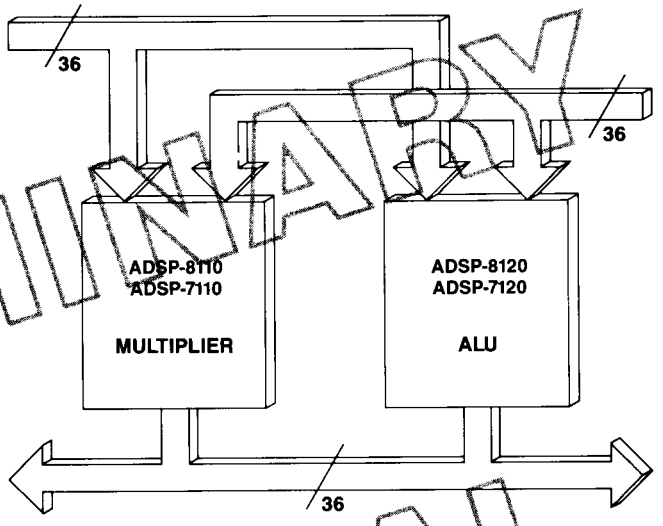
ANALOG DEVICES

November, 1987

ADSP-8110/ADSP-8120
ADSP-7110/ADSP-7120
Floating Point Chip Set

FEATURES

- Complete floating point and integer processor chip set supports the ANSI/IEEE STD.754 and DEC (F&G) formats.
- Four data formats
  64-bit floating point
  32-bit floating point
  64-bit integer (fixed point)
  32-bit integer (fixed point)
- Flowthrough architecture
  20 MFLOPS double precision multiply data rate
  40 MFLOPS double precision ALU data rate
  100 MIPS integer data rate
- Complete instruction set
  Floating point instructions include:
  Multiply, divide, square root, add, subtract, absolute value, negate, min/max, compare
  Integer instructions include:
  Multiply, add, subtract, boolean functions, shift and rotate
  Conversion operations to/from all supported formats
- Three port architecture
  Parity generation and checking
  Scan paths through all registers
- Fast or wrapped underflow and overflow in IEEE mode
- Input and output latches independently configurable as edge-triggered registers (FALU)
- Output latches configurable as edge-triggered registers (FMPY)
- Synchronous or asynchronous output enable option for status flags and output ports
- ECL 10KH compatible interface (ADSP-8110/ADSP-8120)
- TTL compatible interface (ADSP-7110/ADSP-7120)
- 169-pin pin-grid-array package



DESCRIPTION

The ADSP-8110/ADSP-7110 (FMPY) floating point multiplier and ADSP-8120/ADSP-7120 (FALU) floating point ALU provide very high performance floating point and integer operations. Because they are fabricated with a high performance VLSI process, the need for multiple pipeline stages normally associated with floating point processors can be eliminated. This architecture allows higher performance than can be attained with heavily pipelined floating point units, simplifying microcode and compiler code generation, system timing and control hardware. All floating point operations can be either single or double precision and are fully compatible with the IEEE standard 754 or DEC F and G formats. The floating point instruction set includes add, subtract, multiply and conversion operations. In addition, floating point divide, square-root, minimum, maximum and compare instructions are provided. All four IEEE rounding modes are supported.

The floating point chip set also has a large repertoire of 32- and 64-bit integer functions. A 64-bit integer ALU and barrel shifter provide very high performance for both 32- and 64-bit integer operations. Functions include add and subtract (with and without carry/borrow), negate, absolute value, all 16 boolean functions, rotate, logical shift, arithmetic shift, (32-bit) bit reverse, and rotate two concatenated 32-bit operands. Shifts and rotates use an internal register to define shift distance.

The three port architecture of the FMPY and FALU provides maximum performance. The 72-bit transparent latches at the input ports and the 64-bit latch at the output port of the FALU can be configured as edge triggered registers. The FMPY provides 72-bit edge-triggered registers at the input ports and 64-bit transparent latches or edge-triggered registers on the output port. Individual clock enables are provided for both input registers/latches and the output register/latch. The output port can also be used as an input port to allow intermediate results to be passed between the FMPY and FALU. Thirteen status flags are provided (interrupt, negative, zero, overflow, underflow, invalid operation, inexact result, rounded up, not-a-number, denormalized input, divide by 0, carry and parity error).

Output enables can be either synchronous, asynchronous, or both. The synchronous output enable option helps reduce bus conflicts in TTL based design.

Byte parity on each port is provided for increased system reliability. Built in test features include scan paths through all registers.

The functionality of the ADSP-8110 and ADSP-8120 is identical to the functionality of the ADSP-7110 and ADSP-7120, respectively, except the ADSP-7110 and ADSP-7120 have TTL I/O.

Floating Point Chip Set

■ SIGNAL SUMMARY

DATA		
INPUTS	X0-31, XP0-3, Y0-31, YP0-3	72
BIDIRECTIONAL	T0-31, TP0-3	36
CONTROL		
INSTRUCTIONS	I0-7	8
FLAGS	INT, PE, N, ZR, OV, UF, INX, INV, NaN, DEN, DIVZ (FMPY only), CRY (FALU only)	11
MODE SELECT	R/L1, (FALU only), R/L2, RESET	3
MULTIPLEXER SELECT	XSEL, YSEL (FALU only)	2
CLOCK ENABLES	$\overline{ZEN}$, $\overline{XEN}$, $\overline{YEN}$	3
OUTPUT ENABLES	$\overline{TOEN}$, $\overline{FOEN}$	2
	$\overline{TSOEN}$, $\overline{FSOEN}$	2
CLOCKS	CK1, CK2, MSWSEL, MSWEN	4
SCAN PATH		
INPUT	SMODE, SCK, SIN	3
OUTPUT	SOUT/RND	1
POWER (ADSP-8110/ADSP-8120)		
LOGIC GROUND	VCC1	10
OUTPUT GROUND	VCC2	5
– 5.2V	VEE	7
POWER (ADSP-7110/ADSP-7120)		
LOGIC + 5.0V	VCC1	8
OUTPUT + 5.0V	VCC2	5
LOGIC GND	GND1	4
OUTPUT GND	GND2	5
TOTAL		**169**

■ SIGNAL DESCRIPTION

DATA

X0-X31	32-bit X input port
XP0-3	Byte parity bits corresponding to the X input port
Y0-Y31	32-bit Y input port
YP0-3	Byte parity bits corresponding to the Y input port
T0-31	32-bit T bidirectional port
TP0-3	Byte parity bits corresponding to the T bidirectional port

CONTROL

I0-7	8-bit instruction port. Determines the instruction executed by the floating point chip set.
INT	Interrupt flag; asserted if the appropriate bits are enabled in the interrupt enable register and the corresponding condition is true.
PE	Parity flag; asserted whenever a byte parity error is detected by on-chip circuitry at ports X, Y, or T.
N	Negative flag; asserted whenever a computation produces a result which has its most significant bit set.
ZR	Zero flag; asserted whenever a computation produces a result equal to zero.
OV	Overflow flag; this bit is set if the result of an operation is larger than the maximum representable normalized number in the chosen format.
UF	Underflow flag; this bit is set if the result of an operation is less than the minimum representable normalized number in the chosen format.
INX	Inexact flag; asserted whenever the result of a computation is not infinitely precise.
INV	Invalid operation flag; asserted whenever an operand is invalid for the operation to be performed.
NaN	Not a number flag; asserted whenever operands or the result of a computation has no numerical significance (IEEE), or a reserved operand (DEC).
SOUT/RND	The SOUT/RND pin normally outputs the rounded up flag (SMODE deasserted). Asserted when the magnitude of the infinitely precise result is less than the magnitude of the returned result. In scan I/O mode, scan data is output. The SOUT/RND pin is not affected by output enables.
DEN	Denormalized flag; asserted whenever one of the input operands is a denormalized number.
DIVZ (FMPY)	Divide by zero flag; asserted when a finite non-zero number is divided by zero.
CRY (FALU)	Carry flag; asserted during integer arithmetic operations whenever there is a carry out of the most significant result bit. For shift and rotate operations, set equal to the last bit shifted out.
R/L1	Register/Latch mode select for input. Configures input registers XA, YA and INSTR as transparent latches when low, positive edge triggered registers when high.
R/L2	Register/Latch mode select for output. Configures the Z register as a transparent latch when low and configures the control and status registers to load new data on the falling edge of CK2. Configures the Z, control and status registers as positive edge triggered registers when high.
RESET	Hardware reset. Asynchronously resets the mode, interrupt mask, flag and SC registers when asserted high.
XSEL	Input multiplexer select for register XA. Selects the X port when low. Selects the T port when high.
YSEL	Input multiplexer for operand port Y. Selects the Y port when low. Selects the Z result when high. (FALU only)
$\overline{XEN}$	Active low enable for CK1 at XA. Opcode register/latch is enabled if either XEN or YEN is true.
$\overline{YEN}$	Active low enable for CK1 at YA. Opcode register/latch is enabled if either XEN or YEN is true.
$\overline{ZEN}$	Active low enable for CK2 (The clock is always enabled for $\overline{TSOEN}$ and $\overline{FSOEN}$).

■ SIGNAL DESCRIPTION (cont'd)

OUTPUT ENABLES
(ADSP-8110/ADSP-8120)

$\overline{TOEN}$	Active low output enable for the T output port.
$\overline{FOEN}$	Active low output enable for the flag port.
$\overline{TSOEN}$	Synchronous active low output enable for the T port.
$\overline{FSOEN}$	Synchronous active low output enable for the flag port.

OUTPUT ENABLES
(ADSP-7110/ADSP-7120)

$\overline{TOEN}$	Active low output enable for the three-state T output port.
$\overline{FOEN}$	Active low output enable for the three-state flag port.
$\overline{TSOEN}$	Synchronous active low output enable for the three-state T output port.
$\overline{FSOEN}$	Synchronous active low output enable for the three-state flag port.

CLOCKS

CK1	Input clock for the X and Y ports and the instruction port. In the register mode, data is clocked on the rising edge of the clock. In the latch mode, the latches are transparent when the clock is low.
CK2	Output clock for computation results, flags, and internal register writes. In the register mode, data is clocked on the rising edge of the clock. In the latch mode, the latches are transparent when the clock is high.
MSWSEL	Multiplexer clock for output port T and clock for latch XC. Selects the most significant word of the T port and opens latch XC when high. For single precision operations, the 32-bit result in register Z will be selected, regardless of the state of MSWSEL.
MSWEN	Clock which opens the input demultiplexing latch on X and Y ports. Latches are transparent when high.

SCAN PATH

SMODE	Configures the on-chip registers in the scan path into a serial shift register when high. The RND flag is output on SOUT when the SMODE pin is low.
SCK	Clock which shifts data in the scan path. Ignored when SMODE is low, overrides CK1 and CK2 when SMODE is high. Rising edge triggered.
SIN	Input port to the registers configured in the scan path.
SOUT/RND	Scan path output when SMODE is asserted otherwise rounded up flag (see flag description).

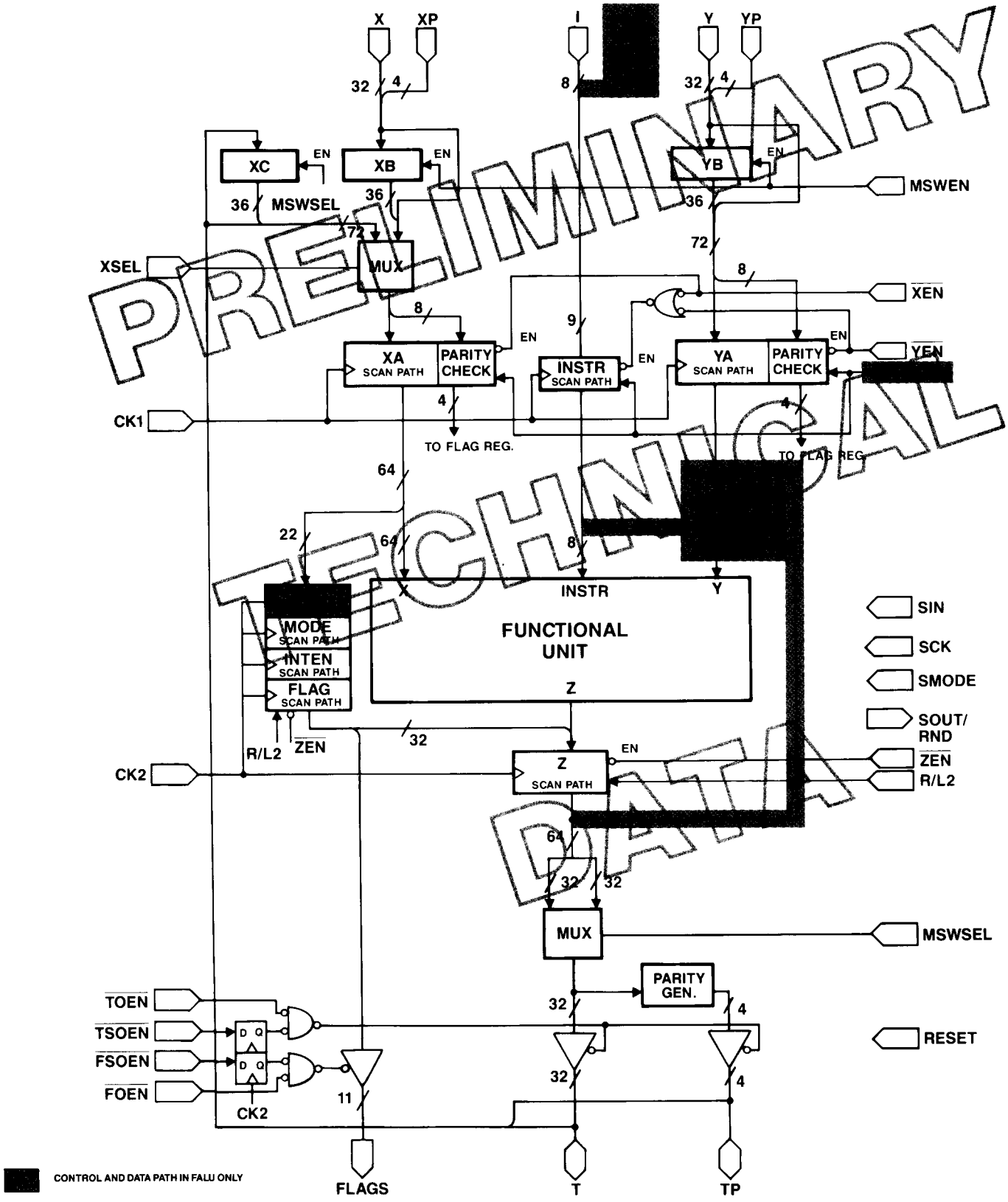
POWER
(ADSP-8110/ADSP-8120)

VCC1	Most positive supply voltage to internal logic circuitry. Usually ground.
VCC2	Most positive supply voltage to output circuitry. Usually ground.
VEE	Most negative supply.

POWER
(ADSP-7110/ADSP-7120)

VCC1	Positive supply voltage to internal logic circuitry.
VCC2	Positive supply voltage to output circuitry.
GND1	Negative supply voltage to internal logic circuitry.
GND2	Negative supply voltage to output circuitry.

BLOCK DIAGRAM (ADSP-8110/ADSP-8120, ADSP-7110/ADSP-7120)



INSTRUCTION SET

The ADSP-8110/ADSP-8120 and ADSP-7110/ADSP-7120 instruction set supports numeric intensive, bit manipulation and general purpose computing applications.

The instruction operation codes of the FMPY and FALU were encoded to allow the same 8-bit instruction stream (10–7) to be sent to each device.

The FMPY performs four types of operations: multiply, divide, square root, and pass operand. The FALU performs all other operations.

Three instructions are provided to perform sum of product operations (MAC/DMAC, MACS/DMACS, SMAC/DSMAC) where the same instruction opcode is interpreted by the FMPY as a multiply and the FALU as an add or subtract.

FLAGS

The following rules apply towards the flag charts in the instruction description section.

An overflow condition will return infinity (or DEC reserved operand) if the wrapped overflow mode is reset. An overflow condition will return a wrapped number if the wrapped overflow mode is set, independent of the wrapped underflow mode.

If the wrapped underflow mode is reset, both the FMPY and FALU treat denormalized inputs as exact zero. If the wrapped underflow mode is set, the FALU handles denormalized inputs directly, whereas the FMPY treats them as inexact zeros.

The following sections have been included:

- Instruction symbols
- Instruction set description

INSTRUCTION SYMBOLS

SYMBOL	DEFINITION
CRY	Carry
DEN	Indicates a denormalized number
DIVZ	Divide by 0
DP	64 bit floating point number
DX	Denormalized input X
DY	Denormalized input Y
FALU	Floating point arithmetic logic unit
FMPY	Floating point multiplier
INF	Infinity
INV	Invalid
INX	Inexact
L	Long integer–64 bits
M	Largest magnitude normalized number
N	Negative
n	User determined binary number
N/A	Not applicable
NaN	Not a number
NORM	Indicates a normalized number
OV	Overflow
Q	Quiet NaN
R	Indicates DEC reserved operand
RND	Rounded up
S	Signaling NaN
sb	Indicates sticky bit
SP	32 bit floating point number
UF	Underflow
WRP	Indicates a wrapped number
ZR	Zero
$\sqrt{\ }$	Square root
/	Divide
•	Multiply
*	Indicates status flags are affected
|	Concatenation
| |	Absolute value
[,]	Items within braces are alternative items, one of them must be used

■ INSTRUCTION SET DESCRIPTION

FLOATING POINT ARITHMETIC INSTRUCTIONS

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
DIV	0 0 0 0 0 0 0 0	X/Y
DDIV	0 0 0 0 0 0 0 1	DP: X/Y

FLAGS AFFECTED

N/A	*	*	*	*	*	*			*	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: Floating Point Division.

Comments: In the tables below, the first entry represents the flag that is set, the second represents the returned result.

IEEE-WRAPPED UNDERFLOW MODE

Y OPERAND \ X OPERAND	0	DEN	NORM	INF	Q	S
0	INV, NaN/Q	DIVZ/INF	DIVZ/INF	INF	NaN/Q	INV, NaN/Q
DEN	ZR/0	INV, NaN/Q	INX, DIVZ/INF	INF	NaN/Q	INV, NaN/Q
NORM	ZR/0	INX, ZR/0	OV/[WRP, INF, M] NORM UF/WRP	INF	NaN/Q	INV, NaN/Q
INF	ZR/0	ZR/0	ZR/0	INV, NaN/Q	NaN/Q	INV, NaN/Q
Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	INV, NaN/Q
S	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q

DEC MODE

Y OPERAND \ X OPERAND	0	NORM	R
0	INV/R	DIVZ/R	INV/R
NORM	ZR/0	OV/R NORM UF, ZR/0	INV/R
R	INV/R	INV/R	INV/R

IEEE-WRAPPED UNDERFLOW MODE DISABLED

Y OPERAND \ X OPERAND	0	DEN	NORM	INF	Q	S
0	INV, NaN/Q	INV, NaN/Q	DIVZ/INF	INF	NaN/Q	INV, NaN/Q
DEN	INV, NaN/Q	INV, NaN/Q	DIVZ/INF	INF	NaN/Q	INV, NaN/Q
NORM	ZR/0	ZR/0	OV/[WRP, INF, M] NORM UF, ZR/0	INF	NaN/Q	INV, NaN/Q
INF	ZR/0	ZR/0	ZR/0	INV, NaN/Q	NaN/Q	INV, NaN/Q
Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	INV, NaN/Q
S	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q

MNEMONIC	OPCODE	FUNCTION
SQRTX	0 0 0 0 0 0 1 0	$\sqrt{X}$
DSQRTX	0 0 0 0 0 0 1 1	DP: $\sqrt{X}$

FLAGS AFFECTED

N/A	0	0	*	*	*	*	*	0	0	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: Floating point square root. The square root of operand X is returned.

Comments: The Y operand should not be changed during square root instructions. In the tables below, the first entry represents the flag that is set, the second represents the result returned to Z.

IEEE-WRAPPED UNDERFLOW MODE

X OPERAND	Z RESULT
S	INV, NaN/Q
Q	NaN/Q
−INF	INV, NaN/Q
−NORM	INV, NaN/Q
−DEN	INV, NaN/Q
−0	ZR/−0
+0	ZR/+0
+DEN	INX, ZR/0
+NORM	+NORM
+INF	+INF

IEEE-WRAPPED UNDERFLOW MODE DISABLED

X OPERAND	Z RESULT
S	INV, NaN/Q
Q	NaN/Q
−INF	INV, NaN/Q
−NORM	INV, NaN/Q
−DEN	ZR/−0
−0	ZR/−0
+0	ZR/+0
+DEN	ZR/+0
+NORM	+NORM
+INF	+INF

DEC MODE

X OPERAND	Z RESULT
R	INV/R
0	ZR/0
+NORM	+NORM
−NORM	INV/R

FLOATING POINT ARITHMETIC INSTRUCTIONS (cont'd)

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
MULTWX	0 0 0 0 0 1 0 0	WRAPPED X • Y
DMULTWX	0 0 0 0 0 1 0 1	DP: WRAPPED X • Y
MULTWY	0 0 0 0 0 1 1 0	X • WRAPPED Y
DMULTWY	0 0 0 0 0 1 1 1	DP: X • WRAPPED Y

FLAGS AFFECTED

N/A	0	*	*	*	*	*	*	*	0	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: Floating point multiplication. The wrapped operand is multiplied by [X,Y].

Comments: The chip must have IEEE Underflow mode set to one, or underflows (wrapped outputs) will be set to zero. The wrapped multiply instructions assume that their operands are wrapped underflows. They do not work on wrapped overflows.

The result of WRAPPED X • WRAPPED Y is always too small to make a denormalized number, thus the result always underflows and no instruction is provided.

The denormalized number flag will not be set if the wrapped number looks like a denormalized number.

If the result is too small to return as a wrapped number, then the underflow and zero flags are raised, and inexact zero is returned.

In the tables below, the first entry represents the flag that is set, the second represents the returned result.

IEEE-WRAPPED UNDERFLOW MODE

UNWRAPPED OPERAND	Z RESULT
0	ZR/0
DEN	INX, ZR/0
NORM	NORM UF/WRP ZR, UF/0[1]
INF	INF
Q	NaN/Q
S	INV, NaN/Q

IEEE-WRAPPED UNDERFLOW MODE DISABLED

UNWRAPPED OPERAND	Z RESULT
0	ZR/0
DEN	ZR/0
NORM	NORM ZR, UF/0
INF	INF
Q	NaN/Q
S	INV, NaN/Q

NOTE[1]: Double underflow–the result is too small to return a wrapped result.

MULT	0 0 0 0 1 0 0 0	X • Y
DMULT	0 0 0 0 1 0 0 1	DP: X • Y
MULTAY	0 0 0 0 1 0 1 0	X • |Y|
DMULTAY	0 0 0 0 1 0 1 1	DP: X • |Y|
MULTAX	0 0 0 0 1 1 0 0	|X| • Y
DMULTAX	0 0 0 0 1 1 0 1	DP: |X| • Y
MULTA	0 0 0 0 1 1 1 0	|X • Y|
DMULTA	0 0 0 0 1 1 1 1	DP: |X • Y|

N/A	0	*	*	*	*	*	*	*	*	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: Floating point multiplication. [|X| , |Y|] represents the absolute value of [X,Y]. |X • Y| is the absolute value of the result.

Comments: In the tables below, the first entry represents the flag that is set, the second represents the returned result. In round to + INF mode, positive underflows return the smallest positive normalized number. In round to – INF mode, negative underflows return the smallest magnitude negative normalized number.

IEEE-WRAPPED UNDERFLOW MODE

Y OPERAND \ X OPERAND	0	DEN	NORM	INF	Q	S
0	ZR/0	ZR/0	ZR/0	INV, NaN/Q	NaN/Q	INV, NaN/Q
DEN	ZR/0	INX, ZR/0	INX, ZR/0	INF	NaN/Q	INV, NaN/Q
NORM	ZR/0	INX, ZR/0	OV/[WRP, INF, M] NORM UF/WRP	INF	NaN/Q	INV, NaN/Q
INF	INV, NaN/Q	INF	INF	INF	NaN/Q	INV, NaN/Q
Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	INV, NaN/Q
S	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q

DEC MODE

Y OPERAND \ X OPERAND	0	NORM	R
0	ZR/0	ZR/0	INV/R
NORM	ZR/0	OV/R NORM ZR, UF/0	INV/R
R	INV/R	INV/R	INV/R

FLOATING POINT ARITHMETIC INSTRUCTIONS (cont'd)

MNEMONIC | **OPCODE** $I_7\ I_6\ I_5\ I_4\ I_3\ I_2\ I_1\ I_0$ | **FUNCTION** | **FLAGS AFFECTED**

IEEE-WRAPPED UNDERFLOW MODE DISABLED

Y OPERAND \ X OPERAND	0	DEN	NORM	INF	Q	S
0	ZR/0	ZR/0	ZR/0	INV, NaN/Q	NaN/Q	INV, NaN/Q
DEN	ZR/0	ZR/0	ZR/0	INV, NaN/Q	NaN/Q	INV, NaN/Q
NORM	ZR/0	ZR/0	OV/[WRP, INF, M] NORM ZR, UF/0	INF	NaN/Q	INV, NaN/Q
INF	INV, NaN/Q	INV, NaN/Q	INF	INF	NaN/Q	INV, NaN/Q
Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	INV, NaN/Q
S	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q

MAC	0 0 0 1 0 0 1 0	X • Y, X + Y
DMAC	0 0 0 1 0 0 1 1	DP: X • Y, X + Y
MACS	0 0 0 1 0 1 0 0	X • Y, X − Y
DMACS	0 0 0 1 0 1 0 1	DP: X • Y, X − Y
SMAC	0 0 0 1 0 1 1 0	X • Y, Y − X
DSMAC	0 0 0 1 0 1 1 1	DP: X • Y, Y − X

FMPY

N/A	0	*	*	*	*	*	*	*	*	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

FALU

0	N/A	*	*	*	*	*	*	*	*	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: Floating point multiply/accumulate instruction. Multiplication is performed by the FMPY, whereas addition or subtraction is performed by the FALU.

Comments: These commands may be used as true multiply/accumulate instructions when the FALU and FMPY outputs are connected and the FALU input multiplexer is configured in "feedback mode." In feedback mode, the FMPY output is fed back as the FALU X operand and the FALU result is fed back as the Y operand. See the ADD and MULT instructions for details regarding the flags and operation result.

MIN	0 0 1 0 0 1 0 0	Floating point MIN
DMIN	0 0 1 0 0 1 0 1	DP: Floating point MIN
MAX	0 0 1 0 0 1 1 0	Floating point MAX
DMAX	0 0 1 0 0 1 1 1	DP: Floating point MAX

*	N/A	*	*	*	0	0	*	*	0	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: These floating point instructions return the smaller of the two operands X and Y (MIN/DMIN) or larger (MAX,DMAX).

Comments: The carry flag is reset if X is returned, otherwise it is set. X is returned if X = Y, except that MAX/DMAX (−0, +0) = +0 and MIN/DMIN (+0, −0) = −0. The Invalid Op flag is set if either operand is a signaling NaN. If either operand is not-a-number, then the result is not-a-number.

In IEEE wrapped underflow mode, a denormalized result is wrapped and the underflow flag is set. If either operand is a NaN, the carry flag is unspecified.

ABSX	0 0 1 0 1 0 0 0	|X|
DABSX	0 0 1 0 1 0 0 1	DP: |X|
NEGX	0 0 1 0 1 0 1 0	− X
DNEGX	0 0 1 0 1 0 1 1	DP: − X
PASSX	0 0 1 0 1 1 0 0	X
DPASSX	0 0 1 0 1 1 0 1	DP: X

*	N/A	0	*	*	0	0	*	*	0	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: These single operand floating point instructions use the X operand input. PASSX/DPASSX returns X through the FALU. If X is denormalized and the wrapped underflow mode is reset, zero is returned, otherwise a wrapped result is returned.

Comments: PASSX/DPASSX with an infinite input sets CRY, otherwise CRY is reset. In IEEE wrapped underflow mode, the denormalized result is wrapped and the underflow flag is set.

ADD	0 0 1 1 0 0 0 0	X + Y
DADD	0 0 1 1 0 0 0 1	DP: X + Y
SUB	0 0 1 1 0 0 1 0	X − Y
DSUB	0 0 1 1 0 0 1 1	DP: X − Y
SUBX	0 0 1 1 0 1 0 0	Y − X
DSUBX	0 0 1 1 0 1 0 1	DP: Y − X
ADDA	0 0 1 1 1 0 0 0	|X| + |Y|
DADDA	0 0 1 1 1 0 0 1	DP: |X| + |Y|
SUBA	0 0 1 1 1 0 1 0	|X| − |Y|
DSUBA	0 0 1 1 1 0 1 1	DP: |X| − |Y|
SUBXA	0 0 1 1 1 1 0 0	|Y| − |X|
DSUBXA	0 0 1 1 1 1 0 1	DP: |Y| − |X|

0	N/A	*	*	*	*	*	*	*	*	*	*
CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N

Description: Floating point addition and subtraction. [|X| , |Y|] represents the absolute value of [X,Y].

Comments: When the sum of two operands with opposite signs (or the difference of two operands with like signs) is exactly zero, the result is +0 for all rounding modes except −INF, in which case the result is −0. Note that (+0) + (+0) = (+0) − (−0) = +0 and (−0) + (−0) = (−0) − (+0) = −0 for all rounding modes.

In the tables below, the first entry represents the flag that is set, the second represents the result returned to Z.

FLOATING POINT ARITHMETIC INSTRUCTIONS (cont'd)

MNEMONIC | **OPCODE** $I_7\ I_6\ I_5\ I_4\ I_3\ I_2\ I_1\ I_0$ | **FUNCTION** | **FLAGS AFFECTED**

IEEE–WRAPPED UNDERFLOW MODE

Y OPERAND \ X OPERAND	0	DEN	NORM	INF	Q	S
0	ZR/0	UF/WRP	NORM	INF	NaN/Q	INV, NaN/Q
DEN	UF/WRP	UF/WRP NORM	OV/[WRP, INF, M] NORM UF/WRP	INF	NaN/Q	INV, NaN/Q
NORM	NORM	OV/[WRP, INF] NORM UF/WRP	OV/[WRP, INF, M] NORM UF/WRP	INF	NaN/Q	INV, NaN/Q
INF	INF	INF	INF	INF[1] INV, NaN/Q[1]	NaN/Q	INV, NaN/Q
Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	INV, NaN/Q
S	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q

DEC MODE

Y OPERAND \ X OPERAND	0	NORM	R
0	ZR/0	NORM	INV/R
NORM	NORM	OV/R NORM ZR, UF/0	INV/R
R	INV/R	INV/R	INV/R

NOTE[1]: (+INF)+(−INF) → NaN (+INF)+(+INF) → +INF
(+INF)−(+INF) → NaN (−INF)+(−INF) → −INF
(−INF)−(−INF) → NaN (+INF)−(−INF) → +INF
(−INF)+(+INF) → NaN (−INF)−(+INF) → −INF

IEEE–WRAPPED UNDERFLOW MODE DISABLED

Y OPERAND \ X OPERAND	0	DEN	NORM	INF	Q	S
0	ZR/0	ZR/0	NORM	INF	NaN/Q	INV, NaN/Q
DEN	ZR/0	ZR/0	NORM	INF	NaN/Q	INV, NaN/Q
NORM	NORM	NORM	OV/[WRP, INF, M] NORM ZR, UF/0	INF	NaN/Q	INV, NaN/Q
INF	INF	INF	INF	INF[1] INV, NaN/Q[1]	NaN/Q	INV, NaN/Q
Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	NaN/Q	INV, NaN/Q
S	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q	INV, NaN/Q

FLOATING POINT SUPPORT INSTRUCTIONS

MNEMONIC | **OPCODE** $I_7\ I_6\ I_5\ I_4\ I_3\ I_2\ I_1\ I_0$ | **FUNCTION** | **FLAGS AFFECTED**

PASSXM 0 0 0 1 0 0 0 0 X
DPASSXM 0 0 0 1 0 0 0 1 DP: X

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
N/A	0	0	0	0	0	0	0	0	0	*	*

Description: The X input is returned unmodified through the FMPY.
Comments: The zero and negative flags are set as if the result is a signed integer. The remaining flags are reset.

SCALE 0 0 1 0 0 0 0 0 EXPONENT X + Y
DSCALE 0 0 1 0 0 0 0 1 DP: EXPONENT X + Y

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	0	0	0	0	0	*	*	0	*

Description: The integer input Y is added to the exponent of X. The sign and mantissa of X are passed unmodified.
Comments: Overflows and underflows always return a wrapped result regardless of the overflow or underflow mode. The least significant 8 bits (single precision) or 11 bits (double precision) of Y are interpreted as a two's complement integer. Other bits of Y are ignored.

MERGE 0 0 1 0 0 0 1 0 SIGN X|EXPONENT Y|MANTISSA X
DMERGE 0 0 1 0 0 0 1 1 DP: SIGN X|EXPONENT Y|MANTISSA X

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	0	0	0	0	0	*	*	0	*

Description: The exponent field of Y is concatenated with the sign and mantissa field of X.
Comments: If a NaN or INF results, the overflow flag is set. If a denormalized number or zero results, the underflow flag is set. Use PASSX/DPASSX to check result type.

NORMX 0 0 1 0 1 1 1 0 NORMALIZE X

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	0	0	0	0	0	0	0	*	0

Description: X is assumed to be a 32-bit unsigned positive integer. The leading one of X will be left shifted to the most significant bit position. The shift count will be placed in the SC register and output as the result.
Comments: The ZR flag is set if the input was zero.

FLOATING POINT SUPPORT INSTRUCTIONS (cont'd)

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
CMPR	0 0 1 1 0 1 1 0	X,Y
DCMPR	0 0 1 1 0 1 1 1	DP: X,Y
CMPRA	0 0 1 1 1 1 1 0	|X|, |Y|
DCMPRA	0 0 1 1 1 1 1 1	DP: |X|, |Y|

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	*	*	*	0	0	*	0	0	*	*

Description: Floating point compare. |X| , |Y| represents the absolute value of X,Y. The following values will be returned to the result based on the relative magnitude of operands X and Y.

Comments: Exactly one of CRY, N, ZR or NaN will be set by compare. For example, if X = Y in round to minus infinity mode, −0 is returned, and the ZR flag is set but the N flag is reset. If a compare is performed on a signaling NaN, the INV flag will be set.

Input	Flag	Output
X > Y	CRY	1
X < Y	N	−1
X = Y	ZR	0 (−0 in round to −INF mode)
[X,Y] NaN	NaN	NaN

When comparing ±INF and ±0, the following states occur:

Input X	Input Y	Flag	Output
±0	±0	ZR	0
+INF	+INF	ZR	0
+INF	−INF	CRY	1
−INF	+INF	N	−1
−INF	−INF	ZR	0

PASSn 0 1 0 0 n n n n X • 16 + n

Description: X is logically left shifted four places and the four least significant bits of the opcode field are added to it.

Comments: Allows microcode to build constants on the datapath. PASSn does not change the flags. In this respect, it acts like a NOP.

SCREGR	0 1 0 1 0 0 0 0	SC register read
SCREGW	0 1 0 1 0 0 0 1	SC register write
FREGAR	0 1 0 1 0 0 1 0	FALU flag register read
FREGAW	0 1 0 1 0 0 1 1	FALU flag register write
FREGMR	0 1 0 1 1 0 1 0	FMPY flag register read
FREGMW	0 1 0 1 1 0 1 1	FMPY flag register write
IREGAR	0 1 0 1 0 1 0 0	FALU int register read
IREGAW	0 1 0 1 0 1 0 1	FALU int register write
IREGMR	0 1 0 1 1 1 0 0	FMPY int register read
IREGMW	0 1 0 1 1 1 0 1	FMPY int register write
MREGAR	0 1 0 1 0 1 1 0	FALU mode register read
MREGAW	0 1 0 1 0 1 1 1	FALU mode register write
MREGMR	0 1 0 1 1 1 1 0	FMPY mode register read
MREGMW	0 1 0 1 1 1 1 1	FMPY mode register write

Description: Register access instructions. SCREGx accesses the SC register (FALU only), FREGx accesses the flag register, IREGx accesses the interrupt enable register and MREGx accesses the mode register.

NOP 0 1 0 1 1 0 0 0 No operation

Description: All registers and flags remain unchanged. The result is unspecified.

Comments: Parity is checked during NOP's (and all unimplemented instructions).

CLRFLAG 0 1 0 1 1 0 0 1 Clear flag register

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	0	0	0	0	0	0	0	0	0	0	0

Description: The flag register is cleared, the SC (FALU only), interrupt enable and mode registers are unaffected.

Comments: If an interrupt has frozen the flag register (see freeze on interrupt mode), CLRFLAG will clear and unfreeze the register.

CONVERSION INSTRUCTIONS

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
FCUI	0 1 1 0 0 0 0 0	SP → 32-bit unsigned integer
DFCUI	0 1 1 0 0 0 0 1	DP → 32-bit unsigned integer
FCSI	0 1 1 0 0 0 1 0	SP → 32-bit signed integer
DFCSI	0 1 1 0 0 0 1 1	DP → 32-bit signed integer
UICF	0 1 1 0 0 1 0 0	SP ← 32-bit unsigned integer
UICDF	0 1 1 0 0 1 0 1	DP ← 32-bit unsigned integer
SICF	0 1 1 0 0 1 1 0	SP ← 32-bit signed integer
SICDF	0 1 1 0 0 1 1 1	DP ← 32-bit signed integer
FCLUI	0 1 1 0 1 0 0 0	SP → 64-bit unsigned integer
DFCLUI	0 1 1 0 1 0 0 1	DP → 64-bit unsigned integer
FCLSI	0 1 1 0 1 0 1 0	SP → 64-bit signed integer
DFCLSI	0 1 1 0 1 0 1 1	DP → 64-bit signed integer
LUICF	0 1 1 0 1 1 0 0	SP ← 64-bit unsigned integer
LUICDF	0 1 1 0 1 1 0 1	DP ← 64-bit unsigned integer
LSICF	0 1 1 0 1 1 1 0	SP ← 64-bit signed integer
LSICDF	0 1 1 0 1 1 1 1	DP ← 64-bit signed integer
FCUIT	0 1 1 1 0 0 0 0	SP → 32-bit unsigned integer (Rnd to 0)
DFCUIT	0 1 1 1 0 0 0 1	DP → 32-bit unsigned integer (Rnd to 0)
FCSIT	0 1 1 1 0 0 1 0	SP → 32-bit signed integer (Rnd to 0)
DFCSIT	0 1 1 1 0 0 1 1	DP → 32-bit signed integer (Rnd to 0)
FCLUIT	0 1 1 1 1 0 0 0	SP → 64-bit unsigned integer (Rnd to 0)
DFCLUIT	0 1 1 1 1 0 0 1	DP → 64-bit unsigned integer (Rnd to 0)
FCLSIT	0 1 1 1 1 0 1 0	SP → 64-bit signed integer (Rnd to 0)
DFCLSIT	0 1 1 1 1 0 1 1	DP → 64-bit signed integer (Rnd to 0)

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	*	*	*	*	*	0	0	*	*

Description: Floating point to integer and integer to floating point conversion instructions. Input operand X is converted to the indicated format. All instructions follow the programmed rounding mode (see mode register), except the xxxT format instructions which always round toward zero.

Comments: When a floating point to integer conversion instruction overflows, the invalid operation flag is set and the result is either the most positive (for positive overflows) or the most negative (for negative overflows) integer. For example:

Input	2^{65}	$-(2^{65})$
32 bit signed result:	7FFFFFFF	80000000
64 bit signed result:	7FFFFFFFFFFFFFFF	8000000000000000
32 bit unsigned result:	FFFFFFFF	00000000
64 bit unsigned result:	FFFFFFFFFFFFFFFF	0000000000000000

If a NaN is converted from floating point to an integer, the result is an overflow with the sign of the NaN.

Integer to floating point instructions can never set NaN flag.

MNEMONIC	OPCODE	FUNCTION
WDNM	0 1 1 1 0 1 0 0	WRAPPED → DENORM
DWDNM	0 1 1 1 0 1 0 1	DP: WRAPPED → DENORM

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	0	0	*	*	0	*	0	*	*

Description: This floating point FALU instruction will convert the wrapped X input to a denormalized number. Inexact and rounded up bits are used as additional inputs.

Comments: The underflow flag is set if the result is inexact. This corresponds to the IEEE specification which says that an underflow shall be signaled if a result is denormalized and inexact.

The inexact and rounded-up flags are used as inputs to prevent a double rounding error. These flags must be set equal to the corresponding flags of the operation that produces the wrapped underflow. The rounding mode must also be the same as when the wrapped underflow was produced. The rounded up flag is unspecified after this operation.

In normal operation, the inexact and rounded up flags are latched externally and written to FALU just before the WDNM/DWDNM instruction.

MNEMONIC	OPCODE	FUNCTION
SDF	0 1 1 1 0 1 1 0	SP → DP

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	*	*	0	0	*	0	0	*	*

Description: Floating point precision conversion instruction. Conversion is carried out on the X operand.

MNEMONIC	OPCODE	FUNCTION
DSDF	0 1 1 1 0 1 1 1	DP → SP

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	*	*	*	*	*	*	*	*	*

Description: Floating point precision conversion instruction. Conversion is carried out on the X operand.

Comments: The carry flag is set if a normalized double precision number is output as a single precision infinity. This will occur if overflows are not wrapped, or if the result is too large to be represented by a wrapped overflow. If a result underflows to the extent that it cannot be wrapped, zero is returned and the zero, underflow and inexact flags are set.

Reference the IEEE Std 754 section 7.3 (overflows) for additional information.

CONVERSION INSTRUCTIONS (cont'd)

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
FFI	0 1 1 1 1 1 0 0	SP → SP format integer
DFFI	0 1 1 1 1 1 0 1	DP → DP format integer
FFIT	0 1 1 1 1 1 1 0	SP → SP format integer (Rnd to 0)
DFFIT	0 1 1 1 1 1 1 1	DP → DP format integer (Rnd to 0)

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	*	*	*	*	-	0	0	*	*

Description: Floating point conversion instruction. The floating point number at the X operand input is rounded to an integral valued floating point number in the same format.
Comments: FFIT/DFFIT always rounds toward zero, regardless of the rounding mode. The INX flag will be set if the result is different from the input. The RND flag will be set if the magnitude of the result is greater than the magnitude of the input.

INTEGER ARITHMETIC INSTRUCTIONS

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
IADD	1 1 1 0 0 0 0 0	X + Y
LIADD	1 0 1 0 0 0 0 0	L: X + Y
ISUB	1 1 1 0 0 1 0 1	X – Y
LISUB	1 0 1 0 0 1 0 1	L: X – Y
ISUBX	1 1 1 0 0 1 1 0	Y – X
LISUBX	1 0 1 0 0 1 1 0	L: Y – X
IADDP	1 1 1 0 0 1 0 0	X + Y + 1
LIADDP	1 0 1 0 0 1 0 0	L: X + Y + 1
ISUBM	1 1 1 0 0 0 0 1	X – Y – 1
LISUBM	1 0 1 0 0 0 0 1	L: X – Y – 1
ISUBXM	1 1 1 0 0 0 1 0	Y – X – 1
LISUBXM	1 0 1 0 0 0 1 0	L: Y – X – 1
IADDC	1 1 1 0 1 0 0 0	X + Y + CARRY
LIADDC	1 0 1 0 1 0 0 0	L: X + Y + CARRY
ISUBC	1 1 1 0 1 0 0 1	X – Y – $\overline{\text{CARRY}}$
LISUBC	1 0 1 0 1 0 0 1	L: X – Y – $\overline{\text{CARRY}}$
ISUBXC	1 1 1 0 1 0 1 0	Y – X – $\overline{\text{CARRY}}$
LISUBXC	1 0 1 0 1 0 1 0	L: Y – X – $\overline{\text{CARRY}}$

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	*	*	*

Description: Integer addition and subtraction instructions.
Comments: The function of the carry input used for ISUBC, LISUBC, ISUBXC, LISUBXC can be changed with mode register bit < 7 > (borrow mode). See mode register section for additional information.

INEGC	1 1 1 0 1 0 1 1	– X – $\overline{\text{CARRY}}$
LINEGC	1 0 1 0 1 0 1 1	L: – X – $\overline{\text{CARRY}}$
IABSX	1 1 1 0 1 1 1 1	|X|
LIABSX	1 0 1 0 1 1 1 1	L: |X|
INEGX	1 1 1 0 0 1 1 1	– X
LINEGX	1 0 1 0 0 1 1 1	L: – X

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	*	*	*

Description: Single operand integer arithmetic instructions.
Comments: The function of the carry input used for INEGC and LINEGC can be changed with mode register bit < 7 > (borrow mode). See mode register section for additional information.

ISMAX	1 1 0 0 0 0 1 0	Signed MAX
LISMAX	1 0 0 0 0 0 1 0	L: Signed MAX
ISMIN	1 1 0 0 0 1 1 0	Signed MIN
LISMIN	1 0 0 0 0 1 1 0	L: Signed MIN
IUMAX	1 1 0 0 1 0 1 0	Unsigned MAX
LIUMAX	1 0 0 0 1 0 1 0	L: Unsigned MAX
IUMIN	1 1 0 0 1 1 1 0	Unsigned MIN
LIUMIN	1 0 0 0 1 1 1 0	L: Unsigned MIN

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	0	*	*

Description: The larger of the two operands X and Y is returned (ISMAX/LISMAX, IUMAX/LIUMAX) or the smaller of the two operands is returned (ISMIN/LISMIN, IUMIN/LIUMIN).
Comments: Sign and zero flags are set based on returned result. The carry flag is reset if X is returned, otherwise it is set. X is returned if X = Y.

INTEGER ARITHMETIC INSTRUCTIONS (cont'd)

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
IMULT	1 1 1 1 1 0 0 0	Unsigned X • Unsigned Y
IMULTSX	1 1 1 1 1 0 0 1	Signed X • Unsigned Y
IMULTSY	1 1 1 1 1 0 1 0	Unsigned X • Signed Y
IMULTS	1 1 1 1 1 0 1 1	Signed X • Signed Y
IMULTH	1 1 1 1 1 1 0 0	Unsigned X • Unsigned Y
IMULTHSX	1 1 1 1 1 1 0 1	Signed X • Unsigned Y
IMULTHSY	1 1 1 1 1 1 1 0	Unsigned X • Signed Y
IMULTHS	1 1 1 1 1 1 1 1	Signed X • Signed Y

FLAGS AFFECTED

IMULT

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
N/A	0	0	0	0	0	0	0	0	0	*	*

IMULTH

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
N/A	0	0	0	0	0	0	0	0	*	*	*

Description: Integer multiplication instructions. IMULT returns a 64 bit result, whereas IMULTH returns the least significant 32 bits.

INTEGER BOOLEAN INSTRUCTIONS

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
INAND	1 1 0 1 0 0 0 0	$\overline{X}$ or $\overline{Y}$
LINAND	1 0 0 1 0 0 0 0	L: $\overline{X}$ or $\overline{Y}$
IORNX	1 1 0 1 0 0 0 1	$\overline{X}$ or Y
LIORNX	1 0 0 1 0 0 0 1	L: $\overline{X}$ or Y
IORNY	1 1 0 1 0 0 1 0	X or $\overline{Y}$
LIORNY	1 0 0 1 0 0 1 0	L: X or $\overline{Y}$
IOR	1 1 0 1 0 0 1 1	X or Y
LIOR	1 0 0 1 0 0 1 1	L: X or Y
IANDNY	1 1 0 1 0 1 0 0	X and $\overline{Y}$
LIANDNY	1 0 0 1 0 1 0 0	L: X and $\overline{Y}$
IAND	1 1 0 1 0 1 0 1	X and Y
LIAND	1 0 0 1 0 1 0 1	L: X and Y
INOR	1 1 0 1 0 1 1 0	$\overline{X}$ and $\overline{Y}$
LINOR	1 0 0 1 0 1 1 0	L: $\overline{X}$ and $\overline{Y}$
IANDNX	1 1 0 1 0 1 1 1	$\overline{X}$ and Y
LIANDNX	1 0 0 1 0 1 1 1	L: $\overline{X}$ and Y
IXNOR	1 1 0 1 1 1 1 0	X XNOR Y
LIXNOR	1 0 0 1 1 1 1 0	L: X XNOR Y
IXOR	1 1 0 1 1 1 1 1	X XOR Y
LIXOR	1 0 0 1 1 1 1 1	L: X XOR Y
ISET	1 1 0 1 1 0 0 0	Z = all ones
LISET	1 0 0 1 1 0 0 0	L: Z = all ones
INOTX	1 1 0 1 1 0 0 1	Z = $\overline{X}$
LINOTX	1 0 0 1 1 0 0 1	L: Z = $\overline{X}$
IPASSY	1 1 0 1 1 0 1 0	Z = Y
LIPASSY	1 0 0 1 1 0 1 0	L: Z = Y
IPASSX	1 1 0 1 1 0 1 1	Z = X
LIPASSX	1 0 0 1 1 0 1 1	L: Z = X
ICLR	1 1 0 1 1 1 0 0	Z = all zeroes
LICLR	1 0 0 1 1 1 0 0	L: Z = all zeroes
INOTY	1 1 0 1 1 1 0 1	Z = $\overline{Y}$
LINOTY	1 0 0 1 1 1 0 1	L: Z = $\overline{Y}$

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
0	N/A	0	0	0	0	0	0	0	0	*	*

Description: Boolean logic instructions.
Comments: Flags are set based on signed operands.

INTEGER SHIFT AND ROTATE INSTRUCTIONS

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
LSS	1 1 1 1 0 0 0 0	Logical shift X w/sb
LLSS	1 0 1 1 0 0 0 0	L: Logical shift X w/sb
LS	1 1 1 1 0 0 0 1	Logical shift X
LLS	1 0 1 1 0 0 0 1	L: Logical shift X

FLAGS AFFECTED

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	0	*	*

Description: Integer shift instructions. The two's complement number in the SC register determines the shift. If SC > 0, then shift left by SC; if SC < 0, then shift right by − SC. If SC = 0, no shift is performed. For LSS/LLSS right shifts, all bits of the result shifted out are ored with the least significant bit position of the result (i.e. sticky bit).

Zeroes are shifted in during right and left shifts.

Comments: Carry bit receives the last bit shifted out of the operand if SC = 0. Flags are set based on a signed operand. The overflow flag is reset. Carry is reset if SC = 0.

AS	1 1 1 1 0 0 1 0	Arithmetic shift X
LAS	1 0 1 1 0 0 1 0	L: Arithmetic shift X

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	*	*	*

Description: Arithmetic shift instruction. The two's complement number in the SC register determines the shift. If SC > 0, then shift left by SC; if SC < 0, then shift right by − SC. If SC = 0, no shift is performed.

For left shifts, zeroes are shifted in. For right shifts, AS/LAS shifts in copies of the sign bit.

Comments: Carry bit receives the last bit shifted out of the operand if SC = 0. Carry is reset if SC = 0. Flags are set based on a signed operand. Left AS/LAS shifts set the OV flag if any bits shifted out differ from the result sign.

ROTX	1 1 1 1 0 0 1 1	Rotate X
LROTX	1 0 1 1 0 0 1 1	L: Rotate X

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	0	*	*

Description: Rotate X by the signed two's complement number in the shift count (SC) register. If SC > 0, rotate left by SC. If SC < 0, rotate right by − SC. If SC = 0, no shift is performed. As bits are shifted out, they are used as shift inputs.

Comments: Carry gets the last bit wrapped from one end of the word to the other. Carry is reset if SC = 0.

ROTC	1 1 1 1 0 1 0 0	Rotate Y|X
LROTC	1 0 1 1 0 1 0 0	L: Rotate Y|X

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	0	*	*

Description: The 32 bit integers Y and X are concatenated and rotated by the signed two's complement number in the shift count register. Before the rotate operation, Y is in the most significant word position. If SC > 0, rotate left by SC. If SC < 0, rotate right by − SC. If SC = 0, no shift is performed. As bits are shifted out, they are used as shift inputs. LROTC returns all 64 bits of the result. ROTC returns the least significant 32 bits of the result.

Comments: Carry gets the last bit wrapped from one end of the word to the other. Carry is reset if SC = 0.

BITR	1 1 1 1 0 1 0 1	Rotate bit reversed X|X

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	0	*	*

Description: The 32 bit integer X is bit-reversed and concatenated with a non-bit-reversed X (bit reverse is defined as Y<0> → Y<31>, Y<1> → Y<30>, etc.) Before the rotate instruction, the non-bit-reversed operand X is in the least significant word position. If SC > 0, rotate left by SC. If SC < 0, rotate right by − SC. If SC = 0, no shift is performed. As bits are shifted out, they are used as shift inputs. After rotation, the least significant 32 bits are returned. To generate the bit reverse of X set SC = 32.

Comments: Carry gets the last bit wrapped from one end of the word to the other. Carry is reset if SC = 0.

ADDSC	1 1 1 1 0 1 1 0	Z, SC ← X + SC
NEGSC	1 1 1 1 0 1 1 1	Z, SC ← − SC

CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N
*	N/A	0	0	0	0	0	0	0	*	*	*

Description: Integer SC register instructions. The 32-bit two's complement operand X is added to the SC register (ADDSC) or the SC register is negated (NEGSC).

Comments: Flags are affected based on signed operations. The new value of the SC register is returned. The operation of the CRY flag is affected by mode register bit <7> (borrow mode) for NEGSC.

UNUSED OPCODES

MNEMONIC	OPCODE I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	FUNCTION
	0 0 0 1 1 x x x	Not used
	0 0 1 0 1 1 1 1	Not used
	1 x 1 0 0 0 1 1	Not used
	1 x 1 0 1 1 0 x	Not used
	1 x 1 0 1 1 1 0	Not used
	1 x 0 0 x x 0 x	Not used
	1 x 0 0 x x 1 1	Not used
	1 0 1 1 0 1 0 1	Not used
	1 0 1 1 0 1 1 x	Not used
	1 0 1 1 1 x x x	Not used

NOTE: Unused opcodes do not affect flags, but data results are undefined. Analog Devices reserves the right to use the opcodes in future products.

RESET OPERATION

Use of the asynchronous hardware reset causes the following events to occur:

- All flags are cleared
- All interrupt enable bits are cleared (prevents interrupt flag from being set)
- The SC register is set to zero
- The Mode register is set to zero

Therefore after a reset the FMPY and FALU will be configured with the following operating modes:

- IEEE mode
- Freeze on interrupt mode disabled
- Parity is disabled
- Underflows and denormalized numbers are set to zero
- Overflows are set to infinity
- Round to nearest
- Borrow mode disabled
- The parity flag is not sticky

DATA PATH OPERATION

The FMPY and FALU provide two 36 bit input data ports (X,Y) and one 36 bit bidirectional data port (T). See the FMPY/FALU block diagrams. Data present at the T port is internally fed back to the X input multiplexer and may be selected with XSEL. In addition, the 64 bit Z result can be fed back to the Y input multiplexer (FALU only) and selected with YSEL. An operation which uses either feedback path must have the same precision as the operation which generated the feedback data.

Two modes are available for clocking the X and Y operands on the FALU. When R/L1 = 1, XA and YA are configured as edge-triggered registers. Data is loaded on the rising edge of CK1. R/L1 = 0 configures XA and YA as latches which are transparent when CK1 = 0. $\overline{XEN}$ enables XA to be loaded, and $\overline{YEN}$ enables YA to be loaded. The operation of the instruction register is identicalto that of XA and YA except that either $\overline{XEN}$ or $\overline{YEN}$ enable instructions to be loaded.

XA, YA and the instruction register on the FMPY are configured as edge-triggered registers. Functionality in this mode is identical to the FALU configured with R/L1 = 1 as described above.

Output data may also pass through a register or a transparent latch on the FALU and FMPY. When R/L2 = 1, Z is configured as an edge-triggered register. Data is loaded on the rising edge of CK2. R/L2 = 0 configures Z as a latch which is transparent when CK2 = 1. $\overline{ZEN}$ enables Z. The enables $\overline{XEN}$, $\overline{YEN}$ and $\overline{ZEN}$ are latched internally. See timing diagrams for more information.

Dynamically changing the state of either R/L1 or R/L2 may disturb the state of internal registers and is therefore not recommended.

CLOCK ENABLES

When R/L1,2 = 1, the clock enables are sampled on the rising edge of CK1 and CK2. When R/L1,2 = 0, the clock enables act as asynchronous control inputs. See the timing diagrams.

DOUBLE PRECISION OPERANDS

Transparent input latches XB and YB are used to store the most significant word of a double precision operand from ports X and Y. The latches are transparent when the MSW/EN clock is HI.

Double precision operand transfers to the FMPY/FALU consist of two steps. First, the most significant word is latched in XB/YB with MSW/EN. The least significant word is then transferred through ports X and Y, concatenated with the contents of latches XB and YB, and then clocked in XA and YA with CK1.

Similarly, double precision operand transfers from the Z register/latch or T port will use XC to latch the most significant word. XC is transparent when MSW/SEL is high. When MSW/SEL is low, the least significant 36 bits of the result is output to the T port and fed back to the X input multiplexer. If XSEL is high, CK1 will clock the fedback operand into the XA register/latch.

The full 64 bit result is clocked into the Z register/latch with CK2. The output multiplexer then selects which half will appear at the T port. When MSW/SEL is high, the most significant 36 bits of the result is output to the T port and is available at X input multiplexer.

SINGLE PRECISION OPERANDS

Single precision operands are clocked directly into registers/latches XA and YA, latches XB/XC and YB are bypassed. The single precision result is always output to port T, regardless of the state of MSW/SEL. Note that when a double precision operand is fed back to the Y input multiplexer (FALU only) and selected for a single precision instruction, only the least significant word will be used in the operation.

Y FEEDBACK PATH

Z result data can be fed back to the Y input multiplexer (FALU only). The input YSEL selects which data path will be used for the Y operand. Note that since the Y feedback path bypasses the YA register/latch, instructions which use Y feedback should only occur when Z is configured as an edge triggered register. This is to prevent an unclocked feedback loop.

If the result of a single precision instruction is fed back along the Y feedback path and used as a double precision operand, the most significant 32-bits are undefined.

MULTICYCLE INSTRUCTIONS

There are several different possible operation times for the chip set. An instruction cycle typically is designed around one instruction type. For example, an instruction cycle could be designed around the floating point add and subtract time of 25ns. A double precison multiply would then become a two-cycle instruction. The clock enables, $\overline{XEN}$ and $\overline{YEN}$, can be used to disable CK1 during the second cycle of the double precision multiply. This ensures that the instruction and operands are valid through the entire two-cycle multiplication operation.

CONTROL AND STATUS REGISTER DESCRIPTION

The FMPY contains three registers which provide control and status information. The FALU contains the same three registers as the FMPY, but also contains an additional register useful for normalization, shift and rotate instructions.

ADSP-8110/ADSP-7110	ADSP-8120/ADSP-7120
Flag Register	Flag Register
Interrupt Enable Register	Interrupt Enable Register
Mode Register	Mode Register
	Shift Count Register

The Flag register holds the state of the device for the most recent instruction. The Interrupt Enable register determines the conditions upon which an interrupt will be generated. The Mode register allows control of operating modes. In addition, the FALU contains a Shift Count register which is used for Rotate and Shift instructions.

Status and control register writes use the X port and reads use the T port. The value output during a write instruction depends on the particular register being referenced and the state of R/L2. The following table describes the possible output values.

REGISTER ACCESSED	VALUE OUTPUT	
	R/L2 = 0	R/L2 = 1
Flag Register	New Contents	Old Contents
Interrupt Enable Register	Old Contents	Old Contents
Mode Register	Old Contents	Old Contents
Shift Count Register	New Contents	New Contents

In the flowthrough output mode (latch Z always transparent), CK2 is held static and the status and control registers will not be updated. This affects several operations which depend on registered data (i.e. shifts, rotates and wrapped to denorm conversion).

Unused/don't care bits in the FMPY are set to logic one. These bits in the FALU are set to logic zero.

FLAG REGISTER

22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B3	B2	B1	B0	PT	PY	PX	dc	dc	CRY	DIVZ	DY	DX	NaN	RND	INX	INV	UF	OV	ZR	N	PE	INT

MSB **LSB**

NOTE: dc is "don't care," these bits are not used

B3, B2, B1, B0 — Parity Error Byte Location flags. When a parity error has been detected at any port, one or more of these bits will be active high to identify which bytes of the 32 bit word contained the error. B3 signifies the most significant byte and B0 signifies the least significant byte.

PT, PY, PX — Parity Error Port Location flags. When a parity error (odd parity is checked) has been detected, one or more of these bits will be active high to signal the ports at which the error has occurred. PT bit identifies the T bidirectional port, PY identifies the Y input port, and PX identifies the X input port.

CRY — Carry flag. Definition applies only to the FALU, don't care for the FMPY. When the borrow bit in the mode register = 1, the carry flag will be asserted when there is a carry out of the accumulator during integer additions or no carry out during integer subtractions. When the borrow mode bit = 0 the carry flag will always indicate a carry out of the ALU during integer operations. During shifts and rotates, the carry flag always contains the last bit to leave the ALU. Floating point arithmetic operations usually reset the carry flag (except: min, max, compare, pass and wrap to denorm).

DIVZ — Divide by zero flag. Definition applies only to the FMPY. This bit is set for divides when a finite non-zero number is divided by zero.

DY, DX — Denormalized input flags. During floating point operations, one or both of these bits will be set if a denormalized number was received on either input port. DY signals a denormalized Y input, and DX flags a denormalized X input. The bits are set without regard to mode register bits <0,3>.

FLAG REGISTER (cont'd)

NaN Not a Number flag. Only valid for floating point operations, this bit indicates that an operand received on one of the two input ports or the result output was not a number. In IEEE mode, a signaling NaN input causes both the NaN flag and the Invalid operation flag to be set. A quiet NaN causes only the NaN flag to be set. In both cases, the output will be a quiet NaN, and the "value" of the NaN is always as follows:

IEEE Single NaN: sign = 0 or 1
exponent = FF hex
fraction = 20 0000 hex
(bit 22 = 0, bit 21 = 1)

IEEE Double NaN: sign = 0 or 1
exponent = 7FF hex
fraction = 4 0000 0000 0000 hex
(bit 51 = 0, bit 50 = 1)

The sign of NaN's follows arithmetic conventions. For example, – NaN multiplied by + NaN = – NaN. During square root operations, the sign of a NaN is the sign of the X input.

In DEC mode, only reserved operands will set both the NaN and INV flags. A reserved operand is output, and the "value" of the output will be " – 0." See DEC format section.

RND Rounded Up output flag. This bit will be set whenever the normalized or wrapped output has been rounded away from zero.

INX Inexact Result flag. This bit will be set whenever the output is not infinitely precise.

INV Invalid Operation flag. This bit is set when an input operand is invalid for the requested operation. In IEEE mode, the following conditions cause an Invalid Operation flag:

1) A signaling NaN on either input
2) Magnitude subtraction of infinities, i.e. (+ INF) + (– INF)
3) Zero multiplied with infinity
4) Zero divided by zero or infinity divided by infinity
5) Square root of a negative number
6) Conversion of a floating point number to an integer format when the operand overflows the integer format or is not representatable

In DEC mode, INV will be active for the following conditions:

1) Reserved operand on either input
2) Zero/zero
3) Square root of a negative number
4) Case (6) above

UF Underflow flag. This flag is set if the result of an operation is less than the minimum representable normalized number in the chosen format.

OV Overflow flag. This bit is set if the result of an operation is larger than the maximum representable normalized number in the chosen format.

ZR Zero flag. This bit is set if the integer result or the normalized floating point result is zero.

N Negative flag. This bit is set if the most significant bit of the result is set. Note that in IEEE mode, negative zero will set this flag.

PE Parity Error flag. If a parity error has been detected at register/latches XA or YA, this bit will be set. Parity errors will not be detected during flag register writes.
Odd parity is checked during every operation except flag register writes. Parity is also checked during operations on other chips.
A parity error will occur if the parity bits plus the data word contain an even number of ones. An input of all zeros is a parity error.

INT Interrupt flag. This bit mirrors the hardware interrupt output. It will be set if one of the above conditions are true, the corresponding interrupt enable bit is set, and the IE bit is set in the Interrupt Enable register.

Access to the flag register is obtained by using the FREGx instructions. The flag port is a reflection of bits in the flag register.

The seven parity error status flags (bits <22..16>) are always "sticky." Once set they remain set until written as 0, the CLRFLAG instruction is executed, or the hardware reset is asserted. The parity error flag (PE) is sticky if mode register bit eight (SP) is set.

Every arithmetic operation updates the flag register (unless the flag register is frozen). With the exception of the parity bits, each update will clear the existing bits and set them according to the current operation. Once frozen, the bits will be cleared by the CLRFLAG instruction, asserting the hardware reset pin, or writing zeroes to the flag register.

In the flowthrough output mode (latch Z always transparent), CK2 is held static and the status and control registers will not be updated. This affects several operations which depend on register data (i.e. shifts, rotates and wrapped to denorm conversion).

INTERRUPT ENABLE REGISTER

13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRY	DIVZ	dc	DEN	NaN	dc	INX	INV	UF	OV	ZR	N	PE	IE

MSB **LSB**

NOTE: dc is "don't care," these bits have no effect
1 = enable, 0 = disable

CRY Carry flag interrupt enable. Used only by the FALU, enables an interrupt to occur if the CRY flag is asserted. Don't care in the FMPY.

DIVZ Divide by zero flag interrupt enable. Used only by the FMPY, enables an interrupt to occur if the DIVZ flag is asserted. Don't care in the FALU.

DEN Denormalized number interrupt enable. Allows an interrupt if either of the denormalized number flags (DX, DY) are asserted.

NaN Not a Number flag interrupt enable.

INX Inexact result flag interrupt enable. In IEEE mode, allows an interrupt whenever the INX flag is asserted.

INV Invalid operation flag interrupt enable.

UF Underflow flag interrupt enable.

OV Overflow flag interrupt enable.

ZR Zero flag interrupt enable.

N Negative interrupt enable.

PE Parity Error interrupt enable.

IE Master Interrupt enable. Must be set to a logic one for an interrupt to be generated for any of the above conditions.

The interrupt enable register allows a user to determine which condition(s) will activate an interrupt. An interrupt will occur if one or more of the above conditions is true, at least one of the corresponding enable bits is set, and the master interrupt enable bit is set.

The interrupt enable bits will not prevent the flags from being set.

Access to the interrupt enable register is accomplished using the IREGx instructions.

MODE REGISTER

13-9	8	7	6	5	4	3	2	1	0
Reserved	SP	BM	R1	R0	IO	IU	IP	FF	ID

MSB **LSB**

1 = enabled, 0 = disabled

SP Sticky Parity Flag mode. When SP = 1, the parity error flag (PE, bit<1> of the flag register) is sticky, once set it will remain set. When sticky, the parity flag can be reset by asserting the reset pin, executing the CLRFLAG instruction, or writing a zero to the parity error flag bit. This mode bit has no effect on bits <16..22> of the flag register.

BM Borrow mode. Only applies to the FALU, don't care for FMPY. When BM = 0, "normal" carry mode is used, i.e. the carry flag is set whenever there is a carry out of the FALU. When BM = 1, "DEC" carry mode is used, i.e. carry flag is set if a carry out of the FALU occurs during addition and if there is no carry (borrow) during subtraction.

R1, R0 Rounding mode. When in DEC mode R1, R0 should be set to 0,0. When in IEEE mode the rounding operation is determined by the following chart.

R1	R0	ROUNDING MODE
0	0	Round to nearest
0	1	Round to zero
1	0	Round to − infinity
1	1	Round to + infinity

IO IEEE Overflow mode. When in IEEE mode, (ID = 0), IO = 1 causes overflows to be returned as wrapped numbers. If IO = 0, then overflows will be set to either infinity or the largest finite number, according to section 7.3 of the IEEE standard 745. When the ID = 1 (DEC mode), this bit should be zero.

IU IEEE Underflow mode. When in IEEE mode (ID = 0), IU = 1 causes underflows to be returned as wrapped numbers. If IU = 0, then underflows will be set to a properly signed zero and denormalized numbers will be flushed to zero. When the ID = 1 (DEC mode), this bit should be zero.

IP Ignore Parity. When IP = 0, all 8 parity error flags (B3, B2, B1, B0, PT, PY, PX and PE) will not be updated. When IP = 1, the 8 parity flags can be updated.

FF Freeze Flags on interrupt mode. When FF is asserted, the bits in the flag register will "freeze" (i.e. remain in their current state) once an interrupt has been generated. The flags will remain frozen until a CLRFLAG instruction is executed, the hardware reset pin is asserted, zeroes are written to the flag register or interrupt enable register, or the FF bit is cleared.
If R/L2 = 0, the status register flags may only be "frozen" on the falling edge of CK2. This bit does nothing in "flowthrough" mode, i.e. output latches always open.
The flag pins reflect the flag register and will be frozen along with the flag register.

MODE REGISTER (cont'd)

ID — IEEE/DEC mode. When this bit is asserted, the FMPY and FALU operate in DEC (F or G format) mode. When de-asserted, IEEE mode is used.

Reserved — During mode register write operations, zeros must be written to the reserved bits to retain compatibility with future versions of the floating point chip set.

Access to the Mode Register is accomplished through the use of the MREGx instructions. If the IO, IU or ID bits are modified, the following cycle must be extended 3ns.

SHIFT COUNT REGISTER

The Shift Count register contains a 7 bit twos complement number which indicates the count and direction for a shift or rotate instruction. Accessing this register is accomplished using the SCREGx instructions which may either write bits <6..0> of the X port or read these same bits (sign extended to 32 bits) from the T port. Only the FALU contains this register.

If the shift count register contains a value greater than zero for a shift or rotate instruction, a shift/rotate left of SC bits will be performed. If the value is zero, no shift or rotate will occur and if the value is less than zero a right shift/rotate of − SC will be performed.

■ FIXED POINT FORMATS (INTEGER)

32 bit twos complement fixed point format.

MSB **LSB**

i_{31}	i_{30}	i_{29}	. . . . .	i_2	i_1	i_0

Sign Bit

64 bit twos complement fixed point format.

LSB

i_{31}	i_{30}	i_{29}	. . . . .	i_2	i_1	i_0

Least Significant Word

MSB

i_{63}	i_{62}	i_{61}	. . . . .	i_{34}	i_{33}	i_{32}

Sign Bit Most Significant Word

■ IEEE FLOATING POINT FORMAT

IEEE Standard 754-1985 binary floating point arithmetic single and double precision basic formats are supported by the floating point multiplier and ALU. Extended formats are NOT supported. The floating point data word is made up of three parts: sign bit, biased exponent and fraction. See the IEEE std 754 for additional information.

	MSB		LSB
Format	s	e	f
Single	1	8	23
Double	(1)	(11)	(52)

Where s = sign bit
e = biased exponent
f = fraction

The value of the floating point word is determined by the following tables.

Single Precision

Sign(s)	Exponent (e)	Fraction (f)	Interpretation
0/1	255	= 0 msb = 1	S
		msb = 0	Q
0	255	0	+ INF
1	255	0	− INF
0/1	1-254	f	$(-1)^s \bullet 1.f \bullet 2^{e-127}$
0/1	0	= 0	DEN
0	0	0	+ 0
1	0	0	− 0

Double Precision

Sign(s)	Exponent (e)	Fraction (f)	Interpretation
0/1	2047	= 0 msb = 1	S
		msb = 0	Q
0	2047	0	+ INF
1	2047	0	− INF
0/1	1-2046	f	$(-1)^s \bullet 1.f \bullet 2^{e-1023}$
0/1	0	= 0	DEN
0	0	0	+ 0
1	0	0	− 0

ROUNDING

The FMPY and FALU support all four IEEE-754 rounding modes. The rounding process will take a number and, if necessary, modify it to fit in the destinations format. The destination format can be single/double precision floating point or single/double precision fixed point.

Round to Nearest

This mode will round the infinitely precise result to the nearest representable value that fits in the destination format. Results that are halfway in between two representable values will be rounded towards the even result (result with LSB = 0 is delivered). This rounding mode is statistically unbiased because over a large number of random numbers half will be rounded up and half rounded down.

Round toward Zero

This mode will round the result to the closest representation whose magnitude is less than or equal to the infinitely precise result. Round to zero truncates all bits less significant than the destination fractions LSB.

Round toward Plus Infinity

This mode will round the result to the closest representation, which is no less than the infinitely precise result. If the prerounded result is greater than the maximum representable normalized number, the result is rounded to plus infinity and the overflow flag is set.

Round toward Minus Infinity

This mode will round the result to the closest representation, which is no greater than the infinitely precise result. If the prerounded result is less than the minimum representable number, the result is rounded to minus infinity and the overflow flag is set.

OVERFLOWS AND UNDERFLOWS (WRAPPED MODE DISABLED)

The result of an operation which overflows or underflows when the wrapped mode is disabled, depends on the sign of the result and the rounding mode. The tables below illustrate the possible results.

Overflows

ROUNDING MODE	– OV	+ OV
Round to nearest	– INF	+ INF
Round to zero	– M	+ M
Round to – infinity	– INF	+ M
Round to + infinity	– M	+ INF

Where M is the largest normalized number.

Underflows

ROUNDING MODE	– UF	+ UF
Round to nearest	–0	+0
Round to zero	–0	+0
Round to – infinity	– E	+0
Round to + infinity	–0	+ E

Where E is the smallest normalized number.

■ DEC (VAX) FLOATING POINT FORMAT

The floating point ALU and Multiplier support the DEC F and G floating point formats. The DEC D and H formats are not supported. DEC floating point arithmetic is very similar to IEEE 754 arithmetic, but does not contain all of the special cases and operands that are defined in the IEEE specification. The F format corresponds to single precision IEEE, while the G format corresponds to double precision. For complete information on DEC format floating point arithmetic, see the VAX Architecture Handbook from Digital Equipment Corporation.

	MSB		LSB
Format	s	e	f
F	1	8	23
G	(1)	(11)	(52)

Where s = sign bit
e = biased exponent
f = fraction

The value of the floating point word is determined by the following tables.

F Format

Sign	Exponent	Fraction	Interpretation
0	0	0	0
0	0	= 0	"dirty" zero
1	0	any	R
0/1	1-255	any	$(-1)^s \cdot 0.1f \cdot 2^{e-128}$

DEC (VAX) FLOATING POINT FORMAT (cont'd)

G Format

Sign	Exponent	Fraction	Interpretation
0	0	0	0
0	0	= 0	"dirty" zero
1	0	any	R
0/1	1-2047	any	$(-1)^S \bullet 0.1f \bullet 2^{e-1024}$

The DEC F and G format sign, exponent, and fraction field widths are the same as their IEEE format counterparts, and both IEEE and DEC formats use a "hidden" bit to increase the resolution of their mantissas. The DEC "hidden" bit, however, is to the right of the binary point, while the IEEE "hidden" bit is to the left. Furthermore, the exponent biases of the two standards differ, leading to different representable number ranges.

Normalized Number Range

	Minimum	Maximum
IEEE Single	2^{-126}	$2^{127} \bullet (2 - 2^{-23})$
DEC F	2^{-128}	$2^{126} \bullet (2 - 2^{-23})$
IEEE Double	2^{-1022}	$2^{1023} \bullet (2 - 2^{-52})$
DEC G	2^{-1024}	$2^{1022} \bullet (2 - 2^{-52})$

Another difference is that the DEC formats lack denormalized numbers and do not have separate representations for positive and negative zero. A number with a sign of zero, an exponent of zero, and a nonzero mantissa is considered to be a "dirty" zero. On input, "dirty" zeroes are treated exactly the same as the normal zero (except that "dirty" zeroes cause the denormalized input flag to be raised), but they are never returned as a result.

DEC reserved operands are similar to IEEE signaling NaNs. When they are used as an operand, the invalid operation flag is raised and a reserved operand is returned as the result. Reserved operands are also output whenever an invalid operation (such as division by zero) or overflow occurs. DEC specifies that when a reserved operand is encountered, the destination register should not be changed. It is up to the user to ensure that this happens. Underflows raise the underflow flag and return a result of zero.

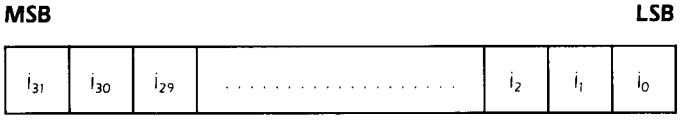
ROUNDING

DEC format arithmetic always rounds the infinitely precise result in the same way; there is no choice of rounding modes. The infinitely precise result is rounded to the nearest representable number. If two representable numbers are equally close to the infinitely precise result, then the one with the larger magnitude is chosen. This is thus slightly different from the IEEE round to nearest mode.

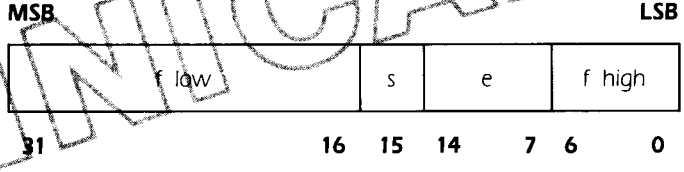
WORD ORDER

The VAX uses different word ordering for integer and floating point numbers. This ordering is NOT supported by the FALU and the FMPY. The distinction is important if the same data path is to be used for VAX compatible integer and floating point operations and conversions. The word order of either the floating point or the integer operands will have to be explicitly swapped before the operands are stored in memory. Single precision numbers can be rotated by 16 bit positions. Double precision numbers need their high and low words swapped and each rotated by 16 bit positions.

VAX 32 bit integer:

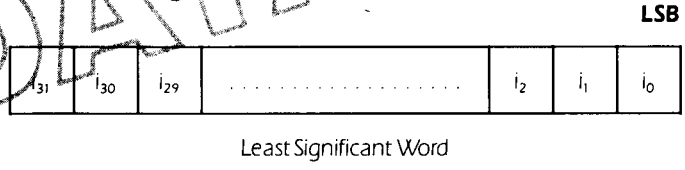


VAX F floating:

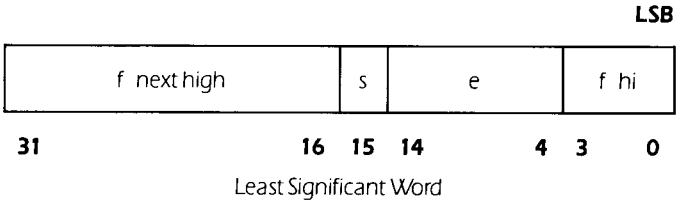


ADI F floating:

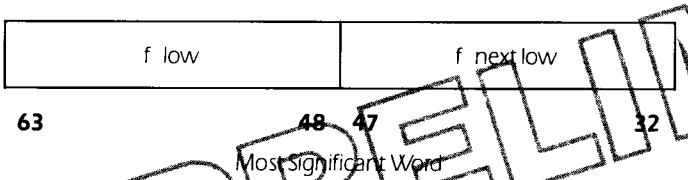
VAX 64 bit integer:



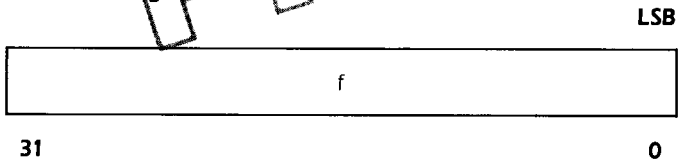
VAX G floating:



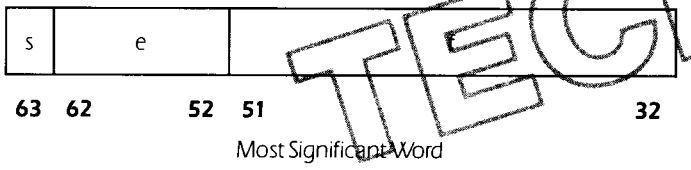
MSB



ADI G floating:



MSB



MODE REGISTER

To implement DEC format arithmetic the following mode register bits must be set/reset.

Bit	Value	Name
0	1	IEEE/DEC Format
3	0	IEEE Underflow mode
4	0	IEEE Overflow mode
5	0	IEEE Rounding mode
6	0	IEEE Rounding mode
7	1	Borrow mode (Borrow mode is only required if integer arithmetic will be used)

■SCAN PATH ORDER

In Scan mode, data is transferred serially into the internal registers through SIN and out of the registers through SOUT on rising edges of SCK. The scan path operates as a first-in-first-out serial shift. The following diagram illustrates the order in which data is transferred.

FMULT:

SCANIN →
$Z_{63} \rightarrow Z_{31} \ldots Z_{32} \rightarrow Z_0$
$X_{32} \rightarrow X_0 \ldots X_{39} \rightarrow X_7$
$XP_4 \rightarrow XP_0$
$X_{40} \rightarrow X_8 \ldots X_{47} \rightarrow X_{15}$
$XP_5 \rightarrow XP_1$
$X_{48} \rightarrow X_{16} \ldots X_{55} \rightarrow X_{23}$
$XP_6 \rightarrow XP_2$
$X_{56} \rightarrow X_{24} \ldots X_{63} \rightarrow X_{31}$
$XP_7 \rightarrow XP_3$
$Y_{32} \rightarrow Y_0 \ldots Y_{39} \rightarrow Y_7$
$YP_4 \rightarrow YP_0$
$Y_{40} \rightarrow Y_8 \ldots Y_{47} \rightarrow Y_{15}$
$YP_5 \rightarrow YP_1$
$Y_{48} \rightarrow Y_{16} \ldots Y_{55} \rightarrow Y_{23}$
$YP_6 \rightarrow YP_2$
$Y_{56} \rightarrow Y_{24} \ldots Y_{63} \rightarrow Y_{31}$
$YP_7 \rightarrow YP_3$
$I_0 \rightarrow I_7$
$MODE_0 \rightarrow MODE_6$
$MODE_8$
$INTEN_0 \rightarrow INTEN_7$
$INTEN_9$
$INTEN_{10}$
$INTEN_{12}$
$FLAG_1 \rightarrow FLAG_7$
$FLAG_9 \rightarrow FLAG_{12}$
$FLAG_{14} \rightarrow FLAG_{32}$
$FLAG_8$
→ SCANOUT

FALU:

SCANIN →
$Z_{63} \rightarrow Z_{31} \ldots Z_{32} \rightarrow Z_0$
$X_{32} \rightarrow X_0 \ldots X_{39} \rightarrow X_7$
$XP_4 \rightarrow XP_0$
$X_{40} \rightarrow X_8 \ldots X_{47} \rightarrow X_{15}$
$XP_5 \rightarrow XP_1$
$X_{48} \rightarrow X_{16} \ldots X_{55} \rightarrow X_{23}$
$XP_6 \rightarrow XP_2$
$X_{56} \rightarrow X_{24} \ldots X_{63} \rightarrow X_{31}$
$XP_7 \rightarrow XP_3$
YSEL
$Y_{32} \rightarrow Y_0 \ldots Y_{39} \rightarrow Y_7$
$YP_4 \rightarrow YP_0$
$Y_{40} \rightarrow Y_8 \ldots Y_{47} \rightarrow Y_{15}$
$YP_5 \rightarrow YP_1$
$Y_{48} \rightarrow Y_{16} \ldots Y_{55} \rightarrow Y_{23}$
$YP_6 \rightarrow YP_2$
$Y_{56} \rightarrow Y_{24} \ldots Y_{63} \rightarrow Y_{31}$
$YP_7 \rightarrow YP_3$
$I_0 \rightarrow I_7$
$MODE_0 \rightarrow MODE_8$
$INTEN_0 \rightarrow INTEN_7$
$INTEN_9$
$INTEN_{10}$
$INTEN_{13}$
$SC_0 \rightarrow SC_6$
$FLAG_1 \rightarrow FLAG_7$
$FLAG_9 \rightarrow FLAG_{11}$
$FLAG_{13}$
$FLAG_{16} \rightarrow FLAG_{22}$
$FLAG_8$
→ SCANOUT

NOTE: $Z_{63} \rightarrow Z_{31} \ldots Z_{32} \rightarrow Z_0$ indicates the pattern $Z_{63} \rightarrow Z_{31} \rightarrow Z_{62} \rightarrow Z_{30} \ldots Z_{32} \rightarrow Z_0$

NOTE: There is a separate parity bit (total of 8 parity bits) for each byte in registers XA, YA and Z.

NOTE: $FLAG_8$ (RND) is out of sequence in the scanpath because the RND pin has been chosen to serve dual purpose as the SCANOUT pin.

■ SPECIFICATIONS (ADSP-7110/ADSP-7120)

Absolute Maximum Ratings

Permanent damage may occur if any one absolute maximum rating is exceeded. Functional operation is not implied and device reliability may be impaired by exposure to higher than recommended voltages for extended periods of time.

Parameter	Symbol	Value	Units
Supply Voltage (GND = 0)	V_{cc}	−0.5 to +7.0	V_{dc}
Input Voltage (GND = 0)	V_{in}	−0.5 to V_{cc} +0.5	V_{dc}
Output Source Current			
Continuous	I_o	30	mA_{dc}
Surge	I_o	100	mA_{dc}
Storage Temperature	T_{st}	−55 to 150	°C
Junction Temperature	T_j	165	°C

Recommended Operating and Test Conditions

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Supply Voltage (GND = 0)	V_{cc}	4.75	5.00	5.25	V_{dc}
Ambient Temperature	T_a	0		70	°C
Junction Temperature	T_{jt}			125	°C

**500 linear feet per minute ambient airflow.*

DC Characteristics

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Supply Current (V_{cc} = Max)					
ADSP-7110L	I_{cc}				mA_{dc}
ADSP-7110J & K	I_{cc}			2.72	mA_{dc}
ADSP-7120K	I_{cc}				mA_{dc}
ADSP-7120J	I_{cc}			2.21	mA_{dc}
Output Current High	I_{oh}			−0.4	mA_{dc}
Output Current Low	I_{ol}			4.0	mA_{dc}
High Impedance Output Current (V_{cc} = Max, V_o = 2.4)	I_{ozh}			40	uA_{dc}
High Impedance Output Current (V_{cc} = Max, V_o = 0.4)	I_{ozl}			−40	uA_{dc}
Input Current High (V_{cc} = Max, V_{ih} = 2.4)	I_{ih}			200	uA_{dc}
Input Current Low (V_{cc} = Max, V_{il} = 0.4)	I_{il}			−200	uA_{dc}
Output Voltage High (V_{cc} = Min, I_{oh} = Max)	V_{oh}	2.4			V_{dc}
Output Voltage Low (V_{cc} = Min, I_{ol} = Max)	V_{ol}			0.4	V_{dc}
Input Voltage High	V_{ih}	2.0			V_{dc}
Input Voltage Low	V_{il}			0.8	V_{dc}

■ SPECIFICATIONS (ADSP-8110/ADSP-8120)

Absolute Maximum Ratings

Permanent damage may occur if any one absolute maximum rating is exceeded. Functional operation is not implied and device reliability may be impaired by exposure to higher than recommended voltages for extended periods of time.

Parameter	Symbol	Value	Units
Supply Voltage (V_{cc} = 0)	V_{ee}	−8.0 to 0	V_{dc}
Input Voltage (V_{cc} = 0)	V_{in}	V_{ee} to 0	V_{dc}
Output Source Current			
Continuous	I_o	30	mA_{dc}
Surge	I_o	100	mA_{dc}
Storage Temperature	T_{st}	−55 to 150	°C
Junction Temperature	T_j	165	°C

Recommended Operating and Test Conditions

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Supply Voltage (V_{cc} = 0)	V_{ee}	−5.46	−5.20	−4.94	V_{dc}
Ambient Temperature	T_a*	0		70	°C
Junction Temperature	T_{jt}			125	°C
Output Termination to −2.0 V_{dc}	R_1		50		Ohms

**500 linear feet per minute ambient airflow.*

DC Characteristics

Parameter	Symbol	Value						Units
		0°C		25°C		70°C		
		Min	Max	Min	Max	Min	Max	
Input Voltage High	V_{ih}	−1.17	−0.84	−1.13	−0.81	−1.07	−0.735	V_{dc}
Input Voltage Low	V_{il}	−1.95	−1.48	−1.95	−1.48	−1.95	−1.45	V_{dc}
Output Voltage High (Terminated)	V_{oh}	−1.02		−0.98		−0.92		V_{dc}
Output Voltage Low (Terminated)	V_{ol}		−1.63		−1.63		−1.60	V_{dc}

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Supply Current (V_{ee} = Min)					
ADSP-8110L	I_{ee}				mA_{dc}
ADSP-8110J & K	I_{ee}			2.75	mA_{dc}
ADSP-8120K	I_{ee}				mA_{dc}
ADSP-8120J	I_{ee}			2.20	mA_{dc}
Input Current High (V_{ih} = Max)	I_{ih}			300	uA_{dc}

■ SWITCHING CHARACTERISTICS (ADSP-8110/ADSP-8120 and ADSP-7110/ADSP-7120)

The operation time of either the FMPY or FALU is a function of the instruction being executed and which switching characteristic is being referenced. Operation times can be calculated by adding the instruction execution time found in Table I for the operation being performed to the delay for the particular switching characteristic of interest. In addition, a delay t_{rd} is included for switching characteristics that, in part, include output delay. This additional delay is only included for the ADSP-7110 or ADSP-7120.

Table I

Operation	Instruction[1] Execution Time OP_{max}			Units
	ADSP-7110L ADSP-8110L	ADSP-7110K ADSP-8110K	ADSP-7110J ADSP-8110J	
FMPY				
Single precision floating point multiply	40	50	60	ns
Double precision floating point multiply	50	60	70	ns
32 x 32 integer multiply	45	55	65	ns
Single precision floating point divide	200	200	200	ns
Double precision floating point divide	300	300	300	ns
Single precision floating point square root	300	300	300	ns
Double precision floating point square root	600	600	600	ns
Single precision pass	40	50	60	ns
Double precision pass	50	60	70	ns
Register operations	40	50	60	ns

Table II

Operation	t_{yls} min	t_{ylh} min	Instruction[1] Execution Time OP_{max}		Units
			ADSP-7120K ADSP-8120K	ADSP-7120J ADSP-8120J	
FALU					
All floating point operations			25	30	ns
All conversions			25	30	ns
All integer operations			12	15	ns
Register operations			25	30	ns

t_{yls} *YSEL latch mode setup time.*
t_{ylh} *YSEL latch mode hold time.*

NOTE[1]: If the IO, IU, or ID bits of the mode register are modified, the following cycle must be extended 6ns to allow these bits to propagate throughout the part.

Specifications for K Grades of Multiplier and J Grades of Multiplier and ALU

(t_{tdmax} = 11ns, t_{tdmin} = 0.5ns)

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Setup and Hold Times					
Input Register Setup Time	t_{rs}	1.5			ns
Input Register Hold Time	t_{rh}				
TTL versions		3.5			ns
ECL versions		3.0			ns
Input Latch Setup Time	t_{ls}	2.0			ns
Input Latch Hold Time	t_{lh}	2.0			ns
Latch Enable Setup Time	t_{les}	2.0			ns
Latch Enable Hold Time	t_{leh}	2.0			ns
MSWEN/MSWSEL to CK1 Hold Time	t_{cdh}	0			ns
MSWEN/MSWSEL Setup Time	t_{ms}	2.5			ns
MSWEN/MSWSEL Hold Time	t_{mh}	2.5			ns
X,Y Register Enable Setup Time	t_{xyrs}	1.5			ns
X,Y Register Enable Hold Time	t_{xyrh}	2.0			ns
Z Register Enable Setup Time	t_{zrs}	1.5			ns
Z Register Enable Hold Time	t_{zrh}	4.0			ns
Register to Register					
Register to Register OP Time	t_{rr}			OP	ns
Register to Register Hold Time	t_{rrh}	0			ns
Register Output Delay Time	t_{rod}	$3 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register Interrupt Output Delay Time	t_{irod}	$3 + t_{tdmin}$		$11 + t_{tdmax}$	ns
Register Parity Output Delay Time	t_{prod}	$3 + t_{tdmin}$		$12 + t_{tdmax}$	ns
Register Flag Output Delay Time	t_{frod}	$3 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register to Latch					
Register to Latch OP Time	t_{rl}			3 + OP	ns
Register to Latch Hold Time	t_{rlh}	0			ns
Latch Output Delay Time	t_{lod}	$3 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register to Data OP Time	t_{rd}	$3 + t_{tdmin}$		$6 + OP + t_{tdmax}$	ns
Latch Interrupt Output Delay Time	t_{ilod}	$3 + t_{tdmin}$		$11 + t_{tdmax}$	ns
Register to Interrupt OP Time	t_{ird}	$3 + t_{tdmin}$		$8 + OP + t_{tdmax}$	ns
Latch Parity Output Delay Time	t_{plod}	$3 + t_{tdmin}$		$12 + t_{tdmax}$	ns
Register to Output Parity OP Time	t_{rp}	$3 + t_{tdmin}$		$9 + OP + t_{tdmax}$	ns
Latch Flag Output Delay Time	t_{flod}	$3 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register to Flag OP Time	t_{rf}	$3 + t_{tdmin}$		$6 + OP + t_{tdmax}$	ns

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Latch to Register					
Data to Register OP Time	t_{dr}			OP	ns
Latch to Register Hold Time	t_{lrh}	0			ns
Latch to Register OP Time	t_{lr}			OP	ns
Register Output Delay Time	t_{rod}	$3+t_{tdmin}$		$9+t_{tdmax}$	ns
Register Interrupt Output Delay Time	t_{irod}	$3+t_{tdmin}$		$11+t_{tdmax}$	ns
Register Parity Output Delay Time	t_{prod}	$3+t_{tdmin}$		$12+t_{tdmax}$	ns
Register Flag Output Delay Time	t_{frod}	$3+t_{tdmin}$		$9+t_{tdmax}$	ns
Latch to Latch					
Data to Latch OP Time	t_{dl}			3 + OP	ns
Latch to Latch Hold Time	t_{llh}	0			ns
Latch to Latch OP Time	t_{ll}			3 + OP	ns
Data to Data OP Time	t_{dd}	$3+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Latch Output Delay Time	t_{od}	$3+t_{tdmin}$		$9+t_{tdmax}$	ns
Latch to Data OP Time	t_{ld}	$3+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Latch Interrupt Output Delay Time	t_{ilod}	$3+t_{tdmin}$		$11+t_{tdmax}$	ns
Data to Interrupt OP Time	t_{idd}	$3+t_{tdmin}$		$8+OP+t_{tdmax}$	ns
Latch to Interrupt OP Time	t_{ild}	$3+t_{tdmin}$		$8+OP+t_{tdmax}$	ns
Latch Parity Output Delay Time	t_{plod}	$3+t_{tdmin}$		$12+t_{tdmax}$	ns
Data to Parity OP Time	t_{dp}	$3+t_{tdmin}$		$9+OP+t_{tdmax}$	ns
Latch to Parity OP Time	t_{lp}	$3+t_{tdmin}$		$9+OP+t_{tdmax}$	ns
Latch Flag Output Delay Time	t_{flod}	$3+t_{tdmin}$		$9+t_{tdmax}$	ns
Data to Flag OP Time	t_{df}	$3+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Latch to Flag OP Time	t_{lf}	$3+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Clock and Enable Pulse Width					
Clock and Latch Pulse Duration					
High FALU	t_{ch}	4.0			ns
High FMPY	t_{ch}	10.0			ns
Low	t_{cl}	4.0			ns
Enable Pulse Duration					
High	t_{enh}	4.0			ns
Low	t_{enl}	4.0			ns

Parameter	Symbol	Value			Units
		Min	Nom	Max	
MSWSEL Output Delay					
Multiplexer Output Delay	t_{mod}	$3+t_{tdmin}$		$8+t_{tdmax}$	ns
Reset					
Reset to Status and Control Register Write Set Up Time	t_{rst}	16.0			ns
Reset Pulse Duration High	t_{rsh}	4.0			ns
Output Enables B3110/B3120					
Synchronous Output Enable Delay	t_{seno}	—		8	ns
Synchronous Output Disable Delay	t_{sdao}	—		8	ns
Output Enable Delay	t_{eno}	—		7	ns
Output Disable Delay	t_{dao}	—		7	ns
Output Enables B2110/B2120					
Synchronous Output Disable Delay					
High State to High Z	t_{sphz}	—		16	ns
Low State to High Z	t_{splz}	—		21	ns
Synchronous Output Enable Delay					
High Z to High State	t_{spzh}	—		13	ns
High Z to Low State	t_{spzl}	—		16	ns
Output Disable Delay					
High State to High Z	t_{phz}	—		15	ns
Low State to High Z	t_{plz}	—		20	ns
Output Enable Delay					
High Z to High State	t_{pzh}	—		12	ns
High Z to Low State	t_{pzl}	—		15	ns
Smode					
Smode Setup Time	t_{ss}	—			ns
Smode Hold Time	t_{sh}	—			ns
Sout Output Delay Time	t_{sod}	—		—	ns

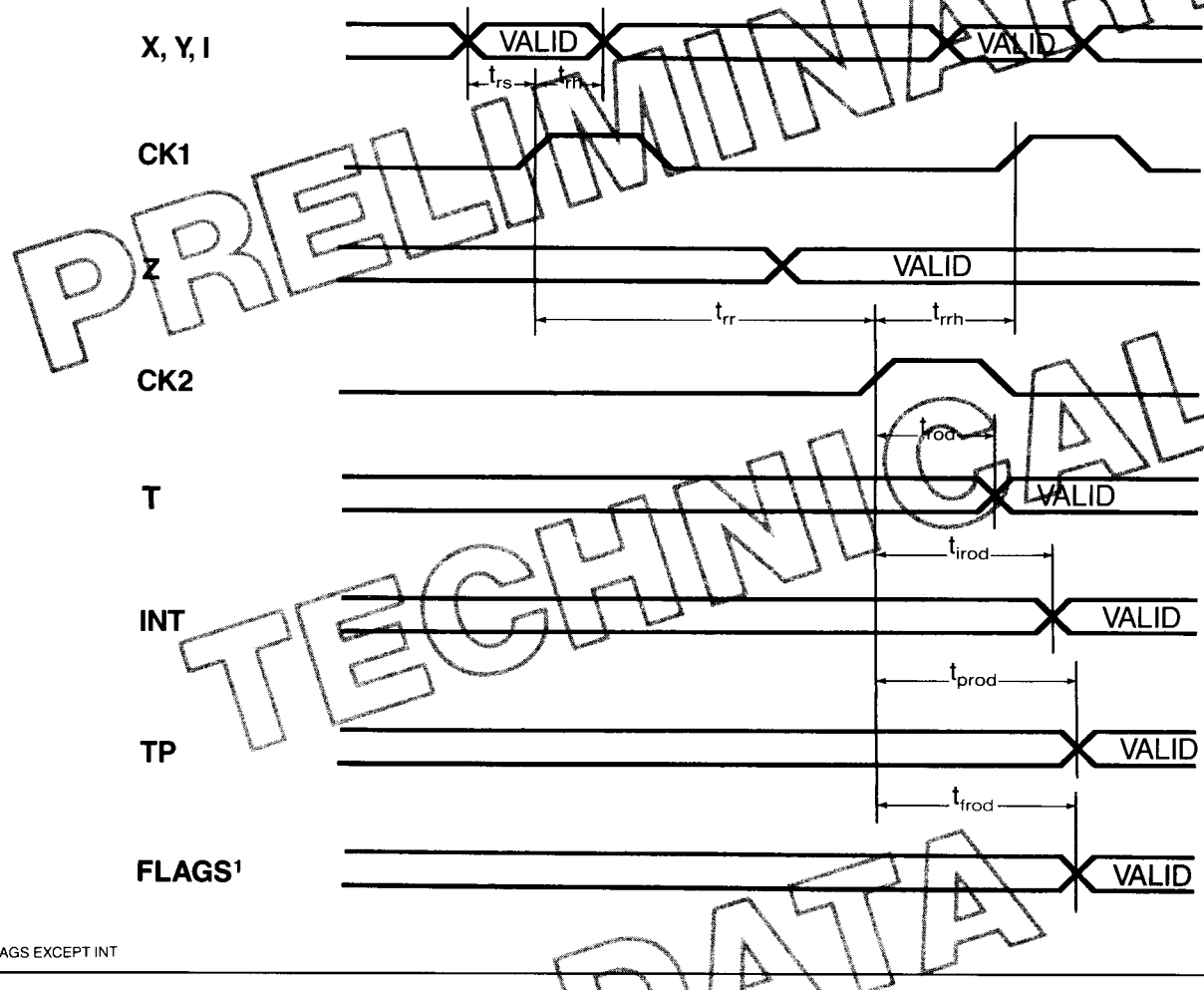
Specifications for L Grade Multipliers, K Grade ALU (t_{tdmax} = 6ns, t_{tdmin} = 1ns)

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Setup and Hold Times					
Input Register Setup Time	t_{rs}	1.5			ns
Input Register Hold Time	t_{rh}				
TTL versions		3.0			ns
ECL versions		2.0			ns
Input Latch Setup Time	t_{ls}	2.0			ns
Input Latch Hold Time	t_{lh}	2.0			ns
Latch Enable Setup Time	t_{les}	2.0			ns
Latch Enable Hold Time	t_{leh}	2.0			ns
MSWEN/MSWSEL to CK1 Hold Time	t_{cdh}	0			ns
MSWEN/MSWSEL Setup Time	t_{ms}	2.5			ns
MSWEN/MSWSEL Hold Time	t_{mh}	2.5			ns
X,Y Register Enable Setup Time	t_{xyrs}	1.5			ns
X,Y Register Enable Hold Time	t_{xyrh}	2.0			ns
Z Register Enable Setup Time	t_{zrs}	1.5			ns
Z Register Enable Hold Time	t_{zrh}	4.0			ns
Register to Register					
Register to Register OP Time	t_{rr}			OP	ns
Register to Register Hold Time	t_{rrh}	0			ns
Register Output Delay Time	t_{rod}	$2 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register Interrupt Output Delay Time	t_{irod}	$2 + t_{tdmin}$		$11 + t_{tdmax}$	ns
Register Parity Output Delay Time	t_{prod}	$2 + t_{tdmin}$		$12 + t_{tdmax}$	ns
Register Flag Output Delay Time	t_{frod}	$2 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register to Latch					
Register to Latch OP Time	t_{rl}			3 + OP	ns
Register to Latch Hold Time	t_{rlh}	0			ns
Latch Output Delay Time	t_{lod}	$2 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register to Data OP Time	t_{rd}	$2 + t_{tdmin}$		$6 + OP + t_{tdmax}$	ns
Latch Interrupt Output Delay Time	t_{ilod}	$2 + t_{tdmin}$		$11 + t_{tdmax}$	ns
Register to Interrupt OP Time	t_{ird}	$2 + t_{tdmin}$		$8 + OP + t_{tdmax}$	ns
Latch Parity Output Delay Time	t_{plod}	$2 + t_{tdmin}$		$12 + t_{tdmax}$	ns
Register to Output Parity OP Time	t_{rp}	$2 + t_{tdmin}$		$9 + OP + t_{tdmax}$	ns
Latch Flag Output Delay Time	t_{flod}	$2 + t_{tdmin}$		$9 + t_{tdmax}$	ns
Register to Flag OP Time	t_{rf}	$2 + t_{tdmin}$		$6 + OP + t_{tdmax}$	ns

Parameter	Symbol	Value			Units
		Min	Nom	Max	
Latch to Register					
Data to Register OP Time	t_{dr}			OP	ns
Latch to Register Hold Time	t_{lrh}	0			ns
Latch to Register OP Time	t_{lr}			OP	ns
Register Output Delay Time	t_{rod}	$2+t_{tdmin}$		$9+t_{tdmax}$	ns
Register Interrupt Output Delay Time	t_{irod}	$2+t_{tdmin}$		$11+t_{tdmax}$	ns
Register Parity Output Delay Time	t_{prod}	$2+t_{tdmin}$		$12+t_{tdmax}$	ns
Register Flag Output Delay Time	t_{frod}	$2+t_{tdmin}$		$9+t_{tdmax}$	ns
Latch to Latch					
Data to Latch OP Time	t_{dl}			3 + OP	ns
Latch to Latch Hold Time	t_{llh}	0			ns
Latch to Latch OP Time	t_{ll}			3 + OP	ns
Data to Data OP Time	t_{dd}	$2+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Latch Output Delay Time	t_{lod}	$2+t_{tdmin}$		$9+t_{tdmax}$	ns
Latch to Data OP Time	t_{ld}	$2+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Latch Interrupt Output Delay Time	t_{ilod}	$2+t_{tdmin}$		$11+t_{tdmax}$	ns
Data to Interrupt OP Time	t_{idd}	$2+t_{tdmin}$		$8+OP+t_{tdmax}$	ns
Latch to Interrupt OP Time	t_{ild}	$2+t_{tdmin}$		$8+OP+t_{tdmax}$	ns
Latch Parity Output Delay Time	t_{plod}	$2+t_{tdmin}$		$12+t_{tdmax}$	ns
Data to Parity OP Time	t_{dp}	$2+t_{tdmin}$		$9+OP+t_{tdmax}$	ns
Latch to Parity OP Time	t_{lp}	$2+t_{tdmin}$		$9+OP+t_{tdmax}$	ns
Latch Flag Output Delay Time	t_{flod}	$2+t_{tdmin}$		$9+t_{tdmax}$	ns
Data to Flag OP Time	t_{df}	$2+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Latch to Flag OP Time	t_{lf}	$2+t_{tdmin}$		$6+OP+t_{tdmax}$	ns
Clock and Enable Pulse Width					
Clock and Latch Pulse Duration					
High FALU	t_{ch}	4.0			ns
High FMPY	t_{ch}	10.0			ns
Low	t_{cl}	4.0			ns
Enable Pulse Duration					
High	t_{enh}	4.0			ns
Low	t_{enl}	4.0			ns

Parameter	Symbol	Value			Units
		Min	Nom	Max	
MSWSEL Output Delay					
Multiplexer Output Delay	t_{mod}	$2 + t_{tdmin}$		$8 + t_{tdmax}$	ns
Reset					
Reset to Status and Control Register Write Set Up Time	t_{rst}	16.0			ns
Reset Pulse Duration High	t_{rsh}	4.0			ns
Output Enables B3110/B3120					
Synchronous Output Enable Delay	t_{seno}	—		6	ns
Synchronous Output Disable Delay	t_{sdao}	—		6	ns
Output Enable Delay	t_{eno}	—		5	ns
Output Disable Delay	t_{dao}	—		5	ns
Output Enables B2110/B2120					
Synchronous Output Disable Delay					
High State to High Z	t_{sphz}	—		13	ns
Low State to High Z	t_{splz}	—		13	ns
Synchronous Output Enable Delay					
High Z to High State	t_{spzh}	—		13	ns
High Z to Low State	t_{spzl}	—		13	ns
Output Disable Delay					
High State to High Z	t_{phz}	—		12	ns
Low State to High Z	t_{plz}	—		12	ns
Output Enable Delay					
High Z to High State	t_{pzh}	—		12	ns
High Z to Low State	t_{pzl}	—		12	ns
Smode					
Smode Setup Time	t_{ss}	—			ns
Smode Hold Time	t_{sh}	—			ns
Sout Output Delay Time	t_{sod}	—		—	ns

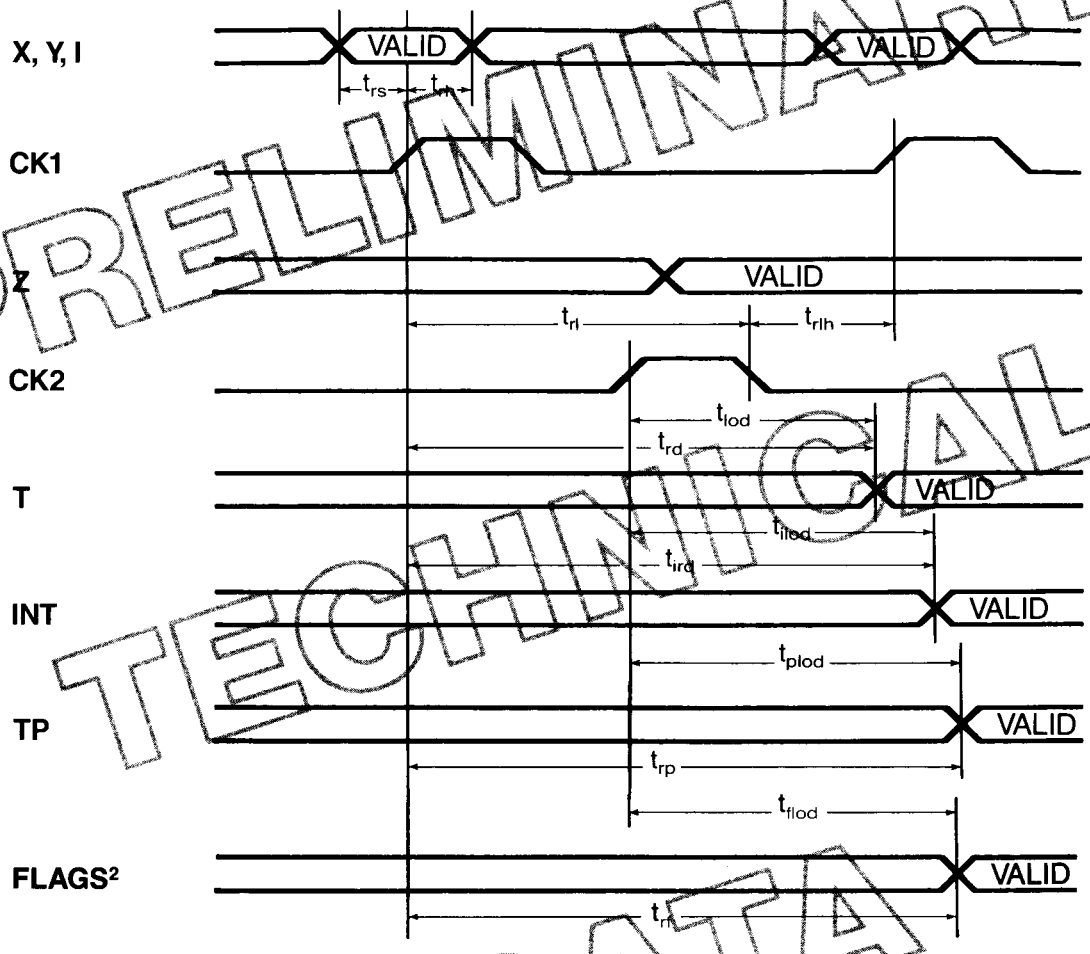
REGISTER TO REGISTER (R/L1 = R/L2 = 1)



NOTE 1: FLAGS EXCEPT INT

REGISTER TO LATCH

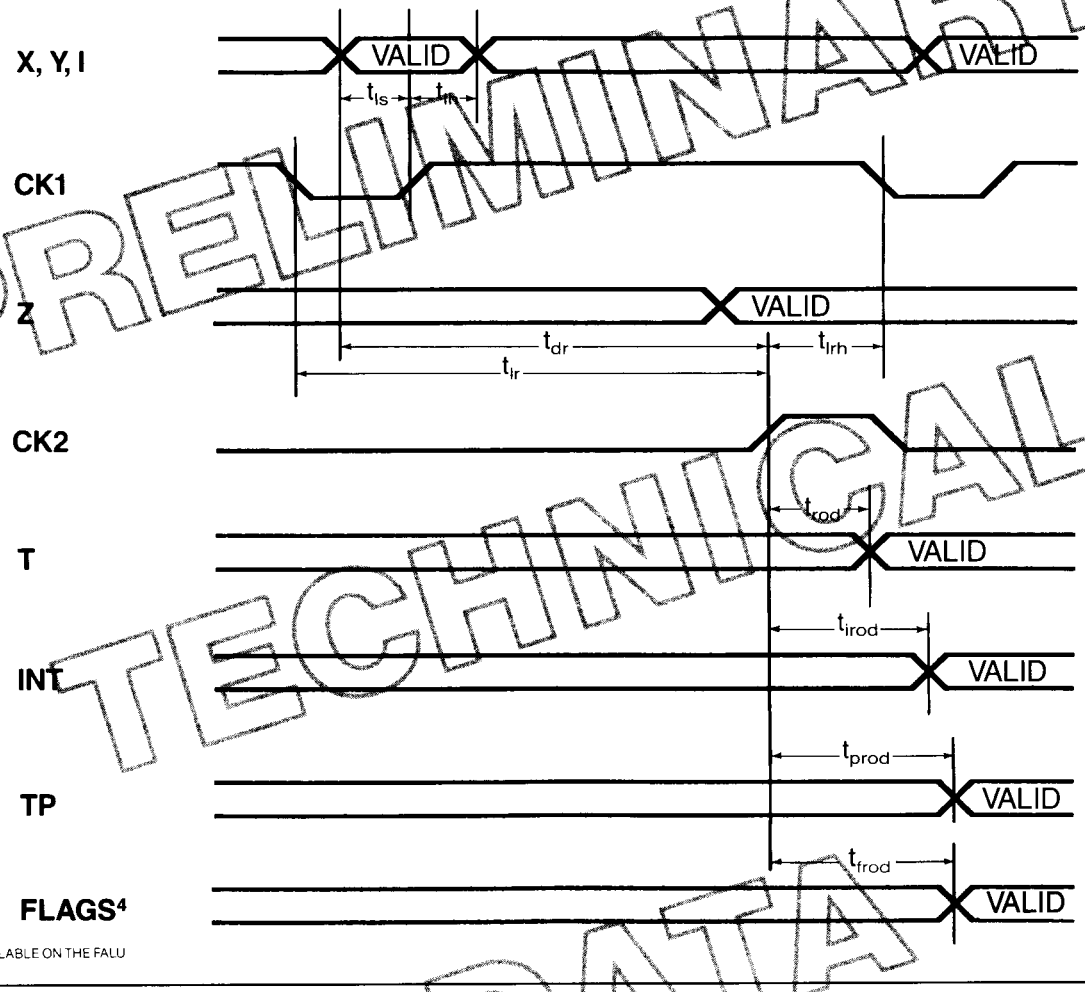
(R/L1 = 1, R/L2 = 0)



NOTE 2. FLAGS EXCEPT INT

LATCH TO REGISTER[3]

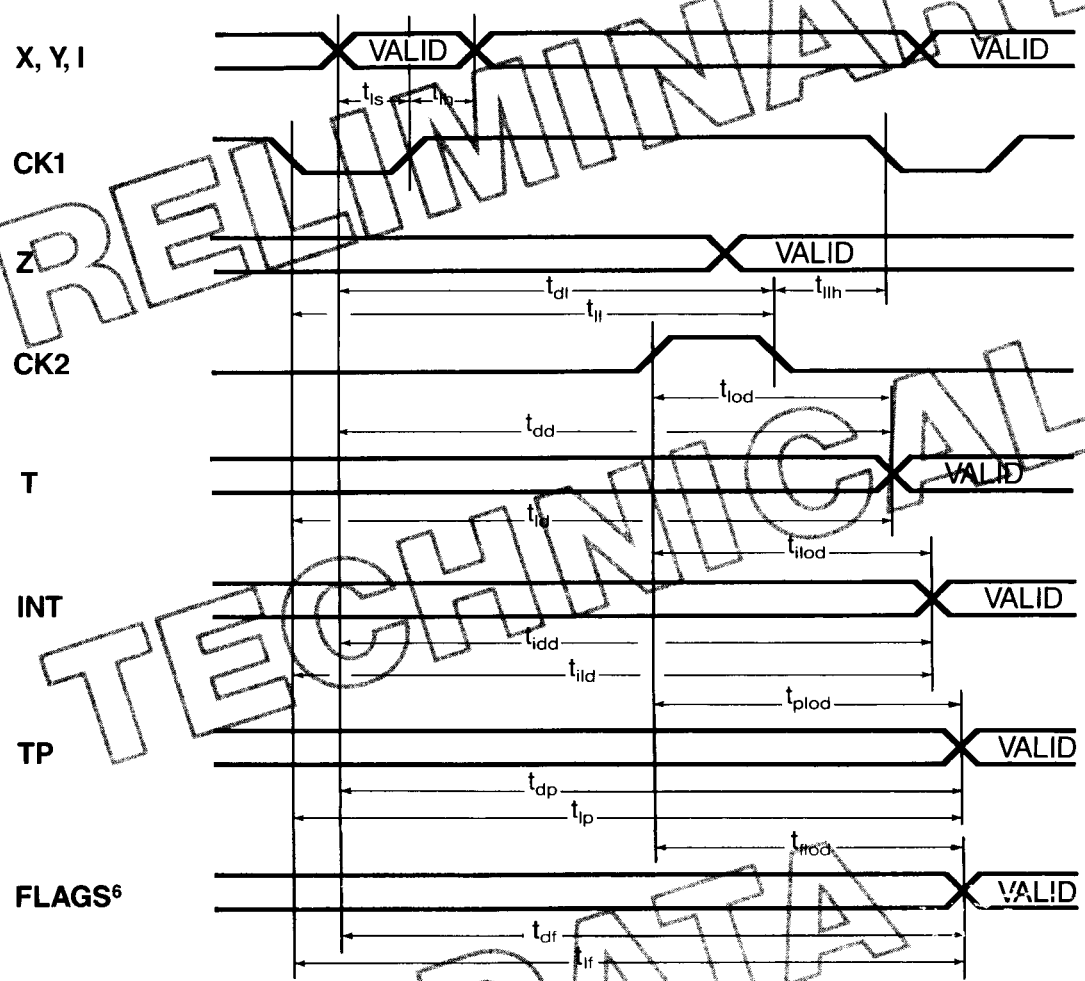
(R/L1 = 0, R/L2 = 1)



NOTE 3. THIS MODE IS ONLY AVAILABLE ON THE FALU
NOTE 4. FLAGS EXCEPT INT

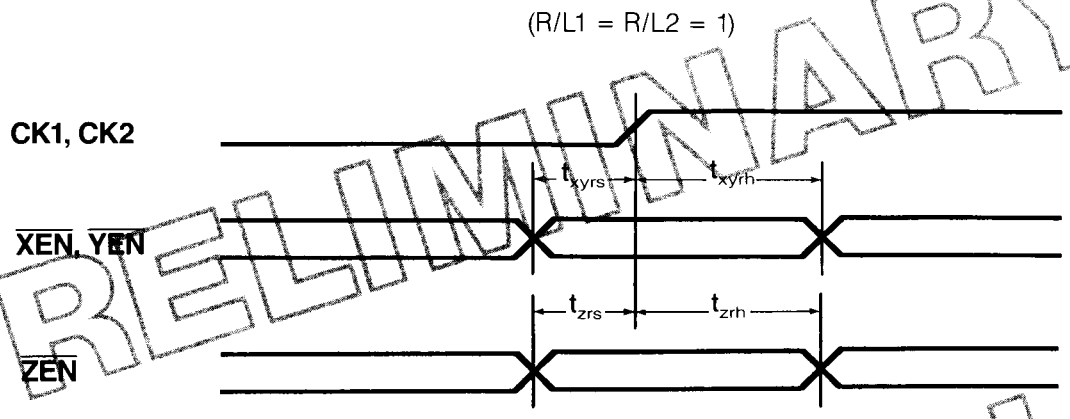
LATCH TO LATCH[5]

(R/L1 = 0, R/L2 = 0)

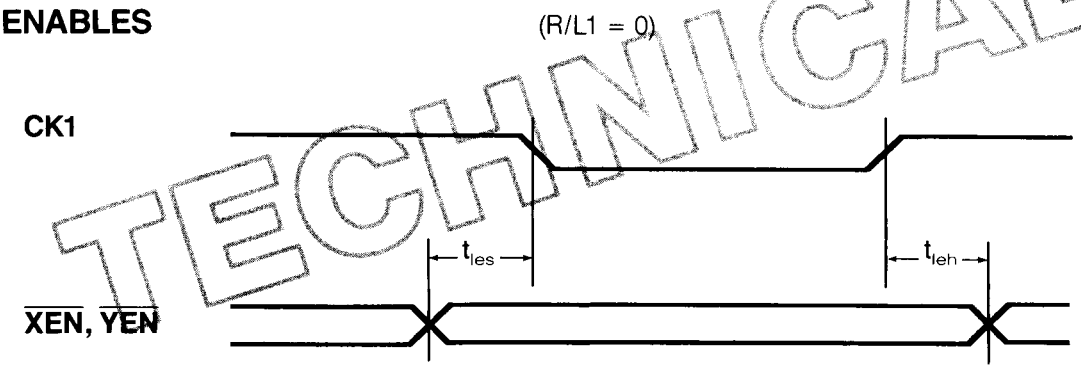


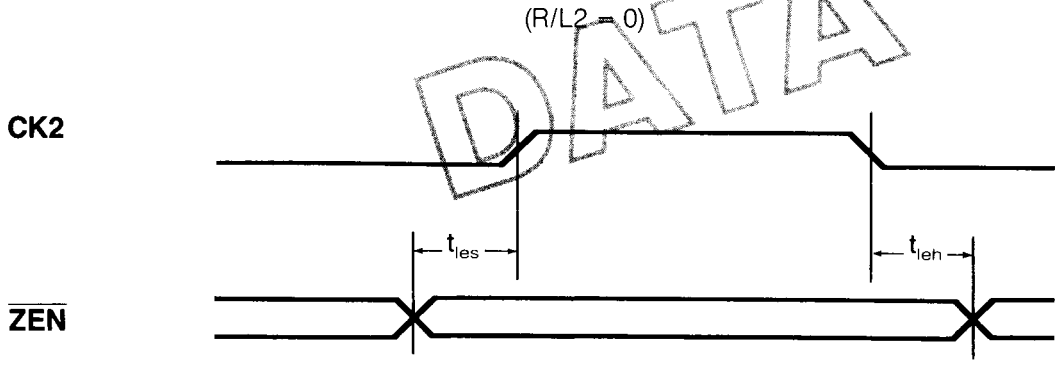
NOTE 5: THIS MODE IS ONLY AVAILABLE ON THE FALU
NOTE 6: FLAGS EXCEPT INT

REGISTER ENABLES[7]



LATCH ENABLES

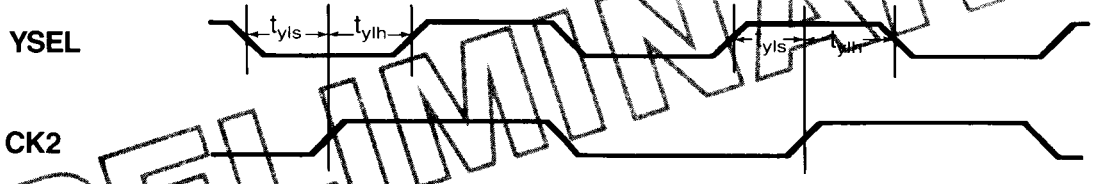




NOTE 7: $\overline{XEN}$, $\overline{YEN}$, AND $\overline{ZEN}$ ARE LATCHED INTERNALLY ON THE RISING EDGE OF CK1, CK2 RESPECTIVELY

YSEL

(R/L1 = 0, R/L2 = 1) See Table 1

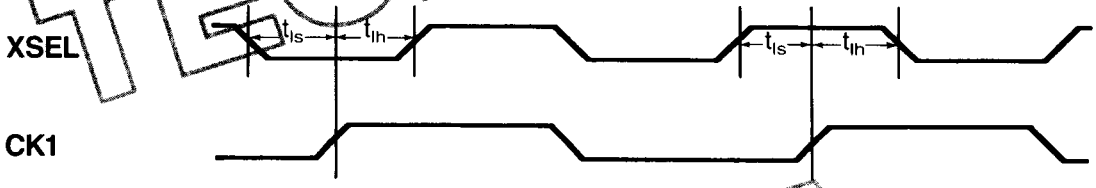


XSEL, YSEL[8]

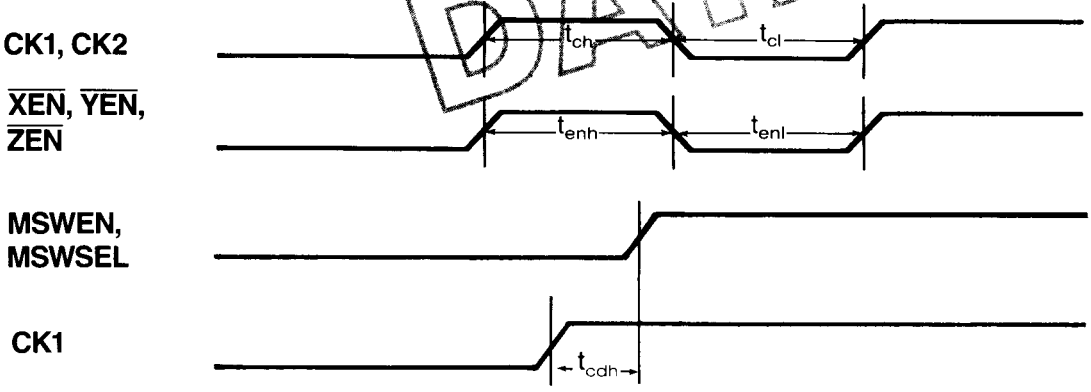
(R/L1 = 1)

XSEL

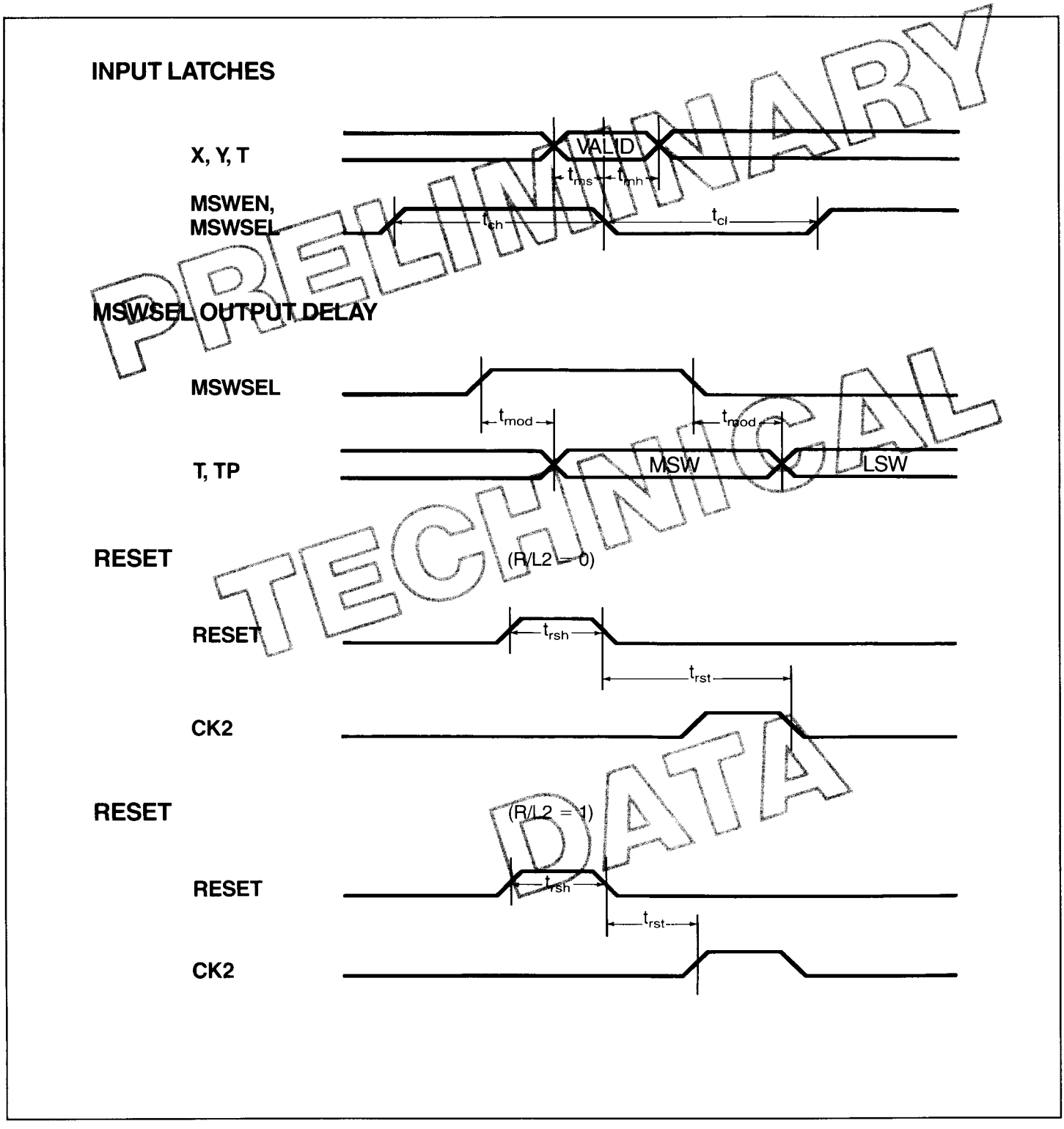
(R/L1 = 0)



CLOCK

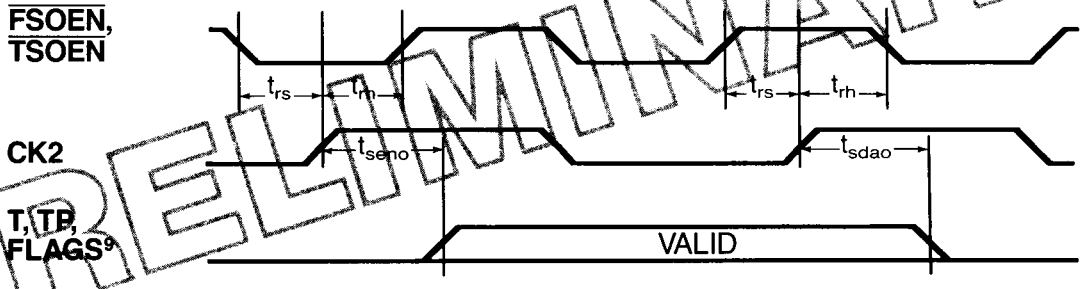


NOTE 8: REFERENCE PAGE 15 FOR ADDITIONAL INFORMATION ON YSEL DATA PATH OPERATION



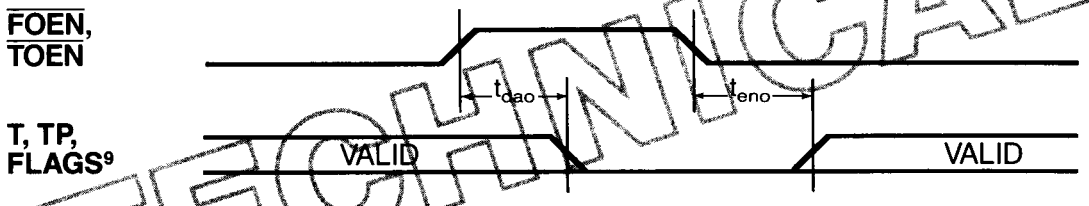
SYNCHRONOUS OUTPUT ENABLES

ADSP-8110/ADSP-8120



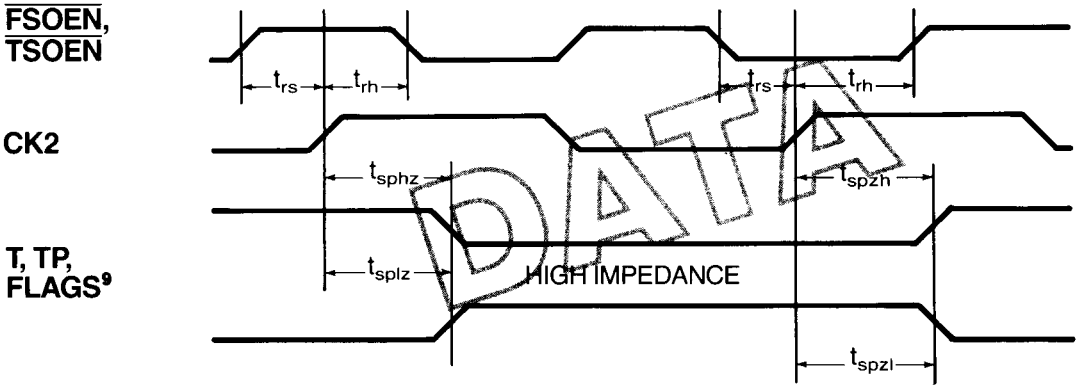
OUTPUT ENABLES

ADSP-8110/ADSP-8120



SYNCHRONOUS OUTPUT ENABLES

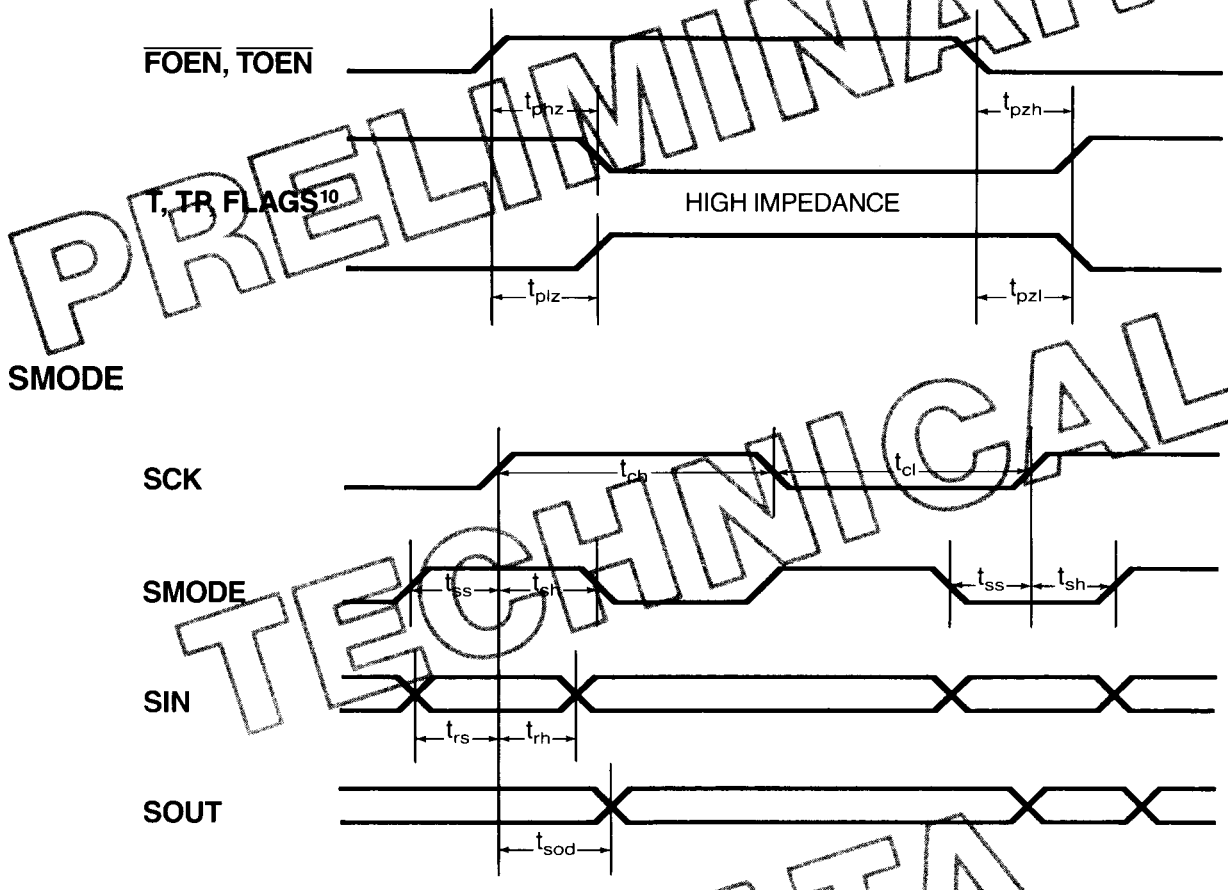
ADSP-7110/ADSP-7120



NOTE 9: THE ROUNDED UP FLAG (RND) IS NOT AFFECTED BY THE OUTPUT ENABLES

OUTPUT ENABLES

ADSP-7110/ADSP-7120



NOTE 10: THE ROUNDED UP FLAG (RND) IS NOT AFFECTED BY THE OUTPUT ENABLES

PINOUTS

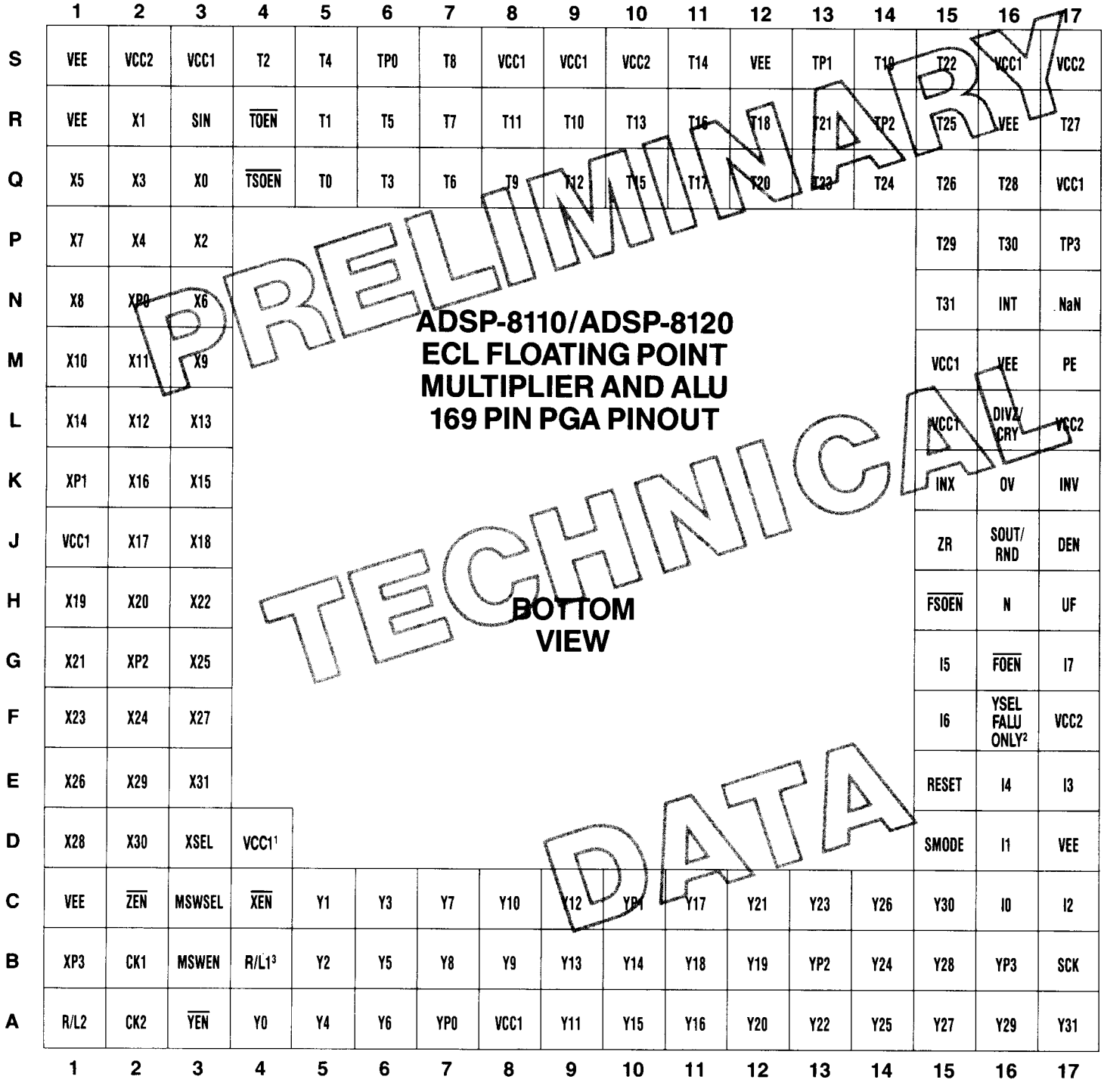
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
S	GND2	VCC2	VCC1	T2	T4	TP0	T8	GND2	VCC1	VCC2	T14	GND1	TP1	T19	T22	VCC1	VCC2	S
R	GND1	X1	SIN	$\overline{\text{TOEN}}$	T1	T5	T7	T11	T10	T13	T16	T18	T21	TP2	T25	GND2	T27	R
Q	X5	X3	X0	$\overline{\text{TSOEN}}$	T0	T3	T6	T9	T12	T15	T17	T20	T23	T24	T26	T28	GND2	Q
P	X7	X4	X2												T29	T30	TP3	P
N	X8	XP0	X6												T31	INT	NaN	N
M	X10	X11	X9												VCC1	GND2	PE	M
L	X14	X12	X13												VCC1	DIVZ/ CRY	VCC2	L
K	XP1	X16	X15												INX	OV	INV	K
J	VCC1	X17	X18												ZR	SOUT/ RND	DEN	J
H	X19	X20	X22												$\overline{\text{FSOEN}}$	N	UF	H
G	X21	XP2	X25												I5	$\overline{\text{FOEN}}$	I7	G
F	X23	X24	X27												I6	YSEL FALU ONLY[2]	VCC2	F
E	X26	X29	X31												RESET	I4	I3	E
D	X28	X30	XSEL	VCC1[1]											SMODE	I1	GND1	D
C	GND1	$\overline{\text{ZEN}}$	MSWSEL	$\overline{\text{XEN}}$	Y1	Y3	Y7	Y10	Y12	YP1	Y17	Y21	Y23	Y26	Y30	I0	I2	C
B	XP3	CK1	MSWEN	R/L1[3]	Y2	Y5	Y8	Y9	Y13	Y14	Y18	Y19	YP2	Y24	Y28	YP3	SCK	B
A	R/L2	CK2	$\overline{\text{YEN}}$	Y0	Y4	Y6	YP0	VCC1	Y11	Y15	Y16	Y20	Y22	Y25	Y27	Y29	Y31	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

ADSP-7110/ADSP-7120
TTL FLOATING POINT
MULTIPLIER AND ALU
169 PIN PGA PINOUT

BOTTOM
VIEW

NOTE 1: OPTIONAL CAN BE LEFT AS A NO CONNECT
NOTE 2: NO CONNECT ON FMPY
NOTE 3: MUST BE TIED HIGH ON FMPY

Floating Point Chip Set



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
S	VEE	VCC2	VCC1	T2	T4	TP0	T8	VCC1	VCC1	VCC2	T14	VEE	TP1	T19	T22	VCC1	VCC2	S
R	VEE	X1	SIN	$\overline{\text{TOEN}}$	T1	T5	T7	T11	T10	T13	T16	T18	T21	TP2	T25	VEE	T27	R
Q	X5	X3	X0	$\overline{\text{TSOEN}}$	T0	T3	T6	T9	T12	T15	T17	T20	T23	T24	T26	T28	VCC1	Q
P	X7	X4	X2												T29	T30	TP3	P
N	X8	XP0	X6												T31	INT	NaN	N
M	X10	X11	X9												VCC1	VEE	PE	M
L	X14	X12	X13												VCC1	DIVZ/ CRY	VCC2	L
K	XP1	X16	X15												INX	OV	INV	K
J	VCC1	X17	X18												ZR	SOUT/ RND	DEN	J
H	X19	X20	X22												$\overline{\text{FSOEN}}$	N	UF	H
G	X21	XP2	X25												I5	$\overline{\text{FOEN}}$	I7	G
F	X23	X24	X27												I6	YSEL FALU ONLY[2]	VCC2	F
E	X26	X29	X31												RESET	I4	I3	E
D	X28	X30	XSEL	VCC1[1]											SMODE	I1	VEE	D
C	VEE	$\overline{\text{ZEN}}$	MSWSEL	$\overline{\text{XEN}}$	Y1	Y3	Y7	Y10	Y12	YP1	Y17	Y21	Y23	Y26	Y30	I0	I2	C
B	XP3	CK1	MSWEN	R/L1[3]	Y2	Y5	Y8	Y9	Y13	Y14	Y18	Y19	YP2	Y24	Y28	YP3	SCK	B
A	R/L2	CK2	$\overline{\text{YEN}}$	Y0	Y4	Y6	YP0	VCC1	Y11	Y15	Y16	Y20	Y22	Y25	Y27	Y29	Y31	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

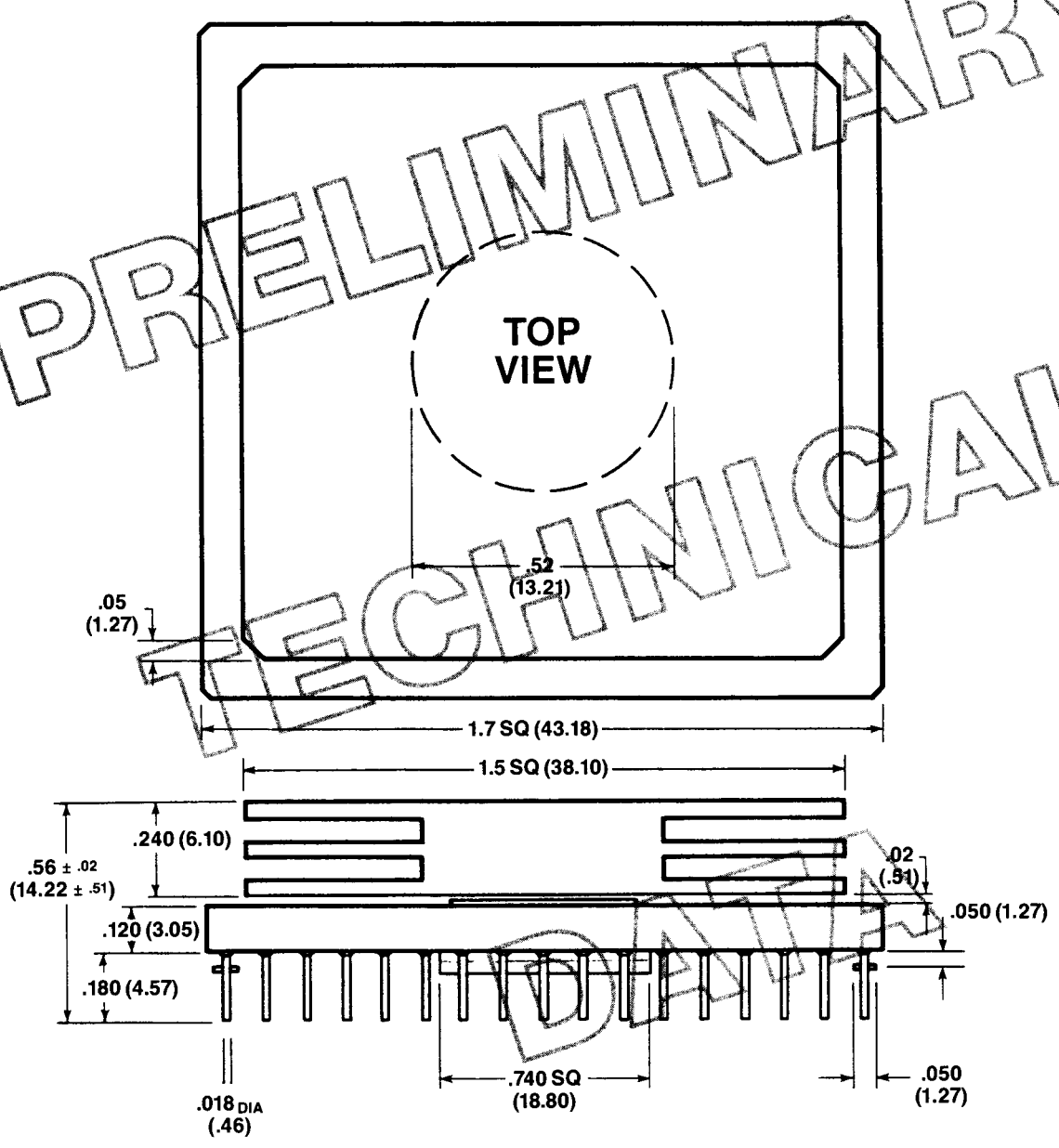
ADSP-8110/ADSP-8120
ECL FLOATING POINT
MULTIPLIER AND ALU
169 PIN PGA PINOUT

BOTTOM
VIEW

NOTE 1: OPTIONAL CAN BE LEFT AS A NO CONNECT
NOTE 2: NO CONNECT ON FMPY
NOTE 3: MUST BE TIED HIGH ON FMPY

■ PACKAGING DRAWING AND DIMENSIONS

DIMENSIONS SHOWN IN INCHES AND (MM).

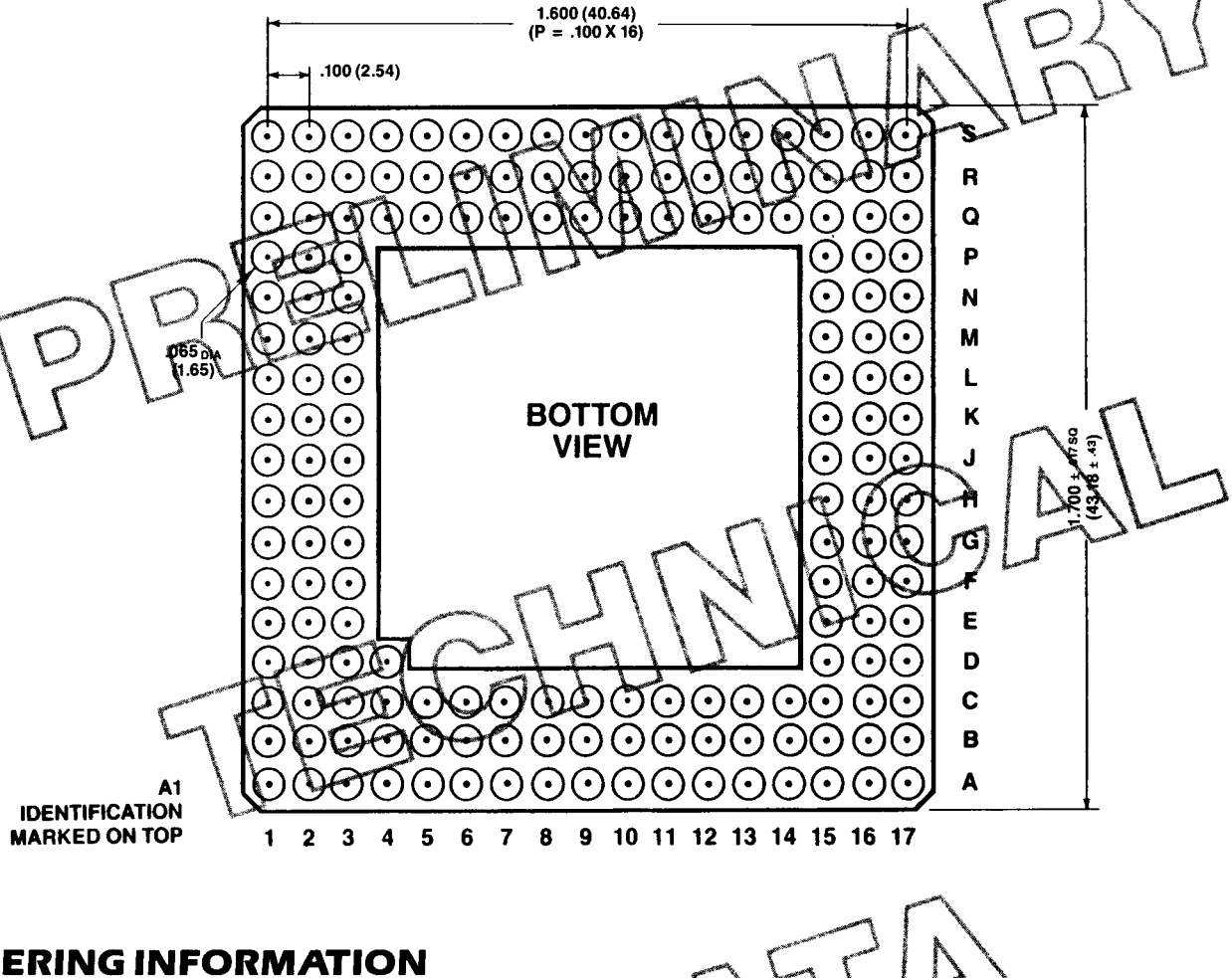


ANALOG DEVICES

One Technology Way
P.O. Box 9106
Norwood, MA 02062-9106
(617) 329-4700

PACKAGING DRAWING AND DIMENSIONS

DIMENSIONS SHOWN IN INCHES AND (MM).



ORDERING INFORMATION

Order Number	Package Type	Temperature Range
ADSP-8110JG	169 Pin Pin-Grid-Array	0–70°
ADSP-8110KG	169 Pin Pin-Grid-Array	0–70°
ADSP-8110LG	169 Pin Pin-Grid-Array	0–70°
ADSP-8120JG	169 Pin Pin-Grid-Array	0–70°
ADSP-8120KG	169 Pin Pin-Grid-Array	0–70°
ADSP-7110JG	169 Pin Pin-Grid-Array	0–70°
ADSP-7110KG	169 Pin Pin-Grid-Array	0–70°
ADSP-7110LG	169 Pin Pin-Grid-Array	0–70°
ADSP-7120JG	169 Pin Pin-Grid-Array	0–70°
ADSP-7120KG	169 Pin Pin-Grid-Array	0–70°