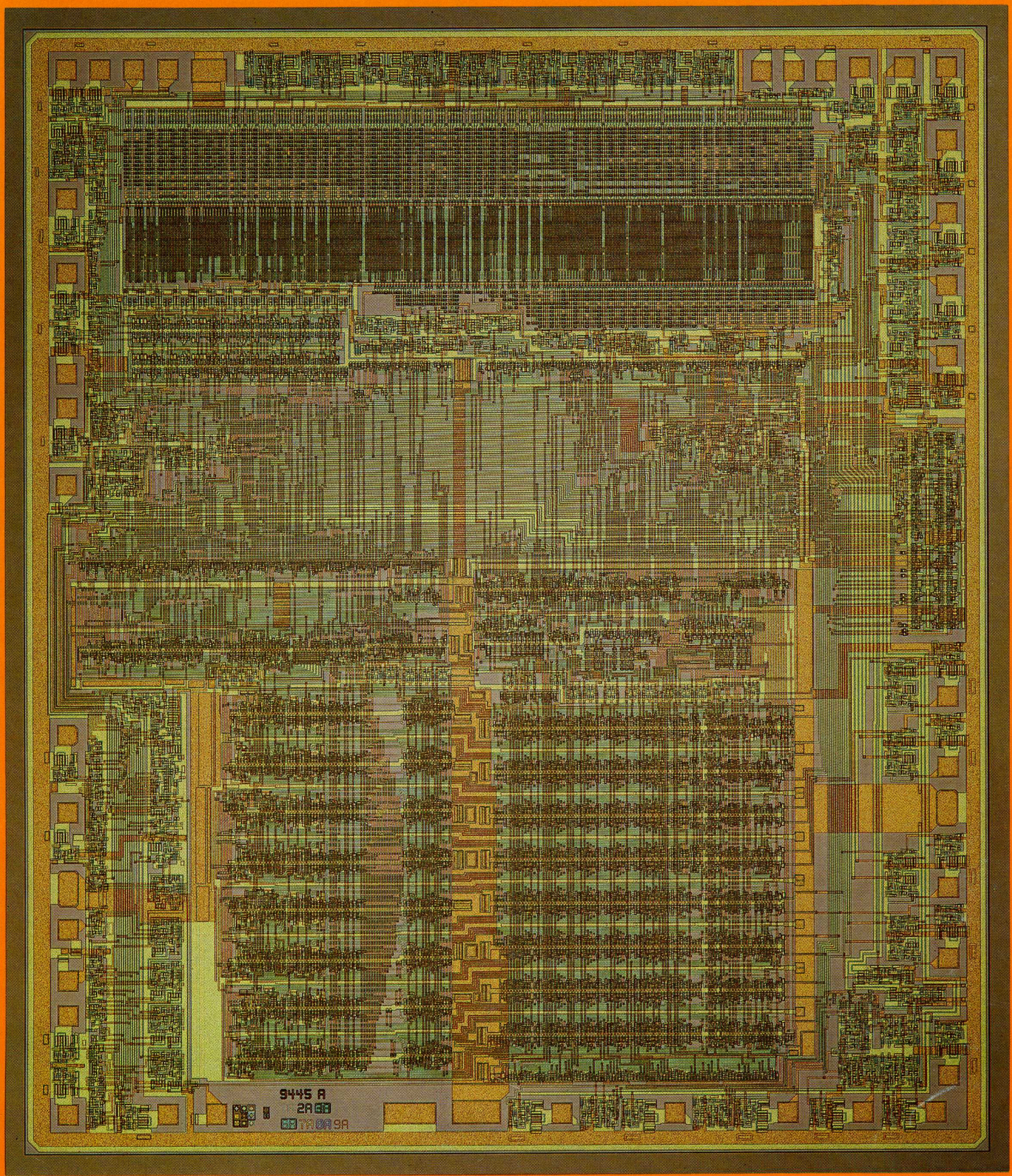
FAIRCHILD
A Schlumberger Company

F9445 16-BIT MICROPROCESSOR



Microprocessor Division
450 National Avenue
Mountain View, California 94042

(Reprinted from Fairchild Journal of Semiconductor Progress, First Quarter 1982)

Printed in U.S.A.

The F9445 Microprocessor Family

The F9445 16-bit, high-speed CPU is supported by a family of several specialized peripheral circuits, a development system capable of supporting all Fairchild 8-bit and 16-bit microprocessors, an in-circuit emulator and other design aids, as well as a software library supporting the power and capabilities of this system approach. Peripheral circuits provide the following auxiliary functions.

Dynamic memory control
Memory management
Input/output control
Communications control
Data channel control
Operator console control

Using a bipolar VLSI technology, Isoplanar Integrated Logic (I^3L^{TM}), a full 16-bit microprocessor has been manufactured and support circuits have been developed which now make it possible to obtain throughput with a single-chip CPU that in the past was generally obtainable only with a multi-device bit-slice implementation. The high speed microprocessor is particularly well-suited for real time control and signal processing applications. Since the device and its supporting peripheral chips have been implemented in bipolar I^3L technology (see page 16), speed does not degrade with temperature. Full speed performance is obtainable at the military standard +125°C operating case temperature. The device is well-suited for the harsh environments that are typical of high-speed real time control applications. *Table 1* shows that the F9445 provides a 1.3 MIP throughput for a typical real time processing instruction mix.

	Weight	F9445 (20 MHz)
Load and Store	36%	0.6
Add and Subtract (Reg-Reg)	14%	0.3
Multiply	6%	3.5
Divide	1%	4.4
Test and Branch	30%	0.6
Logical (Reg-Reg)	8%	0.3
Shift (1 Bit)	4%	0.3
I/O Control	1%	0.6
Throughput (MIPS*)		1.311

*MIPS = million instructions per second.

Table 1 Processor Throughput Computations.

The F9445 Microprocessor

The core of the microprocessor family is the bipolar VLSI 16-bit CPU which has very fast execution times - 250 ns for register-to-register operation and 2.9 μs for multiplication at 24 MHz. At 20 MHz, execution times are 300 ns for register-to-register operation and 3.5 μs for multiplication. The instruction set is a super set to an industry standard and includes powerful floating-point assist instructions such as Normalize, generally not available in 16-bit microprocessors. The F9445 processor design includes provisions for efficient multiprocessor system operations, including basic support for bus sharing. It supports multi-user demand-paged virtual-memory applications via an Instruction Abort function.

Although well suited for general-purpose control applications, the F9445 excels in number-crunching and signal-processing applications, achieving a throughput of over 1.3 million instructions per second (MIPS) at a nominal clock rate of 20 MHz. Recently, an eighth-order elliptic digital low-pass filter was implemented[2] using the following 16-bit microprocessors: MC68000, Z8000, 8086, TMS9900, F9445. This is a class of applications where the F9445 particularly excels as shown by the results of that comparison (*Table 2*).

Performance times (in μs) of 16-bit microcomputers for an eighth-order cascaded filter.

Program	MC68000	Z8000	8086	TMS9900	F9445
Filter (1 loop)	24.75	30.75	53.2	69.1	10.1
Input (no delay)	10.25	15.5	14.8	24.2	2.8
Outp_1D	65.5	127.5	174.4	211.5	40.8
Delay_1D	32.0	30.25	32.8	135.8	33.2
Pre_1D	194.5	380.25	582.4	559.4	108.0
Total Sample Time	327.0	594.0	855.6	1000.0	193.9
Time Lag from Input to Output	82.25	156.25	212.8	253.9	47.8

Multiply times of 16-bit microcomputers using addressing modes selected for 1D module subroutine.

Processor	Clock Rate	16 x 16-Bit Multiply Time	Total Multiply Time in 4 1D Modules
MC68000	8.0 MHz*	9.75 μs	205.0 μs
Z8000	4.0 MHz	18.0 μs	378.0 μs
8086	5.0 MHz	30.6 μs	643.0 μs
TMS9900	3.3 MHz	18.2 μs	382.0 μs
F9445	20 MHz	3.5 μs	73.5 μs

*One internal state = two clock cycles, giving an effective clock rate of 4 MHz.

Table 2. Microcomputer Performance Times, Multiply Times

2. Nagle, H. T., and Nelson, V. P., "Digital Filter Implementation on 16-Bit Microcomputers," IEEE Micro, Vol. 1, No. 1, (February, 1981) pp. 23-41.

Architecture

The F9445 has a 16-bit data path governed by a 1-level pipelined microprogrammed control unit. A 1-level pre-fetch mechanism allows the subsequent instruction to be fetched during execution of the current instruction. This improves execution speed without the additional overhead required for multi-level pre-fetch mechanisms. Efficiency is further improved because all F9445 instructions are one size— 16 bits wide.

There are three basically parallel sections in the F9445 microprocessor (*Figure 2*).

- Data execution
- Bus and address processing
- Pipelined microprogrammed control

Internal and external strobes are generated by the timing and strobe generator. In a typical cycle, the following functions can occur simultaneously.

- Two registers from the register file are processed by the ALU and the result is loaded into one of them
- Next instruction is fetched on the bus and loaded into the Instruction Register (IR)
- Program Counter (PC) is incremented
- Next micro-instruction is fetched from the control store (PLA) and loaded into the microprogram register

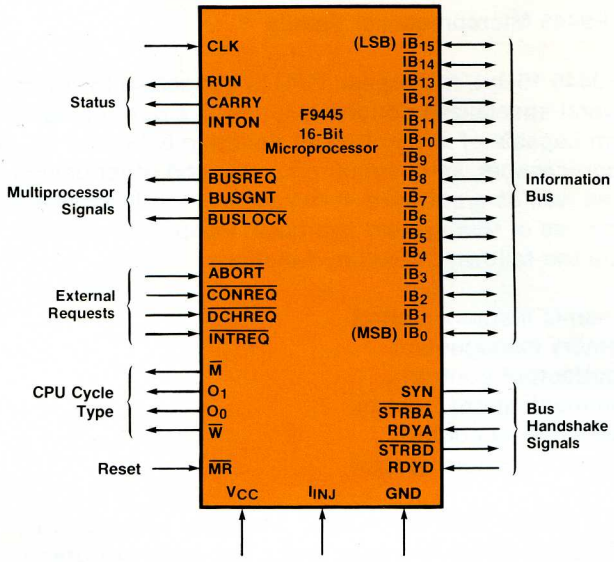


Fig. 1 F9445 Microprocessor

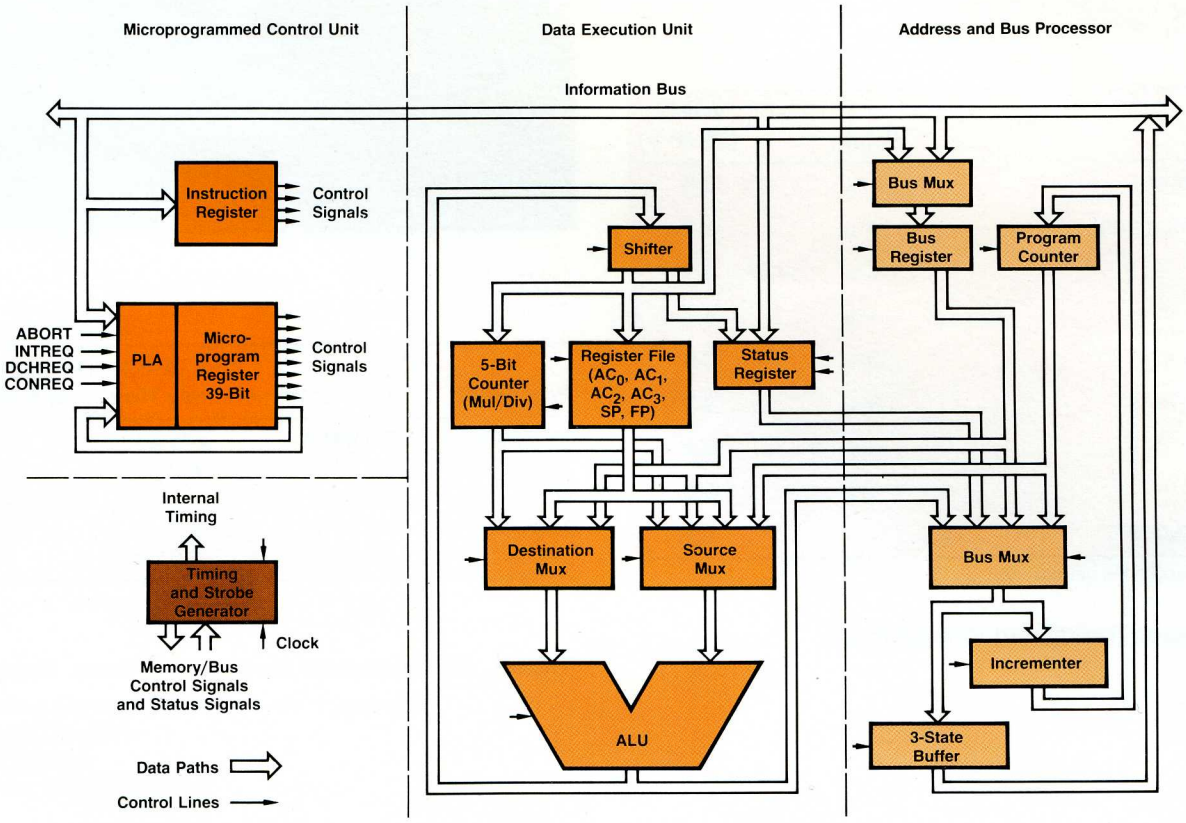


Fig. 2 F9445 Block Diagram.

Data Execution

The data execution unit includes register files (AC0, AC1, AC2, AC3, FP, SP), a 16-bit ALU, and a 17-bit shifter capable of shifting left, right, or swapping 17 bits (ALU plus CARRY). Design efforts have optimized speed for arithmetic operations, making this processor extremely fast with numbers. This goal was achieved by fine tuning the hardware and the micro-algorithms to achieve functions such as shifting two accumulators (AC0, AC1) as one entity, changing ALU codes during operation (ADD/MOV is unsigned multiply, ADD/MOV/SUB is signed multiply, ADD/SUB is divide), a fast overflow scheme, and an on-chip counter. The counter freezes the control unit when the data execution unit executes repetitive cycles (MUL/DIV). It is user-accessible (via AC2) on double shifts (containing the parameter N) and Normalize (containing the resulting shift count required to normalize).

Bus and Address Processing

The bus and address processing unit includes a bus register, responsible for communication with the bus and other related tasks such as sign extending the displacement (8 bits) in memory reference instructions. It contains an incrementer for sequencing the program counter in parallel with the data execution unit.

Microprogram Control

The microprogram control unit is 39 bits wide and is coded diagonally. Additional control bits are supplied by the instruction register. These bits don't change during the microsequence of an instruction execution, e.g., source and destination fields, nor does the no-load bit in an ALU instruction, or the specified register in a memory-reference instruction.

The Register Set

There are eight user-accessible registers in the F9445 (see *Figure 3*), including seven 16-bit registers and a program status word (PSW) register. The seven 16-bit registers are a program counter (PC), four general-purpose accumulators (AC0 through AC3), a stack pointer (SP), and a frame pointer (FP).

The program counter sequences instruction execution. The four accumulators serve as source and destination registers for 16-bit arguments in arithmetic and logic operations. The CPU processes the contents of these accumulators and stores the 16-bit result in the destination accumulator. The carry and overflow flags are set or cleared depending on the result of the ALU operation and the base value of carry (specified in the instruction). Accumulators AC2 and AC3 also serve as index registers during memory address operations. In addition, AC3 functions as a subroutine linkage register, and the AC0/AC1 pair are used as a 32-bit register in the multiply/divide and the normalize and parametric double-shift instructions. AC2 also serves as a linkage register to the internal counter. The counter is directly accessible only in the microprogram level, and thus in parametric double shifts, the parameter (N: number of required shifts) is preloaded in AC2 and then to the counter. In the Normalize instruction, the resulting shift count is loaded to AC2.

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

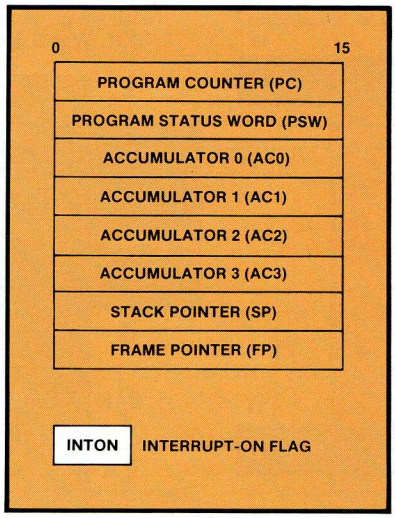


Fig. 3 F9445 Register Model.

The PSW register contains four flags: carry, overflow, 32Kword, and trap enable. The carry flag (C) indicates carry bit status, the overflow flag (V) indicates twos complement overflow, the 32K-word flag (32K) indicates the address range, 32KW (64K byte) or 64KW (128K byte), and the trap enable flag (ETRP) indicates status of the trap.

Separately, 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.

F9445 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 F9445 can operate in two address ranges — 128K-byte (64K-word) or 64K-byte (32K-word) logic address space (*Figure 4*). Upon Master Reset, the F9445 returns to the 64K-byte (32K-word) address range. The 128K-byte (64K-word) address range is enabled or disabled via program control. A detailed list of addressing modes is specified in *Table 3*.

In the 32K-word range, the address field is 15 bits wide, and the available addressing modes are page zero, PC relative and indexed for both direct and indirect modes. An indirect-mode location reference with the high-order bit equal to "1" generates another indirect reference to provide a multi-level indirect addressing mechanism.

A pre-addressing mode with auto-increment and auto-decrement is available when referencing indirectly, performing with auto-increment for address from 20 to 27 (octal) and with auto-decrement for address from 30 to 37 (octal). Byte addressing is provided by a 16-bit byte pointer in the accumulators.

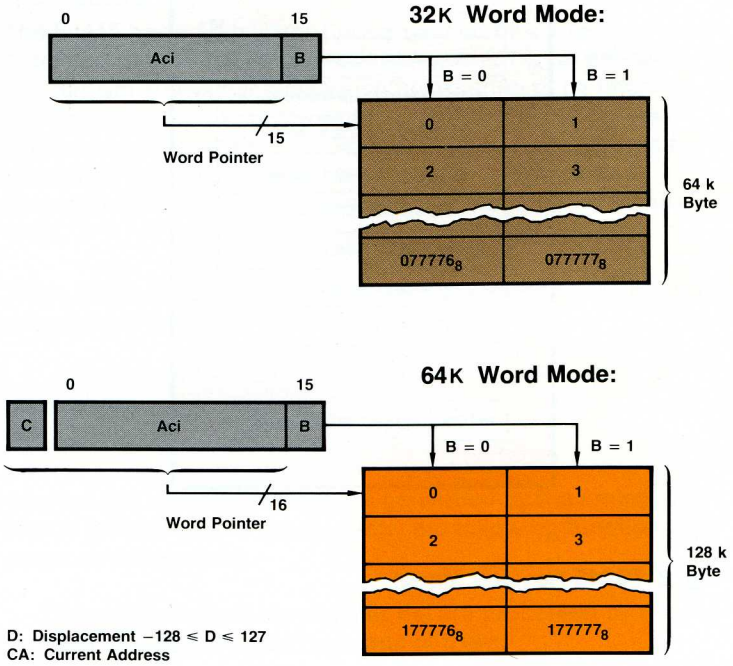


Fig. 4 Byte Addressing

In the 64K-word range, the address field is 16 bits wide. All addressing modes are available, but with these exceptions.

Only one level of indirect — the one specified in the instructions — is available.
Byte addressing has a 17-bit pointer including the specified accumulator plus a carry bit (high order bit).

Table 3. Addressing Modes Summary

Direct	
page zero	EA = D
PC relative	EA = CA + D
PC indexed by AC2	EA = AC2 + D
PC indexed by AC3	EA = AC3 + D
Indirect	
page zero	EA = (D)
PC relative	(CA + D)
PC indexed by AC2	EA = (AC2 + D)
PC indexed by AC3	EA = (AC3 + D)
Auto-increment	If address accessed indirectly is in the range 20-27 octal, then its contents are incremented and written back. The respective EA is: EA = (D) + 1 EA = (CA + D) + 1 EA = (AC2 + D) + 1 EA = (AC3 + D) + 1
Auto-decrement	If address accessed indirectly is in the range 30-37 octal, then its contents are decremented and written back. The respective EA is: EA = (D) − 1 EA = (CA + D) + 1 EA = (AC2 + D) − 1 EA = (AC3 + D) − 1
Multi-level (applies to 32K-word only)	If high order bit of (EA) is HIGH then EA = (EA)
Stack addressing (Read/Write top of stack)	EA = (SP)

F9445 Instruction Set

COM	–	Complement
NEG	–	negate
MOV	–	move
INC	–	increment
ADC	–	add complement
SUB	–	subtract
ADD	–	add
AND	–	and
OR	–	or
MUL	–	unsigned multiply
MULS	–	sign multiply
DIV	–	unsigned divide
DIVS	–	signed divide
NORM	–	normalize
SLLD	–	double logic shift left
SALD	–	double arithmetic shift left
SARD	–	double arithmetic shift right
SLRD	–	double logical shift right
SKNV	–	skip on no overflow
JMP	–	jump
JSR	–	jump to subroutine
ISZ	–	increment and skip if zero

and: no shift/shift/rotate
and: no skip/always skip/skip on any combination of carry and zero result
and: load/no load destination register

DSZ	–	decrement and skip if zero
LDA	–	load word
STA	–	store word
LDB	–	load byte
STB	–	store byte
PSHA	–	push
POPA	–	pop
PSHF	–	push flags
POPF	–	pop flags
POPJ	–	pop and jump
PSHR	–	push return address
TOPR	–	read top of stack
MTSP	–	move to stack pointer
MFSP	–	move from stack pointer
MTFP	–	move to frame pointer
MFFP	–	move from frame pointer
SAV	–	save return block
RET	–	return
DSP	–	decrement stack pointer
NIO	–	no I/O
SKP	–	skip on busy/done
DIA/DIB/DIC	–	input (A/B/C)
DOA/DOB/DOC	–	output (A/B/C)
INTEN	–	interrupt enable
INTDS	–	interrupt disable
READS	–	read switches
INTA	–	interrupt acknowledge
MSKO	–	mask out
IORST	–	clear I/O devices
HALT	–	halt
SKP	–	skip on interrupt
WAIT	–	wait for interrupt
TRAP	–	trap
ETRP	–	enable traps
DTRP	–	disable traps
E64K	–	enable 64K mode
D64K	–	disable 64K mode

Instruction Set

The F9445 has 60 basic fixed-length instructions of 16 bits divided into several fields. These 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 displacements for the calculation of effective addresses of memory locations. The entire instruction set can be considered in five broad categories.

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

Arithmetic and Logic Functions

The arithmetic and logic instructions manipulate the register file, carry, and overflow. In a typical instruction the following activities occur.

Registers are operated
Result may be shifted/rotated with or without carry
Result may be loaded to destination register or not loaded, i.e., compare operations
A skip condition is specified (always skip, never skip and any combination of carry and zero result)

This category includes a number of powerful instructions not available in other 16-bit microprocessors such as a Normalize instruction. The Normalize instruction, when combined with other instructions such as full 32-bit signed and unsigned multiply/divide and arithmetic/logic parametric double shifts, makes floating point packages and number crunching in general extremely fast and efficient.

Memory Reference Instructions

Memory reference instructions modify the contents of memory locations, move operands (words, bytes) between the register file and memory, and alter program execution sequence.

Stack Manipulation Instructions

Stack manipulation instructions manipulate the registers and memory in stack-associated operations. The stack operations use a stack pointer (SP) that points to the top of stack and a frame pointer (FP) that serves as an environment pointer, pointing to the "return block" (*Figure 5*).

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. These instructions and PSHF (push flags), and POPF (pop flags), provide for very efficient context switching. 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: the contents of AC0; the contents of AC1; the contents of AC2; the value of FP before the Save; and the contents of AC3. At this time, 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 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 the values preceding the Save to accumulators AC0, AC1, and AC2. FP is restored to its previous value (pointing to the last previously saved 5-word frame), and PC is loaded with the return address which had been placed in AC3 by the previous Jump-to-Subroutine and pushed

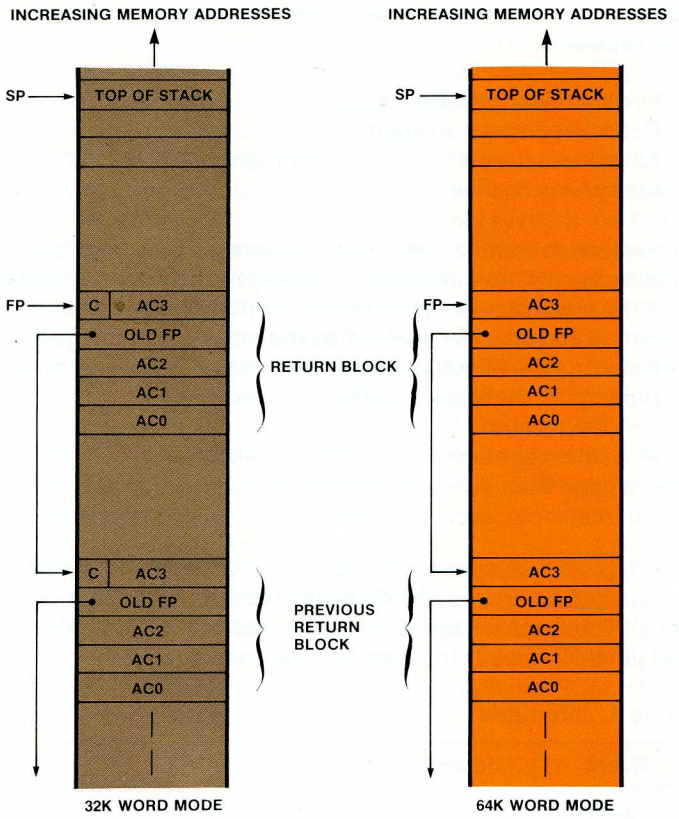


Fig. 5 Data Organization in a Stack (LIFO)

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 Move instructions MTFP, MFFP, MTSP, and MFSP. Thus, multiple stacks can be established with pointers saved in main memory while they are not active. When Push instructions (PSHA, PSHF, PSHR, SAVE, TOP) attempt to write over a page boundary (mod 256), a stack fault is generated, causing an interrupt. If INTON is HIGH, the F9445 completes the instruction and then services the interrupt (see Interrupt discussion).

I/O Instructions

I/O instructions affect data transfers between the register file and I/O devices. In addition to standard data transfer between a register and an I/O device, there are Skip instructions on the device status flags.

Control Instructions

The control instructions modify or interrogate the state of the CPU, the operator console, and I/O devices. These include enabling/disabling the INTON, 32K, and ETRP flags, or a wait for interrupt instruction, extremely useful in multiple microprocessor environments.

Interrupts, Trap, and Abort

The F9445 recognizes the following events.

Interrupts (external and internal)
Trap (software interrupt)
Abort

External interrupts may be initiated by an external device using the $\overline{\text{INTREQ}}$ line. The F9445 completes the instructions being

executed, and if interrupts are enabled proceeds as follows. See *Table 4*.

Further interrupts are disabled
Return address is saved in memory
An indirect jump through the interrupt vector is executed to the service routine.

In the service routine, the F9445 may execute the interrupt acknowledge (INTA) instruction, allowing the interrupting device with the highest priority to identify itself. At the end of the routine, interrupts are enabled by the interrupt enable (INTEN) instruction. The F9445 can selectively mask any device from interrupting via the mask out (MSKO) instruction.

Internal interrupt is initiated when the stack overflows (writing over a page boundary). CPU response is similar to that for external interrupts, except the stack interrupt vector is used.

The trap instruction is a software interrupt that can be enabled or disabled (becomes a NOP) by the user (ETRP/DTRP instructions). The machine saves the return address (PC) in memory and jumps indirectly through the trap vector.

Table 4. Interrupts

Event	Address	
External Interrupt	0	Save Return Address (PC)
	1	Interrupt Vector
Stack Fault	0	Save Return Address (PC)
	3	Stack Interrupt Vector
ABORT	46	Save Aborted Instruction Address
	47	Abort Vector
Software Trap	46	Save Return Address (PC)
	47	Trap Vector

Abort causes the processor to abort the current instruction. The aborted instruction address is saved in memory and a jump indirect through the abort vector is executed. This abort may be used when address fault occurs in a demand-paging virtual-memory mechanism to generate a system call that will start the page-swapping process. All the save cycles can be externally detected by a code on the bus status lines and used, for example, to switch an external mapper to non-mapped mode.

F9445 Interfacing

The F9445 requires a 20 MHz (nominal) clock which can be generated by a monolithic crystal-controlled oscillator. Static memory is easily connected to the F9445 through the set of interface signals provided (*Figure 6*). The bus is multiplexed, so a Memory Address Register (MAR) is required. Two standard octal latches can serve this purpose. The $\overline{\text{STRBA}}$ signal is used to latch the address. The write enable signal for the memories is directly generated by the F9445 ($\overline{\text{W}}$). The chip selects will be a decode of some of the address bits, with the M signal providing one enable for the decode, and the $\overline{\text{STRBD}}$ timing signal providing another enable. The timing variations required by read and write cycles are accommodated by the F9445. For slow memories, the RDYD handshake signal can stretch the $\overline{\text{STRBD}}$ pulse as long as required. If required, RDYD can be generated by a one-shot, a counter, or a shift register. To provide maximum flexibility the address can be held on the bus by using the handshake signal RDYA. This is useful for extended bus systems.

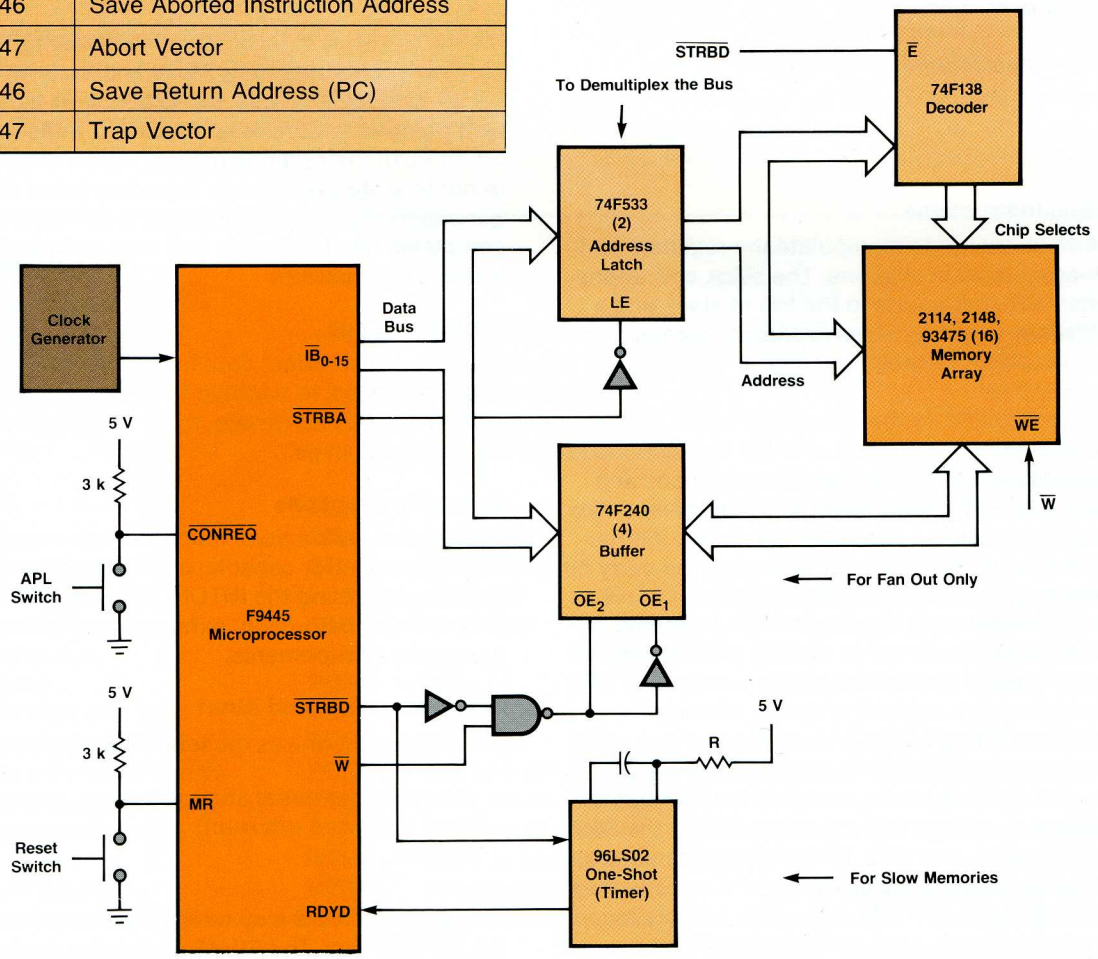


Fig. 6 Static Memory Interface

A set of signals is provided for multiprocessor operation. $\overline{\text{BUSLOCK}}$ is a signal active during read-modify write type instructions, for example DSZ, ISZ. In a multiprocessor system, it provides inter-processor synchronization by inhibiting the shared use of memory during these critical instructions. $\overline{\text{BUSREQ}}$ and BUSGNT provide a bus arbitration handshake. $\overline{\text{BUSREQ}}$ is active whenever the F9445 requires the use of the bus. BUSGNT is an input signal. If this signal is inactive, the F9445 halts when it requires use of the bus, with its information bus pins in a 3-state mode. Multiprocessor operation signals offer a variety of configurations, the simplest being two processors sharing the same bus (*Figure 7*).

Data channel is treated in a different manner. A separate $\overline{\text{DCHREQ}}$ pin indicates a data channel cycle, once execution of the current instruction is finished. The F9445 does not output address or data, but does output the usual $\overline{\text{STRBA}}$ and $\overline{\text{STRBD}}$ strobes, simplifying data channel use.

Input/output instructions result in similar operations to memory cycles. The device code with the rest of the I/O instruction is latched in the MAR instead of an address. The status lines (M, 00, 01) indicate an I/O cycle. Thus, the device code can be externally decoded and used to select a peripheral device in the same manner as memory decode.

The F9445 has a single interrupt request ($\overline{\text{INTREQ}}$) input. However, up to 16 levels of interrupt priority can be accomplished by externally daisy-chaining the requests.

In conventional minicomputers, machine registers are available for examination at any time via a front panel. This is a convenient aid for program and system debug. Microcomputers have not provided this facility, because so many signals from the chip are required. Generally, a firmware solution called a "monitor" has been employed. Such solutions provide all the facilities of a console, but rely on the system being fully operational in order to function. Since operational systems are not always the case, the firmware approach is limited as a debugging aid. In contrast, the F9445 features a microprogrammed console which does not rely on a fully functional memory to operate. To avoid the pin limitation problem, just one dedicated input ($\overline{\text{CONREQ}}$) is used to provide a "console code" from the bus, much the same as instructions are handled. Depending on the code provided, the F9445 either outputs register contents to the bus, inputs data into a register, or reads or writes into memory. One of the console codes initiates a "self-test" within the processor.

By combining CONTINUE and Halt operations, the F9445 can be single-stepped. The APL code is used to start execution from 077777 (in octal, 32K mode). This code is the default code on the bus if no console code is present. Thus the $\overline{\text{CONREQ}}$ pin can be used to initiate execution of a program in PROM, at the end of memory, without a console.

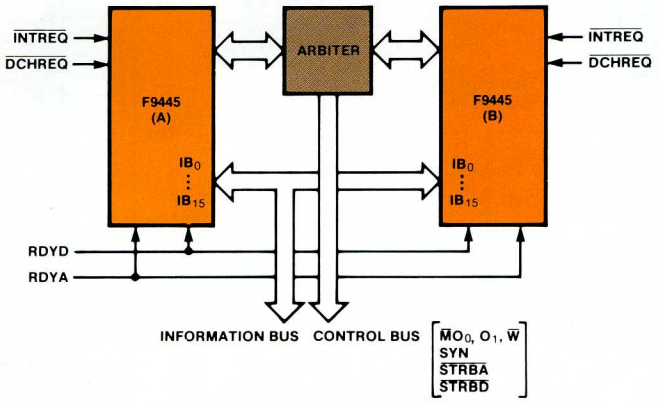


Fig. 7 Multiprocessor Configuration (Bus Sharing)

F9445 Support Circuits

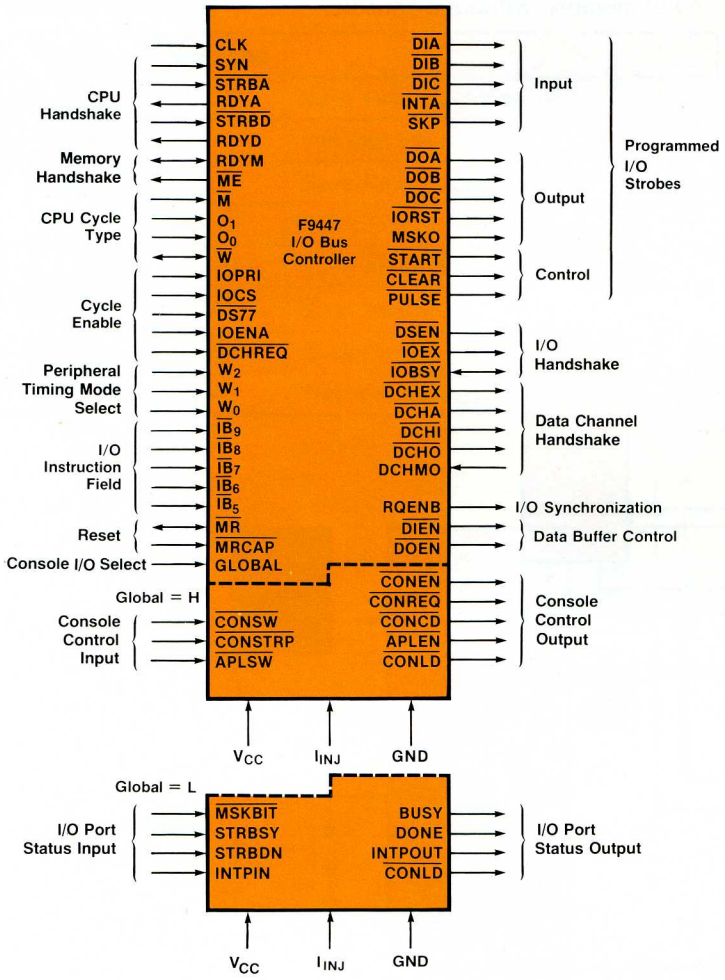
The F9445 family includes a range of support circuits—I/O extension, memory interface and peripheral support—for extending the system capability of the F9445 microprocessor. Except for the F9470 console controller, the circuits are fabricated using I^3L technology and are packaged in compact, 50 mil lead center, 64-pin DIPs.

Extended I/O Capability

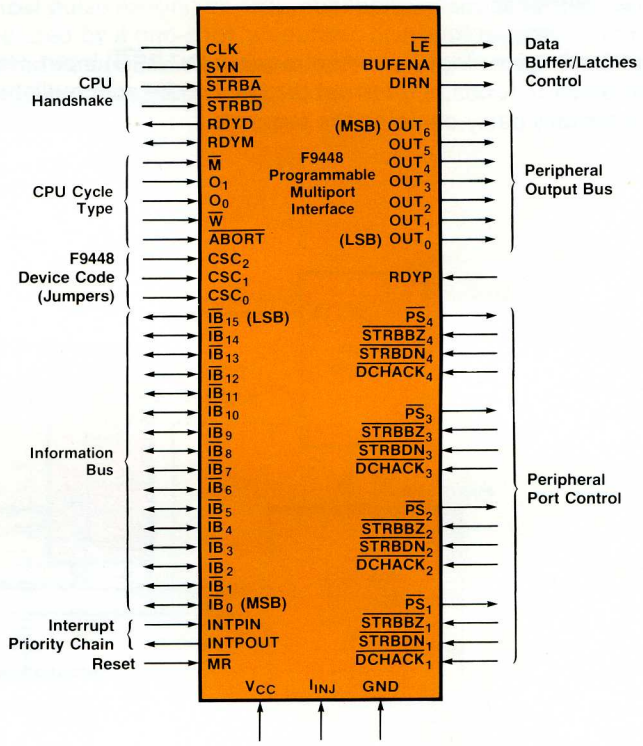
The first group, designed for extending the F9445 I/O capability, externally decode the F9445 I/O instructions and support various I/O bus schemes. The F9447 I/O bus controller is a decoder and timing generator for programmed I/O instructions as well as a timing generator for data channel transfers (*Figure 8*). When interfaced with the industry-standard NOVA-compatible[3] bus, it can control up to 62 peripherals—disk controllers, serial I/O, parallel I/O etc.—with full data channel and interrupt control. In addition, the F9447 includes console-control or a local-mode option.

3. NOVA is a registered trademark of Data General Corporation.

The F9448 programmable multiport interface performs the selecting and handshaking tasks between the F9445 and industry-standard peripherals designed for use with the 6800, 8080, 16000 busses. It offers a number of software-configured timing and control options. The main timing options include 8080/Z80/8086/16000/Z8000, 6800/68000, and the F9445. When the F9448 is in the F9445 timing mode, the F9445 proceeds at maximum speed; when in the other timing modes, wait states are inserted to accommodate the slower circuits. The F9448 decodes the F9445 I/O instructions, performs device decoding and interrupt control, and sets the busy and done flags. It has four independent I/O ports for connecting peripherals with mixed timing requirements and needs no additional decoding. Up to 62 peripherals can be supported using external device address decoding. *Figure 9* shows how 6800-type UARTS can be directly connected to the F9448.



I/O Bus Controller



Programmable Multiport Interface

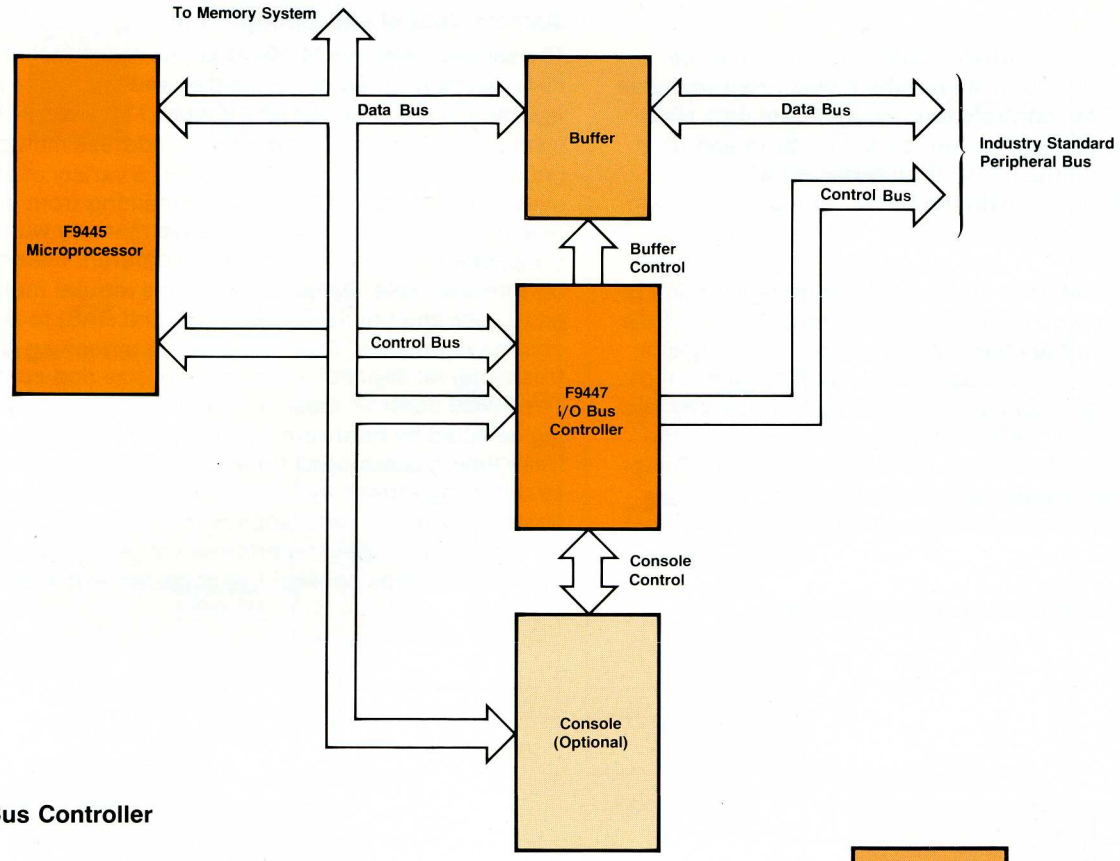


Fig. 8 Data Bus Controller

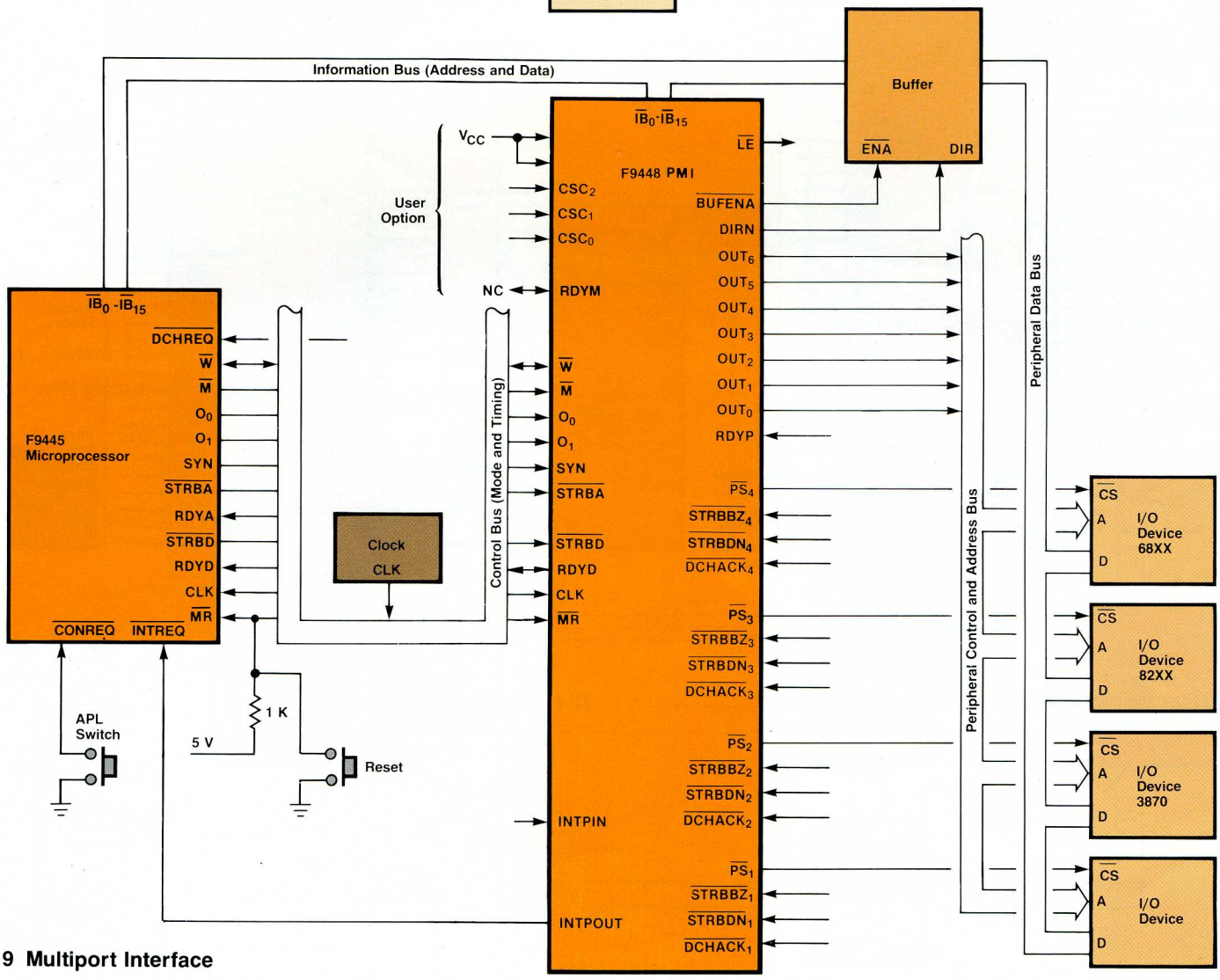


Fig. 9 Multiport Interface

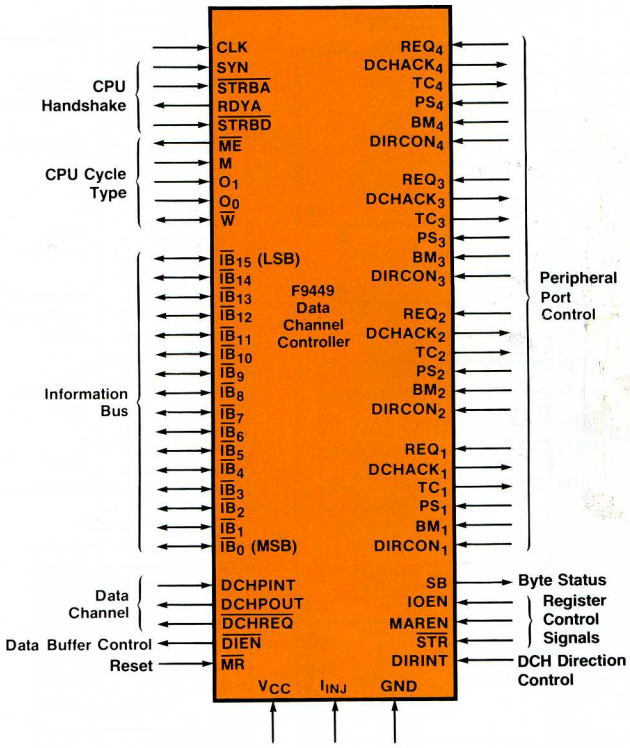
The F9449 data-channel controller, when coupled with the F9445 and the F9447 I/O controller or the F9448 programmable multiport interface PMI, controls four independent data channels so that peripherals can transfer data directly to and from memory. It is used with the F9447 for interfacing with the NOVA-compatible[4] bus and with the F9448 for interfacing with 6800, 8080 or 16000 busses (*Figure 10*).

Each channel has an address and a word-count register and is capable of transferring up to 32,768 8-bit bytes or 16-bit words, depending on whether the channel is operating in the byte or word mode. The four channels have fixed priorities and up to four F9449s can be cascaded, providing automatic priority allocation among the 16 channels. The data rate depends on the I/O timing—typically 4 Mbytes/second—selected in the I/O support circuit, which generates the necessary signals to enable the 3-state buffer or to strobe the data into the peripheral from the information bus.

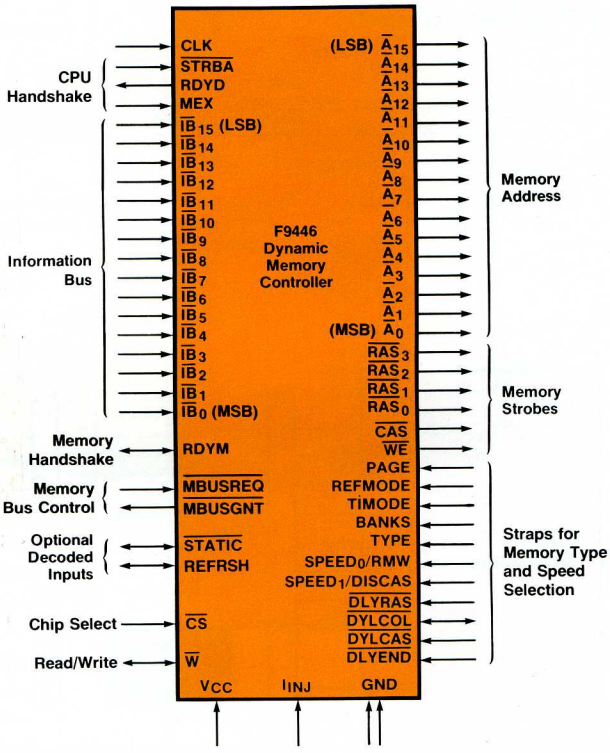
4. NOVA is a registered trademark of Data General Corporation.

Memory Control and Management

The second category of F9445 support circuits includes the F9446 dynamic memory controller and the F9444 memory management and protection unit (*Figure 11*). The F9446 dynamic memory controller incorporates an address multiplexer and memory timing generator to support a variety of static and dynamic memories, 16K or 64K containing from one to four memory blocks. The use of dynamic memory with a microprocessor is complicated by many different memory-speed options available. Dynamic memories require multiplexing of addresses and timing signals ($\overline{CAS}$ and $\overline{RAS}$) to select row and column addresses. They also require refreshing with the refresh interval depending on memory size and configuration. The F9446 controls these functions, with memory size and timing selected by hardware strapping on the package pins. Refresh timing is arranged for least contention with F9445 timing by deferring refresh cycles until a non-memory process cycle is detected. Thus, in most applications, memory refresh imposes no overhead on system performance. An optional page-mode operation speeds consecutive accesses within the same memory page.



Data Channel Controller



Dynamic Memory Controller

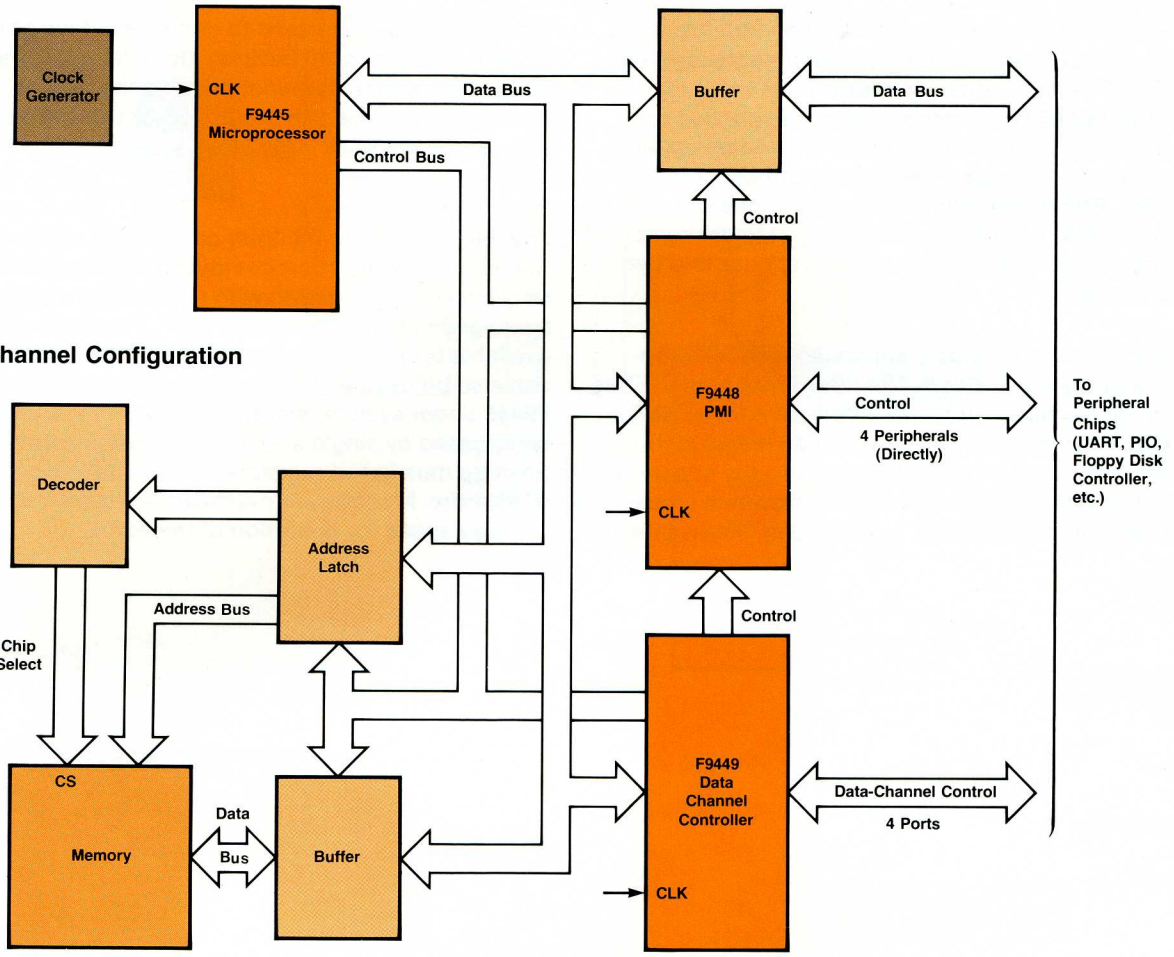


Fig. 10 Data Channel Configuration

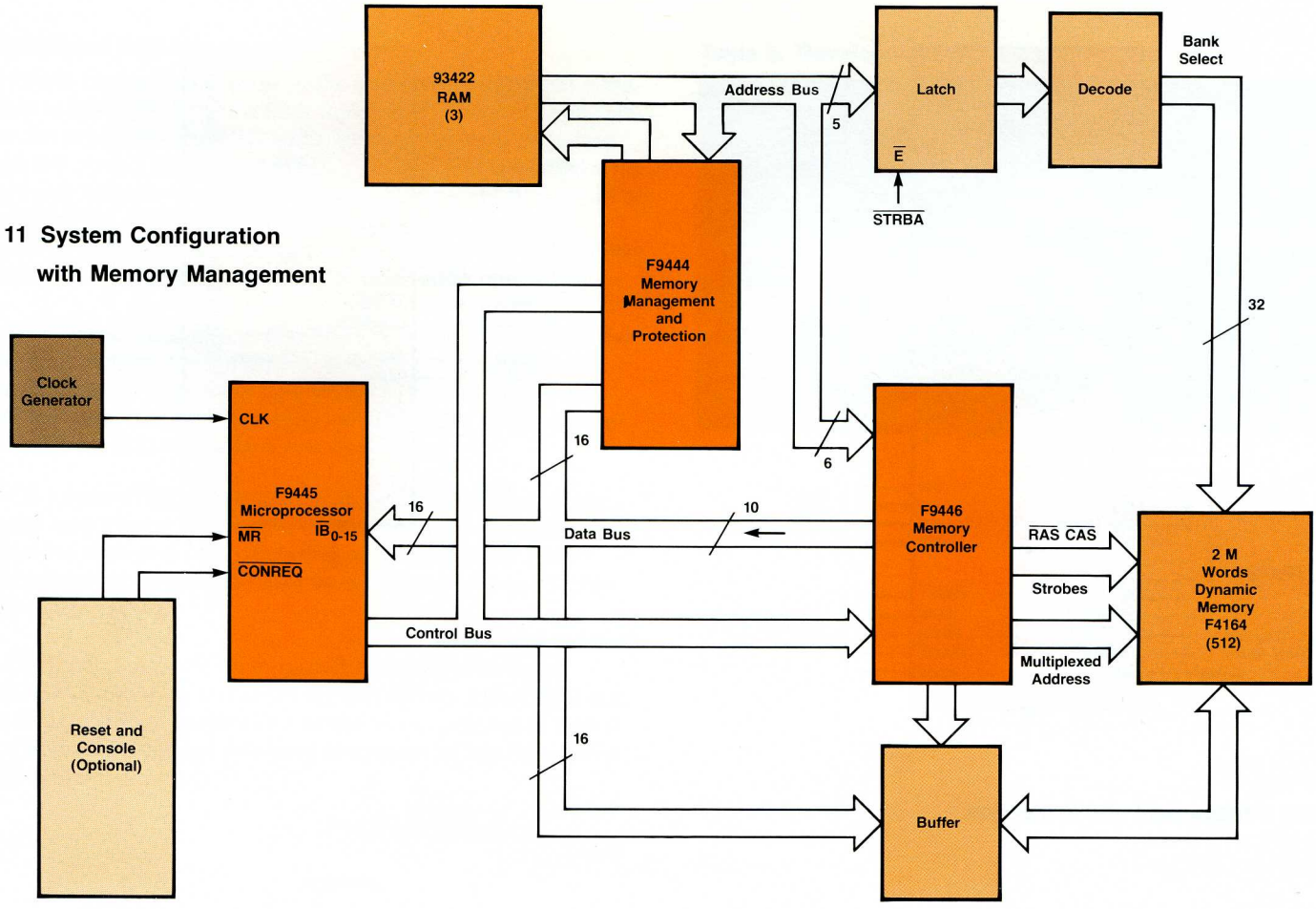
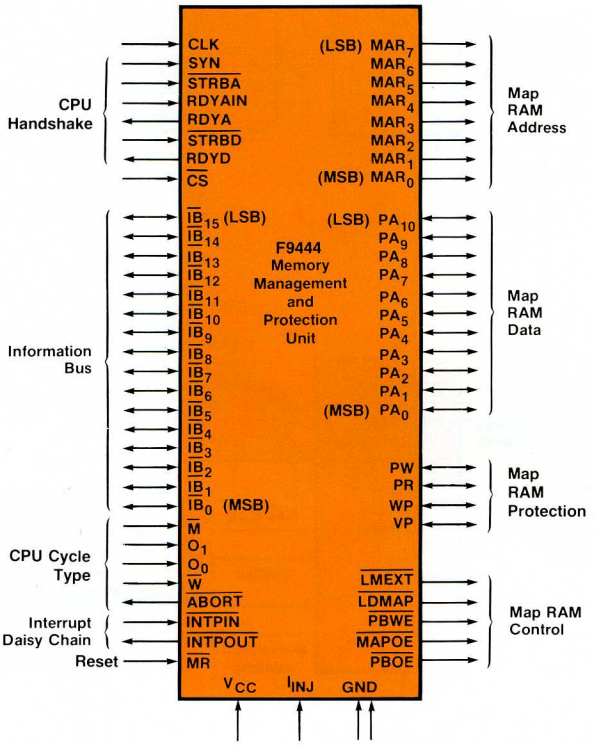


Fig. 11 System Configuration with Memory Management

The F9444 memory management and protection unit has three functions. First, it provides a method of extending the address of the F9445. The F9445 can directly address 128 bytes of memory; with the F9444 the memory space is expanded to 4 Mbytes via a look-up table (map). The top six address bits are used to select one of 64 registers, with each register containing an 11-bit address. Effectively, this provides 64 pages with 1K words per page that can be anywhere in a 4-Mbyte memory space. Four maps are provided, two for program use and two for data channel use.

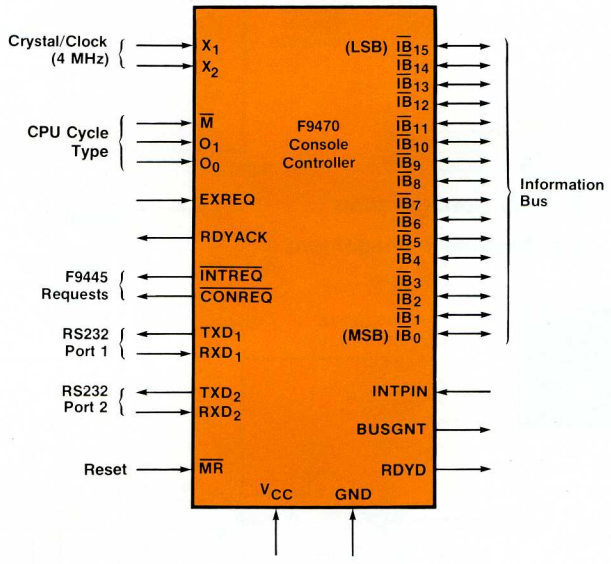
The second function is to provide a supervisor/user environment for operating system support. The F9444 monitors the bus to detect illegal instructions—e.g., I/O, interrupt control, status control, and HALT—in the user mode. When an illegal instruction is detected, there is an immediate branch to the supervisor mode. The third capability of the F9444 is to provide a virtual-memory environment for the F9445. In this mode, status bits in the look-up table are used to determine if the page requested is in main memory, or in backing store. If in backing store, the F9445 operation is terminated using the ABORT signal. The virtual memory size is 64K multiplied by the number of user maps and can be made as large as necessary.



Memory Management and Protection

Peripheral Support

The F9470 falls into the third category of support chips. It is an NMOS single-chip microcomputer, preprogrammed to provide easy access to the F9445 with no external device decoding, packaged in a standard 40-pin DIP. Two serial I/O ports are available for use with a standard RS232 interface. The F9470 can also be used as a virtual console controller to debug the F9445 under system operation (*Figure 12*). This is accomplished by single stepping the F9445, examining and depositing data in the registers, or by examining and loading data in memory. No software overhead is required in the F9445 memory space for operation of the F9470.



Console Controller

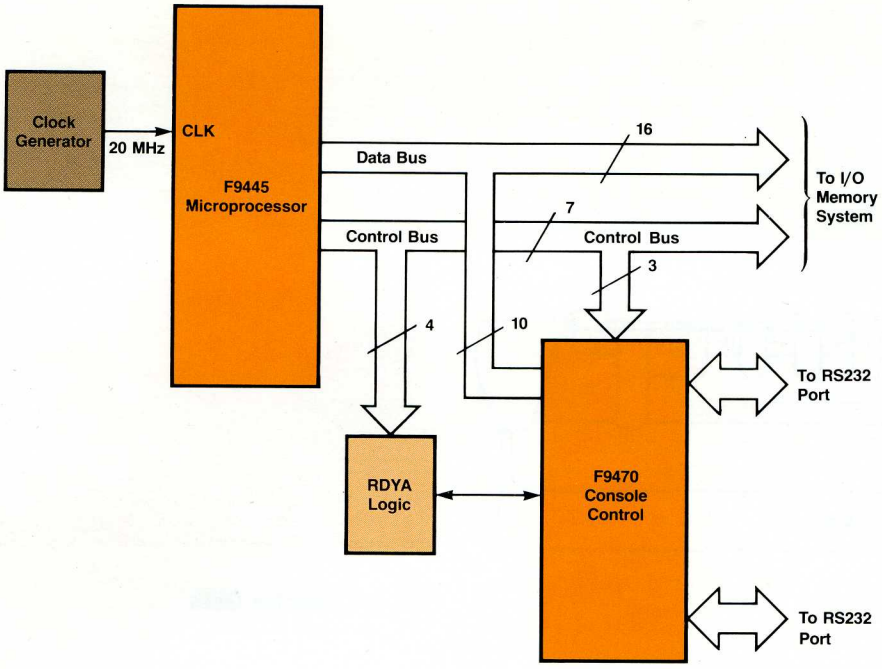


Fig. 12 Console Control Subsystem

Development Systems

The F9445 is supported by versatile design aids (*Table 5*). One system is the PEP 45, a low cost, entry-level, single-board computer for programming evaluation and prototyping requiring only a 5 V power supply and a CRT. This unit can be used with or without firmware, because console commands which can be input from the CRT are included. In addition, a powerful monitor program (PEPBUG) can be used. Also, the PEP 45 can be expanded via a compatible "multibus" edge connector.

Another support system is the FS-1, a complete Winchester Drive disk development system. It has a comprehensive Interactive Multi-user Disk Operating System (IMDOS) that supports many utility programs, including EDIT, a character based editor, and MACRO, a full assembler.

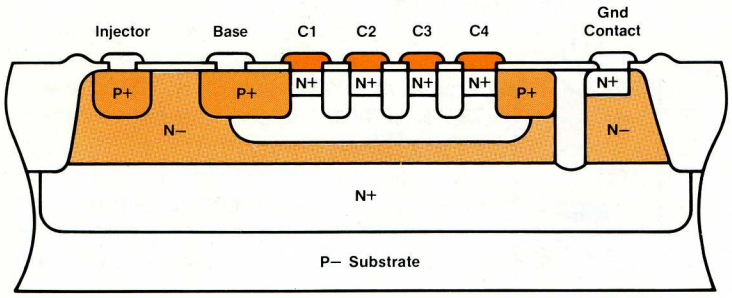
The FS-1 system supports three high-level languages: Basic, Fortran, and Pascal. These standard languages considerably reduce the time and cost of writing software for the F9445. Code produced by the FS-1 can be downloaded into PROMs using utility software.

The FS-1 also supports an in-circuit emulator and tracer (EMUTRAC) allowing complete system debug. EMUTRAC consists of a hardware connection to the target processor board and a software package allowing emulation of the F9445 in a target system.

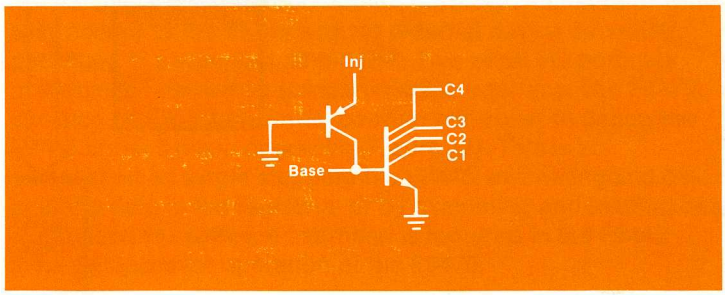
Table 5. Development Systems

System	Software
PEP45	PEPBUG DIAGNOSTICS PEPBASIC
FS-1	PEPBUG IMDOS EDIT UTILITIES BASIC FORTRAN PASCAL

I^3L Logic



I^3L Device Structure



I^3L 4-Collector Gate

Integrated Injection Logic

Integrated injection logic I^2L offers several outstanding advantages as a VLSI technology. These are high packing density, bipolar-compatible processing, low-power and low-voltage operation, low power/delay product, and higher speed than MOS (see Table). Since I^2L is an extension of bipolar technology, other types of conventional bipolar devices such as buffer stages and peripheral circuitry can be placed on the same chip.

While most logic families are built around multiple-input, single-output gates, I^2L is based on a simple single-input, multiple-output inverting gate (see Figure). The vertical npn transistor with multiple collectors operates as an inverter and logic is implemented by wiring the collectors. A lateral pnp, instead of a resistor, serves as both a current source and a load. Most of the gate terminals share the same semiconductor region, e.g., the collector of the lateral pnp transistor is the same p+ region as the base of the vertical npn device; the emitter of the npn is the same n– epitaxial region as the base of the pnp.

Since the individual gates do not require separate isolation regions, they can be placed in a common n-type tub to further enhance chip density. Because of common transistor elements and because no resistors are required, the entire I^2L gate only occupies the space of a single multi-emitter transistor.

Isoplanar Integrated Injection Logic I^3L™

The I^3L process, a direct extension of the Isoplanar bipolar technology, utilizes local-oxide-isolation and ion-implantation techniques to reduce transistor size and improve performance over conventional I^2L. It offers the following advantages.

The I^3L process is very conducive to scaling, i.e., reducing device geometries both vertically and horizontally by changing the design rules. This in turn leads to higher chip density, smaller dice and improved speed. Scaling should not be confused with shrinking, which refers to optically reducing all mask dimensions by a factor that also reduces the die size.

Improved speed due to the shallow structure, which enhances transit time and reduces junction capacitance: the thin n- epitaxial layer below the extrinsic npn base that results in less charge storage in this region, and the p+ diffusion used for the injector and the low-resistance extrinsic base areas.

Improved dc gain and optimum power/delay product due to the oxide collar around the I^3L structures that limits extraneous hole injection and decreases capacitance.

Accurate control of the pnp and npn device characteristics with minimum device interdependence through use of ion implantation and self-aligned device structure.

Compatibility with high-speed low-power Schottky-TTL interface circuits.

Enhanced radiation tolerance — 1×10^5 Rad total dose.

High speed performance over complete military temperature range.

Comparison of LSI Technology Performance

Technology	I^3L	CMOS	NMOS	ECL	LS
Operating Voltage	1	2	3	5	4
Power	2	1	3	5	4
Speed	2	4	5	1	3
Density	1	3	2	5	4
Output Drive Capability	3	4	5	2	1

Note: 1 is best.