A FAMILY OF CMOS FLOATING POINT ARITHMETIC CHIPS

Dr. John A. Eldon
Manager, Advanced Development

TRW LSI Products
La Jolla, California

1.0 BACKGROUND

Although the advantages of floating point arithmetic have long been recognized, hardware complexity and expense have impeded its use in high speed digital signal processing (DSP). Now, however, the availability of a growing number of fast dedicated floating point adder and multiplier chips is spurring renewed interest in floating point for real time filtering and spectral analysis.

As part of this significant trend, TRW LSI Products is introducing a family of CMOS floating point chips, starting with two adders and two multipliers. As discussed below, these chips permit efficient hardware implementation of the IEEE's "754 - Revision 10.0" 32-bit floating point standard and offer some unique features to simplif signal processor system design.

1.1 Formats

Among the several floating point formats which have been proposed and used (IBM's hexidecimally-normalized 32-bit; DEC's 32-bit and high precision and high range 64-bits; Mil Standard 1750A's two's complement 32-bit; and IEEE's 32-bit, 64-bit, and extended), TRW has selected the 32-bit IEEE standard for this first family of CMOS chips. Although this format requires more complex hardware than any other 32-bit, it also offers greater accuracy and less arithmetic noise. If market interest warrants, future chips supporting the other formats can be derived from the IEEE chips relatively easily.

Because direct implementation of the IEEE standard's "gradual underflow" requirements would complicate the chips (particularly the multiplier) considerably, TRW has developed a special 34-bit version of the IEEE single precision format. This format's 10-bit two's complement exponent expands the dynamic range of normalized floating point numbers upward and downward, as illustrated in Figure 1.1. As discussed later, the support of both 32 and 34 bit formats permits the TRW chips to avoid overflow and underflow errors during long computation strings, while retaining full IEEE 32-bit compatibility at the inputs and outputs of the arithmetic processor board. The use of the 34-bit format also avoids the time consuming "wrap/unwrap" overflow/underflow treatment used in other floating point chip sets, while offering even greater protection against all out-of-range errors.

1.2 Architectural Objectives

A floating point arithmetic chip set must offer high accuracy and low arithmetic noise and should comply with a recognized numeric format. Furthermore, the chip's input/output configuration, internal registration structure, and instruction set must support the common DSP algorithms smoothly and efficiently, with a minimum of external registers, multiplexers, and other "adhesive" chips. As discussed below, the new TRW chip set meets all of these requirements.

2.0 ARITHMETIC UNITS

Currently, TRW's family of floating point chips includes two arithmetic units, both of which implement the 32-bit IEEE and 34-bit TRW formats. The TMC3200, offered in an 84 contact leadless chip carrier (LCC) and an 88 pin grid array (PGA), uses three 17-bit timeshared buses, two for input and one for output. Because they are half-size, the buses are run at twice the speed of the chip's internal 10 MHz hardware. Housed in a 132 PGA, the TMC3202 features three full width 34-bit buses. Thus, the user can trade package and bus size against bus bandwidth.

2.1 Architectures

Figure 2.1, a block diagram of the TMC3200, shows the input register configuration, the internal pipelining structure, and the overall data flow for the exponent and significand. The internal pipelining register can be enabled for double speed operation or disabled for simpler recursive and accumulation timing. The internal accumulation path is particularly useful in real time filtering and spectral analysis, where a series of terms or products must be accumulated.

To reduce the load on the system data bus, the TMC3202 features four input registers (Figure 2.2). One of these supports the internal accumulation path, which writes each emerging result back to one input of the adder. A second register, generally wired to the output of a multiplier chip, can be routed to either input of the adder. Finally, the third and fourth registers connect one input of the adder to the remaining parallel input port, which is typically connected to the system's external data bus. The applications discussion illustrates the use of these registers in a Fast Fourier Transform (FFT) butterfly and in a finite impulse response (FIR) filter. To match the FFT performance of a TMC3202/3 set, a TMC3200/1 set would require several external multiplexers and registers.

CH2146-9/85/0000/0101$01.00 © 1985 IEEE

2.2 Instruction Sets

Tables 2.1 A and B summarize the two chips' three-bit operand source instruction sets, which connect the arithmetic cores to the input registers and accumulation paths. These instructions support two-port addition (A+B) and single port accumulation($\sum_i A_i$).

Table 2.2 summarizes the chips' common five-bit operation instruction set. The chips support 32- and 34-bit floating point addition and subtraction, as well as conversion among 24-bit integer and 32- and 34-bit floating point formats. Irrespective of the instruction, the accumulation path always bears full 34-bit data. A fast IEEE-compatible accumulation is implemented using the 32-bit add and "select input A and accumulation path" instructions. Although the inputs and outputs will be in standard 32-bit IEEE format, the internal accumulation will be computed to 34-bit range, allowing user-transparent real time recovery from temporary overflows or underflows. If the final result underflows below the IEEE's minimum normalized value of 2^{-126}, the user can then use one additional cycle to convert the 34-bit accumulation into an IEEE compatible denormalized 32-bit output, if desired. Otherwise, a 32-bit IEEE zero will be output, along with the underflow flag. Handling of denormalized incoming operands is user-transparent: when a 32-bit denormalized number is input, its hidden bit is forced to zero, its exponent to 1. The resulting equivalent value is then used in all internal computations.

The TMC3200/2's handling of denormalized inputs and internal (34-bit) accumulations is fast, convenient, and accurate. However, to minimize arithmetic roundoff noise, the user should try to scale the data base to avoid accumulating denormalized values. If a denormalized 32-bit addend is combined with a small (exponent $\leq$ o) 34-bit cumulative sum, the former's internal exponent of 1 will always force a denormalizing right shift of the latter, irrespective of the relative magnitudes of the two numbers. Excessive or unnecessary denormalization generally increases arithmetic noise in the system.

The adders recognize the standard IEEE traps of 0, not a number (NAN), and infinity. Furthermore, they output overflow (OV), underflow (UN), zero, and illegal operation (IOP) status flags for convenient system monitoring. NAN plus anything or infinity minus infinity is a NAN with IOP flag; infinity plus or minus any normalized number is infinity without OV flag; and the overflowing sum of two normalized numbers is infinity with OV flag. The underflowing small difference between two nearly equal numbers is a 0 with UN flag, whereas the operation "A-A=0" yields a 0 without UN flag. The IOP flag accompanies all NAN outputs and the zero flag accompanies all 0 outputs, simplifying the detection of these conditions.

3.0 MULTIPLIERS

The chip family's two multipliers also support the 32 bit IEEE and 34 bit TRW formats. The TMC3201 multiplier, companion to the TMC3200 arithmetic unit, also has three 17 bit buses and an 84/88 pin package (Figure 3.1). The 132-pin TMC3203 multiplier, with three 34 bit buses, is further enhanced with four input registers, two for each input port (Figure 3.2). This supports complex arithmetic by permitting the user to alternate between the imaginary and real components of the data and coefficient after loading them only once.

The multipliers' instruction sets are very limited:

1. Input format 32/34 bits;

2. Output format 32/34 bits (used in conjunction with instruction #1 for conversions, if required);

3. Rounding to nearest/truncation toward 0, both of which are recognized by the IEEE format.

In the TMC3201, instructions 1 and 2 are merged onto a single global pin. Due to hardware complexity, the multipliers do not support denormalized numbers explicitly, but flush all denormalized inputs to zero instead. Where greater dynamic range is needed, these chips can be operated in 34 bit mode. In a typical FIR filter or FFT butterfly, the TMC3203 chip can be set to read in 32 bit operands and put out 34 bit results, avoiding the potential for overflow or underflow, while avoiding the complications of denormalized numbers. The accompanying arithmetic unit could then be set for 34 bit inputs and 32 bit outputs, completing an arithmetic element which takes in 32 bit information, processes it to 34 bit range, and puts out 32 bit results when finished.

The multipliers recognize the IEEE format's 0, infinity, and NAN special cases. Specifically, attempting to multiply 0 times infinity or a NAN times anything yields a NAN. Infinity times anything is infinity without OV flag; the overflowing product of two normalized numbers is infinity with OV flag. Zero times any noninfinite value is 0 without UN flag, and the underflowing product of two normalized numbers is 0 with UN flag. As in the adder, all outputs of NAN are accompanied by the IOP flag.

4.0 APPLICATON TO A DIGITAL SIGNAL PROCESSOR

Figure 4.1 is a block diagram of a general purpose arithmetic element built out of a TMC3202 and a TMC3203. The inputs and outputs can be connected to separate input and output buses for a full-speed pipelined system, or to a shared I/O data bus, as shown here. The single Exclusive OR gate at the TMC3202's sign output is needed for the 8-cycle pipelined FFT butterfly described below, although it can be omitted if the slower feedthrough mode is acceptable or if only spectral power (and not phase) information is required. (Without the extra gate, half of the final samples are inverted, significantly complicating reconstruction of the original signal via inverse FFT.)

4.1 FFT Butterfly

Table 4.1 lists the operations necessary to execute an 8-cycle radix 2 FFT butterfly with this arithmetic element. Note the use of the chips' extra A-input registers to reduce the load on the data bus; without these registers, either external registers and multiplexers or more clock cycles would be required. The implementation shown uses the pipelined mode, with real and imaginary components "chasing" one another inside each chip. If the chips are used in their slower feedthrough mode, the external exclusive OR gate can be eliminated (Table 4.2).

With the architecture of Figure 4.1 and a 10 MHz master clock, the two chip set can execute the radix 2 butterfly in 0.8µsec (pipelined) or 1.6µsec (feedthrough, using the maximum clock rate of 5 MHz).

Although the implementation of Figure 4.1 will execute the butterfly with very convenient sequencing of inputs and outputs, it uses its multiplier at only a 50% duty cycle, effectively wasting half of the investment in this part. (Of course, as discussed later, it uses the multiplier 100% efficiently in a convolution, matrix multiplication or FIR filter application, where the desired ratio of additions to multiplications is 1:1.) Figure 4.2 suggests an enhanced, equally general purpose arithmetic element architecture, which can execute a 6 cycle nonpipelined FFT butterfly. Unfortunately, this 6 cycle operation runs at 5 MHz rather than 10, taking 1.2 µsec per butterfly and requiring the user to provide separate data input and output buses plus a 32-bit external 2:1 multiplexer.

Theoretically, the fastest possible FFT butterfly for a single-TMC3202 system is 0.6 usec, or 6 cycles at 10 MHz. Figure 4.3 illustrates the external hardware needed to support this type of performance. The registered 8-bit fixed point adder module allows full speed pipelined doubling of data values, facilitating the computational shortcut illustrated in Table 4.3.

4.2 FIR Filter/Convolution

The same general purpose arithmetic element can execute a convolution or vector inner multiplication easily and efficiently, at a rate of one clock cycle per filter tap. Here, the user may prefer to turn off the pipeline register and operate the chip at 5 MHz, to avoid the timing complications of a two-stage feedback path. However, as Table 4.4 illustrates, the pipeline register can still be used advantageously, albeit with more complicated data addressing.

With one multiplier and one adder chip, an arithmetic element can't take advantage of the coefficient symmetry of a linear phase FIR filter by preadding pairs of data point before multiplication. If the user can tolerate the extra address and memory interface complications and wishes to provide a second adder for the preadditions, then the performance of the FIR filter can be doubled, since the number of multiplications is cut in half. Floating point is particularly attractive for this algorithm, since it avoids the word growth problems associated with fixed point addition.

5.0 CONCLUSIONS

The architectures and data formats of the growing family of TRW 32/34 bit floating point chips enable them to execute common DSP operations adroitly. Parts have been developed for system data bus widths of both 16 and 32 bits. The 32-bit parts in particular are designed to require a minimum number of external registers, multiplexers, and other glue chips. Although pipelining was necessary in all four chips to ensure adequate speed performance, the user can generally work around it, as illustrated above. In recursive filtering or other applications in which the pipelining would create serious timing and data flow problems, the user can always use the feedthrough control and strobe the chip's registers at half speed (5 MHz).

FIGURE 1.1 FLOATING POINT FORMATS

REP.	VALUE	REP.	COMMENTS
IEEE 32 BIT		TRW 34 BIT	
FF 000000	INFINITY	1FF 000000	MOST POS EXP (a)
	2^{510} *1.9999	1FE 7FFFFF	MAX NORM TRW (b)
FE 7FFFFF	2^{254} *1.9999	0FE 7FFFFF	MAX NORM IEEE (c)
01 000000	2^{-126} *1.0000	001 000000	MIN NORM IEEE (c)
00 7FFFFF	2^{-126} *0.9999	000 7FFFFE	MAX DEN IEEE (d)
00 000001	$2^{-126-23}=2^{-149}$	3EA 000000	MIN DEN IEEE (e)
	2^{-511} *1.0	101 000000	MIN NORM TRW (f)
00 000000	ZERO	100 000000	MOST NEG EXP (g)

(a) Append 01 to IEEE for TRW
(b) No direct equivalent (flush to infinity) in IEEE
(c) Append 00 to IEEE for TRW throughout normalized IEEE range
(d) LSB=0 for TRW equivalent in this range, since IEEE loses 1 bit of precision (hidden bit = 0)
(e) IEEE denorm numbers with significand MSB=0 (lead digit less than 4 in this table) don't resemble their (normalized) TRW equivalents, due to significand shifting/exponent readjusting
(f) No direct equivalent (flush to zero) in IEEE
(g) Append 10 to IEEE for TRW

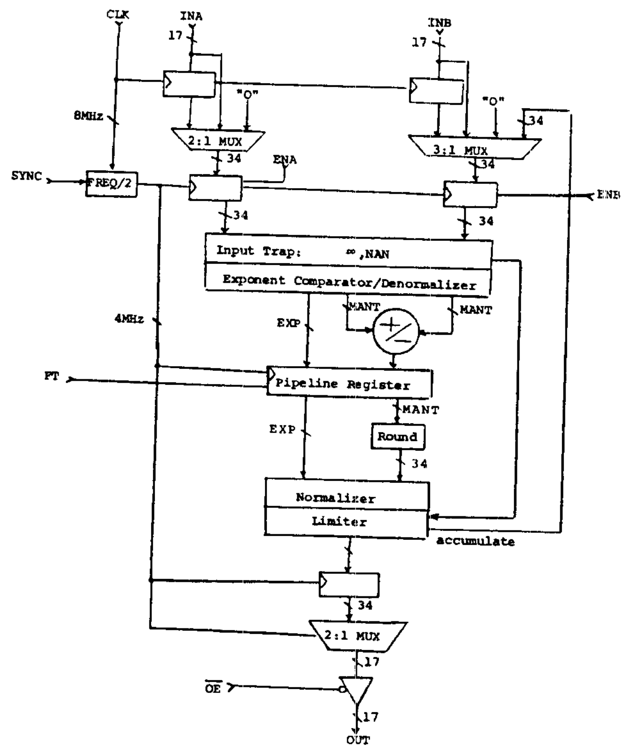

FIGURE 2.1 TMC3200 SIMPLIFIED BLOCK DIAGRAM

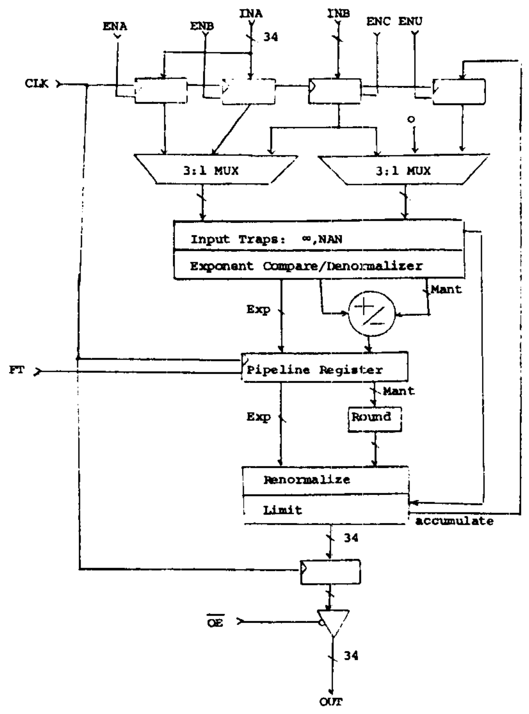

FIGURE 2.2 TMC 3202 BLOCK DIAGRAM

TABLE 2.1A TMC3200 OPERAND SOURCE INSTRUCTIONS

S2	S1	S0	A OPERAND SOURCE	B OPERAND SOURCE
0	0	0	A-BUS	B-BUS
0	0	1	A-BUS	U-BUS*
0	1	0	0	B-BUS
0	1	1	A-BUS	0
1	0	0	Magnitude (A)	Magnitude (B)
1	0	1	Magnitude (A)	Magnitude (U)
1	1	0	0	Magnitude (B)
1	1	1	Magnitude (A)	0

* U is the Accumulate path and will always provide a preceding result that was generated in a prior instruction. Integers in the U path are not supported.

TABLE 2.1B TMC3202 OPERAND SOURCE INSTRUCTIONS

S2	S1	S0	A OPERAND SOURCE	B OPERAND SOURCE(1)
0	0	0	B	0
0	0	1	A_1	0
0	1	0	A_1	U
0	1	1	A_1	B
1	0	0	B	U
1	0	1	A_2	0
1	1	0	A_2	U
1	1	1	A_2	B

In this table,

(1) "A_1" "A_2" "B" and "U" refer to four registers located near the chip's input port (see chip block diagram).

(2) U is the Accumulate path and will always provide a preceding result that was generated in a prior instruction. The U path does not support integers.

Table 2.2 TMC3200, TMC3202 OPERATION INSTRUCTION SET

	M2 M1 M0							
OP1 OP0	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0	A+B F2+F2 ROUND	A+B F2+F2 TRUNC	A+B F4+F2 ROUND	A+B F4+F4 TRUNC	A+B F4+F4 ROUND	A+B F4+F4 TRUNC	A+B F2+F4 ROUND	A+B F2+F4 TRUNC
0 1	A-B F2+F2 ROUND	A-B F2+F2 TRUNC	CONVB F2+F4 ROUND	CONVB F2+F4 TRUNC	A-B F4+F4 ROUND	A-B F4+F4 TRUNC	CONVB F4+F2 (DEN) ROUND	CONVB F4+F2 (DEN) TRUNC
1 0	B-A F2+F2 ROUND	B-A F2+F2 TRUNC	CONVB I+F2 X	CONVB I+F2 X	B-A F4+F4 ROUND	B-A F4+F4 TRUNC	CONVB I+F4 X	CONVB I+F4 X
1 1	-A-B F2+F2 ROUND	-A-B F2+F2 TRUNC	CONVB F2+O TRUNC	CONVB F2+O TRUNC	-A-B F4+F4 ROUND	-A-B F4+F4 TRUNC	CONVB F4+I TRUNC	CONVB F4+I TRUNC

F2, F4 = 32-Bit and 34-Bit Floating-Point formats respectively.
I = Integers (Fixed Point).
X = Don't Care.
NOTE: For all CONVB (ConvertB) instructions, the A-Operand field is ignored.

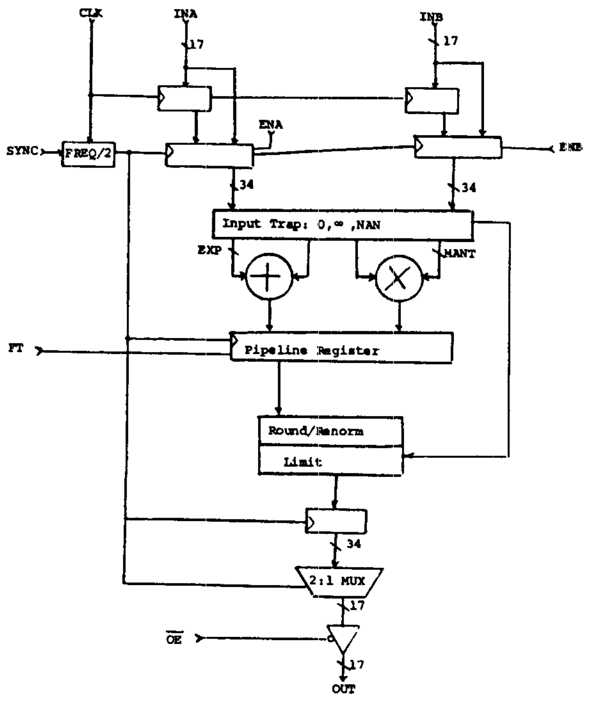

FIGURE 3.1 TMC3201 BLOCK DIAGRAM

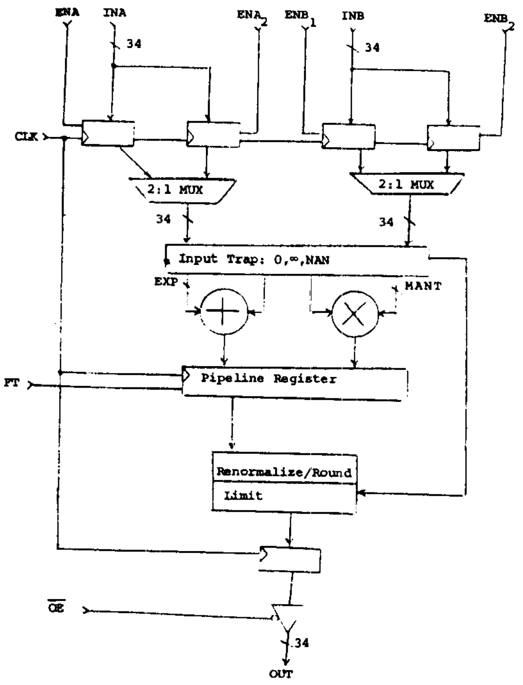

FIGURE 3.2 TMC3203 BLOCK DIAGRAM

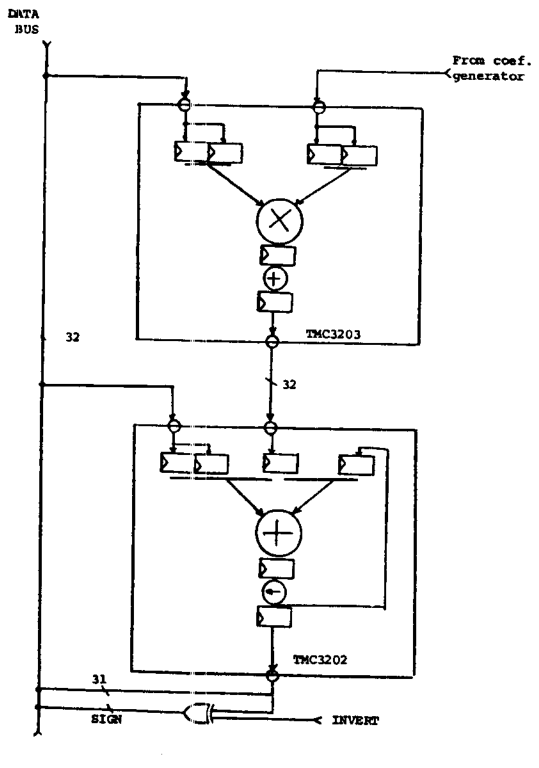

FIGURE 4.1 GENERAL PURPOSE FLOATING POINT ARITHMETIC ELEMENT

TABLE 4.1 8-CYCLE PIPELINED RADIX 2 BUTTERFLY

	Multiplier Registers (a)				Adder Registers (b)					
BUS	MD_1	MD_2	MP	MR	AA	AB	AC	AU	AP	AR
C(c)	C									
D	C	D								
________	C	D								
________	C	D	CR(d)							
A	X	D	CI	CR	A					
B	X	D	DI	CI	A	B	CR	X		
________	X	X	DR	DI	A	B	CI	X	A+CR	
________	X	X	X	DR	A	B	DI	A+CR	B+CI	A+CR
________		X	X	X	A	B	DR	B+CI	A+CR-DI	B+CI
________			X	X	A	B	X	A+CR-DI	B+CI+DR	A+CR-DI
A+CR-DR			X	X	A	B	X	B+CI+DR	CR-DI	B+CI+DR
B+CI+DI				X	A	B	X	CR-DI	CI+DR	CR-DI
________						B	X	CI+DR	-A+CR-DI	CI+DR
________									-B+CI+DR	-A+CR-DI(e)
A-(CR-DR)										-B+CI+DR(e)
B-(CI+DI)										

(a) MDi = multiplier data input registers
MP = multiplier pipeline register
MR = multiplier output (result) register

(b) AA, AB, AC = adder input registers
AU = adder accumulator register
AP = adder pipeline register
AR = adder output (result) register

(c) A, B = real and imaginary components of first data point
C, D = real and imaginary components of second data point

(d). CR = real data * cosine (R)
CI = real data * sine (I)
DR = imaginary data * cosine (I)
DI = imaginary data * sine (R)

(e) Inverter (at sign bit of TMC3202) used to rectify result

TABLE 4.2 SIX CYCLE NONPIPELINED BUTTERFLY

	Multiplier			Adder					
IN BUS	MD_1	MD_2	MR	AA	AB	AC	AU	AR	OUT BUS
C	C								
D	C	D							
(PREV B)	C	D	CR						
________	C	D	DI			CR			
________	C	X	DR	X	DI	CR			
A	X	X	CI	A	X	DR	CR-DI	CR-DI	
(NEXT C)		X	X	A	CI	DR	CR-DI	A+CR-DI	
(NEXT D)			X	X	CI	DR	X	A-CR+DI	A'
B				B	X	X	CI+DR	CI+DR	C'
________				B	X		CI+DR	B+CI+DR	
________							X	B-CI-DR	B'
									D'

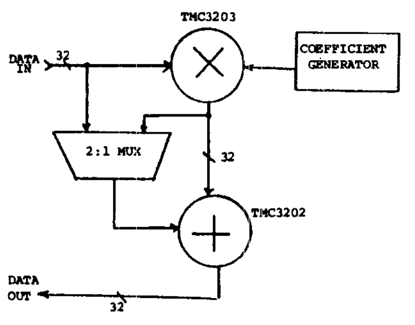

FIGURE 4.2 ARITHMETIC ELEMENT FOR 6-CYCLE NONPIPELINED BUTTERFLY

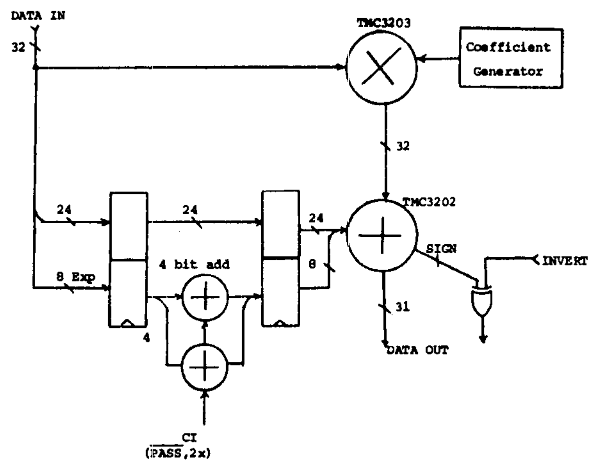

FIGURE 4.3 ARITHMETIC ELEMENT FOR 6 CYCLE PIPELINED BUTTERFLY

TABLE 4.3 SIX CYCLE PIPELINED BUTTERFLY

IN BUS	MD_1	MD_2	MP	MR	AA	AB	AC	AU	AP	AR
C	C									
D	C	D								
A	C	D	CR							
B	X	D	CI	CR						
A	X	D	DI	CI	A	X	CR	X		
B	X	X	DR	DI	X	B	CI	X	A+CR	
		X	X	DR	2A	X	DI	A+CR	B+CI	A+CR
			X	X	2A	2B	DR	B+CI	A+CR-DI	B+CI
				X	2A	2B	X	A+CR-DI	B+CI+DR	A+CR-DI
					X	2B	X	B+CI+DR	-A+CR-DI	B+CI+DR
									-B+CI+DR	-A+CR-DI
										-B+CI+DR

TABLE 4.4 PIPELINED FIR FILTER - 100 NSEC PER TAP

	Multiplier Registers				Adder Registers			
BUS(a)	MD_1	MD_2	MP	MR	AC	AU	AP	AR
	1(c)							
2	2		1A(b)					
			2A	1A				
3	3		2B	2A	1A			
			3B	2B	2A		1A	
4	4		3C	3B	2B	1A	2A	1A
			4C	3C	3B	2A	1A+2B	2A
5	5		4D	4C	3C	1A+2B	2A+3B	1A+2B
			5D	4D	4C	2A+3B	1A+...+3C	2A+3B
6	6		5E	5D	4D	1A+...+3C	2A+...+4C	1A+...+3C
			6E	5E	5D	2A+...+4C	1A+...+4D	2A+...+4C
7		7	6F	6E	5E	1A+...+4D	2A+...+5D	1A+...+4D
3		3	7F	6F	6E	2A+...+5D	1A+...+5E	2A+...+5D
4		4	3A	7F	6F	1A+...+5E	2A+...+6E	1A+...+5E
			4A	3A	7F	2A+...+6E	1A+...+6F	2A+...+6E
5		5	4B	4A	3A	X	2A+...+7F	1A+...+6F(d)
I(d)			5B	4B	4A	X	3A	2A+...+7F(e)
II(e)			5C	5B	4B	3A	4A	3A
			6C	5C	5B	4A		4A
7	7			6C	5C			
					6C			
8	8							
9								

(a) Data in/out common bus
(b) A-F = coefficients; 1A is first data x first coefficient
(c) 1,2,3, --- = data
(d) First full result out = I
(e) Second full result out = II