MI R-FSC

NEP

FAIRCHILD
A Schlumberger Company

S-143

001193

Orig

T1193

FSC

F9450 High-Performance 16-Bit Bipolar Microprocessor

Preliminary Data Sheet – November 1985

Microprocessor Product

DESCRIPTION

The Fairchild F9450 microprocessor is the nucleus of a family of high-performance devices intended for commercial and military applications requiring sophisticated, high-speed, real-time processing. It has, on-chip, all of the functions necessary to perform floating-point operations without the use of a coprocessor. Other on-chip capabilities allow addressing of up to 2M words of memory and, with the addition of the optional F9451 Memory Management Unit (MMU), up to 16M words of memory.

Real-time processing is achieved through advanced architecture that incorporates two programmable timers, user-accessible general-purpose registers, a complete 16-level interrupt processor, and a comprehensive fault handler on the chip. Multiprocessing is supported by a flexible bus arbitration scheme, as well as process synchronization (test and set) instructions.

The F9450 instruction set is optimized for complex real-time applications. It implements the complete MIL-STD-1750A instruction set architecture (ISA) and its floating-point standard on a single chip.

The F9450 family of support circuits and systems provides additional capabilities, including memory-mapped expansion with the F9451 MMU and write protection with the F9452 Block Protect Unit (BPU). Additional support circuits are being developed.

Comprehensive software support for the F9450, including assemblers, loaders, simulators, and compilers, is provided by Fairchild and other sources. Software development for the F9450 can be performed using the VAX-11/7XX™ computers using the VMS™ operating system.

For complete information on available support circuits and software, contact your local Fairchild Sales Office or the Microcontroller Division.

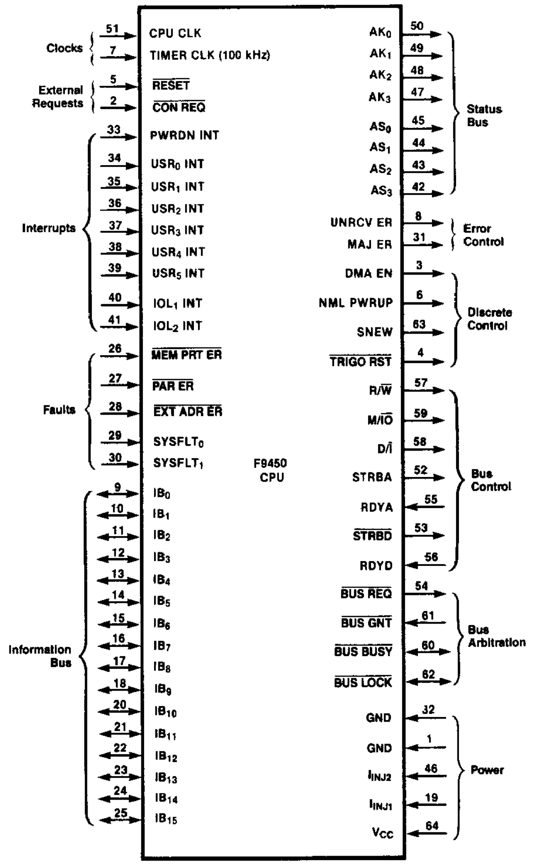
Features

- **Single-Chip 16-Bit Microprocessor with 32- and 48-Bit Floating-Point Arithmetic On-Chip**
- **Fast Operation – 0.25 μs Add, 1.85 μs Integer Multiply, Capable of Greater than 1.5 MIPS Throughput**
- **Real-Time Processing: Two Programmable Timers, 16 Levels of Vectored Interrupt**
- **Address Space of Up to 2M Words, Expandable to 16M Words with Optional F9451**
- **Instruction Set Optimized for Real-Time Applications (MIL-STD-1750A ISA)**
- **Built-In Self-Test, Fault Handling, and Abort**
- **Twenty-Four User-Accessible Registers**
- **Built-In Multiprocessor Capabilities**
- **Single- and Double-Precision Integer Arithmetic**
- **Built-In Console Operations**
- **Complete High-Level Language and Design Development Support Available**
- **Static Operation with Single Clock: 0-20 MHz**
- **TTL Inputs and Outputs with 8 mA Drive Capability**
- **Small Size 64-Pin DIP or Optional Surface-Mount Packages**
- **Full Performance over –55°C to +125°C Operating Temperature Range**
- **Bipolar I³L® Technology**

® I³L is a registered trademark of Fairchild Camera and Instrument Corporation.
™ VAX and VMS are trademarks of Digital Equipment Corporation.

Signal Functions

11562



Contents on Back Cover

Fairchild Camera and Instrument Corporation, Microcontroller Division
(After Dec. 1, 1985: Fairchild Semiconductor Corporation)
450 National Avenue, Mt. View, CA 94043 (415) 962-3888 TWX: 910-379-5091

© 1985 Fairchild Printed in U.S.A. 4027-018-23-115 5M 501973
November 1985 Fairchild reserves the right to make changes in the circuitry or specifications at any time without notice.

F9450 SYSTEM ARCHITECTURE

The F9450 instruction set has been designed for demanding real-time applications. Such operations as single- and double-precision integer arithmetic, including multiply and divide, floating-point arithmetic, bit operations, fault and error handling, interrupt processing, and direct memory access (DMA) are available. Memory management and protection can be accomplished using the F9451 MMU and F9452 BPU peripherals. Console commands permit direct interaction with the user.

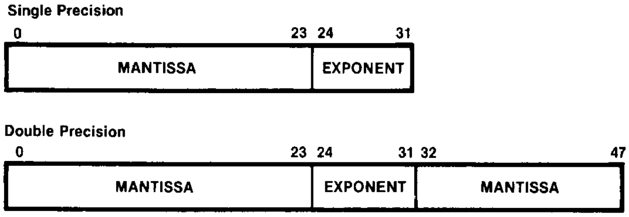
Data Types

The F9450 processes six data types:

- — Bits
- — Bytes (8 bits)
- — Words (16 bits)
- — Double words (32 bits)
- — Single-precision floating-point numbers (32 bits)
- — Extended-precision floating point numbers (48 bits)

The floating-point numbers are represented by a fractional-twos-complement mantissa (24 bits for single-precision and 40 bits for extended-precision) and an 8-bit twos-complement exponent, as shown in figure 1. These floating-point data formats are specified by MIL-STD-1750A.

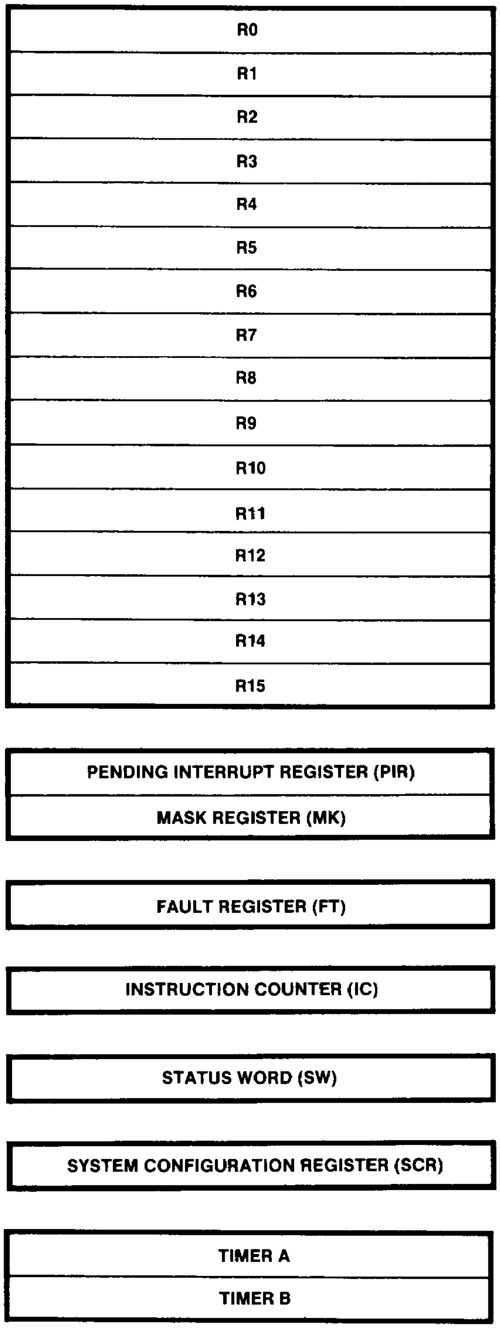
Figure 1 F9450 Floating-Point Formats



Register Set

The F9450 contains 24 user-accessible registers. A model of these registers is shown in figure 2. There are 16 program-accessible general-purpose registers (R0 to R15), a Pending

Figure 2 F9450 Programmer's Register Model



* The SCR is 5 bits only, and is not available to the programmer.

Interrupt Register (PIR), Mask Register (MK), Fault Register (FT), Instruction Counter (IC), Status Word Register (SW), System Configuration Register (SCR), and two Timers (A and B). The PIR, MK, and FT are described in the "Interrupts" section. The 16-bit IC contains the address of the instruction currently being executed.

The F9450 register set also includes six temporary registers, A1, A2, D01, D02, Q1, and Q2, that are internal to the operation of the F9450 and accessible only from the console. These, therefore, are not included in the programmer's model.

Status Word Register (SW)

The Status Word Register is 16 bits wide, defined as:

0 3	4 7	8 11	12 15
C P Z N	RESERVED	AK (PS)	AS

C . . . . Carry
P . . . . Positive
Z . . . . Zero
N . . . . Negative

Results of the most recent arithmetic operation.

Reserved . . . These bits will be 0

AK(PS) . . . . . Access Key/Processor State bits serve two functions:

1) Determine the legal/illegal criteria for privileged instructions. A privileged instruction is executed with PS = 0 only. An attempt to execute a privileged instruction with PS ≠ 0 causes a Major Error, sets bit 10 in the Fault Register, and causes an instruction abort.
2) Define the Access Key that is used in systems with an MMU to match an Access Lock.

AS . . . . Address State defines a page register group in the MMU. For configurations without an MMU, an Address State fault is generated (bit 11 in the Fault Register is set) for any operations attempting to modify the AS field to a nonzero value. Note that this condition is also tested during interrupt processing (i.e., an interrupt service status word with AS ≠ 0 will be aborted).

All usable bits of the status word can be modified under program control or from the console (for details, refer to the F9451 MMU data sheet).

System Configuration Register (SCR)

The SCR defines to the F9450 control circuitry the configuration of the external system connected to the CPU.

The 5-bit-wide SCR is defined as:

0	1	2	3	4	5 15
M	B	C	P	I	NOT USED

M . . . . MMU Present = 1 if an MMU is connected in the system. This must be set if the application ever requires AS ≠ 0.

B . . . . BPU Present = 1 if a BPU is connected in the system.

C . . . . Console Present = 1 if a console is connected in the system.

P . . . . Coprocessor = 1 if a coprocessor is connected in the system. If this bit is not set, the BIF instruction (see the "Built-In Function Implementation" section) will be considered an illegal instruction.

I . . . . Interrupt Mode = Selects the interrupt mode for the PWRDN INT and USR_0 INT–USR_5 INT signals: 1 is level-sensitive, 0 is edge-sensitive (low to high).

To load the SCR, external hardware is required to respond to an I/O Read request at I/O address 8410 (Hex). The internal SCR is not program-accessible, but is automatically loaded from the external hardware via the I/O Read from 8410, which is initiated by the F9450 during either the Reset initialization or the execution of the breakpoint instruction (BPT).

Timer A and Timer B

The two 16-bit-wide timers are started, halted, loaded, and read under software control. Timer A is controlled by the timer clock (100 kHz) and Timer B is controlled by the timer clock divided by 10.

When Timer A and Timer B reach their terminal counts, they set the corresponding bits in the PIR. Both are halted when the CPU is in the console mode and continue on the resumption of program execution.

Instruction Set

The instruction set of the F9450 is optimized for real-time applications in accordance with the MIL-STD-1750A ISA. Table 1 shows addressing modes and related instruction word formats, Derived Address (DA), and Derived Operand (DO). Not all instructions use all the addressing modes; therefore, acceptable addressing modes should be confirmed for each instruction. Table 2 provides the instruction set with applicable addressing modes and table 3 gives the dedicated I/O addresses.

MIL-STD-1750A Description

For a complete description of the instruction set, refer to MIL-STD-1750A ISA or to the F9450 User Guide. The military standard is available from the Department of the Navy, Naval Publications and Forms Center, 5801 Tabor Ave., Philadelphia, PA 19120, telephone (603) 121-3202. The F9450 User Guide will be available from Fairchild sales offices and from the Microprocessor Division.

Addressing Modes

There are ten addressing modes. The smallest addressable memory word is 16 bits. Therefore, the 16-bit address field allows direct addressing of 64K words. There is no restriction on the location of double-word operands in memory.

1. **Register Direct (R)**
   The instruction-specified register contains the required operand. With the exception of this mode, the DA denotes a memory address.

2. **Memory Direct (D)**
   In this mode, the instruction contains the memory address of the operand.

3. **Memory Direct - Indexed (DX)**
   The memory address of the required operand is specified by the sum of the contents of an index register and the instruction address field. Registers R1 through R15 may be specified for indexing.

4. **Memory Indirect (I)**
   The instruction-specified memory address contains the address of the required operand.

5. **Memory Indirect with Pre-Indexing (IX)**
   The sum of the contents of a specified index register and the instruction address field specify the address of a memory location containing the address of the required operand. Registers R1 through R15 may be specified for pre-indexing.

6. **Immediate Long (IM)**
   One method of Immediate Long addressing allows indexing and one does not. The indexable form of immediate addressing is shown in table 3. If the specified index register, RX, is ≠ 0, the contents of RX is added to the immediate field to form the required operand; otherwise, the immediate field contains the required operand.

7. **Immediate Short (IS)**
   The required 4-bit operand is contained within the 16-bit instruction. One method of Immediate Short addressing interprets the contents of the immediate field as positive data and another method interprets tne contents of the immediate field as negative data.

   a. **Immediate Short Positive (ISP)**
      The immediate operand is treated as a positive integer between 1 and 16.

   b. **Immediate Short Negative (ISN)**
      The immediate operand is treated as a negative integer between 1 and 16. Its internal form is a twos-complement, sign-extended 16-bit number.

8. **Instruction Counter Relative (ICR)**
   This mode is used for 16-bit branch instructions. The contents of the instruction counter minus one (i.e., the address of the current instruction) are added to the sign-extended 8-bit displacement field of the instruction. The sum points to the memory address to which control may be transferred if a branch is executed. This mode allows addressing within a memory region of –128 to +127 words, relative to the address of the current instruction.

9. **Base Relative (B)**
   The contents of an instruction-specified base register are added to the 8-bit displacement field of the 16-bit instruction. The displacement field is taken to be a positive number between 0 and 255. The sum points to the memory address of the required operand. This mode allows addressing within a memory region of 256 words, beginning at the address pointed to by the base register.

10. **Base Relative - Indexed (BX)**
    The sum of the contents of a specified index register and a specified base register is the address of the required operand. Registers R1 through R15 may be specified for indexing.

Table 1 F9450 Addressing Modes and Instruction Formats

Mode	Format	Derived Operand (DO)		Derived Address (DA)	
		Single-Precision	Floating-Point and Double Precision	Single-Precision	Floating-Point and Double Precision
1. Register Direct	0 7 8 11 12 15 O.C. RA RB				
"R"		(RB)	(RB, RB + 1)	RB	RB, RB + 1
2. Memory Direct	0 7 8 11 12 15 16 31 O.C. RA RX A				
"D"	RX = 0 (Non-Indexed)	[A]	[A, A + 1]	A	A, A + 1
"DX"	RX ≠ 0 (Indexed)	[A + (RX)]	[A + (RX), A + 1 + (RX)]	A + (RX)	A + (RX), A + 1 + (RX)
3. Memory Indirect	0 7 8 11 12 15 16 31 O.C. RA RX A				
"I"	RX = 0 (Non-Indexed)	[|A|]	[|A|, |A| + 1]	[A]	[A], [A] + 1
"IX"	RX ≠ 0 (Indexed)	[|A + (RX)|]	[|A + (RX)|, |A + (RX)| + 1]	[A + (RX)]	[A + (RX)], [A + (RX)] + 1
4. Immediate Long					
a. Not Indexable	0 7 8 11 12 15 16 31 O.C. RA OCX I				
"IM"		I			
b. Indexable	0 7 8 11 12 15 16 31 O.C. RA RX I				
"IM"	RX = 0 (Non-Indexed)	I			
"IMX"	RX ≠ 0 (Indexed)	I + (RX)			
5. Immediate Short					
a. Positive	0 7 8 11 12 15 O.C. RA I				
"ISP"		+ (I + 1)			
b. Negative	0 7 8 11 12 15 O.C. RA I				
"ISN"		− (I + 1)			
6. IC Relative [1]	0 7 8 15 O.C. DU				
"ICR"				DU + (IC − 1)	
7. Base Relative[2]					
a. Not Indexable[3]	0 5 6 7 8 15 O.C. BR DU				
"B"	BR = BR + 12	[DU + (BR)]	[DU + (BR), DU + 1 + (BR)]	DU + (BR)	DU + (BR), DU + 1 + (BR)
b. Indexable	0 5 6 7 8 11 12 15 O.C. BR OCX RX				
"B"	RX = 0 (Non-Indexed)	[(BR)]	[(BR), (RR) + 1]	(BR)	(BR), (BR) + 1
"BX"	RX ≠ 0 (Indexed)	[(BR) + (RX)]	[(BR) + (RX), (BR) + 1 + (RX)]	(BR) + (RX)	(BR) + (RX), (BR) + 1 + (RX)

Notes:

1. $-128 \leq DU \leq 127$.
2. Base registers: BR = R12, R13, R14, and R15.
3. $0 \leq DU \leq 255$.
4. Extended-precision floating-point instructions require addressing of three operands located at DA, DA + 1, and DA + 2.

Table 2 shows execution time as a function of clock cycles for the instruction set of the F9450. This information is based on the microprogram for the "D" design of the F9450 and is subject to change for future design iterations.

Note that many instructions, including all floating point operations, are data-dependent. This information is given as a guide to the user and includes a great number of these data dependencies. Data dependencies of floating point operations are specifically not included in this table. Such error conditions as overflows or underflows are also not included in this table. Care should be taken to verify these times on a device with actual data for time-critical applications.

Table 2 F9450 Instruction Set

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		Integer Arithmetic/Logic				
Single-Precision Add	A	R	1	0	5	
	A	B	1	1	12	
	A	BX	1	1	12	
	A	ISP	1	0	8	
	A	D	2	1	13	
	A	DX	2	1	13	
	A	IM	2	0	12	
Double-Precision Add	DA	R	1	0	18	
	DA	D	2	1	24	
	DA	DX	2	1	24	
Increment Memory by	INCM	D	2	2	17	Nf/N6 Bus Lock
Positive Integer	INCM	DX	2	2	17	Nf/N6 Bus Lock
Single-Precision Absolute Value	ABS	R	1	0	5	Nf/N6, pos. #
	ABS	R	1	0	10	Nf/N6, neg. #
Double-Precision Absolute Value	DABS	R	1	0	13	Pos. #
	DABS	R	1	0	21	Neg. #
Single-Precision Subtract	S	R	1	0	5	
	S	B	1	1	12	
	S	BX	1	1	12	
	S	ISP	1	0	8	
	S	D	2	1	13	
	S	DX	2	1	13	
	S	IM	2	0	12	
Double-Precision Subtract	DS	R	1	0	18	
	DS	D	2	2	24	
	DS	DX	2	2	24	

Notes:

Nf: Number of Instruction Fetch cycles.

No: Number of Operand bus cycles (either Memory or I/O). I/O cycles are indicated with a note in the comments column. VIO instructions include only the execution of one operation.

Nc: Total number of clocks needed to perform the operation. Variable length instructions require the addition of the base time plus the proper number of increments.

Nf/N6: An Instruction Fetch occurs during a 5-clock-machine cycle, thus hiding 1 wait state in the data phase of the operation.

Bus Lock: Bus Lock is used during the execution of the instruction to retain control of the bus for 2 consecutive machine cycles.

Bus Lock (1): This instruction activates Bus Lock and keeps it active for 2 bus cycles with 1 ALU cycle in between.

Bus Lock (2): This instruction activates Bus Lock and keeps it active for a total of 4 machine cycles, including:
1 Memory Read bus cycle   1 Memory Read bus cycle
1 ALU cycle   1 Memory Write bus cycle

Table 2 F9450 Instruction Set (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		Integer Arithmetic/Logic (Cont.)				
Decrement Memory by Positive Integer	DECM	D	2	2	17	Nf/N6 Bus Lock
	DECM	DX	2	2	17	Nf/N6 Bus Lock
Single-Precision Negate	NEG	R	1	0	5	
Double-Precision Negate	DNEG	R	1	0	18	
Single-Precision Multiply – 16-Bit Product	MS	R	1	0	39	
	MS	ISP	1	0	42	
	MS	ISN	1	0	42	
	MS	D	2	1	47	
	MS	DX	2	1	47	
	MS	IM	2	0	46	
Single-Precision Multiply – 32-Bit Product	M	R	1	0	37	
	M	B	1	1	44	
	M	BX	1	1	44	
	M	D	2	1	45	
	M	DX	2	1	45	
	M	IM	2	0	44	
Double-Precision Multiply	DM	R	1	0	126	
	DM	D	2	2	132	
	DM	DX	2	2	132	
Single-Precision Divide – 16-Bit Dividend	DV	R	1	0	98	Nf/N6
	DV	ISP	1	0	98	Nf/N6
	DV	ISN	1	0	98	Nf/N6
	DV	D	2	1	103	Nf/N6
	DV	DX	2	1	103	Nf/N6
	DV	IM	2	0	102	Nf/N6
			0	0	3	Add for neg. dividend
			0	0	3	Add for neg. divisor
			0	0	3	Add for neg. quotient
			0	0	3	Add for neg. remainder
			0	0	3	Add for remainder correction
Single-Precision Divide – 32-Bit Dividend	D	R	1	0	97	Nf/N6
	D	B	1	1	101	Nf/N6
	D	BX	1	1	101	Nf/N6
	D	D	2	1	102	Nf/N6
	D	DX	2	1	102	Nf/N6
	D	IM	2	0	101	Nf/N6
			0	0	6	Add for neg. dividend
			0	0	5	Add for neg. divisor
			0	0	3	Add for neg. quotient
			0	0	3	Add for neg. remainder
			0	0	3	Add for remainder correction
Double-Precision Divide	DD	R	1	0	239	
	DD	D	2	2	245	
	DD	DX	2	2	245	
			0	0	6	Add for neg. dividend
			0	0	6	Add for neg. divisor
			0	0	3	Add for neg. quotient

Table 2 F9450 Instruction Set (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
Integer Arithmetic/Logic (Cont.)						
Single-Precision Compare	C	R	1	0	8	
	C	B	1	1	15	
	C	BX	1	1	15	
	C	ISP	1	0	11	
	C	ISN	1	0	11	
	C	D	2	1	16	
	C	DX	2	1	16	
	C	IM	2	0	15	
Compare Between Limits	CBL	D	2	2	30	Nf/N6 (RA<DO1)
			2	2	43	Nf/N6 (DO1<RA<DO2)
			2	2	40	Nf/N6 (RA<DO2)
	CBL	DX	2	2	30	Nf/N6 (RA<DO1)
			2	2	43	Nf/N6 (DO1<RA<DO2)
			2	2	40	Nf/N6 (RA<DO2)
Double-Precision Compare	DC	R	1	0	21	
	DC	D	2	2	27	
	DC	DX	2	2	27	
Inclusive-OR	OR	R	1	0	4	
	OR	B	1	1	11	
	OR	BX	1	1	11	
	OR	D	2	1	12	
	OR	DX	2	1	12	
	OR	IM	2	0	11	
AND	AND	R	1	0	4	
	AND	B	1	1	11	
	AND	BX	1	1	11	
	AND	D	2	1	12	
	AND	DX	2	1	12	
	AND	IM	2	0	11	
Exclusive-OR	XOR	R	1	0	4	
	XOR	D	2	1	12	
	XOR	DX	2	1	12	
	XOR	IM	2	0	11	
NAND	NAND	R	1	0	7	
	NAND	D	2	1	15	
	NAND	DX	2	1	15	
	NAND	IM	2	0	14	
Floating Point						
Floating-Point Add	FA	R	1	0	62	No shifts in exponent adjust and in normalization
	FA	B	1	2	67	
	FA	BX	1	2	67	
	FA	D	2	2	68	
	FA	DX	2	2	68	

Table 2 F9450 Instruction Set (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		Floating-Point (Cont.)				
Extended-Precision Floating-Point Add	EFA	R	1	0	71	
	EFA	D	2	3	78	
	EFA	DX	2	3	78	
Floating-Point Absolute Value	FABS	R	1	0	16	Pos. #
	FABS	R	1	0	62	Neg. #
Floating-Point Subtract	FS	R	1	0	62	
	FS	B	1	2	67	
	FS	BX	1	2	67	
	FS	D	2	2	68	
	FS	DX	2	2	68	
Extended-Precision Floating-Point Subtract	EFS	R	1	0	71	
	EFS	D	2	3	78	
	EFS	DX	2	3	78	
Floating-Point Negate	FNEG	R	1	0	56	
Floating-Point Multiply	FM	R	1	0	120	
	FM	B	1	2	125	
	FM	BX	1	2	125	
	FM	D	2	2	126	
	FM	DX	2	2	126	
Extended-Precision Floating-Point Multiply	EFM	R	1	0	251	
	EFM	D	2	3	258	
	EFM	DX	2	3	258	
Floating-Point Divide	FD	R	1	0	234	
	FD	B	1	2	239	
	FD	BX	1	2	239	
	FD	D	2	2	240	
	FD	DX	2	2	240	
Extended-Precision Floating-Point Divide	EFD	R	1	0	480	
	EFD	D	2	3	487	
	EFD	DX	2	3	487	
Floating-Point Compare	FC	R	1	0	52	Nf/N6
	FC	B	1	2	57	Nf/N6
	FC	BX	1	2	57	Nf/N6
	FC	D	2	2	58	Nf/N6
	FC	DX	2	2	58	Nf/N6
Extended-Precision Floating-Point Compare	EFC	R	1	0	52	Nf/N6
	EFC	D	2	3	65	Nf/N6
	EFC	DX	2	3	65	Nf/N6
Convert Floating-Point to 16-Bit Integer	FIX	R	1	0	10	Nf/N6
Convert 16-Bit Integer to Floating-Point	FLT	R	1	0	16	Nf/N6
Convert Extended-Precision Floating-Point to 32-Bit Integer	EFIX	R	1	0	21	

Table 2 **F9450 Instruction Set** (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		Floating-Point (Cont.)				
Convert 32-Bit Integer to Extended-Precision Floating-Point	EFLT	R	1	0	25	Nf/N6
		Bit Operations				
Set Bit	SB	R	1	0	7	
	SB	D	2	2	16	Bus Lock
	SB	DX	2	2	16	Bus Lock
	SB	I	2	3	20	Bus Lock
	SB	IX	2	3	20	Bus Lock
Reset Bit	RB	R	2	0	7	
	RB	D	2	2	16	Bus Lock
	RB	DX	2	2	16	Bus Lock
	RB	I	2	3	20	Bus Lock
	RB	IX	2	3	20	Bus Lock
Test Bit	TB	R	2	0	7	
	TB	D	2	1	15	
	TB	DX	2	1	15	
	TB	I	2	2	19	
	TB	IX	2	2	19	
Test and Set Bit	TSB	D	2	3	23	Bus Lock (2)
	TSB	DX	2	3	23	Bus Lock (2)
Set Variable Bit in Register	SVBR	R	1	0	7	
Reset Variable Bit in Register	RVBR	R	1	0	7	
Test Variable Bit in Register	TVBR	R	1	0	7	
		Shift				
Shift Left Logical	SLL	R	1	0	7	One shift
			0	0	3	Incremental
Shift Right Logical	SRL	R	1	0	7	One shift
			0	0	3	Incremental
Shift Right Arithmetic	SRA	R	1	0	7	One shift
			0	0	3	Incremental
Shift Left Cyclic	SLC	R	1	0	7	One shift
			0	0	3	Incremental
Double Shift Left Logical	DSLL	R	1	0	16	One shift
			0	0	6	Incremental
Double Shift Right Logical	DSRL	R	1	0	16	One shift
			0	0	6	Incremental
Double Shift Right Arithmetic	DSRA	R	1	0	16	One shift
			0	0	6	Incremental
Double Shift Left Cyclic	DSLC	R	1	0	19	One shift
			0	0	9	Incremental

Table 2 **F9450 Instruction Set** (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
Shift (Cont.)						
Shift Logical, Count in Register	SLR	R	1	0	11	Nf/N6, no shift
			1	0	21	Nf/N6, right
			0	0	3	Incremental
	SLR	R	1	0	38	Nf/N6, left
			0	0	5	Incremental
Shift Arithmetic, Count in Register	SAR	R	1	0	11	Nf/N6, no shift
			1	0	21	Nf/N6, right
			0	0	3	Incremental
	SAR	R	1	0	29	Nf/N6, left
			0	0	5	Incremental
Shift Cyclic, Count in Register	SCR	R	1	0	11	Nf/N6, no shift
			1	0	21	Nf/N6, right
			0	0	3	Incremental
	SCR	R	1	0	24	Nf/N6, left
			0	0	3	Incremental
Double Shift Logical, Count in Register	DSLR	R	1	0	11	Nf/N6, no shift
			1	0	30	Nf/N6, right
			0	0	6	Incremental
	DSLR	R	1	0	44	Nf/N6, left
			0	0	8	Incremental
Double Shift Arithmetic, Count in Register	DSAR	R	1	0	11	Nf/N6, no shift
			1	0	30	Nf/N6, right
			0	0	6	Incremental
	DSAR	R	1	0	35	Nf/N6, left
			0	0	8	Incremental
Double Shift Cyclic, Count in Register	DSCR	R	1	0	11	Nf/N6, no shift
			1	0	33	Nf/N6, right
			0	0	9	Incremental
	DSCR	R	1	0	33	Nf/N6, left
			0	0	9	Incremental
Load/Store/Exchange						
Single-Precision Load	L	R	1	0	4	
	L	B	1	1	11	
	L	BX	1	1	11	
	L	ISP	1	0	7	
	L	ISN	1	0	7	
	L	D	2	1	12	
	L	DX	2	1	12	
	L	IM	2	0	11	
	L	IMX	2	0	14	
	L	I	2	2	16	
	L	IX	2	2	16	
Double-Precision Load	DL	R	1	0	16	
	DL	B	1	2	21	

Table 2 F9450 Instruction Set (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		Load/Store/Exchange (Cont.)				
Double-Precision Load (Cont.)	DL	BX	1	2	21	
	DL	D	2	2	22	
	DL	DX	2	2	22	
	DL	I	2	3	26	
	DL	IX	2	3	26	
Extended-Precision Floating-Point Load	EFL	D	2	3	26	
	EFL	DX	2	3	26	
Load from Upper Byte	LUB	D	2	1	15	
	LUB	DX	2	1	15	
	LUB	I	2	2	19	
	LUB	IX	2	2	19	
Load from Lower Byte	LLB	D	2	1	12	
	LLB	DX	2	1	12	
	LLB	I	2	2	16	
	LLB	IX	2	2	16	
Single-Precision Store	ST	B	1	1	11	
	ST	BX	1	1	11	
	ST	D	2	1	12	
	ST	DX	2	1	12	
	ST	I	2	2	16	
	ST	IX	2	2	16	
Store a Non-Negative Constant	STC	D	2	1	12	
	STC	DX	2	1	12	
	STC	I	2	2	16	
	STC	IX	2	2	16	
Double-Precision Store	DST	B	1	2	15	
	DST	BX	1	2	15	
	DST	D	2	2	16	
	DST	DX	2	2	16	
	DST	I	2	3	20	
	DST	IX	2	3	20	
Store Register through Mask	SRM	D	2	2	25	Bus Lock (1)
	SRM	DX	2	2	25	Bus Lock (1)
Extended-Precision Floating-Point Store	EFST	D	2	3	20	
	EFST	DX	2	3	20	
Store into Upper Byte	STUB	D	2	2	16	Bus Lock
	STUB	DX	2	2	16	Bus Lock
	STUB	I	2	3	20	Bus Lock
	STUB	IX	2	3	20	Bus Lock
Store into Lower Byte	STLB	D	2	2	16	Bus Lock
	STLB	DX	2	2	16	Bus Lock
	STLB	I	2	3	26	Bus Lock
	STLB	IX	2	3	20	Bus Lock
Exchange Bytes in Register	XBR	S	1	0	7	
Exchange Words in Registers	XWR	R	1	0	10	

Table 2 **F9450 Instruction Set** (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		Multiple Load/Store				
Push Multiple Registers onto the Stack	PSHM	S	1	1	16	One operation
			0	1	12	Incremental
Pop Multiple Registers off the Stack	POPM	S	1	1	20	Nf/N6
			0	1	16	Incremental
Load Multiple Registers	LM	D	2	1	16	Nf/N6
			0	1	8	Incremental
	LM	DX	2	1	16	Nf/N6
			0	1	8	Incremental
Store Multiple Registers	STM	D	2	1	17	
			0	1	9	Incremental
	STM	DX	2	1	17	
			0	1	9	Incremental
Move Multiple Words, Memory-to-Memory	MOV	S	1	0	9	No move
			1	2	37	One move
			0	2	13	Incremental
		Program Control				
Jump on Condition	JC	D	2	0	9	Nf/N6, no jump
	JC	D	3	0	17	Nf/N6, jump
	JC	DX	2	0	9	Nf/N6, no jump
	JC	DX	3	0	17	Nf/N6, jump
	JC	I	2	0	13	Nf/N6, no jump
	JC	I	3	0	21	Nf/N6, jump
	JC	IX	2	0	13	Nf/N6, no jump
	JC	IX	3	0	21	Nf/N6, jump
Jump to Subroutine	JS	D	3	0	12	
	JS	DX	3	0	12	
Subtract One and Jump	SOJ	D	2	0	13	No jump
	SOJ	D	3	0	17	Jump
	SOJ	DX	2	0	13	No jump
	SOJ	DX	3	0	17	Jump
Branch Unconditionally	BR	ICR	2	0	14	
Branch if Equal to (Zero)	BEZ	ICR	1	0	4	No branch
	BEZ	ICR	3	0	15	Branch
Branch if Less than (Zero)	BLT	ICR	1	0	4	No branch
	BLT	ICR	3	0	15	Branch
Branch if Less than or Equal to (Zero)	BLE	ICR	1	0	4	No branch
	BLE	ICR	3	0	15	Branch
Branch if Greater than (Zero)	BGT	ICR	1	0	4	No branch
	BGT	ICR	3	0	15	Branch
Branch if not Equal to (Zero)	BNZ	ICR	1	0	4	No branch
	BNZ	ICR	3	0	15	Branch

Table 2 F9450 Instruction Set (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
Program Control (Cont.)						
Branch if Greater than or Equal to (Zero)	BGE	ICR	1	0	4	No branch
	BGE	ICR	3	0	15	Branch
Branch to Executive	BEX	S	2	8	92	No MMU attached
			2	8	87	MMU attached
Load Status	LST	D	3	3	42	MMU attached
			3	3	47	No MMU attached
	LST	DX	3	3	42	MMU attached
			3	3	47	No MMU attached
	LST	I	3	4	46	MMU attached
			3	4	51	No MMU attached
	LST	IX	3	4	46	MMU attached
			3	4	51	No MMU attached
Stack IC and Jump to Subroutine	SJS	D	3	1	22	
	SJS	DX	3	1	22	
Unstack IC and Return from Subroutine	URS	S	3	1	15	
No Operation	NOP	S	1	0	9	
Breakpoint	BPT	S	1	1	27	No console (I/O)
Built-In Function (to Implement External Coprocessors)	BIF	D	3	2	34	(I/O)
	BIF	DX	3	2	34	(I/O)
	BIF	I	3	3	38	(2 I/O)
	BIF	IX	3	3	38	(2 I/O)
Programmed Input/Output						
Execute Input/Output	XIO	IM	2	1	31	PI case (I/O)
	XIO	IMX	2	1	34	PI case (I/O)
	XIO	IM	2	1	26	PO case (I/O)
	XIO	IMX	2	1	29	PO case (I/O)
Vectored Input/Output	VIO	DX	2	2	65	PI case (1 I/O – 1 operation)
	VIO	DX	2	2	72	PI case (1 I/O – 1 operation)
	VIO	D	2	2	66	PO case (1 I/O – 1 operation)
	VIO	DX	2	2	69	PO case (1 I/O – 1 operation)
Timer Control						
Timer A Start	TAS	IM	2	1	26	(I/O)
	TAS	IMX	2	1	29	(I/O)
	TAS	D	2	2	66	(1 I/O – 1 operation)
	TAS	DX	2	2	69	(1 I/O – 1 operation)
Timer A Halt	TAH	IM	2	1	26	(I/O)
	TAH	IMX	2	1	29	(I/O)
	TAH	D	2	2	66	(1 I/O – 1 operation)
	TAH	DX	2	2	69	(1 I/O – 1 operation)

Table 2 F9450 Instruction Set (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
Timer Control (Cont.)						
Output Timer A	OTA	IM	2	1	28	(I/O)
	OTA	IMX	2	1	31	(I/O)
	OTA	D	2	2	65	(1 I/O – 1 operation)
	OTA	DX	2	2	68	(1 I/O – 1 operation)
Input Timer A	ITA	IM	2	1	34	(I/O)
	ITA	IMX	2	1	37	(I/O)
	ITA	D	2	2	84	(1 I/O – 1 operation)
	ITA	DX	2	2	87	(1 I/O – 1 operation)
Timer B Start	TBS	IM	2	1	26	(I/O)
	TBS	IMX	2	1	29	(I/O)
	TBS	D	2	2	66	(1 I/O – 1 operation)
	TBS	DX	2	2	69	(1 I/O – 1 operation)
Timer B Halt	TBH	IM	2	1	26	(I/O)
	TBH	IMX	2	1	29	(I/O)
	TBH	D	2	2	66	(1 I/O – 1 operation)
	TBH	DX	2	2	69	(1 I/O – 1 operation)
Output Timer B	OTB	IM	2	1	28	(I/O)
	OTB	IMX	2	1	31	(I/O)
	OTB	D	2	2	68	(1 I/O – 1 operation)
	OTB	DX	2	2	71	(1 I/O – 1 operation)
Input Timer B	ITB	IM	2	1	34	(I/O)
	ITB	IMX	2	1	37	(I/O)
	ITB	D	2	2	84	(1 I/O – 1 operation)
	ITB	DX	2	2	87	(1 I/O – 1 operation)
Interrupt/DMA/Fault Control						
Set Interrupt Mask	SMK	IM	2	1	25	(I/O)
	SMK	IMX	2	1	28	(I/O)
	SMK	D	2	2	65	(1 I/O – 1 operation)
	SMK	DX	2	2	68	(1 I/O – 1 operation)
Clear Interrupt Request	CLIR	IM	2	1	28	(I/O)
	CLIR	IMX	2	1	31	(I/O)
	CLIR	D	2	2	68	(1 I/O – 1 operation)
	CLIR	DX	2	2	71	(1 I/O – 1 operation)
Enable Interrupts	ENBL	IM	2	1	26	(I/O)
	ENBL	IMX	2	1	29	(I/O)
	ENBL	D	2	2	66	(1 I/O – 1 operation)
	ENBL	DX	2	2	69	(1 I/O – 1 operation)
Disable Interrupts	DSBL	IM	2	1	26	(I/O)
	DSBL	IMX	2	1	29	(I/O)
	DSBL	D	2	2	66	(1 I/O – 1 operation)
	DSBL	DX	2	2	69	(1 I/O – 1 operation)
Reset Pending Interrupt	RPI	IM	2	1	36	(I/O)
	RPI	IMX	2	1	39	(I/O)
	RPI	D	2	2	76	(1 I/O – 1 operation)
	RPI	DX	2	2	79	(1 I/O – 1 operation)

Table 2 **F9450 Instruction Set** (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		Interrupt/DMA/Fault Control (Cont.)				
Set Pending Interrupt Register	SPI	IM	2	1	25	(I/O)
	SPI	IMX	2	1	28	(I/O)
	SPI	D	2	2	65	(1 I/O - 1 operation)
	SPI	DX	2	2	68	(1 I/O - 1 operation)
Read Interrupt Mask	RMK	IM	2	1	31	(I/O)
	RMK	IMX	2	1	34	(I/O)
	RMK	D	2	2	78	(1 I/O - 1 operation)
	RMK	DX	2	2	81	(1 I/O - 1 operation)
Read Pending Interrupt Register	RPIR	IM	2	1	31	(I/O)
	RPIR	IMX	2	1	34	(I/O)
	RPIR	D	2	2	78	(1 I/O - 1 operation)
	RPIR	DX	2	2	81	(1 I/O - 1 operation)
Read and Clear Fault Register	RCFR	IM	2	1	34	(I/O)
	RCFR	IMX	2	1	37	(I/O)
	RCFR	D	2	2	81	(1 I/O - 1 operation)
	RCFR	DX	2	2	84	(1 I/O - 1 operation)
DMA Enable	DMAE	IM	2	1	26	(I/O)
	DMAE	IMX	2	1	29	(I/O)
	DMAE	D	2	2	66	(1 I/O - 1 operation)
	DMAE	DX	2	2	69	(1 I/O - 1 operation)
DMA Disable	DMAD	IM	2	1	26	(I/O)
	DMAD	IMX	2	1	29	(I/O)
	DMAD	D	2	2	66	(1 I/O - 1 operation)
	DMAD	DX	2	2	69	(1 I/O - 1 operation)
		MMU Control				
Write Instruction Page Register	WIPR	IM	2	1	26	(I/O)
	WIPR	IMX	2	1	29	(I/O)
	WIPR	D	2	2	66	(1 I/O - 1 operation)
	WIPR	DX	2	2	69	(1 I/O - 1 operation)
Write Operand Page Register	WOPR	IM	2	1	26	(I/O)
	WOPR	IMX	2	1	29	(I/O)
	WOPR	D	2	2	66	(1 I/O - 1 operation)
	WOPR	DX	2	2	69	(1 I/O - 1 operation)
Read Instruction Page Register	RIPR	IM	2	1	31	(I/O)
	RIPR	IMX	2	1	34	(I/O)
	RIPR	D	2	2	69	(1 I/O - 1 operation)
	RIPR	DX	2	2	72	(1 I/O - 1 operation)
Read Operand Page Register	ROPR	IM	2	1	31	(I/O)
	ROPR	IMX	2	1	34	(I/O)
	ROPR	D	2	2	69	(1 I/O - 1 operation)
	ROPR	DX	2	2	72	(1 I/O - 1 operation)

Table 2 F9450 Instruction Set (Cont.)

Function	Mnemonic	Inst. Mode	Nf	No	Nc	Notes
		BPU Control				
Load Memory Protect RAM	LMP	IM	2	1	26	(I/O)
	LMP	IMX	2	1	29	(I/O)
	LMP	D	2	2	66	(1 I/O – 1 operation)
	LMP	DX	2	2	69	(1 I/O – 1 operation)
Read Memory Protect RAM	RMP	IM	2	1	31	(I/O)
	RMP	IMX	2	1	34	(I/O)
	RMP	D	2	2	69	(1 I/O – 1 operation)
	RMP	DX	2	2	72	(1 I/O – 1 operation)
Memory Protect Enable	MPEN	IM	2	1	26	(I/O)
	MPEN	IMX	2	1	29	(I/O)
	MPEN	D	2	2	66	(1 I/O – 1 operation)
	MPEN	DX	2	2	69	(1 I/O – 1 operation)
		Miscellaneous				
Write Status Word	WSW	IM	2	1	33	(I/O) MMU attached
	WSW	IMX	2	1	36	(I/O) MMU attached
	WSW	IM	2	1	38	(I/O) no MMU attached
	WSW	IMX	2	1	41	(I/O) no MMU attached
	WSW	D	2	2	73	(1 I/O – 1 operation) MMU attached
	WSW	DX	2	2	63	(1 I/O – 1 operation) MMU attached
	WSW	D	2	2	73	(1 I/O – 1 operation) no MMU attached
	WSW	DX	2	2	63	(1 I/O – 1 operation) no MMU attached
Read Status Word	RSW	IM	2	1	31	(I/O)
	RSW	IMX	2	1	34	(I/O)
	RSW	D	2	2	78	(1 I/O – 1 operation)
	RSW	DX	2	2	81	(1 I/O – 1 operation)
Reset Normal Power Up Discrete	RNS	IM	2	1	26	(I/O)
	RNS	IMX	2	1	29	(I/O)
	RNS	D	2	2	66	(1 I/O – 1 operation)
	RNS	DX	2	2	69	(1 I/O – 1 operation)
Pulse the $\overline{\text{TRIGO RST}}$ Signal	GO	IM	2	1	26	(I/O)
	GO	IMX	2	1	29	(I/O)
	GO	D	2	2	66	(1 I/O – 1 operation)
	GO	DX	2	2	69	(1 I/O – 1 operation)

Table 2 F9450 Instruction Set (Cont.)

Mnemonic	Function	I/O Command Field (Hex)	
Miscellaneous			
PO	Programmed Output	0YXX	The F9450 is totally transparent to these codes. No special operation results in the F9450 from the execution of these I/O commands. The F9450 treats these operations as any external I/O. Please refer to XIO and VIO instructions for execution times.
PI	Programmed Input	8YXX	
OD	Output Discretes	2008	
CLC	Clear Console	4001	
ESUR	Enable Start Up ROM	4004	
DSUR	Disable Start Up ROM	4005	
RIC1	Read I/O Interrupt Code, Level 1	A001	
RIC2	Read I/O Interrupt Code, Level 2	A002	
RDOR	Read Discrete Output Register	A008	
RDI	Read Discrete Input	A009	
TPIO	Test Programmed Output	A00B	
RMFS	Read Memory Fault Status	A00D	
RCS	Read Console Status	C001	

Table 3 F9450 Dedicated I/O Addresses

I/O Address/Command	Input/Output	Function
8400	Input	Read console command
8401	Input	Read console data
0400	Output	Write result into console
8410	Input	Read system configuration (see the "System Configuration Register (SCR)" section).
0800, 0900, 0A00, 0B00	Output	Write derived address to coprocessor no. 1, 2, 3, or 4, respectively (used to implement the Built-In Function).
0801, 0901, 0A01, 0B01	Output	Write op-code into coprocessor no. 1, 2, 3, or 4, respectively.
1000	Output	Indicate an interrupt acknowledge cycle. Used by external devices to reset their level-generated interrupts.

Note:
These dedicated I/O addresses/commands are in addition to those defined by MIL-STD-1750A.

Instruction Execution Times

The execution times, in μs, for a core instruction set in the different addressing modes are given in table 4, where all times are at 20 MHz with no wait states. The execution times degrade linearly with the clock rate if no wait states are inserted; i.e., the Add Register instruction will be 0.5 μs at 10 MHz. Table 5 gives the execution times in μs for a 15 MHz device under the same conditions.

Note that the times given for floating point operations do not include any shifts for exponent adjustment or result normalization. The following increments of clock periods should be used to estimate execution time for floating point operations that do require these shifts.

Table 4 F9450 Instruction Set Execution Times in μs, for a 20 MHz Device

Basic Instructions	Register	Direct	Direct Index	Indirect	Immediate	Immediate Short	Base Relative	Base Relative Index
Single Precision								
Load/Store	0.20	0.60	0.60	0.80	0.55	0.35	0.55	0.55
Add/Sub	0.25	0.65	0.65		0.60	0.40	0.60	0.60
Multiply	1.85	2.25	2.25		2.20		2.20	2.20
Divide	4.85	5.10	5.10		5.10		5.05	5.05
Compare	0.40	0.80	0.80		0.75	0.55	0.75	0.75
Set/Reset Bit	0.35	0.80	0.80	1.00				
Double Precision								
Load	0.80	1.10	1.10	1.30			1.05	1.05
Store		0.80	0.80	1.00			0.75	0.75
Add/Sub	0.90	1.20	1.20					
Multiply	6.30	6.60	6.60					
Divide	11.95	12.25	12.25					
Compare	1.05	1.35	1.35					
Floating Point								
Add/Sub*	3.10	3.40	3.40				3.35	3.35
Multiply*	6.00	6.30	6.30				6.25	6.25
Divide*	11.70	12.00	12.00				11.95	11.95
Compare*	2.60	2.90	2.90				2.85	2.85
Extended Floating-Point								
Load		1.30	1.30					
Store		1.00	1.00					
Add/Sub*	3.55	3.90	3.90					
Multiply*	12.55	12.90	12.90					
Divide*	24.00	24.35	24.35					
Compare*	2.60	3.25	3.25					
Branch	Taken 0.75	Not Taken 0.20						

*Includes no shifts in exponent adjust or in normalization. See Table 2 for examples.

Table 5 F9450 Instruction Set Execution Times in μs, for a 15 MHz Device

Basic Instructions	Register	Direct	Direct Index	Indirect	Immediate	Immediate Short	Base Relative	Base Relative Index
Single Precision								
Load/Store	0.267	0.800	0.800	1.067	0.733	0.467	0.733	0.733
Add/Sub	0.333	0.867	0.867		0.800	0.533	0.800	0.800
Multiply	2.467	3.000	3.000		2.933		2.933	2.933
Divide	6.467	6.800	6.800		6.800		6.733	6.733
Compare	0.533	1.067	1.067		1.000	0.733	1.000	1.000
Set/Reset Bit	0.467	1.067	1.067	1.333				
Double Precision								
Load	1.067	1.467	1.467	1.733			1.400	1.400
Store		1.067	1.067	1.333			1.000	1.000
Add/Sub	1.200	1.600	1.600					
Multiply	8.400	8.800	8.800					
Divide	15.933	16.333	16.333					
Compare	1.400	1.800	1.800					
Floating Point								
Add/Sub*	4.133	4.533	4.533				4.467	4.467
Multiply*	8.000	8.400	8.400				8.333	8.333
Divide*	15.600	16.000	16.000				15.933	15.933
Compare*	3.467	3.867	3.867				3.800	3.800
Extended Floating Point								
Load		1.733	1.733					
Store		1.333	1.333					
Add/Sub*	4.733	5.200	5.200					
Multiply*	16.733	17.200	17.200					
Divide*	32.000	32.467	32.467					
Compare*	3.467	4.333	4.333					
Branch	Taken 1.000	Not Taken 0.267						

*Includes no shifts in exponent adjust or in normalization. See Table 2 for examples.

Shifts for Standard Precision Floating Point Operations

- If the source exponent must be adjusted:
  add (3+8S) clocks (where S = number of shifts required)
- If the destination exponent must be adjusted:
  add (8+8S) clocks
- For result normalization:
  add (5+13N) clocks (where N = number of shifts required)
- Guidelines refer to Tables 2, 4, and 5.

Shifts for Extended Precision Floating Point Operations

- If the source exponent must be adjusted:
  add (5+9S) clocks
- If the destination exponent must be adjusted:
  add (8+9S) clocks
- For result normalization:
  add (10+16N) clocks

Interrupts

There are 16 levels of interrupt prioritized on-chip, as listed in table 6. Nine interrupts are external (PIR 0, 2, 8, 10-15), of which two are level-sensitive (IOL_1 INT, IOL_2 INT), and the other seven are either level- or edge-sensitive (USR_0 INT – USR_5 INT, PWRDN INT), according to the interrupt mode bit in the System Configuration Register. The remaining seven interrupts are internal to the CPU (PIR 1, 3-7, 9). All interrupts are latched into the Pending Interrupt Register (PIR) and may be disabled or masked by the Mask Register (MK), except as indicated in table 6. All external interrupt inputs (except IOL_1 INT and IOL_2 INT) have hysteresis circuitry for noise immunity.

An enabled interrupt with highest priority that is not masked is processed as shown in figure 3. Upon completion of the current instruction, which is not aborted, further interrupts are disabled. The F9450 then reads, via the Service Pointer (with AS = 0), the new Mask, Status Word, and Instruction Counter. It then stores, via the Linkage Pointer (with the new AS), the old Mask, Status Word, and Instruction Counter.

Interrupts are acknowledged automatically by resetting the appropriate interrupt bit in the PIR and executing an I/O cycle, during which the acknowledged interrupt number is sent to I/O device 1000 (see figure 3). For example, if interrupt level 11 is to be acknowledged, the CPU performs an I/O write cycle to address 1000 (in the I/O space) and writes data equal to FFEF (IB11 = 0, all other bits = 1). Level interrupt requests should be removed within two machine cycles after their respective interrupt acknowledge (IOW) cycle. If an additional delay in removing the interrupt request is needed in a particular system application, this period could be extended by inserting wait states.

The Pending Interrupt Register can be loaded via a privileged XIO instruction to generate simulated interrupts.

The Executive Call (software interrupt), invoked by the BEX instruction, provides a means to jump to a routine in another address state (AS). It is typically used to make controlled, protected calls to an executive, using one of 16 executive entry points.

The BEX instruction does not set PIR bit 5. However, if PIR bit 5 is set to = 1 from program execution of SPI, a BEX = 0 will be executed.

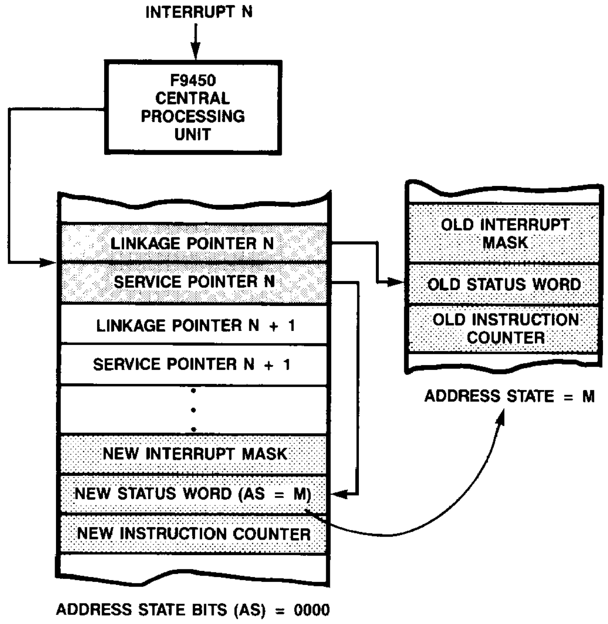
Table 6 Interrupt Priorities

Interrupt	Priority (PIR/MK Bit Number)	Interrupt Linkage Pointer Address (Hex)	Interrupt Service Pointer Address (Hex)
Power Down[1]	0[2]	20	21
Machine Error[3]	1	22	23
User 0	2	24	25
Floating-Point Overflow	3	26	27
Fixed-Point Overflow	4	28	29
Executive Call[1]	5[4]	2A	2B
Floating-Point Underflow	6	2C	2D
Timer A	7	2E	2F
User 1	8	30	31
Timer B	9	32	33
User 2	10	34	35
User 3	11	36	37
I/O Level 1	12	38	39
User 4	13	3A	3B
I/O Level 2	14	3C	3D
User 5	15	3E	3F

Notes:

1. Cannot be masked or disabled.
2. Interrupt level 0 has the highest priority.
3. Cannot be disabled.
4. BEX is not part of the Interrupt priorities; it is the highest priority regardless of the PIR setting. The BEX instruction execution, once started, must complete before another interrupt can occur.

Figure 3 Interrupt Vectors



Fault and Error Handling

Extensive fault handling, as required by real-time applications, involves the Fault Register, interrupt priority processing, and the abort scheme. Four levels of severity exist for faults and errors:

1. **Unrecoverable Errors** - errors with fatal implications for program execution. This class is recorded in the FT and will generate a machine error interrupt (priority level 1) that cannot be disabled. The five unrecoverable errors are:

   —Illegal instruction

   —Memory protect error on an instruction fetch

   —Parity error on an instruction fetch

   —External address error on an instruction fetch

   —Address state fault

2. **Major Errors** - errors without fatal implications for program execution. This class is recorded in the FT and results in a machine error interrupt (priority level 1) after the instruction has completed execution. The three types of major errors are:

   —Privileged instruction violation

   —Memory protect or parity error on a CPU-initiated memory data bus cycle

   —External address error on a CPU-initiated data bus (memory or I/O) read cycle

3. **Unclassified Errors (Warnings)** - are also recorded in the FT and will generate a machine error interrupt (priority level 1). See the "Fault Register (FT)" section for examples of unclassified errors.

4. **Arithmetic Errors** - other errors resulting from arithmetic exception cases (overflow, etc.). These are not recorded in the FT but in the PIR to generate lower-priority interrupts.

Note that the FT records unrecoverable, major, and unclassified errors from violations internal and external to the CPU. Any bit set in the FT will set PIR 1 and cause a level 1 interrupt. The FT can be cleared by the execution of the XIO instruction "CLR INT REQ" (2001 Hex), which also clears the PIR, or the instruction "READ AND CLEAR FT" (A00F Hex), which first transfers the contents of FT to the specified register and then clears FT and PIR 1, or by executing the console command EXAMINE AND CLEAR FAULT REGISTER (FT).

Instruction Abort

An instruction will be aborted by any one of three major errors:

—Memory protect error ($\overline{\text{MEM PRT ER}}$)

—Parity error ($\overline{\text{PAR ER}}$)

—External address error ($\overline{\text{EXT ADR ER}}$) on a CPU-initiated data bus (memory or I/O) read cycle

Aborted instructions will complete execution, with modification of internal registers inhibited. Erroneous information may be written into memory or I/O locations. Outputs UNRCV ER (pin 8) and MAJ ER (pin 31) may be OR-wired together to form the ABORT input for the F9451 MMU (pin 13) and/or the F9452 BPU (pin 11), if these peripherals are part of the system. This external ABORT signal may be connected to the Power Down Interrupt (PWRDN INT) input (pin 33) to ensure proper interrupt handling of the major and unrecoverable error conditions, even if the machine error interrupt (bit 1 of PIR) is masked out.

Three conditions will cause an effective NOP (i.e., the current instruction will not be executed and the next instruction will be fetched):

—Two unrecoverable errors - illegal instruction and address state fault

—One major error - privileged instruction violation

These conditions will set the appropriate bits in the FT and cause a machine error interrupt.

Note: The user should be aware that three unrecoverable errors,

—Memory protect error on instruction fetch

—Parity error on instruction fetch

—External address error on instruction fetch,

result in a machine error interrupt. Because of the F9450 pipeline architecture, faults that occur during the instruction fetch result in an internal machine error interrupt which may occur before the instruction can be executed, providing the machine error interrupt is not masked.

If the last machine cycle of an operation includes an instruction fetch, and the second word of the following instruction is fetched, and a fault occurs on this fetch, the new instruction will be executed before the interrupt occurs. However, the interrupt occurs prior to the execution of the next instruction, as required by MIL-STD-1750A.

Also, some instructions that transfer control to another part of the program may initiate the fetch of the following instruction before flushing the pipeline. If a fault occurs on this fetch, the interrupt will occur prior to the transfer of control.

Provisions should be made by the user to handle these faults in the interrupt service routine.

Fault Register (FT)

The 16-bit-wide Fault Register records major (M), unrecoverable (U), or unclassified (W) faults during data or instruction fetch operations as shown in table 7.

Table 7 Fault Register Bit Assignments

		ERRORS		
Bit	**Fault**	**Major**	**Unrecoverable**	**Unclassified**
0	CPU Memory Protect Error			
	Data Cycle	M		
	Instruction Fetch		U	
1	Non-CPU Memory Protect Error			W
2	Parity Error			
	Data Cycle			W
	Instruction Fetch		U	
3	Spare (Zero)			
4	Spare (Zero)			
5	Illegal I/O Address	M		
6	Spare (Zero)			
7	System Fault 0*			W
8	Illegal Memory Address			
	Data Cycle	M		
	Instruction Fetch		U	
9	Illegal Instruction		U	
10	Privileged Instruction	M		
11	Address State Error		U	
12	Spare (Zero)			
13	BITE (Built-In Test) or System Fault 1 Unc.			W
14	Spare (Zero)			
15	System Fault 1*			W

*Signals $SYSFLT_0$ and $SYSFLT_1$ are asynchronous, edge-sensitive inputs to the F9450 processor and have hysteresis circuitry for noise immunity.

F9450 COMPONENT DESCRIPTION

The F9450 microprocessor architecture is organized in five sections, as illustrated in figure 4.

—Data processor

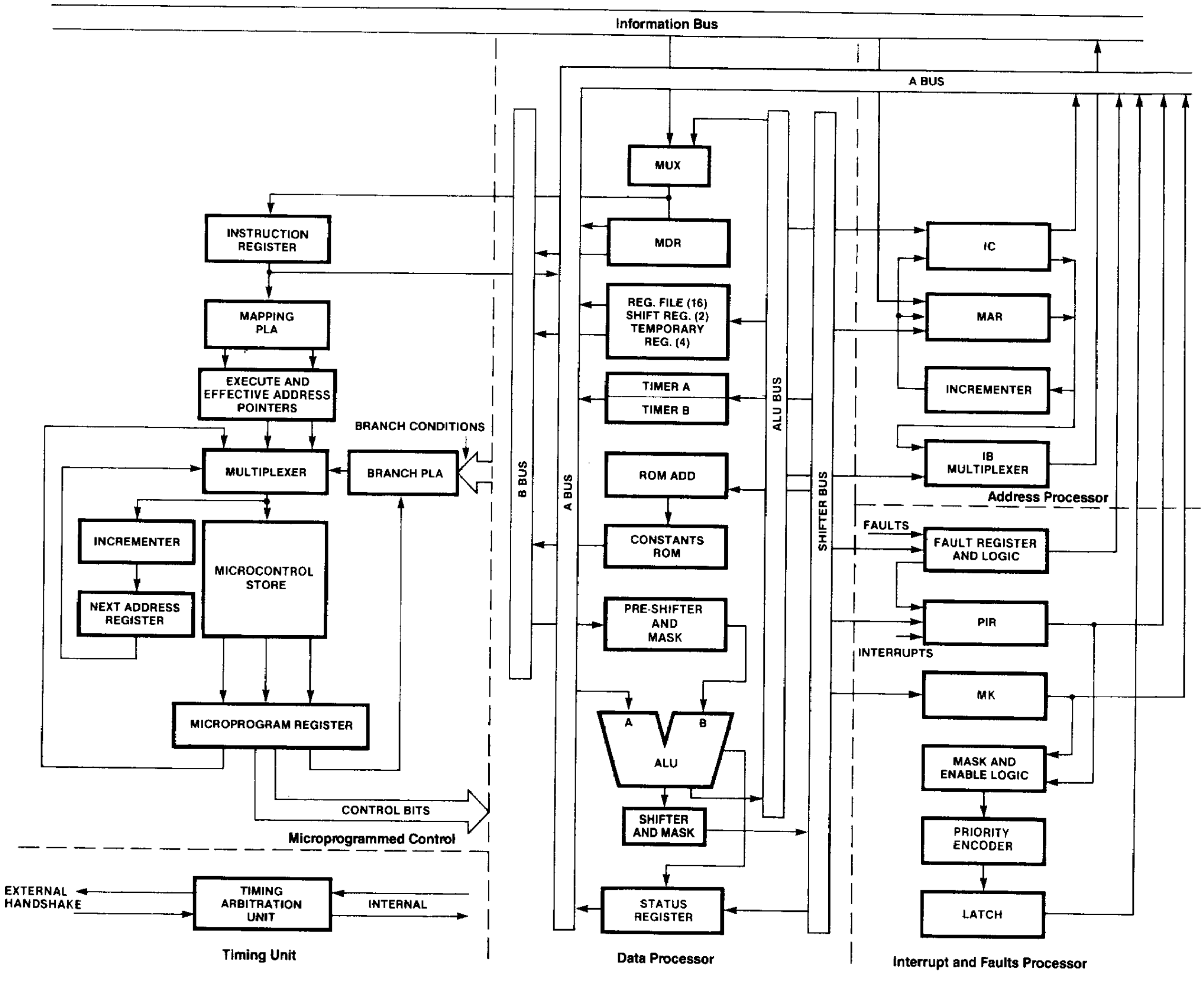
—Microprogrammed control

—Address processor

—Interrupt and fault processor

—Timing unit

Figure 4 F9450 Block Diagram



Data Processor

The 16-bit-wide data processor section is responsible for all data processing in the CPU. It is organized in nine functional blocks:

—17-bit ALU

—Shifter

—16 general-purpose registers (R0–R15)

—Six temporary registers

—Memory Data Register (MDR)

—Two timers

—Constants ROM

—Status Word Register (SW)

—Pre-shifter and mask

Microprogrammed Control

The operation of the CPU is governed by a microprogrammed control section with two levels of pipelining. New instructions are fetched into the instruction register. The mapping PLA is fed from the instruction register and generates the pointers necessary for both execution and the effective address routines that reside in the microcontrol store. The microcontrol store generates three output fields to the microregister. Two—Next Address Field and Branch Field—determine the subsequent microaddress. The Output Field controls the operation of all CPU components.

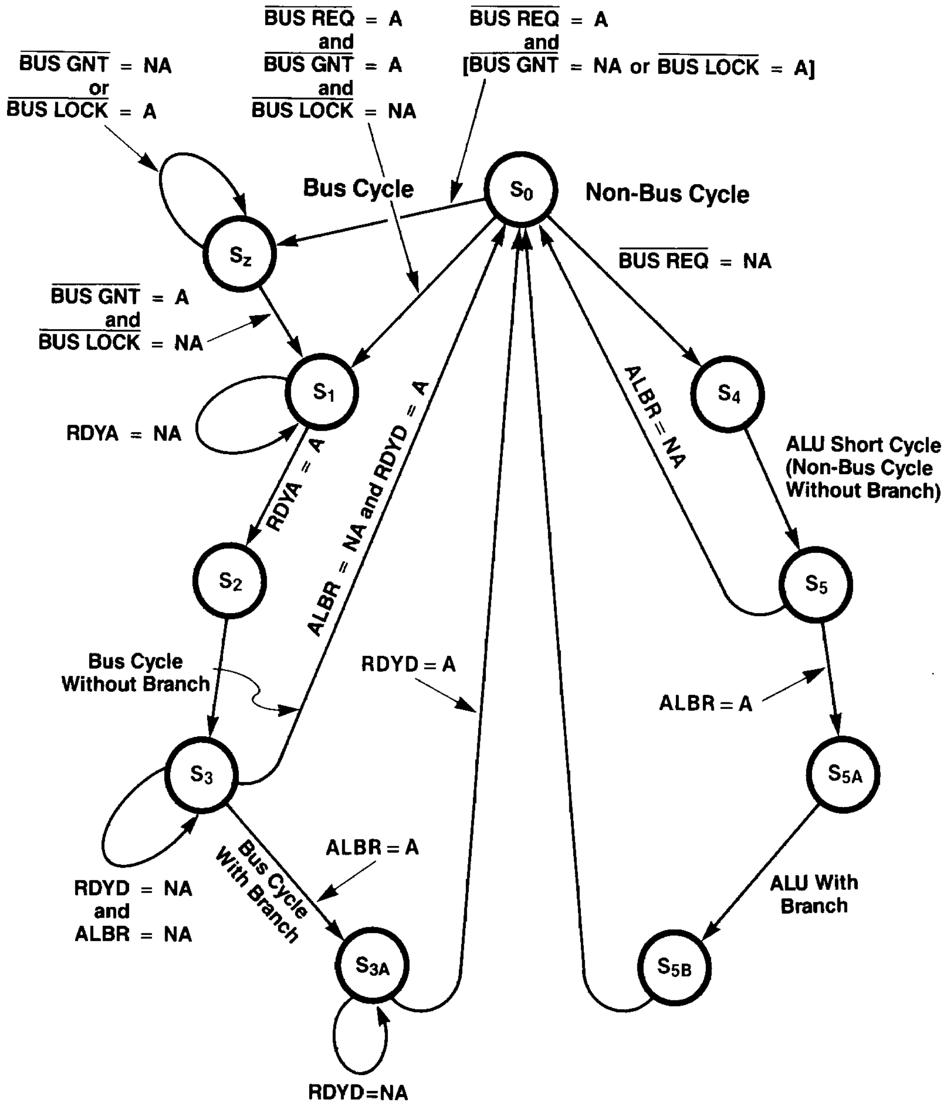
Address Processor

The address processor includes an Instruction Counter (IC) and a Memory Address Register (MAR) that determine the addresses for all instructions and operands. Included in the Address Processor is an independent incrementer that provides Instruction Counter and operand address updates in parallel with data processor operation.

Interrupt and Fault Processor

All faults and interrupts, whether generated internally or externally, are handled by the Interrupt and Fault Processor. It includes the Pending Interrupt Register (PIR), Mask Register (MK), Fault Register (FT), interrupt enabling logic, and a priority encoder. Also included is abort condition detection and activation logic.

Figure 5 F9450 Timing Generator State Diagram



Legend

A	=	Asserted (active)
NA	=	Not Asserted
ALBR	=	ALU Branch Cycle (5 States) – Internal Signal Indicating Conditional Branch Within Microcode
BUS REQ	=	Bus Request Output
BUS GNT	=	Bus Grant Output
BUS LOCK	=	Bus Lock from External Source
RDYA	=	RDYA Input
RDYD	=	RDYD Input
S_z	=	High-Impedance State – CPU Drivers Are Three-State

Timing Unit

The Timing Unit generates the internal and external strobes required for internal CPU operation and the different bus transactions. A basic machine cycle could comprise three, four, or five CPU clock cycles (states), as illustrated in figure 5.

1. A 3-state cycle (S0, S4, S5) for pure internal ALU operations.
2. A 4-state cycle (S0, S1, S2, S3) for minimum length bus cycles.
3. A 5-state cycle (S0, S1, S2, S3, S3A or S0, S4, S5, S5A, S5B) applies for those cycles that use the result of the current ALU operation to determine the next address in the microprogrammed control store.

As shown in figure 5, every timing cycle starts with state S0, in which the timing unit receives the control information needed to initiate a bus cycle or a short ALU cycle.

A bus cycle can be extended indefinitely by manipulating the $\overline{\text{BUS GNT}}$, RDYA, or RDYD external inputs. The $\overline{\text{BUS GNT}}$ signal holds the CPU in the Sz (high-impedance) state when the bus is assigned to another CPU or DMA device. The RDYA signal holds the CPU in the S1 (address phase) state; the RDYD signal holds the CPU in the S3 (data phase) state.

Self-Test and Initialization

The Self-Test is part of the Initialization sequence (see figure 6) that is invoked by asserting the RESET signal. Asserting RESET forces the processor into an S5 state on the next rising edge of the CPU clock, regardless of all other inputs. The processor remains in the S5 state until RESET goes high, at which time the Self-Test sequence begins. The Self-Test functions are:

1. Reads and writes all registers in the register file.
2. Verifies ALU functions.
3. Checks multiply hardware by performing a multiply.
4. Checks divide hardware by performing a divide.
5. Checks ALU shifter in both directions by multiply and divide.
6. Verifies ROM constants can be accessed.
7. Verifies that the IC and MAR can be accessed and incremented.

If the Self-Test is successfully completed, the Normal Power Up (NML PWRUP) discrete output is set; otherwise, bit 13 in the Fault Register is set.

The Self-Test function reads the system configuration from I/O address 8410 into the SCR to determine the presence or absence of the MMU, BPU, Console, and Coprocessor, and also initializes the interrupt mode and then initializes the system, as shown in table 8.

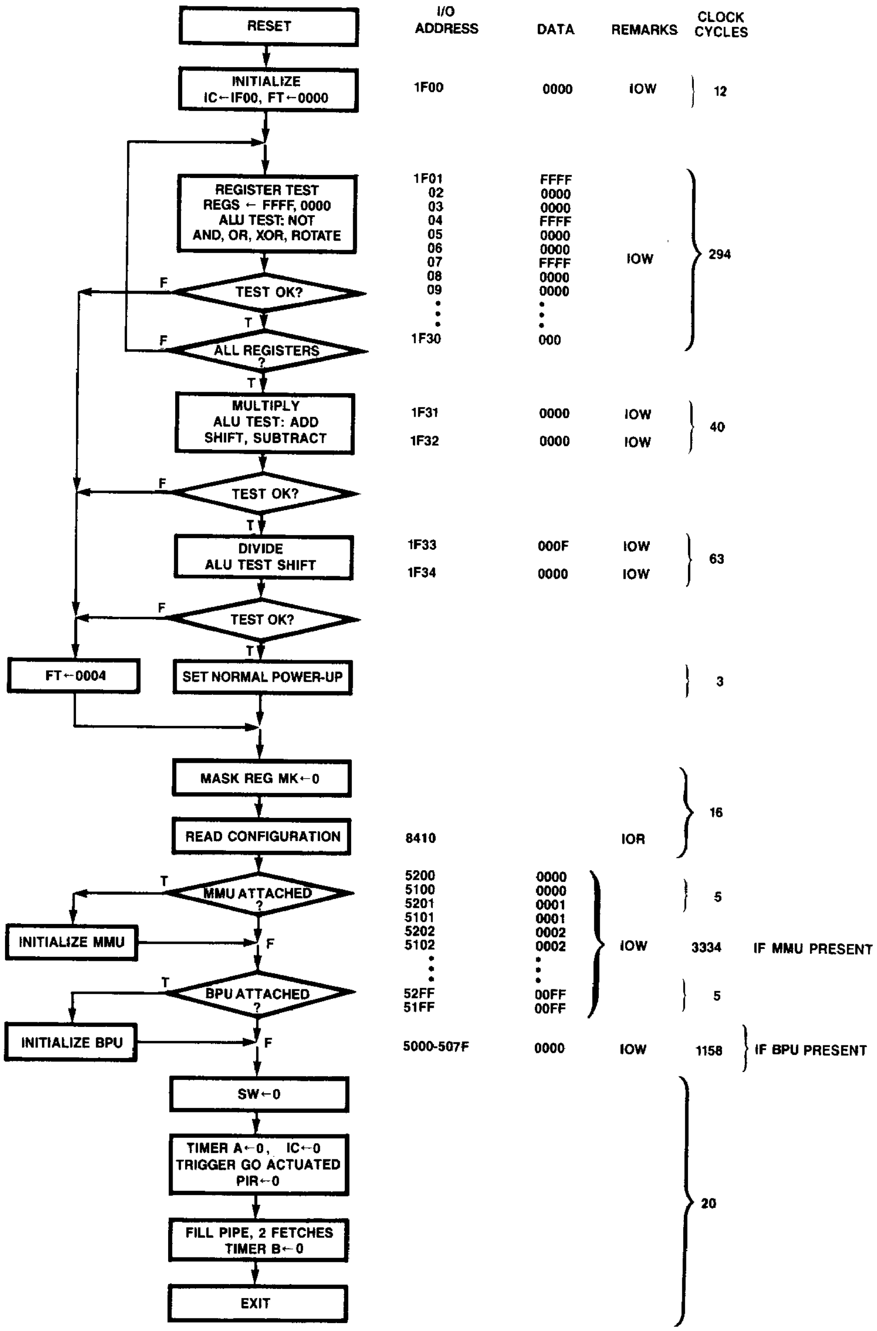
As depicted in the Self-Test flow chart (figure 6), while FT is originally set to all zeros, bit 13 is set any time a partial self-test is failed. Any external fault occurring prior to assertion of Normal Power Up causes a fault to be pending on completion of the self-test. From Normal Power-Up to the end of Initialization, the number of clock cycles can be determined from the figure, including the additional clock cycles spent for initializing the F9451 MMU and/or the F9452 BPU, if they are part of the system.

The Self-Test is intended to test useability of major functional sections of the F9450. It does **not** test all instructions or functions, but ensures basic operation of the F9450.

Table 8 F9450 System Initialization

CPU	
Instruction Counter (IC)	All Zeros
Status Word (SW)	All Zeros
Fault Register (FT)	All Zeros
Pending Interrupt Register (PIR)	All Zeros
Interrupt Mask Register (MK)	All Zeros
Interrupt	Disabled
DMA Enable	Disabled
Timer A and Timer B	All Zeros and Counting Up
Trigger Go Reset ($\overline{\text{TRIGO RST}}$)	Pulsed
Normal Power Up (NML PWRUP)	Set High
MMU	
Page Registers	
AL Field	All Zeros
W Field	All Zeros
E Field	All Zeros
PPA Field	Logical to Physical
BPU	
CPU Write Protect Registers	All Zeros
DMA Write Protect Registers	All Zeros
Global Memory Protect	Enabled

Figure 6 Self-Test and Initialization Sequence



Note:

The purpose of the I/O Write (IOW) addresses is to inform the user of the presently executed instruction in the self-test program. The user must furnish bus control signals to the CPU to enable it to continue the self-test sequence.

CONSOLE OPERATIONS

The F9450 offers a feature that is unique among microprocessors: the capability of being connected to a programmer's console. This connection gives the programmer the ability to examine and change the contents of various registers within the CPU, the system memory, and the I/O subsystems.

The F9450 console is treated as three input/output addresses:

1. Console command 8400H
2. Console data read address (switches) 8401H
3. Console data write address (displays) 0400H

Entering Console Mode

The Console Mode is entered in one of two ways:

1. Drive the $\overline{\text{CON REQ}}$ input low. The CPU completes the current instruction, executes an I/O cycle to read the console command from the information bus (I/O address 8400H), and then executes the console command. (See table 9 for a list of the valid console operations and their corresponding codes.) Console operations typically involve I/O address 8401H to enter console data and/or 0400H to write data to the console.
2. Execute the BPT instruction. The CPU reads the system configuration bits from I/O address 8410H. If the console bit is reset, the CPU treats the BPT instruction as a NOP instruction. Otherwise, it enters Console Mode and waits for a console request.

When responding to a Console Request, the F9450 executes the user-specified console command and then goes into a loop, waiting for the $\overline{\text{CON REQ}}$ signal to be asserted.

Use of Console Mode

The sequence of console operations follows. (See figure 7, a flow chart of the console handshake.)

1. The Console Request ($\overline{\text{CON REQ}}$) input (pin 2) is activated, which has a higher priority than interrupts. It is sampled at the beginning of the last microcycle of each instruction.
2. On completing the instruction that is being executed and recognizing the console request, the processor enters the Console Mode and halts Timer A and Timer B. They resume counting from this same point when the CPU exits console operations.
3. The CPU issues an I/O Read command using I/O address 8400H. At that time, the $\overline{\text{CON REQ}}$ signal should be deactivated and the console command placed on the bus.
4. On decoding this console command, the CPU proceeds through the proper steps to execute the command. Often this includes additional I/O cycles to either read data or an address from the console (I/O address 8401H) or to write data to the console (I/O address 0400H)
5. At this time, the CPU remains in the Console Mode and waits for the $\overline{\text{CON REQ}}$ signal to be reasserted.

Exiting Console Mode

The Console Mode is exited in one of two ways:

1. If an interrupt is pending when the CPU is executing a DISABLE command, the CPU services the interrupt and returns to the normal mode of operation, fetching and executing instructions from the system memory.

   If no interrupt is pending at this time, the CPU remains in an intermediate state, alternately checking for pending interrupts or a console request.
2. The CPU executes a console CONTINUE command.

Notes

The F9450 executes the EXAMINE MEMORY console command by reading the address from I/O address 8401, storing that address into the IC, reading memory at that address, and then writing that data to I/O address 0400. When the EXAMINE NEXT command is executed, the F9450 first increments the IC, reads memory at the address specified by the IC, and then writes that data to I/O address 0400.

The DEPOSIT MEMORY command deposits the user-provided data (from I/O address 8401) into the memory location pointed to by the F9450 Instruction Counter (IC). When the DEPOSIT NEXT command is executed, the F9450 first increments the IC and then stores the user-provided data into the memory location pointed to by the new value of the IC.

The EXAMINE XIO console command reads the I/O address from I/O location 8401 and saves that address in A1 (one of the F9450 internal scratchpad registers). The CPU then performs an I/O read from the address in A1 and writes that data to I/O address 0400. The DEPOSIT XIO command writes the user-specified data to the I/O address contained in A1. This console command expects the I/O address to be preloaded in register A1 by one of two methods:

1. Through the DEPOSIT REGISTER console command.
2. Execute an EXAMINE XIO command before executing a DEPOSIT XIO command.

When executing the EXAMINE NEXT XIO and DEPOSIT NEXT XIO console commands, the F9450 first increments register A1 and then performs the appropriate transfer with the I/O location pointed to by the new value of A1. While normally the most significant bit (MSB) of the I/O address indicates direction (read or write), this is not true for four console commands: EXAMINE XIO, DEPOSIT XIO, EXAMINE NEXT XIO, and DEPOSIT NEXT XIO.

During console commands that generate a bus cycle, the state of the D/I line is 1 (Data cycle). An illegal console command is treated as a NOP instruction and the CPU waits for another console request.

The processor CANNOT be single-stepped by maintaining an active CON REQ signal and repeatedly responding to the I/O read at 8400H with the CONTINUE command. If this is done, the CPU recognizes the CON REQ before executing the instruction and never exits the Console Mode. To perform the single-step operation, the CONTINUE command is executed and the CON REQ signal is activated immediately on recognizing the second fetch from instruction space in the system memory.

If an external fault (EXT ADR ER, MEM PRT ER, or PAR ER) occurs during a console operation, the register file is protected in the same manner as if an instruction were in process. None of the 16 general-purpose registers or the six temporary registers can be modified until the processor exits the Console Mode. To exit the Console Mode, execute a console CONTINUE command or reset the processor.

Fairchild's application Note MC-2, "Simple Console Controller for the F9450", describes how a specially programmed MOSTEK 38P70 microcontroller can be implemented to control the Console function of the F9450.

Table 9 **Console Command Formats**

0 – 7	8	9[1]	10 – 15
CONSOLE CODE	X	X	REGISTER ADDRESS

Console Code and Command (Hex)

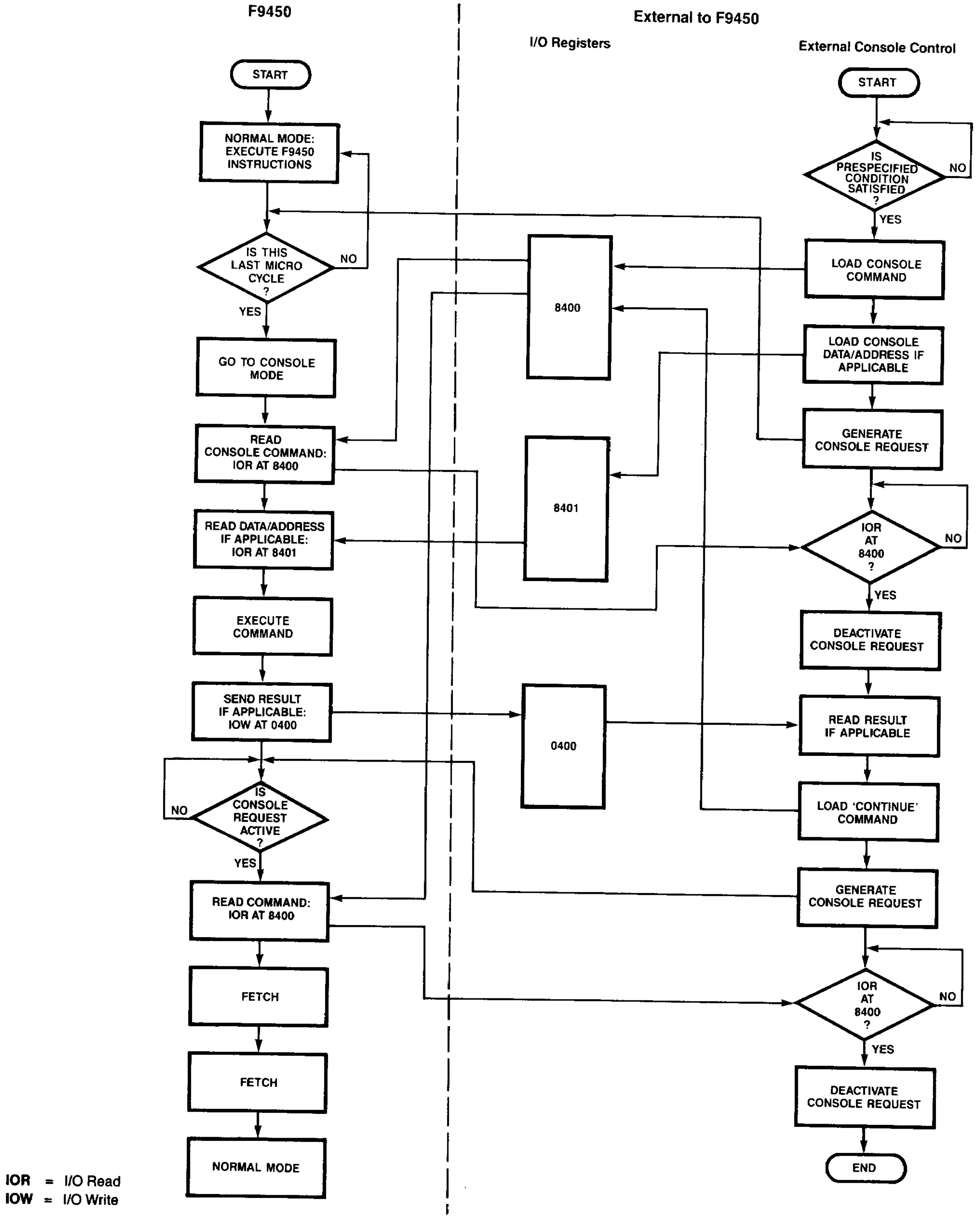
74	:	DISABLE (1)[2, 3]
60	:	EXAMINE REGISTER (1)
61	:	DEPOSIT REGISTER (2)
62	:	EXAMINE AND CLEAR FAULT REGISTER (FT) (1)[3]
63	:	DEPOSIT STATUS WORD (SW) (2)[3]
66	:	EXAMINE MEMORY (2)[3]
67	:	DEPOSIT MEMORY (2)[3]
6A	:	EXAMINE NEXT MEMORY (1)[3]
6B	:	DEPOSIT NEXT MEMORY (2)[3]
75	:	CONTINUE (1)[3]
6C	:	EXAMINE XIO (2)[3]
6D	:	DEPOSIT XIO (2)[3]
6E	:	EXAMINE NEXT XIO (1)[3]
6F	:	DEPOSIT NEXT XIO (2)[3]

Binary						Hex	Reg.
0	0	0	0	0	0	(00)	R0
0	1	0	0	0	0	(10)	R1
0	0	0	0	0	1	(01)	R2
0	1	0	0	0	1	(11)	R3
0	0	0	0	1	0	(02)	R4
0	1	0	0	1	0	(12)	R5
0	0	0	0	1	1	(03)	R6
0	1	0	0	1	1	(13)	R7
0	0	0	1	0	0	(04)	R8
0	1	0	1	0	0	(14)	R9
0	0	0	1	0	1	(05)	R10
0	1	0	1	0	1	(15)	R11
0	0	0	1	1	0	(06)	R12
0	1	0	1	1	0	(16)	R13
0	0	0	1	1	1	(07)	R14
0	1	0	1	1	1	(17)	R15
0	0	1	0	0	0	(08)	A2
0	1	1	0	0	0	(18)	A1
0	0	1	0	0	1	(09)	Q2
0	1	1	0	0	1	(19)	Q1
0	0	1	0	1	0	(0A)	DO0
0	1	1	0	1	0	(1A)	DO1
0	0	1	0	1	1	(0B)	PIR
0	1	1	0	1	1	(1B)	MK
0	0	1	1	0	0	(0C)	FT[4]
0	0	1	1	0	1	(0D)	SW[5]
0	0	1	1	1	0	(0E)	TA
0	1	1	1	1	0	(1E)	TB
1	0	0	1	0	1	(25)	IC[5]
1	0	1	1	X	X	(2C–2F)	IR

Notes:

1. Bits 8 and 9 are always "Don't Care."
2. The (1) and (2) designations indicate a 1- or 2-word-long console command:
   (1) indicates command only
   (2) indicates command and data
3. For commands that do not require a register address, bits 10 through 15 are "Don't Care."
4. In executing the DEPOSIT REGISTER command (61 Hex), only bits 9, 10, 11, and 13 of the Data Word can be deposited into the FT register. These four bits are user-specified with the inverted polarity. The remaining 12 bits are Don't Care; e.g., to set bits 10, 11, 13 to a one and bit 9 to a zero, Data Word XXXX XXXX X100 X0XX must be supplied from I/O location 8401.
5. Only the EXAMINE REGISTER command applies to the Status Word (SW) and Instruction Counter (IC) registers. To deposit (load) the IC with a 16-bit value, the EXAMINE MEMORY console command is used. The address specified in this command (the second word) is automatically loaded into the IC.

Figure 7 F9450 Console Handshake Sequence


F9450
External to F9450
I/O Registers
External Console Control
START
NORMAL MODE: EXECUTE F9450 INSTRUCTIONS
IS THIS LAST MICRO CYCLE ?
NO
YES
GO TO CONSOLE MODE
READ CONSOLE COMMAND: IOR AT 8400
READ DATA/ADDRESS IF APPLICABLE: IOR AT 8401
EXECUTE COMMAND
SEND RESULT IF APPLICABLE: IOW AT 0400
IS CONSOLE REQUEST ACTIVE ?
NO
YES
READ COMMAND: IOR AT 8400
FETCH
FETCH
NORMAL MODE
8400
8401
0400
START
IS PRESPECIFIED CONDITION SATISFIED ?
NO
YES
LOAD CONSOLE COMMAND
LOAD CONSOLE DATA/ADDRESS IF APPLICABLE
GENERATE CONSOLE REQUEST
IOR AT 8400 ?
NO
YES
DEACTIVATE CONSOLE REQUEST
READ RESULT IF APPLICABLE
LOAD 'CONTINUE' COMMAND
GENERATE CONSOLE REQUEST
IOR AT 8400 ?
NO
YES
DEACTIVATE CONSOLE REQUEST
END
IOR = I/O Read
IOW = I/O Write

SIGNAL DESCRIPTIONS

The F9450 input and output signals are described in table 10.

Table 10 F9450 Signal Descriptions

Mnemonic	Pin No.	Name	Description
Clocks			
CPU CLK	51	CPU Clock	Single-phase input clock signal (0-20 MHz, 40% to 60% duty cycle).
TIMER CLK	17	Timer Clock	A 100 kHz input that, after synchronization with CPU CLK, provides the clock for Timer A and Timer B. If timers are used, the CPU CLK signal frequency must be >500 kHz.
External Requests			
$\overline{\text{RESET}}$	5	Reset	An active-low input that initializes the F9450. The processor must be $\overline{\text{RESET}}$ after input power (V_{CC}, I_{INJ}) is within spec and stable, to ensure proper operation.
$\overline{\text{CON REQ}}$	2	Console Request	An active-low input that initiates console operations after the current instruction.
Interrupts			
PWRDN INT	33	Power Down Interrupt	An input interrupt request that cannot be masked or disabled. This signal is active on the positive-going edge or the high level, according to the interrupt mode bit in the configuration register. This input has hysteresis circuitry for noise immunity.
USR_0 INT– USR_5 INT	34-39	User Interrupt	Input signals that are active on the positive-going edge or high level, according to the interrupt mode bit in the configuration register. These inputs have hysteresis circuitry for noise immunity.
IOL_1 INT IOL_2 INT	40, 41	I/O Level Interrupts	Active-high inputs that can be used to expand the number of user interrupts.
Faults			
$\overline{\text{MEM PRT ER}}$	26	Memory Protect Error	An active-low input generated by the MMU or BPU, or both, and sampled by the $\overline{\text{BUS BUSY}}$ signal into the Fault Register (bit 0 if CPU bus cycle, bit 1 if non-CPU bus cycle).
$\overline{\text{PAR ER}}$	27	Parity Error	An active-low input sampled by the $\overline{\text{BUS BUSY}}$ signal into bit 2 of the Fault Register.
$\overline{\text{EXT ADR ER}}$	28	External Address Error	An active-low input sampled by the $\overline{\text{BUS BUSY}}$ signal into the Fault Register (bit 5 or 8), depending on the cycle (memory or I/O).

Table 10 F9450 Signal Descriptions (Cont.)

Mnemonic	Pin No.	Name	Description
Faults (Cont.)			
SYSFLT$_0$ SYSFLT$_1$	29, 30	System Fault 0 System Fault 1	Asynchronous, positive-edge-sensitive inputs to the F9450 that set bit 7 (SYSFLT$_0$) or bits 13 and 15 (SYSFLT$_1$) in the Fault Register. These inputs are protected from system noise through hysteresis circuitry.
Information Bus			
IB$_0$–IB$_{15}$	9-18, 20-25	Information Bus	An active-high bidirectional time-multiplexed address/data bus that is 3-state during bus cycles not assigned to this CPU; IB$_0$ is the most significant bit.
Status Bus			
AK$_0$–AK$_3$	47-50	Access Key	Active-high outputs used to match the Access Lock in the MMU for memory accesses (a mismatch is one of several possible situations that cause the MMU to pull the $\overline{\text{PRT ER}}$ signal low). These signals are 3-state during bus cycles not assigned to this CPU.
AS$_0$–AS$_3$	42-45	Address State	Active-high outputs that select the page register group in the MMU; 3-state bus during bus cycles not assigned to this CPU. These outputs together with D/I can be used to expand the F9450 direct addressing range to 2M words.
Error Control			
UNRCV ER	8	Unrecoverable Error	An active-high output that indicates the occurrence of an error classified as unrecoverable.
MAJ ER	31	Major Error	An active-high output indicating the occurrence of an error classified as major.
Discrete Control			
DMA EN	3	Direct Memory Access Enable	An active-high output that indicates the DMA is enabled. It is disabled when the CPU is initialized (reset) and is enabled under program control.
NML PWRUP	6	Normal Power Up	An active-high output that is set when the CPU has successfully completed the built-in test in the initialization sequence.
SNEW	63	Start New	An active-high output that indicates a new instruction will start executing in the next cycle; useful for instruction tracing function.
$\overline{\text{TRIGO RST}}$	4	Trigger Go Reset	An active-low discrete output. This signal can be pulsed low under program control [I/O address 400B (Hex)] and is automatically pulsed once during processor initialization.
Bus Control			
D/$\overline{\text{I}}$	58	Data or Instruction	An output signal that indicates whether the current bus cycle access is for Data (high) or Instruction (low); 3-state during bus cycles not assigned to this CPU. This line can be used as an additional memory address bit for systems that require separate data and program memory.

Table 10 F9450 Signal Descriptions (Cont.)

Mnemonic	Pin No.	Name	Description
Bus Control (Cont.)			
$R/\overline{W}$	57	Read or Write	An output signal that indicates direction of data flow with respect to the current bus master: a high indicates a read or input operation and a low indicates a write or output operation. The signal is 3-state during bus cycles not assigned to this CPU.
$M/\overline{IO}$	59	Memory or I/O	Output signal that indicates whether the current bus cycle is memory (high) or I/O (low). This signal is 3-state during bus cycles not assigned to this CPU.
STRBA	52	Address Strobe	An active-high output that can be used to externally latch the memory or I/O address at the high-to-low transition of the strobe. The signal is 3-state during bus cycles not assigned to this CPU.
RDYA	55	Address Ready	An active-high input that can be used to extend the address phase of a bus cycle. When RDYA is not active, wait states are inserted by the F9450 timing unit to accommodate slower memory or I/O devices.
$\overline{\text{STRBD}}$	53	Data Strobe	An active-low output that can be used to strobe data in memory and XIO cycles. This signal is 3-state during bus cycles not assigned to this CPU.
RDYD	56	Data Ready	An active-high input that extends the data phase of a bus cycle. When RDYD is not active, wait states are inserted by the F9450 timing unit to accommodate slower memory or I/O devices.
Bus Arbitration			
$\overline{\text{BUS REQ}}$	54	Bus Request	An active-low output that indicates the CPU requires the bus; becomes inactive when the CPU has acquired the bus and started the bus cycle.
$\overline{\text{BUS GNT}}$	61	Bus Grant	An active-low input from an external arbiter that indicates the CPU currently has the highest priority bus request. If the bus is not locked, the CPU may begin a bus cycle, commencing with the next CPU clock.
$\overline{\text{BUS BUSY}}$	60	Bus Busy	An active-low bidirectional signal used to establish the beginning and end of a bus cycle. The trailing edge (low-to-high transition) is used for sampling bits into the Fault Register. It is 3-state in bus cycles not assigned to this CPU; however, the CPU monitors the $\overline{\text{BUS BUSY}}$ line for latching non-CPU bus-cycle faults into the Fault Register.
$\overline{\text{BUS LOCK}}$	62	Bus-Lock	An active-low, bidirectional signal used to lock the bus for successive bus cycles. During non-locked bus cycles, the $\overline{\text{BUS LOCK}}$ signal mimics the $\overline{\text{BUS BUSY}}$ signal. It is 3-state during bus cycles not assigned to this CPU. The following instructions will lock the bus: INCM, DECM, SB, RB, TSB, SRM, STUB, STLB.

Table 10 **F9450 Signal Descriptions** (Cont.)

Mnemonic	Pin No.	Name	Description
Power			
V_{CC}	64	Power Supply	+5 V, 230 mA typical power supply.
GND	1, 32	Ground	0 V reference. **These pins should be tied together as close to the chip as possible.**
I_{INJ1}, I_{INJ2}	19 46	Injector Current	Current source to provide bias for the injection logic. **These pins should be tied together as close to the chip as possible.**

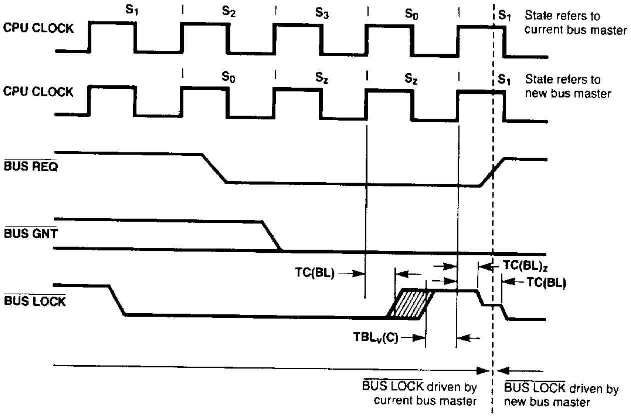
DEVICE OPERATION

Bus Transactions

Bus transactions are four or five states long (a state is equivalent to one CPU clock period). Memory and I/O cycles have identical timing requirements and are distinguished by the status of the M/$\overline{\text{IO}}$ line.

The $\overline{\text{BUS REQ}}$ output is activated during the S0 state, indicating to the external arbiter (if any) that this CPU requests the bus. At the end of state S0, the CPU samples the status of the $\overline{\text{BUS GNT}}$ and $\overline{\text{BUS LOCK}}$ inputs. If $\overline{\text{BUS GNT}}$ has been active and $\overline{\text{BUS LOCK}}$ inactive, satisfying the minimum set-up time, state S1 is entered. Otherwise, the CPU enters the high-impedance state, Sz, waiting for $\overline{\text{BUS GNT}}$ to be active (in a single processor system, no arbiter is needed and the $\overline{\text{BUS GNT}}$ pin can be wired low) and $\overline{\text{BUS LOCK}}$ to be inactive (since $\overline{\text{BUS LOCK}}$ is three-state, it should be pulled high by a resistor to V_{CC}). Refer to figure 8.

Figure 8 Bus Access Signal Requirements



S_1
S_2
S_3
S_0
S_1
State refers to current bus master
CPU CLOCK
S_0
S_z
S_z
S_1
State refers to new bus master
CPU CLOCK
$\overline{\text{BUS REQ}}$
$\overline{\text{BUS GNT}}$
TC(BL)
$TC(BL)_z$
TC(BL)
$\overline{\text{BUS LOCK}}$
$TBL_v(C)$
$\overline{\text{BUS LOCK}}$ driven by current bus master
$\overline{\text{BUS LOCK}}$ driven by new bus master

Notes:

1. All device masters are F9450 or F9450 bus-compatible devices.
2. Bus masters and external arbiter are operating on the same clock.
3. State Sz is the high-impedance state in which all new bus master drivers are in a three-state condition.

Once in S1, the $\overline{\text{BUS REQ}}$ signal is made inactive to allow bus contenders to bid early for the next bus cycle. The CPU activates the $\overline{\text{BUS BUSY}}$, $\overline{\text{BUS LOCK}}$, and status signals, and outputs the address after some delay (measured from the start of the S1 state). At the end of S1, the CPU samples the RDYA input. If low, the CPU stays in the S1 state, extending the address phase on the bus. Otherwise, it proceeds to state S2, and then to state S3.

Once in state S2, the CPU makes the STRBA signal low (this edge is used to latch the address in an external address latch) and, for Read cycles only, activates the $\overline{\text{STRBD}}$ output, where the CPU prepares to receive data by turning the address/data bus around. For Write cycles, the CPU starts driving the bus with the write data immediately after the address. The $\overline{\text{STRBD}}$ signal is activated during S3, allowing both a reasonable setup time for the write data to $\overline{\text{STRBD}}$ falling edge and a reasonable hold time for the $\overline{\text{STRBD}}$ rising edge to write data. The RDYD signal is sampled at the end of S3 and the bus cycle is terminated when RDYD is high; otherwise, it stays in the S3 state. At the end of the bus cycle, all CPU outputs are 3-state. Note that the STRBA, $\overline{\text{STRBD}}$, and IB_0–IB_{15} signals for the Write cycles are actively driven through S0 of the next cycle.

All XIO/VIO output commands and internal input commands (e.g., move contents of such internal special registers as MK to an internal general-purpose register, such as R0) are echoed back externally in the form of an I/O Write cycle. The address is the command itself, and the write data is the result of the execution phase, if applicable. The system must provide the RDYA and RDYD signals in these cycles.

Table 11 gives the typical memory subsystem access times required by the system at various operating frequencies. The access time includes address latches, address decoder delays, and system memory chip enable access.

Table 11 Typical Memory Subsystem Access Time Requirements

CPU Clock (MHz)	System Memory Address Access Time (ns)				
	No Wait	1 Wait	2 Waits	3 Waits	4 Waits
20	90	140	190	240	290
18	107	162	218	273	329
15	140	207	274	340	407
10	240	340	440	540	640

Note:

A wait state is inserted due to either the RDYA or RDYD signal being inactive when sampled by the CPU at the proper time.

TIMING CHARACTERISTICS

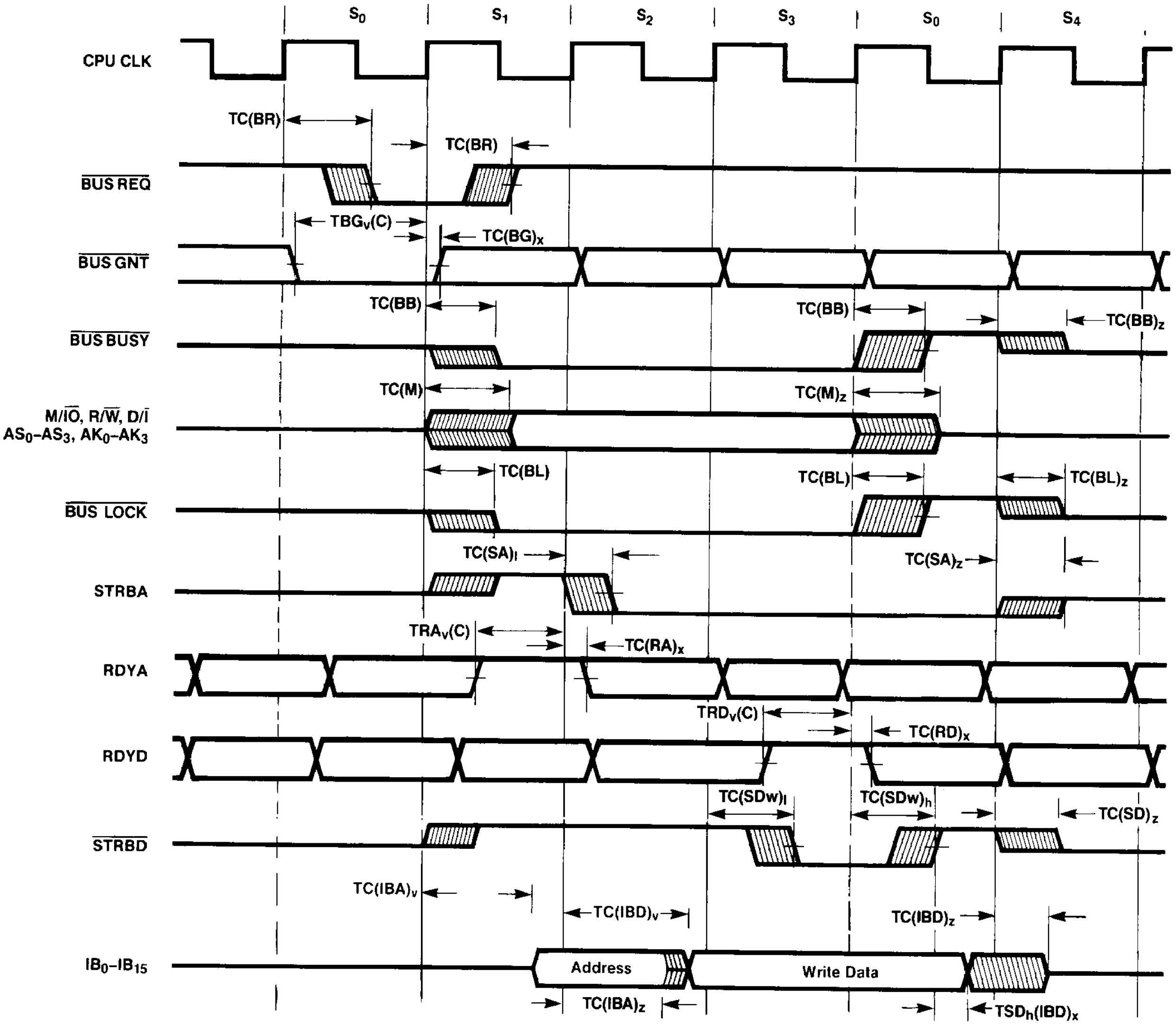
The F9450 timing characteristics are shown in figures 9 through 19 and described in table 12. A switching time test circuit is shown in figure 20.

The abbreviated symbol convention used for timing parameters in this data sheet is TAb(C)d, where:

- Timing symbols all begin with the letter "T".
- The mnemonic in the position represented by "A" indicates the signal node beginning the interval.
- The mnemonic in the position represented by "b" defines the direction of signal transition at the beginning node, if such definition is necessary; the new state of the signal may be low (l), high (h), 3-state (z), don't care (x), or valid (v).
- The mnemonic in the position represented by "(C)", which always appears in the parentheses, indicates the signal node ending the interval.
- The mnemonic in the position represented by "d" is the same as "b", but refers to the state of the signal at the node indicated by the mnemonic in position "(C)".
- The mnemonics in the positions represented by "b" and "d" are not used for reference to the CPU CLK signal, as it is assumed to be active on the rising edge.

For example, TF(BB) is the setup time from a valid fault ($\overline{\text{EXT ADR ER}}$, $\overline{\text{MEM PRT ER}}$, $\overline{\text{PAR ER}}$) input to $\overline{\text{BUS BUSY}}$.

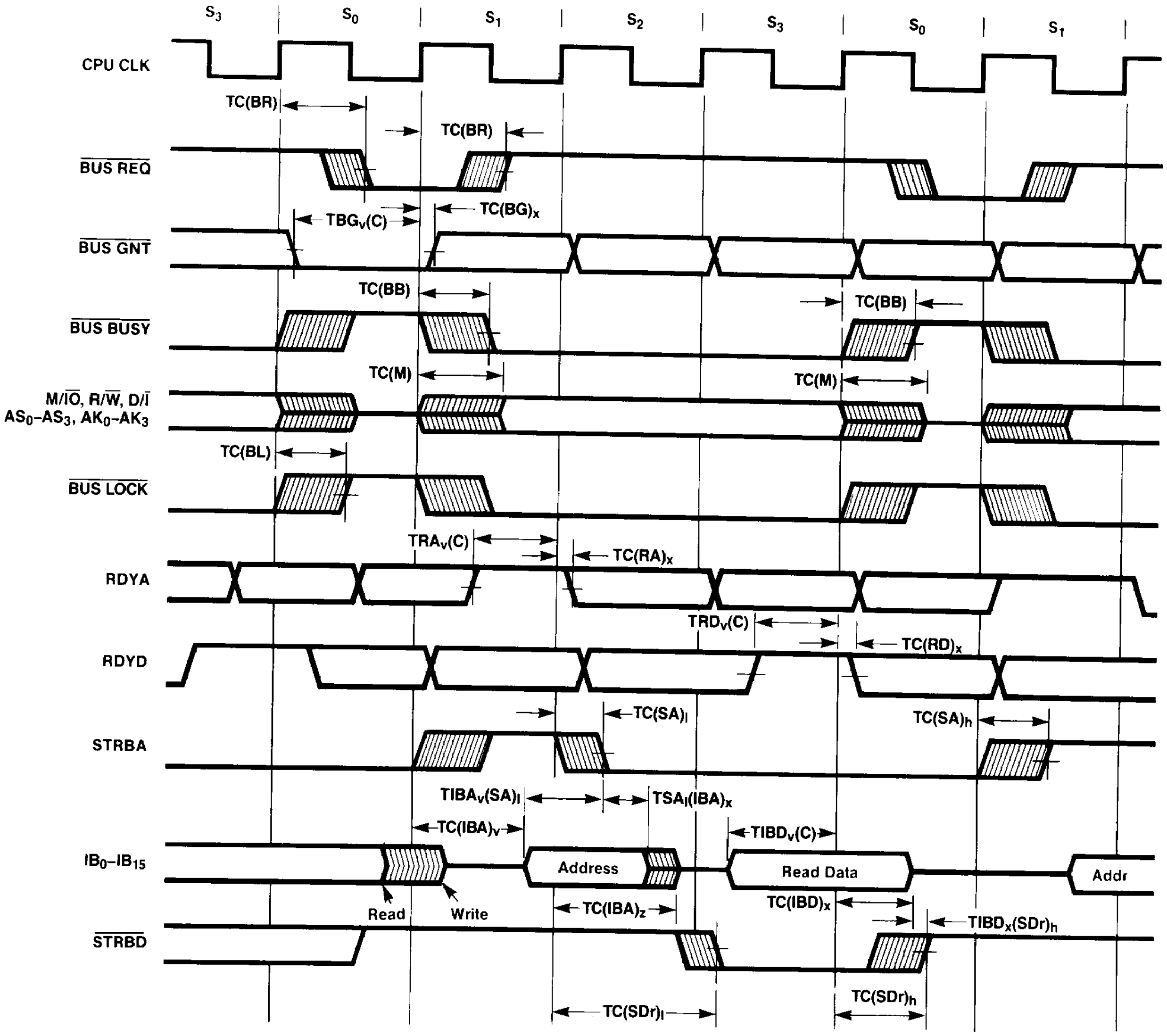
Figure 9 Minimum Write Bus Cycle Timing Diagram



Notes:

1. Shown is a single isolated bus cycle, only one bus master.
2. An intermediate level indicates that the CPU has placed this signal in a three-state condition.

Figure 10 Minimum Read Bus Cycle Timing Diagram



Note:

Shown are three consecutive bus cycles, only one bus master.

Figure 11 RDYA Signal Timing Diagram

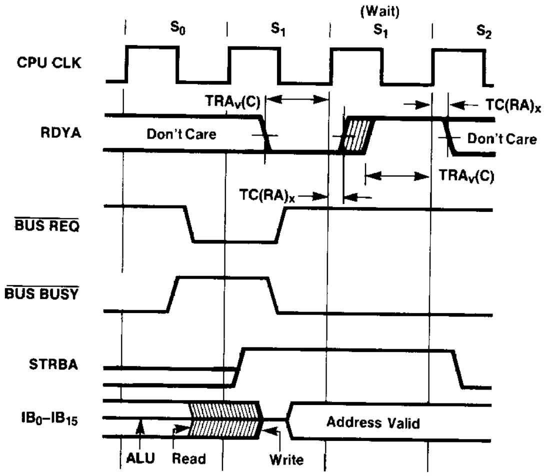


Figure 13 RDYD Signal – Write Bus Cycle Timing Diagram

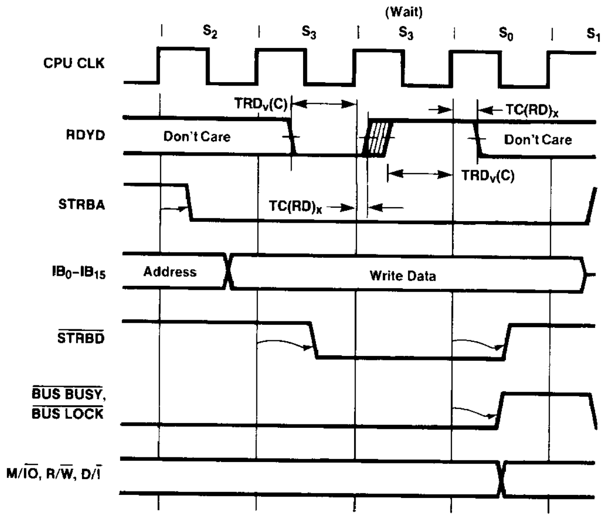


Figure 12 RDYD Signal – Read Bus Cycle Timing Diagram

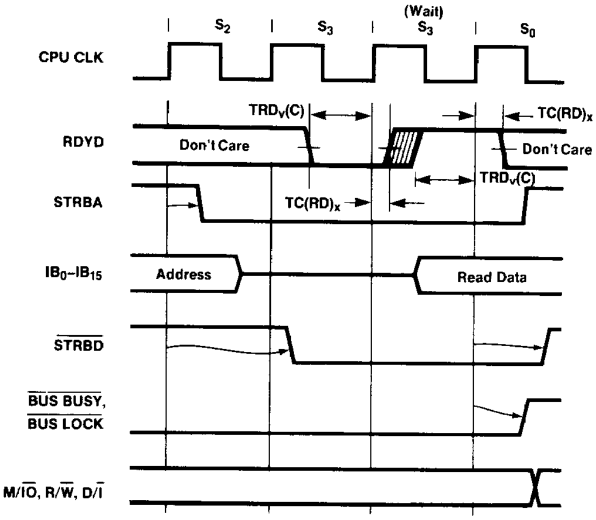
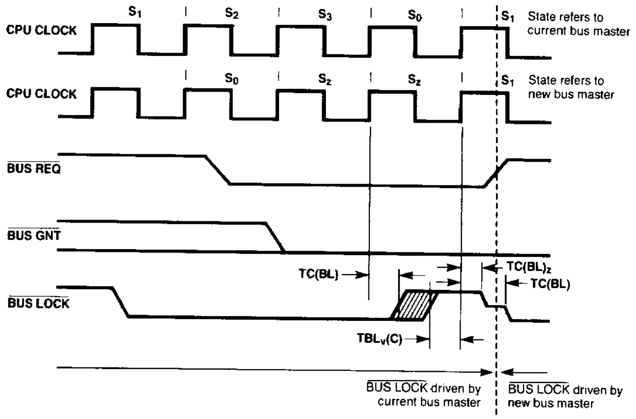


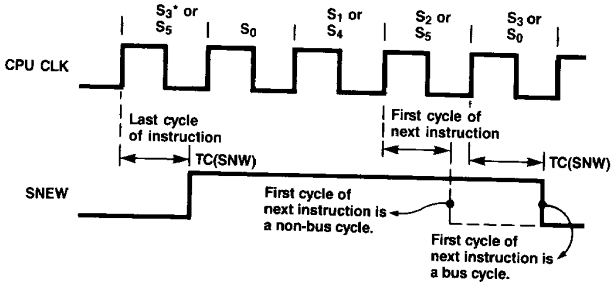
Figure 14 Signal Requirements for Accessing Bus


S_1
S_2
S_3
S_0
S_1
State refers to current bus master
CPU CLOCK
S_0
S_Z
S_Z
S_1
State refers to new bus master
CPU CLOCK
BUS REQ
BUS GNT
TC(BL)
$TC(BL)_Z$
TC(BL)
BUS LOCK
$TBL_V(C)$
BUS LOCK driven by current bus master
BUS LOCK driven by new bus master

Notes:

1. Shown is device response to external control of the BUS LOCK signal. Note that current bus master must release active drive of BUS LOCK before new bus master begins driving.
2. All device masters are F9450 or F9450 bus-compatible devices.
3. Bus masters and external arbiter are operating on the same clock.
4. State Sz is the high-impedance state in which all new bus master drivers are in a three-state condition.

Figure 15 SNEW Discrete Timing Diagram


S_3* or S_5
S_0
S_1 or S_4
S_2 or S_5
S_3 or S_0
CPU CLK
Last cycle of instruction
First cycle of next instruction
TC(SNW)
TC(SNW)
SNEW
First cycle of next instruction is a non-bus cycle.
First cycle of next instruction is a bus cycle.

*If Wait states are included in the data time of the previous cycle, this will be the first S3 state.

Figure 16 $\overline{\text{TRIGO RST}}$ Discrete Timing Diagram

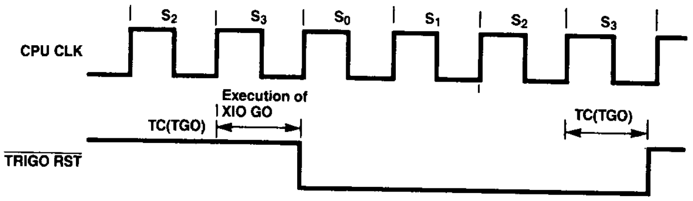


Figure 17 DMA EN Discrete Timing Diagram

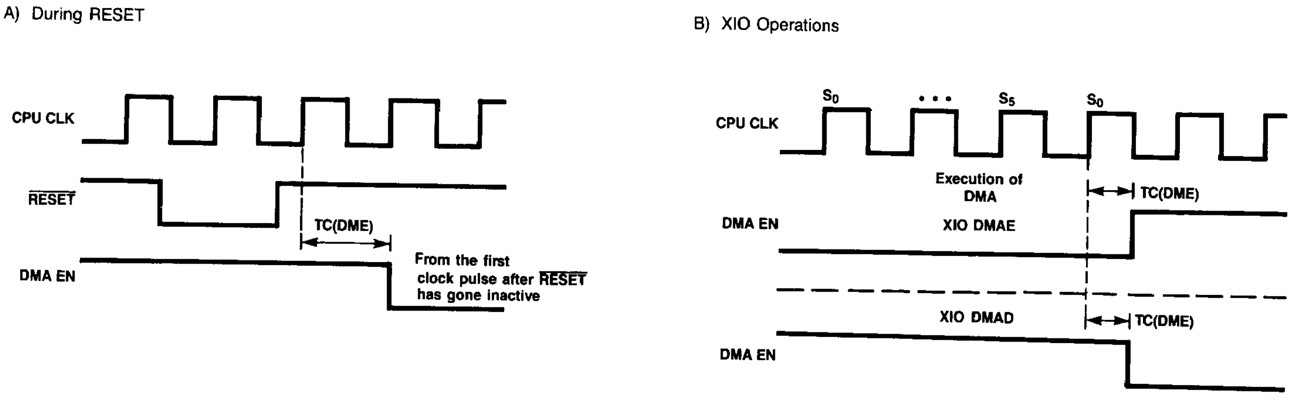
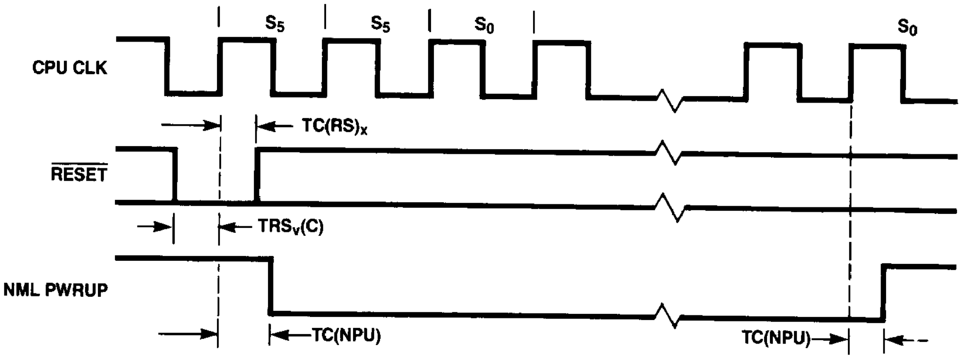


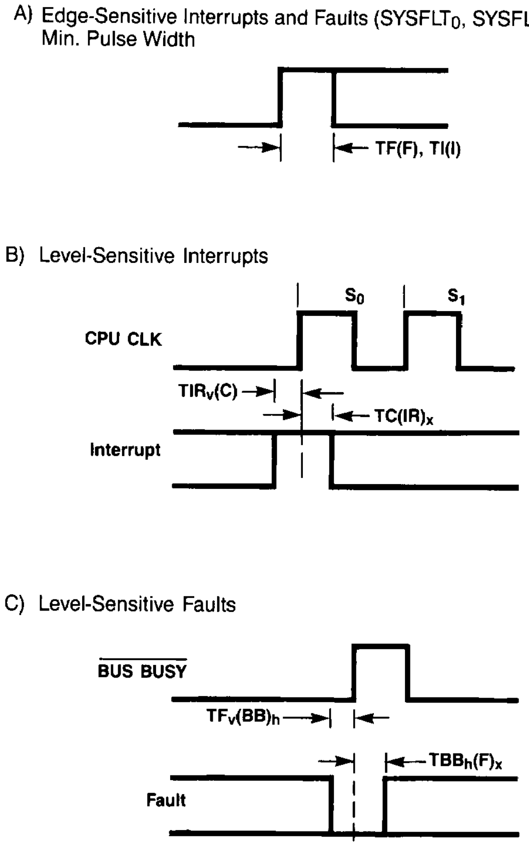
Figure 18 Normal Power Up Discrete Timing Diagram



Note:

The CPU will remain in the S5 state as long as RESET is held low.

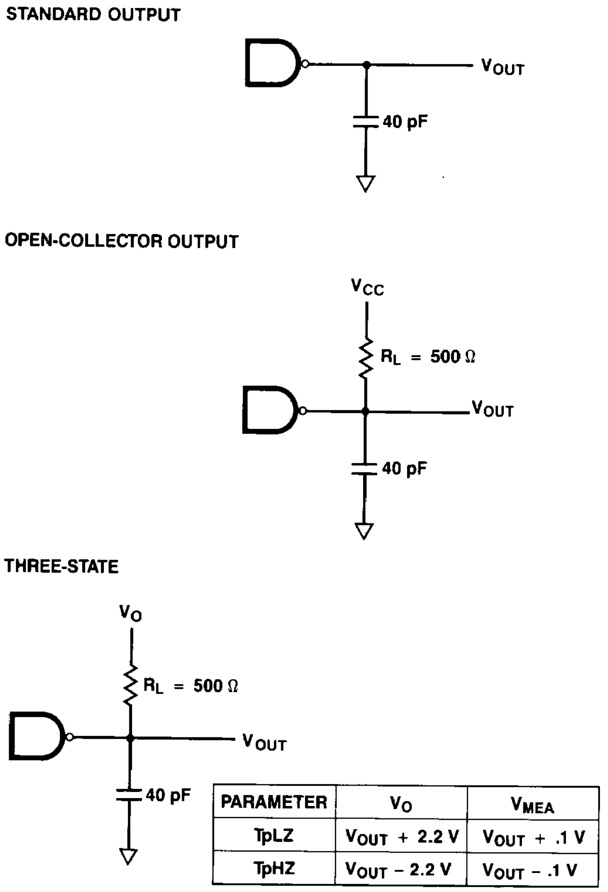
Figure 19 External Faults and Interrupts Timing Diagram


A) Edge-Sensitive Interrupts and Faults ($SYSFLT_0$, $SYSFLT_1$) Min. Pulse Width

B) Level-Sensitive Interrupts

C) Level-Sensitive Faults

Figure 20 Switching Time Test Circuits



PARAMETER	V_O	V_{MEA}
TpLZ	V_{OUT} + 2.2 V	V_{OUT} + .1 V
TpHZ	V_{OUT} – 2.2 V	V_{OUT} – .1 V

Table 12 F9450 CPU Timing Characteristics

Test Conditions:
C_L = 40 pF.
Input Conditioning: Rise and Fall Time = 6 ns, Amplitude = 0 to 3 V; Measurements taken at the 1.5 V Level.
Clock Period (CPRD) = 1/Fmax; Fmax = up to 15 MHz.
Clock Pulse Width (CPW) = 40% to 60% of CPRD.
V_{CC} = 4.75 to 5.25 V, I_{INJ} = 1.0 to 1.2 A, TC = 0°C to 85°C.

Symbol	Description	Units (ns)			Notes
		Min	Typ	Max	
Propagation Delay from Clock					
TC(M)	$M/\overline{IO}$, $D/\overline{I}$, $R/\overline{W}$, AS_0–AS_3, AK_0–AK_3 (Status)		45	60	
TC(BR)	$\overline{\text{BUS REQ}}$		35	50	
TC(BB)	$\overline{\text{BUS BUSY}}$		45	55	
TC(BL)	$\overline{\text{BUS LOCK}}$		40	55	
$TC(BL)_z$	$\overline{\text{BUS LOCK}}$, Three-State		25	40	
$TC(IBA)_v$	IB_0–IB_{15} Address		55	65	
$TC(IBA)_z$	IB_0–IB_{15} Address, Three-State		50	70	
$TC(SA)_l$	STRBA Low		15	25	
$TC(SA)_h$	STRBA High		25	35	
$TC(IBD)_v$	IB_0–IB_{15} Data		50	60	
$TC(IBD)_z$	IB_0–IB_{15} Data, Three-State		25	45	
$TC(SDr)_l$	$\overline{\text{STRBD}}$ Low (Read)		65	85	
$TC(SDr)_h$	$\overline{\text{STRBD}}$ High (Read)		30	45	
$TC(SDw)_l$	$\overline{\text{STRBD}}$ Low (Write)		35	50	
$TC(SDw)_h$	$\overline{\text{STRBD}}$ High (Write)		30	45	
TC(SNW)	SNEW		60	75	
TC(TGO)	$\overline{\text{TRIGO RST}}$		55	70	
TC(DME)	DMA EN		50	60	
TC(NPU)	NML PWRUP		60	70	
TC(ER)	Clock to MAJ ER, UNRCV ER Valid		75	100	
$TC(M)_z$	Clock to Status, Three-State		25	40	
$TC(BB)_z$	Clock to $\overline{\text{BUS BUSY}}$, Three-State		25	40	
$TC(BL)_z$	Clock to $\overline{\text{BUS LOCK}}$, Three State		25	40	
$TC(SD)_z$	Clock to $\overline{\text{STRBD}}$, Three-State		25	40	
$TC(SA)_z$	Clock to STRBA, Three-State		25	40	

Table 12 F9450 CPU Timing Characteristics (Cont.)

Symbol	Description	Units (ns)			Notes
		Min	Typ	Max	
Other Timing Parameters					
TIBA$_v$(SA)$_l$[1]	Address Valid to STRBA Low	10	30		Based on CLK period of 65 ns
TIBD$_v$(SDw)$_h$[1]	Data Valid to $\overline{\text{STRBD}}$ High (Write)	90	115		Based on CLK period of 65 ns
TSA$_l$(IBA)$_x$	Address Valid after STRBA Low Hold Time	0	15		
TM$_v$(SA)$_l$[1]	Status Valid to STRBA Low	15	35		Based on CLK period of 65 ns
TSD$_{wh}$(IBD)$_x$	$\overline{\text{STRBD}}$ Write HIgh to Data Invalid	30	70		
TIBD$_x$(SDr)$_h$	Data Don't Care to $\overline{\text{STRBD}}$ Read High	0	15		
TMEM	Memory or I/O System Time[2]	130	150		
TF(F), TI(I)	Edge-Sensitive Fault Min. PW, Edge-Sensitive Interrupt Min. PW		15	30	
Setup Time Before Clock (Unless Otherwise Specified)					
TRA$_v$(C)	RDYA		20	30	
TRD$_v$(C)	RDYD		20	30	
TIBD$_v$(C)	IB$_0$–IB$_{15}$ Data In		–5	+5	
TREQ$_v$(C)	$\overline{\text{CON REQ}}$		–10	0	
TRS$_v$(C)	$\overline{\text{RESET}}$		0	10	
TBL$_v$(C)	$\overline{\text{BUS LOCK}}$		15	30	
TBG$_v$(C)	$\overline{\text{BUS GNT}}$		20	30	
TIR$_v$(C)	Level-Sensitive Interrupt		0	10	
TF$_v$(BB)$_h$	Level-Sensitive Faults (before $\overline{\text{BUS BUSY}}$)		45	55	
TF$_v$(C)	Level-Sensitive Fault, $\overline{\text{PAR ER}}$, $\overline{\text{MEM PRT ER}}$, $\overline{\text{EXT ADR ER}}$ to Clock		5	15	
TF$_v$(C)	Level-Sensitive Fault, IOL$_1$ INT, IOL$_2$ INT to Clock		–15	0	

Table 12 **F9450 CPU Timing Characteristics** (Cont.)

Symbol	Description	Units (ns)			Notes
		Min	Typ	Max	
Hold Time After Clock (Unless Otherwise Specified)					
$TC(RA)_x$	RDYA		−10	0	
$TC(RD)_x$	RDYD		−10	0	
$TC(IBD)_x$	IB_0–IB_{15} Data In		20	25	
$TC(REQ)_x$	$\overline{\text{CON REQ}}$		30	40	
$TC(RS)_x$	$\overline{\text{RESET}}$		10	20	
$TC(BL)_x$	$\overline{\text{BUS LOCK}}$		−10	0	
$TC(BG)_x$	$\overline{\text{BUS GNT}}$		−10	0	
$TC(IR)_x$	Level-Sensitive Interrupt		5	20	
$TBB_h(F)_x$	Level-Sensitive Faults (after $\overline{\text{BUS BUSY}}$)		−20	0	
$TC(F)_x$	Level-Sensitive Fault, IOL_1 INT, IOL_2 INT		30	40	

Notes:

1. CPRD-dependent characteristic.
2. Memory or I/O system time is the elapsed time from valid address to required valid data without additional wait states. If additional time is required for either memory or I/O, it can be obtained by adding wait states in either the address or data time.

ABSOLUTE MAXIMUM/MINIMUM RATINGS

The absolute minimum and maximum ratings of the F9450 CPU are as follows:

Storage Temperature	−65°C, +150°C
Operating Temperature Under Bias	−55°C, +125°C
V_{CC} Pin Potential to Ground Pin	−0.5 V, +6.0 V
Input Voltage (dc)	−0.5 V, +5.5 V
Input Current (dc)	−20 mA, +5 mA
Output Voltage (Output High)	−0.5 V, +5.5 V
Output Current (dc) (Output Low)	+20 mA
Injector Current (I_{INJ})	1.6 A

These are stress ratings only, and functional operation at these ratings, or under any conditions beyond those indicated in this data sheet, is not implied. Exposure to the absolute rating conditions for extended periods of time may affect device reliability and exposure to stresses greater than those listed may cause permanent damage to the device.

RECOMMENDED OPERATING RANGES

Table 13 lists the recommended operating ranges for the F9450.

Table 13 **Recommended Operating Ranges**

Part Number	Supply Voltage (V_{CC}) (Volts)			Injector Current (I_{INJ}) (Amps)			
	Min	Typ	Max	Min	Typ	Max	Case Temperature (°C)
F9450DC	4.75	5.0	5.25	1.0	1.1	1.2	0 to +85
F9450DM	4.75	5.0	5.25	1.0	1.1	1.2	−55 to +125
F9450GC	4.75	5.0	5.25	1.0	1.1	1.2	0 to +85
F9450GM	4.75	5.0	5.25	1.0	1.1	1.2	−55 to +125

DC CHARACTERISTICS

The F9450 CPU dc characteristics are described in table 14.

Table 14 F9450 DC Characteristics

Symbol	Characteristic	Min	Typ	Max	Unit	Test Conditions
V_{IH}	Input High Level	2.0			V	Guaranteed Input High
V_{IL}	Input Low Level			0.8	V	Guaranteed Input Low
V_{T+}	Positive-Going Threshold Voltage USR_0 INT–USR_5 INT, $SYSFLT_0$, $SYSFLT_1$, PWRDN INT	1.5	1.8	2.0	V	V_{CC} = 5.0 V I_{INJ} = Typ.
V_{T-}	Negative-Going Threshold Voltage USR_0 INT–USR_5 INT, $SYSFLT_0$, $SYSFLT_1$, PWRDN INT	0.6	0.95	1.1	V	V_{CC} = 5.0 V I_{INJ} = Typ.
$(V_{T+})-(V_{T-})$	Hysteresis, USR_0 INT–USR_5 INT, $SYSFLT_0$–$SYSFLT_1$, PWRDN INT	0.4	0.8		V	V_{CC} = 5.0 V I_{INJ} = Typ.
V_{CD}	Input Clamp Voltage		–0.9	–1.5	V	I_{IN} = –18 mA V_{CC} = Min. I_{INJ} = Typ.
V_{OH}	Output High Voltage	2.4	3.2		V	I_{OH} = –0.4 mA V_{CC} = Min. I_{INJ} = Typ.
V_{OL}	Output Low Voltage		0.25	0.5	V	I_{OL} = 8.0 mA V_{CC} = Min. I_{INJ} = Typ.
I_{IH}	Input High Current Except IB_0–IB_{15}, $\overline{\text{BUS BUSY}}$, $\overline{\text{BUS LOCK}}$			40	µA	V_{IN} = 2.7 V I_{INJ} = Typ.
I_{IH}	Input High Current IB_0–IB_{15}, $\overline{\text{BUS BUSY}}$, $\overline{\text{BUS LOCK}}$			100	µA	V_{IN} = 2.7 V I_{INJ} = Typ.
I_{IH}	Input High Current All Inputs			1.0	mA	V_{IN} = V_{CC} Max. I_{INJ} = Typ.
I_{IL}	Input Low Current		–200	–400	µA	V_{IN} = 0.4 V V_{CC} = Max. I_{INJ} = Typ.
I_{OZH}	Output 3-State Current IB_0–IB_{15}, $\overline{\text{BUS BUSY}}$, $\overline{\text{BUS LOCK}}$			140	µA	V_{IN} = 2.4 V V_{CC} = Max. I_{INJ} = Typ.
I_{OZH}	Output 3-State Current AK_0–AK_3, AS_0–AS_3, $R/\overline{W}$, $M/\overline{IO}$, D/I, STRBA, $\overline{\text{STRBD}}$			100	µA	V_{IN} = 2.4 V V_{CC} = Max. I_{INJ} = Typ.

Table 14 **F9450 DC Characteristics** (Cont.)

Symbol	Characteristic	Min	Typ	Max	Unit	Test Conditions
I_{OZL}	Output 3-State Current IB_0–IB_{15}, $\overline{\text{BUS BUSY}}$, $\overline{\text{BUS LOCK}}$			–500	μA	V_{OUT} = 0.5 V V_{CC} = Max. I_{INJ} = Typ.
I_{OZL}	Output 3-State Current AK_0–AK_3, $\overline{AS_0\text{–}AS_3}$, R/W, M/IO, D/I, STRBA, $\overline{\text{STRBD}}$			–100	μA	V_{OUT} = 0.5 V V_{CC} = Max. I_{INJ} = Typ.
I_{OSH}	Output Short Circuit*	–15		–100	mA	V_{OUT} = 0 V V_{CC} = Max. I_{INJ} = Typ.
I_{CC}	Power Supply Current		230	350	mA	V_{CC} = Max. I_{INJ} = Typ.
V_{INJ}	Injector Voltage	1.2	1.4	1.6	V	I_{INJ} = Max.

* Not more than one output shorted at one time.

APPLICATIONS

For examples of sources of injection current, contact a Fairchild Sales Office to obtain a copy of the "Injection Current Sources" Application Note.

F9450 Address Space Implementation

The F9450 implements the direct-address memory standard, as defined by MIL-STD-1750A, using 16 address lines and a data/instruction (D/I) line that differentiates between data or instruction addresses (see figure 21). The sixteen address lines provide 64K words of address space. The additional D/I line provides 128K words of address space: 64K words of data and 64K words of instruction.

Memory Expansion

The F9451 Memory Management Unit (MMU) and F9452 Block Protect Unit (BPU) facilitate system memory expansion and provide comprehensive memory protection. The MMU does logical-to-physical address translation for a system consisting of up to 16M words of memory (1M words for MIL-STD-1750A applications). The MMU also supplies the logic that allows execute protection in the instruction space and write protection in the data space. (Refer to the F9451 data sheet for additional information.)

The BPU provides supplementary protection capability by allowing separate write protection for both CPU and DMA access. This protection is based on a 20-bit physical address for the data space. Refer to the F9452 data sheet for additional information.

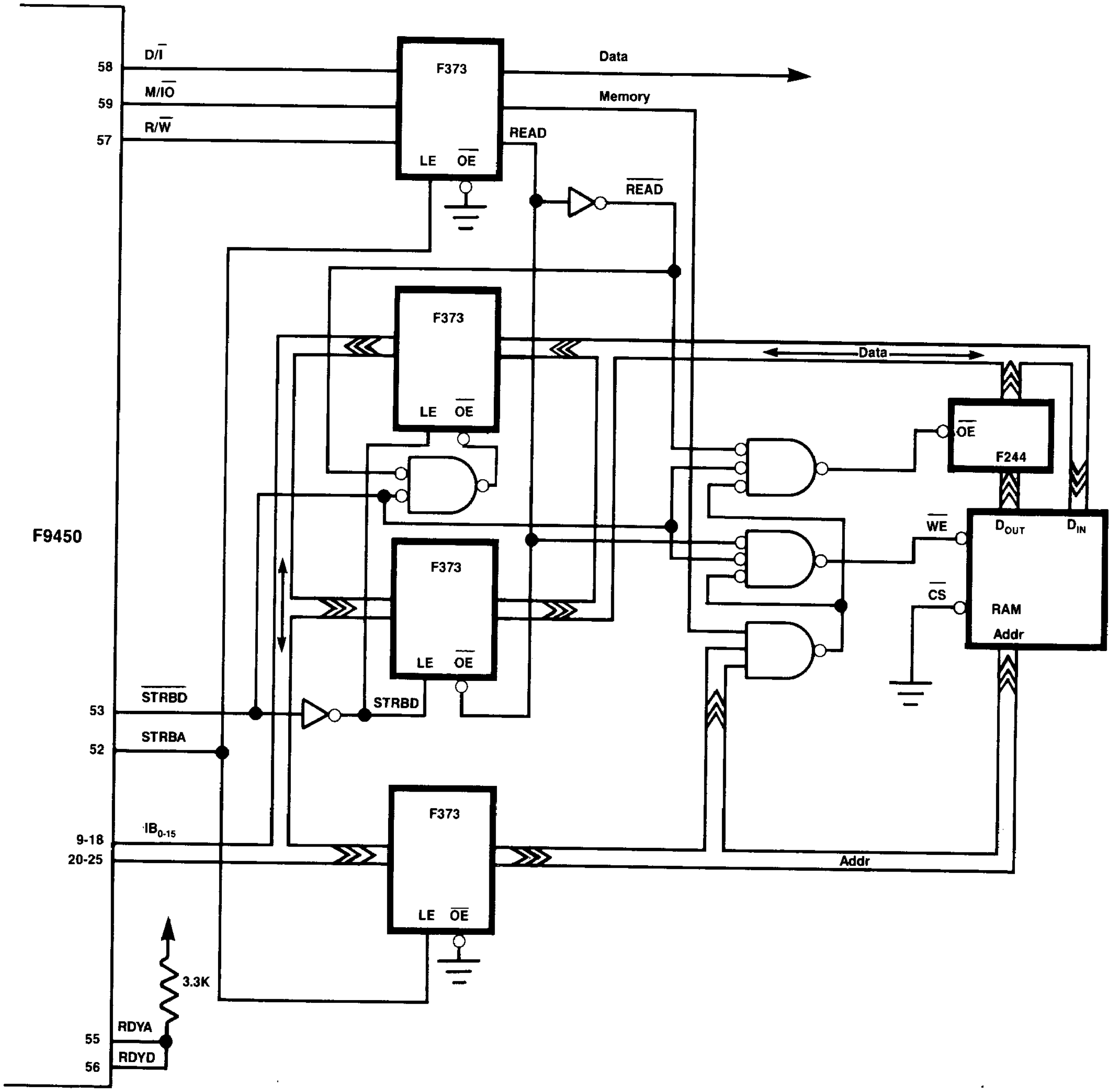
Memory Expansion without an MMU

Note: Most interpretations of MIL-STD-1750A do not permit extending address space without a memory management unit. Therefore, these memory expansion schemes require the CPU to be initialized as though attached to an MMU. Fairchild does not certify the following non-standard uses of the circuit.

In addition to the standard 17-line direct-address scheme (16 direct-address lines plus data/instruction line), MIL-STD-1750A provides four Address State (AS) lines that extend addressing space. The F9450 enables these four AS lines to be implemented directly as address lines, therefore providing a total of 21 address lines and effectively increasing the address space to 2M words: 1M word of data and 1M word of instruction.

These additional bits of address must be viewed as memory "segment" bits, and must be specifically set in the Status Word Register to enter a different segment (see figure 22). The Instruction Counter (IC) is 16 bits wide and no addressing schemes within MIL-STD-1750A will change the AS bits. Therefore, the programmer must directly manipulate the AS bits to enter a different Address State.

Figure 21 Typical Memory Subsystem Minimum Configuration



On initialization after reset, the processor performs a complete test of all major functions and initializes its internal registers in accordance with the available information. (Refer to the "Self-Test and Initialization" section for details.) Since all extensions to the normal addressing range rely on the use of the Address State (AS) bits in the Status Word Register, the F9450 must be initialized as though attached to an MMU.

Attempts to modify the AS bits without initializing the processor as though attached to an MMU result in an Address State Fault (setting bit 11 in the Fault Register). To prevent the fault, the automatic IOR at 8410 (Hex), which initializes the System Configuration Register (SCR), must contain a high (1) in bit 0. Once the SCR is initialized, the AS bits can be toggled without generating an Address State Fault. There are three methods used to change the Address state bits.

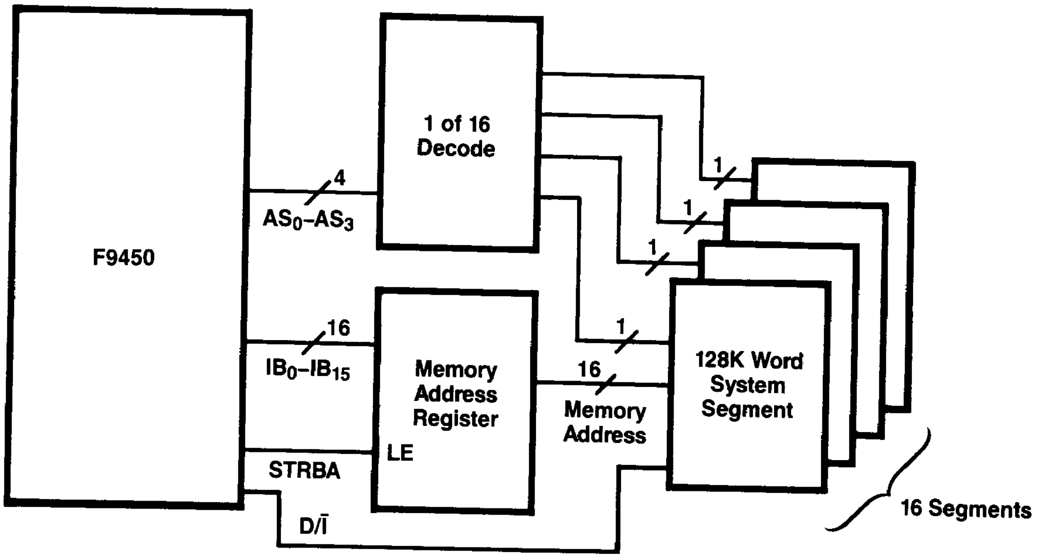
BEX Instruction — The Branch to Executive (BEX) instruction provides the best method of changing the AS bits and all pointers related to normal program execution. A BEX instruction saves the current MK, SW, and IC in locations LP, LP+1, and LP+2. The new MK, SW, and IC are retrieved from SVP, SVP+1, and SVP+2+N. The Service Pointer (SVP) is loaded from location 2B (Hex); the Linkage Pointer (LP) is loaded from location 2A (Hex). The BEX instruction automatically generates the tests for major functions and initializes the SCR, in addition to installing a new value for AS (bits 12 to 15 of the Status Word Register are AS bits). Note that interrupts are automatically disabled when this instruction is executed. During real-time applications, the programmer must reenable them. Return is best accomplished via the Load Status (LST) instruction.

XIO RA,WSW Instruction — The Execute Input/Output, Write Status Word (XIO RA,WSW) instruction transfers the contents of register RA into the Status Word Register (SWR). It contains no provisions for saving/replacing current pointers; the programmer must ensure these steps prior to instruction execution. Basically, this instruction was designed to set/clear the flags and the AK(PS) bits in the Status Register. It allows no control of the IC, which remains the same+1. Therefore, the program will continue execution in a new memory segment, but at the same location (IC) as in the old segment. This expansion scheme requires the programmer to place entry points of routines carefully, and does not allow easy program modification. Return is best accomplished via the Load Status (LST) instruction, which returns control to a known location.

LST Instruction — The Load Status (LST) method, designed to be used as an unconditional jump instruction, loads a new MK, SW, and IC from a known memory location (not the vector table), and begins execution based on these new pointers.

In the previous expansion schemes, the LST instruction is recommended for returning from the expanded memory segment. This instruction is best used to accomplish a return, as a specific location in memory could be reserved to hold return pointers, just as destination pointers are held. The programmer should note that the instruction does not save old pointers prior to the jump.

Figure 22 Typical Segmented 2M Word Memory System



Multiprocessor/Bus Arbitration Functions

A simple bus arbitration scheme uses three bus control signals:

—$\overline{\text{BUS REQ}}$ — generated by any bus master requiring the bus.

—$\overline{\text{BUS GNT}}$ — generated by the arbiter (which is sequenced by the CPU clock) to the requestor with the highest priority.

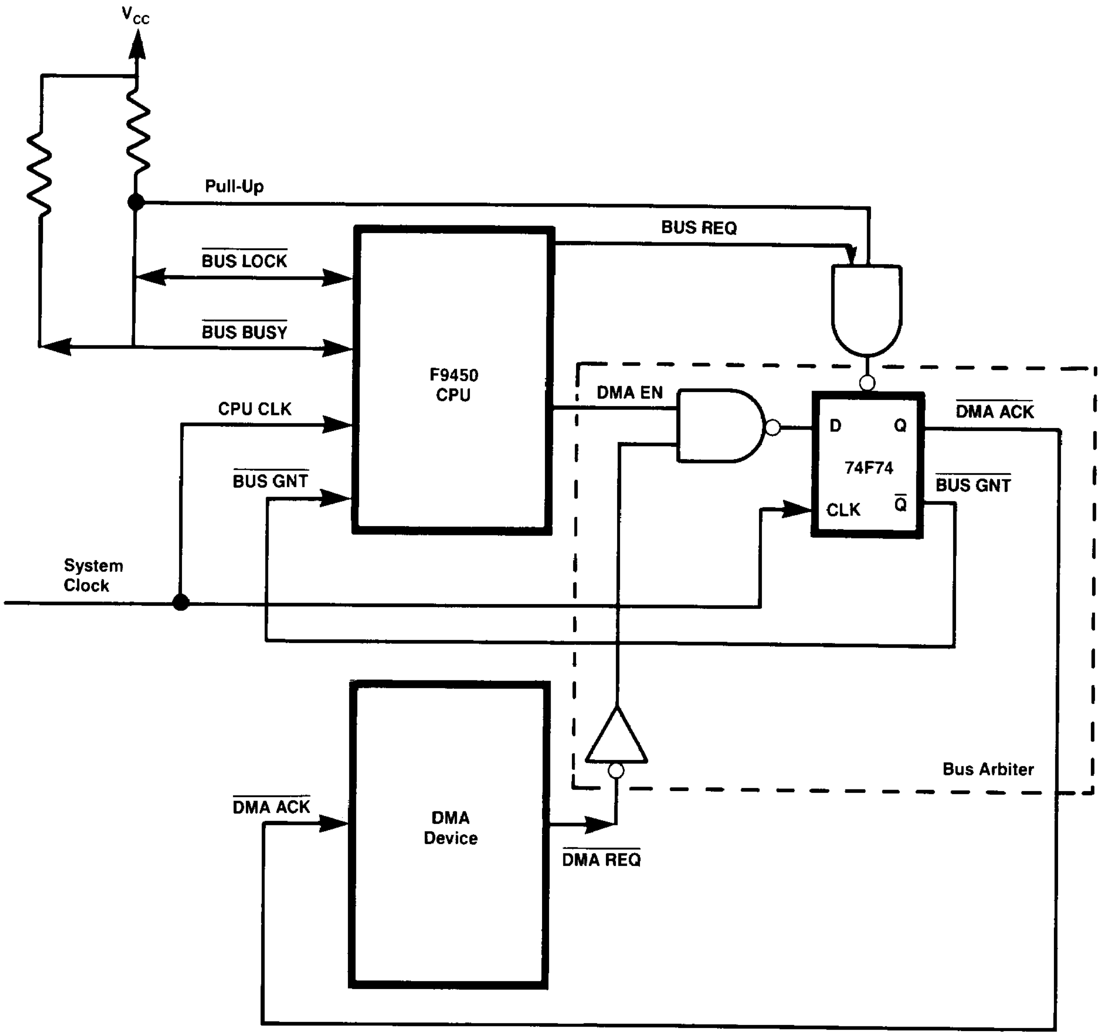
—$\overline{\text{BUS LOCK}}$ — generated by any bus master indicating that the bus is unavailable to other bus contenders.

Three priority arbitrations are considered.

1. CPU/DMA Controller Arbitration
   In a single CPU/DMA controller configuration (figure 23), the DMA device (if enabled by the CPU) has the highest priority. The CPU has access to the bus only if the DMA device doesn't request access.

2. Dual CPU Arbitration
   In a dual CPU configuration (figure 24), one is assigned a higher-priority, and the low-priority CPU has access to the bus only if the higher-priority CPU doesn't request access.

Figure 23 Single CPU/DMA Controller System with External Arbiter



3. Multiple-Bus Masters with Arbitrated Priority
   A system with eight bus masters, seven CPUs operating from the same CPU clock, and one asynchronous DMA controller is illustrated in figure 25. The external arbiter contains few TTL devices, only requiring the 54F175 register when operating at a high CPU clock frequency. The DMA device master can have its bus request to the arbiter qualified by the DMA EN output of its corresponding CPU.

One approach to multiple DMA controller configurations presents arbitration requests to the bus arbiter, causing a single DMA REQ signal to appear at the arbiter. A DMA ACK signal from the arbiter is expected and, once issued by the arbiter, the highest priority requesting DMA device can use the bus.

Because the DMA controller, in general, is not synchronous with the CPU bus masters, and since the DMA REQ signal is the highest priority bus request, the acknowledged DMA device must either keep its request (DMA REQ) active as long as it is using the bus or, after acquiring the bus, must assert the $\overline{\text{BUS LOCK}}$ signal before relinquishing its request.

Figure 24 Dual CPU with External Arbiter

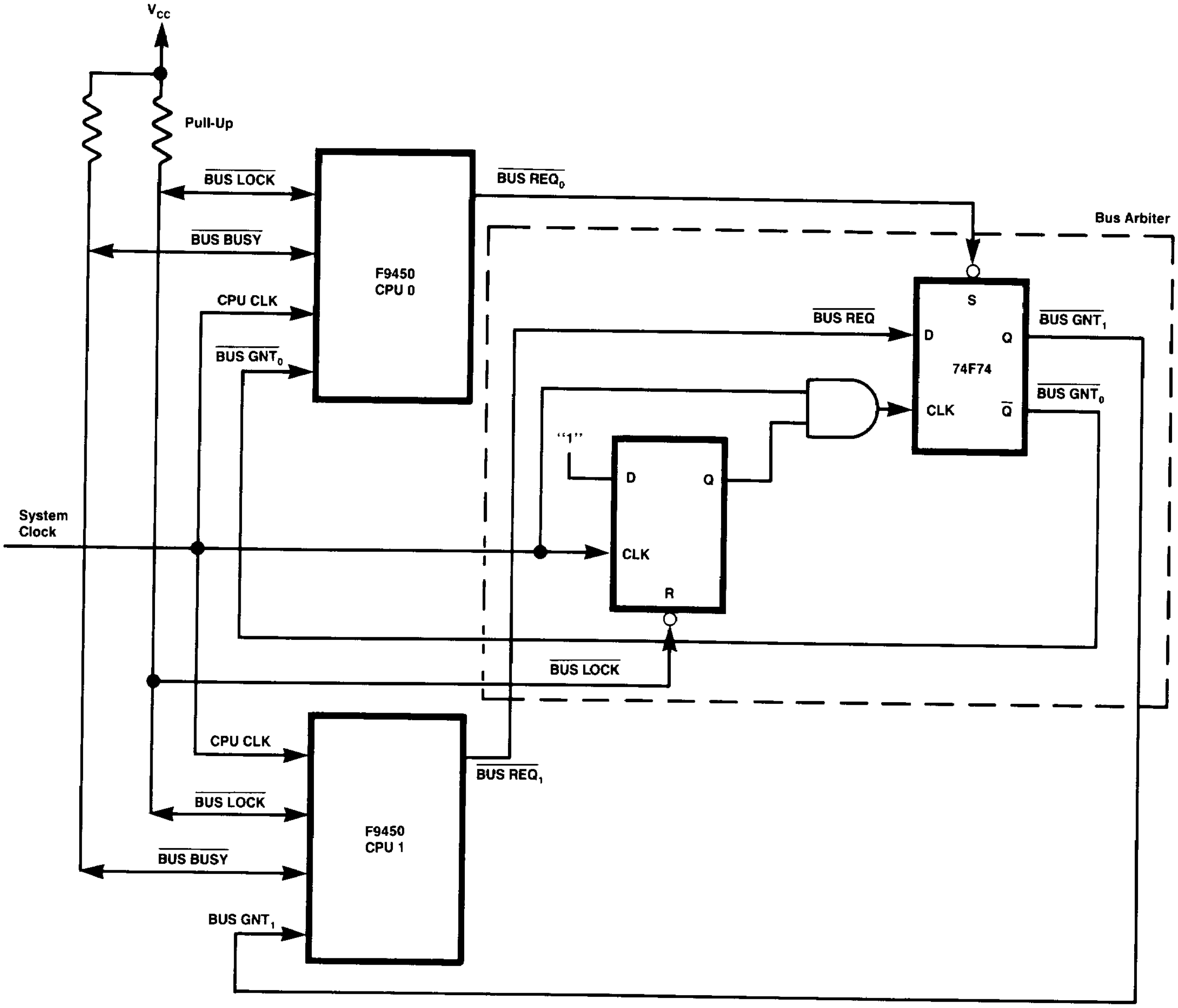
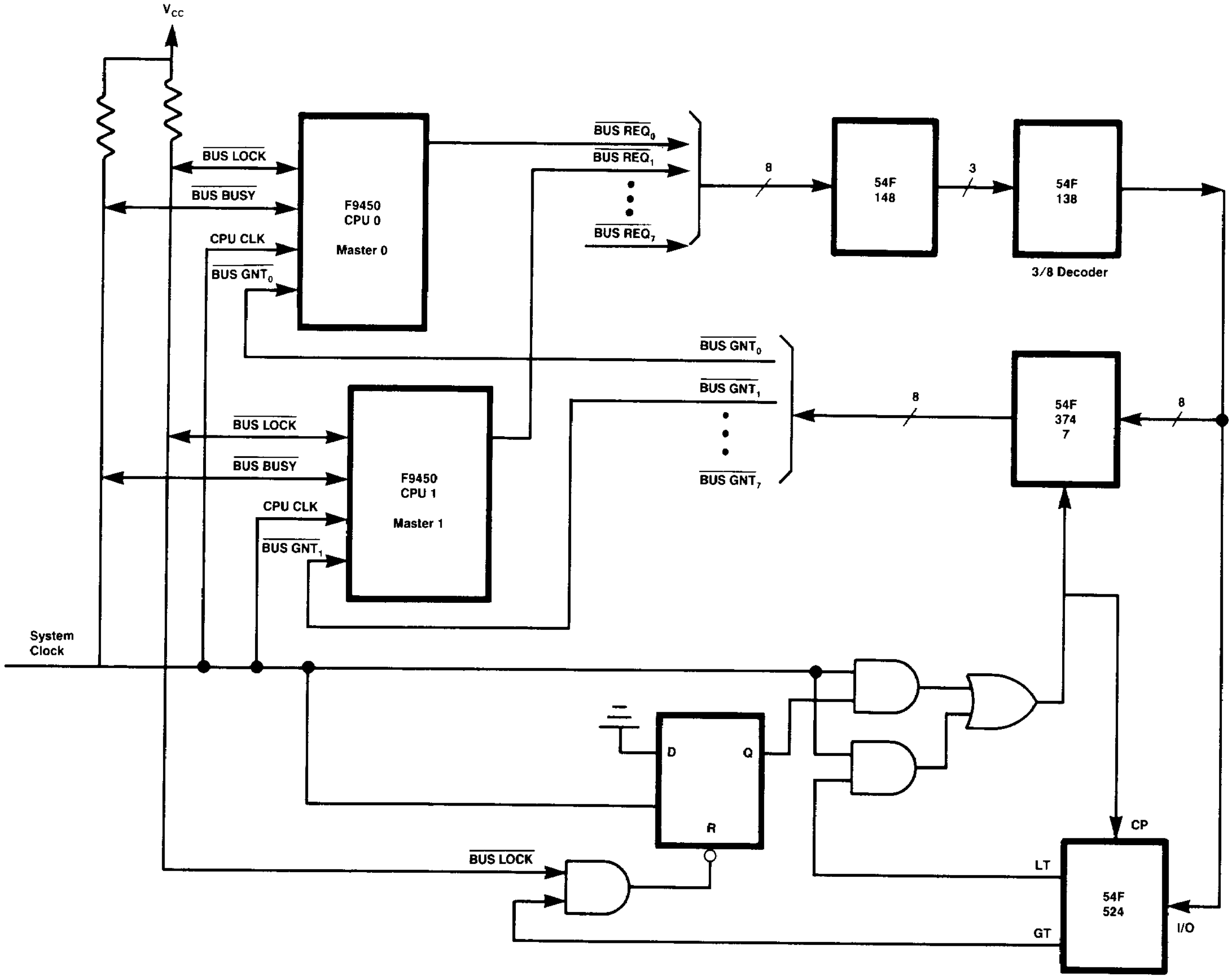


Figure 25 External Arbitration for Up to Eight Bus Masters


V_{CC}
$\overline{\text{BUS LOCK}}$
$\overline{\text{BUS BUSY}}$
CPU CLK
$\overline{\text{BUS GNT}_0}$
F9450
CPU 0
Master 0
$\overline{\text{BUS REQ}_0}$
$\overline{\text{BUS REQ}_1}$
$\overline{\text{BUS REQ}_7}$
8
54F
148
3
54F
138
3/8 Decoder
$\overline{\text{BUS GNT}_0}$
$\overline{\text{BUS GNT}_1}$
$\overline{\text{BUS GNT}_7}$
8
54F
374
7
8
$\overline{\text{BUS LOCK}}$
$\overline{\text{BUS BUSY}}$
CPU CLK
$\overline{\text{BUS GNT}_1}$
F9450
CPU 1
Master 1
System
Clock
D
Q
R
$\overline{\text{BUS LOCK}}$
CP
LT
GT
54F
524
I/O

Built-In Function Implementation with an External Coprocessor

The Built-In Function (BIF) is an escape code in the F9450 instruction set that allows user-defined instructions. The F9450 implements Built-In-Function as a three-word instruction. The MSH of word 0 contains 4F, in accordance with MIL-STD-1750A (Notice 1); the format of the remaining 40 bits is as follows:

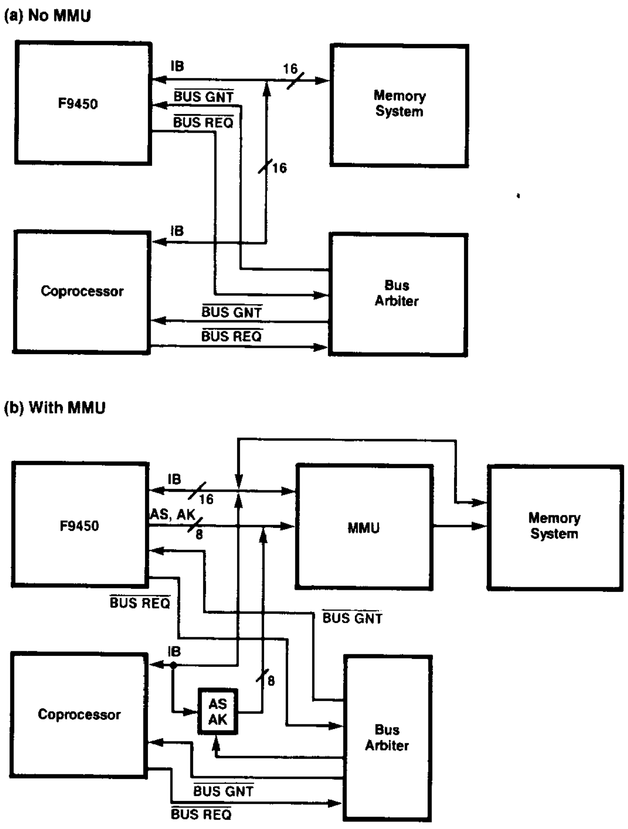
Bit No.	Description
8	*Indicates immediate (0)/direct (1) (referring to coprocessor command)
9	*Indicates two-word (0)/three-word (1) command (must be (1); if (0), bit 9 of FT is set)
10, 11	*Indicates coprocessor number (00 = 0, 01 = 1, 10 = 2, 11 = 3) (the F9450 supports four coprocessors)
12, 13, 14, 15	*Indicates Index Register number (0 through 16, 0 = no Index Register)
16 to 31	Coprocesssor command
32 to 47	Coprocessor data address

* = Bits in Opcode Extension

Use of an external user-defined coprocessor for BIF implementation is shown in figure 26.

The coprocessor receives the command word (defining the instruction) and control word via XIO operations. The operands to the coprocessor are passed from the F9450 by parameter address passing, and the coprocessor becomes a bus contender arbitrated by the bus arbiter (see figure 25). If the system includes an MMU, an additional latch is added to provide an Address State (AS) and Access Key (AK) for the coprocessor.

Figure 26 F9450/Coprocessor Configurations



(a) No MMU

(b) With MMU

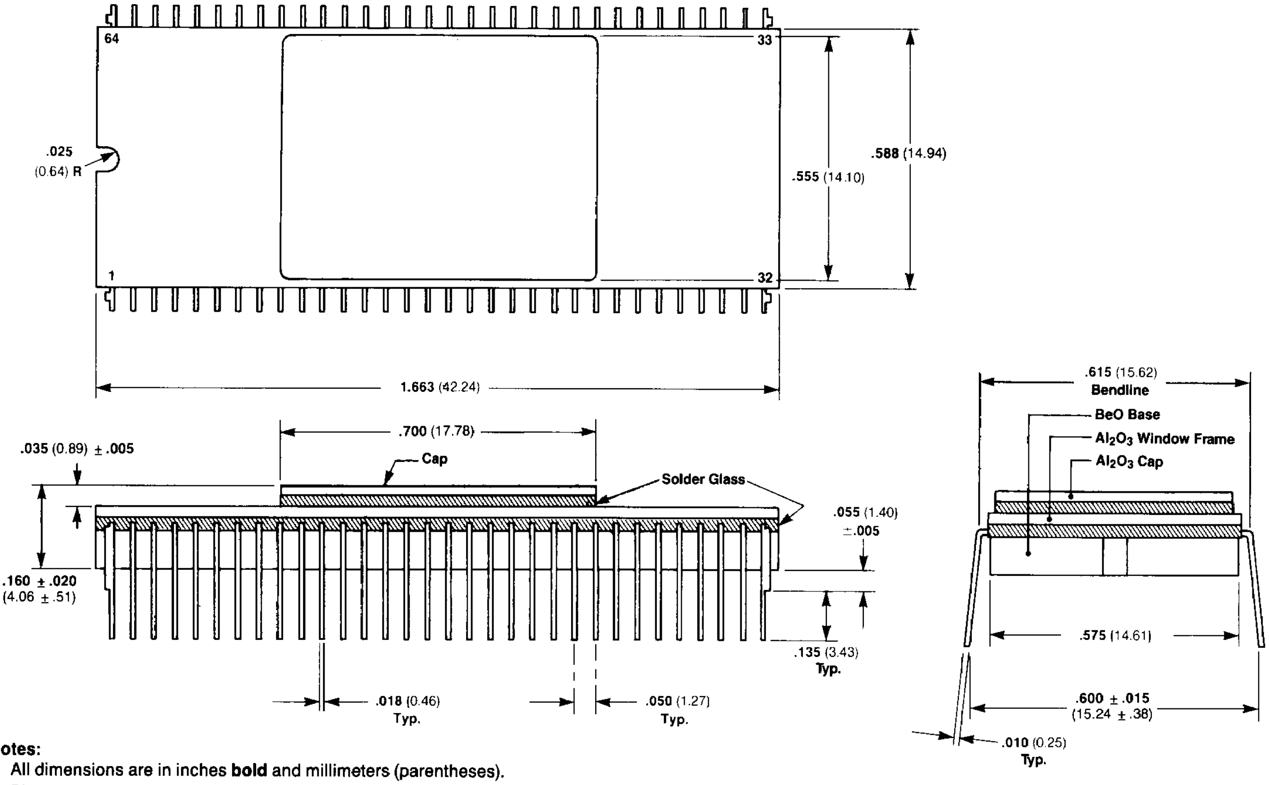
ORDERING INFORMATION

Order Code	Case Temperature (T_C) Operating Range (°C)	Package Type
F9450DC	0 to +85	Ceramic DIP
F9450GC	0 to +85	Gull Wing Surface Mount
F9450-15DC	0 to +85	Hermetic DIP
F9450-20DC	0 to +85	Hermetic DIP
F9450-15GC	0 to +85	Hermetic Gull Wing Surface Mount
F9450-20GC	0 to +85	Hermetic Gull Wing Surface Mount
F9450-15DM	-55 to +125	Hermetic DIP
F9450-20DM	-55 to +125	Hermetic DIP
F9450-15GM	-55 to +125	Hermetic Gull Wing Surface Mount
F9450-20GM	-55 to +125	Hermetic Gull Wing Surface Mount

The F9450 is available with 100 percent screening and with quality conformance inspection required by MIL-STD-883. Contact a Fairchild sales office or the Microcontroller Division for specific product information.

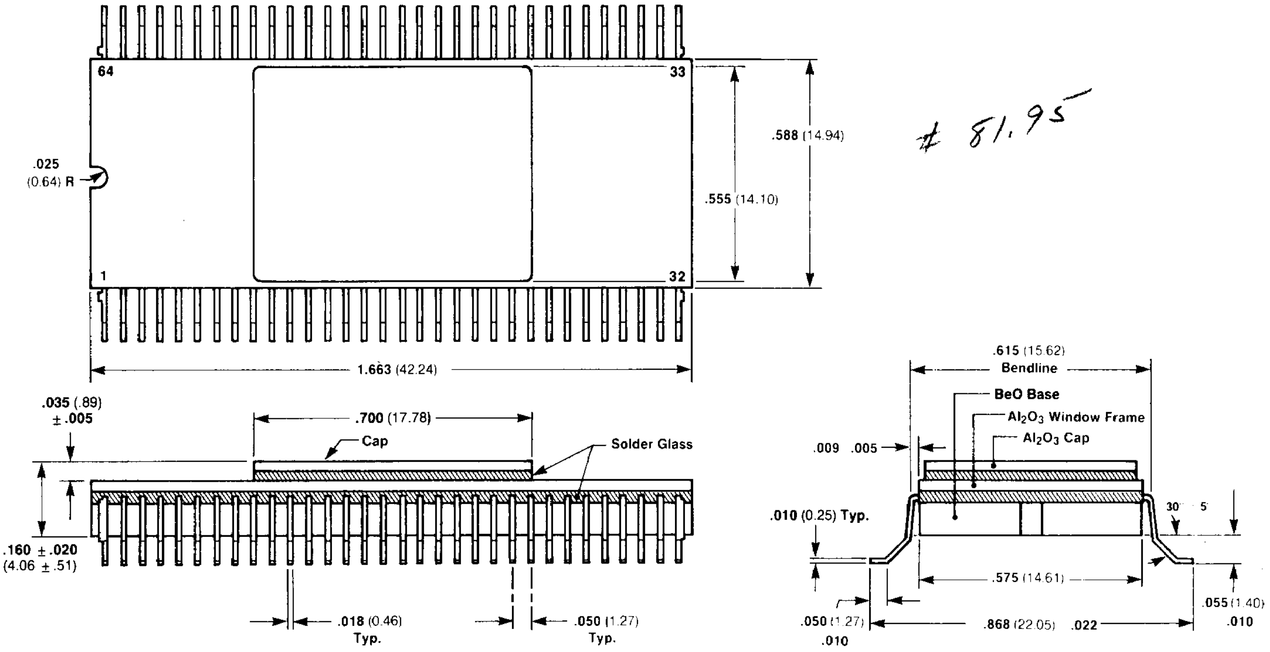
PACKAGE INFORMATION

64-Pin Ceramic Dual In-Line Package (DC or DM)



Notes:
1. All dimensions are in inches **bold** and millimeters (parentheses).
2. Pin material is tin-plated alloy 42.
3. Cap is ceramic.
4. Base is BeO.
5. Package weight is 8.5 grams.

64-Pin Ceramic Gull-Wing Dual In-Line Package (GC or GM)



81.95

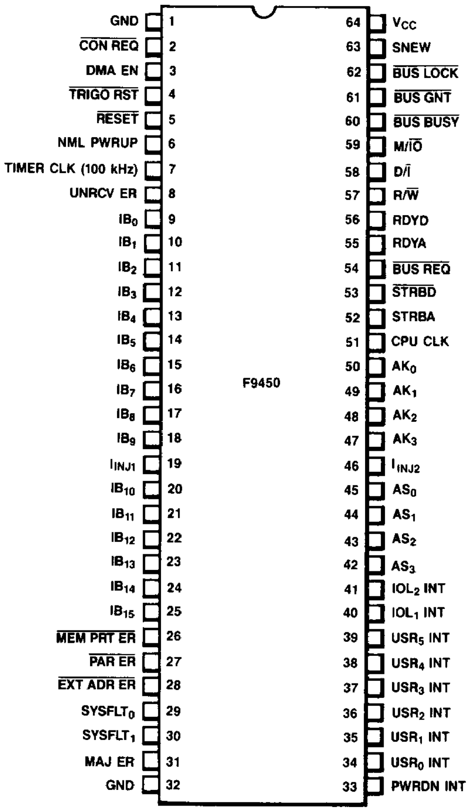
Notes:

1. All dimensions are in inches **bold** and millimeters (parentheses).
2. Pin material is tin-plated alloy 42.
3. Cap is ceramic.
4. Base is BeO.
5. Package weight is 8.5 grams.
6. Plane of lead "feet" parallel to bottom surface of BeO base ±5°.
7. Contact a Fairchild Sales Office or the Microcontroller Division to obtain a copy of the "Sockets for 64-Pin .050-Mil-Center Packages" Application Note.

Contents

Tables

Connection Diagram



Illustrations

Fairchild cannot assume responsibility for use of any circuitry described other than circuitry embodied in a Fairchild product. No other circuit patent licenses are implied.

Manufactured under one of the following U.S. Patents: 2981877, 3015048, 3064167, 3108359, 3117260; other patents pending.