VLSI FLOATING–POINT PROCESSORS

Jan Fandrianto – B. Y. Woo

WEITEK CORPORATION
501 Mercury Drive
Sunnyvale, California 94086

ABSTRACT

The advance of VLSI technology has been the enabling factor in the appearance of VLSI circuits handling floating-point arithmetics. These circuits have found their way into many number–crunching applications such as telecommunications, seismic energy exploration, radar, medical imaging, graphics and simulation. Because of the different requirements for different applications, some processors have a rich repertoire of functions but rather low performance, while some processors aim at having the highest throughput for the most frequent operations such as multiply and add. This paper will review the architecture, the technology, and the design techniques for the current VLSI floating–point processors; and it will also report a high performance chipset (*) implementing complete basic arithmetic functions.

I. Introduction

The motivation for using floating–point data format to handle arithmetic operations is due to its large dynamic range and high precision. However, since the hardware cost of implementing full floating–point arithmetic has been prohibitive, many have been using integer or block floating–point arithmetic instead. As a result, extra care needs to be exercised to take care of overflow, underflow and roundoffs. Block floating–point format has the advantage that the dynamic range can be adjusted to the biggest number. However, small numbers would have large noise to signal ratio. There are two driving forces that are changing the picture. First is the appearance of standards, among them the IEEE Binary Floating Point Arithmetic Standard (Task P754) [1] is the dominant one. This will make the software portable. Second is the advances of VLSI technology. As the level of integration rises, it becomes possible to implement the hardware support for all or most of the floating–point arithmetic operations in one or two chips.

It is the intent of this paper to present an overview of the current VLSI floating–point processors, in terms of their architecture, their interface with the host CPU and their functionality. Basically we can categorize them into two main groups: general–purpose and building block processors. General–purpose processors usually have a rich repertoire of functions. Their hardware consists of a set of basic arithmetic elements such as an adder, on–chip storage and control store. They rely on the built–in microcode routines to execute the arithmetic operations. Versatility and moderate performance are their features. On the other hand, the building–blocks optimize the performance on the key arithmetic functions such as add and multiply. They have dedicated hardware such as multiplier array to offer the utmost speed required. Depending on the applications, the building blocks can be configured accordingly to maximize the performance and throughput.

Table 1 lists some of the key application areas for the VLSI floating–point processors.

II. General–purpose Floating–point Processors

Processors falling in this category are Intel 8087/80287, National 32081, Motorola 68881, Zilog 8070, AMD 9511A/9512 and Fairchild F9450.

A. Intel 8087/80287 [2]

1. Architecture and Interface

8087 is designed to couple tightly with the host CPU, which in this case is 8086. In order to maximize the system throughput, the coprocessor duplicates the instruction queue and operand queue of the host. When one of the ESCAPE instructions is encountered, the host calculates the address and fetches the operand, and the coprocessor captures the address and operand and starts the computation and releases the bus. In doing so no extra bus cycle is needed for handshaking.

There are eight 80–bit registers that serve as a standard register set or as a stack. Associated with each of the registers, there is a two–bit Tag Word to indicate whether its content is valid, zero, special (Not–a–Number, infinity, etc.) or empty. On chip there are a 68–bit wide ALU and a 63–bit wide barrel shifter controlled by microcodes. Modified Cordic algorithm is used to calculate transcendental functions.

2. Functionality and Data Formats

The processor conforms to the IEEE standards and handles seven data types:16–bit, 32–bit and 64–bit two's complement integers, 32–bit, 64–bit and 80–bit floating–point numbers and 80–bit signed packed BCD. Arithmetic functions include add, subtract, multiply,

(*) Weitek WTL1164/1165

CH2146-9/85/0000/0093$01.00 © 1985 IEEE

divide, square root, transcendental functions (exponential, logarithmic and trigonometric functions) and binary to/from decimal conversion. On-chip recovery from overflow and underflow is provided.

B. Motorola 68881 [3]

1. Architecture and Interface

The processor serves as an extension of the host CPU by providing eight 80-bit data registers, a status register, a control register and an instruction register. These registers are memory-mapped into the CPU address space.

The interface between the processor and host CPU is asynchronous. If the host CPU has a direct hardware support for the floating-point processor, such as 68020, it will recognize the 68881 instructions and perform the protocol to exchange machine instructions, operands and results on a 32-bit data bus, which can also be configured as an 8-bit or a 16-bit bus. Otherwise a F-line trap handler is invoked and software routine is used to complete the interface. In any case, the floating-point processor operations start with fetching one of the 68881 instructions (F-line instructions). The host CPU interprets the operation word in the instruction and write the command word to the 68881 command register. 68881 decodes the command word and provides a response to the host in the form of primitives. There are 18 primitives divided into five groups: processor/coprocessor synchronizations, instruction stream manipulation, exception handling, general operand transfer and register transfer. The host will then decide whether no actions is needed or some additional service must be performed, such as calculating the effective address, fetching variable-length operand or handling exception flags.

2. Functionality and Data Formats

68881 implements the IEEE floating-point standards. It handles seven data types: 8-bit (byte), 16-bit (word), 32-bit (long integer) two's complement integers, 32-bit, 64-bit and 80-bit floating-point number and 96-bit packed BCD.

Arithmetic functions include add, subtract, multiply, divide, square root, modulo remainder, IEEE remainder and all transcendental functions. In addition, operations such as compare, absolute value, exponent extract, mantissa extract, scaling and test can also be performed.

Data manipulating instructions include move to and from 68881, branch, save and restore. The last two are for virtual memory support so that the current state can be saved and restored in case of page fault.

C. National Semiconductor 32081 [4]

1. Architecture and Interface

32081 is designed to be a slave to the CPU. CPU is responsible for the fetching of the opcode and operands and sending them to the processor. The communication between the CPU and 32081 is via a two-bit protocol, a slave processor control line and a 16-bit bidirectional data bus.

There are three major functional blocks: the control unit, the execution unit and the interface and storage unit. The micro-instruction includes part of the next instruction's address and part of the condition code. The execution unit has separate hardware for exponent, fraction and sign computation. The interface and storage unit takes care of the control and data interface with external world. There are eight 32-bit data registers, which can be paired together to become four 64-bit registers. In addition, there is a 32-bit Floating Status Register, storing the rounding, underflow, inexact and trap type informations.

2. Functionality and Data Formats

The instruction set includes basic operations such as add, subtract, multiply, divide, compare, negate and absolute value. Also it can perform data type conversion, floating-point to and from integer, single precision to and from double precision. Five data types can be handled: IEEE single and double precision floating-point formats, 8-bit, 16-bit and 32-bit two's complement integers.

D. Others

1. AMD 9511/9512 [5]

These chips have a stack-oriented organization, with only one of the floating-point operands can be supplied from the stack at one time. 9511 handles a special 32-bit floating-point format which is different from the IEEE format, whereas 9512 uses the IEEE single and double precision formats. Data transfer between 9511/9512 and the host is via programmed I/O or DMA. End of execution is signalled as an interrupt to the CPU.

2. Fairchild F9450 [6]

This microprocessor fully implements the military instruction set architecture (MIL-STD-1750A ISA). Data types include bit, byte, 16-bit, 32-bit integer, single-precision(32-bit) and extended-precision (48-bit) floating-point numbers.

3. Zilog Z8070 [7]

Z8070 interprets the opcode fields in the extended instructions and carries out the proper operations. Internally it is structured as two separate processors: the interface processor and the data processor. The former fetches and aligns instructions and data, schedules operations using an instruction queue. The latter carries out the arithmetic operations, which include add, subtract, multiply, divide, square root, remainder, negate, compare and absolute.

Table 2 summarizes the important features of various general-purpose floating-point processors.

III. Building-block Floating-point Processors

The building-block approach is used for the most speed-demanding applications. In the VLSI implementation, the hardware support includes a high-speed multiplier array, a fast carry-look-ahead adder, a priority encoder and barrel shifters (both left and right shifters). As the technology advances, this type of processors starts to emerge, first among them being the Weitek WTL1032/1033 chip-set.

A. Weitek WTL1032/1033 [8]

1. Architecture and Interface

The chip set consists of a multiplier (WTL1032) and an adder (WTL1033). To increase the throughput, the array is partitioned into a three-stage pipe. At a clock rate of 20MHz, a maximum throughput of 10 Megaflops per chip can be achieved. A mode bit can be set to make the pipeline registers transparent in order to reduce the latency for scalar computations. There are three 16-bit data busses, two for the input and one for the output. The I/O bandwidth completely matches with the throughput of the array. The circuits can be configured in such a way that 100% of the pipe speed can be achieved in the applications such as FFT butterfly [9] and a 4 X 4 matrix multiplication for graphics computations.

2. Functionality and Data Formats

This chip-set supports the IEEE single-precision binary floating-point standard. In addition to the basic arithmetic operations such as multiply, add and subtract, the adder can also perform floating-point to integer conversion and vice versa and absolute add. Data types include 32-bit floating-point and 24-bit two's complement integer. To handle gradual underflow, the chip-set also supports operations on wrapped numbers, a normalized floating-point number with the exponent wrapped around to a negative value.

B. TRW TDC1022/1042 [10]

1. Architecture and Interface

TDC 1022 and TDC1042 are a 22-bit floating-point adder and multiplier respectively. Both chips implement a two-stage pipe in order to double the throughput. A feedback path is provided on chip so that continuous accumulation or multiplication can be performed. Both chips have a 22-bit output port. TDC1042 has two 22-bit input ports, whereas TDC1022 has only one.

2. Functionality and Data Formats

Aside from add, subtract and multiply functions, the chips can also perform zeroing, saturating, division by two, normalizing and denormalizing. The floating-point format has a 6-bit two's complement exponent and a 16-bit two's complement mantissa with the binary point to the right of the most significant bit.

C. Weitek WTL1064/1065 [11]

1. Architecture and Interface

The 64-bit floating-point chip-set consists of a multiplier (WTL1064) and an adder (WTL1065). In order to reduce chip area, an iterative multiplier array is used. For single-precision multiply, data has to pass through the array twice, and for double-precision, four times. A two-stage pipe is used for the multiplier and four-stage for the adder. Each of these pipes can be made transparent via a mode bit. At 16 MHz clock, the multiplier delivers 4 Megaflops performance for single precision, 2 Megaflops for double precision, and the adder yields 8 Megaflops for either precision operations.

Normally, the data loading and unloading are via two 32-bit input ports and one 32-bit output port. However, the input and output can be configured via the mode control to be a 64-bit input, single 32-bit port or two 16-bit input ports, and the output, either 32-bit or 16-bit port. User can select whether the most or least significant of operands should be loaded or unloaded first in the case of double precision operations.

2. Functionality and Data Formats

In addition to handling IEEE single and double precision data formats, the chip set also supports DEC F(32-bit) and D(64-bit)formats. Conversion from either of these formats to 32-bit two's complement integer and vice versa can also be performed. Functionality is similar to WTL1032/1033.

C. AMD 29325 [12]

1. Architecture and Interface

This design integrates both adder and multiplier on the same chip. It has a three-port architecture, two 32-bit input ports and one 32-bit output port. It is designed for one cycle latency for all operations. On chip there is a feedback path so that chained operations can be achieved without the results going off chip. The user can have the option of using just one 32-bit input port or have the port split into two 16-bit slices, with the operands loaded at the rising and falling edges of the clock.

2. Functionality and Data Formats.

Besides performing add, subtract and multiply, it can also perform 2-B (for an iterative divide operation where B is one of the input operands), and conversion from floating-point to 32-bit two's complement integer. The processor supports both IEEE single precision and DEC F(32-bit) data formats.

D. Analog Devices ADSP3210/3220 [13]

1. Architecture and Interface

A floating-point multiplier (ADSP3210) and a floating-point adder (ADSP3220) are the components of this chip-set. ADSP3210 has a 32 X 32 parallel multiplier array, implementing Booth and Wallace Tree algorithms. The data path is divided into a two-stage pipe; the first stage performs the mantissa multiply and the second does format adjust, rounding and exception detection. The pipe delay can be 100ns. Thus for single precision multiply, it can deliver 10 Megaflops performance. For double precision, however, iterative multiplication has to be done because of the size of the multiplier array, the performance is thus reduced to 2 Megaflops. Similarly, there is a two-stage pipe on the ADSP3220. The data I/O for both chips is via a 32-bit input bus and a 32-bit output bus. 32-bit operands can be loaded or unloaded every 50ns. Although the pipe delay is only 100ns, for double precision add, the adder can only deliver 5 Megaflops performance because the speed is I/O limited.

2. Functionality and Data Formats

In addition to the arithmetic operations for single and double precision data in IEEE formats, 32-bit two's complement integer arithmetics (multiply and add) and logic functions are supported. It also handles gradual underflow by wrapping denormalized numbers.

Table 3 summarizes the key features of the building-block VLSI processors.

IV. General Purpose Building-block Floating-point Processors

Weitek WTL1164/1165 [14]

The chip-set is designed to provide a high-speed 32-bit and 64-bit floating-point processing capability for general purpose computer systems at supermicrocomputer and superminicomputer levels. The chip-set design goals are to achieve short computational latency for increased scalar throughput, small die sizes for low cost processing and packaging, and an expanded set of operations to meet the requirements for the general purpose application.
The chip-set provides all four basic floating-point operations: add, subtract, multiply, as well as divide, conforming to the IEEE standard [1] with all exception handling including NaN, infinity, overflow and underflow.
Compare operation is also provided, producing comparison result required by the IEEE standard. Other functionality includes 32-bit two's complement numbers conversion to either format of the floating-point number and vice versa.

In order to achieve a superfast arithmetic processing, the chip-set uses dedicated circuit array to perform the required functions. The array flowthrough time for add, subtract, compare and conversions operations is 150ns for both single and double precisions, while multiplication takes 150ns or 250ns for single or double precision respectively. Division operation is implemented to fully conform to the IEEE standard, since it is important in a general purpose computer application to meet the accuracy and precision specified by the IEEE standard. Other division algorithm which uses a look-up table to obtain approximation to the reciprocal is often slower and in some cases will not meet the IEEE standard for accuracy at all.

The chip-set has several mode registers that select optional characteristics which are not often changed. Such mode registers include the IEEE rounding informations, a fast mode operation which substitutes zero for a denormalized number, and ability to control the array latency as a function of the number of clock cycles.

Figure 1 shows a simplified block diagram and the microphotograph of the floating-point multiplier core. The exponent block performs the addition of the exponents of the two operands, and subtract the appropriate bias from it. The exponent block also generates condition for overflow and underflow. The fraction block consists of an array of Carry Save Adders (CSA), and a 53 bit Carry Look-ahead Adder (CLA). To perform the array multiplication, a modified-Booth algorithm is employed to reduce the number of partial products required to be added. For a single precision multiplication, the array is used once and the CLA will provide the final fraction result. For double precision, the array is used twice, and again the CLA will provide the final result.

Figure 2 shows the block diagram and the microphotograph of the adder/divider chip. For add operation, the exponents of both operands are compared and the difference is used to right shift the fraction of the operand with the larger exponent. Now that the exponent is aligned, both fractions are added or subtracted depending on the signs of the operands. A priority encoder hardware is used to detect how many bit position a post-normalization is needed to be performed. The fraction is then left-shifted and the left shift amount is subtracted from the exponent. Detection of overflow and underflow is then performed on the exponent and the rounding is also performed on the fraction at the same time. The datapath is used once for single and double precision add, and conversion among three different data formats (single, double floating-point formats, and 32 bit 2's complement integer). A 8 by 8 'octal' barrel shifter technique is employed to perform up to 63-bit shift. This technique significantly improves the speed and area over a conventional binary shifter. An octal barrel shifter shifts 0 to 7 bit position per stage by the use of 8 to 1 multiplexer gates. Consequently, only 2 instead of 6 MUX stages are required to perform up to 63 bit shift. The gate delay can be roughly compared as log base 8 of 64 versus log base 2 of 64 for octal versus binary shifter. A fast carry look ahead technique (4 bit at a time) is also implemented on the incrementer and priority encoder hardware.

An iterative one-bit at a time restoring divide algorithm is employed to perform single and double precision divisions. On each cycle the divisor is subtracted from the partial remainder, and one quotient bit is generated. It is a '1' if the subtraction result is positive and a '0' otherwise. A 56-bit shift register chain is implemented above the priority encoder hardware. The shift register collects all quotient bits generated on every cycle. 25 and 54 cycles are needed to perform single and double precision division respectively. On the last cycle the priority encoder determines the left shift amount required to normalize a single precision quotient result, since the shift reqister is partially filled. The exponent block, meanwhile, is subtracting the exponent of the divisor from the exponent of the dividend, and an appropriate bias is added accordingly.

Every cycle, an operand may be loaded and unloaded through 32 bit I/O data pins. Internally, a precharged 32-bit bus serves to connect the input and output latches to the I/O pads. To save area, all data bits are stored in half registers (latches) instead of full master-slave registers. As a result of a shared internal I/O bus, a chain of multiplication or addition operations may be performed without having the result to go off-chip first, thereby increasing the scalar performance throughput. Since both chips implement 64 bit wide array datapath, operation involving operands with differing formats can also be handled directly without speed penalty to yield a double precision result (a mixed precision operation).

Figure 3 shows a design of an Extended Floating Point Unit (EFPU) [15] using WTL 1164/1165 as the central arithmetic elements. It is designed as a memory mapped peripheral to work in MC68020 based systems. A peak performance of 2 Megaflops can be expected.

IV. Conclusion

In this paper we have given an overview of the state-of-the-art VLSI implementations for handling floating-point arithmetics. Figure 4 shows where the general purpose building block FP processors stand among other VLSI processors. The IEEE binary floating point standard(P754), although more complex to implement than other standards, is being adopted almost universally because of it is rigorously defined and it offers greater precision and larger dynamic range. Because of the requirements for different applications, two main groups of floating-point processors are developed: general purpose and building blocks. As VLSI technology develops further, we will see that the general-purpose microprocessor will integrate the hardware to handle floating-point arithmetics on the same chip, resulting in the disappearance of the former group of processors. However, the building blocks will still be needed for the most speed-demanding applications. They will have further enhanced functionality and performance and greater I/O capability.

References:

[1]. A Proposed Standard For Binary Floating Point Arithmetic Draft 10.0 of IEEE Task P754; December 2, 1982

[2]. Nave, R. and Palmer, J., " A Numeric Data Processor", ISSCC Digest of Technical Papers, P.108-109; February 1980.

[3]. Huntsman, C. and Cawthron, D.," The MC68881 Floating-point Coprocessor", IEEE Micro, p.44-54, December 1983.

[4]. National Semiconductor, NS16081 Floating Point Unit Data Sheet; May 1983.

[5]. Advanced Micro Devices Am9511A/9512 Floating Processor Manual 1981.

[6]. Mor S., Hingarh H., Vora M., Wilnai D. and Longo T., "A 16b Microprocessor for Realtime Applications", ISSCC Digest of Technical Papers, p.28-29; February 1983.

[7]. Sandesara S., "Zilog's Z8070 Floating-point Processor"

[8]. Woo B., Lin L. and Owen R. E., "ALU, Multiplier Chips Zip Through IEEE Floating-point Operation", Electronics, p.121-126, May 19, 1983.

[9]. Woo B., Lin L. and Ware F., " A High-Speed 32 Bit IEEE Floating-point Chip Set for Digital Signal Processing" ICASSP paper 16.6, March 19-21, 1984.

[10]. Eldon J. and Winter G.E., "Floating-point Chips Carve Out FFT Systems", Electronic Design, p.133-138, August 4, 1983.

[11]. Ware F., Lin L., Wong R., Woo B. and Hansen C., "Fast 64-bit Chip Set Gangs Up for Double-Precision Floating-point Work", Electronics, July 12, 1984.

[12]. Quong D. and Periman R., "Single-chip Accelerators Speed Floating-point and Binary Computations" Electronic Design, p.246-262, November 15, 1984.

[13]. Analog Devices Word Slice Product Family Data Sheet, September, 1984.

[14]. Weitek WTL1164/1165 Preliminary Data Sheet.

[15]. White, G., "Solutions For a High Performance Floating-Point Coprocessor Design", Internal Memo, Feb 14,1985

TABLE 1

KEY APPLICATION FOR FLOATING POINT ARITHMETICS

Computer Aided Engineering
  Simulation
  Modelling

Scientific Computation
  Quantum Physics
  Quantum Chemistry
  EM Boundary Value Problems

Graphics

Seismic Energy Exploration

Medical Imaging

Telecommunications

Radar/Sonar Processing

Weather Analysis

Spectrum Analysis

Noise Cancellation

Digital Filtering

Non-destructive Testing

TABLE 2

Features of General-purpose Floating-Point Processors

1. Intel 8087/80287

Technology	3 micron HMOS		
Arithmetic functions	+,−,x,/,rem,sqrt,transcendental		
Data format/type	IEEE 32,64,80b FP, 16,32,64b int, 18 BCD		
Clock Rate	5 megahertz		
Input/Output	16b bidirectional bus		
Pin Count	40 pins		
Availability	now		
Performance	ADD	MPY	DIV
single prc	17us	19us	39us
double prc	17us	27us	39us

2. Motorola 68881

Technology	2.25 micron HCMOS		
Arithmetic functions	+,−,x,/,rem,mod,sqrt,transcendental		
Data format/type	IEEE 32,64,80b FP, 8,16,32b int, 96b BCD		
Clock Rate	16.67 megahertz		
Input/Output	configurable to 8,16,32b data bus		
Pin Count	64 pin DIP / 68 pin LCC		
Availability	sampling		
Performance	ADD	MPY	DIV
single prc	2.8us	2.8us	3.4us
double prc	2.8us	3.7us	5.6us

3. National 32081

Technology	3 micron XMOS		
Arithmetic functions	+,−,x,/,conversion to integer		
Data format/type	IEEE 32,64b FP, 8,16,32b integer		
Clock Rate	10 megahertz		
Input/Output	16 bit bidirectional bus		
Pin Count	24 pin DIP		
Availability	now		
Performance	ADD	MPY	DIV
single prc	7.4us	4.8us	8.9us
double prc	7.4us	6.2us	12us

4. Fairchild 9450

Technology	I3L−II, bipolar		
Arithmetic functions	+,−,x,/,compare, int convertion		
Data format/type	MIL−STD−1750A ISA 32,48b FP, 8,16,32b integer		
Clock Rate	20 megahertz		
Input/Output	16 bit bidirectional		
Pin Count	64 pin DIP		
Availability	sampling		
Performance	ADD	MPY	DIV
single prc	4.5us	5.6us	9.8us
(48b) double prc	5.75us	12.4us	21.2us

5. Zilog 8070

Technology	2 micron NMOS		
Arithmetic functions	+,−,x,/,remainder,sqrt,compare		
Data format/type	IEEE 32,64,80b FP, 16,32b int, BCD		
Clock Rate	10 megahertz		
Input/Output	16 bit bidirectional		
Pin Count	40 pin DIP		
Availability	no		
Performance	ADD	MPY	DIV
single prc	1.8us	2.8us	2.9us
double prc	1.8us	4.2us	4.3us

6. AMD 9512

Technology	NMOS		
Arithmetic functions	+,−,x,/		
Data format/type	IEEE 32,64b FP		
Clock Rate	3.125 megahertz		
Input/Output	8 bit bidirectional		
Pin Count	24 pin DIP		
Availability	now		
Performance	ADD	MPY	DIV
single prc	163.8us	81.3us	84.5us
double prc	992us	595.2us	1638.4us

TABLE 3

Features of Floating-Point Building-Block

1. Weitek 1032/1033

Technology	3 micron HMOS	
Data format/type	IEEE 32 bit FP, 24 bit integer	
Clock Rate	20 megahertz	
Input/Output	two 16bit input, one 16 bit output	
I/O bandwidth	120 megabytes/s	
Pin Count	64 pin DIP, 68 pin LCC	
Availability	now	
Performance	ADD	MPY
(32b) throughput	10 MFlops	10 MFlops
latency	500 ns	500 ns

2. TRW TDC1022/1042

Technology	1 micron bipolar triple diffusion	
Data format/type	22 bit FP	
Clock Rate	10 megahertz	
Input/Output	one 22b in(1022), two 22b in(1042), one 22b out	
I/O bandwidth	80 megabytes/s (1022/1042)	
Pin Count	64 pin DIP (1022), 84 pin LCC (1042)	
Availability	now	
Performance	ADD	MPY
(22b) throughput	10MFlops	10 MFlops
latency	200ns	200ns

Features of Building–block Floating–Point

(continued)

3. Weitek 1064/1065

Technology	2 micron NMOS
Data format/type	IEEE/DEC 32,64b FP, 32b integer
Clock Rate	16.67 megahertz
Input/Output	two 32b input, one 32b output
I/O bandwidth	200 megabytes/s
Pin Count	144 pin PGA
Availability	now

Performance		ADD	MPY
(32b)	throughput	8 MFlops	4 MFlops
	latency	720ns	600ns
(64b)	throughput	8 MFlops	2 MFlops
	latency	720ns	840ns

4. AMD 29325

Technology	2 micron IMOX bipolar
Data format/type	IEEE 32b FP, 32b integer, DEC 32b FP
Clock Rate	10 megahertz
Input/Output	two 32b input, one 32 bit output
I/O bandwidth	120 megabytes/s
Pin Count	144 pin PGA
Availability	sampling

Performance		ADD	MPY
(32b)	throughput	10MFlops	10 MFlops
	latency	100ns	100ns

5.Analog Devices ADSP3210/3220

Technology	2 micron CMOS
Data format/type	IEEE 32,64b FP, 32b integer
Clock Rate	10 megahertz
Input/Output	one 32 bit input, one 32 bit output
I/O bandwidth	160 megabytes/s
Pin Count	100 pin PGA
Availability	no

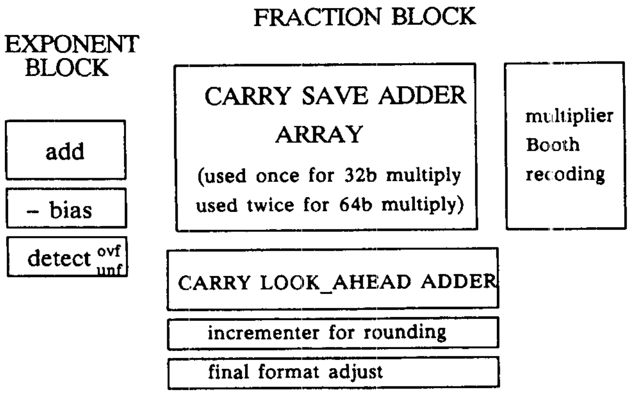
Performance		ADD	MPY
(32b)	throughput	10 MFlops	10 MFlops
	latency	350ns	350ns
(64b)	throughput	5 MFlops	2 MFlops
	latency	500ns	600ns

6. Weitek 1164/1165

Technology	2.5 micron NMOS
Data format/type	IEEE 32,64b FP, 32b integer
Clock Rate	20 megahertz
Input/Output	one 32 bit multiplexed input/output
I/O bandwidth	80 megabytes/s
Pin Count	64 pin DIP, 68 pin LCC
Availability	now

Performance		ADD	MPY	DIV
(32b)	throughput	4.4 MFlops	4.4 MFlops	–
	latency	150ns	150ns	1.25us
(64b)	throughput	2.5 MFlops	2.5 MFlops	–
	latency	150ns	250ns	2.7us

FIGURE 1



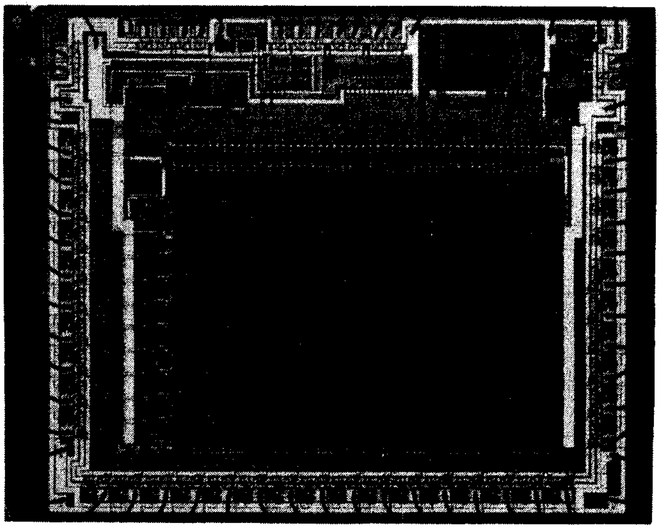
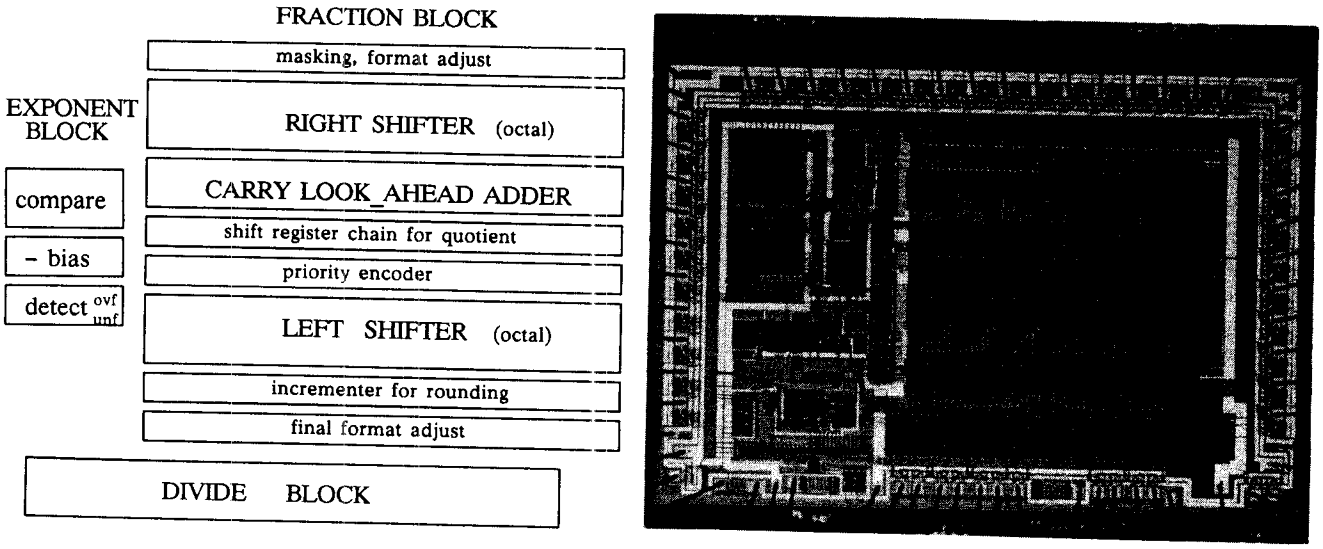


FIGURE 2



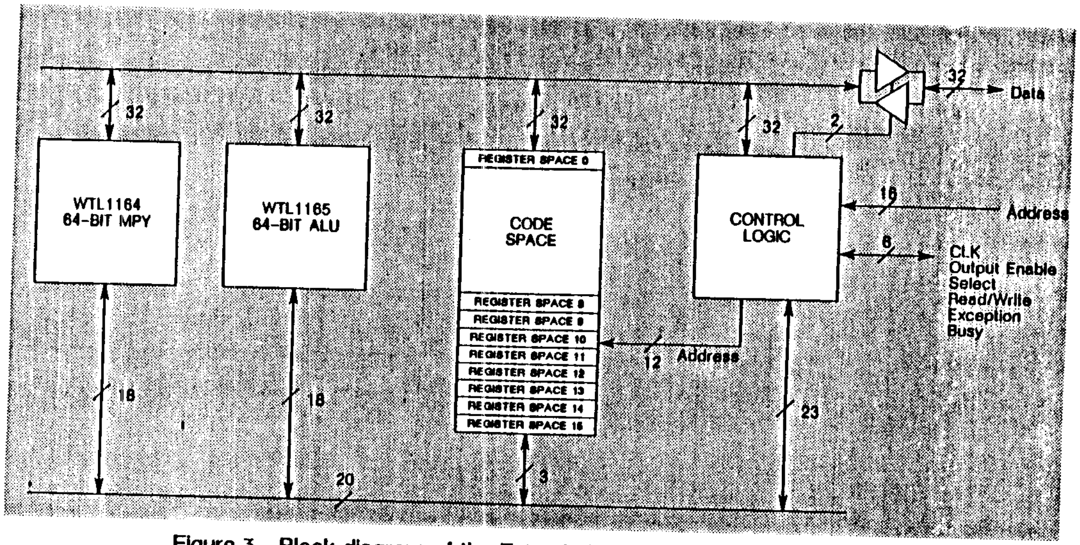


Figure 3 Block diagram of the Extended Floating-Point Unit (EFPU)

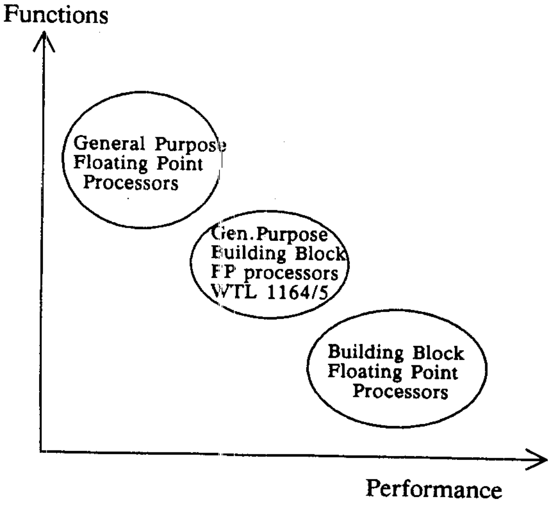

Functions
General Purpose Floating Point Processors
Gen.Purpose Building Block FP processors WTL 1164/5
Building Block Floating Point Processors
Performance

FIGURE 4