the next section we show how to design a read-free device which performs that conversion also.

4. The signed-magnitude adaptor

As seen in the previous section, when we want to convert a number $N = x_R x_{R-1} \dots x_0$ from the signed-magnitude to the B1's-complement representation, we have to produce two different clock bubble strings, namely,

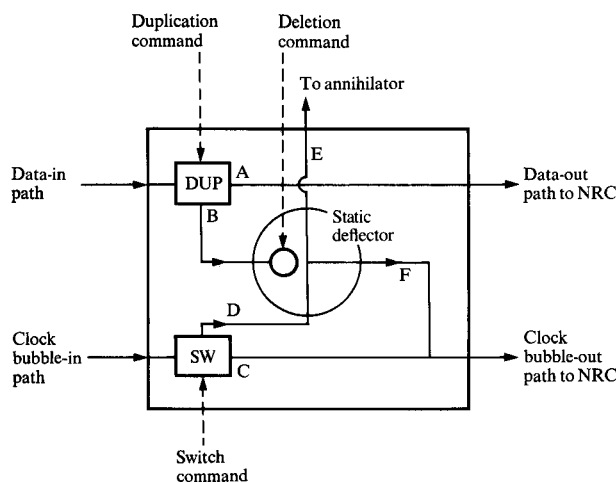


Figure 4 The signed-magnitude adaptor (SMA).

Table 4 Conversions performed by the IGNRC.

Given number representation	Sign bit of number to be converted	Conversion string	Resulting number representation
BI's-complement	0 1	1 0 0 ... 0	1's-complement
BI's-complement	0 1	1 1 1 ... 1 1 0 0 ... 0	signed-magnitude
excess- 2^{m-1} (with different zero)	0 1	1 0 0 ... 0	2's-complement
excess- 2^{m-1}	0 1	0 0 0 ... 0	excess- 2^{m-1}

if $x_R = 0$, then $b_R b_{R-1} b_{R-2} \dots b_0 = 100 \dots 0$ and

if $x_R = 1$, then $b_R b_{R-1} b_{R-2} \dots b_0 = 111 \dots 1$.

To this end, we propose a device, called a *signed-magnitude adaptor* (SMA), through which both the data string and the clock bubble string $111 \dots 1$ are passed before entering the NRC. This device uses the *static deflector* proposed in [5, p. 250] to produce, as a function of x_R , the desired clock bubble string (see Fig. 4). The SMA also uses a duplicator (see [6, p. 55]), indicated as DUP in Fig. 4, to produce a copy of a given bit (the original goes onto path A, the copy onto path B); a switch (see [6, p. 56]), indicated as SW in Fig. 4, to drive bubbles onto either one of the two different paths (C and D); a crossing without interference of two paths (A and E) (see [6, p. 54]); and a merger of two paths (C and F) into a single one (see [2, p. 1184]).

The SMA works as follows: The number $N = x_R x_{R-1} \dots x_0$ to be converted enters on the data-in path, and, when the bit x_R (the sign bit) is in DUP, a duplication signal is activated, so that the original x_R proceeds on data

path A and one copy of x_R goes onto path B to "charge" the static deflector. In the meantime, a sequence $b_R b_{R-1} \dots b_0 = 11 \dots 1$ is entered on the clock bubble-in path. The leading bit b_R of this sequence is driven by the switch SW onto the clock bubble-out path C (since this bit is 1 and is what we want), and the sequence $b_{R-1} b_{R-2} \dots b_0$ is driven by SW onto the path D leading to the static deflector.

Now, if the static deflector (which is already charged) contains 0, the sequence $b_{R-1} b_{R-2} \dots b_0$ is driven onto the path E leading to an annihilator, thus leaving on the clock bubble-out path a sequence $b_R b_{R-1} \dots b_0 = 10 \dots 0$.

If, on the other hand, the static deflector contains 1, then the sequence $b_{R-1} b_{R-2} \dots b_0$ is driven onto the path F, which merges with path C, so as to form on the clock bubble-out path the sequence $b_R b_{R-1} \dots b_0 = 11 \dots 1$.

The following constraints among the different path lengths must be observed, in order to maintain the synchronization needed in the NRC:

1. The length of any path connecting any input with any output of the SMA is the same;
2. The length of path B is smaller than or equal to the length of path D. This is because the static deflector must already contain the sign bit x_R when b_{R-1} enters it. The equality is allowed because x_R enters the device one step before b_{R-1} .

Obviously, these constraints are very easy to realize by adjusting the number of propagating elements on each path in the SMA.

As it is easy to see, what the SMA demands from the external control is only a fixed sequence of signals, depending only on the data length but not on the data value. Thus, it is a read-free device.

5. The general number representation convertor

We are now able to perform all of the four conversions we have mentioned so far. Three of them can be performed by the NRC alone; one needs an SMA cascaded with an NRC.

Obviously, all of the three conversions performed by the NRC alone can also be performed by the SMA and NRC together. In fact, it is sufficient to drive the desired clock bubble string directly onto path C in the SMA, without intervention of the static deflector.

So, the most general structure for implementing all the number representation conversions is depicted in Fig. 5,

and is called the *general number representation convertor* (GNRC).

The different functions performed by the GNRC are summarized with examples in Table 2. Note that all these conversions preserve the ordering of the zero with respect to positive and negative numbers (in the conversion from 1's-complement to B1's-complement, both plus-zero and minus-zero preserve such an ordering).

6. The inverse number representation convertor

To make the on-chip number conversion completely transparent to the outside world, we need also to convert "bubble numbers" back to their external representation before they are used outside the chip.

The reconversions relative to the three conversions which can be performed by the NRC alone can also be performed by the NRC alone, since the reversion functions in these three cases are identical to the conversion functions.

To reconvert a number from the B1's-complement representation back to the signed-magnitude representation, we have the following two cases:

1. If the sign bit x_R of the number N to be reconverted is 0, then we have to feed the NRC with the sequence $b_R b_{R-1} \dots b_0 = 11 \dots 1$ in order to obtain the signed-magnitude representation of N .
2. If the sign bit x_R of N is 1, then we have to feed the NRC with the sequence

$$b_R b_{R-1} \dots b_0 = 10 \dots 0$$

in order to obtain the B1's-complement representation of N .

It is easy to see that by simply interchanging paths E and F in the SMA we can construct the proper conversion strings to be fed into the NRC. We call such a device the *inverse signed-magnitude adaptor* (ISMA).

Now we are ready to construct the *inverse general number representation convertor* (IGNRC), which is identical to the GNRC, except that instead of containing the SMA, it contains the ISMA.

The different conversions performed by the IGNRC are summarized in Table 4.

7. Chip configuration

In the conventional bubble chip design, because of their physical size there are usually only one read port and one write port to read data from and to write data into the

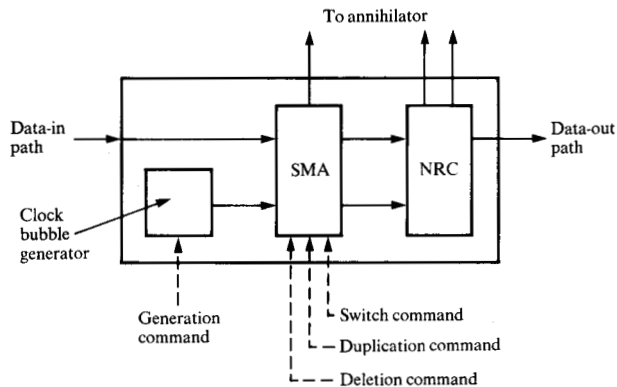


Figure 5 The general number representation convertor (GNRC).

chip. On the other hand, conceivably there may be many bubble string comparators on the chip. Therefore, instead of coupling all the BSC's with NRC's, we need to place only one IGNRC immediately before a read port to convert the numbers from the B1's-complement or excess- 2^{m-1} representation back to any one of the four conventional representations. Similarly, we can place one GNRC immediately after a write port to convert the numbers from their conventional representations to either the B1's-complement or excess- 2^{m-1} representation.

References and note

1. H. Stone, *Introduction to Computer Organization and Data Structures*, McGraw-Hill Book Co., Inc., New York, 1972.
2. B. Antonini, M. Bacchi, and G. Bongiovanni, "A Bubble String Comparator for Information Processing," *IEEE Trans. Magnetics* **MAG-15**, 1183-1185 (1979).
3. A device which has two inputs and two outputs such that one fixed output is always the smaller of the two inputs and the other is the larger is generally referred to as a comparator or a comparator module in the terminology of sorting networks (see [7, p. 220]).
4. R. Sandfort and E. Burke, "Logic Functions for Magnetic Bubble Devices," *IEEE Trans. Magnetics* **MAG-7**, 358-360 (1971).
5. T. C. Chen and H. Chang, "Magnetic Bubble Memory and Logic," *Advances in Computers*, Academic Press, Inc., New York, 1978.
6. H. Chang, *Magnetic Bubble Technology: Integrated Circuit Magnetics for Digital Storage and Processing*, IEEE Press, New York, 1975.
7. D. Knuth, *The Art of Computer Programming*, Addison-Wesley Publishing Co., Reading, MA, 1973.

Received July 10, 1980; revised August 28, 1980

G. Bongiovanni is at the Università di Pisa, Istituto di Scienze dell'Informazione, Italy, but was a Visiting Scientist at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, when this work was done. C. K. Wong is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.