Uwe

Les Wilson
Systems Application Engineer

National Semiconductor

4827 Sepulveda Boulevard
Suite 180
Sherman Oaks, California 91403
Telephone 818 995-8335

NS16081 FPU As a Peripher

- Cautions and Current Bugs
- 68K-to-FPU Interfacing Example

ORDER SAMPLE AS NS16081D-6 OR -10

SOON... NS32081D-6
-8
-10

Les Wilson
Systems Application Engineer

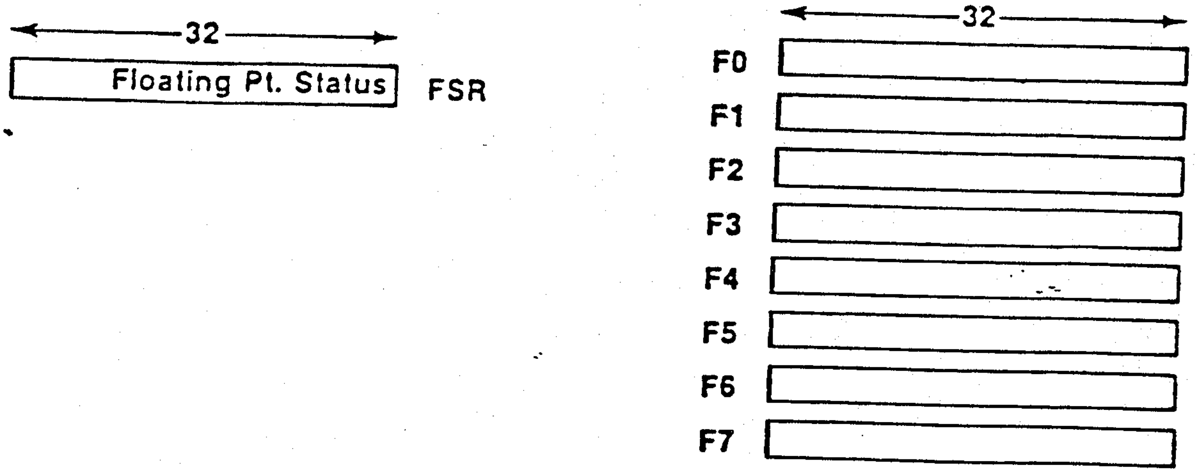
National Semiconductor

4827 Sepulveda Boulevard
Suite 180
Sherman Oaks, California 91403
Telephone 213 995-8335

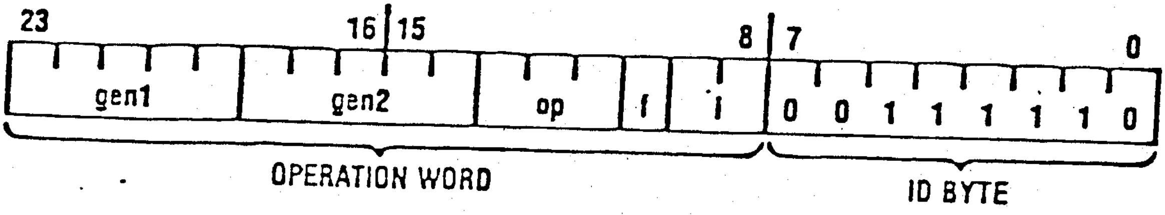
Note: Please refer to NS16081 Data sheet.
New part number for NS16081 is NS32081.
A Final typset version of this is in progress.

March 84

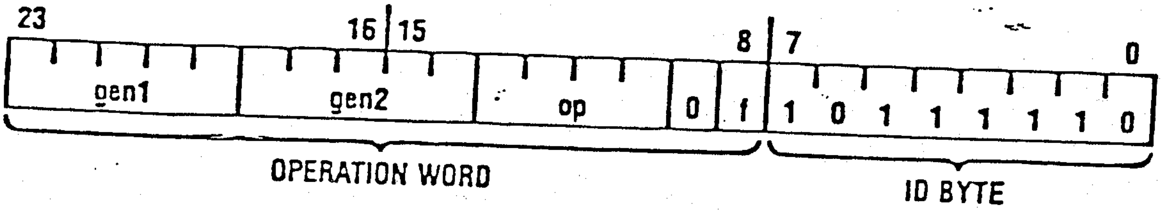
FPU Registers



FPU Instruction Formats



Format 9 : LFSR/SFSR/Conversions



Format 11 : Movement/Calculation

Addressing Modes

0	0		n	

FPU Internal Register:

Fn, n = 0...7

Long Floating: even register only!

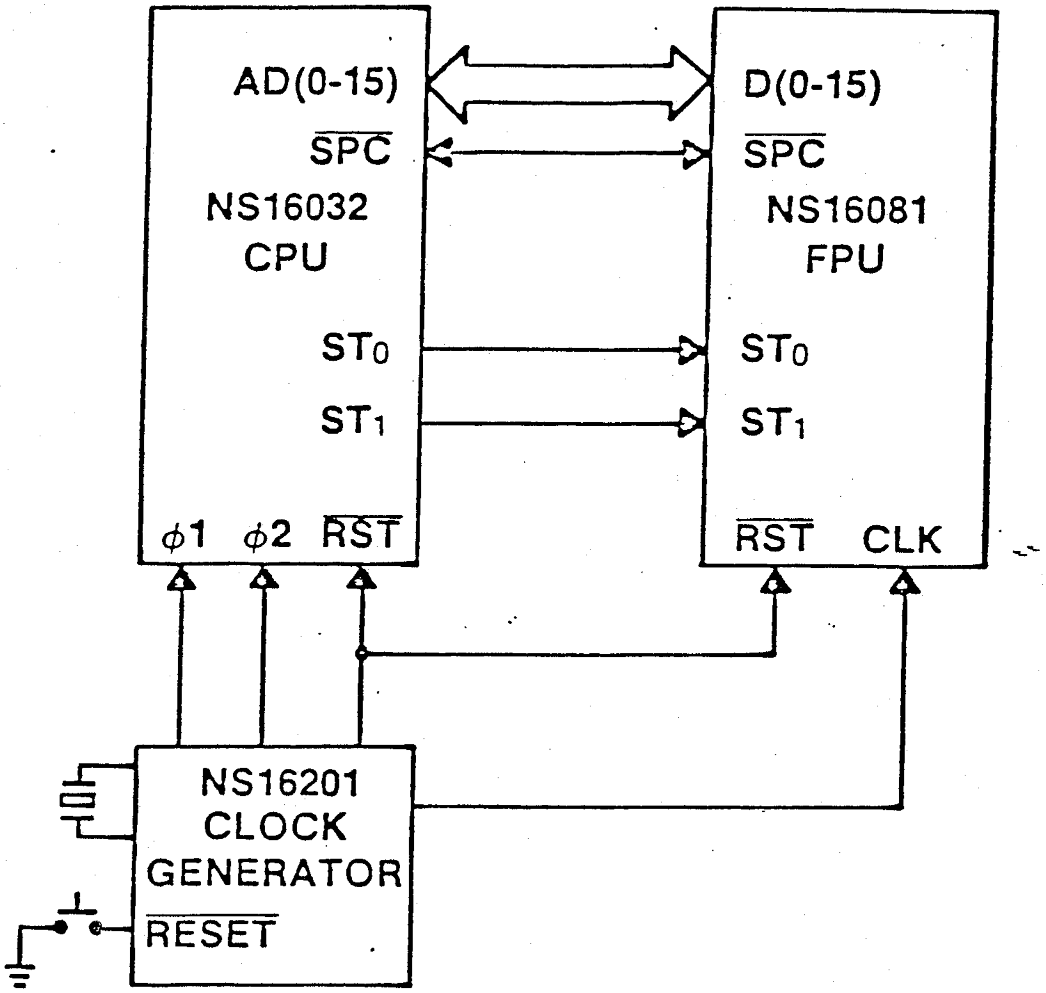
0	1	x	x	x

1	x	x	x	x

External to FPU

Note: All non-Floating operands are always external.

SYSTEM CONNECTION DIAGRAM



SLAVE PROTOCOL

Status Combinations:

11 : Send ID (ID)
01 : Xfer Operand (OP)
10 : Read Status (ST)

Step	Status	Action
1	ID	CPU Sends ID Byte.
2	OP	CPU Sends OPcode. (Bytes Swapped)
3	OP	CPU Sends Required Operands. (Least-Sig. Word first.)
4	—	Slave Starts Execution. CPU Pre-fetches.
5	—	Slave Pulses SPC Low.
6	ST	CPU Reads Status Word. (Trap? Alter Flags?)
7	OP	CPU Reads Result (If Any). (Least-Sig. Word first.)

Summary of Interfacing Considerations

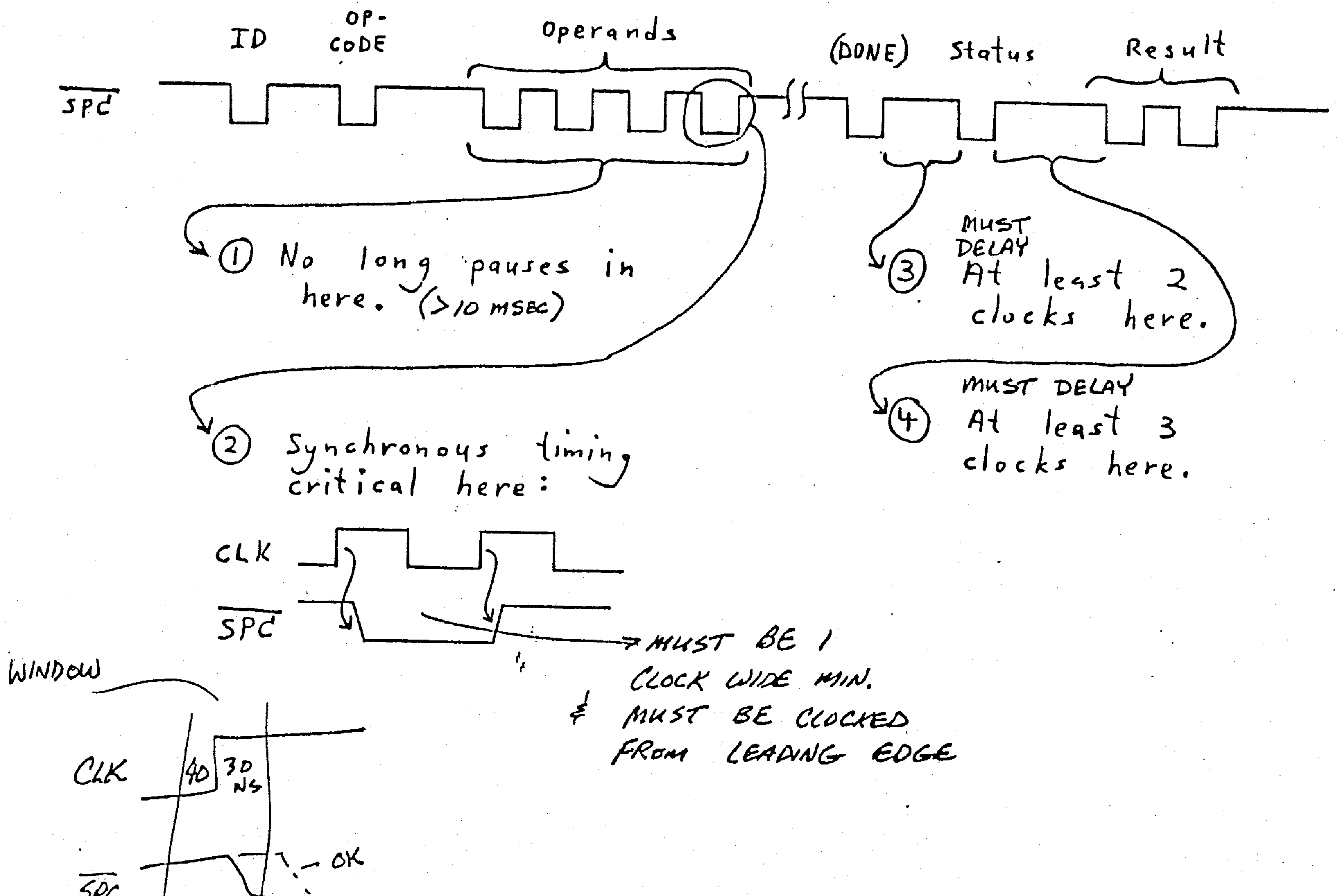
- In multitasking systems, don't interrupt a half-complete FPU instruction and allow another program to use the FPU. Set a semaphore bit in software while an FPU instruction is in progress. This gives the OS enough information to avoid such problems.

- Don't use asynchronous SPC pulses.

  This is a syncronous interface

- Don't "single-step" the protocol manually. (Long pauses cause problems in current FPU revisions.)

Interfacing to FPU : Cautions



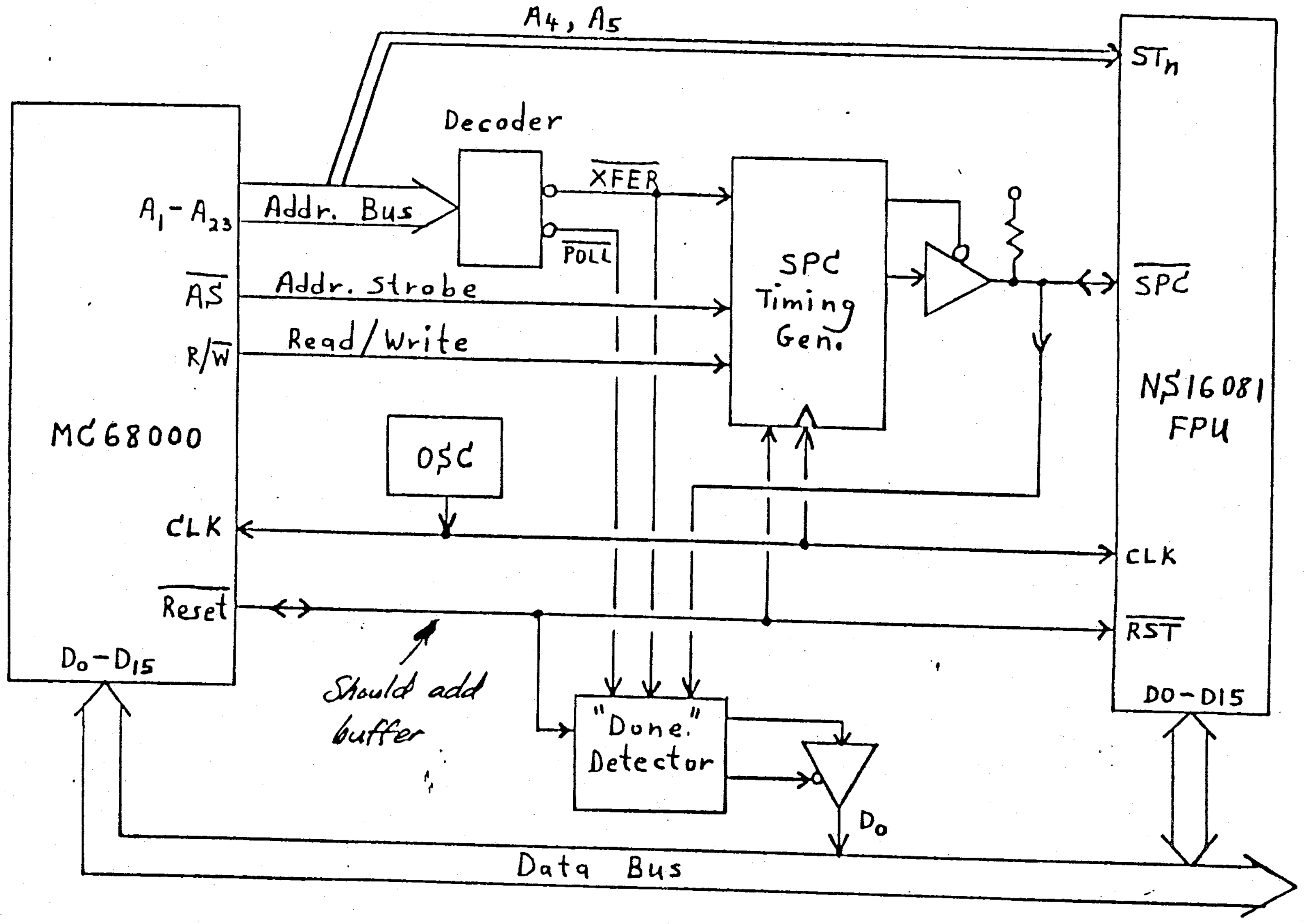
① No long pauses in here. (>10 MSEC)

② Synchronous timing critical here:

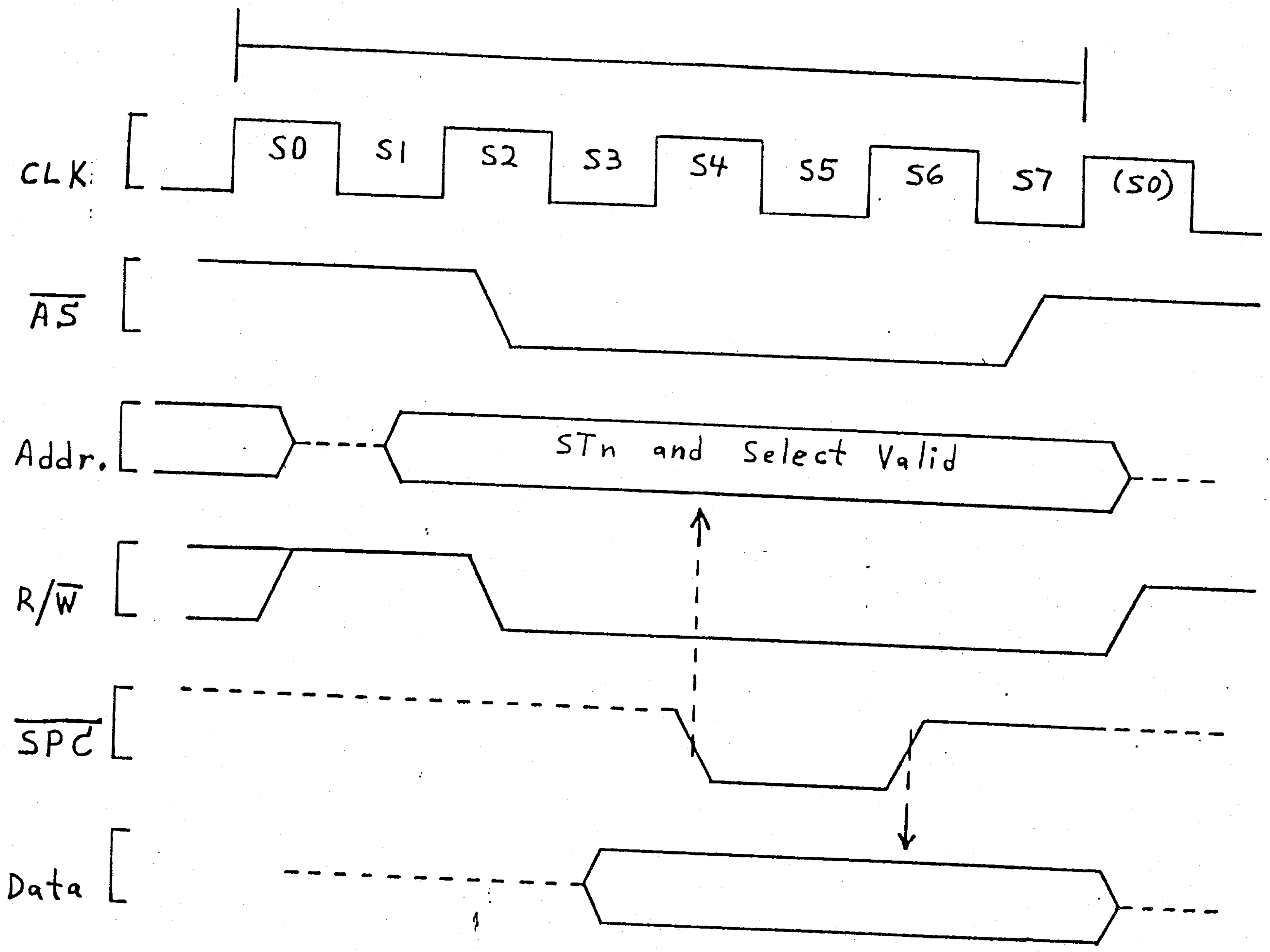
→ MUST BE 1 CLOCK WIDE MIN.
& MUST BE CLOCKED FROM LEADING EDGE

③ MUST DELAY At least 2 clocks here.

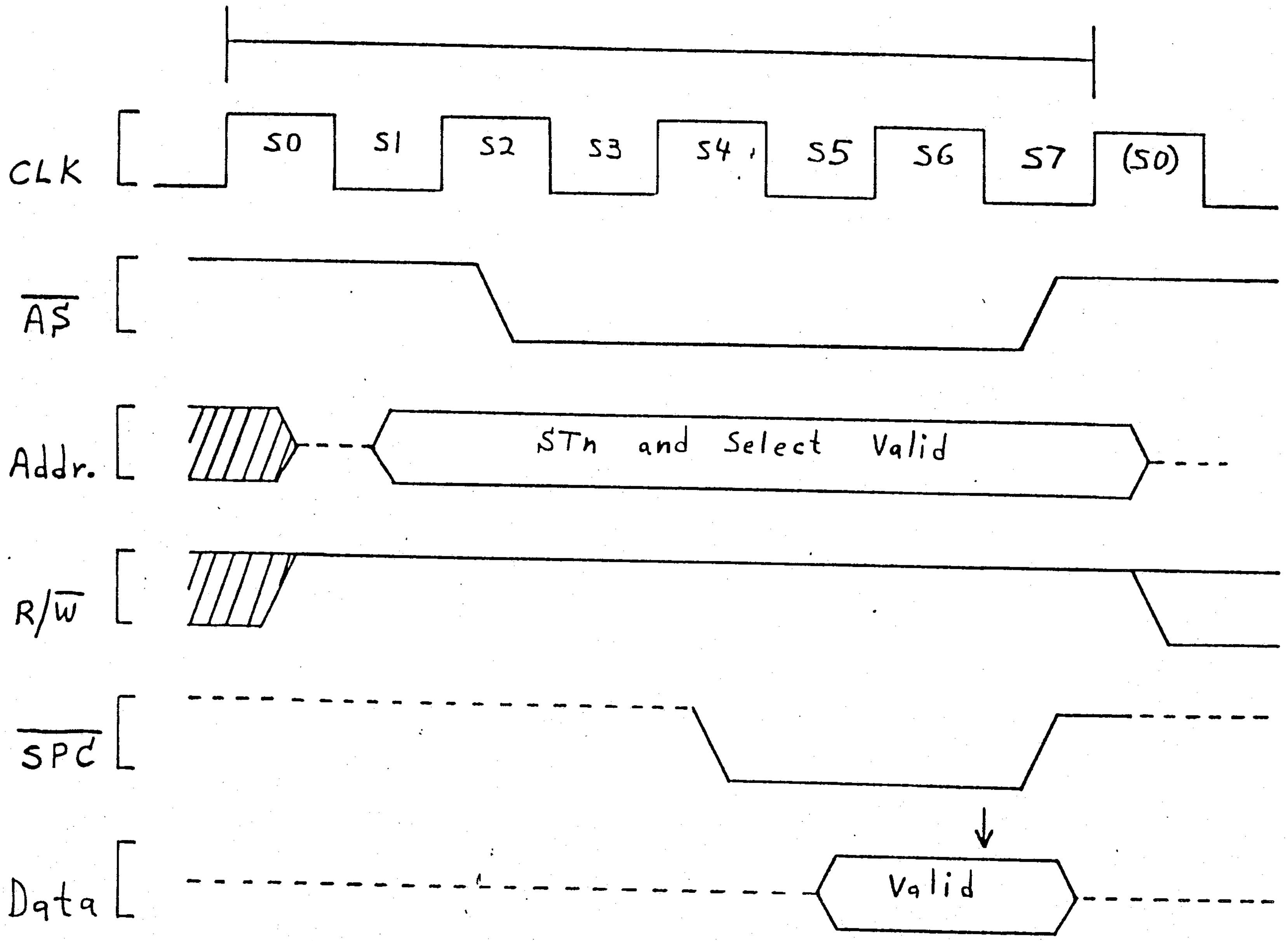
④ MUST DELAY At least 3 clocks here.



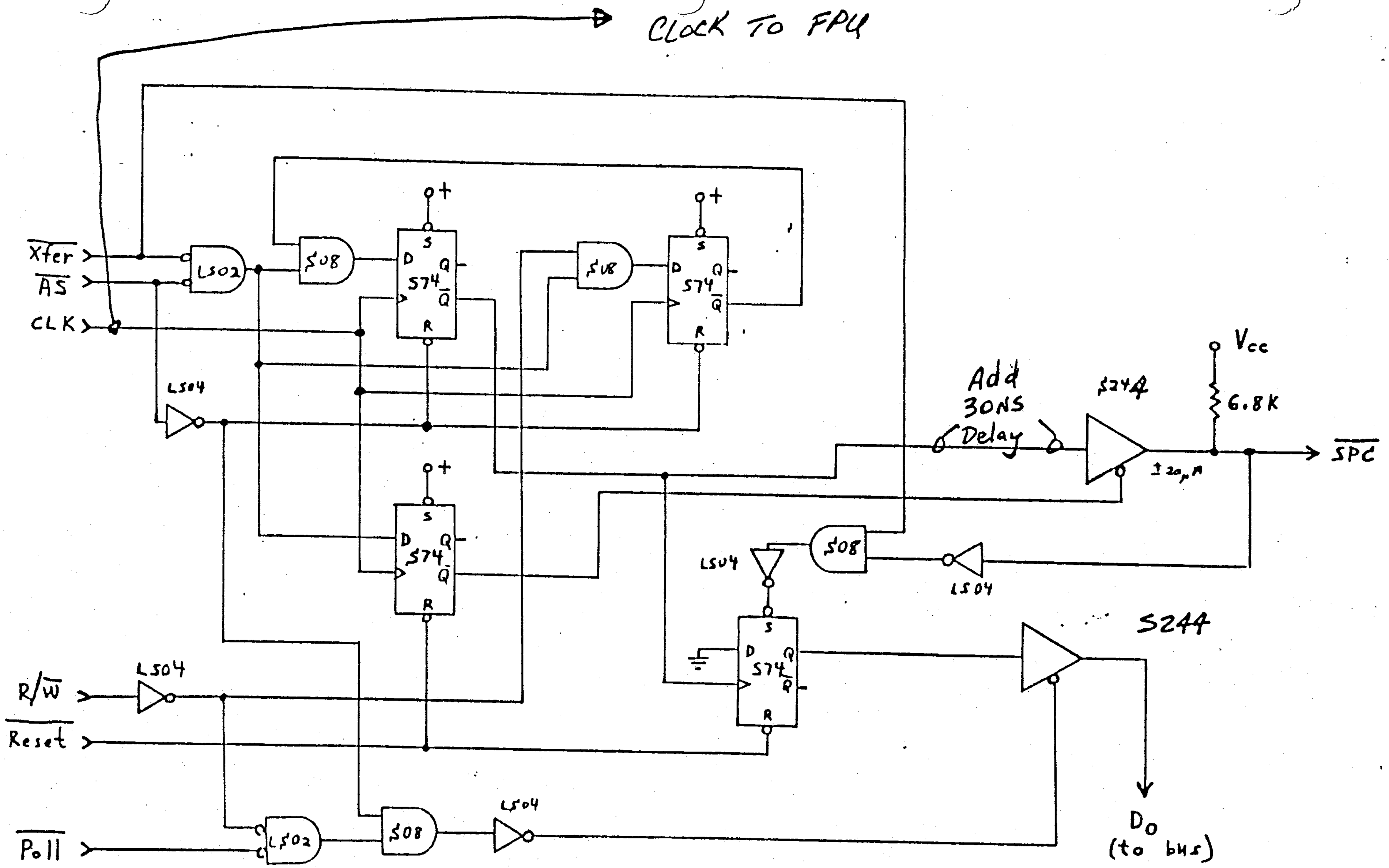
A4, A5
Decoder
$\overline{XFER}$
$\overline{POLL}$
SPC Timing Gen.
ST_n
$\overline{SPC}$
NS16081 FPU
A_1-A_{23}
Addr. Bus
$\overline{AS}$
Addr. Strobe
$R/\overline{W}$
Read/Write
MC68000
OSC
CLK
CLK
$\overline{Reset}$
$\overline{RST}$
D_0-D_{15}
D0-D15
Should add buffer
"Done" Detector
D_0
Data Bus



68K Write to FPU



68K Read from FPU



SPC Generator + "Done" Detector

68K to FPU Interface

Software Cautions

1. Beware of byte alignment:

   In the 68K, the even byte address is at the TOP half of the bus. Transfer only whole words to/from the FPU, or make certain to transfer single bytes at an odd address.

2. The FPU transfers multiple-word operands least-significant word first. The 68K transfers them most-significant word first. Make certain that software sends multiple-word values to the FPU correctly.

```
*                Single-Precision Addition (Demo Routine)
*
*     Register Contents:
*
*        A0 = 00070000   Address of DONE flip-flop.
*        A1 = 00060010   Address for ST=1 transfer (Xfer Operand).
*        A2 = 00060020   Address for ST=2 transfer (Read Status Word).
*        A3 = 00060030   Address for ST=3 transfer (Broadcast ID).
*
*        D0 = 000000BE   ID byte for ADDF instruction.
*        D1 = 00000184   Operation Word for ADDF.  (Note bytes swapped.)
*        D2 = 3F800000   First operand = 1.0  .
*        D3 = 3F800000   Second operand = 1.0 .
*        D4              Receives Status Word from FPU.
*        D5              Receives result from FPU.
*        D7              Scratch register (for DONE bit test).
*
START    MOVE.W   D0,(A3)           Send ID byte.
         MOVE.W   D1,(A1)           Send Operation Word.
         SWAP     D2                Send operands.  The swapping
         MOVE.L   D2,(A1)             is included because the
         SWAP     D2                  FPU expects the least-
         SWAP     D3                  significant word first.
         MOVE.L   D3,(A1)             (Can be avoided, with care.)
         SWAP     D3
*
POLL     MOVE.W   (A0),D7           Check the DONE flip-flop,
         BTST     #0,D7               loop until FPU is finished.
         BEQ.S    POLL
*
         MOVE.W   (A2),D4           Read Status Word.
         MOVE.L   (A1),D5           Read result.
         SWAP     D5                Swap halves of result.
```

Originated 1/9/84

Last Revised 2/22/84

1. When the MOVLF instruction is executed with a source operand containing zero, the FPU generates an Underflow indication.

2. Asynchronous timing of nSPC pulses with respect to CLK does not work reliably. It appears from examination of the FPU logic model (RTL) that the last operand transfer to the FPU must follow 16K CPU timing exactly: i.e. the nSPC pulse must start shortly after a CLK rising edge and terminate shortly after the next CLK rising edge.

3.. At the phase of the protocol in which operands are being transferred to the FPU, no long pauses (on the order of tens of milliseconds) may occur. There is apparently a dynamic discharge problem which causes the FPU's state to decay. This has not yet been confirmed to be the cause of the problem, however (as of 2/21).

4. Documentation errors:

The FSR TT field is loaded with a new error status value (zero if no error) at the end of every floating-point instruction except LFSR or SFSR. (The LFSR instruction loads the TT field, but with the value supplied by the programmer instead of with error status.)

Whenever an FPU error condition occurs, the FSR TT field is loaded with the error code, regardless of whether that condition is enabled to cause a trap. An FPU instruction can therefore complete normally and still display a code of 1 (Underflow) or 6 (Inexact Result). This code remains in the TT field only until the next floating-point instruction (other than SFSR) completes.

Action for NS16000 Instruction Set Manual and the NS16081 data sheet(s).