National Semiconductor

PRELIMINARY

NS32580-20/NS32580-25/NS32580-30 Floating Point Controller

General Description

The NS32580 Floating-Point Controller (FPC) is an interface device designed to couple the NS32532 Microprocessor with the Weitek WTL 3164 Floating-Point Data Path (FPDP). It is a new member of the Series 32000® family and it is fully upward compatible with the existing NS32081 floating-point software. Its performance reaches a peak of 10 Mflops when executing single and double precision ADD, SUB, MUL, and MAC instructions in a pipelined mode.

The FPC/FPDP supports the IEEE 754—1985 standard for Binary Floating-Point Arithmetic. An improved exception handling scheme allows enabling or disabling of each of the IEEE defined traps.

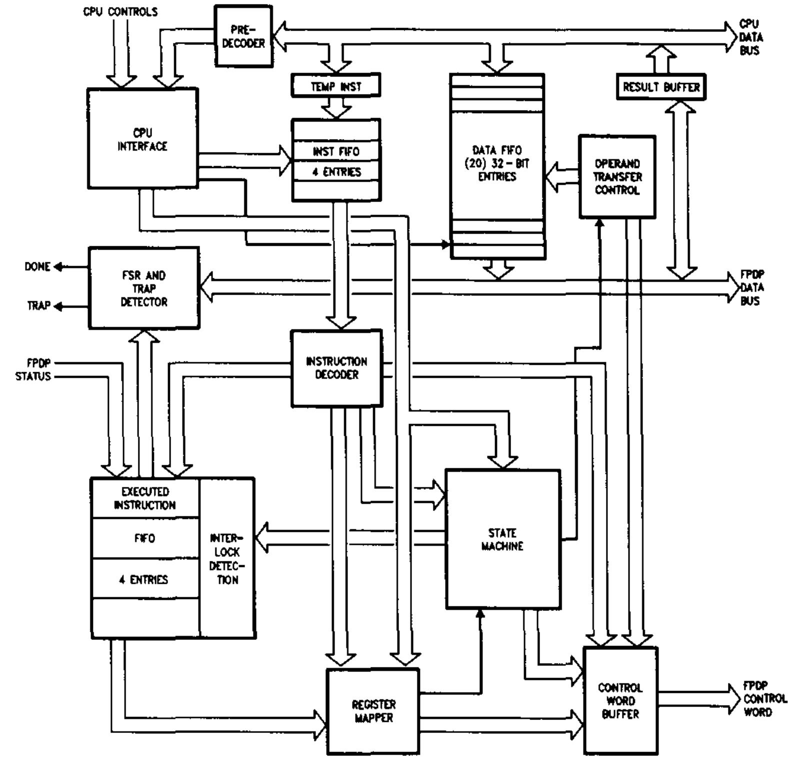
The NS32580 contains three FIFOs and a Floating-Point Status Register (FSR). It executes 18 instructions in conjunction with the WTL 3164 and with the NS32532 forms a tightly coupled computer cluster. The FPC/FPDP appears to the user as a single slave processing unit. The CPU and FPC/FPDP communication is handled automatically, and is user transparent.

The FPC is fabricated with National's advanced double-metal CMOS process and can operate at a frequency of 30 MHz.

Features

- Provides the NS32532 CPU with a complete interface controller for high-speed floating-point arithmetic
- 10 Mflops peak performance for single and double precision ADD, SUB, MUL and MAC instructions with the Weitek WTL 3164 FPDP
- Floating-point format compatible with IEEE 754—1985 standard
- Pipelined Slave Protocol with Data and Instruction FIFOs
- Improved exception handling including support of Infinities and Not a Number (NaN)
- Single (32-bit) and double (64-bit) precision operations
- Upward compatible with existing NS32081 software base
- 20 MHz, 25 MHz and 30 MHz operating frequencies
- 1 μm double-metal CMOS technology
- 172-pin PGA package

Block Diagram


CPU CONTROLS
PRE-DECODER
CPU DATA BUS
TEMP INST
RESULT BUFFER
CPU INTERFACE
INST FIFO
4 ENTRIES
DATA FIFO (20) 32-BIT ENTRIES
OPERAND TRANSFER CONTROL
DONE
TRAP
FSR AND TRAP DETECTOR
FPDP DATA BUS
FPDP STATUS
INSTRUCTION DECODER
EXECUTED INSTRUCTION
FIFO
4 ENTRIES
INTER-LOCK DETEC-TION
STATE MACHINE
REGISTER MAPPER
CONTROL WORD BUFFER
FPDP CONTROL WORD

TL/EE/9421-1

FIGURE 1-1

Table of Contents

1.0 PRODUCT INTRODUCTION
  1.1 IEEE Features Supported
  1.2 Operand Formats
    1.2.1 Normalized Numbers
    1.2.2 Zero
    1.2.3 Reserved Operands
    1.2.4 Integer Formats
    1.2.5 Memory Representations

2.0 ARCHITECTURAL DESCRIPTION
  2.1 Programming Model
    2.1.1 Floating-Point Data Registers
    2.1.2 Floating-Point Status Register (FSR)
      2.1.2.1 FSR Mode Control Fields
      2.1.2.2 FSR Status Fields
      2.1.2.3 FSR Software Field (SWF)
      2.1.2.4 FSR New Fields
      2.1.2.5 FSR Default Values
  2.2 Instruction Set
    2.2.1 Floating Point Instruction Set
  2.3 Exceptions

3.0 FUNCTIONAL DESCRIPTION
  3.1 Power and Grounding
  3.2 Clocking
  3.3 Resetting
  3.4 Bus Operation
    3.4.1 Operand Transfers
  3.5 Instruction Protocols
    3.5.1 General Protocol Sequence
    3.5.2 Byte Sex
    3.5.3 Floating-Point Instruction Protocols

3.0 FUNCTIONAL DESCRIPTION (Continued)
  3.6 FPDP Interface
    3.6.1 Controlling the FPDP
    3.6.2 Instruction Control
    3.6.3 "2 Cycle Mode" and "3 Cycle Mode"
    3.6.4 FPDP Mode Control Registers SR0, SR1
    3.6.5 IEEE Enables Register SR2
      3.6.5.1 FPDP Status Lines (S0–S3)
    3.6.6 FPDP Clocking Requirements

4.0 DEVICE SPECIFICATIONS
  4.1 NS32580 Pin Descriptions
    4.1.1 Supplies
    4.1.2 Input Signals
    4.1.3 Output Signals
    4.1.4 Input/Output Signals
  4.2 Absolute Maximum Ratings
  4.3 Electrical Characteristics
  4.4 Switching Characteristics
    4.4.1 Definitions
    4.4.2 Timing Tables
      4.4.2.1 Output Signal Propagation Delays
      4.4.2.2 Input Signal Requirements

APPENDIX A: Compatibility of FPC-FPDP with NS32081/NS32381

APPENDIX B: Performance Analysis

List of Illustrations

FPC Block Diagram 1-1
Floating-Point Operand Formats 1-2
Single-Precision Operand E and F Fields 1-3
Double-Precision Operand E and F Fields 1-4
Integer Format 1-5
Data Registers 2-1
FSR (Compatible Fields) 2-2
New FSR Mode Control Fields 2-3
Floating-Point Instruction Formats 2-4
Recommended Supply Connections 3-1
Power-On Reset Requirements 3-2
General Reset Timing 3-3
Slave Processor Read Cycle from FPC 3-4
Slave Processor Write Cycle to FPC 3-5
System Connection Diagram 3-6
ID and Operation Word 3-7
FPC Status Word 3-8
Floating-Point Instruction Processing Flowchart 3-9
Byte Sex Connection Diagrams 3-10
FPDP Control Word 3-11
FPDP Multiplier and ALU Bus Control 3-12
IEEE Enables Register (FPDP) 3-13
FPDP Status Timing 3-14
Divide/Sqrt Clock DCLK2/DCLK3 3-15
NS32580 Interface Signals 4-1
172-Pin PGA Package 4-2
Timing Specification Standard (Signal Valid after Clock Edge) 4-3
Timing Specification Standard (Signal Valid before Clock Edge) 4-4
Clock Waveforms 4-5
Power-On Reset 4-6
Non-Power-On Reset 4-7
Read Cycle from FPC 4-8
Write Cycle to FPC 4-9
Slave Processor Done Timing 4-10
$\overline{\text{FSSR}}$ Signal Timing 4-11
FPDP Status Signal Timing 4-12
FPDP Clock Signals Timing 4-13
FPDP Output Signals Timing 4-14

List of Tables

Sample F Fields 1-1
Sample E Fields 1-2
Normalized Number Ranges 1-3
Integer Fields 1-4
FSR Default State Summary 2-1
Exception Handling Summary 2-2
Floating-Point Instruction Sequence 3-1
Floating-Point Instruction Protocols 3-2

1.0 Product Introduction

The NS32580 Floating-Point Controller (FPC) provides complete control for high speed floating-point operations between the NS32532 CPU and the Weitek WTL 3164 Floating-Point Data Path (FPDP). The FPC is fabricated using National high-speed CMOS technology and operates as a slave processor for transparent expansion of the Series 32000 CPU's basic instruction set. The NS32580 is compatible with the IEEE Floating-Point Formats by means of its hardware and software features.

1.1 IEEE FEATURES SUPPORTED

a. Basic floating-point number formats
b. Add, subtract, multiply, divide, sqrt, and compare operations
c. Conversions between different floating-point formats
d. Conversions between floating-point and integer formats
e. Round floating-point number to integer (round to nearest, round toward negative infinity and round toward zero, in double- or single-precision)
f. Exception signaling and handling (invalid operation, divide by zero, overflow, underflow and inexact)
g. Positive and negative infinity (Section 1.2.3)

Note: In addition to supporting the IEEE floating-point overflow, the NS32580 supports integer conversion overflow.

Also, the FPC-FPDP can accept Not-a-Number (NaN) as an operand and generate NaN as a result, but it does not conform to the IEEE 754-1985 Standard since it does not differentiate between signaling and quit Not-a-Number.

Note 1: ABSf NaN and NEGf NaN result in a signaling NaN but not a QUIT NaN.

Note 2: For NaN Op DNRM, where Op is ADDf, SUBf, MULf, DIVf and MACf, and with ROE = 1, the result is a QUIT NaN and not TRAP (INV).

Note 3: If ROE = 1, IVE = 0 and the operand is signalling NaN, the result is NaN with no TRAP (INV).

The remaining IEEE features can be supported in the software library. These items include:

a. Extended floating-point number formats
b. Mixed floating-point data formats
c. Conversions between basic formats, floating-point numbers and decimal strings
d. Remainder
e. Denormalized numbers

1.2 OPERAND FORMATS

The NS32580 FPC operates on two floating-point data types—single precision (32 bits) and double precision (64 bits). Floating-point instruction mnemonics use the suffix F (Floating) to select the single precision data type, and the suffix L (Long Floating) to select the double precision data type.

A floating-point number is divided into three fields, as shown in *Figure 1-2*.

The F field is the fractional portion of the represented number. In Normalized numbers (Section 1.2.1), the binary point is assumed to be immediately to the left of the most significant bit of the F field, with an implied 1 bit to the left of the binary point. Thus, the F field represents values in the range $1.0 \le x < 2.0$, as shown in Table 1-1.

TABLE 1-1. Sample F Fields

F Field	Binary Value	Decimal Value
000 . . . 0	1.000 . . . 0	1.000 . . . 0
010 . . . 0	1.010 . . . 0	1.250 . . . 0
100 . . . 0	1.100 . . . 0	1.500 . . . 0
110 . . . 0	1.110 . . . 0	1.750 . . . 0

↑
Implied Bit

The E field contains an unsigned number that gives the binary exponent of the represented number. The value in the E field is biased; that is, a constant bias value must be subtracted from the E field value in order to obtain the true exponent. The bias value is $011 \ldots 11_2$, which is either 127 (single precision) or 1023 (double precision). Thus, the true exponent can be either positive or negative, as shown in Table 1-2.

TABLE 1-2. Sample E Fields

E Field	F Field	Represented Value
011 . . . 110	100 . . . 0	$1.5 \times 2^{-1} = 0.75$
011 . . . 111	100 . . . 0	$1.5 \times 2^{0} = 1.50$
100 . . . 000	100 . . . 0	$1.5 \times 2^{1} = 3.00$

Two values of the E field are not exponents. 11 . . . 11 signals Not-a-Number (NaN) or Infinity (Section 1.2.3). 00 . . . 00 represents the number zero (Section 1.2.2), if the F field is also all zeroes, otherwise it signals a reserved operand (Section 1.2.4).

The S bit indicates the sign of the operand. It is 0 for positive and 1 for negative. Floating-point numbers are in sign-magnitude form, that is, only the S bit is complemented in order to change the sign of the represented number.

1.2.1 Normalized Numbers

Normalized numbers are numbers which can be expressed as floating-point operands, as described above, where the E field is neither all zeroes nor all ones.

The value of a Normalized number can be derived by the formula:

$$(-1)^{S} \times 2^{(E\text{-Bias})} \times (1 + F)$$

The range of Normalized numbers is given in Table 1-3.

1.2.2 Zero

There are two representatives for zero—positive and negative. Positive zero has all-zero F and E fields, and the S bit is zero. Negative zero also has all-zero F and E fields, but its S bit is one.

1.0 Product Introduction (Continued)

1.2.3 Reserved Operands

Infinity arithmetic is the limiting case of real arithmetic with operands of arbitrarily large magnitudes. The NS32580 does not treat infinity as a reserved operand and in ROUNDfi, TRUNCfi and FLOORfi instructions, when the operand is infinity, the FPC will return the TRAP "Integer overflow" instead of TRAP "Invalid Operation" with the Integer Conversion Overflow Flag, IOF, set to "1" and the Trap type to "2".

Another special case regarding infinity occurs when dividing infinity by zero. In this case NO TRAP "Divide by Zero" will be signaled and infinity will be returned as the result. See *Figures 1-3* and *1-4*.

The NS32580 FPC can treat NaN, not a number, either as a reserved operand (in NS32081 compatibility mode) or as not a reserved operand, depending upon the setting of the FSR ROE bit.

Denormalized numbers have all zeroes in their E fields and non-zero values in their F fields. They are treated as reserved operands except for those special cases listed in the compatibility table of Appendix A.

The NS32580 FPC causes an Invalid Operation Trap (Section 2.1.2.2) if it receives a reserved operand, unless the operation is simply a move (without conversion).

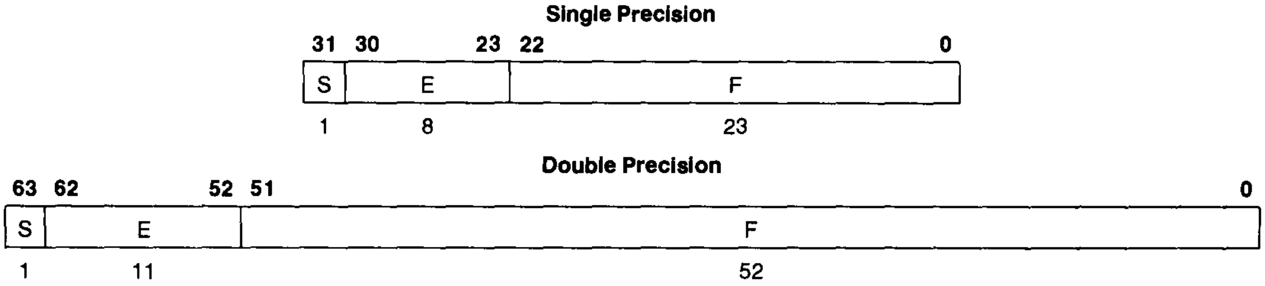


FIGURE 1-2. Floating-Point Operand Formats

TABLE 1-3. Normalized Number Ranges

	Single Precision	Double Precision
Most Positive	$2^{127} \times (2 - 2^{-23})$ $= 3.40282346 \times 10^{38}$	$2^{1023} \times (2 - 2^{-52})$ $= 1.7976931348623157 \times 10^{308}$
Least Positive	2^{-126} $= 1.17549436 \times 10^{-38}$	2^{-1022} $= 2.2250738585072014 \times 10^{-308}$
Least Negative	$-(2^{-126})$ $= -1.17549436 \times 10^{-38}$	$-(2^{-1022})$ $= -2.2250738585072014 \times 10^{-308}$
Most Negative	$-2^{127} \times (2 - 2^{-23})$ $= -3.40282346 \times 10^{38}$	$-2^{1023} \times (2 - 2^{-52})$ $= -1.7976931348623157 \times 10^{308}$

Note: The values given are extended one full digit beyond their represented accuracy to help in generating rounding and conversion algorithms.

E	F	Value	Name	Comments
255	Not 0	None	*NaN	ROE = 0 → Reserved Operand ROE = 1 → NaN Returned as Result
255	0	$(-1)^s$ * Infinity	*Infinity	Not a Reserved Operand
1–254	Any	$(-1)^s * 2^{e-127} * (1.f)$	Normalized Number	
0	Not 0	$(-1)^s * 2^{-126} * (0.f)$	*Denormalized Number	Reserved Operand
0	0	$(-1)^s * 0$	Zero	

FIGURE 1-3. Single-Precision Operand E and F Fields

E	F	Value	Name	Comments
2047	Not 0	None	*NaN	ROE = 0 → Reserved Operand ROE = 1 → NaN Returned as Result
2047	0	$(-1)^s$ * Infinity	*Infinity	Not a Reserved Operand
1–2046	Any	$(-1)^s * 2^{e-1023} * (1.f)$	Normalized Number	
0	Not 0	$(-1)^s * 2^{-1022} * (0.f)$	*Denormalized Number	Reserved Operand
0	0	$(-1)^s * 0$	Zero	

*Special cases listed in the compatibility table of Appendix A.

FIGURE 1-4. Double-Precision Operand E and F Fields

1.0 Product Introduction (Continued)

1.2.4 Integer Formats

The FPC-FPDP performs conversions between integer and floating point operands. Integers are accepted and generated by the FPC-FPDP as two's complement values of byte (8 bits), word (16 bits) or double-word (32 bits).

n − 1	0
S	I

FIGURE 1-5. Integer Format

TABLE 1-4. Integer Fields

S	Value	Name
0	I	Positive Integer
1	$I - 2^n$	Negative Integer

n represents number of bits in the word, 8 for byte, 16 for word and 32 for double-word.

The FPDP supports only 32-bit integers, therefore, the FPC has to sign extend 8- and 16-bit integers prior to integer to floating-point number conversion.

In floating point to integer conversion, FPC has to check possible integer overflow, in case of 8- and 16-bit integer formats.

1.2.5 Memory Representations

The NS32580 FPC does not directly access memory. However, it is cooperatively involved in the execution of a set of two-address instructions with the NS32532 CPU. The CPU determines the representation of operands in memory.

In the Series 32000 family of CPUs, operands are stored in memory with the least significant byte at the lowest byte address. The only exception to this rule is the Immediate addressing mode, where the operand is held (within the instruction format) with the most significant byte at the lowest address.

2.0 Architectural Description

2.1 PROGRAMMING MODEL

The Series 32000 architecture implements nine floating point registers in the FPC; eight data registers and one floating-point status register.

2.1.1 Floating-Point Data Registers (L0–L7)

There are eight registers (L0–L7) in the FPC for providing high-speed access to floating-point operands. Each is 64 bits long. A floating-point register is referenced whenever a floating-point instruction uses the Register addressing mode (Section 2.2.2) for a floating-point operand. All other Register mode usages (i.e., integer operands) refer to the General Purpose Registers (R0–R7) of the CPU, and the FPC transfers the operand as if it were in memory.

Note: These registers are all upward compatible with the 32-bit NS32081 registers, (F0–F7), such that when the Register addressing mode is specified for a double precision (64-bit) operand, a pair of 32-bit registers holds the operand. The programmer specifies the even register of the pair which contains the least significant half of the operand and the next consecutive register contains the most significant half.

2.1.2 Floating-Point Status Register (FSR)

The Floating-Point Status Register selects operating modes and records any exceptional condition encountered during execution of a floating-point operation. The FPC FSR contains all the NS32081/NS32381 FSR bits and additional fields for better exception handling. The FSR is cleared to all zeros during reset.

DATA

← 32 Bits →	← 32 Bits →
F1 / L0 MSDW	F0 / L0 LSDW
L1 MSDW	L1 LSDW
F3 / L2 MSDW	F2 / L2 LSDW
L3 MSDW	L3 LSDW
F5 / L4 MSDW	F4 / L4 LSDW
L5 MSDW	L5 LSDW
F7 / L6 MSDW	F6 / L6 LSDW
L7 MSDW	L7 LSDW

LSDW → Least Significant Double Word

MSDW → Most Significant Double Word

FIGURE 2-1. Data Registers

2.1.2.1 FSR Mode Control Fields

The FSR mode control fields select FPC operation modes. The meanings of the FSR mode control bits are given below:

Rounding Mode (RM bit 8–7). This field selects the rounding method. Floating-point results are rounded whenever they cannot be represented exactly. The rounding modes are:

00 Round to nearest value. The value which is nearest to the exact result is returned. If the result is exactly halfway between the two nearest values the even value (lsb = 0) is returned.

01 Round toward zero. The nearest value which is closer to zero or equal to the exact result is returned.

10 Round toward positive infinity. The nearest value which is greater than or equal to the result is returned.

11 Round toward negative infinity. The nearest value which is less than or equal to the exact result is returned.

Underflow Trap Enable (UEN bit 3). If this bit is set, the FPC requests a trap whenever a result is too small in absolute value to be presented as a Normalized number. If it is not set, FPC returns a result of zero.

Inexact Result Trap Enable (IEN bit 5). If this bit is set, the FPC requests a trap whenever the result of an operation cannot be represented exactly in the operand format of the destination (and no other exception occurred in the same operation) or if the result of an operation overflows and the overflow trap is disabled. If IEN is not set, the result is rounded according to the selected rounding mode.

2.1.2.2 FSR Status Fields

The FSR Status Fields record exceptional conditions encountered during floating-point data processing. The meaning of the FSR status bits are given below:

Trap Type (TT bits 2–0). This 3-bit field indicates the reason for TRAP (FPU) requested by the FPC. The TT field is loaded with zero whenever any floating-point instruction except LFSR or SFSR completes without exception. It is also set to zero by a reset or by writing zero into it with the LFSR instruction. The TT field is updated regardless of the setting of the exception enable bits.

31 … 17	16	15 … 9	8 7	6	5	4	3	2 0
New Fields	RMB	SWF	RM	IF	IEN	UF	UEN	TT

FIGURE 2-2. FSR (Compatible Fields)

000	No exceptional condition occurred.
001	Underflow. This condition occurs whenever a result is too close to zero to be represented as a Normalized number.
010	Overflow. This condition occurs whenever a result is too large in absolute value to be represented (float or integer).
011	Divide by Zero. This condition occurs whenever an attempt was made to divide a non-zero value by zero.
100	Illegal Instruction. An illegal or undefined Floating-Point instruction was passed to the FPC. If the T bit in the Status Word Register (SWR) is a "0", then it indicates that an illegal instruction was passed to the FPC. If the T bit in the SWR is a "1", then it indicates that an undefined instruction was passed to the FPC.
101	Invalid Operation. This condition occurs if: 1. NaN is used as a floating-point operand by any instruction except MOVf and the Reserved Operand Enable (ROE) bit in the FSR is disabled. 2. DNRM is used as a floating-point operand by any instruction except MOVf. 3. Both operands of the DIVf instruction are zero. 4. Sqrt when the floating-point number is negative. 5. Infinity plus negative infinity, infinity minus infinity.
110	Inexact Result. This condition occurs whenever the result of an operation cannot be exactly represented in the precision of the destination (and no other exception occurred in the same operation) or if the result of an operation overflows (floating-point or integer conversion overflow) and the overflow trap is disabled.
111	Reserved.

Underflow Flag (UF bit 4). This bit is set by the FPC whenever a result is too small in absolute value to be represented as a Normalized number. Its function is not affected by the state of the UEN bit. The UF bit is "sticky" therefore it can be cleared only by writing a zero into it with the Load FSR instruction or by a hardware reset.

Inexact Result Flag (IF bit 6). This bit is set by the FPC whenever the result of an operation must be rounded to fit within the destination format (and no other exception occurred in the same operation) or if the result of an operation overflows and the overflow trap is disabled. This situation applies both to floating-point and integer destinations. The IF bit is "sticky" therefore it is cleared only by writing a zero into it with the Load FSR instruction or by a hardware reset.

Register Modify Bit (RMB BIT 16). This bit is set by the FPC whenever writing to a floating-point data register. The RMB bit is cleared only by writing a zero with the LFSR instruction or by a hardware reset. This bit can be used in context switching to determine whether the FPC registers should be saved.

2.1.2.3 FSR Software Field (SWF)

Bits 15–9 of the FSR hold and display any information written to them using the LFSR and SFSR instructions, but are not otherwise used by FPC hardware. They are reserved for use with NSC floating-point extension software.

2.1.2.4 FSR New Fields

New fields were added to the FSR for better exception handling. In the FPC, the user can enable or disable each exception or combination of exceptions by using new "enable bits" implemented in the FSR. After reset the new fields are loaded to the default values (compatible with NS32081). Illegal Instruction always causes TRAP and can't be disabled. The bits are defined as follows:

CONTROL BITS

Reserved Operands Enable (ROE bit 17). If this bit is cleared, the FPC requests an Invalid Operation trap whenever a NaN has been detected by the FPC. When ROE is disabled, the FPC does not generate reserved operands as results. If the ROE is set then NaN will be returned as the result with no trap and the ROF bit is cleared. If Invalid Operation exception is disabled, the ROE bit is overwritten internally (the FPC does not change the ROE bit in the FSR) and the FPC can generate NaN as a result. ROE bit does not affect MOVf instruction.

Invalid Operation Enable (IVE bit 18). If this bit is cleared, the FPC requests a trap whenever the operation is invalid. If this bit is set to "1", the trap is disabled and if invalid operation occurred, NaN will be delivered as result.

Divide By Zero Enable (DZE bit 19). If this bit is cleared the FPC requests a trap whenever an attempt is made to divide by zero. If this bit is set the trap is disabled and if divide by zero occurred, infinity will be delivered as result.

Overflow Enable (OVE bit 20). If this bit is cleared, the FPC requests a trap whenever a floating-point result is too big in absolute value to be represented. If this bit is set, the overflow trap is disabled and if overflow occurred, Infinity or Maximum Number will be delivered as result.

Integer Conversion Overflow Enable (IOE bit 21). If this bit is cleared, the FPC requests a trap whenever an Integer result is too big to be represented. If this bit is set, the integer conversion overflow is disabled and if integer conversion overflow occurred, Max/Min integer will be delivered as result.

STATUS BITS

Reserved Operand Flag (ROF bit 22). This bit is set by the FPC whenever reserved operand DNRM or NaN (when ROE is cleared) is selected by the FPC. The ROF bit is "sticky" and can be cleared only by writing a zero with the Load FSR instruction or by a hardware reset.

Invalid Flag (IVF bit 23). This bit is set by the FPC whenever the operation is invalid. The IVF bit is "sticky" and can be cleared only by writing a zero with the Load FSR instruction or by a hardware reset.

Divide By Zero Flag (DZF bit 24). This bit is set by the FPC whenever an attempt is made to divide a non-zero value by zero. The DZF bit is "sticky" and can be cleared only by writing a zero with the Load FSR instruction or by a hardware reset.

2.0 Architectural Description (Continued)

31 27	26	25	24	23	22	21	20	19	18	17	16
Reserved	IOF	OVF	DZF	IVF	ROF	IOE	OVE	DZE	IVE	ROE	RMB

FIGURE 2-3. New FSR Mode Control Fields

Overflow Flag (OVF bit 25). This bit is set by the FPC whenever a floating-point result is too large in absolute value to be represented. The OVF bit is "sticky" and can be cleared only by writing a zero with the Load FSR instruction or by a hardware reset.

Integer Conversion Overflow Flag (IOF bit 26). This bit is set by the FPC whenever an integer result is too large in absolute value to be represented. The IOF bit is "sticky" and can be cleared only by writing a zero with the Load FSR instruction or by a hardware reset.

Reserved Field

Bits 31–27 in the FSR are reserved by NSC for future use. User should not use this field.

2.1.2.5 FSR Default Values

During Reset the FSR is loaded to a default value (see Table 2-1). The default values for the FSR represent upward compatibility of the FPC-FPDP with the NS32081. The user can change the default values by loading the FSR register with new values.

TABLE 2-1. FSR Default State Summary

Bit Name	Default Value	Default State
TT (bits 2–0)	0	No exceptional condition occurred.
UEN (bit 3)	0	Underflow trap disabled.
UF (bit 4)	0	Underflow flag is cleared.
IEN (bit 5)	0	Inexact result trap disabled.
IF (bit 6)	0	Inexact flag is cleared.
RM (bits 8–7)	0	Round to nearest.
SWF (bits 15–9)	0	Undefined
RMB (bit 16)	0	RMB flag is cleared.
ROE(bit 17)	0	FPC requests a trap whenever an attempt is made to use reserved operand except for MOVf instruction.
IVE (bit 18)	0	FPC requests a trap whenever the operation is invalid.
DZE (bit 19)	0	FPC requests a trap whenever an attempt is made to divide by zero.
OVE (bit 20)	0	FPC requests a trap whenever a floating-point result is too big to be represented.
IOE (bit 21)	0	FPC requests a trap whenever an integer conversion result is too big to be represented.
ROF (bit 22)	0	ROF flag is cleared.
IVF (bit 23)	0	IVF flag is cleared.
DZF (bit 24)	0	DZF flag is cleared.
OVF (bit 25)	0	OVF flag is cleared.
IOF (bit 26)	0	IOF flag is cleared.
RESERVED (bits 31–27)	0	Reserved field is cleared.

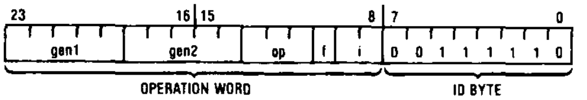
2.2 INSTRUCTION SET

2.2.1 Floating-Point Instruction Set

This section provides a description of the floating-point instructions executed by the FPC in conjunction with the CPU and the FPDP. These instructions form a small subset of the Series 32000 instruction set and their encodings use instruction formats 9, 11, and 12. A list of all the Series 32000 instructions as well as details on their formats and addressing modes can be found in the appropriate CPU data sheets.

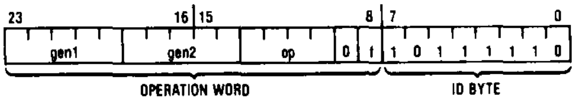
Certain notations in the following instruction description tables serve to relate the assembly language form of each instruction to its binary format in *Figure 2-4*.

Format 9



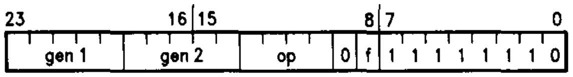
TL/EE/9421-5

Format 11



TL/EE/9421-6

Format 12



TL/EE/9421-7

FIGURE 2-4. Floating-Point Instruction Formats

The Format column indicates which of the three formats in *Figure 2-4* represents each instruction.

The Op column indicates the binary pattern for the field called "op" in the applicable format.

The Instruction column gives the form of each instruction as it appears in assembly language. The form consists of an instruction mnemonic in upper case, with one or more suffixes (i or f) indicating data types, followed by a list of operands (gen1, gen2).

An i suffix on an instruction mnemonic indicates a choice of integer data types. This choice affects the binary pattern in the i field of the corresponding instruction format as follows:

Suffix i	Data Type	i Field
B	Byte	00
W	Word	01
D	Double Word	11

An f suffix on an instruction mnemonic indicates a choice of floating-point data types. This choice affects the setting of the f bit of the corresponding instruction format as follows:

Suffix f	Data Type	f Bit
F	Single Precision	1
L	Double Precision (Long)	0

An operand designation (gen1, gen2) indicates a choice of addressing mode expressions. This choice affects the binary pattern in the corresponding gen1 or gen2 field of the instruction format.

Further details of the exact operations performed by each instruction are found in the Series 32000 Instruction Set Reference Manual.

Movement and Conversion

The following instructions move the gen1 operand to the gen2 operand, leaving the gen1 operand intact.

Format	Op	Instruction		Description
11	0001	MOVf	gen1, gen2	Move without conversion.
9	010	MOVLF	gen1, gen2	Move, converting from double precision to single precision.
9	011	MOVFL	gen1, gen2	Move, converting from single precision to double precision.
9	000	MOVif	gen1, gen2	Move, converting from any integer type to any floating-point type.
9	100	ROUNDfi	gen1, gen2	Move, converting from floating-point to the nearest integer.
9	101	TRUNCfi	gen1, gen2	Move, converting from floating-point to the nearest integer closer to zero.
9	111	FLOORfi	gen1, gen2	Move, converting from floating-point to the largest integer less than or equal to its value.

Note: The MOVLF instruction f bit must be 1 and the i field must be 10.
The MOVFL instruction f bit must be 0 and the i field must be 11.

Arithmetic Operations

The following instructions perform floating-point arithmetic operations on the gen1 and gen2 operands, leaving the result in the gen2 operand.

	Format	Op	Instruction		Description
	11	0000	ADDf	gen1, gen2	Add gen1 to gen2.
	11	0100	SUBf	gen1, gen2	Subtract gen1 from gen2.
	11	1100	MULf	gen1, gen2	Multiply gen2 by gen1.
	11	1000	DIVf	gen1, gen2	Divide gen2 by gen1.
	11	0101	NEGf	gen1, gen2	Move negative of gen1 to gen2.
	11	1101	ABSf	gen1, gen2	Move absolute value of gen1 to gen2.
(N)	12	1010	MACf	gen1, gen2	Move (gen1*gen2) + L1 or F1 to L1 or F1 with two rounding errors.
(N)	12	0001	SQRTf	gen1, gen2	Move the square root of gen1 to gen2.

(N): Indicates NEW instruction.

Comparison

The compare instruction compares two floating-point operands, sending the result to the CPU PSR Z, N and L bits for use as condition codes.

Format	Opcode	Instruction		Description
11	0010	CMPf	gen1, gen2	Compare gen1 to gen2.

2.0 Architectural Description (Continued)

There are four possible results to the CMPf instruction (with normal operands):

Operands are equal	Z bit is set	N, L bits are cleared
Operand1 is less than Operand2		N, L, Z bits are cleared
Operand2 is less than Operand1	N bit is set	L, Z bits are cleared
Unordered (when at least one operand is NaN and ROE is set)	L bit is set	N, Z bits are cleared

Floating-Point Status Register Access

The following instructions load and store the FSR as a 32-bit integer. If the user specifies a register (gen1 in LFSR or gen2 in SFSR) it will be a general purpose register in the CPU.

Format	Opcode	Instruction		Description
9	001	LFSR	gen1	Load FSR with the content of gen1. (gen2 field = 0)
9	110	SFSR	gen2	Store FSR in gen2. (gen1 field = 0)

Note: All instructions support all of the NS32000 family data formats (for external operands) and all addressing modes are supported.

2.3 EXCEPTIONS

An exception for the FPC is a special floating-point condition with a default handling scheme. Seven types of exceptions are supported:

1) Underflows
2) Overflows
3) Divisions by zero
4) Illegal Instructions
5) Invalid Operations
6) Inexact results
7) Undefined Instructions

The FPC has improved exception handling. Except for Illegal and Undefined Instructions, the user can control all of the exception types. In addition, there are some specific exceptions that the user can control:

Overflows — Floating-Point overflow
— Integer conversion overflow
Invalid Operations — Reserved Operands

Most exceptions can be enabled to cause a CPU TRAP or to return a result without a TRAP on their occurrence. The TRAP is signaled by the FPC pulsing the $\overline{\text{FSSR}}$ line for one clock cycle. Illegal and Undefined instructions will always cause a TRAP if they are passed to the FPC.

When a TRAP occurs, the FPC sets the Q bit in the status word register. The CPU responds by reading the status word register while applying status (11110) on the status lines. If the TRAP is caused by an undefined opcode, the TS bit in the status word register will also be set by the FPC indicating a TRAP (UND). The TS bit is clear in all other cases.

When an exception occurs, the type field in the FSR register is also updated. A trapped instruction returns no result (even if the destination is an FPC register) and does not affect the CPU PSR. Instructions that end with a disabled exception will always return a result.

For each exception whose TRAP can be disabled, there is a flag bit to signal the occurrence of the exceptional condition whether or not the TRAP is enabled or disabled. These bits in the FSR can be used for polling the exception status while TRAPs are disabled.

2.0 Architectural Description (Continued)

TABLE 2-2. Exception Handling Summary

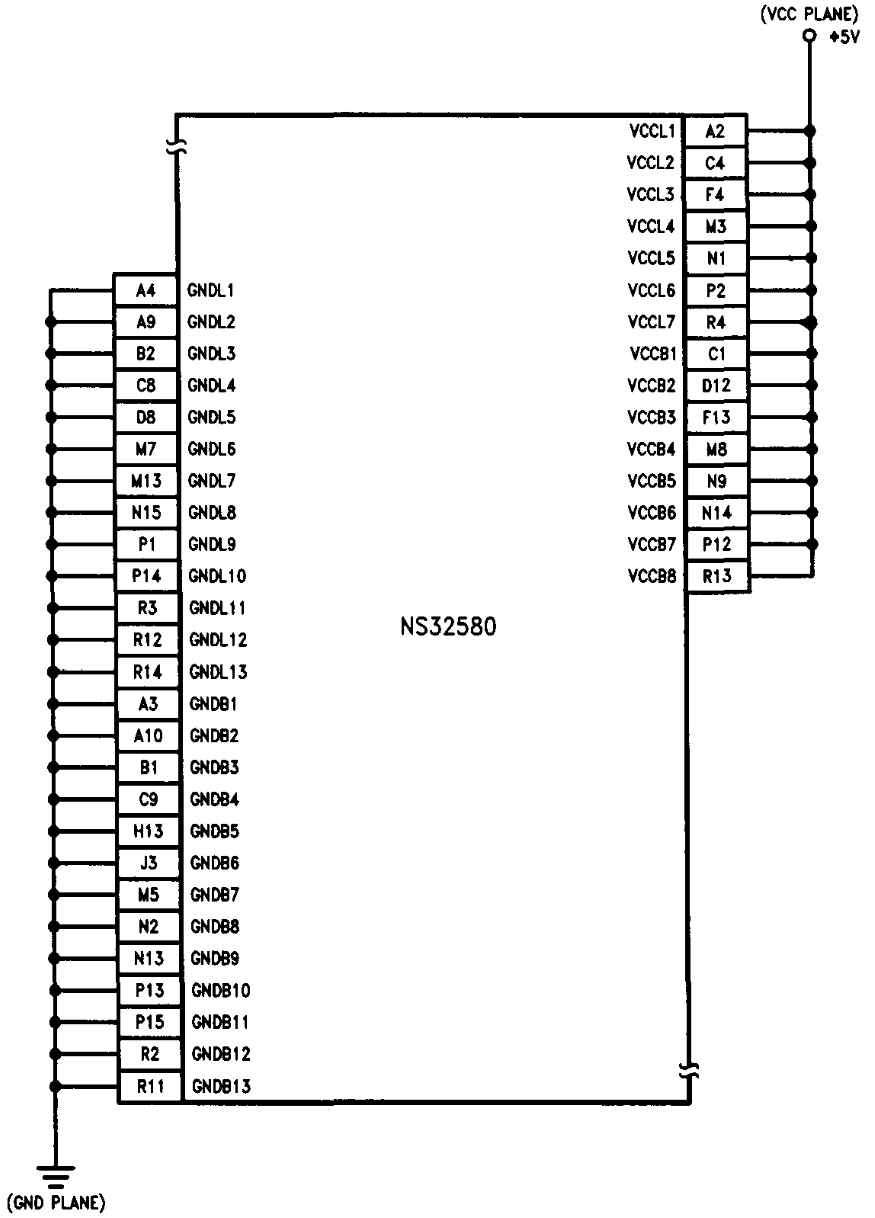
Exception Occurred	Enabled By	Q = 1; Trap Type	Disabled By	Q = 0; Default Result Returned	Flag Bits
Underflow	UEN = 1	001	UEN = 0	Zero	UF = 1
Floating-Point Overflow	OVE = 0	010	OVE = 1 IEN = 0	Infinity or Max NRM Number	OVF = 1
	OVE = 1 IEN = 1	110			OVF = 1 IF = 1
Integer Conversion Ov.	IOE = 0	010	IOE = 1 IEN = 0	Max or Min Integer	IOF = 1
	IOE = 1 IEN = 1	110			IOF = 1 IF = 1
Divide by Zero	DZE = 0	011	DZE = 1	Infinity	DZF = 1
Illegal Instruction	Always Enabled	T bit = 0 and 100	Cannot be Disabled	No Result	No Flags Affected
Invalid Operation	IVE = 0	101	IVE = 1	NaN	IVF = 1
Reserved Op. (NaN)	ROE = 0 IVE = 0	101	ROE = 0 IVE = 1	NaN	ROF = 1 IVF = 1
			ROE = 1 IVE = X	NaN	No Flags Affected
Reserved Op. (DNRM) (Note 1)	ROE = X IVE = 0	101	ROE = X IVE = 1	Undefined	ROF = 1 IVF = 1
Inexact Result	IEN = 1	110	IEN = 0	Correctly Rounded Result	IF = 1
Undefined Instruction	Always Enabled	T bit = 1 and 100	Cannot be Disabled	No Result	No Flags Affected

Exception Occurred	Enabled By	Q = 1; Trap Type	Disabled By	Status Word Register	Flag Bits
CMPf (NaN)	ROE = 0 IVE = 0	101	ROE = 0 IVE = 1	L = 1, N = Z = 0	ROF = 1 IVF = 1
			ROE = 1 IVE = X	L = 1, N = Z = 0	No Flags Affected
CMPf (DNRM)	ROE = X IVE = 0	101	ROE = X IVE = 1	N, L, Z Undefined	ROF = 1 IVF = 1

X = Don't Care

Note 1: For MULf 0 * DNRM
DIVf 0/DNRM
DIVf DNRM/Infinity
NS32580 returns a zero.
For DIVf Infinity/DNRM and MULf Infinity * DNRM, NS32580 returns an infinity.
For DIVf DNRM/0, TRAP (DVZ) will take place.

3.0 Functional Description


(VCC PLANE)
+5V
NS32580
VCCL1 A2
VCCL2 C4
VCCL3 F4
VCCL4 M3
VCCL5 N1
VCCL6 P2
VCCL7 R4
VCCB1 C1
VCCB2 D12
VCCB3 F13
VCCB4 M8
VCCB5 N9
VCCB6 N14
VCCB7 P12
VCCB8 R13
A4 GNDL1
A9 GNDL2
B2 GNDL3
C8 GNDL4
D8 GNDL5
M7 GNDL6
M13 GNDL7
N15 GNDL8
P1 GNDL9
P14 GNDL10
R3 GNDL11
R12 GNDL12
R14 GNDL13
A3 GNDB1
A10 GNDB2
B1 GNDB3
C9 GNDB4
H13 GNDB5
J3 GNDB6
M5 GNDB7
N2 GNDB8
N13 GNDB9
P13 GNDB10
P15 GNDB11
R2 GNDB12
R11 GNDB13
(GND PLANE)

TL/EE/9421-8

FIGURE 3-1. Recommended Supply Connections

3.1 POWER AND GROUNDING

The NS32580 requires a single 5V power supply, applied on 15 pins. The logic voltage pins (VCCL1 to VCCL7) supply the power to the on-chip logic. The buffer voltage pins (VCCB1 to VCCB8) supply the power to the output drivers of the chip. All the voltage pins should be connected together by a power (V_{CC}) plane on the printed circuit board.

The NS32580 grounding connections are made on 26 pins. The logic ground pins (GNDL1 to GNDL13) are the ground pins for the on-chip logic. The buffer ground pins (GNDB1–GNDB13) are the ground pins for the output drivers of the chip. All the ground pins should be connected together by a ground plane on the printed circuit board.

Both power and ground connections are shown in *Figure 3-1.*

3.2 CLOCKING

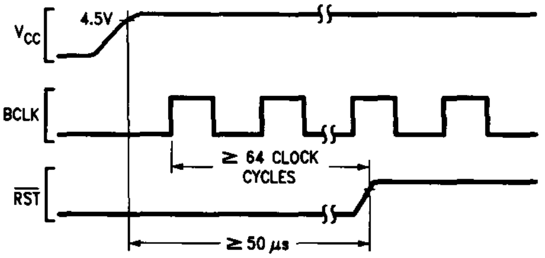
The NS32580 FPC requires a single-phase TTL clock input on its BCLK pin (pin C10) and an inverted TTL clock input on its $\overline{\text{BCLK}}$ pin (pin B8). When the FPC is connected to a NS32532 CPU these signals are provided directly from the CPU's BCLK and $\overline{\text{BCLK}}$ output signals.

3.3 RESETTING

The $\overline{\text{RST}}$ pin serves as a reset for on-chip logic. The FPC may be reset at any time by pulling the $\overline{\text{RST}}$ pin low for at least 64 clock cycles. Upon detecting a reset, the FPC terminates instruction processing, resets its internal logic, clears the FSR to all zeroes, and clears the FIFOs.

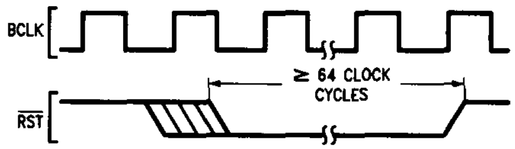
On application of power, $\overline{\text{RST}}$ must be held low for at least 50 μs after V_{CC} is stable. This ensures that all on-chip voltages are completely stable before operation. See *Figures 3-2* and *3-3.*

3.0 Functional Description (Continued)



TL/EE/9421-9

FIGURE 3-2. Power-On Reset Requirements



TL/EE/9421-10

FIGURE 3-3. General Reset Timing

3.4 BUS OPERATION

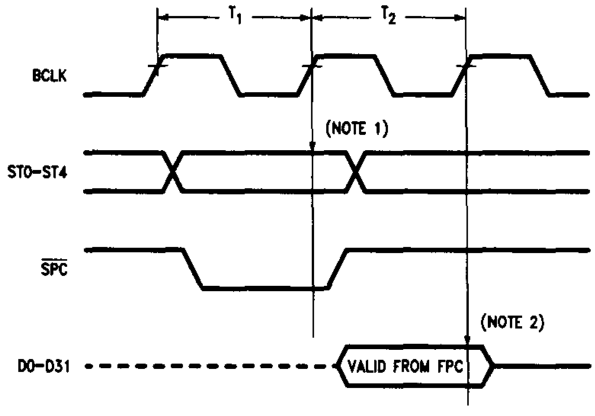
Instructions and operands are passed to the NS32580 FPC with slave processor bus cycles. Each bus cycle transfers one double-word (32 bits) to or from the FPC. During all bus cycles, the $\overline{\text{SPC}}$ line is driven by the CPU as an active low data strobe, and the FPC monitors pins ST0–ST4 to keep track of the sequence (protocol) established for the instruction being executed. This is necessary in a virtual memory environment, allowing the FPC to retry an aborted instruction.

A bus cycle is initiated by the CPU, which asserts the proper status on ST0–ST4 and pulses $\overline{\text{SPC}}$ low. The status lines are sampled by the FPC on the rising edge of BCLK in the T2 state. *Figures 3-4* and *3-5* illustrate these sequences.

3.4.1 Operand Transfers

The CPU fetches operands from memory, aligns them (if needed) and sends them to the slave (with status h'1D) as a 32-bit transfer. If the operand is double-precision the least significant half is transferred first (in 32000 mode). The FPC can not access the memory directly.

From the slave processor point of view there are four possible combinations of locations for operands: (For special cases see next paragraph.)

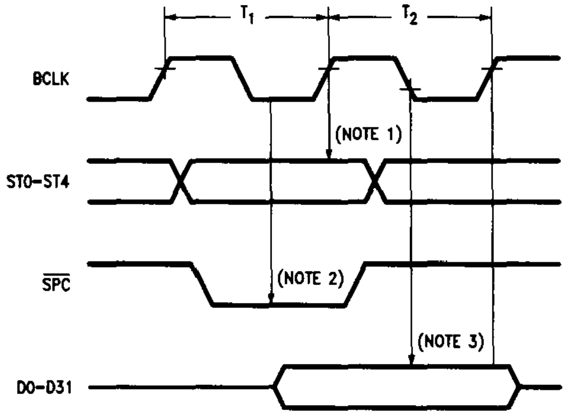


TL/EE/9421-11

Note 1: FPC samples CPU status here.

Note 2: CPU samples FPC data here.

FIGURE 3-4. Slave Processor Read Cycle from FPC



TL/EE/9421-12

Note 1: FPC samples CPU status here.

Note 2: FPC samples $\overline{\text{SPC}}$ here.

Note 3: FPC samples data here.

FIGURE 3-5. Slave Processor Write Cycle to FPC

Register to Register Instructions—Both operands reside in the register file inside the FPDP. No operand fetch or transfer from memory is needed.

Memory to Register—The source operand is in memory, therefore the CPU will transfer the operand (one 32-bit transfer for single-precision and two 32-bit transfers for double-precision). The result is going to the floating-point register in the register file located inside the FPDP.

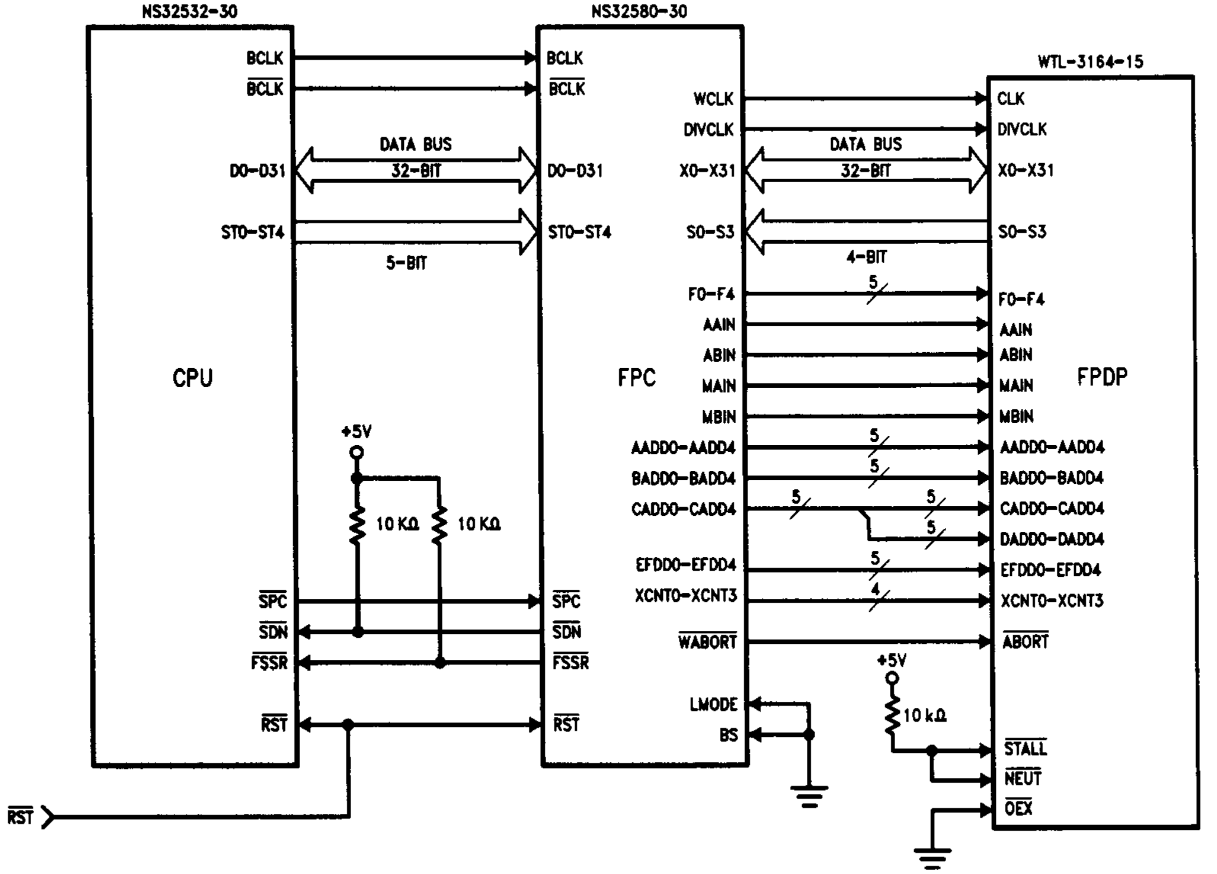
Register to Memory—The source operand resides inside the FPDP. If the instruction is monadic (one operand) the CPU will not transfer the operand to the FPC before the beginning of the instruction (all the information needed to start the operation resides inside the FPDP). For dyadic instructions, the CPU will fetch and transfer one operand from memory.

Memory to Memory—In monadic instructions the source operand is in memory and the CPU will transfer it to the FPC-FPDP. If the instruction is dyadic, two operands will be transferred from memory to the FPC-FPDP by the CPU (gen1 before gen2). The result in both cases is sent back to memory.

When the CPU transfers an operand from memory to the FPC-FPDP it is loaded into one of the registers that create the operand FIFO inside the FPDP. The FPC translates the incoming instruction (mem, reg or mem, mem) to a register-to-register instruction with the same register number. From the incoming instruction addressing mode it should know if the operands are coming from memory or already located in the register file.

The Data FIFO inside the FPC is 10 entries deep, single- or double-precision. If the destination of instruction is memory, the FPC will wait for completion of the instruction. Then, the result will be transferred to the FPC and $\overline{\text{SDN}}$ will be asserted. If the FPC receives a new ID and Opcode before the FPC receives all the operands for the last instruction or before the CPU reads the conplete result for the last instruction (can happen if page fault has been detected on a write and with only one instruction in the FPC's pipe), the FPC will abort the last instruction and will start the execution of the new instruction. The NS32532 CPU can "reset" the FPC at any time by asserting $\overline{\text{SPC}}$ with status 11110. In this case the FPC flushes the instructions currently being executed and the contents of the floating-point registers are undefined.

3.0 Functional Description (Continued)


NS32532-30
NS32580-30
WTL-3164-15
CPU
FPC
FPDP
BCLK
BCLK
D0-D31
DATA BUS
32-BIT
ST0-ST4
5-BIT
+5V
10 KΩ
10 KΩ
SPC
SDN
FSSR
RST
RST
WCLK
DIVCLK
X0-X31
S0-S3
4-BIT
F0-F4
AAIN
ABIN
MAIN
MBIN
AADD0-AADD4
BADD0-BADD4
CADD0-CADD4
EFDD0-EFDD4
XCNT0-XCNT3
WABORT
LMODE
BS
CLK
DIVCLK
X0-X31
S0-S3
F0-F4
AAIN
ABIN
MAIN
MBIN
AADD0-AADD4
BADD0-BADD4
CADD0-CADD4
DADD0-DADD4
EFDD0-EFDD4
XCNT0-XCNT3
ABORT
STALL
NEUT
OEX
10 kΩ
5
4

TL/EE/9421-13

FIGURE 3-6. System Connection Diagram
***(For Two Cycle Latency in Little Endian Mode)**

*See Pin Description for other configurations.

3.5 INSTRUCTION PROTOCOLS

3.5.1 General Protocol Sequence

The FPC supports both the regular and the pipelined slave protocols provided by the NS32532. Detailed information on these protocols is provided in the NS32532 data sheet.

The basic operations performed by the CPU and the FPC are described below.

Floating-point instructions have a three-byte Basic Instruction field, consisting of an ID Byte followed by an Operation Word.

Upon receiving a floating-point instruction, the CPU initiates the sequence outlined in Table 3-1. While applying Status code 11111, the CPU transfers the ID Byte on bits D24–D31, the operation word on bits D8–D23 in a swapped order of bytes and a non-used byte XXXXXXXX (X = don't care) on bits D0–D7 *(Figure 3-7)*.

After transferring the instruction, the CPU sends to the FPC any source operands that are located in memory or the CPU General-Purpose registers.

The CPU action, at this point, depends on whether the regular or the pipelined slave protocol is selected. If the regular protocol is selected, the CPU waits for the FPC to complete the instruction. While the CPU is waiting, it can perform bus cycles to fetch instructions and read source operands for instructions that follow the floating-point instruction being executed. If there are no bus cycles to perform, the CPU is idle with a special Status indicating that it is waiting for a slave .

If the pipelined protocol is selected, the CPU may send a new floating-point instruction to the FPC before the previous instruction has been completed.

Although the CPU can advance as many as four floating-point instructions before receiving a completion pulse on SDN for the first instruction, full exception recovery is assured. This is accomplished through a FIFO mechanism which maintains the addresses of all the floating-point instructions sent to the FPC for execution.

Pipelined execution can occur only for instructions which do not require a result to be read from the FPC.

3.0 Functional Description (Continued)

In cases where a result is to be read back, the CPU will wait for instruction completion before issuing the next instruction. After the FPC asserts $\overline{SDN}$ or $\overline{FSSR}$, the CPU follows one of the two sequences described below.

If the FPC asserts $\overline{SDN}$, then the CPU checks whether the instruction stores any results to memory or the General-Purpose registers. The CPU reads any such results from the FPC by means of 1 or 2 bus cycles and updates the destination. If the instruction had been pipelined, the CPU simply updates the FIFO pointer to point to the next instruction in the FIFO.

If the FPC asserts $\overline{FSSR}$, then the NS32532 reads a 32-bit status word from the FPC. The CPU checks bit 0 in the FPC's status word to determine whether to update the PSR flags or to process an exception. *Figure 3-8* shows the format of the FPC's status word.

If the Q bit in the status word is 0, the CPU updates the N, Z and L flags in the PSR.

If the Q bit in the status word is set to 1, the CPU processes either a Trap (UND) if TS is 1 or a Trap (SLAVE) if TS is 0.

31			0
ID BYTE	OPCODE (LOW)	OPCODE (HIGH)	XXXXXXXX

FIGURE 3-7. ID and Operation Word

31	15		7							0
ZERO	TS	ZERO	N	Z	0	0	0	L	0	Q

FIGURE 3-8. FPC Status Word

TABLE 3-1. Floating-Point Instruction Sequence

Step	Status	Action
1	ID (11111)	CPU sends ID and Operation Word
2	OP (11101)	CPU sends required operands (if any)
3	—	Slaves starts execution (CPU prefetches)
4	—	Slave signals completion by pulsing $\overline{SDN}$ or $\overline{FSSR}$.
5	ST (11110)	CPU Reads Status Word (If an exception occurred or if a CMPf instruction was executed)
6	OP (11101)	CPU Reads Result (if any)

3.0 Functional Description (Continued)

TABLE 3-2. Floating-Point Instruction Protocols

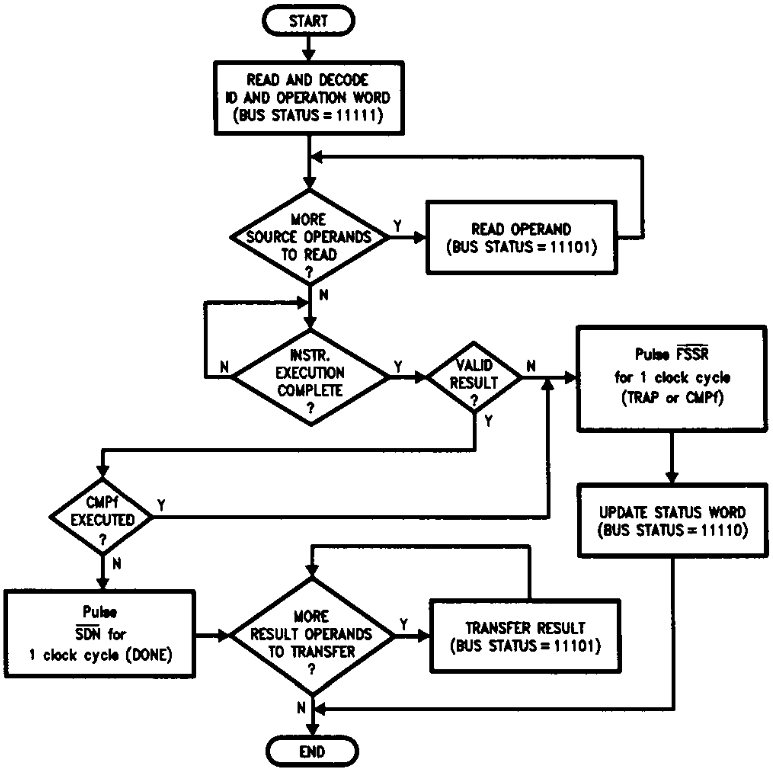
Mnemonic	Operand 1 Class	Operand 2 Class	Operand 1 Issued	Operand 2 Issued	Returned Value	PSR Bits Affected
ADDf	read.f	rmw.f	f	f	f to Op. 2	none
SUBf	read.f	rmw.f	f	f	f to Op. 2	none
MULf	read.f	rmw.f	f	f	f to Op. 2	none
DIVf	read.f	rmw.f	f	f	f to Op. 2	none
MOVf	read.f	write.f	f	N/A	f to Op. 2	none
ABSf	read.f	write.f	f	N/A	f to Op. 2	none
NEGf	read.f	write.f	f	N/A	f to Op. 2	none
CMPf	read.f	read.f	f	f	N/A	N,Z,L
FLOORfi	read.f	write.i	f	N/A	i to Op. 2	none
TRUNCfi	read.f	write.i	f	N/A	i to Op. 2	none
ROUNDfi	read.f	write.i	f	N/A	i to Op. 2	none
MOVFL	read.F	write.L	F	N/A	L to Op. 2	none
MOVLF	read.L	write.F	L	N/A	F to Op. 2	none
MOVif	read.i	write.f	i	N/A	f to Op. 2	none
LFSR	read.D	N/A	D	N/A	N/A	none
SFSR	N/A	write.D	N/A	N/A	D to Op. 2	none
SQRTf	read.f	write.f	f	N/A	f to Op.2	none
MACf	read.f	read.f	f	f	f to L1/F1	none

D = Double Word

i = Integer size (B, W, D) specified in mnemonic.

f = Floating-Point type (F, L) specified in mnemonic.

N/A = Not Applicable to this instruction.


START
READ AND DECODE ID AND OPERATION WORD (BUS STATUS = 11111)
MORE SOURCE OPERANDS TO READ ?
Y
READ OPERAND (BUS STATUS = 11101)
N
INSTR. EXECUTION COMPLETE ?
N
Y
VALID RESULT ?
N
Y
Pulse FSSR for 1 clock cycle (TRAP or CMPf)
UPDATE STATUS WORD (BUS STATUS = 11110)
CMPf EXECUTED ?
Y
N
Pulse SDN for 1 clock cycle (DONE)
MORE RESULT OPERANDS TO TRANSFER ?
Y
TRANSFER RESULT (BUS STATUS = 11101)
N
END

TL/EE/9421-14

FIGURE 3-9. Floating-Point Instruction Processing Flowchart

3.0 Functional Description (Continued)

With the pipelined protocol selected, the FPC can start execution of a new floating-point instruction every two clock cycles.

In the following example three floating-point instructions are executed in pipelined mode:

```
DIVF    O(RO), F1
ADDF    F2, F3
MULF    F4, F5
```

Step	Status	Action
1	ID (h'1F)	CPU sends ID and Opcode of DIVF instruction.
2	OP (h'1D)	CPU sends operand (R0).
3	—	Slave starts execution of DIVF instruction.
4	ID (h'1F)	CPU sends ID and Opcode of ADDF instruction.
5	—	Slave starts execution of ADDF instruction.
6	ID (h'1F)	CPU sends ID and Opcode of MULF instruction.
7	—	Slave starts execution of MULF instruction.
8	—	Slave pulses $\overline{SDN}$ or $\overline{FSSR}$ for the DIVF instruction. If an exception occurred, the rest of the instructions will be aborted.
9	ST (h'1E)	CPU Reads Status Word (if an exception occurred).
10	—	Slave pulses $\overline{SDN}$ or $\overline{FSSR}$ for the ADDF instruction. If an exception occurred, the rest of the instructions will be aborted.
11	ST (h'1E)	CPU Reads Status Word (if an exception occurred).
12	—	Slave pulses $\overline{SDN}$ or $\overline{FSSR}$ for the MULF instruction.
13	ST (h'1E)	CPU Reads Status Word (if an exception occurred).

Note: Instructions that can be pipelined include all instructions except CMPf in Format 11, as well as SQRTf and MAGf in Format 12. All other floating-point instructions will cause the pipe to break, that is, the instruction will be sent to the FPC but the pipe will stop until done or Trap.

3.5.2 Byte Sex

The FPC supports both little Endian and big Endian byte ordering, depending on the state of the BS pin. In little Endian mode (BS = "0"), the FPC receives the least significant half of a double-precision operand first and the most significant half afterward. In Big Endian mode (BS = "1"), the FPC receives the most significant half of a double-precision operand first and the least significant half afterward. The FPC will send the received operands to the correct destination registers inside the FPDP. In Big Endian mode, the user must swap the data bus between the CPU and FPC. See *Figure 3-10* for details. The BS pin is sampled by the FPC during Reset only.

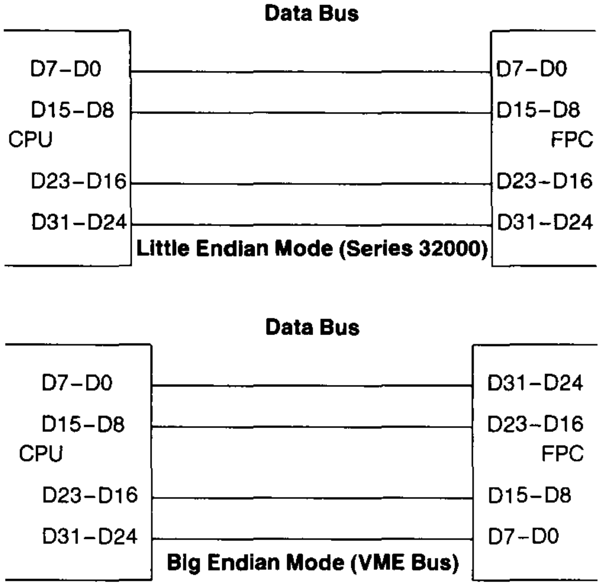


FIGURE 3-10. Byte Sex Connection Diagrams

3.5.3 Floating-Point Instruction Protocols

Table 3-2 gives the protocols followed for each floating-point instruction. The instructions are referenced by their mnemonics. For the bit encodings of each instruction, see Section 2.2.3.

The Operand Class columns give the Access Classes for each general operand, defining how the addressing modes are interpreted by the CPU (see Series 32000 Instruction Set Reference Manual).

The Operand Issued columns show the sizes of the operands transferred to the Floating-Point Controller by the CPU. "D" indicates a 32-bit Double Word. "i" indicates that the instruction specifies an integer size for the operand (B = Byte, W = Word, D = Double Word). "f" indicates that the instruction specifies a floating-point size for the operand (F = 32-bit Standard Floating, L = 64-bit Long Floating).

The Returned Value Type and Destination column gives the size of any returned value and where the CPU places it. The PSR Bits Affected column indicates which PSR bits, if any, are updated from the FPC Status Word *(Figure 3-9)*.

Any operand indicated as being of type "f" will not cause a transfer between CPU and FPC, if the Register addressing mode is specified, since the Floating-Point Registers are physically located in the FPC and are therefore available without CPU assistance.

3.6 FPDP INTERFACE

The FPC uses the Weitek WTL 3164 Floating-Point Data Path (FPDP) as the computational unit.

The FPDP is capable of supporting 32-bit and 64-bit IEEE floating-point operations. The FPDP consists of a Multiplier, ALU, Divide/Sqrt unit, 32 x 64-bit, Six-Port Register file, I/O port and control unit. There are six major internal 64-bit wide data buses used for data transfers between the different blocks inside the FPDP.

3.0 Functional Description (Continued)

Using six data buses allows the input of two double-precision operands to a selected unit and to output one double-precision result in one WCLK cycle, supporting pipelining of a new double-precision instruction every WCLK cycle. For a detailed description of the FPDP, refer to the appropriate data sheet from Weitek.

3.6.1 Controlling the FPDP

The FPC controls the FPDP on an instruction by instruction basis. The instruction control signals are delayed in the FPDP to match the FPDP pipeline stages.

This allows to specify all the controls for a Reg to Reg instruction in a single control word. There are two types of operations that can be executed concurrently on the FPDP. The first operation is a floating-point arithmetic operation done on operands from the register file. The second operation is a Load/Store operation using the X port of the FPDP.

3.6.2 Instruction Control

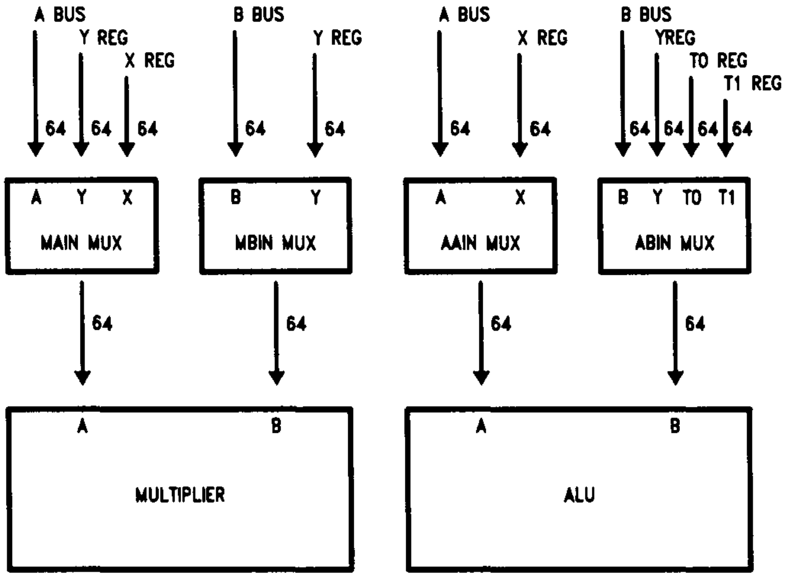
The FPC controls the FPDP using a 33-bit control word. The control word contains all the information needed for the execution of an instruction including the function to be executed, source operands and destination of the result. The controls are pipelined along with the instruction and affect the operation at the appropriate times. The control word is sampled with the rising edge of WCLK.

There are three functional fields in the control word:

1. The FUNC field defines the arithmetic operation to be executed.
2. The AAIN, ABIN, MAIN, MBIN, A ADD, B ADD, C ADD, D ADD bits specify the source and destination for arithmetic operations. Both C ADD and D ADD fields of the FPDP are connected to the D ADD field in the FPC control word.
3. The E/F ADD and XCNT fields control the Load and Store operations. E/F ADD selects the register to be loaded while XCNT selects the operation. XCNT encodings are provided in the following table.

5	1	1	1	1	5	5	5	5	5	4
FUNC	AAIN	ABIN	MAIN	MBIN	A ADD	B ADD	C ADD	D ADD	E/F ADD	XCNT
C41										C4

FIGURE 3-11. FPDP Control Word



TL/EE/9421-15

FIGURE 3-12. FPDP Multiplier and ALU Bus Control

3.0 Functional Description (Continued)

XCNT Field Encodings

The XCNT field specifies the I/O operation to be executed.

Code	Operation	Description
H'0	NOP	No Operation
H'1	EREG LS → XPAD	Transfer the Least Significant half of the register specified by EREG to the X-port (Store LS).
H'2	EREG MS → XPAD	Transfer the Most Significant half of the register specified by EREG to the X-port (Store MS).
H'3	EREG INT → XPAD	Transfer Integer operand in the register specified by EREG to the X-port (Store Int).
H'5	XPAD → XREG/FREG LS	Load the Least Significant half of the data in the X-port into the XREG LS and into the register specified by FREG.
H'6	XPAD → XREG/FREG MS	Load the Most Significant half of the data in the X-port into the XREG MS and into the register specified by FREG.
H'7	XPAD → XREG/FREG INT	Load the Integer operand in X-port into the XREG and into the register specified by FREG.

Data from the FPC is transferred to the FPDP through the XPAD Port. The data is loaded into the XREG and into a register in the register file specified by the E/F ADD.

Loading the data to both locations allows the immediate use of the data by the ALU and MULT, bypassing the register file. Loading the data to a register in the register file prevents data from being lost if the data from memory is needed a few cycles later.

The FPDP I/O Mode is determined by the control bits in the control register SR1 bits 4–0. The FPDP is used in Undelayed Single-Pump mode (code 00000).

3.6.3 "2 Cycle Mode" and "3 Cycle Mode"

The FPDP has two timing modes, "Two cycle latency" and "Three cycle latency". In "Two cycle latency" single- and double-precision operations have latency of two cycles. In "Three cycle latency", double-precision multiply has a three cycle latency, single-precision multiplies and single- or double-precision ALU operations have two cycle latency.

When using the "Three cycle latency" the Divide/Sqrt block uses the same clock as the other functional units in the FPDP. Although the "Three cycle latency" is not optimized for double-precision multiply it may be very useful if the WCLK frequency is higher than the FPDP speed rating.

The FPC has a pin to specify the desired mode. In "Three cycle latency" the LMODE pin should be connected to V_{CC} and in "Two cycle latency" it should be connected to GND. The LMODE line is sampled during reset. After reset, as part of the initialization cycle, the FPC updates the Multiply Latency bit in the FPDP control register SR0 bit-7 (0 = "Two cycle latency", 1 = "Three cycle latency").

In "Three cycle latency" the Divide/Sqrt block uses DCLK3 (same as WCLK), in "Two cycle latency" it uses DCLK2 (2 × WCLK). The FPC uses the latency pin to determine the length of some instructions (number of cycles before FPC can signal DONE or TRAP).

This feature allows the CPU to run at more than twice the maximum FPDP frequency.

The following table shows the system speed versus the FPDP speed and latency selection.

FPDP Speed Grade	WCLK "Two Cycle Latency"	WCLK "Three Cycle Latency"	Max System Speed
120 ns	120 ns	90 ns	45 ns
100 ns	100 ns	75 ns	38 ns
80 ns	80 ns	60 ns	30 ns
60 ns	60 ns	50 ns	25 ns

3.6.4 FPDP Mode Control Registers SR0, SR1

There are few options in the FPDP like Rounding, I/O, IEEE handling, Latency and others that can be controlled by writing into the control registers SR0 and SR1.

After reset and whenever the user changes the relevant fields in the FSR, the FPC updates the FPDP control registers.

Fast/IEEE Mode SR0 bit 0

"1" Set to Fast mode. An underflowed instruction with disabled underflow exception delivers zero to the destination register.

3.0 Functional Description (Continued)

Rounding

SR0 Bit-2	SR0 Bit-1	Rounding Mode
0	0	Round toward nearest value, if tie round toward even significant
0	1	Round toward zero
1	0	Round toward positive infinity
1	1	Round toward negative infinity

IntAbortOn SR0 Bit-3

"0" Internal abort off.

SR0 Bit-4

"0"

IlokOn SR0 Bit-5

"0" Disables Interlocks.

FpexSticky SR0 Bit-6

"0" FPEX is "Pulsed". In this mode, FPEX is asserted for one clock cycle.

Multiply Latency SR0 Bit-7

The FPDP has two multiply latency modes: Two cycle latency mode and Three cycle latency mode. See Section 3.6.3.

SR0 Bit-7	Latency Mode
0	Two Cycle Latency Mode
1	Three Cycle Latency Mode

I/O Mode SR1 Bits 4–0

0 0 0 0 0 Single-Pump Undelayed

The FPDP is used by the FPC in the undelayed single-pump mode for load and store operations.

FpexDelay SR1 Bit-5

"1" Delayed FPEX-Mode.

BypassOn SR1 Bit-6

"1" Enables bypassing of operands between instructions.

SR1 Bit-7

"0"

3.6.5 IEEE Enable Register SR2

The SR2 register has enable bits for each of the exception conditions. The FPC updates the enable bits after Reset and whenever the user changes the relevant bits in the FSR. (See LFSR Instruction.)

	7							0
SR2 ENABLES	NaN	Inv	Dvz	Dnrm	Ovf	Unf	Inx	Iovf

FIGURE 3-13. IEEE Enable Register (FPDP)

FPC updates the Inv, Dvz, Ovf and Iovf, Unf, Inx enable bits to reflect those enable bits in the FSR.

The NaN bit is affected by the ROE bit in the FSR. If the ROE is cleared then NaN should be enabled (signal exception upon detection of NaN). If ROE is set NaN will be disabled.

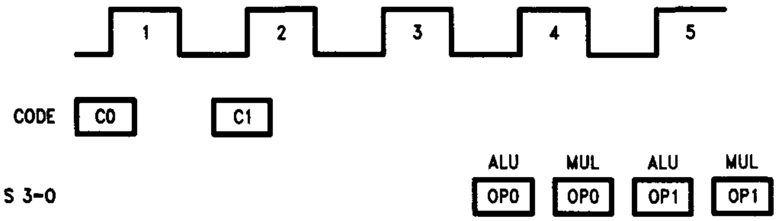
The Dnrm bit is always enabled and detection of Dnrm as operand for operation will cause a source exception.

Whenever the user changes the enable bit in the FSR, the same bit will be updated in the exception enable register in the FPDP.

Registers SR3–SR11 are not used by the FPC.

3.6.5.1 FPDP Status Lines (S0–3)

The status of operation in the FPDP can be obtained by using the FPDP status lines. The status is not "sticky", therefore, the FPC has to sample the status lines in the correct timing. If ALU and MULT instructions end in the same cycle, the ALU status is valid at the end of the cycle and the MULT status is valid at the beginning of the following cycle.



TL/EE/9421-16

FIGURE 3-14. FPDP Status Timing

3.0 Functional Description (Continued)

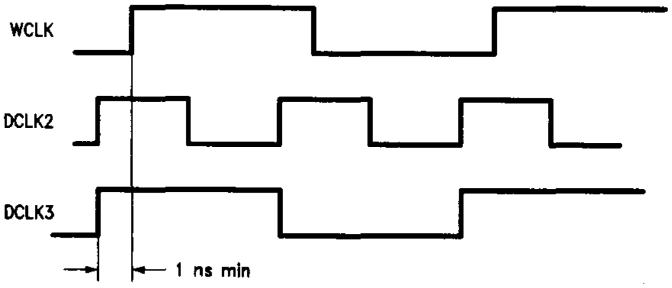
3.6.6 FPDP Clocking Requirements

The FPC uses BLCK and $\overline{\text{BLCK}}$ from the CPU to generate the clock signals required by the FPDP.

The FPDP requires two clock signals: DIVCLK and WCLK. DIVCLK is used by the DIVIDE/SQRT unit, while WCLK is used by all other functional units. The frequency of DIVCLK is dependent on the latency mode selected. It is either same or twice the frequency of WCLK for the "Three Cycle Latency" or "Two Cycle Latency" Modes respectively.

The WCLK frequency is always half the frequency of BCLK.

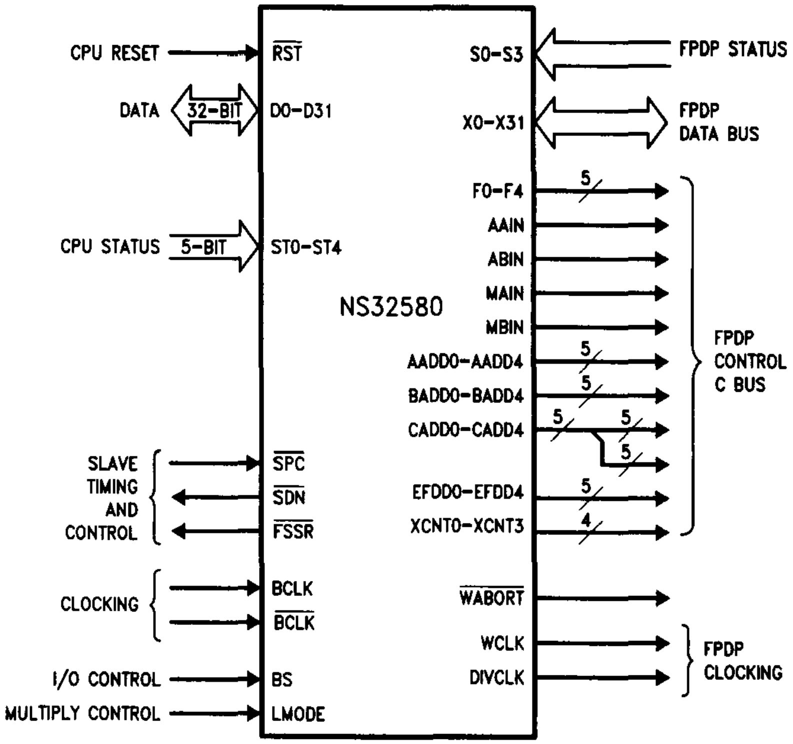
The FPC determines the DIVCLK frequency by using the LMODE pin.


WCLK
DCLK2
DCLK3
1 ns min

TL/EE/9421-17

FIGURE 3-15. Divide/Sqrt Clock DCLK2/DCLK3

4.0 Device Specifications


CPU RESET
RST
DATA
32-BIT
D0-D31
CPU STATUS
5-BIT
ST0-ST4
NS32580
SLAVE TIMING AND CONTROL
SPC
SDN
FSSR
CLOCKING
BCLK
BCLK
I/O CONTROL
BS
MULTIPLY CONTROL
LMODE
S0-S3
FPDP STATUS
X0-X31
FPDP DATA BUS
F0-F4
5
AAIN
ABIN
MAIN
MBIN
AADD0-AADD4
5
BADD0-BADD4
5
CADD0-CADD4
5
5
5
EFDD0-EFDD4
5
XCNT0-XCNT3
4
FPDP CONTROL C BUS
WABORT
WCLK
DIVCLK
FPDP CLOCKING

TL/EE/9421-18

FIGURE 4-1. NS32580 Interface Signals

4.0 Device Specifications (Continued)

4.1 NS32580 PIN DESCRIPTIONS

Descriptions of the NS32580 pins are given in the following sections. *Figure 4-1* shows the NS32580 interface signals grouped according to related functions.

4.1.1 Supplies

VCCL1-7	**Logic Power**— + 5V positive supplies for on-chip logic.
VCCB1-8	**Buffers Power**— + 5V positive supplies for on-chip buffers.
GNDL1-13	**Logic Ground**—Ground references for on-chip logic.
GNDB1-13	**Buffers Ground**—Ground references for on-chip buffers.

4.1.2 Input Signals

BCLK	**Bus Clock**—Input clock from NS32532.
$\overline{\text{BCLK}}$	**Bus Clock Inverse**—Inverted input clock from NS32532.
BS	**Byte Sex**—Specifies the I/O byte ordering of the FPC. If connected to GND the FPC is in Little Endian mode. If connected to V_{CC} the FPC is in Big Endian mode. The BS line must be valid during and after Reset. See Section 3.5.2.
LMODE	**Latency Mode**—Specifies the latency mode of the FPC-FPDP. If connected to GND the FPC-FPDP is in the "Two cycle latency", if connected to V_{CC} the FPC-FPDP is in the "Three cycle latency". LMODE line must be valid during and after Reset.
$\overline{\text{RST}}$	**Reset**—Active low. Resets the last operation, clears the FIFOs and the FSR register to its default state.
S0-3	**FPDP Status**—Indicates any exceptions or conditions that resulted from operations performed by the WTL 3164 floating-point data path.
$\overline{\text{SPC}}$	**Slave Processor Control**—Active low. Data strobe for slave transfers between the CPU and the FPC.
ST0-4	**CPU Status**—Bus cycle status code from CPU. ST0 is the least significant and rightmost bit. 1 1 1 0 0 —Reserved 1 1 1 0 1 —Transferring Operand 1 1 1 1 0 —Reading Status Word 1 1 1 1 1 —Broadcasting Slave ID
AADD0-4	**A Read Port Register Address**—Chooses the inputs to the A bus of the FPDP.
AAIN	**ALU A Input Select**—Controls the A input multiplexers of the FPDP ALU.
ABIN	**ALU B Input Select**—Controls the B input multiplexers of the FPDP ALU.
BADD0-4	**B Read Port Register Address**—Chooses the inputs to the B bus of the FPDP.
CADD0-4	**C Write Port Register Address**—C/D Bus Control. Chooses the destinations of C and D buses. These signals should be connected to both the (CADD0-4) and the (DADD0-4) lines of the FPDP.

4.1.3 Output Signals

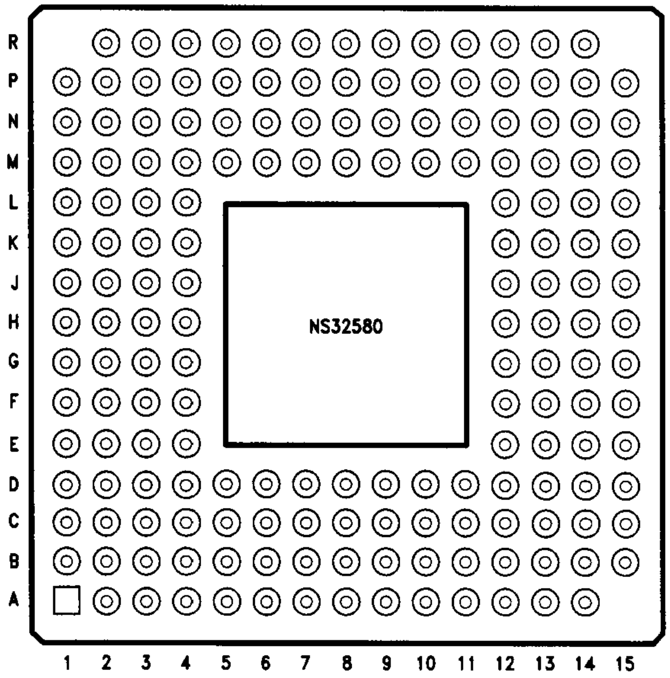
DIVCLK	**Divide/Square Root Clock**—Clock signal for the Divide/Sqrt unit in the FPDP.
EFDD0-4	**E and/or F Port Register Address**—Chooses the source and destination for the Load/Store operations of the FPDP.
F0-4	**Function Code**—Specifies the operation to be performed by the FPDP.
$\overline{\text{FSSR}}$	**Forced Slave Status Read**—Active low. When active, indicates that the FPC status word should be read by the CPU. It is floating before and after being active.
MAIN	**Multiplier A Input Select**—Controls the A input multiplexers of the FPDP multiplier.
MBIN	**Multiplier B Input Select**—Controls the B input multiplexers of the FPDP multiplier.
$\overline{\text{SDN}}$	**Slave Done**—Active low. When active, indicates successful completion by the FPC-FPDP of a floating-point instruction. It is floating before and after being active.
$\overline{\text{WABORT}}$	**FPDP Abort**—Aborts the current and previous instructions in the FPDP.
WCLK	**FPDP Clock**—Clock signal for the FPDP. It is BCLK divided by two. i.e., if BCLK is 30 MHz, WCLK will be 15 MHz.
XCNT0-3	**X Port Control**—They are the Load/Store controls for the FPDP.

4.1.4 Input/Output Signals

D0-31	**CPU Data Bus**—Data bus between FPC and the CPU.
X0-31	**FPDP Data Bus**—Data bus between FPC and the FPDP X port.

4.0 Device Specifications (Continued)

Connection Diagram


R
P
N
M
L
K
J
H
G
F
E
D
C
B
A
NS32580
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

TL/EE/9421-31

Bottom View

FIGURE 4-2. 172-Pin PGA Package

Order Number NS32580-20, NS32580-25 or NS32580-30
See NS Package Number U172B

4.0 Device Specifications (Continued)

NS32580 Pinout Descriptions

Desc	Pin
VCCL1	A2
GNDB1	A3
GNDL1	A4
XCNT0	A5
XCNT3	A6
EFADD1	A7
EFADD2	A8
GNDL2	A9
GNDB2	A10
CADD0	A11
CADD2	A12
CADD3	A13
BADD0	A14
GNDB3	B1
GNDL3	B2
X0	B3
XCNT1	B4
XCNT2	B5
EFADD0	B6
EFADD3	B7
$\overline{\text{BCLK}}$	B8
WCLK	B9
DIVCLK	B10
EFADD4	B11
CADD1	B12
CADD4	B13
BADD1	B14
BADD2	B15
VCCB1	C1
X2	C2
X1	C3
VCCL2	C4
D1	C5
D0	C6
NC	C7
GNDL4	C8
GNDB4	C9
BCLK	C10
$\overline{\text{RST}}$	C11
NC	C12
BADD3	C13
AADD0	C14
BADD4	C15
X3	D1
X4	D2
NC	D3
D2	D4
D17	D5
D16	D6
NC	D7
GNDL5	D8
NC	D9
NC	D10
NC	D11
VCCB2	D12
D15	D13
AADD1	D14
AADD2	D15
X5	E1
X7	E2
D18	E3
D3	E4
D31	E12
D14	E13
AADD3	E14
AADD4	E15
X6	F1
X9	F2
D19	F3
VCCL3	F4
D30	F12
VCCB3	F13
MAIN	F14
MBIN	F15
X8	G1
X10	G2
D4	G3
D20	G4
D13	G12
D29	G13
AAIN	G14
ABIN	G15
X11	H1
X12	H2
NC	H3
D5	H4
D28	H12
GNDB5	H13
F0	H14
F1	H15
X13	J1
X15	J2
GNDB6	J3
D21	J4
D12	J12
D27	J13
F2	J14
F3	J15
X14	K1
X17	K2
D6	K3
D22	K4
D11	K12
NC	K13
SO	K14
F4	K15
X16	L1
X18	L2
D7	L3
D23	L4
$\overline{\text{SPC}}$	L12
$\overline{\text{SDN}}$	L13
S2	L14
S1	L15
X19	M1
Reserved	M2
VCCL4	M3
D8	M4
GNDB7	M5
D26	M6
GNDL6	M7
VCCB4	M8
NC	M9
ST0	M10
ST1	M11
NC	M12
GNDL7	M13
$\overline{\text{WABORT}}$	M14
S3	M15
VCCL5	N1
GNDB8	N2
Reserved	N3
D24	N4
D25	N5
D9	N6
D10	N7
NC	N8
VCCB5	N9
ST2	N10
ST4	N11
$\overline{\text{FSSR}}$	N12
GNDB9	N13
VCCB6	N14
GNDL8	N15
GNDL9	P1
VCCL6	P2
X21	P3
X23	P4
X25	P5
X26	P6
X28	P7
X31	P8
X30	P9
BS	P10
ST3	P11
VCCB7	P12
GNDB10	P13
GNDL10	P14
GNDB11	P15
GNDB12	R2
GNDL11	R3
VCCL7	R4
X20	R5
X22	R6
X24	R7
X27	R8
X29	R9
LMODE	R10
GNDB13	R11
GNDL12	R12
VCCB8	R13
GNDL13	R14

Note: NC = No Connection

4.0 Device Specifications (Continued)

4.2 ABSOLUTE MAXIMUM RATINGS

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias	0°C to +70°C
Storage Temperature	−65°C to +150°C
All Input or Output Voltages with Respect to GND	−0.5V to +7V
Power Dissipation	1.5W

ESD Rating is to be determined.

Note: *Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under Electrical Characteristics.*

4.3 ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND = 0V

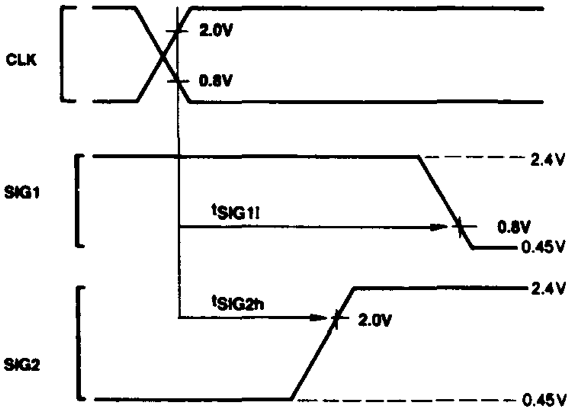
Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage		2.0		$V_{CC} + 0.5$	V
V_{IL}	Low Level Input Voltage		−0.5		0.8	V
V_{OH}	High Level Output Voltage	$I_{OH} = -400\ \mu A$	2.4			V
V_{OL}	Low Level Output Voltage	$I_{OL} = 2\ mA$			0.4	V
I_I	Input Load Current	$0 \leq V_{IN} \leq V_{CC}$				
I_L	Leakage Current (Output and I/O Pins in TRI-STATE®/Input Mode)	$0.4 \leq V_{OUT} \leq 2.4V$				
I_{CC}	Active Supply Current	$I_{OUT} = 0$, $T_A = 25°C$			300	mA
C_{IN}	Input Capacitance					pF
C_{OUT}	Output Capacitance					pF

4.4 SWITCHING CHARACTERISTICS

4.4.1 Definitions

All the Timing Specifications given in this section refer to 0.8V and 2.0V on all the input and output signals as illustrated in *Figures 4.3* and *4.4*, unless specifically stated otherwise.

Note: These voltage levels shown are for 32532-32580 interface only. Levels for 32580-WTL3164 interface are specified in their appropriate timing diagrams.

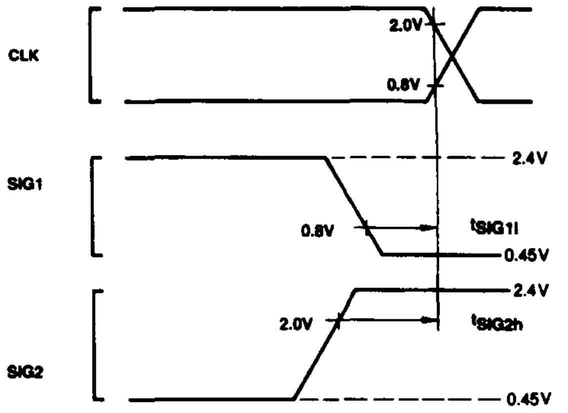


TL/EE/9421-19

FIGURE 4-3. Timing Specification Standard (Signal Valid after Clock Edge)

ABBREVIATIONS

L.E. — Leading Edge	R.E. — Rising Edge
T.E. — Trailing Edge	F.E. — Falling Edge



TL/EE/9421-20

FIGURE 4-4. Timing Specification Standard (Signal Valid before Clock Edge)

4.0 Device Specifications (Continued)

4.4.2 Timing Tables Maximum times assume temperature range 0°C to 70°C

4.4.2.1 Output Signal Propagation Delays Maximum times assume capacitive loading of 100 pF

Symbol	*Figure*	Description	Reference/ Conditions	NS32580-20		NS32580-25		NS32580-30		Units
				Min	Max	Min	Max	Min	Max	
t_{Dv}	*4-8*	CPU Data Valid	After R.E., BCLK T2		35		27		23	ns
t_{Doh}	*4-8*	CPU Data Hold	After R.E., BCLK Next T1/Ti	2		2		2		ns
t_{Dnf}	*4-8*	CPU Data Not Forcing	After R.E., BCLK Next T1/Ti		28		23		19	ns
t_{SDa}	*4-10*	$\overline{\text{SDN}}$ Signal Active	After R.E., BCLK		35		28		22	ns
t_{SDia}	*4-10*	$\overline{\text{SDN}}$ Signal Inactive	After R.E., Next BCLK	2		2		2		ns
t_{SDnf}	*4-10*	$\overline{\text{SDN}}$ Signal Not Forcing	After R.E., BCLK		25		20		17	ns
t_{FSSRa}	*4-11*	$\overline{\text{FSSR}}$ Signal Active	After R.E., BCLK		35		28		22	ns
t_{FSSRia}	*4-11*	$\overline{\text{FSSR}}$ Signal Inactive	After R.E., Next BCLK	2		2		2		ns
t_{FSSRnf}	*4-11*	$\overline{\text{FSSR}}$ Signal Not Forcing	After R.E., BCLK		25		20		17	ns
t_{Cv}	*4-14*	C Bus	After R.E., WCLK		83		63		50	ns
t_{ABRTv}	*4-14*	$\overline{\text{WABORT}}$ Valid	After R.E., WCLK		83		63		50	ns
t_{Ch}	*4-14*	C BUS	After R.E., WCLK	3		3		3		ns
t_{ABRTh}	*4-14*	$\overline{\text{WABORT}}$ Hold Time	After R.E., WCLK	5		5		5		ns
t_{XLv}	*4-14*	FPDP Data Valid	After R.E., WCLK		83		63		50	ns
t_{XLh}	*4-14*	FPDP Data Hold Time	After R.E., WCLK	3		3		3		ns
t_{D2p}	*4-13*	DCLK2 Period	From 2.0V R.E., to 2.0V R.E.	50		40		33.3		ns
t_{D2h}	*4-13*	DCLK2 High Time	From 2.0V R.E., to 0.8V F.E.	22		17		14.5		ns
t_{D2l}	*4-13*	DCLK2 Low Time	From 0.8V F.E. to 2.0V R.E.	22		17		14.5		ns
t_{D3p}	*4-13*	DCLK3 Period	From 2.0V R.E., to 2.0V R.E.	100		80		66.6		ns
t_{D3h}	*4-13*	DCLK3 High Time	From 2.0V R.E., to 0.8V F.E.	45		36		30		ns
t_{D3l}	*4-13*	DCLK3 Low Time	From 0.8V F.E., to 2.0V R.E.	45		36		30		ns
t_{WCLKp}	*4-13*	WCLK Period	From 2.0V R.E., to 2.0V R.E.	100		80		66.6		ns
t_{WCLKh}	*4-13*	WCLK High Time	From 2.0V R.E., to 0.8V F.E.	45		36		30		ns
t_{WCLKl}	*4-13*	WCLK Low Time	From 0.8V F.E. to 2.0V R.E.	45		36		30		ns
t_{DWd}	*4-13*	DCLK2/DCLK3 to WCLK Delay	From 2.0V R.E., to 2.0V R.E.	1	8	1	8	1	8	ns
t_{Wr}	*4-13*	FPDP Clock Rise Time	From 0.8V R.E., to 2.4V R.E.		4		4		4	ns
t_{Wf}	*4-13*	FPDP Clock Fall Time	From 2.4V F.E. to 0.8V F.E.		4		4		4	ns

4.4.2.2 Input Signal Requirements NS32580-20, NS32580-25, NS32580-30

Symbol	*Figure*	Description	Reference/ Conditions	NS32580-20		NS32580-25		NS32580-30		Units
				Min	Max	Min	Max	Min	Max	
t_{BCp}	*4-5*	BCLK Period	R.E., BCLK to Next R.E., BLCK	50	100	40	100	33.3	100	ns
t_{BCh}	*4-5*	BCLK High Time	At 2.0V on BCLK (Both Edges)	$0.5\ t_{BCp} - 5$		$0.5\ t_{BCp} - 4$		$0.5\ t_{BCp} - 3$		
t_{BCl}	*4-5*	BCLK Low Time	At 0.8V on BCLK (Both Edges)	$0.5\ t_{BCp} - 5$		$0.5\ t_{BCp} - 4$		$0.5\ t_{BCp} - 3$		
t_{BCr}	*4-5*	BCLK Rise Time	0.8V to 2.0V on R.E., BCLK		5		4		3	ns
t_{BCf}	*4-5*	BCLK Fall Time	2.0V to 0.8V on F.E., BCLK		5		4		3	ns
t_{NBCp}	*4-5*	$\overline{\text{BCLK}}$ Period	R.E., $\overline{\text{BCLK}}$ to Next R.E., $\overline{\text{BCLK}}$	50	100	40	100	33.3	100	ns

4.0 Device Specifications (Continued)

4.4.2.2 Input Signal Requirements NS32580-20, NS32580-25, NS32580-30 (Continued)

Symbol	*Figure*	Description	Reference/ Conditions	NS32580-20		NS32580-25		NS32580-30		Units
				Min	Max	Min	Max	Min	Max	
t_{NBCh}	*4-5*	$\overline{\text{BCLK}}$ High Time	At 2.0V on $\overline{\text{BCLK}}$ (Both Edges)	0.5 t_{NBCp} −5		0.5 t_{NBCp} −4		0.5 t_{NBCp} −3	120	ns
t_{NBCl}	*4-5*	$\overline{\text{BCLK}}$ Low Time	At 0.8V on $\overline{\text{BCLK}}$ (Both Edges)	0.5 t_{NBCp} −5		0.5 t_{NBCp} −4		0.5 t_{NBCp} −3	120	ns
t_{NBCr}	*4-5*	$\overline{\text{BCLK}}$ Rise Time	0.8V to 2.0V on R.E., $\overline{\text{BCLK}}$		5		4		3	ns
t_{NBCf}	*4-5*	$\overline{\text{BCLK}}$ Fall Time	2.0V to 0.8V on F.E., $\overline{\text{BCLK}}$		5		4		3	ns
$t_{BCNBCrf}$	*4-5*	Bus Clock Skew	2.0V on R.E., BCLK to 0.8V on F.E., $\overline{\text{BCLK}}$	−2	+2	−2	+2	−1	+1	ns
$t_{BCNBCfr}$	*4-5*	Bus Clock Skew	0.8V on F.E., BCLK to 2.0V on R.E., $\overline{\text{BCLK}}$	−2	+2	−2	+2	−1	+1	ns
t_{PWR}	*4-6*	Power Stable to R.E. of $\overline{\text{RST}}$	After V_{CC} Reaches 4.5V	50		40		30		µs
t_{RSTs}	*4-6, 4-7*	$\overline{\text{RST}}$ Setup Time	Before R.E., BCLK	14		12		11		ns
t_{RSTw}	*4-7*	$\overline{\text{RST}}$ Pulse Width	At 0.8V (Both Edges)	64		64		64		t_{BCp}
t_{STs}	*4-8, 4-9*	CPU Status Setup Time	Before R.E., BCLK T2	36		30		24		ns
t_{STh}	*4-8, 4-9*	CPU Status Hold Time	After R.E., BCLK T2	15		12		10		ns
t_{SPCs}	*4-8, 4-9*	$\overline{\text{SPC}}$ Setup Time	Before R.E., BCLK T2	30		23		20		ns
t_{SPCh}	*4-8, 4-9*	$\overline{\text{SPC}}$ Hold Time	After R.E., BCLK T2	0	t_{BCp} +19	0	t_{BCp} +15	0	t_{BCp} +12	ns
t_{Ds}	*4-9*	Data Setup Time	Before R.E., BCLK T2	7		5		3		ns
t_{Dh}	*4-9*	Data Hold Time	After R.E., BCLK Next T1 or Ti	−4		−4		−4		ns
t_{SAs}	*4-12*	FPDP ALU Status Setup Time	Before R.E., WCLK	9		9		8		ns
t_{SAh}	*4-12*	FPDP ALU Status Hold Time	After R.E., WCLK	5		5		5		ns
t_{SMs}	*4-12*	FPDP Multiplier Status Setup Time	Before F.E., WCLK	9		9		8		ns
t_{SMh}	*4-12*	FPDP Multiplier Status Hold Time	After F.E., WCLK	5		5		5		ns
t_{XSs}	*4-14*	FPDP Data Setup Time	Before R.E., WCLK	9		9		9		ns
t_{XSh}	*4-14*	FPDP Data Hold Time	After R.E., WCLK	5		5		5		ns

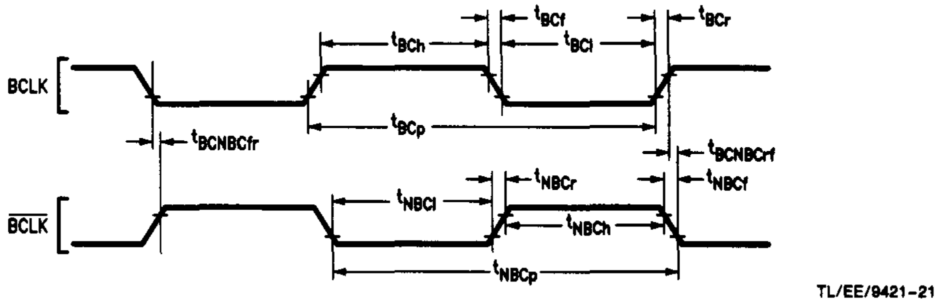
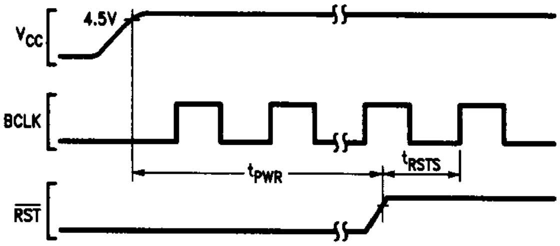


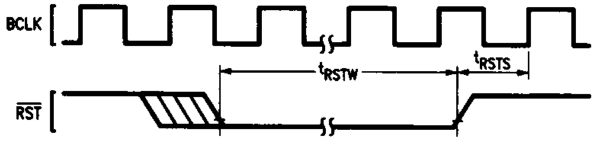
FIGURE 4-5. Clock Waveforms

4.0 Device Specifications (Continued)


V_CC
4.5V
BCLK
RST
t_PWR
t_RSTS

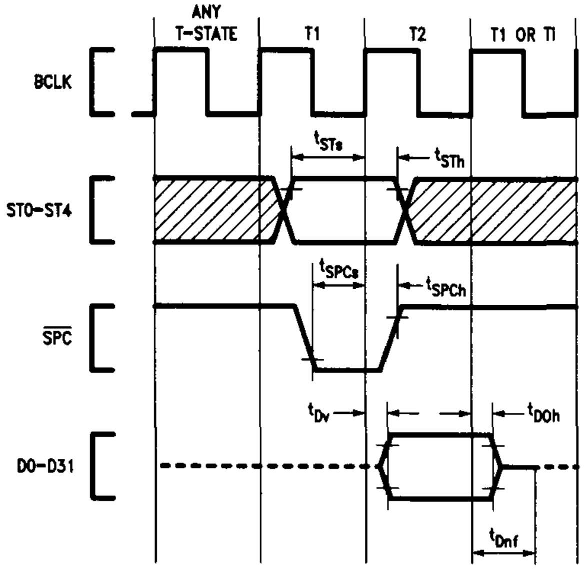
TL/EE/9421-22

FIGURE 4-6. Power-On Reset


BCLK
RST
t_RSTW
t_RSTS

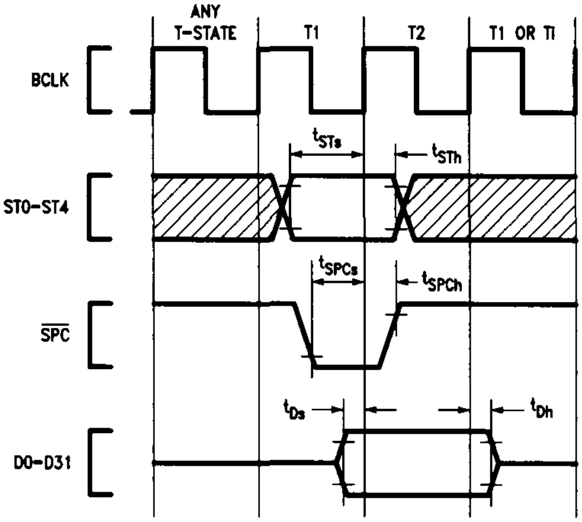
TL/EE/9421-23

FIGURE 4-7. Non-Power-On Reset


ANY T-STATE
T1
T2
T1 OR TI
BCLK
ST0-ST4
SPC
D0-D31
t_STs
t_STh
t_SPCs
t_SPCh
t_Dv
t_DOh
t_Dnf

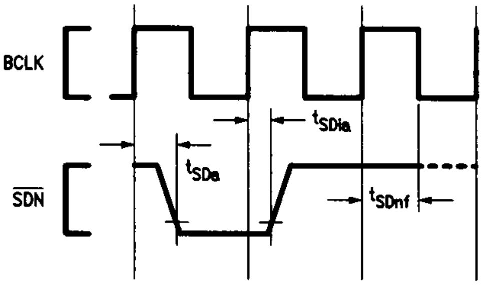
TL/EE/9421-24

FIGURE 4-8. Read Cycle from FPC


ANY T-STATE
T1
T2
T1 OR TI
BCLK
ST0-ST4
SPC
D0-D31
t_STs
t_STh
t_SPCs
t_SPCh
t_Ds
t_Dh

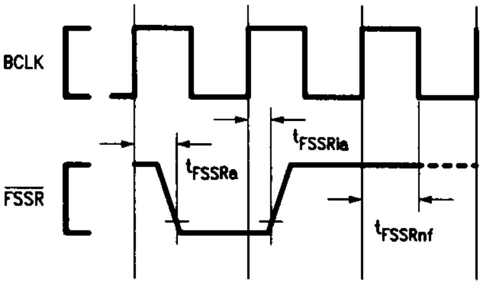
TL/EE/9421-25

FIGURE 4-9. Write Cycle to FPC


BCLK
SDN
t_SDia
t_SDa
t_SDnf

TL/EE/9421-26

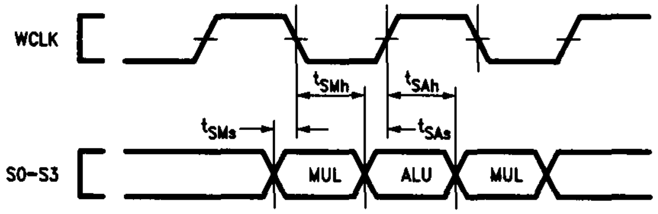
FIGURE 4-10. Slave Processor Done Timing


BCLK
FSSR
t_FSSRia
t_FSSRa
t_FSSRnf

TL/EE/9421-27

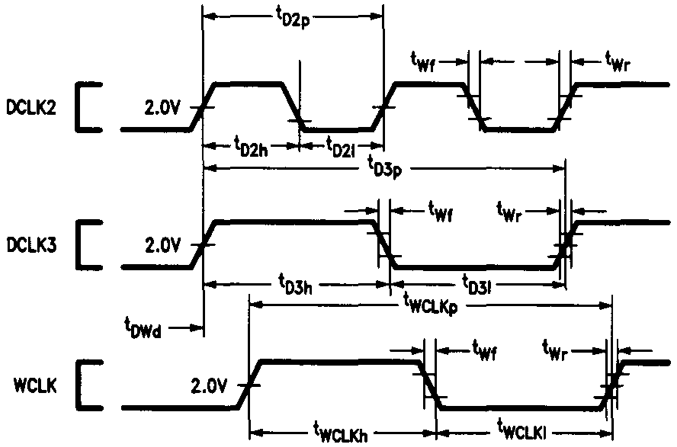
FIGURE 4-11. FSSR Signal Timing

4.0 Device Specifications (Continued)


WCLK
tSMh
tSAh
tSMs
tSAs
S0–S3
MUL
ALU
MUL

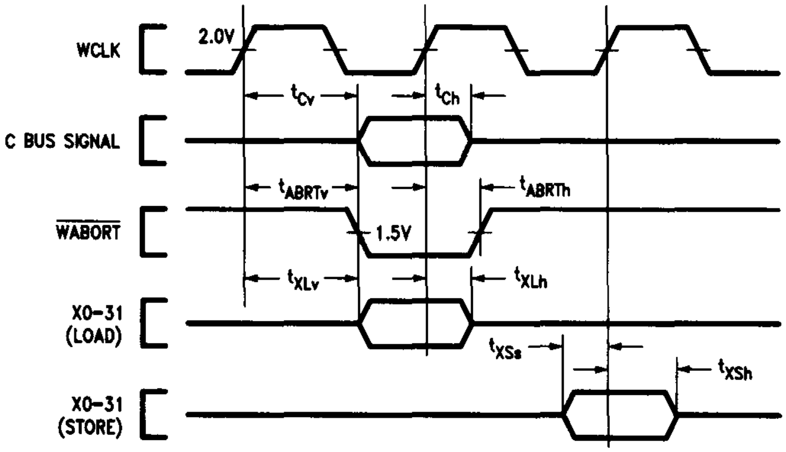
TL/EE/9421-28

FIGURE 4-12. FPDP Status Signal Timing


tD2p
tWf
tWr
DCLK2
2.0V
tD2h
tD2l
tD3p
tWf
tWr
DCLK3
2.0V
tD3h
tD3l
tWCLKp
tDWd
tWf
tWr
WCLK
2.0V
tWCLKh
tWCLKl

TL/EE/9421-29

FIGURE 4-13. FPDP Clock Signal Timing


WCLK
2.0V
tCv
tCh
C BUS SIGNAL
tABRTv
tABRTh
WABORT
1.5V
tXLv
tXLh
X0–31 (LOAD)
tXSs
tXSh
X0–31 (STORE)

TL/EE/9421-30

FIGURE 4-14. FPDP Output Signal Timing

Appendix A

COMPATIBILITY OF FPC-FPDP WITH NS32081/NS32381

NS32081	NS32381	NS32580
INSTRUCTIONS		
	NS32081 + DOTf POLYf SCALBf LOGBf	NS32081 + MACf SQRTf
REGISTERS		
8 x 32 Bit	8 x 64 Bit	8 x 64 Bit
RESERVED OPERANDS		
DNRM	DNRM	DNRM*
NaN	NaN	NaN can be enabled or Disable.*
Infinity	Infinity	Infinity is NOT a reserved operand.*

NS32081	NS32381	NS32580
FSR		
	NS32081 FSR + RMB	NS32081 FSR + RMB ROE IVE DZE OVE IOE ROF IVF DZF OVF IOF

*See compatibility table for special cases.

Compatibility Table

Special Case	NS32081/NS32381	NS32580
ROUNDfi (infinity)	TRAP (INV)	TRAP (OVF), IOF = 1
TRUNCfi (infinity)	TRAP (INV)	TRAP (OVF), IOF = 1
FLOORfi (infinity)	TRAP (INV)	TRAP (OVF), IOF = 1
DIVf 0, infinity	TRAP (DVZ)	Result = infinity
SQRTf (−DNRM)	TRAP (INV)	TRAP (INV), ROF = 0, IVF = 1
DIVf 0, DNRM	TRAP (INV)	TRAP (DVZ)
MULf (0, DNRM) or (DNRM, 0)	TRAP (INV)	Result = 0
DIVf DNRM, 0	TRAP (INV)	Result = 0
DIVf infinity, DNRM	TRAP (INV)	Result = 0
DIVf DNRM, infinity	TRAP (INV)	Result = infinity
MULf (infinity, DNRM) or (DNRM, infinity)	TRAP (INV)	Result = infinity
FSR.ROE = 1 and		
NEGf (NaN)	N/A	Result = −NaN
ABSf (NaN)	N/A	Result = |NaN|
ADDF	N/A	Result = NaN
SUBf Nan, DNRM	N/A	
MULf or	N/A	
DIVf DNRM, NaN	N/A	
MACf	N/A	

Appendix B

PERFORMANCE ANALYSIS

The execution time is calculated from $\overline{\text{SPC}}$ (T1, T2 included) to $\overline{\text{SDN}}$ (including the $\overline{\text{SDN}}$ pulse)

Instruction	Latency reg, reg 2 cycles mode	Latency reg, reg 3 cycles mode	Throughput reg, reg 2 cycles mode	Throughput reg, reg 3 cycles mode	Pipe Break
ADDf/l	13	13	2	2	No
SUBf/l	13	13	2	2	No
MULf	13	13	2	2	No
MULl	13	15	2	4	No
DIVf	29	43	29	43	No
DIV1	43	71	43	71	No
MOVf/l	13	13	2	2	No
ABSf/l	13	13	2	2	No
NEGf/l	13	13	2	2	No
CMPf/l	13 + CPU	13 + CPU	—	—	Yes
FLOORfi	13 + CPU	13 + CPU	—	—	Yes
TRUNCfi	13 + CPU	13 + CPU	—	—	Yes
ROUNDfi	13 + CPU	13 + CPU	—	—	Yes
MOVFL	13 + CPU	13 + CPU	—	—	Yes
MOVLF	13 + CPU	13 + CPU	—	—	Yes
MOVif	17 + CPU	17 + CPU	—	—	Yes
MOVil	13 + CPU	13 + CPU	—	—	Yes
LFSR	13	13	—	—	Yes
SFSR	13 + CPU	13 + CPU	—	—	Yes
MACf	17	17	6	6	No
MACl	17	19	6	8	No
SQRTf	41	65	41	65	No
SQRTl	69	123	69	123	No

Appendix B (Continued)

Add the following CPU cycles to the base (reg, reg) number of cycles for the different cases:

Instruction	Latency 2 Cycles Mode	Latency 3 Cycles Mode	Throughput 2 Cycles Mode	Throughput 3 Cycles Mode	Pipe Break
MONADIC FLOAT (One Operand)					
mem, reg	0	0	2	2	see reg, reg
reg, mem	0 + CPU	0 + CPU	—	—	Yes
mem, mem	0 + CPU	0 + CPU	—	—	Yes
DYADIC FLOAT (Two Operands)					
mem, reg	0	0	2	2	see reg, reg
reg, mem	0 + CPU	0 + CPU	—	—	Yes
mem, mem	2 + CPU	2 + CPU	—	—	Yes
MONADIC LONG (One Operand)					
mem, reg	2	2	4	4	see reg, reg
reg, mem	2 + CPU	2 + CPU	—	—	Yes
mem, mem	2 + CPU	2 + CPU	—	—	Yes
DYADIC LONG (Two Operands)					
mem, reg	2	2	4	4	see reg, reg
reg, mem	6 + CPU	6 + CPU	—	—	Yes
mem, mem	6 + CPU	6 + CPU	—	—	Yes

Note: CPU stands for the time it takes the CPU to take the result from the FPC and resume operation.

This datasheet has been downloaded from:

www.DatasheetCatalog.com

Datasheets for electronic components.

National Semiconductor was acquired by Texas Instruments.

http://www.ti.com/corp/docs/investor_relations/pr_09_23_2011_national_semiconductor.html

This file is the datasheet for the following electronic components:

NS32580-30 - http://www.ti.com/product/ns32580-30?HQS=TI-null-null-dscatalog-df-pf-null-wwe

NS32580-20 - http://www.ti.com/product/ns32580-20?HQS=TI-null-null-dscatalog-df-pf-null-wwe

NS32580-25 - http://www.ti.com/product/ns32580-25?HQS=TI-null-null-dscatalog-df-pf-null-wwe