FAIRCHILD
A Schlumberger Company

F9443
Floating-Point Processor

Advance Product Information

Microprocessor Product

Description

The F9443 Floating-Point Processor is designed to provide enhancement to the numeric capabilities of 16-bit microprocessors by providing a set of floating-point instructions. It can interface with the F9445, F9450 or any other standard 16-bit microprocessor, and it uses the microprocessor memory to directly fetch the required operands. It has eight general-purpose registers on-chip and supports all the basic functions with on-chip microcode. Use of additional off-chip microcode read-only memories (ROMs) provides extended capabilities. Figure 1 is a block diagram of the F9443.

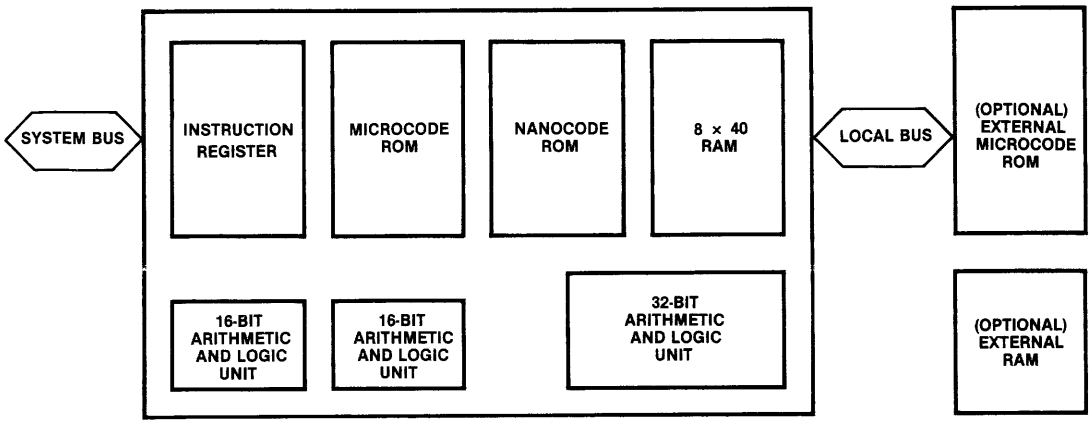
Circuit Description

The F9443 includes special hardware to provide fast algorithms for the basic functions. This hardware includes full-carry look-ahead for add and subtract (ADD/SUB) functions, recoding logic for multiply and square root functions, and partial-remainder-prediction logic for divide functions. An advanced control scheme provides a 2-level microcode/nanocode control with off-chip microcode expansion. The off-chip microcode can be programmable ROM (PROM) or random-access memory (RAM), with easy expansion for fast implementation of user algorithms.

Operation of the F9443 can proceed in parallel with the host processor to maximize throughput. Multiple F9443s can be connected to the host processor for array processing or other high-speed applications.

- **Full IEEE SP D DX Floating-Point Standard Support (80 Bits)**
- **Fast Algorithms for Add, Subtract, Multiply, Square Root, and Divide Functions**
- **Support for Full Set of Trigonometric Exponential and Logarithmic Functions**
- **Expandable Instruction Set That Can Include Macro Operations (e.g., Vector rotate, Fourier Transform, Array and Matrix Applications)**
- **User-Alterable Microcode for User Functions**
- **Support of Integer Decimal and Logical Functions**
- **Standard 64-Pin Package**
- **I^3L® High Speed Bipolar Logic**
- **Low-Power Schottky-Compatible I/O**
- **Very Fast Execution Times**
- **Interface to Any 16-Bit Microprocessor**

Figure 1 9443 Block Diagram



® I^3L is a registered trademark of Fairchild Camera and Instrument Corp.

Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

FAIRCHILD

A Schlumberger Company

F9444
Memory Management and Protection Unit

Advance Product Information

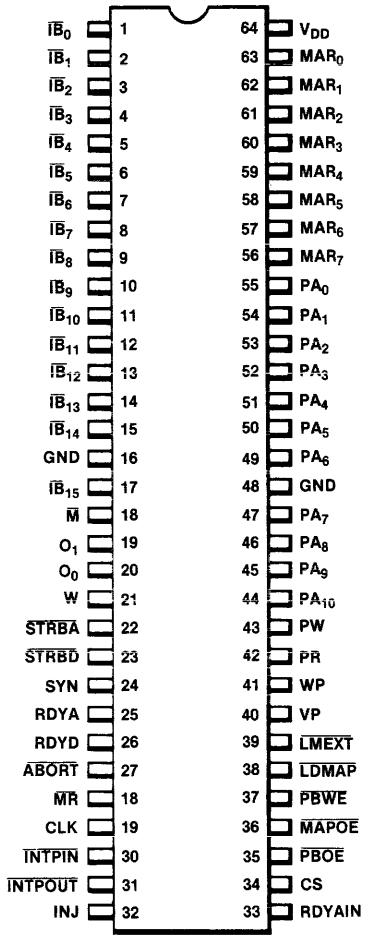
Microprocessor Product

Description

The Fairchild F9444 programmable Memory Management and Protection Unit (MMPU) is designed to support complex multi-user and large single-user environments. With four F93422 bipolar static random-access memories (RAMs) serving as map memory, the F9444 expands the physical address space of the F9445 16-bit microprocessor to 2M words by performing logical-to-physical address translation. That is, the six most significant bits (MSBs) of the logical address are translated into 11 physical address bits, leaving the 10 least significant bits (LSBs) of the logical address unchanged. The memory thus consists of 21 bits (10 LSB and 11 MSB), or 4 megabytes. System integrity is maintained by access protection bits associated with each page. Any violation causes non-maskable interrupt to the F9445 central processing unit. Page-written (PW) and page-referenced (PR) bits permit the implementation of demand-paging algorithms. Figure 1 is a functional diagram of the MMPU.

- **Standard Input/Output (I/O) Instruction Format**
- **Ability to Implement Demand-Paged Virtual Memory System**
- **Ability to Access Up to 2M Words of Memory**
- **2K Pages, With 1K Words for Each Page**
- **Memory Expansion Through Mapping and Demand Paging**
- **Controls for Memory Mapping**
- **Separate RAMs for Storing Maps**
- **Access and I/O Protection to Maintain System Integrity**
- **Special Status Bits for Read/Write Protection, Demand Paging, and I/O Protection**
- **Support for Two User and Two Data Channel Maps**
- **Low-Power Schottky-Compatible I/O**
- **Single +5 V Power Supply**
- **64 Pin Dual-in-Line Package (DIP)**
- **I^3L® Technology**

Connection Diagram



® I^3L is a registered trademark of Fairchild Camera and Instrument Corp.

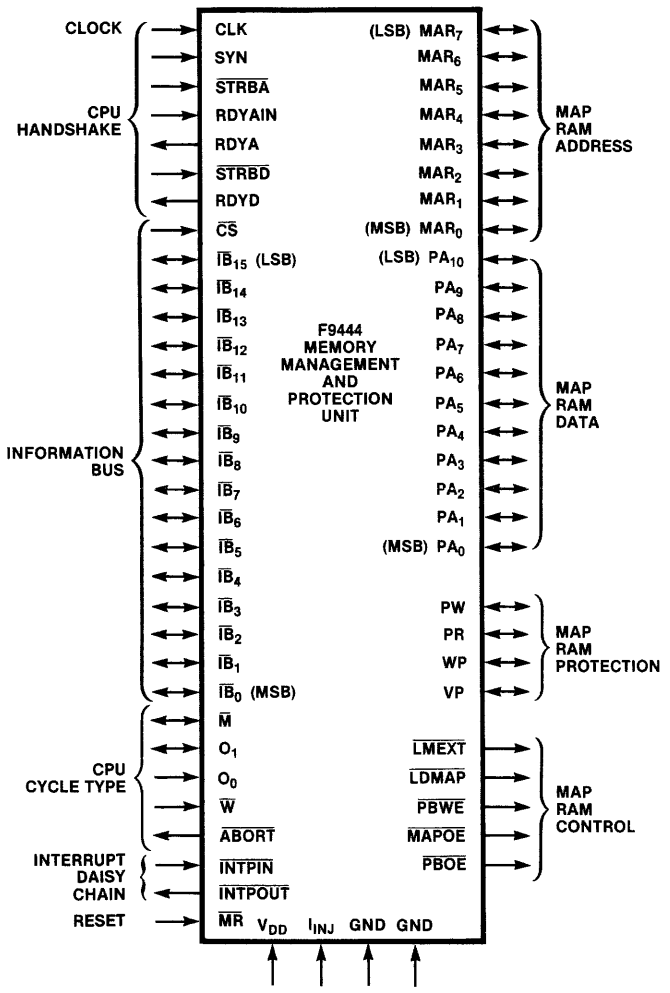
Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

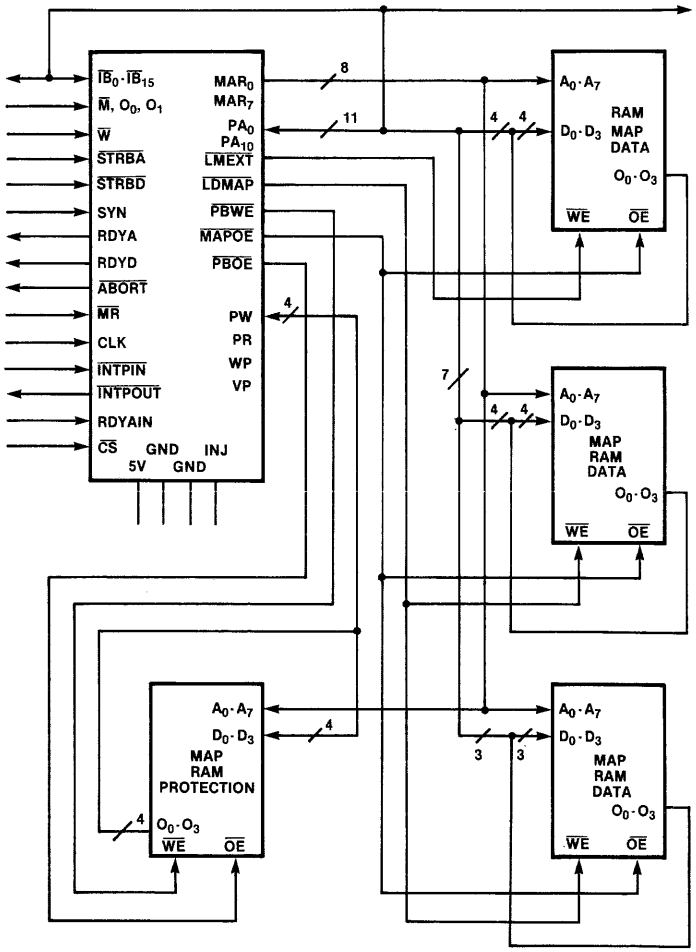
Maps

The MMPU allows two user maps and two data channel maps to reside in map memory. Each data channel map contains 32 1K-word pages and each user map either 32 or 64 1K-word pages that can be relocated anywhere in memory. The two user maps and two data channel maps function independently. Only one user map can be enabled at a time, but both data channel maps are enabled at the same time. The supervisor determines whether the mapping of program address and data channel address are to be enabled at the same time. If either user mapping or data channel mapping is disabled, the physical address space for that function is equal to the logical address space and only the lowest 64K words can be accessed.

Signal Functions

Figure 1 F9444 Functional Diagram





FAIRCHILD
A Schlumberger Company

F9445
16-Bit Bipolar Microprocessor

Preliminary Data Sheet April 1982 Microprocessor Products

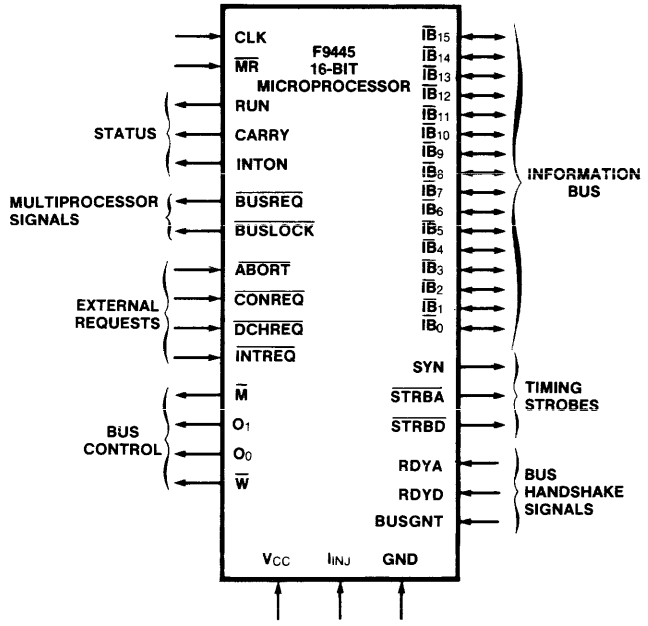
Description

The F9445 is a 16-bit microprocessor implemented using Fairchild's Isoplanar Integrated Injection Logic ($I^3L^®$) technology. This bipolar technology and a sophisticated pipeline architecture combine to give the F9445 very fast execution times. The processor has eight program-accessible registers and the capability of directly addressing 128K bytes (64K words) of memory. Up to 4M bytes of physical memory may be accessed using the F9444 memory management unit. The F9445 can address 62 I/O devices, handle 16 levels of priority interrupt, and perform fast direct memory access. It has control lines to provide operator-console functions and has an on-chip self-test program. The F9445 CPU is supported with a comprehensive family of LSI support circuits to permit cost and performance effective usage in high-performance microcomputer systems. The support circuits include the F9446 Dynamic Memory Controller, F9447 I/O Controller, F9448 Programmable Multiport Interface, F9449 Multiple Data Channel Controller, F9444 Memory Management Unit and F9470 Console Controller. It is also supported with a library of software packages, including editors, debuggers, macro-assembler, relocating loader, real-time executive, interactive multi-user disk operating system and utilities, as well as high-level languages: FORTRAN, BASIC and PASCAL.

- **Advanced Parallel Architecture Leading to Very Fast Execution Times—250 ns Register to Register, 2.9 μs 16 × 16 Bit Multiply**
- **Directly Addresses up to 128K Bytes of Memory with 11 Addressing Modes**
- **Eight Program-Accessible Registers (AC0, AC1, AC2, AC3, SP, FP, PC, PSW)**
- **Versatile Instruction Set Including Memory Reference, ALU, I/O, Stack, Multiply/Divide, and Floating Point Assist (Scale/Normalize) Instructions with 8-Bit Byte, 16-Bit Word or 32-Bit Double-Word Data**
- **Multi-Processing Capabilities**
- **Flexible Operator-Control Functions and Self-Test**
- **Static Operation with Single Clock up to 24 MHz**
- **LS TTL Input/Output Structure with I^3L Internal Circuits**
- **40-Pin DIP Needing a Single +5 V Power Supply**
- **Full Military Temperature and Voltage Ranges**
- **Radiation-Tolerant Technology**
- **Comprehensive Family of Support Circuits**

Contents

Pin Functions



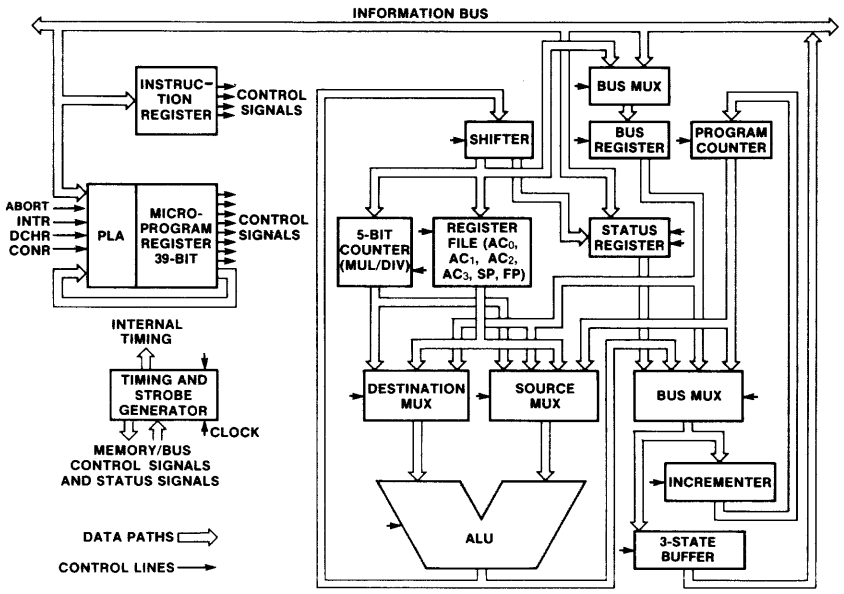
Absolute Maximum Ratings

Beyond these ratings useful life of the device may be impaired.

Storage Temperature	−65 to +150°C
Ambient Temperature Under Bias	−55 to +125°C
V_{CC} Pin Potential to Ground Pin	−0.5 to +6.0 V
Input Voltage (dc)	−0.5 to +5.5 V
Input Current (dc)	−20 to +5 mA
Output Voltage (Output HIGH)	−0.5 to +5.5 V
Output Current (dc) (Output LOW)	+20 mA
Injector Current (I_{INJ})	+600 mA
Injector Voltage (V_{INJ})	−0.5 to +1.5 V

Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

Fig. 1 F9445 Functional Diagram



Architecture

The F9445 microprocessor comprises three main blocks: the data path, the control unit, and the timing generator.

Data Path

The data path is 16 bits wide and is responsible for all the processing of data and address in the system. In many cases, data and address may be processed simultaneously.

The data path includes the following blocks (see *Figure 1*):

Register File (AC0, AC1, AC2, AC3, SP, FP)
Program Counter (PC)
Program Status Word or Status Register (PSW containing: Carry, Overflow, 32KW, ETRP flags)
Interrupt-On Flip-Flop (INTON)
Destination Mux
Source Mux
16-Bit ALU
17-Bit Shifter
5-Bit Counter (for multicycle instructions)
Bus Register Mux
Bus Register
Bus Mux and Buffer
Incrementer

Control Unit

The operations of the data path components are governed by the pipelined, microprogrammed control unit. This unit comprises three main elements (see *Figure 1*): the PLA (control store) to contain the microprogram, the pipeline register (microprogram register) to latch the microinstruction executed in the current cycle, and the instruction register to supply additional control bits during certain instructions. In addition, the control unit has a machine instruction pre-fetch mechanism which overlaps the fetching of the next instruction from memory during execution of short-cycle instructions, such as arithmetic-and-logic (ALU) instructions. This pre-fetch capability and the microprogram pipeline give the F9445 very fast and efficient instruction execution.

Timing Generator

The timing generator produces the system timings for the F9445 internal registers, memory, I/O, and console.

The clock is divided on-chip using a 3-bit twisted ring counter. The divide ratio is 6:1 or 4:1, depending on whether a short or long cycle is required. The long cycle can be extended indefinitely by lowering the inputs BUSGNT, RDYA, or RDYD. These signals hold the processor in state S1 (using BUSGNT or RDYA) or S3 (using RDYD) until the inputs are raised.

The twisted ring counter is also used to generate all the strobes by a combinational decode of its outputs and certain bits of the microprogram register.

Signal Descriptions

All F9445 inputs and outputs are TTL.

Information Bus

$\overline{IB}_0$ through $\overline{IB}_{15}$, Pins 11 through 26 — 16-bit Bus — Active LOW bidirectional; $\overline{IB}_0$ is most significant bit; address valid

with $\overline{STRBA}$ strobe; data valid with $\overline{STRBD}$ strobe; 3-state during data-channel and non-bus cycles.

Timing and Status

SYN, Pin 7 — Synchronize Output — Active every cycle; may be used for external synchronization of memory and I/O control.

$\overline{STRBD}$, Pin 6 — Data Strobe — Active LOW output; active only during memory, I/O, console, or data-channel cycles; used as strobe for data.

$\overline{STRBA}$, Pin 5 — Strobe Memory Address Register — Active LOW output; active only during normal memory cycles; not active during write portion of read-modify-write cycles (DSZ, ISZ, STB instructions and auto-increment/decrement addressing modes); used as strobe for external address register; active on I/O cycles when I/O instruction is output onto bus.

$\overline{M}$, Pin 36 — Memory or I/O Function — Active LOW output.
O_1, Pin 35 — Memory or I/O Function — Active HIGH output.
O_0, Pin 34 — Memory or I/O Function — Active HIGH output; these pins indicate the type of bus transfer as shown in the following table.

	$\overline{M}$	O_1	O_0	Function
Memory	0	0	0	Instruction Fetch
	0	0	1	Operand
	0	1	0	Indirect Address
	0	1	1	Address Save on interrupt, abort, and trap
I/O	1	0	0	Input or Output
	1	0	1	Data Channel Acknowledge
	1	1	0	Read Console Code
	1	1	1	Console Data

If a skip is taken on an arithmetic-and-logic (ALU) instruction, the next instruction is fetched but not executed. In such fetches, the $\overline{M}$ and O lines will indicate the following states.

$\overline{M}$	O_1	O_0	State Indicated
0	0	0	S0 through S4
0	0	1	S5

During machine cycles that do not use the bus, the $\overline{M}$ and O lines will be "111". $\overline{BUSREQ}$ and the bus strobes are inactive in these cycles.

$\overline{W}$, Pin 1 — Write Output — Indicates direction of data flow; HIGH indicates a read or input operation; LOW indicates a write or output operation; 3-state during data-channel cycles and short cycles ($\overline{BUSREQ}$ is HIGH).

RDYD, Pin 8 — Data Ready — Active HIGH input; used to synchronize external devices with the F9445 during data transfer; a LOW level halts the processor.

RDYA, Pin 4 — Address Ready — Active HIGH input; maintains address on bus when LOW.

RUN, Pin 37 — Run Status — Active HIGH output; LOW when in halt state.

CARRY, Pin 39 — Carry Status — Active HIGH output; copy of carry bit.

INTON, Pin 27 — Interrupt-On Status — Active HIGH output; copy of Interrupt-On flag; HIGH when interrupts enabled.

CLK, Pin 40 — Clock Input — Single-phase clock; positive-edge triggered.

Arbitration

$\overline{BUSREQ}$, Pin 38 — Bus Request — Active LOW output; indicates that a bus cycle is required; useful in multi-microprocessor system.

$\overline{BUSLOCK}$, Pin 2 — Bus Lock — Active LOW open collector output; set during read portion of read-modify-write cycles (on DSZ, ISZ, STB, and auto-increment/decrement), reset during write portion of those cycles; used in multi-microprocessor system.

BUSGNT, Pin 3 — Bus Grant — Active HIGH input; used for multi-microprocessor operation; a LOW level inhibits address output and halts the processor.

Service Request

The order of priority of requests and interrupts, from highest to lowest, is as follows: $\overline{MR}$, $\overline{ABORT}$, $\overline{DCHREQ}$, Stack Overflow Interrupt, $\overline{INTREQ}$, and $\overline{CONREQ}$.

$\overline{MR}$, Pin 33 — Master Reset — Active LOW input; a LOW level causes the processor to enter a wait state after completing the next full cycle; if that cycle is a write, it is inhibited (changed to read); sets the F9445 to 32K mode with trap enabled.

$\overline{DCHREQ}$, Pin 29 — Data Channel Request — Active LOW input; initiates data-channel cycles while LOW after current instruction. Must occur before TDRH (c).

$\overline{CONREQ}$, Pin 28 — Console Request — Active LOW input; initiates a console operation after current instruction.

$\overline{\text{INTREQ}}$, Pin 30 — Interrupt Request — Active LOW input; initiates entry to interrupt procedure, if interrupts are enabled, after the current instruction.

$\overline{\text{ABORT}}$, Pin 32 — Abort — Active LOW input; initiates abort sequence in the current microcycle.

Power

V_{CC}, Pin 31 — Power Supply — Requires +5 V.

GND, Pin 9 — Ground.

I_{INJ}, Pin 10 — Injection Current Input — Operates in 200-400 mA range at approximately 1 V; requires > 350 mA for maximum speed.

Register Set

The F9445 has eight user-accessible registers (see Figure 2), including seven 16-bit registers and a program status word (PSW) containing the following four flags: carry (bit 0), 32KW (bit 1), trap enable (bit 2), and overflow (bit 15). The carry flag (C) indicates the state of the carry bit during arithmetic and logic operations. The 32KW flag indicates whether the processor is operating in the 32K-word ("1") or 64K-word ("0") mode. The trap enable/disable flag (ETRP) indicates whether the trap instruction is enabled ("1") or disabled ("0"). The overflow flag (V) indicates twos-complement overflow in arithmetic operations.

In addition, there is an interrupt-on (INTON) flag. The CPU responds to interrupt requests from external I/O devices when the flag is set ("1"). When it is clear ("0"), all interrupt requests are ignored by the CPU. The state of the flag can be altered by the Interrupt-Enable or Interrupt-Disable instruction.

The seven 16-bit registers comprise a program counter (PC) that sequences the execution of instructions, four general-purpose accumulators (AC0 through AC3), the stack pointer (SP) and the frame pointer (FP). The program counter sequences the execution of instructions. It holds the address of the next instruction to be executed and is automatically incremented to fetch instructions from consecutive memory locations. A Skip, Jump, Jump-to-Subroutine, or Trap instruction, an interrupt generated by an I/O device or an Abort can alter the sequential execution of instructions.

The four accumulators serve as source and destination registers for 16-bit arguments in arithmetic-and-logic instructions which process the contents of the source accumulators and a base value for the carry flag and store the 16-bit result in the destination accumulator. The associated carry and overflow flags are set or cleared depending on

Fig. 2 F9445 Register Model

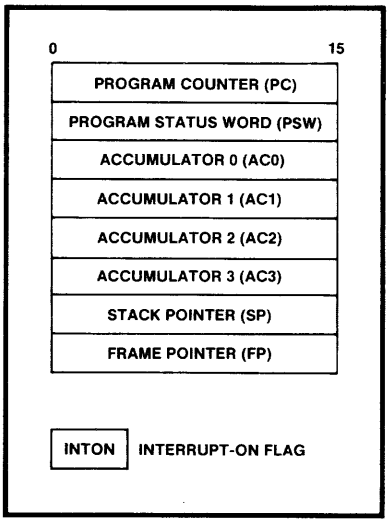
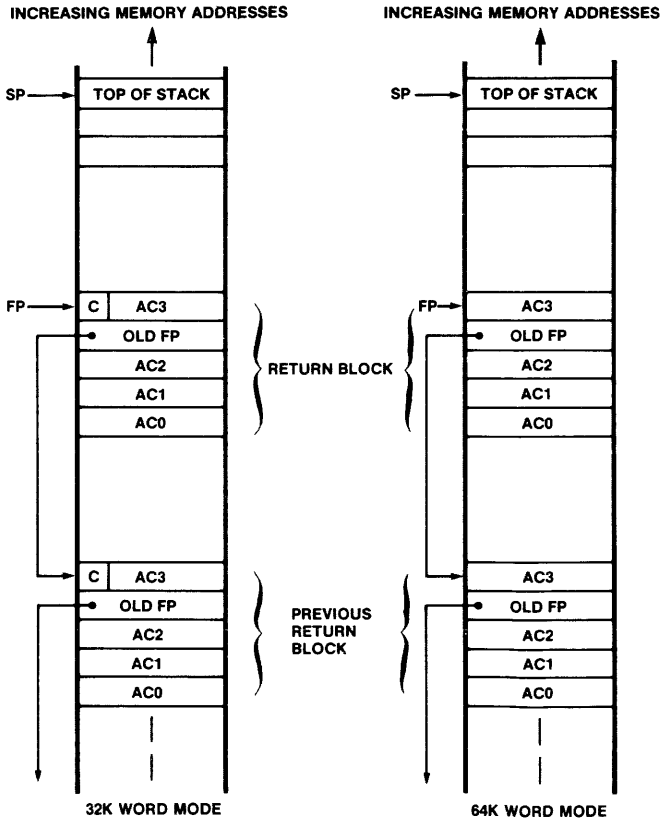


Fig. 3 Data Organization in a Stack (LIFO)



the result of the ALU operation as the base value of carry. Accumulators AC2 and AC3 also serve as index registers during memory addressing operations. In addition, AC3 functions as a subroutine linkage register, and the pair AC0 and AC1 are used as a 32-bit register in the multiply/divide and the normalize and parametric double-shift instructions.

The other two 16-bit registers serve as temporary storage and as the stack pointer (SP) and frame pointer (FP) in the stack manipulation instructions. The stack pointer contains the address of the top of the stack, i.e. the last word "pushed" onto the stack which is also the first word that may be "popped." The frame pointer contains the address of the highest location in a block of five words on the stack, a "frame," containing program status information used to return from a subroutine (see *Figure 3*).

The frame pointer is updated by the Save and Return instructions which are intended to be the first and last instructions, respectively, executed by a subroutine. When a Jump-to-Subroutine instruction is executed, the value PC+1 (and the value of the carry bit in 32K-word mode only) is stored in AC3. The Save instruction then pushes five key words onto the stack in the following order: first, the contents of AC0; second, the contents of AC1; third, the contents of AC2; fourth, the value of FP before the Save; and fifth, the contents of AC3. At this point, SP points to the top of the frame (which is the current top of the stack), and that address becomes the new value of FP. This new value of FP is also placed in AC3. When a Return instruction is executed, the five words stored in the frame referenced by FP are used to restore accumulators AC0 through AC2 to their values at the time preceding the Save. FP is restored to its previous value (pointing to the last previously saved five-word frame) and PC is loaded with the return address which had been placed in AC3 by the previous Jump-to-Subroutine and pushed onto the stack by the previous Save. The restored value of FP is also placed in AC3 by the Return instruction.

Information may also be moved between SP or FP and any of the four accumulators by the instructions MTFP, MFFP, MTSP, and MFSP without affecting the source register of the move or any of the registers not specified with the instruction. This allows setting up multiple stacks whose pointers are saved in main memory when not in use.

Addressing Ranges and Modes

The F9445 memory reference instructions support two address ranges and a variety of addressing modes. These modes include direct/indirect addressing which may be absolute, PC-relative, or indexed by AC2 or AC3. Additional addressing modes include auto-increment, auto-decrement, and address via stack and frame pointers. The two address ranges in which the F9445 can operate are 128K-byte (64K-word) or 64K-byte (32K-word) logical address space. The F9445 master resets to the 64K-byte (32K-word) address range. The 128K-byte (64K word) address range can be enabled or disabled under program control.

64K-Byte (32K-Word) Address Range

After the master reset is activated or the D64K instruction is executed, the F9445 operates in the 64K-byte (32K-word) address range. In this mode of operation, it uses 15-bit addresses to fetch up to 32K words from the memory and uses either the least-significant sixteenth bit to select high or low byte of the word in the byte instructions or the most-significant sixteenth bit to specify the remaining 15 bits of the word as an indirect address in multi-level indirect addressing instructions.

In the Load-Byte (LDB) and Store-Byte (STB) instructions, a 16-bit accumulator is specified as the byte pointer. The most significant 15 bits of the byte pointer are treated as the logical address of the word containing the byte which the least significant bit specifies, selecting the high (if "0") or low (if "1") byte of the word.

The remaining memory reference instructions specify effective addresses of 16-bit words via various (11) addressing modes described below.

Page Zero	In this mode the instruction provides an 8-bit absolute address to access the first 256 words (page zero) of memory.
PC Relative	In this mode the instruction provides an 8-bit twos-complement signed number which is added to the program counter to access 128 locations below and 127 locations above the address specified in the program counter.
Indexed by AC2 (or AC3)	In these two modes the instruction provides an 8-bit twos-complement signed number which is added to AC2 (or AC3) to access 128 locations below and 127 locations above the address specified in the accumulator.

The memory reference instruction may specify any of the above four memory addressing modes to be either direct or indirect. For direct addressing, the effective address computed using the eight address bits of the instruction is the final address of the target word to be stored or retrieved.

For indirect addressing, the effective address computed from the eight address bits of the instruction is used to fetch a 16-bit word that supplies the address of the target word. If the most significant bit of this word is "0", the 15 least significant bits provide the address of the target word. However, if the most significant bit of this word is "1", this specifies a further level of indirect address. In that case, the 15 least significant bits refer to the address of another word which could provide the final address of the target, depending on whether its most significant bit is "0" or "1". Thus, multiple levels of indirect addressing continue until a word is fetched with a most significant bit of "0". Such multiple levels of indirect addressing are only allowed in the 32K-word address range operations.

The next two types of addressing modes are the auto-increment and auto-decrement modes. When locations 20 through 27 (octal) are indirectly addressed, the auto-increment mode is activated: the contents of the specified location are first incremented and stored back and this new value is treated as the effective address (which can, in turn, be either direct or indirect). Locations 30 through 37 (octal) are used as auto-decrement locations in a similar manner.

The last type of addressing is stack addressing in which the address of the memory reference is derived from the stack pointer.

128K-Byte (64K-Word) Addressing Range

After the E64K instruction is executed, the F9445 starts operating with the 128K-byte (64K-word) addressing range. In this range, the F9445 uses 16-bit addresses to fetch up to 64K words from the memory and supports all the 11 addressing modes described previously. However, only one level of indirect addressing is allowed — the one specified in the instruction — since with 16-bit addresses there are no bits available in the words fetched to indicate further indirect addressing.

The byte pointer is also different in the 128K-byte (64K-word) case compared to the 64K-byte (32K-word) case. The 64K-word range byte pointer is 17 bits wide and is composed of the carry flag and the 16-bit accumulator specified in the LDB or STB instruction. The value of the least-significant bit of the 17-bit word selects the high (if "0") or low (if "1") byte of the word to be loaded or stored.

Instruction Set

The F9445 has fixed-length instructions, each of which is 16 bits long and divided into several fields. The fields are used to specify the operation code and other related actions, to define conditions and specify the CPU registers containing arguments, to define I/O device codes, and to provide the displacements for the calculation of effective addresses of memory locations.

The whole instruction set can be divided into five broad groups:

Memory Reference Instructions
Arithmetic-and-Logic Instructions
Stack Manipulation Instructions
I/O Instructions
Control Instructions

The Memory Reference instructions modify the contents of memory locations, alter program execution sequence, and move operands between the accumulators and memory locations. The contents of accumulators and the carry and overflow flags are processed by the Arithmetic-and-Logic instructions. The Stack instructions manipulate the registers and the memory in stack-associated operations. The I/O instructions effect data transfers between the accumulators and I/O devices. The Control instructions modify or interrogate the state of the CPU and operator console, performing such actions as controlling the status of the interrupt-on flag and reading the status of the console switch register.

The F9445 instruction set is shown on the following pages. The assembly-language format of each instruction is shown on the left, followed by the name of the instruction and a symbolic description of its action. The corresponding bit pattern for each instruction is shown on the right side of the page.

Assembly-language mnemonics and binary representations for instruction optional parts (within square brackets) and accumulator codes, to be inserted at the indicated places in the instructions, are shown following each group of instructions.

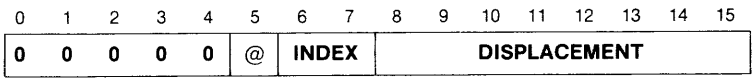
The required separator, indicated in the assembly-language formats by a square (□), may be entered as any number or combination of space or tab characters or a comma for the macro-assembler; the separator must be a single space for the F9445 PEPBUG program.

F9445 Instruction Set

Memory Reference Instructions

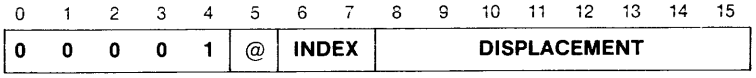
JMP□[@] displacement [,index]

Jump. Jump to effective address.



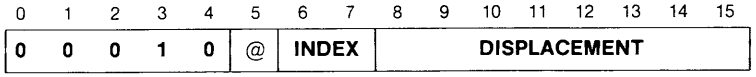
JSR□[@] displacement [,index]

Jump to Subroutine. Jump to subroutine at effective address: then return to PC saved in AC3.



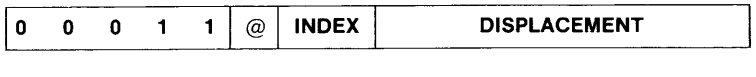
ISZ□[@] displacement [,index]

Increment and Skip if Zero. Increment (EA); if zero, skip next instruction.



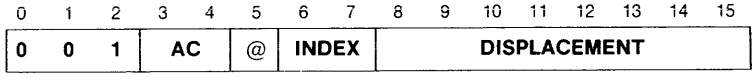
DSZ□[@] displacement [,index]

Decrement and Skip if Zero. Decrement (EA); if zero, skip next instruction.



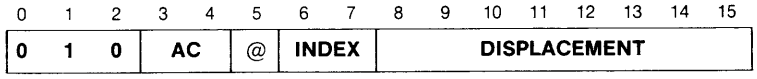
LDA□AC, [@] displacement [,index]

Load Accumulator. (EA) → AC.



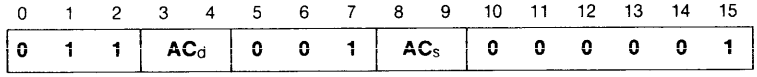
STA□AC, [@] displacement [,index]

Store Accumulator. AC → (EA).



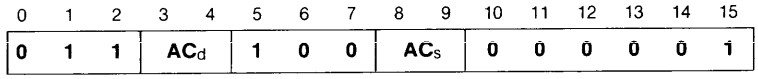
LDB□AC$_S$, AC$_d$,

Load Byte. (Byte Pointer) → AC$_{d\ 8-15}$, 0 → AC$_{d\ 0-7}$; LSB of byte pointer in AC$_S$ selects high-order byte if 0, low-order byte if 1.



STB□AC$_S$, AC$_d$

Store Byte. AC$_{d\ 8-15}$ → (Byte Pointer).



Effective Address Codes

Mnemonics @	Mnemonics Index	Bits 5	6	7	Effective Address
Omitted	0	0	0	0	EA = D; direct page zero
Omitted	1	0	0	1	EA = CA + D; relative to PC
Omitted	2	0	1	0	EA = AC2 + D; indexed by AC2
Omitted	3	0	1	1	EA = AC3 + D; indexed by AC3
@	0	1	0	0	EA = (D); indirect through page zero
@	1	1	0	1	EA = (CA + D); indirect relative to PC
@	2	1	1	0	EA = (AC2 + D); indirect relative to AC2
@	3	1	1	1	EA = (AC3 + D); indirect relative to AC3

Notes

D = Displacement; specified as absolute in current radix or relative via mnemonics.
CA = Current address or PC-1.
EA = Effective address.
(XX) = Contents of location XX, e.g. (EA) = Contents of effective address.
@ = Indirect address bit.
Byte Pointer 32K = 16 bits of AC$_S$.
Byte Pointer 64K = 17 bits of Carry and AC$_S$; upper 32K accessed when Carry = 1, lower 32K when Carry = 0
□ = Required separator.

Accumulator Codes

Mnemonic	Bits 8	9	AC
0	0	0	AC0
1	0	1	AC1
2	1	0	AC2
3	1	1	AC3

Relative Displacement

Mnemonic	Meaning
.+D	Current location plus displacement
.—D	Current location minus displacement

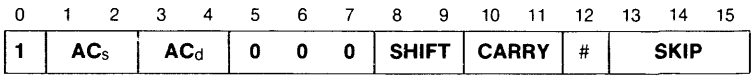
Note

D = Displacement in current radix.

Arithmetic and Logic Instructions

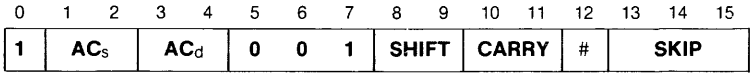
COM[carry][shift][#]□ACs, ACd [,skip]

Complement. $\overline{AC_S}$ → AC_d.



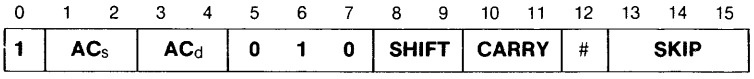
NEG[carry][shift][#]□ACs, ACd [,skip]

Negate. —AC_S → AC_d; affects Carry and Overflow flags.



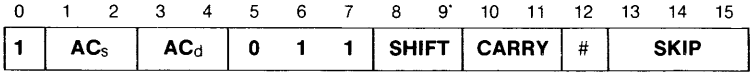
MOV[carry][shift][#]□ACs, ACd [,skip]

Move. AC_S → AC_d.



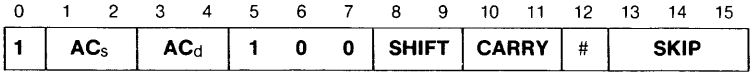
INC[carry][shift][#]□ACs, ACd [,skip]

Increment. $AC_S + 1$ → AC_d; affects Carry and Overflow flags.



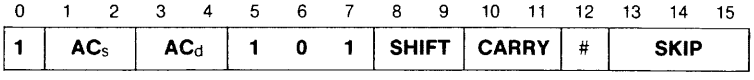
ADC[carry][shift][#]□ACs, ACd [,skip]

Add Complement. $\overline{AC_S} + AC_d$ → AC_d; affects Carry and Overflow.



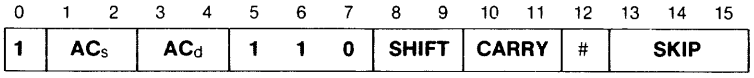
SUB[carry][shift][#]□ACs, ACd [,skip]

Subtract. AC_d — AC_S → AC_d; affects Carry and Overflow.



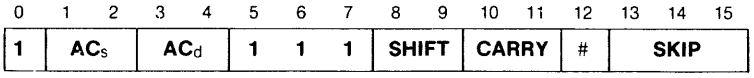
ADD[carry][shift][#]□ACs, ACd [,skip]

Add. $AC_S + AC_d$ → AC_d; affects Carry and Overflow.



AND[carry][shift][#]□ACs, ACd [,skip]

And. $AC_S \wedge AC_d$ → AC_d.



Base Carry Values

Mnemonic	Bits 10	Bits 11	Carry Value Used as Base
Omitted	0	0	Current Carry
Z	0	1	Zero
O	1	0	One
C	1	1	Complement of current Carry

Shift Operation

Mnemonic	Bits 8	Bits 9	Function
Omitted	0	0	No shift
L	0	1	Left rotate
R	1	0	Right rotate
S	1	1	Swap bytes

Load/No-Load Condition

Mnemonic	Bit 12	Operation
Omitted	0	Load result in destination accumulator
#	1	Do not load result

Skip Condition Codes

Mnemonic	Bits 13	Bits 14	Bits 15	Skip Condition
Omitted	0	0	0	Do not skip
SKP	0	0	1	Always skip
SZC	0	1	0	Skip if zero Carry
SNC	0	1	1	Skip if non-zero Carry
SZR	1	0	0	Skip if zero result
SNR	1	0	1	Skip if non-zero result
SEZ	1	1	0	Skip if either Carry or result zero
SBN	1	1	1	Skip if both Carry and result non-zero

Note

= Load/No-Load bit.

□ = Required separator.

A No-Load-No Skip instruction is interpreted as a trap if the Trap Enable flip-flop is set.

Arithmetic and Logic Instructions (Continued)

OR□AC_s, AC_d

Or. AC_s V AC_d → AC_d.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	AC_s		1	1	1	AC_d		0	0	0	0	0	1

MUL

Unsigned Multiply. AC0 + (AC1 x AC2) → AC0, AC1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	0	1	1	0	1	1	0	0	0	0	0	1

MULS

Signed Multiply. AC0 + (AC1 x AC2) → AC0, AC1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1

DIV

Unsigned Divide. (AC0, AC1)/AC2, quotient → AC1, remainder → AC0; Carry and Overflow = 1 if overflow occurs, Carry = 0 if not.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	0	1	1	0	0	1	0	0	0	0	0	1

DIVS

Signed Divide. (AC0, AC1)/AC2, quotient → AC1, remainder → AC0; Carry and Overflow = 1 if overflow occurs, Carry = 0 if not.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1

NORM

Normalize. Move the 32 bits in (AC0, AC1) to the left until high-order bit of AC0 = 1; number of steps required is subtracted from AC2; affects Overflow flag.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	1

SLLD

Shift Logically Left. Shift the 32 bits in (AC0, AC1) logically left n times; zeroes shifted to LSB of AC1; n is contents of AC_2 ($1 \leq n \leq 31$).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1

SALD

Shift Arithmetically Left. Shift the 32 bits in (AC0, AC1) to the left n times; zeroes shifted to LSB of AC1; set Overflow flag on first sign change; n is contents of AC2 ($1 \leq n \leq 31$).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1

SLRD

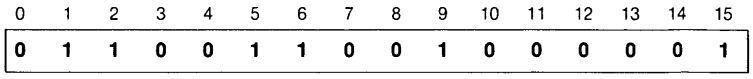
Shift Logically Right. Shift the 32 bits in (AC0, AC1) logically right n times; zeroes shifted to MSB of AC0; n is contents of AC2 ($1 \leq n \leq 31$).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	1	1	0	1	0	0	0	0	0	0	1

Arithmetic and Logic Instructions (Continued)

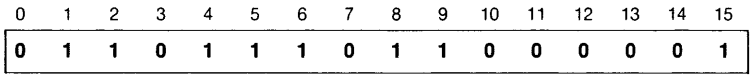
SARD

Shift Arithmetically Right. Shift the 32 bits in (AC0, AC1) arithmetically to the right n times; the MSB (sign) of AC0 is extended; n is contents of AC2 ($1 \leq n \leq 31$).



SKNV

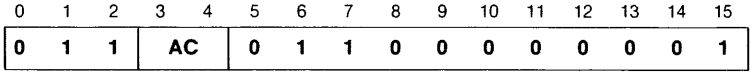
Skip on Not Overflow. Skip next instruction if Overflow = 0; then reset Overflow flag to 0.



Stack Instructions

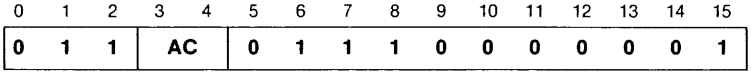
PSHA□AC

Push Accumulator. SP + 1 → SP, AC → (SP).



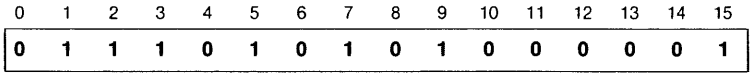
POPA□AC

Pop Accumulator. (SP) → AC, SP-1 → SP.



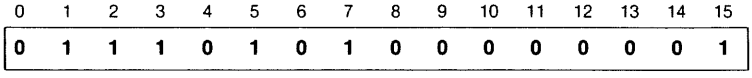
PSHF

Push Flags. SP + 1 → SP, PSW → (SP).



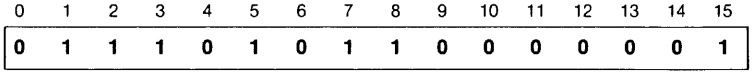
POPF

Pop Flags. (SP) → PSW, SP-1 → SP.



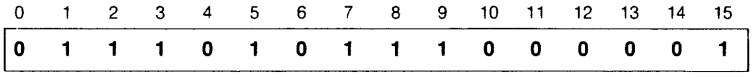
POPJ

Pop PC and Jump. (SP) → PC, SP-1 → SP.



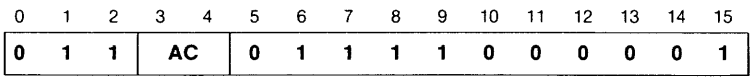
PSHR

Push Return Address. SP + 1 → SP, CA + 2 → (SP).



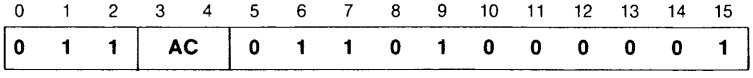
TOPR□AC

Read Top of Stack. (SP) → AC.



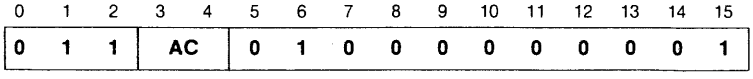
TOPW□AC

Write Top of Stack. AC → (SP).



MTSP□AC

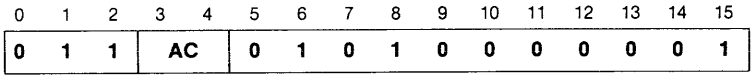
Move to Stack Pointer. AC → SP.



Stack Instructions (Continued)

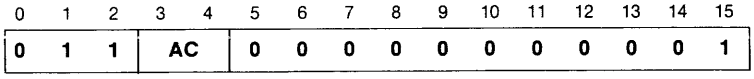
MFSP□AC

Move From Stack Pointer. SP → AC.



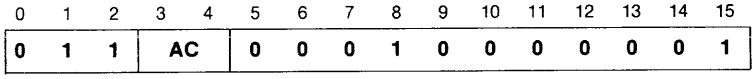
MTFP□AC

Move to Frame Pointer. AC → FP.



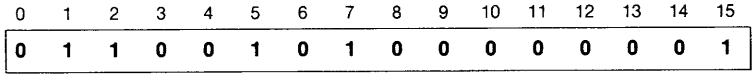
MFFP□AC

Move From Frame Pointer. FP → AC.



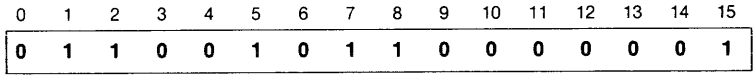
SAV

Save. Push the 5-word return block (AC0, AC1, AC2, FP |Carry*, $AC3_{1-15}$|) on stack, then load FP and AC3 with contents of SP.



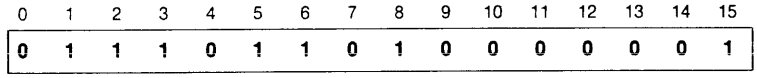
RET

Return. SP is loaded with contents of FP, then the 5-word return block is popped to (Carry*, PC_{1-15}), FP, AC3, AC2, AC1, and AC0, respectively.



DSP

Decrement Stack Pointer. SP-1 → SP.



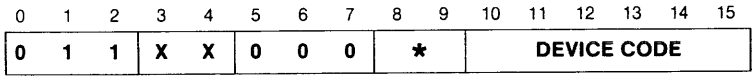
Notes

SP = Stack Pointer
FP = Frame Pointer
PSW = Program Status Word
CA = Current Address (PC-1)
*In 64K-word mode, Carry bit is not involved in SAV and RET and is replaced by $AC3_0$ and PC_0, respectively.

I/O Instructions

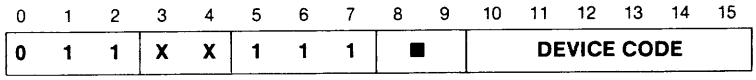
NIO[*]□device

No Data Transfer.



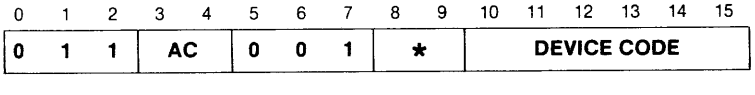
SKP■□ device

Skip on Busy/Done Flags. Skip next instruction if Busy/Done meets test condition.



DIA[*]□ AC, device

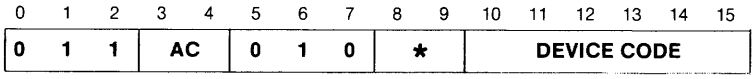
Data In From Register A. A → AC.



I/O Instructions (Continued)

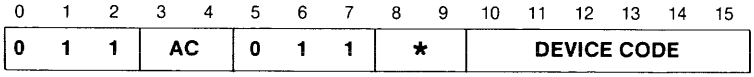
DOA[*]□ AC, device

Data Out to Register A. AC → A.



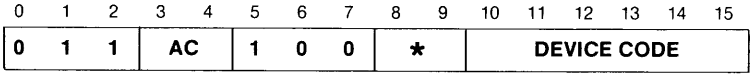
DIB[*]□ AC, device

Data In From Register B. B → AC.



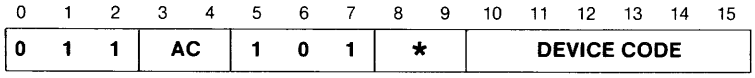
DOB[*]□ AC, device

Data Out to Register B. AC → B.



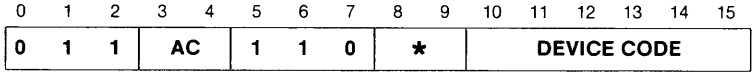
DIC[*]□ AC, device

Data In From Register C. C → AC.



DOC[*]□ AC, device

Data Out to Register C. AC → C.



Note

Device code = any number in the current radix (octal only for PEPBUG 45) between 0 and 77 octal except reserved codes 0 and 1, also may be the following standard mnemonics or other user-defined mnemonics with the macro-assembler:

* Busy/Done Control Codes

Mnemonic	Bits 8	9	Operation
Omitted	0	0	Does not affect Busy and Done flags
S	0	1	START the device by setting Busy = 1 and Done = 0
C	1	0	CLEAR both Busy and Done to 0 and idle the device
P	1	1	PULSE the device. Its effect depends on device

Notes

X = Don't care.
* = Busy/done control code.
■ = Busy/done test code.
□ = Required separator.

■Busy/Done Test Codes

Mnemonic	Bits 8	9	Test Condition
BN	0	0	Busy is Non-Zero
BZ	0	1	Busy is Zero
DN	1	0	Done is Non-Zero
DZ	1	1	Done is Zero

Device Code Symbols

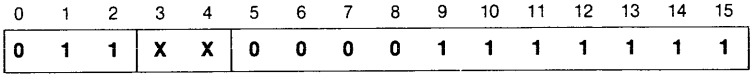
Mnemonic	Octal	Meaning
TTI	10	TTY input
TTO	11	TTY output
PTR	12	Reader
PTP	13	Punch
RTC	14	Real-time clock
LPT	17	Line printer
SMS	61	SMS disk drive
CPU	77	Console

Refer to "Control Instructions" regarding use of device code 77.

Control Instructions

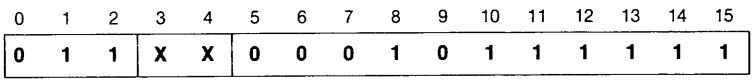
INTEN

Interrupt Enable. 1 → INTON; allows one more instruction to be executed before 1 → INTON.
Alternate assembler format:
NIOS□0, CPU



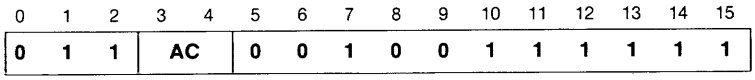
INTDS

Interrupt Disable. 0 → INTON.
Alternate assembler format:
NIOC□0, CPU



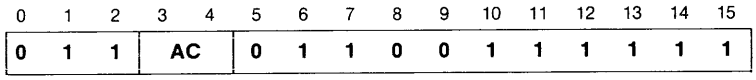
READS□AC

Read Console Switch Register. SW → AC.
Alternate assembler format:
DIA □AC, CPU



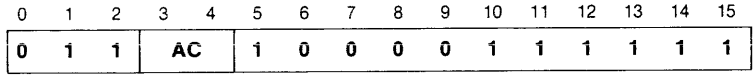
INTA□AC

Interrupt Acknowledge. The device code of the highest priority device requesting interrupt is loaded to bits 10-15 of AC.
Alternate assembler format:
DIB □AC, CPU



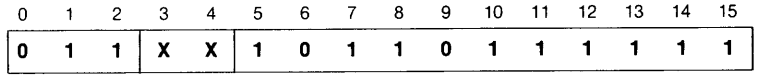
MSKO□AC

Mask Out. Enables specific devices to request interrupts.
Alternate assembler format:
DOB □AC, CPU



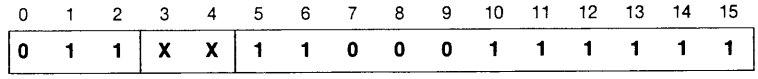
IORST

I/O Reset. Clear busy/done and interrupt enable flags of all I/O devices.
Alternate assembler format:
DICC □0, CPU



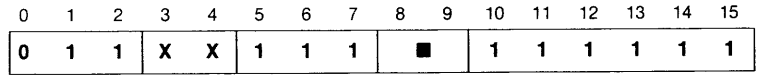
HALT

Halt the Processor. Only console operations are recognized.
Alternate assembler format:
DOC □0, CPU



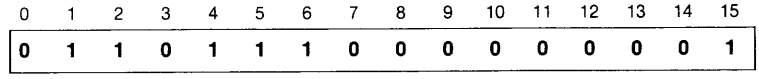
SKP■□ CPU

Skip on Interrupt-On Flag. Skip next instruction if the INTON flag fulfills the specified test conditions.



WAIT

Wait for Interrupt. Console, interrupt, and data channel request are recognized.



F9445 Instruction Set

Control Instructions (Continued)

TRAP

Trap. CA → 46_8, (47_8) → PC.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	X	X	X	X	X	X	X	X	X	X	X	1	0	0	0

ETRP

Enable Trap Instruction. Default state by $\overline{MR}$.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1

DTRP

Disable Trap Instruction.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	0	1	1	0	0	0	0	0	1

E64K

Enable 64K-words mode.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	1	1	1	0	0	1	0	0	0	0	0	1

D64K

Disable 64K-words mode. Default state by $\overline{MR}$.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1

Most of the Control instructions are a subset of I/O instructions using device code 77 octal ("all 1s," mnemonic: CPU); in device code 77 instructions, the Busy/Done control affects the Interrupt-On flag instead of the Busy and Done flags. Use of the Control mnemonics sets the Interrupt-On flag (via bits 8 and 9) as shown; however, with READS, INTA, MSKO, IORST or HALT, the alternate I/O mnemonics, which are shown for each instruction, may be used to control the Interrupt-On flag according to the "Interrupt-On Control Codes" table.

★ Interrupt-On (INTON) Control Codes

Mnemonic	Bits 8	Bits 9	Operation
Omitted	0	0	No effect
S	0	1	Set INTON = 1
C	1	0	Reset INTON = 0
P	1	1	No effect

■ Interrupt-On (INTON) Test Codes

Mnemonic	Bits 8	Bits 9	Test Condition
BN	0	0	Skip on INTON = 1
BZ	0	1	Skip on INTON = 0
DN	1	0	Reserved
DZ	1	1	Reserved

Notes
X = Don't care.
★ = Interrupt-On Control.
■ = Interrupt-On Test.
□ = Required separator.

Input/Output Operations

Input/output devices can transfer data to the F9445-based microcomputer via:

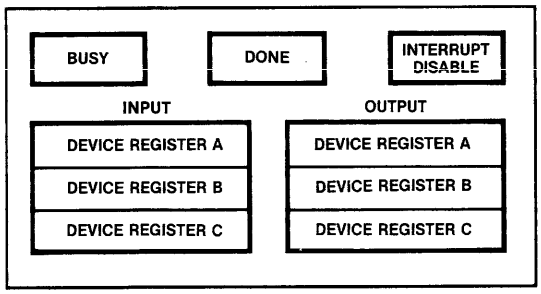
Programmed I/O using the I/O instructions of the F9445,

Memory-mapped I/O using the load/store instructions of the F9445, or

Direct memory access or data-channel transfers.

For programmed I/O, the device consists of up to three (minimum one) bidirectional 16-bit device registers, denoted as A, B, and C, and three 1-bit flags: Busy, Done and Interrupt Disable (see *Figure 4*). The 2-bit status word comprised of Busy and Done represents one of up to four possible states of the device, viz. idle, busy, partially done and completely done (refer to *Device Status Flags* subsection). The F9445 I/O instructions allow data transfers between any of the accumulators (AC0 through AC3) and any of the device registers (A through C), and can test and set the Busy, Done and Interrupt-Disable flags.

Fig. 4 I/O Device Model



The F9445 can transfer the contents of any accumulator to an I/O device by executing a Data-Out instruction. It can load data from an I/O device into any accumulator by executing a Data-In instruction. To test the status of an I/O device, the F9445 can execute a Skip-On-Status instruction.The I/O cycle has the same timing as the memory cycle (see *Figures 13* and *14*). Features of the I/O cycle are:

- 250 ns (at 24 MHz system clock) minimum cycle time
- Cycle time can be extended using RDYA, RDYD
- I/O instruction is output at address time
- $\overline{\text{STRBA}}$ is used to latch the I/O instruction
- $\overline{\text{STRBD}}$ is used to strobe the data
- O lines indicate the type of cycle as follows:

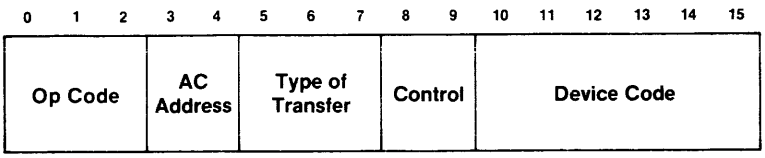
	$\overline{M}$	O_1	O_0	$\overline{W}$
I/O Input Execute	1	0	0	1
Instruction Fetch	0	0	0	1
I/O Output Execute	1	0	0	0
Interrupt Save	0	1	1	0

- The I/O devices can interrupt the normal flow of the program by using the common interrupt request line

Instruction Decode

An I/O instruction in the F9445 system comprises several fields as shown in *Figure 5*. This format accommodates data transfers between a CPU accumulator and any one of up to three bidirectional registers in any one of 62 I/O devices. Bits 10 through 15 are coded to represent device codes 00 through 76 (octal). The all "1s" device code, 77 octal, is reserved for CPU control instructions and should not be assigned to any unique I/O device; for similar reasons, device code 1 is also reserved; by convention, device code 0 is not used.

Fig. 5 Input/Output Instruction Fields



Bits 3 and 4 specify the address of any accumulator involved in an I/O instruction. When no accumulator is involved, both bits are ignored. The function bits 5, 6, and 7 define the I/O operation to be performed. Bits 8 and 9 control or test the status of the device busy and done flags.

The eight standard I/O instructions were listed previously in the *Instruction Set* section. The No-Input/Output (NIO) instruction is a "no data transfer" instruction that can be used to set the busy and done flags as required, by attaching the appropriate flag-setting mnemonic. The F9445 executes a "dummy" data out transfer. The status of a device's busy and done flags is tested by executing a Skip (SKP) instruction that causes a specific I/O device to put its busy and done flag states on lines $\overline{IB}_0$ and $\overline{IB}_1$ of the common information bus. If the flag state satisfies the condition specified by the busy/done flag-testing mnemonic appended to SKP, the CPU skips the next instruction. The remaining six standard I/O instructions first move data between an accumulator and any one of the device registers. A, B, or C. After the transfer is completed, the busy/done flags are set as specified in the I/O instruction.

There are three I/O instructions that are common to all I/O devices: Interrupt-Acknowledge, Mask-Out, and Clear-I/O-Devices. The device code for these three instructions is 77 (octal).

When the F9445 executes the I/O instruction, the $\overline{M}$ and O lines will indicate an I/O operation ("100"). The O lines are valid on the rising edge of SYN. The device address (bits 10-15) must be decoded by each device on the I/O bus. Transfers of information to and from the F9445 are timed with $\overline{STRBD}$ in the same way as the memory cycle.

At the address time, the F9445 outputs the I/O instruction on the information bus. This can be used to generate I/O signals on systems without an I/O controller. $\overline{STRBA}$ is generated and can be used to latch the I/O instruction externally. The interrupt-disable, busy and done flags organize interrupt-driven program-controlled I/O operations. The CPU controls the interrupt-disable flag. Both the CPU and the device can control the busy and done flags.

Device Status Flags

Interrupts from a device are disabled when the interrupt-disable flag of the device is set to "1". Interrupts are enabled when the flag is clear. Interrupt requests are generated whenever the device sets the done flag.

During programmed I/O, the interrupt-disable flag is normally set to disable interrupts, and the busy and done flags define the status of the device for the CPU. The busy and done flag states are coded to represent the indicated device conditions, as follows.

Busy	Done	Device State
0	0	Device idle
1	0	Device busy
0	1	Device completely done
1	1	Device partially done

The sequence of I/O transactions is normally dictated by the speed at which the device can communicate with the CPU. If the CPU operates at a higher speed than a device, it enters a wait loop between each I/O transaction with the device. During execution of the loop, the CPU repeatedly monitors the busy or done flag to determine when the device is ready for the next I/O operation.

During an output operation, one instruction stores data in the desired device register and places the device in the busy state. The CPU then enters a wait loop which terminates when the device has cleared busy and set done to signal readiness for the next output operation.

To initiate an input transaction, the device sets the done flag. One instruction reads data from the appropriate device register and places the device in the busy state. The CPU then enters a wait loop which terminates when the device has cleared busy and set done to indicate that it has the next data ready.

Interrupts

The interrupt request, $\overline{\text{INTREQ}}$, line is common to all I/O devices. When the device completes an I/O operation, it should set the done flag. Concurrently, if the device is enabled to interrupt, it should assert the active LOW on the $\overline{\text{INTREQ}}$ line. The processor responds to the interrupt request after completing execution of the current instruction. It then clears the interrupt-on flag so no further interrupts can be started, saves PC (which points to the next instruction) in location 0, and executes a "jump-indirect-to-location-1" instruction to jump to the interrupt service routine. Location 1 should contain the address of the interrupt routine or an indirect address to the routine. The F9445, when interrupted, can check for the source of the interrupt in two ways:

It can test the state of the done flags in the various devices, one by one, by executing Skip-on-Done instructions; or

It can test the state of the I/O devices by executing the Interrupt-Acknowledge instruction, causing the device that had sent an interrupt request to respond by placing its device code on bits 10 through 15 of the information bus.

As several devices can request interrupt simultaneously, device priority may be established in a daisy-chain fashion by a physical connection of a serially propagated signal, Interrupt Priority. The first device requesting an interrupt and having its Interrupt-Priority-In line HIGH has priority, and it answers the Interrupt-Acknowledge instruction, at the same time blocking the propagation of the interrupt-priority signal by putting its Interrupt-Priority-Out line in a LOW state.

The interrupt-priority signal is generated in the device having the highest priority. The F9445 can disable the interrupt system in each I/O device by placing a mask on the information bus while executing the Mask-Out instruction.

Each bit in the mask is assigned to a specific device. When that bit is "1", the interrupt system is disabled. A "0" in that bit enables the device.

After servicing a device, the routine should restore the pre-interrupt states of the accumulators and carry, turn on the interrupt, and jump to the interrupted program. The instruction that enables the interrupt sets interrupt on (INTON), but the flag has no effect until the next instruction begins. Thus, after the instruction that turns the interrupt back on, the processor always executes one more instruction (assumed to be the return to the interrupted program) before another interrupt service can start. If the service routine allows interrupts by higher priority devices, the routine should turn off the interrupt, before dismissing as indicated above, to prevent further interrupts during dismissal. In dismissing, the routine should re-enable lower priority devices.

The interrupt request input $\overline{\text{INTREQ}}$ is negative-level sensitive and is synchronized in the processor. Externally, interrupt requests may be latched with the leading edge of SYN. The interrupt request may be reset by the external I/O controller from a decode of the I/O instruction INTA.

The F9445 recognizes two other types of interrupts:

Abort Interrupt — This is activated by the active LOW of the $\overline{\text{ABORT}}$ input. The processor responds by:

Aborting the instruction being executed,
Storing the address of the aborted instruction in location 46 (octal), and
Jumping indirect to location 47 (octal).

Stack Overflow interrupt—This is an internal interrupt caused when the stack overflows; i.e., when a stack operation (PSHA, PSHF, PSHR, SAVE, TOPW) writes over a page boundary (mod 256). This interrupt is of higher priority than the external interrupt ($\overline{\text{INTREQ}}$); the processor responds, at completion of the current instruction by:

Clearing the interrupt-on flag (to "0"),
Storing the updated program counter in location 0, and
Jumping indirect to location 3 (octal).

The interrupt-save cycle follows the interrupt. It can be externally detected by the code "011" on the O lines and used, for example, to switch an external mapper to non-mapped mode.

The order of priority of requests and interrupts, from highest to lowest, is as follows: $\overline{\text{MR}}$, $\overline{\text{ABORT}}$, $\overline{\text{DCHREQ}}$, Stack Overflow Interrupt, $\overline{\text{INTREQ}}$, and $\overline{\text{CONREQ}}$.

Data Channel
The data channel has three methods of operation with the F9445:

Data-channel cycle with F9445 controlling the memory, Data-channel cycle with external memory control, and Autonomous-bus cycle using bus arbitration scheme.

The sequence of events during a data-channel cycle is as follows:

1. $\overline{\text{DCHREQ}}$ is set.
2. F9445 responds by setting $\overline{M}$, O_1, and O_0 to "101" and $\overline{\text{BUSREQ}}$ to "1". This is recognized externally as Data-Channel Acknowledge and can be used to reset $\overline{\text{DCHREQ}}$ if it is the last data-channel cycle required.
3. F9445 3-states the bus and sends $\overline{\text{STRBA}}$.
4. The external logic must supply an address at this time. The address time can be extended with RDYA.
5. F9445 outputs $\overline{\text{STRBD}}$.
6. The controller transmits or receives the data-channel data and responds with RDYD, concluding the cycle.

Console Operation
Console operation allows examination and modification of the F9445 internal registers without executing programs in main memory. This is very useful for system diagnostics even when the memory or I/O part of the microcomputer system is not fully functional.

Upon request for console operation, the processor will execute one of a number of console operations depending on a console code on the information bus (see *Figure 6*). This facilitates the connection of an external console for monitoring and test purposes. The following sequence is used to execute a console operation:

1. $\overline{\text{CONREQ}}$ is set LOW.
2. The processor finishes the current instruction.
3. The processor sets the $\overline{M}$ and O lines to "110" (console code in).
4. In response to the $\overline{M}$ and O lines being set to "110", the console logic supplies a code on the information bus corresponding to the desired operation, which is selected onto the bus with $\overline{\text{STRBD}}$.
5. The console logic resets $\overline{\text{CONREQ}}$.
6. The processor executes the console operation.
7. The processor may read or write data from the console switches or console lamps. In this case, the $\overline{M}$ and O lines are set to "111" (console data). In most cases, the processor halts after the console operation by entering a Wait state. The exceptions are Continue and APL.

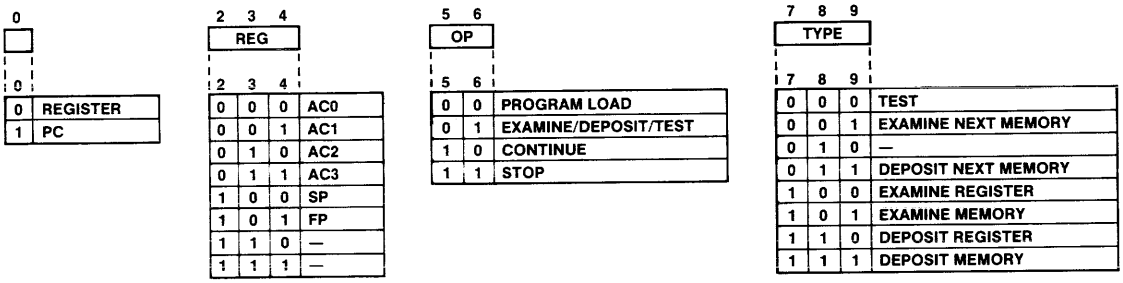
Console logic can be implemented in three levels of simplicity:

No Console Code — If a $\overline{\text{CONREQ}}$ is generated and no console code supplied, the default bus value ("0") will cause the processor to execute APL. This sets the PC to −1, then starts normal execution. This is the minimal console operation required.

Limited Console Operation — A subset of operations can be arranged with a 2-bit console code. These operations are APL, Test, Continue, and Halt.

Full Console Operation — A 9-bit code (see *Figure 6*) defines the full set of console operations. Single-Step is not implemented directly, but can be arranged using Continue: first, the Continue operation is specified; after the first instruction is fetched, a new $\overline{\text{CONREQ}}$ is generated and the operation is changed to Halt.

Fig. 6 Console Codes



0
0 REGISTER
1 PC

2 3 4 REG	
0 0 0	AC0
0 0 1	AC1
0 1 0	AC2
0 1 1	AC3
1 0 0	SP
1 0 1	FP
1 1 0	—
1 1 1	—

5 6 OP	
0 0	PROGRAM LOAD
0 1	EXAMINE/DEPOSIT/TEST
1 0	CONTINUE
1 1	STOP

7 8 9 TYPE	
0 0 0	TEST
0 0 1	EXAMINE NEXT MEMORY
0 1 0	—
0 1 1	DEPOSIT NEXT MEMORY
1 0 0	EXAMINE REGISTER
1 0 1	EXAMINE MEMORY
1 1 0	DEPOSIT REGISTER
1 1 1	DEPOSIT MEMORY

Bus Arbitration

The F9445 contains three signals that allow more than one processor to share a common bus:

$\overline{\text{BUSREQ}}$—This is LOW at the beginning of every cycle in which the F9445 requires use of the bus.

BUSGNT—When LOW, it is used to halt the processor indicating the bus is unavailable.

$\overline{\text{BUSLOCK}}$—This indicates that the current bus cycle and the following bus cycle from the processor must not be interrupted by a cycle from another processor.

The $\overline{\text{BUSLOCK}}$ signal has two purposes. One purpose is to prevent the external memory address register from being overwritten during those instructions that rely on the address remaining in this register. The other purpose is to provide a method of synchronizing separate software tasks using a standard semaphore system. An external arbiter is required to determine which processor has access to the bus.

Applications

Static Memory Interface

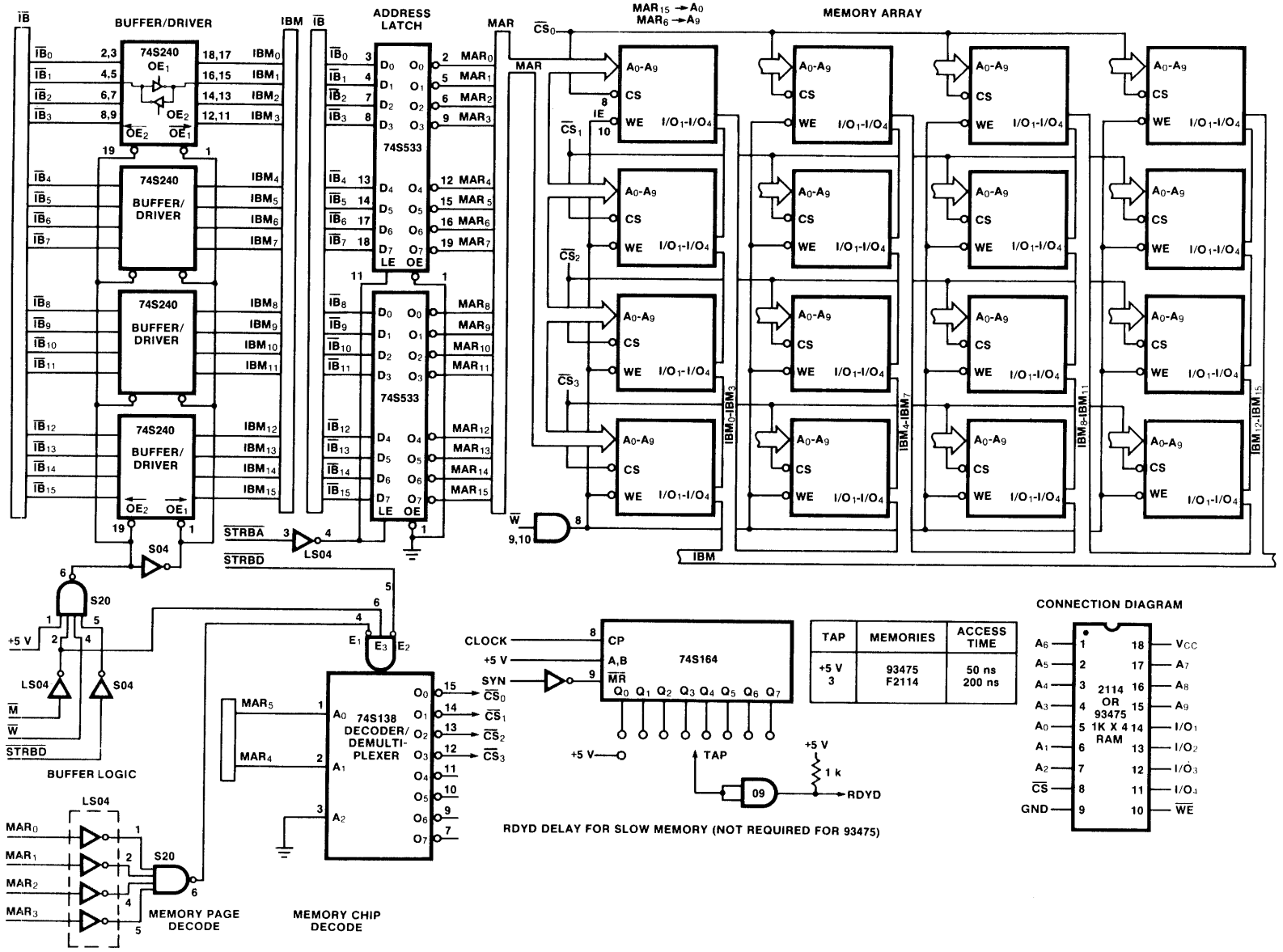
The F9445 bus structure allows easy connection of static memory. Both address and data are multiplexed onto the 16-bit information bus $\overline{\text{IB}}_{(0\text{-}15)}$. The mutually exclusive signals $\overline{\text{STRBA}}$ and $\overline{\text{STRBD}}$ indicate that the information bus is carrying address or data, respectively. The $\overline{\text{M}}$ signal ($\overline{\text{M}}$ = LOW) indicates that a memory cycle is taking place on the bus, while the $\overline{\text{W}}$ signal indicates whether the operation is a read or write. The timing of the $\overline{\text{STRBD}}$ is shorter for a write operation, to allow positive hold time for the memories. The signal RDYD may be held LOW to stretch the memory cycles for slow memories.

A typical scheme is shown in *Figure 7.* This diagram shows a 4K × 16 static RAM configuration (2114-type 1K × 4). The bus is buffered by a 74240 inverting 3-state buffer. Buffering is optional and depends on fan-out requirements of the memories. The buffer is normally connected for output, but is connected for input when necessary by a simple decode of the $\overline{\text{M}}$, $\overline{\text{W}}$ and $\overline{\text{STRBD}}$ lines.

An address latch (74533) is clocked with $\overline{\text{STRBA}}$. The memory address is decoded from these outputs and forms the chip-select ($\overline{\text{CS}}$) inputs to the RAM.

A shift register (74164) provides a time delay for RDYD for slow memories. An alternative for this would be a one-shot (9602). Fast memories do not require RDYD delayed.

Fig. 7 Static Memory Connection Scheme



Input/Output

The F9445 I/O can utilize a simple scheme similar to the memory connection. To take full advantage of standard F9445 I/O instructions, however, I/O instructions must be externally decoded. An F9445 support circuit (F9448, F9447, F9470) can be used for this purpose.

To implement standard F9445 I/O without the support circuits mentioned above requires external logic. This can be implemented with an FPLA (93459). *Table 1* illustrates the PLA for I/O.

Table 1 I/O PLA Listing

```
                                  *A LLLLLLLL
*P 00 *I --L-LLLHLHL--HLL *F -----AAA
*P 01 *I --L-HLLHLHL--LHL *F ------AA
*P 02 *I --L-LLLLHHL--HLL *F -----A-A
*P 03 *I -HH------L------ *F -----A-A
*P 04 *I --L-LLLLHHL--LHL *F -------A
*P 05 *I -LH------L------ *F -------A
*P 06 *I --L-LLLLHHL--LLH *F -----AA-
*P 07 *I --L-LLLLHHL--LHH *F ------A-
*P 08 *I --L-LLLLLHL--LHL *F -----A--
*P 09 *I ---------H---HHH *F A-------
*P 10 *I --L-LLLHLHL--HHH *F --AAA---
*P 11 *I --L-HLLHLHL--HHH *F ---AA---
*P 12 *I --L-LLLLHHL--HHH *F --A-A---
*P 13 *I --L-LLLLHHL--HHL *F --A-A---
*P 14 *I --LHLLLHLHLLH--- *F -AAAA---
*P 15 *I --LHHLLHLHLLH--- *F -A-AA---
*P 16 *I --LHLLLLHHLLH--- *F -AA-A---
*P 17 *I --LHLLLLHHLHH--- *F -A--A---
*P 18 *I --LHLLLHLHLHL--- *F -AAA----
*P 19 *I --LHHLLHLHLHL--- *F -A-A----
*P 20 *I --LHLLLLHHLHL--- *F -AA-----
*P 21 *I --L-HHHHHHH--HLH *F --AA----
*P 22 *I --L-HHHHHHH--HHL *F ---A----
*P 23 *I --L-HHHHHHH--LLH *F --A-----
*P 24 *I --L-HHHHHHH----- *F A-------
```

Key for Table 1:

- *A = Active level of outputs
- *P = Product term number
- *I = Inputs
- *F = Outputs
- – = Don't care
- H = High level
- L = Low level
- A = Active

The schematic (see *Figure 8*) shows a UART connection. The FPLA decodes the instructions and produces outputs from three multiplexers (74138). Spare outputs on these multiplexers can be used to drive other I/O devices.

Busy, done, mask and interrupt latches for both input and output are implemented. The baud rate generator (4702) is programmable for baud rates from 110 to 9600 baud.

In this scheme, the I/O bus is buffered (74240); this is optional. A one-shot (9602) provides a processor cycle delay by holding RDYD low. This allows the use of a slow UART (TR1263B). Converters (1488, 1489) are used for RS232-level connection, and current loop drivers are switch selected as shown. A one-shot (9602) provides a pulse for a TTY reader delay.

Dynamic Memory Control

Since dynamic memory is more difficult than static memory to connect to any processor, the F9445 requires some additional circuitry to drive dynamic memories (see *Figure 9*). There are several approaches to dynamic memory control:

Using an LSI special-purpose dynamic memory controller (e.g. F9446), which is by far the simplest solution;

Using standard SSI or MSI for the controller, requiring considerable board area;

Using software-assisted techniques, which reduces hardware requirements but can result in poorer overall performance; or

Using a standard MSI dynamic memory controller (e.g. 9642) with additional timing and control logic.

The last alternative has the advantage of using standard parts with a low part count and no software overhead. This is the scheme shown in *Figure 9*. The memory address register, data buffer and address decoder are required for any memory, static or dynamic. The 9642 multiplexes the 14-bit address for the dynamic memories, seven bits at a time. The memories require two strobes: a row address strobe ($\overline{RAS}$) and a column address strobe ($\overline{CAS}$). In the scheme shown, all memory chips receive the same $\overline{CAS}$ strobe, but the $\overline{RAS}$ strobe depends on the address. The strobes are sequenced using a combination of F9445 timing signals ($\overline{STRBA}$, SYN, $\overline{STRBD}$) and other signals generated by a 74164 shift register.

Fig. 8 F9445 Input/Output Connection Scheme (1 of 2)

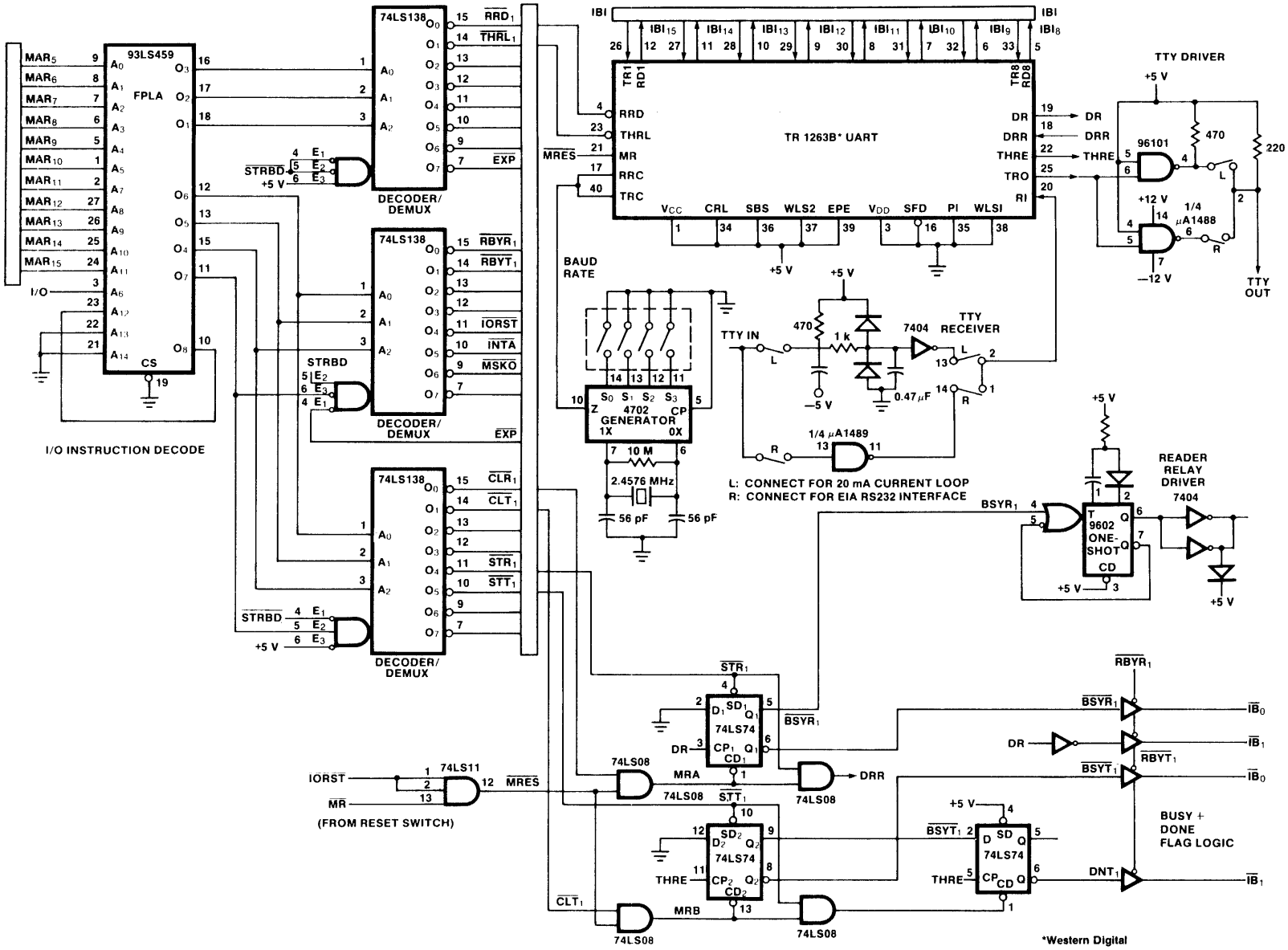


Fig. 8 F9445 Input/Output Connection Scheme (2 of 2)

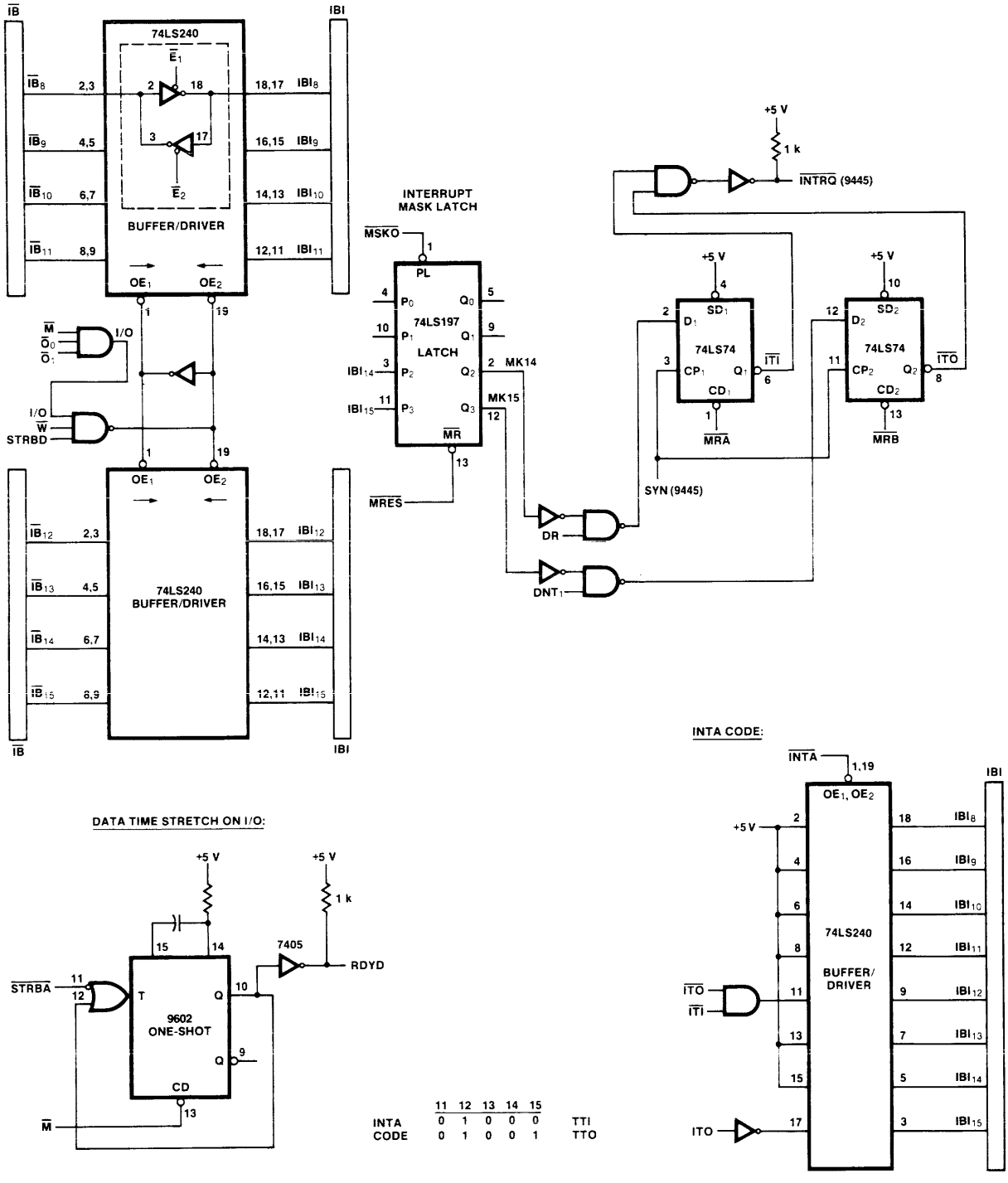
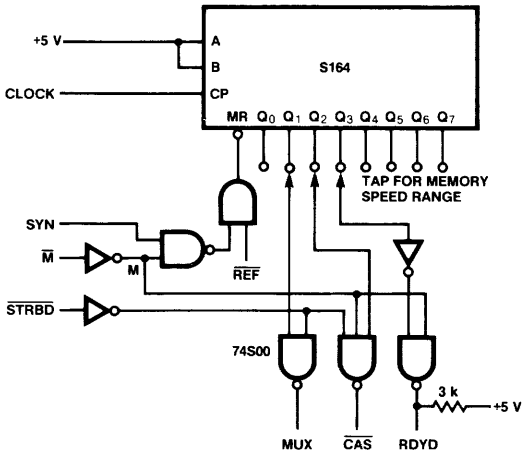
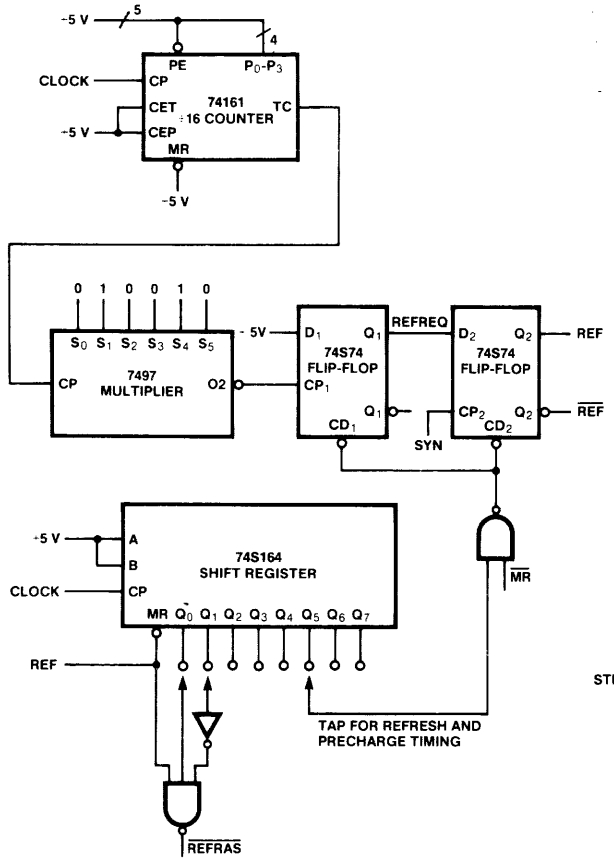


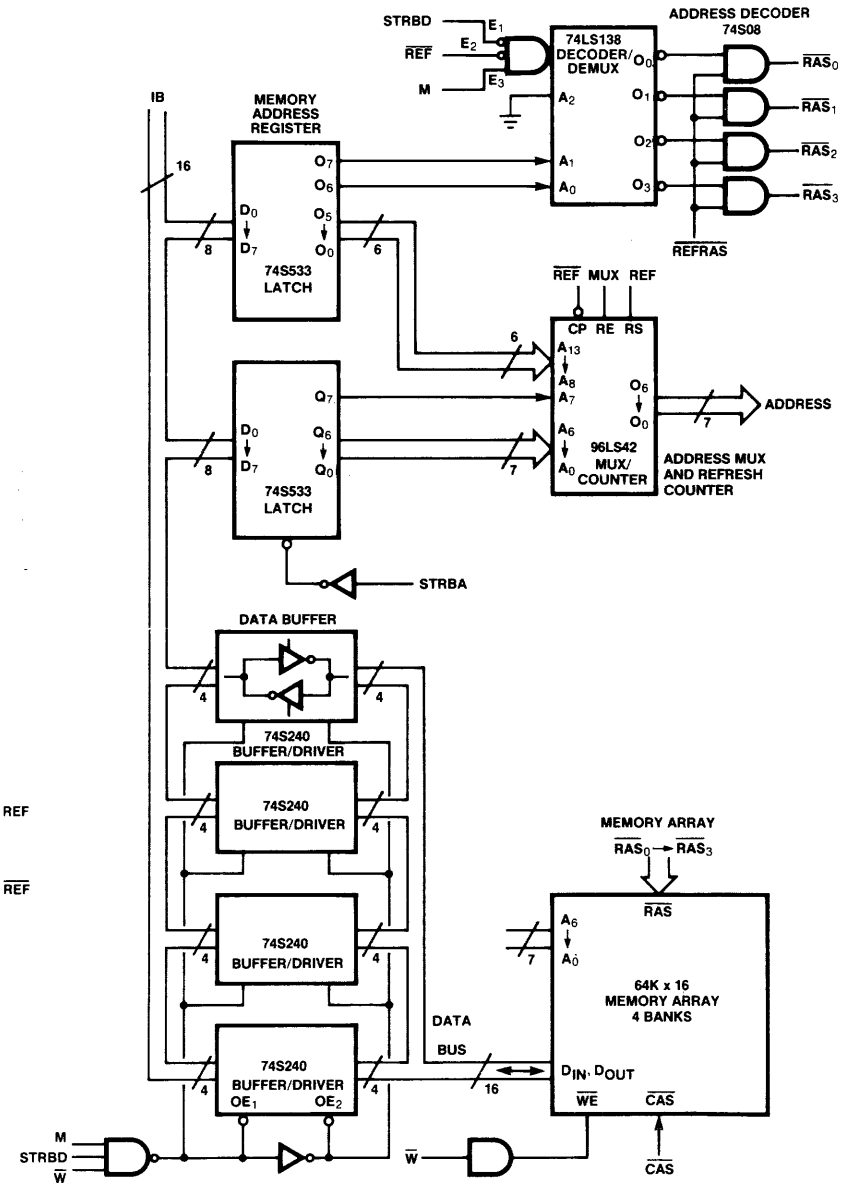
Fig. 9 F9445 Dynamic Memory Connection Scheme

Dynamic Memory Timing



Refresh Timing





Notes

1. D_{IN} connected to D_{OUT} connected to Data Bus.
2. All F16K devices have the same address lines and $\overline{CAS}$, $\overline{WE}$ line.
3. Each bank of 16 F16K has a separate $\overline{RAS}$ line (4 banks).
4. Each slice of 4 F16K is connected to a separate data bus line (16 slices).

The memory requires "refreshing" every 2 ms. The 9642 contains a 7-bit refresh counter. Every 15.63 ms (2/128), the memory controller enters "refresh" mode. This is synchronized with SYN to avoid any conflict. Another 74164 shift register controls the refresh timing which requires only an $\overline{RAS}$ strobe. After the refresh cycle, the refresh counter is incremented and the normal memory timing is resumed.

The refresh cycle takes place when needed and may take place during non-memory processor cycles. In these cases, the processor is not halted, and the refresh cycle is overlapped.

Different memory types have different speed requirements. These requirements can be met by changing the "taps" on the 74164 shift registers.

Console Control

On an application board, a minimal console is usually required. The APL (automatic program load) function can be easily implemented by pulsing the Console Request line LOW. There are no critical timing requirements since this signal is latched internally. The F9445 will continue to execute APL commands until the Console Request is raised. Since the bus must be HIGH for the APL to execute correctly, bits 5 and 6 of the bus may be tied to +5 V through 3 kΩ resistors as pullups.

For debugging and evaluation purposes, a console is a very useful tool. It gives complete control of the processor independent of software and memory operation.

Since the console commands are microprogrammed into the F9445, a full console design is fairly simple, the simplest full console uses the F9470 console-controller circuit, which drives an RS232 terminal and contains two serial I/O ports and a timer. The F9447 I/O controller can also be used to provide some console functions.

Interfacing to standard switches and lamps requires switch debouncing and encoding operations. The circuit shown in *Figure 10* uses R-S latches for switch debouncing and an FPLA (93409) for switch encoding.

An address latch is strobed on every $\overline{STRBA}$, and a data latch is strobed on every $\overline{STRBD}$ except "console code read." This results in the correct display on the lamps. The data switches are enabled with "console data read."

A Single-Step function is included. This function requires two additional latches, plus some decode logic, and implements a Continue followed by a Halt.

The Console Request is set whenever any operation switch is pressed and is reset when the console code is read from the FPLA. The circuit provides for control of two processors sharing the same bus.

All the switches are momentary-action type except the data switches and the select-processor switch.

A full listing of the FPLA is shown in *Table 2*.

The console provides all F9445 console functions, including Self-Test, plus the additional function of Single-Step, and is compact enough to be implemented with all switches, lamps, logic and connectors on a double-sided 17½-by-5½-inch printed-circuit board.

Table 2 Console PLA Listing

```
                                        *A LLLLLLLL
*P 00 *I -H-HHHHHHHHHL--- *F -----A--
*P 01 *I -H-HHHHHHHHL---- *F --A--A--
*P 02 *I -H-HHHHHHHL----- *F --AA-A--
*P 03 *I -H-HHHHHHL---H-- *F ----AA--
*P 04 *I -H-HHHHHHL---L-- *F ----AA-A
*P 05 *I -H-HHHHHL------- *F --A-AA--
*P 06 *I -H-HHHHL-----H-- *F ---AAA--
*P 07 *I HL-H---L-----H-- *F ---AAA--
*P 08 *I HL-H---L-----L-- *F ---AAA-A
*P 09 *I HL-H-----L---L-- *F ----AA-A
*P 10 *I HL-H-----L---H-- *F ----AA--
*P 11 *I -H-HHHHL-----L-- *F ---AAA-A
*P 12 *I -H-HHHL--------- *F --AAAA--
*P 13 *I -H-HHL---------- *F --------
*P 14 *I -H-HL----------- *F ------A-
*P 15 *I ---L------------ *F -----AA-
*P 16 *I HL-H------------ *F ------A-
*P 17 *I LL-H-----H------ *F -----AA-
*P 18 *I -H-HHHHHHHHH--- *F -----AA-
```

Key for Table 2:

*A = Active level of outputs
*P = Product term number
*I = Inputs
*F = Outputs
– = Don't care
H = High level
L = Low level
A = Active

Fig. 10 F9445 Console Connection Scheme (1 of 3)

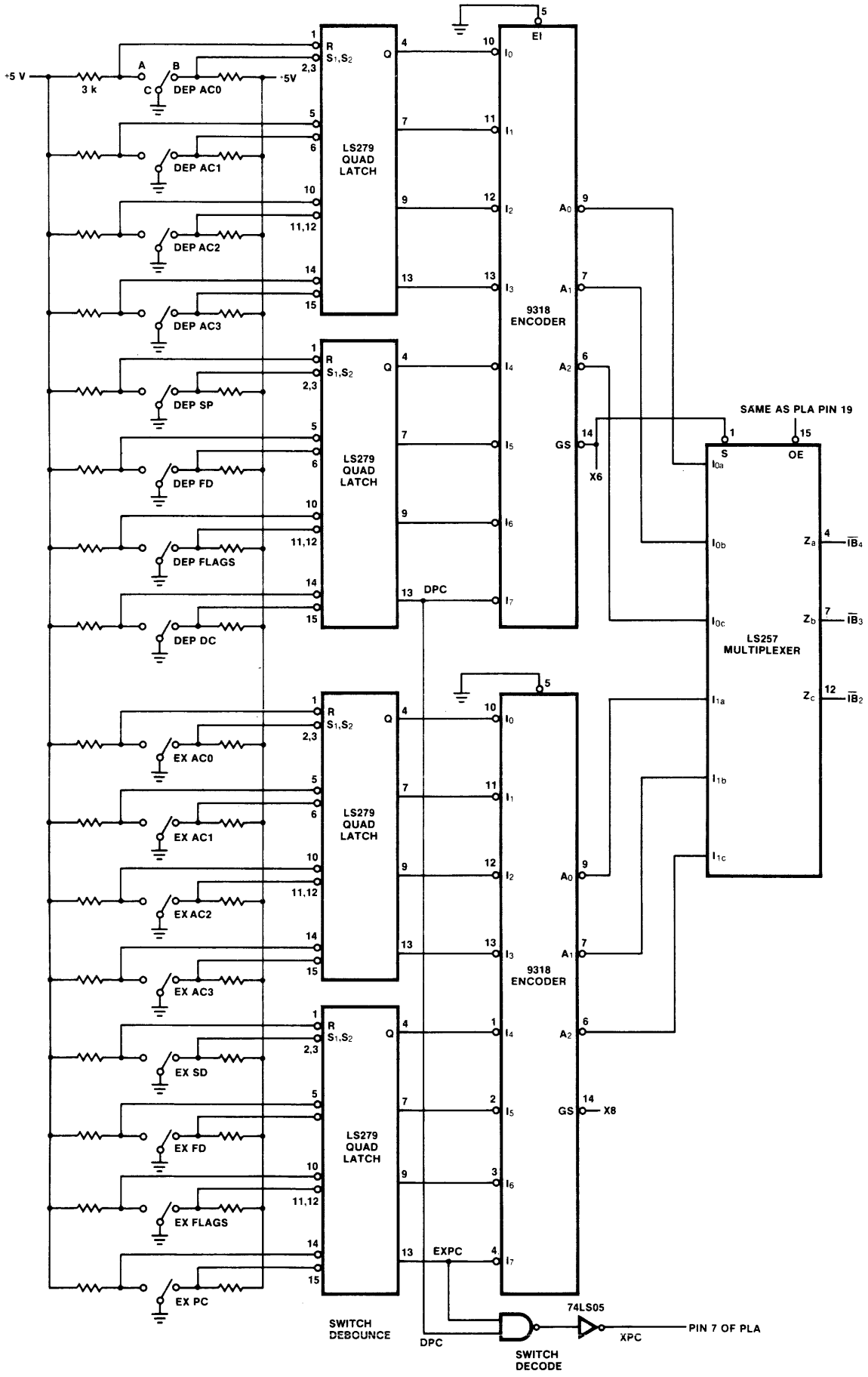


Fig. 10 F9445 Console Connection Scheme (2 of 3)

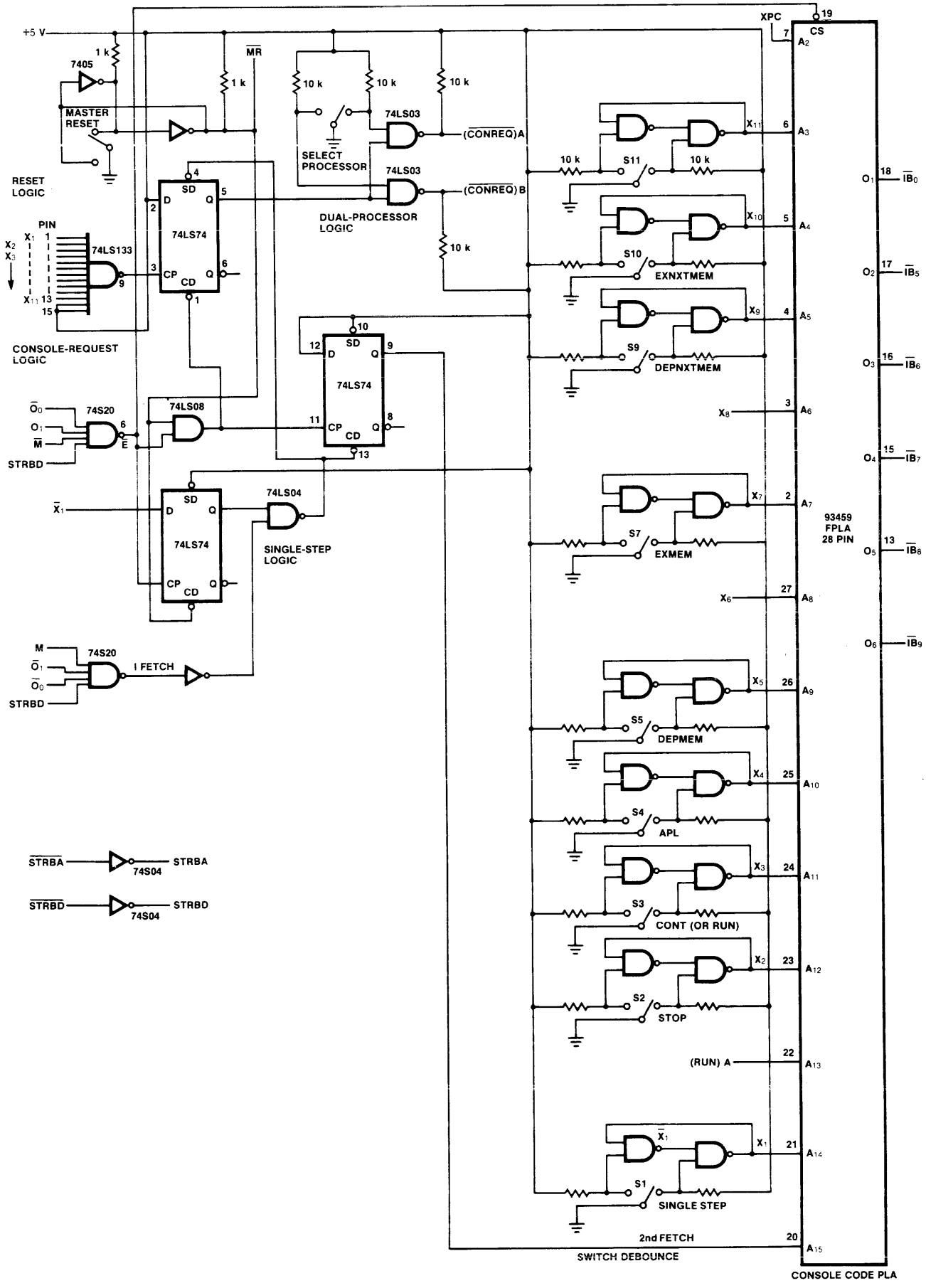
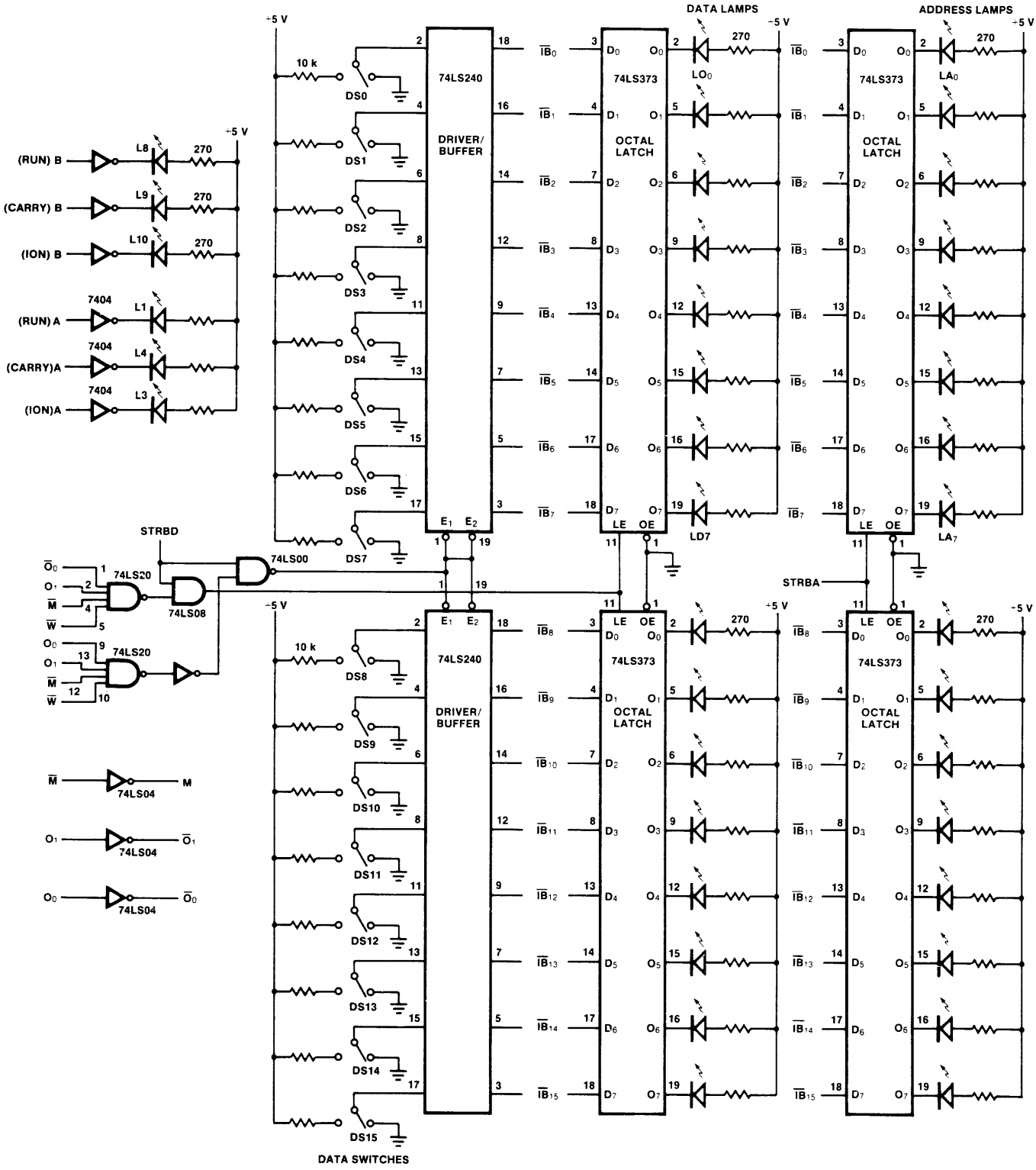


Fig. 10 F9445 Console Connection Scheme (3 of 3)



A Multiprocessor Scheme

There are many ways to envision two or more processors working concurrently. The method of interconnecting the processors depends on the application and the performance objectives. Listed here are a few of the options.

Independent Operation

For those processors which can be made to run independent tasks, this provides the most efficient scheme. Each processor has independent memory and resources.

Shared I/O

Each processor has its own memory but shares an I/O bus. This allows high-speed operation while minimizing system resource requirements.

Local and Common Memory

This gives a good compromise between performance and resource requirements. Each processor normally runs from its own memory at high speed. Accesses to a common memory are rarer and, because of the arbitration problems, slower.

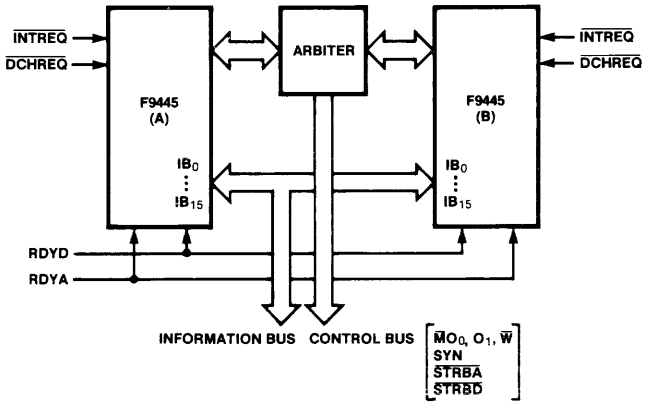
Tightly Coupled

Two or more processors share the same memory and I/O. This scheme is easiest to implement but, because of the completely shared resources, does not give as high performance as the Local and Common Memory scheme. However, for certain applications and for two processors only, this scheme can give a considerable performance increase over a single-processor system with very little hardware overhead. This scheme is described in the following paragraphs.

A general scheme for two tightly coupled F9445 processors is shown in Figure 11. The processors share a common bus and an arbiter selects which processor uses the bus and multiplexes the control lines accordingly. The I/O arbitration scheme is very simple: each processor assigns the bus to the other processor when it commences any cycle that does not use the bus, as long as $\overline{\text{BUSLOCK}}$ is not set.

The scheme is most efficient when the instruction mix includes many "long" instructions, such as Multiply, Divide, Parametric Shift and Normalize. Since only one processor is using the bus at any time, the synchronization signals RDYA and RDYD can be the same for both processors. However, the interrupt request ($\overline{\text{INTREQ}}$) and data-channel request ($\overline{\text{DCHREQ}}$) lines should be separate to avoid any conflicts in I/O handling.

Fig. 11 A Possible General Multiprocessor Scheme



F9445 Instruction Execution Times

Instruction	Clock Cycles	Execution Times			Notes
		16 MHz	20 MHz	24 MHz	
COM	6	0.375	0.3	0.25	
NEG	6	0.375	0.3	0.25	
MOV	6	0.375	0.3	0.25	
INC	6	0.375	0.3	0.25	Times for no-skip or unfulfilled skip;
ADC	6	0.375	0.3	0.25	for fulfilled skip; add 0.3 (0.25 at
SUB	6	0.375	0.3	0.25	24 MHz)
ADD	6	0.375	0.3	0.25	
AND	6	0.375	0.3	0.25	
OR	6	0.375	0.3	0.25	
MUL	70	4.375	3.5	2.9	
MULS	70	4.375	3.5	2.9	
DIV (Normal)	86	5.375	4.3	3.6	
DIV (Overflow)	14	0.875	0.7	0.58	
DIVS (Normal)	114	7.125	5.7	4.7	
DIVS (Overflow)	26	1.625	1.3	1.1	
NORM	10 + 4n	0.625 + 0.25n	0.5 + 0.2n	0.42 + 0.17n	n = number of steps needed for normalization. Time is 0.7 (0.59) if n=0.
SLLD	10 + 4n	0.625 + 0.25n	0.5 + 0.2n	0.42 + 0.17n	n = number of shifts;
SALD	10 + 4n	0.625 + 0.25n	0.5 + 0.2n	0.42 + 0.17n	time is 0.7 (0.59) if n=0.
SARD	10 + 4n	0.625 + 0.25n	0.5 + 0.2n	0.42 + 0.17n	
SLRD	10 + 4n	0.625 + 0.25n	0.5 + 0.2n	0.42 + 0.17n	
SKNV	14	0.875	0.7	0.58	
JMP	6	0.375	0.3	0.25	Times for page-zero addressing; add
JSR	6	0.375	0.3	0.25	0.3 (0.25) for indirect; add 0.3 (0.25)
ISZ	22	1.375	1.1	0.92	for auto-increment/decrement; add
DSZ	22	1.375	1.1	0.92	0.2 (0.18) for indexed.
LDA	12	0.75	0.6	0.5	
STA	12	0.75	0.6	0.5	
LDB	24	1.5	1.2	1.0	
STB	26	1.625	1.3	1.1	
PSHA	16	1.0	0.8	0.67	
POPA	16	1.0	0.8	0.67	
PSHF	16	1.0	0.8	0.67	
POPF	16	1.0	0.8	0.67	
POPJ	16	1.0	0.8	0.67	
PSHR	16	1.0	0.8	0.67	
TOPR	16	1.0	0.8	0.67	
TOPW	16	1.0	0.8	0.67	
MTSP	6	0.375	0.3	0.25	
MTFP	6	0.375	0.3	0.25	
MFSP	6	0.375	0.3	0.25	
MFFP	6	0.375	0.3	0.25	
SAV	60	3.75	3.0	2.5	
RET	80	5.0	4.0	3.3	
DSP	6	0.375	0.3	0.25	
NIO	12	0.625	0.6	0.5	
SKP	16	1.0	0.8	0.67	
DIA/B/C	12	1.0	0.6	0.5	
DOA/B/C	12	1.0	0.6	0.5	
ETRP	10	0.625	0.5	0.42	
DTRP	10	0.625	0.5	0.42	
E64K	10	0.625	0.5	0.42	
D64K	10	0.625	0.5	0.42	

Note: Execution times are given for 20 MHz and 24 MHz clock. The clock may be operated from > 0 to 24 MHz within the specified temperature and voltage range.

Fig. 12 ALU Cycle Timing*

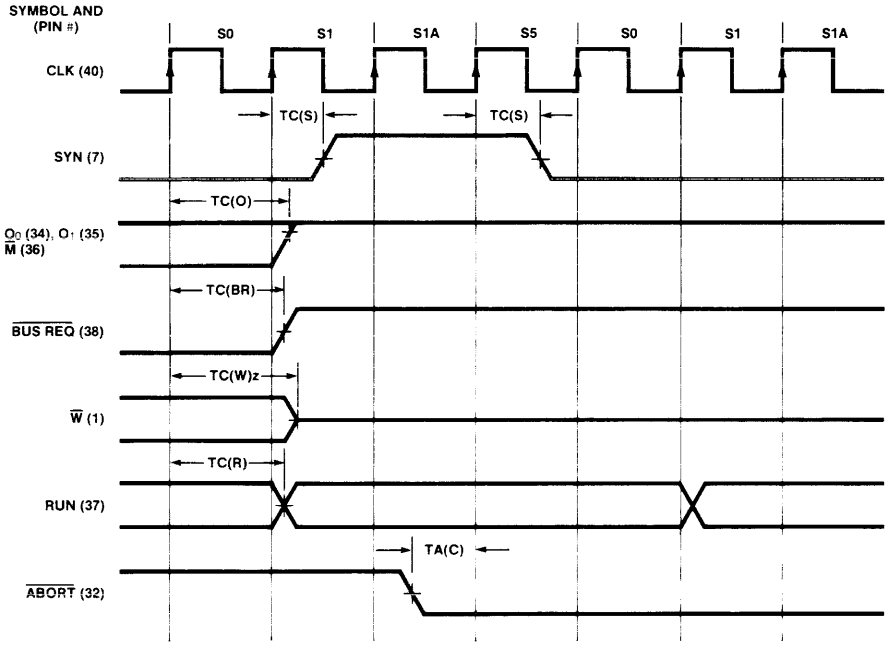
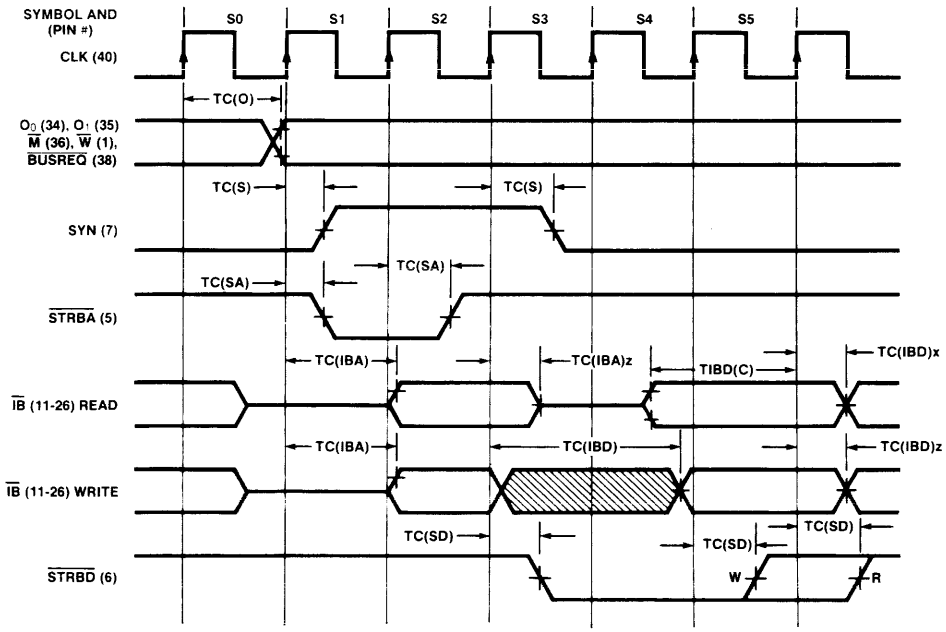
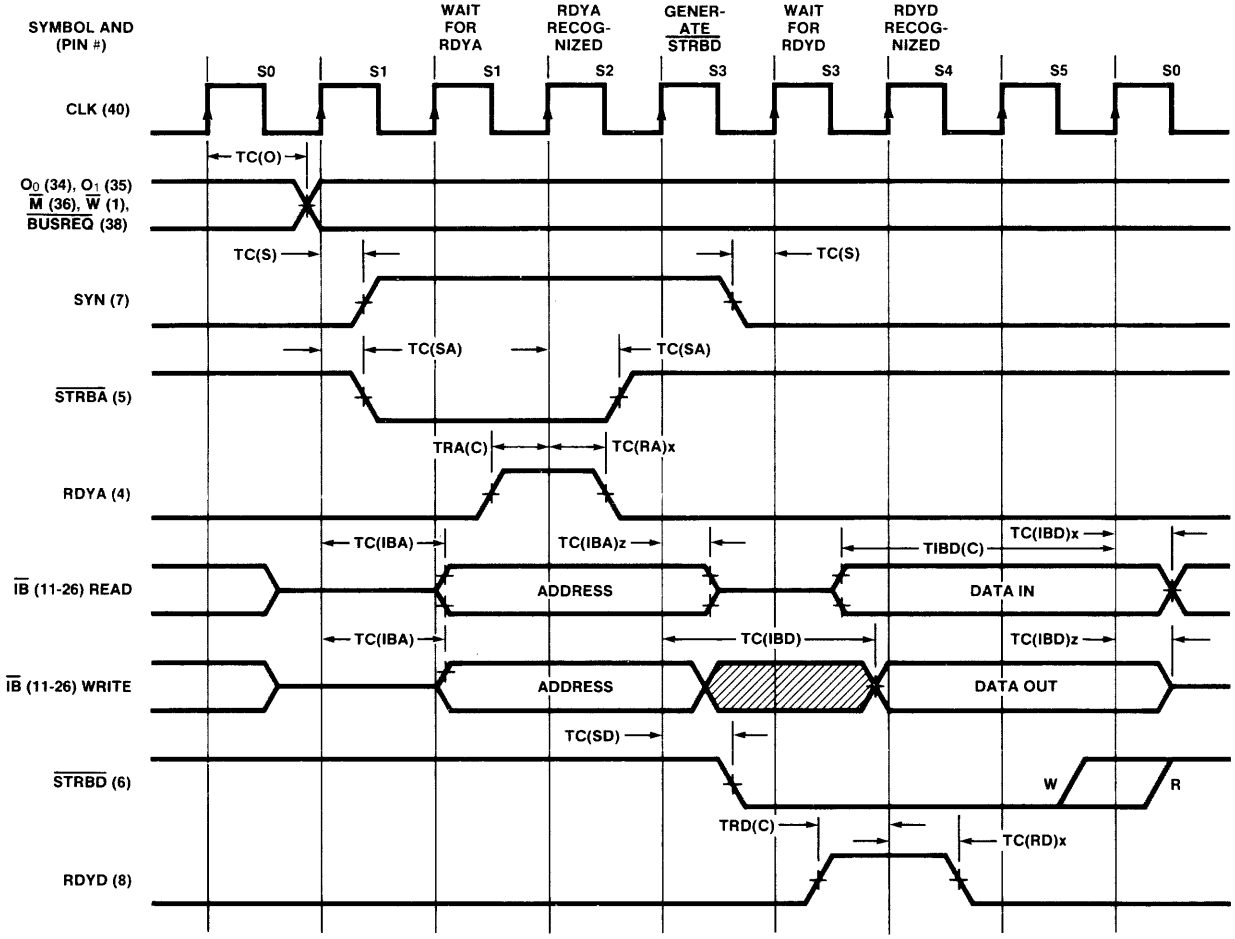


Fig. 13 Minimum Memory Cycle Timing*



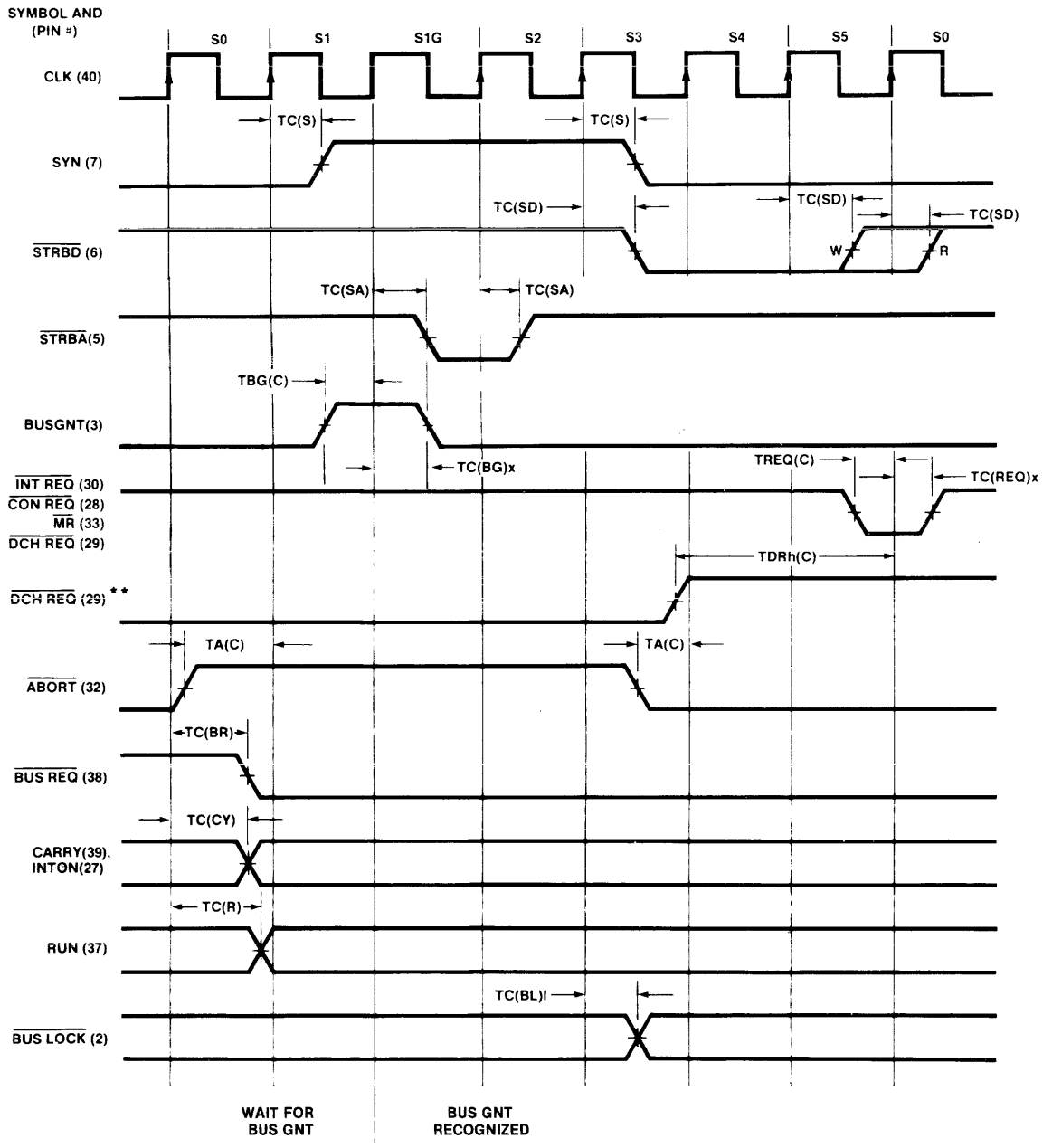
*See Timing Parameter Symbol Conventions at end of data sheet.

Fig. 14 Extended Memory Cycle Timing*



*See *Timing Parameter Symbol Conventions* at end of data sheet.

Figure 15. Bus and Status Control Timing*



*See Timing Parameter Symbol Conventions at end of data sheet.

**If this DCH REQ set-up time is missed, it is not recognized for another complete cycle.

Guaranteed Operating Ranges

Part Number	Supply Voltage (V_{CC})			Ambient Temperature
	Min	Typ	Max	
F9445DC	4.75 V	5.0 V	5.25 V	0 to +75°C
F9445DM	4.5 V	5.0 V	5.5 V	−55 to +125°C

DC Characteristics

(Over guaranteed operating ranges unless other wise noted.)
I_{INJ}(min) = 300 mA; I_{INJ}(max) = 400 mA

Symbol	Characteristic	Min	Typ	Max	Unit	Test Conditions
V_{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage
V_{IL}	Input LOW Voltage			0.8	V	Guaranteed Input LOW Voltage
V_{CD}	Input Clamp Diode Voltage		−0.9	−1.5	V	V_{CC} = Min, I_{IN} = −18 mA, I_{INJ} = 300 mA
V_{OH}	Output HIGH Voltage; RUN, CARRY, INTON, SYN, $\overline{STRBD}$, $\overline{BUSREQ}$, $\overline{STRBA}$, O_0, O_1, $\overline{M}$	2.4	3.4		V	V_{CC} = Min, I_{OH} = −400 μA, I_{INJ} = 300 mA
V_{OH}	Output HIGH Voltage; $\overline{IB}_{(0\text{-}15)}$, $\overline{W}$	2.4	3.4		V	V_{CC} = Min, I_{OH} = −1.0 mA, I_{INJ} = 300 mA
V_{OL}	Output LOW Voltage		0.25	0.5	V	V_{CC} = Min, I_{OL} = 8.0 mA, I_{INJ} = 300 mA
I_{IH}	Input HIGH Current; $\overline{DCHREQ}$, $\overline{INTREQ}$, CLK, $\overline{MR}$, RDYA, RDYD, $\overline{ABORT}$, $\overline{CONREQ}$, BUSGNT		2.0	40	μA	V_{CC} = Max, V_{IN} = 2.7 V, I_{INJ} = 300 mA
I_{IH}	Input HIGH Current; $\overline{IB}_{(0\text{-}15)}$ (3-state)		5.0	100	μA	V_{CC} = Max, V_{IN} = 2.7 V, I_{INJ} − 300 mA
I_{IH}	Input HIGH Current; All inputs			1.0	mA	V_{CC} = Max, V_{IN} = 5.5 V, I_{INJ} = 300 mA
I_{IL}	Input LOW Current; All Inputs		−0.21	−0.4	mA	V_{CC} = Max, V_{IN} = 0.4 V, I_{INJ} = 300 mA
I_{OZH}	Output OFF State (High Impedance) Current $\overline{IB}_{(0\text{-}15)}$, W			100	μA	V_{CC} = Max, V_{OUT} = 2.4 V, I_{INJ} = 300 mA
I_{OZL}	Output OFF State (High Impedance) Current $\overline{IB}_{(0\text{-}15)}$		−210	−400	μA	V_{CC} = Max, V_{OUT} = 0.4 V, I_{INJ} = 300 mA
I_{OZL}	Output OFF State (High Impedance) Current; $\overline{W}$			−100	μA	V_{CC} = Max, V_{OUT} = 0.5 V, I_{INJ} = 300 mA
I_{OSH}	Output Short Circuit Current; All Outputs Except $\overline{BUSLOCK}$	−15		−100	mA	V_{CC} = Max, V_{OUT} = 0.0 V, I_{INJ} = 300 mA*
I_{LOH}	Output Leakage; $\overline{BUSLOCK}$			1.0	mA	V_{CC} = Min, V_{OH} = 5.25 V, I_{INJ} = 300 mA
I_{CC}	Supply Current		160		mA	V_{CC} = Max, I_{INJ} = 300 mA
V_{INJ}	Injector Voltage		1.3		V	I_{INJ} = 400 mA

*Not more than one output to be shorted at a time.

AC Characteristics

T_A = 0 to 75°C; V_{CC} = 4.75 to 5.25 V; I_{INJ} = 300 mA; C_L = 15 pF.
Input conditioning: Rise Time = 6 ns; Fall time = 6 ns; Amplitude = 0 to 3 V

Refer to *Symbol Conventions* at the end of this data sheet for explanation of the timing parameter symbols.

Symbol	Characteristic	Min	Typ	Max	Unit
TC(O)	Propagation delay, CLK to O_0, O_1, $\overline{M}$		60		ns
TC(S)	Propagation delay, CLK to SYN		30		ns
TC(W)	Propagation delay, CLK to $\overline{W}$		70		ns
TC(W)z	Propagation delay, CLK to $\overline{W}$ going 3-state		70		ns
TC(IBA)	Propagation delay, CLK to $\overline{IB}_{(0\text{-}15)}$, address		60		ns
TC(IBA)z	Propagation delay, CLK to $\overline{IB}_{(0\text{-}15)}$, address, going 3-state (read cycle)		35		ns
TC(SA)	Propagation delay, CLK to $\overline{STRBA}$		30		ns
TC(IBD)	Propagation delay, CLK to $\overline{IB}_{(0\text{-}15)}$, data out		75		ns
TC(IBD)z	Propagation delay, CLK to $\overline{IB}_{(0\text{-}15)}$, data out, going three-state		35		ns
TC(SD)	Propagation delay, CLK to $\overline{STRBD}$		25		ns
TRA(C) TC(RA)x	Setup time, RDYA to CLK Hold time, CLK to RDYA		3 10		ns ns
TRD(C) TC(RD)x	Setup time, RDYD to CLK Hold time, CLK to RDYD		2 10		ns ns
TIBD(C)	Setup time, $\overline{IB}_{(0\text{-}15)}$, data in, to CLK (read or fetch cycle)		75		ns
TC(IBD)x	Hold time, $\overline{IB}_{(0\text{-}15)}$, data in, after CLK (read or fetch cycle)		25		ns
TREQ(C)	Setup time, $\overline{INTREQ}$, $\overline{DCHREQ}$, $\overline{CONREQ}$, $\overline{MR}$ to CLK, all are the same timing relative to S5		15		ns
TC(REQ)x	Hold time, $\overline{INTREQ}$, $\overline{DCHREQ}$, $\overline{CONREQ}$, $\overline{MR}$ after CLK, all are the same timing		20		ns
TDRh(C)	Data channel ($\overline{DCHREQ}$) off setup time from CLK (to finish data-channel cycle)		100		ns
TA(C)	Setup time, $\overline{ABORT}$ to CLK		30		ns
TC(BL)1	Propagation delay, CLK to $\overline{BUSLOCK}$ going LOW		35		ns
TBG(C) TC(BG)x	Setup time, BUSGNT to CLK Hold time, CLK after BUSGNT		10 10		ns ns
TC(R)	Propagation delay, CLK to RUN		80		ns
TC(CY)	Propagation delay, CLK to CARRY		50		ns
TC(INT)	Propagation delay, CLK to INTON		50		ns
TC(BR)	Propagation delay, CLK to $\overline{BUSREQ}$		40		ns

Timing Parameter Symbol Conventions

The abbreviated symbols used for ac characteristic timing parameters in this data sheet are defined as follows:

The timing symbol convention is: TAb(C)d

The timing symbols all begin with the letter "T".

The second position, represented by "A", indicates the signal node beginning the interval.

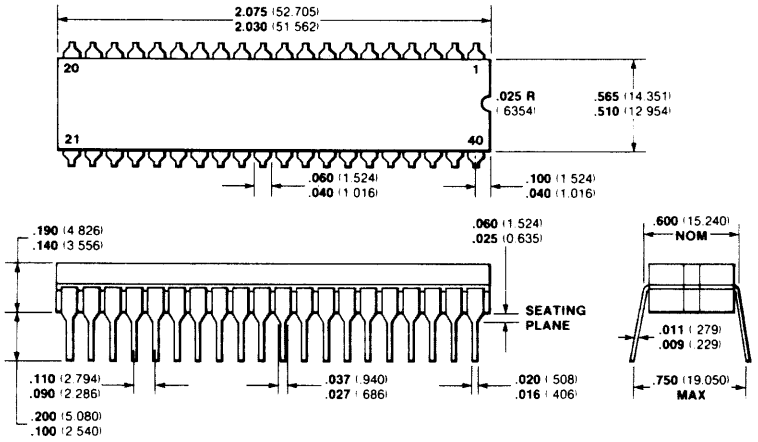
The position "b" defines the direction of signal transition at the beginning node "A", if such definition is necessary; the new state of the signal may be: l = Low; h = High; z = 3-state; x = Don't care; v = Valid

The position "C", which always appears within parentheses, indicates the signal node ending the interval.

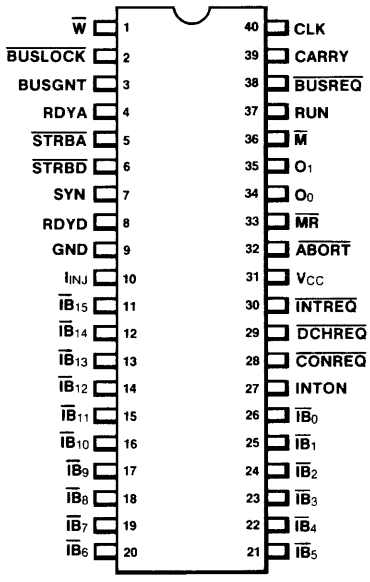
The position "d" is the same as "b" but refers to the state of the signal at the node indicated by the mnemonic in position "C".

Package Information

40-Pin Ceramic Dual In-Line



Connection Diagram
40-Pin DIP (Top View)



Note
All dimensions in inches **bold** and millimeters (parentheses)
Package material is alumina
Sealing material is vitreous glass
Package weight is 12 grams
Mil M38510 lead finishes are available.

Ordering Information

ORDER CODE	SPEED	TEMPERATURE
F9445-24 DC	24 MHz	0°C to +75°C
F9445-24 DM	24 MHz	-55°C to +125°C
F9445-24 DMQB	24 MHz	-55°C to +125°C
F9445-20 DC	20 MHz	0°C to +75°C
F9445-20 DM	20 MHz	-55°C to +125°C
F9445-20 DMQB	20 MHz	-55°C to +125°C
F9445-16 DC	16 MHz	0°C to +75°C
F9445-16 DM	16 MHz	-55°C to +125°C
F9445-16 DMQB	16 MHz	-55°C to +125°C

For other temperature ranges, contact Fairchild Sales Office.

All packages are Ceramic DIPs

FAIRCHILD
A Schlumberger Company

F9446
Dynamic Memory Controller

Advance Product Information

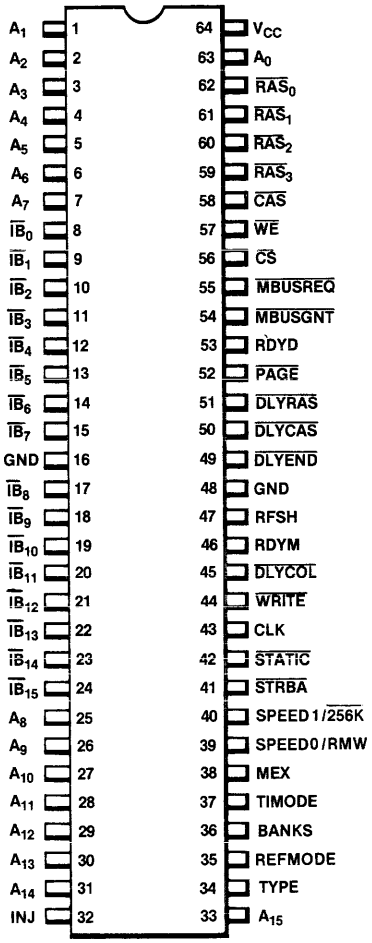
Microprocessor Product

Description

The Fairchild F9446 Dynamic Memory Controller (DMC) is designed to support a variety of memory configurations and provide an interface between 16K and 64K memory chips and the F9445 central processing unit (CPU). It provides a 16-bit memory address register (MAR), an address multiplexer for the row, column, and refresh addresses, a timing generator for the row and column strobe signals and the write enable signal (RAS-CAS-WE), mode arbitration, and page mode logic. It is implemented in I^3L® bipolar technology with low-power Schottky-compatible inputs and outputs.

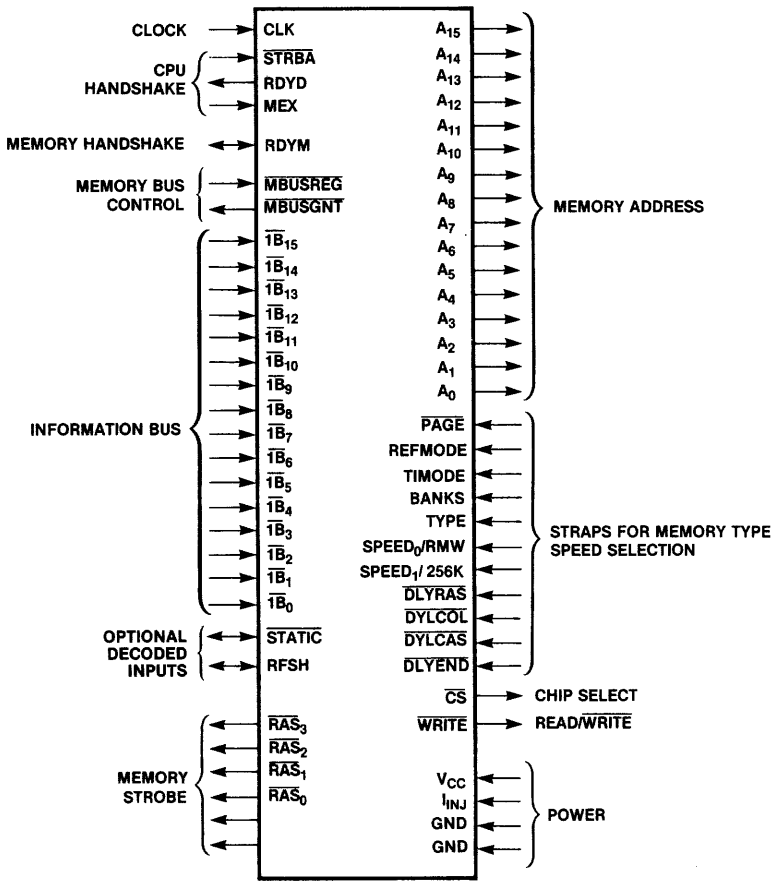
- **16-bit Memory Address Register**
- **Ability to Accommodate a Variety of Memory Speeds**
- **16K or 64K DRAMs**
- **Automatic Page Mode**
- **Internal Refresh Address Counter**
- **Row/Column/Refresh Multiplexer**
- **Complete Memory Timing Signals**
- **Three-state Outputs for Multiport Memories**
- **Internal Refresh Rate Timer**
- **Low-power Schottky-compatible I/0**
- **I^3L Bipolar Technology**
- **64-Pin DIP**
- **Operating Temperature Range of from −55°C to +125°C**

Connection Diagram



Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

Signal Descriptions



The F9446 incorporates an address multiplexer and memory timing generator for use with both static and dynamic memories. The multiplexer selects between row and column segments of the internal 16-bit memory address register (MAR) or an internal refresh address counter (RAC). The upper two address bits provide bank information, while the lower seven, eight, or nine address bits are multiplexed to provide row, column, and refresh address. Assertion of the static line supresses row/column multiplexing of the memory address register outputs, which then provides the full 16-bit address.

The memory timing sequence is initiated by the memory execute signal; it may be inhibited or aborted by removal of the chip select signal. Once started, the memory timing sequence is automatic. A choice of four speed grades accommodates a variety of memory access times, and external controls may be used to further modify the timing.

The four individual RAS lines accommodate from one to four banks of memory chips, with automatic satisfaction of precharge requirements for memory access and refresh, in page mode or not, with distributed refresh or bulk refresh.

Refreshes are initiated to satisfy a rate of 128 per 2 ms per RAS bank. For short intervals, they may automatically be deferred until nonmemory CPU cycles; this makes them semi-transparent.

A memory bus request and memory bus grant are provided to govern 3-state control of memory interface signals for multiport or DMA purposes.

FAIRCHILD
A Schlumberger Company

F9447
I/O Bus Controller

Advance Product Information

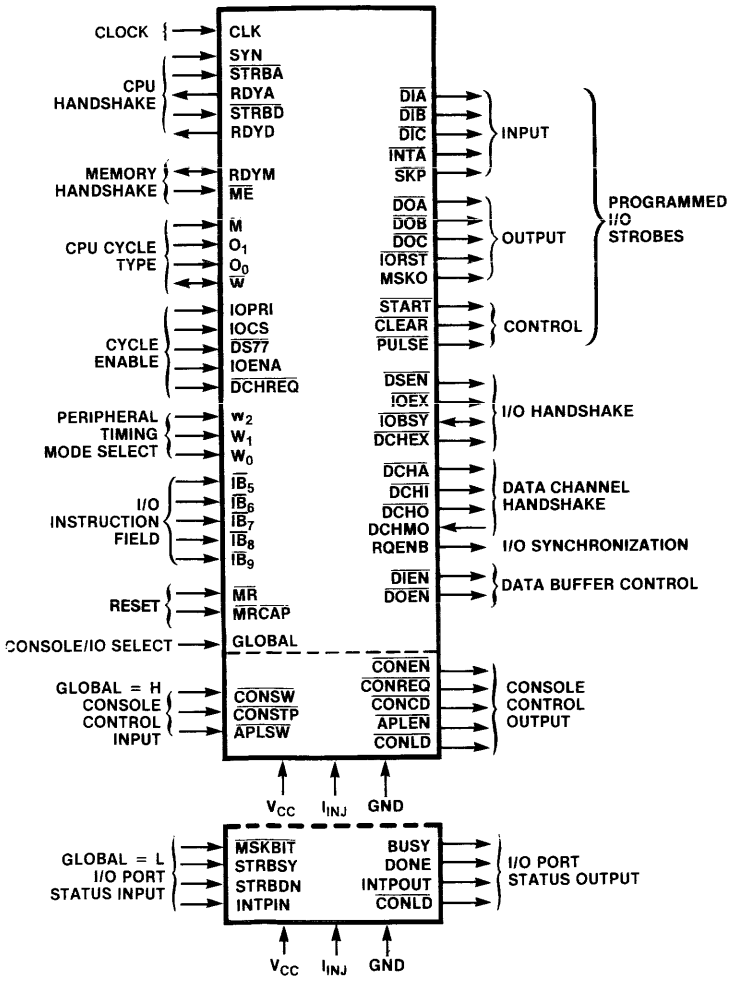
Microprocessor Product

Description

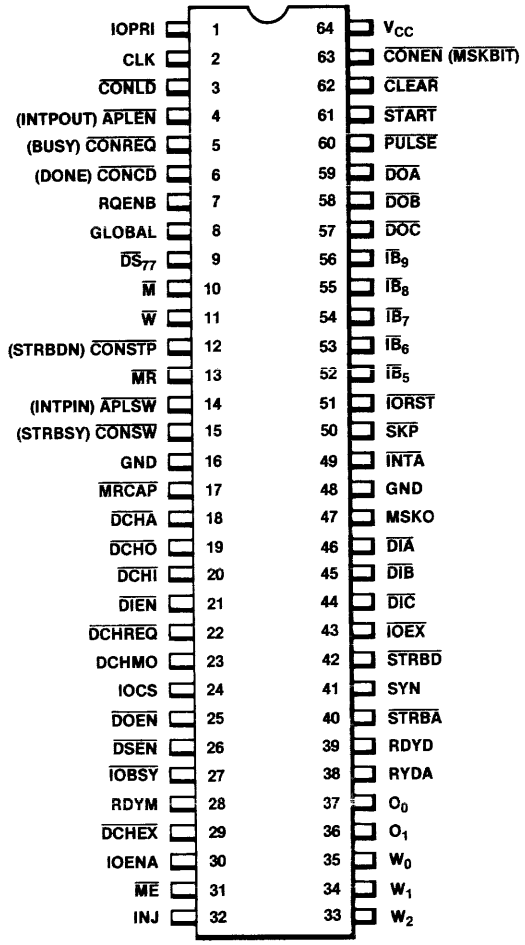
The F9447 I/O Bus Controller (IOC) is used with the F9445 I^3L® 16-Bit Bipolar Microprocessor to demultiplex the I/O instruction and data of the information bus (IB). It provides all the timing and decode signals required for programmed or data channel (DCH) input/output to peripheral device controllers. In the NOVA® -compatible mode, the F9447 provides all the timing and signals required by that I/O bus. For DCH transfers, address generation and handshake can be handled by the F9449 Multiple Data Channel Controller.

- **Interfaces Directly to the F9445**
- **Controls Standard and High-Speed NOVA-Compatible Data Channels**
- **NOVA-Compatible or F9445 I/O and Data Channel Timing**
- **Complete NOVA-Compatible I/O Bus Interface**
- **Automatic Program Load**
- **Power-Up Reset Delay**
- **Console Interface**
- **Local Busy/Done/Interrupt Logic**
- **Low-Power Schottky-Compatible I/O**
- **64-Pin DIP or Optional Chip Carrier Package**
- **I^3L Technology**
- **Operating Temperature Range of from −55°C to +125°C**

Signal Functions



Connection Diagram



® I^3L is a registered trademark of Fairchild Camera and Instrument Corp.

® NOVA is a trademark of Data General Corporation.

Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

The F9447 I/O controller is a decoder and timing generator for programmed I/O instructions and data channel transfers. The timing sequence is selected via a three-bit control code, W_0-W_2. Additional logic is included to implement either the basic console interface or a busy/done/interrupt function. A hysteresis circuit for deriving a power-on-reset, from an external capacitor to ground, is also included. Figure 1 is a typical block diagram of the F9447 I/O controller. Figure 2 shows how the F9447 can be used with the F9445 system.

Signal Description

Table 1 describes the F9447 signals.

Figure 1 F9447 Block Diagram

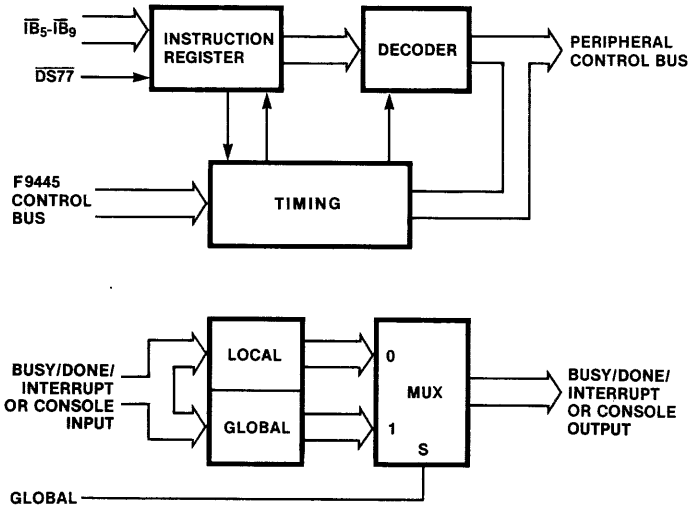


Figure 2 System Configuration

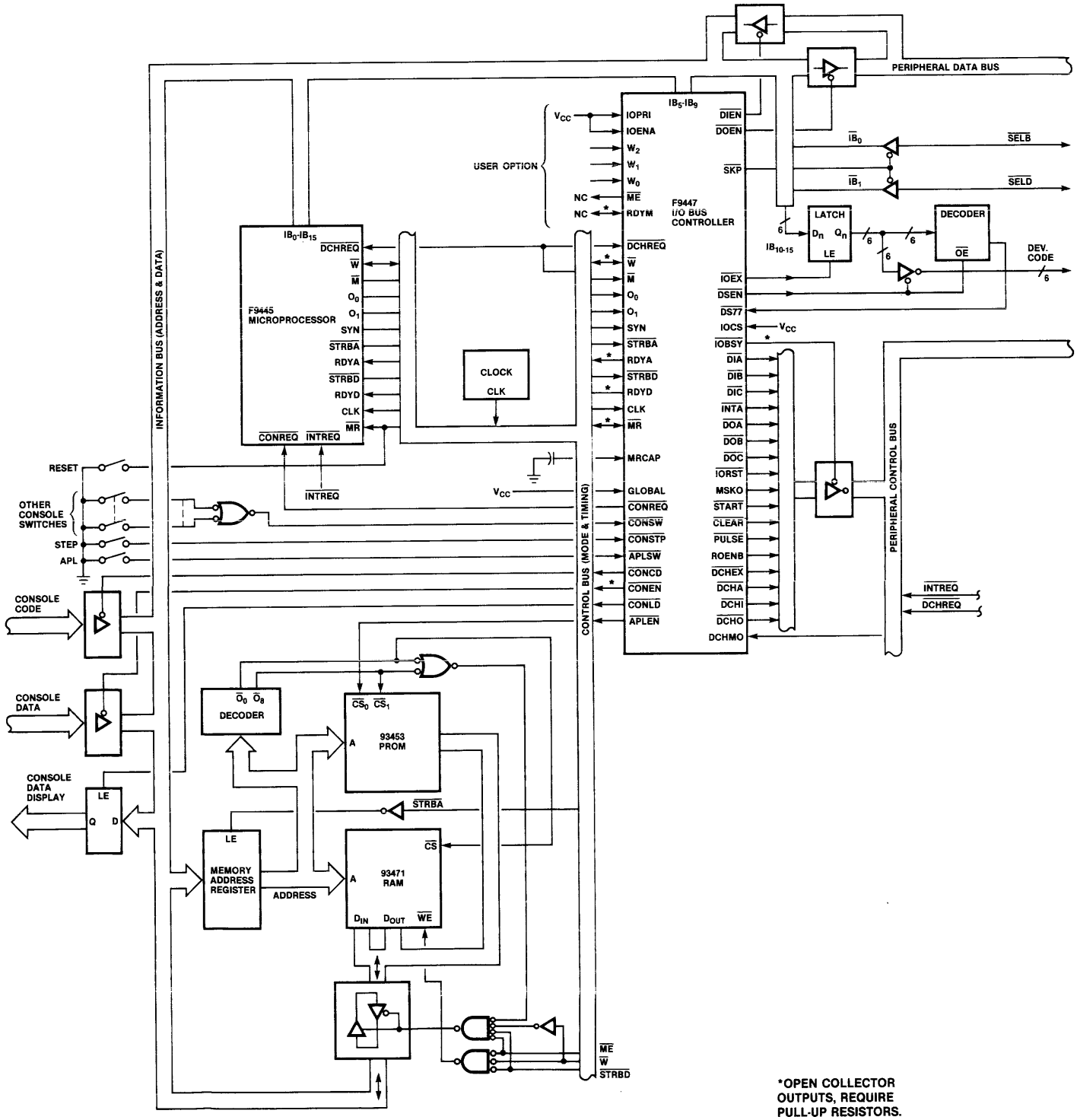


Table 1 F9447 Signal Descriptions

Mnemonic	Pin No.	Name	Description
Clock			
CLK	2	Clock	A synchronizing input signal primarily for timing non-F9445 mode intervals through a state counter clock on alternate positive edges.
CPU Handshake			
SYN	41	Synchronize	An input signal from the CPU that maintains system timing.
$\overline{\text{STRBA}}$	40	Strobe Address	An active low input signal that indicates an address portion of the CPU cycle (and instruction during I/O execute).
RDYA	38	Ready Address	An active high open-collector output signal that allows the CPU to continue beyond the address portion of the cycle.
$\overline{\text{STRBD}}$	42	Strobe Data	An active low input signal that indicates the data portion of the CPU cycle.
RDYD	39	Ready Data	An active high open-collector output signal that allows the CPU to continue beyond the data portion of the cycle.
Memory Handshake			
RDYM	28	Ready Memory	A bidirectional, open collector signal; an active high input during a data channel output cycle indicates that the memory has fetched the data. An active high output during a data channel input cycle indicates to memory that the input data is valid and a write may proceed.
ME	31	Memory Enable	An active low output that modifies the memory controller to respond for CPU memory accesses or for data channel cycles ($ME = M + O_1 \cdot O_0$).
CPU Cycle Type			
$\overline{\text{M}}$	10	Memory	An active low input from the CPU that indicates a memory type of cycle.
O_1 O_0	36 37	O Lines	A pair of input signals from the CPU to indicate the type of cycle.
$\overline{\text{W}}$	11	Write	An active low input signal from the F9445 that indicates a cycle during which data is to be written to a memory or I/O device.

Mnemonic	Pin No.	Name	Description
Cycle Enable			
IOPRI	1	I/O Priority	An active high input that enables the F9447 to begin an I/O or data channel cycle.
IOCS	24	I/O Chip Select	An active high input enables strobes for busy and done, as well as I/O control decodes.
$\overline{DS}_{77}$	9	Device Select 77	An active low input signal that indicates a decode of $\overline{IB}_{10}$-$\overline{IB}_{15}$ all active, a device code 77 (CPU class) I/O instruction.
IOENA	30	I/O Enable	An active high input that, when low, inhibits response by the F9447 to any F9445-programmed I/O cycle. Used with multiple F9447s to allow disables.
$\overline{DCHREQ}$	22	Data Channel Request	An active low input that indicates there is a data channel request.
Peripheral Timing Mode Select			
W_2 W_1 W_0	33 34 35	Timing Options	A three-bit input code; the decodes provide a selection of timing for I/O and data channel cycles.
I/O Instruction Field			
$\overline{IB}_5$-$\overline{IB}_9$	52-56	Information Bus	Active low input signals from the F9445 containing I/O instruction bits 5 through 9 during address phase of I/O execute cycles.
Reset			
$\overline{MR}$	13	Master System Reset	An open-collector bidirectional pin that, when pulled low, initializes the F9447 and activates $\overline{IORST}$. Also an output generated by a low level on $\overline{MRCAP}$.
$\overline{MRCAP}$	17	Capacitive Reset	An active low input with an internal resistive pullup of 10K ohms to V_{CC}.

* The F9447 does not modify the F9445 timing for the following control instructions: READS, ION, APL, HALT.

Table 1 F9447 Signal Descriptions (Cont'd.)

Mnemonic	Pin No.	Name	Description
Console I/O Select			
GLOBAL	8	Global (Local)	An input that selects one of two uses of the global and local modes.
Programmed I/O Strobes			
$\overline{DIA}$	46	Data-In-A	Active low timing strobe outputs that indicate execution of data input instructions.
$\overline{DIB}$	45	Data-In-B	
$\overline{DIC}$	44	Data-In-C	
$\overline{INTA}$	49	Interrupt Acknowledge	
$\overline{SKP}$	50	Skip	
$\overline{DOA}$	59	Data-Out-A	Timing strobe outputs (all active low except active high MSKO) that indicate execution of data output instructions.
$\overline{DOB}$	58	Data-Out-B	
$\overline{DOC}$	57	Data-Out-C	
$\overline{IORST}$	51	I/O Reset	An active low output that indicates either a decode of the execution of the $\overline{IORST}$ instruction or a system reset caused by $\overline{MR}$ or $\overline{MRCAP}$ active.
MSKO	47	Mask Out	
$\overline{START}$	61	Start	Active-low control outputs that indicate decode and time of start, clear, and pulse control functions during programmed I/O execution.
$\overline{CLEAR}$	62	Clear	
$\overline{PULSE}$	60	Pulse	
I/O Handshake			
$\overline{DSEN}$	26	Device Select Enable	An active low output that enables device select.
$\overline{IOEX}$	43	I/O Execute	An active low output indicating that the F9447 is involved in the execution of a programmed I/O cycle.
$\overline{IOBSY}$	27	I/O Busy	An active low input indicating that another source is using the I/O bus. An active low output is provided when the F9447 is about to or is executing an I/O cycle.
$\overline{DCHEX}$	29	Data Channel Execute	An active low output indicating that a data channel transfer is in progress.
Data-Channel Handshake			
$\overline{DCHA}$	18	Data Channel Address	An active low timing strobe output that defines address transfer time of a data channel cycle.
$\overline{DCHI}$	20	Data Channel In	An active low timing strobe output that defines data transfer in (write to memory) of a data channel cycle.

Mnemonic	Pin No.	Name	Description
$\overline{\text{DCHO}}$	19	Data Channel Out	An active low timing strobe that is output during data transfer out (write to peripheral) of a data channel.
DCHMO	23	Data Channel Mode Out	An active high input results in data channel output cycles.
I/O Synchronization			
RQENB	7	Request Enable	A timing output that synchronizes interrupt and data-channel priorities.
Data Buffer Control			
$\overline{\text{DIEN}}$	21	Data In Enable	An active low output that enables peripheral data onto the F9445 information bus during programmed I/O or data-channel input cycles.
$\overline{\text{DOEN}}$	25	Data Out Enable	An active low output that enables information bus data onto the peripheral data bus during programmed I/O or data-channel output cycles.
Console Control Input			
$\overline{\text{CONSW}}$	15	Console Switch	An active low input with internal 2.4K-ohm pullup resistor to V_{CC} and digital delay of approximately 3 ms to eliminate contact bounce.
$\overline{\text{CONSTP}}$	12	Console Step	An active low input with characteristics of CONSW that initiates a console request lasting for two console code cycles of the F9445.
$\overline{\text{APLSW}}$	14	Auto Program Load Switch	An active low input with characteristics of CONSW and CONSTP that initiates a console request cycle with APL enable active.
Console Control Output			
$\overline{\text{CONEN}}$	63	Console Enable	An active low output to enable a console to provide information to the IB of the F9445 during a read or write operation.
$\overline{\text{CONREQ}}$	5	Console Request	An active low output to the F9445 to request console service.
$\overline{\text{CONCD}}$	6	Console Code	An active low output to enable the console code onto the IB of the F9445 in response to a console code cycle of the F9445.
$\overline{\text{APLEN}}$	4	Auto Program Load Enable	An active low output initiated by an APLSW input or the execution of (DOA ac, CPU) instruction and terminated by the execution of a (DOAP ac, CPU) instruction or system reset.
$\overline{\text{CONLD}}$	3	Console Load	An active low output to enable the console to latch data from the IB of the F9445.

Table 1 F9447 Signal Descriptions, Cont'd.

Mnemonic	Pin No.	Name	Description
I/O Port Status Input			
$\overline{\text{MSKBIT}}$	63	Mask Bit	The local logic contains an interrupt disable flag loaded from a select bit of the F9445 information bus during the execution of a mask out instruction. The MSKBIT signal is driven by that selected bit of the IB; a low level at the beginning of MSKO execution sets the interrupt disable flag.
STRBSY	15	Strobe Busy	A negative-going input edge that strobes the BUSY flag to the clear state.
STRBDN	12	Strobe Done	A negative-going input edge that strobes the DONE flag to the true state.
INTPIN	14	Interrupt Priority In	An active high input that determines which peripheral device may interrupt.
I/O Port Status Output			
BUSY	5	Busy Flag	An active high output of a flip-flop is set by the execution of an I/O-Start cycle with I/O Chip Select (IOCS) high; and cleared by a similar I/O-Clear cycle, by a negative transition on STRBSY, by execution of an IORST instruction, or by a low level on MR.
DONE	6	Done Flag	An active high output of a flip-flop is set by a negative transition on STRBDN; cleared by the execution of an I/O-Start or I/O-Clear while I/O Chip Select (IOCS) is high, by the execution of an IORST instruction, or by a low level on $\overline{\text{MR}}$.
INTPOUT	4	Interrupt Priority Out	An active high output that determines which peripheral device may interrupt.
$\overline{\text{CONLD}}$	3	Console Load	An active low output that enables the console to latch data from the F9445 information bus.
Power			
V_{CC}	64	Power Supply	Nominal +5 V DC.
I_{INJ}	32	Injection Current	Constant current (80 mA) obtained by using a dropping resistor from V_{CC} (nominal V_{INJ} = 1.3 V). Use of bypass capacitor to GND is desirable.
GND	16, 48	Ground	Common power and signal return.

FAIRCHILD
A Schlumberger Company

F9448
Programmable Multiport Interface

Advance Product Information

Microprocessor Product

Description

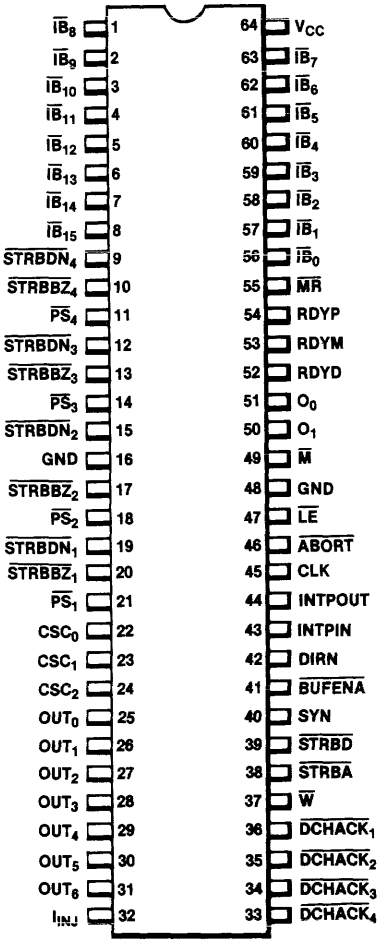
The F9448 Programmable Multiport Interface (PMI) is 64 pin bipolar I^3L device that facilitates the interface between an F9445 16-bit bipolar microprocessor and many industry-standard input/output (I/O) devices. It decodes I/O instructions and memory addresses from the central processing unit (CPU) to communicate with devices tied to its four external ports. When used with the F9449 Multiple Data Channel Controller, it handles peripheral selection and timing during data channel cycles. Some of the features of the F9448 are:

- **Four Independent I/O Ports**
- **Memory-Mapped and Programmed I/O**
- **Interface with Serial, Parallel, DMA, and Other Special-Purpose Devices**
- **Programmable Peripheral Timing**
- **Compatibility with Many Industry-Standard Interfaces**
- **Ability to Implement F9445-Programmed I/O Flags**
- **Interrupt Arbitration and Response Handling**
- **Fabricated in I^3L® Bipolar VLSI Technology**
- **Operating Temperature Range −55° to +125°C**
- **64-Pin Package**
- **Low-Power Schottky Compatible I/O**

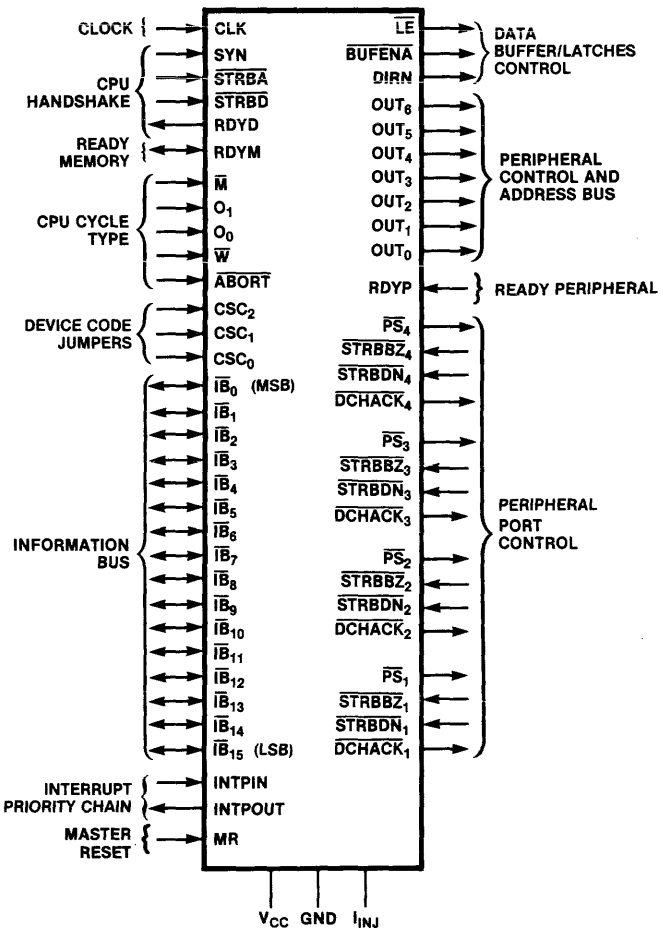
The F9448 PMI ties the F9445 CPU to many industry-standard microprocessor interfaces. It easily links I/O devices designed for the F6800 or 8080 buses and those directly suitable for the F9445 I/O bus. The system configuration in figure 1 shows how the F9448 can be used to interface F6800, 8086 family I/O devices or the F3870 to the F9445 system.

® I^3L is a registered trademark of Fairchild Camera and Instrument Corp.

Connection Diagram

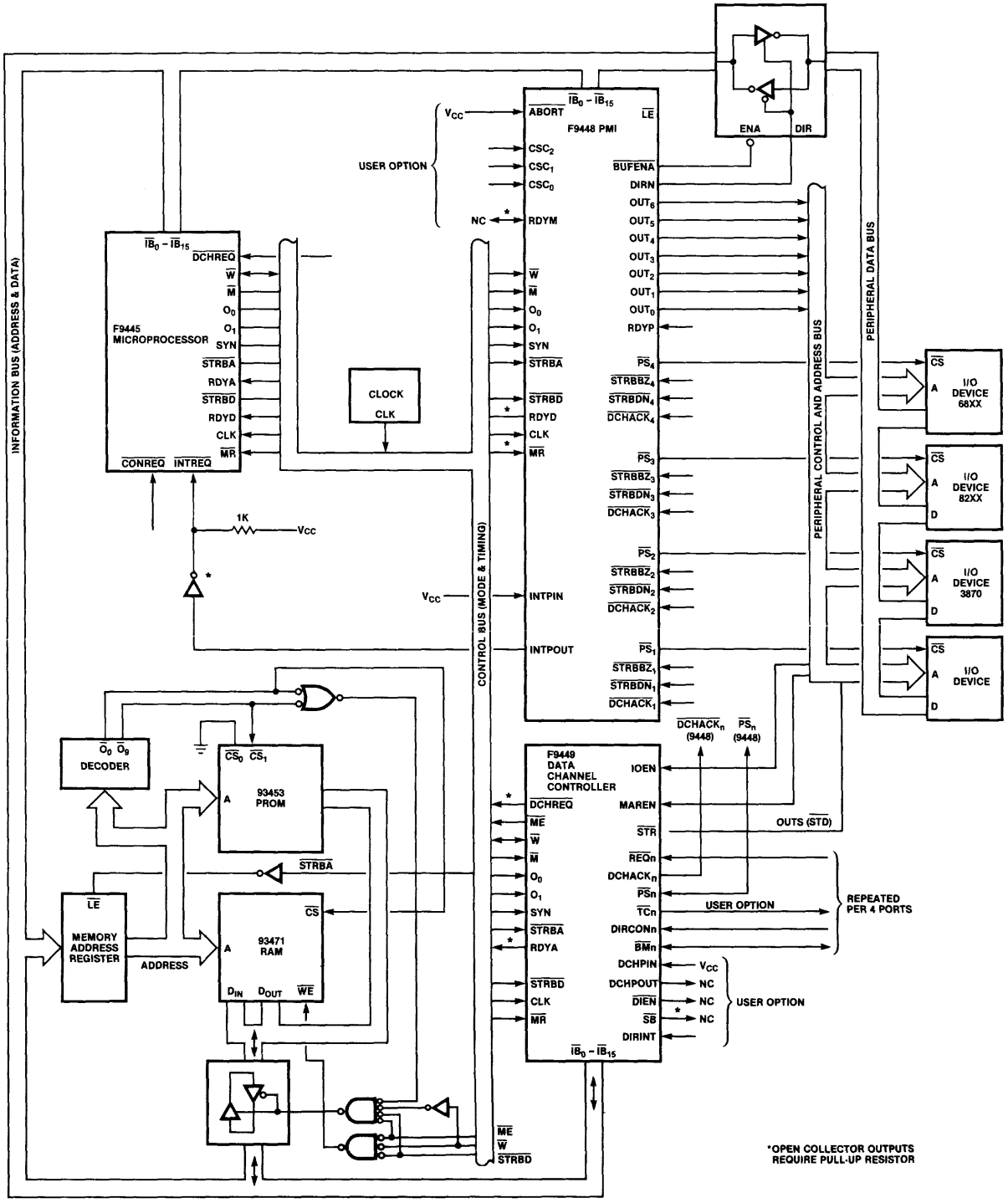


Signal Functions



Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

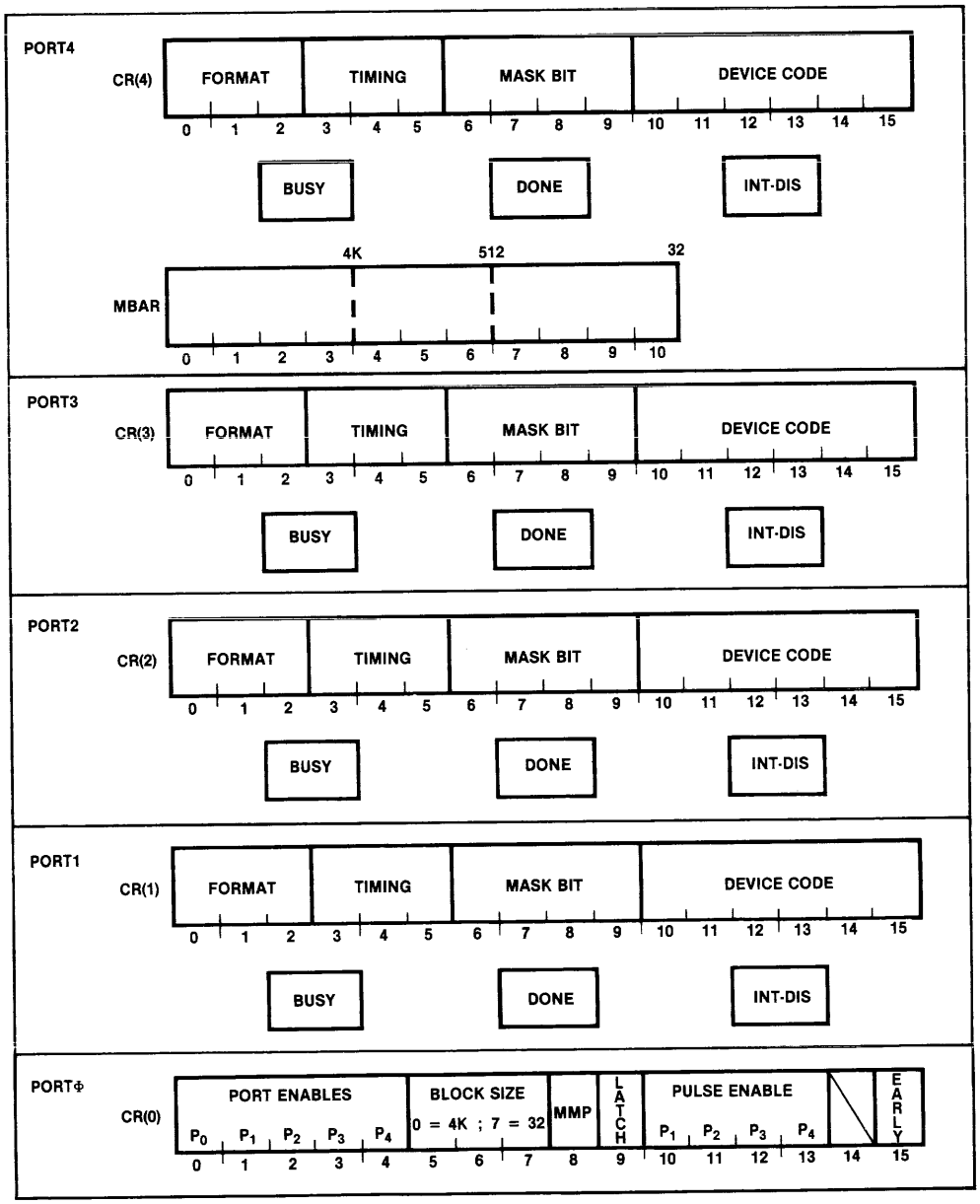
Figure 1 System Configuration



Registers

Each port has a 16-bit configuration register and busy, done, and interrupt-disable flags. Port 4 also has an 11-bit memory-base address register used in conjuction with memory-mapped I/O. Figure 2 shows the F9448 program-accessible registers.

Figure 2 F9448 Software Model



I/O Ports

The F9448 performs selecting and handshaking with connected peripheral devices. It has five bidirectional ports numbered 0 through 4. Port 0 is used as a bootstrap port by the F9445 to read and write, address and configuration registers inside the F9448; ports 1 through 4 are used to communiate with external peripheral devices. Ports 0 through 3 respond only to programmed I/O instructions; port 4 responds to either I/O instructions or memory cycles. This latter feature enables a block of up to 4096 memory addresses to be used for memory-mapped I/O.

Transactions between the F9448 and peripheral devices are organized using several select, address, and timing signals. Out 0-6 signals are shared by all ports, while each set of Peripheral Port Controls is associated with a specific port.

Signal Descriptions

Table 1 describes the signals for the F9448.

Table 1 F9448 Signal Descriptions

Mnemonic	Pin No.	Name	Description
Clock			
CLK	45	Clock	Input signal from the positive-edge-triggered master clock from which all F9448 timing is generated.
CPU Handshake			
SYN	40	Synchronize	An input signal from the F9445 for synchronizing the F9445 with external devices. Active during every CPU cycle.
$\overline{\text{STRBA}}$	38	Strobe Address	An input signal generated by the F9445 during external bus cycles.
$\overline{\text{STRBD}}$	39	Strobe Data	An input signal generated by the F9445 during data transfer time and used by F9448 to organize transfers.
RDYD	52	Data Ready	An active high open collector output signal synchronizing F9448 with F9445 during data transfers. A low level stalls the F9445 until the peripheral is ready.
Ready Memory			
RDYM	53	Memory Ready	Handshake between the memory controller and F9448 during data-channel cycles. Input to F9448 during data-channel read from memory; output from F9448 during data-channel write to memory.
CPU Cycle Type			
$\overline{\text{M}}$	49	Memory	Input status lines from the F9445 indicating the type of bus cycle.
O_1	50	O-Line	
O_0	51	O-Line	
$\overline{\text{W}}$	37	Write	An input signal indicating the direction of data transfer on the IB.
$\overline{\text{ABORT}}$	46	Abort	A low input signal from the Memory Management and Protection Unit that prevents the F9448 from starting another cycle but allows completion of the current cycle.

Table 1 F9448 Signal Descriptions (Cont'd.)

Mnemonic	Pin No.	Name	Description
Device Code Jumpers			
CSC_0-CSC_2	22-24	Chip Select Code	Input signals tied to V_{CC}, GND, SYN, or SYN to define a 6-bit device code, DS_{10} through DS_{15} for port 0 of F9448.
Information Bus			
$\overline{IB}_0$-$\overline{IB}_{15}$	56-63, 1-8	Information Bus	A 16-bit, three-state address and data bus for transmitting information between the F9445 and external devices.
Interrupt Priority Chain			
INTPIN	43	Interrupt Priority Input	An input for determining which peripheral device will respond to an INTA instruction.
INTPOUT	44	Interrupt Priority Output	A priority-signal output from the F9448 to lower-priority devices for determining which peripheral device may interrupt.
Master Reset			
$\overline{MR}$	55	Master Reset	An input signal that initializes the F9448 by clearing all F9448 user-accessible registers to 0 and clearing all busy, done, and interrupt-disable flags.
Data Buffer/ Latches Control			
$\overline{LE}$	47	Latch Enable	An output signal that may be used to load the data from the F9445 on the IB into peripheral data bus latches.
$\overline{BUFENA}$	41	Buffer Enable	An active low output during memory, I/O, or data channel cycles to enable IB transceivers or latches if data is to be transferred between the IB and a device controlled by the F9448.
DIRN	42	Direction	An output signal controlling the direction of any bus transceivers on the IB bus between the F9445 and the F9448 or of any data latches/transceivers between the IB bus and a peripheral data bus.
Peripheral Control And Address Bus			
OUT_0-OUT_6	25-31	Outputs	A 7-bit peripheral output control bus that is to be shared by all peripheral devices controlled by F9448.
Ready Peripheral			
RDYP	54	Ready Peripheral	Open-collector handshake input signal from peripherals to the F9448.

Table 1 F9448 Signal Descriptions (Cont'd.)

Mnemonic	Pin No.	Name	Description
Peripheral Port Control			
$\overline{PS}_1$-$\overline{PS}_4$	21, 18, 14, 11	Port Select	Outputs for selecting the devices being controlled by ports 1 through 4 of the F9448.
$\overline{STRBBZ}_1$-$\overline{STRBBZ}_4$	20, 17, 13, 10	Strobe Busy	A low-to-high transition on the $\overline{STRBBZ}$ input signal clears the associated port's busy flag.
$\overline{STRBDN}_1$-$\overline{STRBDN}_4$	19, 15, 12, 9	Strobe Done	A low-to-high transition on the $\overline{STRBDN}$ input sets the associated port's done flag.
$\overline{DCHACK}_1$-$\overline{DCHACK}_4$	33-36	Data Channel Acknowledge	Active low select inputs from the F9449 data channel controller.
Power			
V_{CC}	64	Power Supply	Nominal +5 V DC.
GND	16, 48	Ground	Ground for both supply and signals.
I_{INJ}	32	Injection Current	A constant current obtainable by use of a dropping resistor from V_{CC} ($V_{INJ} \approx 13$ V) supply. Use of a bypass capacitor to GND is desirable.

FAIRCHILD

A Schlumberger Company

F9449

Multiple Data Channel Controller

Microprocessor Product

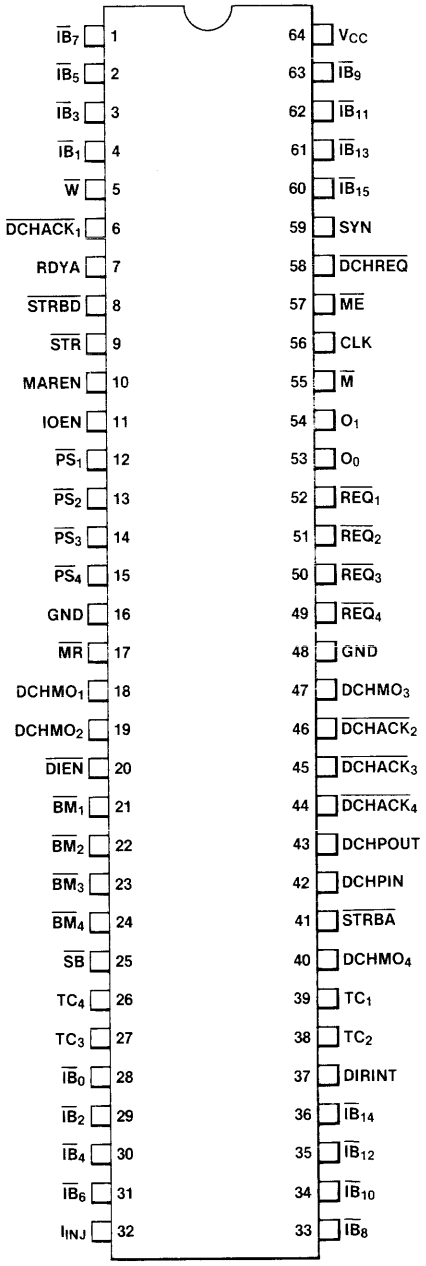
Preliminary Data Sheet

Description

The F9449 Multiple Data Channel Controller is a 4-port controller that is used with the Fairchild F9445 16-Bit Bipolar Microprocessor, and either an F9447 I/O Bus Controller or an F9448 Programmable Multiport Interface, to control direct data transfer to and from memory by peripheral devices. It contains four pairs of program-controlled address and word count registers that are multiplexed to control four fully independent data channels (DCHs) through which data transfers can occur. Data channel transfers are similar to direct memory access (DMA) channel transfers, except that the F9445 architecture time-shares its information bus (IB).

- **Provides Control of Four Independent Channels**
- **Has Separate Word Count and Memory Address Registers for Each Channel**
- **Supports Byte- or Word-mode Operation on Each Channel**
- **Performs Internal Priority Arbitration**
- **Supports Memory-to-Memory Transfers**
- **Implemented in I^3L® Technology, with Low-power Schottky TTL-compatible Input and Output**
- **Available in a 64-Pin Package.**
- **Operating Temperature Range of from −55°C to +125°C**

Connection Diagram



® I^3L is a registered trademark of Fairchild Camera & Instrument Corp.

Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

F9449 Signal Functions

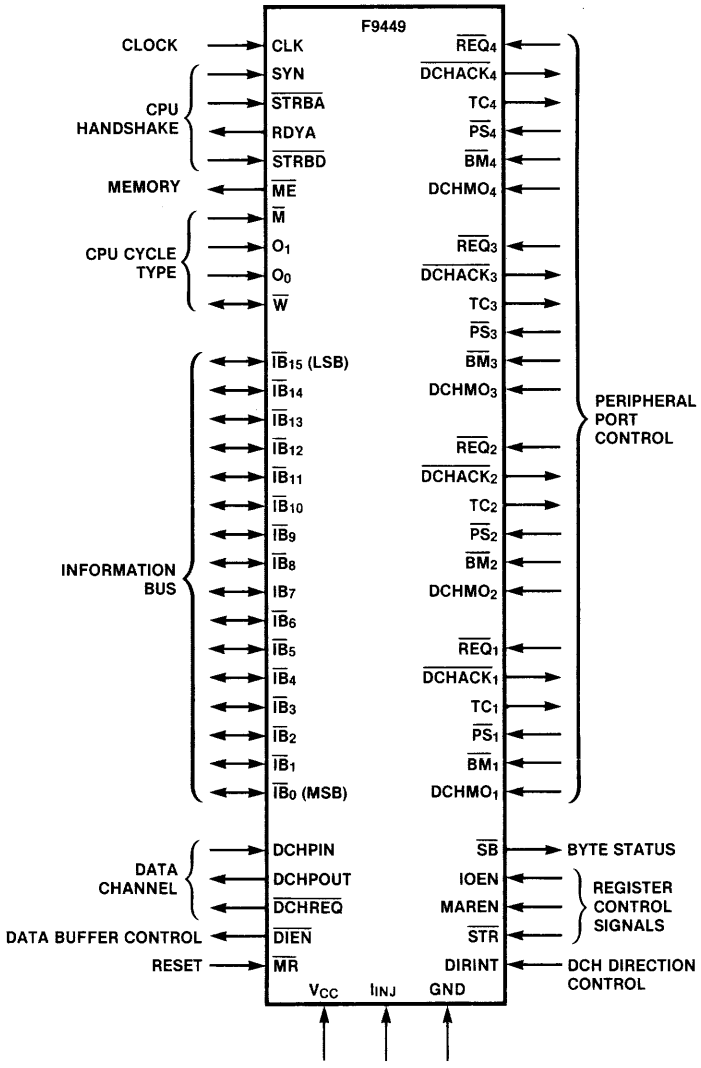
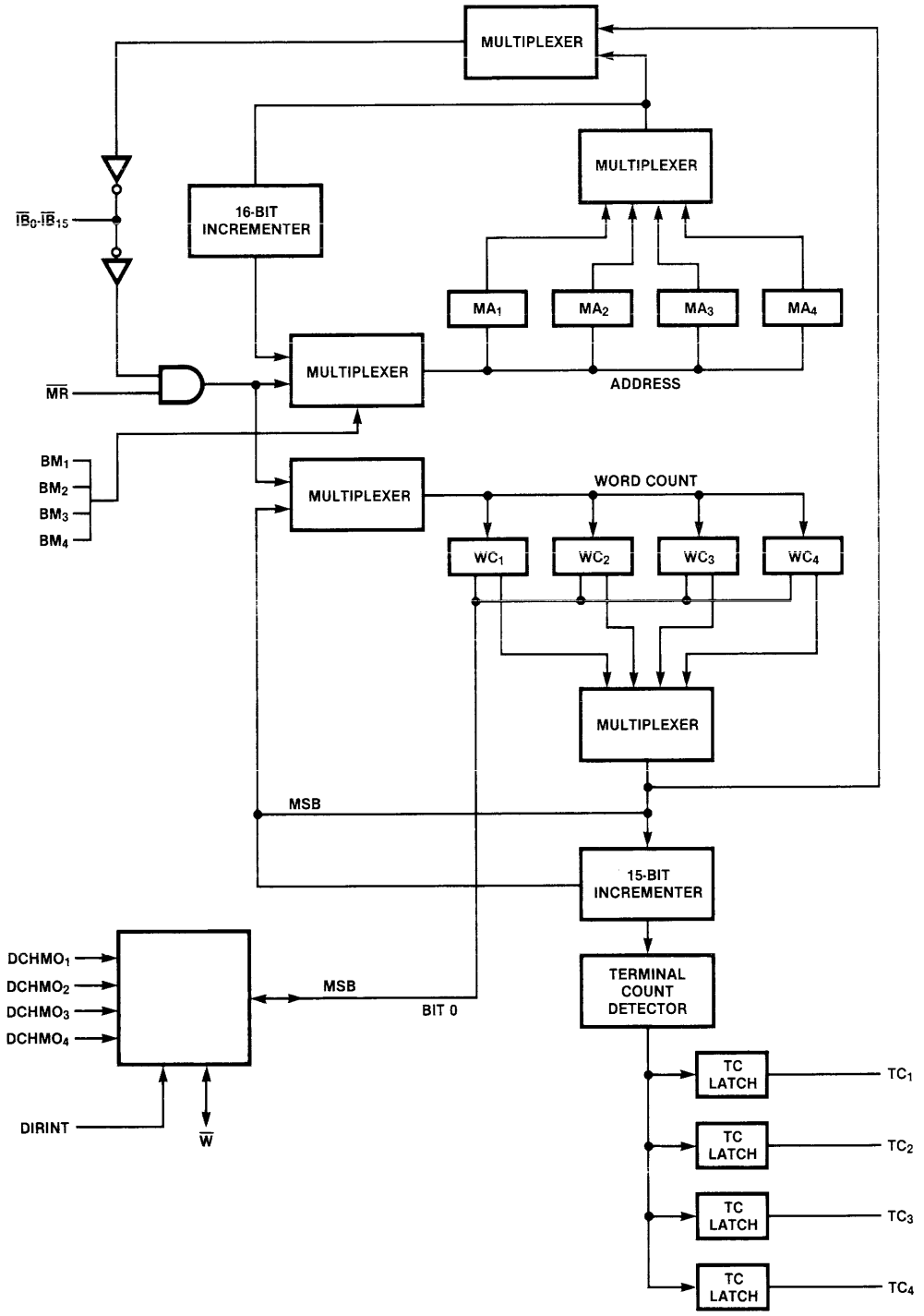


Fig. 1 F9449 Block Diagram

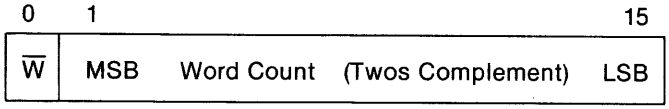


Register Operation

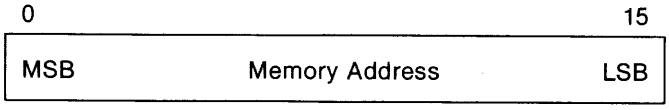
The eight F9449 registers (MA_{1-4}, WC_{1-4}), shown in figure 1, provide four fully independent data channels, numbered 1 to 4, each of which is capable of transferring up to 32K 8-bit bytes or 16-bit words, depending upon whether it is strapped for byte-mode or word-mode operation. Figure 2 illustrates the data formats of these registers.

A word count (WC) register associated with each channel contains the number of bytes or words to be transferred by that channel. The WC registers, which are loaded with the twos complement of the number of bytes or words, are automatically incremented after each DCH cycle (i.e., after every byte or word transfer), regardless of operating mode. When a WC register increments from all ones to zero, a terminal count (TC) signal for that port is set by the F9449. This is normally wired to the F9447 bus controller or F9448 multiport interface to generate an interrupt to the F9445. (Figures 3 and 4 illustrate system configurations using the F9447 and F9448, respectively.) The TC signal can optionally be used to terminate any further requests from that peripheral channel.

Figure 2 Register Data Formats



Format for word- or byte-count register load (MAREN = 0). Word count range is from -2^{15} to 1 (loaded with twos complement). $\overline{W}$ is an internal direction bit: 0 is from peripheral to memory; 1 is from memory to peripheral.



Format for memory address register load (MAREN = 1). Address range is from 0 to $2^{16}-1$.

A memory address (MA) register associated with each channel contains the address at which the next transfer is to occur; each MA register provides a 16-bit address space (0 to 65535). The MA registers are incremented after each transfer in word mode and after every second transfer in byte mode.

F9449 I/O Cycle

The F9449 registers are under software control. They are loaded with starting address and word count information through F9445 programmed output instructions, which are decoded by the F9447 I/O controller or F9448 multiport interface. The F9445 also generates the clock (CLK), synchronize (SYN), address strobe ($\overline{STRBA}$), and data strobe ($\overline{STRBD}$) bus timing signals. (Figure 5 illustrates the I/O cycle timing; refer to the "Timing Characteristics" section for a description of the cycle characteristics and specifications.)

The F9447 or F9448 selects the register to be loaded by generating the appropriate port select ($\overline{PS}$) signal, together with input/output enable (IOEN) and strobe ($\overline{STR}$) signals, as shown in table 1. The low-to-high transition of the STR signal during the write time loads the addressed port WC or MA register, selected by the memory address enable (MAREN) signal, with data from the information bus.

All eight registers are cleared when master reset ($\overline{MR}$) goes low to allow hardware implementation of auto load/bootstrap routines (i.e., to fill the memory beginning at address 0).

The F9445 can read the contents of any register by means of a programmed input instruction, which is decoded by the F9447 or F9448 in the same manner as the output instruction.

F9449 DCH Cycle

A peripheral device requests service by asserting its request ($\overline{REQ_n}$) line to the F9449, which then determines priority and generates a data channel request ($\overline{DCHREQ}$) signal to the F9445. After completing its current program instruction, the F9445 responds to the $\overline{DCHREQ}$ or data channel request performing a DCH cycle, which is a long bus cycle similar to an F9445 memory cycle, but with the information bus and the write ($\overline{W}$) line not driven.

The F9445 sets the bus control lines as follows: $\overline{M}$ high, O_1 low, and O_0 high (i.e., $\overline{M}$, O_1, O_0 to 101). It then generates the CLK, SYN, $\overline{STRBA}$ and $\overline{STRBD}$ bus timing signals. The high-to-low transition of $\overline{STRBA}$ latches the priority resolution logic, and starts the internal DCH sequence of the F9449. The F9449 asserts the appropriate data channel acknowledge ($\overline{DCHACK_n}$) line to signal the requesting peripheral and the F9447 or F9448 that a DCH cycle for it has begun. This may cause the peripheral to remove the REQ_n signal. (Figure 6 illustrates the DCH cycle timing; refer to the "Timing

Fig. 3 F9449/F9447 Configuration

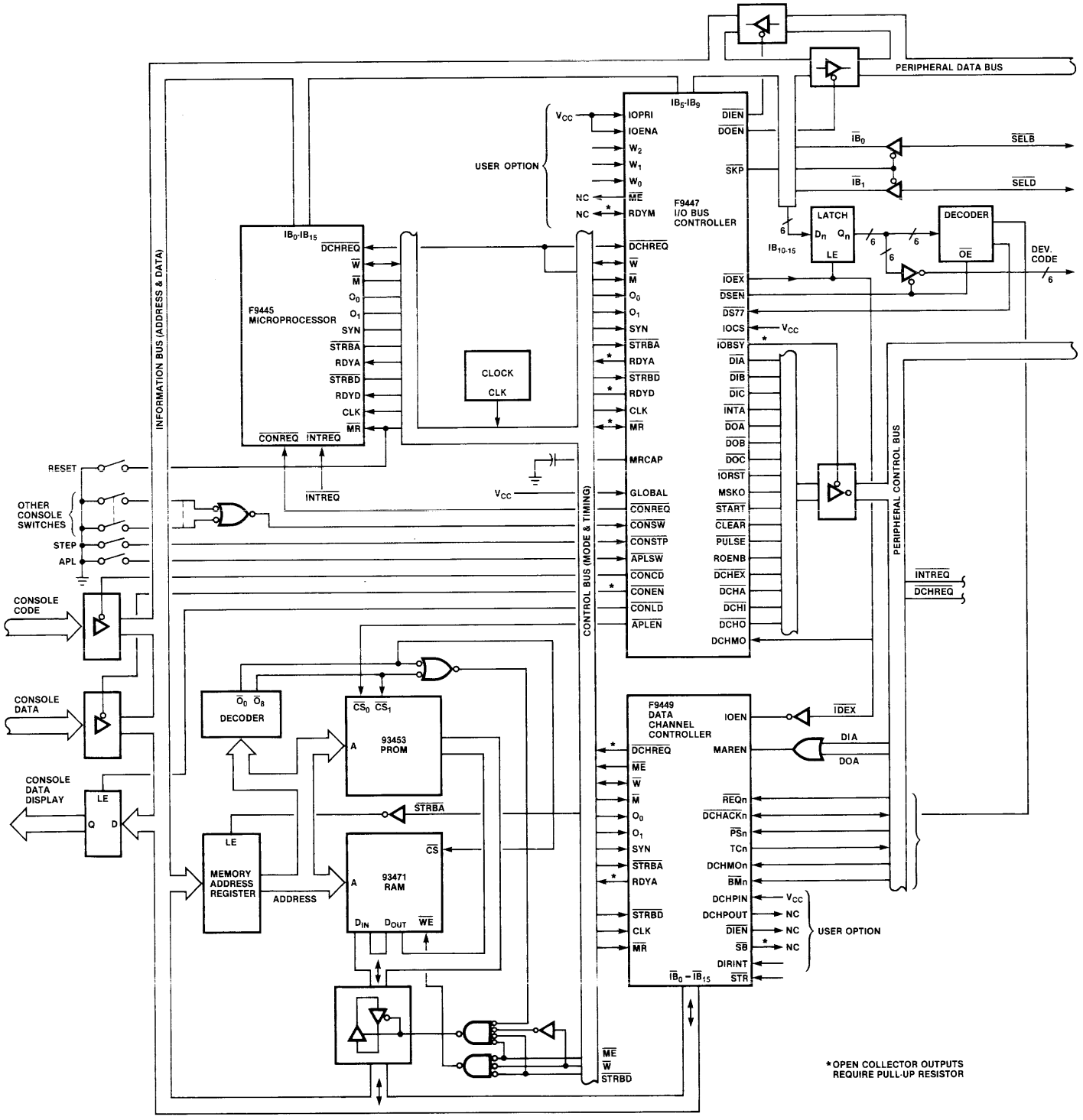


Fig. 4 F9449/F9448 Configuration

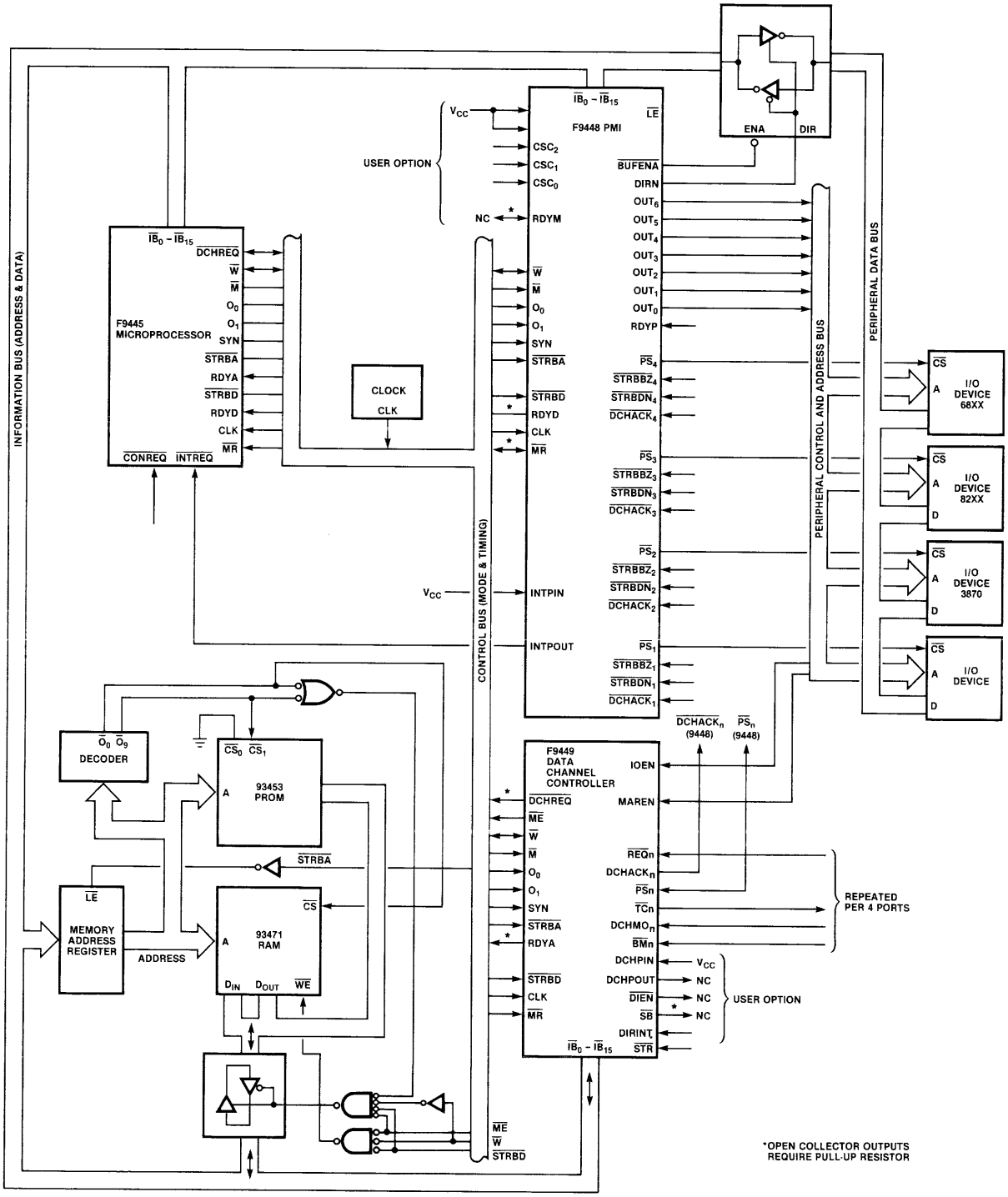


Table 1 F9449 I/O Control

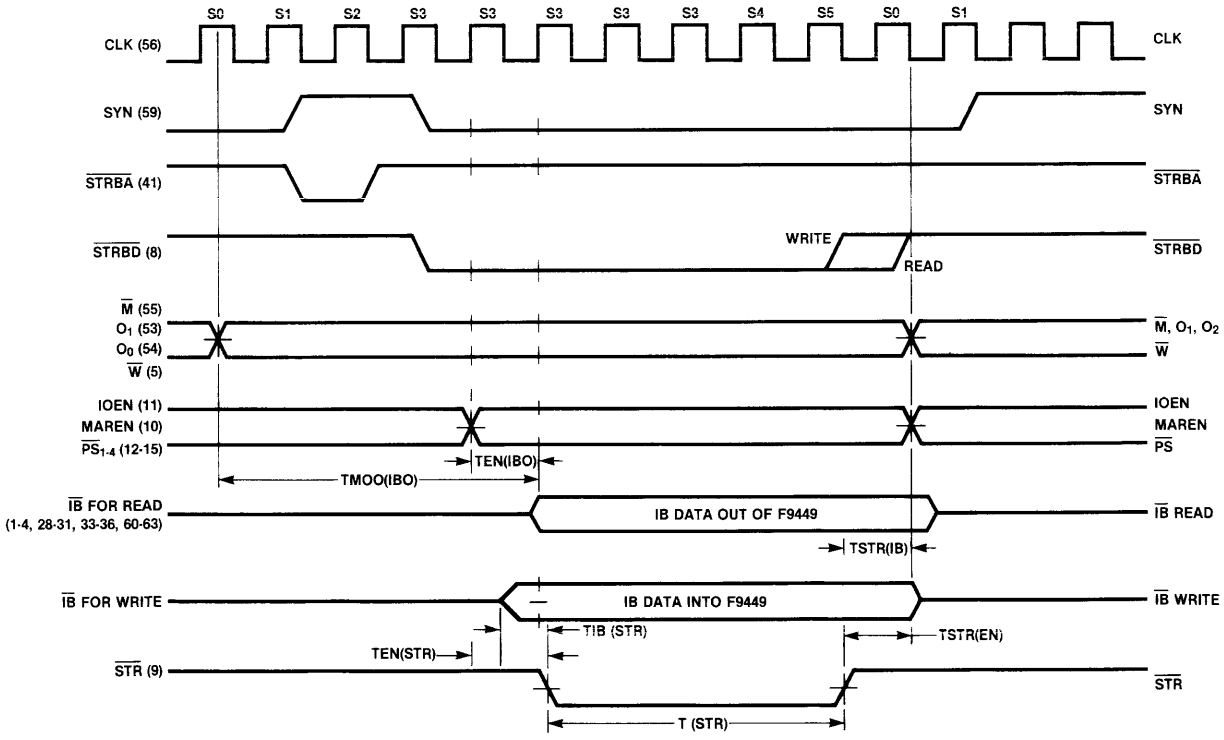
Signal State								Operation Performed
IOEN	$\overline{W}$	MAREN	$\overline{PS_1}$	$\overline{PS_2}$	$\overline{PS_3}$	$\overline{PS_4}$	$\overline{STR}$	
0	X	X	X	X	X	X	X	No operation
X	X	X	1	1	1	1	1	No operation
1	0	0	*	*	*	*	U	Loads IB data into selected word count register
1	0	1	*	*	*	*	U	Loads IB data into selected memory address register
1	1	0	*	*	*	*	X	Loads selected word count register data onto IB
1	1	1	*	*	*	*	X	Loads selected memory address register data onto IB

*One active-low input, selected by programmed I/O instruction device code.

Note
Multiple port selects result in unpredictable results.

0 = Low
1 = High
X = Don't care
U = Low-to-high transition

Fig. 5 Programmed I/O Timing



Characteristics" section for a description of the cycle characteristics and specifications.)

During the address phase of a DCH cycle, the F9449 determines which peripheral is to be served, places the contents of the appropriate MA register onto the information bus and drives the $\overline{W}$ line to the memory controller so that the memory performs either a read or a write cycle. If internal direction (DIRINT) is high, read/write selection can be programmed from the F9445 as the most significant bit of the WC register contents load. When this bit is at a logic 0 it causes a read of memory. If the DIRINT pin is low, the read/write selection is controlled by the peripheral through a data channel mode ($DCHMO_n$) signal. A high $DCHMO_n$ signal indicates a memory read (DCH out operation). If required, the data in enable ($\overline{DIEN}$) and second byte ($\overline{SB}$) lines are also asserted at this time.

The F9449 RDYA signal causes the microprocessor to generate three additional F9445 address strobe (S1G) states, allowing the address sufficient time to propagate from the F9449 to the memory controller.

The F9449 does not actually perform the data transfer between memory and peripheral. Instead, the end of the STRBA signal causes the F9449 to stop driving the address onto the information bus and allows the F9447 or F9448 to provide data control during the data phase of the DCH cycle. It enables a peripheral three-state input buffer, or strobes data out from the IB into the peripheral. The data phase of the DCH cycle can be extended as required by additional data (S3) states generated from the F9445 in response to the F9447 or F9448 data ready (RDYD) output being low.

Because it must communicate with the peripheral during F9445 programmed I/O cycles, the F9447 or F9448 normally accommodates the data timing peculiarities of the peripheral.

The end of the data strobe ($\overline{STRBD}$) causes the WC and MA registers to increment, a TC signal to be asserted (if the WC register has reached zero), and terminates the $\overline{W}$, $\overline{DCHACK_n}$, RDYA, and $\overline{SB}$ signals.

Priority Arbitration

The F9449 arbitrates DCH requests from multiple peripherals on a fixed-priority basis, with channel 1 having the highest priority and channel 4 the lowest. The priority arbitration scheme allows cascading of up to four F9449 controllers by interconnecting the data channel priority out (DCHPOUT) of a higher priority controller to data channel priority in (DCHPIN) of the next, thereby permitting the system to serve a total of 16 data channel peripherals.

Priority resolution occurs during every cycle, at the high-to-low transition of the SYN signal. In a multiple-F9449 system, all pending $\overline{REQ_n}$ inputs are latched at that time, and the DCHPIN/DCHPOUT signals ripple from device to device.

Priorities are reestablished during every cycle, including "short" F9445 cycles. Additional states are generated by the F9449 address ready (RDYA) signal to allow priority ripple when the F9445 responds to a DCH request from a "wait" cycle.

Signal Descriptions

The F9449 input and output signals are described in table 2.

Timing Characteristics

The timing characteristics of the F9449 are illustrated in figure 5 (Programmed I/O Timing) and figure 6 (Data Channel Cycle Timing).

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 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".

Figure 6 Data Channel Cycle Timing

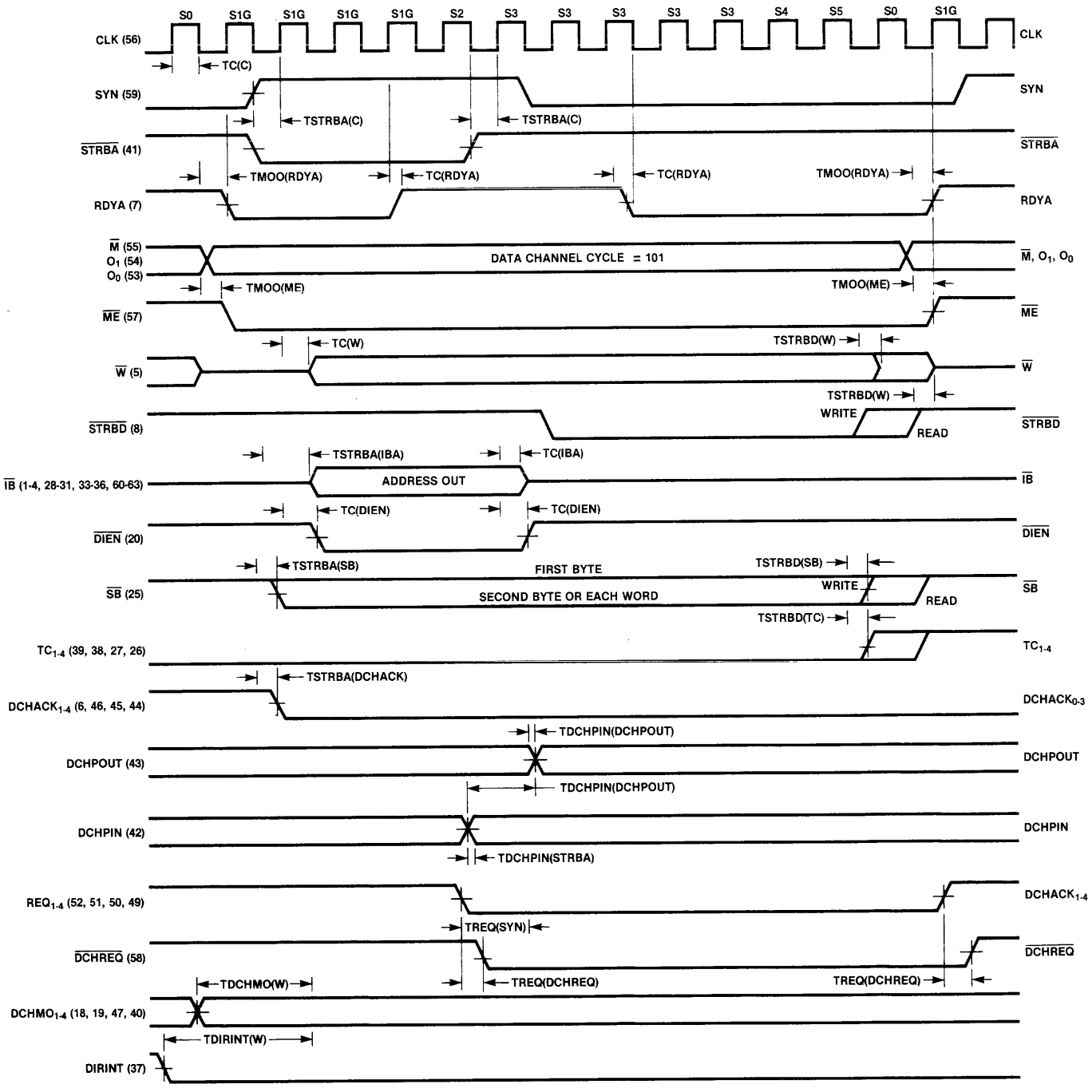


Table 2 F9449 Signal Descriptions

Mnemonic	Pin No.	Name	Description
Clock			
CLK	56	Clock	An input signal from the F9445. The rising edge of the single-phase system clock causes action in the F9449. This line can also be single-stepped for debugging.
CPU Handshake			
SYN	59	Synchronize	An active-high input signal from the CPU that maintains system timing. The start of SYN indicates the start of a CPU cycle with valid $\overline{M}$, O_1, O_0 code.
$\overline{STRBA}$	41	Address Strobe	An active-low input signal from the F9445. In a DCH cycle, the low-to-high transition is used by the memory controller to strobe the memory address from the IB into the selected MA register. (This can be delayed indefinitely by RDYA.)
RDYA	7	Address Ready	An active-high open-collector output signal to the F9445. A low level prolongs the $\overline{STRBA}$ signal to allow time for the F9449 to perform priority resolution and propagate the memory address to the memory controller over the IB.
$\overline{STRBD}$	8	Data Strobe	An active-low input signal from the F9445. The low-to-high transition during a DCH cycle causes the selected WC and MA registers to increment and the $\overline{W}$, $\overline{DCHACK}_n$, RDYA and SB signals to terminate.
Memory			
$\overline{ME}$	57	Memory Enable	An active-low output signal to the memory controller. When low, it informs the memory controller that either the F9445 $\overline{M}$ is low or a DCH cycle is in progress.
CPU Cycle Type			
$\overline{M}$	55	Memory	An active-low input signal from the F9445 that serves as a status indicator. When it is low, the F9445 is performing a memory cycle.
O_1 O_0	54 53	Memory or I/O Function	Active-high "O" line input signals from the F9445, used with the $\overline{M}$ input to indicate the type of cycle the F9445 is performing. During a DCH cycle, $\overline{M}$, O_1, O_0 are set at 101.
$\overline{W}$	5	Write	An active-low input/output signal to and from the components in the system, normally driven by the F9445 to control the read and write operations. Placed in a high-impedance state by the F9445, during a DCH cycle, when it is driven by the F9449, it is low if the system is writing to memory and high if the system is reading from memory.

Table 2 F9449 Signal Descriptions (Cont.)

Mnemonic	Pin No.	Name	Description
Information Bus Signals			
$\overline{IB}_0$-$\overline{IB}_{15}$	1-4 28-31 33-36 60-63	Information Bus	A set of 16 input/output signals to and from the system. This active-low, bidirectional bus is used to load and examine the contents of the selected WC and MA registers. These signals are driven by the selected MA register during the $\overline{STRBA}$ state of an F9449 DCH cycle. The most significant bit is $\overline{IB}_0$; the least significant bit is $\overline{IB}_{15}$.
Data Channel			
DCHPIN	42	Data Channel Priority Input	An active-high input signal from a higher-priority F9449 that is used to extend priority resolution logic throughout a multiple-F9449 system. When this signal is low, it prevents the F9449 from being in a DCH cycle. The highest priority F9449 should have DCHPIN connected high.
DCHPOUT	43	Data Channel Priority Output	An active-high output signal to a lower-priority F9449 that is used to extend priority resolution logic throughout a multiple-F9449 system. When the signal is high, none of the four channels are requesting a DCH cycle and DCHPIN is high.
$\overline{DCHREQ}$	58	Data Channel Request	An active-low open-collector output used by the F9449 to request a DCH from the F9445. Multiple simultaneous requests will be sorted by priority resolution logic during a DCH cycle and will result in additional consecutive DCH cycles. A low level requests a data channel cycle.
Data Buffer Control			
$\overline{DIEN}$	20	Data in Enable	An active-low output signal that can be used to enable an optional bus transceiver placed between the F9449 and the IB. When low, the F9449 is putting out an address during a DCH cycle or data during an I/O read operation.
Reset			
$\overline{MR}$	17	Master Reset	An input signal that is active-low from a power-up, front-panel, or programmed initialization signal. It is used to load the WC and MA registers with zeros, set the internal direction control bit to zero, set the four TC signal lines high and clear the four $\overline{DCHACK}$ lines.
Peripheral Port Control			
$\overline{REQ}_1$-$\overline{REQ}_4$	52 51 50 49	Port Request	A set of four active-low input signals from the corresponding requesting peripherals. A low signal on a $\overline{REQ}_n$ line indicates that its associated peripheral wishes a DCH cycle. Priority resolution logic arbitrates multiple requests and generates a single acknowledge, $\overline{REQ}_1$ having the highest priority and $\overline{REQ}_4$ the lowest.

Table 2 F9449 Signal Descriptions (Cont.)

Mnemonic	Pin No.	Name	Description
$\overline{DCHACK}_1$- $\overline{DCHACK}_4$	6 46 45 44	Data Channel Acknowledge	A set of four active-low output signals to the requesting peripherals and to the F9447 or F9448. When low, it informs the appropriate peripheral that its requested DCH cycle is in progress. The $\overline{DCHACK}$ signal is used by the peripheral to clear the $\overline{REQ}_n$ line. It is also used by the F9447 or F9448 and by the peripheral to enable data buffers to and from the IB.
TC_1-TC_4	39 38 27 26	Terminal Count	A set of four active-high output signals to the associated peripherals, indicating completion of a DCH block. When a WC register is incremented to zero during the last phase of a DCH cycle, the corresponding TC line goes high. When the WC register is loaded with any value from the IB during an I/O write operation to the F9449, the corresponding TC line is cleared to low. All four TC signals are set high by a low level on MR.
$\overline{PS}_4$	12 13 14 15	Port Select	A set of four active-low input signals from a programmed I/O device. The IB bits are decoded by an F9448, which outputs a port select signal to the F9449. When low, the associated port is selected during an I/O read or write operation. No more than one PS line should be low at a time.
$\overline{BM}_1$-$\overline{BM}_4$	21 22 23 24	Byte Mode	A set of four active-low input lines that are used to establish operating modes. When strapped low, the corresponding channel is set for 8-bit byte-mode operation; when strapped high, the associated channel is set for 16-bit word-mode operation.
$DCHMO_1$- $DCHMO_4$	18 19 47 40	Data Channel Mode Out	A set of input signals from the requesting peripherals. When DIRINT is low, a $DCHMO_n$ low indicates that the corresponding peripheral is writing to memory during a DCH cycle (IN). When the $DCHMO_n$ signal is high, it indicates that the peripheral is reading from memory (OUT).
BYTE Status			
$\overline{SB}$	25	Second Byte	An active-low open-collector output signal to the memory controller. During $\overline{STRBD}$ timing, this signal is high during the first byte of a byte-mode DCH cycle and low during the second byte of a byte-mode cycle and during every word in a word-mode DCH cycle. It can be used to strobe either the left or right half of the memory array during a $\overline{STRBD}$ operation.
Register Control Signals			
IOEN	11	Input/Output Enable	An active-high input signal from the F9447 or F9448. It is used to enable the F9449 when the F9445 wishes to read from or write to a WC or MA register during an I/O cycle. When the signal is high, a programmed I/O operation is in progress.

Table 2 **F9449 Signal Descriptions** (Cont.)

Mnemonic	Pin No.	Name	Description
MAREN	10	Memory Address Register Enable	An active-high input signal from the F9447 or F9448. It is used to select the source/destination register for an F9445 programmed I/O operation. A high signal selects an MA register; a low signal selects a WC register.
$\overline{STR}$	9	Strobe	An input signal from the F9447 or F9448. The low-to-high transition causes the F9449 to load the IB data into the selected WC or MA register during an I/O write operation.
DCH Direction Control			
DIRINT	37	Internal Direction	Input line that is used to establish the control source of the $\overline{W}$ line. When DIRINT is high during a DCH cycle, the $\overline{W}$ line is controlled internally by IB_0, the most significant bit of the data word in the WC register. When DIRINT is low, the $\overline{W}$ line is controlled externally by the $DCHMO_n$ input from the corresponding requesting peripheral. It may be driven low by selected $\overline{DCHACK}_n$ outputs if some channels need internal control and others external control.
Power			
V_{CC}	64	Power Supply	Supply voltage (+5 Vdc).
I_{INJ}	32	Injection Current	A constant 250 mA current supply; may be derived by use of an external resistor to V_{CC}. (Nominal $V_{INJ} = 1.2$ V.)
GND	16, 48	Ground	Common power and signal return.

Table 3 DC Characteristics

Symbol	Characteristic	Min	Typ	Max	Unit	Test Conditions
V_{INJ}	Injector Voltage.		1.3		V	I_{INJ} = Max
V_{IH}	Input High Voltage.	2.0			V	Guaranteed Input High Voltage
V_{IL}	Input Low Voltage.			0.8	V	Guaranteed Input Low Voltage
V_{CD}	Input Clamp Diode Voltage.		-0.9	-1.5	V	V_{CC} = Min, I_{IN} = -18 mA, I_{INJ} = Min
V_{OH}	Output High Voltage.	2.4	3.2		V	V_{CC} = Min, I_{OH} = -400 μA, I_{INJ} = Min
V_{OL}	Output Low Voltage.		0.2	0.5	V	V_{CC} = Min, I_{OL} = 8.0 mA, I_{INJ} = Min
I_{IH}	Input High Current All Inputs.			1.0	mA	V_{CC} = Max, V_{IN} = 5.5 V, I_{INJ} = 300 mA
I_{IL}	Input Low Current.		-0.21	-0.4	mA	V_{CC} = Max, V_{IN} = 0.4 V, I_{INJ} = Min
I_{OZH}	Output Off (High-Impedance) State High Current IB_0-IB_{15}, W.			100	μA	V_{CC} = Max, V_{OUT} = 2.4 V, I_{INJ} = Min
I_{OZL}	Output Off (High-Impedance) State Low Current IB_0-IB_{15}, W.		-210	-500	μA	V_{CC} = Max, V_{OUT} = 0.4 V I_{INJ} = Min
I_{OSH}	Output Short Circuit Current.	-15		-100	mA	V_{CC} = Max, V_{OUT} = 0.0 V, I_{INJ} = Min*
I_{LOH} O_{HH}	Output Leakage Current (Open Collector) RDYA, SB, DCHREQ.			1.0	mA	V_{CC} = Min, V_{OH} = 5.25 V, I_{INJ} = Min
I_{CC}	Supply Current.		125		mA	V_{CC} = Max, I_{INJ} = Min

*Not more than one output to be shorted at a time.

Absolute Maximum Ratings

These are stress ratings only, and functional operation at these ratings, or under any conditions above those indicated in this data sheet, is not implied. Exposure to the absolute maximum 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.

Storage Temperature	−65 °C, +150 °C
Ambient 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})	+500 mA
Injector Voltage (V_{INJ})	−0.5 V, +2.0 V

Recommended Operating Ranges

Part Number	Supply Voltage (V_{CC})		
	Min	Typ	Max
F9449DC	4.75 V	5.0 V	5.25 V
F9449DM	4.5 V	5.0 V	5.5 V

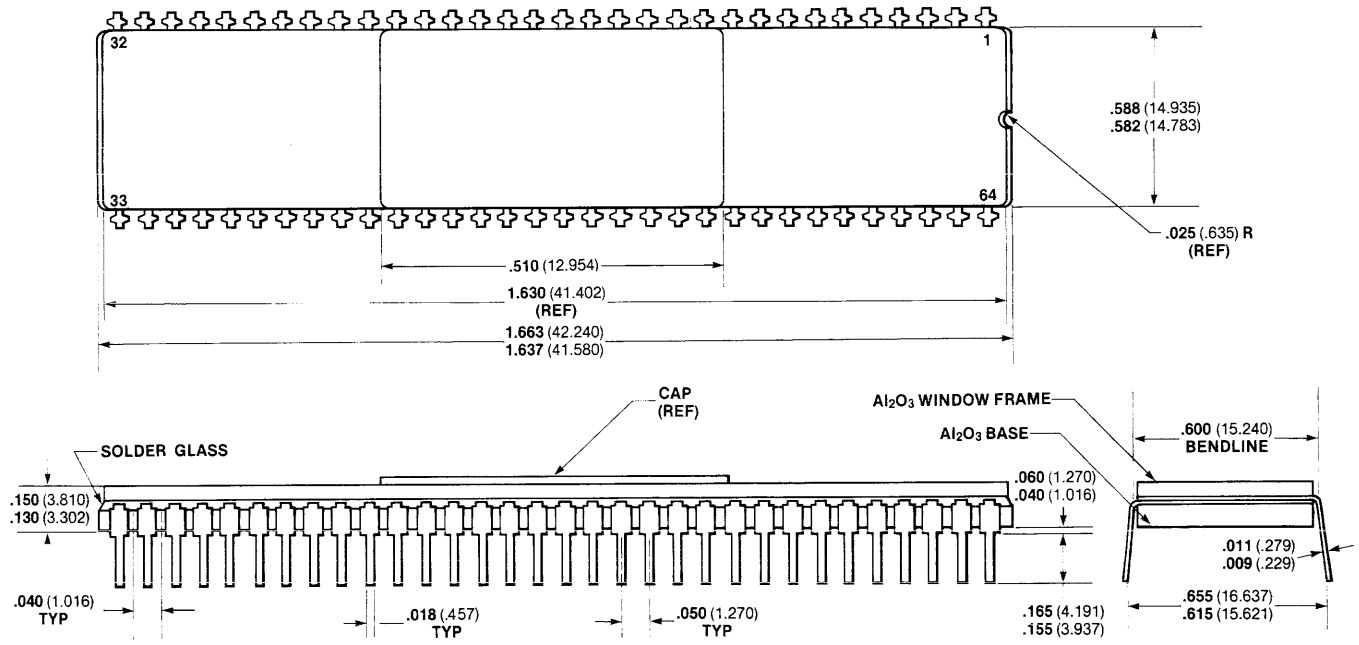
Part Number	Injector Current (I_{INJ})		
	Min	Typ	Max
F9449DC	200 mA	250 mA	300 mA
F9449DM	200 mA	250 mA	300 mA

Ordering Information

Order Code	Temperature Range
F9449DC	0 °C to +75 °C
F9449DM	−55 °C to +125 °C

Package Outline

64-Pin Ceramic Dual In-Line



FAIRCHILD

A Schlumberger Company

F9450 (MIL STD 1750A) 16-Bit Bipolar Microprocessor Family

Advance Information, November, 1982

Microprocessor Products

Features

- **Single-chip microprocessor fully implements MIL-STD-1750A (Notice 1) ISA**
- **High performance over military temperature range - 700 KIPS DAIS mix with floating point - 0.2 μs ADD, 1.85 μs MULTIPLY**
- **Real-time processing - two programmable timers - 16 levels of vectored interrupt**
- **32 and 48-bit floating point arithmetic on chip**
- **Bipolar VLSI I^3L® -II - 1×10^5 radiation**
- **Multiprocessor capabilities**
- **Single and double precision arithmetic**
- **Direct address to 64K words, expandable to 1M words**
- **16 General-purpose registers**
- **Static operation with single clock - 0-20 MHz**
- **TTL inputs and outputs with 8 mA drive capability**
- **Single 5 V supply - injector current source required**
- **64-pin DIPs with 50-mil pin centers**

Description

The F9450 microprocessor, in a single chip, completely implements MIL-STD-1750A (Notice 1) Instruction Set Architecture. This microprocessor is currently being developed as the heart of a high-performance processor family for commercial and military applications requiring high-speed, sophisticated, real-time processing.

Utilizing 16-bit architecture, the F9450 provides 16 user-accessible general purpose registers and performs floating point operations on-chip. The I^3L-II Bipolar VLSI technology affords static operation with 200 ns bus cycle times, LSTTL-compatible input/output, inherent radiation tolerance (1×10^5 rads), and operation at 20 MHz over the full military temperature range.

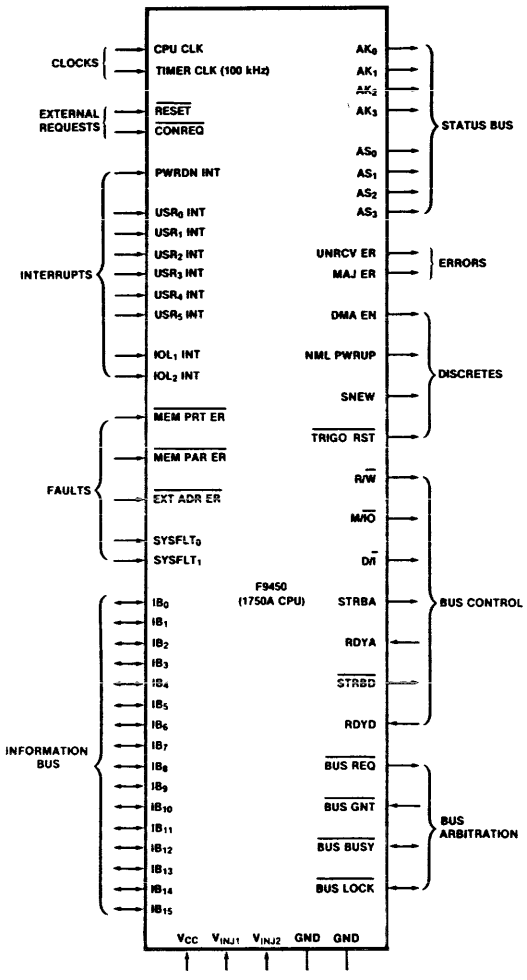
Real-time processing is achieved through advanced design and architecture, incorporating two programmable timers, a complete 16-level interrupt processor, and a comprehensive fault handler on the chip.

Multiprocessing is supported by a flexible bus arbitration scheme as well as process synchronization (test and set) instructions.

Several support circuits and systems can provide additional capability. These include the F9446 Dynamic Memory Controller; the F9451 Memory Management Unit, providing memory-mapped expansion to 1M words; and the F9452 Block Protect Unit. A multi-user development system (FS-1) has been developed, as well as EMUTRAC™, which offers real-time system emulation and debugging. Information on these products is available in the Microprocessor Products Data Book.

Comprehensive software support for the F9450 (MIL-STD-1750A), including assemblers, loaders, simulators and compilers, is available from various sources. An ADA compiler for MIL-STD-1750A is also under development. For a list of available software contact Fairchild - see last page of data sheet.

F9450 Pin Functions



Contents

Fairchild Camera and Instrument Corporation
464 Ellis Street, Mountain View, California 94042

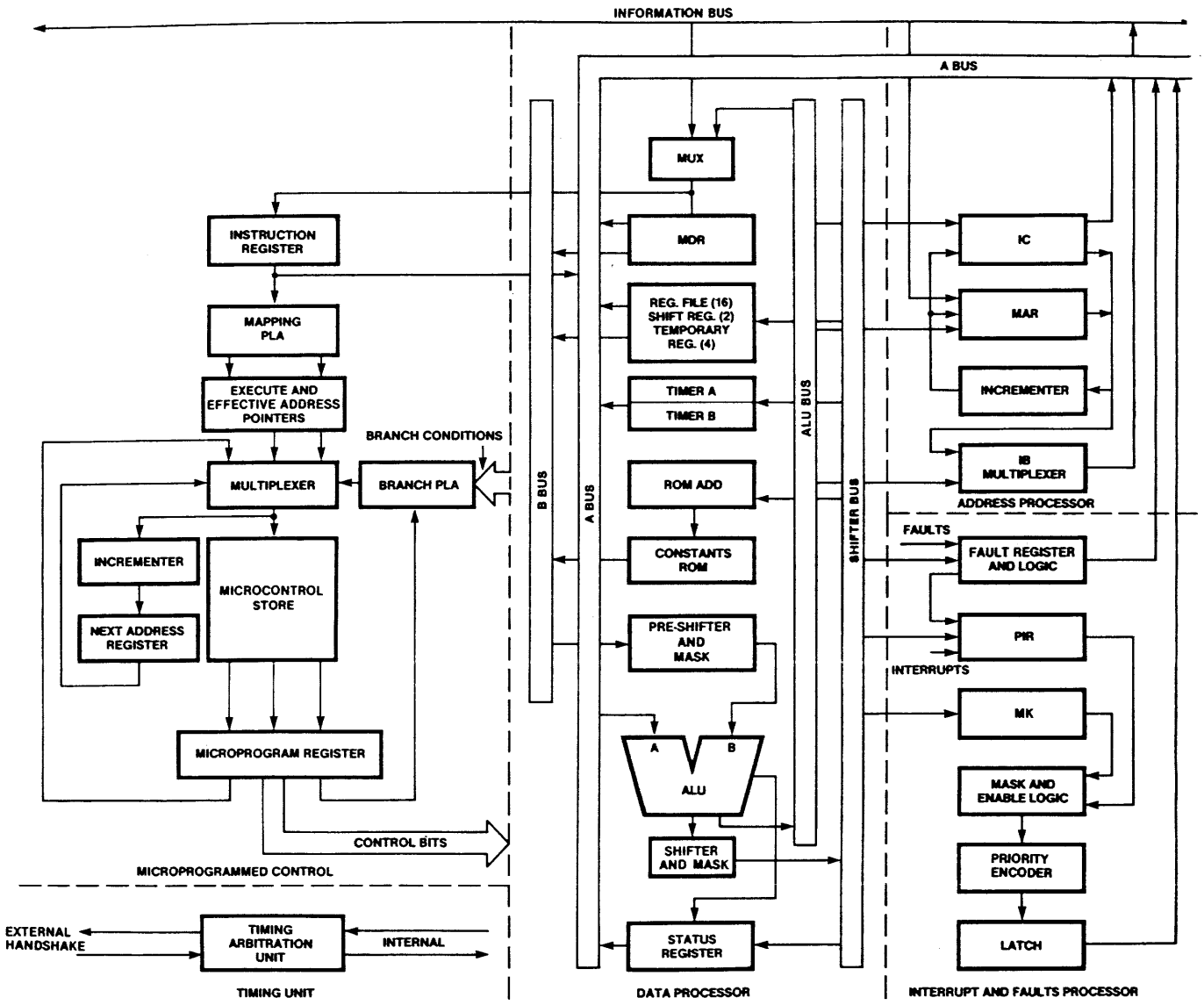


Fig. 1 F9450 Block Diagram

Architecture

There are five main sections of the F9450 microprocessor as represented in the block diagram, *Figure 1*. Each is discussed separately as follows.

Data processor
Microprogrammed control
Address processor
Interrupt and fault processor
Timing unit

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 (R_0 - R_{15})
Six working registers
Memory Data Register (MDR)
Two timers
Constants ROM
Status Register (SR)
Pre-Shifter and Mask

Microprogrammed Control

CPU operation 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 which reside in the micro control store. The micro control store generates three output fields to the micro register. Two of them - Next Address Field and Branch Field - determine the subsequent micro address. The third output field controls operation of all CPU components.

Address Processor

The address processor includes an Instruction Counter (IC) and a Memory Address Register (MAR) which 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 paralleling 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 a Pending Interrupt Register (PIR), a Mask Register (MK), a Fault Register (FT), interrupt enabling logic and a priority encoder. Also included is abort condition detection and activation logic.

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) (refer to the state diagram in *Figure 2*):

(a) A 3-state cycle (S_0, S_4, S_5) for pure internal ALU operations.

(b) A 4-state cycle (S_0, S_1, S_2, S_3) for minimum length bus cycles.

(c) A 5-state cycle (S_0, S_1, S_2, S_3, S_{3A} or S_0, S_4, S_5, S_{5A}, S_{5B}) applies for those cycles that use the result of the current ALU operation to determine the next address in the microprogrammed control store. It also applies to those cycles following an abort condition.

Note that every timing cycle starts with state S_0 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 BUS GNT, RDYA or RDYD external inputs. These signals hold the CPU in states S_Z (high impedance) when the bus is assigned to another CPU or DMA device, S_1 (address phase) or S_3 (data phase) respectively, as shown in the state diagram (*Figure 2*).

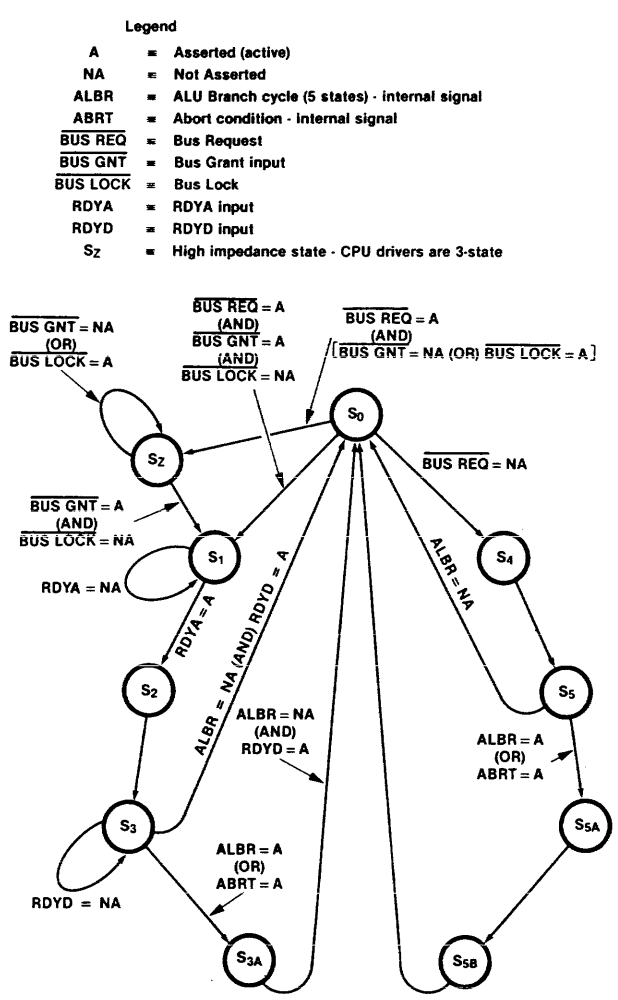


Fig. 2 F9450 Timing Generator State Diagram

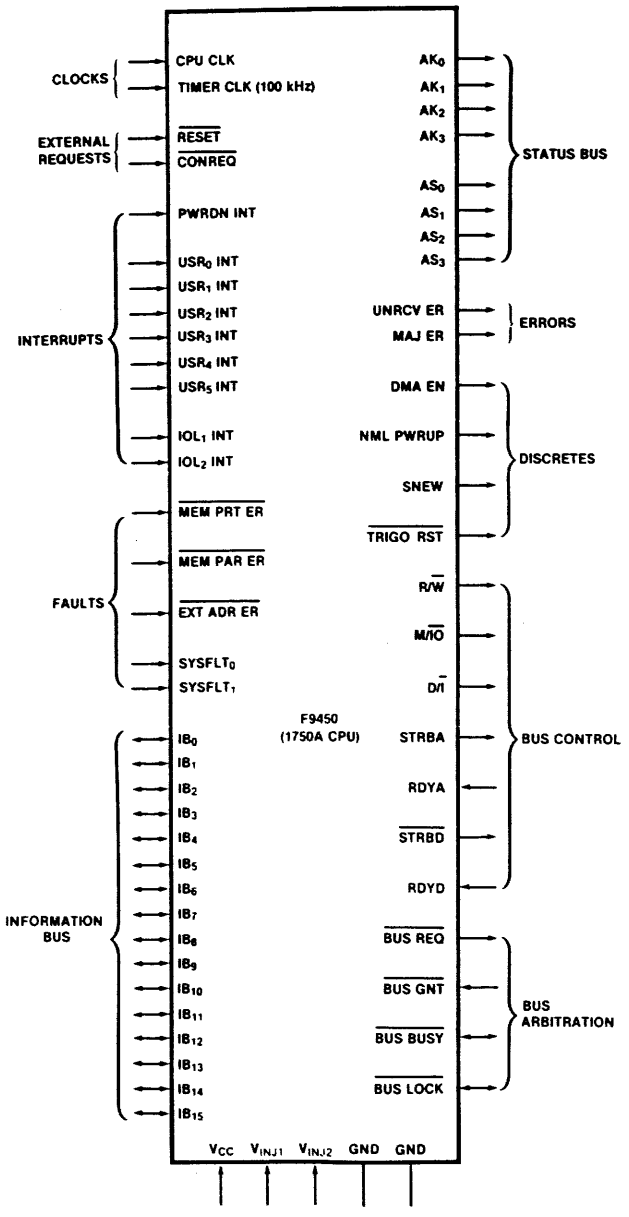


Fig. 3 F9450 Pin Functions

F9450 Signal Descriptions

Clocks

CPU CLK, Pin 51 - CPU clock (0-20 MHz)

TIMER CLK, Pin 7 - Timer clock (100 kHz)

External Requests

$\overline{\text{RESET}}$, Pin 6 - Reset - Active LOW input initializes the CPU.

$\overline{\text{CONREQ}}$, Pin 2 - Console Request - Active LOW input initiates console operations after current instruction.

Interrupts

PWRDN INT, Pin 33 - Power Down Interrupt - Active on the positive-going edge or HIGH level, according to the interrupt mode bit in the configuration register.

USR_0 INT through USR_5 INT, Pins 34 through 39 - User Interrupts - Active on the positive-going edge or HIGH level, according to the interrupt mode bit in the configuration register.

IOL_1 INT and IOL_2 INT, Pins 40 and 41 - Input/Output Level Interrupts - Active HIGH inputs that can be used to expand the number of user interrupts.

Faults

$\overline{\text{MEM PRT ER}}$, Pin 26 - Memory Protect Error - Active LOW input generated by the MMU or BPU or both and sampled by $\overline{\text{BUS BUSY}}$ into the Fault Register (bit 0 if CPU bus cycle, bit 1 if non-CPU bus cycle).

$\overline{\text{MEM PAR ER}}$, Pin 27 - Memory Parity Error - Active LOW input sampled by $\overline{\text{BUS BUSY}}$ into bit 2 of the Fault Register.

$\overline{\text{EX ADR ER}}$, Pin 28 - External Address Error - Active LOW input sampled by $\overline{\text{BUS BUSY}}$ into the Fault Register (bit 5 or bit 8).

$SYSFLT_0$, Pin 29 - System Fault 0 - Active on the positive-going edge setting bit 7 in the Fault Register.

$SYSFLT_1$, Pin 30 - System Fault 1 - Active on the positive-going edge setting bits 13 and 14 in the Fault Register.

Information Bus

IB_0 through IB_{15}, Pins 9 through 18 and 20 through 25 - 16-bit Bus - Active HIGH bidirectional time-multiplexed Address/Data bus; 3-state during bus cycles not assigned to this CPU. IB_0 is the most significant bit.

Status Bus

AK_0 through AK_3, Pins 47 through 50 - Address Key - Active HIGH outputs used to match with the Access Lock in the MMU for memory accesses (a mismatch is one of several possible situations causing the MMU to ... $\overline{\text{MEM PRT ER}}$ LOW).

AS_0 through AS_3, Pins 42 through 45 - Address State - Active HIGH outputs select the page register group in the MMU.

Errors

UNRCV ER, Pin 8 - Unrecoverable Error - Active HIGH output indicating the occurrence of an error classified as unrecoverable. The instruction in which the error occurred is aborted.

MAJ ER, Pin 31 - Major Error - Active HIGH output indicating the occurrence of an error classified as major. The instruction in which the error occurred is aborted.

Discretes

DMA EN, Pin 3 - Direct Memory Access Enable - Active HIGH output indicates that DMA is enabled. It is disabled when the CPU is initialized (RESET).

NML PWRUP, Pin 6 - Normal Power Up - Active HIGH output set when the CPU has completed the Built-In Test in the initialization sequence successfully.

SNEW, Pin 63 - Start New - Active HIGH output indicating that a new instruction will start executing in the next cycle, useful for instruction tracing function.

TRIGO RST, Pin 4 - Trigger Go Reset - Active LOW discrete output.

Bus Control

R/$\overline{W}$, Pin 57 - Read or Write Output - Indicates direction of data flow. A HIGH indicates a read or input operation and a LOW indicates a write or output operation; 3-state during bus cycles not assigned to this CPU.

M/$\overline{IO}$, Pin 59 - Memory or I/O Output - Indicates whether the current bus cycle is Memory (HIGH) or I/O (LOW); 3-state during bus cycles not assigned to this CPU.

D/$\overline{I}$, Pin 58 - Data or Instruction Output - Indicates whether the current bus cycle access is for Data (HIGH) or Instruction (LOW); 3-state during bus cycles not assigned to this CPU.

STRBA, Pin 52 - Address Strobe - Active HIGH output used for latching the memory or XIO address in an external latch at the HIGH-to-LOW transition of the strobe; 3-state during bus cycles not assigned to this CPU.

RDYA, Pin 55 - Address Ready - Active HIGH input used to extend the address phase of a bus cycle.

$\overline{\text{STRBD}}$ - Pin 53 - Data Strobe - Active LOW output used for strobing data in Memory and XIO cycles; 3-state during bus cycles not assigned to this CPU.

RDYD - Pin 56 - Data Ready - Active HIGH input used to extend the data phase of a bus cycle. WAIT states are inserted as long as RDYD is not active to accommodate slower memory devices.

Bus Arbitration

$\overline{\text{BUS REQ}}$, Pin 54 - Bus Request - Active LOW output indicates that the CPU requires the bus; becomes inactive as soon as the CPU has acquired the bus and started the bus cycle.

$\overline{\text{BUS GNT}}$, Pin 61 - Bus Grant - Active LOW input from an external arbiter indicates that 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.

$\overline{\text{BUS BUSY}}$, Pin 60 - Bus Busy - 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.

$\overline{\text{BUS LOCK}}$, Pin 62 - Bus Lock - Active LOW, bidirectional signal used to lock the bus for successive bus cycles. During non-locked bus cycles, $\overline{\text{BUS LOCK}}$ mimics $\overline{\text{BUS BUSY}}$. It is 3-state during bus cycles not assigned to this CPU.

GND, Pins 1 and 32 - Ground

V_{CC}, Pin 64 - Nominal +5 V, 225 mA.

V_{INJ1} and V_{INJ2}, Pins 19 and 46 - Injector Supply - Nominal 1.3 V, 1.4 A, (*Figure 4*).

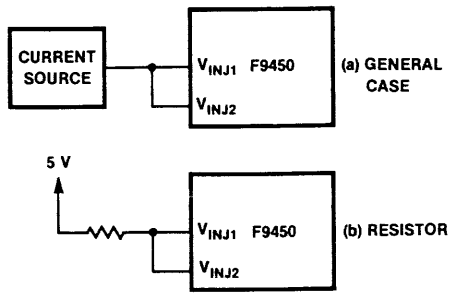


Fig. 4 V_{INJ} Supply.

Bus Transactions

Bus transactions are four 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/IO line.

The master internal clock (NRCK - *Figures 5 and 6*) is active LOW during state S_0. The falling edge of NRCK activates $\overline{\text{BUS REQ}}$ output indicating to the external arbiter (if any) that this CPU requests the bus. At the end of state S_0, the CPU samples the status of $\overline{\text{BUS GNT}}$ input. If this input has been active, satisfying the minimum set up time, state S_1 is entered. Otherwise, the CPU enters the high-impedance state, S_Z, waiting for $\overline{\text{BUS GNT}}$ to be active (In a uniprocessor system, no arbiter is needed and $\overline{\text{BUS GNT}}$ is wired LOW). Simultaneously, the $\overline{\text{BUS LOCK}}$ input should be not active (HIGH) for the CPU to enter state S_1 (*Figure 10*). Once in state S_1, $\overline{\text{BUS REQ}}$ is deactivated to allow other bus contenders bidding early for the next bus cycle. The CPU activates $\overline{\text{BUS BUSY}}$, $\overline{\text{BUS LOCK}}$, status information and outputs the address after some delay measured from the start of S_1 state. At the end of S_1, the CPU samples the RDYA input. If LOW, the CPU stays in the S_1 state (*Figure 7*), extending the address phase on the bus. Otherwise, it proceeds to state S_2 and then unconditionally to state S_3.

Once in state S_2, the CPU drops STRBA LOW (this edge is used to latch the address in an external address latch) and activates $\overline{\text{STRBD}}$ output for Read cycles only, where the CPU gets ready to receive read data by turning the address/data bus around. For Write cycles, the CPU starts outputting the write data immediately after the address. $\overline{\text{STRBD}}$ is activated during S_3 allowing both a reasonable set-up time for write data-to-$\overline{\text{STRBD}}$ falling edge and reasonable hold time for $\overline{\text{STRBD}}$ rising edge-to-write data going away. RDYD is sampled at the end of S_3 and the bus cycle is terminated when RDYD is HIGH; otherwise, it stays in S_3 (*Figures 8 and 9*). At the end of the bus cycle, all CPU outputs are 3-state.

All XIO and VIO commands are echoed back to the external world in a 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. (It is the system's responsibility to provide RDYA and RDYD in these cycles.)

Table 1 depicts the maximum access time required by the system at various operating frequencies. The access time includes address latches, address decoders delays and system memory chip enable access time.

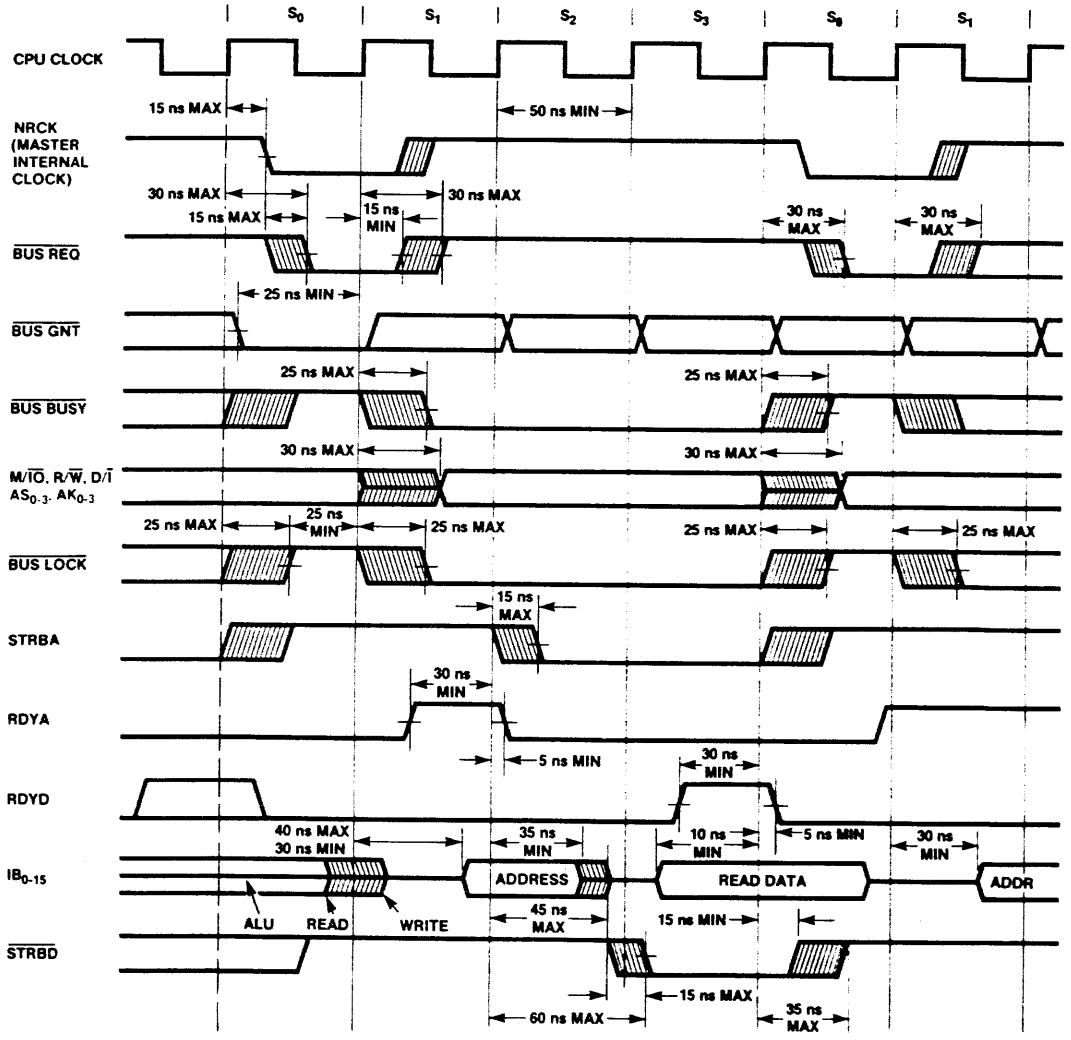


Fig. 5 Minimum Read Bus Cycle

CPU Clock (MHz)	System Memory Address Access Time**(ns)				
	No* Wait	1* Wait	2* Waits	3* Waits	4* Waits
20	100	150	200	250	300
18	117	172	228	283	339
16	138	200	263	325	388
10	250	350	450	550	650

Table 1 Memory Access Time Requirements

* A Wait state is inserted due to either RDYA or RDYD not being active when sampled by the CPU at the proper time.

** System memory address access time is maximum and includes address latch and address decoder delays.

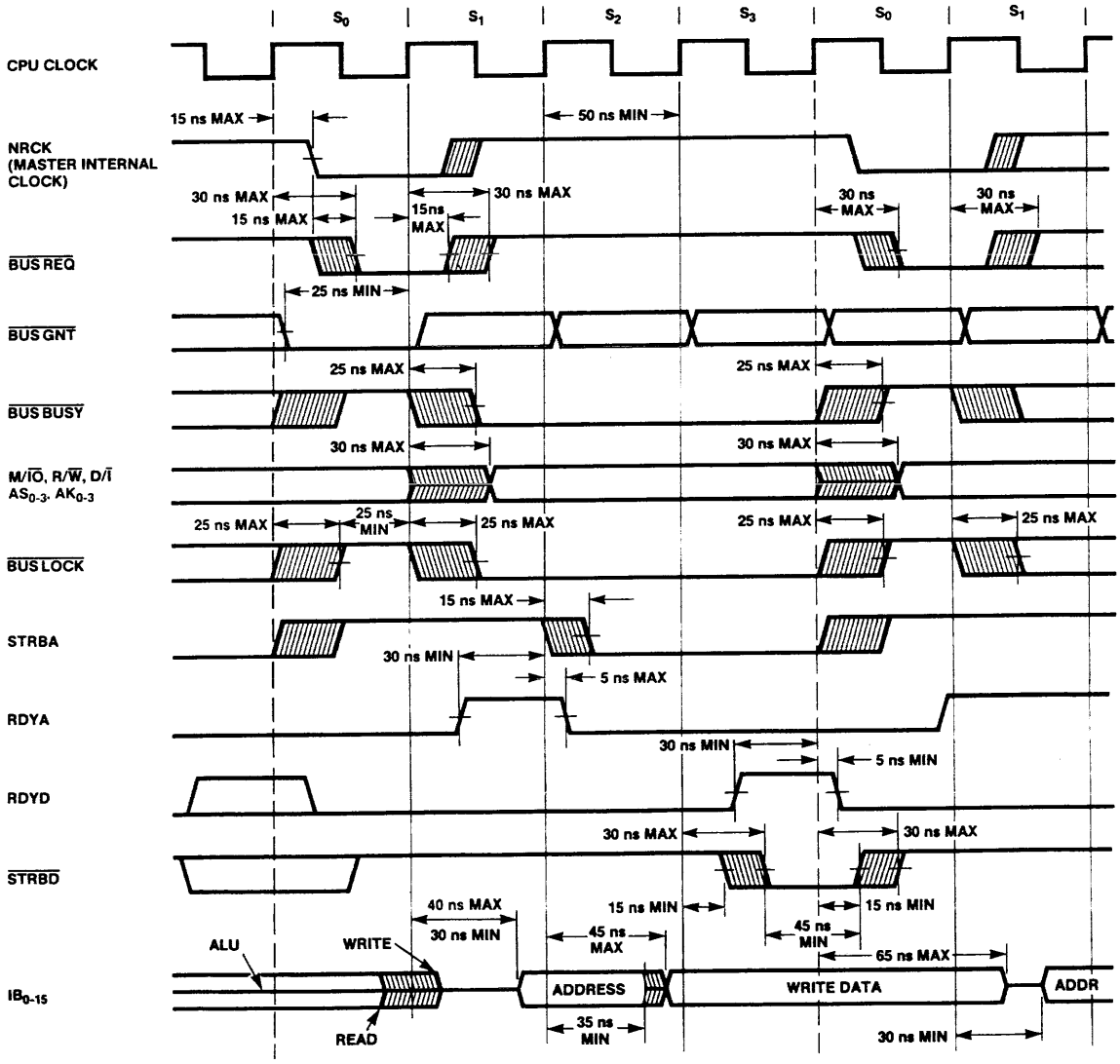


Fig. 6 Minimum Write Bus Cycle

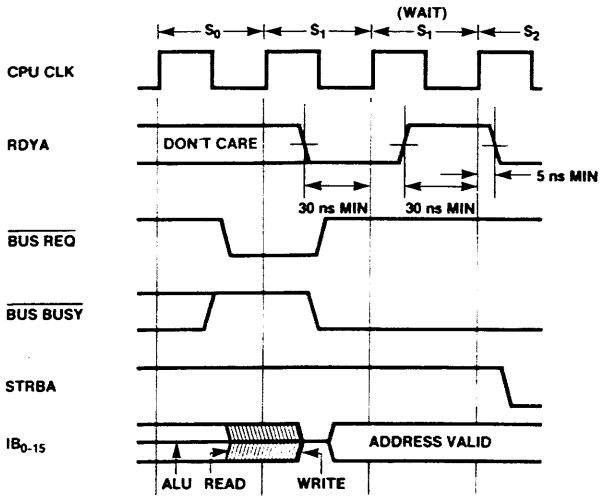


Fig. 7 RDYA Signal Timing

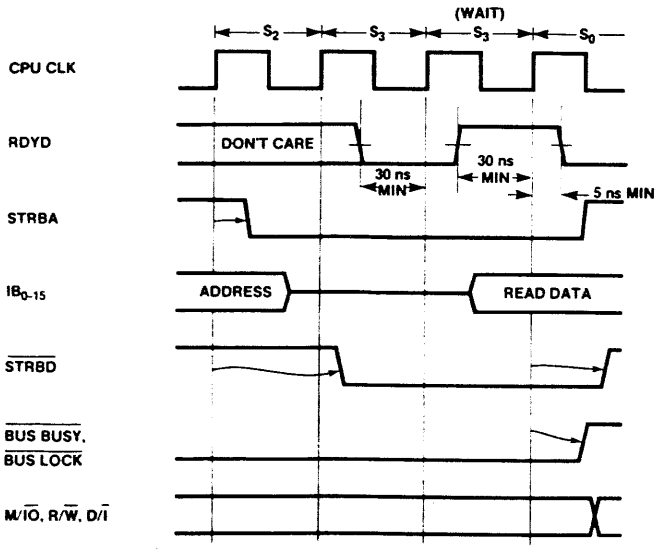


Fig. 8 RDYD Signal - Read Bus Cycle

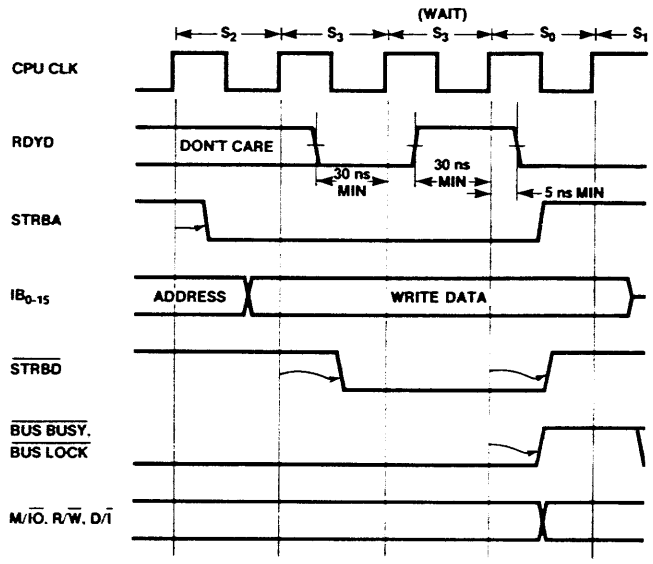
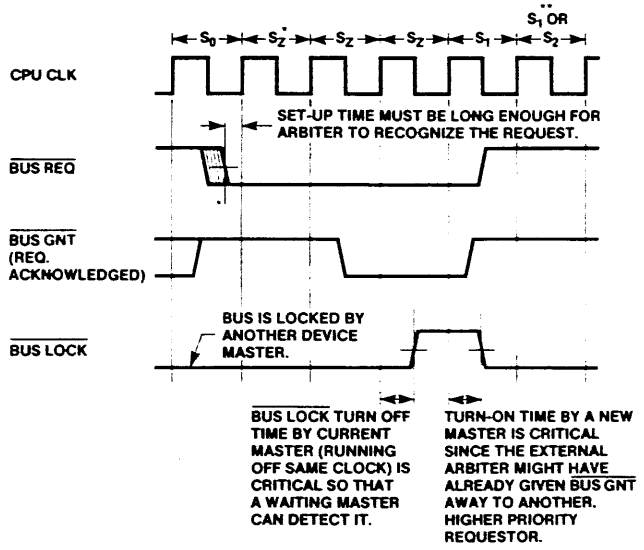


Fig. 9 RDYD Signal - Write Bus Cycle



Notes:
All device masters are F9450s or F9450-compatible devices.
Bus masters and external arbiter are running off same clock.
*S_Z is the high impedance state in which all CPU drivers are 3-stated.
**Depending on the status of RDYA input.

Fig. 10 Signal Requirements for Acquiring Bus Mastership

Data Types
The F9450 processes the following data types.

Bytes (8 bits)
Words (16 bits)
ıble words (32 bits)
...ıgle precision floating point (32 bits)
Extended precision floating point (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, (*Figure 11*).

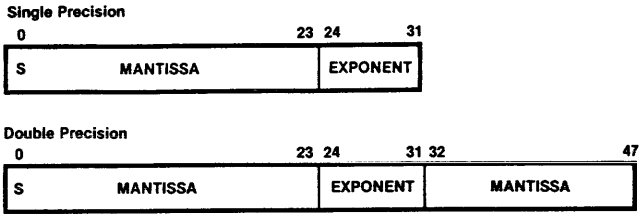
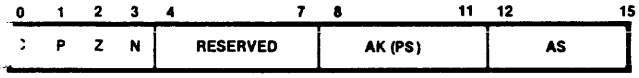


Fig. 11 F9450 Floating Point Formats

Register Set
There are 16 user-accessible registers, *Figure 12*, plus the Pending Interrupt Register, Mask Register, Fault Register, Status Register and two timers.

Status Register
The Status Word (SW) is 16 bits wide, defined as follows.



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

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 will cause a Major Error, set bit 10 in the Fault Register, and cause an instruction abort (See Fault Register, page 15).

(2) Defines the Access Key that is used in systems with an MMU to match with an access lock.

AS: Address state defines a page register group in the Memory Management Unit (MMU). For implementations that do not include 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 non-zero value.

System Configuration Register (SCR)

The System Configuration Register is five bits wide and is defined as follows (bits 0 through 4):

...MU Present: "1" if MMU is connected in the system

BPU Present: "1" if BPU is connected in the system

Console Present: "1" if console is connected in the system

Co-Processor: "1" if co-processor is connected in the system

Interrupt Mode: Selects interrupt mode for PWRDN INT and USR_0 INT - USR_4 INT. "1" level sensitive, "0" - edge (LOW-to-HIGH) sensitive.

Timer A and Timer B

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

When Timer A and Timer B reach their terminal counts, they set the corresponding bits in the PIR (see Interrupts). Both are halted when the CPU is in console mode.

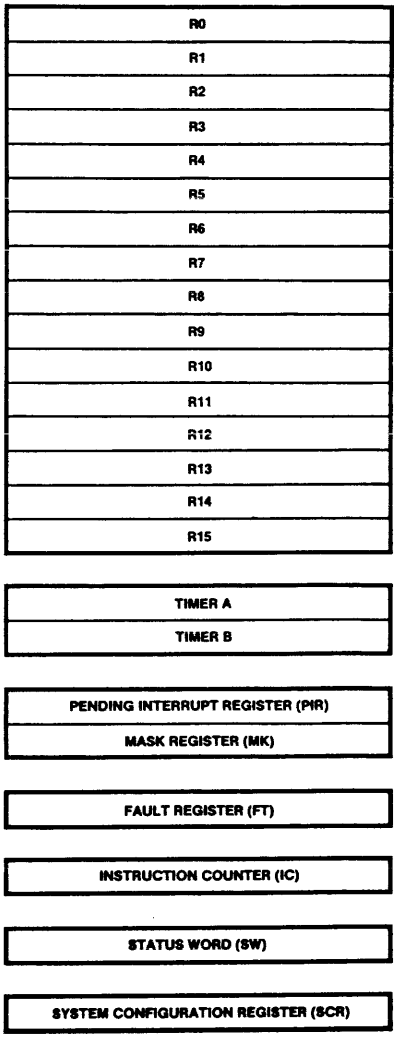
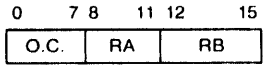
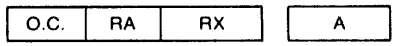
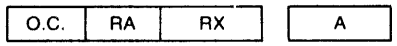
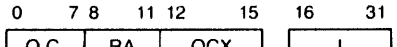
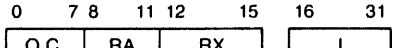
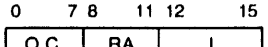
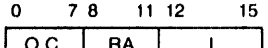
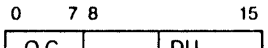
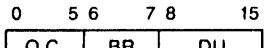
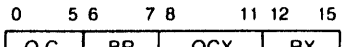


Fig. 12 F9450 Register Model

Table 2 Addressing Modes and Instruction Formats

Mode	Format	Derived Operand (DO) Single Precision	Derived Operand (DO) Floating Point and Double Precision	Derived Address (DA) Single Precision	Derived Address (DA) Floating Point and Double Precision	Notes
1. Register Direct "R"		(RB)	(RB, RB + 1)	RB	RB, RB + 1	Extended Precision Floating Point Instructions require addressing of three operands located at DA, DA + 1, and DA + 2.
2. Memory Direct "D" "DX"	 RX = 0 (Non-Indexed) RX ≠ 0 (Indexed)	|A| |A + (RX)|	|A, A + 1| |A + (RX), A + 1 + (RX)|	A A + (RX)	A, A + 1 A + (RX), A + 1 + (RX)	
3. Memory Indirect "I" "IX"	 RX = 0 (Not-Indexed) RX ≠ 0 (Indexed)	||A|| ||A + (RX)||	||A|, |A| + 1| ||A + (RX)|, |A + (RX)| + 1|	||A|| ||A + (RX)||	||A||, ||A|| + 1 ||A + (RX)||, ||A + (RX)|| + 1	
4. Immediate Long						
a. Not Indexible "IM"		I				
b. Indexible "IM" "IMX"	 RX = 0 (Non-Indexed) RX ≠ 0 (Indexed)	I I + (RX)				
5. Immediate Short						
a. Positive "ISP"		+ (I + 1)				
b. Negative "ISN"		− (I + 1)				
6. IC Relative "ICR"				D + (IC − 1)		128 ≤ DU ≤ 127
7. Base Relative						Base registers, BR = R12, R13, R14, and R15
a. Not Indexible "B"	 BR = BR + 12	|DU + (BR)|	|DU + (BR), DU + 1 + (BR)|	DU + (BR)	DU + (BR), DU + 1 + (BR)	0 ≤ DU ≤ 255
b. Indexible "B" "BX"	 RX = 0 (Non-Indexed) RX ≠ 0 (Indexed)	|(BR)| |(BR) + (RX)|	|(BR), (RR) + 1| |(BR) + (RX), (BR) + 1 + (RX)|	(BR) (BR) + (RX)	(BR), (BR) + 1 (BR) + (RX), (BR) + 1 + (RX)	

Instruction Set

The following is the list of instructions for the F9450 with the applicable addressing modes. For a complete description, refer to MIL-STD-1750A ISA.

Mnemonic	Addressing Mode	Function
Integer Arithmetic/Logic		
A	R, B, BX, ISP, D, DX, IM	Single precision Add
DA	R, D, DX	Double precision Add
INCM	D, DX	Increment Memory by positive integer
ABS	R	Single precision Absolute value
DABS	R	Double precision Absolute value
S	R, B, BX, ISP, D, DX, IM	Single precision Subtract
DS	R, D, DX	Double precision Subtract
DECM	D, DX	Decrement Memory by positive integer
NEG	R	Single precision Negate
DNEG	R	Double precision Negate
MS	R, ISP, ISN, D, DX, IM	Single precision Multiply – 16-bit product
M	R, B, BX, D, DX, IM	Single precision Multiply – 32-bit product
DM	R, D, DX	Double precision Multiply
DV	R, ISP, ISN, D, DX, IM	Single precision Divide – 16-bit dividend
D	R, B, BX, D, DX, IM	Single precision Divide – 32-bit dividend
DD	R, D, DX	Double precision Divide
C	R, B, BX, ISP, ISN, D, DX, IM	Single precision Compare
CBL	D, DX	Compare between limits
DC	R, D, DX	Double precision Compare
OR	R, B, BX, D, DX, IM	Inclusive OR
AND	R, B, BX, D, DX, IM	AND
XOR	R, D, DX, IM	Exclusive OR
NAND	R, D, DX, IM	NAND
Floating Point		
FA	R, B, BX, D, DX	Floating point Add
EFA	R, D, DX	Extended precision Floating point Add
FABS	R	Floating point Absolute Value
FS	R, B, BX, D, DX	Floating point Subtract
EFS	R, D, DX	Extended precision Floating point Subtract
FNEG	R	Floating point Negate
FM	R, B, BX, D, DX	Floating point Multiply
EFM	R, D, DX	Extended precision Floating point Multiply
FD	R, B, BX, D, DX	Floating point Divide
EFD	R, D, DX	Extended precision Floating point Divide
FC	R, B, BX, D, DX	Floating point Compare
EFC	R, D, DX	Extended precision Floating point Compare
FIX	R	Convert Floating point to 16-bit integer
FLT	R	Convert 16-bit integer to Floating point
EFIX	R	Convert extended precision Floating point to 32-bit integer
EFLT	R	Convert 32-bit integer to extended precision Floating point
Bit Operations		
SB	R, D, DX, I, IX	Set Bit
RB	R, D, DX, I, IX	Reset Bit
TB	R, D, DX, I, IX	Test Bit
TSB	D, DX	Test and Set Bit
SVBR	R	Set Variable Bit in Register
RVBR	R	Reset Variable Bit in Register
TVBR	R	Test Variable Bit in Register

Shift		
SLL	R	Shift Left Logical
SRL	R	Shift Right Logical
SRA	R	Shift Right Arithmetic
SLC	R	Shift Left Cyclic
DSLL	R	Double Shift Left Logical
DSRL	R	Double Shift Right Logical
DSRA	R	Double Shift Right Arithmetic
DSLC	R	Double Shift Left Cyclic
SLR	R	Shift Logical, count in Register
SAR	R	Shift Arithmetic, count in Register
SCR	R	Shift Cyclic, count in Register
DSLR	R	Double Shift Logical, count in Register
DSAR	R	Double Shift Arithmetic, count in Register
DSCR	R	Double Shift cyclic, count in Register
Load/Store/Exchange		
L	R, B, BX, ISP, ISN, D, DX, IM, IMX, I, IX	Single precision Load
DL	R, B, BX, D, DX, I, IX	Double precision Load
EFL	D, DX	Extended precision Floating point Load
LUB	D, DX, I, IX	Load from Upper Byte
LLB	D, DX, I, IX	Load from Lower Byte
S	B, BX, D, DX, I, IX	Single precision Store
STC	D DX, I, IX	Store a non-negative Constant
DST	B, BX, D, DX, I, IX	Double precision Store
SRM	D, DX	Store Register through Mask
EFST	D, DX	Extended precision Floating point Store
STUB	D, DX, I, IX	Store into Upper Byte
STLB	D, DX, I, IX	Store into Lower Byte
XBR	S*	Exchange Bytes in Register
XWR	R	Exchange Words in Registers
Multiple Load/Store		
PSH M	S*	Push Multiple registers onto the stack
POP M	S*	Pop Multiple registers off the stack
LM	D, DX	Load Multiple registers
STM	D, DX	Store Multiple registers
MOV	S*	Move multiple words, memory to memory
Program Control		
JC	D, DX, I, IX	Jump on Condition
JS	D, DX	Jump to Subroutine
SOJ	D, DX	Subtract One and Jump
BR	ICR	Branch unconditionally
BEZ	ICR	Branch if Equal to (Zero)
BLT	ICR	Branch if Less Than (zero)
BLE	ICR	Branch if Less than or Equal to (zero)
BGT	ICR	Branch if Greater Than (zero)
BNZ	ICR	Branch if Not equal to (Zero)
BGE	ICR	Branch if Greater than or Equal to (zero)
BEX	ICR	Branch to Executive
LST**	D, DX, I, IX	Load Status
SJS	D, DX	Stack IC and Jump to Subroutine
URS	S*	Unstack IC and Return from Subroutine
NOP	S*	No Operation
BPT	S*	Breakpoint
BIF	S*	Built In Function (escape code)

* S Special Format

** Privileged Instruction

Input/Output Instructions **

Programmed Input/Output		
XIO	IM, IMX	Execute Input/Output
VIO	D, DX	Vectored Input/Output
Timer Control		
TAS		Timer A Start
TAH		Timer A Halt
OTA		Output Timer A
ITA		Input Timer A
TBS		Timer B Start
TBH		Timer B Halt
OTB		Output Timer B
ITB		Input Timer B
Interrupt/DMA/Fault Control		
SMK		Set Interrupt Mask
CLIR		Clear Interrupt Request
ENBL		Enable Interrupts
DSBL		Disable Interrupts
RPI		Reset Pending Interrupt
SPI		Set Pending Interrupt Register
RMK		Read Interrupt Mask
RPIR		Read Pending Interrupt Register
RCFR		Read and Clear Fault Register
DMAE		DMA Enable
DMAD		DMA Disable
MMU Control		
WIPR		Write Instruction Page Register
WOPR		Write Operand Page Register
RIPR		Read Instruction Page Register
ROPR		Read Operand Page Register
BPU Control		
LMP		Load Memory Protect RAM
RMP		Read Memory Protect RAM
MPEN		Memory Protect Enable
MISC		
WSW		Write Status Word
RSW		Read Status Word
RNS		Reset Normal Power Up discrete
GO		Reset Trigger GO indicator

**Privileged Instruction

Performance

Table 3 lists execution times for a core instruction set in the different addressing modes. All times are at 20 MHz with no wait states (See *Table 1* for memory system requirements). Execution times will degrade linearly with the clock if no wait states are inserted, i.e., Add Register will be 0.4 μs at 10 MHz.

Basic Instructions

Single Precision	Register	Direct	Direct Indx.	Indirect	Imm.	Imm. Short	Base Rel.	Base Rel. Indx.
Load/Store	0.2	0.6	0.6	0.8	0.4	0.35	0.55	0.55
Add/Sub	0.2	0.6	0.6	0.8			0.55	0.55
Multiply	1.85	2.25	2.25		2.05	2.0	2.2	2.2
Divide	4.7	5.1	5.1			4.9	5.05	5.05
Compare	0.35	0.75	0.75		0.55	0.5	0.7	0.7
Set/Reset Bit	0.35	0.95						
Double Precision								
Load	0.65	1.25	1.25	1.6			1.2	1.2
Store	0.65	0.95	0.95	1.3			0.9	0.9
Add/Sub	0.8	1.4	1.4					
Multiply	5.75	6.35	6.35					
Divide	12.0	12.6	12.6					
Compare	0.6	1.2	1.2					
Floating Point								
Add/Sub (Typical)	4.5	5.1	5.1				5.2	5.2
Multiply	5.6	6.2	6.2				6.3	6.3
Divide	9.8	10.4	10.4				10.5	10.5
Compare	3.15	3.75	3.75				3.85	3.85
Extended Floating Point								
Load		1.3	1.3					
Store		1.45	1.45					
Add/Sub (Typical)	5.75	6.55	6.55					
Multiply (Typical)	12.4	13.2	13.2					
Divide (Typical)	21.15	21.95	21.95					
Compare (Typical)	3.85	4.65	4.65					

Branch	Taken	Not Taken
	0.75	0.2

Table 3 Execution Times

Fault Register (FT)
The Fault Register is 16 bits wide. Bit functions are described next.

Bit 0 CPU memory protect error
Bit 1 Non-CPU memory protect error
Bit 2 Memory parity error
Bit 3 Spare
Bit 4 Spare
Bit 5 Illegal I/O address
Bit 6 Spare
Bit 7 System Fault 0
Bit 8 Illegal memory address
Bit 9 Illegal instruction
Bit 10 Privileged instruction
Bit 11 Address State error
Bit 12 Spare
Bit 13 BITE (Built-in Test) or System Fault 1
Bit 14 Spare
Bit 15 System Fault 1

System Fault 0 and System Fault 1 are asynchronous, edge-sensitive inputs to the F9450 processor.

Any bit set in the Fault Register can cause a Level 1 interrupt. Major or unrecoverable errors as defined below will cause the CPU to abort the current instruction.

Major Errors
- Privileged instruction
- CPU access mode
- CPU write protect
- CPU data illegal address
- Illegal I/O address

Unrecoverable Errors
- Illegal instruction
- Instruction protect fault
- Instruction parity
- Instruction illegal address
- Address state fault

Interrupts
There are 16 levels of interrupt prioritized on chip as indicated in *Table 4*. Nine are external, of which two are level sensitive (IOL_1 INT, IOL_2 INT). The other seven external interrupts are either level or edge-sensitive, according to the interrupt mode bit in the configuration register. 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 the table.

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

Interrupts are acknowledged by resetting the acknowledged interrupt bit in the PIR and executing an I/O cycle during which the acknowledged interrupt number is sent to I/O device 1000. Level interrupt requests should be removed within two machine cycles after the I/O acknowledged cycle. 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 is invoked by the BEX instruction that 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.

	Priority (PIR/MK bit number)	Interrupt Linkage Pointer Address (Hex)	Interrupt Service Pointer Address(Hex)
Power Down[1]	0[3]	20	21
Machine Error[2]	1	22	23
User 0	2	24	25
F.P. Overflow	3	26	27
Fixed Point Overflow	4	28	29
Executive Call[1]	5	2A	2B
F.P. 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. Cannot be disabled
3. Interrupt level 0 has the highest priority

Table 4 Interrupt Priorities

Self Test and Initialization
The Self Test is part of the Initialization sequence (*Figure 13*) that is invoked by the assertion of $\overline{\text{RESET}}$.

Self Test Functions

Reads/writes all registers in register file

Causes address state fault and verifies that the fault bit is set and a machine error interrupt is pending

Verifies ALU functions

Checks Booth hardware by performing a multiply

Checks divide hardware by performing a divide

ALU shifter is checked right/left by Booth/divide

Verifies ROM constants can be accessed

Verifies IC/MAR can be accessed and incremented

If completed successfully, the Normal Power Up discrete output (NML PWRUP) will be set, else bit number 13 in the Fault Register is set.

Reads the configuration from I/O device 8410 into the SCR to determine the presence or absence of the MMU, BPU, Console, Co-processor - it also initializes the interrupt mode and then initializes the system as follows.

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
Timers A&B	All Zeros & Counting
Trigger GO Reset ($\overline{\text{TRIGO RST}}$)	Pulsed

MMU (see Page 18)

Page Registers	
AL Field	All Zeros
W Field	All Zeros
E Field	All Zeros
PPA Field	Logical to Physical

BPU (see Page 22)

CPU Write Protect Registers	All Zeros
DMA Write Protect Registers	All Zeros
Global Memory Protect	Enabled

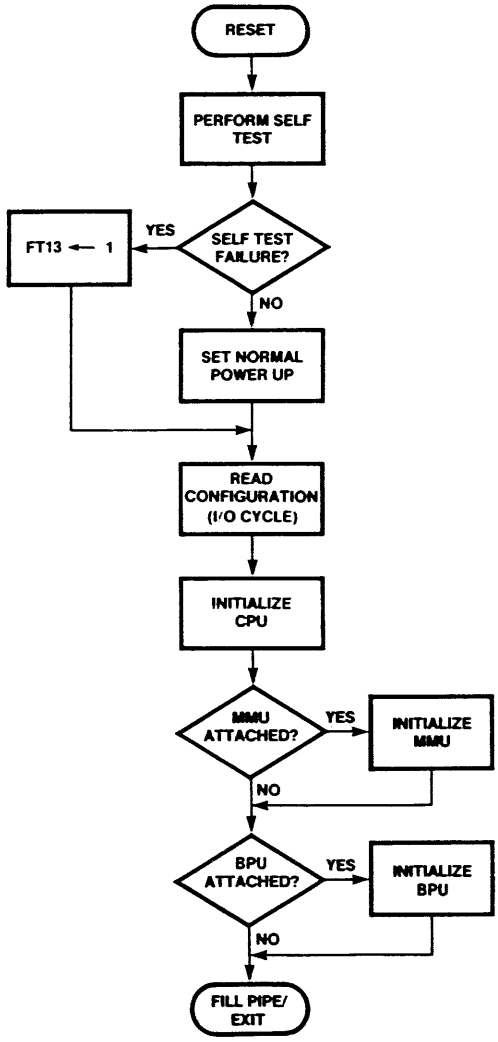


Fig. 13 Self Test/Initialization

Console Operations

Consoles are treated by the F9450 microprocessor as three Input/Output addresses: console command (8400), console read address (switches) (8401), or console write address (displays) (0400).

Console operation is initiated when the $\overline{\text{CONREQ}}$ input goes LOW. The CPU completes the current instruction, executes an Input/Output cycle to read the console command from the information bus, (I/O address 8400), and then executes the console command. Typically, it involves I/O addresses 8401 to enter console data and 0400 to write data to the console. Console operations and corresponding codes are listed in *Table 5.*

0 — 7	8	9	10 — 15
CONSOLE CODE	*	*	**REGISTER ADDRESS**
0, 1, 2, 3, 4, 5, 6, 7			10, 11, 12, 13, 14, 15
74 : DISABLE			000000 : R0
60 : EXAM REG			010000 : R1
61 : DEP REG			000001 : R2
62 : EXAM FT			010001 : R3
66 : EXAM MEM			000010 : R4
67 : DEP MEM			010010 : R5
6A : EXAM NEXT			000011 : R6
6B : DEP NEXT			010011 : R7
75 : CONTINUE			000100 : R8
6C : EXAM XIO			010100 : R9
6D : DEP XIO			000101 : R10
6E : EXAM XIO NEXT			010101 : R11
6F : DEP XIO NEXT			000110 : R12
			010110 : R13
			000111 : R14
			010111 : R15
			001000 : A2
			011000 : A1
			001001 : Q2
			011001 : Q1
			001010 : DO0
			011010 : DO1
			001011 : PIR
			011011 : MK
			001100 : FT
			001101 : SW
			001110 : TA
			011110 : TB
			1000XX : MDR
			100101 : IC
			1010XX : MAR
			1011XX : IR

* Bits 8 and 9 are always "Don't Care"

Table 5 Console Command Format

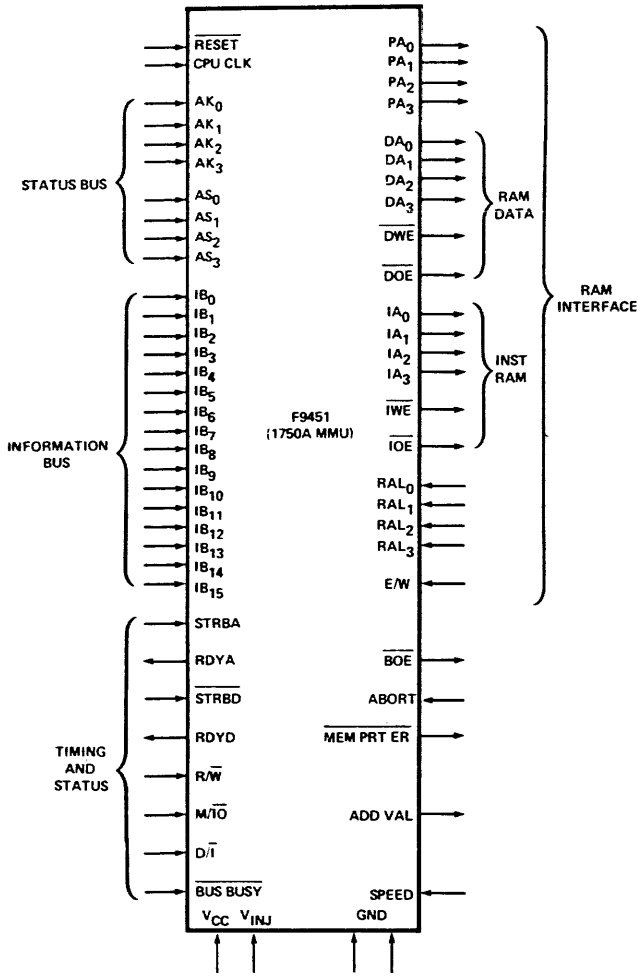


Fig. 14 F9451 MMU Pin Functions

F9451 Memory Management Unit (MMU)
The Memory Management Unit provides the following functions

- Logical to physical address translation
- Address space of 1M words (20 bits of address)
- Two translation maps
  - Instruction map (256 x 16)
  - Data map (256 x 16)
- Protection in the logical space of 4K word pages for
  - Access Key to Access Lock match
  - Write protect
  - Execute protect

Address Translation and Protection Mechanism
The memory is mapped in 4K word pages. The mapping mechanism incorporates 512 page registers that are organized in 16 pairs of groups, one for instruction memory space and one for data memory space. Each group contains 16 page registers accommodating a total of 256 registers for data space and 256 registers for instruction space (*Figure 15*).

One of 16 pairs of groups is selected by the four AS bits. The selection of the group within the pair is done by the D/I (Data/Instruction) bit. The page register within the group is selected by the four most significant bits of the address. The eight bits of the PPA field (Physical page Address) are concatenated with the 12 least significant bits of the address to become the 20 bits of physical address.

The MMU also checks for protection violations by comparing the Access Key (AK) to the access lock (AL field in the page register), and by examining the Write protect bit (W in page register) if data is accessed, or the Execute protect bit (E in page register) if instruction is accessed. If a violation occurs, it activates the memory protect error output. Page Registers are read or written by means of XIO instructions.

Principle of Operation
The MMU (*Figure 16*) comprises the F9451 MMU chip plus four F93479 (instruction and data maps) and two bidirectional buffers (54F245).

The F9451 generates the addresses and controls for the map RAMs that provide the eight most significant bits of the physical address. To avoid the access time to the map RAMs every CPU memory access, the F9451 employs on-chip cache mechanisms.

The last AS bits and four most significant address bits (for Data and Instruction) are latched, for comparison during the next memory access. Only in the case of a cache miss is a Map RAM access necessary. The last Access Lock (AL) and Execute (E)/Write (W) protect bits from the Map RAM are latched as well to be used for the following cycles if cache hits occur.

F9451 Signal Descriptions

Status Bus
AK_0 through AK_3, Pins 14, 15, 18 and 19 Access Key Bus - Four inputs from CPU or DMA used to convey the memory access key. This is compared with the lock information which is retrieved from the MMU MAP and used to produce $\overline{\text{MEM PRT ER}}$ if a violation occurs.

AS_0 through AS_3, Pins 20 through 23 - Address State Bus - Four inputs from CPU or DMA used to specify the state (0-15) of the user who is referencing memory. Used by the F9451 to select the page group of the MMU MAP appropriate for this particular user.

Information Bus
IB_0 through IB_{15}, Pins 24 through 31 and 33 through 40 - 16-bit Bus - 16 inputs from main system Address/Data bus used to transfer Address or I/O code from CPU or DMA to the F9451 during the Address phase of a Bus cycle.

Timing and Status
STRBA, Pin 50 - Address Strobe - Active HIGH input from CPU or DMA. HIGH: Address phase of Memory or I/O cycle, HIGH-to-LOW transition latches address.

RDYA, Pin 42 - Address Ready - Open-collector output to the CPU or DMA used by the F9451 to extend the Address phase of a Memory or I/O cycle. The F9451 pulls RDYA LOW prior to the beginning of a Bus cycle and then releases it after a variable number of S1 states.

Other devices can insert additional S1 states by OR-tying their own open-collector RDYA outputs with this pin.

$\overline{\text{STRBD}}$, Pin 51 - Data Strobe - Active LOW input from CPU or DMA. LOW: Data phase of Bus cycle, LOW-to-HIGH transition used by F9451 to terminate $\overline{\text{IOE}}$ or $\overline{\text{IWE}}$.

RDYD, Pin 43 - Data Ready - 3-state output to the CPU used to extend the Data phase of I/O cycles when the CPU is writing to the MAP RAMs. Inserts one additional S3 state. Not enabled during Memory cycles. HIGH: Ready, LOW: another S3.

R/$\overline{\text{W}}$, Pin 44 - Read/Write - Input from CPU or DMA to specify direction of data transfer during Memory or I/O cycles. HIGH: Data moves from MMU MAP RAM or Memory to the CPU or DMA, LOW: Data moves from CPU or DMA.

M/$\overline{\text{IO}}$, Pin 46 - Memory-I/O - Input from CPU to specify type of cycle. HIGH: Memory Cycle, LOW: I/O cycle.

D/$\overline{\text{I}}$, Pin 45 - Data/Instruction - Input from CPU to specify type of Memory cycle. HIGH: Operand reference, LOW: CPU Instruction fetch.

$\overline{\text{BUS BUSY}}$, Pin 47 - Bus Busy - Active LOW input from CPU or DMA LOW-to-HIGH transition marks the end of the current Bus cycle. HIGH: bus idle or being rearbitrated.

MMU MAP RAM Interface

PA_0 through PA_3, Pins 63 through 60 - Page Address Bus - Four outputs from the F9451 to the four MSB address inputs to all four of the F93479 MMU MAP RAMs. Directly related to AS_{0-3}.

DA_0 through DA_3, Pins 55 through 52 - Data Address Bus - Four outputs used to address four LSB address inputs of the two MAP RAMs used during operand references.

$\overline{\text{IOE}}$ and $\overline{\text{DOE}}$, Pins 9 and 6 - Instruction Output Enable and Data Output Enable - Two outputs used to control the Output Enables of the two groups of F93479 MMU MAP RAMs for instruction fetches and operand references, respectively. Active also during I/O cycles which read the contents of the RAMs to the CPU. LOW: Enable - HIGH: Disable.

$\overline{\text{IWE}}$ and $\overline{\text{DWE}}$, Pins 10 and 7 - Instruction Write Enable and Data Write Enable - Two outputs used to control the writing of the MMU MAP RAMs by the CPU connected to the Instruction and Data (Operand) RAMs respectively. LOW: Write, HIGH: Read.

IA_0 through IA_3, Pins 59 through 56 - Instruction Address Bus - Four outputs from the F9451 used as the four LSB address inputs to the two MMU MAP RAMs used for instruction references.

AL_0 through RAL_3, Pins 4 through 1 - RAM Access Lock - Four inputs to the F9451 from the MMU MAP RAMs to signify the lock loaded in the MAP for the current AS and the area of memory the user is trying to access. Compared with AK to generate $\overline{\text{MEM PRT ER}}$.

E/W, Pin 5 - Execute/Write Protect - Input to the F9451 from the MMU MAP RAMs to signify whether the CPU is allowed to fetch an instruction from the addressed area of memory or if the CPU or DMA is allowed to alter (by writing) memory. HIGH: protect, LOW: allow access.

General Purpose Signals

$\overline{\text{RESET}}$, Pin 17 - Reset - Active LOW input from system used to initialize F9451 at power up. Proper initialization requires that $\overline{\text{RESET}}$ be asserted for at least four clock cycles. LOW: Reset, HIGH: Normal operation.

CPU CLK, Pin 49 - CPU Clock - System clock input to the F9451. The LOW-to-HIGH transition causes a change of state in the F9451. Frequency range is 0 to 20 MHz.

$\overline{\text{BOE}}$, Pin 11 - Buffer Output Enable - Output from the F9451 used to control a 16-bit wide bidirectional bus transceiver between the Information bus and Extended Address (including the extra three reserved bits), RAM Access Lock and Execute/Write Protect. This Buffer is enabled only during I/O operations to the MMU MAP RAMS. The direction input of this buffer can be driven directly from the R/$\overline{\text{W}}$ signal. LOW: Enable, HIGH: Disable (normal).

ABORT, Pin 13 - Aborted Cycle - Active HIGH input from the CPU (MAJ ER, UNRCV ER) used by the F9451 to prevent alteration of any of its registers during Aborted cycles initiated by the CPU. This also forces RDYA HIGH after only S1 state. HIGH: Abort cycle, LOW: Normal operation.

$\overline{\text{MEM PRT ER}}$, Pin 41 - Memory Protect - Open-collector output to the CPU, memory, and other circuits, which signals that the user's Address Key did not match the Access Lock. It also can indicate that the user was attempting to execute an instruction from a protected area of memory, or attempting to write into a protected area of memory. Signal is stable prior to the HIGH-to-LOW transition of STRBA. LOW: violation, HIGH: memory reference valid.

ADD VAL, Pin 12 - Address Valid - Active HIGH output to the Block Protect Unit and other circuits. LOW-to-HIGH transition signals that the Extended Address from the MMU MAP RAMs is stable. HIGH-to-LOW transition follows STRBA.

SPEED, Pin 8 - Speed Select - Active HIGH input used to inform the F9451 of the speed range of the accompanying CPU. HIGH: 20 MHz CLK, LOW: 10 MHz CLK.

The number of wait states introduced by the MMU is thrown in *Table 6.*

GND, Pins 16 and 48 - Ground

V_{CC}, Pin 64 - Nominal +5 V, 70 mA

V_{INJ}, Pin 32 - Injector Supply - Nominal (at approximately 1.3 V), 80 mA.

		Clock Rate			
		20 MHz		10 MHz	
Type of Cycle		S_1	S_3	S_1	S_3
I/O	MMU	3	1	1	0
	$\overline{\text{MMU}}$	2	0	1	0
Memory	Page Boundary Crossed (Miss)	4	0	2	0
	No Page Boundary Crossed (Hit)	2	0	1	0

Table 6 MMU Wait States

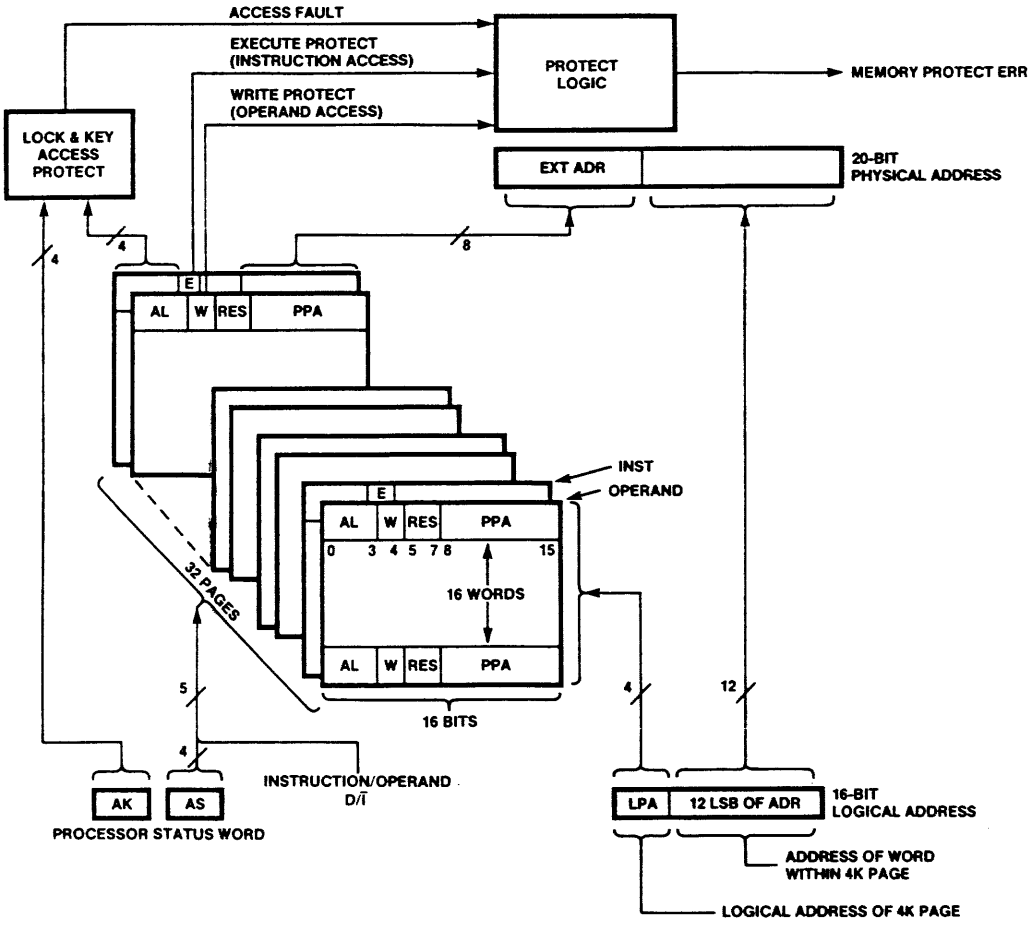


Fig. 15 MMU Mapping Structure

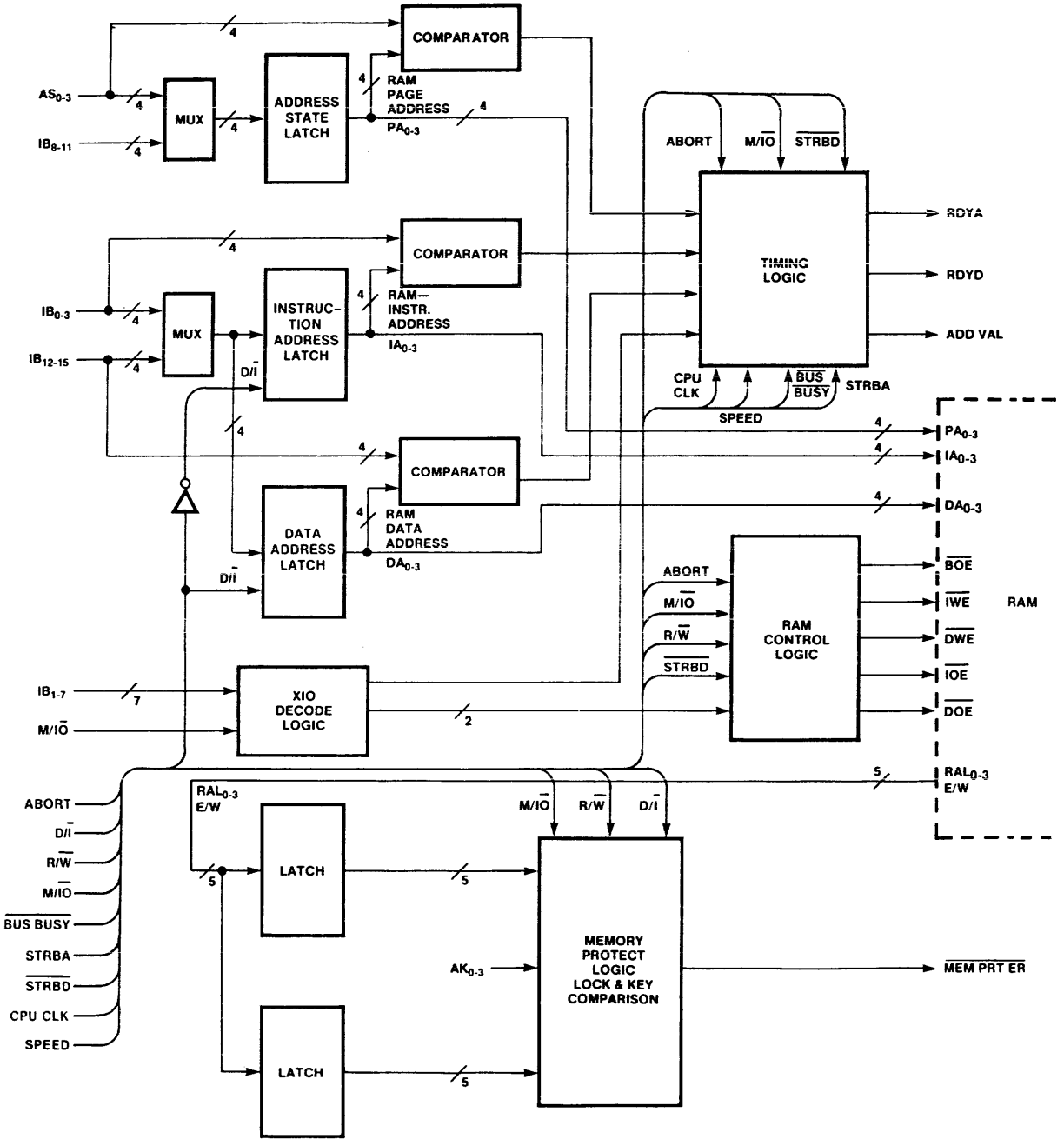


Fig. 16 F9451 MMU Block Diagram

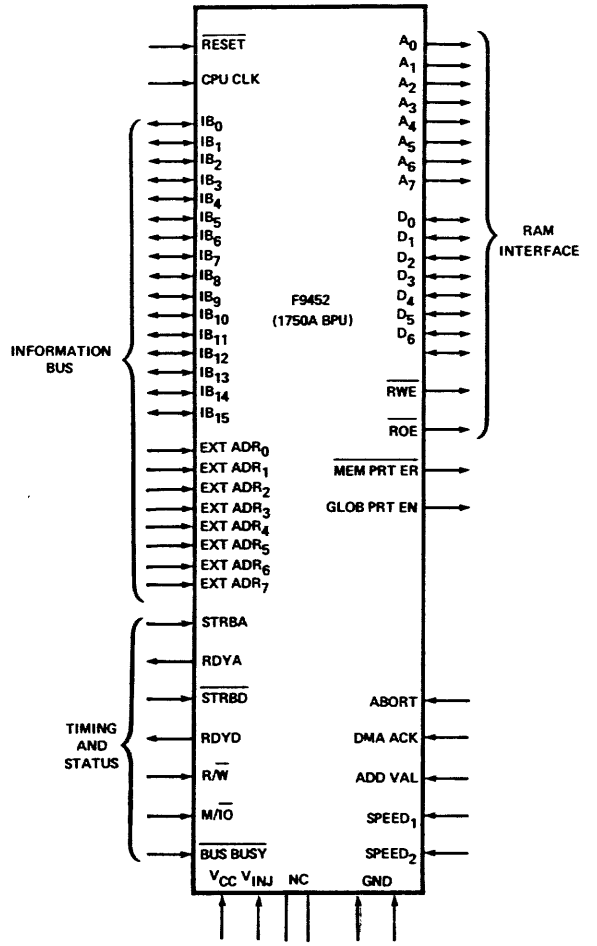


Fig. 17 F9452 BPU Pin Functions

F9452 Block Protect Unit

The Block Protect Unit (BPU) provides the following functions.

- Write protection of physical CPU memory space
- Write protection of physical DMA memory space
- Protection is done on 1K pages
- Global memory write protection from initialization until being enabled.

Principle of Operation

The BPU provides write protection via a look-up table (RAM), in which each bit represents 1K word page in physical memory. The look-up table is organized in 16-bit words and contains 128 x 16 words (64 for CPU and 64 for DMA) when an MMU is incorporated, or 8 x 16 words (four for CPU and four for DMA) when an MMU is not incorporated.

The F9452 BPU chip (*Figure 17*) communicates with the look-up table RAM in bytes. It employs a cache mechanism in which the last memory address and the last 16-bit write protect word read from the RAM (representing 16 contiguous 1K pages) are latched on chip. In the subsequent memory access, the look-up table RAM is only accessed if there is a cache miss. The look-up table words are read or written by means of XIO instructions.

F9452 Signal Descriptions

Information Bus

IB_0 through IB_{15}, Pins 24 through 31 and 33 through 40 - 16-bit Bus - Active HIGH bidirectional bus for communication between CPU and F9452.

EXT ADD_0 through EXT ADD_7, Pins 14, 15 and 18 through 23 - Extended Address - Active HIGH inputs, eight bits of extended address, coming from the MMU, *Figure 21*.

Timing and Status

STRBA, Pin 50 - Address Strobe - Active HIGH input used to latch the memory or I/O address from the Information bus on the HIGH-to-LOW transition.

RDYA, Pin 42 - Address Ready - Active HIGH open-collector output used by the BPU to maintain the address on the information bus by pulling it LOW.

$\overline{\text{STRBD}}$, Pin 51 - Data Strobe - Active LOW input used as a strobe for the data phase of the information bus.

RDYD, Pin 43 - Data Ready - Active HIGH output used to synchronize the Block Protect RAM with the CPU during I/O cycles with the look-up table RAM.

R/W, Pin 44 - Read/Write - Input indicating the direction of data transfer. HIGH: data transfers to CPU, LOW: data transfers to Block Protect Unit.

M/$\overline{\text{IO}}$, Pin 46 - Memory-Input/Output - Input indicating whether a Memory Bus cycle or an I/O Bus cycle is taking place. HIGH: Memory Bus cycle, LOW: I/O Bus cycle.

$\overline{\text{BUS BUSY}}$, Pin 47 - Bus Busy - Active LOW input indicating that a Bus cycle is in progress.

RAM Interface

A_0 through A_7, Pins 63 through 56 - RAM Address - Active HIGH outputs, eight bits of address data to the look-up table F93479 RAM.

D_0 through D_7, Pins 8 through 1 - RAM Data - Active HIGH bidirectional signals, eight bits of data to the look-up table F93479 RAM.

$\overline{RWE}$, Pin 54 - RAM Write Enable - Active LOW output that provides a Write Enable signal to the look-up table F93479 RAM when LOW.

ROE, Pin 53 - RAM Output Enable - Active LOW output that controls the output enable of the look-up table F93479 RAM - LOW: Enable; HIGH: Disable.

General Purpose Signals

$\overline{RESET}$, Pin 17 - Reset - Active LOW input used to initialize internal circuitry and to disable memory protection mechanism. Sets GLOB PRT EN.

CPU CLK, Pin 49 - CPU Clock - Active HIGH single-phase clock input. LOW-to-HIGH transition causes the F9452 to change states. Frequency range is 0-20 MHz.

$\overline{MEM\ PRT\ ER}$, Pin 41 - Memory Protect Error - Active LOW open-collector output indicating that a memory write was attempted to a protected location.

GLOB PRT EN, Pin 55 - Global Memory Protect Enable - Active HIGH output indicating that memory is to be globally write protected. It is set by $\overline{RESET}$, and reset by executing an MPEN instruction.

ABORT, Pin 11 - Abort - Active HIGH input indicating that the cycle is aborted (MAJ ER, UNRCV ER in F9450). Write to the look-up table RAM is inhibited.

DMA ACK, Pin 13 - Direct Memory Access Acknowledge - Active HIGH input from the aribiter used to select the appropriate bank in the memory look-up table.

ADD VAL, Pin 12 - Address Valid - Active HIGH input from the Memory Management Unit used to enter the extended address.

$SPEED_1$ and $SPEED_2$, Pins 9 and 10 - Clock Speed Data - Active HIGH inputs used to control the clock frequency as indicated in the table below, and the number of wait states introduced by the BPU as a function of the clock ranges is shown in *Table 7*.

CPU CLOCK Freq. Range MHz	$SPEED_1$	$SPEED_2$
15–20	1	1
12–15	1	0
8–12	0	1
less than 8	0	0

GND, Pins 16 and 48 - Ground

V_{CC}, Pin 64 - Nominal +5 V, 70 mA.

V_{INJ}, Pin 32 - Injector Supply - Nominal (at approximately 1.3 V), 140 mA.

Type of Cycle

Clock		15-20 MHz		12-15 MHz		8-12 MHz		< 8 MHz	
Cycle		S_1	S_3	S_1	S_3	S_1	S_3	S_i	S_3
I/O	BPU	2	6	2	4	1	4	1	2
	$\overline{BPU}$	2	0	2	0	1	0	1	0
Memory Write	Miss	3	0	2	0	1	0	1	0
	Hit	2	0	1	0	0	0	0	0
Memory Read		1	0	1	0	1	0	0	0

Table 7 BPU Wait States

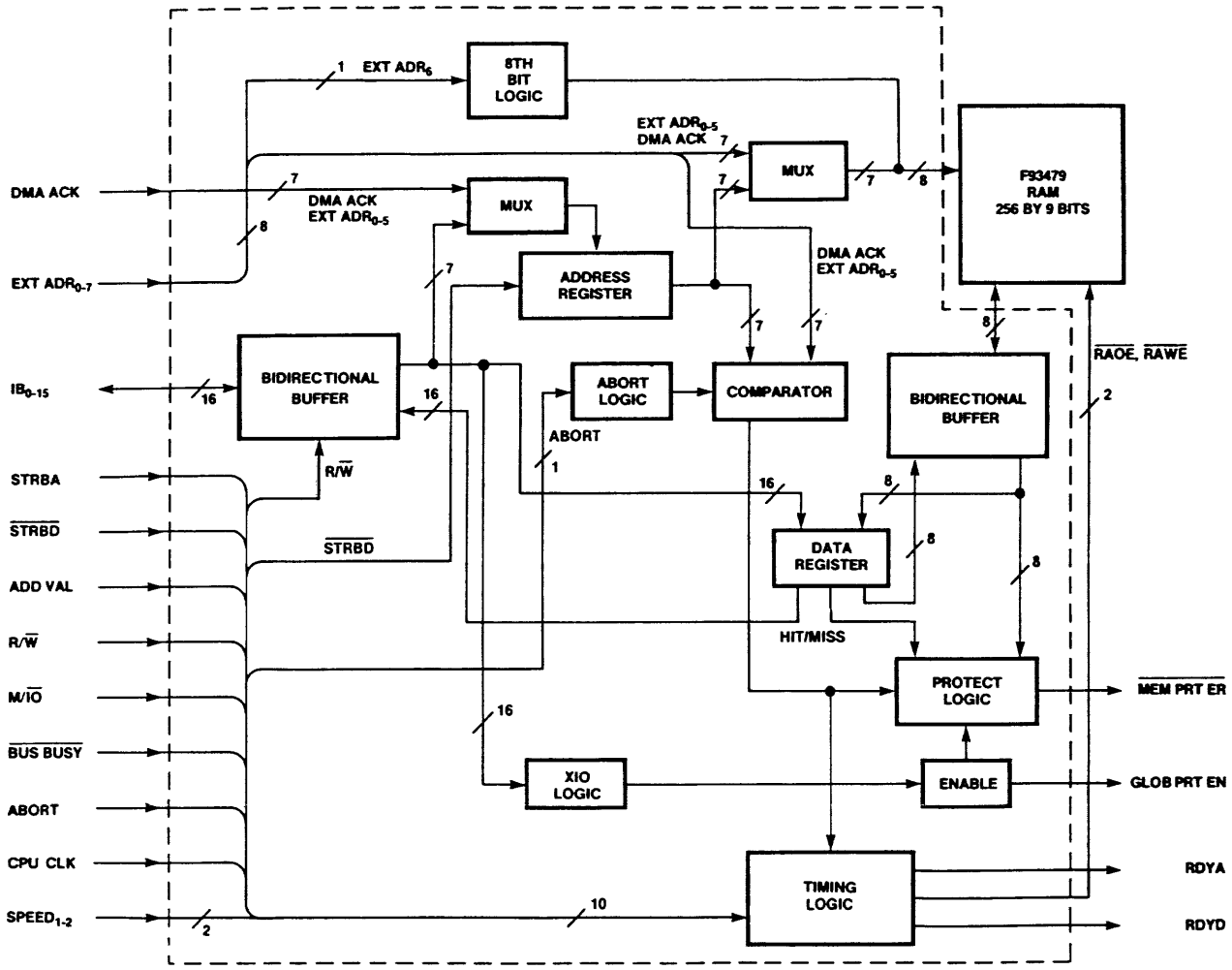


Fig. 18 F9452 BPU Block Diagram

Applications

Basic System Configurations

The basic system configurations using the F9450, F9451 and F9452 are shown in *Figures 19-21*. A CPU/MMU configuration (*Figure 19*) provides a 1M word memory space with access lock/access key, execute and write protection. The MMU indicates the validity of the eight extended address bits (concatenated with the 12 least significant bits of the address to constitute the 20-bit physical address) by activating the ADD VAL pin. It will also hold the RDYA input to the CPU LOW as long as the extended address bits are not valid.

Any protection violation is reported to the CPU (to start the fault handling process) by pulling the $\overline{\text{MEM PRT ER}}$ pin LOW. The two 54F245s provide the data path between the Map RAM (four 93479s) and the CPU.

In a CPU/BPU configuration (*Figure 20*), the BPU provides write protection for 1K word pages. The protection is done for the CPU memory space and DMA memory space as determined by the DMA ACK input. Global memory write protection is provided by the GLOB PRT EN output as long as the BPU is not enabled. The BPU pulls the DYA input HIGH when it is ready and reports write protect error by pulling the $\overline{\text{MEM PRT ER}}$ output LOW.

A CPU/MMU/BPU configuration will provide a 1M words addressing space with 4K page protection in logical space for lock/key, write and execute protections, as well as 1K page write protection in the physical space. In this case, the RDYA and $\overline{\text{MEM PRT ER}}$ inputs to the CPU are wired-OR between the MMU and the BPU.

Multiprocessor/Bus Arbitration Functions

A simple bus arbitration scheme uses three bus control signals, $\overline{\text{BUS REQ}}$, $\overline{\text{BUS GNT}}$, and $\overline{\text{BUS LOCK}}$,

$\overline{\text{BUS REQ}}$ - 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

Timing requirements for the arbitration sequence is illustrated in *Figure 10*. Three configurations are discussed (*Figures 22-24*).

1. CPU and DMA with DMA device having highest priority
2. Dual CPU with one CPU assigned highest priority
3. Multiple bus masters with arbitrated priority

CPU/DMA Arbitration - In a single CPU/DMA configuration (*Figure 22*), 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 it.

Dual CPU Arbitration - In a dual CPU configuration (*Figure 23*), one is arbitrarily assigned the highest priority, and the low priority CPU has access to the bus only if the high priority CPU doesn't request it.

Multiple Bus Masters with Arbitrated Priority - *Figure 24* depicts a system with eight bus masters, seven CPUs operating from the same CPU clock, and one asynchronous DMA device. 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 when using multiple DMA devices is as follows. Multiple DMA devices would present their requests to the bus arbiter, causing a single $\overline{\text{DMA RQ}}$ to appear at the arbiter. A $\overline{\text{DMA ACK}}$ signal from the arbiter is expected, and once issued by the arbiter, the highest priority requesting DMA device (assuming more than one) can use the bus.

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

Built-In Function Implementation with an External Co-processor

The Built-In Function (BIF) is an escape code in the F9450 instruction set that allows user defined instructions. Use of an external co-processor for BIF implementation is shown in *Figure 25*.

The co-processor is receiving the command word (defining the instruction) and control word via XIO instructions. The operands to the co-processor are passed from the F9450 by parameter address passing, and the co-processor becomes a bus contender arbitrated by the bus arbiter (*Figure 25a*). If the system includes an MMU, an additional latch has to be added (*Figure 25b*) to provide the Address State (AS) and Access Key (AK) for the co-processor.

I/O addresses for the co-processor as well as other dedicated I/O addresses are listed in *Table 8*.

IO 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
0800, 0900, 0A00, 0B00	Output	Write derived address to coprocessor No. 1, 2, 3 or 4 respectively. (used to implement Built-in functions.)
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.

Table 8 Dedicated I/O Addresses

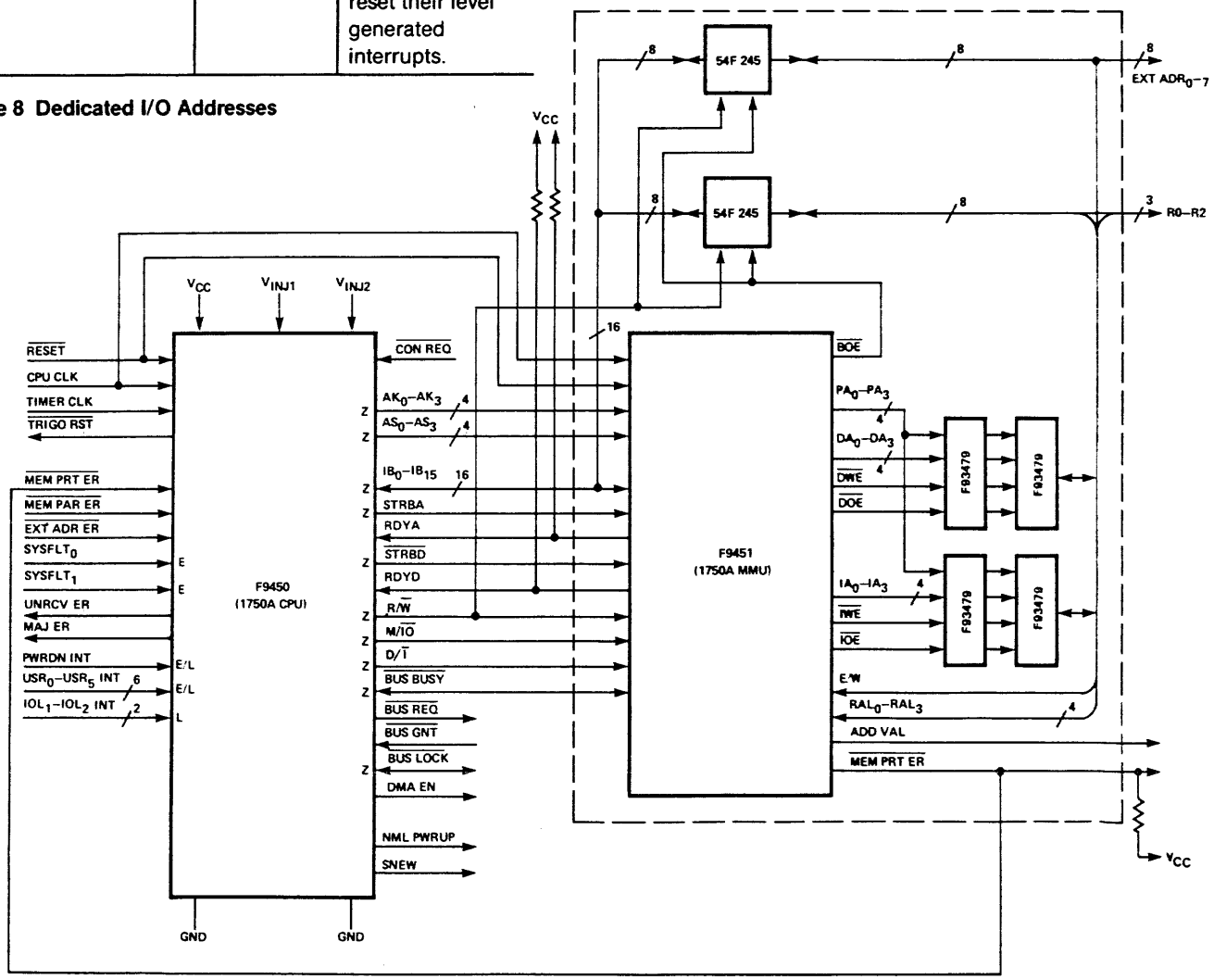


Fig. 19 CPU/MMU Configuration

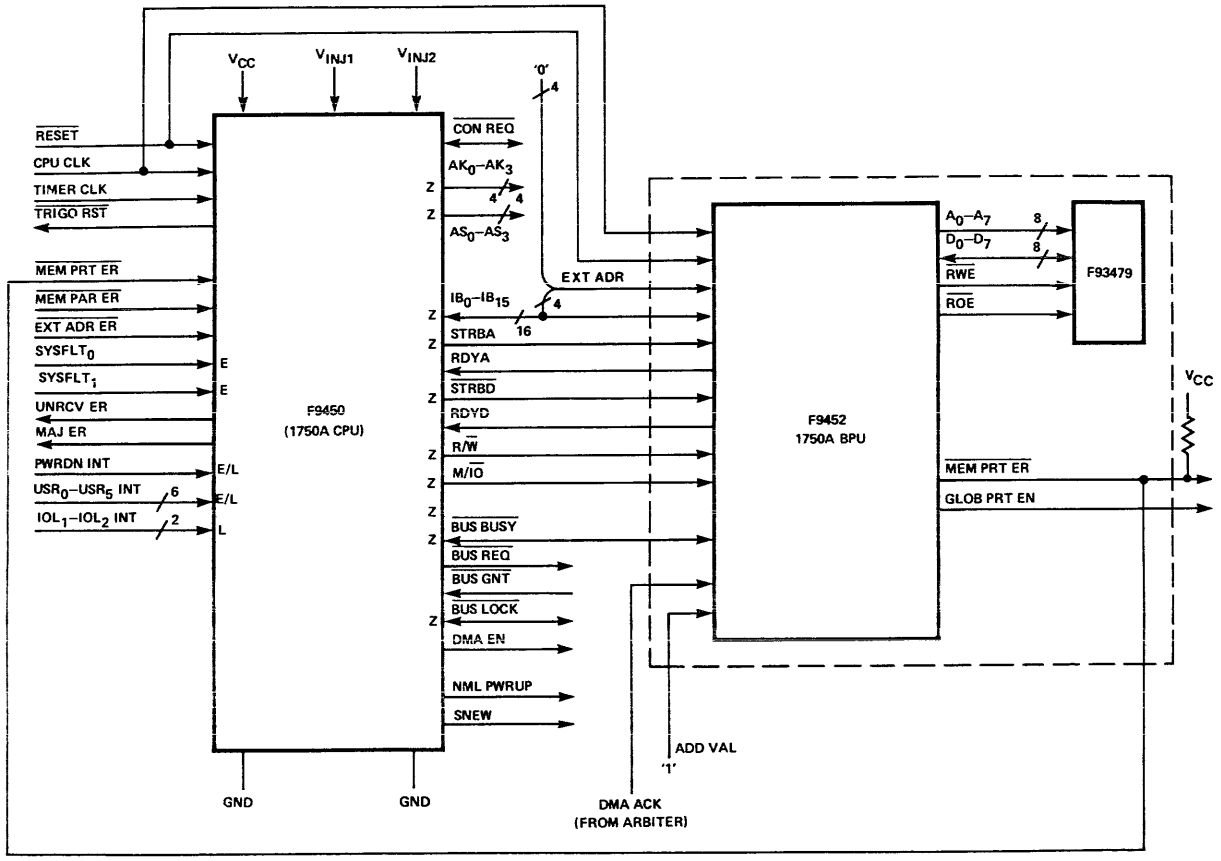


Fig. 20 CPU/BPU Configuration

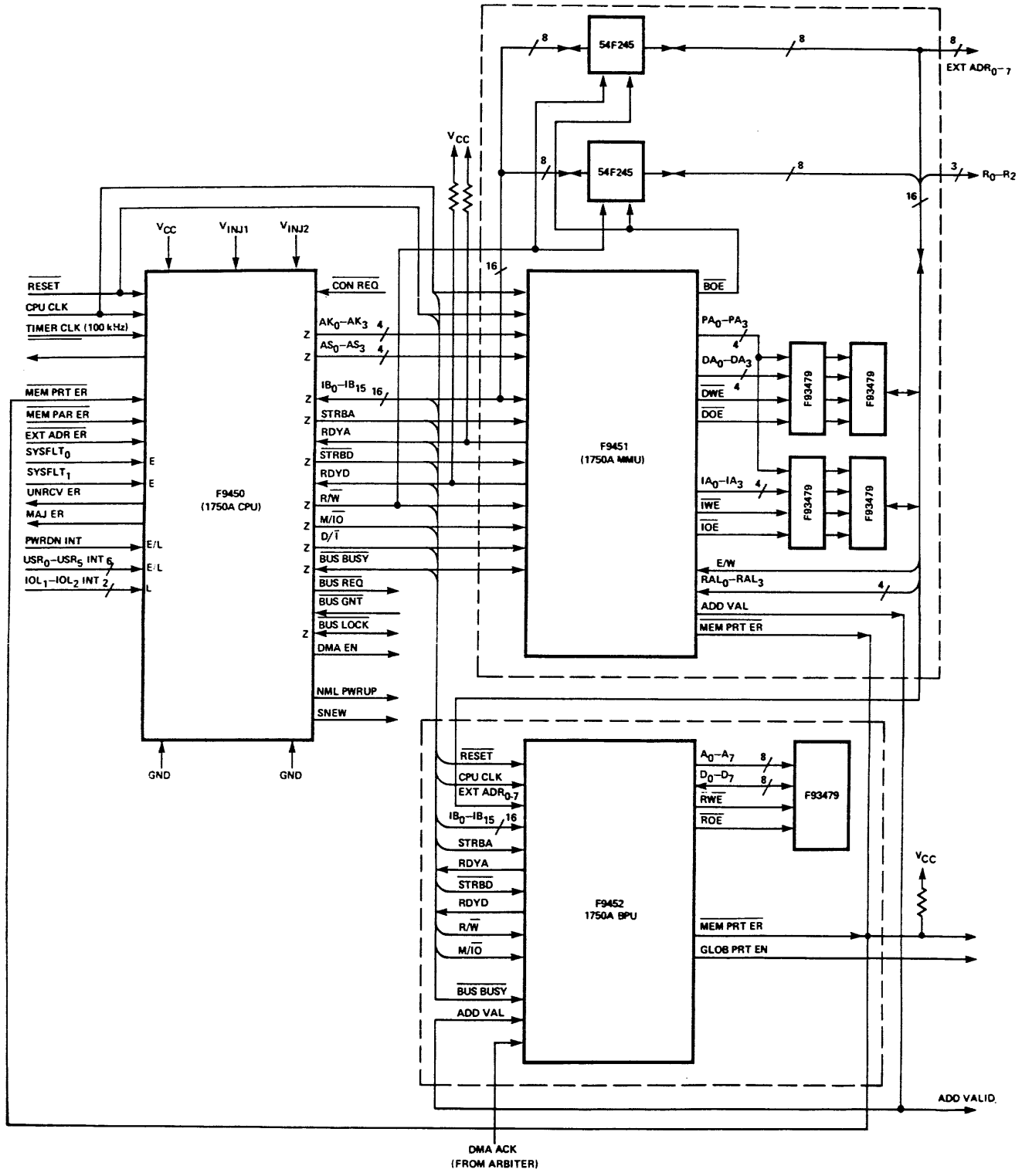


Fig. 21 CPU/MMU/BPU Configuration

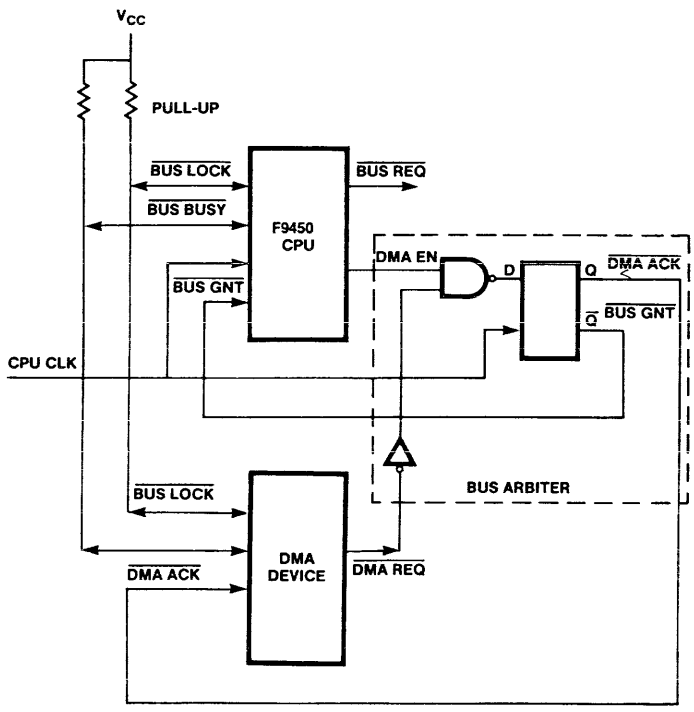


Fig. 22 Single CPU/DMA System with External Arbiter

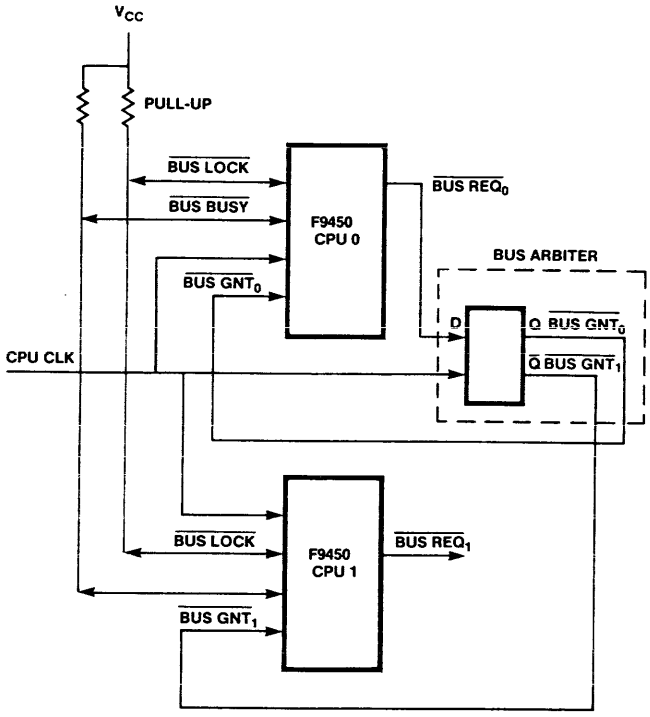


Fig. 23 Dual CPU with External Arbiter

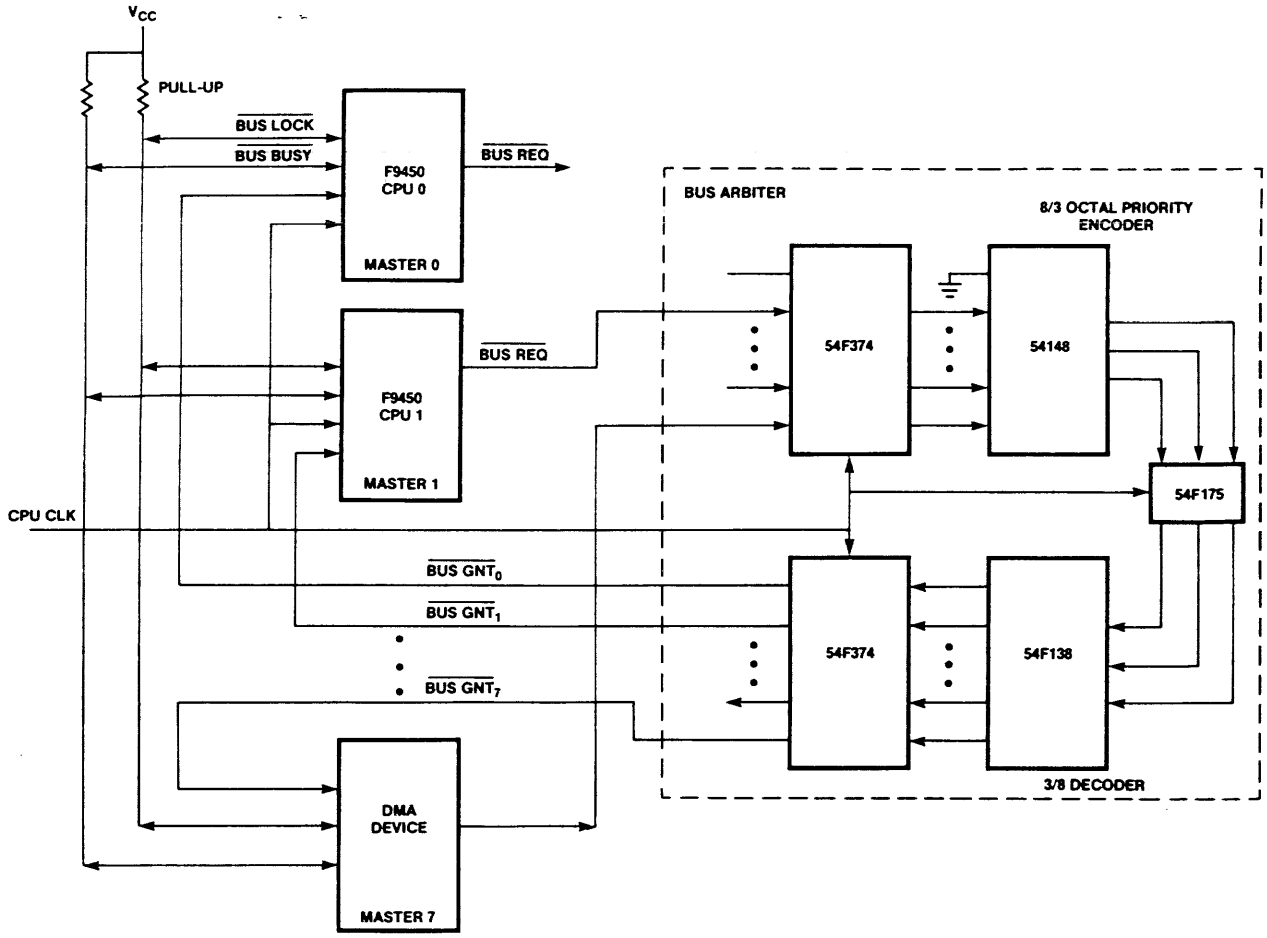


Fig. 24 External Arbitration for up to 8 Bus Masters

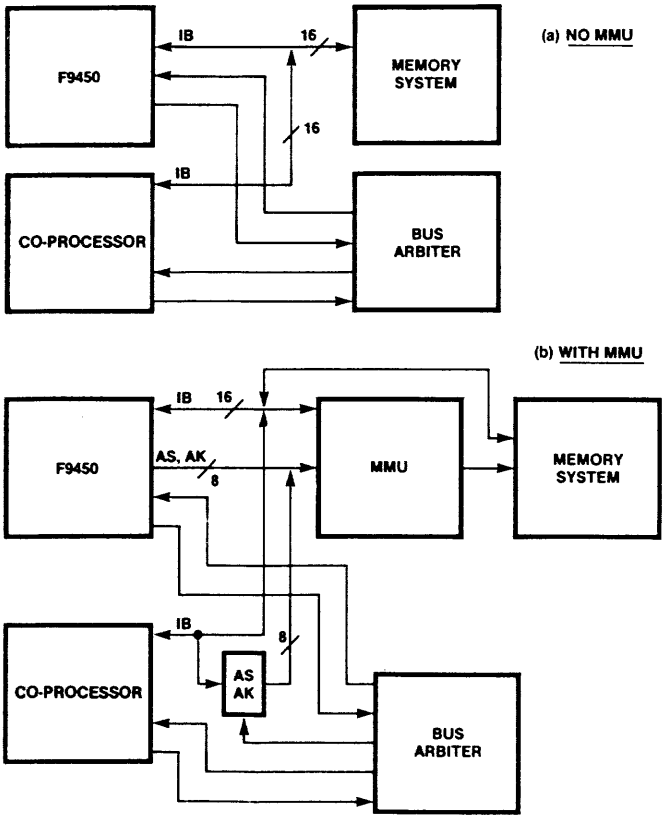
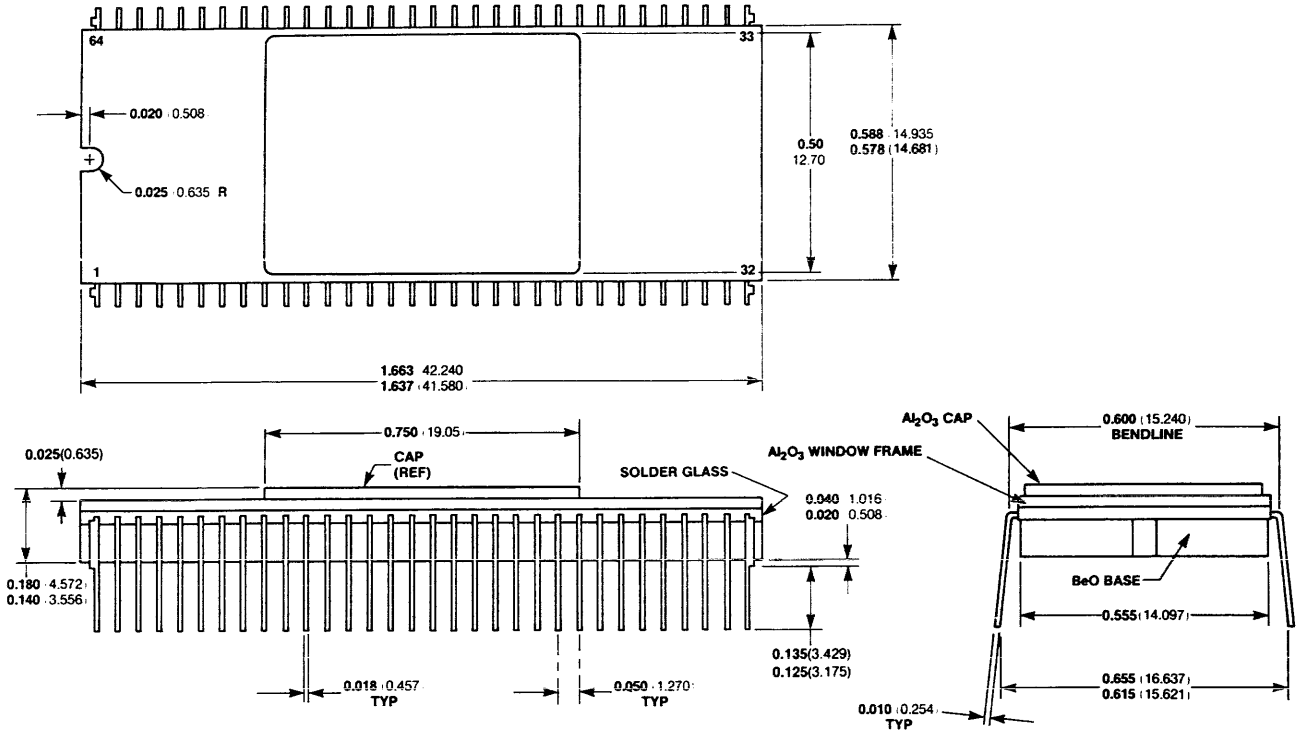
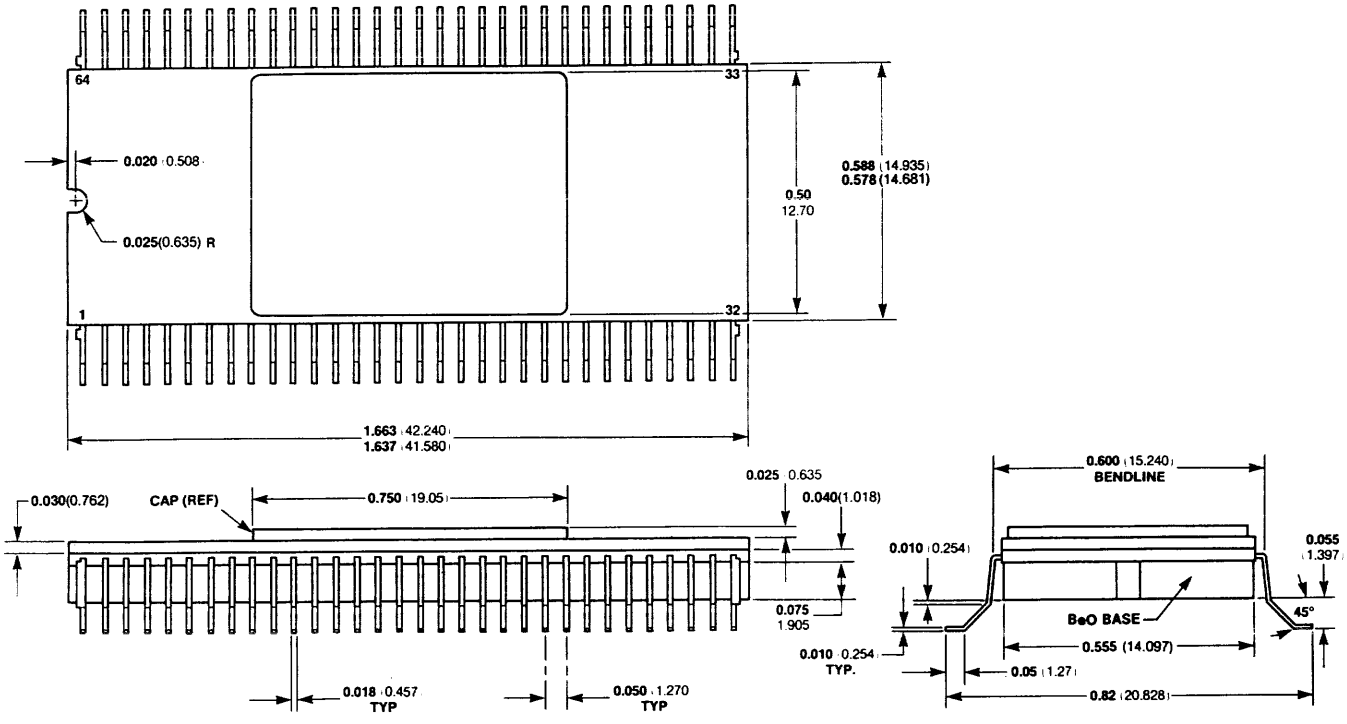


Fig. 25 F9450/ Co-Processor



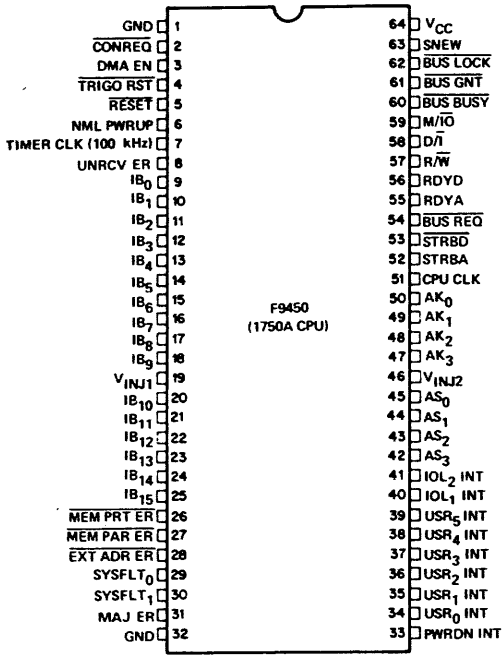
NOTES:
All dimensions are in inches **bold** and millimeters (parentheses).
Pin material is tin plated alloy 42.
Cap is ceramic.
Base is ceramic.
Package weight is 8.5 grams.

64-Pin Ceramic Dual In-Line Package

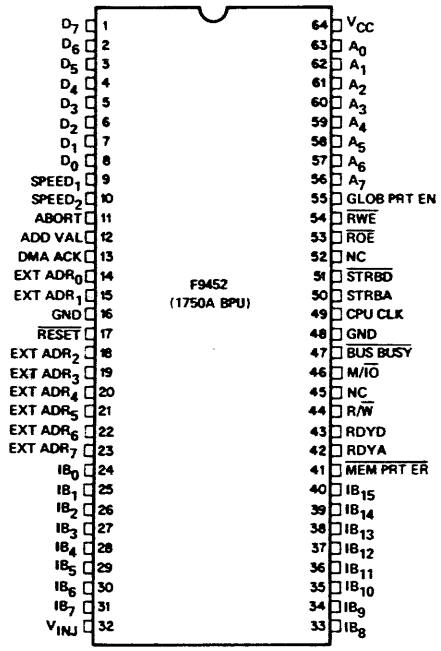


NOTES:
All dimensions are in inches **bold** and millimeters (parentheses).
Pin material is tin plated alloy 42.
Cap is ceramic.
Base is ceramic.
Package weight is 8.5 grams.

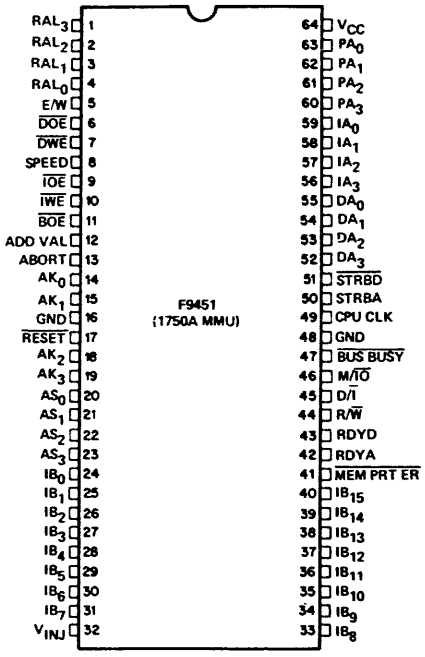
64-Pin Ceramic Gull-Wing Dual In-Line Package



F9450 Connection Diagram



F9452 Connection Diagram



F9451 Connection Diagram

Ordering Information

Device	Package	Order Code*
F9450	Ceramic DIP	F9450 XX DM
F9450	Ceramic DIP	F9450 XX DMQB
F9450	Ceramic Gull Wing DIP	F9450 XX GMQB
F9451	Ceramic DIP	F9451 XX DM
F9451	Ceramic DIP	F9451 XX DMQB
F9451	Ceramic Gull Wing DIP	F9451 XX GMQB
F9452	Ceramic DIP	F9452 XX DM
F9452	Ceramic DIP	F9452 XX DMQB
F9452	Ceramic Gull Wing DIP	F9452 XX GMQB

* XX = Speed, to be defined

Inquiries regarding the F9450 Family, including available software and development tools, should be directed to:

M. Grisham, Fairchild R&D, 4001 Miranda Avenue, Palo Alto, CA 94304, 415/493-7250

or

J. R. Byrne, Fairchild Microprocessor Products, 3420 Central Expressway, Santa Clara, CA 95051, 408/773-1000.

FAIRCHILD
A Schlumberger Company

F9470
Communication and Console Controller

Advance Product Information

Microprocessor Product

Description

The Fairchild F9470 Communication and Console Controller is an LSI MOS device that provides the Fairchild F9445 16-bit I^{3}L® microprocessor with virtual console control functions via a pair of asynchronous communication ports.

The F9470 provides a variety of useful console functions, including examine and deposit to memory and accumulators, jump to a specified location, and trace the F9445 instruction execution.

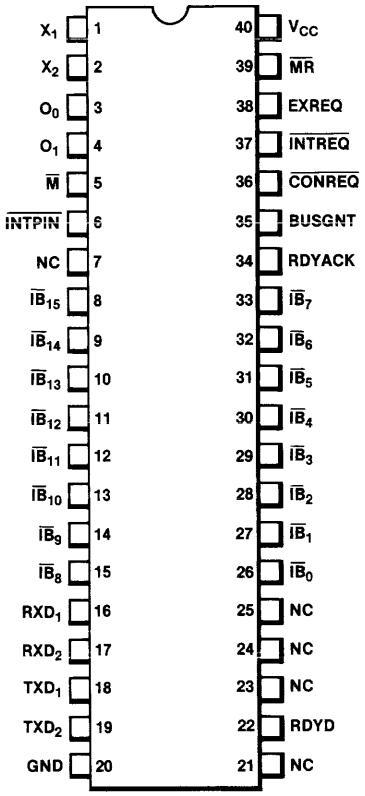
The F9470 operates in two modes: console control and I/O service. In the console mode, all communication with the F9445 is controlled by the F9470, which interprets the seven console commands, requests the appropriate information from the F9445, and then outputs it to the operator's terminal.

In the I/O service mode, the F9470 acts as a serial I/O controller, interfacing the serial I/O devices to the F9445 through device codes 10-13 and 77. The console commands are not available while in the I/O service mode: all I/O in this mode must be programmed through the F9445.

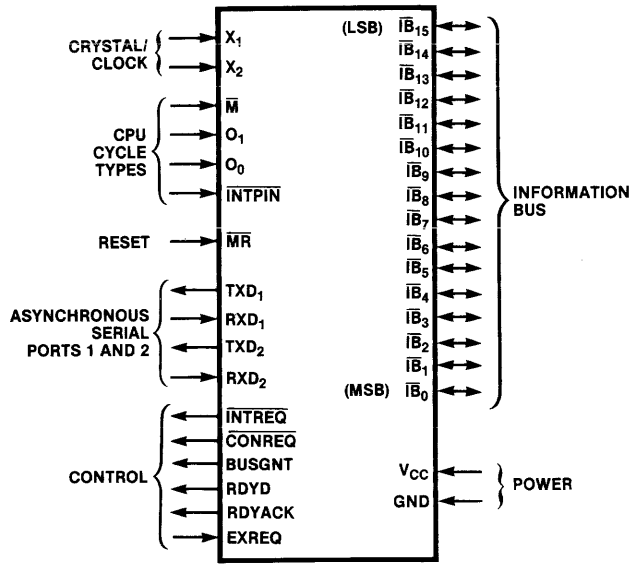
- **Accesses Microprocessor Internal Registers**
- **Two Asynchronous Serial Ports**
- **Allows VDU to Operate as a Console for an F9445 System**
- **40-Pin DIP Requiring Single +5 V Power Supply**
- **NMOS Technology.**

Figure 1 illustrates the pin configuration of the console controller.

Connection Diagram



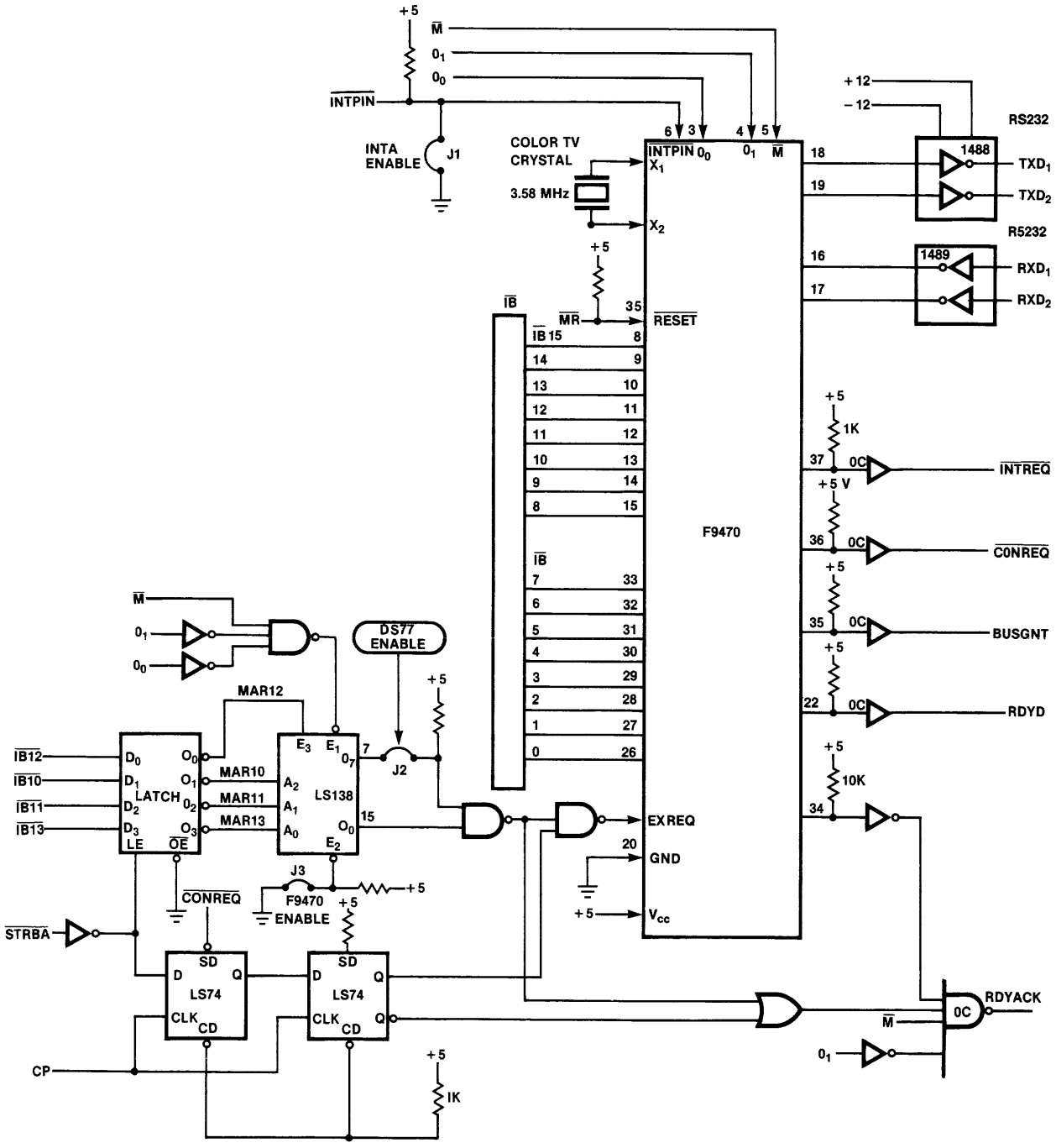
Signal Functions



® I^{3}L is a registered trademark of Fairchild Camera and Instrument Corp.

Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

Figure 1 Pin Configuration of the Console Controller



The three jumpers in figure 1 perform the following functions.

Jumper	Name	Description
J1	INTA ENABLE	Jumpered low, the F9470 is the highest proirity I/O device, or the only I/O device in the system. Removing the jumper and driving $\overline{\text{INTPIN}}$ provides a daisy-chain priority-interrupt scheme. A high level on this pin allows interrupts from the I/O terminal to be disabled when a device with a higher priority interrupts the CPU.
J22	DS77 ENABLE	With this jumper in place, the F9470 responds to device code 77 (octal) programmed I/O instructions. Removing the jumper allows faster interrupt response to a system device that interrupts the CPU at a higher frequency. Notice that MSKO, INTA, and IORST are no longer available with the jumper removed.
J3	F9470 I/O SERVICE ENABLE	Connect the jumper to ground to use the I/O service mode. The F9470 decodes and executes I/O instructions for a processor in order to control the serial output. The F9470 responds to device codes 10, 11, 12, 13, and 77 (teletype in and out, paper tape reader and punch, CPU), and stalls the CPU to get time to decode and respond to the instructions using these device codes.

Console Mode Operation

The procedure used to communicate with the F9470 in the console mode is as follows:

Entering Data and Commands

The console prompt is an asterisk(*), which indicates the F9470 will accept commands. The commands are executed by typing the desired capital letter or, where appropriate, the octal number(s) and letter. No carriage return (CR) is necessary. Once within the examine (E) mode, some keystroke is used to complete each individual entry. The strokes are (CR) and Λ. Each closes the present entry, and then respectively goes to the next or previous address location. To write a program with the E command, enter the octal values equivalent to the F9445 binary instruction codes. (See the F9445 data sheet for a description of the instruction set.)

Recovering from Keystroke Error

If an error is made by typing the wrong letter command, there is no recovery, since execution begins immediately. There are many ways to correct data errors. (Characteristics vary among terminals; the following text refers to what occurs when using the Zenith terminal.)

1. Force an overflow. The F9470 buffer accepts six octal numbers, so only the last six entries will be used. For example, 77777777 is interpreted to be 177777. Only the least significant bit of the most significant number is read.

2. CTRL-H. This backs up over each mis-typed number and echoes each entry, so the number of character deletions is apparent.

3. Pressing the backspace key is identical to pressing CTRL-H.

4. Pressing any key other than 0 through 7, or any accepted command letter, returns to the console mode.

5. After executing the examine command, xxxE, pressing any key other than 0 through 7 &(CR), or Λ returns to the console mode.

Return to Console Mode from I/O Service

BREAK (or CTRL-BREAK) enters the console mode from the I/O service mode. The F9445 will continue execution of any currently running code. (BREAK and CTRL-BREAK are keystrokes. Some terminals require that the key labeled CONTROL be held while the BREAK key is struck.)

The seven console commands processed by the F9470 are given in table 1.

Table 1 F9470 Console Mode Commands

Command	Description
A	Displays seven fields, the present octal contents of the PC, AC0, AC1, AC2, AC3, SP, and FP registers. PC contains the next instruction to be executed.
n,vC	Changes any of six registers to the new octal value, v. To select which register, choose the octal code associated with that register. Octal code (n): 0 1 2 3 4 5 Register: AC0 AC1 AC2 AC3 SP FP If v is omitted, the chosen register is set to 0. Comma is the only proper delimiter to use.
xE	Examines the contents of the memory cell at address x. The address and its present octal contents are displayed on the terminal. New octal contents may then be typed. A carriage return enters the new value and closes the cell. The next address (x + 1) and its contents are then displayed, and its new value can be typed. Entry continues in sequential locations until the ESC key is pressed and terminates the examining session. Pressing multiple carriage returns effectively displays the contents of consecutive cells. If ∧ (circumflex) is pressed instead of (CR), an open cell is closed, new data (if any) is entered, and the previous address cell is opened.
xJ	Jumps to location x and begins executing the program located there in F9445 absolute assembly language. Transfers from console mode to service I/O mode.
xR	Jumps to location x and begins program execution. This is identical to the jump (J) command, except the F9470 remains in the console mode. The console responds to keystrokes, such as A, T, or more R commands. When the R command is used, the F9470 does not respond to I/O instructions from the F9445, and program execution halts when another console command (e.g., A) is entered. Back-to-back R commands are not recommended.
S	Allows the user to reset the baud rate. Issue this command, attach the CRT cable to another device or re-set the baud rate of the terminal in use, then press (CR). The software of the F9470 sets the board baud rate to agree with the rate of the attached device. A second rate can then be software-programmed, in accord with the restrictions noted in figure 2.
nT	Traces through the user program n (octal) steps, beginning at the address pointed to by the PC counter or where the previous trace left off, whichever was last. If n = 0 or 177777, it will trace forever; if n is omitted, it traces one step. To start tracing at location x, set the PC counter to x with the command xE, then press the ESC key to terminate the examine mode; finally, use the appropriate T command to begin tracing. For every program step that is traced, the command T displays eight fields: the memory address, the instruction in octal form, the four accumulators, the stack pointer, and the frame pointer.

I/O Service Mode Operation

When performing I/O functions with the F9470, the following restrictions should be noted.

1. The busy flag is not set with input device codes 10 and 12, therefore, SKPBZ and SKPBN should not be used with these codes.

2. The F9470 requires some time after clearing the done flag to remove the associated interrupt request. When performing interrupt-driven I/O to the F9470, the operator needs to add a delay between the clear request and the next INTEN. Alternatively, the interrupt handler must be able to tolerate a bogus interrupt from the F9470. In this latter case, an INTA command will return a zero (0), which the interrupt handler should ignore, and then re-enable interrupts.

When in the I/O service mode, the F9470 responds to the F9445 data control instructions presented in table 2. These commands are a subset of the F9445 mnemonic instructions described in the F9445 data sheet.

Table 2 F9470 I/O Service Instructions

Instruction	Description
DIAx ACC,DEV	Data In from A
DOAx ACC,DEV	Data Out from A
NIOx DEV	No I/O; Used to Start or Clear a Device
SKPBN DEV*	Skip if Busy = 1
SKPBZ DEV*	Skip if Busy = 0
SKPDN DEV	Skip if Done = 1
SKPDZ DEV	Skip if Done = 0

NOTES:
If x = S (start), set busy flag, clear done.
If x = C (clear), clear busy flag, set done
ACC = Accumulator 0, 1, 2, or 3.
DEV = Device Codes = 10, 11, 12, 13
* Note that Busy is not defined for input devices (TTI and PTR); hence, SKPBN and SKPBZ should not be used with these devices.

The F9470 responds to the interrupt control instructions listed in table 3 when in the I/O service mode.

Table 3 F9470 Interrupt Control Commands

Instruction	Description
IORST	Clears all busy and done flags, disables interrupts.
MSKO,ACC,CPU	Enables or disables device interrupts by clearing or setting the interrupt disable flag in the device. The interrupt disable flag of each device is associated with a specific data line, and is set if its mask bit is 1, cleared if 0*.
INTA,ACC,CPU	Reads device code of highest priority device that is requesting an interrupt. The 6-bit code is loaded into ACC bits 10-15. All 16 bits are set to 0 of no device is interrupting.

***NOTE:**

Interrupt Disable Bits

IB Bits	8	9	10	11	12	13	14	15
Mnemonics				PTR		PTP	TTI	TTO
Function				CH2 In		CH2 Out	CH1 In	CH1 Out

The F9470 responds to device codes 10, 11, 12, 13, and 77, which are octal codes of the six least significant instruction bits:

Device Code	Mnemonic	Description	Action
10	TTI	Teletype In	Input on Channel 1
11	TTO	Teletype Out	Output on Channel 1
12	PTR	Paper Tape Reader	Input on Channel 2
13	PTP	Paper Tape Punch	Output on Channel 2
77	CPU	CPU	

Serial I/O

The F9470 has two asynchronous serial input/output ports. Each port can select 110, 300, 1200, 1800, 2400, or 4800 baud. The initial carriage return after power-up or after typing the S command allows the software to define the first baud rate. The second rate is programmed according to the mnemonic keystroke next entered (see figure 2 for valid rates). The second baud rate must be less than or equal to the rate of the first one.

Figure 2 Serial I/O Port Baud Rate Selections

MNEMONIC		0[2]	1	2	3	4	5
BAUD RATE FOR PORT 2		None	110	300	1200	1800	2400
	110		*[1]				
	300	*		*			
	1200	*	*	*	*		
FIRST BAUD RATE	1800	*				*	
	2400	*	*	*	*		*
	4800	*[1]					

* = Valid 2nd baud rate
1. Automatically
2. 0 signifies no second line

DC Characteristics

The dc characteristics of the console controller are provided in table 4.

Table 4 Console Controller DC Characteristics

Symbol	Parameter	Min	Max	Unit	Test Conditions
I_{CC}	Power Supply Current		100	mA	Outputs Open
P_D	Power Dissipation		550.0	mW	Outputs Open
V_{IHEX}	External Clock Input HIGH Voltage	2.4	5.8	V	
V_{ILHEX}	External Clock Input LOW Voltage	−0.3	0.6	V	
I_{HEX}	External Clock Input HIGH Current		100	mA	V_{IHEX} = 2.4 V
I_{ILEX}	External Clock Input LOW Current		−100.0	mA	V_{ILEX} = 0.6 V
V_{IH}	Input HIGH Voltage	2.0	5.8	V	
V_{IL}	Input LOW Voltage	−0.3	0.8	V	
I_{IH}	Input HIGH Current (except open drain and direct drive I/O ports)		100	mA	V_{IH} = 2.4 V, internal pull-up
I_{IL}	Input LOW Current (except open drain and direct drive ports)		−1.6	mA	V_{IL} = 0.4 V
I_{LOD}	Leakage Current (open drain ports)		±10.0	mA	Pull-down, device off V_{OH} = 13.2 V
I_{OH}	Output HIGH Current (except open drain and direct drive ports)	−100		mA	V_{OH} = 2.4 V
I_{OHDD}	Output Drive Current (direct drive ports)	−1.5	−8.0	mA	V_{OH} = 0.7 V to 1.5 V
I_{OL}	Output LOW Current	1.8		mA	V_{OL} = 0.4 V
I_{OHS}	Output HIGH Current (STROBE Output)	−300		mA	V_{OH} = 2.4 V
I_{OLS}	Output LOW Current (STROBE Output)	5.0		mA	V_{OL} = 0.4 V

(T_A = 0 °C to 70 °C, V_{CC} = 5 V ± 10%, I/O Power Dissipation = 100 mW)

Absolute Maximum Ratings

These are stress ratings only, and functional operation at these ratings, or under any conditions above those indicated in this document, is not implied. Exposure to the absolute maximum 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.

Temperature (ambient) under bias	
F9470DC	0 °C, + 70 °C
F9470DM	− 55 °C, + 125 °C
Storage temperature	− 55 °C, + 150 °C
Voltage on all open drain pins	− 1.0 V, + 13.2 V
Voltage on all other pins, with respect to ground	− 1.0 V, + 7.0 V
Power dissipation	1.5W

Ordering Information

Part Number	Package	Temperature Range*
F9470DC	Ceramic	C
F9470DL	Ceramic	L
F9470DM	Ceramic	M

*C = Commercial Temperature Range 0 °C to + 70 °C
L = Limited Temperature Range − 40 °C to + 85 °C
M = Military Temperature Range − 55 °C to + 125 °C

FAIRCHILD
A Schlumberger Company

FS-I
Fairchild System-I

Advance Product Information | Microprocessor Product

Description

The Fairchild System-I (FS-I) is a versatile, multi-user development system designed to support software development and hardware prototyping for applications using Fairchild microprocessors, including the F8, F3870, F6800, F6809, F9445, F16000, and such upcoming microprocessors as the F9450.

Three principal versions of the FS-I are available: The FS-I Standard System, the FS-I Multi-User System, and the FS-I Entry-Level System. Numerous software and hardware options are available that operate under Fairchild's Interactive Multi-User Disk Operating System (IMDOS). The FS-I also supports the in-circuit emulation and tracing (EMUTRAC™) system for the F3870, the F6800, F6809, and the F9445 microprocessors. (For a description of the EMUTRAC system, see *EMUTRAC Advance Product Information.*)

Standard System

System features include:

- **CPU with 128K-Byte RAM and F9445 Instruction Set.**
- **A Winchester and a Double-Density Floppy Drive Provide Approximately 10M-Byte of Mass Storage.**
- **I/O Controller Board Provides Winchester/Floppy Disk Controller Interface.**
- **Nine Asynchronous Serial RS-232C Ports (Up to 19.2K Baud) Provide Support for CRT Terminal, Optional Letter-Quality Printer, Modem, and Other Serial Devices.**
- **One Synchronous Serial RS-232C Port (Up to 19.2K Baud) and Selectable Protocols, such as BISYNC, DDCMP, SDLC, and HDLC.**
- **PROM Programmer Port to Interface to the Optional Fairchild PROM Programmer Unit.**
- **Parallel Printer Port (Centronics-Compatible Interface).**
- **Programmable Real-Time Clock.**
- **One CRT Terminal.**
- **Single-User Version of IMDOS, System Processors, and System Utility Programs (see "System Software").**
- **BASIC Language Interpreter with Interface to Custom F9445 Assembly Language Programs.**
- **FS-I Diagnostic Programs.**
- **Provides Full Support for the F9445 and for the PEP 45 Microcomputer System.**
- **Hardware and Software Upgradable to Multi-User System.**
- **EMUTRAC Can Be Added to the Standard System.**

Multi-User System

System features include:

- **Fully Equipped for Four Timesharing Users (Expandable to Eight Simultaneous Users with Additional Terminals and Cables).**
- **A 16-Bit CPU with 128K-Byte RAM and F9445 Instruction Set.**
- **A Winchester and a Double-Density Floppy Drive Provide Approximately 10M-Byte of Mass Storage.**
- **Memory Management and Protection Unit (MMPU) Board with 384K Bytes of RAM (Gives the System 512K Words of RAM).**
- **I/O Controller Board Provides Winchester/Floppy Disk Controller Interface.**
- **Nine Asynchronous Serial RS-232C Ports (Up to 19.2K Baud) Provide Support for CRT Terminals, Optional Letter-Quality Printer, Modem, and Other Serial Devices.**
- **One Synchronous Serial RS-232C Port (Up to 19.2K Baud) and Selectable Protocols, such as BISYNC, DDCMP, SDLC, and HDLC.**
- **PROM Programmer Port to Interface to the Optional Fairchild PROM Programmer Unit.**
- **Parallel Printer Port (Centronics-Compatible Interface).**
- **Programmable Real-Time Clock.**
- **Four CRT Terminals.**
- **Multi-User Version of IMDOS, System Processors, and System Utility Programs (see "System Software").**
- **BASIC Language Interpreter with Interface to Custom F9445 Assembly Language Programs.**
- **FS-I Diagnostic Programs.**
- **Provides Full Support for the F9445 and for the PEP 45 Microcomputer System.**
- **EMUTRAC Can Easily Be Added to the Multi-User System.**

Entry-Level System

System features include:

- **A 16-Bit CPU with 128K-Byte RAM and F9445 Instruction Set.**
- **Two Double-Density Floppy Disk Drives Provide Approximately 1M-Byte of Mass Storage.**
- **I/O Controller Board Provides Floppy Disk Controller Interface.**
- **Nine Asynchronous Serial RS-232C Ports (Up to 19.2K Baud) Provide Support for CRT Terminal, Optional Letter-Quality Printer, Modem, and Other Serial Devices.**

™ EMUTRAC is a trademark of Fairchild Camera and Instrument Corp.

Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

 Printed in U.S.A. 4027-054-A-23-022 5M 501995
March 1982

- **One Synchronous Serial RS-232C Port (Up to 19.2K Baud) and Selectable Protocols, such as BISYNC, DDCMP, SDLC, and HDLC.**
- **PROM Programmer Port to Interface to the Optional Fairchild PROM Programmer Unit.**
- **Parallel Printer Port (Centronics-Compatible Interface).**
- **Programmable Real-Time Clock.**
- **One CRT Terminal.**
- **Single-User Version of IMDOS, System Processors, and System Utility Programs (see "System Software").**
- **BASIC Language Interpreter with Interface to Custom F9445 Assembly Language Programs.**
- **FS-I Diagnostic Programs.**
- **Full Support for the F9445 and for the PEP 45 Microcomputer System.**
- **Hardware and Software Factory-Upgradeable to Standard or Multi-User System.**
- **EMUTRAC and MMPU Can Be Added to System.**

System Hardware

The hardware comprising the FS-I development system is housed in a single enclosure that contains the mainframe CPU, I/O board, optional boards, and disk drives.

The mainframe consists of:

- **Single-Board 16-Bit CPU with 128K Bytes of RAM, 4K-Byte PEPBUG45 PROMs for Bootstrapping the System, Real-Time Clock, an RS-232C-Compatible Port, and a Centronics-Parallel Compatible Port.**
- **Power Supplies.**
- **I/O Controller Board with the Following:**
  - **Eight Asynchronous Serial RS-232C Ports, with Four Ports Having Full Modem Control and All Ports Having Data Rate Selectable Up to 19.2K Baud, that Allow Timesharing by Up to Eight Concurrent Users on Systems Equipped with MMPU Board and Multi-User Operating System Software.**
  - **One Synchronous Serial RS-232C Port (Up to 19.2K Baud) and Selectable Protocols, such as BISYNC, DDCMP, SDLC, and HDLC.**
  - **A Parallel Data Channel Interface Compatible with Shugart Associates System Interface for Communicating with Disk Units.**
  - **8-Bit Parallel Port to Interface with Optional Fairchild PROM Programmer.**
- **A Total of Nine Asynchronous Serial Ports (RS-232C-Compatible, DB25-Pin Female Connectors).**
- **One Parallel Printer Port (Centronics-Compatible Interface, DB25-Pin Connector).**
- **Expansion Slots for Fairchild's Optional I/O Controller Boards, Optional EMUTRAC Controller Board, Memory Expansion Boards, MMPU Board, and Industry-Standard, Nova® I/O-Compatible Interface Boards.**
- **Depending Upon System Configuration, the Mainframe Contains a Single 10M-Byte Winchester and a Single 0.5M-Byte Double-Density Floppy Disk Drive or Two 0.5M-Byte Double-Density Floppy Disk Drives.**

The MMPU board expands the physical address space of the FS-I to 4M words by performing logical-to-physical address translation. This board is required for multi-user system software. With its 384K bytes of RAM, the MMPU board extends the FS-I memory to 256K words.

Hardware Options

The FS-I systems support the following Fairchild-supplied hardware options:

- **Additional I/O Controller Boards that Provide Asynchronous RS-232C Ports (Up to 19.2K Baud) in Sets of Eight, a Synchronous RS-232C Port for each I/O Controller Board, Data Channel Interface to Disk Units, and a PROM Programmer Port for each I/O Controller Board.**
- **Fairchild's PROM Programmer Unit.**
- **MMPU Board that Provides Memory Mapping and Protection Expansion in Increments of 384K Bytes, Optional Multi-User Software Allows the MMPU Board to Support Eight Simultaneous Users.**
- **Memory Expansion Board that Provides 384K Bytes of Additional RAM (Requires an MMPU Board in the Chassis).**
- **EMUTRAC System Controller Board that Provides the Hardware Interface Between the CPU Board in the FS-I and Processor-Specific EMUTRAC Modules.**
- **EMUTRAC Modules and EMUTRAC Control Software that Support the F3870, the F6800, the F6809, and the F9445 Microprocessors.**
- **Additional CRT Terminals.**
- **Dot Matrix Printer—Texas Instruments Model 810 Basic RO Terminal (150 CPS), Centronics Parallel Interface, and Cable.**
- **Daisywheel Letter-Quality Printer—Qume Model Sprint 9145 with Bidirectional Forms Tractor (45 CPS), Serial Interface, and Cable.**

® Nova is a registered trademark of Data General Corp.

This powerful software package, which is included with the standard, multi-user, and entry-level systems, offers advanced capabilities that the user would normally expect from a much larger system, such as:

- **Multi-User Timesharing**
- **System Executive, Including File Management System with Version Numbers for Automatic Backup**
- **Memory Management and Protection by Memory Mapping**
- **Password Protection**
- **Interactive Command Language and Command Files**
- **Multiple Directory Devices**
- **Device-Independent I/O**
- **Hard Disk, Magnetic Tape, Modem, and Real-Time Clock Support**
- **Documentation Aids**
- **Concurrent Processing and Spooling**

System Software

The interactive multi-user disk operating system (IMDOS) is the principal operating system for the FS-I. In addition to being an operating system, the IMDOS includes the following features that are useful for developing F9445-based systems:

IMDOS	Single-User Supervisor—The supervisor manages the FS-I resources and controls the I/O.
IMDOS	Multi-User Supervisor—The supervisor manages the FS-I resources for up to eight simultaneous users, controls the I/O, and interfaces transparently to the MMPU board (included only with the multi-user system).
IMDOS	Executive—The executive provides the command language interface between the user and the supervisor.
EDIT	The EDIT program provides the ability to create and modify text files.
MACRO	The MACRO program is the macroassembler for F9445 macro assembly language.
RELOAD	The RELOAD program is used to link relocatable macro assembly language programs to create executable F9445 absolute assembly language programs.
PEPBUG45	The PEPBUG45 program is a virtual console and debugging tool for F9445 absolute assembly language programs. The PEPBUG45 program is also available in PROM.
PEPLINK45	Provides capability to download programs from the FS-I to PROM or RAM on the PEP 45 microcomputer system.
Utility Library	Implements the utility functions listed in the IMDOS and utility library users guides.
PHONE	The PHONE program establishes communication between the FS-I and a modem or telephone line. Software switches govern communication protocols.
SCRIPT	The SCRIPT program processes a text file that contains SCRIPT commands to produce an aesthetically pleasing document.
TYPESET	The TYPESET program processes a text file that contains TYPESET commands to produce an aesthetically pleasing document.
DEBUG	The DEBUG program is a debugger for F9445 macro assembly language programs.
DIAGNOSTICS	A series of programs that test the FS-I hardware. The diagnostic programs are available on diskette in a version suitable for downloading to an F9445-based system.
BASIC	Language interpreter with interface to custom F9445 assembly language programs.

Software Options

F9445 MICROFORTRAN	An extended subset of FORTRAN66 that interfaces with custom F9445 assembly language subroutines. MICROFORTRAN produces "ROMable" F9445 code and can be operated under the real-time executive (REX).
F9445 PASCAL	A Jensen and Wirth-compatible PASCAL. The F9445 PASCAL compiler generates F9445 code and interfaces with custom F9445 assembly language subroutines.
FS-I/PEP 38 System Software	Includes F8/F3870 cross assembler and program for downloading to the PEP 38 system.
FS-I/PEP 68 System Software	Includes F6800 cross assembler, F6809 cross assembler, F6800-to-F6809 translator program, and program for downloading to the PEP 68 system.
F16000 Cross Software	Assembler, debugger, and downloader allow the FS-I to generate 16000 code that can be downloaded to an F16000-based system.
F9445 REX	A real-time executive for F9445-based systems. The REX system allows creation of custom REX programs, linkable using RELOAD.
F9445 PEPBASIC	A diskette version of PEPBASIC (supplied on PROM with the PEP 45 system). A 2K-word subset of BASIC, which accepts abbreviations, that is extendable with custom F9445 assembly language subroutines.
EMUTRAC Control Software	Optional EMUTRAC control software packages provide support for each processor-specific EMUTRAC module. (Refer to *EMUTRAC Advance Product Information.*)

In addition, all Fairchild software for the FS-I is independently available without system purchase under an appropriate software license agreement.

Dimensions and Power Requirements

The FS-I standard mainframe enclosure measures only 26 inches long by 19 inches wide by 13 inches high. It requires a 115 V, 60 Hz ac power source. A 50 Hz system is also available.

FAIRCHILD

A Schlumberger Company

PEP-45
Prototyping, Evaluation and Programming Board

Advance Product Information

Microprocessor Product

Description

The Fairchild PEP-45 is a single-board microcomputer for Prototyping, Evaluation, and Programming of microprocessor-based system applications using the F9445 microprocessor. When used with the Fairchild System-I (FS-I) development system, the PEP-45 board provides capability for executing and debugging software directly on the F9445 microprocessor.

- **Stand-alone Prototyping, Evaluation, and Programming Board.**
- **Provides a Powerful Development Tool to Support F9445 Microprocessor-based System Development.**
- **Utilizes All the Advantages of the F9445 Microprocessor, with its Powerful Instruction Set and High Throughput.**
- **Memory Options for Bipolar and NMOS Memories.**
- **Interfaces with IEEE 796 Standard Bus.**
- **Buffered F9445 bus.**
- **On-board EPROM Programmer.**
- **Adapts to 16K or 32K Byte EPROMs or 64K Byte Masked ROMs.**
- **Standard- and High-Speed RAM Options.**
- **Console Commands.**
- **Two Serial I/O Ports.**
- **16-Bit Parallel Input/Output.**
- **Four Interrupt Sources.**
- **Five Status Lines.**
- **On-board +12 V and +25 V Voltage Converter.**
- **Requires Single +5 V Power Supply.**

The PEP-45 board is primarily intended for use in hardware prototyping and software development applications. It may also be tied to a host computer, such as the FS-I, for large program editing, assembling/compiling, and general file storage and handling. Cross-assembler software packages are available for creating machine-executable programs in formatted form. These programs may be down-loaded from the host computer system into the PEP-45 board via one of the two serial I/O channels. Since the PEP-45 board can operate in a transparent fashion, it may be placed in-line between the user's in-house terminal and the host computer, giving the PEP-45 the power of the host.

Also useful for incoming inspector of F9445 parts and as a microcomputer training tool, the PEP-45 interacts with the user at the control terminal, with prompts that assist programming. The control terminal may be a video terminal, printer terminal, or from a microcomputer control console.

Software Support

In addition to serving as an efficient stand-alone evaluation module, the PEP-45 is designed to operate as a key module of the FS-I development system. A PEPLINK utility transparently couples the FS-I video terminal to the PEP-45 board.

A powerful PROM-based PEPBUG debugging monitor provides commands for trouble-shooting assembly language programs and for developing and testing peripheral circuits and custom interfaces. A PROM-based PEPBASIC language allows programming in a high-level language.

Hardware Specifications

CPU	F9445
Data word size	16 bits
Instruction word size	16 bits
Address capability	128K bytes
Console controller	F9470
Memory	
RAM	8K bytes (4K words) static RAM (or optional high-speed RAM)
ROM	Eight sockets for 16K bytes of F2716 EPROMs (8K words), or up to 32K bytes using F2732 EPROMs (16K words), or masked 64K byte ROMs using F3564
Expansion	External memory in any combination of RAM or ROM up to 64K bytes maximum (in 16-bit-wide only)
Input/Output	
Parallel I/O	Two TTL-compatible, 16-bit I/O ports (one input, one output)
Serial I/O	Two programmable, asynchronous channels, with RS-232 interfaces. Each channel is software-selectable to a baud rate of 110, 300, 1200, 1800, 2400, or 4800 baud
Real-Time Clock	Continuously selectable real-time clock interrupts from approximately 200 μs to 200 ms

Fairchild Camera and Instrument Corporation, Microprocessor Division
3420 Central Expressway, Santa Clara, California 95051
(408) 773-1000 TWX MICPSNTA

System Buses

Dual backplane buses	PI — An 86-pin asynchronous system bus compatible with standard Multibus 16-bit slave boards and multi-master option
	P2 — A 60-pin buffered F9445 bus that allows complete expansion of processor capabilities and faster operating speeds
I/O buses	J1 — A 9-pin RS-232C serial I/O interface for control terminal
	J2 — A 9-pin RS-232C second serial I/O interface for a serial printer or a host computer
	P3 — A 40-pin applications connector with two parallel I/O ports (one input and one output), and with status and control bits. May be used for connection to the microcomputer control console or to a high-speed parallel printer (Centronics-type)
Connectors	P1 — An 86-contact, double-sided edge connector on 0.156″ centers
	P2 — A 60-contact, double-sided edge connector on 0.100″ centers
	J1, J2 — 9-pin, D-type subminiature right-angle connectors
	P3 — A 40-pin, D-type subminiature right-angle connector

Power Supply Requirements[1] +5 V ±5% at 3.5 A (typ)

Environmental Requirements

Temperature	0°C to +50°C
Humidity	0% to 90% (noncondensing)

Physical Envelope Dimensions[2]

Height	10.0 (254)
Length	12.0 (305)
Thickness	0.75 (19.05)
Weight	17 oz. (approximately)

Notes

1. Power may be applied to the board either through the card-edge backplane connector or by connection of discrete wires to the board.
2. All dimensions are in inches and millimeters (in parentheses).

Fairchild cannot assume responsibility for use of any circuitry described other than circuitry embodied in a Fairchild product.

Fairchild reserves the right to make changes in the circuitry or specifications at any time without notice.

Ordering Data

Part Number	Product Code	Description
PEP 9445SFX	A F944516PEP	PEP-45 Board with 8K byte PROM sockets populated with PEPBASIC and PEPBUG firmware. Firmware carries copywriter notice. Minimum of four PROM sockets will not be populated. PEP-45 Users Guide, PEPBASIC, and PEPBUG Users Guide supplied.
PEP 9445SXX	A F944516PEP	PEP-45 Board with 8K byte static MOS and eight PROM sockets not populated. PEP-45 Users Guide supplied. No firmware included.
PEP 9445HXX	A F944520PEP	PEP 9445 Board with 8K byte high speed RAM and PROM sockets not populated. PEP-45 Users Manual supplied. No firmware or users guides included.