Military Standard Products

UT69R000 Rad-Hard MicroController

T-49-19-16

Data Sheet

UNITED TECHNOLOGIES MICROELECTRONICS CENTER

November 1991

- ☐ Harvard architecture
  - 64K data space
  - 1M instruction space
- ☐ High throughput engine
  - 2 clocks per instruction
  - 8 MIPS @ 16 MHz
  - Static design
- ☐ 15 levels of interrupts
  - 8 external user defined interrupts
  - Machine error and power fail
- ☐ Two on-board 16-bit interval timers
  - Timer A, 10 µs/bit
  - Timer B, 100 µs/bit resolution
- ☐ 8-bit software controlled output discrete bus
- ☐ Register-oriented architecture has 21 user-accessible registers
  - 16-bit or 32-bit register configurations
- ☐ Supports direct memory access (DMA) system configuration
- ☐ Built-in 9600 baud UART
- ☐ Full military operating temperature range, -55°C to +125°C, screened to specific test methods listed in Table I MIL-STD-883 Method 5004 for Class S or Class B
- ☐ Radiation-hardened process and design; total dose irradiation testing to MIL-STD-883 Method 1019
  - Total dose: 1.0E6 rads(Si)
  - Dose rate upset: 1.0E9 rads(Si)
  - Dose rate survival: 1.0E11 rads(Si)
  - Single event upset: < 25.6E-6 errors/device-day
- ☐ Post-radiation AC/DC performance characteristics guaranteed by MIL-STD-883 Method 1019 testing at 1.0E6 rads(Si)
- ☐ Latchup immune 1.5-micron CMOS, epitaxial, double-level-metal technology
- ☐ Packaging options:
  - 132-lead flatpack
  - 144-pin pingrid array

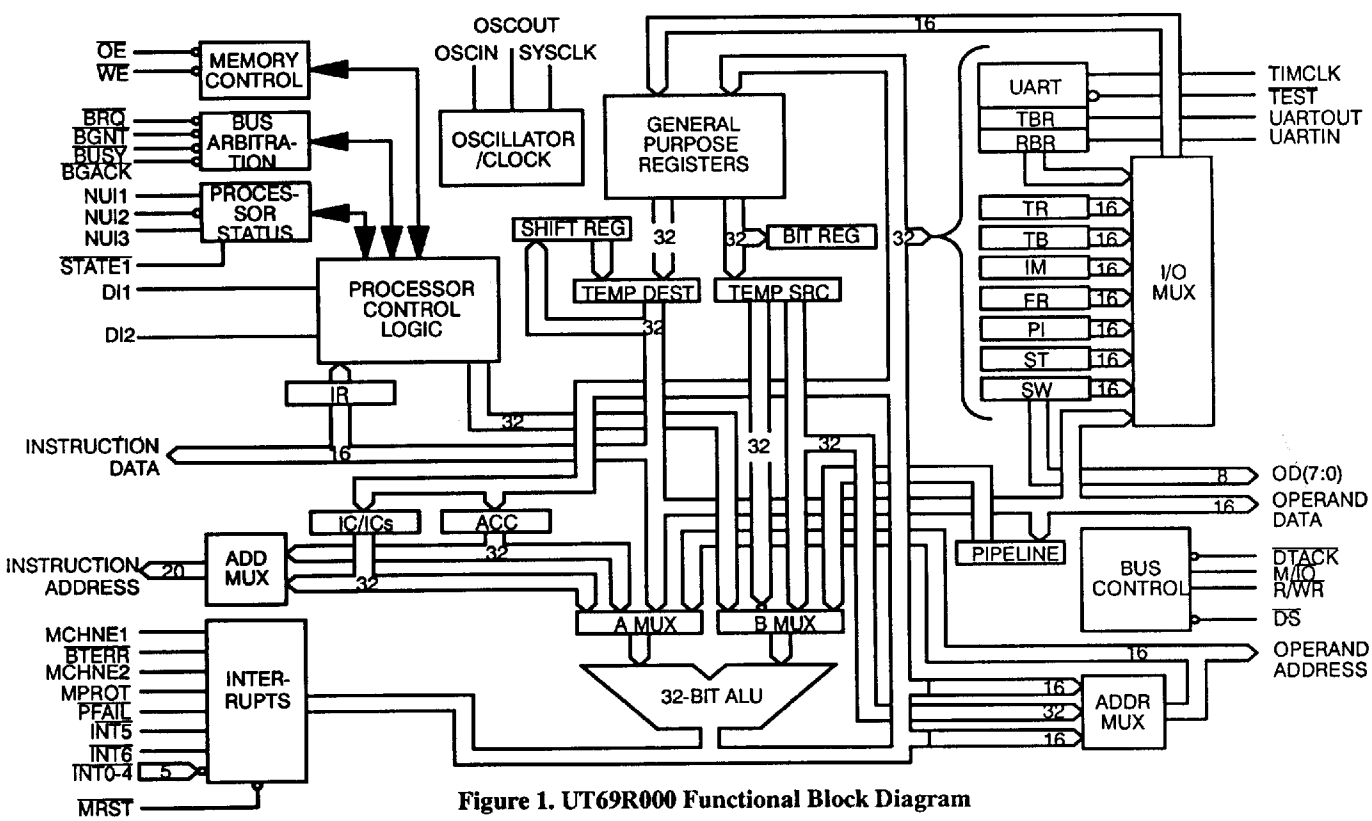


Figure 1. UT69R000 Functional Block Diagram

T-49-19-16

Table of Contents

1.0 Introduction 4
1.1 General Description 4
1.2 General Operation 4
2.0 Register File 6
2.1 General Purpose Registers 6
2.2 Specialized Registers 6
2.2.1 Specialized Register Description 6
3.0 Instruction Port 16
3.1 Instruction Port Operations 17
3.1.1 STRI Instruction Bus Cycle 17
3.1.2 LRI Instruction Bus Cycle 18
4.0 Operand Port 19
4.1 Operand Bus Cycle Operation 20
4.2 DMA Operation and Bus Arbitration 23
5.0 Discrete Input/Output 25
5.1 Output Discrete Bus 25
5.2 Discrete Inputs 26
6.0 Interrupts 26
6.1 Interrupt Control 26
6.1.1 Interrupt Status 27
6.1.2 Interrupt Processing and Vectoring 27
6.2 Interrupt Sources 28
6.3 Interrupt Hardware 28
6.4 Interrupt Latency 28
7.0 Monitor 28
7.1 Using the Monitor 29
7.1.1 Examine Command 33
7.1.2 Modify Command 33
7.1.3 Continue Command 34
7.1.4 Run Command 34
8.0 Internal UART Operation 34
8.1 UART Transmitter Operation 34
8.2 UART Receiver Operation 35
9.0 Programming Interface 35
9.1 Data Formats 35
9.2 Instruction Formats 36
9.3 Addressing Modes 37
9.4 Data Movement Operations 38
10.0 Pin Description 39
11.0 Absolute Maximum 46
12.0 Recommended Operating Conditions 46
13.0 DC Electrical Characteristics 47
14.0 AC Electrical Characteristics 48
15.0 Packaging 58
16.0 Ordering 60

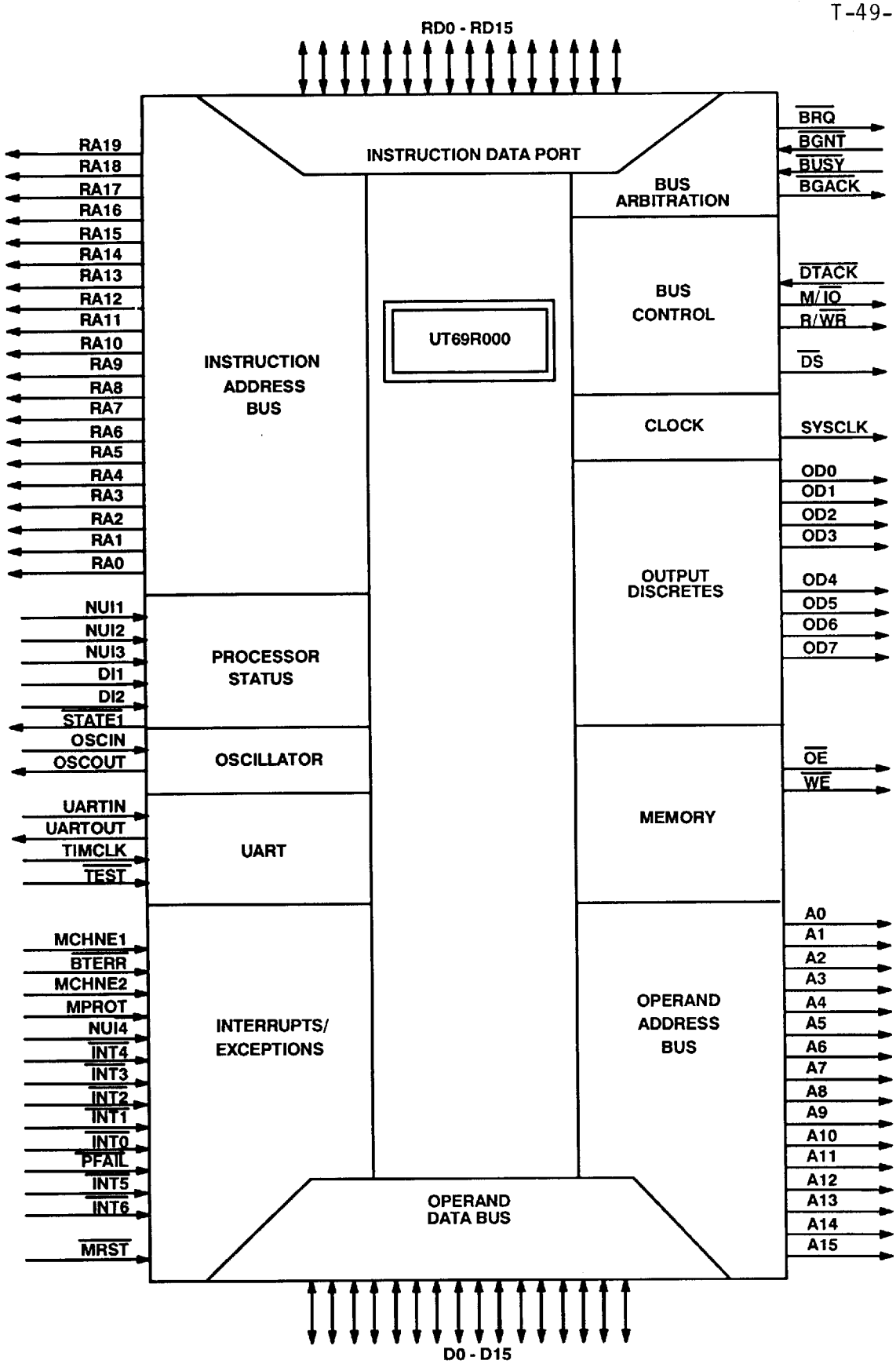


Figure 2. UT69R000 Pin Function Diagram

1.0 Introduction

The UT69R000 is a radiation-hardened high-performance microcontroller designed, manufactured, and tested to meet the rigorous radioactive environments. UTMC designed and implemented the UT69R000 using an advanced radiation-hardened twin-well CMOS process. The combination of radiation-hardness, high throughput, and low power consumption makes the UT69R000 ideal for high-speed systems in satellites, missiles, and avionics applications.

1.1 General Description

The UT69R000 is a versatile microcontroller designed to meet real-time control type applications. Support functions often found external to a microprocessor are integrated within the microcontroller. Functions include UART, interval timers, 10 external interrupt vectors, and a 8-bit output discrete bus.

The UT69R000 core (machine) is a two port microcontroller that accesses instructions from a 1M x 16 instruction port; a second port (64K x 16 data port) is available for data storage. Data transfer acknowledge allows the addition of wait states on the data port. The machine performs overlapping fetches and executes speeding instruction throughput. A 12 MHz operating clock frequency provides up to 6 MIPS of throughput. A later section of this data sheet expands on this concept.

The UT69R000 architecture is based on 20 16-bit general purpose registers providing the programmer with extensive register support. The UT69R000's flexibility is enhanced by the concatenation of 16-bit registers into 32-bit registers. In addition, all registers are available for use as either the source or destination for any register operation.

All UT69R000 circuitry is of static design. Internal registers, counters, and latches do not require refresh as with dynamic circuit design. Therefore the UT69R000 can operate from DC to the upper frequency limit of 16 MHz. This type of operation is especially useful in power critical application such as satellites.

The UT69R000 fully supports multiprocessor systems, DMA, and complex bus arbitration. Bus control passes among bus masters operating on the same bus. The bus master can be one of several UT69R000s or any other device requiring DMA.

The UT69R000 supports 15 levels of vectored interrupts. Ten of these are external interrupts, all of which are user-definable. All interrupts are serviced in order of priority.

The UT69R000's three basic instruction formats support 16-bit and 32-bit instruction. The formats are Register-to-Register, Register-to-Literal, and Register-to-Long-Immediate instructions.

Figure 3 shows the UT69R000's general system architecture.

1.2 General Operation

The UT69R000 reduced instruction set consists of 35 separate instructions, most of these instructions execute in two clock cycles providing high-throughput. The UT69R000 has a Harvard architecture which incorporates two address and two data buses. One set of address and data buses interfaces with instruction memory (instruction port) and the other interfaces with data memory (data port). The instruction port consists of a 20-bit address bus and 16-bit data bus. The maximum program length of any program is 1 mega-word. The data port consists of a 16-bit address and data bus, allowing access to 64K x 16 of data storage.

The instruction port is dedicated to the storage of instruction code, however two instructions exist that allow the instruction port manipulation by the machine. These instructions are the Load Register from Instruction Memory (LRI) and Store Register to Instruction Memory (STRI).

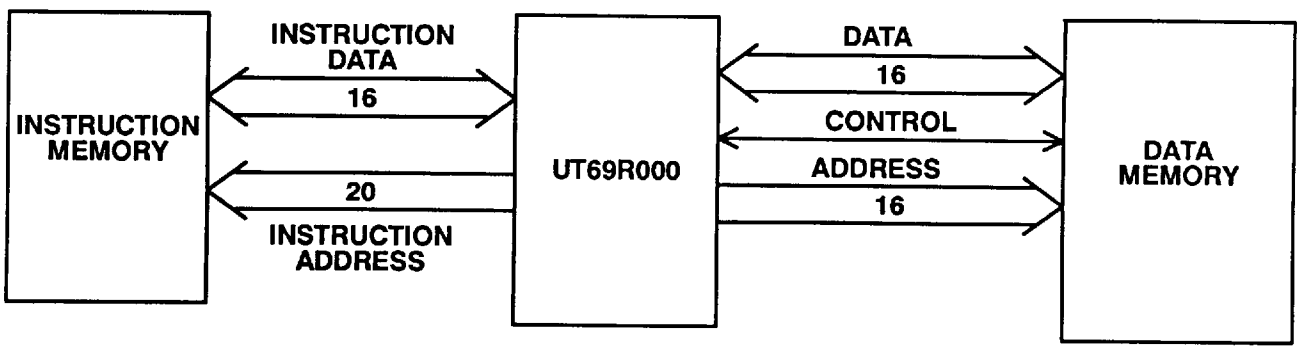


Figure 3. UT69R000 General System Architecture

The UT69R000 begins operation by first generating an address on the instruction port; valid data (instruction) is then latched into the Primary Instruction Register (PIR). After the machine stores the instruction in the PIR, the machine begins execution of the instruction in the Instruction Register (IR). If the present instruction in the IR requires only internal processing, the machine does not exercise the data bus. If the machine needs additional data to complete the instruction the machine begins arbitration for the data port.

Data port arbitration begins with the machine asserting the Bus Request ($\overline{BRQ}$) signal. The machine samples the Bus Grant ($\overline{BGNT}$) and Bus Busy ($\overline{BUSY}$) signals on the falling edge of the clock (OSCIN). When the machine detects that the previous bus controller has relinquished control of the bus, the machine generates a Bus Grant Acknowledge ($\overline{BGACK}$) signal signifying that it has taken control of the bus (i.e., data port).

After the UT69R000 takes control of the bus, it generates valid address and data information. If the machine is interfacing to slow memory or other peripheral devices that require long memory-access times, the Data Transfer Acknowledge ($\overline{DTACK}$) signal extends the memory cycle time. By holding off the assertion of $\overline{DTACK}$, the slow device lengthens the memory cycle until it can provide data for the machine.

The UT69R000 controls the vectoring and prioritizing of interrupt service. Internal logic selects one of 15 interrupt vectors, each interrupt vector is allocated four memory locations. Use the four memory locations to store return from interrupt service address information along with the interrupt service routine's location. The UT69R000 controls prioritizing of coincident interrupts.

Perform UART control and maintenance via input/output commands OTR and INR. These commands allow the programmer to read UART status, and error information, as well as upload and download information to the receive and transmit buffers respectively.

Figure 4 shows an example of a system configuration.

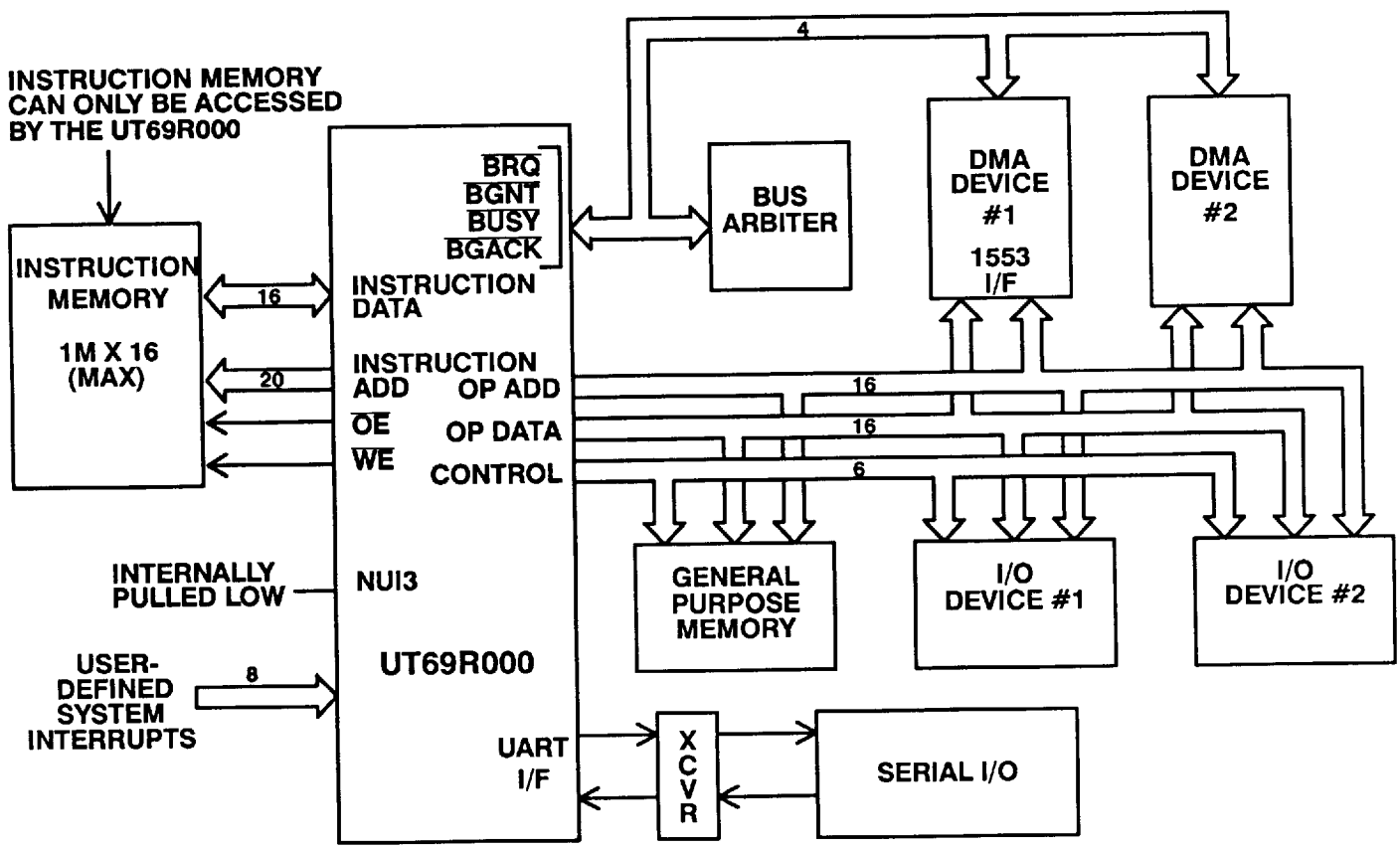


Figure 4. The UT69R000 Example System Configuration

2.0 Register File

The UT69R000 has a register-oriented architecture. The registers within the machine fall into two categories, general purpose and specialized registers. All registers are accessible to the programmer through the instruction set. The programmer uses data from these registers to perform arithmetic and logical functions, alter program flow, detect various system and machine faults, determine machine status, control UART and timer functions, and exception handling.

2.1 General Purpose Registers

Figure 5 shows the UT69R000's 20 general purpose registers. The UT69R000 normally accesses these registers as single-word 16-bit registers although the machine can concatenate these registers into 32-bit double-word register pairs. When the programmer uses the general purpose registers as a double-word register pair, the most significant 16 bits of the 32-bit words are stored in the even-numbered register of the register pair. For instance, if a 32-bit word is stored in Register Pair XR6, the most significant word is stored in register R6 and the least significant word is stored in register R7.

In addition to the 20 general purpose registers, the UT69R000 has a 32-bit accumulator (ACC). The ACC is normally a destination register, although under certain circumstances it can be the source register (INR RD, ACC). The accumulator retains the most significant half of the product during a multiply instruction or the remainder during a divide operation.

16 BITS	16 BITS	CONCATENATED 32-BIT REGISTER PAIR
R0	R1	XR0
R2	R3	XR2
R4	R5	XR4
R6	R7	XR6
R8	R9	XR8
R10	R11	XR10
R12	R13	XR12
R14	R15	XR14
R16	R17	XR16
R18	R19	XR18
ACCUMULATOR		ACC

Figure 5. General Register Set

2.2 Specialized Registers

The UT69R000 has 13 special purpose registers. These registers control machine configuration, report status, and interrupts. Below is a list of the special purpose registers. The values in the brackets indicate the power-up condition.

1. Stack Pointer Register (SP) [XXXX (hex)]
2. System Status Register (STATUS) [XXXX (hex)]
3. UART Receiver Buffer Register (RCVR) [XX00 (hex)]
4. UART Transmitter Buffer Register (TXMT) [XX00 (hex)]
5. Pending Interrupt Register (PI) [0000 (hex)]
6. Fault Register (FT) [0000 (hex)]
7. Interrupt Mask Register (MK) [XXXX (hex)]
8. Status/Output Discrete Register (SW) [XXFF (hex)]
9. Instruction Counter Register (IC) [0000 (hex)]
10. Instruction Counter Save Register (ICS) [XXXXX (hex)]
11. Instruction Register (IR) [0000 (hex)]
12. Timer A (TA) [0000 (hex)]
13. Timer B (TB) [0000 (hex)]

The instruction set provides access to most of the special purpose registers.

2.2.1 Register Description

Stack Pointer Register

The UT69R000 uses the 16-bit Stack Pointer Register as an address pointer on PUSH and POP instructions. The machine pre-increments (POP) and post-decrements (PUSH) the Stack Pointer contents. The programmer loads and stores the SP by executing the INR and OTR commands to the stack pointer. Bit 15 is the most significant bit, the least significant bit is bit zero.

System Status Register

The System Status Register provides status information on the UT69R000's internal operation, including status of the internal UART. Register is read via the INR Rd, STATUS instruction. Bit definitions follow.

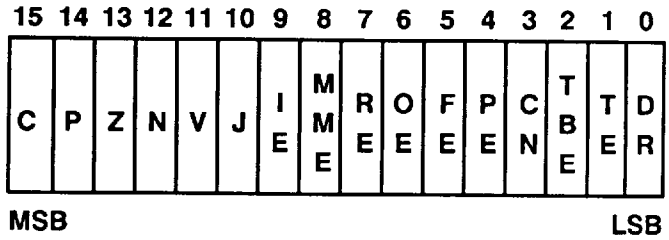


Figure 6. The System Status Register (STATUS)

Bit Number	Mnemonic	Description
Bit 15	C	Carry. This conditional status is set if a carry is generated or no borrow. [0] **Carry Equations:** $C = (\overline{Dm} * \overline{Sm} * \overline{Rm}) + (Dm * \overline{Sm} * Rm) + (Dm * Sm * \overline{Rm})$ Where: Dm - destination register most significant bit Sm - source register most significant bit Rm - result most significant bit (stored in destination register)
Bit 14	P	Positive. This conditional status is set if the result of an operation is positive. [0] **Positive Equation:** $P = \overline{N} * \overline{Z}$
Bit 13	Z	Zero. This conditional status is set if the result of an operation is negative. [0] **Zero Equation:** $Z = \overline{Rm} * \overline{Rm\text{-}1} * \overline{Rm\text{-}2} * \overline{R0}$
Bit 12	N	Negative. This conditional status is set if the result of an operation is negative. [0] **Negative Equation:** N = Rm
Bit 11	V	Overflow. This conditional status is set if the result when an overflow condition occurs. [0] **Overflow Equation:** $V = (\overline{Dm} * Sm * \overline{Rm}) + (Dm * \overline{Sm} * Rm)$
Bit 10	J	Normalized. This conditional status is set as the result of a long instruction and the result is normalized. [0] **Normalized Equation:** J = (R32 XOR R31)
Bit 9	IE	Interrupts Enabled. This bit reflects whether interrupts are disabled or enabled. OTR Rd, ENBL and OTR Rd, DSBL control this bit and function. [0]
Bit 8	DI1	Discrete Input 1. This bit reflects the input stimulus applied to the input pin.
Bit 7	RE	Receiver Error. This bit is the logical OR combination of OE, FE, and PE status bits. [0]
Bit 6	OE	Overrun Error. When active, this bit indicates that at least one data word was lost because the Data Ready (DR bit 0 of the Status Register) signal was active twice consecutively without an INR Rd, RCVR. [0]

Bit Number	Mnemonic	Description
Bit 5	FE	Framing Error. When active, this bit indicates a stop bit was missing from the serial transmission string. Cleared on next transmission. [0]
Bit 4	PE	Parity Error. When active, this bit indicates the serial transmission was received with the incorrect parity. Cleared on next transmission. [0]
Bit 3	DI2	Discrete Input 2. This bit reflects the input stimulus applied to the input pin.
Bit 2	TBE	UART Transmitter Buffer Empty. This bit indicates the Transmitter Buffer Register is empty and ready for data. [0]
Bit 1	TE	UART Transmitter Empty. This bit is low while the UART is transmitting data and goes high when the transmission is complete. [0]
Bit 0	DR	UART Data Ready. This active-high signal indicates the UART received a serial data word and this data is available. Cleared on the execution of INR Rd, RCVR. [0]

UART Receiver Register (RCVR)

The UART Receiver Buffer Register (see figure 7) receives 9600-baud asynchronous serial data through the UARTIN input pin on the UT69R000. Each serial data string contains an active-low Start bit, eight Data bits, an odd Parity bit, and an active-high Stop bit. Figure 8 shows a single serial data string.

While receiving a serial data string, the UT69R000 generates four status flags: Data Ready (DR), Overrun Error (OE), Framing Error (FE), and Parity Error (PE). The UT69R000 stores these bits in the System Status Register.

Receiver buffer register bits 15-8 are always low. Bit numbers, 7 to 0 (RCD7 - RCD0) contain data the UT69R000 receives via the serial data port. RCD7 is the MSB; RCD0 is the LSB.

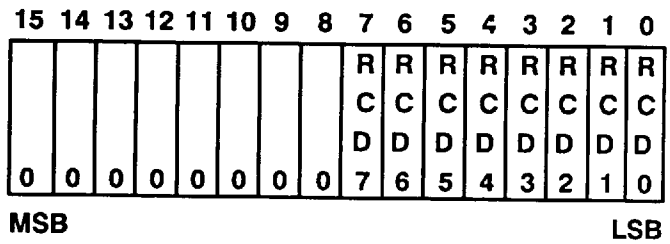


Figure 7. The UART Receiver Buffer Register (RCVR)

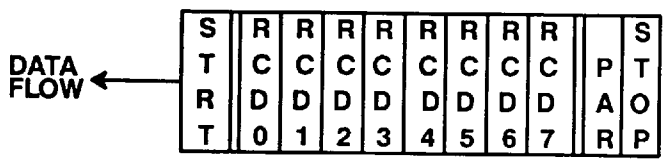


Figure 8. UART Receiver Data String

UART Transmitter Buffer Register

The UT69R000's internal UART forms an 11-bit serial string by combining a Start bit, the eight Data bits from the Transmitter Buffer Register, an odd Parity bit, and a Stop bit. Figure 9 shows the composition of the serial data string. The UT69R000 transmits this serial string through the UARTOUT pin at a rate of 9600 baud (TIMCLK = 12MHz).

The UT69R000's internal UART has a double-buffered data transmission scheme (figure 10). The UT69R000 first loads the data for transmission into the Transmitter Buffer Register. If the UART Transmitter Register is empty, data from the Transmit Buffer Register automatically transfers to the UART Transmitter Register. At this time, the TBE bit goes active indicating more data may be loaded into the Transmit Buffer Register. This double-buffering scheme allows contiguous transmission of serial data streams and also decreases the UT69R000's required overhead for the UART interface. The UT69R000 loads the 8-bit Transmit Buffer Register via the OTR Rd, TXMT instruction.

Two status signals are associated with transmitting serial data. These signals are the UART Transmitter Buffer Empty (TBE) and UART Transmitter Register Empty (TE). TBE and TE are both active high and provide information on the status of double buffering the UART's transmitted data. TBE and TE are read from the System Status Register bits 2 and 1 respectively.

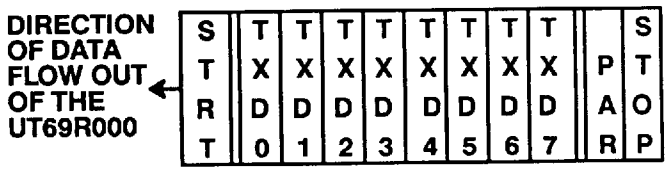


Figure 9. UART Transmitter Data String

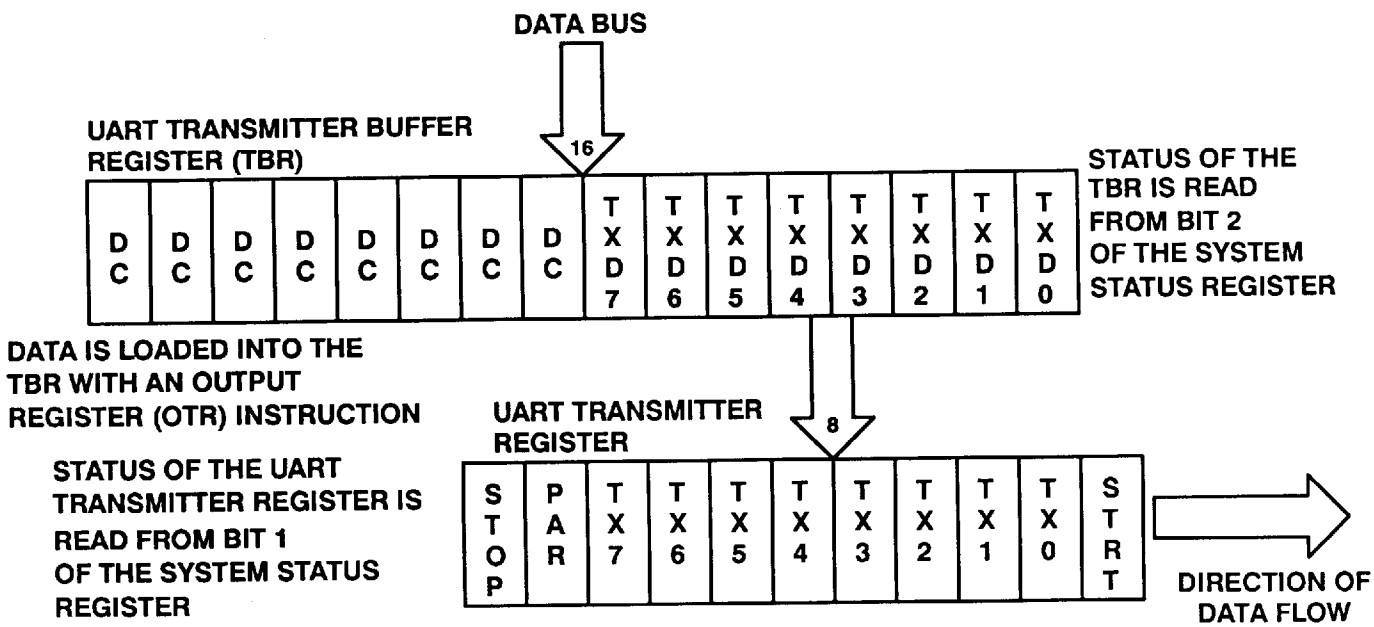


Figure 10. The UT69R000 UART Double-Buffered Transmitter Register

Pending Interrupt Register

The Pending Interrupt Register (PI) contains information on pending interrupts attempting to vector the Instruction Counter Register to a new location. Software or hardware controls the Pending Interrupt Register contents. Any system interrupt, when active, sets the corresponding bit in the register. OTR and INR instruction can also set, clear, and read the Pending Interrupt Register (figure 11).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWD N	MCH E	INT O	USR 3	FIP O	USR 2	USR 1	TIM A	INT 1	TIM B	INT 2	INT 3	INT 6	INT 4	INT 7	N/U

MSB LSB

Figure 11. The Pending Interrupt Register (PI)

Instruction INR Rd, PI stores the PI contents in the destination register. OTR Rd, PI loads the PI with the contents of the destination register. OTR Rd, RPI clears the PI register. For each bit set, to a logic one, in the destination register the corresponding PI bit is cleared. To clear the PI, first read the PI, then clear only the bits set to a logic one. Reading, then clearing the PI prevents the inadvertent clearing of interrupts occurring during execution of an OTR Rd, RPI command.

Example:

```
CLEAR: INR R1, PI
       OTR R1, RPI
```

To generate a software interrupt clear the corresponding bit in the PI register before writing to the PI register.

Example:

```
WRITE: MOV R1, 1000 (hex)
       OTR R1, RPI
       OTR R1, PI
```

Note: Do not enable interrupts while the PI is non-zero.

Bit Number	Mnemonic	Description
Bit 15	PFAIL	Power Fail
Bit 14	MCHE	Machine Error Interrupt
Bit 13	INT0	External Interrupt 0
Bit 12	USR3	Software Interrupt 3
Bit 11	FIPO	Fix Point Overflow
Bit 10	USR2	Software Interrupt 2
Bit 9	USR1	Software Interrupt 1
Bit 8	TIMA	Timer A Interrupt
Bit 7	INT1	External Interrupt 1
Bit 6	TIMB	Timer B Interrupt
Bit 5	INT2	External Interrupt 2

Bit 4	INT3	External Interrupt 3
Bit 3	INT6	External Interrupt 6
Bit 2	INT4	External Interrupt 4
Bit 1	INT7	External Interrupt 7
Bit 0	NU	Not Used*

***Note:**

The UT69R000 interrupt control hardware is designed such that the following sequence always occurs:

The machine will vector to the lowest priority interrupt (vector 43Chex) if interrupts are disabled after an interrupt is latched into the PI register. Interrupts are latched into the PI register on the falling edge of STATE1, interrupts are disabled on the falling edge of clock CK1 internal (rising edge of CK2). The UT69R000 vectors to address 43C (hex) during the fetch cycle of the command following the disable command.

Work Around: Vector 43C hex is not used as a normal interrupt; instead it is used to store the interrupt that occurred coincident with disabling of interrupts. The coincident interrupt is serviced after the interrupts are re-enabled by software.

```
43C:     INR RD, ICS
         JC x, PSEUDO
         NOP
PSEUDO:  INR RD, PI          ; read PI
         OTR RD, RPI         :clear PI
         OTR RD, PI          ;reset PI register, saves coincident interrupt
         CALL RD, RD         ;return
```

The interrupt service routine saves the interrupt that was coincident with the disable interrupt instruction. The UT69R000 will vector to that interrupt after interrupts are re-enabled. The interrupt service routine (@43C) does not affect the condition code flags.

Fault Register

The UT69R000 uses the Fault Register (FT) to indicate the occurrence of a machine-error fault. A machine-error fault cannot be disabled. The UT69R000 uses the logical OR combination of the 16 bit FT to generate a Machine Error interrupt, bit 14 of the PI. Any bits in the FT the UT69R000 does not use are set to a logic zero. The UT69R000 reads, writes, and clears the FT with INR and OTR instructions. Fault Register bits reflecting external pins are level sensitive; bit(s) cannot be reset until the external signal is negated.

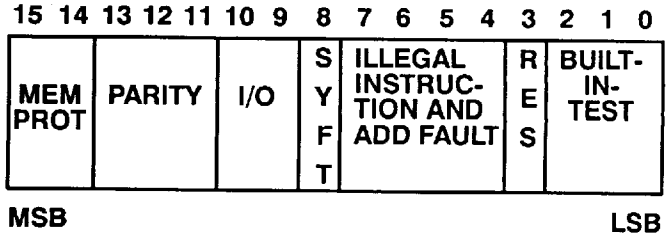


Figure 12. The Fault Register (FT)

Bit Number	Mnemonic	Description
Bit 15	CMPF	CPU Memory Protect Fault. This bit indicates the UT69R000 has detected an access fault on the operand bus (i.e., attempted access to write or read-protected memory). This bit is asserted only when UT69R000 is controlling Operand Data Bus and the MPROT input is asserted. [0]
Bit 14	DMPF	DMA Memory Protect Fault. This bit indicates a DMA device has detected an access fault (i.e., attempted access to write or read-protected memory). This bit is asserted when the UT69R000 is not controlling the Operand Data Bus and the MPROT input is asserted.
Bit 13	MCHNE2	Machine Error 2. This bit indicates that a user-defined machine error has occurred. Reflects status of external input pin MCHNE2.
Bit 12	SFT8	Software Error 8. This bit indicates that a user-defined software machine error has occurred.
Bit 11	SFT7	Software Error 7. This bit indicates that a user-defined software machine error has occurred.
Bit 10	BTERR2	I/O Bus Timer Error. This bit indicates that a bus or time-out occurred. Bit is set if the UT69R000 is performing an I/O access and $\overline{\text{BTERR}}$ input is asserted.
Bit 9	SFT6	Software Error 6. This bit indicates that a user-defined software machine error has occurred.
Bit 8	MCHNE1	Machine Error 1. This bit indicates that a user-defined machine error has occurred. Reflects status of external input pin MCHNE1.
Bit 7	BTERR1	Memory Bus Timer Error. This bit indicates that a bus or time-out occurred. Bit is set if the UT69R000 is performing a memory access and the $\overline{\text{BTERR}}$ input is asserted.

Bit Number	Mnemonic	Description
Bit 6	SFT6	Software Error 6. This bit indicates that a user-defined software machine error has occurred.
Bit 5	SFT5	Software Error 5. This bit indicates that a user-defined software machine error has occurred.
Bit 4	SFT4	Software Error 4. This bit indicates that a user-defined software machine error has occurred.
Bit 3	SFT3	Software Error 3. This bit indicates that a user-defined software machine error has occurred.
Bit 2	SFT2	Software Error 2. This bit indicates that a user-defined software machine error has occurred.
Bit 1	SFT1	Software Error 1. This bit indicates that a user-defined software machine error has occurred.
Bit 0	SFT0	Software Error 0. This bit indicates that a user-defined software machine error has occurred.

Interrupt Mask Register

The Interrupt Mask Register (MK) contains one mask bit for each of the 15 system interrupts. All bits in the MK are set or reset under software control, setting bits 15 and 10 has no effect on the UT69R000's interrupt operation because these interrupts cannot be masked. The UT69R000 reads and writes the MK with instructions OTR Rd,MK and INR Rd,MK. A logical one unmasks an interrupt; logic zero masks an interrupt.

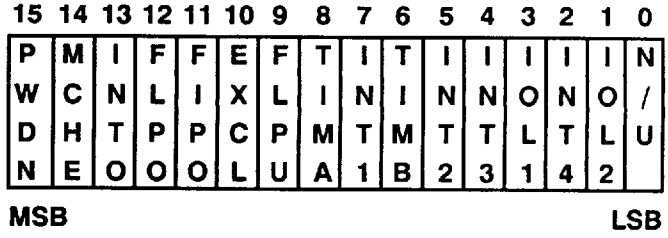


Figure 13. The Interrupt Mask Register (MK)

Bit Number	Mnemonic	Description
Bit 15	NMI	Non-Maskable Interrupt
Bit 14	MCHE	Machine Error Interrupt
Bit 13	INT0	External Interrupt 0
Bit 12	USR3	Software Interrupt 3
Bit 11	FIPO	Fix Point Overflow
Bit 10	USR2	Software Interrupt 2
Bit 9	USR1	Software Interrupt 1
Bit 8	TIMA	Timer A Interrupt
Bit 7	INT1	External Interrupt 1
Bit 6	TIMB	Timer B Interrupt
Bit 5	INT2	External Interrupt 2
Bit 4	INT3	External Interrupt 3
Bit 3	INT6	External Interrupt 6
Bit 2	INT4	External Interrupt 4
Bit 1	INT7	External Interrupt 7
Bit 0	NU	Not Used (see Note on page 11)

Status/Output Discrete Register

The Status/Output Discrete Register reports the C, P, Z, and N bits of the System Status Register along with controlling the output discrete bus (OD(7:0)). Register bits C, P, Z, and N contain the same information as the System Status Register bits 15, 14, 13, and 12. Control of the output discrete bus is allowed via bit 0 through 7 of this register. Use input and output instructions INR Rd, SW and OTR Rd, SW to write and read this register. Use this register to restore the condition codes after interrupt service routines.

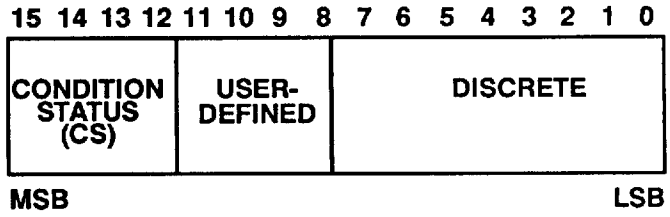


Figure 14. The Status Register Output Discrete (SW)

Bit Number	Mnemonic	Description
Bit 15	C	Carry
Bit 14	P	Positive
Bit 13	Z	Zero
Bit 12	N	Negative
Bit 11	U/D	User-defined
Bit 10	U/D	User-defined
Bit 9	U/D	User-defined
Bit 8	U/D	User-defined
Bit 7	OD7	Output Discrete 7
Bit 6	OD6	Output Discrete 6
Bit 5	OD5	Output Discrete 5
Bit 4	OD4	Output Discrete 4
Bit 3	OD3	Output Discrete 3
Bit 2	OD2	Output Discrete 2
Bit 1	OD1	Output Discrete 1
Bit 0	OD0	Output Discrete 0

Instruction Counter and Instruction Register
The UT69R000's instruction port interface consists of a 20-bit instruction address and a 16-bit data bus. The Instruction Counter (IC) supplies the 20-bit address to memory. The instruction read from memory is then stored into the Instruction Register (IR, 16-bits wide). The IR consists of two sets of internal latches, a Primary Instruction Register latch (PIR, 16-bits wide) and the Instruction Register latch (IRL, 16-bits wide). These two sets of latches allow the UT69R000 to perform overlapping memory fetch and execute cycles. This means the UT69R000 performs a delayed branch when the flow of the program is interrupted. A delayed branch implies that the UT69R000 fetches and executes the instruction following the branch condition before the UT69R000 executes the first instruction at the branch location.

Instruction Counter Save Register
The UT69R000 uses the Instruction Counter Save Register (ICS) when servicing interrupts and branch instructions. When an interrupt or branch occurs, the UT69R000 saves the IC in the ICS. Read the ICS immediately after entering the target routine to save the return location before any other IC save occurs. The UT69R000 reads the ICS using input instruction INR XRd, ICS. Please note that the ICS read requires a 32-bit wide register.

Timer A and Timer B
Timer A and B registers are 16-binary counters. Input/output instructions start, halt, read, and write these counters. Timer A resolution is 10 µs per bit, Timer B has a resolution of 100 µs per bit (TIMCLK at 12 MHz). Each timer generates a time-out interrupt when the counter transitions from FFFF (hex) to 0000 (hex). Time intervals before interrupt are defined as the difference between the loaded value and 0000 (hex). For example, load Timer A with the value FFFE (hex), start Timer A; an interrupt occurs 20 µs later as the timer transitions from FFFF (hex) to 0000 (hex). The Pending Interrupt Register reflects this time-out condition.

Modify the resolution of Timer A and B by scaling the TIMCLK input. For example, to decrease Timer A resolution from 10 µs to 64 µs per bit, TIMCLK is decreased to 1.88 MHz.

3.0 Instruction Port

Instruction port signals include a 20-bit address bus RA(19:0), a 16-bit data bus RD(15:0), and two control signals $\overline{OE}$ and $\overline{WE}$. During instruction and data fetch cycles $\overline{OE}$ is asserted ($\overline{WE}$ negated). Write operations to the port asserts $\overline{WE}$ and negates $\overline{OE}$. Primarily designed for fast access of instruction information, the instruction port does not allow for the inclusion of wait states.

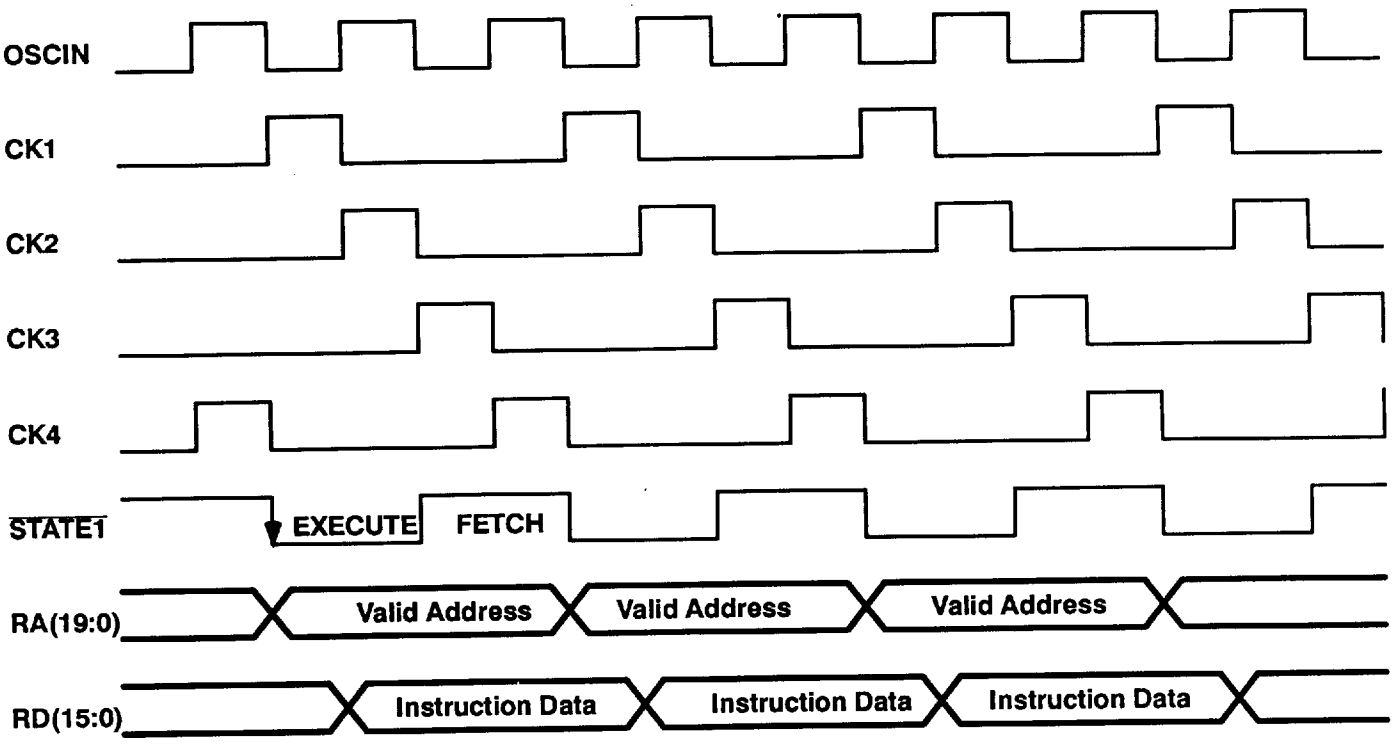
The UT69R000 divides all operations into four distinct time periods (CK1 through CK4). These time periods are based on the processor clock. The UT69R000 performs a separate function during each of these four time periods.

During CK1, the UT69R000 begins executing the instruction in the Primary Instruction Register (PIR). The instruction executed is the instruction the UT69R000 fetched during the previous bus cycle. Also during CK1, the instruction address for the next to fetch from memory becomes valid. Thus the overlapping fetch and execute cycles of the UT69R000. $\overline{STATE1}$ output goes low, indicating the UT69R000 is executing an instruction.

The UT69R000 begins variable width clock period CK2 after completing CK1. For 2 and 3 clock cycle instructions CK2 remains one-half clock cycle in length. During four clock cycle instructions CK2 is stretched to one and a half clock cycles. The following conditions extend time period CK2: (1) Executing a STRI instruction, (2) Executing a LRI instruction, or (3) Executing any instruction access to the operand port. The UT69R000 also extends clock period CK2 because the Operand Port arbitration process. The UT69R000 samples the logical AND combination of $\overline{BUSY}$ and inverted $\overline{BGNT}$ during CK2. If this combination is low, time period CK2 extends until the combination of the two signals is high, indicating the UT69R000 now controls the Operand Port. $\overline{STATE1}$ output remains low for the entire CK2 time period.

At the beginning of CK3, $\overline{STATE1}$ goes high indicating the next instruction is being fetched from memory. The UT69R000's operand address and data buses become active at the beginning of CK3 along with the Bus Grant Acknowledge ($\overline{BGACK}$), M/$\overline{IO}$, and R/$\overline{WR}$ signals. Data Strobe ($\overline{DS}$) asserts one clock cycle after the beginning of CK3, one and a half clock cycles after the start of CK4.

Following CK3 is variable length clock period CK4. The stretch of CK4 occurs during the following instruction execution: (1) Executing a STRI instruction, (2) Executing a LRI instruction, (3) Executing any instruction with Long Immediate data (e.g., MOV Rd, FFFFh), or (4) Executing any operand port access. After time period CK4 starts, the transparent latches that make up the Primary Instruction Register enable allowing the UT69R000 to input the instruction from memory.



Note:

1. Examples of two clock cycle instructions include (internal operations):

   MOV Rd, Rs

   ADD Rd, Rs

Figure 15. Machine Cycle 1 (2 Clock Cycle Instructions) (1)

If the instruction being executed requires access to the operand bus $\overline{DS}$ goes active. The UT69R000 samples the Data Transfer Acknowledge ($\overline{DTACK}$) on the next and every subsequent rising edge of the processor clock. If $\overline{DTACK}$ is not low, the UT69R000 extends time period CK4 until $\overline{DTACK}$ become active or until an error condition is detected -- either Bus Error ($\overline{BTERR}$) or Memory Protect (MPROT) becomes active. $\overline{STATE1}$ remains high during the entire CK4 time period. Figures 15, 16, and 17 show the timing relationships for CK1, CK2, CK3, and CK4 during 2, 3, and 4 clock cycle instructions.

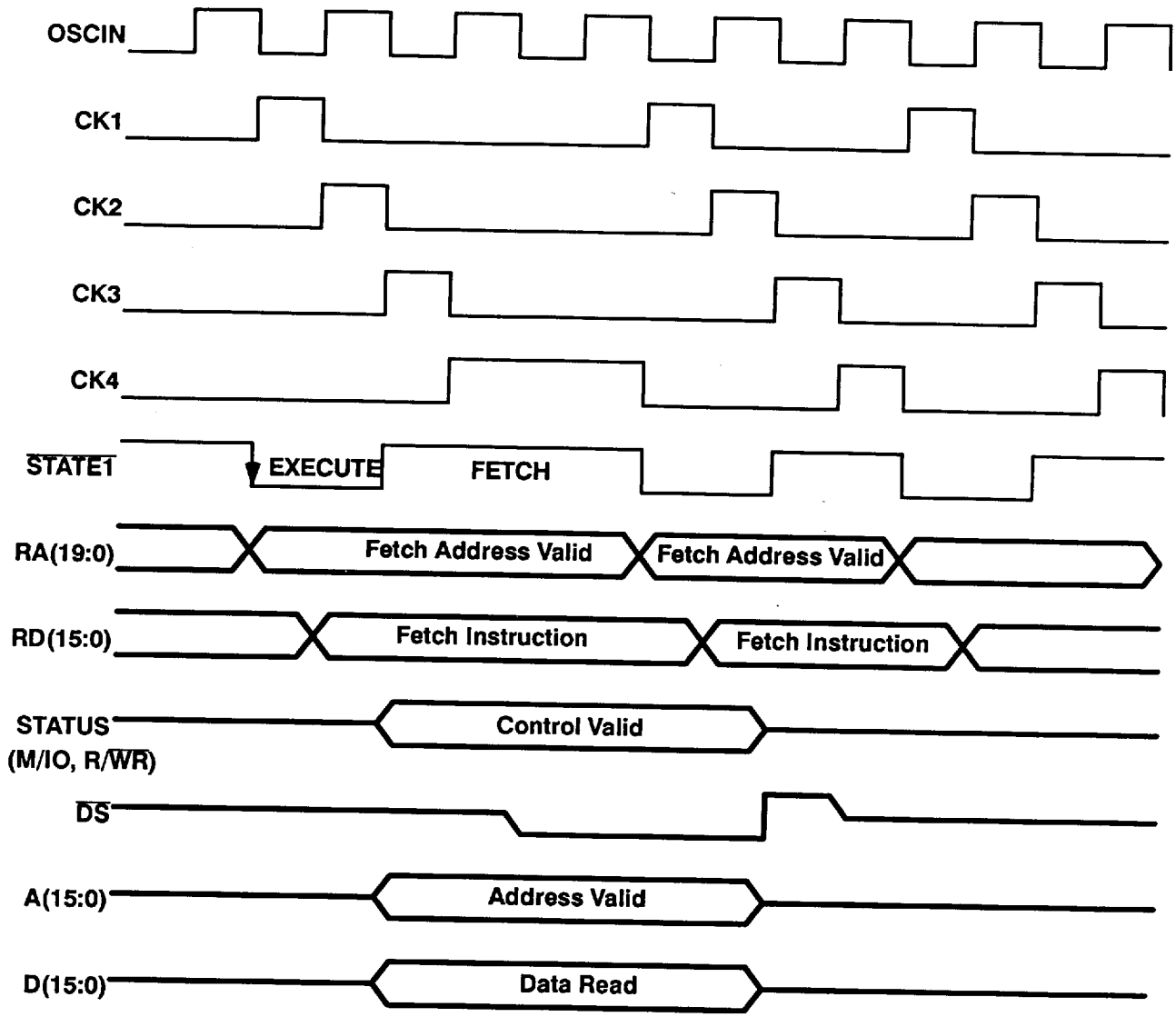
3.1 Instruction Port Operations

Most applications dedicate the instruction port to program information, for these applications $\overline{WE}$ is always negated. The UT69R000 can manipulate the instruction port through instructions Store Register to Instruction Memory (STRI, write access) and Load Register from Instruction Memory (LRI, read access). Section 3.1.1 and 3.1.2 review the STRI and LRI instructions.

3.1.1 STRI Instruction Bus Cycle

During an STRI instruction, instruction data moves from the UT69R000 to the instruction memory. Figure 18 shows the timing diagram of the signal relationships for the UT69R000 during STRI Instruction Bus Cycle Operation. Before the UT69R00 executes the STRI instruction, the system programmer must load the Accumulator Register with the address which will receive the data. When the ACC is loaded with the address information, the UT69R000 can begin executing the STRI instruction.

Executing the STRI instruction begins when the falling edge of OSCIN signals the start of time period CK1. At the beginning of CK1, the data previously stored in the ACC becomes a valid address on the instruction port address bus (RA(19:0)) and $\overline{STATE1}$ output becomes active, indicating the UT69R000 is executing an instruction.



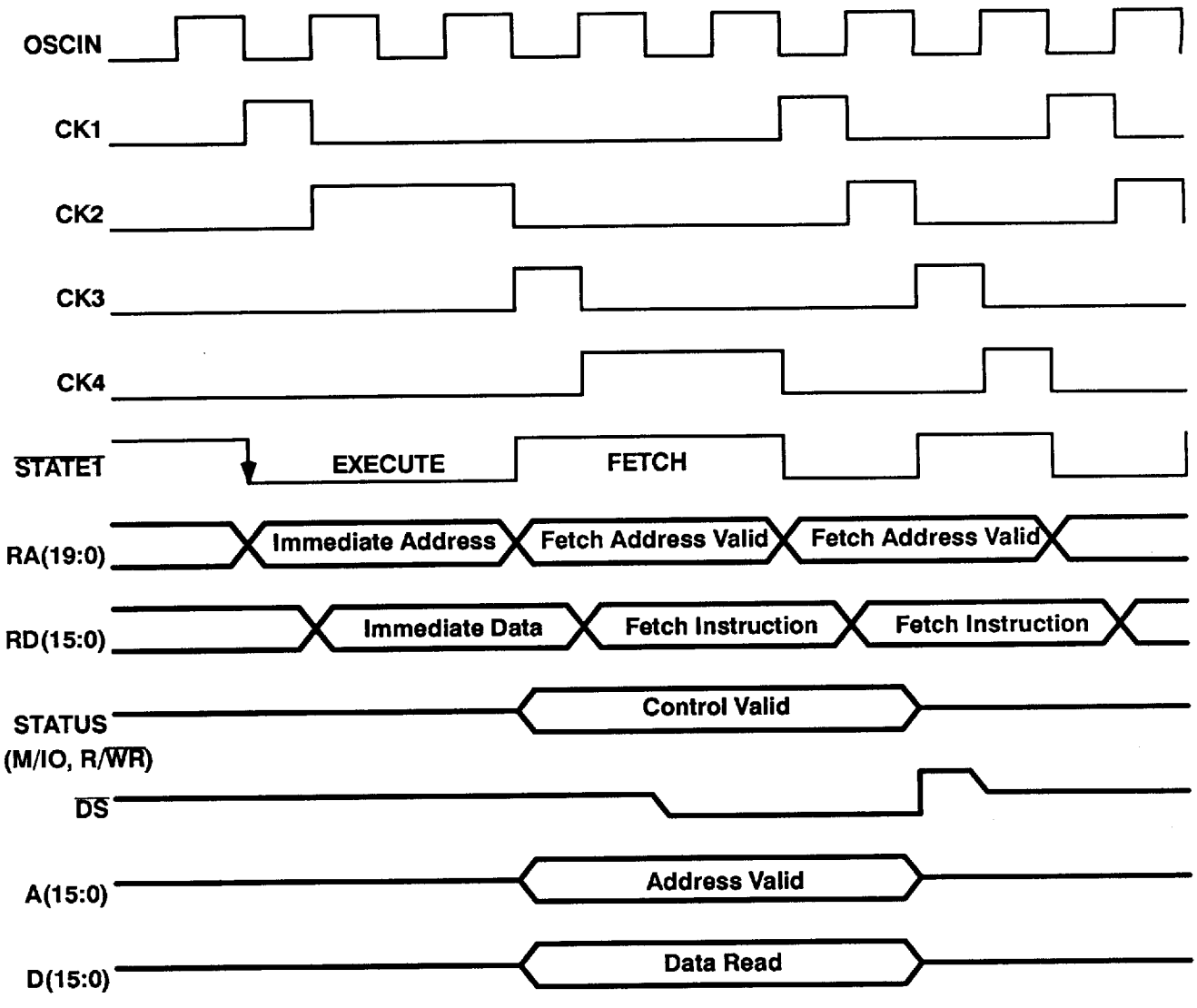
Note:
1. Examples of three clock cycle instructions include (operand port accesses):
   LR Rd, Rs
   STR Rd, Rs

Figure 16. Machine Cycle 2 (3 Clock Cycle Instructions) (1)

3.1.2 LRI Instruction Bus Cycle

During an LRI instruction, the UT69R000 moves the instruction data from the instruction memory to the UT69R000. Figure 19 shows the timing diagram of the signal relationships for the UT69R000 during an LRI Instruction Bus Cycle.

Just as with the STRI instruction, before the UT69R000 executes the LRI instruction the system programmer must load the UT69R000's accumulator with the address from which the data will be read. After the ACC is loaded with the address information, LRI instruction execution can take place.



Note:
1. Examples of three clock cycle instructions include (long immediate accesses):
MOV Rd, FFFF (hex)
ADD Rd, FFFF (hex)

Figure 17. Machine Cycle 3 (4 Clock Cycle Instructions)

Executing the LRI instruction begins when the falling edge of OSCIN signals the start of time period CK1. At the beginning of CK1, the data previously stored in the ACC becomes a valid address on the instruction port address bus (RA(19:0)) and $\overline{\text{STATE1}}$ output becomes active indicating the UT69R000 is executing an instruction.

The data on the data bus is read into the UT69R000 during time period CK2. The function of the remainder of the bus cycle (time periods CK3 and CK4) is the same as for other instructions. $\overline{\text{STATE1}}$ is high, indicating the next instruction is being fetched from memory and is ready for execution during the next bus cycle.

4.0 Operand Port

The UT69R000 Operand Data bus interface supports multiple processor and direct memory access (DMA) configurations. The Operand Address bus A(15:0), data bus D(15:0), and memory control bus signals ($\overline{\text{DS}}$, R/$\overline{\text{WR}}$, and M/$\overline{\text{IO}}$) are TTL-compatible outputs that may be placed in a high-impedance state. These signals are only active during bus cycles when the UT69R000 is the current bus master. On other bus cycles, these signals enter a high-impedance state so an alternate bus master can control the port.

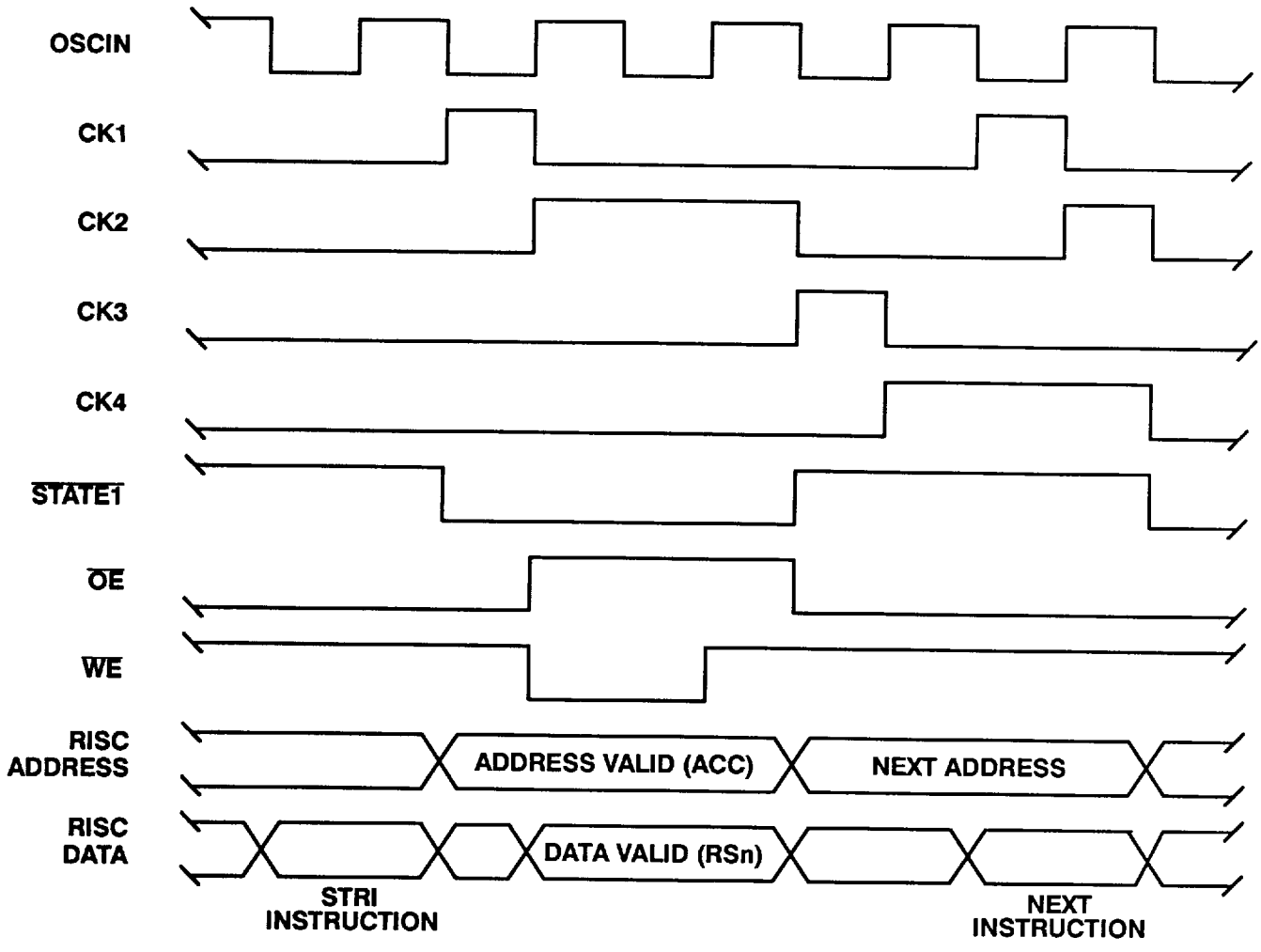


Figure 18. STRI Instruction Typical Timing

Four signals make up the arbitration control bus -- Bus Request ($\overline{BRQ}$), Bus Grant ($\overline{BGNT}$), Bus Busy ($\overline{BUSY}$), and Bus Grant Acknowledge ($\overline{BGACK}$) .

4.1 Operand Bus Cycle Operation

The timing diagram in figures 20, 21, and 22 show signal relationships for the UT69R000 during an operand bus cycle operation. The UT69R000 performs one of four operations involving bus cycles on the Operand buses. These bus cycles are: (1) Memory Read; (2) Memory Write; (3) I/O Read; and (4) I/O Write. The UT69R000 performs all four bus cycle operations similarly. The M/$\overline{IO}$ and R/$\overline{WR}$ signals determine the precise type of bus cycle operation. For the following discussion, refer to figures 20, 21, and 22.

When the Operand bus arbitration process is complete and the UT69R000 controls the Operand address and data buses, time period CK3 begins. The UT69R000 signal controls the Operand port at the beginning of time period CK3 by asserting $\overline{BGACK}$. $\overline{STATE1}$ transitions from low to high. At the same time, the following signals become valid: R/$\overline{WR}$, M/$\overline{IO}$, and the Operand Address bus RA(15:0). Control signals R/$\overline{WR}$ and M/$\overline{IO}$ determine the direction and type of bus cycle taking place.

One-half clock cycle after the beginning of time period CK4 or one full clock cycle after the start of time period CK3, $\overline{DS}$ goes active low. After $\overline{DS}$ has asserted, the UT69R000 samples the $\overline{DTACK}$ input on every subsequent rising edge of OSCIN to determine the duration of CK4. A bus cycle terminates one-half clock cycle after the rising edge of OSCIN when the UT69R000 detects assertion of $\overline{DTACK}$. At this time, the Operand Address Bus A (15:0) and the Operand bus control signals (R/$\overline{WR}$, M/$\overline{IO}$) select the memory or I/O location from which the Operand Data is read, or to which the Operand Data is written. The UT69R000 also samples the MPROTand $\overline{BTERR}$ inputs on the same rising edge of OSCIN. These two inputs indicate an error condition and terminate the current bus cycle.

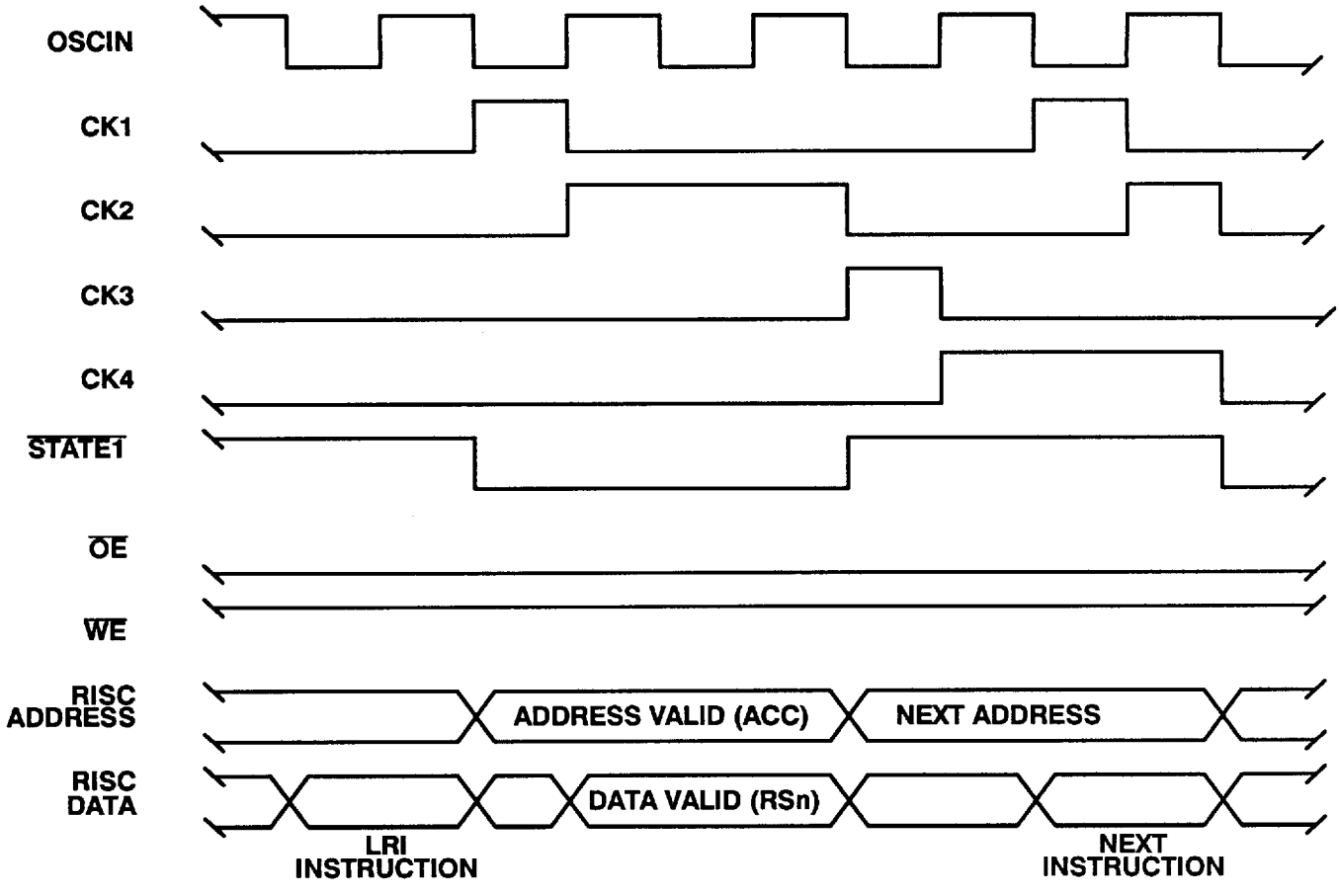


Figure 19. LRI Instruction Typical Timing

After the UT69R000 recognizes the current bus cycle is finished, $\overline{DS}$ becomes inactive (transition from low to high) on the first rising edge of OSCIN after the end of time period CK4. The bus cycle completely ends one full clock cycle after the end of time period CK4, when $\overline{BGACK}$, R/$\overline{WR}$, and the Operand Address and Data buses enter a high-impedance state.

4.2 DMA Operation and Bus Arbitration

Figure 22 shows the timing diagram of the signal relationships for the UT69R000 during a DMA operation. For DMA operations, multiprocessor, and Operand bus arbitration functions, the UT69R000 provides four active-low control signals for managing the Operand bus and preventing bus contention. These signals are Bus Request ($\overline{BRQ}$), Bus Grant ($\overline{BGNT}$), Bus Busy ($\overline{BUSY}$), and Bus Grant Acknowledge ($\overline{BGACK}$).

Each of the four bus control signals provides a specific function for controlling Operand bus operation. The function of each of the four signals is given below.

Bus Request ($\overline{BRQ}$)

The UT69R000 generates $\overline{BRQ}$ to indicate a request to use the Operand buses. The UT69R000 retains control of the buses by keeping the $\overline{BGACK}$ signal active until it no longer requires the buses.

Bus Grant ($\overline{BGNT}$)

An external arbiter generates this input indicating to the UT69R000 that it has the highest priority. This informs the UT69R000 to control the Operand buses as soon as the present bus master relinquishes bus control by asserting $\overline{BUSY}$.

Bus Busy ($\overline{BUSY}$)

Another bus master generates $\overline{BUSY}$ input to the UT69R000, indicating another bus master is using the bus.

Bus Grant Acknowledge ($\overline{BGACK}$)

The UT69R000 generates this signal to indicate it is the present bus master. $\overline{BGACK}$ enters a high-impedance state when the UT69R000 gives up control of the Operand buses.

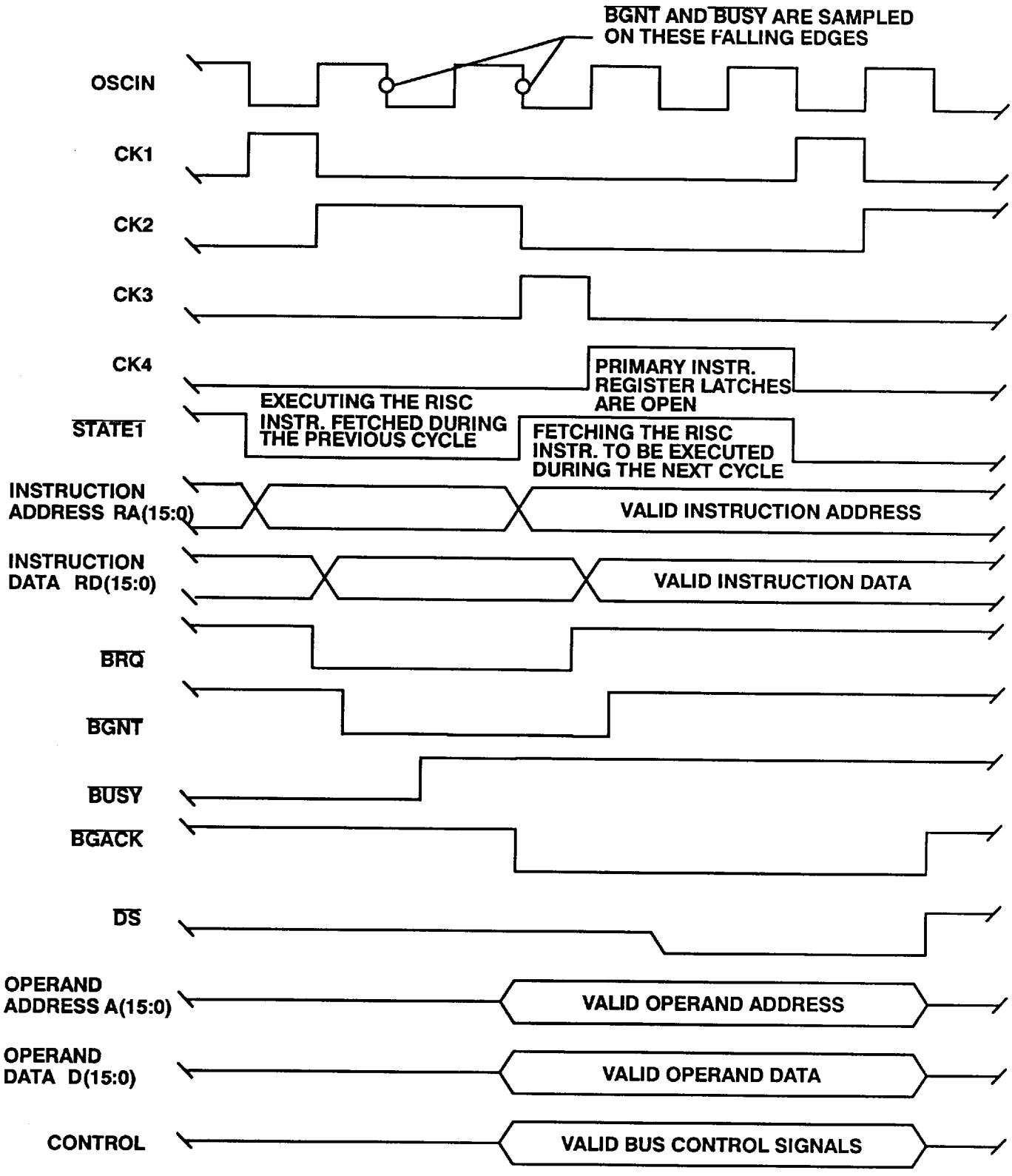
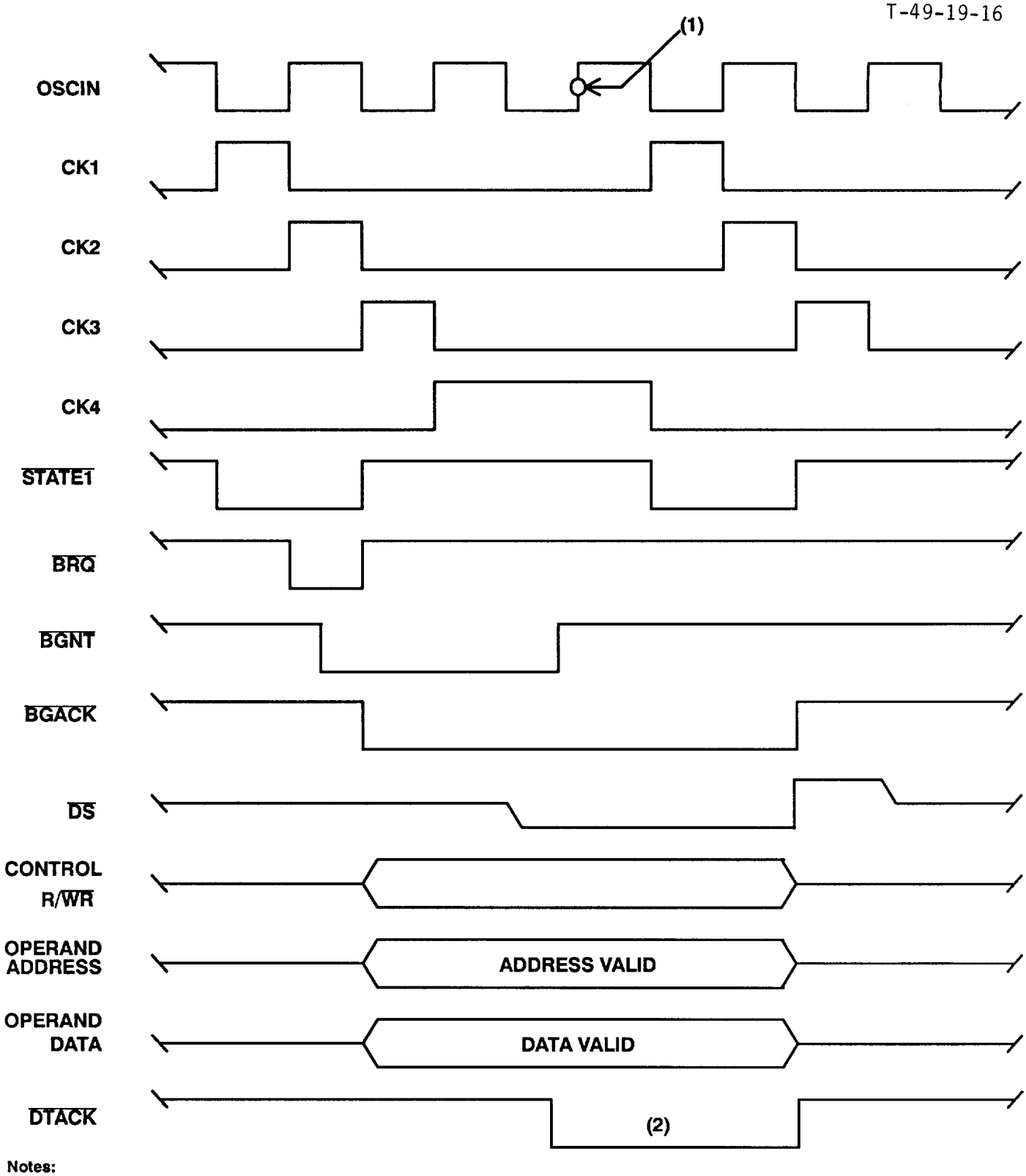


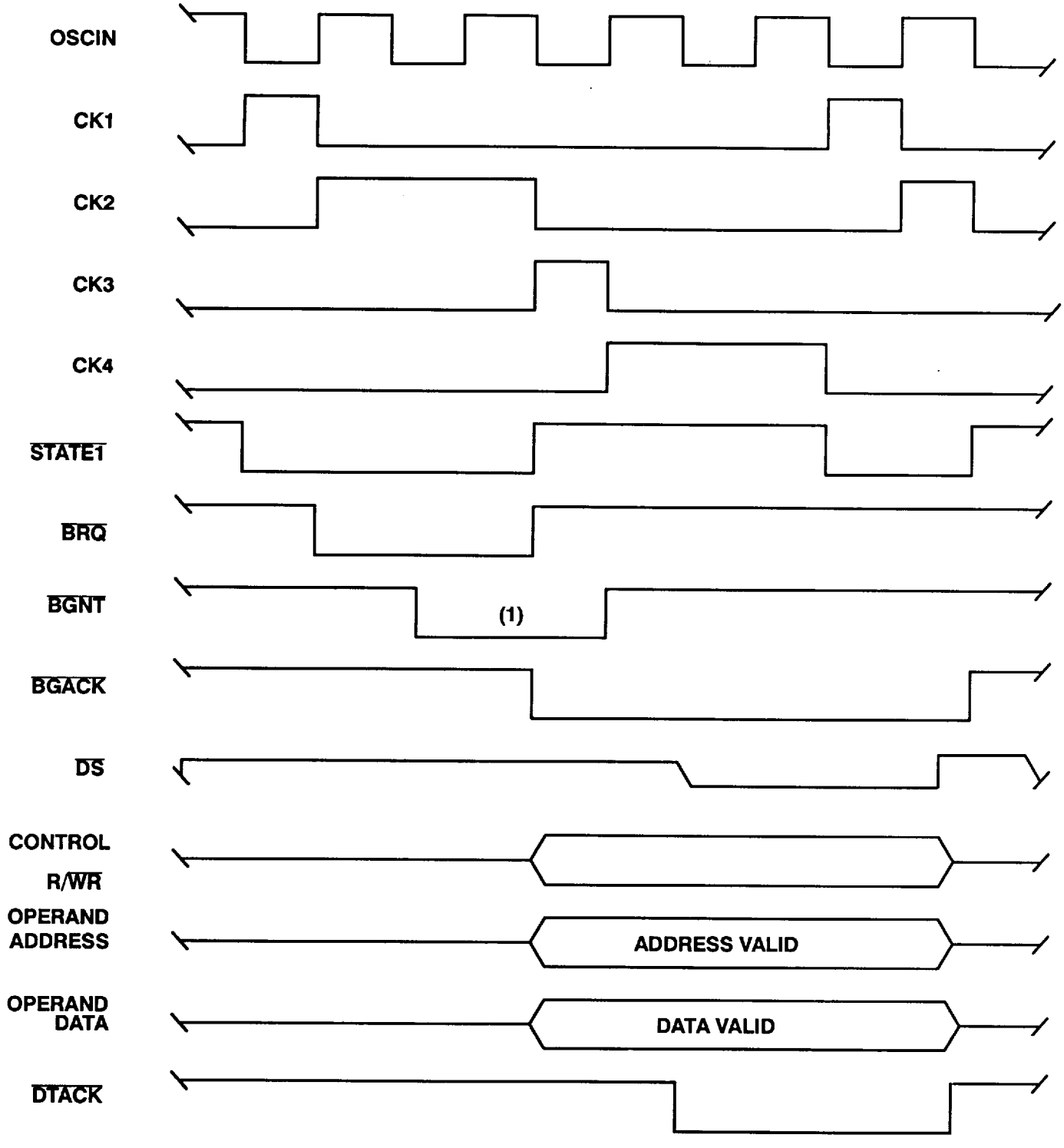
Figure 20. Typical UT69R000 Bus Cycle With Extended Clock Cycles

T-49-19-16



Notes:
1. DTACK must be active by this edge to avoid wait states.
2. DTACK is sampled by the rising edges of OSCIN.

Figure 21. Typical UT69R000 Data Bus Cycle Operation



Note:
1. $\overline{\text{BGNT}}$ is sampled by the falling edges of OSCIN. Wait states are inserted until $\overline{\text{BGNT}}$ is low and $\overline{\text{BUSY}}$ is high.

Figure 22. Typical UT69R000 DMA Bus Cycle

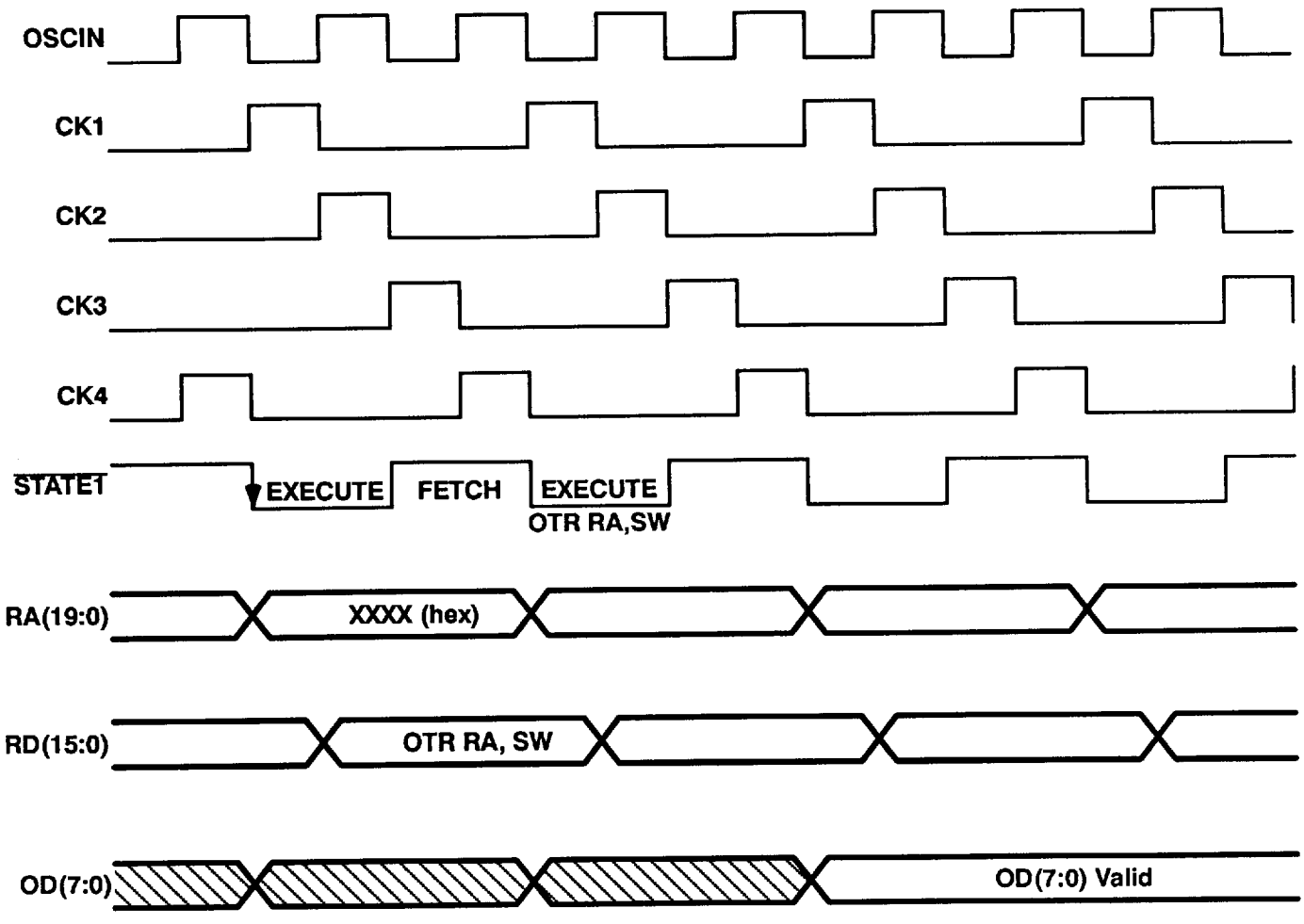


Figure 23. Output Discrete Bus Timing

The UT69R000 requests control of the Operand buses at the beginning of time period CK2 by asserting $\overline{BRQ}$. On every subsequent falling edge of OSCIN, the UT69R000 samples the $\overline{BGNT}$ and $\overline{BUSY}$ inputs. When the UT69R000 detects on the falling edge of OSCIN that $\overline{BGNT}$ has gone low and $\overline{BUSY}$ has gone high, this tells the UT69R000 that it is the new bus master and can now control the Operand buses. The UT69R000 locks out any other bus master from controlling the Operand buses by asserting $\overline{BGACK}$ at the beginning of time period CK3 and holding $\overline{BGACK}$ active until it is ready to relinquish control of the Operand buses. The UT69R000 holds the $\overline{BGACK}$ signal active until the beginning of the CK3 time period of the next bus cycle when the UT69R000 no longer controls the Operand buses.

5.0 Discrete Input/Output

To control external hardware and receive external information the UT69R000 has an 8-bit output discrete bus and two discrete inputs. The discrete input function allows for easy gathering of information from the subsystem. The output discrete bus allows the UT69R000 to control subsystems via a combination of hardware and software.

5.1 Output Discrete Bus

The UT69R000 has eight user-defined output discretes (OD(7:0)). Output Register Instruction OTR Rd,SW governs the logic state of each output discrete. The Status/Output Discrete Register reflects the state of the output discretes. Software can read the contents of this register by executing. Input Register Instruction INR Rd,SW.

Table 1. Interrupt Definitions

INTERRUPT NUMBER	DESCRIPTION
0 (Highest Priority)	Power-Down Interrupt. Cannot be masked or disabled.
1	Machine Error. Cannot be disabled.
2	INT0. External user interrupt.
3	Software interrupt (USR3)
4	Fixed-point overflow. (V bit)
5	Software interrupt (USR2)
6	Software interrupt (USR1)
7	Timer A (If implemented).
8	INT1. External user interrupt.
9	Timer B (If implemented).
10	INT2. External user interrupt.
11	INT3. External user interrupt.
12	INT5. External user interrupt.
13	INT4. External user interrupt.
14 (Lowest Priority)	INT6. External user interrupt.

Useful in the control of external subsystem hardware, the output discrete function is fully static and remains unchanged until rewritten. Outputs can drive standard (i.e., sink or source) TTL loads. These outputs three-state on the assertion of the $\overline{\text{TEST}}$ input pin. Figure 23 shows the timing relationships for a write to the output discrete bus.

5.2 Discrete Inputs

Status register bits DI1 and DI2, bits 8 and 3 respectively, reflect the stimulus applied to the input pins. In a system application the software would make decisions based on the state (i.e., logic one or zero) of either or both of these bits. The system software would poll the Status Register by executing an Input Register Instruction INR Rd,SW; the software then proceeds to perform a test bit on the appropriate bit (i.e., 3 or 8). The result of the test bit determines the next task performed by the software. Section 7.0 discusses an example of using a discrete input to control program for entering the monitor program. Both DI1 and DI2 input buffers have pull-down resistors and can float if not in use.

6.0 Interrupts

The UT69R000 has 15 levels of internal interrupt prioritizing. Upon the occurrence of an enabled non-masked interrupt, the UT69R000 program flow (i.e., instruction counter) transfers to the appropriate interrupt vector. The interrupt vector points to an interrupt service routine. After completing the interrupt service routine the program flow is returned to the main program location. Table 1 shows a list of UT69R000 interrupts.

6.1 Interrupt Control

The Pending Interrupt Register, Mask Register, Status Register, and Fault Register control and report interrupt processing. These registers contain the following interrupt information:

- Interrupt events (PI)
- Interrupt status, masked versus unmasked (MK)
- Interrupt status, enabled versus disabled (STATUS bit 9)
- Machine error interrupts (FT)

The interrupt architecture allows for the disabling and masking of certain interrupts. Output Register Instruction OTR Rd,ENBL and OTR Rd,DSBL control the disable and enable of interrupts. The content of the Rd register is a "don't care" for these commands. Status Register bit 9 reflects the state of interrupts (i.e., enabled or disabled). The Mask Register provides the ability to mask the service of user selected interrupts. Interrupts awaiting service are reflected in the PI Register. Execution of Input Register and Output Register Instructions INR Rd,PI, OTR Rd,PI and OTR Rd,RPI read, write, and clear the PI Register. To latch an interrupt into the PI Register the corresponding bit must be a logic zero before the event occurs. An integral part of interrupt service should include the clearing of the appropriate bit in the PI Register. Section 2.2.1 shows an example of clearing the PI register.

6.1.1 Interrupt Status

The architecture of the UT69R000 allows for the disabling and masking of interrupts. If the software cannot support interrupt service the software can disable (i.e., not recognize) interrupts. The disable feature will prevent the servicing of all interrupts with the exception of power fail ($\overline{\text{PFAIL}}$) and software interrupt (USR2). The UT69R000 will log these interrupts into the PI Register but does not alter program flow to the interrupt vector. Re-enabling interrupts with a non-zero PI Register will result in the UT69R000 vectoring to the highest priority interrupt. To prevent the service of these interrupt clear the PI Register before re-enabling interrupts.

The mask feature allows the software to select particular interrupts for service while masking others. The selection of interrupts, via the mask feature, for service is controlled through the MK Register. Input Register and Output Register Instructions INR Rd,MK and OTR Rd,MK read and write the MK register. The mask feature prevents the servicing of all interrupts with the exception of $\overline{\text{PFAIL}}$ and USR2. Similiar to the disable feature, unmasking and interrupt with a non-zero PI Register results in the vectoring to the appropriate interrupt vector. Writing a logical zero into a Mask Register bit location will prevent the recognition of the specific interrupt (i.e., mask). To un-mask all interrupts write FFFF (hex) to the MK register.

To enable the UT69R000 interrupts architecture the software program enables interrupts by executing instruction OTR Rd,ENBL, followed by a write to the Mask Register, OTR Rd,MK. Interrupts are enabled and disabled on the falling edge of internal clock cycle CK1 (rising edge of CK2).

6.1.2 Interrupt Processing and Vectors

The occurrence of an enabled and non-masked interrupts results in the altering of program flow. Interrupt processing begins by saving the present Instruction Counter Register (IC) in the Instruction Counter Save Register (ICS) followed by automatic disabling of all interrupts (Status Register Bit 9 equals logic 0). The UT69R000 then loads the designated interrupt vector location into the Instruction Counter. The UT69R000 begins interrupt service by executing the code residing at the interrupt vector location.

Table 2. Interrupt Instruction Counter Load Location

INTERRUPT NUMBER	LOCATION (HEX)	MASK-ABLE (Y/N)	CAN USER DISABLE (Y/N)
0	0400	N	N
1	0404	Y	N
2	0408	Y	Y
3	040C	Y	Y
4	0410	Y	Y
5	0414	N	N
6	0418	Y	Y
7	041C	Y	Y
8	0420	Y	Y
9	0424	Y	Y
10	0428	Y	Y
11	042C	Y	Y
12	0430	Y	Y
13	0434	Y	Y
14	0438	Y	Y
15*	043C	Y	Y

* See note on page 11.

Interrupt vectors reside from memory location 400 (hex) to 43C (hex). Each interrupt is assigned a vector with four memory locations (see table 2). These four memory locations allow for storage of the Instruction Counter Save Register (ICS) and a jump (JC), branch (BR), or call (CALL) to the interrupt service routine. An example is shown below.

```
ISR0_INT0: 408 (hex) INR xR0, ICS
           409 (hex) CALL xR18, ISR0
           40A (hex) ISR0
           40B (hex) NOP
```

Read the ICS register with an Input Register Instruction INR Rd,ICS before interrupts are re-enabled or before executing a program branch to assure that the return address in the ICS is not overwritten. The CALL instruction saves the IC into the ICS register and overwrites the interrupt return address with the CALL return address. Similarly, if the interrupts are re-enabled before the interrupt return address is read from the ICS, the occurrence of a new interrupt causes the old return address to be overwritten. It is suggested for CALL instructions the software reserve register pair xR16 for ICS storage; for interrupts the software reserve register pair xR18 for ICS storage. When nested CALLs or interrupts are encountered, the address values stored in register pairs xR16 and xX18, respectively, must be stored in system memory to provide the UT69R000 with full return information.

6.2 Interrupt Sources

Interrupt sources include nine externally generated hardware interrupts, two internally generated hardware interrupts, and four internally generated software interrupts. External interrupts include: $\overline{\text{INT}}$(6:0), MCHNE(2:1), $\overline{\text{PFAIL}}$, $\overline{\text{BTERR}}$, MPROT, and MPAR. Internal hardware interrupts include TIMA and TIMB. Software interrupts include USR(3:1) and FIPO.

User-defined hardware interrupts $\overline{\text{INT}}$(6:0) are available to signal the occurrence of events which require special action by the UT69R000. User-defined interrupts are entered into PI Register bits 2, 8, 10, 11, 12, 13, and 14. Internal hardware interrupts TIMA and TIMB signal the wrap-around of either of these 16-bit counters from FFFF (hex) to 0000 (hex).

Machine error interrupts MCHNE(2:1), $\overline{\text{BTERR}}$, MPROT, and MPAR designate machine error interrupts. The UT69R000 enters machine error interrupts into the Fault Register, the logical OR of all Fault Register bits generates the stimulus to control bit 14 of the PI Register. On the occurrence of a Machine Error Interupt the host examines the Fault Register to determine the specific event that generated the interrupt. Input Register and Output Register Instructions INR Rd,FT, OTR Rd,FT, and OTR Rd,RFT read, write, and clear the Fault Register. Clear the Fault Register before clearing the PI Register.

Generate software interrupts by executing an Output Register Instruction OTR Rd,PI. User-defined software interrupts include USR3, USR2, and USR1. A fourth software interrupt includes FIPO, fixed-point overflow. When enabled and not masked interrupt FIPO signals the assertion of condition code bit V to a logical one. Generate user-defined interrupt USR3, USR2, and USR1 by writing to the PI Register. Please note; clear the specific bit in the PI Register before attempting to generate a software interrupt.

6.3 Interrupt Hardware

All the UT69R000 external interrupts are level triggered. Interrupts $\overline{\text{INT}}$(6:0) and $\overline{\text{PFAIL}}$ are sampled on the rising edge of the OSCIN and latched into the PI Register on the falling edge of $\overline{\text{STATE1}}$ (rising edge of CK1). The minimum pulse width for these inputs is 500 ns.

Machine error interrupts MCHNE(2:1), $\overline{\text{BTERR}}$, MPAR, and MPROT provide stimulus to the PI Register through an S-R flip-flop. The architecture requires removal of the interrupt signal before the Fault Register (FT) and PI Register can be cleared. If the FT and PI Register is cleared while the interrupt input is asserted the specific FT and PI Register bit is re-asserted.

6.4 Interrupt Latency

Figures 24, 25, and 26 display the latency associated with servicing of interrupts. When an interrupt is sampled into the UT69R000 before the falling edge of $\overline{\text{STATE1}}$ (figure 24) interrupt service begins during the following execute machine cycle ($\overline{\text{STATE1}}$ low). If the interrupt is sampled into the UT69R000 after the falling edge of $\overline{\text{STATE1}}$ (figure 25) interrupt service is delayed one execution cycle. Interrupts are first sampled into the device and then latched into the PI Register.

When the interrupt is latched coincident with the fetch and execution of a CALL instruction the interrupt latency increases. Figure 26 shows interrupt latency associated to the CALL instruction. The increase in interrupt latency is due to the temporary disable of the latching of interrupts into the PI register. This temporary disable is due to the fetch of the CALL instruction. The disable is necessary to allow for the UT69R000 to execute the CALL instruction before servicing the interrupt.

7.0 Monitor

Communication between the UT69R000 and a dumb terminal or IRSIM is established via a monitor program written to support the internal UART. When operating in the monitor mode the programmer can (1) examine and modify the UT69R000's internal registers; (2) examine and modify the contents of the operand port memory; (3) examine and modify the contents of I/O subsystems; (4) control program execution. UTMC offers a monitor shell program for the UT69R000. The software programmer can tailor the monitor program to meet specific application.

Assertion of a discrete input can signal the UT69R000 to enter the monitor mode of operation. To perform this function the application software polls the Status Register looking for the assertion (i.e., transition to logic one) of the appropriate discrete input. The UT69R000 then enters the monitor program via a CALL or BR instruction. Interrupts can also be used to access the monitor program. An example of this technique is

accessing the monitor on a specific condition. A specific interrupt event (e.g., memory access location 100) can generate a hardware interrupt to the $\overline{\text{PFAIL}}$ input. Accessing the monitor allows the software programmer to evaluate the state of the UT69R000 and system (i.e., memory or I/O subsystem). Figures 27 and 28 show an example.

7.1 Using The Monitor

When the UT69R000 enters the Monitor mode, it begins executing the monitor program stored in the instruction port. The UT69R000 initially sets its internal UART as the default monitor interface.

To control the UT69R000 with the Monitor, the user simply transmits a predefined set of ASCII characters over the serial data port. The list of the predefined ASCII characters meaningful to the UT69R000's Monitor mode are described in detail in the following sections. The UT69R000 can receive these Monitor control commands with its internal UART, decode them, and then take the appropriate action. *All ASCII characters must be capitalized for the UT69R000 to recognize them.*

The four primary ASCII control characters are E, M, C, and R. These control characters permit the system user to Examine or Modify instruction memory, Operand memory, external I/O, and internal registers, Continue Execution, and Run From a set starting location.

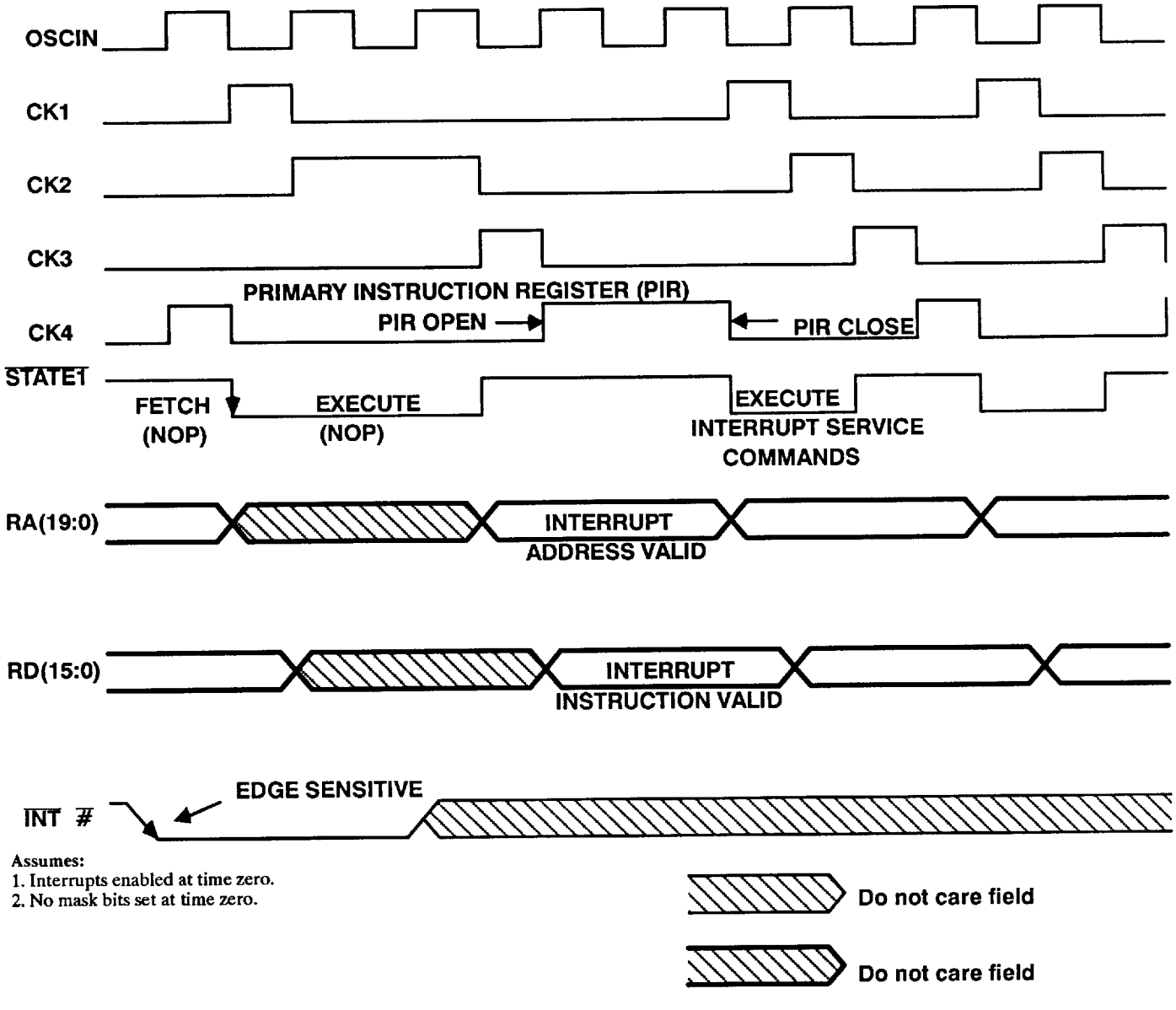
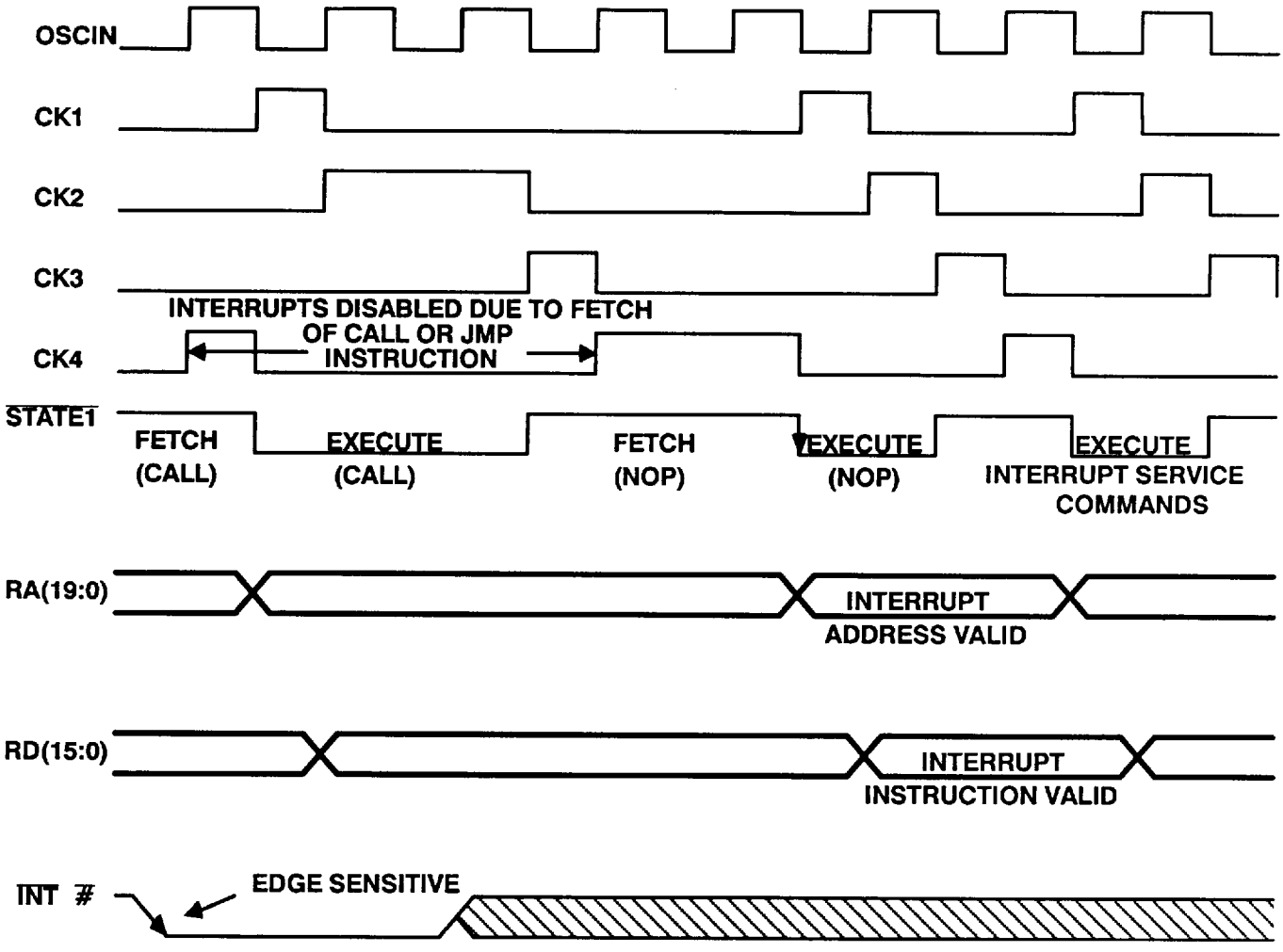


Figure 24. Interrupt Timing



Assumes:
1. Interrupts enabled at time zero.
2. No mask bits set at time zero.

Do not care field

Do not care field

Figure 25. Interrupt Timing

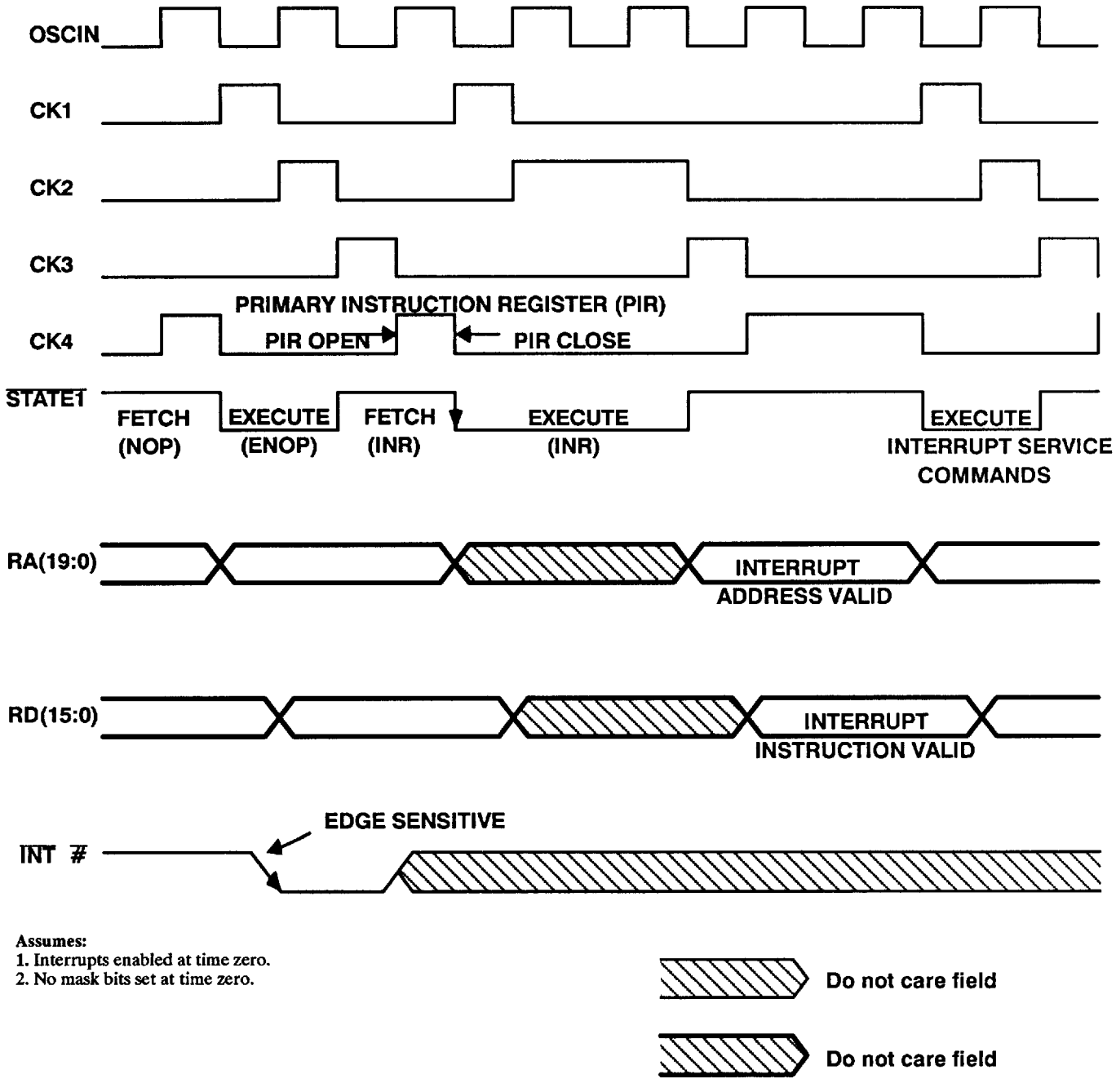


Figure 26. Interupt Timing

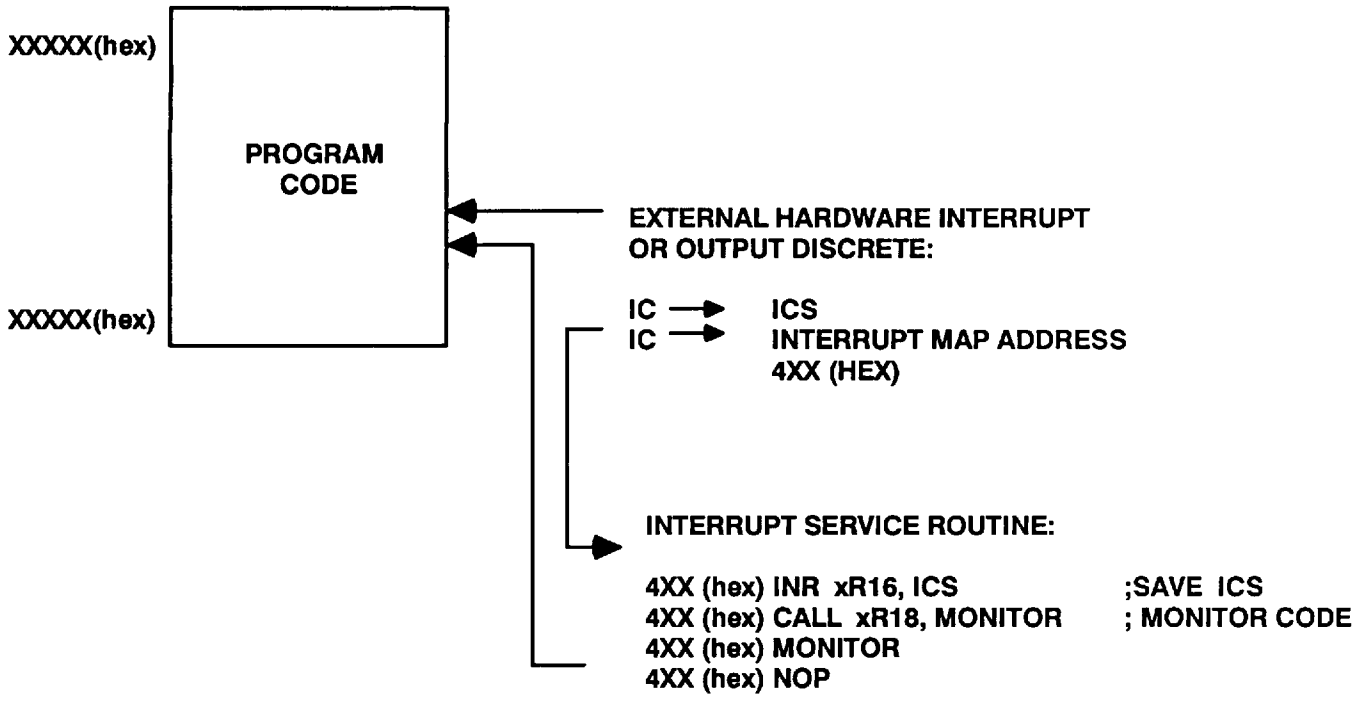


Figure 27. Monitor Operation

Monitor Mode

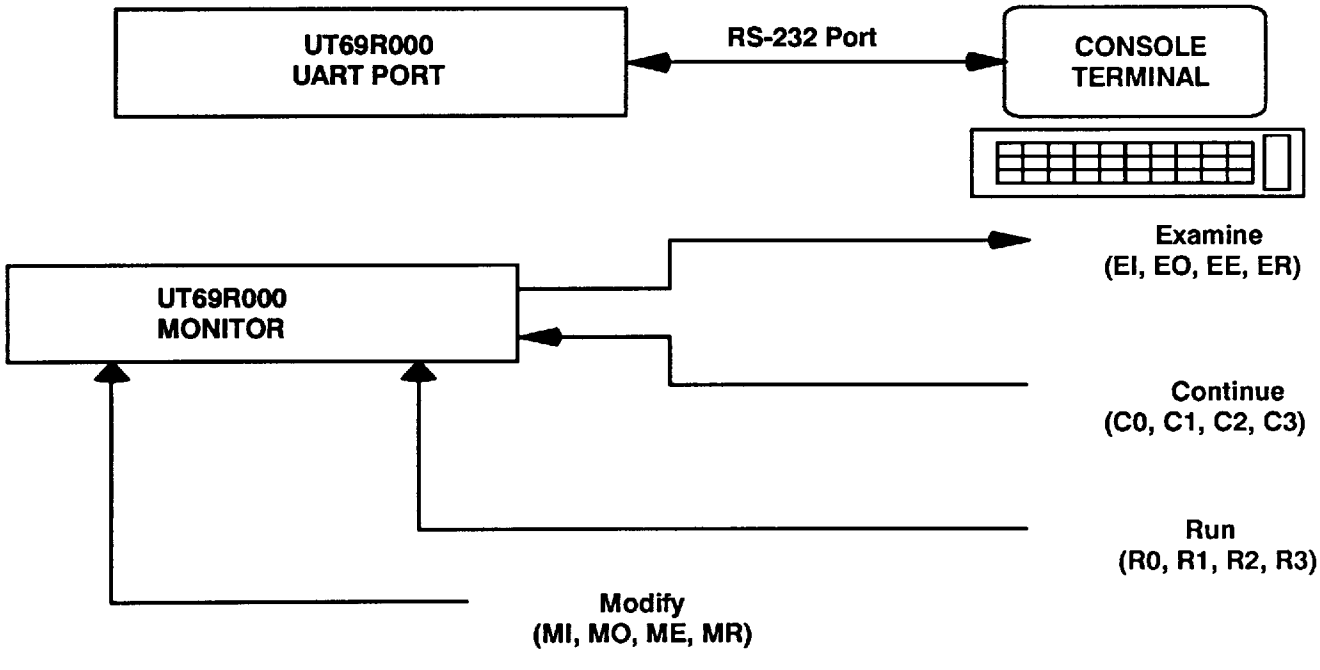


Figure 28. Monitor Mode Operation

7.1.1 Examine Command

The Examine Command has four variations:

(1) EIxxxx - The Examine Instruction memory command. This command permits the user to examine any memory location within the 64K instruction memory space. The EI command is followed by the 16-bit Hex address, above as "xxxx," of the memory location to examine. Valid characters for the instruction address field (xxxx) are 0-9 and A-F.

The user can examine consecutive memory locations by repeatedly entering Space characters. The Monitor continues to display the contents of contiguous memory locations until any non-Space character is received. When the Monitor receives a non-Space character, it terminates EI command execution and waits for the next valid Monitor command.

(2) EOxxxx - The Examine Operand memory command. This command works exactly the same as the EI command except that the user can now examine Operand memory.

(3) EExxxx - The Examine External (I/O) command. This command works exactly the same as the EI and EO commands except that the user can now examine any external I/O location.

(4) ER - The Examine Register command. The Examine Register command allows the user to look at most of the UT69R000's internal registers.

After the UT69R000 has received the ER command, it displays the contents of register R0. The user can examine additional registers by repeatedly transmitting Space characters to the UT69R000. The Monitor displays the registers one after another in the following order: R0 through R15, Status/Output Discrete register (SW), Pending Interrupt Register (PI), Interrupt Mask Register (MK), Fault Register (FT), Timer A (TA) and Timer B (TB). The UT69R000 continues to display its registers until the UT69R000 receives a non-Space character or until the UT69R000 has displayed the complete list of registers. At this time the UT69R000 terminates the ER command and waits for the next valid Monitor command.

7.1.2 Modify Command

The Modify Command has four variations:

(1) MIxxxx,vvvv - The Modify Instruction memory command. This command permits the user to modify any memory location within the 64K instruction memory space. The MI command is followed by the 16-bit Hex address denoted above as "xxxx," of the memory location to examine and the 16 bit Hex value denoted above as "vvvv," the user wishes to place in this memory location. Valid characters for the instruction address field (xxxx) and value field (vvvv) are 0-9 and A-F.

The user can modify consecutive memory locations by entering multiple 16-bit values in the MI command. The MI command would then take the form: MIxxxx,vvvv,vvvv,...,vvvv where the user can enter as many new values as desired. The commas are optional as delimiters. The UT69R000 now modifies instruction memory starting at the given address (xxxx) and continues to modify memory until all new values are in memory.

(2) MOxxxx,vvvv - The Modify Operand memory command. This command works exactly the same as the MI command except that the user can now modify Operand memory. The form of the MO command to alter multiple Operand memory locations is: MOxxxx,vvvv,vvvv,...,vvvv.

(3) MExxxx,vvvv - The Modify External I/O command. This command works exactly the same as the MI and MO commands except that the user can now modify any external I/O location. The form of the ME command to alter multiple external I/O locations is: MExxxx,vvvv,vvvv,...,vvvv.

(4) MRrr,vvvv - The Modify Register command. The Modify Register command allows the user to modify most of the UT69R000's internal registers. The MR command is followed by an 8-bit register ID code, denoted as rr, and a 16-bit value, denoted as vvvv. Table 4 lists the register IDs that the UT69R000 recognizes. Valid characters for the register ID field (xxxx) and value fields (vvvv) are 0-9 and A-F.

The user can use only one MR command to modify one UT69R000 register. Modifying additional registers requires transmitting a separate MR command for each change.

7.1.3 Continue Command
The Continue Execution Command allows the user to resume program execution from the point where the Monitor mode of operation was entered. The Continue Execution command takes the form:

C0 - Resume execution with Timers A and B halted.
C1 - Resume execution with Timer A on and Timer B off.
C2 - Resume execution with Timer A off and Timer B on.
C3 - Resume execution with Timers A and B on.

7.1.4 Run Command
The Run From Memory Location Command allows the user to start program execution from any point within the 1M port space. This command takes the form Rxxxxn where "xxxxx" denotes the 20-bit starting address. Valid characters for the address field (xxxx) are 0-9 and A-F. The value n is either 0,1,2, or 3 and is defined:

0 - Resume execution with Timers A and B halted.
1 - Resume execution with Timer A on and Timer B off.
2 - Resume execution with Timer A off and Timer B on.
3 - Resume execution with Timers A and B on.

8.0 UART Operation

The UT69R000 has an internal UART. Figure 29 shows a diagram of the UT69R000 connected to a serial bus. The UART operates at a fixed frequency of 9600 baud with eight bits, one stop bit, and odd parity. The idle state for the UART is logic zero. The TIMCLK input fixes the baud rate of the UART (9600 baud at TIMCLK equal to 12 MHz). TIMCLK also controls the frequency of the internal timers (TA and TB). The status of the UART is read from the System Status Register (STATUS) bits 7 through 0.

8.1 UART Transmitter Operation
The transmitter portion of the UT69R000's UART is a double-buffered configuration consisting of a Transmitter Register and a Transmitter Buffer Register. The Transmitter Register contains the serial data stream the UT69R000 is currently transmitting through the UART; the Transmitter Buffer Register contains the next message to transmit through the UART. The system programmer reads the status of the Transmitter Register from bit 1 (TE) of the Status and the status of the Transmitter Buffer Register from bit 2 (TBE) of the Status Register. If bit 2 of the Status register is a logical one, the UART transmitter buffer is ready for data, once loaded with data, bit 2 transitions to a logical zero. Bit 1 is a logical zero during serial transmission and transitions to a logical one when transmission from the Transmitter Register is complete. The Status register is read using Input Register Instruction INR Rd,STATUS.

To initiate a serial data transmission, the system designer must first load the data to transmit into the Transmitter Buffer Register with the Output Register Instruction OTR Rd, TXMT. This instruction loads the least significant byte of the source register specified in the instruction into the Transmitter Buffer Register. At this time, TBE goes low and the UT69R000 automatically transfers the data word into the Transmitter Register. After the transfer is complete, TE goes low and TBE transition to a logical one indicating a serial transmission is about to begin and the next data word can be loaded into the Transmitter Buffer Register.

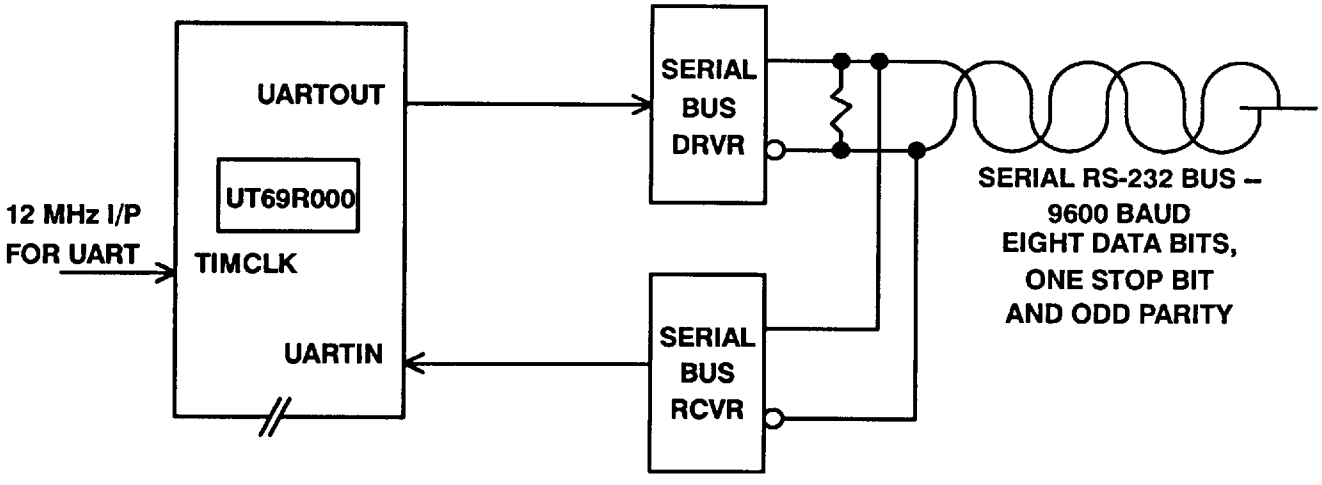


Figure 29. Serial Data Bus Interface to the UT69R000

This double-buffering process allows transmitting contiguous serial data streams. The process of alternately loading the Transmitter Buffer Register with new data and then reading the transmitter status from the STATUS register continues until completion of all serial transmission. An example of UART transmitter software follows:

```
WRITE_UART:  INR R11, STATUS
             TBR R11, 1DH
             BR EQ, WRITE_UART
             NOP
             INR R15, TXMT
```

8.2 UART Receiver Operation

The UT69R000's internal UART has one register associated with the receive function. This register is the UART Receiver Buffer Register (RBR). The least significant byte of the RCVR contains the received serial data. The Status Register contains error information about the serial data in the receiver. Four error bits reflect information status, bit 7 (Receiver Error, RE), which is the logical OR combination of the other three error bits; bit 6 (Overrun Error, OE); bit 5 (Framing Error, FE); bit 4 (Parity Error, PE). An additional status bit for the Receiver is the Data Ready (DR) bit. DR is the least significant bit of the Status Register.

The UT69R000 is ready to receive serial data through the internal UART, it must poll the Status Register to determine when the Data Ready (DR) bit transitions from a logical zero to logical one to signal the UART has indeed received a serial transmission. When DR equals a logic one, the software reads the Receiver by executing and Input Register Operation INR RD, RCVR. The INR instruction takes the eight bits of received data in the and places this data in the least significant byte of the destination register (Rd) specified in the instruction.

When the UT69R000 is finished executing the Input Register Instruction, the software can then determine the validity of the message by testing the RE bit. After the software has checked for a valid message, the data is stored. If the UT69R000 is to receive more data through the UART, the software must return to polling the Status Register to determine the reception of the next valid serial transmission. The Input Register Instruction INR Rd, RCVR clears the DR bit. An example of receiver software follows:

```
READ_UART:   INR R11, STATUS
             TBR R11, 1FH
             BR EQ, READ_UART
             NOP
             OTR R15, RCVR 9.0
```

9.0 PROGRAMMING INTERFACE

9.1 Data Formats

The UT69R000 instruction set supports 16-bit integer single-precision data and 32-bit integer double-precision data. All data is in 2's complement representation.

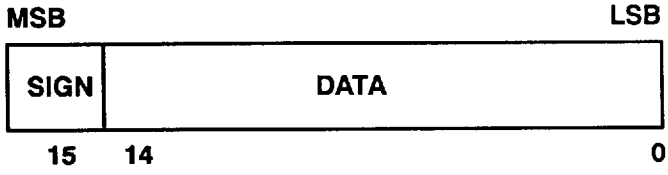


Figure 30a. Single-Precision Fixed-Point Data

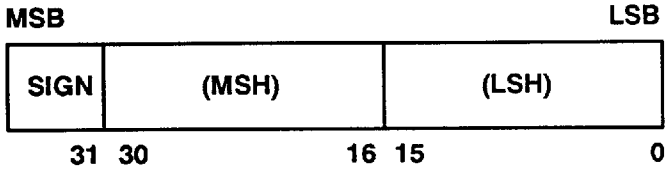


Figure 30b. Double-Precision Fixed-Point Data

The UT69R000 represents the fixed-point data formats as a 2's complement integer with the MSB as the sign bit (figures 30a and 30b).

Operand Size

The UT69R000's instruction set supports three operand sizes: (1) Byte (eight bits); (2) Word (16 bits); and (3) Long Word (32 bit). Byte operands are only allowed with byte instructions. All other instructions support word and long-word operands.

Organization of Data in General Purpose Registers

All 20 of the UT69R000's general purpose data registers support bit, byte, and word operations. When the system programmer specifies a byte operation in a specific instruction, the instruction expects to find the byte of Operand Data in the least significant eight bits of the data register. The least significant bit of each of the data registers is bit 0 and the most significant bit of each of the data registers is bit 15. Any one of the data registers may be the source or destination for the operand.

For long-word operands, the UT69R000 organizes the 20 general purpose data registers as 10 even/odd register pairs. The even-numbered register of the register pair contains the most significant word. All register pairs may be the source or destination operands.

Special Purpose Data Registers
In addition to the 20 general purpose data registers, the UT69R000 has three special purpose data registers: (1) The ACCUMULATOR (ACC); (2) the Stack Pointer (SP); and (3) the Instruction Counter Save Register (ICS).

The Accumulator (ACC) is a 32-bit register used only with multiply, divide, extended shift, Load Register from Instruction memory (LRI), and Store Register to Instruction memory (STRI) instructions. For multiply instructions, the ACC retains the most significant half of the product, and for divide instructions, the ACC retains the remainder. For LRI and STRI instructions, the ACC contains the instruction memory pointer. Note that the ACC can be used as a general purpose register for most operations.

The Stack Pointer (SP) is a 16-bit register usable only with POP and PUSH instructions.

The Instruction Counter Save (ICS) register is a 20-bit register used during calls, jumps, and interrupts.

Register Notation
The UT69R000's instruction descriptions contain a definition of the Register Transfer Language (RTL) that the Assembler uses to describe how the instructions operate. The RTL description of the UT69R000's internal registers is as follows:

- RS*n* -- Source Register where *n* specifies the register number.
- RD*n* -- Destination Register where *n* specifies the register number.
- XRS*n* -- Long-Data Source Register where *n* specifies the register number.
- XRD*n*-- Long-Data Destination Register where *n* specifies the register number.
- IC -- Instruction Counter
- SP -- Stack Pointer
- ACC -- 32-bit Accumulator
- ICS -- Instruction Counter Store Register
- @RS*n*-- Data Register Indirect where *n* specifies the register number
- @SP -- Stack Pointer Indirect
- # -- Immediate Data
- @# -- Immediate Data Indirect

9.2 Instruction Formats
The UT69R000 has three instruction formats (figure 32): (1) Register-to-Register; (2) Register-to-Short Immediate; and (3) Register-to-Immediate.

All the UT69R000's instructions are either word (16-bit) or long-word (32-bit) in length. The only time the UT69R000 uses the long-word instruction format is for the Immediate Source Operand Address Mode.

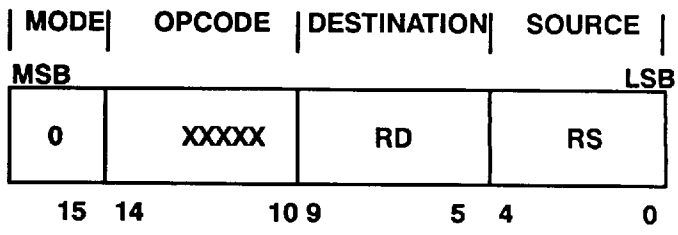


Figure 31a. Register-to-Register Instruction

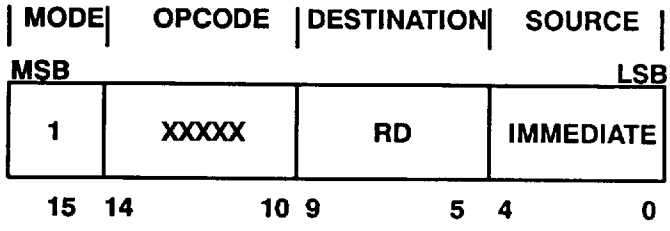


Figure 31b. Register-to-Short Immediate Instruction Format

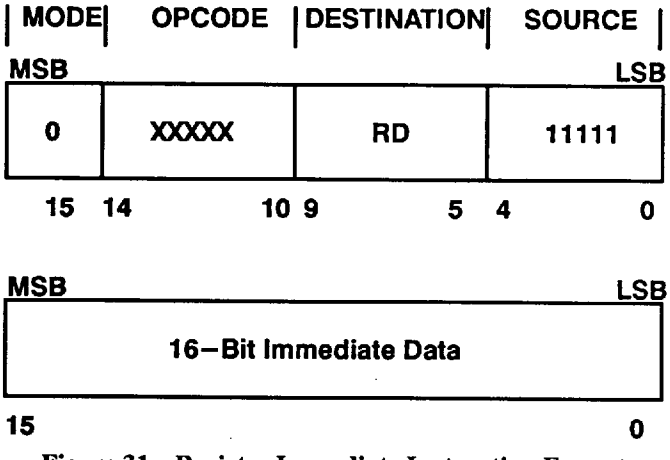


Figure 31c. Register Immediate Instruction Format

The bits in the instructions are defined as follows:

M: Instruction Mode Bit. When M = 1, the UT69R000 interprets the Instruction Source field as a five-bit literal value. If M = 0, the UT69R000 uses the Instruction Source field to specify the source register for the instruction.

Opcode: This field is the five-bit opcode the UT69R000 uses to decode the instruction into a machine operation.

Destination: This field specifies the register the UT69R000 uses for the destination of the instruction.

Source: This field specifies the register the UT69R000 uses for the Instruction Source.

Immediate: If needed, this field contains the 16-bits of immediate data the UT69R000 requires for the long-immediate instruction.

9.3 Operand Addressing Modes

The UT69R000's instruction set supports four basic addressing modes. All instructions require a source operand and a destination operand. The destination operand is a data register (RDn or XRDn) for all instructions, except the Jump on Condition (JC) instruction where the destination register contains a template for the jump condition tested for in the instruction. The source operand can be either a data register or immediate data for all instructions.

The source operand can also be addressed in an indirect mode. In an indirect addressing mode, the source data register or the Stack Pointer contains an effective address. This address points to the memory location for operand data the UT69R000 uses during the current instruction execution. This type of memory addressing is only used with the Load (LR), Store (STR), PUSH, and POP instructions.

Destination Addressing Mode

The destination operand is given explicitly for all UT69R000 instructions. The UT69R000 encodes a five-bit field, bits 9 through 5, in each instruction as follows:

R0 -- 00000	XR0 -- 10000
R1 -- 00001	R16 -- 10001
R2 -- 00010	XR2 -- 10010
R3 -- 00011	R17 -- 10011
R4 -- 00100	XR4 -- 10100
R5 -- 00101	XR16 -- 10110
R6 -- 10110	
R7 -- 00111	XR8 -- 11000
R8 -- 01000	R18 -- 11001
R10 -- 01010	XR10 -- 11010
R11 -- 01011	R19 -- 11011
R12 -- 01100	XR12 -- 11100
R13 -- 01101	XR18 -- 11101
R14 -- 01110	XR14 -- 11110
R15 -- 01111	ACC -- 11111
	NUL -- 10111

Source Addressing Modes

The UT69R000 directly addresses the source operand by using one of three normal modes: (1) Data Register Direct; (2) Literal; and (3) Immediate Long Data.

Data Register Direct

When the UT69R000 uses the Data Register Direct mode, the source operand is one of the data registers. The data register is explicitly stated for all instructions. The UT69R000 encodes a 5-bit field, bits 4 through 0, in each instruction as follows:

R0 -- 00000	XR0 -- 10000
R1 -- 00001	R16 -- 10001
R2 -- 00010	XR2 -- 10010
R3 -- 00011	R17 -- 10011
R4 -- 00100	XR4 -- 10100
R5 -- 00101	XR16 -- 10101
R6 -- 00110	XR6 -- 10110
R7 -- 00111	
R8 -- 01000	XR8 -- 11000
R9 -- 01001	R18 -- 11001
R10 -- 01010	XR10 -- 11010
R11 -- 01011	R19 -- 11011
R12 -- 01100	XR12 -- 11100
R13 -- 01101	XR18 -- 11101
R14 -- 01110	XR14 -- 11110
R15 -- 01111	Reserved -- 10111
and 11111	

Literal

When the UT69R000 uses the Literal mode, the source operand is a 5-bit literal data value. The UT69R000 explicitly states this literal data value for the instructions. The UT69R000 encodes a 5-bit field, bits 4 through 0, in each instruction as follows:

0 -- 00000	-16 -- 10000
+1 -- 00001	-15 -- 10001
+2 -- 00010	-14 -- 10010
+3 -- 00011	-13 -- 10011
+4 -- 00100	-12 -- 10100
+5 -- 00101	-11 -- 10101
+6 -- 00110	-10 -- 10110
+7 -- 00111	- 9 -- 10111
+8 -- 01000	- 8 -- 11000
+9 -- 01001	- 7 -- 11001
+10 -- 01010	- 6 -- 11010
+11 -- 01011	- 5 -- 11011
+12 -- 01100	- 4 -- 11100
+13 -- 01101	- 3 -- 11101
+14 -- 01110	- 2 -- 11110
+15 -- 01111	- 1 -- 11111

Immediate Long

When the UT69R000 uses the Immediate Long mode, the source operand is a 16-bit data value. The UT69R000 explicitly states this data for all instructions and encodes the 16-bit data in a second 16-bit instruction word (figure 32). The UT69R000 encodes the 5-bit field of the instruction source field, bits 4 through 0, as follows:

IMM -- 11111

Special Source Operand Addressing Modes
In addition to its three direct addressing modes, the UT69R000 also supports three modes of indirect addressing: (1) Data Register Indirect; (2) Stack Pointer Indirect; and (3) Absolute.

Data Register Indirect
When the UT69R000 uses the Data Register Indirect mode, the source operand is a memory location addressed by the contents of the specified data register. The data register is explicitly stated for all instructions. This mode is only available on the LR, STR, INR, and STR instructions. The UT69R000 encodes a 5-bit field, bits 4 through 0, in each instruction as follows:

R0 -- 00000	XR0 -- 10000
R1 -- 00001	R16 -- 10001
R2 -- 00010	XR2 -- 10010
R3 -- 00011	R17 -- 10011
R4 -- 00100	XR4 -- 10100
R5 -- 00101	XR16 -- 10101
R6 -- 00110	XR6 -- 10110
R7 -- 00111	
R8 -- 01000	XR8 -- 11000
R9 -- 01001	R18 -- 11001
R10 -- 01010	XR10 -- 11010
R11 -- 01011	R19 -- 11011
R12 -- 01100	XR12 -- 11100
R13 -- 01101	XR18 -- 11101
R14 -- 01110	XR14 -- 11110
R15 -- 01111 and 11111	Reserved -- 10111

Stack Pointer Indirect
When the UT69R000 uses the Stack Pointer Indirect mode, the source operand is a memory location addressed by the contents of the Stack Pointer (SP) register. This mode is only available with POP and PUSH instructions. The UT69R000 encodes a 5-bit field, bits 11 through 15, of each instruction when in the Stack Pointer Indirect mode as follows:

SP -- 10111.

Absolute
When the UT69R000 uses the Absolute mode, the source operand is the memory location addressed by the contents of the 16-bit immediate-data field accompanying the instruction. This mode is only available on the LR, STR, INR, and OTR instructions. The system programmer encodes the immediate data field as a second 16-bit instruction word.

9.4 Data Movement Operations
The UT69R000 places no restrictions on operand size during data movement. This means the size (Byte, Word, or Long Word) of the data in the source and destination do not have to match. The UT69R000 handles the data movement for all instructions.

When a instruction specifies a word destination, a 16-bit result is always stored in the destination. If the instruction specifies a 5-bit literal source operand, then the UT69R000 sign-extends this source data to produce a 16-bit operand. If the instruction specifies a word-length source operand, there is no manipulation of the source data. If the instruction specifies a long-word source operand, the UT69R000 only retains the least significant 16 bits of the result. The UT69R000 truncates the most significant 16 bits of the result.

When a instruction specifies a long-word destination, a 32-bit result is always stored in the destination. If the instruction specifies a 5-bit literal source operand, then the UT69R000 sign-extends this source data to produce a 32-bit operand. If the instruction specifies a word-length source operand, then the UT69R000 also sign-extends this source data to produce a 32-bit operand. If the instruction specifies a long-word-length source operand, there is no manipulation of the source data.

When the system programmer specifies a byte instruction, the UT69R000 only stores eight bits of the result regardless of whether the instruction specifies a word or long-word destination register.

Operation Code Matrix
The UT69R000 performs 30 basic operations, each with its own operation code. All the UT69R000's operations are explicit, and are encoded in bits 14 through 10 of the instruction.

10.0 PIN DESCRIPTION

Legend for TYPE and ACTIVE fields:

TO = TTL output
TI = TTL input
TUI = TTL input (pull-up)
TDI = TTL input (pull-down)
TTO = Three-state TTL output
TTB = Three-state TTL bidirectional
CO = CMOS output
OSC = Oscillator input to a Pierce Oscillator inverter
AH = Active High
AL = Active Low

OSCILLATOR AND CLOCK SIGNALS

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
OSCIN	50	P14	OSC	--	Oscillator Input. A 50% duty cycle crystal-drive input for driving the UT69R000.
OSCOUT	51	P15	CO	--	Oscillator Output. A 50% duty cycle, single-phase clock output at the same frequency as the OSCIN input.
SYSCLK	52	M14	TO	--	System Output. The buffered equivalent of the OSCOUT signal.

PROCESSOR STATUS

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
NUI1	129	H2	TI	--	Not used input 1. Internal UTMC use only. Tie either high or low.
NUI2	44	P12	TUI	--	Not used input 2. Internal UTMC use only. Tie low.
NUI4	61	K15	TUI	--	Not used input 4. Internal UTMC use only. Tie high.
NUO3	126	G3	TTO	--	Not used output 3. Internal UTMC use only. NUO3 enter high impedance state when the UT69R000 is in the test mode ($\overline{TEST}$=0)
NUI3	45	N11	TDI	--	Not used input 3. Internal UTMC use only. Tie low.
$\overline{STATE1}$	54	N15	TTO	--	Processor State. This signal indicates the internal state of the UT69R000. A low on $\overline{STATE1}$ indicates the UT69R000 is executing a new instruction. A high on $\overline{STATE1}$ indicates the UT69R000 is fetching an instruction. $\overline{STATE1}$ enters a high-impedance state when the UT69R000 is in the test mode ($\overline{TEST}$=0).

OPERAND DATA BUS ARBITRATION

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
$\overline{BRQ}$	118	D2	TTO	AL	Bus Request. The UT69R000 asserts this signal to indicate it is requesting control of the Operand data bus (D0 - D15). $\overline{BRQ}$ enters a high-impedance state when the UT69R000 is in the test mode ($\overline{TEST}$ = 0).
$\overline{BGNT}$	119	E3	TUI	AL	Bus Grant. When asserted, this signal indicates the UT69R000 may take control of the Operand data bus. It is tied to an internal pull-up resistor.
$\overline{BUSY}$	120	C1	TUI	AL	Bus Busy. A bus master asserts this input to inform the UT69R000 that another bus master is using the Operand data bus. It is tied to an internal pull-up resistor.
$\overline{BGACK}$	117	B1	TTO	AL	Bus Grant Acknowledge Output. The UT69R000 asserts this signal to indicate it is the current bus master. When low, $\overline{BGACK}$ inhibits other devices from becoming the bus master. When the UT69R000 relinquishes control of the bus, $\overline{BGACK}$ enters a high-impedance state.

OPERAND DATA BUS CONTROL

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
$\overline{DTACK}$	121	E2	TUI	AL	Data Transfer Acknowledge. This signal tells the UT69R000 that a data transfer has been acknowledged and the UT69R000 can complete the bus cycle. To assure the UT69R000 operates with no wait states, $\overline{DTACK}$ can be tied low. $\overline{DTACK}$ is tied to an internal pull-up resistor.
M/$\overline{IO}$	112	B3	TTO	--	Memory or I/O. Indicates whether the current bus cycle is for memory (high) or I/O (low). It remains in the high-impedance state during bus cycles when the UT69R000 does not control the Operand buses.
R/$\overline{WR}$	114	C4	TTO	--	Read/Write. Indicates the direction of data flow with respect to the UT69R000. R/$\overline{WR}$ high means the UT69R000 is attempting to read data from an external device, and R/$\overline{WR}$ low means the UT69R000 is attempting to write data to an external device. R/$\overline{WR}$ remains in a high-impedance state when the UT69R000 does not control the Operand buses.

Continued on page 41.

OPERAND DATA BUS CONTROL *Continued from page 40.*

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
$\overline{DS}$	116	B2	TTO	AL	Data Strobe. Indicates valid data is on the Operand Data bus. The UT69R000 places $\overline{DS}$ in a high-impedance state when it does not control the Operand buses.

INSTRUCTION MEMORY CONTROL

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
$\overline{OE}$	42	R12	TTO	AL	Output Enable Instruction Memory. This signal allows memory to place data on the instruction data bus. The Store Register to Instruction Memory (STRI) instruction removes $\overline{OE}$ during the CK2 internal clock cycle. $\overline{OE}$ enters a high-impedance state when the UT69R000 is in the test mode ($\overline{TEST}$ = 0).
$\overline{WE}$	43	R13	TTO	AL	Write Enable Memory. This signal allows the UT69R000 to write to instruction memory. The Store Register to Instruction Memory (STRI) instruction asserts $\overline{WE}$ during the CK2 internal clock cycle. $\overline{WE}$ enters a high-impedance state when the UT69R000 is in the test mode ($\overline{TEST}$ = 0).

UART CONTROL/TIMER CLOCK

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
UARTIN	127	F1	TUI	AH	UART Input. The UT69R000 receives serial data through this input. The serial data is stored in the UT69R000's Receiver Buffer Register (RCVR). It is tied to an internal pull-up resistor.
UARTOUT	128	G1	TTO	AH	UART Output. The serial data stored in the UT69R000's Transmitter Buffer Register (TXMT) is transmitted through this output. The UART output is fixed at 9600 baud, with eight data bits, odd-parity, and one stop bit. UARTOUT enters a high-impedance state when the UT69R000 is in the test mode ($\overline{TEST}$=0). (9600 baud @ TIMCLK = 12 MHz)

Continued on page 42.

UART CONTROL/TIMER CLOCK *Continued from page 41*

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
TIMCLK	53	L13	TI	--	Timer Clock. This 12 MHz clock input generates the baud rate for the UT69R000's internal UART. The input also provides the clock for the UT69R000's two internal timers (TIMER A and TIMER B).
DI2	48	N12	TDI	--	Discrete Input 2. Asserting this input sets bit 3 in the System Status Register Bit 3 is read with the Input Register Instruction (INR). Tied to an internal pull-down resistor. (asynchronous input).
$\overline{\text{TEST}}$	46	P13	TUI	AL	Test (Input). Asserting this input places the UT69R000 into a test mode. In this mode, all the UT69R000's outputs, except OSCOUT and SYSCLK, enter a high-impedance state. When using $\overline{\text{TEST}}$, the UT69R000 must have a $\overline{\text{MRST}}$. $\overline{\text{MRST}}$ must be held active for at least one SYSCLK period after $\overline{\text{TEST}}$ is deasserted to assure proper operation (see figure 42b). $\overline{\text{TEST}}$ is tied to an internal pull-up resistor.
DI1	49	N13	TDI	--	Discrete Input 1. Asserting this input sets bit 8 in the System Status Register. Bit 8 is read with the Input Register Instruction (INR). Tie to a internal pull-down resistor. (asynschronous input).

PROCESSOR MODE

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
OD0 OD1 OD2 OD3 OD4 OD5 OD6 OD7	104 105 106 107 108 109 110 111	B7 B6 C6 A5 A4 A3 B4 C5	TTO	--	Output Discrete Bus (OD(7:0)). These outputs reflect the status of bits 0 through 7 of the Status/ Output Discrete Register. Write to this register using Output Register Instruction (OTR). Outputs enter a high-impedance state when the UT69R000 is placed in the test mode ($\overline{\text{TEST}}$ = 0).

INTERRUPTS/EXCEPTIONS

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
MCHNE1	125	G2	TUI	AH	System Fault. This positive edge-triggered input sets bit 8 (MCHNE1) in the UT69R000's Fault Register. Under no circumstances should MCHNE1 be tied in its active state. It is tied to an internal pull-up resistor. Interrupt is not cleared via software until the negation of the input signal.
$\overline{\text{BTERR}}$	122	D1	TUI	AL	Bus Time Error. It is asserted when a bus error or a timeout occurs. During I/O bus cycles, an active $\overline{\text{BTERR}}$ sets bit 10 of the Fault Register. During Memory bus cycles, an active $\overline{\text{BTERR}}$ sets bit 7 of the Fault Register. Under no circumstances should $\overline{\text{BTERR}}$ be tied in its active state. It is tied to an internal pull-up resistor. Interrupt is not cleared via software until the negation of the input signal.
MCHNE2	124	F2	TDI	AH	Memory Parity (Error). Asserting this input indicates a machine error. Bit 13 of the UT69R000's Fault Register, is set when MCHNE2 is active. Under no circumstances should MCHNE2 be tied in its active state. It is tied to an internal pull-down resistor. Interrupt is not cleared via software until the negation of the input signal.
MPROT	123	F3	TUI	AH	Memory Protect Fault. When asserted, it informs the UT69R000 that a memory-protect fault has occurred on the Operand Data Bus. An access fault, a write-protect fault, or an execute-protect fault causes a memory-protect fault. If the UT69R000 is using the bus and MPROT is asserted, bit 15 of the Fault Register (CPU Fault) is set. If the UT69R000 is not using the bus and MPROT is asserted, bit 14 of the Fault Register (DMA Error) is set. It is tied to an internal pull-up resistor. Interrupt is not cleared via software until the negation of the input signal.
$\overline{\text{INT0}}$ $\overline{\text{INT1}}$ $\overline{\text{INT2}}$ $\overline{\text{INT3}}$ $\overline{\text{INT4}}$ $\overline{\text{INT5}}$ $\overline{\text{INT6}}$	56 57 58 59 60 62 63	M15 K13 K14 J14 J13 J15 H14	TUI	AL	User Interrupts. These interrupts are active on a negative-going edge and each will set, when active, its associated bit in the Pending Interrupt Register. The interrupts are maskable by setting the associated bits in the Interrupt Mask Register. Asserting $\overline{\text{MRST}}$ resets all interrupts. They are tied to an internal pull-up resistor.
$\overline{\text{PFAIL}}$	55	L14	TUI	AL	Power Fail (Interrupt). Asserting this input informs the UT69R000 that a power failure has occurred and the present process will be interrupted. This input sets bit 15 in the Pending Interrupt Register. A Power Fail Interrupt (bit 15) cannot be disabled or masked. It is tied to an internal pull-up resistor.
$\overline{\text{MRST}}$	47	R14	TUI	AL	Master Reset. This input initializes the UT69R000 to a reset state. The UT69R000 must be reset after power (Vcc) is within specification and stable to ensure proper operation. The system must hold $\overline{\text{MRST}}$ active for at least one period of SYSCLK to assure the UT69R000 will be reset. It is tied to an internal pull-up resistor.

OPERAND BUSES

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
A0	84	A14	TTO	--	Address Bus - Operand. When asserted, this bus is unidirectional and represents the Operand Address. The bus is in the high-impedance state when the UT69R000 does not control the bus. A15 is the most significant bit. The Operand Address enters a high-impedance state when the UT69R000 is in the test mode ($\overline{TEST}$ = 0).
A1	85	B12			
A2	86	C11			
A3	87	A13			
A4	88	B11			
A5	89	A12			
A6	90	C10			
A7	91	B10			
A8	92	B9			
A9	93	C9			
A10	94	A10			
A11	95	A9			
A12	96	B8			
A13	97	A8			
A14	102	A7			
A15	103	A6			
D0	64	H15	TTB	--	Data Bus - Operand. This bidirectional data bus remains in a high-impedance state when the UT69R000 does not control the bus. D15 is the most significant bit. The Operand Data Bus enters a high-impedance state when the UT69R000 is in the test mode ($\overline{TEST}$ = 0).
D1	69	G15			
D2	70	F15			
D3	71	G14			
D4	72	F14			
D5	73	F13			
D6	74	E15			
D7	75	D15			
D8	76	C15			
D9	77	D14			
D10	78	E13			
D11	79	C14			
D12	80	B15			
D13	81	D13			
D14	82	C13			
D15	83	B14			

INSTRUCTION BUSES

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
RA0	18	R2	TTO	--	Instruction Address Bus. This unidirectional bus represents the address of the data in instruction memory. RA19 is the most significant bit. The address enters a high-impedance state only when the UT69R000 is in the test mode ($\overline{TEST}$ = 0).
RA1	19	P4			
RA2	20	N5			
RA3	21	R3			
RA4	22	P5			
RA5	23	R4			
RA6	24	N6			
RA7	25	P6			
RA8	26	P7			
RA9	27	N7			
RA10	28	R6			
RA11	29	R7			
RA12	30	P8			
RA13	31	R8			
RA14	36	R9			
RA15	37	R10			
RA16	38	P9			
RA17	39	P10			
RA18	40	N10			
RA19	41	R11			

Continued on page 45.

INSTRUCTION BUSES *Continued from page 44.*

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
RD0	130	H1	TTB	--	Instruction Data Bus. This bidirectional data bus is the interface with the memory. RD15 is the most significant bit. The Data Bus enters a high-impedance state only when the UT69R000 is in the test mode ($\overline{TEST}$ = 0).
RD1	3	J1			
RD2	4	K1			
RD3	5	J2			
RD4	6	K2			
RD5	7	K3			
RD6	8	L1			
RD7	9	M1			
RD8	10	N1			
RD9	11	M2			
RD10	12	L3			
RD11	13	N2			
RD12	14	P1			
RD13	15	M3			
RD14	16	N3			
RD15	17	P2			

POWER AND GROUND

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
V_{DD}	34	H3	--	--	+5 VDC Power. Power supply input.
	67	N9			
	100	G13			
	132	C7			
V_{SS}	1	J3	--	--	Reference Ground. Zero Volts DC, logic ground.
	33	N8			
	66	H13			
	99	C8			

11.0 ABSOLUTE MAXIMUM RATINGS (1)
(Referenced to V_{SS})

SYMBOL	PARAMETER	LIMITS	UNIT
V_{DD}	DC supply voltage	−0.3 to +7.0	V
$V_{I/O}$	Voltage on any pin	−0.3 to V_{DD} + .3	V
I_I	DC input current	±10	mA
T_{STG}	Storage temperature	−65 to +150	°C
I_{LU}	Latchup immunity (2)	±150	mA
P_D	Maximum power dissipation	600	mW
T_J	Maximum junction temperature	+175	°C
ΘQ_{JC}	Thermal resistance, junction-to-case (3)	10	°C/W

Notes:
1. Stresses outside the listed absolute maximum ratings may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions beyond limits indicated in the operational sections of this specification is not recommended. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. See discussion of test technique (figure 42).
3. Test per MIL-STD-883, Method 1012.

12.0 RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	LIMITS	UNIT
V_{DD}	DC supply voltage	4.5 to 5.5	V
T_C	Temperature range	−55 to +125	°C
V_{IN}	DC input voltage	0 to V_{DD}	V

13.0 ELECTRICAL CHARACTERISTICS

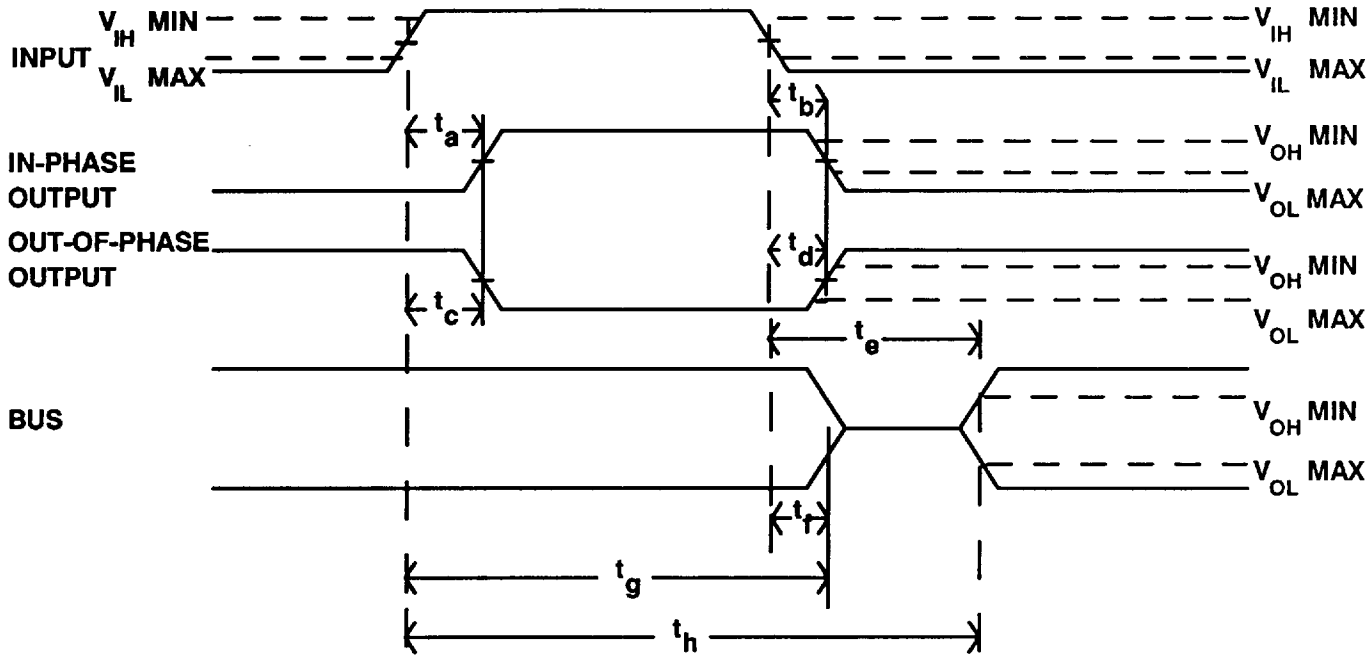
V_{DD} = 5.0V ± 10%; −55 C° < T_C < + 125 C°)

SYMBOL	PARAMETER	CONDITION	MINIMUM	MAXIMUM	UNIT
V_{IL}	Low-level input voltage TTL inputs			0.8	V
V_{IH}	High-level input voltage TTL inputs		2.2		V
I_{IN}	Input leakage current TTL inputs Inputs with pull-down resistors Inputs with pull-up resistors	 $V_{IN} = V_{DD}$ or V_{SS} $V_{IN} = V_{DD}$ $V_{IN} = V_{SS}$	 −10 150 −900	 10 900 −150	 µA µA µA
V_{OL}	Low-level output voltage TTL outputs CMOS outputs	I_{OL} = 3.2 mA I_{OL} = 6.4 mA Note 6. I_{OL} = 1µA		0.4 0.4 0.05	V V V
V_{OH}	High-level output voltage TTL outputs CMOS outputs	I_{OH} = −400 µA I_{OH} = −800 µA Note 6. I_{OH} = −1µA	2.4 2.4 V_{DD} − 0.05		V V V
I_{OZ}	Three-state output leakage current TTL outputs	$V_O = V_{DD}$ or V_{SS}	 −10 −20 Note 6.	 +10 +20 Note 6.	 µA
I_{OS}	Short-circuit output current (2,3)	V_{DD} = 5.5 V, $V_O = V_{DD}$ V_{DD} = 5.5 V, V_O = 0 V	 −100 −200 Note 6.	100 200 Note 6.	mA mA mA mA
C_{IN}	Input capacitance (3)	F = 1 MHz @ 0 V		10	pF
C_{OUT}	Output capacitance (3)	F = 1 MHz @ 0 V		15	pF
C_{IO}	Bidirect I/O capacitance (3)	F = 1 MHz @ 0 V		20	pF
I_{DD}	Average operating current (1,5)	F = 16 MHz, C_L = 50 pF F = 12 MHz, C_L = 50 pF		75 50	mA mA
Q_{IDD}	Quiescent current	Note 4.		1	mA

Notes:

1. Supplied as a design limit, but not guaranteed or tested.
2. Not more than one output may be shorted at a time for a maximum duration of one second.
3. Measured only for initial qualification, and after process or design changes that could affect input/output capacitance.
4. All inputs with internal pull-ups or pull-downs should be left open circuit, all other inputs tied low or high. $\overline{TEST}$ input pin asserted.
5. Includes current through input pull-ups. Instantaneous surge currents on the order of 1 ampere can occur during output switching. Voltage supply should be adequately sized and decoupled to handle a large current surge.
6. Double buffer output pins (i.e., $\overline{DS}$, R/$\overline{WR}$, M/$\overline{IO}$,).

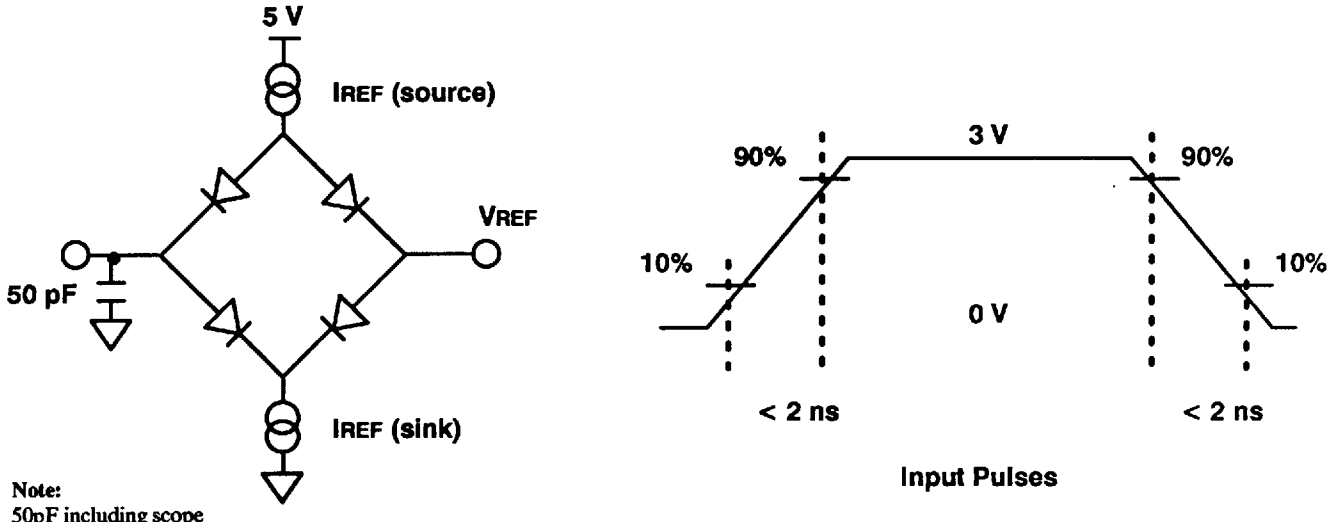
14.0 AC ELECTRICAL CHARACTERISTIC



SYMBOL	PARAMETER
t_a	INPUT ↑ to response ↑
t_b	INPUT ↓ to response ↓
t_c	INPUT ↑ to response ↓
t_d	INPUT ↓ to response ↑
t_e	INPUT ↓ to data valid
t_f	INPUT ↓ to high Z
t_g	INPUT ↑ to high Z
t_h	INPUT ↑ to data valid

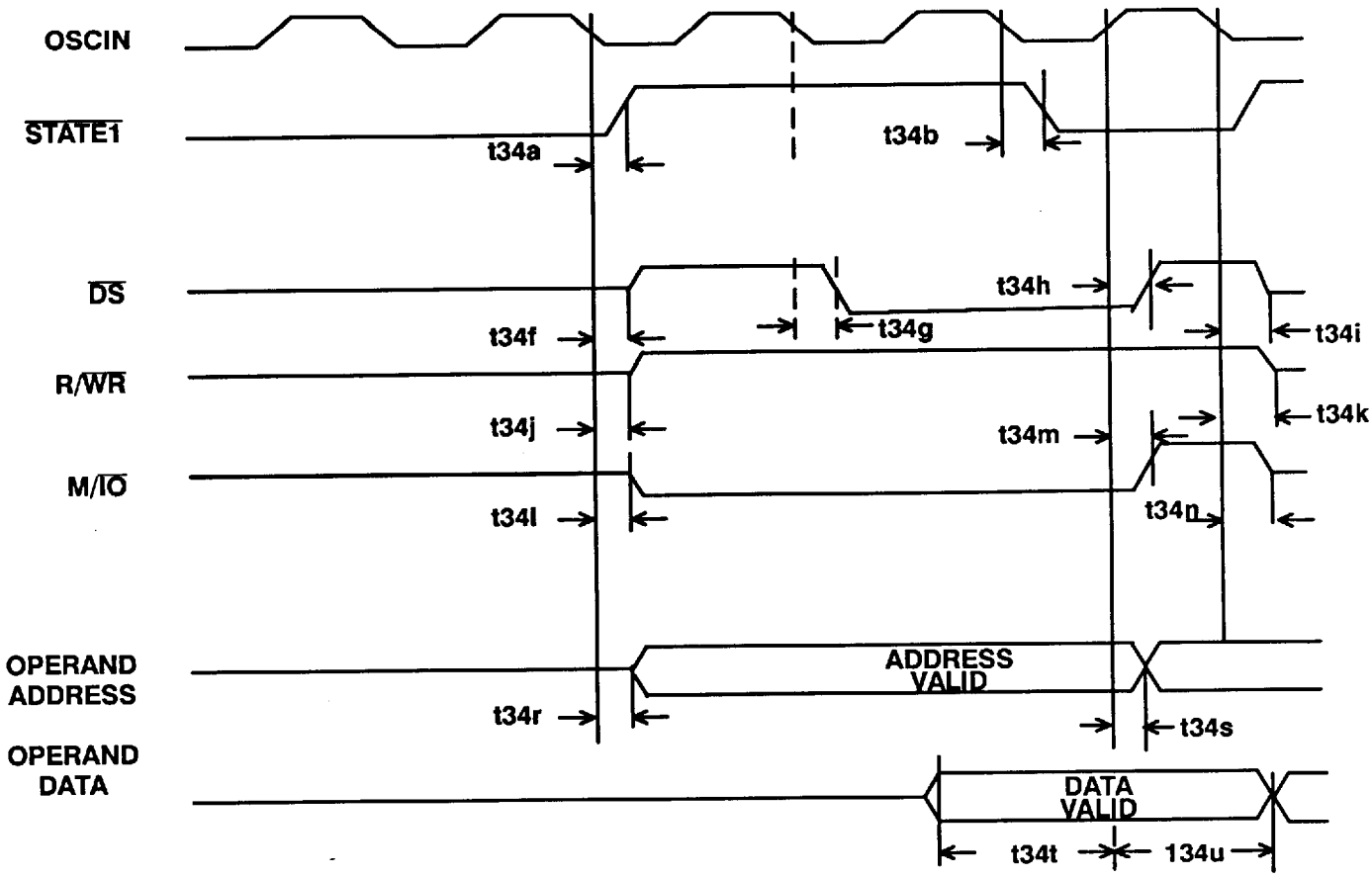
*Unless otherwise noted, all AC electrical characteristics are guaranteed by design or characterization.

Figure 32a. Typical Timing Measurements



Note:
50pF including scope probe and test socket.

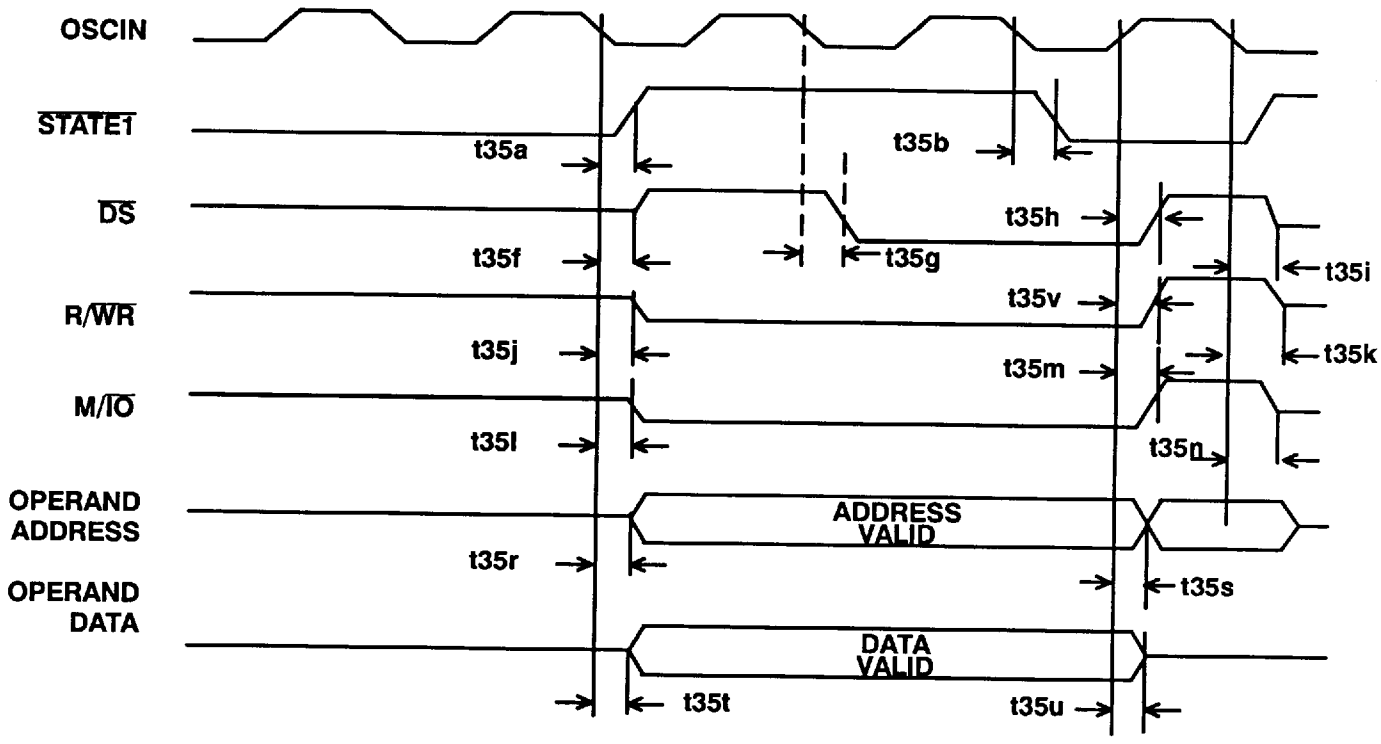
Figure 32b. AC Test Loads and Input Waveforms



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t34a *	OSCIN low to STATE1 high	0	42	0	33	ns
t34b *	OSCIN low to STATE1 low	0	39	0	33	ns
t34f *	OSCIN low to DS inactive	0	54	0	45	ns
t34g *	OSCIN low to DS active	0	37	0	35	ns
t34h *	OSCIN high to DS inactive	0	50	0	38	ns
t34i	OSCIN low to DS high Z	--	50	--	38	ns
t34j *	OSCIN low to R/WR active	0	54	0	41	ns
t34k	OSCIN low to R/WR high Z	--	50	--	38	ns
t34l *	OSCIN low to M/IO low	0	51	0	42	ns
t34m*	OSCIN high to M/IO high	0	73	0	55	ns
t34n	OSCIN low to M/IO high Z	--	50	--	38	ns
t34r *	OSCIN low to address valid	0	57	0	45	ns
t34s	OSCIN high to address invalid	--	55	--	41	ns
t34t	Data setup time	0	--	0	--	ns
t34u	Data hold time	34	--	26	--	ns

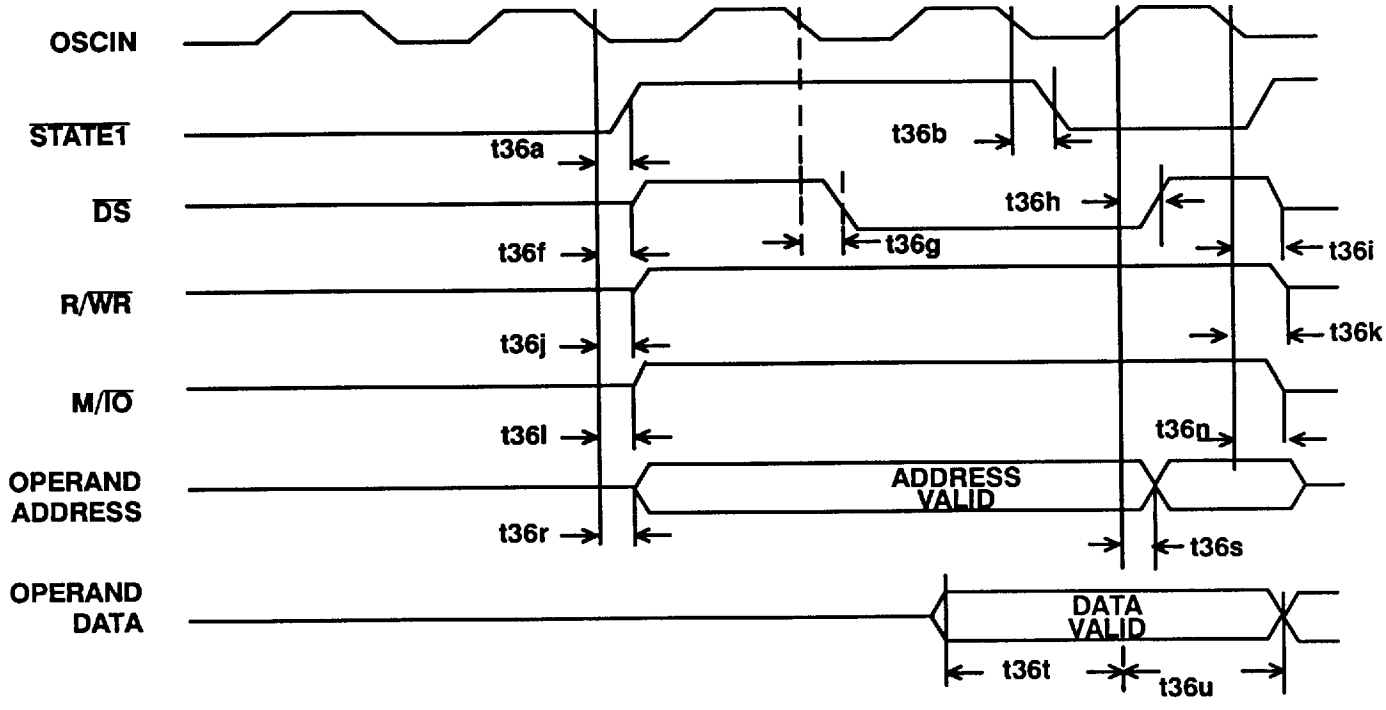
Note:
*Guaranteed by test.

Figure 33. I/O Read Cycle



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t35a *	OSCIN low to STATE1 high	0	42	0	33	ns
t35b *	OSCIN low to STATE1 low	0	39	0	33	ns
t35f *	OSCIN low to DS inactive	0	54	0	45	ns
t35g *	OSCIN low to DS active	0	37	0	35	ns
t35h *	OSCIN high to DS inactive	0	50	0	38	ns
t35i	OSCIN low to DS high Z	–	50	–	38	ns
t35j *	OSCIN low to R/WR inactive	0	51	0	42	ns
t35k	OSCIN low to R/WR high Z	–	50	–	38	ns
t35l *	OSCIN low to M/IO low	0	51	0	42	ns
t35m*	OSCIN high to M/IO high	0	73	0	55	ns
t35n	OSCIN low to M/IO high Z	–	50	–	38	ns
t35r *	OSCIN low to address valid	0	57	0	45	ns
t35s	OSCIN high to address invalid	–	55	–	41	ns
t35t *	OSCIN low to data valid	0	64	0	48	ns
t35u	OSCIN high to data invalid (high Z)	–	80	–	60	ns
t35v *	OSCIN high to R/WR high	0	72	0	54	ns

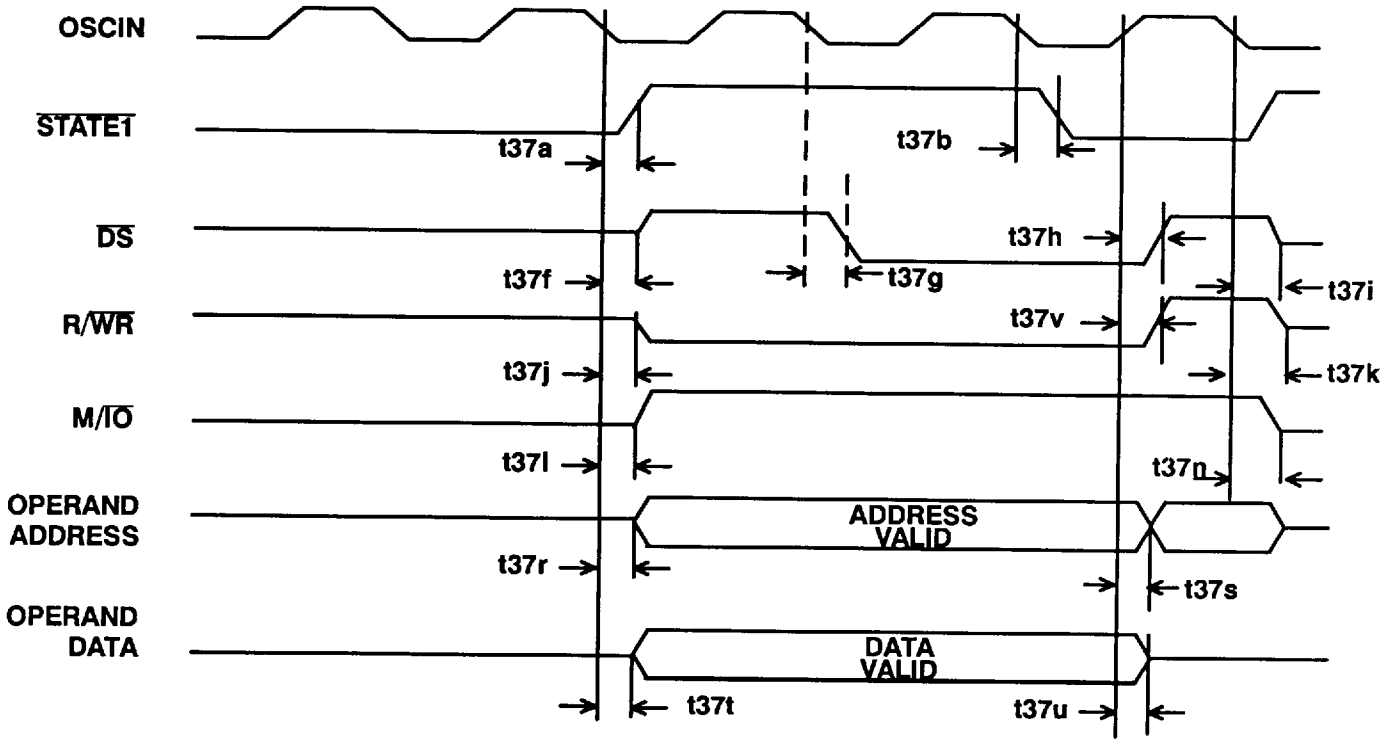
Figure 34. I/O Write Cycle



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t36a *	OSCIN low to STATE1 high	0	42	0	33	ns
t36b *	OSCIN low to STATE1 low	0	39	0	33	ns
t36f *	OSCIN low to DS inactive	0	54	0	45	ns
t36g *	OSCIN low to DS active	0	37	0	35	ns
t36h *	OSCIN high to DS inactive	0	50	0	38	ns
t36i	OSCIN low to DS high Z	--	50	--	38	ns
t36j *	OSCIN low to R/WR inactive	0	54	0	42	ns
t36k	OSCIN low to R/WR high Z	--	50	--	38	ns
t36l *	OSCIN low to M/IO high	0	53	0	42	ns
t36n	OSCIN low to M/IO high Z	--	50	--	38	ns
t36r *	OSCIN low to address valid	0	57	0	45	ns
t36s	OSCIN high to address invalid	--	55	--	41	ns
t36t	Data setup time	0	--	0	--	ns
t36u	Data hold time	34	--	26	--	ns

Note:
*Guaranteed by test.

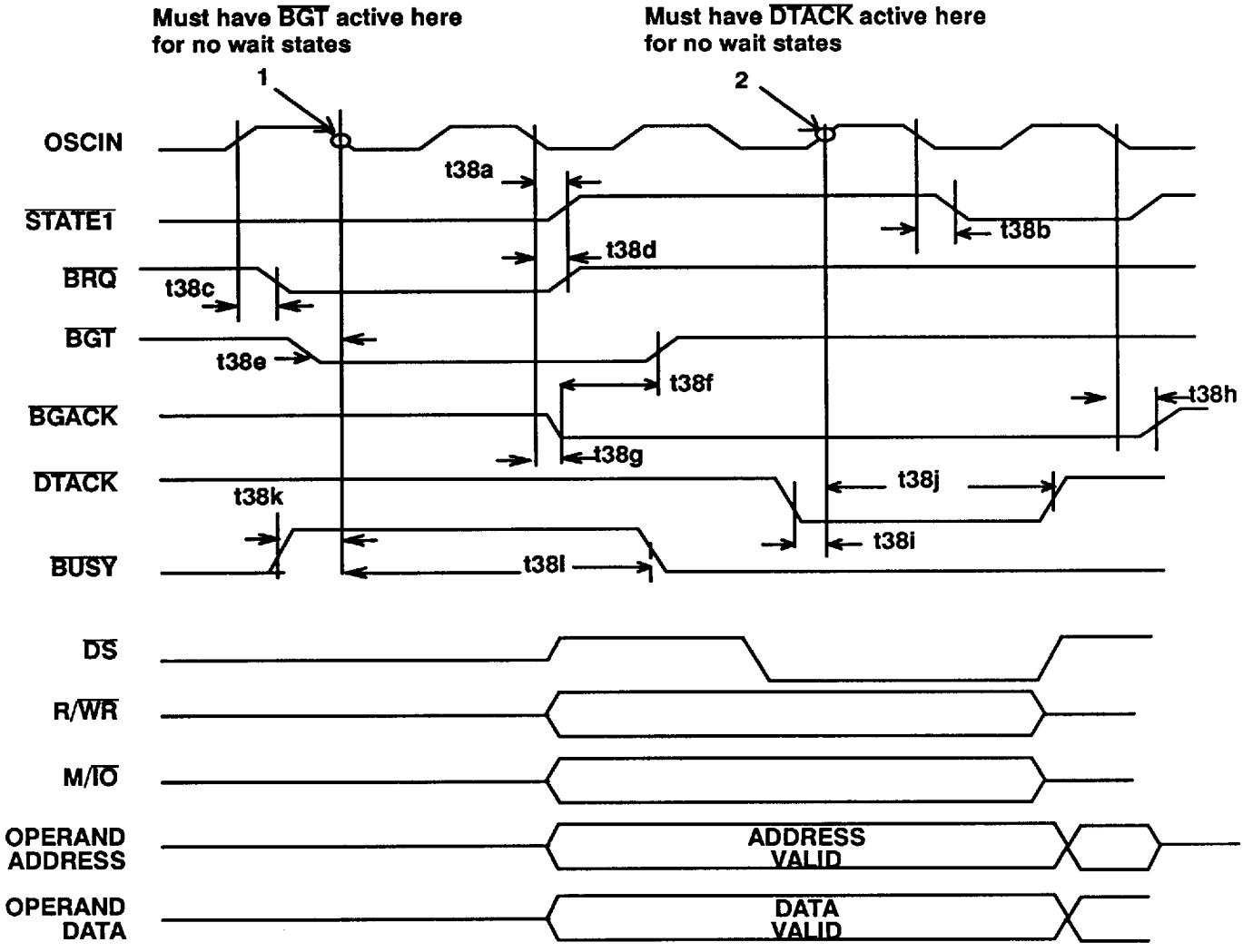
Figure 35. Operand Port Read Cycle



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t37a *	OSCIN low to STATE1 high	0	42	0	33	ns
t37b *	OSCIN low to STATE1 low	0	39	0	33	ns
t37f *	OSCIN low to DS inactive	0	54	0	45	ns
t37g *	OSCIN low to DS active	0	37	0	35	ns
t37h *	OSCIN high to DS inactive	0	50	0	38	ns
t37i	OSCIN low to DS high Z	–	50	–	38	ns
t37j *	OSCIN low to R/WR active	0	51	0	42	ns
t37k	OSCIN low to R/WR high Z	–	50	–	38	ns
t37l *	OSCIN low to M/IO high	0	53	0	42	ns
t37n	OSCIN low to M/IO high Z	–	50	–	38	ns
t37r *	OSCIN low to address valid	0	57	0	45	ns
t37s	OSCIN high to address invalid	–	55	–	41	ns
t37t *	OSCIN low to data valid	0	64	0	48	ns
t37u	OSCIN high to data invalid (high Z)	–	80	–	60	ns
t37v *	OSCIN high to R/WR high	0	72	0	54	ns

Note:
*Guaranteed by test.

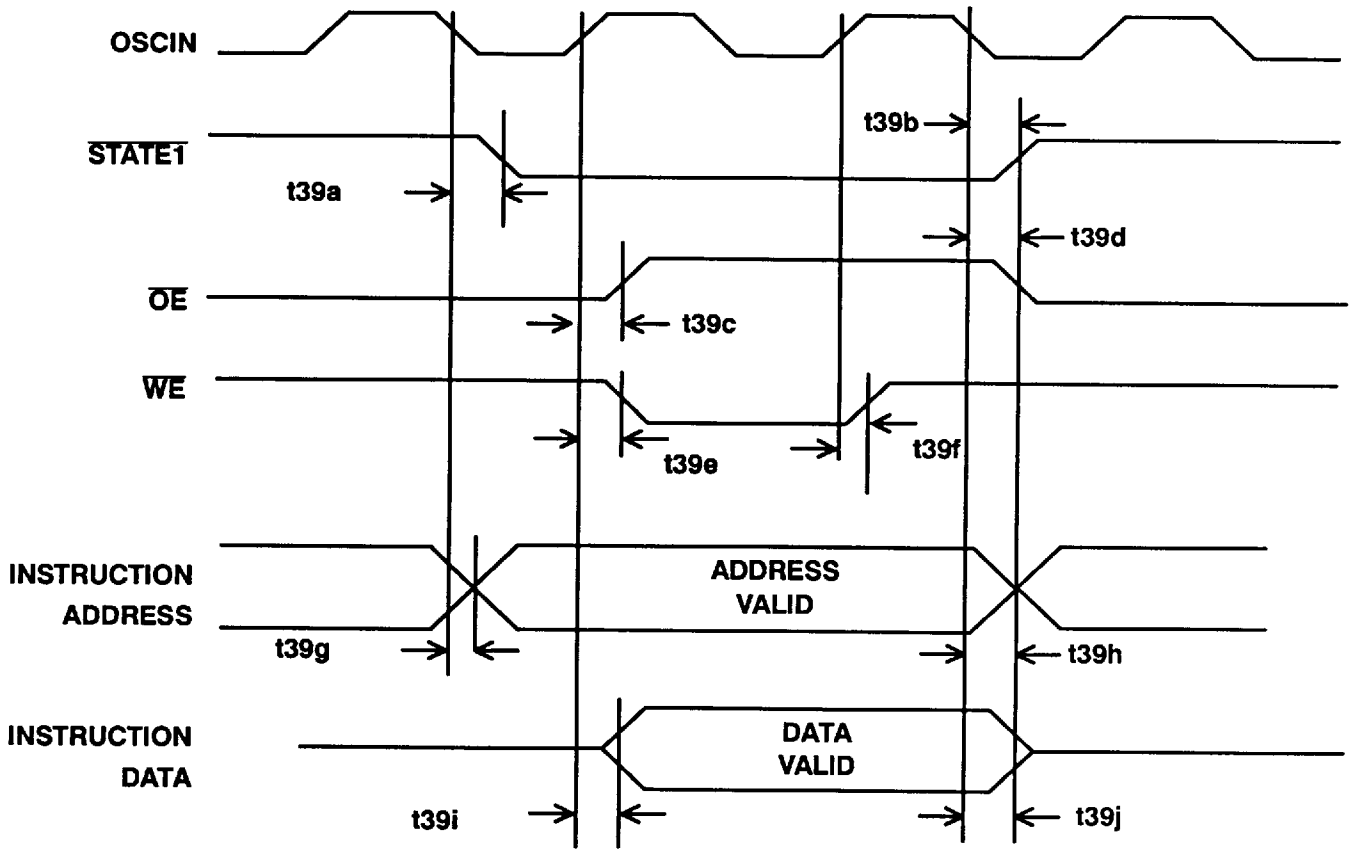
Figure 36. Operand Port Write Cycle



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t38a *	OSCIN low to STATE1 high	0	42	0	33	ns
t38b *	OSCIN low to STATE1 low	0	39	0	33	ns
t38c *	OSCIN high to BRQ low	0	54	0	41	ns
t38d *	OSCIN low to BRQ high	0	58	0	44	ns
t38e	BGT setup time	15	--	15	--	ns
t38f	BGT hold time	0	--	0	--	ns
t38g *	OSCIN low to BGACK active	0	53	0	42	ns
t38h	OSCIN low to BGACK high Z	--	55	--	41	ns
t38i	DTACK setup time	10	--	10	--	ns
t38j	DTACK hold time	0	--	0	--	ns
t38k	BUSY setup time	15	--	10	--	ns
t38l	BUSY hold time	10	--	10	--	ns

Notes:
* Guaranteed by test.
1. BGT must be active and BUSY high at this clock edge or wait states will occur.
2. To avoid wait states, DTACK must be active here.

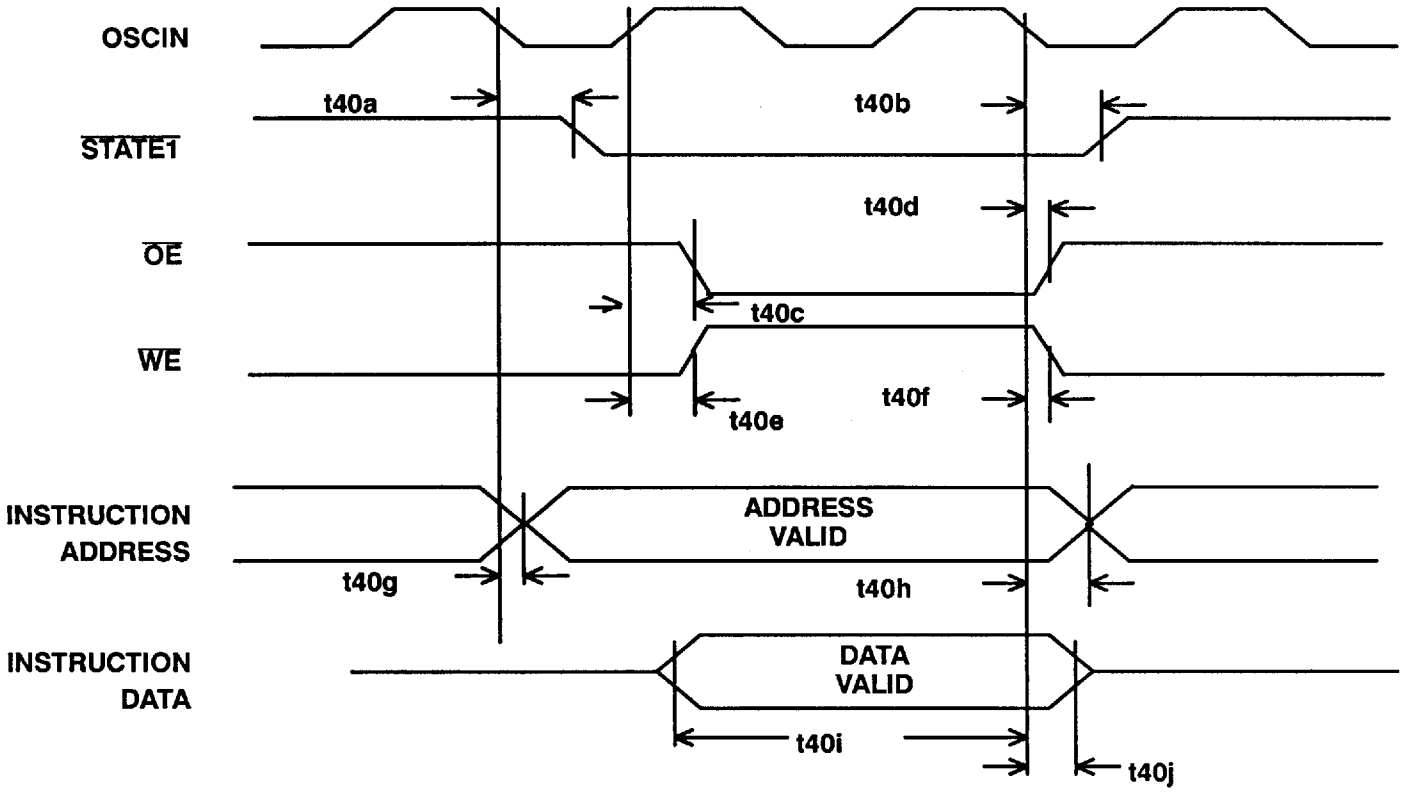
Figure 37. DMA No Wait State



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t39a *	OSCIN low to $\overline{\text{STATE1}}$ low	0	39	0	33	ns
t39b *	OSCIN low to $\overline{\text{STATE1}}$ high	0	42	0	33	ns
t39c *	OSCIN high to $\overline{\text{OE}}$ high	0	52	0	39	ns
t39d *	OSCIN low to $\overline{\text{OE}}$ low	0	46	0	37	ns
t39e *	OSCIN high to $\overline{\text{WE}}$ low	0	50	0	40	ns
t39f *	OSCIN high to $\overline{\text{WE}}$ high	0	49	0	37	ns
t39g *	OSCIN low to address valid	0	65	0	49	ns
t39h	OSCIN low to address high Z	--	50	--	38	ns
t39i	OSCIN high to data valid	--	55	--	41	ns
t39j	OSCIN low to data high Z	--	52	--	39	ns

Note:
*Guaranteed by test.

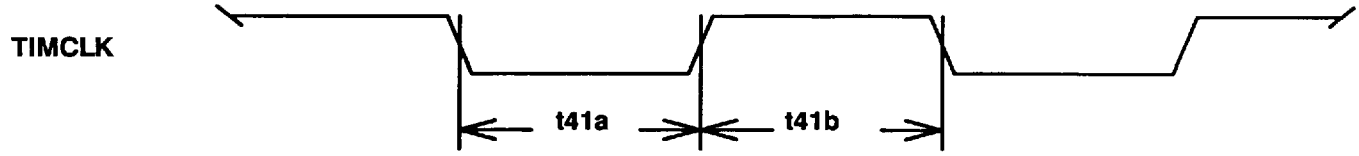
Figure 38. STRI Command Timing



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t40a *	OSCIN low to $\overline{\text{STATE1}}$ low	0	39	0	33	ns
t40b *	OSCIN low to $\overline{\text{STATE1}}$ high	0	42	0	33	ns
t40c	OSCIN high to $\overline{\text{OE}}$ low	0	46	0	35	ns
t40d	OSCIN low to $\overline{\text{OE}}$ high	0	52	0	39	ns
t40e	OSCIN high to $\overline{\text{WE}}$ high	0	49	0	37	ns
t40f	OSCIN low to $\overline{\text{WE}}$ low	0	47	0	35	ns
t40g *	OSCIN low to address valid	0	65	0	49	ns
t40h	OSCIN low to address high Z	--	50	--	38	ns
t40i	Data setup time	0	--	0	--	ns
t40j	Data hold time	27	--	20	--	ns

Note:
*Guaranteed by test.

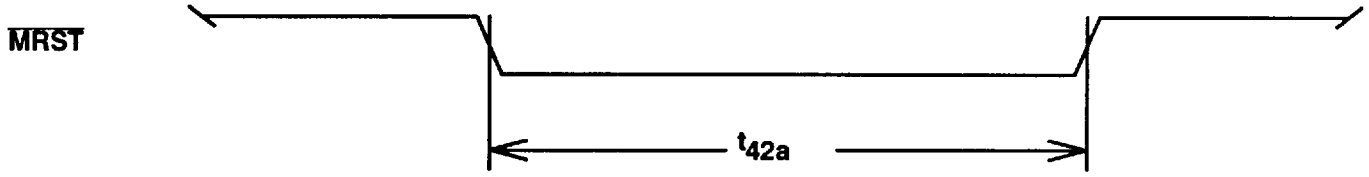
Figure 39. LRI Command Timing



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t41a *	TIMCLK low time	32	--	24	--	ns
t41b *	TIMCLK high time	--	50	--	38	ns

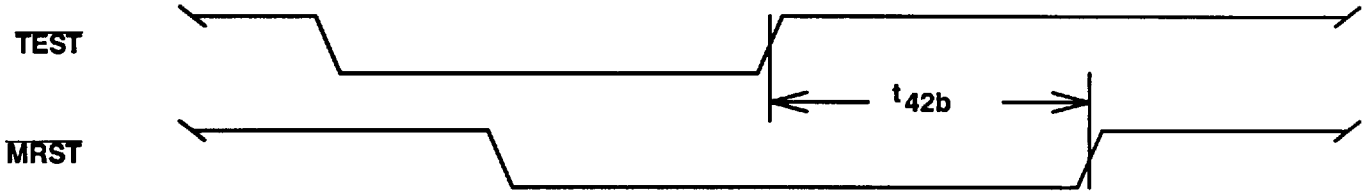
Note:
*Guaranteed by test.

Figure 40. UART and Timer A/B TIMCLK Timing



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t_{42a}	MRST Pulse Width	83	--	62	--	ns

Figure 41a. Master Reset Timing



SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t_{42b}	$\overline{\text{MRST}}$ Timing with TEST active	83	--	62	--	ns

Figure 41b. Master Reset Timing when $\overline{\text{TEST}}$ is Active

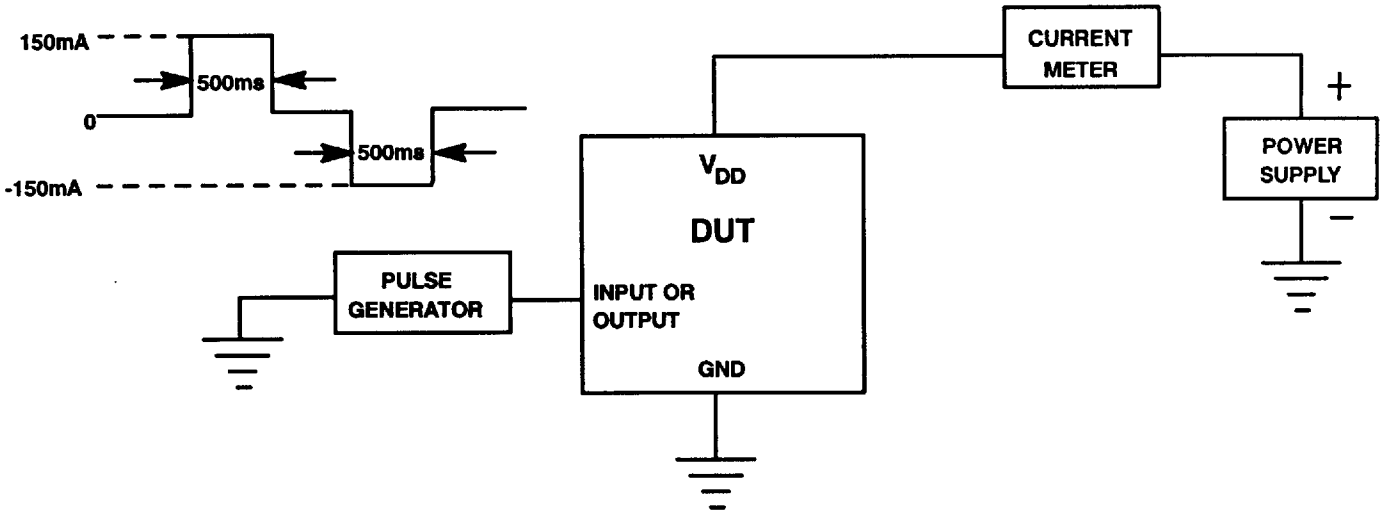


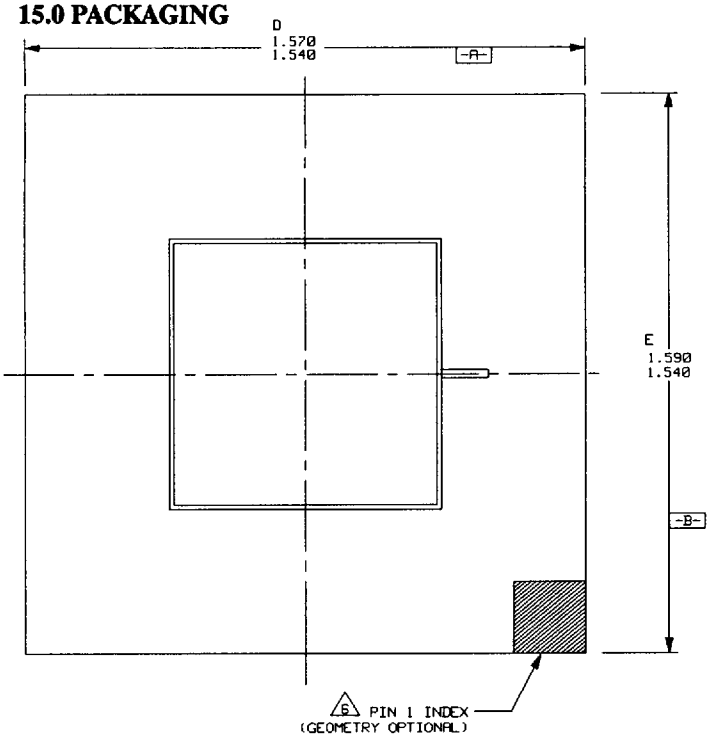
Figure 42. Latchup Test

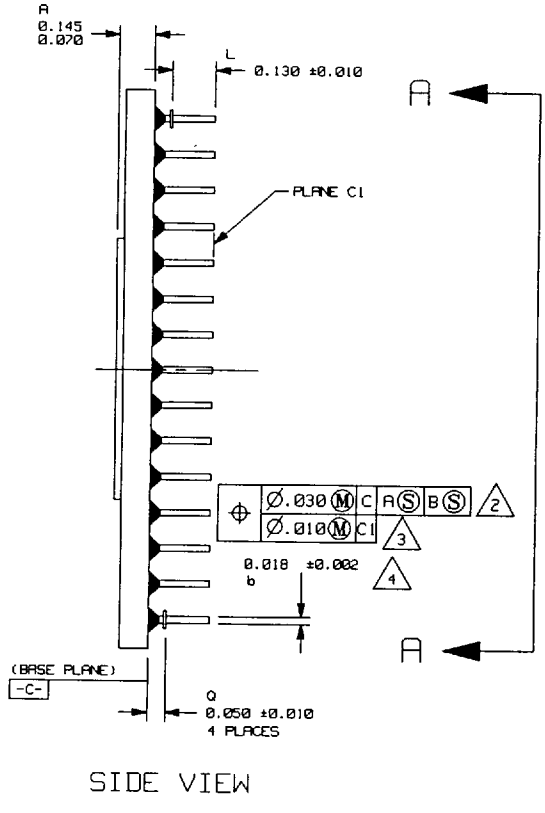
LATCHUP TEST CONFIGURATION

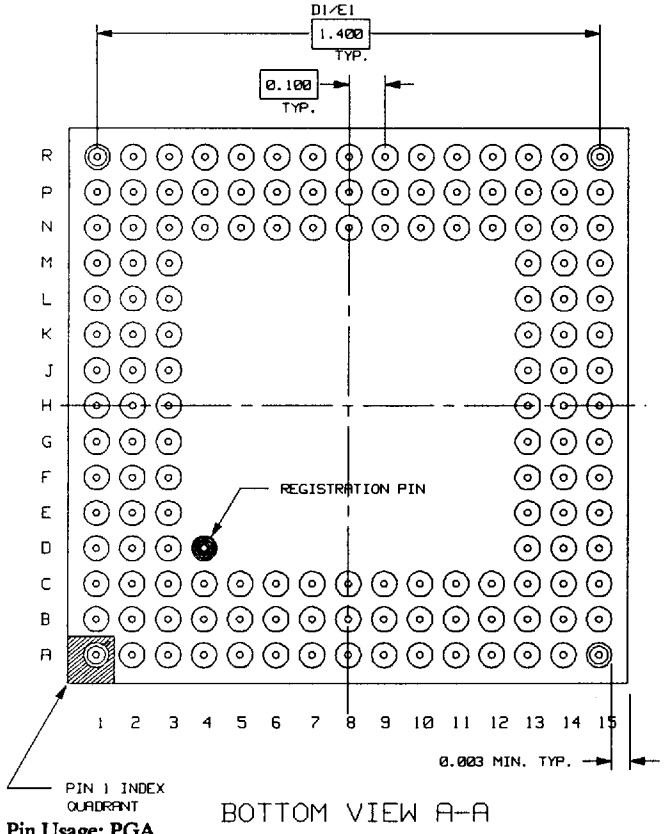
Figure 42 shows the latchup test. VDD holds at +5.5 VDC, and VSS holds at ground. The device test is at 125°C. Each type of I/O alternately receives a positive and then negative 150 mA pulse of 500 ms duration. The current is monitored after the pulse for latchup condition. To prevent burnout, the supply current is limited to 400 mA.

The UT69R000AR has latchup immunity in excess of +150 mA for 500 ms.

15.0 PACKAGING







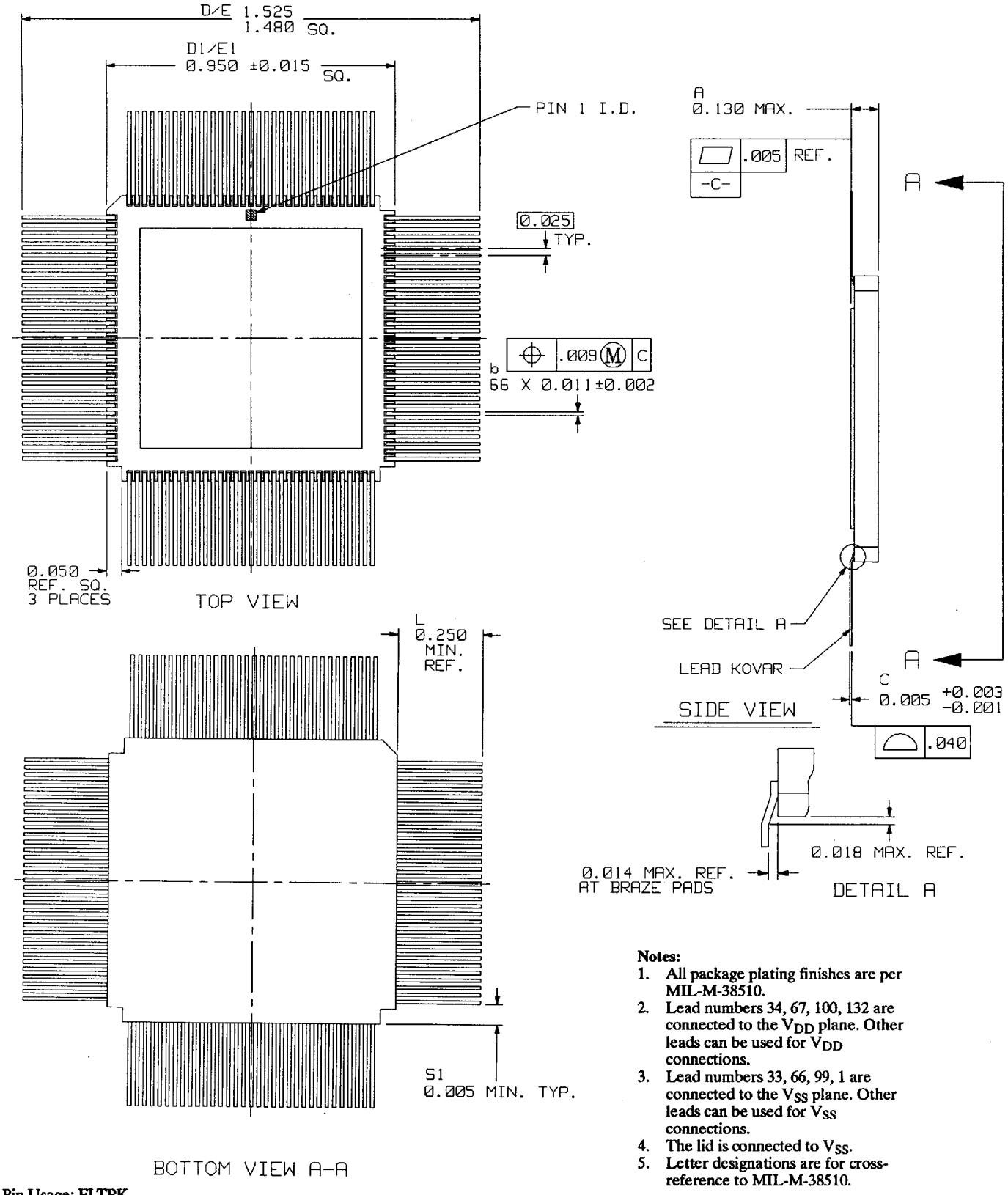
Notes:

1. Package material: Opaque ceramic.
2. True position applies at base plane (Datum C).
3. True position applies at pin tips (Datum C1).
4. All package plating finishes are per MIL-M-38510.
5. Letter designations are for cross-reference MIL-M-38510.
6. Geometry of index mark cannot be an alpha or numeric symbol.
7. All VDD pads are connected to the power plane, die-attach pad and external pins H3, N9, G13, and C7.
8. All VSS pads are connected to the ground plane and external pins J3, N8, H13, and C8.

Pin Usage: PGA

113 - I/O
8 - Power and ground
20 - No connect (B13, C2, N14, P3, R1, D3, M13, A15, E1, A1, L2, N4, R5, B5, P11, A11, C12, E14, R15, L15)

Figure 43. 144-Pin Pingrid Array



Notes:

1. All package plating finishes are per MIL-M-38510.
2. Lead numbers 34, 67, 100, 132 are connected to the V_{DD} plane. Other leads can be used for V_{DD} connections.
3. Lead numbers 33, 66, 99, 1 are connected to the V_{SS} plane. Other leads can be used for V_{SS} connections.
4. The lid is connected to V_{SS}.
5. Letter designations are for cross-reference to MIL-M-38510.

Pin Usage: FLTPK

113 - I/O
8 - Power and ground
8 - No connect (2, 32, 35, 65, 68, 98, 101, 131)

Figure 44. 132-Lead Flatpack (Unformed Leads)

16.0 ORDERING INFORMATION

To order the Standard Military Drawing UT69R000, use the following part number guide: (Check factory for availability)

UT69R000 ** * *

Lead finish
- A = solder
- C = gold finish
- X = any

Package Options
- X = 144-pin PGA
- Y = 132-lead Flatpack - unformed leads

Device Type
- 01 = 12 MHz
- 02 = 16 MHz