

- (21) Application No. 53430/77 (22) Filed 22 Dec. 1977
- (31) Convention Application No. 2432391
- (32) Filed 22 Dec. 1976 in
- (33) Soviet Union (SU)
- (44) Complete Specification published 23 April 1980
- (51) INT CL<sup>3</sup> G06F 7/385
- (52) Index at acceptance  
G4A 18F 18Y AS
- (72) Inventor ALEXEI PETROVICH STAKHOV



(54) IMPROVEMENTS IN OR RELATING TO FIBONACCI CODE ADDERS

(71) We, TAGANROGSKY RADIO-TEKHNICHESKY INSTITUT IMENI V.D. KALMYKOVA, of ulitsa Chekhova, 22, Taganrog, Union of Soviet Socialist Republics, a Corporation organised and existing under the laws of the Union of Soviet Socialist Republics, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

This invention relates to Fibonacci code adders. Such adders may be used in digital data processing systems to perform Fibonacci code adding operations.

The invention is applicable to digital computers utilizing Fibonacci codes.

According to the invention, there is provided a Fibonacci code adder comprising an n-digit half-adder having at least multidigit augend input and a multidigit addend input to accept numbers represented in the minimal form of Fibonacci codes and having its multidigit intermediate sum and carry outputs coupled, respectively, to a multidigit intermediate sum input and to a multidigit carry input of a rewriting device having its multidigit intermediate sum and carry outputs and its control output coupled, respectively, to a multidigit intermediate sum input, a multidigit carry input and a control input of a Fibonacci code converter, a fault acknowledgement output of the n-digit half-adder being coupled to a fault acknowledgement input of the Fibonacci code converter, where n is the length of the Fibonacci code.

The preferred Fibonacci code adder comprises an n-digit half-adder provided with at least two multidigit inputs through which addends of numbers represented in the Fibonacci code minimal form are input, a multidigit intermediate sum output and a multidigit carry output of the n-digit half-adder being coupled, respectively, to a multidigit intermediate sum input and a carry input of a rewriting device having its multidigit intermediate sum input, its multidigit carry input and its control input coupled, respec-

tively, to a multidigit intermediate sum input, to a multidigit carry input and to a control input of a Fibonacci code converter, and a fault acknowledgement output of the n-digit half-adder being coupled to a fault acknowledgement input of the Fibonacci code converter, where n is the length of the Fibonacci code.

Advantageously, a Fibonacci code adder, comprises an n-digit half-adder incorporating n identical one-digit half-adders each having its fault acknowledgement output coupled to a respective input of an OR gate of the n-digit half-adder, a primary carry output of an ith one-digit half-adder being coupled to one primary carry input of an (i-1)th one-digit half-adder and to the other primary carry input of an (i-2)th one-digit half-adder, a secondary carry output of the ith one-digit half-adder being coupled to one secondary input of the (i-1)th one-digit half-adder and to the other secondary carry input of the (i-2)th one-digit half-adder, a carry output and an initial sum output of the ith one-digit half-adder being coupled, respectively, to a carry input of the (i-1)th one-digit half-adder and to an initial sum input of an (i+1)th one-digit half-adder, where  $i=1, 2 \dots n$ .

Preferably, an ith one-digit half-adder comprises OR gates, AND gates and an inverter, first and second inputs of an addend analysis OR gate being coupled, respectively, to first and second inputs of an addend analysis AND gate, the output of the addend analysis OR gate being coupled to first inputs of an initial sum OR gate, a first test signal selection OR gate and a carry analysis AND gate, the output of the initial sum OR gate being coupled to a first input of an intermediate sum OR gate, a second input of the initial sum OR gate being coupled to a first input of a first test signal selection AND gate having its second input coupled to the output of the first test signal selection OR gate and having its output coupled to a first input of a test OR gate having two other inputs coupled, respectively, to the output of a second test signal selection AND gate and to the output

of a third test signal selection AND gate, a first input of the second test signal selection AND gate being coupled to the output of a secondary carry AND gate, a second input of the second test signal selection AND gate being coupled to the output of a second test signal selection OR gate having its first input coupled to a first input of a carry AND gate and to the input of the inverter, having its second input coupled to a first input of the third test signal selection AND gate and to the output of the carry AND gate and having its third input coupled to the output of the addend analysis AND gate, one of two other inputs of the intermediate sum OR gate being coupled to one of two other inputs of the first test signal selection OR gate, the remaining input of the intermediate sum OR gate being coupled to a second input of the carry analysis AND gate, to the remaining input of the first test signal selection AND gate and to the output of a carry analysis OR gate, the output of the carry analysis AND gate being coupled to a second input of the carry AND gate and to a first input of the secondary carry AND gate whose second input is coupled to the output of the inverter.

It is preferable that a rewriting device with a control input comprises k identical rewriting cells each being provided with a delay having its output coupled to a first input of a high-order intermediate sum AND gate, to a first input of a high-order carry AND gate, to a first input of a low-order intermediate sum AND gate and to a first input of a low-order carry AND gate, the input of a delay of an mth rewriting cell being coupled to the output of a delay of an (m+1)th rewriting cell of the rewriting device, the input of a delay of a kth rewriting cell being used as the control input of the rewriting device and the output of a delay of a first rewriting cell being used as a control output of the rewriting device, where

$$m \begin{cases} 1, 2 \dots \frac{n}{2} & \text{at an even } n \\ 1, 2 \dots \frac{n+1}{2} & \text{at an odd } n, \text{ and} \end{cases}$$

$k=1, 2 \dots n.$

It is advantageous that a Fibonacci code converter comprises a store for flip-flop and at least n identical conversion cells provided with message inputs and message outputs and with check acknowledgement outputs coupled to respective inputs of a test OR gate of the Fibonacci code converter, the output of the test OR gate being coupled to an input of a test OR gate of the Fibonacci code adder, a first communication output of each conversion

cell being coupled to a first input of a transient analysis OR gate whose output is coupled to the input of an inverter and to the input of an electrical filter having its output coupled to the remaining input of the test OR gate of the Fibonacci code converter, the output of the inverter being coupled to a first input of an end-of-addition AND gate having its second input coupled to the "1" output of an end-of-rewriting flip-flop, a first communication input of the ith conversion cell being coupled to a message output of an (i-1)th conversion cell, a second communication input of the ith conversion cell being coupled to a second communication output of an (i+1)th conversion cell, a first communication output of the ith conversion cell being coupled to a third communication input of the (i+1)th conversion cell and to a fourth communication input of the (i-1)th conversion cell, and a first communication output of a high-order conversion cell being coupled to the "1" set input of the store flip-flop.

Preferably, each conversion cell comprises an intermediate sum flip-flop and a carry flip-flop whose "1" set inputs are used as message inputs of the conversion cell, the "0" set input of the intermediate sum flip-flop being coupled to the output of a flip-flop state analysis AND gate having its first input coupled to the "0" output of carry flip-flop and having its second input coupled to the "0" set input of the carry flip-flop and to the output of an OR gate of the conversion cell, a first input of the OR gate being coupled to the output of a convolution AND gate having its input coupled to the "1" output of the intermediate sum flip-flop used as the intermediate sum output of the conversion cell, the "0" output of the intermediate sum flip-flop being coupled to a first input of the test AND gate of the conversion cell, whose output is used as a fault acknowledgement output of the conversion cell, the second input of the test AND gate of the conversion cell being coupled to the "1" output of the carry flip-flop.

The invention will be further described, by way of example, with reference to the accompanying drawings in which:

Fig. 1 is a block diagram of a preferred Fibonacci code adder;

Figure 2 is a block diagram of an n-digit half-adder;

Figure 3 is a block diagram of a rewriting device; and

Figure 4 is a block diagram of a Fibonacci code converter.

Figure 1 illustrates a block diagram of a Fibonacci code adder comprising an n-digit half-adder 1 whose inputs are a multidigit augend input 2 and a multidigit addend input 3, respectively, which accept numbers represented in Fibonacci codes. There is also pro-

vided a rewriting device 4 having a control input 5 to deliver a rewrite data signal, a multi-digit intermediate sum input 6 and a multi-digit carry input 7 coupled to an intermediate sum output and a carry output, respectively, of the n-digit half-adder. 1. The Fibonacci code adder also incorporates a Fibonacci code converter 8 having its multidigit intermediate sum input 9, its intermediate carry input 10, its control input 11 and its check acknowledgement input 12 coupled to an intermediate sum output, a carry output, a control output of the rewriting device 4 and a check acknowledgement output of the n-digit half-adder 1, respectively. One output of the Fibonacci code adder, while the other output of the Fibonacci code converter 8 is used as an end-of-addition denote signal output 14 of the Fibonacci code adder. The remaining output of the Fibonacci code converter 8 is used as a fault acknowledgement output 15 of the Fibonacci code adder, which produces an error signal in the case of improper operation of the Fibonacci code adder.

Figure 2 illustrates a block diagram of an n-digit half-adder 1 (with  $n=4$ ). The n-digit half-adder 1 includes four one-digit half-adders  $16_1$ — $16_4$  and an OR gate 17 whose output is used as a fault acknowledgement output of the n-digit half-adder 1. An ith one-digit half-adder  $16_i$  (in this case  $i=1, 2, 3, 4$ ) incorporates a series of AND and OR gates. All one-digit half-adders  $16_1$ — $16_4$  are identical.

Consider a block diagram of an ith one-digit half-adder  $16_i$  in the case of  $i=2$ , namely, a second one-digit half-adder  $16_2$ . A first input of an addend analysis OR gate  $18_2$  is a one-digit augend input  $2_2$  of the Fibonacci code adder, while a second input of the addend analysis OR gate  $18_2$  is a one-digit addend input  $3_2$  of the Fibonacci code adder. When combined, the one-digit inputs  $2_1$ — $2_4$  and  $3_1$ — $3_4$  of the one digit half-adders  $16_1$ — $16_4$  form, respectively, multidigit inputs 2 and 3 (Fig. 1). The inputs of the OR gate  $18_2$  are coupled to first and second inputs of an addend analysis AND gate  $19_2$  provided with an output  $20_2$ . An output  $21_2$  of the addend analysis OR gate  $18_2$  is coupled to one input of an initial sum OR gate  $22_2$ , to one input of a first test signal selection OR gate  $23_2$  and to one input of a carry analysis AND gate  $24_2$ .

The output of an OR gate  $22_2$  is coupled to a first input of an intermediate sum OR gate  $25_2$  and is used as an initial sum output  $26_2$  of the second one-digit half-adder  $16_2$ . A second input of the initial sum OR gate  $22_2$  is used as a first primary carry input  $27_2$  of the second one-digit half-adder  $16_2$  and is coupled to a first input of a first test signal selection AND gate  $28_2$  whose second input

is coupled to the output of the first test signal selection OR gate  $23_2$ . The output of the first test signal selection AND gate  $28_2$  is coupled to a first input of the test OR gate  $29_2$  whose output is used as a fault acknowledgement output  $30_2$  of the second one-digit half-adder  $16_2$ . Fault acknowledgement outputs  $30_1$ — $30_4$  of the one-digit half-adders  $16_1$ — $16_4$  are coupled to respective inputs of the OR gate 17. One of two other inputs of the test OR gate  $29_2$  is coupled to the output of a second test signal selection AND gate  $31_2$  and the remaining input of the test OR gate  $29_2$  is coupled to the output of a third test signal selection AND gate  $32_2$ . A first input of the second test signal selection AND gate  $31_2$  is coupled to the output of a secondary carry AND gate  $33_2$  which is used as a secondary carry output  $34_2$  of the second one-digit half-adder  $16_2$ . A second input of the second test signal selection AND gate  $31_2$  is coupled to the output of a second test signal selection OR gate  $35_2$ . A first input of the second test signal selection OR gate  $35_2$  is coupled to a first input of a carry AND gate  $36_2$  and to the input of an inverter  $37_2$  and is used as an initial sum input of the second one-digit half-adder  $16_2$ , coupled to the initial sum output  $26_1$  of the first one-digit half-adder  $16_1$ . A second input of the second test signal selection OR gate  $35_2$  is coupled to a first input of the third test signal selection AND gate  $32_2$  and to the output of the carry AND gate  $36_2$ , used as a carry output  $38_2$  of the second one-digit half-adder  $16_2$ . The remaining input of the second test signal selection OR gate  $35_2$  is coupled to the output  $20_2$  of the addend analysis AND gate  $19_2$ , used as a primary carry output of the second one-digit half-adder  $16_2$ . One of two other inputs of the intermediate sum OR gate  $25_2$  is coupled to one of two other inputs of the first test signal selection OR gate  $23_2$  and is used as a first secondary carry input of the second one-digit half-adder  $16_2$ , coupled to a secondary carry output  $34_3$  of a third one-digit half-adder  $16_3$ . The remaining input of the intermediate sum OR gate  $25_2$  is coupled to the remaining input of the carry analysis AND gate  $24_2$ , to the remaining input of the first test signal selection OR gate  $23_2$  and to the output of a carry analysis OR gate  $39_2$  whose first and second inputs are used, respectively, as a second primary carry input and a second secondary carry input of the second one-digit half-adder  $16_2$ . The output of the carry analysis AND gate  $24_2$  is coupled to the remaining input of the carry AND gate  $36_2$  and to a first input of the secondary carry AND gate  $33_2$  having its second input coupled to the output of the inverter  $37_2$ . The output of the intermediate sum OR gate  $25_2$  is used as an intermediate sum output  $40_2$  of the second one-digit half-adder  $16_2$  and the remaining input of the

third test signal selection AND gate  $32_2$  is used as a carry input of the second one-digit half-adder  $16_2$ , coupled to a carry output  $38_3$  of the third,  $(i+1)$ th, one-digit half-adder  $16_3$ . The primary carry input  $27_2$  of the second one-digit half-adder  $16_2$  is coupled to a primary carry output of the third,  $(i+1)$ th one-digit half-adder  $16_3$  and to a second primary carry input of the first,  $(i-1)$ th, one-digit half-adder  $16_1$ . A secondary carry output  $34_3$  of the third one-digit half-adder  $16_3$  is coupled to a first secondary carry input of the second,  $i$ th, one-digit half-adder  $16_2$  and to a second secondary carry input of the first one-digit half-adder  $16_1$ . The initial sum output  $26_2$  of the second one-digit half-adder  $16_2$  is coupled to an initial sum input of the third one-digit half-adder  $16_3$ . When combined, the carry outputs  $38_1$ — $38_4$  of the one-digit half-adders  $16_1$ — $16_4$  form a multidigit carry output of the  $n$ -digit half-adder 1 (Fig. 1) and the intermediate sum outputs  $40_1$ — $40_4$  (Fig. 2) of the one-digit half-adders  $16_1$ — $16_4$  form a multidigit intermediate output of the  $n$ -digit half-adder 1.

The addend analysis OR gate  $18_2$ , the initial sum OR gate  $22_2$  and the intermediate sum OR gate  $25_2$  are used to form intermediate and initial sum signals for the one-digit half-adder  $16_2$ . The carry analysis OR gate  $39_2$  and the intermediate sum OR gate  $25_2$  are used to form intermediate sum signals, while the addend analysis AND gate  $19_2$  is used to form primary carry signals. The carry analysis AND gate  $24_2$  and the carry AND gate  $36_2$  are designed to form carry signals. The carry analysis AND gate  $24_2$ , the secondary carry AND gate  $33_2$  and the inverter  $37_2$  are used to form secondary carry signals. The first test signal selection OR gate  $23_2$  and the first test signal selection AND gate  $28_2$  are designed to form the first test signal. The second test signal selection OR gate  $35_2$  and the second test signal selection AND gate  $31_2$  are used to form the second test signal, while the third test signal selection AND gate  $32_2$  is used to form the third test signal. The test OR gate  $29_2$  is used to form an error signal at the fault acknowledgement output  $30_2$  of the one-digit half-adder  $16_2$ . The one-digit half-adders  $16_1$ ,  $16_3$  and  $16_4$  feature a design similar to that described for the second one-digit half-adder  $16_2$ .

Figure 3 illustrates a block diagram of the rewriting device 4 comprising two identical rewriting cells  $41_1$  and  $41_2$ . The rewriting cell  $41_2$  comprises, for example, a delay  $42_2$  which provides for a pulse delay time  $\tau$  which exceeds the time required for the formation of all output signals in two adjacent one-digit half-adders  $16_3$  and  $16_4$  (Fig. 2) having their intermediate sum and carry outputs coupled, respectively, to two one-digit inputs  $6_3$  and  $6_4$  and to one-digit inputs  $7_3$  and  $7_4$  of the rewriting cell  $41_2$ . When combined, one-digit

inputs  $6_4$ ,  $6_3$ ,  $6_2$ ,  $6_1$  and one-digit inputs  $7_4$ ,  $7_3$ ,  $7_2$ ,  $7_1$  of the rewriting cells  $41_2$  and  $41_1$  form, respectively, the multidigit intermediate sum input 6 and the multidigit carry input 7 (Fig. 1) of the rewriting device 4.

The intermediate sum one-digit input  $6_4$  (Fig. 3) is used as one input of a high-order intermediate sum AND gate  $43_2$ , while the intermediate sum one-digit input  $6_2$  is used as one input of a low-order intermediate sum AND gate  $44_2$ . The outputs of the AND gates  $43_2$  and  $44_2$  form a two-digit intermediate sum output of the rewriting cell  $41_2$ . The one-digit carry inputs  $7_4$  and  $7_3$  are used, respectively, as inputs of a high-order carry AND gate  $45_2$  and a low-order carry AND gate  $46_2$  whose outputs are used to form a two-digit carry output of the rewriting cell  $41_2$ . Other inputs of the AND gates  $43_2$ ,  $44_2$ ,  $45_2$ , and  $46_2$  are coupled to the output of the delay  $42_2$ , which is coupled, in turn, to the input of a delay  $42_1$  of the rewriting cell  $41_1$ , and the output of the delay  $42_1$  is used as a control output of the rewriting device 4. When combined, the two-digit intermediate sum outputs and the two-digit carry outputs of the rewriting cells  $41_1$  and  $41_2$  form, respectively, a multidigit intermediate output and a multidigit carry output of the rewriting device 4. The latter operates to rewrite in succession data from the  $n$ -digit half-adder 1 (Fig. 1) to the Fibonacci code converter 8, starting at high-order positions and in accordance with the production of intermediate sum and carry signals at the outputs  $40_1$ — $40_4$  and  $38_1$ — $38_4$  of the one-digit half-adders  $16_1$ — $16_4$ .

Figure 4 illustrates a block diagram of the Fibonacci code converter 8 used to convert codewords representing intermediate sums and carries to the Fibonacci code minimal form representing the final sum. The Fibonacci code converter 8 includes  $(n+1)$  conversion cells, namely five conversion cells  $47_1$ — $47_5$  which are identical devices designed to perform convolution operations relating to respective positions of a given codeword. Message inputs  $9_1$ — $9_4$  of the conversion cells  $47_1$ — $47_4$  are used to form the multidigit intermediate sum input 9 (Fig. 1) of the Fibonacci code converter 8, while message inputs  $10_1$ — $10_4$  (Fig. 4) of the conversion cells  $47_1$ — $47_4$  form the multidigit carry input 10 (Fig. 1) of the Fibonacci code converter 8. Message inputs  $9_5$  and  $10_5$  (Fig. 4) of the fifth conversion cell  $47_5$  are coupled to a zero bus. The fifth conversion cell  $47_5$  is designed to store and convert codeword positions in case the number of positions of the codeword of the final sum exceeds the number of positions contained in the codewords representing initial addends. Fault acknowledgement outputs  $48_1$ — $48_5$  of the conversion cells  $47_1$ — $47_5$  are coupled to respective inputs of the test OR gate 49 of the Fibonacci code converter 8. The output of the test OR gate 49

70

75

80

85

90

95

100

105

110

115

120

125

130

produces an error signal in the case of improper operation of the conversion cells 47<sub>1</sub>—47<sub>5</sub>. First communication outputs 50<sub>1</sub>—50<sub>5</sub> of the conversion cells 47<sub>1</sub>—47<sub>5</sub> are coupled to respective inputs of a transient analysis OR gate 51 which produces transient denote signals relating to the Fibonacci code converter 8. The output of the transient analysis OR gate 51 is coupled to the input of an inverter 52 and to the input of an electrical filter 53 having its time constant exceeding the maximum time taken by a transient that takes place in the Fibonacci code converter 8 during code conversion. The output of the electrical filter 53 is coupled to the remaining input of the test OR gate 49. The output of the inverter 52 is coupled to a first input of an end-of-addition AND gate 54 whose output is the end-of addition denote signal output 14 of the Fibonacci code adder 8. A second input of the end-of-addition AND gate 54 is coupled to the "1" output of an end-of-rewriting flip-flop 55 whose "1" set input is the control input 11 of the Fibonacci code converter 8. Each conversion cell, for example, the conversion cell 47<sub>2</sub>, comprises an intermediate sum flip-flop 56<sub>2</sub> and a carry flip-flop 57<sub>2</sub> having their "1" set inputs used, respectively, as the message inputs 9<sub>2</sub>, 10<sub>2</sub> of the conversion cell 47<sub>2</sub>. The "0" set input 58<sub>2</sub> of the intermediate sum flip-flop 56<sub>2</sub> is coupled to the output of an analysis and gate 59<sub>2</sub> used to analyze the state of the carry flip-flop 57<sub>2</sub>. An input 60<sub>2</sub> of the analysis AND gate 59<sub>2</sub> is coupled to the "0" output of the carry flip-flop 57<sub>2</sub>, while an input 61<sub>2</sub> is coupled to the "0" set input of the carry flip-flop 57<sub>2</sub> and to the output of an OR gate 62<sub>2</sub> which produces a reset signal for the carry flip-flop 57<sub>2</sub>. An input of the OR gate 62<sub>2</sub> is coupled to the output of a convolution AND gate 63<sub>2</sub> used to form convolution signals to handle positions of a given codeword. The output of the convolution AND gate 63<sub>2</sub> is used as the first communication output 50<sub>2</sub> of the conversion cell 47<sub>2</sub>. A first input of the convolution AND gate 63<sub>2</sub> is a first communication input of the conversion cell 47<sub>2</sub>, coupled to a one-digit message output 13<sub>1</sub> of the first conversion cell 47<sub>1</sub>. A second input of the convolution AND gate 63<sub>2</sub> is a second communication input 64<sub>2</sub> of the conversion cell 47<sub>2</sub>, coupled to a second communication output of the conversion cell 47<sub>3</sub>. The remaining input of the convolution AND gate 63<sub>2</sub> is coupled to the "1" output of the intermediate sum flip-flop 56<sub>2</sub>, which is used as a one-digit message output 13<sub>2</sub> of the second conversion cell 47<sub>2</sub>. The "0" output of the intermediate sum flip-flop 56<sub>2</sub> is used as a second communication output of the conversion cell 47<sub>2</sub> and is coupled to a first input of a test AND gate 65<sub>2</sub> of the conversion cell 47<sub>2</sub>. The output of the test AND gate 65<sub>2</sub> is used as a fault acknowledgement

output 48<sub>2</sub> of the conversion cell 47<sub>2</sub>, while a second input of the test AND gate 65<sub>2</sub> is coupled to the "1" output of the carry flip-flop 57<sub>2</sub>. A first communication output 50 of the (i+1)th, (third) conversion cell 47<sub>3</sub> is coupled to a third communication input of the (i+2)th (fourth) conversion cell 47<sub>4</sub>, which is a counting input of the intermediate sum flip-flop 56<sub>2</sub>, and to a fourth communication input of the ith (second) conversion cell 47<sub>2</sub>, which is the remaining input of the OR gate 62<sub>2</sub>. A first communication output 50<sub>5</sub> of the fifth conversion cell 47<sub>5</sub> is coupled to the "1" set input of a store flip-flop 66 which holds data when the number of positions of the final sum codeword exceeds the number of positions of codewords of the initial addends. One-digit outputs 13<sub>1</sub>—13<sub>5</sub> of the conversion cells 47<sub>1</sub>—47<sub>5</sub> and the "1" output 13<sub>6</sub> of the store flip-flop 66 form the multidigit message output 13 (Fig. 1) of the Fibonacci code adder, which produces the final sum codeword.

The output of the test OR gate 49 (Fig. 4) is coupled to a first input of a test OR gate 67 of the Fibonacci code adder, while a second input of the test OR gate 67 is used as the fault acknowledgement input 12 of the Fibonacci code converter 8 and is connected to the output of the OR gate 17 (Fig. 2). The output of the test OR gate 67 is used as the fault acknowledgement output of the Fibonacci code adder.

The Fibonacci code adder operates as follows. Codewords representing numbers in the Fibonacci code minimal form are applied to the multidigit inputs 2 and 3 of the n-digit half-adder 1 which operates to form codewords representing intermediate sums and carries, starting at high-order positions of the Fibonacci code and moving to low-order positions of the code. In this case, the codewords representing intermediate sums and carries take a form that differs from the minimal one. At the moment when the multidigit inputs 2 and 3 accept addend signals, the control input 5 accepts a rewrite signal which causes the rewriting device 4 to transfer data from the n-digit half-adder 1 to the Fibonacci code converter 8. The rewriting of data in the rewriting device 4 is a time-dependent process which starts at the one-digit half-adders 16<sub>4</sub> and 16<sub>3</sub> (Fig. 2) corresponding to high-order positions of the Fibonacci code. As the codewords representing intermediate sums and carries appear in the Fibonacci code converter 8, the latter converts them to the Fibonacci code minimal form which is the final sum. On completion of the rewriting process, the control output of the rewriting device 4 produces an end-of-rewriting denote signal which indicates that the rewriting is terminated, that signal being applied to the control input 11 of the Fibonacci code converter 8 and stored in the

end-of-writing flip-flop 55 (Fig. 4) of the Fibonacci code converter 8. On the arrival of the end-of-writing denote signal and completion of conversion in the Fibonacci code converter 8, the end-of-addition denote signal output 14 of the latter produces a respective signal. In the case of improper operation of the n-digit half-adder 1 (Fig. 1) or the Fibonacci code converter 8, the fault acknowledgement output 15 of the Fibonacci code adder produces logic 1 to indicate that the Fibonacci code adder is at fault.

The codewords of intermediate sums and carries are formed in the n-digit half-adder 1 (for example, with  $n=4$ ) as follows. With an augend (for example, 1010) and an addend (for example, 1010) applied, respectively, to the multidigit inputs 2, 3, the production of the initial sum and the primary carry commences in the one-digit half-adders 16<sub>1</sub>—16<sub>4</sub> (Fig. 2) so that respective positions are arranged in a right-to-left succession. For example, in the case of the one-digit half-adder 16<sub>2</sub>, logic 1's applied to the one-digit inputs 2<sub>2</sub> and 3<sub>2</sub> result in the appearance of logic 1's at the output of the addend analysis OR gate 18<sub>2</sub> and at the output of the addend analysis AND gate 19<sub>2</sub>. Logic 1 from the output of the addend analysis OR gate 18<sub>2</sub> passes via the initial sum OR gate 22<sub>2</sub> and the intermediate sum OR gate 25<sub>2</sub> to the intermediate sum output 40<sub>2</sub> and to the initial sum output 26<sub>2</sub> of the one-digit half-adder 16<sub>2</sub>, that signal being passed from the initial sum output 26<sub>2</sub> to the initial sum input of the third one - digit half - adder 16<sub>3</sub>. The signal present on the initial sum output 26<sub>2</sub> is the initial sum signal for the second one-digit half-adder 16<sub>2</sub>. Logic 1 from the output of the addend analysis AND gate 19<sub>2</sub>, which is a carry signal, comes to the primary carry input 27<sub>1</sub> of the first one-digit half-adder 16<sub>1</sub> and passes through its own initial sum OR gate (not shown in Fig. 2) similar to the initial sum OR gate 22<sub>2</sub> to the initial sum output 26<sub>1</sub> and to the initial sum output of the second one-digit half-adder 16<sub>2</sub>. Also, that logic 1 passes through a first test signal selection OR gate (not shown in Fig. 2) similar to the intermediate sum OR gate 25<sub>2</sub> to the intermediate sum output 40<sub>1</sub>. In the one-digit half-adders 16<sub>1</sub>, 16<sub>2</sub> and 16<sub>3</sub>, intermediate sum signals are formed in a similar manner with the result that a codeword 1111 appears at the intermediate sum outputs 40<sub>1</sub>, 40<sub>2</sub>, 40<sub>3</sub> and 40<sub>4</sub>.

Now consider how a carry codeword is produced (with the initial codeword equal to 1010). In the fourth one-digit half-adder 16<sub>4</sub>, the output of an addend analysis AND gate similar to the addend analysis AND gate 19<sub>2</sub> produces logic 1 which is a primary carry signal applied to the first primary carry input 27<sub>3</sub> of the third one-digit half-adder 16<sub>3</sub> and to the second primary carry input of the

second one-digit half-adder 16<sub>2</sub>, namely, to an input of the carry analysis OR gate 39<sub>2</sub>. After passing through the carry analysis OR gate 39<sub>2</sub>, logic 1 appears at its output and at the output of the carry analysis AND gate 24<sub>2</sub>. This results in the appearance of logic 1 at the output of the carry gate 36<sub>2</sub> and, therefore, at the carry output 38<sub>2</sub> of the second one-digit half-adder 16<sub>4</sub>. No carry signals are produced in the fourth, third and first one-digit half-adders 16<sub>1</sub>, 16<sub>3</sub>, and 16<sub>1</sub>, since logic 0's are present on the second primary carry inputs of these one-digit half-adders. As a result, a carry codeword 0010 appears at the carry outputs 38<sub>3</sub>, 38<sub>3</sub>, 38<sub>2</sub>, and 38<sub>1</sub>.

Another example may be considered when Fibonacci codes of an augend 1010 and an addend 1000 are added. In this case, the above description applies to the production of intermediate sum and carry signals in the fourth, third and first one-digit half-adders 16<sub>4</sub>, 16<sub>3</sub>, and 16<sub>1</sub>.

In the second one-digit half-adder 16<sub>2</sub>, logic 1 appears at an input of the addend analysis OR gate 18<sub>2</sub>, thereby resulting in the appearance of logic 1 at the intermediate sum output 40<sub>2</sub>. Since one of the input signals is logic 0, there is logic 0 at the output of the addend analysis AND gate 19<sub>2</sub>, i.e., at the primary carry output of the second one-digit half-adder 16<sub>2</sub> and at the primary carry input 27<sub>1</sub> of the first one-digit half-adder 16<sub>1</sub>. Since logic 0's are present on the inputs 2<sub>1</sub>, 3<sub>1</sub>, then logic 0 appears at the initial sum output 26<sub>1</sub> of the first one-digit half-adder 16<sub>1</sub>. That logic 0, after passing through the inverter 37<sub>2</sub> of the second one-digit half-adder 18<sub>2</sub>, results in the appearance of logic 1 at an input of the secondary carry AND gate 33<sub>2</sub>. Applied to the primary carry input 27<sub>3</sub> of the third one-digit half-adder 16<sub>3</sub> and to the other primary carry input of the second one-digit half-adder 16<sub>2</sub> is logic 1 produced in the fourth one-digit half-adder 16<sub>4</sub> at the output of the addend analysis AND gate similar to the addend analysis AND gate 19<sub>2</sub>. As a result, the output of the carry analysis OR gate 39 produces logic 1 which causes the appearance of logic 1 at the output of the carry analysis AND gate 24<sub>2</sub> and, therefore, at the output of the secondary carry AND gate 33<sub>2</sub> which is the secondary carry output 34<sub>2</sub> of the second one-digit half-adder 16<sub>3</sub>, coupled to an input of the intermediate sum OR gate similar to the intermediate sum OR gate 25<sub>2</sub>. Now, logic 1 from the secondary carry output 34<sub>2</sub> passes to the intermediate sum output 40<sub>1</sub> with the result that codewords 0000 and 1111 representing, respectively, the carry and the intermediate sum appear at the carry and intermediate sum outputs of the n-digit half-adder 1.

Consider a test procedure for the n-digit half-adder 1 when Fibonacci codes 0100 and 0100 are added. The actions performed in

5  
10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65

70  
75  
80  
85  
90  
95  
100  
105  
110  
115  
120  
125  
130

this case are similar to those described above. Therefore, logic 1 present on the primary carry input  $27_2$  passes to an input of the first test signal selection AND gate  $28_2$ . If any one of the inputs  $2_2, 3_2$  accepts erroneously logic 1, then the latter also passes to an input of the first test signal selection AND gate  $28_2$  with the result that its output produces logic 1 delivered to the output of the test OR gate  $29_2$ , that logic 1 being handled as an error signal. If logic 1 appears erroneously at the secondary carry output  $34_4$  of the fourth one-digit half-adder  $16_4$ , or at the primary carry output of the latter, coupled to the primary carry input  $27_3$  and to the carry analysis OR gate  $39_2$ , then logic 1 appears at the output of the latter and passes via the first test signal selection OR gate  $23_2$ , the first test signal selection AND gate  $28_2$  and the carry analysis OR gate  $39_2$  to cause the appearance of logic 1, namely an error signal, at the fault acknowledgement output  $30_2$ .

If logic 1 erroneously appears at the secondary carry output  $34_3$  of the third one-digit half-adder  $16_3$ , that logic 1, after passing through the first test selection OR gate  $23_2$ , the first test signal selection AND gate  $28_2$  and the carry analysis OR gate  $29_2$ , causes the appearance of logic 1, namely, an error signal, at the fault acknowledgement output  $30_2$ .

Note that due to the availability of the inverter 37 and a coupling established between the output of the addend analysis OR gate  $19_2$  and the initial sum output 26, through the initial sum OR gate of the first one-digit half-adder  $16_1$ , similar to the carry analysis AND gate  $24_2$ , logic 1's cannot appear concurrently at the secondary carry output  $34_2$  and even at one of the inputs of the second test signal selection OR gate  $35_2$ . If logic 1's appear, however, due to a hardware check concurrently at the secondary carry output  $34_2$  and even at one of the inputs of the second test signal selection OR gate  $35_2$ , then logic 1 passes through the second test signal selection AND gate  $31_2$  and the test OR gate  $29_2$  to the check acknowledgement output  $30_2$ .

In the case of proper operation of the n-digit half-adder 1 handling Fibonacci codes represented in the minimal form, logic 1's cannot appear concurrently, for example, at the carry outputs  $38_2, 38_3$  of the second and third one-digit half-adders  $16_2, 16_3$ . If these logic 1's appear concurrently due to a hardware check, then the output of the third test signal selection AND gate  $32_2$  produces logic 1 which, after passing through the test OR gate  $29_2$ , appears at the fault acknowledgement output  $30_2$  as an error signal. With the latter present on even one of the fault acknowledgement outputs  $30_1, \dots, 30_n$ , logic 1 appears at the output of the OR gate 17, which constitutes the fault acknowledgement output of the n-digit half-adder 1.

The rewriting device 4 operates as follows.

The codewords representing intermediate sums and carries are delivered from the outputs of the n-digit half-adder 1 (Fig. 1) to the multidigit inputs 6, 7 of the rewriting device 4. Consider the case where a carry codeword 0010 and an intermediate sum codeword 1111 (relating to the first example described above) are rewritten. A rewrite signal is applied to the control input 5 whereas the codewords representing the augend and the addend are applied to the multidigit inputs 2, 3. After the delay time  $\tau$ , which considerably exceeds the time required for the production of the values of two high-order positions in the fourth and third one-digit half-adders  $16_3, 16_4$  (Fig. 2) has elapsed, the output of the delay 42 (Fig. 3) and, therefore, respective inputs of the AND gates  $43_2, 44_2, 45_2, 46_2$  accept a control signal. After signals 00 and 11 belonging to the high-order positions of the intermediate sum and the carry have been formed, they are passed to the one-digit inputs  $6_4, 6_3$  and  $7_4, 7_3$  of the rewriting cell  $41_2$  and, therefore, to respective inputs of the AND gates  $43_2, 44_2, 45_2$  and  $46_2$ . After the delay time  $\tau$  has elapsed, other inputs of the AND gates  $43_2, \dots, 46_2$  accept logic 1 from the output of the delay  $42_2$  with the result that input data is precisely reproduced at the outputs of the AND gates  $43_2, \dots, 46_2$ . The same logic 1 from the output of the delay  $42_2$  is used to activate the delay  $42_1$  whose output produces logic 1, the delay time  $\tau$  later, and carry signal 10 and intermediate sum signal 11 belonging to low-order positions are rewritten. Thus, the appearance of data at the outputs of the rewriting cells  $41_2$  and  $41_1$  is delayed by the delay time  $\tau$ . Logic 1 from the output of the delay  $42_1$ , after a  $2\tau$  time interval has elapsed since a moment when a respective signal is applied to the control input 5, appears at the control input of the rewriting device 4 and, therefore, at the control input 11 (Fig. 1) of the Fibonacci code converter 8. With the intermediate sum signal 00 and the carry signal 11 applied, respectively, to the inputs  $9_4, 9_3$  and  $10_4, 10_3$  of the conversion cells  $47_4$  and  $47_3$  (Fig. 4) of the Fibonacci code converter 8, said cells begin to convert the codewords to the Fibonacci code minimal form. The conversion consists in performing the convolution operation which involves the high-order (third and fourth) positions of the intermediate sum codeword and the fifth position, corresponding to the fifth conversion cell  $47_5$ , as follows: 0 1 1.

The sign  $\uparrow \text{---} \text{---} \text{---}$  is used to designate the convolution operation. When the "1" set inputs of the flip-flops  $56_2$  and  $56_3$  accept logic 1's the "1" outputs of these flip-flops produce logic 1's which pass through the convolution AND gates  $63_4$  and the OR gate  $62_4$  of the conversion cell  $47_4$  and through the convolution AND gate  $63_3$  and the OR gate

61<sub>3</sub> of the conversion cell 47<sub>3</sub> to the inputs 61<sub>4</sub> and 61<sub>3</sub> of the analysis AND gates 59<sub>4</sub> and 59<sub>3</sub>. The inputs 60<sub>4</sub>, 60<sub>3</sub> of the latter accept logic 1's from the "0" outputs of the flip-flops 57<sub>4</sub>, 57<sub>3</sub> with the result that the outputs of the analysis AND gates 59<sub>4</sub>, 59<sub>3</sub> produce logic 1's applied to the "0" inputs 58<sub>4</sub>, 58<sub>3</sub> of the flip-flops 56<sub>4</sub>, 56<sub>3</sub> so that the latter take up their "0" states. Logic 1 from the output of the convolution AND gate 63<sub>4</sub> comes to the counting input of the flip-flop 56<sub>5</sub> of the fifth conversion cell 47<sub>5</sub>, is stored in that flip-flop, then passes through the transient analysis OR gate 51 and is converted by the inverter 52 to logic 0 which inhibits the appearance of logic 1 at the output of the end-of-addition AND gate 54, i.e., at the end-of-addition denote signal output 14. After the convolution of two high-order positions 11 of the intermediate sum codeword has been completed the values 0 1 0 0 are stored, respectively, in the flip-flops 66, 56<sub>5</sub>, 56<sub>4</sub>, 56<sub>3</sub>, these values being applied to the one-digit sum inputs 13<sub>6</sub>, 13<sub>5</sub>, 13<sub>4</sub>, and 13<sub>3</sub>. After the delay time  $\tau$  has elapsed since a moment when the intermediate sum and carry signals are applied to the inputs 9<sub>4</sub>, 9<sub>3</sub>, 10<sub>4</sub>, 10<sub>3</sub>, the inputs 9<sub>2</sub>, 9<sub>1</sub> accept signals corresponding to two remaining positions of the intermediate sum code, i.e., 11, while the inputs 10<sub>2</sub>, 10<sub>1</sub> accept signals corresponding to two remaining positions of the carry code, i.e., 10. At the same time, the end-of-rewriting denote logic 1 is passed from the output of the rewriting device 4 (Fig. 1) to the control input 11 of the Fibonacci code converter 8. That logic 1 causes the end-of-rewriting flip-flop 55 (Fig. 4) to take up its "1" state, thereby resulting in the appearance of logic 1 at one of the inputs of the end-of-addition AND gate 54. On arrival of the intermediate sum and carry signals, the flip-flops 56<sub>2</sub>, 57<sub>2</sub>, 56<sub>1</sub> assume their "1" states, while the flip-flop 57<sub>1</sub> is held in its "0" state. The input 60<sub>2</sub> of the analysis AND gate 59<sub>2</sub> receives logic 0 from the "0" output of the flip-flop 57<sub>2</sub> which inhibits the appearance of logic 1 at the output of the analysis AND gate 59. Since the flip-flop 56<sub>3</sub> is held in its "0" state, its "0" output and, therefore, the output 64<sub>2</sub> and a respective input of the convolution AND gate 63<sub>2</sub> as well accept logic 1's. Since logic 1's from the "1" outputs of the flip-flops 56<sub>2</sub>, 56<sub>1</sub> comes to the remaining inputs of the convolution AND gate 63<sub>2</sub>, the output of the latter produces logic 1 which passes through the OR gate 62<sub>2</sub> to the "0" set input of the flip-flop 57<sub>2</sub> so as to send the latter into "0" state. Logic 1 from the output of the convolution AND gate 63<sub>2</sub> also comes to the input of the OR gate 62<sub>1</sub> of the conversion cell 47<sub>1</sub> and, from the output of the OR gate 62<sub>2</sub>, to one input of the analysis AND gate 59<sub>1</sub> whose other input accepts logic 1 from the "0" output of the flip-flop 57<sub>1</sub>. As a result,

logic 1 appears at the output of the analysis gate 59<sub>1</sub> and passes to the "0" set input of the flip-flop 56<sub>1</sub> so as to place the latter in its "0" state. Logic 1 from the first communication output 50<sub>2</sub> of the second conversion cell 47<sub>2</sub> comes to the counting input of the flip-flop 56<sub>3</sub> of the third conversion cell 47<sub>3</sub> and causes the flip-flop 56<sub>3</sub> to take up its "1" state. The same logic 1 also comes to a respective input of the transient analysis OR gate 51 and to the inverter 52 with the result that logic 0 appears at the output of the latter. That logic 0 comes to the end-of-addition AND gate 54 so that no signal is present at the end-of-addition denote signal output 14. During this processing step, the flip-flops 66 and 56<sub>5</sub>—56<sub>1</sub> receive the codeword 0 1 0 1 1 0 and the flip-flops 57<sub>5</sub>—57<sub>1</sub> receive the codeword 0 0 0 0 0, thereby resulting in the appearance of logic 1 at the inputs 60<sub>1</sub>—60<sub>5</sub> of the analysis AND gates 59<sub>1</sub>—59<sub>5</sub>. The codeword written in the flip-flops 56<sub>4</sub>, 56<sub>3</sub>, 56<sub>2</sub> of the conversion cells 47<sub>4</sub>, 47<sub>3</sub>, 47<sub>2</sub> is represented as 0 1 1, a form which differs

from the Fibonacci code minimal form. Therefore, the convolution condition for these positions is satisfied. The convolution operation similar to that described above is accomplished with the result that the codeword 0 1 1 0 0 0 is stored in the flip-flops 66 and 56<sub>5</sub>—56<sub>1</sub>. Again the convolution condition for the fifth, fourth and sixth positions is satisfied and the flip-flop 66 with its "1" set input coupled to the first communication output 50<sub>5</sub> of the fifth conversion cell 47<sub>5</sub> corresponds in this case to the sixth position. The convolution is performed same as described above and the codeword 1 0 0 0 0 0 appears which corresponds to the Fibonacci code minimal form one of the final sum. As a rule, logic 1 is present on at least one of the first communication outputs 50<sub>1</sub>, 50<sub>2</sub>, 50<sub>3</sub>, 50<sub>4</sub>, and 50<sub>5</sub>, that logic 1 being applied to the input of the inverter 52 through the transient analysis OR gate 51. Therefore, logic 0 to inhibit the appearance of logic 1 at the end-of-addition denote signal output 14 is always present during convolution on the input of the end-of-addition AND gate 54. After the conversion is terminated, which means that all convolutions are performed, logic 0's appear at the first communication outputs 50<sub>1</sub>—50<sub>5</sub> with the result that the end-of-addition denote signal is produced at the end-of-addition denote signal output 14 to enable data to be read off the multidigit message output 13 (Fig. 1).

When the Fibonacci code adder is checked for serviceability, one of its proper operation conditions requires that a carry logic 0 be always present in the presence of an intermediate sum logic 0 in one and the same position. If this condition is not satisfied, then logic 1's appear at the inputs of the test

AND gates 65<sub>1</sub>—65<sub>5</sub> (Fig. 4), coupled to the "0" outputs of the flip-flops 57<sub>1</sub>—57<sub>5</sub>. This results in the appearance of logic 1's at the fault acknowledgement outputs 48<sub>1</sub>—48<sub>5</sub> of the conversion cells 47<sub>1</sub>—47<sub>5</sub>. These signals pass through the test OR gate 49 of the Fibonacci code converter 8 and through the test OR gate 67 of the Fibonacci code adder to the check acknowledgement output 15 of the Fibonacci code adder to indicate improper operation of the latter.

The Fibonacci code converter 8 can be checked for proper convolution operation as follows. If at least one of the first communication outputs 50<sub>1</sub>—50<sub>5</sub> generates a continuous signal whose length exceeds a maximum time required for the execution of all convolutions, then that signal passes through the transient analysis OR gate 51 to the input of the electrical filter 53 and appears at the output of the latter and, therefore, at the fault acknowledgement output 15 after a time interval determined by the time constant of the electrical filter 53 has elapsed.

The present invention provides for an increased speed of the Fibonacci code adder due to the fact that a large number of add microcycles are excluded and the entire addition process is reduced to two operations as follows: the production of the intermediate sum and carry and the conversion of all codewords representing them to the codeword of the final sum in the Fibonacci code minimal form. The speed of the Fibonacci code adder is also increased since the production of the intermediate sum and carry and the conversion of respective Fibonacci codes are performed in a parallel mode due to the availability of the rewriting device 4.

The Fibonacci code adder proposed features an increased hardware controllability provided by an increased number of test relationships available to each one-digit half-adder of the n-digit half-adder 1 and also resulted from the fact that one can check operations relating to the writing of data into the Fibonacci code converter and to the convolution of bits in the latter. An increase in the speed of the Fibonacci code adder is also attained as a result of the notifying of a moment when addition is terminated, thereby lessening the mean time required for the summation of a number of codewords.

#### WHAT WE CLAIM IS:—

1. A Fibonacci code adder comprising an n-digit half-adder having at least multidigit augend input and a multidigit addend input to accept numbers represented in the minimal form of Fibonacci codes and having its multidigit intermediate sum and carry outputs coupled, respectively, to a multidigit intermediate sum input and to a multidigit carry input of a rewriting device having its multidigit intermediate sum and carry outputs and

its control output coupled, respectively, to a multidigit intermediate sum input, a multidigit carry input and a control input of a Fibonacci code converter, a fault acknowledgement output of the n-digit half-adder being coupled to a fault acknowledgement input of the Fibonacci code converter, where n is the length of the Fibonacci code.

2. A Fibonacci code adder as claimed in Claim 1, wherein the n-digit half-adder comprises n identical one-digit half-adders each having its fault acknowledgement output coupled to a respective input of an OR gate of the n-digit half-adder, a primary carry output of an ith one-digit half-adder being coupled to one primary carry input of an (i-1)th one-digit half-adder and to the other primary carry input of an (i-2)th one-digit half-adder, a secondary carry output of the ith one-digit half-adder being coupled to one secondary carry output of the (i-1)th one-digit half-adder and to the other secondary carry input of the (i-2)th one-digit half-adder, a carry output and an initial sum input of the ith one-digit half-adder being coupled, respectively, to the carry input of the (i-1)th one-digit half-adder and to the initial sum input of an (i+1)th one-digit half-adder, where  $i=1, 2, \dots, n$ .

3. A Fibonacci code adder as claimed in claim 2, wherein each one-digit half-adder comprises OR gates, AND gates and an inverter, first and second inputs of an addend analysis OR gate being coupled, respectively, to first and second inputs of an addend analysis OR gate, an output of the addend analysis OR gate being coupled to first inputs of an initial sum OR gate, a first test signal selection OR gate and a carry analysis AND gate, the output of the initial sum OR gate being coupled to a first input of an intermediate sum OR gate, a second input of the initial sum OR gate being coupled to a first input of a first test signal selection AND gate having its second input coupled to the output of the first test signal selection OR gate and having its output coupled to a first input of a test OR gate having two other inputs coupled, respectively, to the output of a second test signal selection AND gate and to the output of a third test signal selection AND gate, a first input of the second test signal selection AND gate being coupled to the output of a secondary carry AND gate, a second input of the second test signal selection AND gate being coupled to the output of a second test signal selection OR gate having its first input coupled to a first input of a carry AND gate and to the input of the inverter, having its second input coupled to a first input of the third test signal selection AND gate and to the output of the carry AND gate and having its third input coupled to the output of the addend analysis AND gate, one of two other inputs of the intermediate sum OR gate being coupled to

one of two other inputs of the first test signal selection OR gate, the remaining input of the intermediate sum OR gate being coupled to a second input of the carry analysis AND gate, to the remaining input of the first test signal selection OR gate and to the output of a carry analysis OR gate, the output of the carry analysis AND gate being coupled to a second input of the carry AND gate and to a first input of the secondary carry AND gate whose second input is coupled to the output of the inverter.

4. A Fibonacci code adder as claimed in claim 1, wherein the rewriting device with a control input comprises k identical rewriting cells, each rewriting cell being provided with a delay having its output coupled to a first input of a high-order intermediate sum AND gate, to a first input of a high-order carry AND gate, to a first input of a low-order intermediate sum AND gate and to a low-order carry AND gate, second inputs of the high- and low-order intermediate sum AND gates of all rewriting cells being used as the multidigit intermediate sum input of the rewriting device, second inputs of the high- and low-order carry AND gates being used as the multidigit carry input of the rewriting device, the outputs of the high- and low-order intermediate sum AND gates of all rewriting cells being used as a multidigit intermediate sum output of the rewriting device, the outputs of the high- and low-order carry AND gates of all rewriting cells being used as a multidigit carry output of the rewriting device, the input of the delay of an mth rewriting cell being coupled to the output of a delay of an (m+1)th cell of the rewriting device, the input of a delay of a kth cell being used as the control input of the rewriting device, and the output of a delay of a first cell being used as the control output of the rewriting device, where

$$k \left\{ \begin{array}{l} \frac{n}{2} \text{ at an even } n \\ \frac{n+1}{2} \text{ at an odd } n, \text{ and} \end{array} \right.$$

45  $m=1, 2 \dots k.$

50 5. A Fibonacci code adder as claimed in claim 1, further comprising a Fibonacci code converter incorporating at least n identical conversion cells and a store flip-flop, fault acknowledgement outputs of the conversion cells being coupled to respective inputs of a test OR gate of the Fibonacci code converter, the output of the test OR gate being coupled

to a first input of a test OR gate of the Fibonacci code adder, a first communication output of each conversion cell being coupled to a first input of a transient analysis OR gate whose output is coupled to the input of an inverter and to the input of an electrical filter having its output coupled to the remaining input of the test OR gate of the Fibonacci code converter the output of the inverter being coupled to a first input of an end-of-addition AND gate whose output is used as an end-of-addition denote signal output of the Fibonacci code adder, a second input of the end-of-addition AND gate being coupled to the "1" output of an end-of-rewriting flip-flop, a first communication input of the ith conversion cell being coupled to a one-digit message output of an (i-1)th conversion cell, a second communication input of the ith conversion cell being coupled to a second communication output of an (i+1)th conversion cell, a first communication output of the ith conversion cell being coupled to a third communication input of the (i+1)th conversion cell and to a fourth communication input of the (i-1)th conversion cell, and a first communication output of a high-order conversion cell being coupled to the "1" set input of the store flip-flop.

6. A Fibonacci code adder as claimed in claim 5, wherein each conversion cell comprises an intermediate sum flop-flop and a carry flip-flop whose "1" set inputs are used as message inputs of the conversion cell, the "0" set input of the intermediate sum flop-flop being coupled to the output of an analysis AND gate having its input coupled to the "0" output of the carry flip-flop and having its another input coupled to the "0" set input of the carry flip-flop and to the output of an OR gate of the conversion cell, a first input of the OR gate being coupled to the output of a convolution AND gate having its input coupled to the "1" output of the intermediate sum flop-flop whose "0" output is coupled to a first input of a test AND gate of the conversion cell, whose first output is used as a fault acknowledgement output of the conversion cell and whose second output is coupled to the "1" output of the carry flip-flop, and the counting input of the intermediate sum flop-flop being used as a third communication input of the conversion cell.

7. A Fibonacci code adder as claimed in claim 1, substantially as hereinbefore described with reference to the accompanying drawings.

MARKS & CLERK,  
Chartered Patent Agents,  
57-60 Lincoln's Inn Fields,  
London, WC2A 3LS.  
Agents for the Applicant(s).

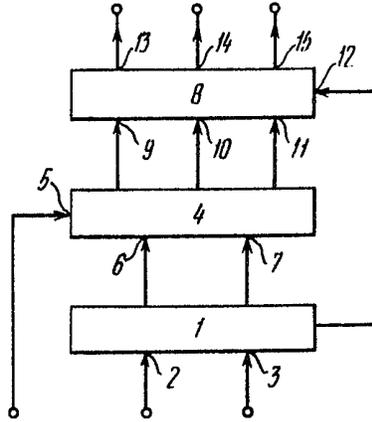


FIG. 1

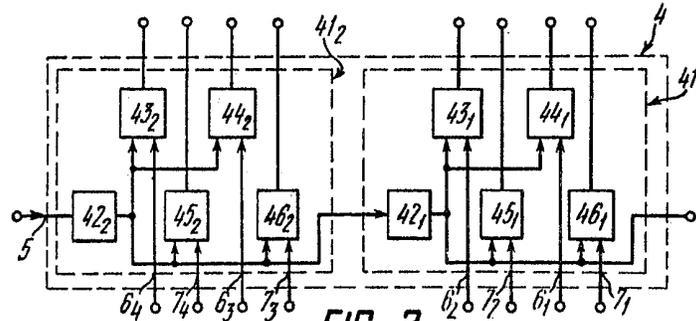


FIG. 3

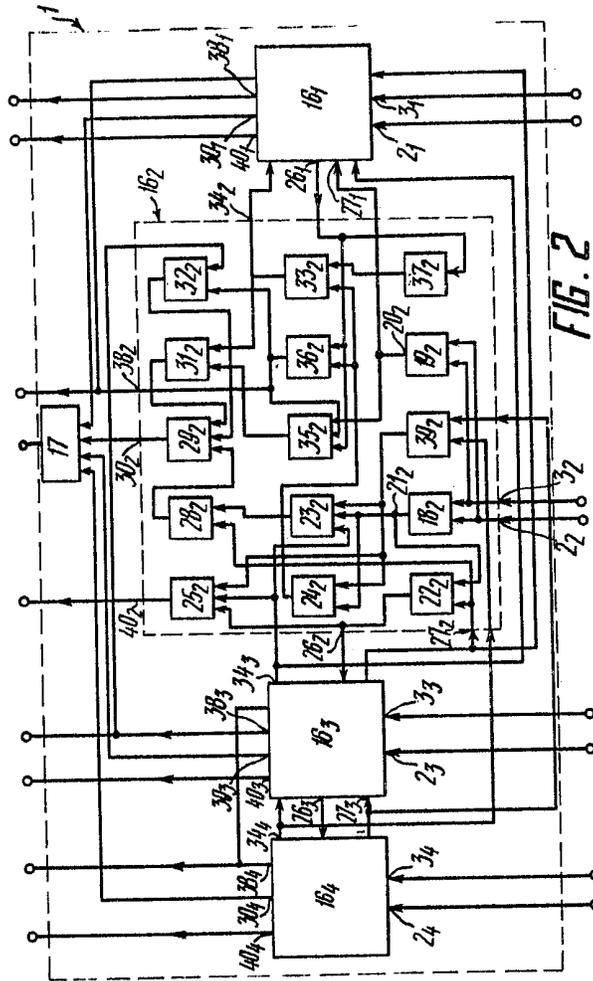


FIG. 2

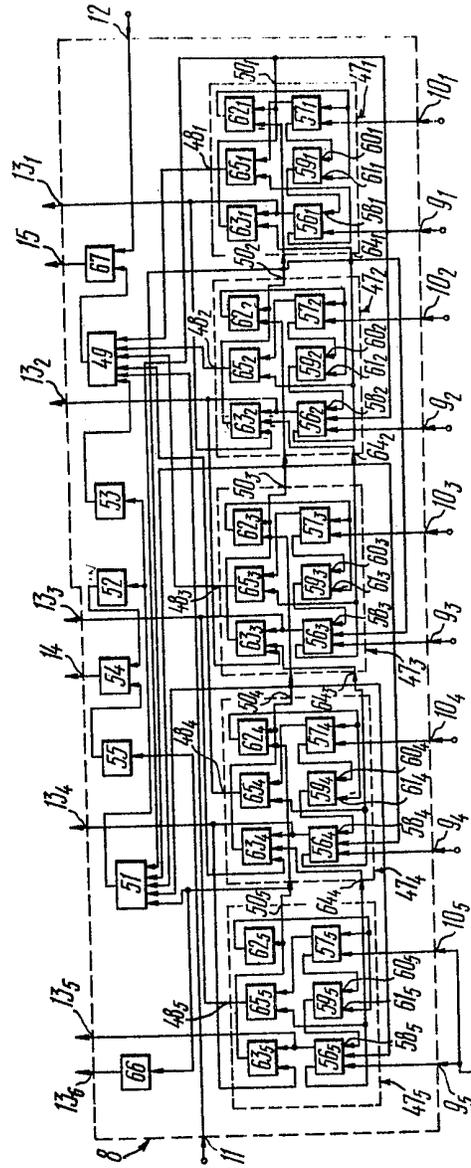


FIG. 4