APPENDIX B

NS32016 INSTRUCTION EXECUTION TIMES

B.1 Assumptions

The entire instruction, with all displacements and immediate operands, is assumed to be present in the instruction queue when needed.

Interference from instruction prefetches, which is very dependent upon the preceding instruction(s), is ignored. This assumption will tend to affect the timing estimate in an optimistic direction.

It is assumed that all memory operand transfers are completed before the next instruction begins execution. In the case of an operand of access class rmw in memory, this is pessimistic, as the Write transfer occurs in parallel with the execution of the next instruction.

It is assumed that there is no overlap between the fetch of an operand and the following sequences of microcode. This is pessimistic, as the fetch of Operand A will generally occur in parallel with the effective address calculation of Operand B, and the fetch of Operand B will occur in parallel with the execution phase of the instruction.

Where possible, the values of operands are taken into consideration when they affect instruction timing, and a range of times is given. Where this is not done, the worst case is assumed.

B.2 Definitions

TEA - The time required to calculate an operand's Effective Address. For a Register or Immediate operand, this includes the fetch of that operand.

TMMU - The extra clock cycle required for translation of memory addresses if an MMU is present.

TOPB - The time needed to read or write a memory byte.

TOPW - The time needed to read or write a memory word.

TOPD - The time needed to read or write a memory double-word.

TOPi - The time needed to read or write a memory operand, where the operand size is given by the operation length of the instruction. It is always equivalent to either TOPB, TOPW or TOPD.

TCY - Internal processing overhead, in clock cycles.

L - Internal processing whose duration depends on the operation length. The number of clock cycles is derived by multiplying this value by the number of bytes in the operation length.

B.3 Equations

```
TMMU -  If an MMU is present then TMMU=1
        else TMMU=0

TOPB -  If operand is in a register or is immediate then TOPB=0
        else TOPB = 3 + TMMU

TOPW -  If operand is in a register or is immediate then TOPW=0
        else if word-aligned (even address) then TOPW = 3 + TMMU
             else TOPW - 7 + 2 * TMMU

TOPD -  If operand is in a register or is immediate then TOPD=0
        else if word-aligned (even address) then TOPD = 7 + 2 * TMMU
             else TOPD = 11 + 3 * TMMU

TOPi -  If operand is in a register or is immediate then TOPi=0
        else if i=byte then TOPi = TOPB
        else if i=word then TOPi = TOPW
        else (i=double-word) then TOPi = TOPD

TCY  -  TCY = 1

L    -  If i (operation length) = byte then L = 1
        else if i = word then L = 2
        else (i = double-word) L = 4

TEA  -  If REGISTER addressing then TEA = 2
        if IMMEDIATE or ABSOLUTE addressing then TEA = 4
        if REGISTER RELATIVE or MEMORY SPACE addressing then TEA = 5
        if MEMORY RELATIVE addressing then TEA = 7 + TOPD
        if TOP OF STACK addressing then
             if access class = write then TEA = 4
             if access class = read then TEA = 2
             else TEA = 3
        if EXTERNAL addressing then TEA = 11 + 2 * TOPD
        if SCALED INDEXED addressing then TEA = TI1 + TI2

             where TI1 depends on scale factor:
                    if byte indexing TI1 = 5
                    if word indexing TI1 = 7
                    if double-word indexing TI1 = 8
                    if quad-word indexing TI1 = 10

             and   TI2 = TEA of the basemode except:
                             if basemode is REGISTER then TI2 = 5
                             if basemode is TOP OF STACK then TI2 = 4
```

B.4 Calculation of Total Execution Time (TEX)

TEX is obtained by performing the following steps:

1. Find the desired instruction in the table.

2. Calculate the values for TEA, TOPB, etc. using the numbers in the table and the equations given on the preceding page.

3. The result derived by adding together these values is the execution time (TEX) in clock cycles.

B.5 Notes on Table Use

Values in the TEA column indicate the number of effective addresses to be calculated. If the value in this column is less than the number of general operands in the instruction, this is because one or both operands are in registers and that instruction has an optimized form which eliminates TEA for such operands.

In the L column, multiply the entry by the operation length in bytes (1, 2 or 4).

In the TCY column, special notations sometimes appear:

n1-n2 means n1 minimum, n2 maximum.

n1%n2 means that the instruction flushes the instruction queue after n1 clock cycles and nonsequentially fetches the next instruction. The value n2, indicating the total number of clock cycles in internally executing the instruction (including n1), is not generally useful. The most accurate technique for determining such timing depends on the size and alignment of the Basic Instruction portion of the next instruction, plus Index Bytes. If this portion can be read in one memory cycle, then the execution time is n1+10 (including the memory cycle). If more memory cycles are required, the value is n1+5+4*m, where m is the number of memory cycles required.

In the Notes column, notations held within angle brackets <> indicate alternatives in the form of the instruction which affect the execution time. A table entry which is affected by the form of the instruction may have multiple values, separated by slashes, corresponding to the alternatives. The notations are:

<M>	Memory form
<R>	Register form
<MM>	Memory-to-Memory form
<RM>	Register-to-Memory form
<MR>	Memory-to-Register form
<RR>	Register-to-Register form
x	either Register or Memory

B.6 Example of Table Usage

Calculate TEX for the instruction:

```
CMPW  R0,TOS
```

Operand A is in a register; Operand B is in memory. This means that we must use the table values corresponding to the <xM> case as given in the Notes column (<xM> meaning "anything to memory").

Only the TEA, TOPi and TCY columns have values assigned for the CMPi instruction. Therefore, they are the only ones that need to be calculated to find TEX. The blank columns are irrelevant to this instruction.

The TEA column contains 2 for the <xM> case. This means that effective address times have to be calculated for both operands. (For the <MR> case, the Register operand would have required no TEA time, therefore only the Memory operand TEA would have been necessary.) From the equations:

```
TEA (Register mode)  =  2,
TEA (Top of Stack mode, read access class) =   2,

Total TEA = 2+2 = 4.
```

The TOPi column represents potential operand transfers to or from memory. For a Compare instruction, each operand is read once, for a total of two operand transfers.

```
TOPi (Word, Register) = 0,
TOPi (Word, TOS) = TOPW = 3  (assuming aligned, no MMU)
Total TOPi = 3
```

TCY is the time required for internal operation within the CPU. The TCY value for this case is 3.

TEX = TEA + TOPi + TCY = 4 + 3 + 3 = 10 machine cycles.

If the CPU is running at 10 MHz then a machine cycle (clock cycle) is 100 nsec. Therefore, this instruction would take 10x100 nsec, or 1.0 micro-seconds, to execute.

Table B-1 Basic and Memory Management Instructions

This table does not include the timings for Floating-Point instructions. See Section B.7 and Table B-2.

MNEMONIC	TEA	TOPB	TOPW	TOPD	TOPi	TCY	L	NOTES
ABSi	2	-	-	-	2	9/8	-	src<0 / src>=0
ACBi	1	-	-	-	2	16/15%20	-	<M>, no branch / branch
ACBi	-	-	-	-	-	18/17%22	-	<R>, no branch / branch
ADDi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>
ADDCi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>
ADDPi	2	-	-	-	3	16/18	-	no carry / carry
ADDQi	1/0	-	-	-	2/0	6/4	-	<M>/<R>
ADDR	2/1	-	-	1/0	-	2/3	-	<xM>/<xR>
ADJSPi	1	-	-	-	1	6	-	
ANDi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>
ASHi	2	1	-	-	2	14-45	-	
Bcond	-	-	-	-	-	7/6%10	-	no branch / branch
BICi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>
BICPSRB	1	1	-	-	-	18%22	-	
BICPSRW	1	-	1	-	-	30%34	-	

Table B-1 (Cont.)

MNEMONIC	TEA	TOPB	TOPW	TOPD	TOPi	TCY	L	NOTES
BISPSRB	1	1	-	-	-	18%22	-	
BISPSRW	1	-	1	-	-	30%34	-	
BPT	-	-	4	3	-	40	-	
BR	-	-	-	-	-	6%10	-	
BSR	-	-	-	1	-	6%16	-	
CASEi	1	-	-	-	1	4%9	-	
CBITi	2/1	2/0	-	-	1	15/7	-	<xM>/<xR>
CBITIi	2/1	2/0	-	-	1	15/7	-	<xM>/<xR>
CHECKi	2	-	-	-	3	7/10/11	-	high / low / ok
CMPi	2/1/0	-	-	-	2/1/0	3	-	<xM>/<MR>/<RR>
CMPMi	2	-	-	-	2*n	9*n+24	-	n = # of elements in block
CMPQi	1/0	-	-	-	1/0	3	-	<M>/<R>
CMPSi	-	-	-	-	2*n	35*n+53	-	n = # of elements, not Translated
CMPST	-	n	-	-	2*n	38*n+53	-	Translated
COMi	2	-	-	-	2	7	-	
CVTP	2	-	-	1	-	7	-	
CXP	-	-	3	4	-	16%21	-	

Table B-1 (Cont.)

MNEMONIC	TEA	TOPB	TOPW	TOPD	TOPi	TCY	L	NOTES
CXPD	1	-	3	3	-	13%18	-	
DEIi	2/1	-	-	-	5/1	38/31	16	<xM>/<xR>
DIA	-	-	-	-	-	3%7	-	
DIVi	2	-	-	-	3	58-68	16	
ENTER	-	-	-	n+1	-	4*n+18	-	n = # of general registers saved
EXIT	-	-	-	n+1	-	5*n+17	-	n = # of general registers restored
EXTi	2	-	-	1	1	19-29	-	field in memory
EXTi	2	-	-	-	1	17-51	-	field in register
EXTSi	2	-	-	1	1	26-36	-	
FFSi	2	2	-	-	1	24-28	24	
FLAG	-	-	0/4	0/3	-	6/44	-	no trap / trap
IBITi	2/1	2/0	-	-	1	17/9	-	<xM>/<xR>
INDEXi	2	-	-	-	2	25	16	
INSi	2	-	-	2	1	29-39	-	field in memory
INSi	1	-	-	-	1	28-96	-	field in register
INSSi	2	-	-	2	1	39-49	-	
JSR	1	-	-	1	1	5%15	-	

Table B-1 (Cont.)

MNEMONIC	TEA	TOPB	TOPW	TOPD	TOPi	TCY	L	NOTES
JUMP	1	-	-	-	-	2%6	-	
LMR	1	-	-	-	1	30%34	-	
LPRi	1	-	-	-	1	19-33	-	
LSHi	2	1	-	-	2	14-45	-	
MEIi	2	-	-	-	4	23	16	
MODi	2	-	-	-	3	54-73	16	
MOVi	2/1/0	-	-	-	2/1/0	1/3/3	-	<xM>/<MR>/<RR>
MOVMi	2	-	-	-	2*n	3*n+20	-	n = # of elements in block
MOVQi	1/0	-	-	-	1/0	2/3	-	<M>/<R>
MOVSi	-	-	-	-	2*n	13*n+18	-	n = # of elements, no options
MOVSi	-	-	-	-	2*n	24*n+54	-	B, W and/or U option in effect
MOVST	-	n	-	-	2*n	27*n+54	-	Translated
MOVSUi	2	-	-	-	2	33%37	-	
MOVUSi	2	-	-	-	2	33%37	-	
MOVXBD	2	1	-	1	-	6	-	
MOVXBW	2	1	1	-	-	6	-	
MOVXWD	2	-	1	1	-	6	-	

Table B-1 (Cont.)

MNEMONIC	TEA	TOPB	TOPW	TOPD	TOPi	TCY	L	NOTES
MOVZBD	2	1	-	1	-	5	-	
MOVZBW	2	1	1	-	-	5	-	
MOVZWD	2	-	1	1	-	5	-	
MULi	2	-	-	-	3	15	16	
NEGi	2	-	-	-	2	5	-	
NOP	-	-	-	-	-	3	-	
NOTi	2	-	-	-	2	5	-	
ORi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>
QUOi	2	-	-	-	3	49-55	16	
RDVAL	1	1	-	-	-	21	-	
REMi	2	-	-	-	3	57-62	16	
RESTORE	-	-	-	n	-	5*n+12	-	n = # of general registers restored
RET	-	-	-	1	-	2%8	-	
RETI	-	1	3	3	-	39%45	-	
RETT	-	-	2	2	-	35%41	-	
ROTi	2	1	-	-	2	14-45	-	
RXP	-	-	1	2	-	2%6	-	

Table B-1 (Cont.)

MNEMONIC	TEA	TOPB	TOPW	TOPD	TOPi	TCY	L	NOTES
Scondi	1	-	-	-	1	9/10	-	False / True
SAVE	-	-	-	n	-	4*n+13	-	n = # of general registers saved
SBITi	2/1	2/0	-	-	1	15/7	-	<xM>/<xR>
SBITIi	2/1	2/0	-	-	1	15/7	-	<xM>/<xR>
SETCFG	-	-	-	-	-	15	-	
SKPSi	-	-	-	-	n	27*n+51	-	n = # of elements, not Translated
SKPST	-	n	-	-	n	30*n+51	-	Translated
SMR	1	-	-	-	1	25	-	
SPRi	1	-	-	-	1	21-27	-	
SUBi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>
SUBCi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>
SUBPi	2	-	-	-	3	16/18	-	no carry / carry
SVC	-	-	4	3	-	40	-	
TBITi	2/1	1/0	-	-	1	14/4	-	<xM>/<xR>
WAIT	-	-	-	-	-	6-?	-	? = until an interrupt/reset
WRVAL	1	1	-	-	-	21	-	
XORi	2/1/0	-	-	-	3/1/0	3/4/4	-	<xM>/<MR>/<RR>

B.7 Floating-Point Execution Times

Table B-2 gives execution timing information for Floating-Point instructions. Some additional timing definitions are used, as given below.

f - The floating-point operation length.

Standard Floating (32 bits): f=1
Long Floating (64 bits): f=2

Tf - The time required to transfer 32 bits of a floating-point value to or from the NS32081 Floating-Point Unit.

Tf = 4 always.

Ti - The time required to transfer an integer value to or from the NS32081 Floating-Point Unit.

Byte: Ti = 2
Word: Ti = 2
Double-Word: Ti = 4

Table B-2 Floating-Point Instruction Execution Times

INSTRUCTION	CASE	TEA	TOPD	TOPi	Ti	Tf	TCY
MOVf	<MM>	2	2f	-	-	2f	23
	<RR>	-	-	-	-	-	27
	<MR>	1	f	-	-	f	23
	<RM>	-	f	-	-	f	27
ADDf, SUBf	<MM>	2	3f	-	-	3f	70
	<RR>	-	-	-	-	-	74
	<MR>	1	f	-	-	f	70
	<RM>	1	2f	-	-	2f	70
MULf	<MM>	2	3f	-	-	3f	30+14f
	<RR>	-	-	-	-	-	34+14f
	<MR>	1	f	-	-	f	30+14f
	<RM>	1	2f	-	-	2f	30+14f
DIVf	<MM>	2	3f	-	-	3f	55+30f
	<RR>	-	-	-	-	-	59+30f
	<MR>	1	f	-	-	f	55+30f
	<RM>	1	2f	-	-	2f	55+30f
ABSf, NEGf	<MM>	1	2f	-	-	2f	20
	<RR>	-	-	-	-	-	24
	<MR>	1	f	-	-	f	20
	<RM>	-	f	-	-	f	24
CMPf	<MM>	2	2f	-	-	2f	45
	<RR>	-	-	-	-	-	49
	<MR>	1	f	-	-	f	45
	<RM>	1	f	-	-	f	45
MOVLF	<MM>	1	3	-	-	3	23
	<RR>	-	-	-	-	-	27
	<MR>	1	2	-	-	2	23
	<RM>	-	1	-	-	1	27
MOVFL	<MM>	1	3	-	-	3	22
	<RR>	-	-	-	-	-	26
	<MR>	1	1	-	-	1	22
	<RM>	-	2	-	-	2	26
MOVif	<MM>	1	f	1	1	f	53
	<MR>	1	-	1	1	-	53
ROUNDfi, TRUNCfi, FLOORfi	<MM>	1	f	1	1	f	53
	<RM>	-	-	1	1	-	66
SFSR	<M>	-	1	-	-	1	13
LFSR	<M>	1	1	-	-	1	18