

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com

**Full-service, independent repair center**
with experienced engineers and technicians on staff.

**We buy your excess, underutilized, and idle equipment**
along with credit for buybacks and trade-ins.

**Custom engineering**
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

**Expert team | Trust guarantee | 100% satisfaction**

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

**Visit our website - *Click HERE***

### 1.1 Scope.

This specification covers the detail requirements for a CMOS monolithic 32-bit and 64-bit IEEE Standard 754 format floating-point arithmetic and logic unit (ALU).

### 1.2 Part Number.

The complete part number per Table 1 of this specification is as follows:

| Device | Part Number |
|---|---|
| −1 | ADSP-3221SG/883B |
| −2 | ADSP-3221TG/883B |

### 1.2.3 Case Outline.

See Appendix 1 of General Specification ADI-M-1000: package outline: G-144A.

### 1.3 Absolute Maximum Ratings.

| | |
|---|---|
| Supply Voltage | −0.3 V to 7 V |
| Input Voltage | −0.3 V to $V_{DD}$ |
| Output Voltage | −0.3 V to $V_{DD}$ |
| Operating Temperature Range (Ambient) | −55°C to +125°C |
| Storage Temperature Range | −65°C to +150°C |
| Lead Temperature (Soldering 10 sec) | +300°C |

### 1.5 Thermal Characteristics.

Maximum Thermal Resistance $\theta_{JC}$: see MIL-M-38510, Appendix C.



Downloaded from Partstack.com

# ADSP-3221 – SPECIFICATIONS

| Test | Symbol | Device | Design Limit @ +25°C | Sub Group 1 | Sub Group 2, 3 | Sub Group 9 | Sub Group 10, 11 | Test Condition[1] | Units |
|---|---|---|---|---|---|---|---|---|---|
| Digital Input High Voltage | $V_{IH}$ | −1, 2 | 2.0 | 2.0 | 2.0 | | | $V_{DD}$ = max | V min |
| Digital Input High Voltage, CLK and Asynchronous Controls (RESET, MSWSEL, OEN & IPORT0:1) | $V_{IHA}$ | −1, 2 | 3.0 | 3.0 | 3.0 | | | $V_{DD}$ = max | V min |
| Digital Input Low Voltage | $V_{IL}$ | −1, 2 | 0.8 | 0.8 | 0.8 | | | $V_{DD}$ = min | V max |
| Digital Output High Voltage | $V_{OH}$ | −1, 2 | 2.4 | 2.4 | 2.4 | | | $V_{DD}$ = min<br>$I_{OH}$ = −1 mA | V min |
| Digital Output Low Voltage | $V_{OL}$ | −1, 2 | 0.4 | 0.6 | 0.6 | | | $V_{DD}$ = min<br>$I_{OL}$ = +4 mA | V max |
| Digital Input High Current | $I_{IH}$ | −1, 2 | 10 | 10 | 10 | | | $V_{DD}$ = max<br>$V_{IN}$ = +5.0 V | µA max |
| Digital Input Low Current | $I_{IL}$ | −1, 2 | 10 | 10 | 10 | | | $V_{DD}$ = max<br>$V_{IN}$ = 0.0 V | µA max |
| Three-State Leakage Current | $I_{OZH}$<br>$I_{OZL}$ | −1, 2 | 50 | 50 | 50 | | | $V_{DD}$ = max<br>High Z, $V_{IN}$ = 0 V or max | µA max |
| Supply Current | $I_{DD1}$ | −1, 2 | 150 | 200 | 200 | | | @ max Clock Rate, TTL Inputs | mA max |
| | $I_{DD2}$ | | 50 | 60 | 60 | | | All $V_{IN}$ = 2.4 V | mA max |
| Clock Cycle | $t_{CY}$ | −1 | 125 | | | 150 | 150 | Note 2 | ns max |
| | | −2 | 100 | | | 125 | 125 | | |
| Clock LO | $t_{CL}$ | −1, 2 | 20 | | | 30 | 30 | Note 2 | ns min |
| Clock HI | $t_{CH}$ | −1, 2 | 20 | | | 30 | 30 | Note 2 | ns min |
| Data & Control Setup | $t_{DS}$ | −1 | 20 | | | 25 | 25 | Note 2 | ns min |
| | | −2 | 15 | | | 20 | 20 | | |
| Data & Control Hold | $t_{DH}$ | −1, 2 | 3 | | | 3 | 3 | Note 2 | ns min |
| Data Output Delay | $t_{DO}$ | −1 | 30 | | | 35 | 35 | Note 2 | ns max |
| | | −2 | 25 | | | 30 | 30 | | |
| Status Output Delay | $t_{SO}$ | −1 | 30 | | | 35 | 35 | Note 2 | ns max |
| | | −2 | 25 | | | 30 | 30 | | |
| MSWSEL-to-Data Delay | $t_{ENO}$ | −1 | 25 | | | 30 | 30 | Note 2 | ns max |
| | | −2 | 20 | | | 25 | 25 | | |
| Three-State Disable Delay | $t_{DIS}$ | −1 | 18 | | | 25 | 25 | Notes 2 & 3 | ns max |
| | | −2 | 15 | | | 20 | 20 | | |
| Three-State Enable Delay | $t_{ENA}$ | −1 | 25 | | | 30 | 30 | Notes 2, 3 & 4 | ns max |
| | | −2 | 20 | | | 25 | 25 | | |
| RESET Setup | $t_{SU}$ | −1, 2 | 25 | | | 25 | 25 | Note 2 | ns min |
| RESET Pulse Duration | $t_{RS}$ | −1, 2 | 75 | | | 75 | 75 | Note 2 | ns min |
| Operation Time (With or Without Direct Operand Feed): | $t_{OPD}$ | | | | | | | | |
| 32-Bit Multiplication | | −1 | 125 | | | 150 | 150 | Note 2 | ns max |
| | | −2 | 100 | | | 125 | 125 | | |
| 64-Bit ALU Operation | $t_{OPD}$ | −1 | 125 | | | 150 | 150 | Note 2 | ns max |
| | | −2 | 100 | | | 125 | 125 | | |

*Table 1. (Continued on next page)*

REV. A



Downloaded from Partstack.com

| Test | Symbol | Device | Design Limit @ +25°C | Sub Group 1 | Sub Group 2, 3 | Sub Group 9 | Sub Group 10, 11 | Test Condition[1] | Units |
|---|---|---|---|---|---|---|---|---|---|
| Hold Setup | $t_{HS}$ | −1 | 20 | | | 22 | 22 | Note 2 | ns min |
| | | −2 | 15 | | | 18 | 18 | | |
| Hold Hold | $t_{HH}$ | −1, 2 | 3 | | | 3 | 3 | Note 2 | ns min |
| Total Latency[5] (With Direct Operand Feed): | $t_{LAD}$ | | | | | | | | |
| 32-Bit ALU Operation | | −1 | 300 | | | 360 | 360 | Note 2 | ns max |
| | | −2 | 240 | | | 300 | 300 | | |
| 64-Bit ALU Operation | | −1 | 363 | | | 435 | 435 | Note 2 | ns max |
| | | −2 | 290 | | | 363 | 363 | | |
| 32-Bit Division | | −1 | 2.175 | | | 2.61 | 2.61 | Note 2 | µs max |
| | | −2 | 1.74 | | | 2.175 | 2.175 | | |
| 64-Bit Division | | −1 | 3.925 | | | 4.71 | 4.71 | Note 2 | µs max |
| | | −2 | 3.14 | | | 3.925 | 3.925 | | |
| 32-Bit Square Root | | −1 | 3.8 | | | 4.56 | 4.56 | Note 2 | µs max |
| | | −2 | 3.04 | | | 3.8 | 3.8 | | |
| 64-Bit Square Root | | −1 | 7.425 | | | 8.91 | 8.91 | Note 2 | µs max |
| | | −2 | 5.94 | | | 7.425 | 7.425 | | |

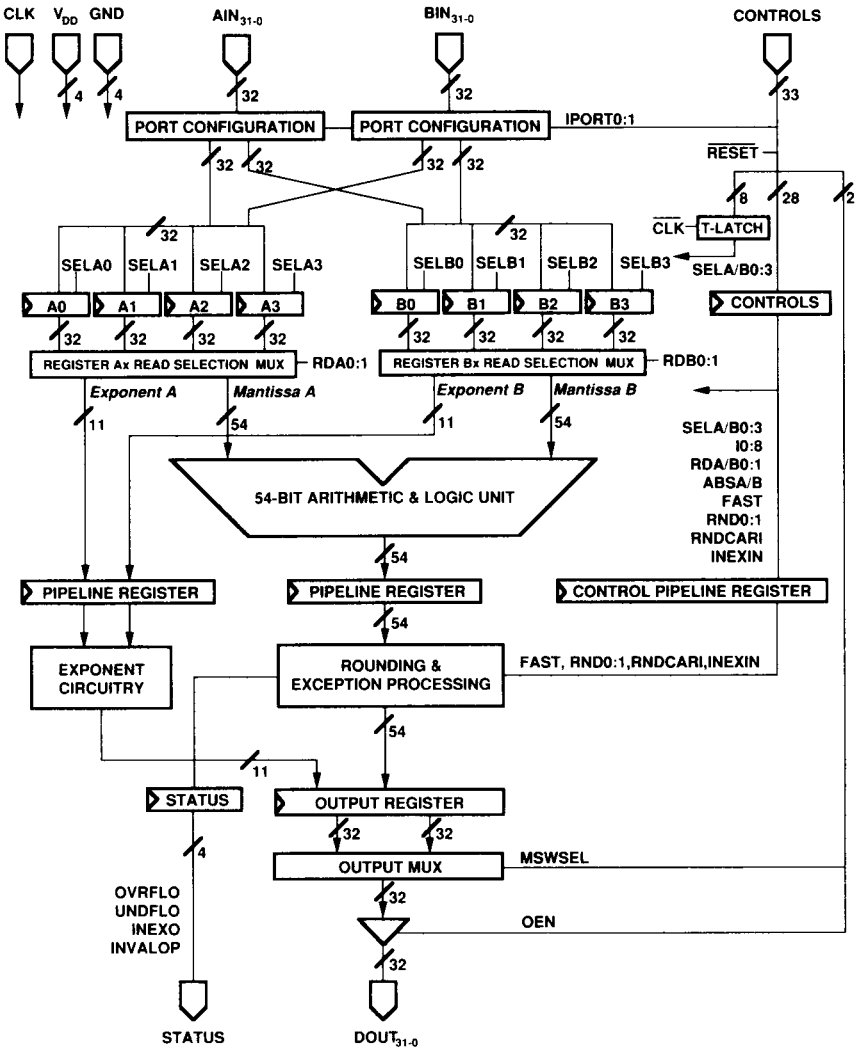
NOTES
[1]$T_A$ = +25°C; $V_{DD}$ = +4.5 V min to +5.5 V max (unless otherwise noted).
[2]Input levels are GND and +3.0 V; $V_{DD}$ = +4.5 V, and timing transitions, per Figure 1 through 8, measured at +1.5 V.
[3]Transitions measured per Figure 1.
[4]3 ns minimum.
[5]Total latency = (data setup + processing + output delay of MSW) in Direct Operand Feed Mode.

*Table 1.*



Downloaded from Partstack.com

### 3.2.1 Functional Block Diagrams and Terminal Assignments.




**Pin Assignments**

| PIN | FUNCTION | PIN | FUNCTION |
|---|---|---|---|
| B2 | DIN3 | M12 | DOUT28 |
| B1 | DIN2 | M13 | DOUT29 |
| C2 | DIN1 | L12 | DOUT30 |
| C1 | DIN0 | L13 | DOUT31 |
| D2 | WRAPB | K12 | GND |
| D1 | RDB0 | K13 | GND |
| E2 | SELB0 | J12 | GND |
| E1 | SELB1 | J13 | DENORM |
| F3 | ABSB | H11 | INVALOP |
| F2 | DP | H12 | OVRFLO |
| F1 | SP | H13 | UNDFLO |
| G2 | CLK | G12 | OEN |
| G3 | RESET | G11 | MSWSEL |
| G1 | RND1 | G13 | SHLP |
| H1 | RND0 | F13 | FAST |
| H2 | RNDCARO | F12 | ABSA |
| H3 | $V_{DD}$ | F11 | SELA1 |
| J1 | $V_{DD}$ | E13 | SELA0 |
| J2 | $V_{DD}$ | E12 | RDA0 |
| K1 | INEXO | D13 | WRAPA |
| K2 | DOUT0 | D12 | DIN31 |
| L1 | DOUT1 | C13 | DIN30 |
| M1 | DOUT2 | B13 | DIN29 |
| L2 | DOUT3 | C12 | DIN28 |
| N1 | N/C | A13 | N/C |
| M2 | DOUT4 | B12 | DIN27 |
| N2 | DOUT5 | A12 | DIN26 |
| M3 | DOUT6 | B11 | DIN25 |
| N3 | DOUT7 | A11 | DIN24 |
| M4 | DOUT8 | B10 | DIN23 |
| N4 | DOUT9 | A10 | DIN22 |
| M5 | DOUT10 | B8 | DIN21 |
| N5 | DOUT11 | A9 | DIN20 |
| L6 | DOUT12 | C8 | DIN19 |
| M6 | DOUT13 | B8 | DIN18 |
| N6 | DOUT14 | A8 | DIN17 |
| M7 | DOUT15 | B7 | DIN16 |
| L7 | DOUT16 | C7 | DIN15 |
| N7 | DOUT17 | A7 | DIN14 |
| N8 | DOUT18 | A6 | DIN13 |
| M8 | DOUT19 | B6 | DIN12 |
| L8 | DOUT20 | C6 | DIN11 |
| N9 | DOUT21 | A5 | DIN10 |
| M9 | DOUT22 | B5 | DIN9 |
| N10 | DOUT23 | A4 | DIN8 |
| M10 | DOUT24 | B4 | DIN7 |
| N11 | DOUT25 | A3 | DIN6 |
| N12 | DOUT26 | A2 | DIN5 |
| M11 | DOUT27 | B3 | DIN4 |
| N13 | N/C | A1 | N/C |

### 3.2.4 Microcircuit Technology Group.

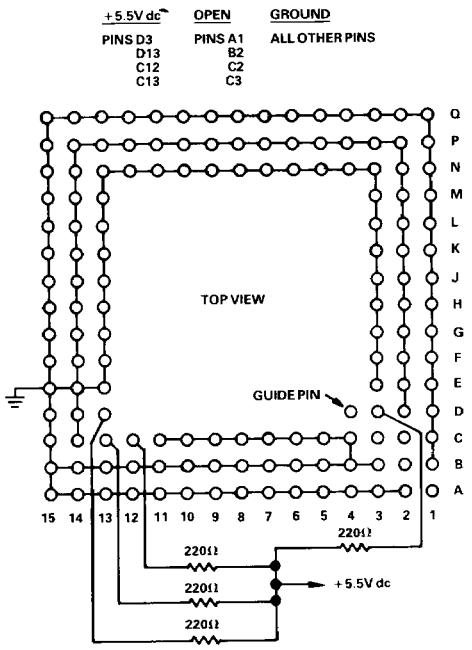
This microcircuit is covered by technology group (105).



Downloaded from Partstack.com

### 4.2.1 Life Test/Burn-In Circuit.

Steady state life test is per MIL-STD-883 Method 1005. Burn-in is per MIL-STD-883 Method 1015 test condition (B).




Output disable time, $t_{DIS}$, is measured from the time the output enable control signal reaches 1.5 V to the time when all outputs have ceased driving. This is calculated by measuring the time, $t_{MEASURED}$, from the same starting point to when the output voltages have changed by 0.5 V toward +1.5 V. From the tester capacitive loading, $C_L$, and the measured current, $i_L$, the decay time, $t_{DECAY}$, can be approximated to first order by:

$$t_{DECAY} = \frac{C_L \cdot 0.5\ V}{i_L}$$

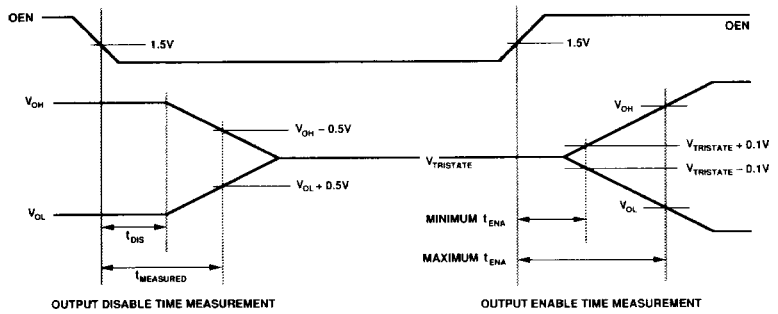
from which

$$t_{DIS} = t_{MEASURED} - t_{DECAY}$$

is calculated. Disable times are longest at the highest specified temperature.

The minimum output enable time, minimum $t_{ENA}$, is the earlist that outputs begin to drive. It is measured from the control signal OEN reaching 1.5 V to the point at which the fastest outputs have changed by 0.1 V from $V_{TRISTATE}$ toward their final output voltages. Minimum enable times are shortest at the lowest specified temperature.

The maximum output enable time, maximum $t_{ENA}$, is also measured from output enable control signal at 1.5 V to the time when all outputs have reached TTL input levels ($V_{OH}$ or $V_{OL}$). This could also be considered as "data valid." Maximum enable times are longest at the highest specified temperature.




REFER TO THE DISCUSSION IN THE SECTION "TIMING" IN THE TEXT OF THE DATA SHEET FOR A DESCRIPTION OF THIS FIGURE.

*Figure 1. Three-State Disable and Enable Timing*

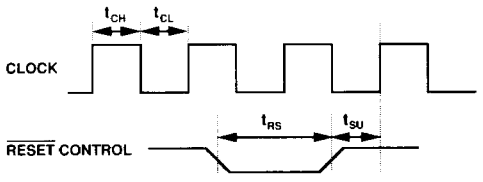
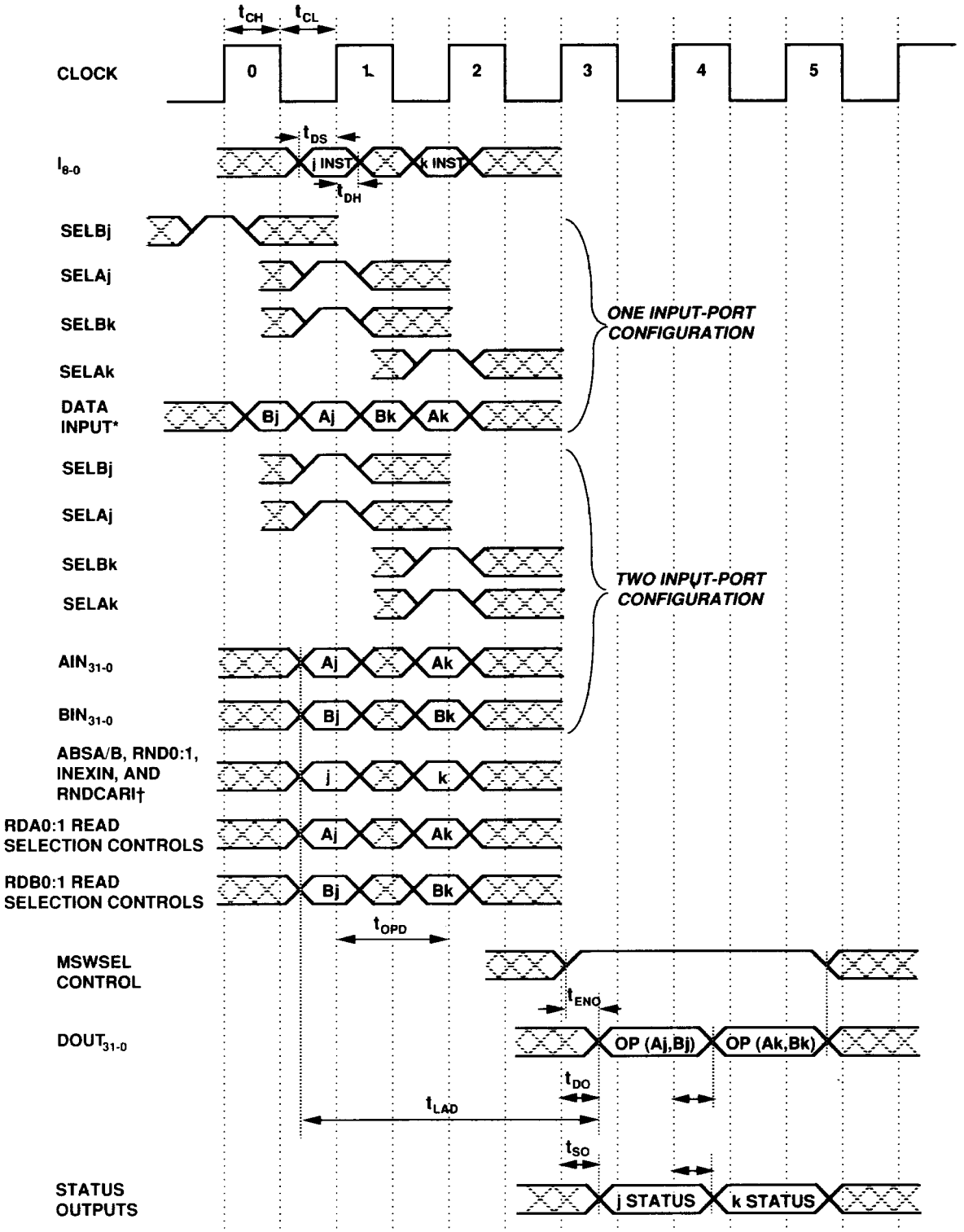



*Figure 2. Reset Timing*

REV. A



Downloaded from Partstack.com

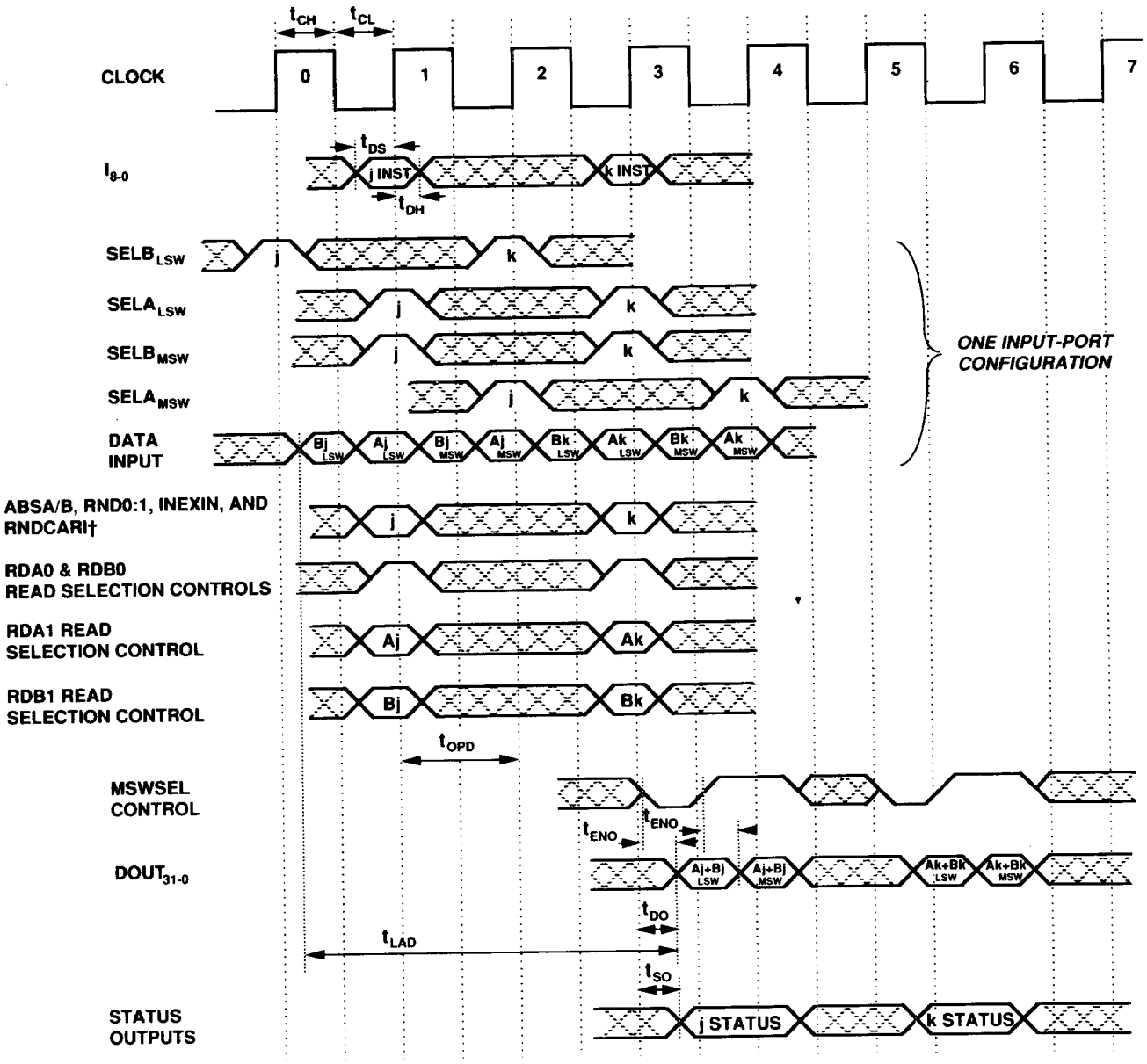



†RNDCARI AND INEXIN SHOULD BE LO EXCEPT FOR UNWRAP, DIVISION, AND SQUARE ROOT OPERATIONS.

*Figure 3. 32-Bit Single-Precision Floating-Point Logical and Fixed-Point ALU Operation*



Downloaded from Partstack.com

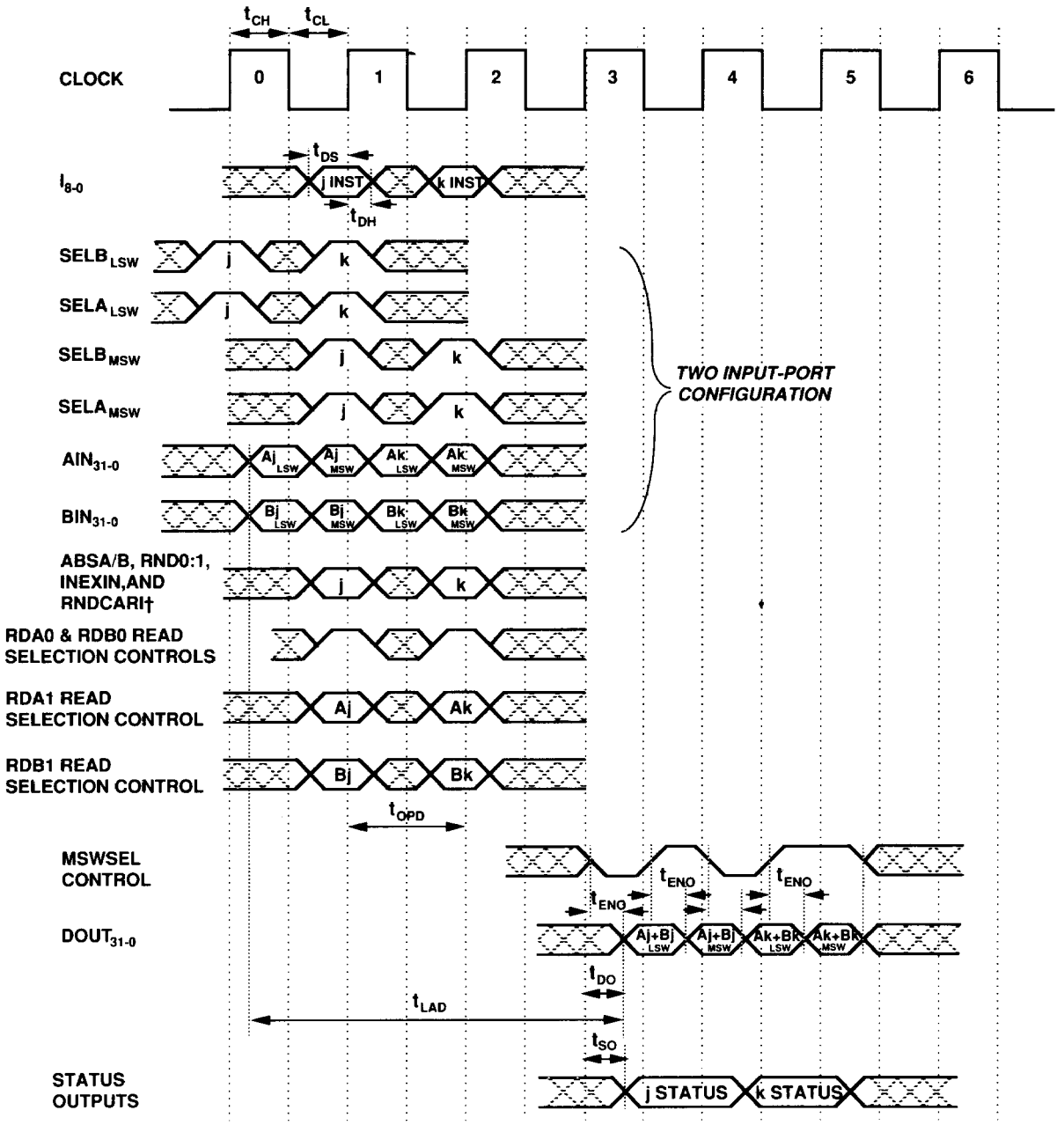



†RNDCARI AND INEXIN SHOULD BE LO EXCEPT FOR UNWRAP, DIVISION, AND SQUARE ROOT OPERATIONS.

*Figure 4. 64-Bit Double-Precision Floating Point ALU Operations – One Port Configuration*



Downloaded from Partstack.com

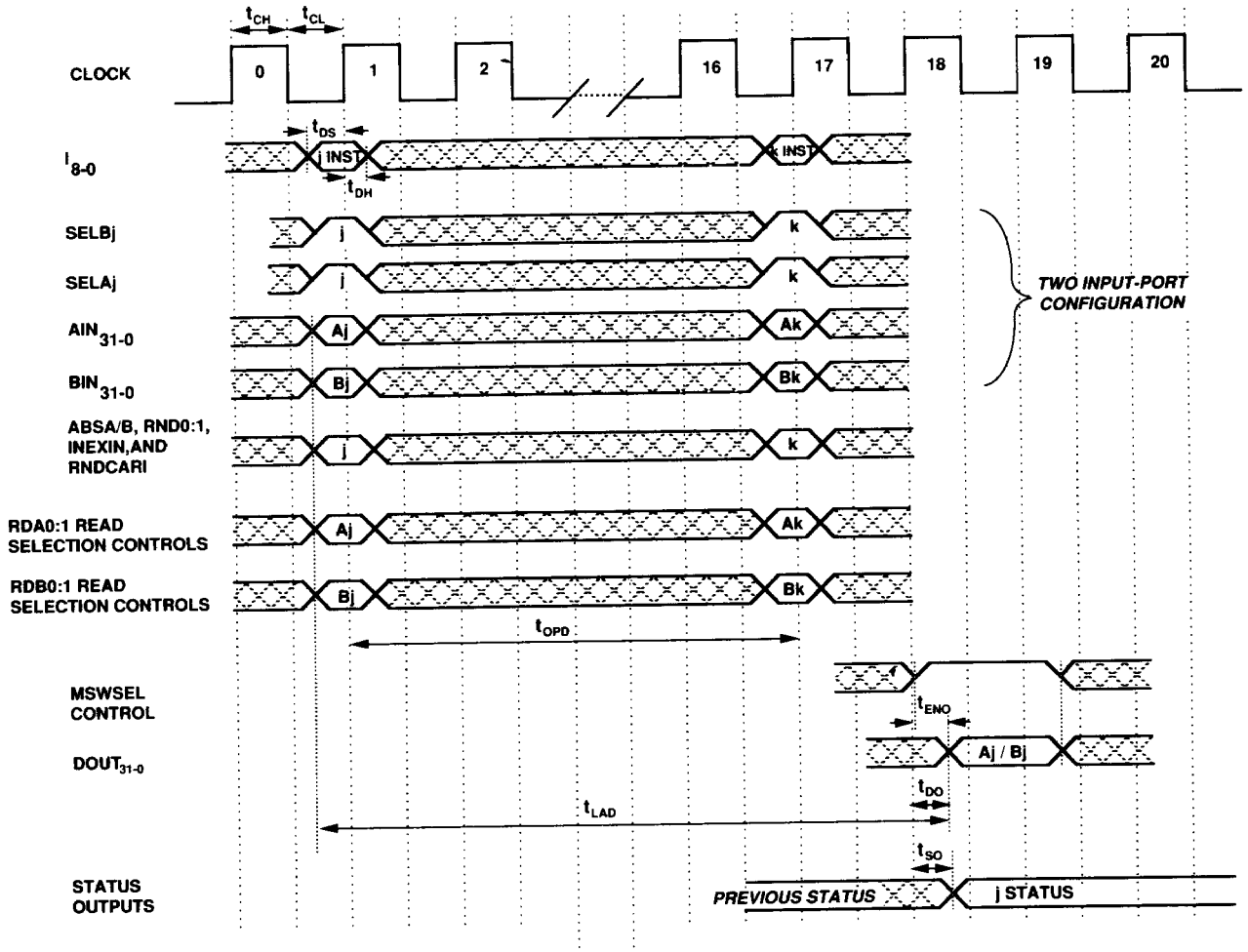



†RNDCARI AND INEXIN SHOULD BE LO EXCEPT FOR UNWRAP, DIVISION, AND SQUARE ROOT OPERATIONS.

*Figure 5. 64-Bit Double-Precision Floating-Point ALU Operations – Two-Port Configuration*



Downloaded from Partstack.com

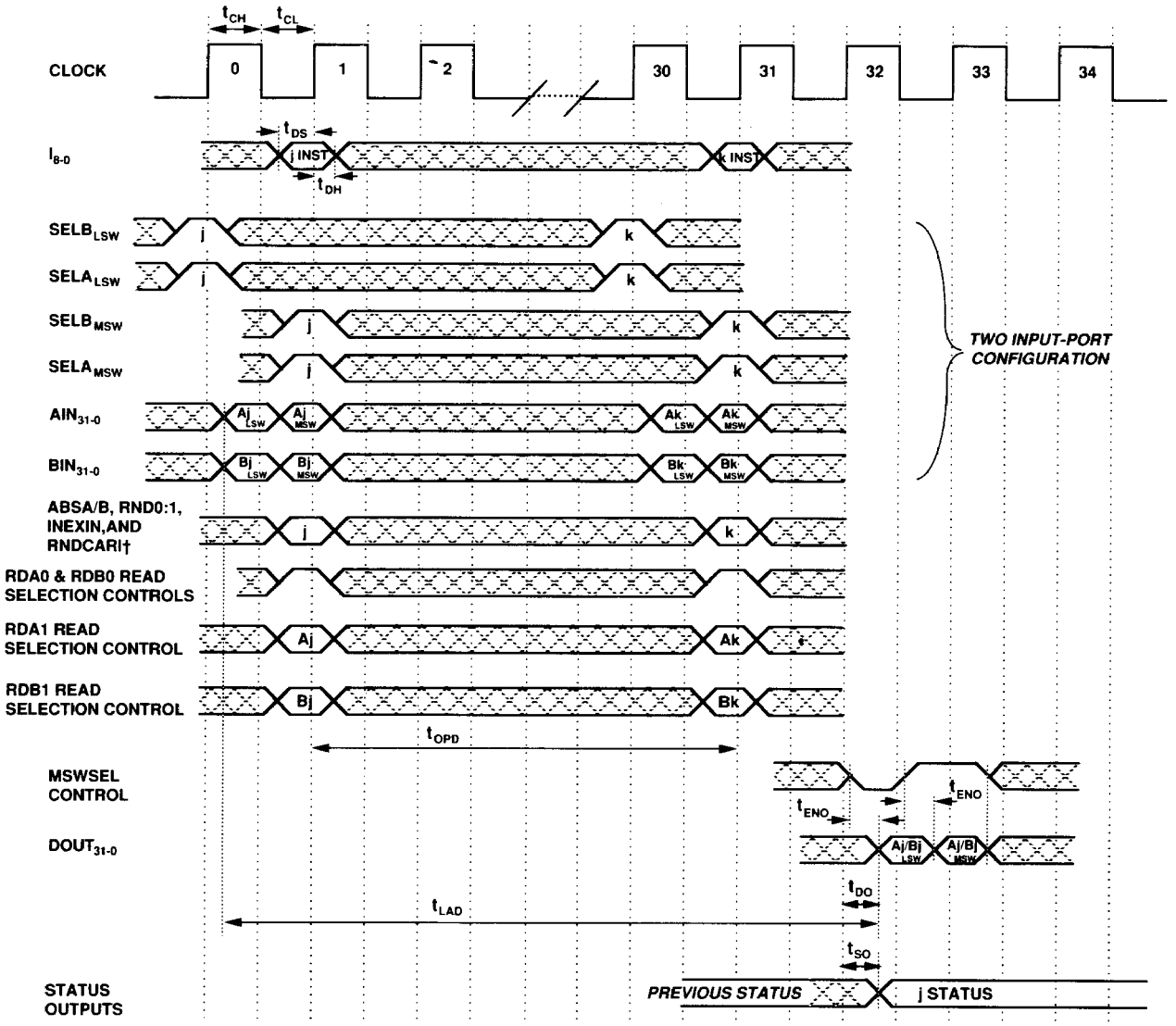



†RNDCARI AND INEXIN SHOULD BE LO EXCEPT FOR UNWRAP, DIVISION, AND SQUARE ROOT OPERATIONS.

*Figure 6. 32-Bit Single-Precision Floating-Point Division – Two Input-Port Configuration*



Downloaded from Partstack.com

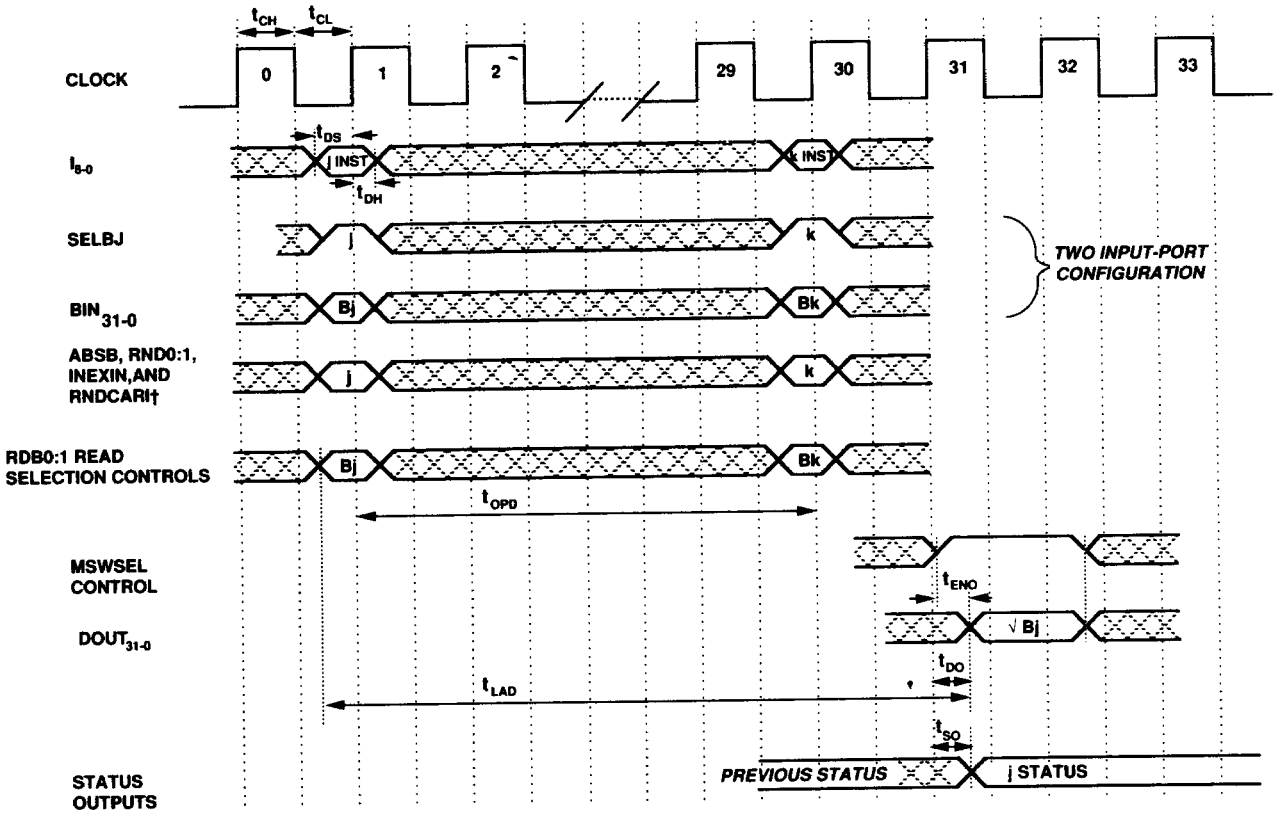



†RNDCARI AND INEXIN SHOULD BE LO EXCEPT FOR UNWRAP, DIVISION, AND SQUARE ROOT OPERATIONS.

*Figure 7. 64-Bit Double-Precision Floating-Point Division – Two Input-Port Configuration*



Downloaded from Partstack.com

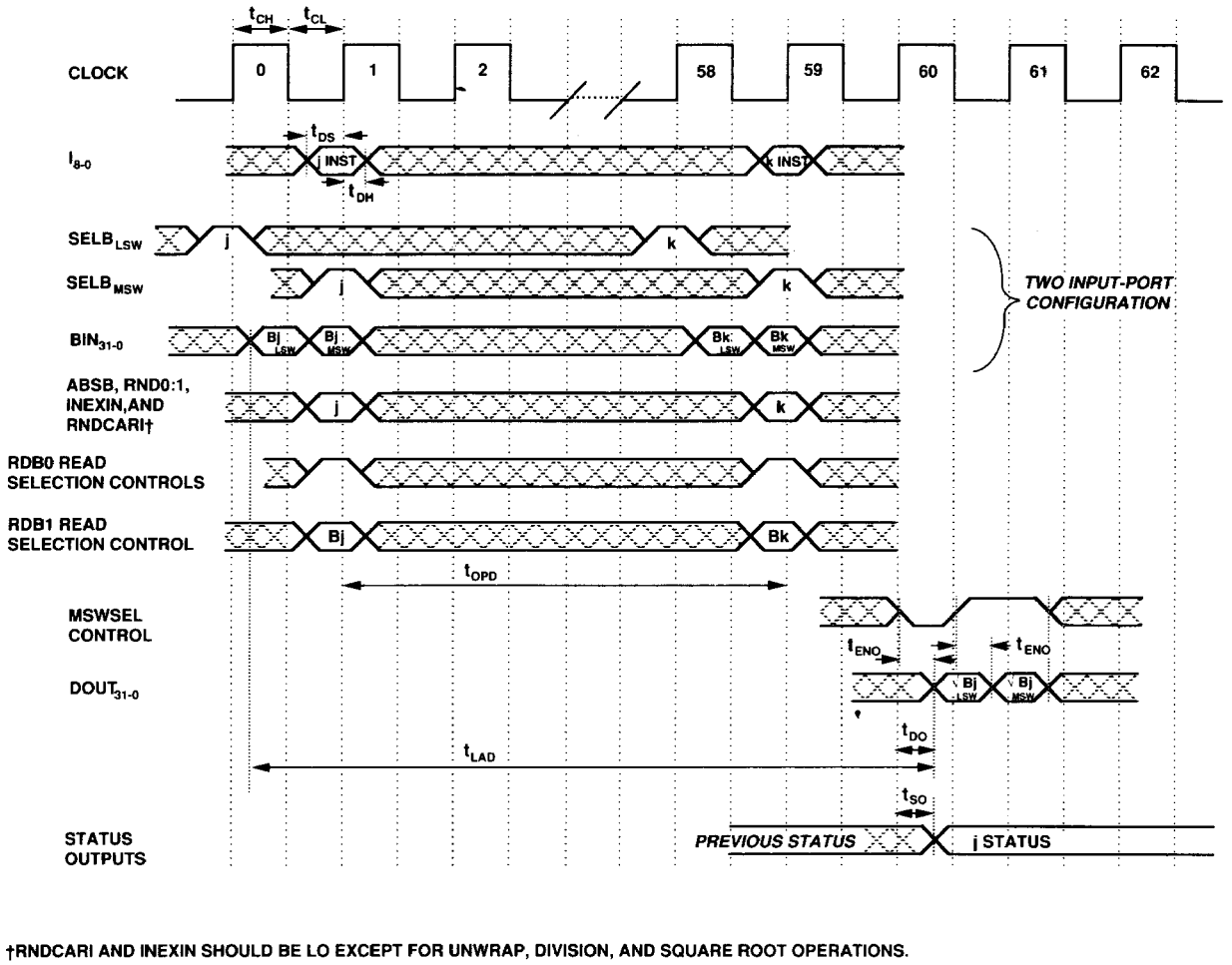



†RNDCARI AND INEXIN SHOULD BE LO EXCEPT FOR UNWRAP, DIVISION, AND SQUARE ROOT OPERATIONS.

*Figure 8. 32-Bit Single-Precision Floating-Point Square Root – Two Input-Port Configuration*



Downloaded from Partstack.com




†RNDCARI AND INEXIN SHOULD BE LO EXCEPT FOR UNWRAP, DIVISION, AND SQUARE ROOT OPERATIONS.

*Figure 9. 64-Bit Double-Precision Floating-Point Square Root – Two Input-Port Configuration*

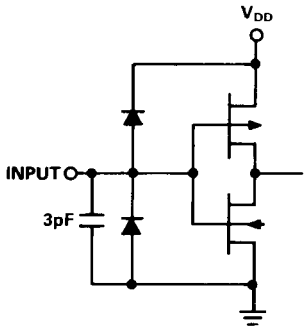



*Figure 10. Equivalent Input Circuits*

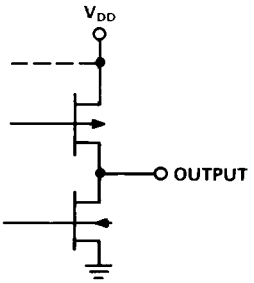



*Figure 11. Equivalent Output Circuits*

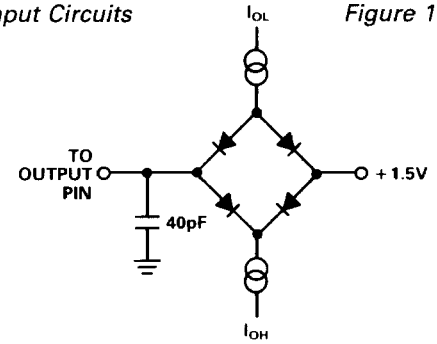



*Figure 12. Normal Load for AC Measurements*



Downloaded from Partstack.com

## 5.0 ADSP-3221 Floating-Point ALU Instructions.

For all operations involving two operands, the operands may be in SP Floating point format. DP Floating point, or 32-bit FIXED point, but both operands must be of the same data type.

Note that for all arithmetic operations (SP, DP, and FIXED), the ABSA and ABSB controls can be used with the input operands, causing the absolute value of the respective operand to be used in the operation.

### ADSP-3221 Fixed-Point Operation

| INSTRUCTION | OPCODE | | | | | | | | | COMMENT |
|---|---|---|---|---|---|---|---|---|---|---|
| | $I_8$ | $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | |
| IADD | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | Integer Add (A+B) |
| ISUBA | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | Integer Subtract (B−A) |
| ISUBB | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Integer Subtract (A−B) |
| IADDWC | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Integer Add with Carry (A+B)[1] |
| ISUBWBA | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | Integer Subtract with Borrow (B−A)[1] |
| ISUBWBB | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | Integer Subtract with Borrow (A−B)[1] |
| INEGA | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | Perform Twos Complement Negation of the A Operand |
| INEGB | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | Perform Twos Complement Negation of the B Operand |
| IADDAS | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | Absolute Value of (A+B) |
| ISUBAAS | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | Absolute Value of (B−A) |
| ISUBBAS | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | Absolute Value of (A−B) |

### ADSP-3221 Logical Operations

| INSTRUCTION | OPCODE | | | | | | | | | COMMENT |
|---|---|---|---|---|---|---|---|---|---|---|
| | $I_8$ | $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | |
| COMPL A | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | Ones Complement the A Operand |
| COMPL B | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | Ones Complement the B Operand |
| PASS A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Passes the A Operand (selected with RDA1, RDA0) unmodified to the output. |
| PASS B | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | Passes the B Operand (selected with RDB1, RDB0) unmodified to the output. |
| A AND B | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Bitwise AND |
| A OR B | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Bitwise OR |
| A XOR B | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Bitwise Exclusive OR |
| NOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | No Operation. Preserves status flags but does not preserve the contents of the output register. |
| CLR | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Clear. Sets the UNDFLO, OVERFLO, INVALOP, and INEXO flags to zero The Output Register result is not meaningful. |

NOTE
[1]The Carry bit in fixed point arithmetic is latched internally, for one-cycle only, to act as a Carry-In pin for the operations. Add with Carry or Subract with Borrow. These instructions are used in performing 64-bit addition or subtraction. See note on Special Instructions in the commercial data sheet.

*Table 2. ADSP-3221 Instruction Set*



Downloaded from Partstack.com

**ADSP-3221 Single-Precision Floating-Point Operations**

| INSTRUCTION | OPCODE | | | | | | | | | COMMENT |
|---|---|---|---|---|---|---|---|---|---|---|
| | $I_8$ | $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | |
| SADD | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | Floating Point SP Add (A+B) |
| SSUBB | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | Floating SP Subtract (A−B) |
| SSUBA | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Floating SP Subtract (B−A) |
| SCOMP | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | SP Compare. Operation is (A−B); Output Result is (A−B); Flags OVRFLO and UNDFLO generated conditional on Greater Than or Less Than conditions, respectively. If equal, output is zero and UNDFLO and OVRFLO are both LOW. |
| SADDAS | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | SP Absolute Value of (A+B) |
| SSUBBAS | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | SP Absolute Value of (B−A) |
| SSUBAAS | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | SP Absolute Value of (B−A) |
| FIX A | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | Convert 32-bit Single-Precision Floating-Point A operand→32-bit Integer |
| FIX B | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | Convert 32-bit Single-Precision Floating-Point B operand→32-bit Integer |
| FLOAT A | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | Convert 32-bit FIXED A operand→32-bit Single-Precision Floating-Point |
| FLOAT B | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | Convert 32-bit FIXED B operand→32-bit Single-Precision Floating-Point |
| DOUBLE A | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | Convert 32-bit Single-Precision A operand→64-bit Double-Precision Floating[1] |
| DOUBLE B | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | Convert 32-bit Single-Precision B operand→64-bit Double-Precision Floating[1] |
| SPASS A | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | Pass SP A operand. A NAN is passed unmodified but causes the INVALOP exception |
| SPASS B | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | Pass SP B operand. A NAN is passed unmodified but causes the INVALOP exception |
| SWRAP A | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | Converts SP denormalized A operand→Wrapped format[2] |
| SWRAP B | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | Converts SP denormalized B operand→Wrapped format[2] |
| SUNWRAP A | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Converts Wrapped A operand→SP denormalized format[2] |
| SUNWRAP B | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Converts Wrapped B operand→SP denormalized format[2] |
| SSIGN[3] | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | Transfers sign of the B operand to the A operand. Result is Sign B, Expt A, Fraction A. |
| SXSUB[3] | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | Exponent Subtract. Subtracts the B exponent from the A exponent. Result is Sign A, (Expt A − Expt B), Fraction A. NANs and Infinities are operated on directly without flagging the INVALOP exception. If the Expt of the Result underflows to −127 or less, then the Output Result returned is Zero. If the Expt of the Result overflows to +128, then the Output Result returned is Infinity. |
| SITRN[3] | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | Integer Shift and Truncation. The A mantissa with hidden bit restored (24 bits) is treated as a right-justified unsigned integer and shifted down logically by the unbiased B exponent.[4] The 32-bit right-justified result appears in the MSW and is a left-zero-filled unsigned-magnitude integer. The shift value given by the unbiased B exponent is interpreted as an unsigned number; negative unbiased exponents cause very large down shifts. |

*Table 2. ADSP-3221 Instruction Set (Continued)*



Downloaded from Partstack.com

**ADSP-3221 Single-Precision Floating-Point Operations (Continued)**

| INSTRUCTION | OPCODE | | | | | | | | | COMMENT |
|---|---|---|---|---|---|---|---|---|---|---|
| | $I_8$ | $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $\overline{I}_1$ | $I_0$ | |
| SDIV | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | SP Fltg Pt. (A÷B) |
| SSQR | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | SP Fltg Pt. $\sqrt{B}$ |

NOTES
[1]Conversion from Single to Double format are always exact, so INEXO will always be LOW for these conversions.
[2]For information on handling denormals, wrapped, and unwrapped numbers, see Application Note on "Handling IEEE Exceptions".
[3]See note on Special Instructions in the commercial data sheet.
[4]An unbiased exponent is the exponent after the proper exponent bias has been subtracted.

**ADSP-3221 Double-Precision Floating-Point Operations**

| INSTRUCTION | OPCODE | | | | | | | | | COMMENT |
|---|---|---|---|---|---|---|---|---|---|---|
| | $I_8$ | $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | |
| DADD | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | DP Add (A+B) |
| DSUBB | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | DP Subtract (A−B) |
| DSUBA | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | DP Subtract (B−A) |
| DCOMP | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | DP Compare. Operation is (A−B); Output Result is (A−B); Flags OVRFLO and UNDFLO generated conditional on Greater Than or Less Than conditions, respectively. If equal, output is zero and UNFLO and OVRFLO are both LOW. |
| DADDAS | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | DP Absolute Value of (A+B) |
| DSUBBAS | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | DP Absolute Value of (A−B) |
| DSUBAAS | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | DP Absolute Value of (B−A) |
| DFIX A | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | Convert 64-bit Double-Precision A operand→32-bit Integer |
| DFIX B | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | Convert 64-bit Double-Precision B operand→32-bit Integer |
| DFLOAT A[1] | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | Convert 32-bit FIXED A operand→64-bit Double-Precision Floating-Point |
| DFLOAT B[1] | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | Convert 32-bit FIXED B operand→64-bit Double-Precision Floating-Point |
| SINGLE A | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | Convert 64-bit Double-Precision A operand→32-bit Single-Precision |
| SINGLE B | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | Convert 64-bit Double-Precision B operand→32-bit Single-Precision |
| DPASSA | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Pass DP A operand. A NAN is passed unmodified but causes the INVALOP exception. |
| DPASS B | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Pass DP B operand. A NAN is passed unmodified but causes the INVALOP exception. |
| DWRAP A | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | Converts DP denormalized A operand→Wrapped format[2] |
| DWRAP B | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Converts DP denormalized B operand→Wrapped format[2] |
| DUNWRAP A | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Converts Wrapped A operand→DP denormalized format[2] |
| DUNWRAP B | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Converts Wrapped B operand→DP denormalized format[2] |

*Table 2. ADSP-3221 Instruction Set (Continued)*



Downloaded from Partstack.com

**ADSP-3221 Double-Precision Floating-Point Operations (Continued)**

| INSTRUCTION | OPCODE | | | | | | | | | COMMENT |
|---|---|---|---|---|---|---|---|---|---|---|
| | $I_8$ | $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | |
| DSIGN | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | Transfers sign of the B operand to the A operand. Result is Sign B, Expt A, Fraction A. |
| DXSUB | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | Exponent subtract. Subtracts the B exponent from the A exponent. Result is Sign A, (Expt A – Expt B), Fraction A. NANs and Infinities are operated on directly without flagging the INVALOP exception. If the Expt of the Result underflows to −1023 or less, then the Output Result returned is Zero. If the Expt of the Result overflows to +1024, then the Output Result returned is Infinity. |
| DITRN | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | Integer Shift and Truncation. The A mantissa with hidden bit restored (53 bits) is treated as a right-justified unsigned integer and shifted down logically by the unbiased B exponent.[3] The 32-bit right-justified result appears in the MSW and is a left-zero-filled unsigned-magnitude integer. The shift value given by the unbiased B exponent is interpreted as an unsigned number; negative unbiased exponents cause very large down shifts. |
| DDIV | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | DP Fltg Pt. (A÷B) |
| DSQR | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | DP Fltg Pt. $\sqrt{B}$ |

NOTES
[1]Conversion from FIXED to Double format are always exact. The 32-bit FIXED source operand must be fed from one of the registers A0. A2, B0 or B2 only.
[2]For information on handling denormals, wrapped and unwrapped numbers, see Application Note on "Handling IEEE Exceptions."
[3]An unbiased exponent is the exponent after the proper exponent bias has been subtracted.

*Table 2. ADSP-3221 Instruction Set*

**Notations Used in the Tables Below:**

| | | |
|---|---|---|
| RN | = | Round to the Nearest Number |
| RP | = | Round toward Plus Infinity |
| RM | = | Round toward Minus Infinity |
| RZ | = | Round toward Zero |
| NORM | = | Normalized Number |
| DNRM | = | Denormalized Number. A Denormalized number is treated as zero internally. |
| WNRM | = | Wrapped Number. A wrapped number is a number with a normalized fraction and an exponent that has been decremented through zero to take on a twos-complement negative value. |
| UNRM | = | Unnormalized Number. An Unnormal is an Underflowed and Wrapped Number. An UNRM can result from a multiplication of 1 or 2 Wrapped Numbers. |
| NORM.MAX | = | Maximum Normalized Number Representable in the Destination Format |
| NORM.MIN | = | Minimum Normalized Number Representable in the Destination Format |
| OVF | = | Overflowed Number |
| UNDF | = | Underflowed Number |
| INV | = | Invalid Operand |
| INF | = | Infinity |
| OK | = | No Exception Status Generated |


Downloaded from Partstack.com

In the tables below, the first mnemonic in each box describes the flag that is set, the second is the result on the DOUT pins.

| A operand \ B operand | ZERO | | DNRM | | NORM | | INF | | NAN | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* |
| **ZERO** | ZERO[2] | | DNRM | | NORM | | INF | | NAN | INVALOP |
| **DNRM** | DNRM | | NORM<br>DNRM<br>ZERO | | INF,NORM.MAX[1]<br>NORM<br>DNRM | OVRFLO | INF | | NAN | INVALOP |
| **NORM** | NORM | | INF,NORM.MAX[1]<br>NORM<br>DNRM | OVRFLO | INF,NORM.MAX[1]<br>NORM<br>DNRM<br>ZERO | OVRFLO | INF | | NAN | INVALOP |
| **INF** | INF | | INF | | INF | | INF[3]<br>NAN[3] | INVALOP | NAN | INVALOP |
| **NAN** | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP |

1. Either INF or NORM.MAX, depending on rounding mode. See "Round Controls."
2. (± ZERO)+ (± ZERO) = (±ZERO)–(∓ZERO) ⇒ ± ZERO
   (± ZERO)+(∓ ZERO) = (±ZERO)–(±ZERO)⇒ + ZERO (RN, RZ, RP rounding modes)
   (± ZERO)+(∓ ZERO) = (±ZERO)–(±ZERO)⇒ – ZERO (RM rounding mode)
3. (± INF)+(± INF) = (±INF)–(∓INF) ⇒ ± INF
   (± INF)+(∓ INF) = (±INF)–(±INF) ⇒ +NAN (RN, RZ, RP rounding modes)
   (± INF)+(∓ INF) = (±INF)–(±INF) ⇒ –NAN (RM rounding mode)
4. If DNRM result is inexact, UNDFLO will be set.

*Table 3. ADSP-3221 Floating-Point Additional Subtraction (IEEE Mode)*

| A operand \ B operand | ZERO | | DNRM | | NORM | | INF | | NAN | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* |
| **ZERO** | ZERO[2] | | ZERO | UNDFLO | NORM | | INF | | NAN | INVALOP |
| **DNRM** | ZERO | UNDFLO | NORM<br>ZERO | | INF,NORM.MAX[1]<br>NORM<br>ZERO | OVRFLO<br>UNDFLO | INF | | NAN | INVALOP |
| **NORM** | NORM | | INF,NORM.MAX[1]<br>NORM<br>ZERO | OVRFLO<br>UNDFLO | INF,NORM.MAX[1]<br>NORM<br>ZERO[4]<br>ZERO[4] | OVRFLO<br>UNDFLO | INF | | NAN | INVALOP |
| **INF** | INF | | INF | | INF | | INF[3]<br>NAN[3] | INVALOP | NAN | INVALOP |
| **NAN** | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP |

1. Either INF or NORM.MAX, depending on rounding mode. See "Round Controls."
2. (± ZERO)+ (± ZERO) = (±ZERO)–(∓ZERO) ⇒ ± ZERO
   (± ZERO)+(∓ ZERO) = (±ZERO)–(±ZERO)⇒ + ZERO (RN, RZ, RP rounding modes)
   (± ZERO)+(∓ ZERO) = (±ZERO)–(±ZERO)⇒ – ZERO (RM rounding mode)
3. (± INF)+(± INF) = (±INF)–(∓INF) ⇒ ± INF
   (± INF)+(∓ INF) = (±INF)–(±INF)⇒ +NAN (RN, RZ, RP rounding modes)
   (± INF)+(∓ INF) = (±INF)–(±INF) ⇒ –NAN (RM rounding mode)
4. Exact result.
5. In FAST mode, WRAP inputs are illegal.

*Table 4. ADSP-3221 Floating-Point Addition/Subtraction (FAST Mode)*



Downloaded from Partstack.com

| Sign | HB | f22 . . . f1 | f0 | | Unbiased Expnt | Source Name |
|---|---|---|---|---|---|---|
| 0 | 1 | X . . . X | X | 2** | 128 | +NAN |
| 0 | 1 | 0 . . . 0 | 0 | 2** | 128 | +INF |
| 0 | 1 | 0 . . . 0 | 0 | 2** | 31 | |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 23 | |
| 0 | 1 | 0 . . . 0 | 0 | 2** | 23 | |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 22 | |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 22 | |
| 0 | 1 | 0 . . . 0 | 0 | 2** | 0 | one |
| 0 | 1 | 1 . . . 1 | 1 | 2** | −1 | one − 1LSB |
| 0 | 1 | 1 . . . 1 | 1 | 2** | −1 | one − 1LSB |
| 0 | 1 | 0 . . . 0 | 1 | 2** | −1 | 1/2 + 1LSB |
| 0 | 1 | 0 . . . 0 | 1 | 2** | −1 | 1/2 + 1LSB |
| 0 | 1 | 0 . . . 0 | 0 | 2** | −1 | 1/2 |
| 0 | 1 | 0 . . . 0 | 0 | 2** | −1 | 1/2 |
| 0 | 1 | 0 . . . 0 | 0 | 2** | −126 | +NORM.MIN |
| 0 | 1 | 0 . . . 0 | 0 | 2** | −126 | +NORM.MIN |
| 0 | 0 | 0 . . . 0 | 1 | 2** | −126 | +DENORM.MIN |
| 0 | 0 | 0 . . . 0 | 1 | 2** | −126 | +DENORM.MIN |
| 0 | 0 | 0 . . . 0 | 0 | | 0 | +ZERO |
| 1 | 0 | 0 . . . 0 | 1 | 2** | −126 | −DENORM.MIN |
| 1 | 0 | 0 . . . 0 | 1 | 2** | −126 | −DENORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 2** | −126 | −NORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 2** | −126 | −NORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 2** | −1 | −1/2 |
| 1 | 1 | 0 . . . 0 | 0 | 2** | −1 | −1/2 |
| 1 | 1 | 0 . . . 0 | 1 | 2** | −1 | −1/2 − 1LSB |
| 1 | 1 | 0 . . . 0 | 1 | 2** | −1 | −1/2 − 1LSB |
| 1 | 1 | 1 . . . 1 | 1 | 2** | −1 | −one + 1LSB |
| 1 | 1 | 1 . . . 1 | 1 | 2** | −1 | −one + 1LSB |
| 1 | 1 | 0 . . . 0 | 0 | 2** | 0 | −one |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 22 | |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 22 | |
| 1 | 1 | 0 . . . 0 | 0 | 2** | 23 | |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 23 | |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 1 | 1 | 0 . . . 0 | 0 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 1 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 2** | 128 | −INF |
| 1 | 1 | X . . . X | X | 2** | 128 | −NAN |

| Sign | i30 | i29 | i28 | i27 | i26 | i25 | i24 | i23 | i22 . . . i7 | i6 | i5 | i4 | i3 | i2 | i1 | i0 | Rounding Modes | Status Flags |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all | INVALOP |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all | INVALOP |
| U* | U | U | U | U | U | U | U | U | U . . . U | U | U | U | U | U | U | U | all | OVRFLO |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | all | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | all | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RN,RP | INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RZ,RM | INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | all | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | RN, RP | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RZ,RM | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | RN,RP | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RZ,RM | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | RP | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RM,RN,RZ | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | RP | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RM,RN,RZ | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | RP | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RM,RN,RZ | UNDFLO,INEXO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | all | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RM | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RP,RN,RZ | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RM | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RP,RN,RZ | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RM | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RP,RN,RZ | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RM,RN | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RP,RZ | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RM,RN | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RP,RZ | UNDFLO,INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RM,RN | INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | RP,RZ | INEXO |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | all | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | all | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | all | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 . . . 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | all | |
| U | U | U | U | U | U | U | U | U | U . . . U | U | U | U | U | U | U | U | all | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all | INVALOP |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 . . . 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all | INVALOP |

*"U" denotes an undefined result.

*Table 5. Conversion of 32-Bit Single-Precision Floating-Point to 32-Bit Twos-Complement Integer*



Downloaded from Partstack.com

| Sign | HB | f51 . . . f22 | f21 | f20 | f19 . . . f1 | f0 | | Unbiased Expnt | Source Name |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | X . . . X | X | X | X . . . X | X | 2** | 1024 | +NAN |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 1024 | +INF |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 31 | |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 0 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 0 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 1 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 1 | 2** | 30 | |
| 0 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 0 | 2** | 30 | |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 0 | one |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1 | one − 1LSB |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1 | one − 1LSB |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1 | 1/2 + 1LSB |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1 | 1/2 + 1LSB |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1 | 1/2 |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1 | 1/2 |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | +NORM.MIN |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | +NORM.MIN |
| 0 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | +DENORM.MIN |
| 0 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | +DENORM.MIN |
| 0 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | | 0 | +ZERO |
| 1 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | −DENORM.MIN |
| 1 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | −DENORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | −NORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | −NORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1 | −1/2 |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1 | −1/2 |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1 | −1/2 − 1LSB |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1 | −1/2 − 1LSB |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1 | −one + 1LSB |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1 | −one + 1LSB |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 0 | −one |
| 1 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 0 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 1 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 1 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 0 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 0 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 30 | |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 1 | 0 . . . 0 | 0 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 1 | 0 . . . 0 | 0 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 1 | 1 . . . 1 | 1 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 1 | 1 . . . 1 | 1 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 1 | 0 | 0 . . . 0 | 0 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 1 | 0 | 0 . . . 0 | 1 | 2** | 31 | |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 32 | |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 1024 | −INF |
| 1 | 1 | X . . . X | X | X | X . . . X | X | 2** | 1024 | −NAN |

| Sign | i30 . . . i1 | i0 | Rounding Modes | Status Flags |
|---|---|---|---|---|
| 0 | 1 . . . 1 | 1 | all | INVALOP |
| 0 | 1 . . . 1 | 1 | all | INVALOP |
| U* | U . . . U | U | all | OVRFLO |
| U | U . . . U | U | RP,RN | OVRFLO,INEXO |
| 0 | 1 . . . 1 | 1 | RZ,RM | INEXO |
| U | U . . . U | U | RP,RN | OVRFLO,INEXO |
| 0 | 1 . . . 1 | 1 | RZ,RM | INEXO |
| U | U . . . U | U | RP | OVRFLO,INEXO |
| 0 | 1 . . . 1 | 1 | RM,RN,RZ | INEXO |
| U | U . . . U | U | RP | OVRFLO,INEXO |
| 0 | 1 . . . 1 | 1 | RM,RN,RZ | INEXO |
| 0 | 1 . . . 1 | 1 | all | |
| 0 | 0 . . . 0 | 1 | all | |
| 0 | 0 . . . 0 | 1 | RN,RP | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RZ,RM | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 1 | RN,RP | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RZ,RM | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 1 | RP | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RM,RN,RZ | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 1 | RP | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RM,RN,RZ | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 1 | RP | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RM,RN,RZ | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | all | |
| 1 | 1 . . . 1 | 1 | RM | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RP,RN,RZ | UNDFLO,INEXO |
| 1 | 1 . . . 1 | 1 | RM | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RP,RN,RZ | UNDFLO,INEXO |
| 1 | 1 . . . 1 | 1 | RM | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RP,RN,RZ | UNDFLO,INEXO |
| 1 | 1 . . . 1 | 1 | RM,RN | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RP,RZ | UNDFLO,INEXO |
| 1 | 1 . . . 1 | 1 | RM,RN | UNDFLO,INEXO |
| 0 | 0 . . . 0 | 0 | RP,RZ | UNDFLO,INEXO |
| 1 | 1 . . . 1 | 1 | all | |
| 1 | 0 . . . 0 | 1 | all | |
| 1 | 0 . . . 0 | 0 | RM | INEXO |
| 1 | 0 . . . 0 | 1 | RP,RN,RZ | INEXO |
| 1 | 0 . . . 0 | 0 | RM | INEXO |
| 1 | 0 . . . 0 | 1 | RP,RN,RZ | INEXO |
| 1 | 0 . . . 0 | 0 | RM,RN | INEXO |
| 1 | 0 . . . 0 | 1 | RP,RZ | INEXO |
| 1 | 0 . . . 0 | 0 | RM,RN | INEXO |
| 1 | 0 . . . 0 | 1 | RP,RZ | INEXO |
| 1 | 0 . . . 0 | 0 | all | |
| 1 | 0 . . . 0 | 0 | RP,RN,RZ | INEXO |
| U | U . . . U | U | RM | OVRFLO,INEXO |
| 1 | 0 . . . 0 | 0 | RP,RZ | INEXO |
| U | U . . . U | U | RM,RN | OVRFLO,INEXO |
| 1 | 0 . . . 0 | 0 | RP,RZ | INEXO |
| U | U . . . U | U | RM,RN | OVRFLO,INEXO |
| U | U . . . U | U | all | OVRFLO |
| U | U . . . U | U | all | OVRFLO,INEXO |
| U | U . . . U | U | all | OVRFLO |
| 1 | 1 . . . 1 | 1 | all | INVALOP |
| 1 | 1 . . . 1 | 1 | all | INVALOP |

*"U" denotes an undefined result.

NOTE: Heavy line indicates rounding boundary in source.

*Table 6. Conversion of 64-Bit Double-Precision Floating-Point to 32-Bit Twos-Complement Integer*



Downloaded from Partstack.com

| Sign | HB | f51 . . . f30 | f29 | f28 | f27 . . . f1 | f0 | | Unbiased Expnt | Source Name |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | X . . . X | X | X | X . . . X | X | 2** | 1024 | + NAN |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 1024 | + INF |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 1023 | + NORM.MAX |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 1023 | + NORM.MAX |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 0 . . . 0 | 0 | 2** | 127 | |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 0 . . . 0 | 0 | 2** | 127 | |
| 0 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 1 | 2** | 127 | |
| 0 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 1 | 2** | 127 | |
| 0 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | 127 | |
| 0 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 0 | 2** | 127 | |
| 0 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 1 | 2** | 127 | |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −126 | |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −127 | |
| 0 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −127 | |
| 0 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | −127 | |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −149 | |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | + NORM.MIN |
| 0 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | + NORM.MIN |
| 0 | 0 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1022 | + DNRM.MAX |
| 0 | 0 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1022 | + DNRM.MAX |
| 0 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | + DNRM.MIN |
| 0 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | + DNRM.MIN |
| 0 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | | 0 | + ZERO |
| 1 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | | 0 | − ZERO |
| 1 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | − DNRM.MIN |
| 1 | 0 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 1 | 2** | −1022 | − DNRM.MIN |
| 1 | 0 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1022 | − DNRM.MAX |
| 1 | 0 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −1022 | − DNRM.MAX |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | − NORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −1022 | − NORM.MIN |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −149 | |
| 1 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 0 | 2** | −127 | |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −127 | |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | −127 | |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | −126 | |
| 1 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 1 | 2** | 127 | |
| 1 | 1 | 1 . . . 1 | 0 | 0 | 0 . . . 0 | 1 | 2** | 127 | |
| 1 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | 127 | |
| 1 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 1 | 2** | 127 | |
| 1 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 1 | 2** | 127 | |
| 1 | 1 | 1 . . . 1 | 1 | 0 | 0 . . . 0 | 0 | 2** | 127 | |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 0 . . . 0 | 0 | 2** | 127 | |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 1023 | − NORM.MAX |
| 1 | 1 | 1 . . . 1 | 1 | 1 | 1 . . . 1 | 1 | 2** | 1023 | − NORM.MAX |
| 1 | 1 | 0 . . . 0 | 0 | 0 | 0 . . . 0 | 0 | 2** | 1024 | − INF |
| 1 | 1 | X . . . X | X | X | X . . . X | X | 2** | 1024 | − NAN |

| Sign | HB | f22 . . . f1 | f0 | | Unbiased Expnt | Result Name | Rounding Modes | Status Flags |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 . . . 1 | 1 | 2** | 128 | + NAN | all | INVALOP |
| 0 | 1 | 0 . . . 0 | 0 | 2** | 128 | + INF | all | |
| 0 | 1 | 0 . . . 0 | 0 | 2** | 128 | + INF | RP,RN | OVRFLO,INEXO |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 127 | + NORM.MAX | RZ,RM | OVRFLO,INEXO |
| 1 | 1 | 1 . . . 0 | 0 | 2** | 127 | + INF | RP,RN | OVRFLO,INEXO |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 127 | + NORM.MAX | RZ,RM | INEXO |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 127 | + INF | RP | OVRFLO,INEXO |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 127 | + NORM.MAX | RM,RN,RZ | INEXO |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 127 | + NORM.MAX | all | |
| 0 | 1 | 1 . . . 1 | 1 | 2** | 127 | + NORM.MAX | RP | INEXO |
| 0 | 1 | 1 . . . 1 | 0 | 2** | 127 | | RM,RN,RZ | INEXO |
| 0 | 1 | 0 . . . 0 | 0 | 2** | −126 | + NORM.MIN | all | |
| 0 | 0 | 1 . . . 1 | 1 | 2** | −126 | + NORM.MIN | RP,RN | INEXO |
| 0 | 0 | 1 . . . 1 | 1 | 2** | −126 | + DNRM.MAX | RZ,RM | UNDFLO,INEXO |
| 0 | 0 | 1 . . . 1 | 1 | 2** | −126 | + DNRM.MAX | all | |
| 0 | 0 | 0 . . . 0 | 1 | 2** | −126 | + DNRM.MIN | all | |
| 0 | 0 | 0 . . . 0 | 1 | 2** | −126 | + DNRM.MIN | RP | UNDFLO, INEXO |
| 0 | 0 | 0 . . . 0 | 0 | 2** | −126 | + ZERO | RM,RN,RZ | UNDFLO, INEXO |
| 0 | 0 | 0 . . . 0 | 1 | 2** | −126 | + DNRM.MIN | RP | UNDFLO, INEXO |
| 0 | 0 | 0 . . . 0 | 0 | 2** | −126 | + ZERO | RM,RN,RZ | UNDFLO, INEXO |
| 0 | 0 | 0 . . . 0 | 1 | 2** | −126 | + DNRM.MIN | RP | UNDFLO, INEXO |
| 0 | 0 | 0 . . . 0 | 0 | 2** | −126 | + ZERO | RM,RN,RZ | UNDFLO, INEXO |
| 0 | 0 | 0 . . . 0 | 0 | | 0 | + ZERO | all | |
| 1 | 0 | 0 . . . 0 | 0 | | 0 | − ZERO | all | |
| 1 | 0 | 0 . . . 0 | 1 | 2** | −126 | − DNRM.MIN | RM | UNDFLO, INEXO |
| 1 | 0 | 0 . . . 0 | 0 | 2** | −126 | − ZERO | RP,RN,RZ | UNDFLO, INEXO |
| 1 | 0 | 0 . . . 0 | 1 | 2** | −126 | − DNRM.MIN | RM | UNDFLO, INEXO |
| 1 | 0 | 0 . . . 0 | 0 | 2** | −126 | − ZERO | RP,RN,RZ | UNDFLO, INEXO |
| 1 | 0 | 0 . . . 0 | 1 | 2** | −126 | − DNRM.MIN | RM | UNDFLO, INEXO |
| 1 | 0 | 0 . . . 0 | 0 | 2** | −126 | − ZERO | RP,RN,RZ | UNDFLO, INEXO |
| 1 | 0 | 0 . . . 0 | 1 | 2** | −126 | − DNRM.MIN | all | |
| 1 | 0 | 1 . . . 1 | 1 | 2** | −126 | − DNRM.MAX | all | |
| 1 | 0 | 1 . . . 1 | 1 | 2** | −126 | − NORM.MIN | RM,RN | INEXO |
| 1 | 0 | 1 . . . 1 | 1 | 2** | −126 | − DNRM.MAX | RP,RZ | UNDFLO,INEXO |
| 1 | 1 | 0 . . . 0 | 0 | 2** | −126 | − NORM.MIN | all | |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 127 | − NORM.MAX | RM | INEXO |
| 1 | 1 | 1 . . . 1 | 0 | 2** | 127 | | RP,RN,RZ | INEXO |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 127 | − NORM.MAX | all | |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 127 | − INF | RM | OVRFLO,INEXO |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 127 | − NORM.MAX | RP,RN,RZ | INEXO |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 127 | − INF | RM,RN | OVRFLO,INEXO |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 127 | − NORM.MAX | RP,RZ | INEXO |
| 1 | 1 | 0 . . . 0 | 0 | 2** | 128 | − INF | RM,RN | OVRFLO,INEXO |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 127 | − NORM.MAX | RP,RZ | OVRFLO,INEXO |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 128 | − INF | all | |
| 1 | 1 | 1 . . . 1 | 1 | 2** | 128 | − NAN | all | INVALOP |

NOTE: Heavy line indicates rounding boundary in source.

*Table 7. Conversion of 64-Bit Double-Precision Floating-Point to 32-Bit Single-Precision Floating Point (IEEE Mode)*



Downloaded from Partstack.com

| A operand \ B operand | ZERO | | DNRM | | WRAP | | NORM | | INF | | NAN | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* |
| **ZERO** | NAN | INVALOP | ZERO | | ZERO | | ZERO | | ZERO | | NAN | INVALOP |
| **DNRM** | INF[1] | OVRFLO& INVALOP | NAN | UNDFLO& INVALOP | NAN | UNDFLO INVALOP | NAN | UNDFLO INVALOP | ZERO | | NAN | INVALOP |
| **WRAP** | INF[1] | OVRFLO& INVALOP | NAN | UNDFLO& INVALOP | NORM | | NORM WRAP UNRM | UNDFLO UNDFLO | ZERO | | NAN | INVALOP |
| **NORM** | INF[1] | OVRFLO& INVALOP | NAN | UNDFLO& INVALOP | INF,NORM.MAX[1] NORM | OVRFLO | INF,NORM.MAX[1] NORM WRAP UNRM | OVRFLO UNDFLO UNDFLO | ZERO | | NAN | INVALOP |
| **INF** | INF | | INF | | INF | | INF | | NAN | INVALOP | NAN | INVALOP |
| **NAN** | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP |

1. Either INF or NORM.MAX, depending on rounding mode. In FAST mode, WRAP inputs are illegal.

*Table 8. ADSP-3221 Floating-Point Division (A ÷ B) (IEEE Mode)*

| A operand \ B operand | ZERO | | DNRM | | NORM | | INF | | NAN | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* |
| **ZERO** | NAN | INVALOP | NAN | INVALOP | ZERO | | ZERO | | NAN | INVALOP |
| **DNRM** | NAN | INVALOP | NAN | INVALOP | ZERO | | ZERO | | NAN | INVALOP |
| **NORM** | INF[1] | OVRFLO& INVALOP | INF[1] | OVRFLO& INVALOP | INF,NORM.MAX[1] NORM ZERO | OVRFLO UNDFLO | ZERO | | NAN | INVALOP |
| **INF** | INF | | INF | | INF | | NAN | INVALOP | NAN | INVALOP |
| **NAN** | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP | NAN | INVALOP |

1. Either INF or NORM.MAX, depending on rounding mode.

*Table 9. ADSP-3221 Floating-Point Division (A ÷ B) (FAST Mode)*

| Mode \ B operand | B<ZERO | | ±ZERO | | +DNRM | | +WRAP | | +NORM | | +INF | | ±NAN | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* | *result* | *status* |
| **IEEE** | –NAN | INVALOP | ±ZERO | | +NAN | UNDFLO& INVALOP | NORM | | NORM | | +INF | | ±NAN | INVALOP |
| **FAST** | –NAN | INVALOP | ±ZERO | | +ZERO | | NORM | | NORM | | +INF | | ±NAN | INVALOP |

*Table 10. ADSP-3221 Floating-Point Square Root ($\sqrt{B}$)*



Downloaded from Partstack.com

**Artisan Technology Group** is an independent supplier of quality pre-owned equipment

## Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

## We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

## Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP