National Semiconductor

PRELIMINARY
April 1988

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

General Description

The NS32580 Floating-Point Controller (FPC) is an interface controller 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. The performance of the NS32580 (FPC) and the WTL 3164 (FPDP) with the NS32532 has been significantly enhanced for high-performance floating-point applications. It reaches the peak performance of 15 Mflops when executing single and double precision ADD, SUB, MUL, and MAC instructions in a pipelined mode while maintaining precise exception handling.

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. It supports Infinity and Not a Number (NaN) and can flush the result to zero or trap on underflowed instructions.

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. All addressing modes, including two address operations, are available with the floating-point instructions. In addition, 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
- 15 Mflops peak performance for single and double precision ADD, SUB, MUL and MAC instructions with the Weitek WTL 3164 FPDP
- Conforms to IEEE 754—1985 standard for Binary Floating-Point Arithmetic
- 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

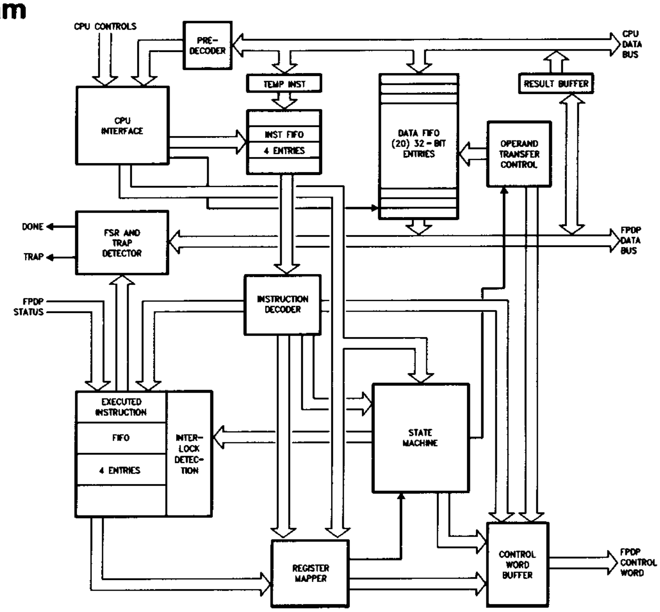


FIGURE 1-1

Series 32000® and TRI-STATE® are registered trademarks of National Semiconductor Corporation.

RRD-B20M48/Printed in U. S. A.

Table of Contents

List of Illustrations

List of Tables

1.0 Product Information

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 infinities 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 and Not-a-Number (NaN) (Section 1.2.3)

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

The remaining IEEE features are 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 $1.0 \le x \le 2.0$, 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.

1.2.3 Reserved Operands

NaN is a symbolic entity which is defined to have two types, Signaling NaN and Quiet NaN. The NS32580 can accept any form of NaN as operands but will only return Quiet NaN as a result.

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 is fully supported per the IEEE standard. See *Figures 1-3* and *1-4*.

The NS32580 FPC treats only Denormalized numbers as reserved operands if the Floating-Point Status Register ROE bit is set (Section 2.1.2). Denormalized numbers have all zeroes in their E fields and non-zero values in their F fields.

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).

Single Precision

31	30 … 23	22 … 0
S	E	F
1	8	23

Double Precision

63	62 … 52	51 … 0
S	E	F
1	11	52

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 → Quiet NaN Returned as Result
255	0	$(-1)^s$ * Infinity	Infinity	Not a Reserved Operand
1–254	Any	$(-1)^s * 2^{0-127} * (1.f)$	Normalized Number	
0	Not 0	$(-1)^s * 2^{-126} * (0.f)$	Denormalized Number	Reserved Operand
0	0	$(-1)^s * 0$	Zero	

31	30 … 23	22 … 0	
0	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Quiet NaN

FIGURE 1-3. Single-Precision Operand E and F Fields with Quiet NaN Format

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

63	62 … 52	51 … 0	
0	1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Quiet NaN

FIGURE 1-4. Double-Precision Operand E and F Fields with Quiet NaN Format

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 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 includes nine registers; 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 FPDP 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 FPU 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 ← 64 → ← 32 →	← 32 →	
F1	L0 MSDW	L0 LSDW	F0
	L1 MSDW	L1 LSDW	
F3	L2 MSDW	L2 LSDW	F2
	L3 MSDW	L3 LSDW	
F5	L4 MSDW	L4 LSDW	F4
	L5 MSDW	L5 LSDW	
F7	L6 MSDW	L6 LSDW	F6
	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 exactly 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.

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 LFSRs and SFSR, 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.

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. Infinities are not reserved operands in the FPC. When ROE is disabled, the FPC does not generate reserved operands as results. Denormalized Numbers (DNRM) are always treated as reserved operands. If Invalid Operation or Divide by Zero or Overflow exceptions are disabled, the ROE bit is overwritten internally (the FPC does not change the ROE bit in the FSR) and the FPC can accept or generate quiet NaN as results. 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, quiet 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.

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 (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 OVERFLOW (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	
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.

TABLE 2-1. FSR Default State Summary (Continued)

Bit Name	Default Value	Default State
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 General Instruction Format

Figure 2-4 shows the general format of a Series 32000 instruction. The Basic Instruction is one to three bytes long and contains the opcode and up to two 5-bit General Addressing Mode (Gen) fields. Following the Basic Instruction field is a set of optional extensions, which may appear depending on the instruction and the addressing modes selected.

The only form of extension issued to the NS32580 FPC is an Immediate operand. Other extensions are used only by the CPU to reference memory operands needed by the FPC.

Index Bytes appear when either or both Gen fields specify Scaled Index. In this case, the Gen field specifies only the Scale Factor (1, 2, 4 or 8), and the Index Byte specifies which General Purpose Register to use as the index, and which addressing mode calculation to perform before indexing. See *Figure 2-5*.

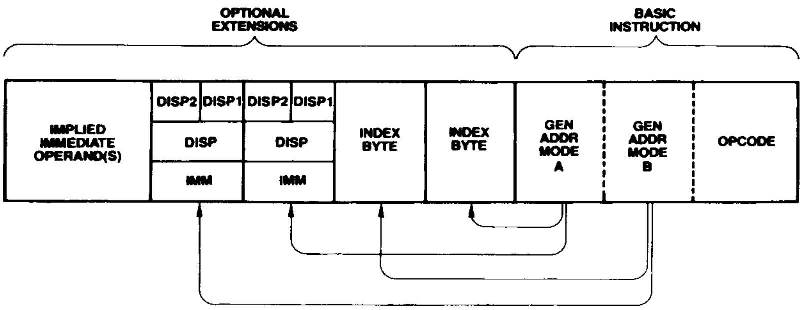
Following Index Bytes come any displacements (addressing constants) or immediate values associated with the selected addressing modes. Each Disp/Imm field may contain one or two displacements, or one immediate value. The size of a Displacement field is encoded within the top bits of that field, as shown in *Figure 2-6*, with the remaining bits interpreted as a signed (two's complement) value. The size of an immediate value is determined from the Opcode field. Both Displacement and Immediate fields are stored most significant byte first.

Some non-FPC instructions require additional, "implied" immediates and/or displacements, apart from those associated with addressing modes. Any such extensions appear at the end of the instruction, in the order that they appear within the list of operands in the instruction definition.

2.2.2 Addressing Modes

The Series 32000 Family CPUs generally access an operand by calculating its Effective Address based on information available when the operand is to be accessed. The method to be used in performing this calculation is specified by the programmer as an "addressing mode."

Addressing modes in the Series 32000 family are designed to optimally support high-level language accesses to variables. In nearly all cases, a variable access requires only



TL/EE/9421-2

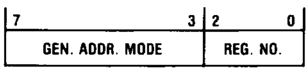
FIGURE 2-4. General Instruction Format

one addressing mode within the instruction which acts upon that variable. Extraneous data movement is therefore minimized.

Series 32000 Addressing Modes fall into nine basic types:

Register: In floating-point instructions, these addressing modes refer to a Floating-Point Register (F0–F7) or (L0–L7) if the operand is of a floating-point type. Otherwise, a CPU General Purpose Register (R0–R7) is referenced. See Section 2.1.1.

Register Relative: A CPU General Purpose Register contains an address to which is added a displacement value from the instruction, yielding the Effective Address of the operand in memory.



TL/EE/9421-3

FIGURE 2-5. Index Byte Format

Memory Space: Identical to Register Relative above, except that the register used is one of the dedicated CPU registers PC, SP, SB or FP. These registers point to data areas generally needed by high-level languages.

Memory Relative: A pointer variable is found within the memory space pointed to by the CPU SP, SB or FP register. A displacement is added to that pointer to generate the Effective Address of the operand.

Immediate: The operand is encoded within the instruction. This addressing mode is not allowed if the operand is to be written. Floating-point operands as well as integer operands may be specified using Immediate mode.

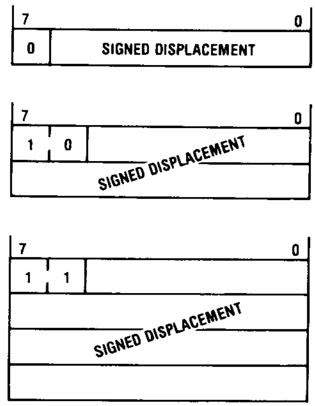
Absolute: The address of the operand is specified by a Displacement field in the instruction.

External: A pointer value is read from a specified entry of the current Link Table. To this pointer value is added a displacement, yielding the Effective Address of the operand.

Top of Stack: The currently-selected CPU Stack Pointer (SP0 or SP1) specifies the location of the operand. The operand is pushed or popped, depending on whether it is written or read.

Scaled Index: Although encoded as an addressing mode, Scaled Indexing is an option on any addressing mode except Immediate or another Scaled Index. It has the effect of calculating an Effective Address, then multiplying any General Purpose Register by 1, 2, 4 or 8 and adding it into the total, yielding the final Effective Address of the operand.

The following table, Table 2-2, is a brief summary of the addressing modes. For a complete description of their actions, see the Series 32000 Instruction Set Reference Manual.



TL/EE/9421-4

FIGURE 2-6. Displacement Encodings

TABLE 2-2. Series 32000 Family Addressing Modes

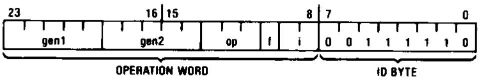
Encoding	Mode	Assembler Syntax	Effective Address
REGISTER			
00000	Register 0	R0, F0 or L0	None: Operand is in the specified register.
00001	Register 1	R1, F1 or L1	
00010	Register 2	R2, F2 or L2	
00011	Register 3	R3, F3 or L3	
00100	Register 4	R4, F4 or L4	
00101	Register 5	R5, F5 or L5	
00110	Register 6	R6, F6 or L6	
00111	Register 7	R7, F7 or L7	
REGISTER RELATIVE			
01000	Register 0 relative	disp(R0)	Disp + Register.
01001	Register 1 relative	disp(R1)	
01010	Register 2 relative	disp(R2)	
01011	Register 3 relative	disp(R3)	
01100	Register 4 relative	disp(R4)	
01101	Register 5 relative	disp(R5)	
01110	Register 6 relative	disp(R6)	
01111	Register 7 relative	disp(R7)	
MEMORY SPACE			
11000	Frame memory	disp(FP)	Disp + Register; "SP" is either SP0 or SP1, as selected in PSR.
11001	Stack memory	disp(SP)	
11010	Static memory	disp(SB)	
11011	Program memory	* + disp	
MEMORY RELATIVE			
10000	Frame memory relative	disp2(disp1(FP))	Disp2 + Pointer; Pointer found at address Disp1 + Register. "SP" is either SP0 or SP1, as selected in PSR.
10001	Stack memory relative	disp2(disp1(SP))	
10010	Static memory relative	disp2(disp1(SB))	
IMMEDIATE			
10100	Immediate	value	None: Operand is issued from CPU instruction queue.
ABSOLUTE			
10101	Absolute	@disp	Disp.
EXTERNAL			
10110	External	EXT (disp1) + disp2	Disp2 + Pointer; Pointer is found at Link Table Entry number Disp1.
TOP OF STACK			
10111	Top of Stack	TOS	Top of current stack, using either User or Interrupt Stack Pointer, as selected in PSR. Automatic Push/Pop included.
SCALED INDEX			
11100	Index, bytes	mode[Rn:B]	Mode + Rn.
11101	Index, words	mode[Rn:W]	Mode + 2 × Rn.
11110	Index, double words	mode[Rn:D]	Mode + 4 × Rn.
11111	Index, quad words	mode[Rn:Q]	Mode + 8 × Rn. "Mode" and "n" are contained within the Index Byte.
10011	(Reserved for Future Use)		

2.2.3 Floating-Point Instruction Set

The NS32580 FPC-FPDP instructions occupy formats 9, 11 and 12 of the Series 32000 Family instruction set *(Figure 2-7)*. A list of all Series 32000 family instruction formats is found in the applicable CPU data sheet.

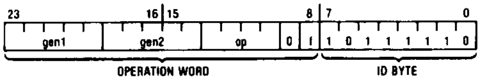
Certain notations in the following instruction description tables serve to relate the assembly language form of each instruction to its binary format in *Figure 2-7*.

Format 9



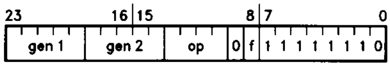
TL/EE/9421-5

Format 11



TL/EE/9421-6

Format 12



TL/EE/9421-7

FIGURE 2-7. Floating-Point Instruction Formats

The Format column indicates which of the three formats in *Figure 2-7* 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 *(Figure 2-7)* 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 (*Figure 2-7*) 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 *(Figure 2-7)*. Refer to Table 2-2 for the options available and their patterns.

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.

	Format	Op	Instruction		Description
(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.

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/TRAPS

An exception for the FPU 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 FPU 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

Each exception or type that is controlled by the user can be set-up to cause an interrupt or to return a result without an interrupt on the occurrence of the exception. The interrupt is called a TRAP and is signaled by the FPU pulsing the $\overline{\text{FSSR}}$ line for one clock cycle. Illegal and Undefined instructions are not under control of the user and will always cause a TRAP if they are passed to the FPU.

Enabling an exception will cause a TRAP whenever the exception occurs and disabling an exception will return a result without a TRAP.

When the FPU TRAPS it sets the Q bit in the status word register. The CPU responds by reading the status word register while applying status (1110) on the status lines. If the CPU sent the FPU ID with an undefined opcode, the T bit in the status word register would also be set by the FPU indicating a TRAP (UND). If the T bit is clear after the TRAP it indicates a TRAP(FPU) and the reason for the TRAP resides in the FSR TRAP TYPE field. A trapped instruction returns no result (also if the destination is an FPU register) and does not affect the CPU PSR.

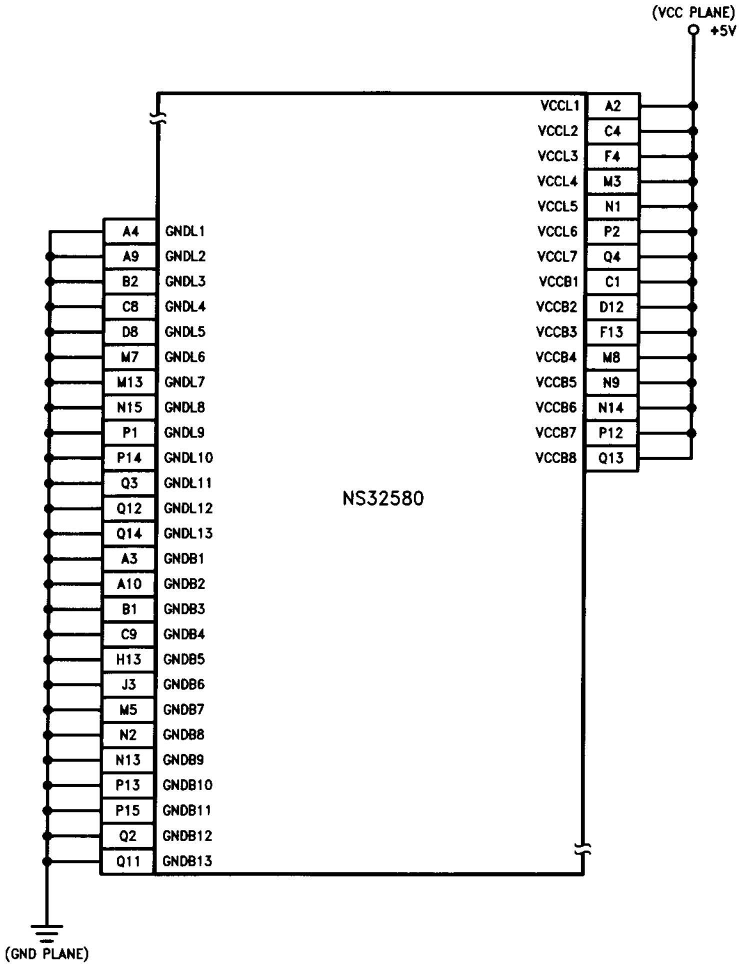
In addition there is a flag bit, for each exception under user control, which will mark the occurrence of the exceptional condition whether or not the exception is enabled or disabled. These bits in the FSR can be used for polling the exception status while TRAPS are disabled.

Floating-point instructions that end with an enabled exception will trap, activating the $\overline{\text{FSSR}}$ signal, but will not update the destination register. In this case, the FPC will ABORT the instruction that ended with the exception to prevent destruction of the data in the destination register. Instructions that ended with a disabled exception update the destination register with the default result.

TABLE 2-3. Exception Enabled/Disabled 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	Infinity or Max NRM Number	OVF = 1
Integer Conversion Ov.	IOE = 0	010	IOE = 1	Max or Min Integer	IOF = 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
Invalid Operation	IVE = 0	101	IVE = 1	Quiet NaN	IVF = 1
Reserved Op. (NaN)	ROE = 0 IVE = 0	101	ROE = 0 IVE = 1	Quiet NaN	ROF = 1 IVF = 1
Reserved Op. (NaN)		000	ROE = 1 IVE = X	Quiet NaN	No Flags
Reserved Op. (DNRM)	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
CMPf (NaN)	ROE = 0 IVE = 0	101	ROE = 0 IVE = 1	L = 1, N = Z = 0 Status Word Register	ROF = 1 IVF = 1
CMPf (NaN)		000	ROE = 1 IVE = X	L = 1, N = Z = 0	No Flags
CMPf (DNRM)	ROE = X IVE = 0	101	ROE = X IVE = 1	N, L, Z Udefined	ROF = 1 IVF = 1

3.0 Functional Description



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.*

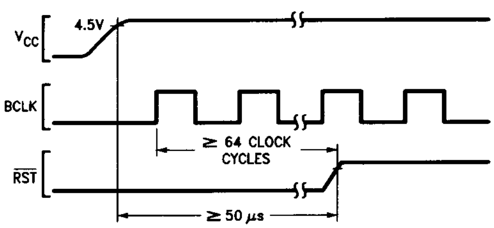
3.2 CLOCKING

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

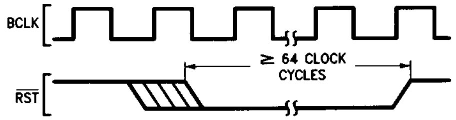
The $\overline{\text{RST}}$ pin serves as a reset for on-chip logic. The FPC may be reset at any time by pulling the $\overline{\text{RST}}$ pin low for at least 64 clock cycles. Upon detecting a reset, the FPC terminates instruction processing, resets its internal logic, clears the FSR to all zeroes,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



TL/EE/9421-9

FIGURE 3-2. Power-On Reset Requirements



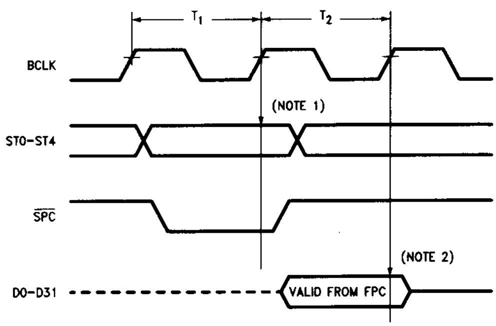
TL/EE/9421-10

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 $\overline{SPC}$ 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 $\overline{SPC}$ 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.

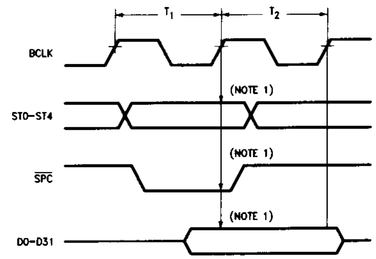


Note 1: FPC samples CPU status here.

Note 2: CPU samples FPC data here.

TL/EE/9421-11

FIGURE 3-4. Slave Processor Read Cycle from FPC



TL/EE/9421-12

Note 1: FPC samples CPU status, control and data here.

FIGURE 3-5. Slave Processor Write Cycle to FPC

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 Less 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.)

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 FPDP. If the instruction is dyadic, two operands will be transferred from memory to the 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 FPDP it is loaded into one of the registers that create the operand FIFO. 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 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 $\overline{SDN}$ will be signaled. If the FPC receives a new ID and Opcode before the CPU finishes reading the result, (can happen if page fault has been detected on a write) the FPC will abort the last instruction and will start the execution of the new instruction. The NS32532 CPU can "reset" the FPC by issuing $\overline{SPC}$ and status h'1E when there was no $\overline{FSSR}$ from FPC. In this case FPC flushes the instructions currently being executed and the contents of the floating-point registers are undefined.

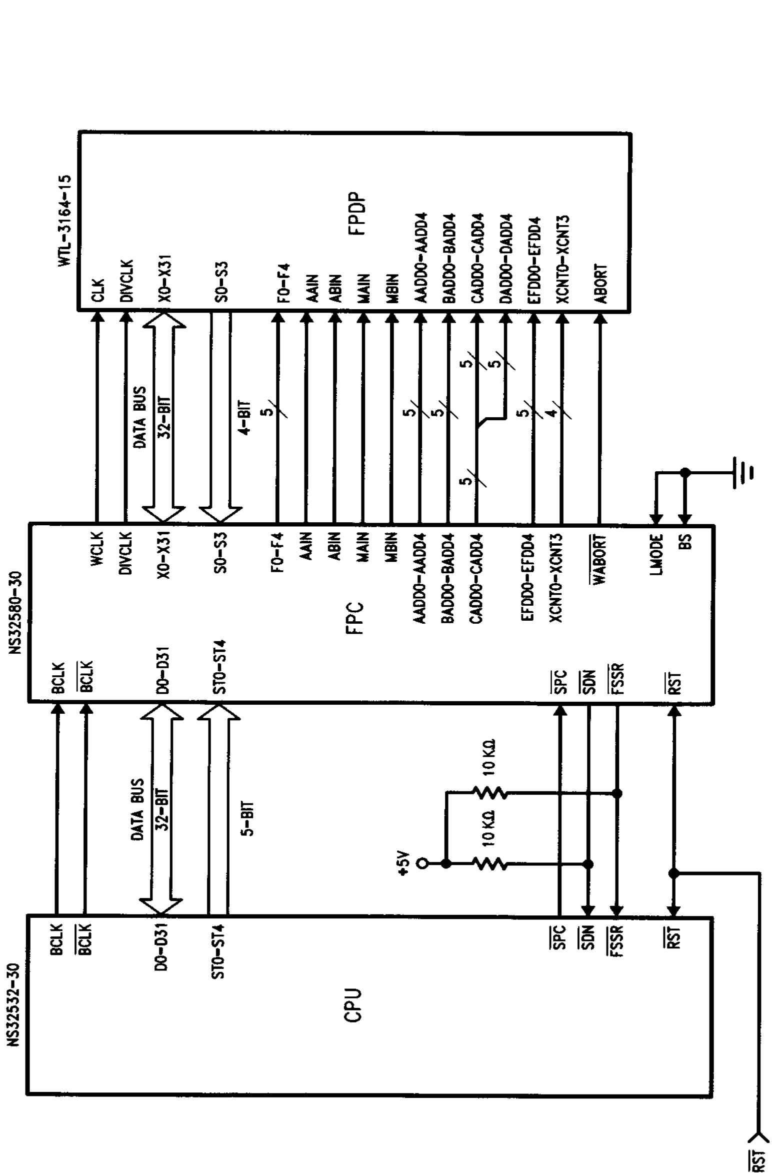


FIGURE 3-6. System Connection Diagram

3.5 INSTRUCTION PROTOCOLS

3.5.1 General Slave Protocol Sequence

The FPC interfaces with the CPU using the Slave-Protocol. The slave protocol is a well defined protocol for instruction and operand transfers between the CPU and the slave co-processors (FPC and Custom Slave). Only the CPU can initiate slave cycle or access memory to fetch operands. The communication between the CPU and the FPC occurs at the beginning of the floating-point instruction, when the CPU transfers the Opcode and possible operands. At the end of the instruction, the FPC signals successful or unsuccessful conclusion of floating-point instruction and the CPU transfers operands from the FPC, if applicable.

The CPU broadcasts the ID and Opcode to all slave processors, one of which will recognize it and from this point the CPU is communicating only with one slave processor.

The CPU puts the slave ID (different ID for each format) on byte 3 (D31–D24), puts the Opcode low on byte 2 (D23–D16) and puts the Opcode high on byte 1 (D15–D8). Byte 0 (D7–D0) is not used.

31	23	15	7 0
ID	OPCODE low	OPCODE high	XXXXXXXX
Byte 3	Byte 2	Byte 1	Byte 0

FIGURE 3-7. ID and Opcode Format

CPU Status Combinations

11101 (h'1D) Transfer Slave Processor Operands—The CPU is transferring an operand to or from a slave processor.

11110 (h'1E) Read Slave Processor Status—The CPU is reading the Status Word Register after the FPC signaled TRAP.

11111 (h'1F) Broadcast Slave ID—The CPU is initiating the execution of a slave processor instruction.

The floating-point unit has three different instruction formats:

Format 9

23 … 16	15 …			… 8	7							0
Gen 1	Gen 2	Op	f	i	0	0	1	1	1	1	1	0

MOVif	—000	MOVLF	—010	ROUND	—100	SFSR	—110
LFSR	—001	MOVFL	—011	TRUNC	—101	FLOOR	—111

Format 11

23 … 16	15 …			… 8	7							0
Gen 1	Gen 2	Op	0	f	1	0	1	1	1	1	1	0

ADDf	—0000	SUBf	—0100	DIVf	—1000	MULf	—1100
MOVf	—0001	NEGf	—0101	Trap(FPU)	—1001	ABSf	—1101
CMPf	—0010	Trap(UND)	—0110	Trap(UND)	—1010	Trap(UND)	—1110
Trap(FPU)	—0011	Trap(UND)	—0111	Trap(UND)	—1011	Trap(UND)	—1111

Format 12

23 … 16	15 …			… 8	7							0
Gen 1	Gen 2	Op	0	f	1	1	1	1	1	1	1	0

SREMf*	—0000	SCALBf*	—0100	Trap(UND)	—1000	Trap(UND)	—1100
SQRTf	—0001	LOGBf*	—0101	Trap(UND)	—1001	Trap(UND)	—1101
POLYf*	—0010	Trap(UND)	—0110	MACf	—1010	Trap(UND)	—1110
DOTf*	—0011	Trap(UND)	—0111	Trap(UND)	—1011	Trap(UND)	—1111

*All the marked instructions are not supported by the NS32580 and will cause Trap(UND).

TABLE 3-1. 32-Bit General Slave Instruction Protocol

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 DONE, TRAP or CMPf
5	ST (11110)	CPU Reads Status Word (If TRAP was signaled or if a CMPf instruction was executed)
6	OP (11101)	CPU Reads Result (if destination is memory and if no TRAP occurred)

TABLE 3-2. Floating-Point Instruction Protocols

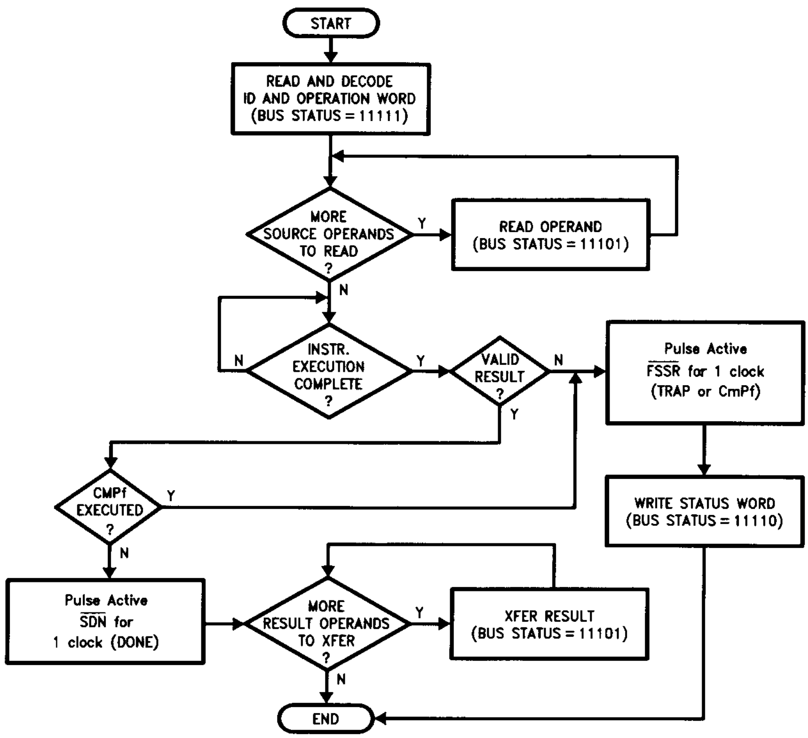
Mnemonic	Operand 1 Class	Operand 2 Class	Operand 1 Issued	Operand 2 Issued	Returned Value	PSR Bits Affected
ADDf	read.f	rmw.f	f	f	f to Op. 2	none
SUBf	read.f	rmw.f	f	f	f to Op. 2	none
MULf	read.f	rmw.f	f	f	f to Op. 2	none
DIVf	read.f	rmw.f	f	f	f to Op. 2	none
MOVf	read.f	write.f	f	N/A	f to Op. 2	none
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.



TL/EE/9421-14

FIGURE 3-8. 32-Bit General Slave Instruction Protocol

3.0 Functional Description (Continued)

3.5.2 Pipelined Slave Protocol Sequence

The NS32532 can communicate with the FPC using the pipelined Slave Protocol. In the pipelined slave protocol, the CPU proceeds to the next floating-point instruction if the destination of the current floating-point instruction is register, without waiting for $\overline{SDN}$ signal. The FPC from the other end can receive new instructions before the end of the previous instruction. The FPC can internally store up to five new instructions, with up to 10 single- or double-precision operands. The CPU saves the PC of the floating-point instructions in the Floating-Point Instruction FIFO (FIF).

If exception occurs, the floating-point instruction can be reexecuted using the PC saved in the FIF (if exception occurs the CPU will flush the FIF and the FPC will flush the instruction and the operand's FIFOs).

The FPC-FPDP can start execution of a new floating-point instruction every two CPU clock cycles.

In the following example three floating-point instructions are being pipelined:

```
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 O(RO).
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 DONE or TRAP for the DIVF instruction. if TRAP occurred, the rest of the instructions will be aborted.
9	ST (h'1E)	CPU Reads Status Word (if TRAP was signaled).
10	—	Slave pulses DONE or TRAP for the ADDF instruction. If TRAP occurred, the rest of the instructions will be aborted.
11	ST (h'1E)	CPU Reads Status Word (if TRAP was signaled).
12	—	Slave pulses DONE or TRAP for the MULF instruction.
13	ST (h'1E)	CPU Reads Status Word (if TRAP was signaled).

3.5.3 Status Word Register

There is a Status Word Register (SWR) that holds the compare results and an exception flag, which indicates TRAP. This register can be read by the CPU by applying status code h'1E (read slave status) on the status line and NSPC as a timing signal. The FPC updates the status word register after compare float instruction or if TRAP has occurred. The content of SWR is valid only after the FPC signaled TRAP.

31	23	15	7 0
00000000	00000000	T0000000	NZ000L0Q

FIGURE 3-9. FPC Status Word Format

Status Bits

N BIT: The N bit is set to "1" if the second operand is less than the first operand. Otherwise, it is set to "0".

Z BIT: The Z bit is set to "1" if the second operand is equal to the first operand. Otherwise, it is set to "0".

L BIT: The L bit is set to "1" if the operands in CMPf operation are "Unordered" (i.e., one of them is NaN). If ROE bit is cleared, the L bit is always cleared by the FPC.

Q BIT: The Q bit is set to "1" if TRAP occurred. The T bit will distinguish between TRAP(UND) and TRAP(FPU).

T BIT: The T bit is set to "1" if the TRAP is TRAP(UND) and "0" if the TRAP is TRAP(FPU). The CPU examines this bit when TRAP is signaled.

3.5.4 Termination of Instruction (Not Including CMPf)

Floating-Point Instructions that ended without exception will signal done by pulsing the $\overline{SDN}$ line for one clock cycle. The CPU will read the result from the FPC if the destination is memory. The CPU can try to read the result immediately after detecting the $\overline{SDN}$ signal. Therefore, the DONE must be signaled after loading the result to the FPC. To read the result the CPU uses the Read from FPC cycle as described in the previous paragraph. Upon detecting an exceptional condition in executing a floating point instruction, the FPC requests a TRAP by pulsing the $\overline{FSSR}$ line for one clock cycle. In addition, it sets the Q bit in the status word register. The CPU responds by reading the status word register while applying status h'1E (transferring status word) on the status lines. A trapped instruction returns no result (also if the destination is FPC register) and does not affect the CPU PSR.

The FPC displays the reason for the TRAP(FPU) in the TRAP TYPE (TT) field of the FSR. If the CPU sends FPC ID with illegal opcode, the FPC generates TRAP(UND) by signaling TRAP and setting the T bit in the status word register. The TT field in the FSR will be set to Illegal Instruction (h'100). POLYf, DOTf, SREMf, SCALBf, LOGBf and all the unused opcodes in formats 11 and 12 will cause a TRAP(UND).

3.5.5 Byte Sex

The FPC supports VME or 32000 bus, depend on the BS pin. In 32000 mode (BS = "0"), the FPC is ready to receive the less significant half of a double-precision operand first and the more significant half afterward. In VME mode (BS = "1"), the FPC is ready to receive the more significant half of a double-precision operand first and the less significant half afterward. The FPC will send the received operands to the correct destination registers inside the FPDP. In VME mode, the user must swap the data bus between the CPU and FPC. Byte 0 in the CPU should be connected to Byte 3 in the FPC, Byte 1 in the CPU should be connected to Byte 2 in the FPC, byte 2 in the CPU should be connected to Byte 1 in the FPC and Byte 3 in the CPU should be connected to Byte 0 in the FPC. The BS line is sampled by the FPC during Reset only.

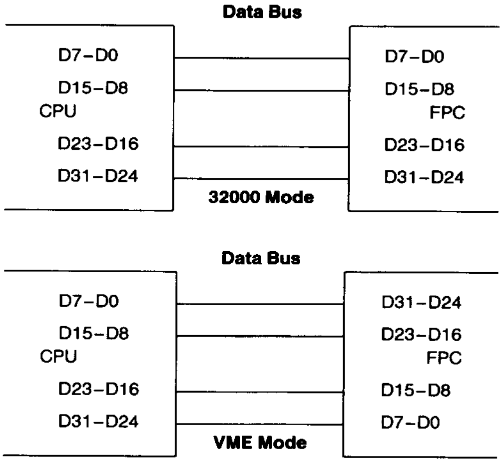


FIGURE 3-10. Byte Sex Connection Diagrams

3.5.6 Floating-Point 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 issued to the Floating-Point Unit 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 FPU Status Word *(Figure 3-7)*.

Any operand indicated as being of type "f" will not cause a transfer if the Register addressing mode is specified, because the Floating-Point Registers are physically on the Floating-Point Unit 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.

Using six data buses allows an 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. (WCLK is half the frequency of BCLK.)

3.6.1 Controlling the FPDP

The FPC controls the FPDP on an instruction by instruction basis and not clock by clock. The instruction's control signals are delayed in the FPDP to match the pipeline stages inside the FPDP.

This allows the specifying of 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 the WCLK (system clock divided by two).

There are three functional fields in the control word:

1. The FUNC bits define 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 operation. Both C ADD and D ADD fields of the FPDP are connected to the D ADD field in the FPC control (C) word.
3. The E/F ADD and XCNT control the Load and Store operations.

FUNC, AAIN, ABIN, MAIN, MBIN Fields

The five-bit FUNC field specifies the arithmetic operation to be executed. The MAIN and AAIN controls the muxes on the A inputs to the MULT and ALU respectively. MBIN and ABIN controls the muxes on the B inputs to the MULT and ALU.

Aain, Main: A = "1", X = "0"
Abin, Mbin: B = "1", Y = "0"

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

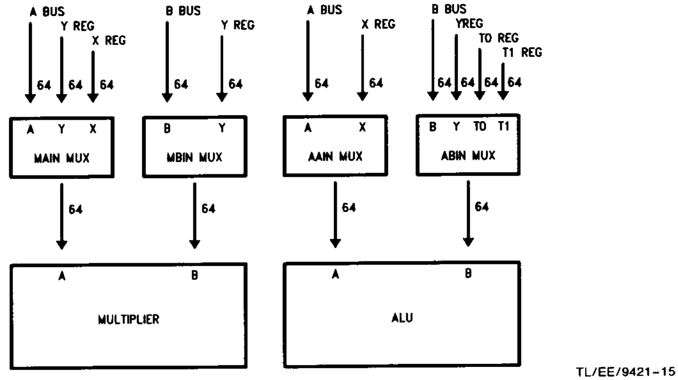


FIGURE 3-12. FPDP Multiplier and ALU Bus Control

XCNT Field

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

Code	Operation	Description
H'0	Nop	
H'1	EREG LS → XPAD	Transfer the Less Significant half of the register specified by EREG to the X-port (Store LS).
H'2	EREG MS → XPAD	Transfer the More 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 Less 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 More 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 FPC is transferred to the FPDP through the XPAD (32-bit I/O 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 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 being 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 latency of two cycles.

When using the "Three cycle latency" the Divide/Sqrt block uses the same clock as the FPDP (can not use the 2× clock). Although the "Three cycle latency" is not optimized for double-precision multiply it may be very useful if the system speed divided by two (WCLK output from FPC) is faster 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" Divide/Sqrt block uses the DCLK3 (same as WCLK), in "Two cycle latency" it uses the DCLK2 (2 × WCLK). 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 speed.

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 other options 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.

Rounding

SR0 Bit-2	SR0 Bit-1	Rounding Mode
0	0	Round toward nearest value, if tie round toward even significant (the default mode after reset)
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 separate paragraph on Latency Modes.)

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 being used in the undelayed single-pump mode for loads and stores. (See I/O OPERATIONS paragraph.)

FpexDelay SR1 Bit-5

"1" Delayed FPEX- Mode.

BypassOn SR1 Bit-6

"1" Enables bypassing of operands between instructions.

SR1 Bit-7

Not used.

3.6.5 IEEE Enables 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 Enables 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.

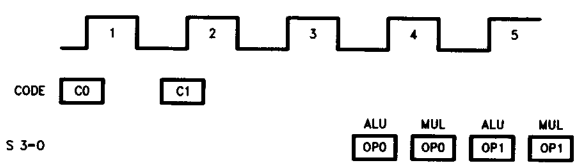
The Dnrm bit is always enabled and detection of Dnrm as operand for operation will cause source exception.

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 (S3–S0)

The status of operation in the FPDP can be obtained by using the FPDP status lines (S3–S1). 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 in 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

3.6.6 FPC-FPDP Clocks

FPC runs off BCLK and $\overline{BCLK}$, which is generated by the CPU. FPDP uses two clock signals, one clock signal for most of the chip and a special clock for the Divide unit. Both FPDP clock signals are supplied by the FPC.

3.6.6.1 FPC Clock

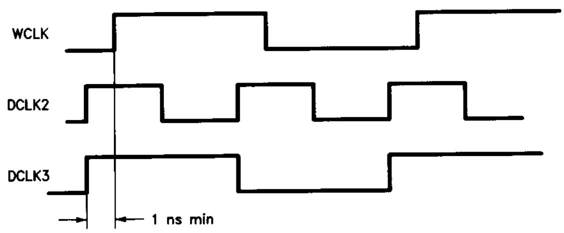
The FPC uses the system clocks (BCLK and $\overline{BCLK}$) generated by the NS32532. All the timing for signals between the CPU and the FPC are referenced to the BCLK. BCLK is a 30 MHz, TTL level clock (for timing characteristics refer to the timing chapter).

3.6.6.2 FPDP Main Clock (WCLK)

The FPDP uses a TTL level clock supplied by the FPC. The FPC generates the WCLK by dividing the BCLK by two. All the FPDP control signal timings are specified relative to the rising edge of the WCLK.

3.6.6.3 Divide/Sqrt Unit Clock (DIVCLK)

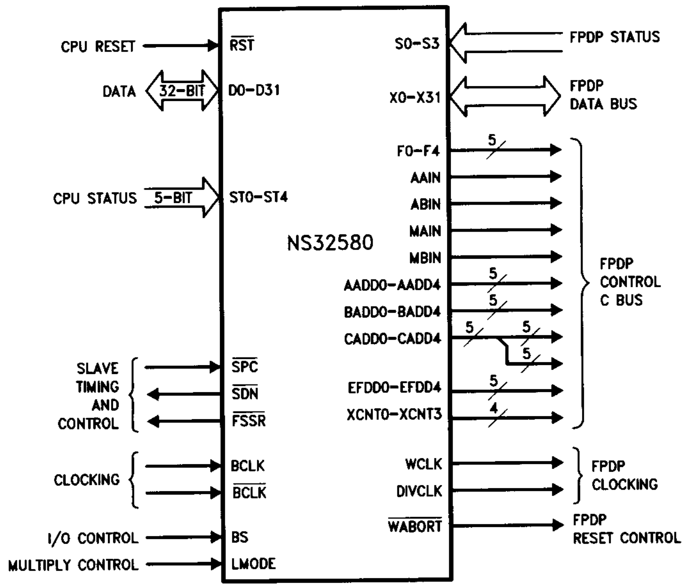
The Divide/Sqrt unit in "Two cycle latency" mode uses a clock signal that is twice the WCLK (DCLK2). If the FPDP is in "three cycle latency", the Divide/Sqrt unit uses a clock signal that has the same frequency as WCLK (DCLK3). The FPC generates the correct DCLK automatically using the LMODE pin.



TL/EE/9421-17

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

4.0 Device Specifications



TL/EE/9421-18

FIGURE 4-1. NS32580 Interface Signals

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 for CPU bus timing; NS32532 system clock.

$\overline{\text{BCLK}}$ — **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 32000 mode. If connected to V_{CC} the FPC is in VME mode. The BS line must be valid during and after Reset. See Section 3.6.5.

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-S3 — **FPDP Status**—Indicates any exceptions or conditions that resulted from operations performed by the WTL3164 floating-point units.

$\overline{\text{SPC}}$ — **Slave Processor Control**—Active low. Data strobe for slave transfers between the CPU and the FPC.

ST0-ST4 — **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-AADD4 — **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-BADD4 — **B Read Port Register Address**—Chooses the inputs to the B bus of the FPDP.

CADD0-CADD4 — **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-CADD4) and the (DADD0-DADD4) 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-EFDD4 — **E and/or F Port Register Address**—Chooses the source and destination for the Load/Store operations of the FPDP.

F0-F4 — **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 slave 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 multiplier of the FPDP.

MBIN — **Multiplier B Input Select**—Controls the B input multiplexers of the multiplier of the FPDP.

$\overline{\text{SDN}}$ — **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-XCNT3 — **Z Port Control**—They are the Load/Store controls for the FPDP.

4.1.4 Input/Output Signals

D0-D31 — **CPU Data Bus**—Data bus between FPC and the CPU.

X0-X31 — **FPDP Data Bus**—Data bus between FPC and the FPDP X port.

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
$\overline{BCLK}$	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
$\overline{RST}$	C11
NC	C12
BADD3	C13
AADD0	C14
BADD4	C15

Desc	Pin
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

Desc	Pin
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
$\overline{SPC}$	L12
$\overline{SDN}$	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
WABORT	M14
S3	M15

Desc	Pin
VCCL5	N1
GNDB8	N2
Reserved	N3
D24	N4
D25	N5
D9	N6
D10	N7
NC	N8
VCCB5	N9
ST2	N10
ST4	N11
$\overline{FSSR}$	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	Q2
GNDL11	Q3
VCCL7	Q4
X20	Q5
X22	Q6
X24	Q7
X27	Q8
X29	Q9
LMODE	Q10
GNDB13	Q11
GNDL12	Q12
VCCB8	Q13
GNDL13	Q14

Note: NC = No Connection

4.0 Device Specifications (Continued)

4.2 ABSOLUTE MAXIMUM RATINGS

If Military/Aerospace specified devices are required, 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	
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.*

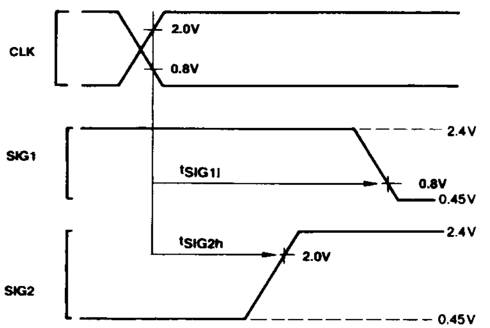
4.3 ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±5%, GND = 0V

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	Logical 1 Input Voltage		2.0		V_{CC} + 0.5	V
V_{IL}	Logical 0 Input Voltage		−0.5		0.8	V
V_{OH}	Logical 1 Output Voltage	I_{OH} = −400 μA	2.4			V
V_{OL}	Logical 0 Output Voltage	I_{OL} = 2 mA			0.4	V
I_I	Input Load Current	$0 \leq V_{IN} \leq V_{CC}$				
$I_{O(OFF)}$	Output Leakage Current (Output Pins in TRI-STATE® Condition)	$0.4 \leq V_{OUT} \leq 2.4V$				
I_{CC}	Active Supply Current	I_{OUT} = 0, T_A = 25°C	100		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 4.2* and *4.3*, unless specifically stated otherwise.

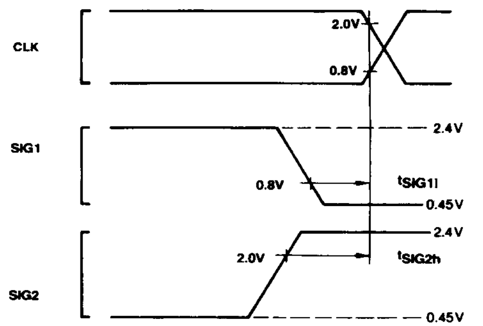


TL/EE/9421-19

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

ABBREVIATIONS

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



TL/EE/9421-20

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

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

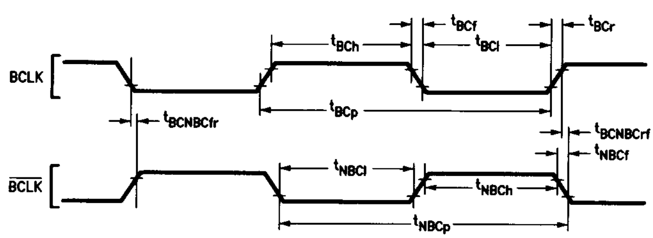
4.4.2.1 Output Signal Propagation Delays Maximum times assume capacitive loading of 100 pF

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_{Dh}	4-8	CPU Data Hold	After R.E., BCLK Next T1/Ti	2		2		2		ns
t_{Dnf}	4-8	CPU Data Not Floating	After R.E., BCLK Next T1/Ti		28		23		19	ns
t_{SDa}	4-10	$\overline{SDN}$ Signal Active	After R.E., BCLK		35		28		22	ns
t_{SDia}	4-10	$\overline{SDN}$ Signal Inactive	After R.E., Next BCLK	2		2		2		ns
t_{SDnf}	4-10	$\overline{SDN}$ Signal Not Floating	After R.E., BCLK		25		20		17	ns
t_{FSSRa}	4-11	$\overline{FSSR}$ Signal Active	After R.E., BCLK		35		28		22	ns
t_{FSSRia}	4-11	$\overline{FSSR}$ Signal Inactive	After R.E., Next BCLK	2		2		2		ns
t_{FSSRnf}	4-11	$\overline{FSSR}$ Signal Not Floating	After R.E., BCLK		25		20		17	ns
t_{Cv}	4-14	C Bus and $\overline{WABORT}$ Valid	After R.E., WCLK		83		63		50	ns
t_{Ch}	4-14	C BUS and $\overline{WABORT}$ Hold Time	After R.E., WCLK	2		2		2		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	2		2		2		ns
t_{D2p}	4-13	DCLK2 Period	From 1.5V R.E., to 1.5V R.E.	50		40		33.3		ns
t_{D2h}	4-13	DCLK2 High Time	From 1.5V R.E., to 1.5V F.E.	22		17		14.5		ns
t_{D2l}	4-13	DCLK2 Low Time	From 1.5V F.E. to 1.5V R.E.	22		17		14.5		ns
t_{D3p}	4-13	DCLK3 Period	From 1.5V R.E., to 1.5V R.E.	100		80		66.6		ns
t_{D3h}	4-13	DCLK3 High Time	From 1.5V R.E., to 1.5V F.E.	45		36		30		ns
t_{D3l}	4-13	DCLK3 Low Time	From 1.5V F.E., to 1.5V R.E.	45		36		30		ns
t_{WCLKp}	4-13	WCLK Period	From 1.5V R.E., to 1.5V R.E.	100		80		66.6		ns
t_{WCLKh}	4-13	WCLK High Time	From 1.5V R.E., to 1.5V F.E.	45		36		30		ns
t_{WCLKl}	4-13	WCLK Low Time	From 1.5V F.E. to 1.5V R.E.	45		36		30		ns
t_{DWd}	4-13	DCLK2/DCLK3 to WCLK Delay	From 1.5V R.E., to 1.5V R.E.	2.5	8	2.5	8	2.5	8	ns
t_{Wr}	4-13	FPDP Clock Rise Time	From 0.4V R.E., to 2.4V R.E.		9		9		9	ns
t_{Wf}	4-13	FPDP Clock Fall Time	From 2.4V F.E. to 0.4V F.E.		2		2		2	ns
t_{BCp}	4-5	BCLK Period	R.E., BCLK to Next R.E., BLCK	50	200	40	200	33.3	200	ns
t_{BCh}	4-5	BCLK High Time	At 2.0V on BCLK (Both Edges)	$0.5\ t_{BCp} - 4$		$0.5\ t_{BCp} - 3.5$		$0.5\ t_{BCp} - 3$		
t_{BCl}	4-5	BCLK Low Time	At 0.8V on BCLK (Both Edges)	$0.5\ t_{BCp} - 4$		$0.5\ t_{BCp} - 3.5$		$0.5\ t_{BCp} - 3$		
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	$\overline{BCLK}$ Period	R.E., $\overline{BCLK}$ to Next R.E., $\overline{BCLK}$	50	200	40	200	33.3	200	ns
t_{NBCh}	4-5	$\overline{BCLK}$ High Time	At 2.0V on $\overline{BCLK}$ (Both Edges)	$0.5\ t_{NBCp} - 5$		$0.5\ t_{NBCp} - 4$		$0.5\ t_{NBCp} - 4$	120	ns

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

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

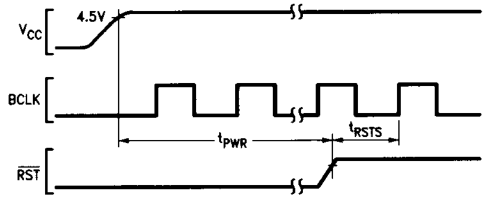
Symbol	*Figure*	Description	Reference/ Conditions	NS32580-20		NS32580-25		NS32580-30		Units
				Min	Max	Min	Max	Min	Max	
t_{NBCl}	*4-5*	$\overline{BCLK}$ Low Time	At 0.8V on $\overline{BCLK}$ (Both Edges)	0.5 t_{NBCp} −5		0.5 t_{NBCp} −4		0.5 t_{NBCp} −4	120	ns
t_{NBCr}	*4-5*	$\overline{BCLK}$ Rise Time	0.8V to 2.0V on R.E., $\overline{BCLK}$		4		3.5		3	ns
t_{NBCf}	*4-5*	$\overline{BCLK}$ Fall Time	2.0V to 0.8V on F.E., $\overline{BCLK}$		4		3.5		3	ns
$t_{BCNBCrf}$	*4-5*	Bus Clock Skew	2.0V on R.E., BCLK to 0.8V on F.E., $\overline{BCLK}$	−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., $\overline{BCLK}$	−2	+2	−2	+2	−1	+1	ns
t_{PWR}	*4-6*	Power Stable to R.E. of $\overline{RST}$	After V_{CC} Reaches 4.5V	50		40		30		µs
t_{RSTs}	*4-6, 4-7*	$\overline{RST}$ Setup Time	Before R.E., BCLK	14		12		11		ns
t_{RSTw}	*4-7*	$\overline{RST}$ 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	24		ns
t_{STh}	*4-8, 4-9*	CPU Status Hold Time	After R.E., BCLK T2	15		12	10	10		ns
t_{SPCs}	*4-8, 4-9*	$\overline{SPC}$ Setup Time	Before R.E., BCLK T2	30		23	20	20		ns
t_{SPCh}	*4-8, 4-9*	$\overline{SPC}$ 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., Next T1 or Ti	−4		−4		−4		ns
t_{SAs}	*4-12*	FPDP ALU Status Setup Time	Before R.E., WCLK	9		9		9		ns
t_{SAh}	*4-12*	FPDP ALU Status Hold Time	After R.E., WCLK	2		2		2		ns
t_{SMs}	*4-12*	FPDP Multiplier Status Setup Time	Before F.E., WCLK	9		9		9		ns
t_{SMh}	*4-12*	FPDP Multiplier Status Hold Time	After F.E., WCLK	2		2		2		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	2		2		2		ns



TL/EE/9421–21

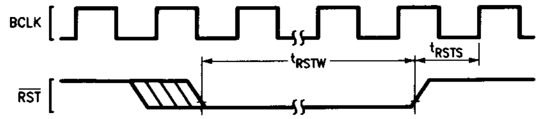
FIGURE 4-5. Clock Waveforms

4.0 Device Specifications (Continued)



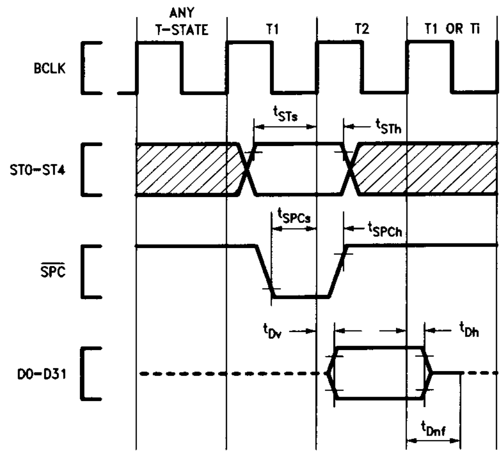
TL/EE/9421-22

FIGURE 4-6. Power-On Reset



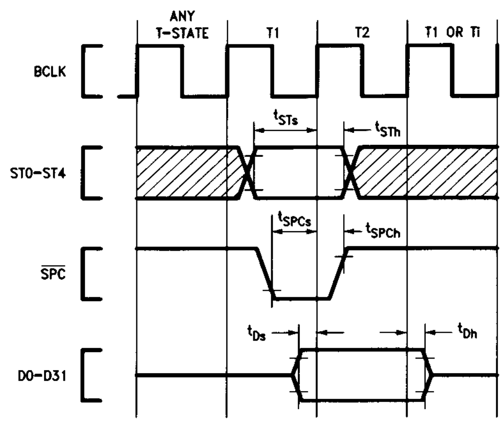
TL/EE/9421-23

FIGURE 4-7. Non-Power-On Reset



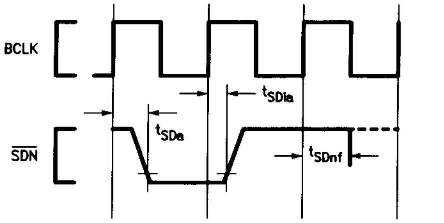
TL/EE/9421-24

FIGURE 4-8. Read Cycle from FPC



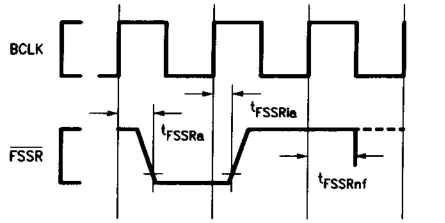
TL/EE/9421-25

FIGURE 4-9. Write Cycle to FPC



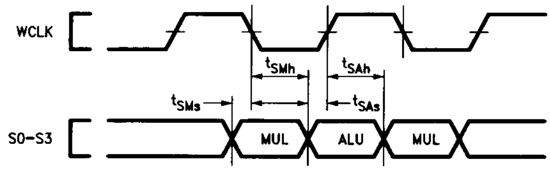
TL/EE/9421-26

FIGURE 4-10. Slave Processor Done Timing



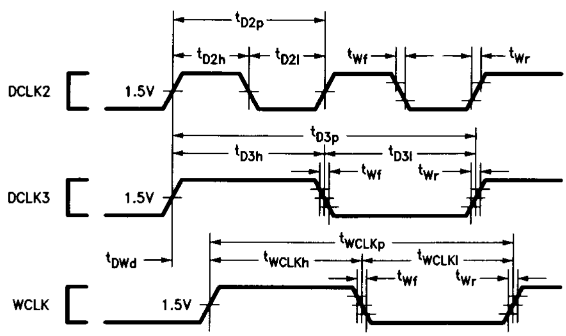
TL/EE/9421-27

FIGURE 4-11. $\overline{FSSR}$ Signal Timing



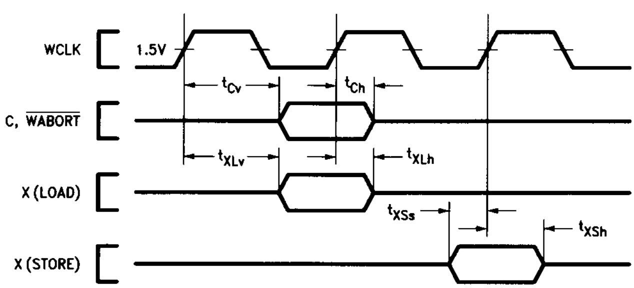
TL/EE/9421-28

FIGURE 4-12. FPDP Status Signal Timing



TL/EE/9421-29

FIGURE 4-13. FPDP Clock Singals Timing



TL/EE/9421-30

FIGURE 4-14. FPDP Output Signals 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

PERFORMANCE ANALYSIS

The execution time is calculated from $\overline{SPC}$ (T1, T2 included) to $\overline{SDN}$ (including the $\overline{SDN}$ 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/I	13	13	2	2	No
SUBf/I	13	13	2	2	No
MULf MULI	13 13	13 15	2 2	2 4	No No
DIVf DIV1	29 43	43 71	Up to 29 Up to 43	Up to 43 Up to 71	No No
MOVf/I	13	13	2	2	No
ABSf/I	13	13	2	2	No
NEGf/I	13	13	2	2	No
CMPf/I	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 MOVil	17 + CPU 13 + CPU	17 + CPU 13 + CPU	— —	— —	Yes Yes
LFSR	13	13	—	—	Yes
SFSR	13 + CPU	13 + CPU	—	—	Yes
MACf MACI	15 15	15 17	6 6	6 8	No No
SQRTf SQRTI	41 69	65 123	Up to 41 Up to 69	Up to 65 Up to 123	No No

Appendix A (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.

Physical Dimensions inches (millimeters)

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

National Semiconductor GmbH
Westendstrasse 193-195
D-8000 Munchen 21
West Germany
Tel: (089) 5 70 95 01
Telex: 522772

NS Japan Ltd.
4-403 Ikebukuro,
Toshima-ku,
Tokyo 171, Japan
Tel: (03) 988-2131
FAX: 011-81-3-988-1700

National Semiconductor Hong Kong Ltd.
Southeast Asia Marketing
Austin Tower, 4th Floor
22-26A Austin Avenue
Tsimshatsui, Kowloon, H.K.
Tel: 3-7231290, 3-7243645
Cable: NSSEAMKTG
Telex: 52996 NSSEA HX

National Semicondutores Do Brasil Ltda.
Av. Brig. Faria Lima, 830
8 Andar
01452 Sao Paulo, SP. Brasil
Tel: (55/11) 212-5066
Telex: 391-1131931 NSBR BR

National Semiconductor (Australia) PTY, Ltd.
21/3 High Street
Bayswater, Victoria 3153
Australia
Tel: (03) 729-6333
Telex: AA32096