Appendix A
CPU Instruction Timing

A.1 Introduction

The execution time for an instruction depends on:

- The type of instruction executed
- The mode of addressing used
- The type of memory being referenced

In general, the total execution time is the sum of the base instruction fetch/execute time plus the operand(s) address calculation/fetch time.

You can use the tables in this section to calculate the length of an instruction in terms of microcycles (MC). Tables A-1 through A-8 list the standard and floating point instructions, their op code listing, and execution times (MC). The "Execution MC" column specifies the number of microcycles required to fetch/execute the base instruction. The R/W column specifies the number of read microcycles (R) and write microcycles (W) in the Execution MC column. Any remaining microcycles are non-I/O (NIO).

If the instruction involves the calculation/fetch of one or more operands, a reference to a separate table (a source or destination table) is made in the last column. The column is usually labeled "Table" or "Dest Table." The tables referenced are A-9, A-10 through A-15, and A-16 through A-20: they are located at the end of the appendix. The source/destination tables specify the number of microcycles the source/destination calculation/fetch requires, and how many of these are read or write microcycles. As before, any remaining microcycles are NIO.

The numbers contained in the tables are based on the assumptions that:

- A memory read must last a minimum of four CLK periods
- A memory write must last a minimum of eight CLK periods
- An NIO lasts four CLK periods (no DMA)

Any wait states caused by slower memory or a DMA transfer must be added to the total instruction time. If wait states are required, the first wait state of a nonstretched read or NIO cycle will last four clock periods and can continue in increments of two clock periods. Further wait states for stretched cycles occur in increments of two clock periods.

Floating-point instruction execution times are given as a range. The actual execution time will vary depending on the type of data being operated on.

The following examples illustrate how to use the tables.

Example 1

How long does a MOV R0,@ 2044 instruction last?

Step 1 From Table A-2, the execution time for the MOV base instruction is 1 MC, or 4 CLK periods. This consists of one read and no write microcycles (R/W column). Depending on the type of memory in the system, the microcycle may be stretched. If so, the microcycle lasts at least 8 CLK periods and may be stretched thereafter in increments of 2 CLK periods.

Step 2 To find the operand calculation/fetch time for the source operand (R0), refer to Table A-9. As shown in Table A-9, a mode 0 register 0 calculate/fetch takes 0 MC. Note that the operand is already available to the DCJ11 (in the register file).

Step 3 To find the operand calculation/fetch time for the destination operand (the contents of memory location 2044), see Table A-12. Table A-12 specifies that a mode 3 register 7 calculate/fetch requires three microcycles (that is, one read microcycle and one write microcycle). Note that the remaining microcycle is an NIO microcycle.

The type of memory in the system must be taken into account. If the read cycle is stretched, the stretched cycle lasts at least 8 CLK periods and may be stretched thereafter in increments of 2 CLK periods. The write microcycle lasts at least 8 CLK periods and may be stretched in increments of 2 CLK periods.

Step 4 For a determination of the minimum time required, total up the microcycles. In this example, it is 1 + 0 + 3, or 4 MC (16 CLK periods if no microcycle stretching occurs).

Example 2

The source and destination tables for floating point instructions show a negative number in the microcycle column for certain mode 2 register 7 operations. For example, to determine how long a CLRD 2000 instruction lasts, you can follow steps 1 through 3:

Step 1 As specified in Table A-8, the base instruction time for the CLRD instruction is 14 MC.

Step 2 From Table A-17, the calculation/fetch time for the operand (a mode 2 register 7 reference) is shown as (-1) under Double Precision. This means that you subtract 1 MC from the base instruction time. However you add 1 MC for the memory write operation. There are no memory read cycles.

Step 3 Total the microcycles:

14 - 1 + 1 = 14 MC minimum.

Note that this example assumes no cycle stretching.

Table A-1 Single Operand Instructions

Mnemonic	Instruction	Op Code Listing	Timing			
			Execution MC	R/W	Source Table	Dest. Table
CLR(B)	Clear	0050DD	1	1/0	-	A-12
COM(B)	Complement (1's)	0051DD	1	1/0	-	A-13
INC(B)	Increment	0052DD	1	1/0	-	A-13
DEC(B)	Decrement	0053DD	1	1/0	-	A-13
NEG(B)	Negate (2's complement)	0054DD	1	1/0	-	A-13
TST(B)	Test	0057DD	1	1/0	-	A-13
Rotate and Shift						
ROR(B)	Rotate right	0060DD	1	1/0	-	A-13
ROL(B)	Rotate left	0061DD	1	1/0	-	A-13
ASR(B)	Arithmetic shift right	0062DD	1	1/0	-	A-13
SWAB	Swap bytes	0003DD	1	1/0	-	A-13
Multiple Precision						
ADC(B)	Add carry	0055DD	1	1/0	-	A-13
SBC(B)	Subtract carry	0056DD	1	1/0	-	A-13
SXT	Sign extend	0067DD	1	1/0	-	A-12
Multiprocessing						
TSTSET	Test and set (low bit interlocked)	0072DD	5	1/1	-	A-13
WRTLCK	Write interlocked	0073DD	4	1/1	-	A-13

Table A-2 Double Operand Instructions

Mnemonic	Instruction	Op Code Listing	Timing			
			Execution MC	R/W	Source Table	Dest. Table
MOV(B)	Move	01SSDD	1	1/0	A-9	A-13
CMP(B)	Compare	02SSDD	1	1/0	A-9	A-13
ADD	Add	06SSDD	1	1/0	A-9	A-13
SUB	Subtract	16SSDD	1	1/0	A-9	A-13
Logical						
BIT(B)	Bit test (AND)	03SSDD	1	1/0	A-9	A-11
BIC(B)	Bit clear	04SSDD	1	1/0	A-9	A-13
BIS(B)	Bit set (OR)	05SSDD	1	1/0	A-9	A-13
Register						
MUL	Multiply	0704SS	22 (Notes 5, 11)	1/0	-	A-10
DIV	Divide	071RSS	34 (Notes 6, 7, 12)	1/0	-	A-10
ASH	Shift automatically	072RSS	4	1/0	-	A-10
ASHC	Arithmetic shift combined	073RSS	5 (Note 13)	1/0	-	A-10
XOR	Exclusive (OR)	074RDD	1	1/0	-	A-10

Table A-3 Branch Instructions

Mnemonic	Instruction	Branch Op Code Listing	Timing: Branch Not Taken MC	Timing: Branch Not Taken R/W	Timing: Branch Taken MC	Timing: Branch Taken R/W
BR	Branch (unconditional)	000400	2	1/0	4	2/0
BNE	Br if not equal (to 0)	001000	2	1/0	4	2/0
BEQ	Br if equal (to 0)	001400	2	1/0	4	2/0
BPL	Br if plus	100000	2	1/0	4	2/0
BMI	Br if minus	100400	2	1/0	4	2/0
BVC	Br if overflow is clear	102000	2	1/0	4	2/0
BVS	Br if overflow is set	102400	2	1/0	4	2/0
BCC	Br if carry is clear	103000	2	1/0	4	2/0
BCS	Br if carry is set	103400	2	1/0	4	2/0
Signed Conditional Branches						
BGE	Br if greater or equal (to 0)	020000	2	1/0	4	2/0
BLT	Br if less than (0)	002400	2	1/0	4	2/0
BGT	Br if greater than (0)	003000	2	1/0	4	2/0
BLE	Br if less or equal (to 0)	003400	2	1/0	4	2/0

Table A-3 (Cont.) Branch Instructions

Unsigned Conditional Branches						
BHI	Br if higher	101000	2	1/0	4	2/0
BLOS	Br if lower or same	101400	2	1/0	4	2/0
BHIS	Br if higher or same	103000	2	1/0	4	2/0
BLO	Br if lower	103400	2	1/0	4	2/0
SOB	Subtract 1 and branch (if not equal to 0)	077RNN	3	1/0	5	2/0

Table A-4 Jump and Subroutine

			Timing		
Mnemonic	Instruction	Op Code Listing	Execution MC	R/W	Dest. Table
JMP	Jump	0001DD	-	-	A-15
JSR	Jump to subroutine	004RDD	-	-	A-15 (Note 4)
RTS	Return from subroutine	00020R	5	3/0	- (Note 14)
MARK	Stack cleanup	0064NN	10	3/0	

Table A–5 Trap and Interrupt Instructions

Mnemonic	Instruction	Op Code Listing	Timing Execution MC	Timing R/W
EMT	Emulator trap	104000-104377	20	4/2
TRAP	Trap	104400-104777	20	4/2
BPT	Breakpoint trap	000003	20	4/2
IOT	Input/output trap	000004	20	4/2
RTI	Return from interrupt	000002	9	4/0
RTT	Return from interrupt	000006	9	4/0

Table A–6 Condition Code Operators

Mnemonic	Instruction	Op Code Listing	Timing Execution MC	Timing R/W
CLC	Clear C	000241	3	1/0
CLV	Clear V	000242	3	1/0
CLZ	Clear Z	000244	3	1/0
CLN	Clear N	000250	3	1/0
CCC	Clear all CC bits	000257	3	1/0
SEC	Set C	000261	3	1/0
SEV	Set V	000262	3	1/0
SEZ	Set Z	000264	3	1/0
SEN	Set N	000270	3	1/0
SCC	Set all C bits	000277	3	1/0

Table A-7 Miscellaneous Instructions

Mnemonic	Instruction	Op Code Listing	Timing		
			Execution MC	R/W	Dest. Table
HALT	Halt	000000	-		-
WAIT	Wait for interrupt	000001	-		-
RESET	Reset external bus	000005	-		-
NOP	(No operation)	000240	3	1/0	-
SPL	Set priority level to N		7	1/0	-
MFPI	Move from previous instr space	00023N	5	1/1	A-10
MTPI	Move to previous instr space	0056DD	3	2/0	A-12
MFPD	Move from previous data space	1065SS	5	1/1	A-10
MTPD	Move to previous data space	1066DD	3	2/0	A-12
MTPS	Move byte to PSW PS	1064SS	8	1/0	A-10
MFPS	Move byte from PSW PS	1067DD	1	1/0	A-12
MFPT	Move from processor	000007	2	1/0	-
CSM	Call to supervisor mode	0070DD	28	3/3	A-10

Table A–8 Floating-Point Instructions

Mnemonic	Instruction	Op Code Listing	Timing			
			Min	Execution MC Non-Mode 0 Typical	Max	Table
ABSD	Make absolute	1706 fdst	23		24	A-18
ABSF	Make absolute	1706 fdst	19		20	A-18
ADDD	Add	172 (AC) fsvc	41	48	119	A-16
ADDF	Add	172 (AC) fsvc	31	35	102	A-16
CFCC	Copy Floating Condition Codes	170000	5		5	-
CLRD	Clear	1704 fdst	14		14	A-17
CLRF	Clear	1704 fdst	12		12	A-17
CMPD	Compare	173 (AC + 4)	24		25	A-17
CMPF	Compare	173 (AC + 4)	18		19	A-16
DIVD	Divide	174 (AC + 4)	160		167	A-16
DIVF	Divide	174 (AC + 4)	59		63	A-16
LDCDF	Ld & C from D to F	177 (AC + 4)	24		26	A-16
LDCFD	Ld & C from F to D	177 (AC + 4)	20		21	A-16
LDCID	Ld & C Integer to D	177 (AC) src	31		42	A-19
LDCIF	Ld & C Integer to F	177 (AC) src	26		36	A-19
LDCLD	Ld & C Long Integer to D	177 (AC) src	31		42	A-19
LDCLF	Ld & C Long Integer to F	177 (AC) src	26		44	A-19
LDD	Load	172 (AC + 4)	16		17	A-16

Table A-8 (Cont.) Floating-Point Instructions

Mnemonic	Instruction	Op Code Listing	Timing Min	Execution MC Non-Mode 0 Typical	Max	Table
LDEXP	Load Exponent	176 (AC + 4)	17		18	A-19
LDF	Load	172 (AC + 4)	12		13	A-19
LDFPS	Load FPP Program Status	1701 src	6		6	A-19
MODD	Multiply and Separate	171 (AC + 4)	202	217	268	A-16
MODF	Integer and Fraction	171 (AC + 4)	82	94	115	A-16
MULD	Multiply	171 (AC) fsrc	165		173	A-16
MULF	Multiply	171 (AC) fsrc	56		61	A-16
NEGD	Negate	1707 fdst	22		23	A-18
NEGE	Negate	1707 fdst	18		19	A-18
SETD	Set Floating Double Mode	170011	6		6	-
SETF	Set Floating Mode	170001	6		6	-
SETI	Set Integer Mode	170002	6		6	-
SETL	Set Long Integer Mode	170012	6		6	-
STCDF	St & C from D to F	176 (AC) fdst	17		20	A-17
STCDI	St & C from D to Integer	176 (AC) fdst	26		38	A-20
STCDL	St & C from D to Long Integer	176 (AC) fdst	26		54	A-20
STCFD	St & C from F to D	176 (AC) fdst	19		20	A-17
STCFI	St & C from F to Integer	175 (AC + 4)	23		35	A-20

Table A-8 (Cont.) Floating-Point Instructions

Mnemonic	Instruction	Op Code Listing	Timing Min	Execution MC Non-Mode 0 Typical	Max	Table
STCFL	St & C from F to Long Integer	175 (AC + 4)	23		51	A-20
STD	Store	174 (AC) fdst	12		12	A-17
STEXP	Store Exponent	175 (AC) dst	16		16	A-20
STF	Store	174 (AC) fdst	8		8	A-17
STFPD	Store FPP Program Status	1702 dst	9		9	A-20
STST	Store FPP Status	1703 dst	7		7	A-20
SUBD	Subtract	173 (AC) fsrc	47	55	122	A-16
SUBF	Subtract	173 (AC) fsrc	37	41	104	A-16
TSTD	Test	1705 fdst	11		12	A-16
TSTF	Test	1705 fdst	9		10	A-16

Table A-9 Source Address Times: All Double Operand

Source Mode	Source Register	Microcode Cycles	Read Memory Cycles
0	0-7	0	0
1	0-7	2	1
2	0-6	2	1
2	7	1	1
3	0-6	4	2
3	7	3	2
4	0-6	3	1
4	7	6	2 (Note 1)
5	0-6	5	2
5	7	8	3 (Note 1)
6	0-7	4	2
7	0-7	6	3

Table A–10 Destination Address: Read-Only Single Operand

Destination Mode	Destination Register	Microcode Cycles	Read Memory Cycles
0	0-7	0	0
1	0-7	2	1
2	0-6	2	1
2	7	1	1
3	0-6	4	2
3	7	3	2
4	0-6	3	
4	7	7	2 (Note 2)
5	0-6	5	2
5	7	9	3 (Note 3)
6	0-7	4	2
7	0-7	6	3

Table A–11 Destination Address Times: Read-Only Double Operand

Destination Mode	Destination Register	Microcode Cycles	Read Memory Cycles
0	0-7	0	0
1	0-7	3	1
2	0-6	3	1
2	7	2	1
3	0-6	5	2
3	7	3	2
4	0-6	4	1
4	7	8	2 (Note 2)
5	0-6	6	2
5	7	10	3 (Note 3)
6	0-7	5	2
7	0-7	7	3

Table A-12 Destination Address Times: Write-Only

Destination Mode	Destination Register	Microcode Cycles	Memory Read	Cycles Write
0	0-6	0	0	0
0	7	5	1	0
1	0-6	2	0	1
1	7	6	1	1
2	0-6	2	0	1
2	7	6	1	1
3	0-6	4	1	1
3	7	3	1	1
4	0-6	3	0	1
4	7	7	1	1
5	0-6	5	1	1
5	7	9	2	1
6	0-7	4	1	1
7	0-7	6	2	1

Table A-13 Destination Address Times: Read Modify Write

Destination Mode	Destination Register	Microcode Cycles	Memory Read	Cycles Write
0	0-6	0	0	0
0	7	5	1	0
1	0-6	3	1	
1	7	7	2	1
2	0-6	3	1	1
2	7	7	2	1
3	0-6	5	2	1
3	7	4	2	1
4	0-6	4	1	1
4	7	8	2	1 (Note 2)
5	0-6	6	2	1
5	7	10	3	1 (Note 3)
6	0-7	5	2	1
7	0-7	7	3	1

Table A-14 Destination Address Times: JMP

Destination Mode	Destination Register	Microcode Cycles	Memory Read	Cycles Write
1	0-7	4	2	0
2	0-7	6	2	0
3	0-7	5	3	0
4	0-7	5	2	0
5	0-7	6	3	0
6	0-6	6	3	0
6	7	5	3	0
7	0-7	7	4	0

Table A-15 Destination Address Times: JSR

Destination Mode	Destination Register	Microcode Cycles	Memory Read	Cycles Write
1	0-7	9	2	1
2	0-7	10	2	1
3	0-6	10	3	1
3	7	9	3	1
4	0-7	10	2	1
5	0-7	11	3	1
6	0-6	10	3	1
6	7	9	3	1
7	0-7	12	4	1

Table A-16 Floating Source 1-7

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Single Precision				
1	0-7	3	2	0
2	0-6	3	2	0
2	7	1	1	0
3	0-6	4	3	0
3	7	3	3	0
4	0-7	4	2	0
5	0-7	5	3	0
6	0-7	4	3	0
7	0-7	6	4	0
Double Precision				
1	0-7	5	4	0
2	0-6	5	4	0
2	7	0 (Note 15)	1	0
3	0-6	6	5	0
3	7	5	5	0
4	0-7	6	4	0
5	0-7	7	5	0
6	0-7	6	5	0
7	0-7	8	6	0

Table A-17 Floating Destination Modes 1-7

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Single Precision				
1	0-7	3	0	2
2	0-6	3	0	2
2	7	1	0	1
3	0-6	4	1	2
3	7	3	1	2
4	0-7	4	0	2
5	0-7	5	1	2
6	0-7	4	1	2
7	0-7	6	2	2
Double Precision				
1	0-7	5	0	4
2	0-6	5	0	4
2	7	(-1) (Note 15)	0	1
3	0-6	6	1	4
3	7	5	1	4
4	0-7	6	0	4
5	0-7	7	1	4
6	0-7	6	1	4
7	0-7	8	2	4

Table A–18 Floating Read-Modify-Write Modes 1-7

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Single Precision				
1	0-7	5	2	2
2	0-6	5	2	2
2	7	1 (Note 15)	1	1
3	0-6	6	3	2
3	7	5	3	2
4	0-7	6	2	2
5	0-7	7	3	2
6	0-7	6	3	2
7	0-7	8	4	2
Double Precision				
1	0-7	9	4	4
2	0-6	9	4	4
2	7	(-2) (Note 15)	1	1
3	0-6	10	5	4
3	7	9	5	4
4	0-7	10	4	4
5	0-7	11	5	4
6	0-7	10	5	4
7	0-7	12	6	4

Table A-19 Integer Source Modes 1-7

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Integer				
1	0-7	2	1	0
2	0-6	2	1	0
2	7	0 (Note 15)	1	0
3	0-6	3	2	0
3	7	2	2	0
4	0-7	3	1	0
5	0-7	4	2	0
6	0-7	3	2	0
7	0-7	5	3	0
Long Integer				
1	0-7	4	2	0
2	0-6	4	2	0
2	7	0 (Note 15)	1	0
3	0-6	5	3	0
3	7	4	3	0
4	0-7	5	2	0
5	0-7	6	3	0
6	0-7	5	3	0
7	0-7	7	4	0

Table A-20 Integer Destination Modes 1-7

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Integer				
1	0-7	2	0	1
2	0-6	2	0	1
2	7	2	0	1
3	0-6	3	1	1
3	7	2	1	1
4	0-7	3	0	1
5	0-7	4	1	1
6	0-7	3	1	1
7	0-7	5	2	1
Long Integer				
1	0-7	4	0	2
2	0-6	4	0	2
2	7	2	0	1
3	0-6	5	1	2
3	7	4	1	2
4	0-7	5	0	2
5	0-7	6	1	2
6	0-7	5	1	2
7	0-7	7	2	2

A.2 Source and Destination Table Notes

1. Subtract 2 MC and 1 read if both source and destination modes autodecrement PC, or if write-only or read-modify-write mode 07 or 17 is used.
2. Read-only and read-modify-write destination mode 47 references actually perform 3 READ operations. For bookkeeping purposes, one of the reads is accounted for in the execute, fetch timing.
3. READ-ONLY and READ-MODIFY-WRITE destination mode 57 references actually perform 4 READ operations. For bookkeeping purposes, one of the READs is accounted for in the EXECUTE, FETCHING TIMING.
4. Subtract 1 MC if the link register is PC.
5. Add 1 MC if the source operand is negative.
6. Subtract 1 MC if the source mode is not 0.
   a. Add 1 MC if the quotient is even.
   b. Add 2 MC if overflow occurs.
   c. Add 5 MC and 1 read if the PC is used as a destination register, but only if source mode 47 or 57 is not used.
7. Add 1 MC per shift.

8. Add 1 MC if source operand <15:6> is not 0.
9. Subtract 1 MC if one shift only.
10. Add 4 MC and 1 read if the PC is used as a destination register, but only if source mode 47 or 57 is not used.
11. Divide by zero executes in 5 MC (see Note 6).
12. Timing for no shift. Add 1 MC if a left shift. (Notes 8, 9, 11 apply.) Add 2 MC for a right shift. (Notes 8, 10, 11 apply.)
13. Add 1 MC if a register other than R7 is used.
14. Mode 27 references only access single-word operands. The execution time listed has been compensated in order to computer the total execution time accurately.

Appendix D
Floating-Point Instruction Timing

Since the FPJ11 is a coprocessor operating in parallel with the J-ll chip set, the calculation of floating-point instruction times for J-11 systems (using the FPJ11 option) must take this parallel processing into account.

Part	Definition
FPJ11 cycle	Two clock periods (110 ns at 18 MHz)
J-11 nonstretched cycle	Two FPJ11 cycles (220 ns at 18 MHz)
J-11 read cycle	J-11 nonstretched cycle if cache hit. Dependent on read access time of system if cache miss. The minimum is two J-11 nonstretched cycles, after which the J-11 stretches in half-cycle increments until MCONT is asserted.
J-11 write cycle	Dependent on write access time of system (two J-11 cycles + half cycles until MCONT).
Instruction Decode	A decode/prefetch cycle followed by a MOV microinstruction that allows the FPJ11 to assert DMR prior to the start of the next microinstruction (INPR for REG mode). This time equals two nonstretched cycles if the prefetch is a cache hit; otherwise nonstretched plus one read cycle.
Address Calculation Time	J-11 time required to calculate the address of the operand. This time is dependent on the addressing mode of the instruction, the frequency of the system clock, and whether any indirect data required is present in the cache (see Table D-1).
Argument Transfer Time	J-11 time required to load or store floating-point operands. This time is one nonstretched cycle (address relocation μcycle) plus one read cycle per 16-bit word read from memory for load class instructions, or one nonstretched plus one write cycle per 16-bit word to memory for store class instructions.
INPR (FEATEMP,TEMP)	J-11 support code microinstruction execute for all FPJ11 instructions. Moves the PC of the previous FPJ11 instruction to a TEMP register in case that instruction resulted in a floating-point exception. If the FPJ11 is still executing the previous instruction when the J-11 reaches its INPR microinstruction, the FPJ11 asserts STALL causing the J-11 INPR μcycle to stretch. The J-11 then waits for the FPJ11 to deassert STALL, signaling the system interface to assert MCONT before executing the next microinstruction (OUTR).

Part	Definition
WAIT	J-11 time waiting for the completion by the FPJ11 of the previous FP instruction. For load class or REG mode instructions, the time from when the J-11 INPR cycle stretches at the trailing edge of male until the FPJ11 deasserts STALL. This time equals zero if a stall was not required or if the FPJ11 deasserted the stall signal after the INPR cycle began but prior to the trailing edge of male. Although the wait time for the latter case is zero, RESYNC time is required. For store class instructions the wait time equals the time between the assertion of SCTL (that is, when the system interface is ready to execute the first write cycle of an FP store) and the assertion of FPA-RDY (data ready) by the FPJ11.
RESYNC	For load class and REG mode instructions the time required to continue a stretched INPR. This is the time for the system interface to recognize the deassertion of STALL and assert MCONT, plus the time required for the J-11 to synchronize MCONT and advance to the next microinstruction. Store class instructions normally do not have RSYNC time since the J-11 is waiting in a stretched write cycle and the continuation time is part write cycle. However, if the FPJ11 is executing a previous MODF/D or DIVD, the FPJ11 will assert STALL in order to stretch a non-I/O cycle prior to the first bus write. This allows the system interface to service DMA, thus limiting the worst case DMA latency when waiting for FPJ11 output. In this case, a wait and RESYNC time associated with the stretched non-I/O cycle is added to the effective execution time of the store class instruction.
OUTR (PC,FEATEMP), TESTPLA FPE	Last J-11 support microinstruction unless there is an FPE from the previous FP instruction. Saves address of PC in FEATEMP.
PRDC SYNC	Time required by FPJ11 to decode FP instruction and begin execution after receiving PRDC. This time equals two or three FPJ11 cycles depending upon synchronization. PRDC SYNC is not added to FPJ11 instruction execution times when the FPJ11 is executing a previous FP instruction at the assertion of PDRC.
Floating-Point Execution Time	Time required by FPJ11 to complete an FP instruction once it has received all arguments. For store class instructions, floating-point execution time includes the time from the start of the instruction until the FPJ11 asserts FPA-RDY, indicating the first 16-bit word is available for output (see Table D-2).
Effective Execution Time	Total J-11 time required to execute an FP instruction.
Load class	Instruction Decode + Address Calculation + Argument Transfer + INPR + WAIT + RSYNC + OUTR
REG mode	Instruction Decode + INPR + WAIT + RSYNC + OUTR
Store class	Instruction Decode + Address Calculation + INPR + Argument Transfer + WAIT + OUTR

Table D-1 shows address calculation times. Table D-2 shows floating-point instruction times.

Table D-1 Address Calculation Times

Mode	Load Class	Store Class
0	0	0
1	3	3
2	3	2
3	3 + RD[1]	2 + RD
4	4	4
5	3 + RD	3 + RD
6	3 + RDI[2]	2 + RDI
7	3 + RDI + RD	3 + RDI + RD
27	2	2
37	2 + RDI	1 + RDI
67	3 + RDI	2 + RDI
77	4 + RDI + RD	4 + RDI + RD

[1]RD = J-11 Read Cycle

[2]RDI = J-11 Istream Request

Table D–2 FPJ11 Instruction Times

Instruction	Min Cycles	Typ Cycles	Max Cycles	Stretch Cycles[1]	18 MHz[2] Typ(μs)
ADDF/SUBF	7	9	19	5	1.0
ADDD/SUBD	7	9	30	5	1.0
MULF	15	15	16	11	1.7
MULD	26	26	27	22	2.9
DIVF	17	24	30	25	2.7
DIVD	33	48	62	57	5.4
MODF	28	34	43	15	3.7
MODD	39	45	71	26	5.0
CMPF/D	3	4	6	2	0.4
LDF/D	3	3	3	0	0.3
LDEXP	2	3	2	0	0.2
LDCIF/D	10	10	10	3	1.1
LDCLF/D	10	10	10	3	1.1
LDCFD	4	4	4	1	0.4
LDCDF	4	4	8	1	0.4
STF/D	3	3	3	0	0.3
STCFI	8	10	13	1	1.1
STCFL	8	12	16	1	1.3
STCFD	4	4	4	0	0.4
STCDF	6	6	6	1	0.7
STEXP	5	5	5	0	0.6
TSTF/D, LDFPS STFPS, CFCC, SET	3	3	3	0	0.3
ABSF/D, NEGF/D	4	4	5	0	0.4

[1]Stretch cycles indicate the number of cycles out of max cycles that a data dependent stretch of one additional cycle could occur with probability less than 1 percent for each additional cycle.

[2]18 MHz = 111 ns Cycle

Load class instructions require input data and deposit results to the destination FP accumulator. REG mode instructions are FP accumulator to FP accumulator.

Execution of a load class FP instruction by the FPJ11 occurs in parallel with J-11 operation and can be overlapped as shown in the following flow.

J-11	FPJ11
Load class instruction is prefetched. This occurs during previous instruction execution	
Instruction Decode Prefetch next instruction	PRDC SYNC
Address Calculation	
Argument Transfer	FPJ11 loads operands
INPR	FPJ11 execution starts
WAIT if any	
RSYNC if any	
OUTR	
Decode next instruction	
	FPJ11 only stalls if next instruction is FP and REG mode. The FPJ11 can overlap the loading of operands for subsequent load class instructions.
	FPJ11 execution unit done

Store class instructions can be overlapped by the J-11 as the FPJ11 will complete a previously started load class or REG mode instruction and then continue to the store instruction. Execution of the store class instruction must be completed before the result can be stored in memory, thus eliminating further parallel processing for store class FP instructions. See the following flow.

J-11	FPJ11
Store class instruction is prefetched. This occurs during previous instruction execution	
Instruction Decode Prefetch next instruction	PRDC SYNC
Address Calculation	FPJ11 starts execution
INPR Argument Transfer	FPJ11 places operands in output buffer and sets FPA_RDY
J-11 waits during first write if FPA-RDY not asserted	
J-11 completes argument transfer	
OUTR	
Decode next instruction	