Pascal CHOUR

Z8108

Zilog

Advance Information Bulletin

October 1982

Zilog

Zilog

Zilog

Zilog

A 2 M
6, AVENUE CH. DE GAULLE, HALL A
78150 LE CHESNAY
TÉL. 954.91.19

Zilog

Information Advance Bulletin

October 1982

FEATURES

- Object code compatible with Z80, plus Multiply, Divide, and other instructions, and new addressing modes.
- 512K byte memory address space with on-chip paged memory management.
- 6-25 MHz CPU.
- On-chip clock oscillator with programmable scaler for bus timing.
- Z80-Bus compatible.
- Supports the three Z80 interrupt modes, plus a new fourth mode that allows trap handling and interrupt nesting.
- Breakpoint and single step implemented as traps to facilitate debugging.
- System and User modes.
- Optional separation of program and data.
- On-chip refresh with 10-bit addressing.

GENERAL DESCRIPTION

The Z8108 is a 40-pin microprocessor that operates on the Z80 bus. It incorporates advanced architectural features for greater throughput and increased memory addressing, while maintaining Z80 object code compatibility. These new features provide a continuing growth path for present Z80-based designs and a high-performance microprocessor for future designs.

The Z8108 is an enhanced version of the Z80 Central Processing Unit (CPU). Advanced features such as dual mode (System/User) operation and a sophisticated interrupt and trap handling mechanism are supported by the CPU architecture. The Z80 instruction set has been retained to provide binary code compatibility with the present Z80. The augmented instruction set provides enhancements and extensions to many Z80 instructions. Four new addressing modes have been added to increase the flexibility and power of the instruction set. The Z8108 also contains an on-chip clock oscillator and a refresh controller that provides 10-bit refresh addresses for dynamic memories.

One important requirement for today's microprocessor-based system design is to increase memory address space beyond 64K bytes for 8-bit machines. The Z8108 provides this capability with its on-chip Memory Management Unit (MMU). The MMU dynamically translates memory addresses to allow the microprocessor to address 512K bytes. In addition to enabling an expanded address space, the MMU performs other memory management functions previously handled by dedicated off-chip memory management devices.

CPU

The following sections describe the Z8108 System/User mode, data types, address spaces, and register sets.

System/User Mode

The Z8108 microprocessor can run in either System or User mode. In System mode, all of the instructions can be executed and all of the CPU registers

can be accessed. System mode is intended for use by programs that perform operating system functions. In User mode, some instructions cannot be executed and some registers of the CPU are inaccessible. In general, User mode of operation is intended for use with application programs.

This separation of CPU resources promotes the integrity of the system. Programs operating in User mode cannot access those aspects of the CPU that deal with system interface events. To further support the System and User modes, there are two Stack Pointers--one each for both System and User mode stacks. To ensure that the user stack is free of system information, the information saved on the occurrence of interrupts or traps is always pushed onto the System stack before the new program status is loaded into the Program Counter and Master Status register.

Data Types

The CPU of the Z8108 microprocessor can operate on bits, BCD digits (4 bits), bytes (8 bits), words (16 bits), and byte strings. Bits in registers or memory can be set, cleared, and tested. BCD digits, packed two to the byte, can be manipulated with the Decimal Adjust Accumulator and Rotate Digit instructions. Bytes are operated on by 8-bit load, arithmetic, logical, and shift-and-rotate instructions. Words are operated on by the 16-bit load and 16-bit arithmetic instructions. Block move and search operations can manipulate byte strings up to 64K bytes long.

Address Spaces

The Z8108 microprocessor supports four separate address spaces. These address spaces correspond to the different kinds of locations that can be addressed, the method by which the logical addresses are formed, and the translation mechanism used to map the logical address into physical locations. The four address spaces are discussed below.

CPU Registers. This address space consists of the CPU registers available to the programmer that are used for data or address manipulation. These registers include F, A, B, C, D, E, H, L, F', A', B', C', D', E', H', L', IX, IY, SSP, USP, PC, I, and R.

CPU Status and Control Registers. This address space consists of all the CPU control and status registers. The CPU status and control registers are the Master Status register, Bus Timing and Initialization register, Bus Timing and Control register, System Stack Limit register, Trap Control register, Interrupt/Trap Vector Table Pointer register, and I/O Page register.

Memory. Two memory address spaces, one each for both System and User mode of operation, are supported. The appropriate memory space is selected by the System/User bit in the Master Status register. Only memory addresses are translated by the memory management mechanism.

I/O. The I/O address space is accessed by I/O instructions only.

REGISTER SET

The Z8108 microprocessor uses a superset of the present Z80 CPU registers. The additional registers of the Z8108 are: User Stack Pointer register, Master Status register, Bus Timing and Initialization register, Bus Timing and Control register, System Stack Limit register, Trap Control register, Interrupt/Trap Vector Table Pointer register, and I/O Page register. A short description of the CPU registers follows.

Primary and Alternate Register Files. The Primary and Alternate [designated by ' (prime)] register files each contain an 8-bit accumulator (A), an 8-bit Flag register (F), and six 8-bit general-purpose registers (B, C, D, E, H, L). The Primary register file is normally used as the working file, but the Exchange instructions allow the Alternate file to be substituted for it. The accumulator is the destination register for 8-bit arithmetic and logical instructions and the Flag register contains status information. The HL register pair is used as a 16-bit accumulator. The six general-purpose registers can be paired (BC, DE, HL) to form three 16-bit general-purpose registers.

Index Registers. The two index registers, IX and IY, each hold a 16-bit base address that is used in indexed addressing modes. They each can also be used as two 8-bit general-purpose registers.

Stack Pointers. Two hardware Stack Pointers, the System Stack Pointer (SSP) and the User Stack Pointer (USP), support the two modes of operation

of the microprocessor and serve as a base for relative addresses. The System Stack Pointer is used for saving information when an interrupt or trap occurs and for supporting subroutine calls and returns in System mode. The User Stack Pointer is used for supporting subroutine calls and returns in User mode.

Program Counter. The Program Counter is used to step through instructions in the currently executing program and as a base for relative addresses.

Interrupt Register. The Interrupt register (I) is used in Interrupt Mode 2 to generate the 16-bit logical address of an interrupt service routine. The Interrupt register supplies the upper eight bits of the indirect address and the interrupting peripheral supplies the lower eight bits.

R Register. The R register can be used as a general-purpose 8-bit read/write register. The R register is not associated with the refresh address, and its contents are changed only by a system mode program.

Master Status Register. The Master Status Register is a 16-bit register that contains status about the currently-executing program.

Bus Timing and Initialization Register. The Bus Timing and Initialization register is used to specify the scaling of the CPU clock for bus transactions and the control of automatic memory Wait state generation. The CPU clock can be scaled by 1, 2, or 4 to achieve reduced system clock rates. Zero, one, two, or three Wait states can be automatically inserted in each memory transaction.

Bus Timing and Control Register. The Bus Timing and Control register is used to specify the timing of I/O and interrupt transactions, and whether interrupts are to be vectored or not. Zero, one, two, or three additional Wait states may be inserted in each I/O transaction. Similarly zero, one, two, or three Wait states may be inserted during interrupt acknowledge to allow additional time for the interrupt acknowledge daisy chain to settle.

System Stack Limit Register. The System Stack Limit register is used to set the boundary conditions for a System Stack Overflow Warning trap.

Trap Control Register. The Trap Control register enables the maskable traps.

Interrupt/Trap Vector Table Pointer Register. The Interrupt/Trap Vector Table Pointer is a 16-bit register used to point to the start of the Interrupt/Trap Vector Table.

I/O Page Register. The I/O Page Register is an 8-bit register that is used to expand the I/O address space. During an I/O operation, the contents of the I/O Page register are appended to the normal 16 bits that are output during a Z80 I/O operation, enabling groups of I/O locations to be accessed.

INTERRUPT AND TRAP STRUCTURE

The following sections describe interrupts and traps, interrupt modes, and trap conditions.

Interrupts and Traps

The Z8108 contains a flexible and powerful interrupt and trap structure. Interrupts are external asynchronous events requiring CPU attention and are generally triggered by peripherals needing service. Traps are synchronous events that are the response by the CPU to certain events detected during the attempted execution of an instruction. Both interrupts and traps are processed in a similar manner by the CPU.

Two types of interrupt are supported, nonmaskable and maskable. The nonmaskable interrupt cannot be disabled by software and is typically used for high-priority events that require immediate attention. The maskable interrupt can be disabled or masked by clearing the appropriate bit in the Master Status register. The maskable interrupt can be programmed to be either vectored or nonvectored in Interrupt Mode 3.

Interrupt Modes

The four Z8108 interrupt modes are described below.

Mode 0, 1, and 2

When operating in Interrupt Modes 0, 1, or 2, the CPU responds to external interrupts in a manner similar to that of the Z80 microprocessor. All nonmaskable interrupts are serviced by jumping to

restart location 0066H. Maskable interrupts are serviced differently, depending on which interrupt mode is active. When in Mode 0, the CPU executes an instruction placed on the data bus by the interrupting device. In Mode 1 operation the CPU restarts at location 0038H. In Mode 2, the CPU uses a vector placed on the bus by the interrupting device combined with the value of the I register to access a table of addresses for interrupt service routines.

Mode 3

Mode 3 is a new interrupt service mode intended to improve interrupt and trap handling. In Interrupt Mode 3, upon receiving an interrupt or trap, the current program status (the Program Counter value and contents of the Master Status register) is pushed on the stack and a new program status is loaded from memory. The saving of the old Master Status register allows for complete nesting of interrupts, since the state of the previous interrupt enable is saved on the stack. For traps and nonvectored interrupts, the new Program Counter and Master Status values are loaded from pre-assigned locations in the Interrupt/Trap Vector Table. If a vectored maskable interrupt is received in Mode 3, a vector is used to index into the Interrupt/Trap Vector table to fetch a new Program Counter value. The new Master Status register is loaded from a pre-assigned location in the table.

Trap Conditions

The Z8108 allows for the efficient processing of traps. The Z8108 supports the following traps:

Privileged Instruction Trap. This trap occurs whenever the CPU attempts to execute a privileged instruction while in User mode.

System Call Trap. This trap occurs whenever a System Call (SC) instruction is executed. It allows for an orderly transition to be made from User to System mode.

Access Violation Trap. This trap occurs whenever a logical address to be translated by the MMU uses a Page Descriptor register with the Valid bit cleared to 0, or the Write Protect bit set to 1.

System Stack Overflow Warning Trap. This trap occurs whenever a Push performed on the system stack causes the stack to pass the boundary set in the System Stack Limit register. This trap can be disabled by clearing the stack overflow warning trap enable flag.

Division Exception Trap. This trap occurs whenever the division instructions are executed with either a divisor of zero or when an overflow occurs in the quotient.

Single Step Trap. This trap occurs whenever the Single Step Pending control bit of the Master Status register is set. This trap allows programs to be reliably stepped one instruction at a time.

Breakpoint on Halt Trap. This trap occurs whenever a Halt instruction is encountered while the Breakpoint on Halt control bit of the Master Status register is set. This trap provides a breakpoint facility that is useful for debugging.

ADDRESSING MODES

Addressing modes are used by the CPU to calculate the effective address of an operand needed for execution of an instruction. Nine addressing modes are supported by the Z8108. Of the nine addressing modes, four modes are new to the Z8108: Indexed (with 16-bit displacement), Stack Pointer Relative, Program Counter Relative, and Base Indexed. The remaining modes are the same as the Z80 addressing modes. The Z8108 addressing modes are summarized in Figure 1.

Mode	Operand Addressing			Operand Value
	In the Instruction	In a Register	In Memory or I/O	
Register		REGISTER ADDRESS → OPERAND		The contents of the register
Immediate	OPERAND			In the instruction
Register Indirect		REGISTER ADDRESS → ADDRESS → OPERAND		The contents of the location whose address is in the register
Direct Address			ADDRESS → OPERAND	The contents of the location whose address is in the instruction
*Index			REGISTER ADDRESS + BASE ADDRESS + INDEX → OPERAND	The contents of the location whose address is the address in the instruction, offset by the contents of the register
Short Index			REGISTER ADDRESS + DISPLACEMENT + ADDRESS → OPERAND	The contents of the location whose address is in the register, offset by the signed displacement in the instruction
*Relative			DISPLACEMENT + PC VALUE → OPERAND	The contents of the location whose address is the contents of the program counter, offset by the displacement in the instruction
*Stack Pointer Relative			DISPLACEMENT + SP VALUE → OPERAND	The contents of the location whose address is the contents of the stack pointer, offset by the displacement in the instruction
*Base Index			REGISTER ADDRESS 1 → ADDRESS; REGISTER ADDRESS 2 → DISPLACEMENT; + → OPERAND	The contents of the location whose address is the contents of a register, offset by the displacement in a register

* = New addressing modes on the Z8108

Figure 1. Addressing Modes

Register. The effective address of the operand is one of the 8-bit registers (A, B, C, D, E, H, L, IX--high or low byte, or IY--high or low byte) or one of the 16-bit registers (BC, DE, HL, IX, IY, or SP) or one of the special byte registers (I or R).

Immediate. The operand is in the instruction itself and has no effective address.

Register Indirect. The contents of a register specify the effective address of the operand. For memory accesses, the HL register is most often used. The C register is used for I/O and control register space accesses.

Direct Address. The operand is in the location whose address is contained in the instruction. Depending on the instruction, the specified operand will be either in I/O or memory space.

Index. The effective address of the operand is specified by adding the 16-bit address contained in the instruction to a two's complement index contained in the HL, IX, or IY register.

Short Index. The effective address of the operand is computed by adding the 8-bit two's complement signed displacement contained in the instruction to the contents of the IX or IY register. This addressing mode is equivalent to the Z80 indexed mode.

Relative. An 8- or 16-bit displacement contained in the instruction is added to the Program Counter to generate the effective address of the operand.

Stack Pointer Relative. The effective address of the operand is computed by adding a 16-bit two's complement displacement contained in the instruction to the contents of the Stack Pointer.

Base Index. The effective address of the operand is the location whose address is computed by adding the contents of either HL, IX, or IY to the contents of another of these three registers.

INSTRUCTION SET GROUPINGS

The Z8108 operates with an instruction set that is divided into ten basic groups. These groups are:

8-bit Load. Instructions in this group load an 8-bit byte into a register or memory location.

16-bit Load and Exchange. Instructions in this group load two bytes of data into memory or registers.

Block Transfer and Search. Instructions in this group support block transfer and string search functions on byte strings up to 64K bytes long.

8-bit Arithmetic and Logic. Instructions in this group perform arithmetic and logical byte operations. New instructions include Multiply and Divide.

16-bit Arithmetic and Logic. Instructions in this group perform 16-bit arithmetic operations. New instructions include Compare, Multiply, and Divide.

Rotate, Shift, and Bit-Manipulation. Instructions in this group test, set, and reset bits within bytes, rotate and shift byte data one bit position, and rotate BCD digits left and right.

Program Control. Instructions in this group affect the Program Counter (PC) and as a result control program flow. The System Call instruction provides controlled entry into System mode from User mode programs.

I/O. Instructions in this group transfer bytes, or strings of bytes between peripheral devices and the CPU registers or memory.

CPU Control. Instructions in this group affect the CPU control and status registers, as well as other miscellaneous instructions.

MEMORY MANAGEMENT UNIT (MMU)

The Z8108 microprocessor contains an on-chip MMU. The MMU allows access to more than 64K bytes of physical memory (the Z8108 can access up to 512K bytes) and provides memory protection features typical of those found on large systems. Access violations such as invalid mapping and writes to write-protected pages are detected by the MMU, which initiates a trap sequence to signal the error. When first powered up the processor looks like a Z80 because the MMU features are not enabled. It is only after the MMU is enabled through software that the memory addressing is expanded.

The physical address space is expanded by dividing the 64K byte logical address space (the space manipulated by the program) into pages. These pages are then mapped into the larger physical address space of the Z8108. The actual size of the page depends on whether program/data separation is enabled or not. If program/data separation mode is enabled, each page is 8K bytes in length. If not, the page length is reduced to 4K bytes. Using this page mapping technique, 16-bit logical addresses are translated into 19-bit physical addresses in the Z8108. Address translation can occur both in system and in user mode, with separate translation facilities used for each mode. The MMU further provides for separating instruction references from data references, enabling programs of up to 64K bytes to manipulate up to 64K bytes of data without operating system intervention.

The on-chip mapping mechanism consists of two sets of 16 Page Descriptor registers--a set of 16 for both System and User mode. Associated with the Page Descriptor registers is a control register that governs the MMU activities. All MMU registers are accessed through the I/O space of the CPU. Each 16-bit Page Descriptor register consists of a 4-bit attribute field and a 12-bit page frame address field. Only seven bits of the page frame address field are used by the MMU. The bits in the attribute field are the Valid, Modified, and Write Protect bits, with one bit reserved for future use. The Valid bit indicates whether the descriptor contains valid information. An access violation will be generated if an attempt is made to use an invalid descriptor for translation. The Modified bit indicates whether or not the page frame has been written, and can be used by operating systems to determine whether or not a page needs to be written back to mass storage when changing pages. The Write Protect bit, when set, will cause an access violation to occur when an attempt is made to write into the associated page frame.

Address Translation Without Program/Data Separation

When program/data separation is not enabled, the 16-bit logical address is divided into two fields, a 4-bit index field used to select one of 16 Page Descriptor registers, and a 12-bit offset field which will form the lower 12 bits of the final physical address. The final physical address is composed of seven bits of the page frame address supplied by the selected Page Descriptor register and the 12-bit offset supplied by the logical address. This method of translation is illustrated in Figure 2.

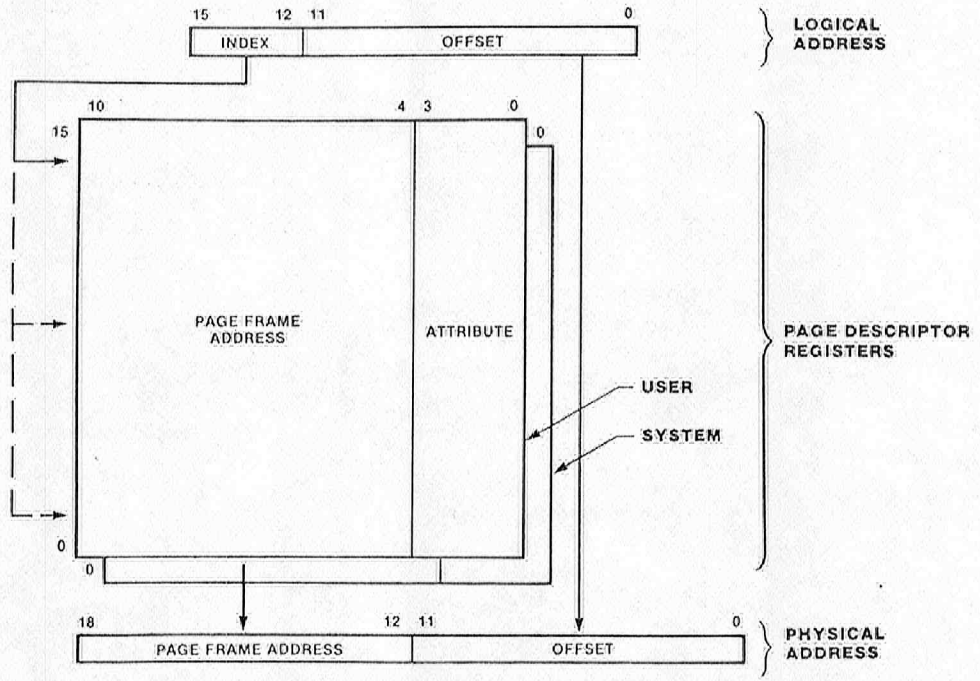

Figure 2. Address Translation Without Program/Data Separation

Address Translation with Program/Data Separation

Figure 3 illustrates address translation when program/data separation is enabled. The Page Descriptor register consists of a 12-bit page frame address field. When program/data separation is enabled, only six bits of the page frame address field are used by the MMU. The logical address is divided into a 3-bit index field and a 13-bit offset field. The physical address is the result of concatenating the six bits from the page frame address to the 13-bit logical offset. The Page Descriptor register is chosen by a 4-bit index field, which consists of the Program/Data Address status signal from the CPU and the three index bits from the logical address.

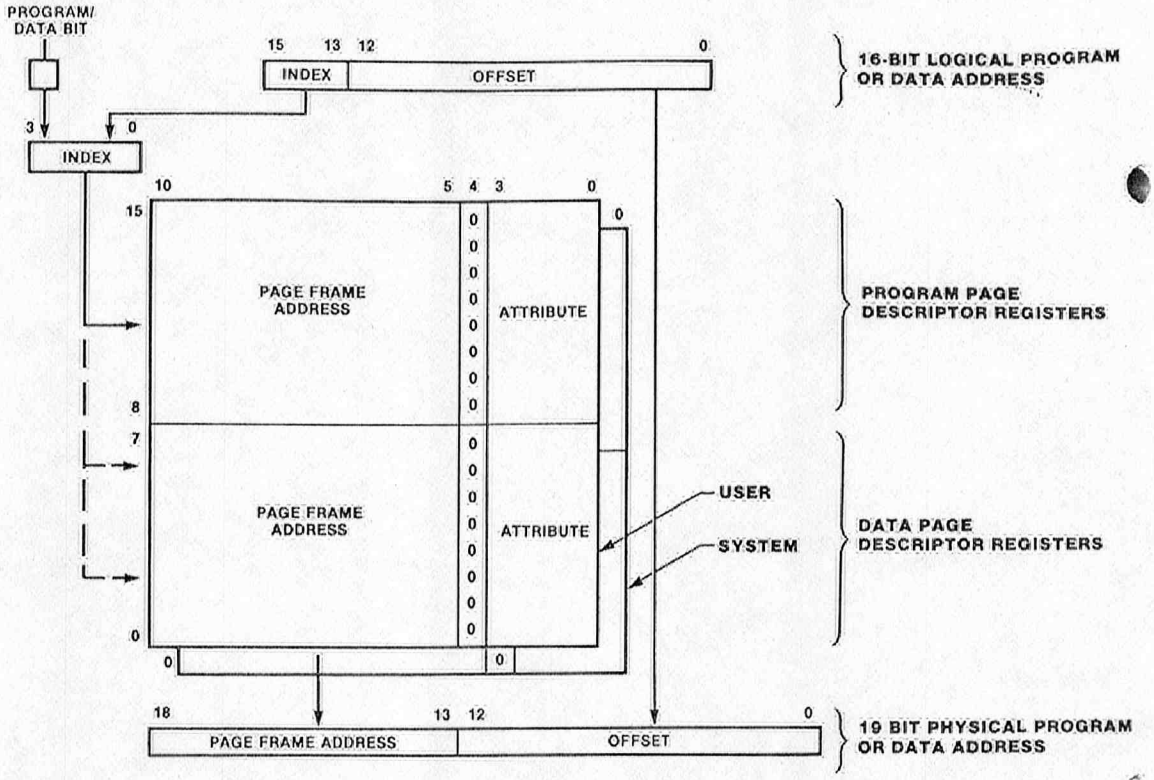

Figure 3. Address Translation with Program/Data Separation

ON-CHIP OSCILLATOR

The Z8108 has an on-chip oscillator/clock generator that can be connected to a crystal or other suitable clock source. The frequency of the processor clock is one-half that of the fundamental frequency of the crystal or the external clock source. This clock can then be further divided by 1, 2, or 4 before being output as the bus timing clock for use by the rest of the system.

REFRESH MECHANISM

The Z8108 contains an internal mechanism for refreshing dynamic memories. Enabling the refresh mechanism causes memory refresh to be performed periodically at a user-specified rate. Refresh transactions are identical to memory transactions except that no data is transferred. Each refresh transation generates a 10-bit refresh address enabling the Z8108 to interface with large dynamic memories.

The Z8108 functional pinout is shown in Figure 4.

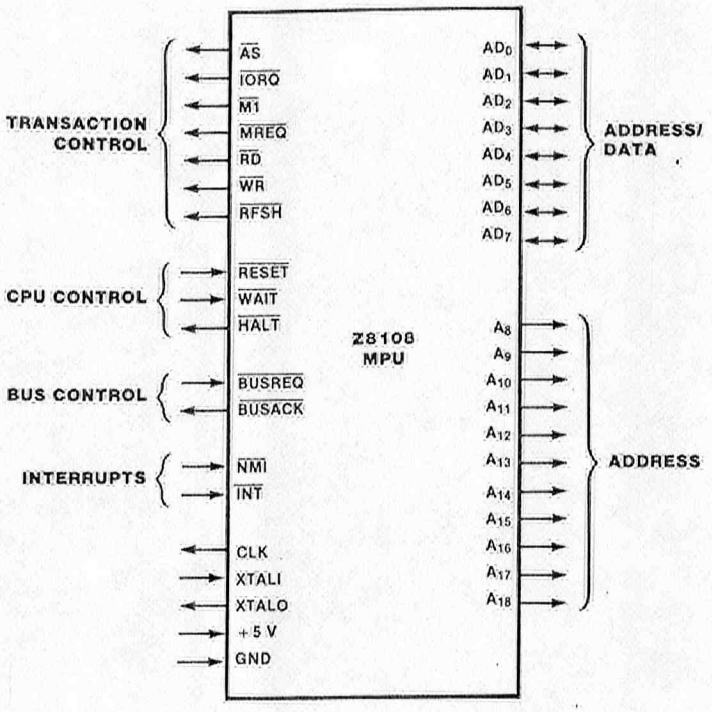

Figure 4. Z8108 Functional Pinout

Transaction Pins

These signals provide timing, control, and data transfer for bus transactions.

AD_0-AD_7. Address/Data (Bidirectional, active High, 3-state). These multiplexed address and data lines carry I/O addresses, memory addresses, and data during bus transactions.

A_8-A_{18}. Address (Output, active High, 3-state). These address lines carry I/O and memory addresses during bus transactions.

$\overline{AS}$. Address Strobe (output, active Low, 3-state). The rising edge of $\overline{AS}$ indicates the beginning of a transaction and shows that the address is valid. This signal can be used to gate the address on the AD_0-AD_7 lines into an external latch.

$\overline{IORQ}$. Input/Output Request (Output, active Low, 3-state). A Low on this line indicates that the address lines hold valid I/O addresses for an I/O read or write operation. An I/O request signal is also generated with an $\overline{M1}$ signal when an interrupt is being acknowledged. This indicates that an interrupt response vector must be placed on the bus.

$\overline{M1}$. Machine Cycle One (Output, active Low, 3-state). A Low on this line indicates that the current transaction is the op code fetch cycle of an RETI instruction execution. $\overline{M1}$ also occurs

... an I/O request to indicate an Interrupt Acknowledge cycle.

MREQ. Memory Request (Output, active Low). A Low on this line indicates that the address bus holds a valid address for a memory read or write operation.

RD. Read (Output, active Low, 3-state). A Low on this line indicates that data is being read from memory or an I/O device.

WR. Write (Output, active Low, 3-state). A Low on this line indicates that the bus holds valid data to be stored at the addressed memory locations.

RFSH. Refresh (Output, active Low). A Low on this line indicates that the lower ten bits of the address bus contain a refresh address for dynamic memories.

CPU Control Pins

These pins carry signals that control the overall operation of the CPU.

RESET. Reset (Input, active Low). A Low on this line resets the CPU.

WAIT. Wait (Input, active Low). A Low on this line indicates that the addressed memory or I/O devices need more time to complete a transaction. Wait states are inserted as long as the signal is active.

HALT. Halt (Output, active Low, 3-state). A Low on this line indicates that the CPU has executed a Halt instruction and is waiting for an interrupt before operation can resume.

Bus Control Pins

These pins carry signals for requesting and obtaining control of the bus from the CPU.

BUSREQ. Bus Request (Input, active Low) A Low on this line indicates that a bus requester has obtained or is trying to obtain control of the bus.

BUSACK. Bus Acknowledge (Output, active Low). A Low on this line indicates that the CPU has relinquished control of the bus in response to a bus request.

Interrupt Pins

These pins carry signals that convey interrupt requests to the CPU.

NMI. Nonmaskable Interrupt (Input, Edge activated). A High-to-Low transition on NMI requests a nonmaskable interrupt.

INT. Maskable Interrupt (Input, active Low). A Low on this line requests a maskable interrupt.

System Pins

CLK. Clock Output (Output). The frequency of the bus timing clock is derived from the oscillator input (external oscillator) or crystal frequency (internal oscillator), by dividing the crystal or external oscillator by two, four, or eight (as programmed).

XTALI. Clock/Crystal Input (Time-base input). Connects a series resonant crystal or an external single phase clock to the on-chip oscillator.

XTALO. Crystal Output (Time-base output). Connects a series resonant crystal to the on-chip oscillator.

+5V.

GND.