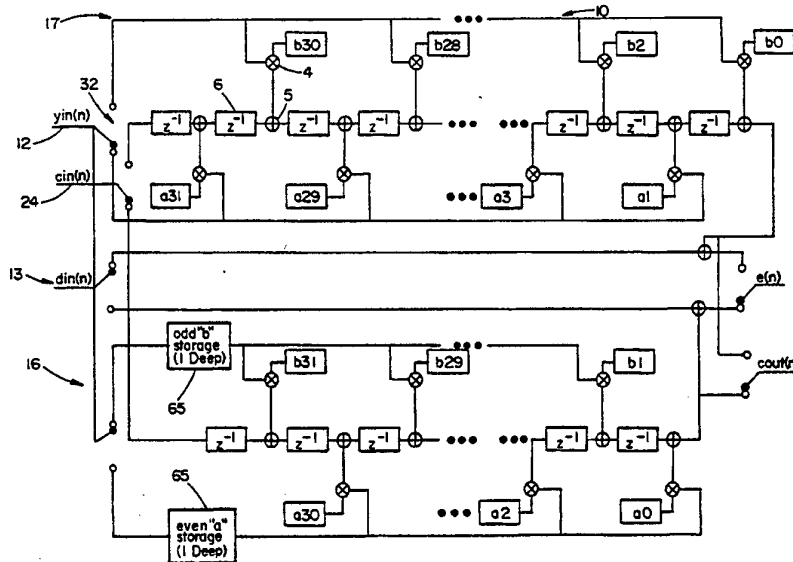




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification <sup>5</sup> : <b>G06F 15/31</b></p>	<p><b>A1</b></p>	<p>(11) International Publication Number: <b>WO 94/28499</b> (43) International Publication Date: 8 December 1994 (08.12.94)</p>
<p>(21) International Application Number: <b>PCT/US94/05831</b> (22) International Filing Date: <b>24 May 1994 (24.05.94)</b> (30) Priority Data: 068,932                      28 May 1993 (28.05.93)                      US (71) Applicant: <b>GRUMMAN CORPORATION [US/US]; M.S. A25-GHQ, Bethpage, NY 11714-3586 (US).</b> (72) Inventors: <b>WEDGWOOD, Janet, E.; 30 Lafayette Avenue, Bethpage, NY 11714 (US). PETRSORIC, John, F.; 1001 N. 4th Street, New Hyde Park, NY 11040 (US).</b> (74) Agents: <b>DiGIGLIO, Frank, S. et al.; Scully, Scott, Murphy &amp; Presser, 400 Garden City Plaza, Garden City, NY 11530 (US).</b></p>		<p>(81) Designated States: European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).  <b>Published</b> <i>With international search report.</i></p>

(54) Title: **COMPLEX ADAPTIVE FIR FILTER**



(57) Abstract

A complex adaptive Finite-Impulse-Response filter (10) for processing complex digital data having real and imaginary data portions that includes a first data path (17) for receiving and processing the real data portion of the complex input and a second data path (16) for receiving and processing the imaginary data portion. Each first and second data path includes a corresponding plurality of adaptive weight circuits (4) and accumulator circuits (5) to perform respectively, multiplications of input data with a set of updatable coefficient data and accumulations of the multiplication results obtained. Two sets of multiplication results are obtained for each data path and a subset from each are accumulated to form a real output portion and imaginary output portion for the filter. A time delay circuit (6) is provided to allow for proper time multiplexing of the imaginary output portion. The plurality of adaptive weight circuits have associated therewith first and second memory storage banks (24, 27) for storing the current coefficient data values.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LU	Luxembourg	SN	Senegal
CN	China	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				



-2-

1 having a fast array multiplier and accumulator may  
execute the single multiply-accumulate operation in a  
single clock cycle.

The DSP chip may be utilized as an  
5 application-specific device that can perform one  
function more accurately, faster, and more cost-  
effectively. For example, an increasing number of  
digital filter and Fourier transform applications  
require DSP chips. A special type of digital filter, a  
10 Finite-Impulse-Response ("FIR") filter, i.e., a filter  
that is characterized as having an impulse response of  
finite length, is ideally used to achieve exact linear  
(or near linear) phase response. Ideally this is used  
in systems where it is paramount that the phase affects  
15 the shape of the output of the FIR filter by, at most, a  
time delay. In radar systems incorporating phased-array  
antennas, for example, digital FIR filters are required  
to process complex digital signals, i.e., signals having  
real and imaginary components. Usually, the filter  
20 functions to get rid of noise caused by antenna mismatch  
or clutter. When filtering complex digital signals, the  
real and imaginary parts of the digital signal and their  
time delayed versions are each multiplied by a complex  
weight and the results are then summed. To perform the  
25 filtering function, both real and imaginary parts of the  
weight and the negative of the imaginary part of the  
weight, must be made available as multiplicands  
(coefficients). Most digital filters are thus  
conveniently implemented using DSP chips having  
30 coefficient multiplier elements, adders or summing  
devices, and delay elements (memory storage devices).  
The filter is considered "adaptive" when the

1 coefficients are variable and updated by a host  
processor that implements an updating algorithm.

There are presently available integrated  
circuit chips that are used for filtering digital  
5 signals. For instance, the INMOS signal processing chip  
IMSA100 is able to do up to 32 parallel multiplies and  
additions (or subtractions) and that makes it useful as  
a FIR filter. This particular DSP filter is designed  
for use in a microprocessor system, so that the  
10 coefficients may be loaded through an input port to one  
or two banks of registers or alternatively, to external  
ROM or RAM. When filtering real data, one bank of  
memory is used to provide the coefficients for the  
filter. For adaptive filtering, one bank of memory is  
15 used while the other is being updated with new  
coefficients. For complex adaptive filtering however,  
the two bank architecture becomes unusable, even at low  
data rates. This is because both banks of coefficients  
are required to produce the filter output, i.e., one  
20 bank produces the real output and the other produces the  
imaginary output. To update the coefficients, the  
filtering process has to be stopped while the new  
coefficients are being loaded in. Usually, the time  
required to load new coefficients is long compared to  
25 the data rate of most existing processors. In fact,  
most existing processors can do complex processing at  
only one-half the data rate.

To alleviate excessive delays in updating the  
coefficients, alternative designs have been implemented  
30 where the coefficients are accessed circularly from an  
external memory. Using appropriate control circuitry,  
one bank of memory can provide the coefficients to the

-4-

1 filter while the other bank of memory is being updated.  
This arrangement relieves the updating problem but it  
does not take full advantage of the geometry of the  
complex filter and requires a significant amount of  
5 external control circuitry.

Additionally, in order to properly perform  
adaptive filtering, an error signal, which is calculated  
as the sum of the filter output signal and a "desired"  
input signal, must be provided as an output to the host  
10 processor to calculate the updated weight coefficients.  
Present DSP processors do not provide this capability  
and external adder circuitry is usually required to  
calculate the error for updating the weights. The need  
for an external adder and corresponding control  
15 circuitry degrades filter performance and takes up  
valuable circuit board area.

Most of the available FIR filter chips also  
require the need to fix and truncate the floating point  
result coming out of the host processor. This is due to  
20 the fact that the internal processing is not floating  
point. One consequence of this, is that the dynamic  
range usually required for adaptive processing, is not  
preserved.

In view of the foregoing limitations of the  
25 present generation of adaptive FIR filters, there exists  
the need for a complex adaptive FIR filter that can  
process complex digital signals more quickly and  
efficiently when the signal to be processed is a complex  
digital data and when the processing speed requirements  
30 of the data rate make commercially available FIR filter  
chips functionally prohibitive.

1           Therefore, it is the objective of the instant  
invention to provide a complex adaptive FIR filter  
having an internal processing architecture that features  
two parallel data paths, one which filters the real data  
5 stream of the complex signal and another which filters  
the complex data stream whereby the architecture takes  
maximum advantage of the time available to do the  
required calculations.

          It is a further objective to provide a complex  
10 adaptive FIR filter that is able to do complex adaptive  
digital filtering without having to interrupt or stop  
the filtering process when the weight coefficients are  
being updated.

          It is another object of the invention to  
15 provide a complex adaptive FIR filter having an input  
for receiving a "desired" input signal and an adder  
element for enabling the generation of an error signal  
for more efficient adaptive filter implementation.

          Another objective of the present invention is  
20 to provide a complex adaptive FIR filter that  
incorporates floating-point processing to preserve the  
dynamic range required for adaptive processing.

          It is still another objective to provide a  
complex adaptive FIR filter that is designed as a  
25 standalone device that requires no external circuitry to  
interface to a host processor.

          Still another objective is to provide a  
complex adaptive FIR filter that can be cascaded to  
allow the creation of longer filters for applying longer  
30 coefficient sets to the data stream.

          Yet another objective of the present invention  
is to provide a complex adaptive FIR filter that

-6-

1 provides a unique and efficient way to update the weight  
coefficients without having to provide significant  
amounts of hardware support.

The present invention is directed to a complex  
5 Finite-Impulse-Response filter with adaptive weights for  
processing complex digital data having real and  
imaginary input data portions. The filter comprises a  
first data path for processing the real input data  
portion of the input complex data. The first data path  
10 includes a first plurality of adaptive weight circuits  
for multiplying the real input data by predetermined  
coefficients to generate a set of multiplication  
results. A second data path is also provided for  
processing the imaginary input data portion of the input  
15 complex data. The second path includes a second  
plurality of adaptive weight circuits for multiplying  
the imaginary input data by predetermined coefficients  
to generate a set of multiplication results. The filter  
is also provided with a control circuit that provides  
20 the real input data simultaneously to the first  
plurality of adaptive weight circuits during a first  
time interval and successive first time intervals  
thereafter, and provides the imaginary input data  
simultaneously to the second plurality of adaptive  
25 weight circuit during a second time interval and  
successive second time intervals thereafter. A first  
plurality of accumulator circuits interconnect a first  
subset of the first plurality of adaptive weight  
circuits and a first subset of the second plurality of  
30 adaptive weight circuit to accumulate a first set of  
multiplication results therefrom to form a real data  
output portion for the filter. A one clock cycle delay

35

-7-

1 circuit located in said first data path delays the  
multiplication of the real input data with a second  
subset of the first plurality of adaptive weight  
circuits by one time interval and a one clock cycle  
5 delay circuit located in the second data path delays the  
multiplication of the imaginary input data with a second  
subset of the second plurality of adaptive weight  
circuits. These first and second delay circuits ensure  
proper time multiplexing of the imaginary data output  
10 portion of the filter. A second plurality of  
accumulator circuits interconnect the second subset of  
the first plurality of adaptive weight circuits and the  
second subset of the second plurality of adaptive weight  
circuits to accumulate a second set of multiplication  
15 results therefrom to form the imaginary output data  
portion for the filter. The real and imaginary output  
data portions are multiplexed at the output of the  
filter to form the complex output signals.

The coefficient data values of the adaptive  
20 weight circuits are stored in two memory storage banks  
for access by the filter or host processor. These  
values can also be stored in alternative memory storage  
banks so that when data is accessed by the filter from  
the memory storage bank, data can be written to the  
25 alternate memory storage bank at the request of the host  
processor and vice versa. Continuous filter operation is  
assured by a coefficient swapping circuit that will only  
allow access to the updated predetermined coefficient  
values of the first subset of the first plurality and  
30 the second subset of the second plurality of adaptive  
weights during second time intervals. Likewise, the  
coefficient swapping circuit will only allow access to

1 the predetermined coefficient values of the first subset  
of the second plurality and the second subset of the  
first plurality of adaptive weights during first time  
intervals.

5 Further benefits and advantages of the  
invention will become apparent from a consideration of  
the following detailed description given with reference  
to the accompanying drawings, which specify and show  
preferred embodiments of the invention.

10 Figure 1 is a block diagram of a system  
implementing the adaptive FIR filter of the present  
invention.

Figure 2 shows a general block diagram  
conceptually illustrating the adaptive filter having two  
15 sets of parallel paths; one set for processing the real  
data portion of the input signal and the other set for  
processing the imaginary portion of the input signal.

Figure 3 shows a typical application for an  
adaptive filter which can be implemented using the  
20 filter of the present invention.

Figure 4 shows a timing diagram for the input  
signals and the internal control sequences of the  
complex adaptive FIR filter.

Figure 5 shows the block diagram  
25 representation of the filter when designed as a 3  
complex weight filter.

Figure 6a illustrates the tabulated results of  
the calculations obtained when processing the real data  
portion of a complex digital signal input to the filter  
30 of Figure 5.

Figure 6b illustrates the tabulated results of  
the calculations obtained when processing the imaginary

1 data portion of a complex digital signal input to the filter of Figure 5.

Figure 7a shows a timing diagram of the signals utilized during the weight coefficient updating 5 (i.e., weight swapping) process when implemented in the adaptive filter of the present invention.

Figure 7b shows the logic truth table and its corresponding legend setting forth the signal conditions in the weight swapping circuitry necessary to perform 10 weight swapping function.

Figure 7c shows the digital logic circuit implemented to command the weight swapping to occur.

Figure 8 shows a generalized block diagram of the circuitry that determines which memory bank is being 15 written to and which memory bank is receiving the updated coefficients.

Figure 9 is a pin-out diagram summarizing the Input/Output of the complex adaptive FIR filter of the present invention.

20 The present invention relates to a finite impulse response filter designed to process complex data. The filter comprises two parallel data paths, one which filters the real data stream and a second which filters the imaginary data stream. The filter processes 25 complex digital signal by multiplying the signals and time delayed versions of the signals each by a complex weight and summing the results of these multiplications. Since the filter utilizes two parallel data paths, both the real and imaginary part of the complex weight, as 30 well as the negative of the imaginary part of the weight must be available as multiplicands which are hereinafter referred to as coefficients. The coefficients are

-10-

1 variable, therefore, the filter is capable of operation  
as an adaptive filter when interfaced to a host  
processor which executes the adaptive algorithm and  
updates the coefficients. Due to the particular  
5 geometry of complex digital signal processing, these  
parallel data paths take maximum advantage of the time  
available to do the required filtering calculations.  
The filter of the present invention also includes a  
"desired input" which eliminates the need for an  
10 external summing device to calculate the error term  
often needed to update the coefficients. Fig. 3  
illustrates an application for the adaptive filter of  
the present invention which eliminates the need for  
external summing device 71.

15 Fig. 1, shows a block diagram of the complex  
adaptive finite impulse response filter 10 of the  
present invention. Correspondingly, Fig. 9 shows a pin-  
out summary of the I/O for the filter 10. Referring to  
Fig. 1, the complex digital signal being processed by  
20 the filter 10,  $Y_{in}(n)$ , is input to the filter 10 via  
reference input signal line 12.  $Y_{in}(n)$  is a multiplexed  
digital signal having a real input data portion and  
imaginary input data portion. It can be, for e.g., a  
pulsed radar return signal that has been sampled at a  
25  $X$ MHz clock rate, separated into its real and imaginary  
components, and time multiplexed. The reference input  
signal line 12 is connected to a first switching circuit  
32 which provides  $Y_{in}(n)$  to both the real processing  
circuit 17 and the imaginary processing circuit 16 in  
30 either position. For instance, in one position,  $Y_{in}(n)$   
is multiplied with adaptive weights having real  
coefficient data in both the real and imaginary

-11-

1 processing circuits 17 and 16, and in the second  
position,  $Yin(n)$  is multiplied with adaptive weights  
having imaginary coefficient data in both the real and  
imaginary processing circuits 17 and 16. It should be  
5 understood that the inputs  $Yin(n)$ ,  $Din(n)$ ,  $Cin(n)$  and  
the output signals  $Cout(n)$  and  $E(n)$  are complex digital  
data signals having real and imaginary parts. These  
inputs are a multiplexed stream of data with the real  
input data portion input first and the imaginary input  
10 data portion input next. Switching circuits 32, 33, and  
34 are essentially demultiplexers which provide the real  
and imaginary input data portions to the respective real  
data and imaginary data processing circuits. The  
switching circuits are all shown in the Fig. 1 as single  
15 throw double pole switches, but it should be understood  
that any circuitry which switches a signal between two  
inputs on every clock can be used for the demultiplexing  
function. Likewise, switching circuits 35 and 36  
function as multiplexers that form a continuous stream  
20 of real and imaginary parts of a complex signal along a  
single data output line.

The desired input signal  $Din(n)$ , is input to  
the filter 10 via desired input signal line 13. The  
desired input signal line 13 is connected to a second  
25 switching circuit 33 which provides  $Din(n)$  to either a  
first summing circuit 20 or a second summing circuit 22.  
The real data portion of the desired input signal  $Din(n)$   
is added to the real portion of the output signal  
 $Cout(n)$  when the second switching circuit 33 is switched  
30 to route  $Din(n)$  to the first summing circuit 20. The  
imaginary data portion of  $Din(n)$  is added to the  
imaginary portion of the output signal  $Cout(n)$  when the

-12-

1 second switching circuit 33 routes  $Din(n)$  to the second  
summing circuit 22. In the preferred embodiment, the  
Din(n) input provided to the respective summing circuits  
is switched in at the XMHz clock rate however, it must  
5 be externally delayed depending upon the length of the  
filter i.e., the number of weights, in order to account  
for the latency of the filter 10.

A write strobe or write enable signal,  $Wr$ , is  
input to the filter 10 via a write strobe signal line  
10 14. The write enable signal,  $Wr$ , is transmitted from  
the host processor (not shown) and functions to latch  
the coefficients into the respective  
coefficient/coefficient update memory banks 24 and 27  
during the coefficient updating process to be described  
15 in detail below.

Address lines  $Adr(2:\emptyset)$ , via signal lines 15,  
are the address lines which access the individual  
coefficient registers in memory banks 24 and 27. These  
are transmitted by a host processor and input to address  
20 decode circuit 28 where they are decoded to obtain six  
 $REGSELECT(5:\emptyset)$  signals, 44, that address the individual  
coefficient registers in the coefficient/coefficient  
update memory banks 24 and 27. Signal  $Ce$ , input to  
filter 10 via signal line 18, is used to gate the  
25 address decoder 28 to make the filter 10 look like a  
memory-mapped peripheral to a host processor.

The clock input to the filter 10 is labelled  
XMHz and is shown in Fig. 1 input line 19. This clock  
controls the filter and runs at the real data rate which  
30 is two times the complex data rate. As will be  
described in detail below with reference to the timing  
diagram of Fig. 4, the control signals 40, 42 that are

-13-

1 used to control the flow of data into the real and  
imaginary processing circuits, are generated by control  
circuitry 30. Control circuitry 30, it will be  
appreciated, also provides the signals that control the  
5 switching of input switching circuits 32, 33, and 34 and  
output switching circuits 35 and 36.

The signal WSWAPREQ that is input to the  
filter via line 21 is the weight swap request signal and  
it provides an indication to the weight swapping  
10 circuitry 38 that a change of banks is required.

The DATAIN signal input via line 23 provides  
the floating point coefficient data to the coefficient  
registers addressed by  $adr(2:\phi)$  and  $REGSELECT(5:\phi)$ .  
DATAIN, 23 may comprise a 32-bit bus. Both DATAIN and  
15 WSWAPREQ signals are transmitted from a host processor.

The filter 10 is expandable in that it can be  
cascaded to provide for more coefficients and hence,  
handle larger amounts of data. The cascade input  
 $Cin(n)$ , is input to the filter 10 via cascade input  
20 signal line 24. The cascade input signal line 24 is  
connected to a third switching circuit 34 which may  
provide the real data portion of a  $Cout(n)$  signal from a  
preceding filter stage to the real processing half 17 of  
the filter as well as provide the imaginary data portion  
25 of  $Cout(n)$  from a preceding filter stage to the  
imaginary processing half 16 of the filter. The cascade  
output signal  $Cout(n)$ , is the cascade output of filter  
10 and is output via signal line 25, which is connected  
to a fourth switching circuit 36. Switching circuit 36  
30 is a demultiplexer that switches between the real output  
data from the real processing circuit 17 and the  
imaginary output data from the imaginary processing

-14-

1 circuit 16 to produce the complex signal  $C_{out}(n)$  at  
output signal line 25.

The error signal  $E(n)$  is the error output  
which is essentially a summation of the desired input  
5 signal  $D_{in}(n)$ , 13, with both the real and imaginary  
output data available at  $C_{out}(n)$ . The error signal is  
output via output signal line 31, which is connected to  
a fifth switching circuit 35. The real output portion  
of error signal  $E(n)$  is a summation in the first summing  
10 circuit 20, wherein real data portion of  $D_{in}(n)$  is added  
to the real output data from the real processing circuit  
17. The imaginary output portion of error signal  $E(n)$   
is a summation in the second summing circuit 22, where  
the imaginary data portion of  $D_{in}(n)$  is added to the  
15 imaginary output data from the imaginary processing  
circuit 16. Due to the circuit which implements filter  
10, the real part of the first term of a complex  
correlation or convolution will be present at the output  
of the filter after  $2*N$  clock cycles, where 'N' is the  
20 number of weights the filter is implementing. Thus, as  
mentioned above, the  $D_{in}(n)$  signal is delayed by  
external control means at least  $2*N$  clock cycles to  
account for the latency of the filter.

Referring now to Fig. 2 there is shown a  
25 generalized block diagram of a 32 complex weight FIR  
filter 10 having a real processing circuit 17 and  
imaginary processing circuit 16. The switching circuit  
32 is shown in a first position where the real input  
data portion of the input signal (hereinafter  
30 "Re[Y<sub>in</sub>(n)]" is connected to the half of the real  
processing circuit 17 having the odd numbered "a"  
coefficients and simultaneously connected to the half of

-15-

1 the imaginary processing circuit 16 having the odd  
numbered "b" coefficients through a single clock cycle  
storage element 65. Hereinafter, the connection to the  
odd "a" and odd "b" coefficients will be referred to as  
5 a connection to the odd side coefficients. As can be  
seen, this is configured conceptually as a first  
parallel data path for processing the real data input of  
Yin(n). When the switching circuit 32 is in a second  
position, as dictated by control circuitry 30 to be  
10 explained below, the imaginary input data portion of the  
input signal hereinafter "Im[Yin(n)]" is connected to  
the half of the real processing circuit 17 having the  
even numbered "b" coefficients and simultaneously  
connected to the half of the imaginary processing  
15 circuit 16 having the even numbered "a" coefficients  
through a single clock cycle storage element 65.  
Hereinafter the connection to even "a" and even "b"  
coefficients will be referred to as a connection to the  
even side coefficients. This configuration illustrates  
20 conceptually a second parallel data path for processing  
the imaginary data input of Yin(n). The storage  
elements 65 in the paths of the imaginary processing  
circuit 16 provide the delay necessary to multiplex the  
real and imaginary output of the filter.

25 It should be readily understood to those  
skilled in the art that complex signal processing (e.g.  
convolution or correlation) involves multiplication of  
the input signal with a select set of real and imaginary  
multiplicands, i.e. coefficients, such that when the  
30 results are accumulated, a complex output signal is  
formed. Note that the geometry of the complex filter of  
the present invention is configured so that the top

-16-

1 portion 17 of Fig. 2 will have coefficients such that  
when they are multiplied with  $Y_{in}(n)$  and selectively  
accumulated, a real output data portion of the filter 10  
is obtained. The bottom portion 16 will have the  
5 coefficients that will produce the imaginary output data  
portion of the filter 10. All even and odd side  
coefficients are weights having values that are  
dependent upon the type of function the filter is to  
perform, e.g. complex correlation or convolution. The  
10 real and imaginary data portions of  $Y_{in}(n)$  are  
multiplied with the corresponding coefficients stored in  
the coefficient memory banks 24, 27 and each  
multiplication is accomplished by a multiplier circuit  
4. Delay elements, labeled 6 in Fig. 2, are used for  
15 illustrative purposes to signify that the inputs are to  
be stored in the accumulator for one clock cycle before  
they appear at the output. In the preferred embodiment,  
the delay elements 6 are not implemented as such, but  
rather the control circuit 30 and control signals 39  
20 (see Fig. 1) controlling the multiplier circuits 4 and  
accumulator circuits 5 properly account for the delay.  
Each accumulator circuit 5 is a summing circuit that  
adds two inputs to produce an output out of the right  
hand side of the accumulator. For instance, the  
25 accumulator 5 shown in Fig. 2 will add the  
multiplication result of  $a_{31} * \text{Re}[Y_{in}(n)]$  with the result  
of the multiplication of  $b_{30}$  with  $\text{Im}[Y_{in}(n)]$  one clock  
cycle after the 'b' side multiplication has taken place.  
With reference to the description of the timing diagram  
30 of Fig. 4 it will be appreciated that the use of  
parallel paths takes maximum advantage of the time

-17-

1 available to do the required multiplications and  
accumulations.

Fig. 4 illustrates the timing sequence that is  
implemented by control circuitry 30 and control signals  
5 39 to steer the real and imaginary inputs to the  
multipliers 4 and accumulators 5. The input  $Yin(n)$  is  
clocked in on a  $XMHz$  Clock 19. The data input to the  
real and imaginary processing circuitry is controlled by  
the signal QSWITCH which is signal 40. This signal is  
10 used to control switching circuit 32 to switch the input  
signal  $Yin(n)$ , signal 12, between the multipliers 4  
connected to the 'a' side coefficients and the 'b' side  
coefficients. The rising edge of QSWITCH signal 40 is  
used to clock the  $Re[Yin(n)]$  into the odd side while the  
15 falling edge of QSWITCH, via signal QSWITCH signal 42,  
clocks  $Im[Yin(n)]$  into the even side. [As can be seen  
from the Figure, QSWITCH and QSWITCH run at one-half the  
 $XMHz$  clock frequency.] Since the real and imaginary  
data processing proceed in parallel, the real data is  
20 multiplied by both the odd 'b' and odd 'a' coefficients  
and the imaginary data is multiplied by both the even  
'a' and even 'b' coefficients. The imaginary processing  
is delayed by one clock in order to provide the correct  
multiplexed output.

25 Once the real data is made available to the  
odd side multipliers, the multiplication of  
 $Re[Yin(n)] * 'a'_{odd}$  and  $Im[Yin(n-1)] * 'a'_{even}$  coefficients  
will begin on the rising edge of  $XMHz$  (signal 43)  
which is the inverted  $XMHz$  clock signal. This 'a' side  
30 multiplication will take place when QSWITCH 40 is high  
(i.e. Logic 1). When QSWITCH 40 is low, the rising edge  
of  $XMHz$  will start the 'b' side multiplication, i.e.,

35

-18-

1  $\text{Im}[\text{Yin}(n)] * 'b'$  even coefficient and  $\text{Re}[\text{Yin}(n-1)] * 'b'$  odd  
coefficients. The accumulation of results obtained from  
the previous multiplications involving 'b' side  
coefficients is performed at the time the multiplication  
5 of 'a' side coefficients take place. Similarly, the  
accumulation of results obtained from the previous  
multiplications involving 'a' side coefficient is  
performed at the time the multiplication of 'b' side  
coefficients take place. Once the accumulating has  
10 begun on the 'a' side, it is possible to switch to the  
alternate 'a' coefficient registers since the old  
coefficients are no longer needed and are not being  
utilized by the multipliers. Likewise, when the  
accumulating has begun on the 'b' side, it is possible  
15 to switch to the alternate 'b' coefficient registers.  
Changing access of data from one set of 'a' and 'b'  
coefficient registers from memory bank 24 (e.g., Bank 0)  
to the other set of 'a' and 'b' coefficient memory  
registers from memory bank 27 (e.g. Bank 1) will be  
20 referred to as "weight swapping" and this feature will  
be described in detail below.

As can be seen from the Fig. 4, QSWITCH 40 and  
QSWITCH 42 run at one-half the XMHz clock frequency.  
Thus, the internal calculations proceed at a rate of X/2  
25 MHz while the output of the filter is maintained at the  
XMHz data rate. In this particular embodiment, each  
multiply and accumulate cycle takes  $2/X \times 10^3$  ns to  
complete (using an XMHz clock). In conventional FIR  
filter designs, the multiply must take place in the  
30 shorter time span  $1/X \times 10^3$  ns (i.e., at the XMHz clock  
rate).

-19-

1           As mentioned above, the coefficient/-  
coefficient update memory bank 24 and the alternative  
coefficient/ coefficient update memory bank 27 supply  
all the 'a' side and 'b' side coefficient values to the  
5   respective multipliers 4 (Fig. 2). The arrangement  
shown in Fig. 1 allows for easy conceptualization of the  
process. As shown therein, both coefficient memory  
banks 24 and 27 will provide the even 'b' and odd 'a'  
coefficients to the real processing circuit 17 of the  
10 filter and the even 'a' and odd 'b' coefficients to the  
imaginary processing half 16 of the filter. The  
respective 'a' and 'b' coefficients are broadcast to the  
multipliers in the real or imaginary processing circuits  
17 or 16 via respective weight use signal lines 26 and  
15 29.

Different arrangements may be possible for  
swapping the coefficients in memory bank 24 or Bank  $\phi$   
and memory bank 27 or Bank 1. For example, a more  
efficient arrangement might require half as many  
20 coefficients and consequently half as many registers if  
the register values can be broadcast to two multipliers  
and the accumulators are given the capability of  
subtraction as well as addition. This is true in view  
of the fact that in complex correlation calculations the  
25 first imaginary coefficient e.g.,  $a_0$  is equal to the  
succeeding real coefficient  $a_1$  and the first real  
coefficient  $b_0$  is equal to the negative of the  
succeeding imaginary coefficient  $b_1$  as will be shown  
below.

30           The weight swapping feature of the present  
invention will now be described with reference to the  
timing diagrams of Figs. 4 and 7. In view of Fig. 4,

35

-20-

1 and as heretofore mentioned, the 'a' side and 'b' side  
coefficients are updated when each respective 'a' side  
and 'b' side accumulation is taking place i.e., when the  
coefficients are not being used by the multipliers 4.  
5 As can be seen, there is no time when both the 'a' side  
and the 'b' side coefficients are not being used by the  
multipliers, so each must be swapped separately.

The weight swap circuitry 38 (see Fig. 1)  
provides the timing and control signals necessary to  
10 accomplish the weight swapping function in the real and  
imaginary coefficient/coefficient update memory banks 24  
and 27. The control and timing signals used during the  
weight swapping process are shown in Fig. 7. The  
signals WSWAPA, 54, and WSWAPB, 58, that are generated  
15 by weight swapping circuitry 38, (Fig. 1) actually cause  
the corresponding 'a' side and 'b' side weight swapping  
to occur. Input to the weight swapping circuitry 38 are  
the XMHz clock, signal 19,  
WSWAPREQ $\bar{\sim}$ , signal 21, and MULTIPLYA and MULTIPLYB, which  
20 are signals 53 and 56 generated from control circuitry  
30.

With reference to Figs. 7a and 8, the swapping  
of the 'a' side coefficients between Bank  $\phi$  and Bank 1  
will now be explained. It should be understood that the  
25 same timing diagram shown in Fig. 7a applies to the  
swapping of 'b' side coefficients. The input WSWAPREQ $\bar{\sim}$   
(21) generated from the host indicates that the host has  
updated the coefficients in the register bank not being  
used by the filter. The weight swap request WSWAPREQ $\bar{\sim}$   
30 (21) is synchronized with the XMHz clock (19) to produce  
the weight swap request synch WSWAPREQSA $\bar{\sim}$  which is  
signal 48. When swapping 'a' side coefficients, this

-21-

1 signal is represented as WSWAPREQSA<sup>~</sup>. Fig. 7b sets  
forth the truth table 70 describing the logic governing  
the production of the following signals: WSWAPREQSA<sup>~</sup>  
(signal 48); weight swap request delay A, WSWAPREQDELA<sup>~</sup>  
5 (signal 52); weight swap request delay B, WSWAPREQDELB<sup>~</sup>  
(signal 55); weight swap acknowledge A, WSWAPAKA<sup>~</sup>  
(signal 57); weight swap acknowledge B, WSWAPAKB<sup>~</sup>  
(signal 59); weight swap acknowledge, WSWAPAK<sup>~</sup> (signal  
10 60). The legend shown in Fig. 7b provides the  
definitions of the symbols used in the truth table of  
Fig 7b. For instance, the WSWAPAK<sup>~</sup> signal 60 will go  
low (logic  $\emptyset$ ) on the next clock if signal 57, WSWAPAKA<sup>~</sup>,  
and signal 59, WSWAPAKB<sup>~</sup>, are both low. The WSWAPAK<sup>~</sup>  
signal 60 is thus generated and output from the filter  
15 when both 'a' side and 'b' side coefficients have been  
swapped. Note that the swapping logic circuitry 38  
implementing the logic set forth in the truth table 70  
of Fig. 7b may be implemented by programmable array  
logic, but preferably is implemented in the ASIC.

20 As previously mentioned, the WSWAPA signal 54  
causes the weight swapping on the 'a' side to occur.  
The signal WSWAPREQDELA<sup>~</sup> (52) waits for the MULTIPLYA,  
signal 53, before generating the WSWAPA signal 54. This  
condition indicates that on the next clock, the 'a' side  
25 multiply will be completed and the accumulate will  
begin. As can be seen in Figs. 7a and 7c, WSWAPA is  
generated as an output of the J-K flip-flop 61 at the  
falling edge 62 of WSWAPREQDELA<sup>~</sup> (signal 52). Note  
that after an 'a' side swap has occurred, the WSWAPAKA<sup>~</sup>,  
30 signal 57, is asserted according to the logical  
condition as set forth in truth table 70.

-22-

1           The WSWAPB signal 58, which causes the  
swapping of the 'b' side coefficients after the 'a' side  
swap has occurred, is generated as an output of a second  
J-K flip-flop (not shown) at the falling edge of  
5 WSWAPREQDELB<sup>~</sup> (signal 55). Before the 'b' side swap  
occurs, WSWAPREQDELB<sup>~</sup> waits for the combination of  
signals MULTIPLYB (signal 56) and WSWAPAKA<sup>~</sup> (signal 57)  
in accordance with the logic as dictated by the swapping  
logic truth table 70 of Fig. 7b. After a 'b' side swap  
10 has occurred, the WSWAPAKB<sup>~</sup> signal 59 is asserted in  
accordance with the logical condition as set forth in  
truth table 70 of Fig. 7b.

The advantages of the two memory bank  
architecture for each half of the filter are readily  
15 apparent. First of all, maximum filter processing time  
is made available because, for e.g., while the  
accumulation of the results obtained from the  
multiplications of the input data with 'a' side  
coefficients presently loaded into Bank 0 are taking  
20 place, a new set of updated 'a' side coefficients in  
Bank 1 memory can be made accessible to the multipliers.  
This ensures that valid 'a' side coefficients are always  
made available so there is no interruption of the  
filtering process. Likewise, for the 'b' side  
25 coefficients although there is a one clock cycle delay.

With conventional prior art filters, it takes  
more than one clock cycle to write data into a memory  
bank of registers after that bank has supplied the  
coefficients for multiplication with the input during a  
30 previous clock cycle. Thus, in a filter having one  
coefficient memory bank, the new set of coefficient data  
could not be loaded into that memory bank during the

-23-

1 accumulation cycle which nominally lasts for one clock  
cycle (see Fig. 4). Hence, after the accumulate cycle  
is complete, there would be a filter processing time  
delay existing for the amount of time it takes to finish  
5 loading new coefficient data into the memory bank.  
During this time delay, no multiplication could take  
place and filter performance is degraded.

With applicant's two Bank architecture, while  
the filter is operating with data present in the first  
10 coefficient memory bank, the alternative memory bank,  
i.e., the one not being used by the filter, is being  
updated with the new set of coefficients. This ensures  
that under the control of the weight swap circuitry,  
after the accumulation has taken place, multiplication  
15 with new coefficients loaded into the alternative bank  
can immediately take place with no time lapse. Hence,  
with no filter processing time delay, the complex data  
output rate of the filter is equal to the complex data  
input rate. As noted above, it should be appreciated  
20 that coefficient data will be initially loaded into the  
first coefficient memory bank (e.g. Bank  $\emptyset$ ). Subsequent  
updated coefficient data is successively loaded into the  
alternate memory bank, i.e., the one not being used by  
the multipliers, after each weight swap request under  
25 the automatic control of the weight swap circuitry.

The writing of coefficient data to the memory  
banks during the weight swapping process will now be  
explained with respect to Fig. 8. Fig. 8 shows how the  
'a' side coefficients are swapped between coefficient  
30 registers in Bank  $\emptyset$  and Bank 1. It is understood that  
when the coefficient registers containing 'a' side  
coefficients in Bank  $\emptyset$  are being accessed by the filter

-24-

1 10, the coefficient registers containing 'a' side  
coefficients in Bank 1 may be accessed by the host  
computer for writing updated 'a' coefficient data  
thereto and vice versa. 'b' side coefficients are  
5 swapped similarly, i.e., the 'b' side coefficients  
registers in Bank  $\emptyset$  are either being accessed by the  
filter 10 or host computer at any one time.

In operation, the individual 'a' side  
coefficient registers are accessed to receive data that  
10 is present on the DATAIN bus 23. As shown in Fig. 8,  
WSWAPA signal 54, together with  $Wr\sim$  signal 14, determine  
which coefficient/ coefficient update memory bank is  
being accessed by the filter 10, (e.g. Bank  $\emptyset$ ), and  
which memory bank is to receive the updated coefficients  
15 (e.g. Bank 1). The REGSELECT(5: $\emptyset$ ), signal 44, selects  
which individual coefficient register in the selected  
memory bank, i.e. registers  $C_4$  through  $C_5$ , is to receive  
the data present on DATAIN. As mentioned previously,  
DATAIN contains floating point coefficients, up to 32  
20 bits, for enhanced mathematical precision. An identical  
operation is performed when coefficients are being  
updated to the alternate coefficient/coefficient update  
memory Bank  $\emptyset$  while the coefficients present in Bank 1  
are being accessed by the filter 10. Again, WSWAPA,  
25 signal 54, together with  $Wr\sim$ , signal 14, determine that  
the coefficient registers in Bank  $\emptyset$  are to receive  
updated coefficient data. The REGSELECT(5: $\emptyset$ ), signal  
44, will then select which individual coefficient  
register in the Bank  $\emptyset$  is to receive the updated  
30 coefficient data present on DATAIN.

The same processes mentioned above apply to  
swapping the 'b' side coefficients as well. For the

-25-

1 case of updating 'b' side coefficients, the signal  
WSWAPB (58) together with  $W_r\sim$  (14) determine which  
weight/weight update memory bank is being accessed by  
the filter 10, (e.g. Bank  $\phi$ ), and which memory bank is  
5 to receive the updated coefficients (e.g. Bank 1,) and  
vice versa. It will be appreciated that the filter  
looks like a memory mapped peripheral to the host and  
that the coefficients are downloaded to sequential  
memory locations. Although it is not shown in Fig. 8,  
10 the  $C_e\sim$  signal 18 in Fig. 1 must be asserted while the  
 $W_r\sim$  (14) signal is activated as is standard practice  
with memory mapped peripherals.

Example of a 3 complex weight filter:

A complex correlation of two vectors will be  
15 described to illustrate how the filter of the present  
invention is utilized for digital signal processing  
applications. An N-weight complex correlation  
conceptually requires  $2 \times 2N$  coefficients. In this  
example, a 3-weight complex correlation is performed,  
20 and accordingly, 12 coefficients are required. The  
scope of the present invention should however not be  
limited to any particular number of coefficients except  
that these must be an even number per section for the  
cascading to work correctly. After the derivation of  
25 the coefficients for a 3-weight complex correlation, it  
will be shown that there are actually only six unique  
coefficients. In a preferred embodiment, a method for  
implementing the filter using only six coefficients will  
be shown.

30 A 3-weight filter is illustrated in Fig. 5.  
The implementation of the 3-weight complex filter is  
described to prove the design including the cascade in

-26-

1 from the higher order coefficients to the lower order  
coefficients and the summing of the filter output with  
the desired input to produce the error term.

A common digital signal processing function  
5 which is implemented as a FIR filter is a complex  
correlation. The complex correlation of two vectors is  
given by

$$R_{wy}(m) = \sum_{k=0}^{N-1} w^*(k)y(k+m) \quad (1)$$

10 where  $w^*(k)$  is the complex conjugate of the weight  
vector  $w(k)$ ,  $y(k+m)$  is a time shifted version of  
reference input signal  $y(m)$ , and  $N$  is the number of  
complex coefficients, which is equal to 3 in this  
15 example. Expanding the summation in equation (1) for  $m$   
equal to 0 results in

$$R_{wy}(0) = w^*(0)y(0) + w^*(1)y(1) + w^*(2)y(2) \quad (2)$$

The weight vector,  $w(k)$ , is given by

$$w(k) = rw(k) + j[iw(k)] \quad (3)$$

20 wherein  $rw(k)$  is the real part of the weight vector and  
[ $iw(k)$ ] is the imaginary part of the weight vector.

Expanding equation (3) for values of  $k$  ranging from  $\phi$  to  
 $N-1$  results in

$$\begin{aligned} w(0) &= rw(0) + j[iw(0)] \\ 25 \quad w(1) &= rw(1) + j[iw(1)] \\ w(2) &= rw(2) + j[iw(2)] \end{aligned} \quad (4)$$

The complex conjugate of the weight vectors of equation  
(4) are given by

$$\begin{aligned} w^*(0) &= rw(0) - j[iw(0)] \\ 30 \quad w^*(1) &= rw(1) - j[iw(1)] \\ w^*(2) &= rw(2) - j[iw(2)] \end{aligned} \quad (5)$$

The reference input vector,  $y(k+m)$ , is given by

35

-27-

$$\begin{aligned}
 1 \quad y(0) &= ry(0)+j[iy(0)] \\
 y(1) &= ry(1)+j[iy(1)] \\
 y(2) &= ry(2)+j[iy(2)]
 \end{aligned} \tag{6}$$

Substituting equations (5) and (6) into equation (2)

5 results in

$$\begin{aligned}
 Rwy(0) &= (rw(0)-j[iw(0)])(ry(0)+j[iy(0)]) \\
 &+ (rw(1)-j[iw(1)])(ry(1)+j[iy(1)]) \\
 &+ (rw(2)-j[iw(2)])(ry(2)+j[iy(2)])
 \end{aligned} \tag{7}$$

Expanding equation (7) results in

$$\begin{aligned}
 10 \quad Rwy(0) &= rw(0)ry(0)-j[iw(0)]ry(0)+iw(0)iy(0)+rw(0)j[iy(0)] \\
 &+ rw(1)ry(1)-j[iw(1)]ry(1)+iw(1)iy(1)+rw(1)j[iy(1)] \\
 &+ rw(2)ry(2)-j[iw(2)]ry(2)+iw(2)iy(2)+rw(2)j[iy(2)]
 \end{aligned} \tag{8}$$

Rearranging equation (8) into the real and imaginary parts results in

$$\begin{aligned}
 15 \quad Rwy(0) &= rw(0)ry(0)+iw(0)iy(0) \\
 &+ rw(1)ry(1)+iw(1)iy(1) \\
 &+ rw(2)ry(2)+iw(2)iy(2) \\
 &+ j[rw(0)iy(0)-ry(0)iw(0) \\
 &+ rw(1)iy(1)-ry(1)iw(1) \\
 20 \quad &+ rw(2)iy(2)-ry(2)iw(2)]
 \end{aligned} \tag{9}$$

The reference input,  $Y_{in}$ , enters the filter real part first, and accordingly, the values for  $Y_{in}$  are given by

$$\begin{aligned}
 25 \quad Y_{in}(0) &= ry(0) \\
 Y_{in}(1) &= iy(0) \\
 Y_{in}(2) &= ry(1) \\
 Y_{in}(3) &= iy(1) \\
 Y_{in}(4) &= ry(2) \\
 Y_{in}(5) &= iy(2)
 \end{aligned} \tag{10}$$

30

35

-28-

1 The coefficients are chosen based upon the specific  
 implementation of the filter shown in Fig. 5, and are  
 given by:  $a_0 = rw(2)$                        $b_0 = iw(2)$   
                   $a_1 = rw(2)$                        $b_1 = -iw(2)$   
 5                    $a_2 = rw(1)$                        $b_2 = iw(1)$                       (11)  
                   $a_3 = rw(1)$                        $b_3 = -iw(1)$   
                   $a_4 = rw(0)$                        $b_4 = iw(0)$   
                   $a_5 = rw(0)$                        $b_5 = -iw(0)$

It is clear that  $a_0 = a_1$ ,  $a_2 = a_3$ ,  $a_4 = a_5$ ,  $b_0 = -b_1$ ,  $b_2$   
 10  $= -b_3$ , and  $b_4 = -b_5$ . Thus, in a preferred embodiment,  
 only the coefficients  $a_0$ ,  $a_1$ ,  $a_2$ ,  $a_4$ ,  $b_0$ ,  $b_2$ , and  $b_4$   
 need be provided. Thus, with reference to Fig. 5,  $a_0$   
 would be broadcast to both multiplier circuits 82' and  
 80;  $a_2$  would be broadcast to multiplier circuits 78' and  
 15 76;  $a_4$  would go to multiplier circuits 74' and 72;  $b_0$   
 would be broadcast to multiplier circuits 82 and 80';  
 $b_2$  would be broadcast to 76' and 78; and  $b_4$  would be  
 broadcast to 72' and 74. Substituting the coefficients  
 of equation (11) and the values of the reference inputs  
 20 of equation (10) into equation (9) results in

$$\begin{aligned} Rwy(0) = & a_5 Y_{in}(0) + b_4 Y_{in}(1) + a_3 Y_{in}(2) + b_2 Y_{in}(3) \\ & + a_1 Y_{in}(4) + b_0 Y_{in}(5) + j[b_5 Y_{in}(0) + a_4 Y_{in}(1) \\ & + b_3 Y_{in}(2) + a_2 Y_{in}(3) + b_1 Y_{in}(4) + a_0 Y_{in}(5)] \end{aligned} \quad (12)$$

The complex correlation,  $Rwy(0)$ , in terms of filter  
 25 output is given by

$$Rwy(0) = Cout(6) + jCout(7) \quad (13)$$

wherein  $Cout(6)$  is the real part of  $Rwy(0)$  and  $Cout(7)$   
 is the imaginary part of  $Rwy(0)$ . Equation (13) is the  
 result of the mathematical calculation of the complex  
 30 correlation of two vectors.

Referring to Fig. 5, the block diagram  
 representation of the filter is utilized to demonstrate

1 that the operation of the filter corresponds' to the  
mathematical result achieved above. Each device in Fig.  
5 is identified by a name and a number corresponding to  
the column headings in Figs. 6a and 6b. The processing  
5 which occurs in the top half of the FIR filter of Fig. 5  
i.e., the real processing circuit 17, is tabulated in  
Fig. 6a and corresponds to the real output of the 3-  
weight complex filter. The bottom half of Fig. 5 i.e.,  
the imaginary processing circuit 16, is tabulated in  
10 Fig. 6b and corresponds to the imaginary output of the  
3-weight complex filter design. The grid of the  
processing table 6a, corresponding to the real  
processing circuit 17, makes use of equations 14 through  
19 below.

15 Referring to the real path of the filter, the  
output accum5 of the sixth summing circuit 71 is given  
by

$$\text{accum5} = \text{Cin} + \text{mult5}, \quad (14)$$

wherein Cin is the cascade input 19 and mult5 represents  
20 the output of the sixth multiplying circuit 72. The  
output accum4 of the fifth summing circuit 73 is given  
by

$$\text{accum4} = \text{accum5} + \text{mult4}, \quad (15)$$

where accum5 is given in equation (14) and mult4  
25 represents the output of the fifth multiplying circuit  
74. The output accum3 of the fourth summing circuit 75  
is given by

$$\text{accum3} = \text{accum4} + \text{mult3}, \quad (16)$$

where accum4 is given in equation (15) and mult3  
30 represents the output of the fourth multiplying circuit  
76. The output accum2 of the third summing circuit 77  
is given by

-30-

$$1 \quad \text{accum2} = \text{accum3} + \text{mult2}, \quad (17)$$

where accum3 is given in equation (16) and mult2 represents the output of the third multiplying circuit 78. The output accum1 of the second summing circuit 79 is given by

$$\text{accum1} = \text{accum2} + \text{mult1}, \quad (18)$$

where accum2 is given in equation (17) and mult1 represents the output of the second multiplying circuit 80. The output accum $\emptyset$  of the first summing circuit 81 is given by

$$\text{accum}\emptyset = \text{accum1} + \text{mult}\emptyset \quad (19)$$

where accum1 is given in equation (18) and mult $\emptyset$  represents the output of the first multiplying circuit 82.

15 Thus the results for the real output part of this 3-weight filter, Cout(n), is the output at time n, which in this case is equal to six. Thus, at the sixth clock cycle, the result Cout(6) corresponds to the result obtained mathematically in equation (12).

20 The processing table 6b corresponding to the imaginary processing circuit 16 of filter 10, makes use of equations 20 through 25 below. Referring to the imaginary path of the filter shown in Fig. 5, the output accum5 of the sixth summing circuit 71' is given by

$$25 \quad \text{accum5} = \text{Cin} + \text{mult5}, \quad (20)$$

wherein Cin is the cascade input 19 and mult5 represents the output of the sixth multiplying circuit 72'. The output accum4 of the fifth summing circuit 73' is given by

$$30 \quad \text{accum4} = \text{accum5} + \text{mult4}, \quad (21)$$

where accum5 is given in equation (20) and mult4 represents the output of the fifth multiplying circuit

-31-

1 74'. The output accum3 of the fourth summing circuit  
75' is given by

$$\text{accum3} = \text{accum4} + \text{mult3}, \quad (22)$$

where accum4 is given in equation (21) and mult3

5 represents the output of the fourth multiplying circuit  
76'. The output accum2 of the third summing circuit 77'  
is given by

$$\text{accum2} = \text{accum3} + \text{mult2}, \quad (23)$$

where accum3 is given in equation (22) and mult2

10 represents the output of the third multiplying circuit  
78'. The output accum1 of the second summing circuit  
79' is given by

$$\text{accum1} = \text{accum2} + \text{mult1}, \quad (24)$$

where accum2 is given in equation (23) and mult1

15 represents the output of the second multiplying circuit  
80'. The output accum $\emptyset$  of the first summing circuit 81'  
is given by

$$\text{accum}\emptyset = \text{accum1} + \text{mult}\emptyset \quad (25)$$

where accum1 is given in equation (24) and mult $\emptyset$

20 represents the output of the first multiplying circuit  
82'.

For the preferred embodiment of the 3-weight filter where only 2N or six coefficients are used, the filter would have to perform the subtraction wherein for  
25 e.g.:  $\text{accum1} = \text{accum2} - \text{mult1}$ ; and  $\text{accum3} = \text{accum4} - \text{mult3}$  to accommodate the (-) signs in the  $b_1$ ,  $b_3$ , and  $b_5$  coefficients.

Thus the results for the imaginary output part of this 3-weight filter,  $\text{Cout}(n)$ , is the output at time  
30  $n$ , which in this case is equal to seven. Note that at the seventh clock cycle, the result  $\text{Cout}(7)$  corresponds to the imaginary portion of the result obtained

-32-

1 mathematically in equation (12). The imaginary output  
Cout(7) is available at the seventh clock cycle due to  
the initial coefficient storage delay of one clock cycle  
implemented by the storage devices 65 in the imaginary  
5 processing path 16 of the filter. The delay of one  
clock cycle in this path is necessary to assure proper  
multiplexing of the Cout(n) signal 25, (filter output).

While the invention has been particularly  
shown and described with respect to the preferred  
10 embodiments thereof, it should be understood by those  
skilled in the art that the foregoing and other changes  
in form and detail may be made therein without departing  
from the spirit and scope of the invention which should  
be limited only by the scope of the appended claims.

15

20

25

30

35

-33-

1 WHAT IS CLAIMED IS

1. A finite impulse response filter with adaptive weights for processing a complex data signal having both real and imaginary input data, said finite  
5 impulse response filter comprising:

a) a first data path for calculating a real data output portion for said filter, said first path having a first plurality of adaptive weight means for multiplying said real input data by a predetermined  
10 coefficient set during a first time interval and successive first time intervals, and a second plurality of adaptive weight means for multiplying said imaginary input data by a predetermined coefficient set during a second time interval and successive second time  
15 intervals, each respective multiplication forming a plurality of multiplication results thereof;

b) a second data path for calculating an imaginary data output portion for said filter, said second path having a third plurality of adaptive weight  
20 means for multiplying said real input data by a predetermined coefficient set during said second time interval and successive second time intervals, and a fourth plurality of adaptive weight means for multiplying said imaginary input data by a predetermined  
25 coefficient set during said first time interval and successive first time intervals, each respective multiplication forming a plurality of multiplication results thereof;

c) control circuit means for providing said  
30 real input data portion to said first plurality and said third plurality of adaptive weight means during said first time interval and successive first time intervals

-34-

1 thereafter, and for providing said imaginary input data  
portion to said second plurality and said fourth  
plurality of adaptive weight means during said second  
time interval and successive second time intervals  
5 thereafter;

d) a first input storage means located in  
said second data path for delaying the multiplication of  
said real input data with said third plurality of  
adaptive weight means for one time interval and a second  
10 input storage means located in said second data path for  
delaying the multiplication of said imaginary input data  
with said fourth plurality of adaptive weight means for  
one time interval;

e) first plurality of accumulator circuit  
15 means interconnecting said first and second plurality of  
adaptive weight means for accumulating therefrom a  
selective set of said plurality of multiplication  
results to form said real data output portion, and a  
second plurality of accumulator circuit means  
20 interconnecting said third and fourth plurality of  
adaptive weight means for accumulating therefrom a  
selective set of said plurality of multiplication  
results to form said imaginary data output portion, said  
real data output portion and said imaginary data output  
25 portion together forming a complex output signal;

wherein said control circuit means directs  
said first plurality and second plurality of accumulator  
circuit means to accumulate multiplication results  
obtained respectively from said first and fourth  
30 plurality of adaptive weight means during said second  
time interval, and to accumulate multiplication results  
obtained respectively from said second and third

35

-35-

1 plurality of adaptive weight means during said first  
time interval; and

f) coefficient swapping circuit means for  
updating the values of said predetermined coefficient  
5 sets, wherein said means simultaneously updates the  
values of the coefficients for said first plurality and  
said fourth plurality of adaptive weights means during  
said second time interval, and wherein said means  
simultaneously updates the values of the coefficients  
10 for said second plurality and said third plurality of  
adaptive weight means during said first time interval.

2. The filter according to Claim 1 wherein  
during each of said second time intervals, said control  
circuit means simultaneously directs said first and said  
15 second accumulator circuit means to accumulate said  
multiplication results obtained respectively from said  
first plurality and said fourth plurality of adaptive  
weight means.

3. The filter according to Claim 1 wherein  
20 during each of said first time intervals, said control  
circuit means simultaneously directs said first and said  
second accumulator circuit means to accumulate said  
multiplication results obtained respectively from said  
second plurality and said third plurality of adaptive  
25 weight means.

4. The filter according to Claim 1 wherein  
said first data path has associated therewith a first  
memory storage means for storing said predetermined  
coefficients sets used for multiplication by said first  
30 plurality and second plurality of adaptive weight means.

5. The filter according to Claim 1 wherein  
said second data path has associated therewith a second

-36-

1 memory storage means for storing said predetermined  
coefficient sets used for multiplication by said third  
plurality and fourth plurality of adaptive weight means.

6. The filter according to Claims 4 or 5  
5 wherein when said filter is implemented in a system  
having a processor means, said coefficient swapping  
circuit means directs said predetermined coefficients to  
be loaded into either of said first or second memory  
storage means at the request of said processor means.

10 7. The filter according to Claim 6 wherein  
said first memory storage means has associated therewith  
a first alternative memory storage means and said second  
memory storage means has associated therewith a second  
alternate memory storage means, wherein said coefficient  
15 swapping circuit means directs said first alternate and  
said second alternate memory storage means to receive  
updated values of said predetermined coefficients from  
said processor means while said predetermined  
coefficients stored in said first and second memory  
20 storage means are being utilized by their respective  
adaptive weight means for multiplication thereof.

8. The filter according to Claim 7 wherein  
said coefficient swapping circuit means directs said  
updated predetermined coefficient data to be loaded into  
25 said first and second memory storage means while said  
predetermined coefficients stored in said first  
alternate and second alternate memory storage means are  
being utilized by their respective adaptive weight means  
for multiplication thereof.

30 9. The filter according to Claim 7 wherein  
said coefficient swapping circuit means directs said  
updated predetermined coefficient values to be loaded

-37-

1 into said first alternate and second alternate memory  
storage means at the request of said processor means.

10. The filter according to Claims 7 or 8  
wherein said coefficient swapping circuit means  
5 generates a weight swap acknowledge signal to inform  
said processor means that either said first memory and  
second memory storage means are available to receive  
updated coefficient values or said first alternate  
memory and second alternate memory storage means are  
10 available to receive updated coefficient values.

11. A finite impulse response filter with  
adaptive weights for processing complex data having real  
and imaginary input data portions, said finite impulse  
response filter comprising:

15 a) a first data path for processing a real  
input data portion of said complex data, said first path  
having a first plurality of adaptive weight means for  
multiplying said real input data by predetermined  
coefficients to generate a set of multiplication  
20 results;

b) a second data path for processing an  
imaginary input data portion of said complex data, said  
second path having a second plurality of adaptive weight  
means for multiplying said imaginary input data by  
25 predetermined coefficients to generate a set of  
multiplication results;

c) control circuit means for providing said  
real input data portion to said first plurality of  
adaptive weight means during a first time interval and  
30 successive first time intervals thereafter, and for  
providing said imaginary input data portion to a second  
plurality of adaptive weight means during a second time

-38-

- 1 interval and successive second time intervals  
thereafter;
- d) a first plurality of accumulator circuit  
means interconnecting a first subset of said first  
5 plurality of adaptive weight means and a first subset of  
said second plurality of adaptive weight means for  
accumulating a selective first set of multiplication  
results therefrom to form a real data output portion for  
said filter;
- 10 e) first input storage means located in said  
first data path for delaying the multiplication of said  
real input data with a second subset of said first  
plurality of adaptive weight means by one time interval  
and a second input storage means located in said second  
15 data path for delaying the multiplication of said  
imaginary input data with a second subset of said second  
plurality of adaptive weight means by one time interval;
- f) a second plurality of accumulator circuit  
means interconnecting said second subset of said first  
20 plurality of adaptive weight means and said second  
subset of said second plurality of adaptive weight means  
for accumulating a selective second set of  
multiplication results therefrom to form an imaginary  
data output portion for said filter; and
- 25 g) coefficient swapping circuit means for  
updating the predetermined coefficient values of said  
first subset of said first plurality of adaptive weight  
means and said second subset of said second plurality of  
adaptive weight means during said second time interval,  
30 and for updating the predetermined coefficient values of  
said first subset of said second plurality of adaptive  
weight means and said second subset of said first

35

-39-

1 plurality of adaptive weight means during said first  
time interval.

12. The filter according to Claim 11, wherein  
said control circuit means simultaneously directs said  
5 first plurality and said second plurality of accumulator  
circuit means to accumulate respectively said  
multiplication results of said first subset of said  
first plurality of adaptive weight means and said second  
subset of said second plurality of adaptive weight means  
10 during said second time interval.

13. The filter according to Claim 11, wherein  
said control circuit means simultaneously directs said  
first plurality and said second plurality of accumulator  
circuit means to accumulate respectively said  
15 multiplication results of said first subset of said  
second plurality of adaptive weight means and said  
second subset of said first plurality of adaptive weight  
means during said first time interval.

14. The filter according to Claim 11, wherein  
20 said first plurality of adaptive weight means and said  
second plurality of adaptive weight means further  
includes respectively, a first memory storage means and  
second memory storage means for storing said  
predetermined coefficients.

25 15. The filter according to Claim 14, wherein  
when said filter is implemented in a system having a  
processor means, said coefficient swapping circuit means  
directs said predetermined coefficients to be loaded  
into either said first or second memory storage means at  
30 the request of said processor means.

16. The filter according to Claim 15, wherein  
said first plurality of adaptive weight means and said

-40-

1 second plurality of adaptive weight means further  
includes respectively, a first alternate memory storage  
means and a second alternate memory storage means for  
receiving updated values of said predetermined  
5 coefficients from said processor means while said  
predetermined coefficients stored in said first and  
second memory storage means are being utilized by each  
respective adaptive weight means for multiplication  
thereof.

10           17. The filter according to Claim 16, wherein  
said coefficient swapping circuit means directs said  
updated predetermined coefficient values to be loaded  
into either of said first alternate or second alternate  
memory storage means at the request of said processor  
15 means.

          18. The filter according to Claim 17 wherein  
said coefficient swapping circuit means directs said  
updated predetermined coefficient data to be loaded into  
said first alternate or second alternate memory storage  
20 means while said predetermined coefficients stored in  
said first and second memory storage means are being  
utilized by their respective adaptive weight means for  
multiplication thereof.

          19. The filter according to Claim 17 wherein  
25 said coefficient swapping circuit means directs said  
updated predetermined coefficient data to be loaded into  
said first and second memory storage means while said  
predetermined coefficients stored in said first  
alternate and second alternate memory storage means are  
30 being utilized by their respective adaptive weight means  
for multiplication thereof.

1           20. The filter according to Claim 1 wherein  
said first data path is a chain connecting each of said  
first plurality of accumulator circuit means, each  
alternately connected accumulator circuit means of said  
5 chain having a one of said first plurality of adaptive  
weight means connected thereto and wherein each of said  
alternately connected accumulator circuit means thereof  
is connected adjacent to an accumulator circuit means  
having a one of said second plurality of adaptive weight  
10 means connected thereto.

          21. The filter according to Claim 1 wherein  
said second data path is a chain connecting each of said  
second plurality of accumulator circuit means, each  
alternately connected accumulator circuit means of said  
15 chain having a one of said third plurality of adaptive  
weight means connected thereto and wherein each of said  
alternately connected accumulator circuit means thereof  
is connected adjacent to an accumulator circuit means  
having a one of said fourth plurality of adaptive weight  
20 means connected thereto.

          22. The filter according to Claim 1 wherein  
said control circuit means includes a delay means for  
enabling each one of said first and second plurality of  
accumulator circuit means to store said multiplication  
25 results obtained from a respective adaptive weight means  
for a period of one time interval.

          23. The filter according to Claim 11 wherein  
each one of said first plurality of accumulator circuit  
means is connected to a corresponding one of said first  
30 subset of said first plurality of adaptive weight means  
or a corresponding one of said first subset of said  
second plurality of adaptive weight means wherein each

-42-

1 of said first plurality of accumulator circuit means  
having a one of said first subset of said first  
plurality of adaptive weight means connected thereto is  
connected adjacent to a one of said first plurality of  
5 accumulator circuit means having a one of said first  
subset of said second plurality of adaptive weight means  
connected thereto.

24. The filter according to Claim 11 wherein  
each one of said second plurality of accumulator circuit  
10 means is connected to a corresponding one of said second  
subset of said first plurality of adaptive weight means  
or a corresponding one of said second subset of said  
second plurality of adaptive weight means wherein each  
of said second plurality of accumulator circuit means  
15 having a one of said second subset of said first  
plurality of adaptive weight means connected thereto is  
connected adjacent to a one of said second plurality of  
accumulator circuit means having a one of said second  
subset of said second plurality of adaptive weight means  
20 connected thereto.

25. The filter according to Claim 11 wherein  
said control circuit means includes a delay means for  
enabling each one of said first and second plurality of  
accumulator circuit means to store said multiplication  
25 results obtained from a respective adaptive weight means  
for a period of one time interval.

26. A method for processing complex digital  
data having both real and imaginary input data said  
method comprising the steps of:

30 a) alternately loading real input data into a  
first data path and imaginary input data into a second  
data path during successive time intervals, said first

35

-43-

1 data path having a first plurality and second plurality  
of adaptive weight means having coefficient data with  
predetermined values and said second data path having a  
third and fourth plurality of adaptive weight means  
5 having coefficient data with predetermined values;

b) storing said real input data in said first  
data path having said second plurality of adaptive  
weight means for one time interval after loading said  
real input data into said first data path, and storing  
10 said imaginary input data in said second data path  
having said fourth plurality of adaptive weight means  
for one time interval after loading said imaginary input  
data into said second data path;

c) simultaneously multiplying during a first  
15 and successive first time intervals said real input data  
and imaginary input data respectively by said  
coefficient data values of said first plurality of  
adaptive weight means and fourth plurality of adaptive  
weight means to obtain respectively therefrom a first  
20 plurality and fourth plurality of multiplication  
results;

d) simultaneously multiplying during a second  
and successive second time intervals said imaginary  
input data and real input data respectively by said  
25 coefficient data values of said second plurality of  
adaptive weight means and third plurality of adaptive  
weights means to obtain respectively therefrom a second  
plurality and third plurality of multiplication results;

30 e) alternately accumulating the results of  
each of said first plurality and third plurality of

-44-

1 multiplication results to form therefrom a real data  
output portion for said filter; and

f) alternately accumulating the results of  
said second plurality and said fourth plurality of  
5 multiplication results to form therefrom an imaginary  
data output portion for said filter,

wherein said first plurality and fourth  
plurality of multiplication results are accumulated  
simultaneously and said second plurality and said third  
10 plurality of multiplication results are accumulated  
simultaneously.

27. The method according to Claim 26 further  
including the steps of updating the coefficient data  
values of said first plurality and fourth plurality of  
15 adaptive weight means during the accumulation of said  
first plurality and fourth plurality of multiplication  
results, and updating the coefficient data values of  
said second plurality and third plurality of adaptive  
weight means during the accumulation of said second  
20 plurality and third plurality of multiplication results.

28. The method according to Claim 27 wherein  
said first and fourth plurality of adaptive weight means  
has associated therewith a first and first alternate  
memory storage means for storing said coefficient data,  
25 said updating step further including the step of  
alternately accessing said coefficient data values from  
either said first or alternate first memory storage  
means during an accumulation of said first plurality and  
fourth plurality of multiplication results.

29. The method according to Claim 27 wherein  
30 said second and third plurality of adaptive weight means  
has associated therewith a second and second alternate

1 memory storage means for storing said coefficient data,  
said updating step further including the step of  
alternately accessing said coefficient data values from  
either said second or alternate second memory storage  
5 means during an accumulation of said second plurality  
and third plurality of multiplication results.

30. The filter according to Claims 1 or 11  
further including means for generating an error signal  
for use by said processor means.

10 31. The filter according to Claim 30 wherein  
said means for generating an error signal includes a  
first summing circuit means for adding said real output  
portion for said filter with a desired real output  
signal and a second summing circuit means for adding  
15 said imaginary output portion for said filter with a  
desired imaginary output signal.

32. The filter according to Claim 21 further  
including a cascade input circuit means for extending  
the length of said first and second data paths, said  
20 cascade circuit means connected to an input end of said  
first and second data paths.

33. The filter according to Claim 1 wherein  
said control circuit means further includes means for  
multiplexing said real output data portion from said  
25 first data path and said imaginary output data portion  
from said second data path to form said complex output  
signal at an output end of said filter.

34. The filter according to Claim 33 further  
including cascade output means for connecting said  
30 multiplexing means at said output end of said filter to  
a cascade input circuit means of a succeeding finite

1 impulse response filter stage to thereby cascade said  
finite impulse response filters.

35. The method according to Claims 28 or 29  
wherein said updating step further includes the step of  
5 generating an acknowledge signal to inform a host  
processor that one of said first, alternate first,  
second and alternate second memory storage means is  
available for receiving coefficient data.

10

15

20

25

30

35

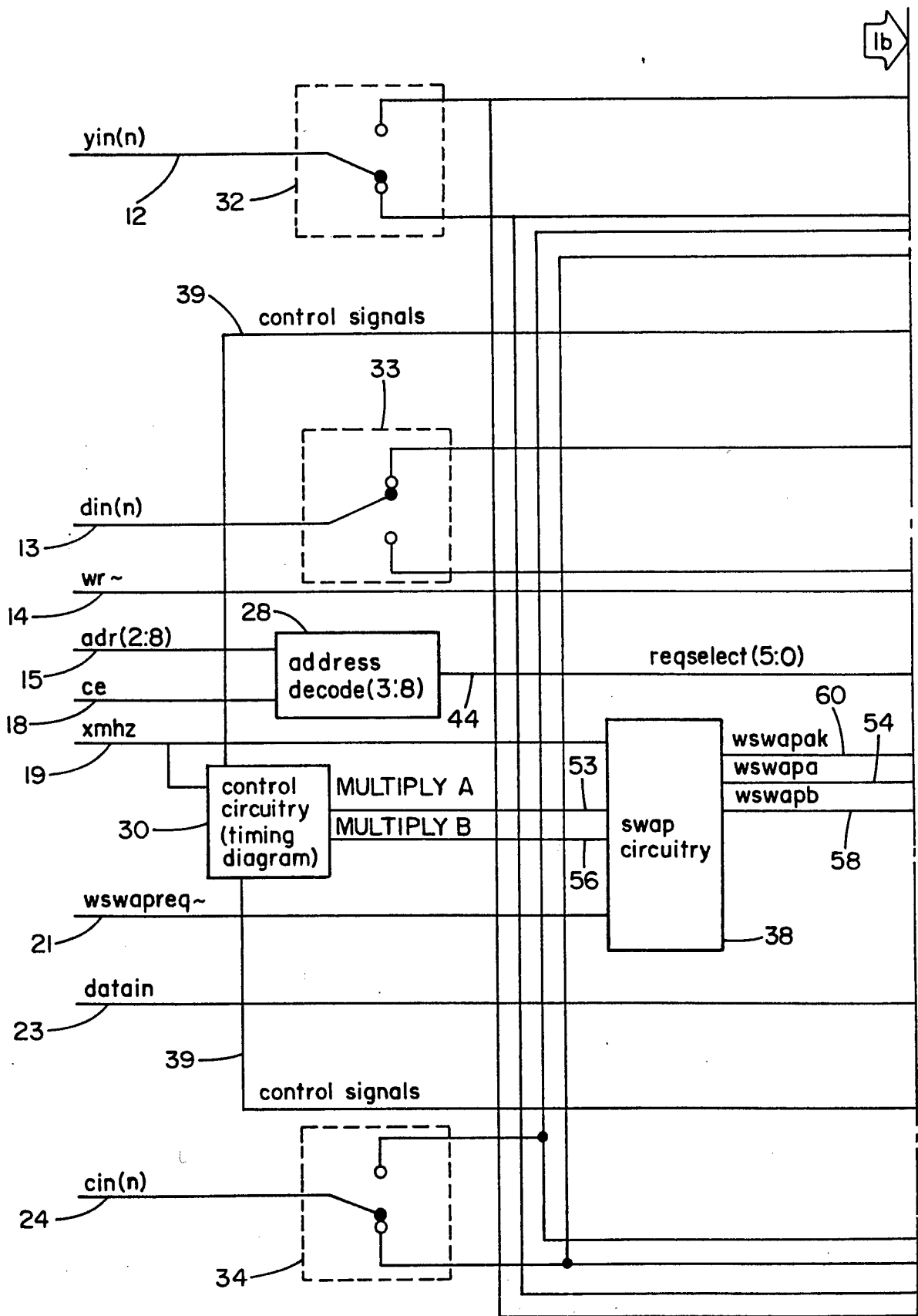


FIG. 1a

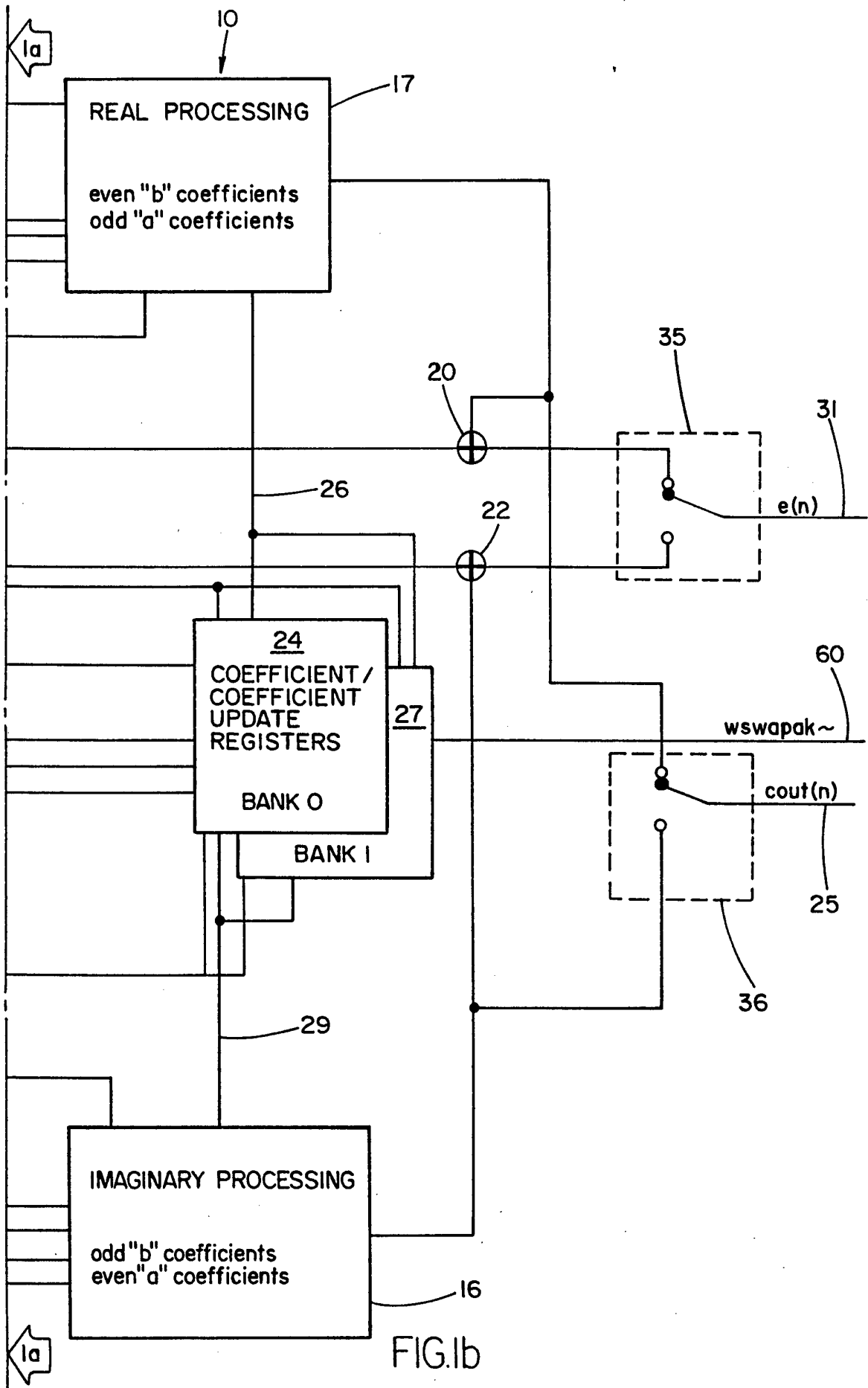


FIG. 1b

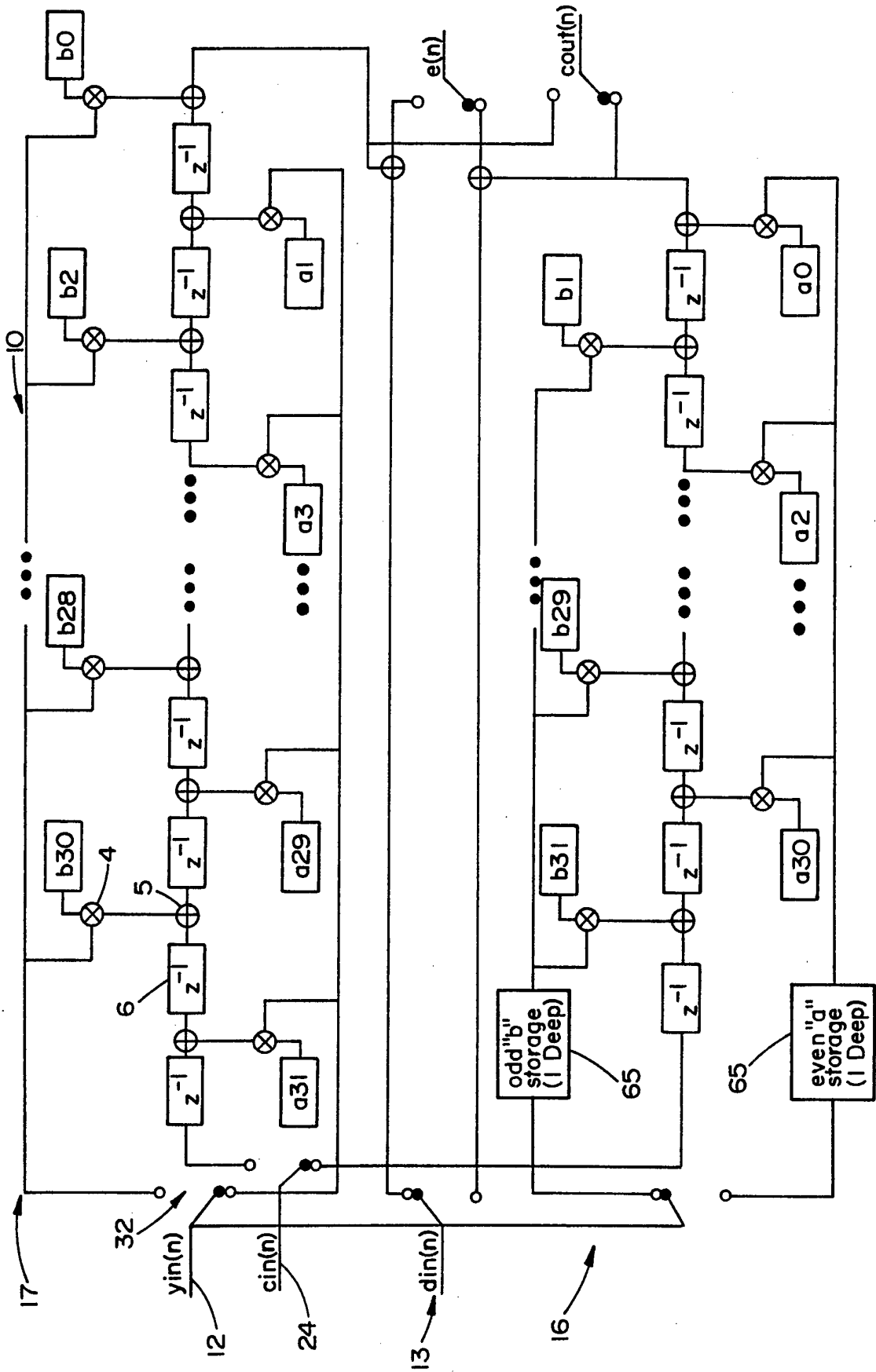


FIG. 2

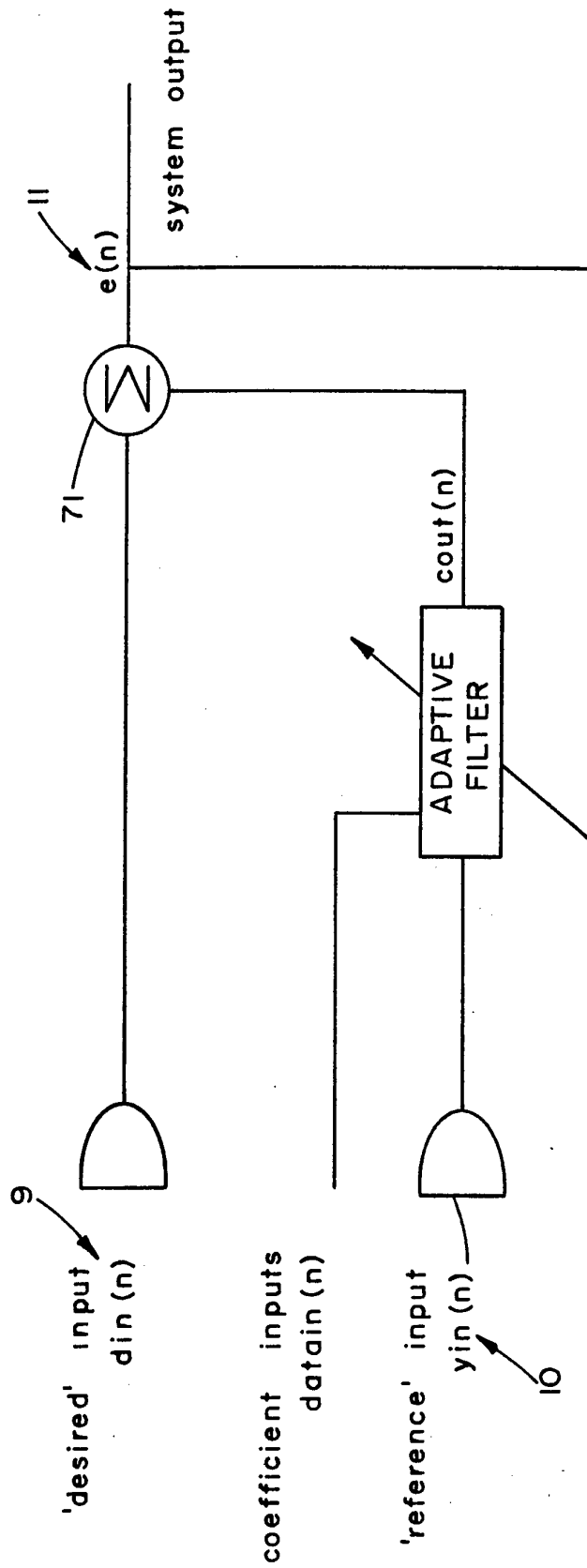
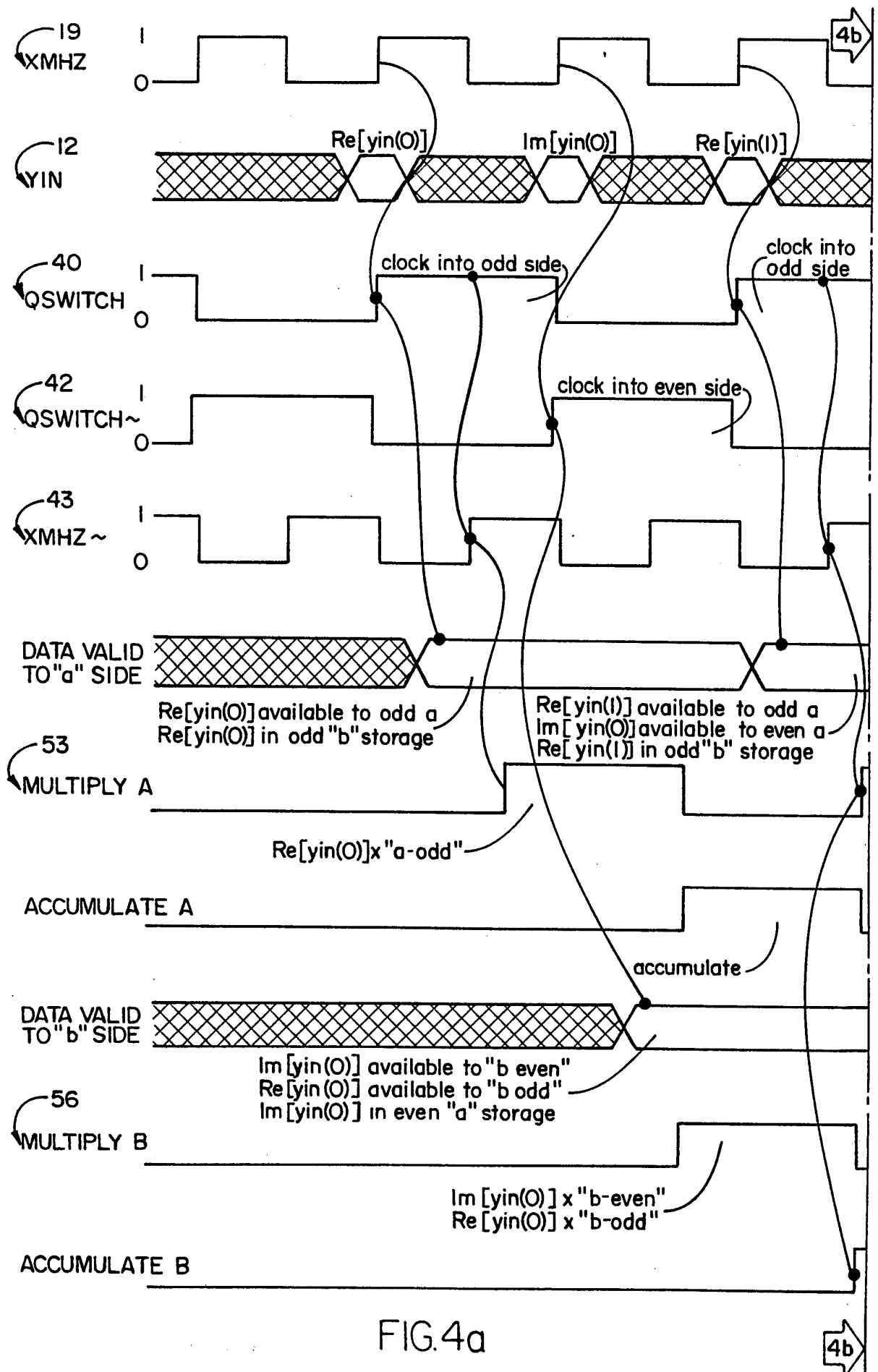


FIG.3



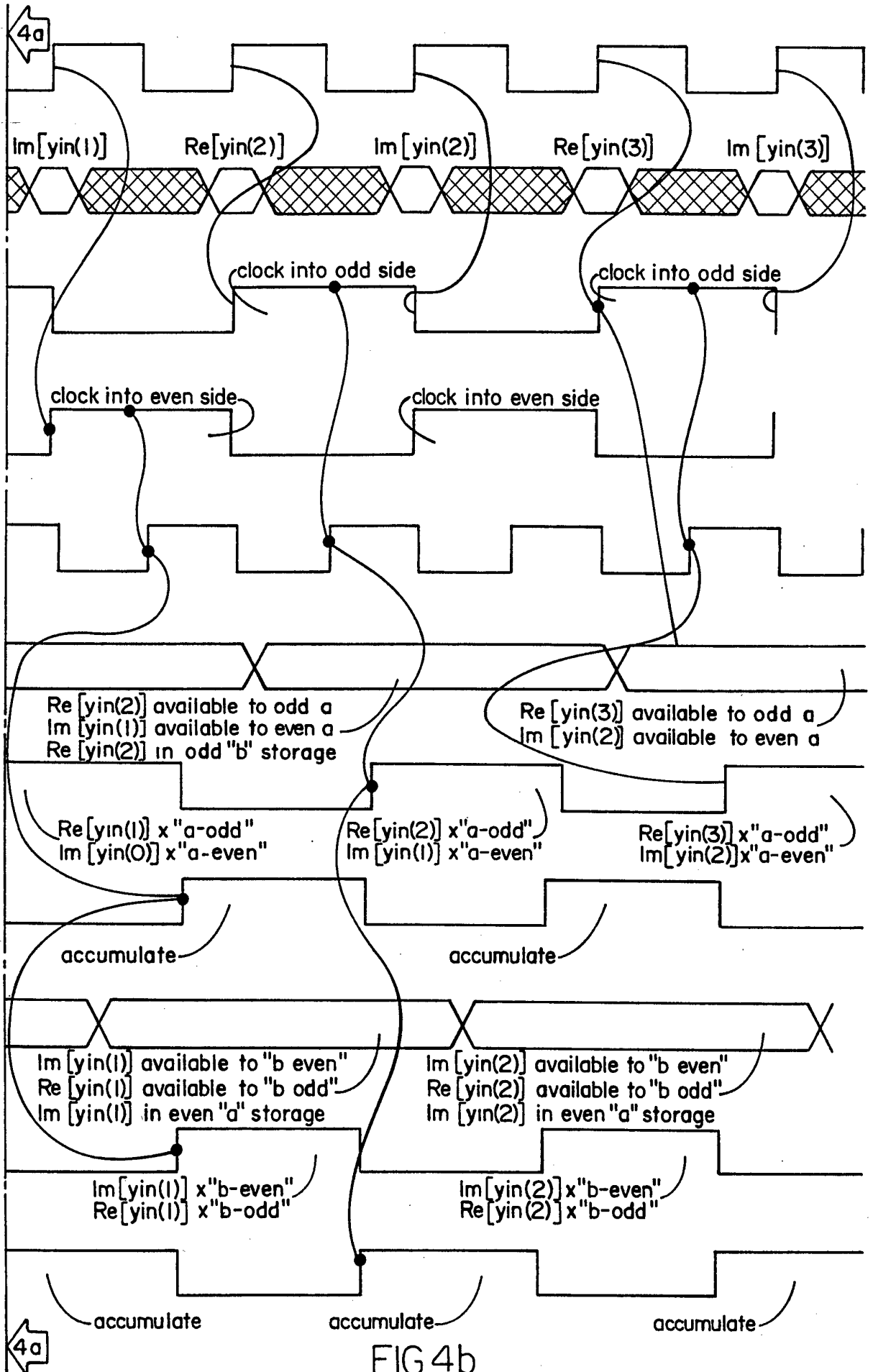


FIG 4b

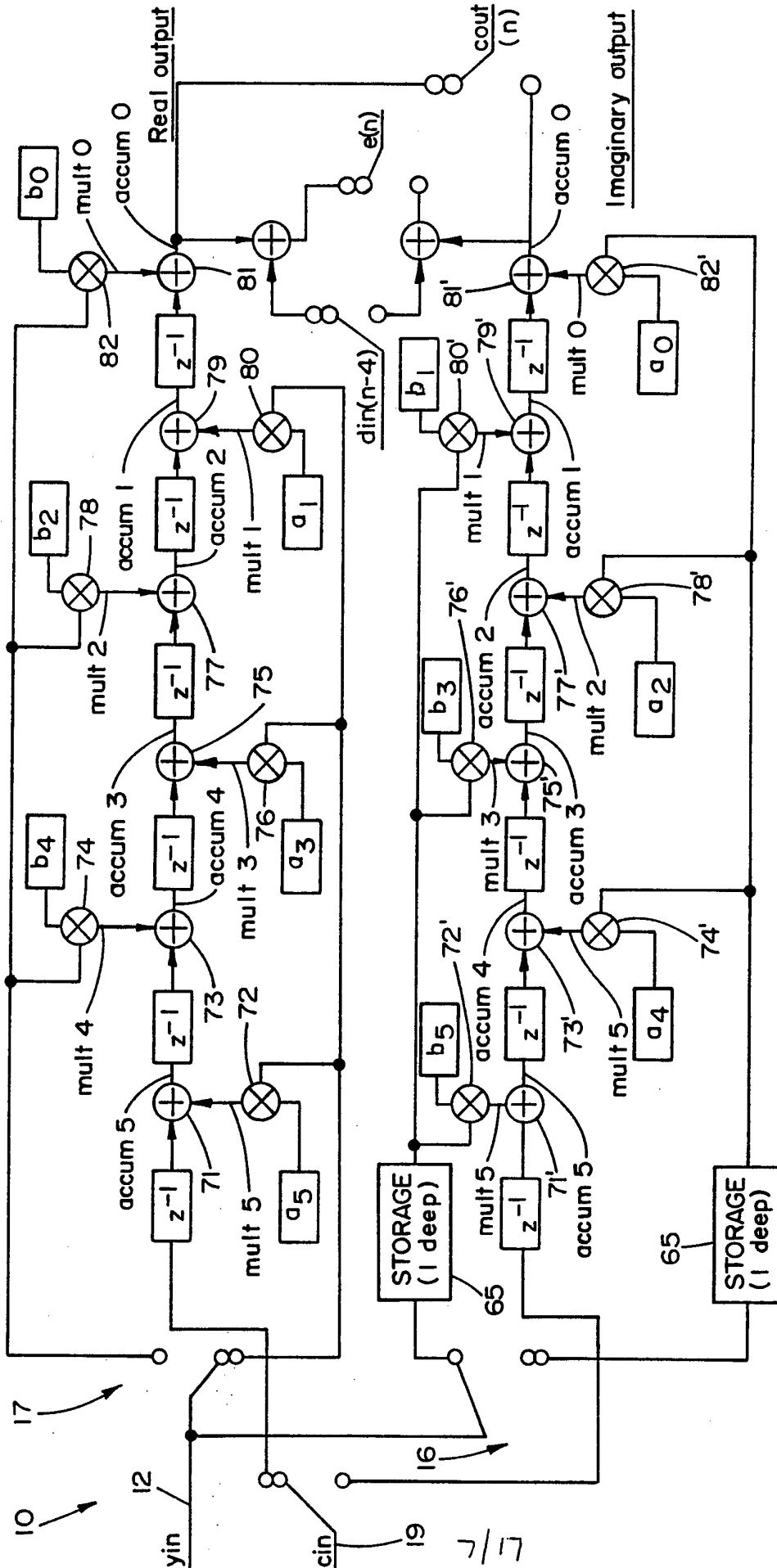


FIG.5

CLOCK	INPUT	mult	accum	mult	accum	mult
		5	5	4	4	3
0	yin(0)	$yin(0) \cdot a_5$	0	0	0	$yin(0) \cdot a_3$
1	yin(1)	0	$yin(0) \cdot a_5$	$yin(1) \cdot b_4$	0	0
2	yin(2)	$yin(2) \cdot a_5$	0	0	$yin(0) \cdot a_5 + yin(1) \cdot b_4$	$yin(2) \cdot a_3$
3	yin(3)	0	$yin(2) \cdot a_5$	$yin(3) \cdot b_4$	0	0
4	yin(4)	$yin(4) \cdot a_5$	0	0	$yin(2) \cdot a_5 + yin(3) \cdot b_4$	$yin(4) \cdot a_3$
5	yin(5)	0	$yin(4) \cdot a_5$	$yin(5) \cdot b_4$	0	0
6	yin(6)	$yin(6) \cdot a_5$	0	0	$yin(4) \cdot a_5 + yin(5) \cdot b_4$	$yin(6) \cdot a_3$
7	yin(7)	0	$yin(6) \cdot a_5$	$yin(7) \cdot b_4$	0	0
8	yin(8)	$yin(8) \cdot a_5$	0	0	$yin(6) \cdot a_5 + yin(7) \cdot b_4$	$yin(8) \cdot a_3$
9	yin(9)	0	$yin(8) \cdot a_5$	$yin(9) \cdot b_4$	0	0

6A2

6A2

FIG.6A1

← 6A1	accum	mult	accum	mult	accum	mult → 6A3
	3	2	2	1	1	0
	0	0	0	$y_{in}(0) \cdot a_1$	0	0
	$y_{in}(0) \cdot a_3$	$y_{in}(1) \cdot b_2$	0	0	$y_{in}(0) \cdot a_1$	$y_{in}(1) \cdot b_0$
	0	0	$y_{in}(0) \cdot a_3 +$ $y_{in}(1) \cdot b_2$	$y_{in}(2) \cdot a_1$	0	0
	$y_{in}(0) \cdot a_5 +$ $y_{in}(1) \cdot b_4 +$ $y_{in}(2) \cdot a_3$	$y_{in}(3) \cdot b_2$	0	0	$y_{in}(0) \cdot a_3 +$ $y_{in}(1) \cdot b_2 +$ $y_{in}(2) \cdot a_1$	$y_{in}(3) \cdot b_0$
	0	0	$y_{in}(0) \cdot a_5 +$ $y_{in}(1) \cdot b_4 +$ $y_{in}(2) \cdot a_3 +$ $y_{in}(3) \cdot b_2$	$y_{in}(4) \cdot a_1$	0	0
	$y_{in}(2) \cdot a_5 +$ $y_{in}(3) \cdot b_4 +$ $y_{in}(4) \cdot a_3$	$y_{in}(5) \cdot b_2$	0	0	$y_{in}(0) \cdot a_5 +$ $y_{in}(1) \cdot b_4 +$ $y_{in}(2) \cdot a_3 +$ $y_{in}(3) \cdot b_2 +$ $y_{in}(4) \cdot a_1$	$y_{in}(5) \cdot b_0$
	0	0	$y_{in}(2) \cdot a_5 +$ $y_{in}(3) \cdot b_4 +$ $y_{in}(4) \cdot a_3 +$ $y_{in}(5) \cdot b_2$	$y_{in}(6) \cdot a_1$	0	0
	$y_{in}(4) \cdot a_5 +$ $y_{in}(5) \cdot b_4 +$ $y_{in}(6) \cdot a_3$	$y_{in}(7) \cdot b_2$	0	0	$y_{in}(2) \cdot a_5 +$ $y_{in}(3) \cdot b_4 +$ $y_{in}(4) \cdot a_3 +$ $y_{in}(5) \cdot b_2 +$ $y_{in}(6) \cdot a_1$	$y_{in}(7) \cdot b_0$
	0	0	$y_{in}(4) \cdot a_5 +$ $y_{in}(5) \cdot b_4 +$ $y_{in}(6) \cdot a_3 +$ $y_{in}(7) \cdot b_2$	$y_{in}(8) \cdot a_1$	0	0
	$y_{in}(6) \cdot a_5 +$ $y_{in}(7) \cdot b_4 +$ $y_{in}(8) \cdot a_3$	$y_{in}(9) \cdot b_2$	0	0	$y_{in}(4) \cdot a_5 +$ $y_{in}(5) \cdot b_4 +$ $y_{in}(6) \cdot a_3 +$ $y_{in}(7) \cdot b_2 +$ $y_{in}(8) \cdot a_1$	$y_{in}(9) \cdot b_0$

FIG 6A2

6A2 accum 0= YHAT	OUTPUT error = accum 0+ din (n-6)	din							comments
		n	n-1	n-2	n-3	n-4	n-5	n-6	
0	0	din							
0	0	din	din						
yin(0) · a1+ yin(1) · b0	0	din	din	din					
0	yin(0) · a1+ yin(1) · b0+ junk	din	din	din	din				
yin(0) · a3+ yin(1) · b2+ yin(2) · a1+ yin(3) · b0	0	din	din	din	din	din			
0	yin(0) · a3+ yin(1) · b2+ yin(2) · a1+ yin(3) · b0+ junk	din	din	din	din	din	din		
yin(0) · a5+ yin(1) · b4+ yin(2) · a3+ yin(3) · b2+ yin(4) · a1+ yin(5) · b0	0	din	din	din	din	din	din	din	YHAT = Re [Ryw(0)] = cout(6)
0	yin(0) · a5+ yin(1) · b4+ yin(2) · a3+ yin(3) · b2+ yin(4) · a1+ yin(5) · b0 + bin(0)	din	din	din	din	din	din	din	OUTPUT = Re [Ryw(0)] Re [d(0)]
yin(2) · a5+ yin(3) · b4+ yin(4) · a3+ yin(5) · b2+ yin(6) · a1+ yin(7) · b0	0	din	din	din	din	din	din	din	YHAT = Re [Ryw(1)] = cout(8)
0	yin(2) · a5+ yin(3) · b4+ yin(4) · a3+ yin(5) · b2+ yin(6) · a1+ yin(7) · b0+ din(2)								OUTPUT = Re [Ryw(1)] + Re [d(1)]

6A2

FIG.6A3

6B2

CLOCK	INPUT (from storage)	mult 5	accum 5	mult 4	accum 4	mult 3
0	Store yin(0)	0	0	0	0	0
1	yin(0)	$yin(0) \cdot b_5$	0	0	0	$yin(0) \cdot b_3$
2	yin(1)	0	$yin(0) \cdot b_3$	$yin(1) \cdot a_4$	0	0
3	yin(2)	$yin(2) \cdot b_5$	0	0	$yin(2) \cdot a_5$ $yin(1) \cdot a_4$	$yin(2) \cdot b_3$
4	yin(3)	0	$yin(2) \cdot b_5$	$yin(3) \cdot a_4$	0	0
5	yin(4)	$yin(4) \cdot b_5$	0	0	$yin(2) \cdot b_5 +$ $yin(3) \cdot a_4$	$yin(4) \cdot b_3$
6	yin(5)	0	$yin(4) \cdot b_5$	$yin(5) \cdot a_4$	0	0
7		$yin(6) \cdot b_5$	0	0	0	$yin(6) \cdot b_3$
8		0	$yin(6) \cdot b_5$	$yin(7) \cdot a_4$	0	0
9		0	0	0	0	0
10						

FIG.6B1

6B2

6B1						6B3	
accum	mult	accum	mult	accum	mult		
3	2	2	1	1	0		
0	0	0	0	0	0		
0	0	0	$yin(0) \cdot b_1$	0	0		
$yin(0) \cdot b_3$	$yin(1) \cdot a_2$	0	0	$yin(0) \cdot b_1$	$yin(1) \cdot a_0$		
0	0	$yin(0) \cdot b_3 + yin(1) \cdot a_2$	$yin(2) \cdot b_1$	0	0		
$yin(0) \cdot b_5 + yin(1) \cdot a_4 + yin(2) \cdot b_3$	$yin(3) \cdot a_2$	0	0	$yin(0) \cdot b_3 + yin(1) \cdot a_2 + yin(2) \cdot b_1$	$yin(3) \cdot a_0$		
0	0	$yin(0) \cdot b_5 + yin(1) \cdot a_4 + yin(2) \cdot b_3 + yin(3) \cdot a_2$	$yin(4) \cdot b_1$	0	0		
$yin(2) \cdot b_5 + yin(3) \cdot a_4 + yin(4) \cdot b_3$	$yin(3) \cdot a_2$	0	0	$yin(0) \cdot b_5 + yin(1) \cdot a_4 + yin(2) \cdot b_3 + yin(3) \cdot a_2 + yin(4) \cdot b_1$	$yin(5) \cdot a_0$		
0	0	$yin(2) \cdot b_5 + yin(3) \cdot a_4 + yin(4) \cdot b_3 + yin(5) \cdot a_2$	$yin(6) \cdot b_1$	0	0		
0	$yin(7) \cdot a_2$	0	0	$yin(2) \cdot b_5 + yin(3) \cdot a_4 + yin(4) \cdot b_3 + yin(5) \cdot a_2 + yin(6) \cdot b_1$	$yin(7) \cdot a_0$		
0	0	0	0	0	0		

FIG.6B2

← 6B2	accum	OUTPUT	din							comments	
	O = YHAT	error = accum O + din(n-6)	n	n-1	n-2	n-3	n-4	n-5	n-6		
	0	0	din								
	0	0	din	din							
	0	0	din	din	din						
	yin(0) · b <sub>1</sub> + yin(1) · a <sub>0</sub>	0	din	din	din	din					
	0	yin(0) · b <sub>1</sub> + yin(1) · a <sub>0</sub>	din	din	din	din	din				
	yin(0) · b <sub>3</sub> + yin(1) · a <sub>2</sub> + yin(2) · b <sub>1</sub> + yin(3) · a <sub>0</sub>	0	din	din	din	din	din	din			
	0	yin(0) · b <sub>3</sub> + yin(1) · a <sub>2</sub> + yin(2) · d <sub>1</sub> + yin(3) · a <sub>0</sub>	din	din	din	din	din	din	din		
	yin(0) · b <sub>5</sub> + yin(1) · a <sub>4</sub> + yin(2) · b <sub>3</sub> + yin(3) · a <sub>2</sub> + yin(4) · b <sub>1</sub> + yin(5) · a <sub>0</sub>	0	din	din	din	din	din	din	din		YHAT = Im [Ryw(0)] = cout(7)
	0	0	din	din	din	din	din	din	din		OUTPUT = Im [Ryw(0)] Im [d(0)]
	yin(2) · b <sub>5</sub> + yin(3) · a <sub>4</sub> + yin(4) · b <sub>3</sub> + yin(5) · a <sub>2</sub> + yin(6) · b <sub>1</sub> + yin(7) · a <sub>0</sub>	0	din	din	din	din	din	din	din		YHAT = Im [Ryw(1)] = cout(9)
											OUTPUT = Im [Ryw(1)] + Im [d(1)]

← 6B2

FIG.6B3

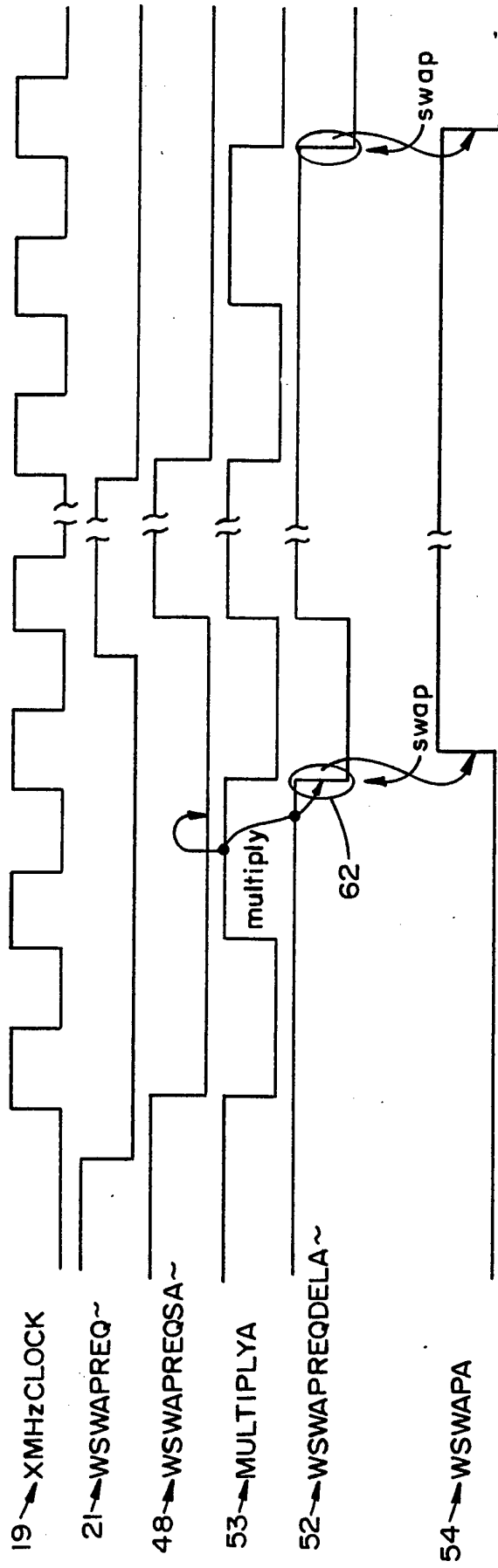


FIG. 7a

SWAPPING LOGIC

- 48 → WSWAPREQSA~!(WSWAPREQ~)
- 52 → WSWAPREQSDELA~!(WSWAPREQSA~ & WSWAPREQDELA~ & MULTIPLYA)
- 55 → WSWAPREQSELB~!:(WSWAPREQSA~ & MULTIPLYB & !WSWAPAKA~)
- 57 → WSWAPAKA~!:(WSWAPREQSDELA~ & WSWAPAKA~# !WSWAPAK~ & !WSWAPREQS~)
- 59 → WSWAPAKB~!:(WSWAPREQSELB~ & WSWAPAKB~# !WSWAPAK~ & !WSWAPREQS~)
- 60 → WSWAPAK~!:(WSWAPAKA~ & !WSWAPAKB~)

70 {

LEGEND:

- : = registered output
- | x is true when x is low
- y: = !(!x) means y will go low on the next clock if x is low
- # = Logical OR
- & = Logical AND

FIG.7b

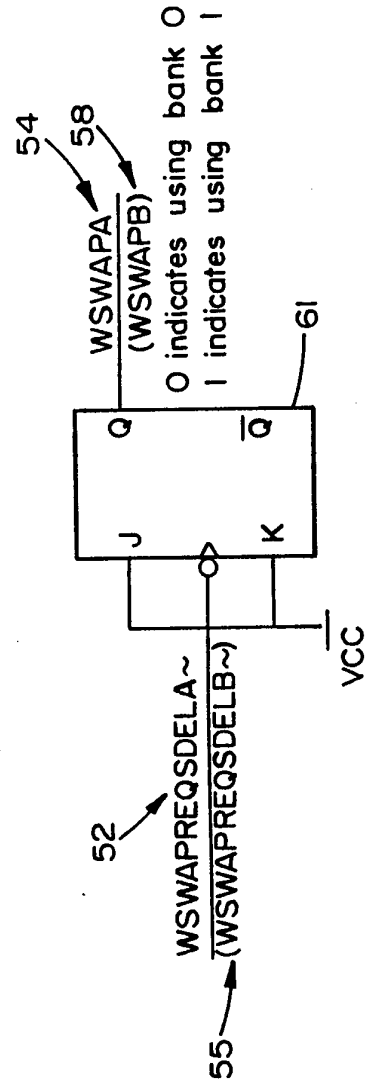


FIG.7c



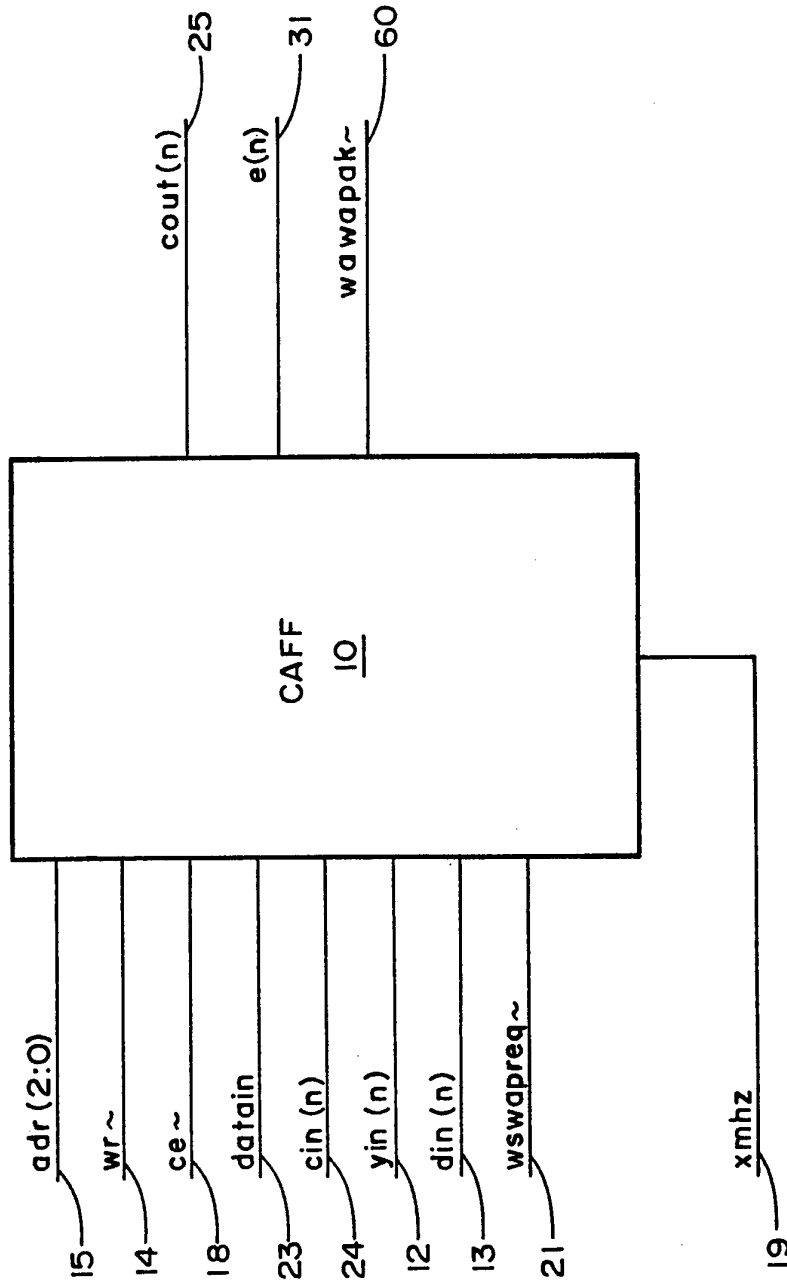


FIG.9

**INTERNATIONAL SEARCH REPORT**

International application No.  
PCT/US94/05831

<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC(5) :G06F 15/31 US CL :364/724.16, 724.19 According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) U.S. : 364/724.16, 724.19, 724.13 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US, A, 4,893,265 (HIROSAKI) 09 JANUARY 1990, SEE THE ENTIRE DOCUMENT	1-35
A	US, A, 4,947,362 (BUI) 07 AUGUST 1990, SEE THE ENTIRE DOCUMENT	1-35
A, P	US, A, 5,245,561 (SUGIYAMA) 14 SEPTEMBER 1993, SEE THE ENTIRE DOCUMENT	1-35
A, P	US, A, 5,280,255 (NISHIKAWA) 18 JANUARY 1994, SEE THE ENTIRE DOCUMENT	1-35
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: *A* document defining the general state of the art which is not considered to be part of particular relevance *E* earlier document published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art *Z* document member of the same patent family	
Date of the actual completion of the international search 21 JULY 1994		Date of mailing of the international search report 01 AUG 1994
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230		Authorized officer <i>B. Malzahn</i> DAVID MALZAHN <i>for</i> Telephone No. (703) 305-9600