#234

EQUIVALENCE OF SOME BINARY, TERNARY AND QUATERNARY FIBONACCI COMPUTERS

P. Ligomenides and R. Newcomb
Intelligent Machines Program and
Electrical Engineering Department
University of Maryland
College Park, MD. 20742
Phone: (301)454-2442

Abstract

Use of the Fibonacci recursion relation and completeness of the corresponding binary coefficient sequence yields complete ternary and quaternary Fibonacci number representations suitable for construction of efficient Fibonacci computers.

I. Introduction

Multi-valued logic has a rich history both in philosophical and computer hardware contexts with an extensive bibliography built up even prior to 1965.[1] Since that time one finds it applied further in many and varied situations, such as to neural modeling[2] and "possible" automata.[3] Here we discuss it in the context of Fibonacci computers through an interesting equivalence between binary, ternary and quaternary logic as applied to such computers.

The Fibonacci computer has been introduced[4] as a computer which performs calculations in terms of Fibonacci numbers rather than binary ones. As such it has apparent advantages for error correction because of the redundancy built into the number base. Too, it allows construction in terms of normal binary logic hardware since any integer can be represented in terms of the Fibonacci number base using only binary coefficients[5]. But if one goes to ternary coefficients then special error correcting properties come to light, with the Fibonacci number base with ternary coefficients having special advantages for detecting and correcting burst errors.[6]

Here we point out equivalences between some binary, ternary, and quaternary representations of arbitrary integers for new classes of Fibonacci computers. Since finite sums are appropriate for computers, we investigate the restrictions imposed by finiteness, where completeness within the range still holds. Along with this investigation comes that of efficiency of the representation for logic construction. In Section II we make these investigations while in Section III we briefly discuss further the significance for computer systems.

II. Binary to Ternary and Quaternary-Fibonacci Number Representations

In this section we show the equivalence of binary weighted and a certain ternary and quaternary weighted Fibonacci base number representation.

As background we recall that it has been shown[7] that the Fibonacci representation

$$\sum_{j=2}^{m} b_j u_j \quad , \quad b_j \in \{0,1\} \tag{1}$$

where

$$u_j = u_{j-1} + u_{j-2} \quad , \quad u_0 = 0,\ u_1 = 1 \tag{2}$$

is a complete representation of <u>all</u> positive integers in the range [0,M],

where

$$M = u_{m+2} - 2 \tag{3}$$

This holds true for various initial values or "seeds," $A=u_0$ and $B=u_1$, with the one chosen in (2) giving the Fibonacci numbers and A=2,B=1, giving the Lucas Numbers. Thus any positive integer N, $0 \le N \le M$, can be expanded in the Fibonacci representation using binary coefficients on the Fibonacci numbers, $F_i = u_i$, as a complete base or radix set.

1. A Quaternary-Fibonacci Number Representation

We use the Fibonacci number sequence to provide us with the radix numbers in the weighted binary representation of (1)

$$N = \sum_{j=2}^{m} b_j F_j \tag{4}$$

of all positive numbers less than or equal to $M = F_{m+2} - 2$. From the characterizing property of the series we have the difference equation applied to odd subscripts, as follows:

$$F_{2n+1} = F_{2n+2} - F_{2n} \tag{5}$$

where $n = 1, 2, \ldots, \frac{m-2}{2}$ then

$$b_{2n+1}F_{2n+1} = b_{2n+1}(F_{2n+2} - F_{2n}) = b_{2n+1}F_{2n+2} - b_{2n+1}F_{2n} \tag{6a}$$

$$b_{2n-1}F_{2n-1} = b_{2n-1}(F_{2n} - F_{2n-2}) = b_{2n-1}F_{2n} - b_{2n-1}F_{2n-2} \tag{6b}$$

adding

$$b_{2n-1}F_{2n-1} + b_{2n}F_{2n} + b_{2n+1}F_{2n+1} = -b_{2n-1}F_{2n-2} + (b_{2n} + b_{2n-1} - b_{2n+1})F_{2n} + b_{2n+1}F_{2n+2} \tag{7}$$

which has only even subscripted terms on the right. Therefore, substitutions will convert the binary representation $\sum_{j=2}^{m} b_j F_j$ into one in which <u>only even</u> subscripted Fibonacci numbers, F_{2n}, appear, each with coefficients

$$d_n = b_{2n} + b_{2n-1} - b_{2n+1} \quad , \quad n = 1, 2, \ldots, k \tag{8}$$

Indeed the (m-1) bits $(b_2, b_3, \ldots, b_m)$ give

$$\sum_{j=2}^{m} b_j F_j = b_2F_2 + b_3F_3 + \ldots + b_mF_m \tag{9a}$$

$$= (b_2 + b_1 - b_3)F_2 + (b_4 + b_3 - b_5)F_4 + (b_6 + b_5 - b_7)F_6 +$$

0195-623X/81/0000/0082$00.75 © 1981 IEEE

$$+\dots+(b_{2k-2}+b_{2k-3}-b_{2k-1})F_{2k-2}+(b_{2k}+b_{2k-1}-b_{2k+1})F_{2k} \quad (9b)$$

$$= \sum_{i=1}^{k} (b_{2i}+b_{2i-1}-b_{2i+1})F_{2i} \quad (9c)$$

$$= \sum_{i=1}^{k} d_i F_{2i} \quad (9d)$$

where $k = \frac{m}{2}$ if m is even, or $k = \frac{m+1}{2}$ if m is odd, which yields (8) with $b_1=0$, and $b_{m+1}=0$ (9e)

The possible values of d_i are shown in Table 1.

Table 1
Coefficient Relations

b_{2i-1}	b_{2i}	b_{2i+1}	d_i
0	0	0	0
0	0	1	-1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	2
1	1	1	1

It should be observed that the transformation of the b_i's to the d_i's is a many-one transformation, showing the added redundancy in the binary representation. We see that d_i are quaternary (four-level) digits, and the k-digit representation.

$$\sum_{i=1}^{k} d_i F_{2i} \quad (d_1 d_2 \dots d_k) \quad (10)$$

is a complete representation of all positive numbers in the range (0,M), (where $M=F_{m+2}-2$, m=2k or m=2k-1), using only k quaternary digits, i.e. one half the number needed in the binary representation. An important point to make, however, is also that by using only even subscripted Fibonacci numbers as radix numbers, negative numbers are also naturally represented by simply extending the representation to negative subscripts. We should recall that $F_{-2i} = -F_{2i}$ for the Fibonacci numbers.

The range of integer numbers represented has now been extended (doubled) to (-M,M). The representation is complete and may be formulated as

$$\sum_{i=-k}^{k} d_i F_{2i} \rightarrow (d_{-k} d_{-k+1} \dots d_{-1} d_0 d_1 \dots d_k) \quad (11)$$

Notice that the digit d_0 is superficial because it weights on F_0 which is zero ($F_0=0$ for the Fibonacci series), and therefore it does not contribute anything to the sum. The essential quaternary-Fibonacci representation of all integers in the range (-M,M) is $(d_{-k} d_{-k+1} \dots d_{-1} d_1 \dots d_k)$ and uses 2 k digits.

It may be observed here that this representation is peculiar in that the left half k digits are all zeros in the representation of a positive number, while the right half k digits are all zeros in the representation of a negative number. This suggests a more economical representation using only k+1 digits

$$(d_0 d_1 d_2 \dots d_k)$$

where d_0 is used to denote whether the positive subscripted or the negative subscripted Fibonacci numbers are used correspondingly with the weights $d_1 d_2 \dots d_k$. The digit d_0 may be used on a control input line into a Fibonacci number generator (a circuit or a software program) to control the generation of the sequence of positive or of negative subscripted radix numbers. Being a four-level digit, d_0 also allows the control of two more conditions, such as the representation of "zero" (without need to generate, set or check the weights $d_1 d_2 \dots d_k$) or the designation of an arithmetic overflow condition.

2. A Ternary-Fibonacci Number Representation

In the binary representation

$$\sum_{j=2}^{m} b_j F_j$$

we may observe a possible transformation of a sequence $\dots+1\cdot F_{j-1}+1\cdot F_j+0\cdot F_{j+1}+\dots$ into its equivalent $\dots +0\cdot F_{j-1}+0\cdot F_j+1\cdot F_{j+1}+\dots$ i.e. of a bit sequence...110... into the bit sequence ...001... . (Notice that this transformation denotes the redundancy that exists in the Fibonacci representation of integers).

This property may be used to eliminate the possibility of encountering the 001 combination in the bits $(b_{2i-1} b_{2i} b_{2i+1})$ in Table 1. If every 001 triplet, which is encountered in a sequence of odd-even-odd subscripted weights, b_j, in the binary representation, is replaced by the triplet 110, then the possibility of $d_i=-1$ is eliminated and the k-digit quaternary-Fibonacci representation is reduced into a k-digit ternary-Fibonacci representation

$$\sum_{i=1}^{k} c_i F_{2i} \rightarrow (c_1 c_2 \dots c_k) \quad (12a)$$

where

$$c_i = b_{2i}+b_{2i-1}-b_{2i+1} \quad (12b)$$

now takes only the values 0,1 and 2.

III. Observations and Discussion

The binary-Fibonacci number representation, $\sum_{j=2}^{m} b_j F_j$, is complete in the range of the first $F_{m+2}-1$ nonnegative integers, i.e. in the representation of the integers $0,1,2,\dots,F_{m+2}-2$. The representation uses (m-1) binary coefficients (weights) b_j. Since 2^{m-1} binary combinations (codes) are available for the representation of the $F_{m+2}-1$ integers, the redundancy gained by this representation has an "average multiplicity factor", Q, equal to

$$Q = \frac{2^{m-1}}{F_{m+2}-1} \tag{13}$$

This multiplicity is dependent on m and its normalized rate of change may be computed as follows:

$$\Delta Q = \frac{2^m}{F_{m+3}-1} - \frac{2^{m-1}}{F_{m+2}-1} = \frac{2^{m-1}[2F_{m+2}-F_{m+3}-1]}{(F_{m+2}-1)\ (F_{m+3}-1)} =$$

$$= \frac{2^{m-1}(F_m-1)}{(F_{m+2}-1)(F_{m+3}-1)} = Q\ \frac{F_m-1}{F_{m+3}-1} \tag{14}$$

Therefore

$$\frac{\Delta Q}{Q} = \frac{F_m-1}{F_{m+3}-1} \simeq \frac{F_m}{F_{m+3}} = \frac{F_m}{F_{m+1}+F_{m+2}} \quad \text{in large m.} \tag{15}$$

In the binary to quatenary transformation some of the multiplicity, therefore also redundancy, is absorbed, as demonstrated from Table 1. We observe that more than one bit-combination for the triplet $b_{2i-1}b_{2i}b_{2i+1}$ may correspond to the same value for d_i. The same is true for the binary to ternary transformation, which is again represented in Table 1 if we delete the second row.

The elimination of the second row of Table 1 in the binary to ternary transformation implies that a number of binary combinations are discarded as "out of code" in the transformation, which therefore reduces the available multiplicity in the representation of the natural numbers.

Here we have primarily treated number system ideas. Hardware and fault tolerant system ideas are discussed in part, for binary implementations, in Hoang.[6] We will later treat ternary and quaternary implementations and extensions in terms of hardware and fault tolerance.

In summary we have shown useful equivalences valid for Fibonacci computers which can be realized by hardware. For example in the ternary case the circuits of [8] can be readily adapted to Fibonacci computers. Thus the fault tolerant computing possibilities of Fibonacci computers may be realized by implementing the representations treated above. We point out that a very readable paperback on the properties of Fibonacci numbers is that of Hoggatt[9], it being a book which makes the topic area available to engineers and scientists at all levels.

References

[1] N. Rescher, "Many-Valued Logic," McGraw-Hill Book Co., New York, 1969.

[2] R. F. Mattrey, D. D. Givone, and C. M. Allen, "Applying Multiple-Valued Algebra Concepts to Neural Modeling," Conference Record of the 1973 International Symposium on Multiple-Valued Logic, May 1973, pp. 127-136.

[3] B. R. Gaines and L. Kohout, "Possible Automata," Proceedings of the 1975 International Symposium on Multiple-Valued Logic, May 1975, pp. 183-196.

[4] R. Newcomb, "Fibonacci Numbers as a Computer Base," Conference Proceedings of the Second Interamerican Conference on Systems and Informatics, Mexico City, November 27, 1974.

[5] P. Monteiro and R.W. Newcomb, "Minimal and Maximal Fibonacci Representations: Boolean Generation," The Fibonacci Quarterly, Vol. 14, No. 1, February 1976, pp. 9-12.

[6] V.-D. Hoang, "A Class of Arithmetic Burst-Error-Correcting Codes for the Fibonacci Computer," Ph.D. Dissertation, University of Maryland, Department of Electrical Engineering, December 1979.

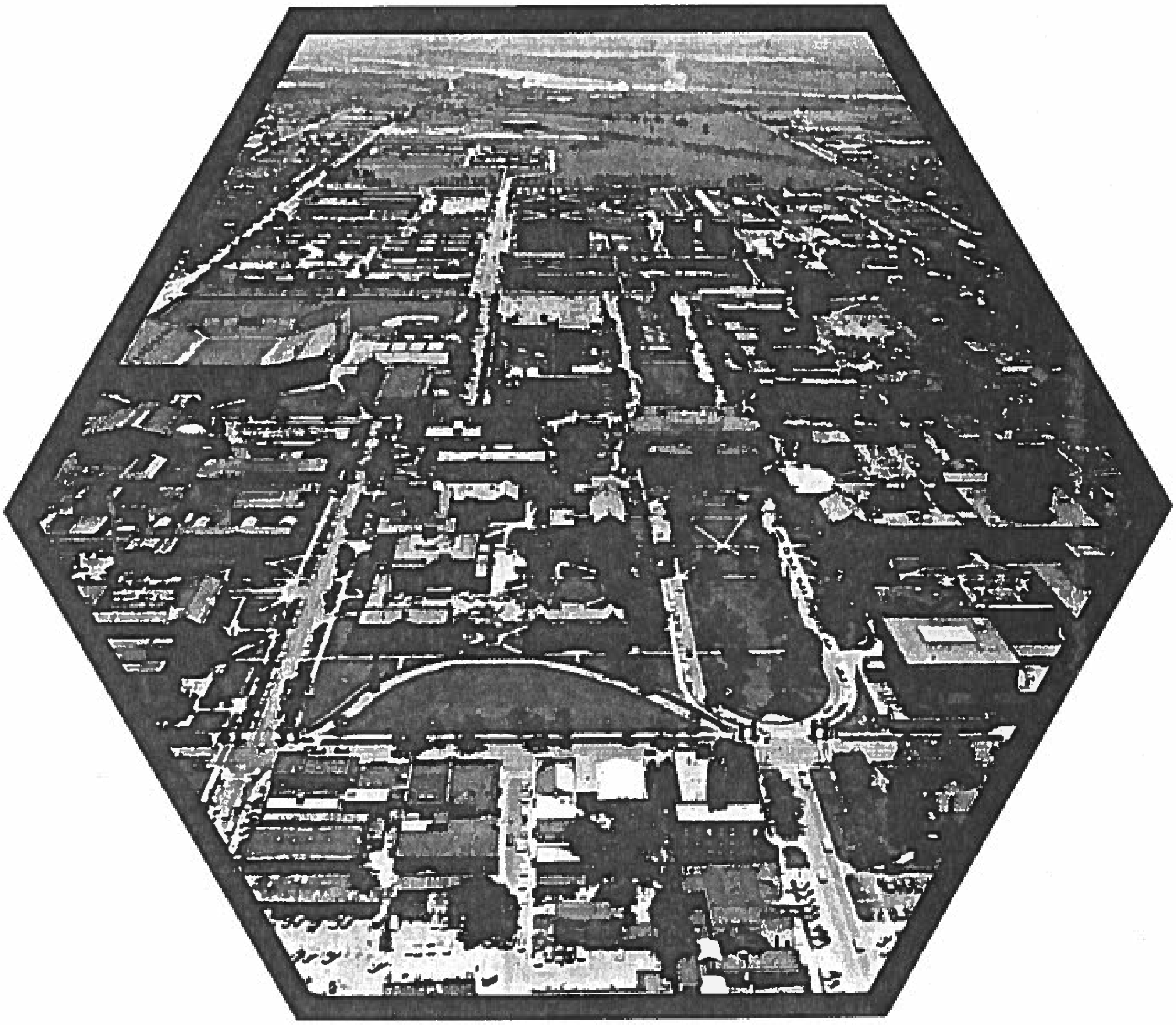
[7] J. L. Brown, Jr., "Note on Complete Sequence of Integers," The American Mathematical Monthly, Vol. 68, No. 6, June-July 1961, pp. 557-560.

[8] T. Higuchi and M. Kameyana, "Ternary Logic System Based on T-Gate," Proceedings of the 1975 International Symposium on Multiple-Valued Logic, May 1975, pp. 290-304.

[9] V. E. Hoggatt, Jr., "Fibonacci and Lucas Numbers," Houghton Mifflin Company, Boston, 1969.

PROCEEDINGS

THE ELEVENTH INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC



University of Oklahoma, Norman, OK
May 27–29, 1981

IEEE   IEEE Computer Society   The University of Oklahoma

ISSN 0195-623X
Library of Congress No. 79-641110
IEEE Catalog Number 81CH1611-3
Computer Society Order Number 348

COMPUTER SOCIETY PRESS