LSI LOGIC

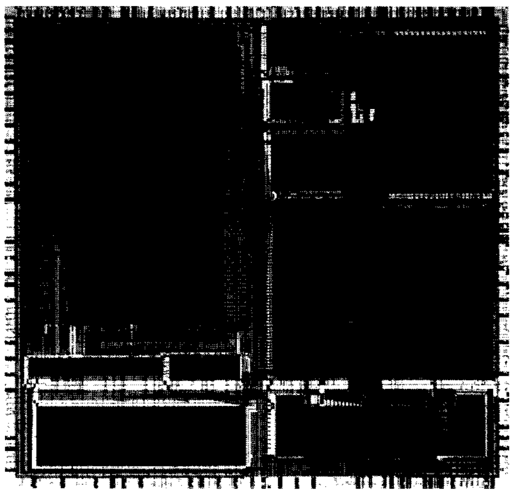
L64814 Floating-Point Unit (FPU)
Preliminary

Description

The L64814 Floating-Point Unit (FPU) is a high-performance, CMOS implementation of the SPARC (Scalable Processor ARChitecture) FPU. The FPU combines a floating-point controller with a high-throughput floating-point processor to provide a single-chip floating-point solution for SPARC-based systems.

The FPU implements the IEEE 754-1985 standard for floating-point arithmetic. It operates concurrently with the IU to execute single- and double-precision floating-point operations, as well as register-to-register move instructions, floating-point loads and stores, and floating-point queue and state register instructions. Supported floating-point operations are: add, subtract, multiply, divide, square root, compare, and convert. Each instruction not implemented in the L64814 hardware generates an instruction trap, so that the instruction can be emulated in software. Note that the FPU handles all IEEE exceptions in hardware, except for denormals in the floating-point multiplier unit.

The L64814 FPU is part of the LSI Logic L64811 Chip Family which implements and supports SPARC-based system development.


L64814 Die

Features

- High-performance operation – Provides double-precision Linpack floating-point operation at:

Device	Performance
L64814-25 MHz	3.8 MFlops
L64814-33 MHz	5.0 MFlops
L64814-40 MHz	6.0 MFlops

- Low-cost solution – Integrates a floating-point controller and floating-point processor on a single chip for cost-efficient system implementation
- Wide range of operating frequencies – 25, 33, and 40 MHz versions
- Implements IEEE exception handling directly in hardware
- 64-bit wide internal datapath for all floating-point operations results in highly efficient double-precision performance
- Connects directly to the L64811 Integer Unit (IU).
- Pin-compatible with the Weitek Abacus 3171 and Texas Instruments TMS390C602 floating-point units
- Advanced, 143-pin cavity-up plastic or ceramic pin grid array package

July 1990 Order Number L64814

\

Interconnect Diagram

Figure 1 shows the FPU, the L64811 Integer Unit (IU), and the L64815 Memory Management, Cache Control, and Cache Tags Unit (MCT) in a system configuration.

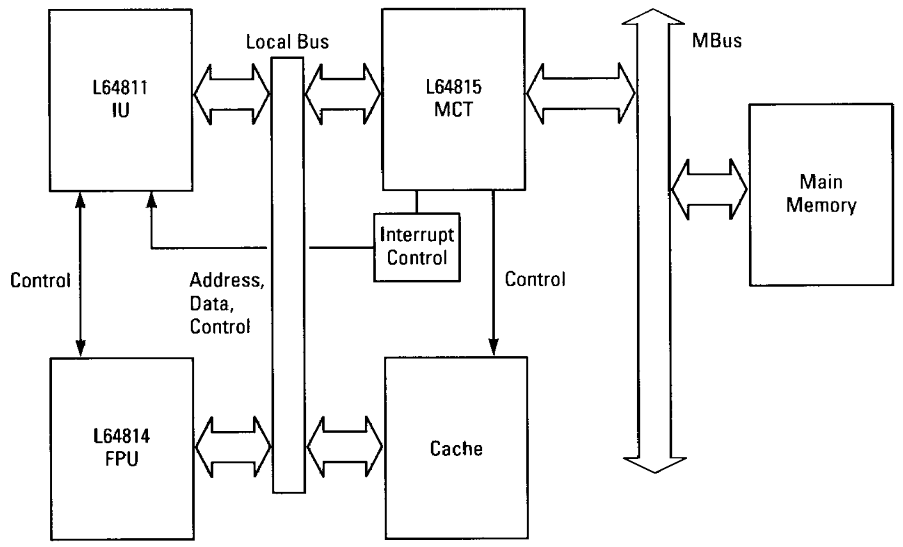


Figure 1. FPU-IU-MCT System Interconnect Diagram

Instruction Set

The SPARC architecture specifies a complete set of instructions for a SPARC-compatible FPU. An FPU which executes the floating-point instruction set may actually implement a subset of the instructions in hardware, and trap any instructions which are not implemented in hardware; the system software then performs the trapped instructions. The L64814 implements in hardware all SPARC floating-point instructions which operate on integer, single-precision, and double-precision data. It traps all extended-precision floating-point instructions for execution in software.

Table 1 lists the FPU instruction set, which includes the floating-point load, store, and floating-point operate (FPop) instructions. Table 2 shows the cycle count for the FPops. Note that the cycle count for floating-point load and store instructions varies depending on several factors, including the precision of the data and the availability of the instruction and data.

Instruction Set
(Continued)

Table 1. FPU Instruction Set

Mnemonic	Instruction
FiTO(s,d,x[1])	Convert integer to (single, double, extended[1])-precision floating-point
F(s,d,x[1])TOi	Convert (single, double, extended[1])-precision floating-point to integer
FsTO(d,x[1])	Convert single-precision floating-point to (double, extended[1])-precision floating-point
FdTO(s,x[1])	Convert double-precision floating-point to (single, extended[1])-precision floating-point
FxTO(s,d)[1]	Convert extended-precision floating-point to (single, double)-precision floating-point[1]
FMOVs	Move a word from one f-register to another
FNEGs	Negate the operand (invert the sign bit)
FABSs	Take the absolute value (clear the sign bit)
FSQRT(s,d,x[1])	Calculate the (single, double, extended[1])-precision square root
FADD(s,d,x[1])	Add the (single, double, extended[1])-precision operands
FSUB(s,d,x[1])	Subtract the (single, double, extended[1])-precision operands
FMUL(s,d,x[1])	Multiply the (single, double, extended[1])-precision operands
FDIV(s,d,x[1])	Divide the (single, double, extended[1])-precision operands
FCMP(s,d,x[1])	Compare the (single, double, extended[1])-precision operands
FCMPE(s,d,x[1])	Compare the (single, double, extended[1])-precision operands and cause an exception if unordered (if at least one is a NaN, i.e. Not a Number)
LDF	Load floating-point
LDDF	Load double-precision floating-point
LDFSR	Load floating-point status register
STF	Store floating-point
STDF	Store double-precision floating-point
STFSR	Store floating-point status register
STDFQ	Store double-precision floating-point queue

Note:
1. Trapped instruction

Table 2. FPop Instruction Cycle Count

Instruction Type	Instruction	Latency
Instructions which use the FPU Multiplier	FMULs,d	4
	FDIVs,d	33
	FSQRTs,d	45
Instructions which use the FPU Adder	FADDs,d	4
	FSUBs,d	4
	FiTOs,d	4
	FsTOi,d	4
	FdTOi,s	4
	FMOVs	2
	FNEGs	2
	FABSs	2
	FCMPs,d	3
	FCMPEs,d	3

Block Diagram

Figure 2 shows a block diagram of the FPU.

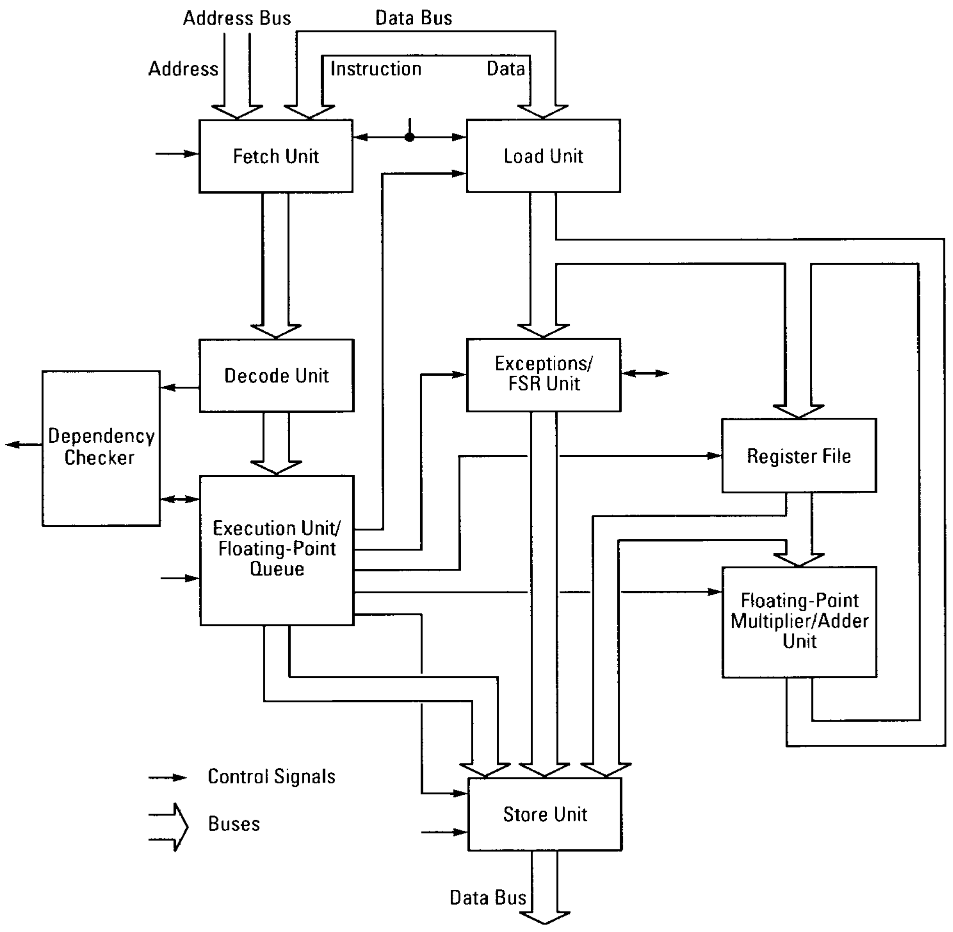


Figure 2. L64814 FPU Functional Block Diagram

In the diagram, the **Fetch Unit** captures each instruction and its address from the Data and Address buses, respectively. The **Decode Unit** decodes the instruction opcodes and makes them available to the Execution Unit.

The **Execution Unit and Floating-Point Queue** handles floating-point instruction execution. When the L64811 Integer Unit (IU) decodes a valid floating-point operate (FPop) or floating-point load/store instruction, it signals the FPU. The FPU latches the instruction and address from the Decode Unit and starts execution. The two-deep floating-point queue (FQ) holds the instructions and addresses for the currently executing instructions.

The **Dependency Checker** determines whether the instruction depends on the results or the resources required by other floating-point instructions ahead of it in the queue. If a dependency exists, then the Dependency Checker freezes the instruction pipeline until the dependency is cleared.

The **Load Unit** holds data fetched from memory until the FPU writes it to the **Register File**. The Register File, also referred to as the f-registers, consists of 32, 32-bit registers. These registers store data (operands) for FPops and for floating-point load/store instructions.

The **Floating-Point Multiplier/Adder Unit** contains the 64-bit adder and 64-bit multiplier which FPops use to operate on data in the Register File. Because the FPU includes a separate multiplier and adder, it supports parallel execution of FPops.

Block Diagram (Continued)

The **Exceptions/FSR Unit** maintains the status of FPops completing execution, as well as that of the operating mode of the FPU. The Floating-Point Status Register (FSR) is a 32-bit register whose fields store this information.

The **Store Unit** holds data which the FPU drives onto the Data Bus during execution of a floating-point store instruction.

Pin Descriptions

This section lists and describes the FPU signals.

A[31:2]
Address Bus[31:2] – A[31:0] comprise the instruction address bus common to the FPU, IU, and memory subsystem. From this bus, the FPU latches the instruction address for each instruction fetched. Because instructions are stored on 32-bit boundaries, the FPU does not need the two lowest-order bits of the address, A[1:0].

CCCV
Coprocessor Condition Codes Valid – The coprocessor uses this signal to notify the IU and FPU when the coprocessor condition codes are valid. When CCCV is deasserted, the IU instruction pipeline freezes until the instruction that generates the coprocessor condition codes completes execution and the codes become valid.

$\overline{\text{CHOLD}}$
Coprocessor Hold – When the coprocessor detects a condition where it cannot accept any more instructions, it asserts this signal to freeze the IU instruction pipeline. This signal is similar to the $\overline{\text{FHOLD}}$ signal generated by the FPU.

CLK
Clock – This signal provides the system clock to the FPU.

$\overline{\text{DOE}}$
Data Bus Driver Output Enable – Asserting this signal enables the FPU data bus drivers. The system deasserts this signal when another bus master needs to use the data bus.

D[31:0]
Instruction/Data Bus [31:0] – D[31:0] comprise the instruction/data bus common to the FPU, IU, and memory subsystem.The FPU fetches all instructions from this bus. In addition, on floating-point load/store instructions, the FPU receives/sends data from/to memory on this bus.

FCCV
Floating-Point Condition Codes Valid – The FPU asserts this signal when the floating-point condition codes, FCC[1:0], become valid. When the FPU deasserts this signal, the IU instruction pipeline freezes.

FCC[1:0]
Floating-Point Condition Codes [1:0] – These signals are the FPU condition code; they are valid only when FCCV is asserted. During the execution of a Branch on floating-point condition code (Bfcc) instruction, the IU uses these bits to make branching decisions. These signals have the same state as the FCC field of the FSR.

$\overline{\text{FEXC}}$
Floating-Point Exception – The FPU asserts this signal to notify the IU that a floating-point exception has occurred and that the IU should take the trap. The signal remains asserted until the IU acknowledges that it has taken the trap by asserting FXACK.

$\overline{\text{FHOLD}}$
Floating-Point Hold – The FPU asserts this signal to freeze the instruction pipeline when it cannot accept any more floating-point instructions due to resource or data dependencies. The FPU deasserts the signal when the dependency is resolved.

FINS1
Floating-Point Instruction Select – The IU asserts this signal during the decode stage of a floating-point instruction to notify the FPU that it should start executing the last instruction fetched.

FINS2
Floating-Point Instruction Select – The IU asserts this signal during the decode stage of a floating-point instruction to notify the FPU that it should start executing the second-to-last instruction fetched.

FLUSH
FPU Instruction Pipeline Flush – The IU asserts this signal to notify the FPU to abort the floating-point instructions that are still in the pipeline and have not yet entered the queue. The IU typically asserts this signal when it takes a trap, and it restarts the aborted instructions after the trap handler completes execution. Instructions

Pin Descriptions (Continued)

that are already in the queue complete execution.

FNULL
Floating-Point Null – The FPU asserts this signal to notify the memory subsystem that the FPU is freezing the instruction pipeline. It asserts FNULL whenever it asserts $\overline{\text{FHOLD}}$ or deasserts FCCV. The memory system uses FNULL in the same fashion as the IU's INULL signal; it needs the additional signal because INULL does not take into account FPU holds.

$\overline{\text{FP}}$
Floating-Point Unit – This signal tells the IU that a floating-point unit is present in the system. The signal typically has a pullup resistor holding it HIGH at the IU input; when the FPU is plugged into the board, the FPU pulls the signal LOW.

FXACK
Floating-Point Exception Acknowledge – The IU asserts FXACK to signal the FPU that it has taken the requested floating-point exception trap. In response, the FPU deasserts $\overline{\text{FEXC}}$.

INST
Instruction – The IU asserts this signal when it is fetching a new instruction; it signals the FPU to copy the instruction being fetched.

$\overline{\text{MHOLDA}}$, $\overline{\text{MHOLDB}}$, $\overline{\text{BHOLD}}$
Memory Hold – The memory subsystem asserts these signals to freeze the IU instruction pipeline.

$\overline{\text{MDS}}$
Memory Data Strobe – The memory subsystem asserts this signal to strobe an instruction or data into the FPU during a cache miss situation. One or more memory hold signals are active at the same time.

$\overline{\text{RESET}}$
System Reset – The system asserts this pin to reset the FPU.

$\overline{\text{TOE}}$
Test Output Enable – For chip and board test, tying this pin HIGH floats all of the FPU output drivers, including the D bus.

Specifications

This section provides the electrical specifications for the L64814.

Table 3. Absolute Maximum Ratings

Symbol	Parameter	Limits[1]	Unit
VDD	DC Supply	-0.3 to +7	V
VIN	Input Voltage	-0.3 to VDD +0.3	V
IIN	DC Input Current	±10	mA
TSTG	Storage Temperature Range (plastic)	-40 to +125	°C
TSTG	Storage Temperature Range (ceramic)	-65 to +150	°C

Note:
1. Referenced to VSS

Table 4. Recommended Operating Conditions

Symbol	Parameter	Limits	Unit
VDD	DC Supply	+4.75 to +5.25	V
TA	Ambient Temperature	-0 to +70	°C

Specifications (Continued)

Table 5. DC Characteristics

Symbol	Parameter	Condition[1]	Min	Typ	Max	Units
VIL	Voltage Input LOW				0.8	V
VIH	Voltage Input HIGH		2.0			V
VOH	Voltage Output HIGH	IOH = -8.0 mA	2.4	4.5		V
VOL	Voltage Output LOW	IOL = 8.0 mA		0.2	0.4	V
IIH	Current Input HIGH	VIN = VDD			10	µA
IIL	Current Input LOW	VIN = VSS			-10	µA
IOH	Current Output HIGH	VOH = 2.4 V	-4.0			mA
IOL	Current Output LOW	VOL = 0.4 V	4.0			mA
IOZ	Current 3-state Output Leakage	VOH = VDD or VSS	-10	±1	10	µA
IOS	Current Output Short Circuit	VDD = Max, output shorted to VDD	15	50	130	mA
		VDD = Max, output shorted to VSS	-5	-25	-150	mA
IDD	Quiescent Supply Current	VIN = VDD or VSS			10	mA
ICC	Dynamic Supply Current	f = 10 MHz at 5.25 V f = 33 MHz at 5.25 V f = 40 MHz at 5.25 V			90 300 380	mA mA mA

Note:
1. Specified at VDD equals 5V ± 5%; ambient temperature over the specified range

Table 6. Capacitance

Symbol	Parameter	Condition	Min	Typ	Max	Units
CIN	Input Capacitance	VIN = 5.0 V, TA = 25°C, f = 1 MHz			10	pF
COUT	Output Capacitance	VIN = 5.0 V, TA = 25°C, f = 1 MHz			12	pF

LSI LOGIC

L64814 Floating-Point Unit (FPU)

Preliminary

Pinout, Package and Ordering Information

The L64814 is available in a 143-pin, cavity-up ceramic (CPGA) or plastic (PPGA) pin grid array package. Table 7 provides ordering information for the various package and speed options.

Figure 3 shows the pinout for the L64814 in either package. Figure 4 is the mechanical drawing for the packages.

Table 7. L64814 Ordering Information

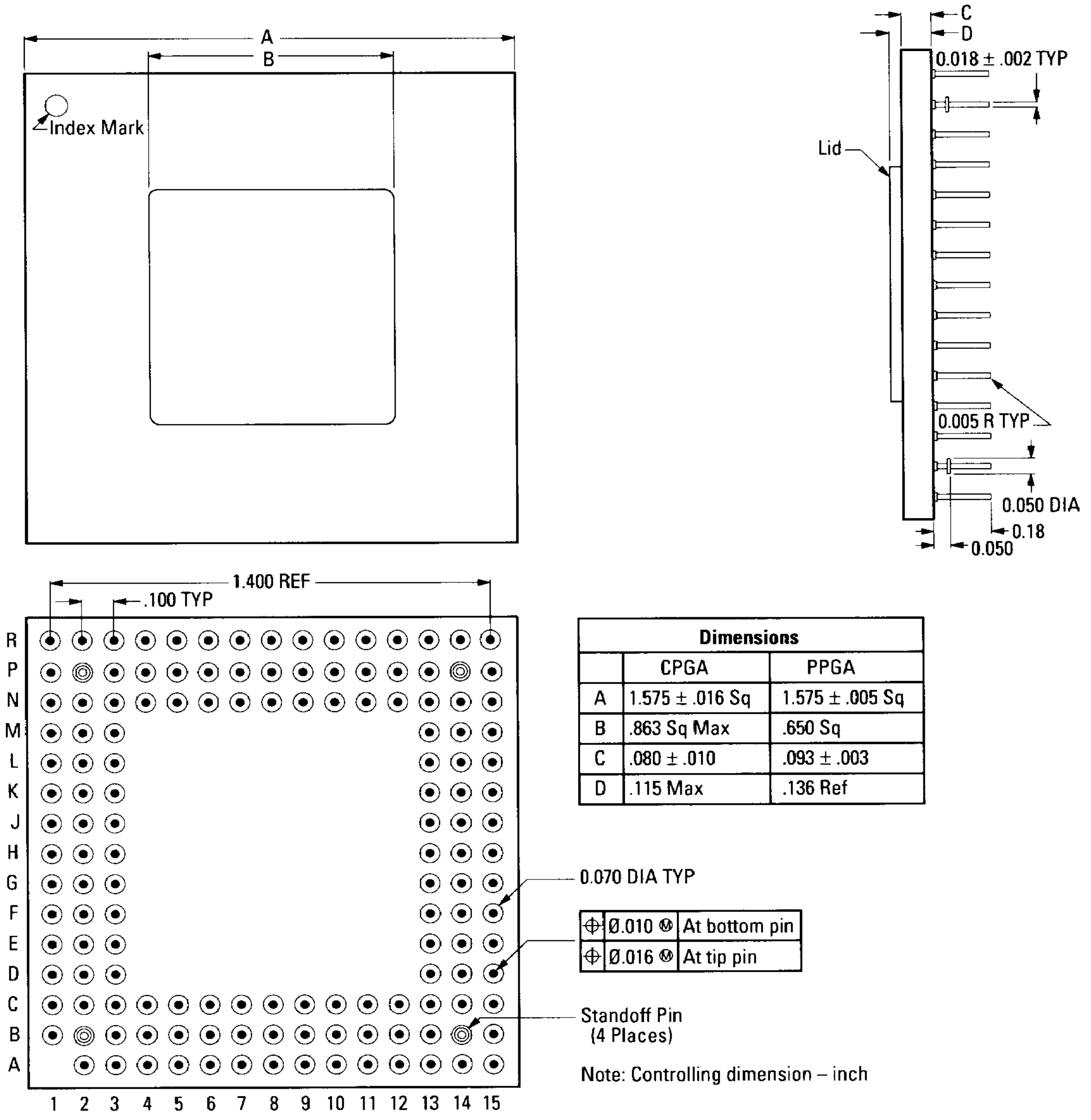
Order Number	Clock Frequency	Package Type	Operating Range
L64814CG-25	25 MHz	143 CPGA	Commercial
L64814NC-25	25 MHz	143 PPGA	Commercial
L64814CG-33	33 MHz	143 CPGA	Commercial
L64814NC-33	33 MHz	143 PPGA	Commercial
L64814CG-40	40 MHz	143 CPGA	Commercial
L64814NC-40	40 MHz	143 PPGA	Commercial

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	*missing pin*	D22	A22	D24	A24	A25	D26	A26	A27	A28	A29	A30	A31	D31	GND
B	D21	VCC	VCC	A23	D23	VCC	D25	VCC	D27	D28	D29	D30	VCC	VCC	VCC
C	D20	A21	GND	GND	VCC	GND	*	VCC	GND	GND	GND	GND	GND	VCC	FCCV
D	D19	VCC	GND										GND	GND	FCC1
E	A18	A19	A20										CCCV	FCC0	FXACK
F	A16	D17	D18										$\overline{\text{RESET}}$	GND	$\overline{\text{FEXC}}$
G	D16	A17	GND										CLK	GND	FNULL
H	A00	A01	D00										GND	$\overline{\text{CHOLD}}$	$\overline{\text{FHOLD}}$
J	D01	$\overline{\text{DOE}}$	*										VCC	$\overline{\text{MHOLDA}}$	$\overline{\text{BHOLD}}$
K	D02	VCC	GND										VCC	$\overline{\text{MDS}}$	$\overline{\text{MHOLDB}}$
L	A02	D03	GND										FLUSH	VCC	VCC
M	A03	VCC	D05										GND	FINS1	INST
N	D04	VCC	GND	GND	GND	D08	GND	D10	$\overline{\text{TOE}}$	GND	D14	GND	GND	VCC	FINS2
P	A04	VCC	GND	A06	VCC	A08	VCC	A11	D12	VCC	VCC	VCC	D15	VCC	VCC
R	A05	VCC	D06	A07	D07	A09	D09	A10	D11	A12	A13	D13	A14	A15	$\overline{\text{FP}}$

* *Reserved Pins*

Figure 3. 143-Pin CPGA and PPGA Pinout – Top View

Pinout, Package and Ordering Information (Continued)



Dimensions		
	CPGA	PPGA
A	1.575 ± .016 Sq	1.575 ± .005 Sq
B	.863 Sq Max	.650 Sq
C	.080 ± .010	.093 ± .003
D	.115 Max	.136 Ref

⌖	Ø.010 Ⓜ	At bottom pin
⌖	Ø.016 Ⓜ	At tip pin

Note: Controlling dimension – inch

Figure 4. 143-Pin CPGA and PPGA Mechanical Drawing

LSI LOGIC

L64814 Floating-Point Unit (FPU)
Preliminary

Sales Offices and Design Resource Centers

LSI Logic Corporation Headquarters Milpitas, CA
■ **408.433.8000**

Alabama
■ 205.883.3527

Arizona
602.951.4560

California
San Jose
■ 408.954.1561

Irvine
■ 714.553.5600

San Diego
619.541.7092

Encino
■ 818.379.2400

Colorado
303.756.8800

Florida
Altamonte Springs
407.339.2242

Boca Raton
■ 407.395.6200

Georgia
404.448.4898

Illinois
■ 708.773.0111

Maryland
Bethesda
■ 301.897.5800

Columbia
301.740.5664

Massachusetts
■ 617.890.0180 (Design Ctr)
617.890.0161 (Sales Ofc)

Michigan
313.930.6975

Minnesota
■ 612.921.8300

New Jersey
■ 201.549.4500

New York
914.226.1620

North Carolina
■ 919.872.8400

Oregon
503.645.9882

Pennsylvania
215.638.3010

Texas
Austin
512.338.2140

Dallas
■ 214.788.2966

Washington
■ 206.822.4384

LSI Logic Corporation of Canada, Inc.
Headquarters
Calgary
■ 403.262.9292

Edmonton
■ 403.450.4400

Ottawa
■ 613.592.1263

Montreal
■ 514.694.2417

Toronto
■ 416.620.7400

Vancouver
■ 604.433.5705

France
LSI Logic S.A.
■ 33.1.46212525

Israel
LSI Logic Limited
■ 972.3.5403741/4

Italy
LSI Logic SPA
■ 39.39.6056881

Japan
LSI Logic K.K.
Tokyo
■ 81.3.589.2711

Tsukuba-Shi
■ 81.298.52.8371

Osaka
■ 81.6.947.5281

Yokohama
81.45.902.4111

LSI Logic Corporation of Korea Limited
■ 82.2.561.2921

Netherlands
LSI Logic/Arcobel
■ 31.4120.30335

Scotland
LSI Logic Limited
■ 44.506.416767

Sweden
LSI Logic Export AB
46.8.703.4680

Switzerland
LSI Logic/Sulzer
■ 41.32.515441

Taiwan
LSI Logic Corporation
■ 02.755.3433

United Kingdom
LSI Logic Limited
Bracknell
■ 44.344.426544

West Germany
LSI Logic GmbH
Headquarters
Munich
■ 49.89.926903.0

Dusseldorf
■ 49.211.5961066

Stuttgart
■ 49.711.2262151

AE Advanced Electronics
Hannover
■ 49.511.3681756

AE Advanced Electronics
Munich
■ 49.89.93009855

■ Sales Offices with Design Resource Centers

018670 X — X

This document is preliminary. As such, it contains data derived from functional simulations and performance estimates. LSI Logic has not verified the functional descriptions or electrical and mechanical specifications using production parts.

Printed in USA
030.7111.6538.004.15K.TP.J