MAY 2 4 '91

National Semiconductor

PRELIMINARY
April 1991

NS32181-15/NS32181-20/NS32181-25 Floating-Point Unit

NS32181-15/NS32181-20/NS32181-25 Floating-Point Unit

General Description

The NS32181 is a CMOS floating-point slave processor that is fully software compatible with the NS32081 FPU. The NS32181 FPU functions with Series 32000® and Series 32000/EP CPUs in a tightly coupled slave configuration. The performance of the NS32181 has been increased over the NS32081 by architecture improvements, hardware enhancements, and higher clock frequencies. Key improvements include an early done algorithm to increase CPU/FPU parallelism, an expanded register set, an automatic power down feature, expanded math hardware, and additional instructions.

The NS32181 FPU contains eight 64-bit data registers and a Floating-Point Status Register (FSR). The FPU executes 20 instructions, and operates on both single and double-precision operands. Three separate processors in the NS32181 manipulate the mantissa, sign, and exponent.

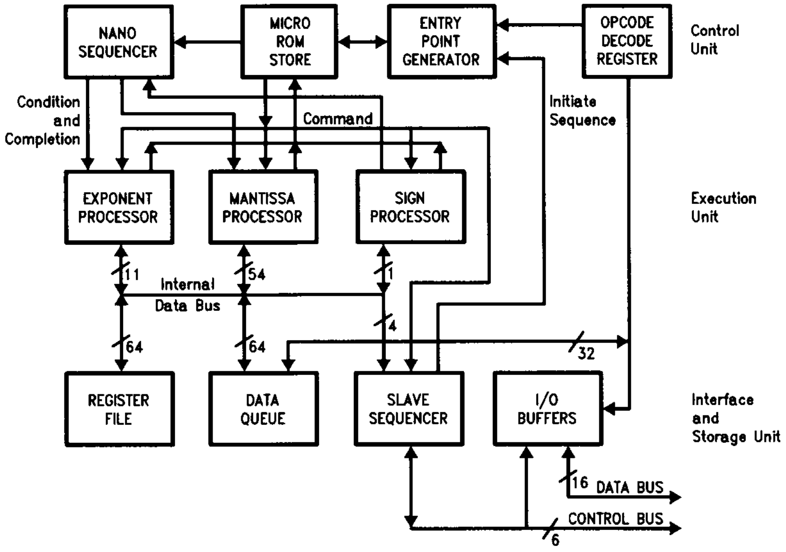
The CPU and NS32181 FPU form a tightly coupled computer cluster, which appears to the user as a single processing unit. The CPU and FPU communication is handled automatically, and is user transparent.

The FPU is fabricated with National's advanced double-metal CMOS process.

Features

- Compatible with NS32CG16, NS32CG160, NS32FX16 and previous Series 32000 CPUs
- 16-bit slave protocol
- Compatible with IEEE Standard 754-1985 for binary floating point arithmetic
- Early done algorithm
- Single (32-bit) and double (64-bit) precision operations
- Eight on-chip (64-bit) data registers
- Automatic power down mode
- Full upward compatibility with existing 32000 software
- High speed double-metal CMOS design
- 68-Pin PLCC package

FPU Block Diagram



TL/EE/10828-1

FIGURE 1-1

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

RRD-B20M51/Printed in U. S. A.

Table of Contents

List of Illustrations

List of Tables

1.0 Product Introduction

The NS32181 Floating-Point Unit (FPU) provides high speed floating-point operations for the Series 32000/EP family, and is fabricated using National high-speed CMOS technology. It operates as a slave processor for transparent expansion of the Series 32000 basic instruction set. The FPU can also be used with other microprocessors as a peripheral device by using additional TTL and CMOS interface logic. The NS32181 is compatible with the IEEE Floating-Point Formats.

1.1 IEEE STD 754 FEATURES SUPPORTED BY THE NS32181

a) Basic floating-point number formats

b) Add, subtract, multiply, divide 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)

1.2 OPERAND FORMATS

The NS32181 FPU 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 $1.0 \le x < 2.0$.

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 a reserved operand (Section 1.2.3). 00 . . . 00 represents the number zero if the F field is also all zeroes, otherwise it signals a reserved operand.

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.

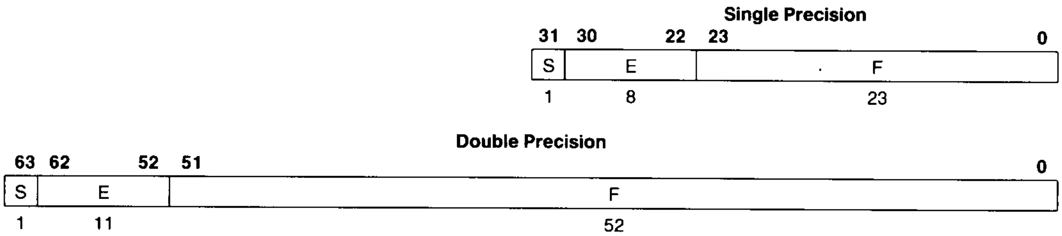


FIGURE 1.2. Floating-Point Operand Formats

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	$-(2^{-125})$ $= -1.17549436 \times 10^{-38}$	$-(2^{-1022})$ $= -2.2250738585072014 \times 10^{-308}$
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.

1.2.2 Zero

There are two representations 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

The IEEE Standard for Binary Floating-Point Arithmetic provides for certain exceptional forms of floating-point operands. The NS32181 FPU treats these forms as reserved operands. The reserved operands are:

- Positive and negative infinity
- Not-a-Number (NaN) values
- Denormalized numbers

Both Infinity and NaN values have all ones in their E fields. Denormalized numbers have all zeroes in their E fields and non-zero values in their F fields.

The NS32181 FPU 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). The FPU does not generate reserved operands as results.

1.2.4 Integers

In addition to performing floating-point arithmetic, the NS32181 FPU performs conversions between integer and floating-point data types. Integers are accepted or generated by the FPU as two's complement values of byte (8-bits), word (16-bits) or double word (32-bits) length.

See *Figure 1-3* for the Integer Format and Table 1-4 for the Integer Fields.

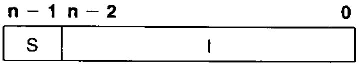


FIGURE 1-3. Integer Format

TABLE 1-4. Integer Fields

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

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

1.2.5 Memory Representations

The NS32181 FPU does not directly access memory. However, it is cooperatively involved in the execution of a set of two-address instructions with its Series 32000/EP Family CPU. The CPU determines the representation of operands in memory.

In the Series 32000/EP 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 that are implemented on the NS32181 Floating-Point Unit (FPU).

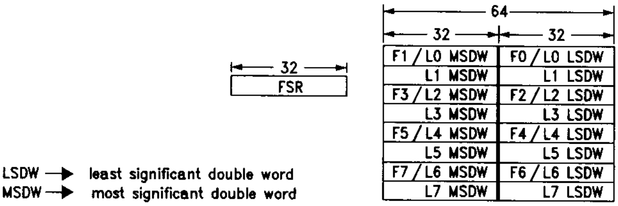
2.1.1 Floating-Point Registers

There are eight registers (L0–L7) on the NS32181 FPU 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.

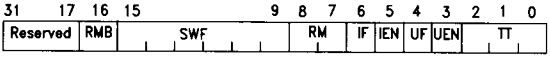
2.1.2 Floating-Point Status Register (FSR)

The Floating-Point Status Register (FSR) selects operating modes and records any exceptional conditions encountered during execution of a floating-point operation. *Figure 2-2* shows the format of the FSR.



TL/EE/10828-2

FIGURE 2-1. Register Set



TL/EE/10828-3

FIGURE 2-2. Floating-Point Status Register

2.1.2.1 FSR Mode Control Fields

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

Rounding Mode (RM): Bits 7 and 8. This field selects the rounding method. Floating-point results are rounded whenever they cannot be exactly represented. 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 exact 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 FPU requests a trap whenever a result is too small in absolute value to be represented as a normalized number. If it is not set, any underflow condition returns a result of exactly zero.

Inexact Result Trap Enable (IEN): Bit 5. It this bit is set, the FPU requests a trap whenever the result of an operation cannot be represented exactly in the operand format of the destination. If it 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 meanings of the FSR status bits are given below:

Trap Type (TT): Bits 0–2. This 3-bit field records any exceptional condition detected by a floating-point instruction. The TT field is loaded with zero whenever any floating-point instruction except LFSR or SFSR completes without encountering an exceptional condition. It is also set to zero by a hardware reset or by writing zero into it with the Load FSR (LFSR) instruction. Underflow and Inexact Result are always reported in the TT field, regardless of the settings of the UEN and IEN bits.

000 No exceptional condition occurred.

001 Underflow. A non-zero floating-point result is too small in magnitude to be represented as a normalized floating-point number in the format of the destination operand. This condition is always reported in the TT field and UF bit, but causes a trap only if the UEN bit is set. If the UEN bit is not set, a result of Positive Zero is produced, and no trap occurs.

010 Overflow. A result (either floating-point or integer) of a floating-point instruction is too great in magnitude to be held in the format of the destination operand. Note that rounding, as well as calculations, can cause this condition.

011 Divide by zero. An attempt has been made to divide a non-zero floating-point number by zero. Dividing zero by zero is considered an Invalid Operation instead (below).

100 Illegal Instruction. Any instruction forms not included in the NS32181 Instruction Set are detected by the FPU as being illegal.

101 Invalid Operation. One of the floating-point operands of a floating-point instruction is a Reserved operand, or an attempt has been made to divide zero by zero using the DIVf instruction.

110 Inexact Result. The result (either floating-point or integer) of a floating-point instruction cannot be represented exactly in the format of the destination operand, and a rounding step must alter it to fit. This condition is always reported in the TT field and IF bit unless any other exceptional condition has occurred in the same instruction. In this case, the TT field always contains the code for the other exception and the IF bit is not altered. A trap is caused by this condition only if the IEN bit is set; otherwise the result is rounded and delivered, and no trap occurs.

111 (Reserved for future use.)

Underflow Flag (UF): Bit 4. This bit is set by the FPU 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 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 FPU whenever the result of an operation must be rounded to fit within the destination format. The IF bit is set only if no other error has occurred. 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 FPU 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 FPU registers should be saved.

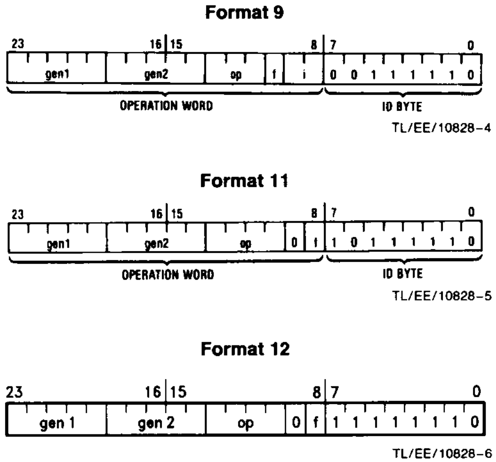
2.1.2.3 FSR Software Field (SWF)

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

2.2 INSTRUCTION SET

This section describes the floating-point instructions executed by the FPU in conjunction with the CPU. These instructions form a subset of the Series 32000 instruction set and take 9, 11, and 12 encoding formats. 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 2-3*.


Format 9

TL/EE/10828-4

Format 11

TL/EE/10828-5

Format 12

TL/EE/10828-6

FIGURE 2-3. Floating-Point Instruction Formats

The Format column indicates which of the three formats in *Figure 2-3* 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. Refer to Table 2-1 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.

Note: POLY and DOT use the additional third implied operand.
POLY and DOT put their results to LO/FO register and not the gen2.

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.

	Format	Op	Instruction		Description
	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	0100	SCALBf	gen1, gen2	Move gen2*2^{gen1} to gen2, for integral values of gen1 without computing 2^{gen1}.
(N)	12	0101	LOGBf	gen1, gen2	Move the unbiased exponent of gen1 to gen2.
(N)	12	0011	DOTf	gen1, gen2	Move (gen1*gen2) + L0 to L0.(*)
(N)	12	0010	POLYf	gen1, gen2	Move (L0*gen1) + gen2 to L0.(*)

Notes:

(N): Indicates NEW instruction.

(*): The third implied operand used by theses instructions can be either F0 or L0 depending on whether "floating" or "long" data type is specified in the opcode.

Comparison

The Compare instruction compares two floating-point values, sending the result to the CPU PSR Z and N bits for use as condition codes. See *Figure 3-11*. The Z bit is set if the gen1 and gen2 operands are equal; it is cleared othewise. The N bit is set if the gen1 operand is greater than the gen2 operand; it is cleared otherwise. The CPU PSR L bit is unconditionally cleared. Positive and negative zero are considered equal.

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

Floating-Point Status Register Access

The following instructions load and store the FSR as a 32-bit integer.

Format	Op	Instruction		Description
9	001	LFSR	gen1	Load FSR
9	110	SFSR	gen2	Store FSR

Note: All instructions support all of the Series 32000/EP family data formats and addressing modes.

Rounding

The FPU supports all IEEE rounding options: Round toward nearest value or even significant if a tie. Round toward zero, Round toward positive infinity and Round toward negative infinity.

2.3 EXCEPTIONS

The FPU supports five types of exception: Invalid operation, Division by zero, Overflow, Underflow and Inexact Result. When an exception occurs, the FPU may or may not generate a trap depending upon the bit setting in the FSR Register. The user can disable the Inexact Result and the Underflow traps. If an undefined Floating-Point instruction is passed to the FPU an Illegal Instruction trap will occur. The user can't disable trap on Illegal Instruction.

Upon detecting an exceptional condition in executing a floating-point instruction, the FPU requests a TRAP by pulsing the $\overline{\text{SPC}}$ line for one clock cycle.

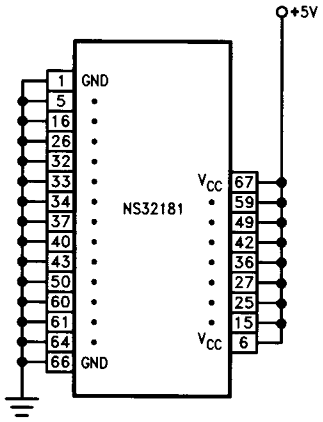
In addition, the FPU sets the Q bit in the status word register. The CPU responds by reading the status word register (refer to Section 3.6.1 for its format) while applying status code 1110 on the status lines. A trapped instruction returns no result (even if the destination is FPU register) and does not affect the CPU PSR. The FPU records exceptional cause in the trap type (TT) field of the FSR. If an illegal opcode is detected, the FPU sets the TS bit in the slave processor status word register, indicating a trap (UND).

3.0 Functional Description

3.1 POWER AND GROUNDING

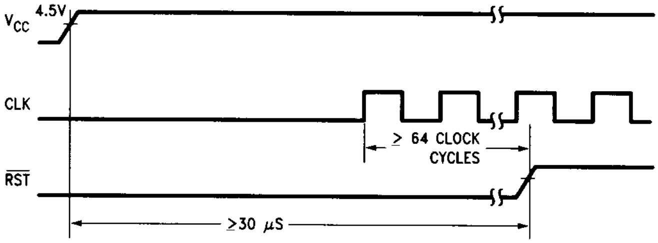
The NS32181 requires a single 5V power supply, applied on the V_{CC} pins. These pins should be connected together by a power (V_{CC}) plane on the printed circuit board. See *Figure 3-1.*

The grounding connections are made on the GND pins. These pins should be connected together by a ground (GND) plane on the printed circuit board. See *Figure 3-1.*



TL/EE/10828-7

FIGURE 3-1. Recommended Supply Connections



TL/EE/10828-8

FIGURE 3-2. Power-On Reset Requirements

3.2 AUTOMATIC POWER DOWN MODE

The NS32181 supports a power down mode in which the device consumes only 20% of its orginal power at 30 MHz. The NS32181 enters the power down mode (internal clocks are stopped with phase two high) if it does not receive an $\overline{\text{SPC}}$ pulse from the CPU within 256 clocks.

The FPU exits the power down mode and returns to normal operation after it receives an $\overline{\text{SPC}}$ from the CPU. There is no extra delay caused by the FPU being in the power down mode.

3.3 CLOCKING

The NS32181 FPU requires a single-phase TTL clock input on its CLK pin. Different Clock sources can be used to provide the CLK signal depending on the application. When the FPU is connected to a Series 32000/EP CPU, the clock signal is provided by the CTTL signal from the CPU.

3.4 RESETTING

The $\overline{\text{RST}}$ pin serves as a reset for on-chip logic. The FPU 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 FPU terminates instruction processing, resets its internal logic, and clears the FSR to all zeroes.

On application of power, $\overline{\text{RST}}$ must be held low for at least 30 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.*

3.5 BUS OPERATION

Instructions and operands are passed to the NS32181 FPU with slave processor bus cycles. Each bus cycle transfers either one byte (8 bits) or one word (16 bits) to or from the FPU. During all bus cycles, the $\overline{\text{SPC}}$ line is driven by the CPU as an active low data strobe, and the FPU monitors pins ST0 and ST1 to keep track of the sequence (protocol) established for the instruction being executed. This is necessary in a virtual memory environment, allowing the FPU to retry an aborted instruction.

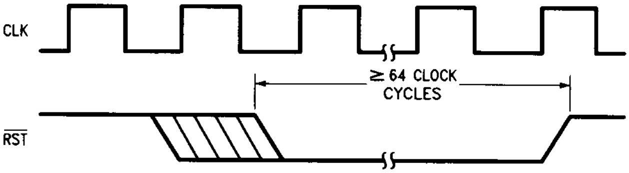
3.5.1 Bus Cycles

A bus cycle is initiated by the CPU, which asserts the proper status on (ST0–ST3) and pulses $\overline{\text{SPC}}$ low. The status lines are sampled by the FPU on the leading (falling) edge of the $\overline{\text{SPC}}$ pulse. If the transfer is from the FPU (a slave processor read cycle), the FPU asserts data on the data bus for the duration of the $\overline{\text{SPC}}$ pulse. If the transfer is to the FPU (a slave processor write cycle), the FPU latches data from the data bus on the trailing (rising) edge of the $\overline{\text{SPC}}$ pulse. *Figures 3-6* and *3-7* illustrate these sequences.

The direction of the transfer and the role of the bidirectional $\overline{\text{SPC}}$ line are determined by the instruction protocol being performed. $\overline{\text{SPC}}$ is always driven by the CPU during slave processor bus cycles. Protocol sequences for each instruction are given in Section 3.6.

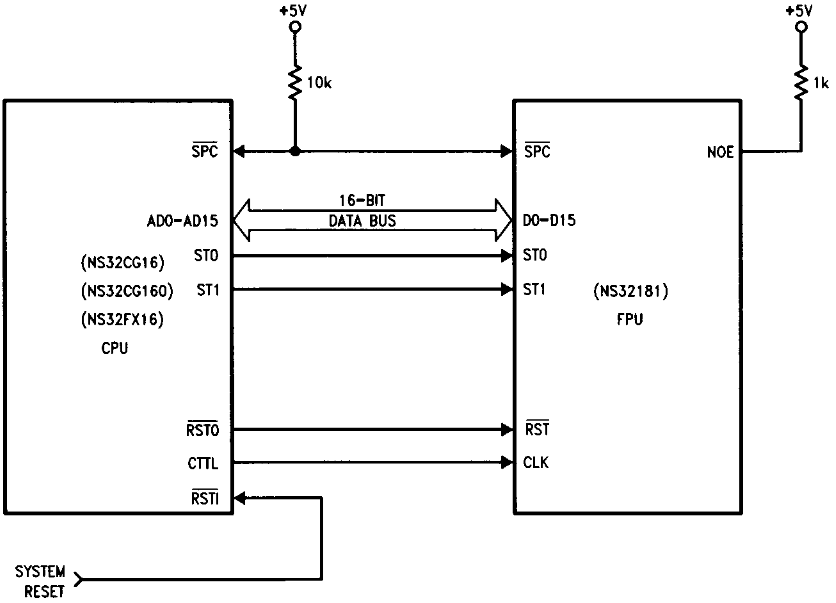
3.5.2 Operand Transfer Sequences

An operand is transferred in one or more bus cycles. A 1-byte operand is transferred on the least significant byte of the data bus (D0–D7). A 2-byte operand is transferred on the entire bus. A 4-byte or 8-byte operand is transferred in consecutive bus cycles, least significant word first.



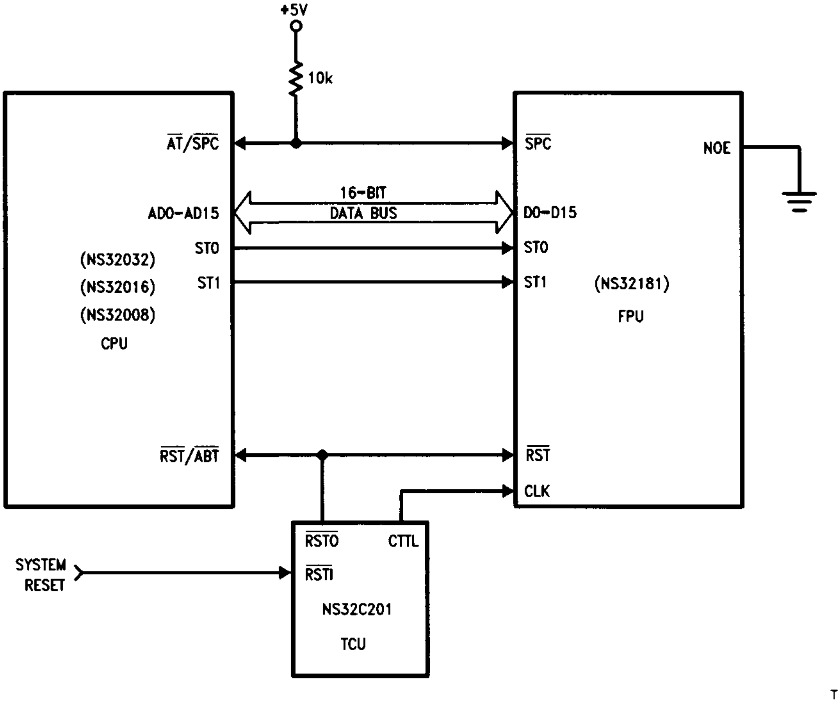
TL/EE/10828-9

FIGURE 3-3. General Reset Timing



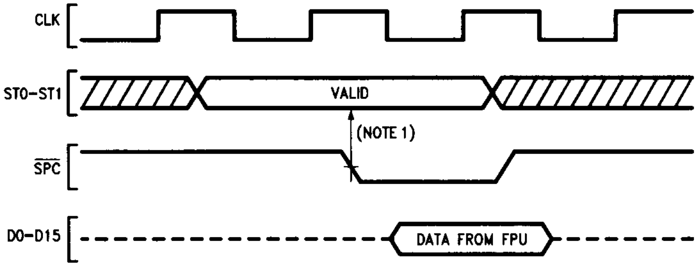
TL/EE/10828–10

FIGURE 3-4. System Connection Diagram with the NS32CG16, NS32CG160 or NS32FX16 CPU



TL/EE/10828–11

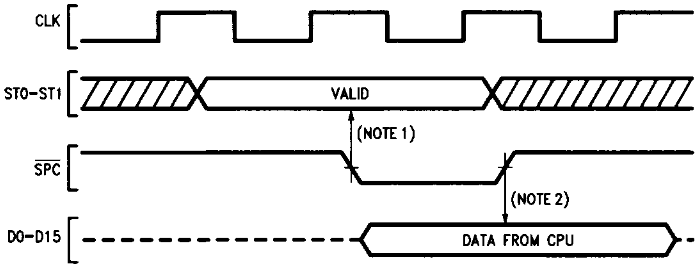
FIGURE 3-5. System Connection Diagram with the NS32008, NS32016 or NS32032 CPU



TL/EE/10828–12

Note 1: FPU samples CPU status here.

FIGURE 3-6. Slave Processor Read Cycle



TL/EE/10828–13

Note 1: FPU samples CPU status here.

Note 2: FPU samples data bus here.

FIGURE 3-7. Slave Processor Write Cycle

3.6 INSTRUCTION PROTOCOLS

3.6.1 General Protocol Sequences

Slave Processor instructions have a three-byte Basic Instruction field, consisting of an ID byte followed by an Operation Word. See Section 2.2 for FPU instruction encodings. The ID Byte has three functions:

1. It identifies the instruction to the CPU as being a Slave Processor instruction.
2. It specifies which Slave Processor will execute it.
3. It determines the format of the following Operation Word of the instruction.

Upon receiving a slave processor instruction, the CPU initiates a sequence outlined in Table 3-1.

TABLE 3-1. General Slave Instruction Protocol

Step	Status	Action
1	ID (1111)	CPU sends ID Byte
2	OP (1101)	CPU sends Operation Word
3	OP (1101)	CPU sends required operands (if any)
4	—	Slaves starts execution (CPU prefetches)
5	—	Slave pulses $\overline{SPC}$ low
6	ST (1110)	CPU Reads Status Word
7	OP (1101)	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 Type and Destination	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
SCALBf	read.f	rmw.f	f	f	f to Op.2	none
LOGBf	read.f	write.f	f	N/A	f to Op.2	none
DOTf	read.f	read.f	f	f	*f to F0/L0	none
POLYf	read.f	read.f	f	f	*f to F0/L0	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.

*The "returned value" can go to either F0 or L0 depending on the "f" bit in the opcode, i.e., whether "floating" or "long" data type is used.

While applying Status Code 1111 (Broadcast ID), the CPU transfers the ID Byte on the least significant half of the Data Bus (D0–D7). All Slave Processors input this byte and decode it. The Slave Processor selected by the ID Byte is activated, and from this point the CPU is communicating only with it. If any other slave protocol was in progress (e.g., an aborted Slave instruction), this transfer cancels it.

The CPU next sends the Operation Word while applying Status Code 1101 (Transfer Slave Operand). Upon receiving it, the FPU decodes it, and at this point both the CPU and the FPU are aware of the number of operands to be transferred and their sizes. The Operation Word is swapped on the Data Bus; that is, bits 0–7 appear on pins D8–D15, and bits 8–15 appear on pins D0–D7.

Using the Addressing Mode fields within the Operation Word, the CPU starts fetching operands and issuing them to the FPU. To do so, it references any Addressing Mode extensions appended to the FPU instruction. Since the CPU is solely responsible for memory accesses, these extensions are not sent to the Slave Processor. The Status Code applied is 1101 (Transfer Slave Processor Operand).

After the CPU has issued the last operand, the FPU starts the actual execution of the instruction. Upon completion, it will signal the CPU by pulsing $\overline{SPC}$ low. To allow for this, the CPU releases the $\overline{SPC}$ signal, causing it to float. $\overline{SPC}$ must be held high by an external pull-up resistor.

Upon receiving the pulse on $\overline{SPC}$, the CPU uses $\overline{SPC}$ to read a Status Word from the FPU, applying Status Code 1110. This word has the format shown in *Figure 3-8*. If the Q bit is set, this indicates that an error has been detected by the FPU. The CPU will not continue the protocol, but will immediately trap through the Slave vector in the Interrupt Table. If the instruction being performed is CMPf (Section 2.2) and the Q bit is not set, the CPU loads Processor Status Register (PSR) bits N, Z and L from the corresponding bits in the Status Word. The FPU always sets the L bit to zero.

The last step will be for the CPU to read the result, provided there are no errors and the result's destination is either in memory or in a CPU register. Here again the CPU uses $\overline{SPC}$ to read to result from the FPU and transfer it to its destination. These Read cycles from the FPU are performed by the CPU while applying Status Code 1101 (Transfer Slave Operand).

15							8	7							0
TS	0	0	0	0	0	0	0	N	Z	0	0	0	L	0	Q

Bit **Description**

(0) Q: Set to "1" if an FPU TRAP (error) occurred. Cleared to "0" by a valid CMPf.

(2) L: Cleared to "0" by the FPU.

(6) Z: Set to "1" if the second operand is equal to the first operand. Otherwise it is cleared to "0".

(7) N: Set to "1" if the second operand is less than the first operand. Otherwise it is cleared to "0".

(15) TS: Set to "1" if the TRAP is (UND) and cleared to "0" if the TRAP is (FPU). Section 2.3.

FIGURE 3-8. FPU Status Word Format

3.6.2 Early Done Algorithm

The NS32181 has the ability to modify the General Slave protocol sequences and to boost the performance of the FPU by 20% to 40%. This is called the Early Done Algorithm.

Early Done is defined by the fact that the destination of an instruction is an FPU register and that the instruction and range of operands cannot generate a TRAP. When these conditions are met the FPU will send an $\overline{SPC}$ pulse after receiving all of the operands from the CPU and before executing the instruction. Hence this becomes an early done as compared to the General Slave Protocol.

Since the CPU always reads the slave status word, the FPU will force all zeroes to be read. The CPU can then send the next instruction to the FPU and save the general protocol overhead. The FPU will start the new instruction immediately after finishing the previous instruction.

SFSR, CMPF and CMPL do not generate an Early Done.

3.6.3 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

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 3-8)*.

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.

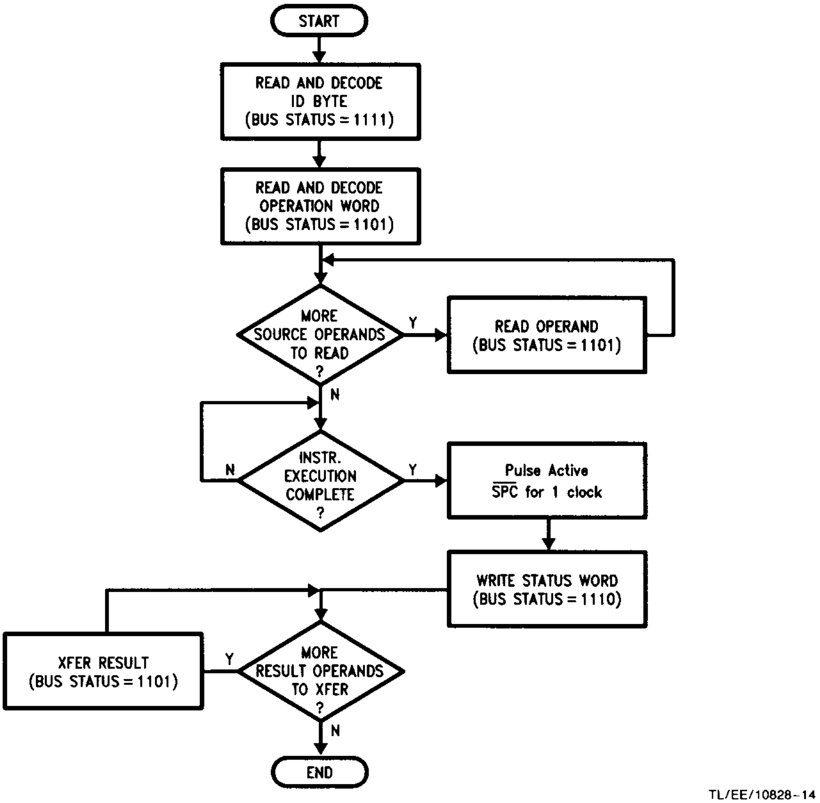

START
READ AND DECODE ID BYTE (BUS STATUS = 1111)
READ AND DECODE OPERATION WORD (BUS STATUS = 1101)
MORE SOURCE OPERANDS TO READ ?
Y
READ OPERAND (BUS STATUS = 1101)
N
INSTR. EXECUTION COMPLETE ?
N
Y
Pulse Active $\overline{\text{SPC}}$ for 1 clock
WRITE STATUS WORD (BUS STATUS = 1110)
XFER RESULT (BUS STATUS = 1101)
Y
MORE RESULT OPERANDS TO XFER ?
N
END
TL/EE/10828-14

FIGURE 3-9. General Slave Instruction Protocol: FPU Actions

4.0 Device Specifications

4.1 PIN DESCRIPTIONS

4.1.1 Supplies

The following is a brief description of all NS32181 pins.

V_{CC} **Power:** + 5V positive supply.

GND **Ground:** Ground reference for both on-chip logic and output drivers.

CLK **Clock:** TTL-level clock signals.

ST0–ST1 **Status:** Two least-significant bits of the CPU status. ST0 is the rightmost bit. The upper bits of the CPU status are not monitored by the FPU. Encodings are:

00— Reserved

01— Transferring Operation Word or Operand

10— Reading Status Word

11— Broadcasting Slave ID

$\overline{\text{RST}}$ **Reset:** A low level on this pin initiates a reset operation. Section 3.4.

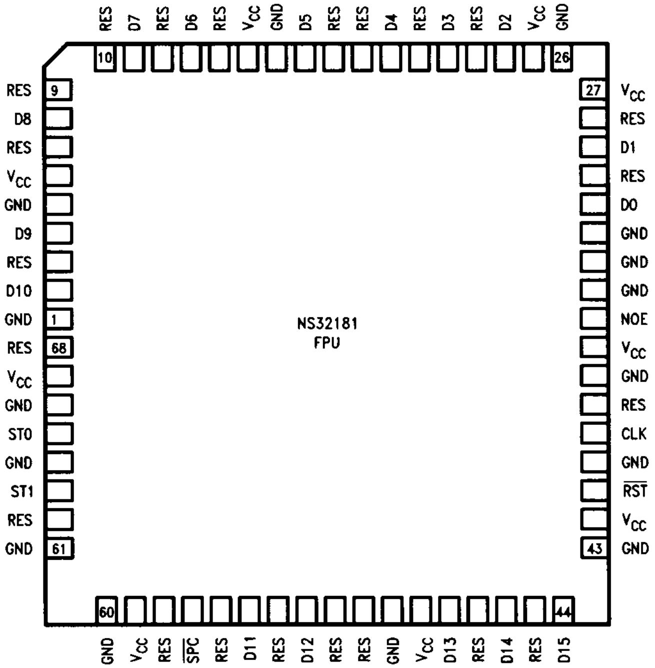
NOE **New Opcode Enable:** Active high. This signal enables the new opcodes available in the NS32181.

4.1.3 Input/Output Signals

$\overline{\text{SPC}}$ **Slave Processor Control:** Active low. Driven by the CPU as the data strobe for bus transfers to and from the FPU. Driven by the FPU to signal completion of an operation, Section 3.6.1. Must be held high with an external pull-up resistor while floating.

D0–D15 **Data Bus:** 16-bit bus for data transfer. D0 is the least significant bit. Section 3.5

Connection Diagram



TL/EE/10828-15

Bottom View

Order Number NS32181V-15, NS32181V-20, NS32181V-25
See NS Package Number V68A

FIGURE 4-1. 68-Pin Plastic Chip Carrier Package

Note: All pins marked RES (Reserved) must be left open.

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.

Maximum Case Temperature	95°C
Storage Temperature	−65°C to +150°C
All Input or Output Voltages with Respect to GND	−0.5V to +7.0V
ESD Rating	2000V (in human body model)

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

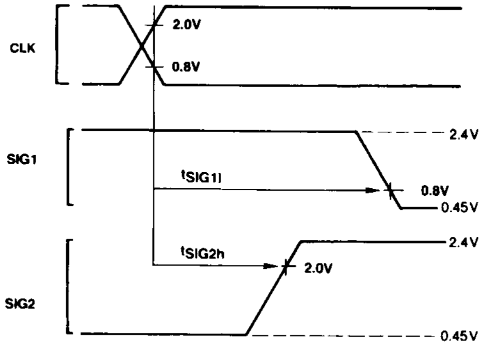
Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage*		2.0		V_{CC} + 0.5	V
V_{IL}	Low Level Input Voltage*		−0.5		0.8	V
V_{OH}	High Level Output Voltage	I_{OH} = −400 μA	2.4			V
V_{OL}	Low Level Output Voltage	I_{OL} = 4 mA			0.4	V
I_I	Input Load Current*	$0 \leq V_{IN} \leq V_{CC}$	−10.0		10.0	μA
V_{IH}	High Level Input Voltage for NOE		3.5		V_{CC} + 0.5	V
V_{IL}	Low Level Input Voltage for NOE		−0.5		1.5	V
I_I	Input Load Current for NOE	$0 \leq V_{IN} \leq V_{CC}$	−100		100	μA
I_L	Leakage Current (Output and I/O Pins in TRI-STATE®/Input Mode)	$0.4 \leq V_{OUT} \leq 2.4V$	−20.0		20.0	μA
I_{CC}	Active Supply Current	I_{OUT} = 0, T_A = 25°C, V_{CC} = 5V			300	mA
I_{CC}	Power Down Current	I_{OUT} = 0, T_A = 25°C, V_{CC} = 5V			60	mA

*Except NOE

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 *Figure 4.2* and *4.3* unless specifically stated otherwise.

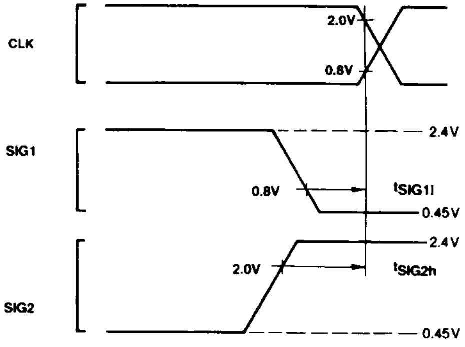


TL/EE/10828-16

FIGURE 4-2. 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/10828-17

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

4.0 Device Specifications (Continued)

4.4.2 Timing Tables

4.4.2.1 Output Signal Propagation Delays

(Maximum times assume capacitive loading of 100 pF at 15 MHz and 50 pF at 20 MHz and 25 MHz)

Symbol	Figure	Parameter	Reference/ Conditions	NS32181-15		NS32181-20		NS32181-25		Units
				Min	Max	Min	Max	Min	Max	
t_{Dv}	4-7	Data Valid (D0–D15)	After $\overline{\text{SPC}}$ L.E.		30		23		18	ns
t_{Df} (Note 1)	4-7	D0–D15 Floating	After $\overline{\text{SPC}}$ T.E.		30		30		30	ns
t_{SPCF_w}	4-9	$\overline{\text{SPC}}$ Pulse Width from FPU	At. 0.8V (Both Edges)	$t_{CLK_p}-10$	$t_{CLK_p}+10$	$t_{CLK_p}-10$	$t_{CLK_p}+10$	$t_{CLK_p}-10$	$t_{CLK_p}+10$	ns
t_{SPCF_a}	4-9	$\overline{\text{SPC}}$ Output Active	Active CLK R.E.		17		17		15	ns
$t_{SPCF_{ia}}$	4-9	$\overline{\text{SPC}}$ Output Inactive	Active CLK R.E.		38		33		25	ns
t_{SPCF_f} (Note 1)	4-9	$\overline{\text{SPC}}$ Output Floating	After CLK F.E.		35		30		25	ns

4.4.2.2 Input Signal Requirements

Symbol	Figure	Parameter	Reference/ Conditions	NS32181-15		NS32181-20		NS32181-25		Units
				Min	Max	Min	Max	Min	Max	
t_{PWR}	4-5	Power-On Reset Duration	After CLK R.E.	30		30		30		µs
t_{RST_w}	4-6	Reset Pulse Width	At 0.8V (Both Edges)	64		64		64		t_{CLK_p}
t_{RST_s}	4-6	Reset Setup Time	Before CLK R.E.	10		14		12		ns
t_{RST_h}	4-6	Reset Hold	After CLK R.E.	0		0		0		ns
t_{S_s}	4-8	Status (ST0–ST1) Setup	Before $\overline{\text{SPC}}$ L.E.	20		20		15		ns
t_{S_h}	4-8	Status (ST0–ST1) Hold	After $\overline{\text{SPC}}$ L.E.	20		20		20		ns
t_{D_s}	4-8	Data Setup (D0–D15)	Before $\overline{\text{SPC}}$ T.E.	25		20		15		ns
t_{D_h}	4-8	Data Hold (D0–D15)	After $\overline{\text{SPC}}$ T.E.	20		20		15		ns
t_{SPC_w}	4-8	$\overline{\text{SPC}}$ Pulse Width from CPU	At 0.8V (Both Edges)	35		35		28		ns
t_{SPC_s}	4-14, -15	$\overline{\text{SPC}}$ Setup	Before CLK R.E.	35		35		28		ns
t_{SPC_h}	4-14, -15	$\overline{\text{SPC}}$ Hold	After CLK R.E.	0		0		0		ns

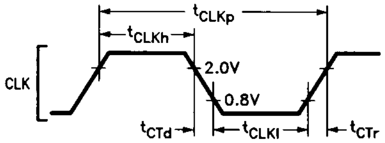
4.4.2.3 Clocking Requirements

Symbol	Figure	Parameter	Reference/ Conditions	NS32181-15		NS32181-20		NS32181-25		Units
				Min	Max	Min	Max	Min	Max	
t_{CLK_h}	4-4	Clock High Time	At 2.0V (Both Edges)	$0.5\ t_{CLK_p}-8.3$		$0.5\ t_{CLK_p}-5$		$0.5\ t_{CLK_p}-5$		ns
t_{CLK_l}	4-4	Clock Low Time	At 0.8V (Both Edges)	$0.5\ t_{CLK_p}-8.3$		$0.5\ t_{CLK_p}-5$		$0.5\ t_{CLK_p}-4$		ns
t_{CT_r} (Note 1)	4-4	Clock Rise Time	Between 0.8V and 2.0V		7		5		4	ns
t_{CT_d} (Note 1)	4-4	Clock Fall Time	Between 2.0V and 0.8V		7		5		4	ns
t_{CLK_p} (Note 2)	4-4	Clock Period	CLK R.E. to Next CLK R.E.	66	1000	50	1000	40	1000	ns

Note 1: Guaranteed by characterization. Due to tester conditions, this parameter is not 100% tested.

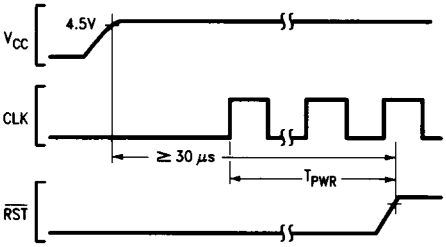
Note 2: The 1 MHz clock frequency is allowed for power-down mode only. Floating point instructions can be executed only after the clock frequency has been brought back to normal operating frequency.

4.4.3 Timing Diagrams



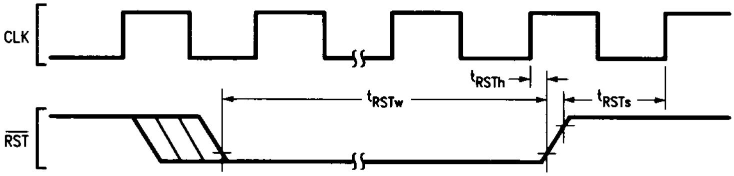
TL/EE/10828-18

FIGURE 4-4. Clock Timing



TL/EE/10828-19

FIGURE 4-5. Power-On Reset



TL/EE/10828-20

FIGURE 4-6. Non-Power-On Reset

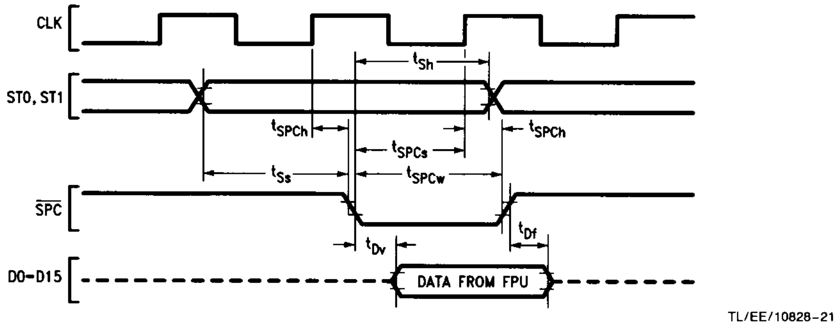


FIGURE 4-7. Read Cycle from FPU

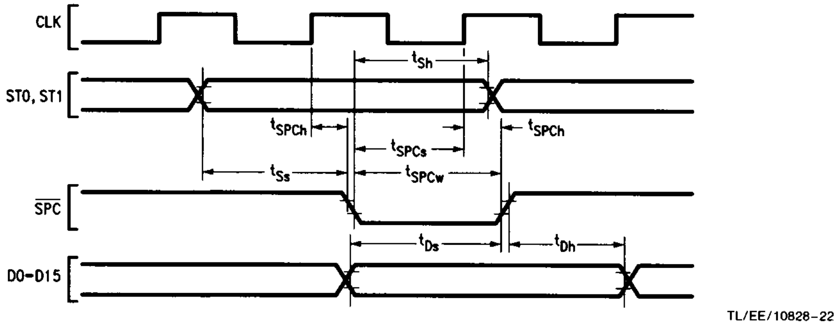


FIGURE 4-8. Write Cycle to FPU

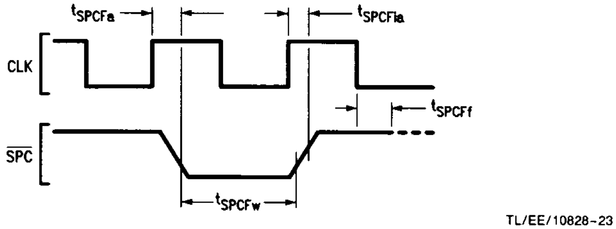


FIGURE 4-9. $\overline{SPC}$ Pulse from FPU

Appendix A: Instruction Execution Times

The FPU execution times are provided in Table A-1. The values given in the TFPU column represent FPU performance numbers which are CPU independent.

They provide the number of clock cycles from the $\overline{\text{SPC}}$ pulse from the CPU for the last operand transfer until the activation of $\overline{\text{SPC}}$ by the FPU to signal completion of the instruction.

Some FPU instructions show substantially different values depending upon whether the result is in a register or in memory. This is due to the Early Done algorithm described in Section 3.6.2.

If an FPU instruction immediately follows another FPU instruction that takes advantage of the Early Done algorithm, the FPU will delay execution of the second instruction until the first instruction is completed.

Since the execution time depends on a variety of factors like data, rounding mode and trap-enable bit settings, there may be variations of few clock cycles from the values shown.

NOTATIONS

n1 → n2 = Range of Values
n1 = minimum
n2 = maximum

i = Integer Type Field
B = Byte
W = Word
D = Double Word

f = Floating-Point Type Field
F = Std. Floating: 32 bits
L = Long Floating: 64 bits

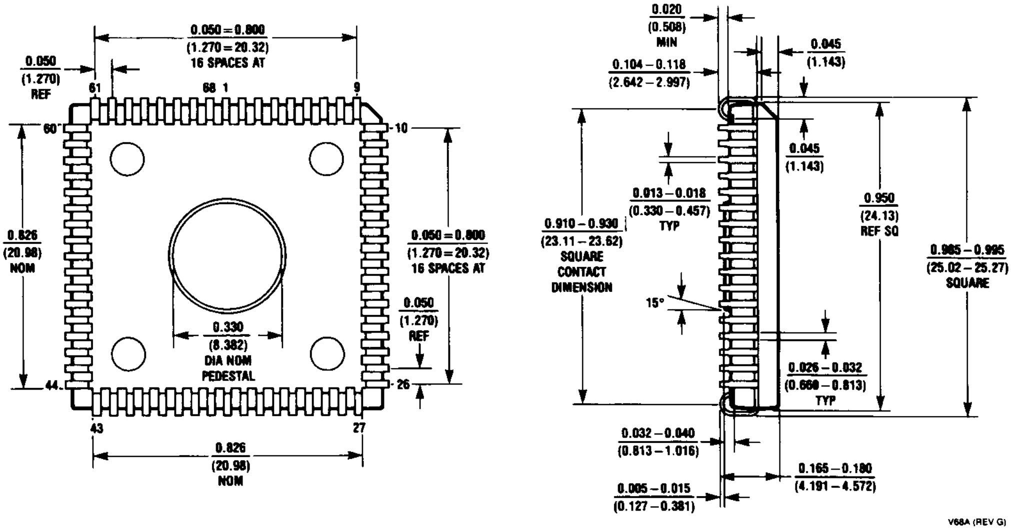
TABLE A-1. FPU Execution Times

Mnemonic	TFPU	Notes
MOVLF	19	
ROUNDf i, TRUNCf i, FLOORf i,	39 → 59	
DIVF	56 11	Result in Memory Result in Register
DIVL	72 11	Result in Memory Result in Register
ABSf	9	
NEGf	9	
MOVFL	11 9	Result in Memory Result in Register
LFSR	7	
SFSR	7	
MOVBf, MOVWf, MOVDf,	45 → 57 5	Result in Memory Result in Register
Trap (ILL)	6	Illegal Instruction Trap
MULF	33	Result in Memory
MULL	31 → 40	Result in Memory
MULf	11	Result in Register
MOVf	7	
CMPf	11 → 27	
ADDF, SUBF	28 → 72	Result in Memory
ADDL, SUBL	28 → 296	Result in Memory
ADDf, SUBf	11	Result in Register

NS32181-15/NS32181-20/NS32181-25 Floating-Point Unit

Physical Dimensions inches (millimeters)

Lit. # 114273



68-Pin Plastic Leaded Chip Carrier (V)
Order Number NS32181V-15, NS32181V-20 or NS32181V-25
NS Package Number V68A

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.

023743 ✓

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Industriestrasse 10
D-8080 Furstenfeldbruck
West Germany
Tel: (0-81-41) 103-0
Telex: 527-649
Fax: (08141) 103554

National Semiconductor Japan Ltd.
Sanseido Bldg. 5F
4-15 Nishi Shinjuku
Shinjuku-Ku,
Tokyo 160, Japan
Tel: 3-299-7001
FAX: 3-299-7000

National Semiconductor Hong Kong Ltd.
Suite 513, 5th Floor
Chinachem Golden Plaza,
77 Mody Road, Tsimshatsui East,
Kowloon, Hong Kong
Tel: 3-7231290
Telex: 52996 NSSEA HX
Fax: 3-3112536

National Semicondutores Do Brasil Ltda.
Av. Brig. Faria Lima, 1383
6.0 Andor-Conj. 62
01451 Sao Paulo, SP, Brasil
Tel: (55/11) 212-5066
Fax: (55/11) 211-1181 NSBR BR

National Semiconductor (Australia) PTY, Ltd.
1st Floor, 441 St. Kilda Rd.
Melbourne, 3004
Victory, Australia
Tel: (03) 267-5000
Fax: 61-3-2677458

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.