Standard Products

UT1750AR RadHard RISC Microprocessor

Data Sheet

May 2003

FEATURES

- Operates in either RISC (Reduced Instruction Set Computer) mode or MIL-STD-1750A mode
- Supports MIL-STD-1750A 32-bit floating-point operations and 48-bit extended-precision floating-point operations on chip
- Built-in 9600 baud UART
- Supports defined MIL-STD-1750A Console Mode of Operation
- Full 64K-word address space. Expandable to 1M words with optional MMU (operand port)
- Register-oriented architecture has 21 user-accessible registers
- Registers may be in 16-bit word or 32-bit double-word configurations
- Built-in multiprocessor bus arbitration and Direct Memory Access support (DMA)
- TTL-compatible I/O
- Stable 1.5-micron CMOS technology
- Full military operating range, -55°C to +125°C, in accordance with MIL-PRF-38535 for Class Q and V
- Typical radiation performance
  - Total dose: 1.0E6 rads(Si)
  - SEL Immune . 100 MeV-cm^2/mg
  - $LET_{TH}(0.25)$ = 60 MeV-cm^2/mg
  - Saturated Cross Section (cm^2) per bit, 1.2E-7
  - 2.3E-11 errors/bit-day, Adams to 90% geosynchronous heavy ion
- Standard Military Drawing 5962-01502

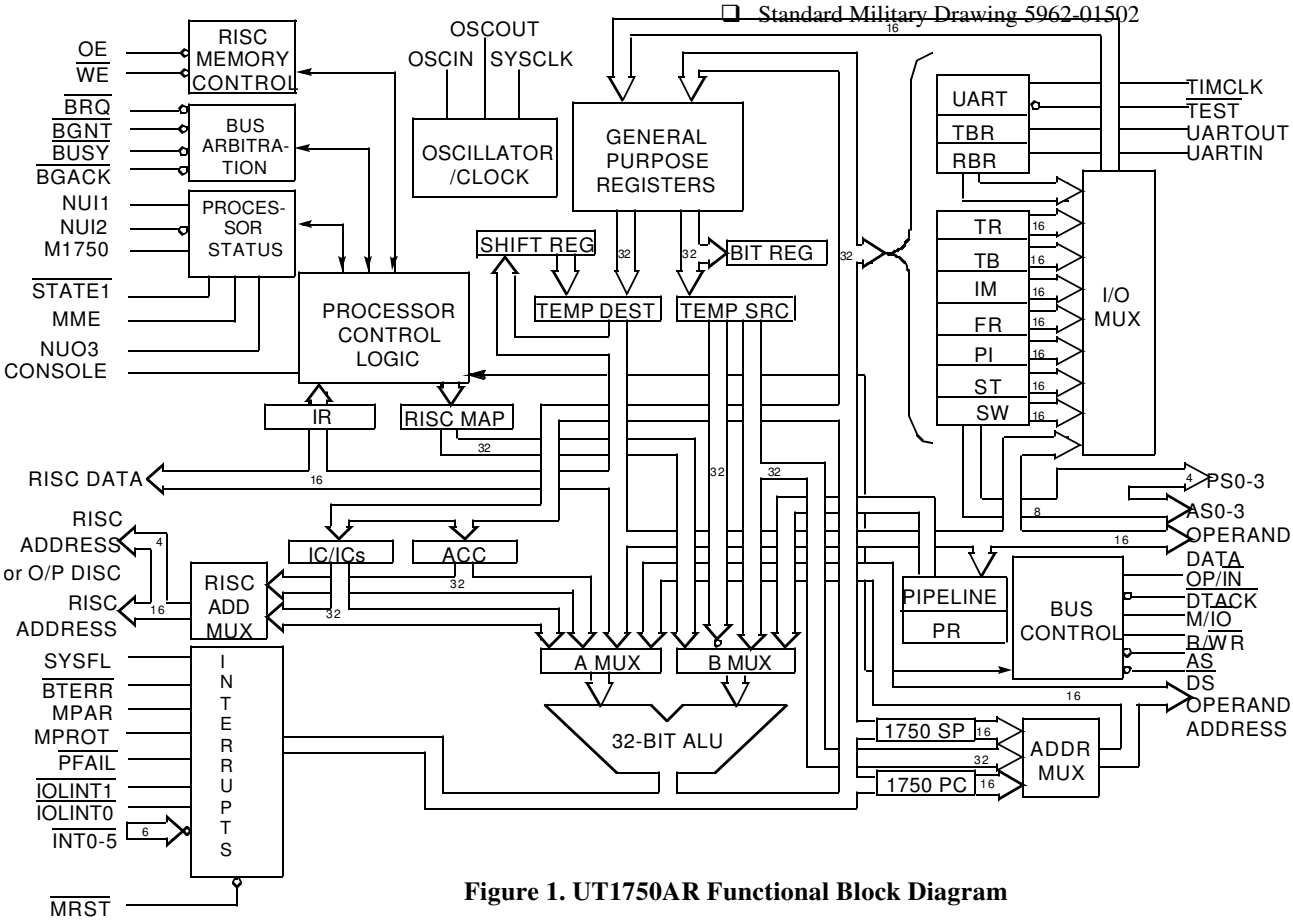


Figure 1. UT1750AR Functional Block Diagram

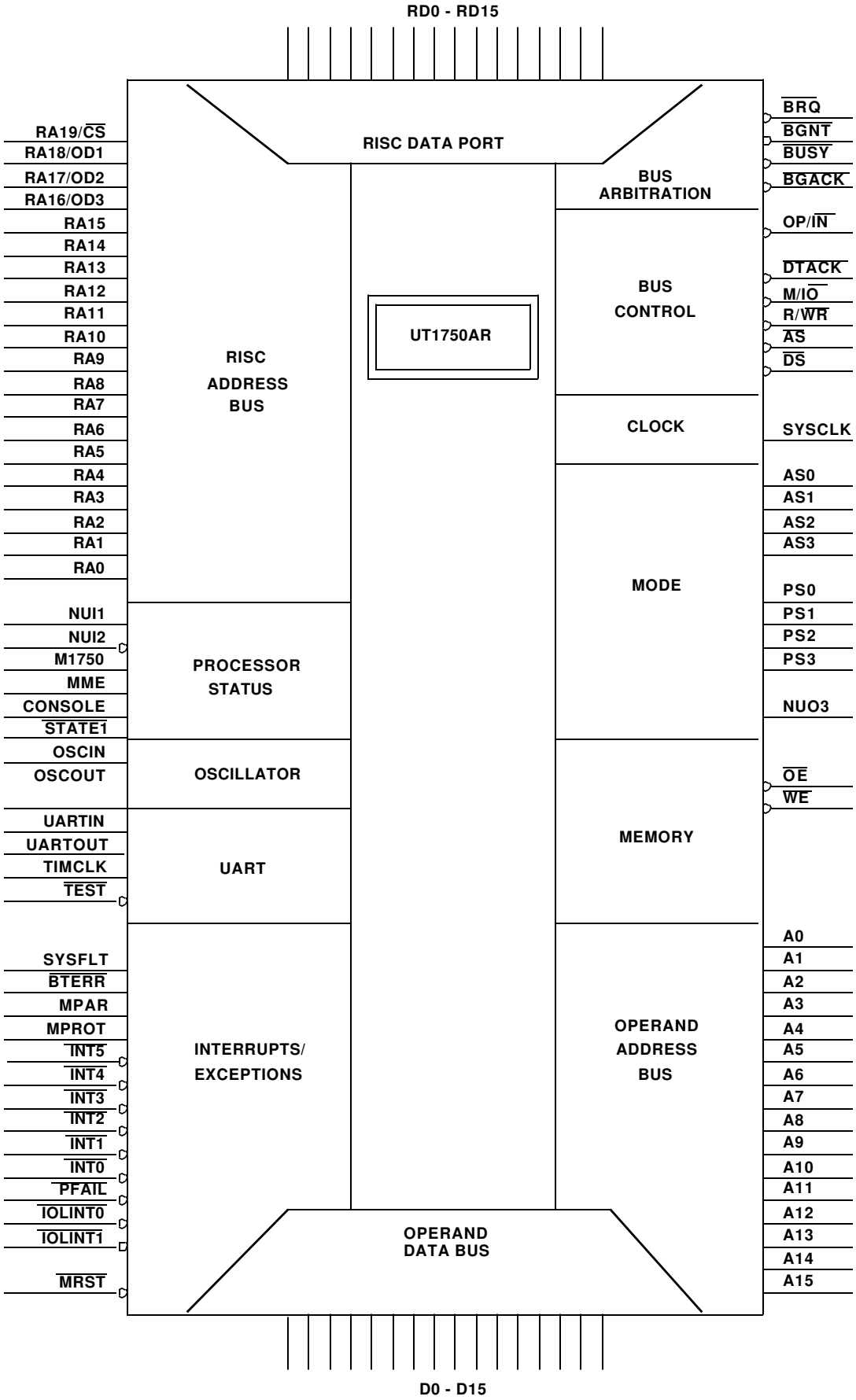


Figure 2. UT1750AR Pin Function Diagram

GENERAL DESCRIPTION

The UT1750AR (figures 1 and 2) is a high performance monolithic CMOS 16-bit RISC microprocessor that supports the complete MIL-STD-1750A Instruction Set Architecture (ISA). Underlying the MIL-STD-1750A support is a high-performance RISC that provides MIL-STD-1750A emulation capability. Developed to provide effective real-time avionics processing, the high performance of the native RISC machine is available to the MIL-STD-1750A systems designer through the MIL-STD-1750A Built-In-Function (BIF) opcode.

The UT1750AR is the first member of a family of high-performance MIL-STD-1750 processors and support peripherals from UTMC.

PRODUCT DESCRIPTION

The UTMC UT1750AR operates in its native RISC language mode or MIL-STD-1750A ISA mode. As a MIL-STD-1750A microprocessor, the UT1750AR requires 8K x 16 of ROM to map the MIL-STD-1750A instruction set into the native RISC machine language instructions. Each MIL-STD-1750A opcode has a unique RISC code macro in the external ROM. The UT1750AR executes the corresponding resident RISC code macro to perform the MIL-STD-1750A instruction requirements. When in this mode and operating with a 12 MHz clock, the UT1750AR can throughput 600 KIPS using the DAIS mix (800 KIPS @ 16 MHz).

The native RISC language mode is available to the user when the UT1750AR is operating as MIL-STD-1750A processor through MIL- STD-1750A's Built-In Function (BIF) opcode. When operating as a RISC processor, the UT1750AR executes most RISC instructions in two clock cycles. Thus, a 12 MHz operating clock frequency provides up to 6 MIPS of RISC throughput (8 MIPS @16 MHz). This high execution rate, along with its efficient architecture, make the RISC mode especially effective in applications requiring real-time processing.

The architecture of the UT1750AR is based around 20 user-accessible, 16-bit general purpose registers providing the programmer with extensive register support. The UT1750AR's flexibility is enhanced by its ability to concatenate the 16-bit registers into ten 32-bit registers. In addition, all registers are available for use as either the source or the destination for any register operation.

The UT1750AR fully supports multiprocessor, DMA, and complex bus arbitration for managing the system bus and preventing bus contention. Bus control passes among bus masters operating on the same bus. The bus masters can be several UT1750ARs or any other device requiring Direct Memory Access, such as a MIL-STD-1553B interface.

The UT1750AR supports 16 levels of vectored interrupts. Ten of these are external interrupts, eight of which are user-definable. All 16 interrupt levels are prioritized and serviced in order of priority.

When used as a MIL-STD-1750A microprocessor, the UT1750AR's instruction set supports 16-bit fixed-point single-precision and 32-bit fixed-point double-precision data formats. Also, the UT1750AR can emulate 32-bit floating-point and 48-bit floating-point extended-precision data in two's complement representation.

In its native RISC mode, the UT1750AR's three basic instruction formats support 16-bit and 32-bit instructions. The formats are Register-to-Register, Register-to-Literal, and Register-to-Long-Immediate instructions.

Figure 3 shows the UT1750AR's general system architecture, its emulation ROM, instruction and data memory, and the system interface. The emulation ROM is isolated from the system; only the UT1750AR microprocessor accesses it.

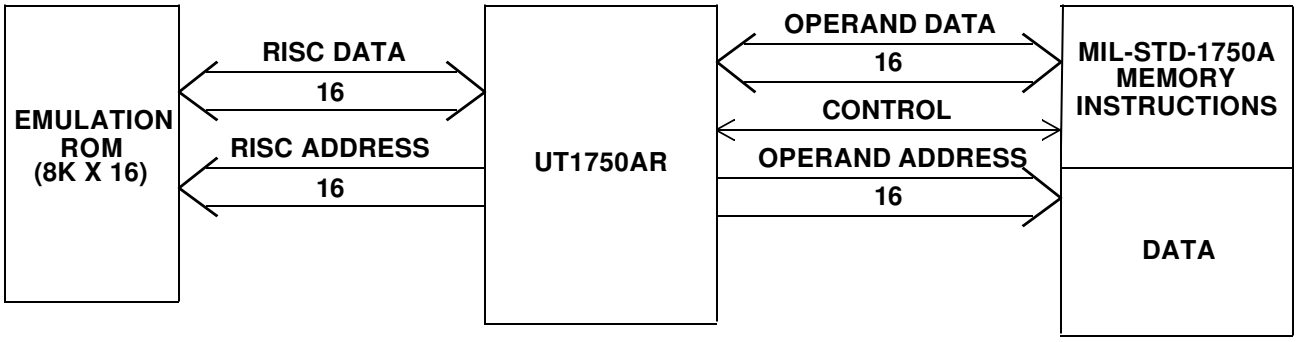


Figure 3. UT1750AR MIL-STD-1750A General System Architecture

FUNCTIONAL PINOUT

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 UT1750AR.
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.
NUO3	126	G3	TTO	--	Not used output 3. Internal UTMC use only. NUO3 enter high impedance state when the UT1750AR is in the test mode (TEST=0)
M1750	45	N11	TDI	AH	Mode Select RISC/1750. A high on M1750 places the UT1750AR into the MIL-STD-1750A emulation mode. A low on M1750 places the UT1750AR into the RISC mode. It is tied to an internal pull-down resistor.
$\overline{\text{STATE1}}$	54	N15	TTO	--	Processor State. This signal indicates the internal state of the UT1750AR. A low on STATE1 indicates the UT1750AR is executing a new RISC instruction. A high on STATE1 indicates the UT1750AR is fetching a RISC instruction. STATE1 enters a high-impedance state when the UT1750AR is in the test mode (TEST=0).

OPERAND DATA BUS ARBITRATION

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
$\overline{BRQ}$	118	D2	TTO	AL	Bus Request. The UT1750AR asserts this signal to indicate it is requesting control of the Operand data bus (D0 - D15). BRQ enters a high-impedance state when the UT1750AR is in the test mode (TEST = 0).
$\overline{BGNT}$	119	E3	TUI	AL	Bus Grant. When asserted, this signal indicates the UT1750AR 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 UT1750AR 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 UT1750AR 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 UT1750AR relinquishes control of the bus, BGACK enters a high-impedance state.

OPERAND DATA BUS CONTROL

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
OP/$\overline{IN}$	113	A2	TTO	--	Operand/Instruction. This indicates whether the UT1750AR's current bus cycle is for Data (high) or Instruction (low) acquisition. OP/IN remains in a high state whenever a bus cycle (Memory or I/O) is not an instruction fetch.
$\overline{DTACK}$	121	E2	TUI	AL	Data Transfer Acknowledge. This signal tells the UT1750AR that a data transfer has been acknowledged and the UT1750AR can complete the bus cycle. To assure the UT1750AR 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 UT1750AR does not control the Operand busses.
R/$\overline{WR}$	114	C4	TTO	--	Read/Write. Indicates the direction of data flow with respect to the UT1750AR. R/$\overline{WR}$ high means the UT1750AR is attempting to read data from an external device, and R/$\overline{WR}$ low means the UT1750AR is attempting to write data to an external device. R/$\overline{WR}$ remains in a high-impedance state when the UT1750AR does not control the Operand busses.

Continued on page 6.

OPERAND DATA BUS CONTROL

Continued from page 5.

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
$\overline{AS}$	115	C3	TTO	AL	Address Strobe. Indicates a valid address on the Operand Address bus. UT1750AR places AS in a high-impedance state when it does not control the Operand busses.
$\overline{DS}$	116	B2	TTO	AL	Data Strobe. Indicates valid data is on the Operand Data bus. The UT1750AR places DS in a high-impedance state when it does not control the Operand busses.

RISC MEMORY CONTROL

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
$\overline{OE}$	42	R12	TTO	AL	Output Enable RISC Memory. This signal allows RISC memory to place data on the RISC instruction data bus. The Store Register to Instruction Memory (STRI) instruction removes OE during the CK2 internal clock cycle. OE enters a high-impedance state when the UT1750AR is in the test mode (TEST = 0).
$\overline{WE}$	43	R13	TTO	AL	Write Enable RISC Memory. This signal allows the UT1750AR to write to RISC instruction memory. The Store Register to Instruction Memory (STRI) instruction asserts WE during the CK2 internal clock cycle. WE enters a high-impedance state when the UT1750AR is in the test mode (TEST = 0).

UART CONTROL/TIMER CLOCK

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
UARTIN	127	F1	TUI	AH	UART Input. The UT1750AR receives serial data through this input. The serial data is stored in the UT1750AR'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 UT1750AR'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 UT1750AR is in the test mode (TEST=0). (9600 baud @ TIMCLK = 12MHz)

Continued on page 7.

UART CONTROL/TIMER CLOCK *Continued from page 6*

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 UT1750AR's internal UART. The input also provides the clock for the UT1750AR's two internal MIL-STD-1750A timers (TIMER A and TIMER B).
CONSOLE	48	N12	TDI	AH	Console (Command). Asserting this input sets bit 3 in the System Status Register. Bit 3 is read with the Input Register Instruction (INR). When the UT1750AR is operating in the MIL-STD-1750 mode, asserting CONSOLE during a Master Reset invokes the maintenance console option. Tied to an internal pull-down resistor.
$\overline{\text{TEST}}$	46	P13	TUI	AL	Test (Input). Asserting this input places the UT1750AR into a test mode. In this mode, all the UT1750AR's outputs, except OSCOUT and SYSCLK, enter a high-impedance state. When using $\overline{\text{TEST}}$, the UT1750AR must have a MRST. 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.
MME	49	N13	TDI	AH	Memory Management Enable. This signal indicates to the UT1750AR that a Memory Management Unit (MMU) is present and that the memory management option is enabled. MME is tied to an internal pull-down resistor.

PROCESSOR MODE

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
AS0 AS1 AS2 AS3	104 105 106 107	B7 B6 C6 A5	TTO	AH	Address State. These outputs indicate the current address state of the UT1750AR. Using these outputs with a Memory Management Unit (MMU) allows selecting the MMU's page register group. These outputs enter a high-impedance state when the UT1750AR is placed in the test mode (TEST=0) or during bus cycles not assigned to this processor.
PS0 PS1 PS2 PS3	108 109 110 111	A4 A3 B4 C5	TTO	AH	Processor State. These outputs indicate the current state of the processor. These outputs enter a high-impedance state when the UT1750AR is in the test mode (TEST=0) or during bus cycles not assigned to this processor.

INTERRUPTS/EXCEPTIONS

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
SYSFLT	125	G2	TUI	AH	System Fault. This positive edge-triggered input sets bit 8 (SYSFLT) in the UT1750AR's Fault Register. Under no circumstances should SYSFLT be tied in its active state. It is tied to an internal pull-up resistor.
$\overline{\text{BTERR}}$	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 BTERR sets bit 10 of the Fault Register. During Memory bus cycles, an active BTERR sets bit 7 of the Fault Register. Under no circumstances should 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.
MPAR	124	F2	TDI	AH	Memory Parity (Error). Asserting this input indicates a MIL-STD-1750 memory parity error. Bit 13 of the UT1750AR's Fault Register, Memory Parity Fault, is set when MPAR is active. Under no circumstances should MPAR 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 UT1750AR 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 UT1750AR is using the bus and MPROT is asserted, bit 15 of the Fault Register (CPU Fault) is set. If the UT1750AR 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.
$\overline{\text{INT0}}$ $\overline{\text{INT1}}$ $\overline{\text{INT2}}$ $\overline{\text{INT3}}$ $\overline{\text{INT4}}$ $\overline{\text{INT5}}$	56 57 58 59 60 61	M15 K13 K14 J14 J13 K15	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 MRST resets all interrupts. They are tied to an internal pull-up resistor.
$\overline{\text{IOLINT0}}$ IOLINT1	62 63	J15 H14	TUI	AL	I/O Level Interrupts. These inputs are active on a negative-going edge and each sets, 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 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 UT1750AR 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. When operating in the RISC mode, the UT1750AR must be reset after a PFAIL to assure normal operation. It is tied to an internal pull-up resistor.
$\overline{\text{MRST}}$	47	R14	TUI	AL	Master Reset. This input initializes the UT1750AR to a reset state. The UT1750AR must be reset after power (Vcc) is within specification and stable to ensure proper operation. The system must hold MRST active for at least one period of SYSCLK to assure the UT1750AR will be reset. It is tied to an internal pull-up resistor.

OPERAND BUSSES

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 UT1750AR does not control the bus. A15 is the most significant bit. The Operand Address enters a high-impedance state when the UT1750AR is in the test mode (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 UT1750AR does not control the bus. D15 is the most significant bit. The Operand Data Bus enters a high-impedance state when the UT1750AR is in the test mode (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			

RISC BUSSES

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
RA0	18	R2	TTO	--	RISC (Instruction) Address Bus. This unidirectional bus represents the address of the data in RISC memory. With the MIL-STD-1750A mode of operation selected (M1750 = 1), the data from RISC memory is from the emulation ROMs. This data is the RISC instructions that the UT1750AR executes to emulate MIL-STD-1750A instructions. RA15 is the most significant bit. The RISC address enters a high-impedance state when the UT1750AR is in the test mode (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			

Continued on page 10.

RISC BUSSES

Continued from page 9.

PIN NAME	PIN NUMBER		TYPE	ACTIVE	DESCRIPTION
	FLTPK	PGA			
RA16/OD3 RA17/OD2 RA18/OD1 RA19/$\overline{CS}$	38 39 40 41	P9 P10 N10 R11	TTO	--	RISC Instruction Address Bus/Output Discretes. When the UT1750AR is operating in the RISC mode (M1750 = 0) these four bits represent the four most significant address bits. In the MIL- STD-1750A mode (M1750 = 1) these four bits are user-programmable output discretes defined as follows: RA19/$\overline{CS}$ = Chip Select (AL) RA18/OD1 = Output Discrete 1 RA17/OD2 = Output Discrete 2 RA16/OD3 = Output Discrete 3 These output discretes are programmed with the Output Register (OTR) RISC opcode. These signals enter a high-impedance state when the UT1750AR is in the test mode ($\overline{TEST}$ = 0).
RD0 RD1 RD2 RD3 RD4 RD5 RD6 RD7 RD8 RD9 RD10 RD11 RD12 RD13 RD14 RD15	130 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	H1 J1 K1 J2 K2 K3 L1 M1 N1 M2 L3 N2 P1 M3 N3 P2	TTB	--	RISC Instruction Data Bus. This bidirectional data bus is the interface with the RISC memory. When the UT1750AR is in the MIL-STD-1750A mode of operation, the data comes from the emulation ROMs. This data is executed to emulate the MIL-STD-1750A Instruction Set. RD15 is the most significant bit. The RISC Data Bus enters a high-impedance state only when the UT1750AR is in the test mode ($\overline{TEST}$ = 0).

POWER AND GROUND

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

GENERAL OPERATION

The UT1750AR can operate in two modes. The first operating mode is the Reduced Instruction Set Computer (RISC) mode; the second is the MIL-STD-1750A Instruction Set Architecture (ISA) emulation mode. The mode-select input pin (M1750) determines the UT1750AR's operating mode. M1750 must be tied high to enable the MIL-STD-1750A ISAemulation mode of operation; otherwise, an internal pull-down resistor pulls M1750 low, selecting the RISC mode.

The UT1750AR has a Harvard architecture when it operates in the RISC mode (M1750 = 0). A processor with a Harvard architecture has two sets of address and data busses; one set interfaces with instruction memory and the other set interfaces with operand memory. This architecture allows the UT1750AR to perform overlapping instruction fetch-and-execute bus cycles that enhance processor throughput.

The UT1750AR's reduced instruction set consists of 30 separate instructions. The UT1750AR executes most of these instructions in two clock cycles providing fast execution of RISC-coded programs. All the UT1750AR's processing capabilities in the RISC mode are available to the system programmer by using the companion RISC Assembler (RASM)/Linker (RLNK), RISC Interactive Software Simulator (IRSIM), and hardware development debug tools.

In the MIL-STD-1750A mode of operation (M1750 = 1), the UT1750AR has a Von Neumann architecture. A processor with a Von Neumann architecture has a common set of address and data busses that make instructions and operand data available to the processor.

The UT1750AR emulates the MIL-STD-1750A instruction set when it has a specially programmed set of RISC PROMs. These PROMs contain RISC-coded macros that correspond to each MIL-STD-1750 instruction. When the UT1750AR fetches a 1750 instruction from memory, it decodes this instruction's opcode and generates an address for the RISC PROMs. This address points to a RISC macro that, when executed, performs the operation the 1750 instruction requires.

The high execution rate of the UT1750AR's native RISC language is also available when the UT1750AR is in the MIL-STD-1750 mode of operation by using the MIL-STD-1750 Built-in-Function (BIF) opcode. The system designer can develop a RISC macro for a specific function, such as power-on self-test routines, built-in-test routines, signal-processing routines, or any routine that requires real-time processing. The UT1750AR executes this function when it encounters the BIF in the MIL-STD-1750 program flow.

The RISC Mode of Operation

The configuration for the UT1750AR in the RISC mode of operation is shown in figure 4. RISC is the default mode of operation for the UT1750AR since the M1750 input is tied to an internal pull-down resistor.

When the UT1750AR operates in the RISC mode, the system designer stores the executable RISC program in RISC memory. The UTMC RISC Assembler generates this executable RISC program. All 20 of the RISC address lines can access a user-defined program in RISC memory. This means the maximum length of any RISC program is 1 mega- word.

Although the executable RISC program is all that is stored in RISC memory, two RISC instructions allow the programmer to manipulate the data in RISC memory. These instructions are the Load Register from (RISC) Instruction Memory (LRI) and the Store Register to (RISC) Instruction Memory (STRI).

When operating in the RISC mode, the UT1750AR first generates an address on the RISC address bus for the instruction it stores in the Primary Instruction Register (PIR). After the UT1750AR stores the RISC instruction in the PIR, the UT1750AR begins executing the instruction in the Instruction Register (IR). If the present instruction in the IR requires only internal processing, the UT1750AR does not exercise the Operand Address and data busses. If, on the other hand, the instruction in the IR requires some type of Operand Data, the UT1750AR begins an Operand bus arbitration cycle midway through the next processor clock cycle.

The Operand bus arbitration cycle begins with the UT1750AR asserting the Bus Request ($\overline{BRQ}$) signal. The UT1750AR samples the Bus Grant ($\overline{BGNT}$) and the Bus Busy ($\overline{BUSY}$) signals on every falling edge of the processor clock. When the UT1750AR detects that the previous bus controller has relinquished control of the bus, the UT1750AR generates the Bus Grant Acknowledge ($\overline{BGACK}$) signal signifying that it has taken control of the bus.

After the UT1750AR has taken control of the bus, it generates the Operand address and data. The Address Strobe ($\overline{AS}$) and Data Strobe ($\overline{DS}$) signals indicate when the Operand address and data are valid. If the UT1750AR 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 memory device lengthens the memory cycle until it can provide data for the UT1750AR.

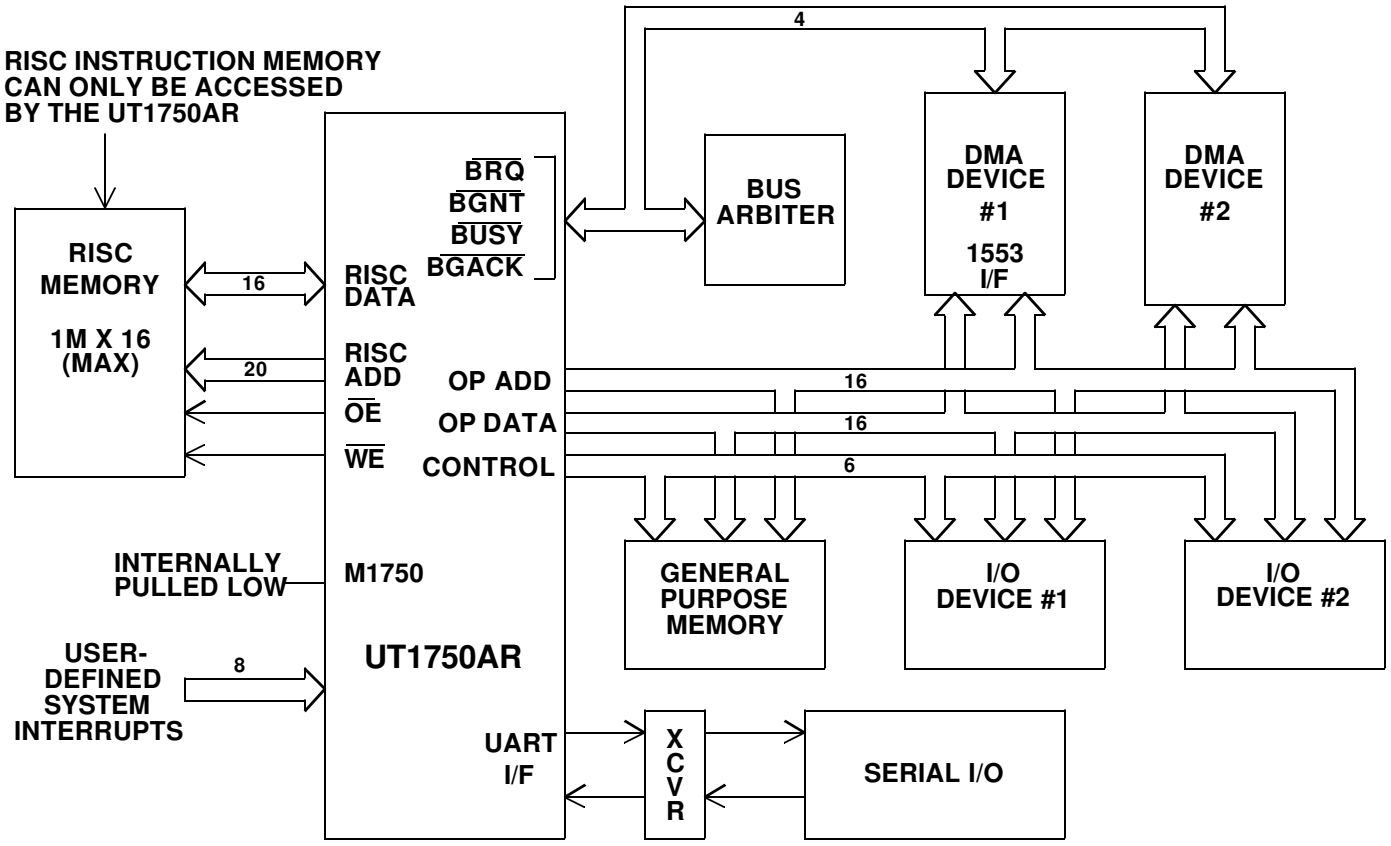


Figure 4. The UT1750AR in the RISC Mode of Operation

All user-definable interrupts are available when the UT1750AR is operating as a RISC. In addition, the system programmer can read or write to virtually all of the UT1750AR's internal registers, either general purpose or specialized, when the UT1750AR is in the RISC mode by using the Internal I/O command (INR) or the Output Register command (OTR), respectively.

The 1750A Mode of Operation

The configuration for the UT1750AR in the MIL-STD-1750A mode of operation is shown in figure 5. The UT1750AR enters the 1750 mode of operation when the mode input, M1750, is pulled high.

The functional operation of the UT1750AR in the MIL-STD-1750 mode is similar to the RISC mode of operation, although it has two important differences. The first difference is that when the system designer selects the MIL-STD-1750 mode, the UT1750AR requires a specific set of RISC PROMs specially programmed to allow the UT1750AR to emulate the 1750 ISA.

This special set of RISC PROMs contains a set of RISC-coded macros that allow the UT1750AR to serve as a full-feature MIL-STD-1750A microprocessor. In this respect, the RISC PROMs hold external microcode, or "Mili"-code. This "Mili"-code tells the UT1750AR how to function as a 1750 processor and, if necessary, the user can change the "Mili"-code if the application requires additional capability for real-time processing.

The second difference between the operation of the UT1750AR in the 1750 mode and the RISC mode is that in the 1750 mode the RISC address bus is limited to 16 address lines or 64K words instead of the UT1750AR's 20-bit RISC address bus in the RISC mode. When in the 1750 mode, the UT1750AR uses the four most significant bits of the RISC address bus for output discretes. The output discrete that replaces the most significant address bit (RA19) is a dedicated chip select.

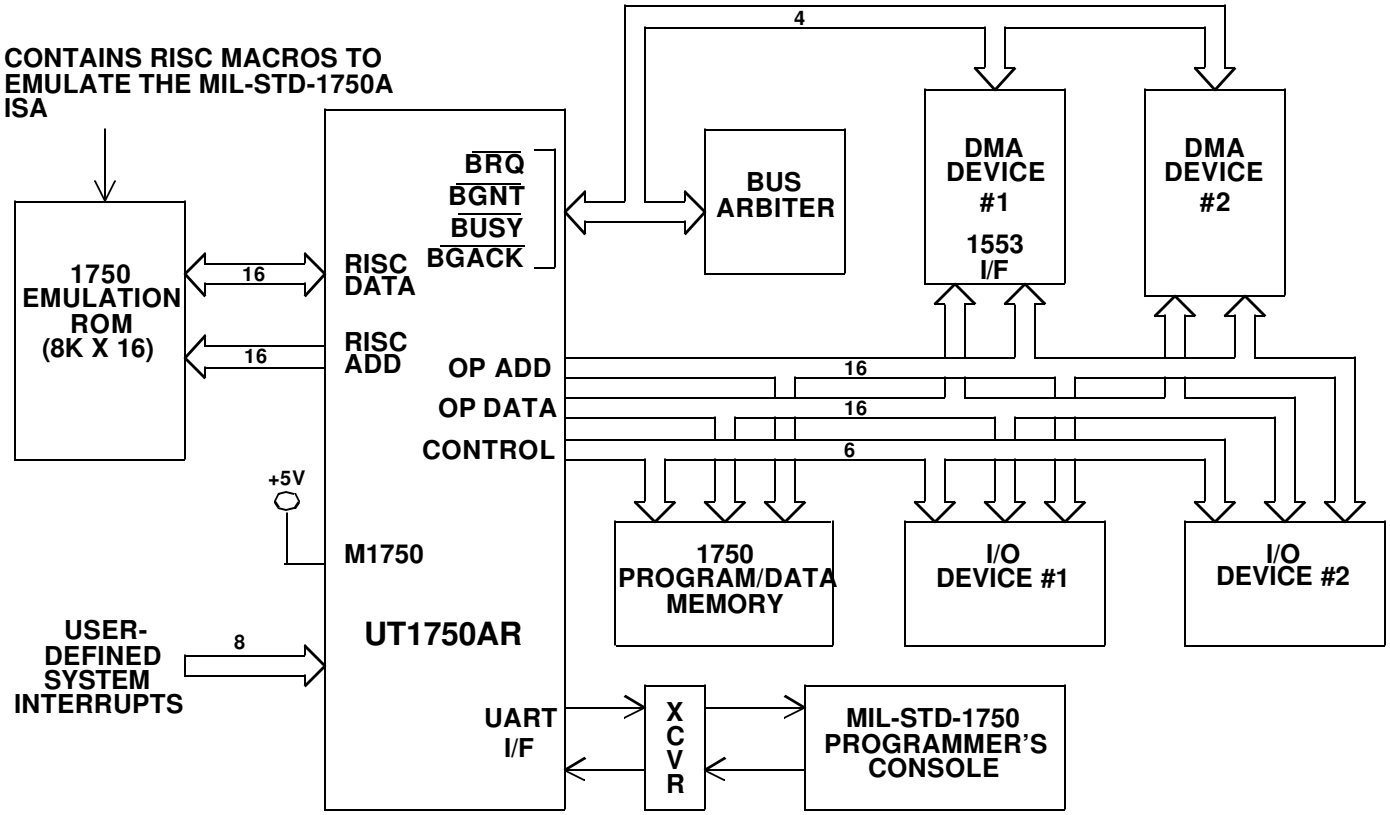


Figure 5. The UT1750AR in the MIL-STD-1750 Mode of Operation

The next three RISC address bits (RA16-RA18) are user-definable discrete outputs. These outputs are defined as:

RA16/OD3 ⟶ DMA enable ($\overline{\text{DMAEN}}$)
RA17/OD2 ⟶ power-up ($\overline{\text{GOOD}}$)
RA18/OD1 ⟶ start-up ROM enable ($\overline{\text{SUREN}}$)

After reset these signals will be in the following states:
RA16 ⟶ 1, RA17 ⟶ 0, RA18 ⟶ 0.

When the UT1750AR operates in the MIL-STD-1750 mode, it generates an address on the Operand address bus for the next 1750 instruction. If the UT1750AR has just been initialized or has just been reset, the first memory location placed on the Operand Address Bus is 0000H; this instruction is the first one fetched from the 1750 memory. After this instruction is fetched and entered into the UT1750AR, the UT1750AR uses the opcode to "map" or point to a specific address in the RISC memory. Since the RISC PROM programming provides 1750 emulation capability, this address in RISC memory contains a specific RISC-coded macro allowing the UT1750AR to perform the requisite 1750 function.

When the UT1750AR begins executing this RISC macro for 1750 emulation, the UT1750AR begins to operate as if it were in the RISC mode (see the previous section on RISC mode of operation). The processor cycles of all the RISC instructions that make up the particular macro are executed as if the UT1750AR were operating purely as a RISC.

During RISC macro execution for the MIL-STD-1750 instruction, the internal registers of the UT1750AR hold the intermediate results from the execution of the RISC instructions. When the macro is complete, the UT1750AR's registers contain the data the MIL-STD-1750A instruction requires.

If the UT1750AR receives an interrupt during RISC macro execution, the RISC macro completes execution before the UT1750AR recognizes the interrupt. This is similar to completing a single 1750 instruction rather than allowing its interruption. The only exception is with the multiple-word MOV 1750 instruction. For this instruction, the UT1750AR interrupts macro execution after transferring the current word.

After the RISC macro is complete, all the UT1750AR's internal registers, including the status registers and/or memory locations, contain the results of the MIL-STD-1750A instruction that has just completed execution. The UT1750AR now fetches the next 1750 instruction from Operand memory and the process repeats.

The advanced architecture of the UT1750AR allows the system designer to define RISC macros accessible through the MIL-STD-1750A Built-In Function (BIF) opcode. These user-defined RISC macros can be any regularly-used function requiring the UT1750AR's high-speed, real-time processing capabilities. The UT1750AR fetches the BIF instruction from Operand memory just like any other 1750 instruction; it then decodes the BIF. The resulting UT1750AR-generated RISC address points to the location of the user-defined macro in RISC memory. RISC macro execution proceeds just as it would for any other 1750 instruction. MIL-STD-1750A permits the system designer to define up to 256 BIF variations.

REGISTER ARCHITECTURE

The UT1750AR has a register-oriented architecture (figure 1). The registers within the UT1750AR fall into two categories: general purpose registers, and specialized registers. All the UT1750AR's registers are accessible to the programmer through the RISC instruction set. The programmer uses data from these registers to perform arithmetic and logical functions, alter program flow, detect various system and processor faults, determine processor status, provide control for UART and timer functions, and provide interrupt processing and exception-handling control.

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 6. General Register Set

General Purpose Registers

Figure 6 shows the UT1750AR's 20 general purpose registers. All RISC instructions use these registers; any register or register pair can be either the source or the destination for any RISC instruction. The UT1750AR normally accesses these registers as single-word 16-bit registers although the UT1750AR 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 UT1750AR has a 32-bit Accumulator (ACC). The ACC is normally a destination register, although under certain circumstances it can be the source register. The Accumulator retains the most significant half of the product during a multiply instruction or the remainder during a divide operation.

Specialized Registers

The UT1750AR has 16 special purpose registers (figures 7 through 24). The values in the brackets indicate the power-up condition. They are:

1. Stack Pointer Register (SP) [XXXX16]
2. System Status Register (STATUS)
3. UART Receiver Buffer Register (RCVR) [XX0016]
4. UART Transmitter Buffer Register (TXMT) [XX0016]
5. Pending Interrupt Register (PI) [000016]
6. Fault Register (FT) [000016]
7. Interrupt Mask Register (MK) [XXXX16]
8. 1750 Status Register (SW) [000016]
9. RISC Instruction Counter Register (IC) [0000016]
10. RISC Instruction Counter Save Register (ICS) [XXXXX16]
11. RISC Instruction Register (IR) [000016]
12. 1750 Pipeline Register (PIPE) [XXXX16]
13. 1750 Program Register (PR) [XXXX16]
14. 1750 Program Counter (PC) [XXXX16]
15. 1750 Timer A Register (TA) [000016]
16. 1750 Timer B Register (TB) [000016]

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

The Stack Pointer Register

Figure 7. The UT1750AR uses the 16-bit Stack Pointer Register as an address pointer on Push and

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

MSB LSB

Figure 7. The Stack Pointer Register (SP)

Pop instructions. When the UT1750AR is operating in the RISC mode, it pre-increments (pops) and post-decrements (pushes) the SP. In the 1750 mode, the UT1750AR pre-increments (pops) and post-increments (pushes) the SP.

The programmer accesses the SP by using local I/O commands to Load and Store the Stack Pointer.

The System Status Register

Figure 8. The System Status Register provides additional status information on the UT1750AR's internal signals, including the status of the internal UART. The bit definitions for STATUS are given below.

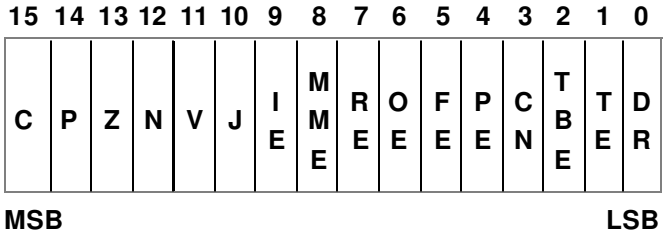


Figure 8. The System Status Register (STATUS)

Bit Definitions

All bits in the System Status Register are active high. The values in the brackets indicate the power-up state.

BIT NUMBER	MNEMONIC	DESCRIPTION
15	C	Carry. This conditional status is set if a carry generated. [0]
14	P	Positive. This conditional status is set if the result of operation is positive. [0]
13	Z	Zero. This conditional status is set if the result of an operation is equal to zero. [0]
12	N	Negative. This conditional status is set if the result of an operation is negative. [0]
11	V	Overflow. This conditional status is set when an overflow condition occurs. [0]
10	J	Normalized. Thisconditional status is set as the result of a long instruction. [0]
9	IE	Interrupts enabled. [0]
8	MME	Memory Management enabled. [0]
7	RE	Receiver Error. This bit is the logical OR combination of the OE, FE, and PE status bits. [0]
6	OE	Overrun Error. When active, this bit indicates that at least one data word was lost because the Data Ready (DR is bit 0 of the STATUS) signal was active twice consecutively without an RBR read. [0]
5	FE	Framing Error. When active, this bit indicates a stop bit was missing from the serial transmission. [0]
4	PE	Parity Error. When active, this bit indicates the serial transmission was received with the incorrect parity. [0]
3	CN	MIL-STD-1750A Console Enabled. When active, this bit indicates the CONSOLE discrete input is active. CONSOLE active sets bit 3 in the System Status Register.
2	TBE	UART Transmitter Buffer Empty. This bit indicates the Transmitter Buffer Register is empty and ready for data. [0]
1	TE	UART Transmitter Empty. This bit is low while the UART is transmitting data and goes high when the transmission is complete. [0]
0	DR	UART Data Ready. This active-high signal indicates the UART received a serial data word and this data is available. [0]

UART Receiver Register (RCVR)

The UART Receiver Buffer Register (see figure 9) receives 9600-baud asynchronous serial data through the UARTIN input pin on the UT1750AR. Each serial data string contains an active-low Start bit, eight Data bits, an odd Parity bit, and an active-high Stop bit. Figure 10 shows a single serial data string.

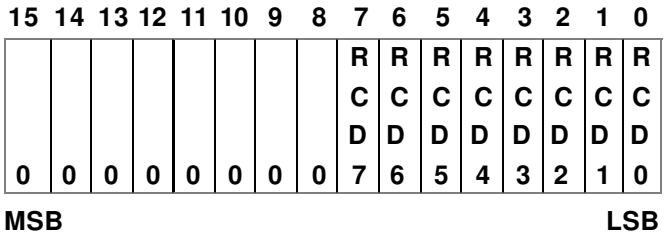


Figure 9. The UART Receiver

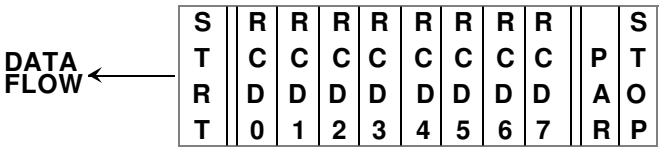


Figure 10. UART Receiver Data String

While receiving a serial data string, the UT1750AR generates four status flags: Data Ready (DR); Overrun Error (OE); Framing Error (FE); and Parity Error (PE). The UT1750AR stores these status bits in the System Status Register (SYSTAT).

Receiver buffer register bits 15-8 are always low. Bit numbers 7-0 (RCD7-RCD0) contain the data the UT1750AR receives via the serial data port. RCD7 is the MSB; RCD0 is the LSB.

UART Transmitter Buffer Register (TXMT)

The UT1750AR's internal UART forms an 11-bit serial data string by combining a Start bit, the eight Data bits from the Transmitter Buffer Register (TXMT), an odd Parity bit, and a Stop bit. Figure 11 shows the composition of the serial data string.

The UT1750AR transmits this serial data string through the UARTOUT pin at a rate of 9600 baud.

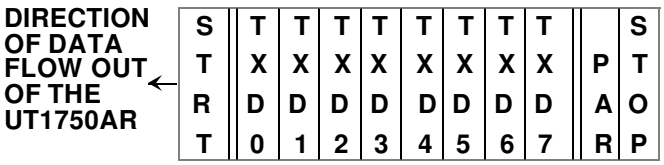


Figure 11. UART Transmitter Data String

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 as bits 2 and 1, respectively.

The UT1750AR's internal UART has a double-buffered data transmission scheme (figure 12). The UT1750AR first loads the data for transmission into the Transmitter Buffer Register. If the UART Transmitter Register is empty, data from the TXMT automatically transfers to the UART Transmitter Register. At this time, the TBE bit goes active indicating more data may be loaded into the TXMT. This double-buffering scheme allows contiguous transmission of serial data streams and also decreases the UT1750AR's required overhead for the UART interface.

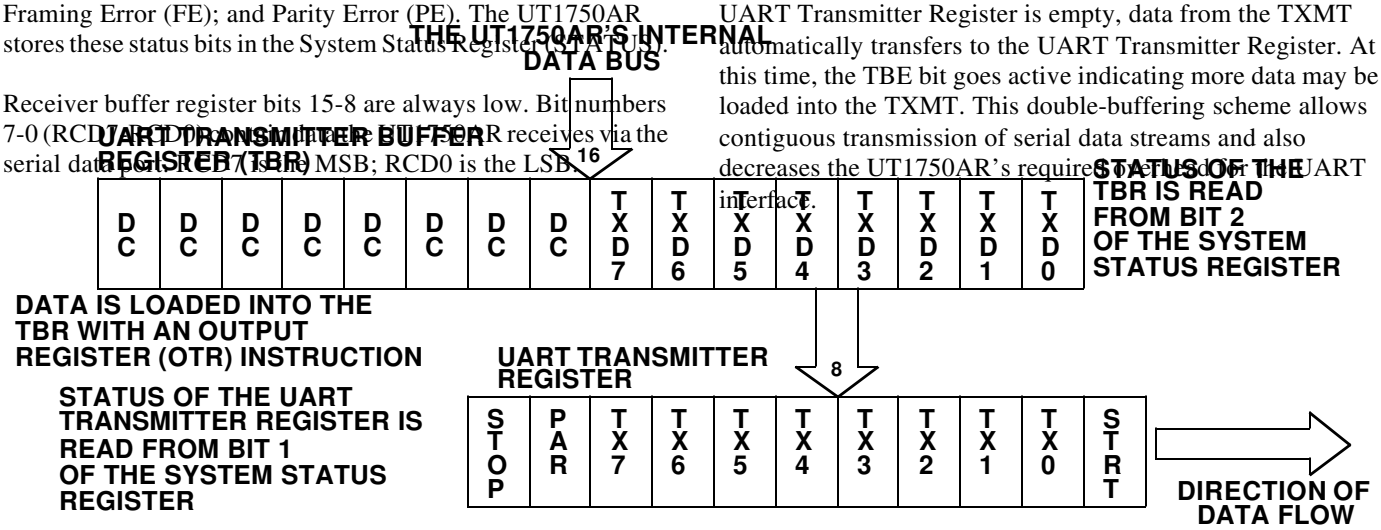


Figure 12. The UT1750AR UART Double-Buffered Transmitter Register

The UT1750AR loads the eight bits of serial data into the lower eight bits of the TXMT (figure 13).

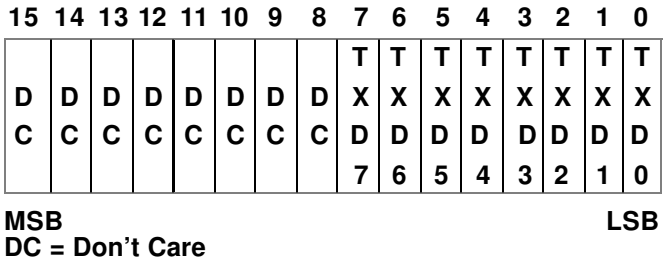


Figure 13. The UART Transmitter

The Pending Interrupt Register (PI)
The Pending Interrupt Register (PI) contains information on pending interrupts attempting to vector the Instruction Counter Register (IC) to a new location. Software or hardware controls the PI. Any system interrupt, when active, sets the corresponding bit in the PI. RISC I/O instructions can also set, clear, and read the PI (figure 14).

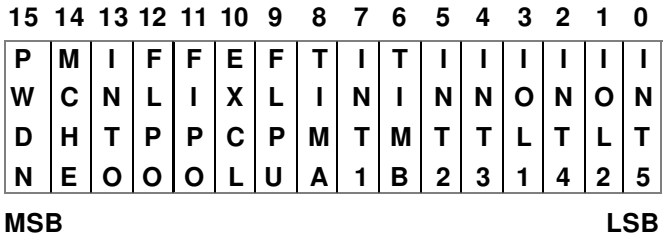


Figure 14. The Pending Interrupt Register (PI)

The Fault Register (FT)
The UT1750AR uses the Fault Register (FT) (figure 15) to indicate the occurrence of a machine-error fault. A machine-error fault cannot be disabled. The UT1750AR uses the logical OR combination of the 16 FT bits to generate the Machine Error interrupt, bit 14 of the PI. Any bits in the FT the UT1750AR does not use are set to a logic zero. The UT1750AR reads, loads, and clears the FT with RISC I/O instructions. The configuration of the FT is shown in figure 15.

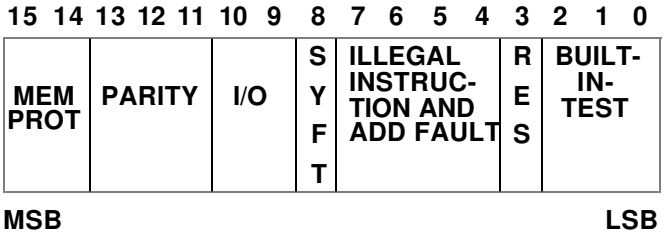


Figure 15. The Fault Register (FT)

Bit Definitions
All bits in the Fault Register are active when high.

BIT NUMBER	MNEMONIC	DESCRIPTION
15	CMPF	CPU Memory Protect Fault. This bit indicates the UT1750AR has detected an access fault, write-protect fault, or an execute-protect fault. [0]
14	DMPF	DMA Memory Protect Fault. This bit indicates a DMA device has detected an access fault or a write-protect fault. [0]
13	MPF	Memory Parity Fault. [0]
12	PCPF	Parallel I/O (PIO) Channel Parity Fault. [0] No user access.
11	DCPF	DMA Channel Parity Fault. [0] No user access.
10	ICF	Illegal Command Fault. This bit indicates an attempt to execute an unimplemented or reserved I/O command. [0]
9	PTF	PIO Transmission Fault. Can wire-OR I/O error-checking devices together and feed them into this input to indicate an error. [0] No user access.
8	SYSFLT	System Fault. [0]
7	IAF	Illegal Address Fault. This bit indicates addressing a memory location not physically present. [0]
6	IIF	Illegal Instruction Fault. This bit indicates an attempt to

		execute a reserved code. [0]
5	PIF	Privileged Instruction Fault. This bit indicates an attempt to execute a privileged instruction with the Processor State not equal to zero. [0]
4	ASF	Address State Fault. This bit indicates an attempt to establish an Address State value for an unimplemented page register set. [0]
3		Reserved.
2	BITF	Built-In-Test Fault. This bit indicates the UT1750AR has detected a hardware built-in-test error. [0]
1 - 0		Spare BIT. The user defines these bits as additional BIT parameters. [0]

The Interrupt Mask Register (MK)
The Interrupt Mask Register (MK) (figure 16) contains one mask bit for each of the 16 system interrupts. All bits in the MK are set or reset under software control, although setting bits 15 and 10, Power Down Interrupt and Executive Call respectively, has no effect on the UT1750AR's operation because these interrupts cannot be masked. The UT1750AR reads or loads the MK with RISC I/O instructions.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWDN	MCHE	INTO	FLPO	FIPO	EXCL	FLPU	TIMA	INT1	IMB	INT2	INT3	IOL1	INT4	IOL2	INT5

MSB **LSB**

Figure 16. The Interrupt Mask Register (MK)

The 1750 Status Word Register (SW)
The MIL-STD-1750A Instruction Set Architecture (ISA) defines the Status Word Register (SW). The UT1750AR reads and loads the SW with RISC I/O instructions. Figure 17 shows the definitions of various bits in the SW.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONDITION STATUS (CS)				RESERVED				PROCESSOR STATE (PS)				ADDRESS STATE (AS)			

MSB **LSB**

Figure 17. The 1750 Status Register (SW)

Bit Definitions

BIT NUMBER	MNEMONIC	DESCRIPTION
15	C	Carry. This bit is set if the result of an addition operation generates a carry or if the result of a subtraction generates no borrow.
14	P	Positive. This bit is set if the result of an operation is greater than zero.
13	Z	Zero. This bit is set if the result of an operation is equal to zero.
12	N	Negative. This bit is set if the result of an operation is less than zero.
11 - 8		Reserved Bits.
7 - 4	PS3 - PS0	Processor State. This four bit field determines the legal illegal criteria for privileged instructions.
3 - 0	AS3 - AS0	Address State. Used in conjunction with the optional UT1750 MMU Memory Management Unit, this four-bit field determines the current extended address page.

Note: If condition codes are turned on (default after reset) the condition codes reflect the corresponding bits in the STATUS register.

The RISC Instruction Counter Register (IC) and The RISC Instruction Register (IR)

The UT1750AR's RISC interface consists of a 20-bit instruction address and a 16-bit data bus. The RISC Instruction Counter Register (IC) supplies the 20-bit address to RISC memory. The RISC's instruction data that is read from memory is then input into the RISC's Instruction Register (IR). The IR consists of two sets of latches, a Primary Instruction Register latch (PIR) and the Instruction Register latch (IRL). These two sets of latches allow the UT1750AR to perform overlapping memory fetch and execute cycles. This means the UT1750AR performs a delayed branch when the flow of the program is interrupted. A delayed branch implies that the UT1750AR fetches and executes the instruction following the branch condition BEFORE the UT1750AR executes the first instruction at the branch location.

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC19	IC18	IC17	IC16	IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0

MSB LSB

Figure 18. RISC Instruction Counter Register (IC)

The RISC Instruction Register (IR) is made of two 16-bit latches: the Primary Instruction Register (PIR) latch, and the Instruction Register (IRL) latch.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR15	IR14	IR13	IR12	IR11	IR10	IR9	IR8	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0

MSB LSB

Figure 19. Instruction Register (IR)

The RISC Instruction Counter Save Register (ICS)

The UT1750AR uses the RISC's Instruction Counter Save Register (ICS) (figure 20) when servicing interrupts and branch instructions. When an interrupt or branch occurs, the UT1750AR saves the IC in the ICS. Read the ICS IMMEDIATELY after entering the target routine so the return location can be stored before any other IC saves. The UT1750AR reads the ICS using the RISC Input instruction. The configuration of the ICS is shown below.

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICS19	ICS18	ICS17	ICS16	ICS15	ICS14	ICS13	ICS12	ICS11	ICS10	ICS9	ICS8	ICS7	ICS6	ICS5	ICS4	ICS3	ICS2	ICS1	ICS0

MSB LSB

Figure 20. RISC Instruction Counter Save Register (ICS)

Pipe Register (PIPE)

The PIPE Register (figure 21) holds the pre-fetched MIL-STD-1750A instruction. The UT1750AR reads the PIPE Register with the RISC I/O instruction.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPE15	PIPE14	PIPE13	PIPE12	PIPE11	PIPE10	PIPE9	PIPE8	PIPE7	PIPE6	PIPE5	PIPE4	PIPE3	PIPE2	PIPE1	PIPE0

MSB LSB

Figure 21. The PIPE Register (PIPE)

Program Register (PR)

The Program Register holds the present MIL-STD-1750A instruction. Figure 22 shows the configuration of the Program Register (PR).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
Opcode								IRS				IRD			

MSB LSB

Figure 22. Program Register (PR)

Program Counter Register (PC)
The Program Counter Register (PC) (figure 23) contains the 16-bit address for the present MIL-STD-1750A instruction. The RISC I/O instruction reads from or writes to the PC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

MSB LSB

Figure 23. The Program Counter Register (PC)

1750 Timer A (TA) and 1750 Timer B (TB)
The Timer A (TA) and Timer B (TB) registers, figures 24a and 24b respectively, are 16-bit binary counters as defined by MIL-STD-1750A. The RISC I/O instruction starts, halts, reads, and loads them. When one of the timers reaches its programmed time setting, such as going from FFFFH to 0000H, a timeout occurs. This timeout sets the appropriate bit in the Pending Interrupt Register (PI).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TA15	TA14	TA13	TA12	TA11	TA10	TA9	TA8	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0

MSB LSB

Figure 24a. 1750 Timer A (TA)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB15	TB14	TB13	TB12	TB11	TB10	TB9	TB8	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0

MSB LSB

Figure 24b. 1750 Timer B (TB)

SYSTEM INTERFACE

The System Interface describes how the Instruction and Operand address and data busses operate during the UT1750AR's many machine cycles and bus operations. The discussion about the UT1750AR's machine cycles and bus operations applies to both the RISC mode and the MIL-STD-1750A mode of operation, since in the 1750 mode of operation the UT1750AR executes a specialized set of RISC macros that allow the UT1750AR to emulate the MIL-STD-1750A Instruction Set Architecture.

The UT1750AR has the following seven types of machine operations or bus cycle operations:

- Data Bus Cycle Operation
- DMA Operation and Bus Arbitration
- Interrupt Operation and Exception Handling
- RISC Instruction Bus Cycle Operation
- Internal UART Operation
- Console Mode of Operation
- 1750 Instruction Memory Mapping

Operand Bus and Instruction Bus Interfaces
The UT1750AR Operand Data Bus interface supports multiple processor and Direct Memory Access (DMA) configurations. The Operand Address Bus (A15-A0), Data Bus (D15-D0), and memory control bus signals (AS, DS, R/WR, M/IO, and OP/IN) are TTL-compatible signals that may be placed in a high-impedance state. These signals are only active during bus cycles when the UT1750AR is the current bus master. On other bus cycles, these signals enter a high-impedance state so an alternate bus master can control the busses.

The four signals that make up the Arbitration Control Bus -- Bus Request (BRQ), Bus Grant (BGNT), Bus Busy (BUSY), and Bus Grant Acknowledge (BGACK) -- control the UT1750AR's Operand Data Bus arbitration process. The arbitration process allows asynchronous bus arbitration.

The Instruction Bus does not allow any type of bus arbitration. The UT1750AR is the only device permitted to access Instruction memory; this access is generally confined to reading RISC instructions the UT1750AR subsequently executes, although the RISC instruction set does provide one instruction the UT1750AR uses to alter RISC memory. This instruction is the Store Register to Instruction Memory (STRI).

The Instruction address and data busses only enter a high-impedance state when the TEST input is low.

A TYPICAL UT1750AR BUS CYCLE

Figure 25a (see page 21), a generalized diagram for a typical UT1750AR bus cycle, shows the UT1750AR's bus cycle separated into four distinct time periods (CK1 through CK4). These time periods are based on the processor clock. The UT1750AR performs a separate function during each of these four time periods.

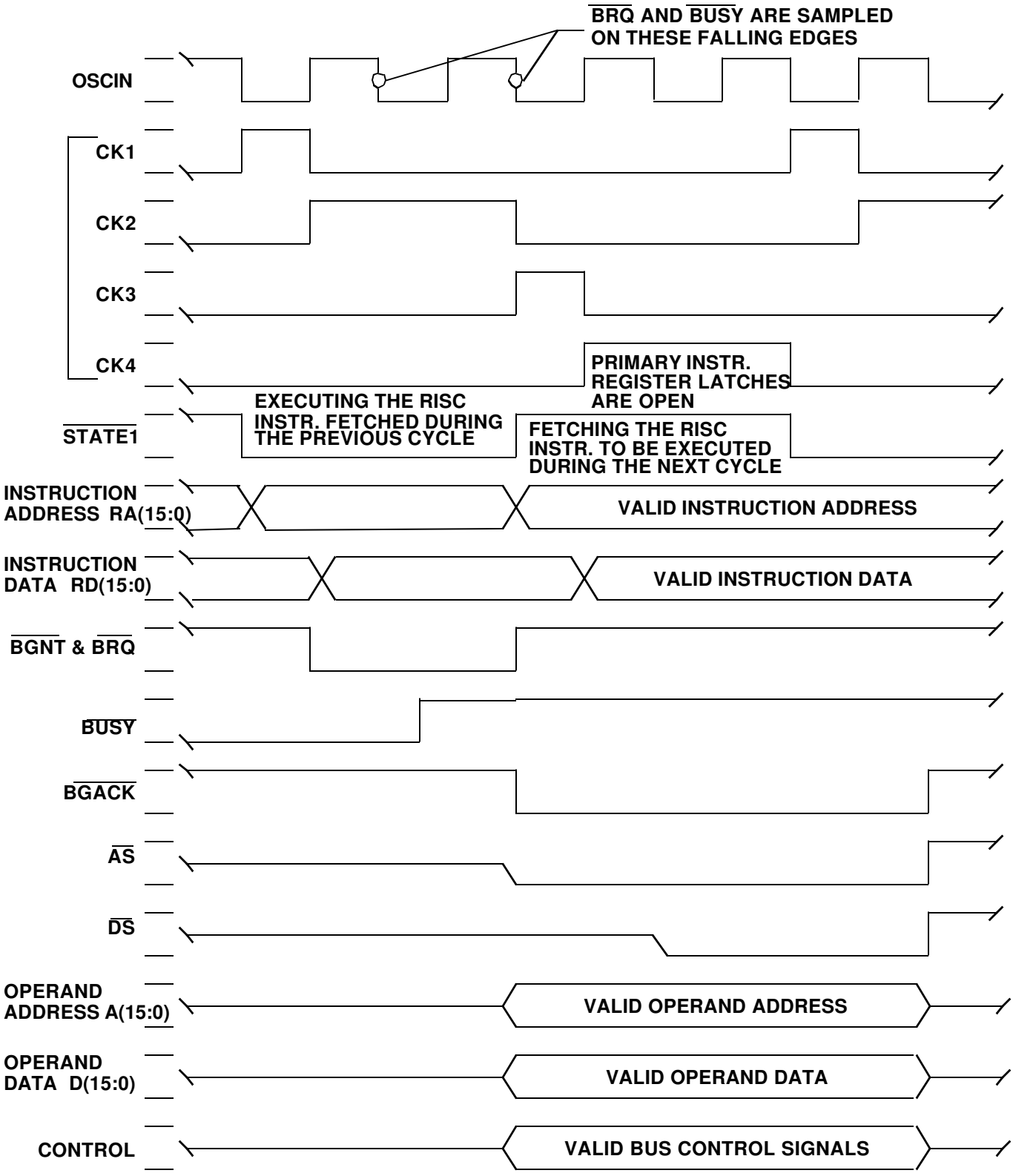


Figure 25a. Typical UT1750AR Bus Cycle With Extended Clock Cycles

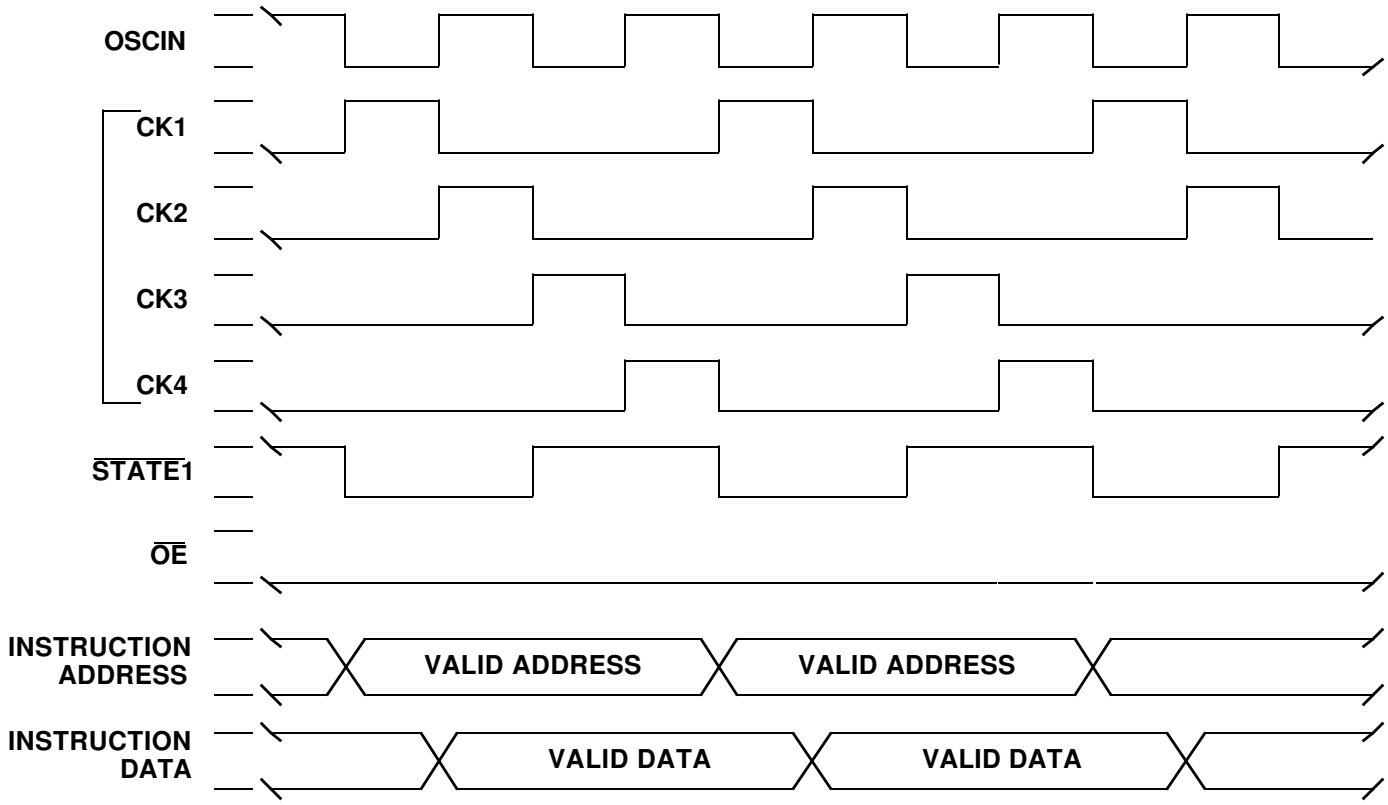


Figure 25b. Typical UT1750AR Bus Cycle

During the time period CK1, the UT1750AR begins executing the instruction in the Primary Instruction Register (PIR). The instruction executed is the instruction the UT1750AR fetched during the previous bus cycle, thus the overlapping fetch and execute cycles of the UT1750AR. During CK1, the RISC address for the next instruction to fetch from memory becomes valid. Also, the STATE1 output goes low, indicating the UT1750AR is executing an instruction.

At the beginning of time period CK2, the data addressed during CK1 becomes valid. The following conditions extend time period CK2 one clock cycle: (1) Executing a STRI instruction, (2) Executing a LRI instruction, or (3) Executing any instruction with Long Immediate data. The UT1750AR also extends clock period CK2 because of the Operand bus arbitration process. The UT1750AR samples the logical AND combination of the Bus Busy ($\overline{BUSY}$) and Bus Grant ($\overline{BGNT}$) inverted on the falling edge of CK2. If this combination is low during the falling edge of CK2, time period CK2 extends until the combination of the two signals is high, indicating the UT1750AR now controls the Operand busses. The STATE1 output remains low for the entire CK2 time period.

At the beginning of time period CK3, the $\overline{STATE1}$ output goes high indicating the next instruction is being fetched from memory. The UT1750AR's Operand address and data busses become active at the beginning of CK3 along with the Bus Grant Acknowledge ($\overline{BGACK}$), the Address Strobe ($\overline{AS}$), the Memory or I/O (M/$\overline{IO}$), the Operand/ Instruction (OP/IN), and the Read/ Write (R/$\overline{WR}$) signals.

After time period CK4 starts, the transparent latches that make up the Primary Instruction Register open up allowing the UT1750AR to input the instruction from RISC memory. Since the instruction being executed requires Operand data, the Data Strobe ($\overline{DS}$) goes active on the falling edge of the processor clock, one-half clock period after the rising edge of CK4. The UT1750AR now samples the Data Transfer Acknowledge ($\overline{DTACK}$) signal on the next and every subsequent rising edge of the processor clock. If $\overline{DTACK}$ is not low, the UT1750AR extends time period CK4 until $\overline{DTACK}$ becomes active or until an error condition is detected -- either Bus Error ($\overline{BTERR}$) or Memory Protect (MPROT) becomes active. STATE1 remains high during the entire CK4 time period.

The Processor bus cycle just described is for an instruction that requires some type of Operand data. Figure 25b shows a UT1750AR bus cycle when no Operand data is required. This cycle is typical of the bus cycle occurring for instructions that only require internal processing. An example of this type of instruction is a Move Register-to-Register instruction. For this type of instruction, each instruction requires two processor clock cycles for execution. Neither time period (CK2 nor CK4) is extended because of Operand bus arbitration or a delayed DTACK.

Operand Bus Cycle Operation

The timing diagram in figure 26 (see page 24) shows signal relationships for the UT1750AR during an operand bus cycle operation. The UT1750AR performs one of four operations involving bus cycles on the Operand busses. These bus cycles are: (1) Memory Read; (2) Memory Write; (3) I/O Read; and (4) I/O Write. The UT1750AR performs all four bus cycle operations similarly. The M/IO and R/WR signals determine the precise type of bus cycle operation. For the following discussion, please refer to figure 26.

When the Operand bus arbitration process is complete and the UT1750AR controls the Operand address and data busses, time period CK3 begins. Because the UT1750AR took control of the Operand busses at the beginning of time period CK3, BGACK becomes active. STATE1 transitions from low to high and AS goes active low. At the same time, the following signals become valid: R/WR, M/IO, OP/IN, and the Operand Address Bus. The three control signals 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, DS goes active low. After DS has gone low, the UT1750AR samples the DTACK input on every subsequent rising edge of OSCIN to determine the duration of CK4. This bus cycle terminates one-half clock cycle after the rising edge of OSCIN when the UT1750AR detects DTACK has gone active. The UT1750AR also samples the MPROT and BTERR inputs on the same rising edge of OSCIN. These two inputs indicate an error condition and terminate the current bus cycle.

After the UT1750AR recognizes the current bus cycle is finished, AS and DS become inactive (transition from low to high) on the first rising edge of OSCIN after the end of time period CK4. At this time, the Operand Address Bus (A0-A15) and the Operand bus control signals (R/WR, M/IO, OP/IN) select the memory or I/O location from which the Operand data (D0-D15) is read, or to which the Operand data (D0-D15) is written. The bus cycle completely ends one full clock cycle after the end of time period CK4 (the next rising edge of STATE1) when BGACK, R/WR, OP/IN, and the Operand address and data busses enter a high-impedance state.

DMA Operation and Bus Arbitration

Figure 27 (see page 25) shows the timing diagram of the signal relationships for the UT1750AR during a DMA operation. For DMA operations, multiprocessor, and Operand bus arbitration functions, the UT1750AR provides four active-low control signals for managing the Operand bus and preventing bus contention. These signals are Bus Request (BRG), Bus Grant (BGNT), Bus Busy (BUSY), and Bus Grant Acknowledge (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 (BRO)

The UT1750AR generates BRG to indicate a request to use the Operand busses. When the UT1750AR controls the Operand busses, if it then requires successive bus cycles, multiple Bus Requests are not generated. The UT1750AR retains control of the busses by keeping the BGACK signal active until it no longer requires the busses.

Bus Grant (BGNT)

An external arbiter generates this input indicating to the UT1750AR that it has the highest priority. This informs the UT1750AR to control the Operand busses as soon as the present bus master relinquishes bus control by setting BUSY = 1.

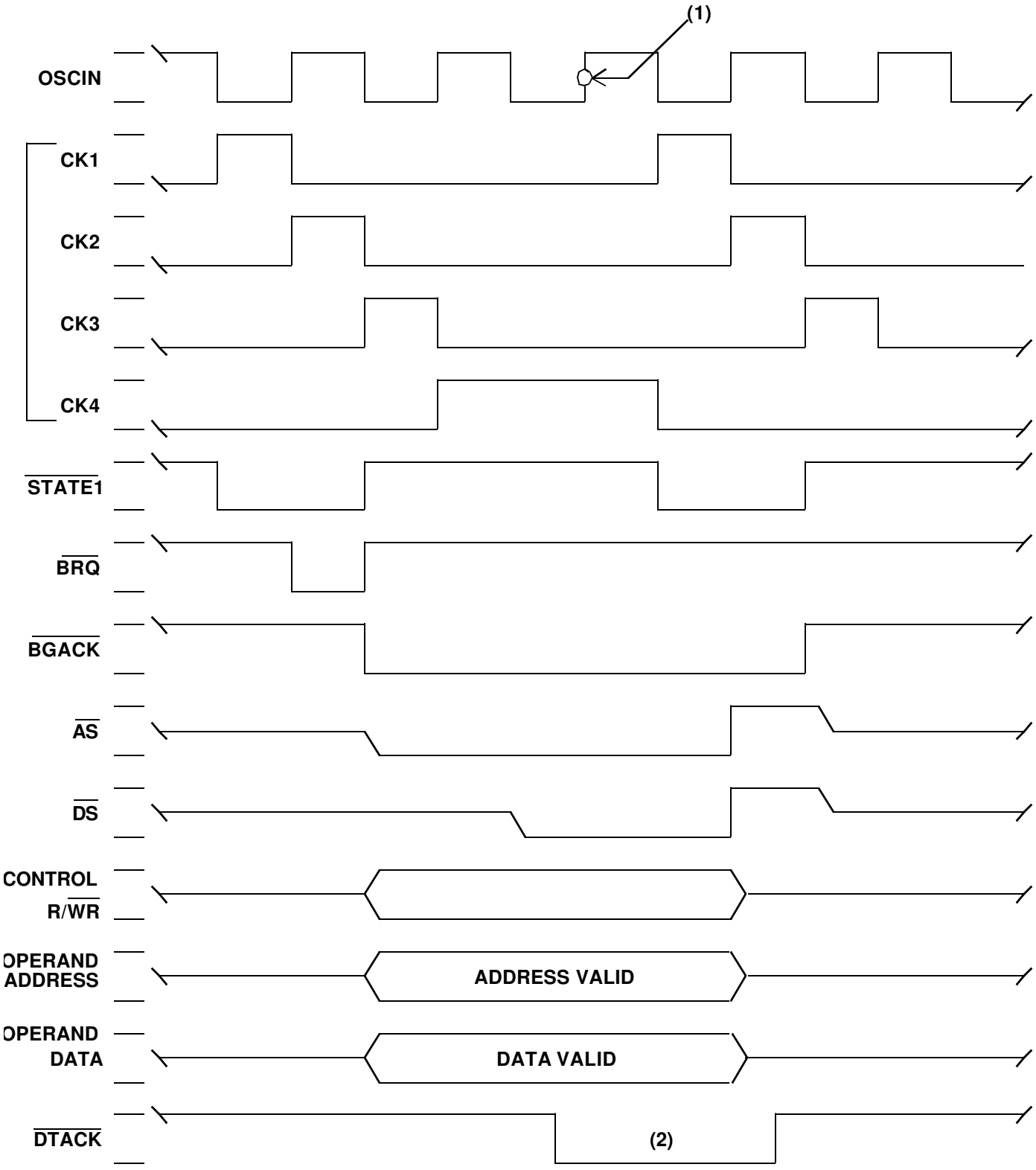
Bus Busy (BUSY)

Another bus master generates BUSY input to the UT1750AR, indicating another bus master is using the bus.

Bus Grant Acknowledge (BGACK)

The UT1750AR generates this signal to indicate it is the present bus master. BGACK enters a high-impedance state when the UT1750AR gives up control of the Operand busses.

The UT1750AR requests control of the Operand busses at the beginning of time period CK2 by asserting BRG. On every subsequent falling edge of OSCIN, the UT1750AR samples the BGNT and BUSY inputs. When the UT1750AR detects on the falling edge of OSCIN that BGNT has gone low and BUSY has gone high, this tells the UT1750AR that it is the new bus master and can now control the Operand busses. The UT1750AR locks out any other bus master from controlling the Operand busses by asserting BGACK at the beginning of time period CK3 and holding BGACK active until it is ready to give up control of the Operand busses. The UT1750AR holds the BGACK signal active until the beginning of the CK3 time period of the next bus cycle when the UT1750AR no longer controls the Operand busses.

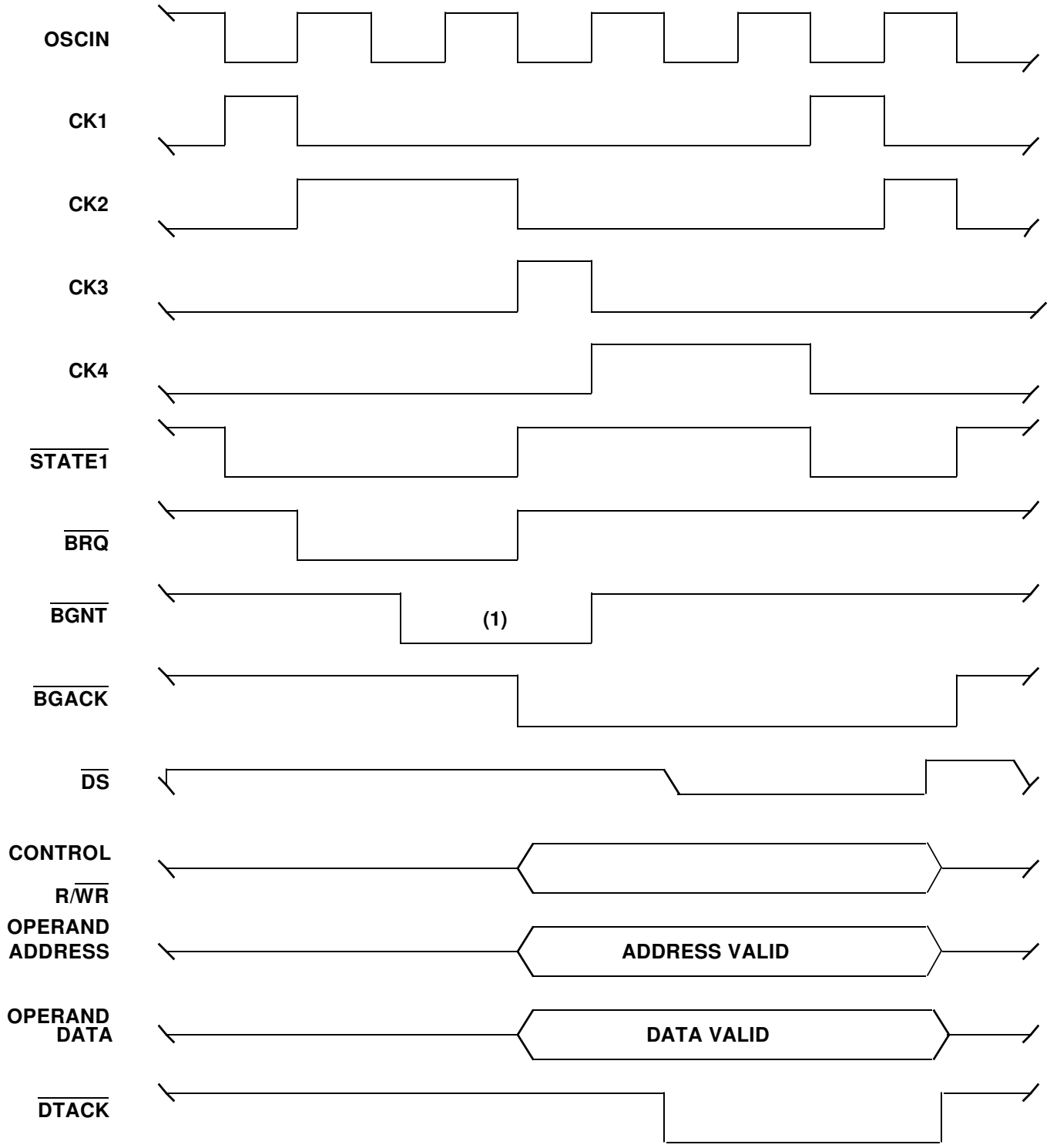


Note:

(1) DTACK must be active bythis edge to avoid wait states.

(2) DTACK is sampled by the rising edges of OSCIN.

Figure 26. Typical UT1750AR Data Bus Cycle Operation



Note:

1. BGNT is sampled by the falling edges of OSCIN. Wait states are inserted until BGNT is low and BUSY is high.

Figure 27. Typical UT1750AR DMA Bus Cycle

Table 1. Interrupt Definitions

INTERRUPT NUMBER	DESCRIPTION
0 (Highest Priority)	Power-Down Interrupt.Cannot be masked or disabled.
1	Machine Error. Cannot bedisabled.
2	INT0. External user interrupt.
3	Floating-point overflow.
4	Fixed-point overflow.
5	Branch Executive. Cannot be masked or disabled.
6	Floating-point underflow.
7	1750 Timer A (If implemented).
8	INT1. External user interrupt.
9	1750 Timer B (If implemented).
10	INT2. External user interrupt.
11	INT3. External user interrupt.
12	Input/Output level 1.
13	INT4. External user interrupt.
14	Input/Output level 2.
15 (Lowest Priority)	INT5. External user interrupt.

Interrupt Operation and Exception Handling

The UT1750AR supports 16 levels of interrupts (table 1). Eight (INT0 through INT5, IOL1, and IOL2) of the 16 interrupts are externally available for system use when the UT1750AR operates in the RISC mode. The UT1750AR internally defines the remaining interrupts for specific purposes. The UT1750AR internally prioritizes the 16 interrupts; Interrupt 0 (Power Down Interrupt) has the highest priority, and Interrupt 15 (INT5) has the lowest. Interrupts 0 and 5 are cleared when a Master Reset ($\overline{\text{MRST}}$) is asserted.

All the UT1750AR's 16 interrupts are edge-triggered, except Interrupt 3 (Floating-Point Overflow), Interrupt 5 (Executive Call), and Interrupt 6 (Floating-Point Underflow). If any one of the 16 interrupts becomes active, the UT1750AR latches the bit corresponding to the active interrupt into the Pending Interrupt Register (PI). The program can now read the PI to determine which of the 16 interrupts has occurred.

When the UT1750AR is operating in the RISC mode and an interrupt alters the RISC program flow, the UT1750AR first saves the present value of the Instruction Counter (IC) in the Instruction Counter Save Register (ICS), and then disables the interrupts. The UT1750AR then loads the IC with the memory location (table 2) corresponding to that interrupt.

When programming the UT1750AR, the ICS must be read with an Input instruction before the interrupts are re-enabled or before executing a CALL or JC (BR) instruction to assure that the return address in the ICS is not overwritten. The CALL instruction also saves the IC in the ICS 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. Therefore, for CALL instructions the system programmer should reserve register pair XR16 for ICS storage; for interrupts, the system programmer should reserve register pair XR18 for ICS storage. When nested CALLS or interrupts are encountered, the address values stored in register pairs XR16 and XR18, respectively, must be stored in system memory to provide the UT1750AR with full return information.

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

When the UT1750AR is in the 1750 mode, the UT1750AR handles the Interrupt Linkage Pointer Address and Interrupt Service Pointer Address with the MIL-STD-1750A emulation programming stored in the RISC PROMs. The addresses used for each of the 16 interrupts are in table 3.

Any one of the 16 UT1750AR interrupts can be enabled at any time during processor operation by setting the appropriate bit in the Interrupt Mask Register (MK). If an interrupt occurs but happens to have its corresponding bit masked out in the MK, then the UT1750AR ignores that interrupt, although the Power-Down Interrupt (Interrupt 0) and the Branch Executive Interrupt (Interrupt 5) cannot be masked or disabled.

RISC Instruction Bus Cycle Operation

The Instruction Bus Cycle Operation refers to the only two RISC instructions that can manipulate the data in the RISC memory. These two RISC instructions are Store Register to Instruction Memory (STRI) and Load Register from Instruction Memory (LRI).

Table 3. UT1750AR MIL-STD-1750 Interrupt Pointer Addresses

INTERRUPT NUMBER	INTERRUPT LINKAGE POINTER ADDRESS (HEX)	INTERRUPT SERVICE POINTER ADDRESS (HEX)
0	20	21
1	22	23
2	24	25
3	26	27
4	28	29
5	2A	2B
6	2C	2D
7	2E	2F
8	30	31
9	32	33
10	34	35
11	36	37
12	38	39
13	3A	3B
14	3C	3D
15	3E	3F

STRI Instruction Bus Cycle Operation

During an STRI instruction, RISC instruction data moves from the UT1750AR to the RISC instruction memory. Figure 28 (see page 28) shows the timing diagram of the signal relationships for the UT1750AR during an STRI Instruction Bus Cycle Operation.

Before the UT1750AR executes the STRI instruction, the system programmer must load the UT1750AR's Accumulator (ACC) with the RISC address which will receive the data. When the ACC is loaded with the address information, the UT1750AR 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 RISC address bus (RA0-RA20) and the $\overline{\text{STATE1}}$ output becomes active, indicating the UT1750AR is executing a RISC instruction.

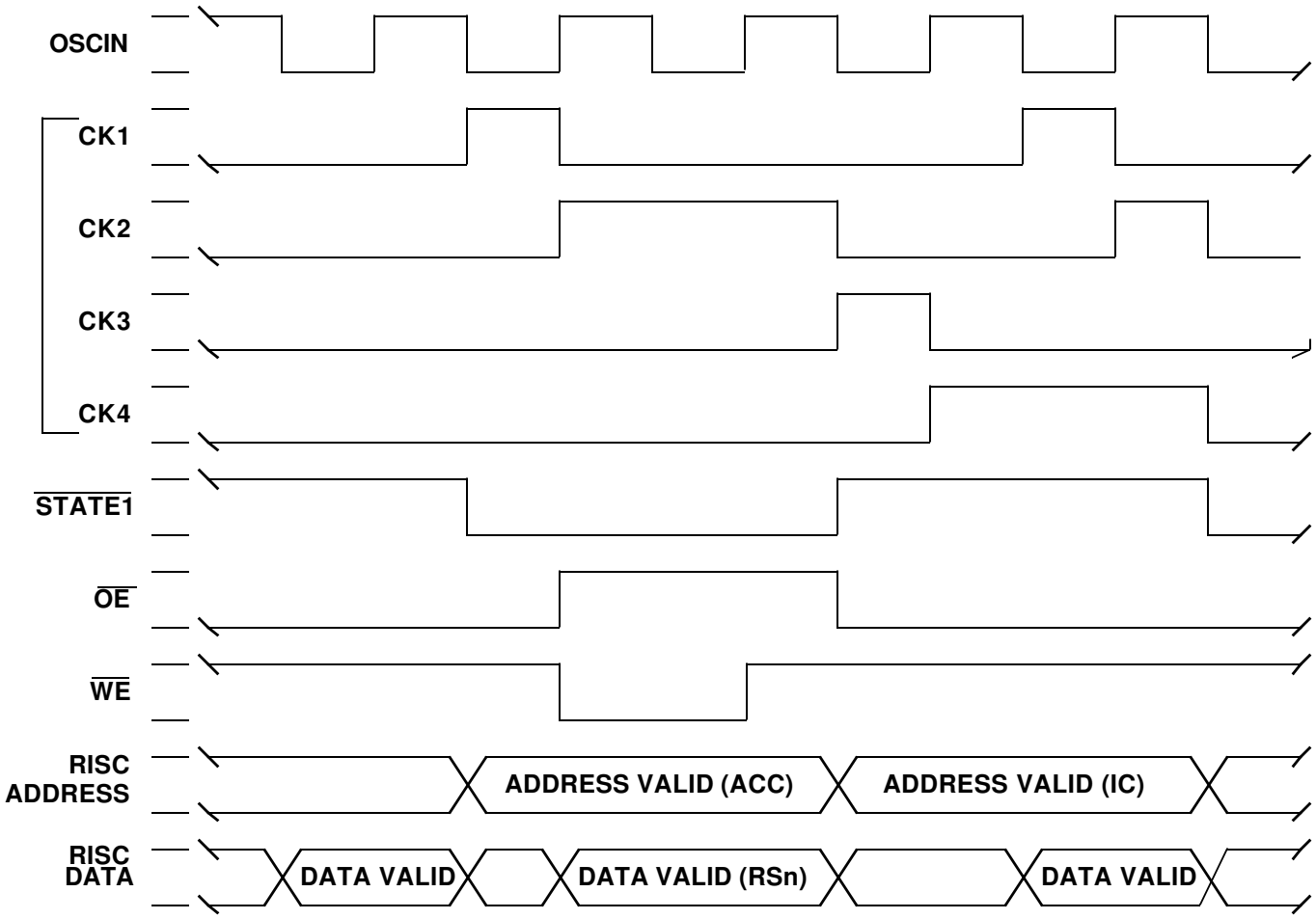


Figure 28. STRI Instruction Typical Timing

- The UT1750AR de-asserts the Output Enable(RISC Instruction) ($\overline{OE}$). This inhibits the RISC instruction from placing any data on the RISC data bus.

- The UT1750AR asserts the Write Enable (RISC Instruction) ($\overline{WE}$) so the UT1750AR can write to RISC Instruction memory.

- The data from the register selected in the STRI instruction is valid on the RISC Data bus during time period CK2.

LRI Instruction Bus Cycle Operation

During an LRI instruction, the UT1750AR moves the RISC instruction data from the RISC instruction memory to the UT1750AR. Figure 29 shows the timing diagram of the signal relationships for the UT1750AR during an LRI Instruction Bus Cycle Operation.

Just as with the STRI instruction, before the UT1750AR executes the LRI instruction the system programmer must load the UT1750AR's Accumulator (ACC) with the RISC address from which the data will be read. After the ACC is loaded with the address information, LRI instruction execution can take place.

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 RISC Address bus (RA0-RA20) and the STATE1 output becomes active indicating the UT1750AR is executing a RISC instruction.

The data on the RISC Data bus is read into the UT1750AR during time period CK2. The function of the remainder of the bus cycle (time periods CK3 and CK4) is the same as for other RISC instructions. STATE1 is high, indicating the next RISC instruction is being fetched from memory and is ready for execution during the next bus cycle.

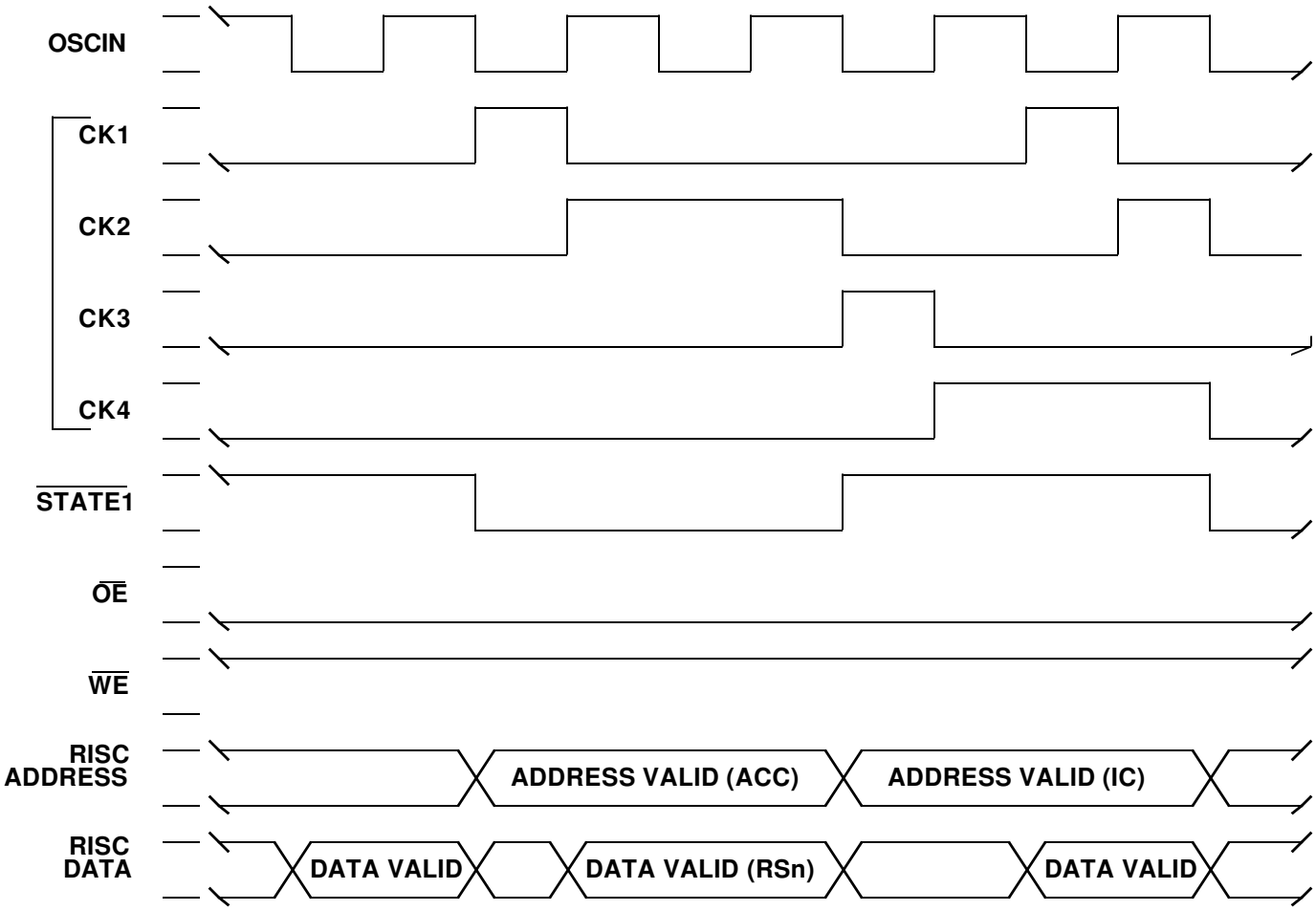


Figure 29. LRI Instruction Typical Timing

INTERNAL UART OPERATION

The UT1750AR has an internal UART. Figure 30 (see page 30) shows a diagram of the UT1750AR connected to a serial data bus. The UART operates at a fixed frequency of 9600 baud with eight data bits, one stop bit, and odd parity. The TIMCLK input fixes the baud rate of the UART. This input also controls the frequency of the internal 1750 timer registers (TA and TB).

The UART's Transmitter Buffer Register (TXMT) and Receiver Buffer Register (RCVR) are UT1750AR internal registers and are treated as such when programming the UT1750AR. The status of the UT1750AR's internal UART is read from the System Status Register (STATUS) bits 7 through 0.

UART Transmitter Operation

The transmitter portion of the UT1750AR'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 UT1750AR 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. If bit 2 of the STATUS is high, the UART transmitter is ready for data. Bit 1 is low during the serial transmission and transitions to a high when a transmission from the Transmitter Register is complete.

To initiate a serial data transmission, the system designer must first load the data to transmit into the Transmitter Buffer Register with the output instruction. 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 UT1750AR automatically transfers the data word into the Transmitter Register. After the transfer is complete, TE goes low and TBE returns high indicating a serial transmission is about to begin and the next data word can be loaded into the Transmitter Buffer Register.

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 continues until completion of all serial data transmission.

UART Receiver Operation

The UT1750AR'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 System Status Register (STATUS) contains error information about the serial data in the RCVR. These four error bits are (1) Bit 7, the Receiver Error (RE), which is the logical OR combination of the other three error bits; (2) Bit 6, an Overrun Error (OE); (3) Bit 5, a Framing Error (FE); and (4) Bit 4, a 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.

When the UT1750AR is ready to receive serial data through the internal UART, it must poll the STATUS to determine when the Data Ready (DR) bit transitions from a low to a high to signify that the UART has indeed received a serial transmission. When DR = 1, the system programmer reads the RCVR by executing an Input instruction. The INR instruction takes the eight bits of received data in the RCVR and places this data in the least significant byte of the destination register specified in the instruction.

When the UT1750AR is finished executing the Input instruction, the system programmer can then determine the validity of the message by testing the RE bit. After the programmer has checked for a valid message, the data can be stored. If the UT1750AR is to receive more data through the UART, the programmer must return to polling the STATUS to determine the reception of the next valid serial transmission.

1750 CONSOLE MODE OF OPERATION

The UT1750AR supports a defined Console mode of operation when operating as a MIL-STD-1750 processor. The Console mode of operation is a unique mode of operation that allows the system programmer to connect the UT1750AR directly to a programmer's console. The actual console can be any type of I/O device, such as a computer terminal, that allows the programmer to interface with the UT1750AR's internal UART.

While operating the UT1750AR in the Console mode, the programmer can (1) examine and modify the UT1750AR's internal registers; (2) examine and modify the contents of the Operand memory; (3) examine and modify the contents of the RISC memory; (4) examine and modify the contents of the I/O subsystems; (5) continue the execution of a 1750 program; and (6) have the UT1750AR begin program execution from any address.

The CONSOLE input is a discrete input to the UT1750AR and is read as bit 3 in the System Status Register (STATUS). The definition of this input is not inherent to the UT1750AR, but is defined only by the programming within the RISC PROMs. Since, as with many other operational features of the UT1750AR, the Console mode is a function of the programming in the RISC PROMs, the user can tailor the UT1750AR's Console mode to a specific application. For example, the user can modify the Console mode program in the RISC PROMs so when the UT1750AR executes this code, it performs a system-level test. When complete, the UT1750AR reports the results to the programmer's console where the user can ascertain the functional integrity of the system.

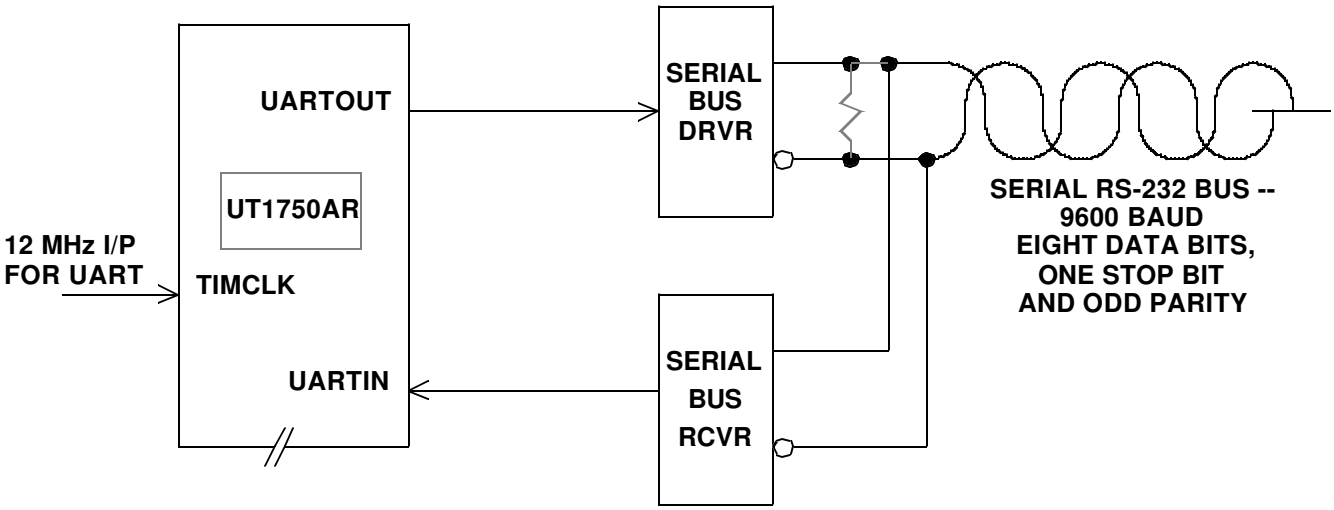


Figure 30. Serial Data Bus Interface to the UT1750AR

Entering the Console mode
The UT1750AR enters the Console mode in one of two ways:

- If the CONSOLE input is active (high) when the UT1750AR is reset ($\overline{MRST}$ = 0).

- Upon executing a Breakpoint (BPT) instruction. When the UT1750AR encounters a BPT instruction, the UT1750AR first reads the data in the STATUS. If the Console Enable bit (bit 4) in the STATUS is low, t he UT1750AR treats the BPT instruction like a NOP. If, on the other hand, the Console Enable bit is high, the UT1750AR enters the Console mode and waits for the first console ommand.

When the UT1750AR enters the Console mode, it begins executing the program stored in the RISC PROMs. The UT1750AR initially sets its internal UART as the default console interface. Although the internal UART is the default console interface, the user can select another interface, such as a MIL-STD-1553 bus, another external serial interface, or a parallel interface, as the console interface by changing the programming in the RISC PROMs.

Using the Console mode
To control the UT1750AR with the Console mode, 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 UT1750AR's Console mode are described in detail in the following sections. The UT1750AR can receive these Console control commands with its internal UART, decode them, and then take the appropriate action. *All ASCII characters must be capitalized for the UT1750AR 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.

The Examine (E) Command
The Examine Command has four variations:
(1) EIxxxx - The Examine Instruction (RISC) 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 Console continues to display the contents of contiguous memory locations until any non-Space character is received. When the Console receives a non-Space character, it terminates EI command execution and waits for the next valid Console 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 UT1750AR's internal registers.

After the UT1750AR 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 UT1750AR. The Console mode displays the registers one after another in the following order: R0 through R15, 1750 Status Word (SW), Pending Interrupt Register (PI), Interrupt Mask Register (MK), Fault Register (FT), 1750 Program Counter (PC), 1750 Timer A (TA) and Timer B (TB). The UT1750AR continues to display its registers until the UT1750AR receives a non-Space character or until the UT1750AR has displayed the complete list of registers. At this time the UT1750AR terminates the ER command and waits for the next valid Console command.

The Modify (M) Command
The Modify Command has four variations:
(1) MIxxxx,vvvv - The Modify Instruction (RISC) 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 UT1750AR 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 UT1750AR'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 UT1750AR 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 UT1750AR register. Modifying additional registers requires transmitting a separate MR command for each change.

The Continue Execution (C) Command
The Continue Execution Command allows the user to resume program execution from the point where the Console 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.

Table 4. Console Command Register ID Numbers

REGISTER	ID NUMBER (HEX)
R0	00
R1	01
R2	02
R3	03
R4	04
R5	05
R6	06
R7	07
R8	08
R9	09
R10	0A
R11	0B
R12	0C
R13	0D
R14	0E
R15	0F
SW	10
PI	11
MK	12
FT	13
TA	14
TB	15
DISCON	16
DISCOFF	17

The Run From Memory Location (R) Command
The Run From Memory Location Command allows the user to start program execution from any point within the 64K operand memory space. This command takes the form Rxxxxn where "xxxx" denotes the 16-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.

Exiting the Console mode

The UT1750AR exits the Console mode of operation by executing either Continue Execution (C) command or a Run From Memory Location (R) command. After the UT1750AR leaves the Console mode, it resumes operating in a normal 1750 mode.

1750 Mode Built-In Test

In the 1750 mode of operation, the UT1750AR features a built-in test function which executes upon device power-up or reset. The built-in test function performs "stuck-at" tests on all internal UT1750AR registers, Timer A, and Timer B. In addition to testing the UT1750AR registers, the built-in test also checks for the 1750 emulation code. The 1750 emulation ROM is tested via a checksum test of all memory locations.

Test failures are recorded in the UT1750AR's Fault Register.

- UT1750AR failure: Fault Register = 5 (hex)
- Emulation code checksum failure: Fault Register = 6 (hex)
- Output Discrete 2 (RA17/OD1) = Active (logic 1)

If the CONSOLE pin is asserted (logic 1) during power-up or reset, the emulation code will enter the Console mode after finishing the built-in tests. The Fault Register contents indicate the failure mode.

A failure in the built-in test without the Console mode implemented results in Output Discrete 2 (RA17/OD1) being set to a logic one. In addition to the Output Discrete 2 being set to a logic one, the UT1750AR will not begin program execution if failure occurs in PI or FT registers.

1750 XIO

The UT1750AR emulation code does not implement the following optional XIO command fields and mnemonics:

2008 OD-- Output Discretes
200A RNS-- Reset Normal Power-Up Discrete
4001 CLC-- Clear Console
4003 MPEN-- Memory Protect Enable
50XX LMP-- Load Memory Protect RAM
A001 RIC1-- Read Input/Output Interrupt Code, Level 1
A002 RIC2-- Read Input/Output Interrupt Code, Level 2
A008 RDOR--Read Discrete Output Register
A009 RDI-- Read Discrete Input
A00B TPIO -- Test Programmed Output
D0XX RMP--Read Memory Protect RAM

The UT1750AR internal UART is I/O mapped as follows:

XIO RA, FFFE (hex)- RISC Status Register contents loaded into register RA

XIO RA, FFFF (hex) - Contents of UART Receiver Buffer

Register (RCVR) loaded into register R

XIO RA, 7FFF (hex)- Contents of register RA loaded into UART Transmitter Buffer Register (TBR)

MIL-STD-1750 Console XIO's result in the following:

1750 INSTRUCTION	EFFECTIVE RESULT
4000 CO	XIO RA, 7FFF (hex)
4001 CLC	NOP
C000 CI	XIO RA, FFFF (hex)
C001 RCS	XIO RA, FFFE (hex)

1750 INSTRUCTION MEMORY MAPPING

The UT1750AR emulates the MIL-STD-1750A ISA by mapping each of the 1750A opcodes into a specific location within the UT1750AR's RISC memory space. This memory mapping is accomplished by internal UT1750AR hardware. The memory mapping for the valid 1750 opcodes between 00H and 4FH is shown in table 5.

For the Base Relative and Indexed Base Relative 1750 instructions, the UT1750AR maps multiple instructions to the same address. The UT1750AR determines the correct operation for these opcodes by using the Input Register (INR) RISC instruction. For more information on the operation of the INR instruction, please refer to the UT1750AR Assembly Language Manual.

For the remainder of the valid 1750 opcodes between 50H and FFH, the UT1750AR follows a straightforward memory-mapping scheme. To determine the RISC memory location for these 1750 opcodes, the UT1750AR masks off the lower byte of the instruction and logically shifts the result four times to the right.

For example, the 1750 opcode for the POPM instruction is 8FxxH. The location of the POPM macro in the UT1750AR's RISC memory space is 08F0H.

Table 5. RISC Macro Locations for Valid 1750 Opcodes Between 00H and 4FH

1750 INSTRUCTION	1750 OPCODE(S)	RISC MACRO LOCATION
LB	00 TO 03	0020
DLB	04 TO 07	0060
STB	08 TO 0B	00A0
DSTB	0C TO 0F	00E0
AB	10 TO 13	0120
SBB	14 TO 17	0160
MB	18 TO 1B	01A0
DB	1C TO 1F	01E0
FAB	20 TO 23	0220
FSB	24 TO 27	0260
FMB	28 TO 2B	02A0
FDB	2C TO 2F	02E0
ORB	30 TO 33	0320
ANDB	34 TO 37	0360
CB	38 TO 3B	03A0
FCB	3C TO 3F	03E0
LBX	400 TO 430	0030
DLBX	401 TO 431	0070
STBX	402 TO 432	00B0
DSTX	403 TO 433	00F0
ABX	404 TO 434	0130
SBBX	405 TO 435	0170
MBX	406 TO 436	01B0
DBX	407 TO 437	01F0
FABX	408 TO 438	0230
FSBX	409 TO 439	0270
FMBX	40A TO 43A	02B0
FDBX	40B TO 43B	02F0
CBX	40C TO 43C	0330
FCBX	40D TO 43D	0370
ANDX	40E TO 43E	03B0
ORBX	40F TO 43F	03F0
XIO	48	0480
VIO	49	0490
AIM	4AX1	0050
SIM	4AX2	0090
MIM	4AX3	00D0
MSIM	4AX4	0110
DIM	4AX5	0150
DVIM	4AX6	0190
ANDM	4AX7	01D0
ORIM	4AX8	0210
XORM	4AX9	0250
CIM	4AXA	0290
NIM	4AXB	02D0
BIF	4F	04F0

PROGRAMMING INTERFACE

Data Formats

The UT1750AR instruction set supports 16-bit integer single-precision data and 32-bit integer double- precision data. When the UT1750AR is operating in the 1750 mode with the 1750 emulation code in the RISC PROMs, the UT1750AR can emulate 32-bit floating-point and 8-bit floating-point extended-precision data. All data is in 2's complement representation.

Figure 31a. Single 6Precision Fixed-Point Data

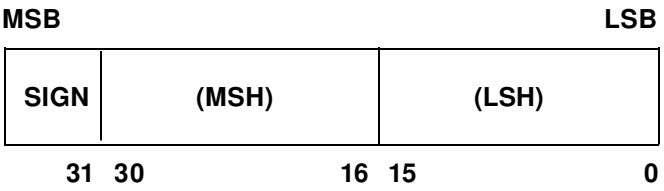


Figure 31b. Double 06Precision Fixed-Point Data

The UT1750AR represents the fixed-point data formats as a 2's complement integer with the MSB as the sign bit (figures 31a and 31b).

Operand Size

The UT1750AR'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 UT1750AR'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 UT1750AR 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 UT1750AR 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 UT1750AR's RISC instruction descriptions contain a definition of the Register Transfer Language (RTL) that the RISC Assembler uses to describe how the RISC instructions operate. The RTL description of the UT1750AR'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

Instruction Formats
The UT1750AR has three instruction formats (figure 32): (1) Register-to-Register; (2) Register-to-Short Immediate; and (3) Register-to-Immediate.

All the UT1750AR's instructions are either word (16-bit) or long-word (32-bit) in length. The only time the UT1750AR uses the long-word instruction format is for the Immediate Source Operand Address Mode.

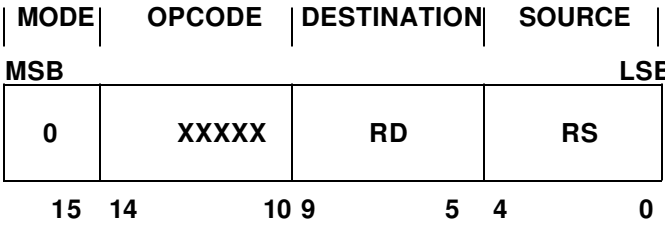


Figure 32a. RegisterX0106to-Register Instruction Format

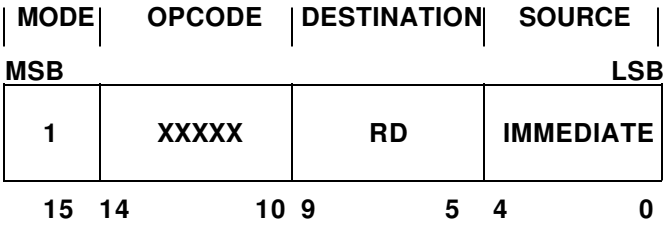


Figure 32b. RegisterX 106to-Short Immediate Instruction Format

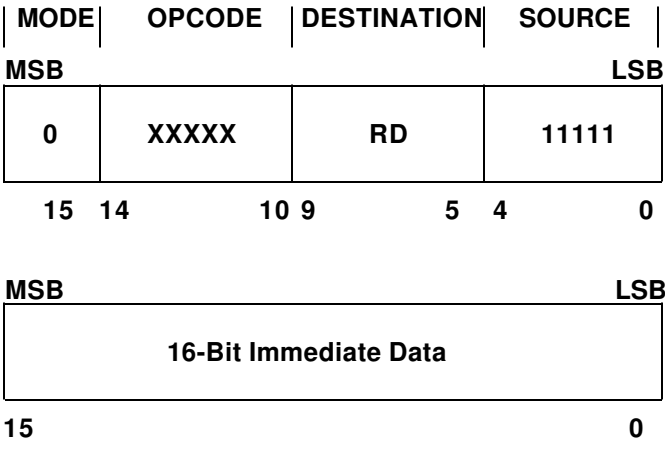


Figure 32c. Register Immediate Instruction Format

The bits in the RISC instructions are defined as follows:

M: Instruction Mode Bit. When M = 1, the UT1750AR interprets the Instruction Source field as a five-bit literal value. If M = 0, the UT1750AR uses the Instruction Source field to specify the source register for the instruction.

Opcode: This field is the five-bit opcode the UT1750AR uses to decode the RISC instruction into a machine operation.

Destination: This field specifies the register the UT1750AR uses for the destination of the instruction.

Source: This field specifies the register the UT1750AR uses for the Instruction Source.

Immediate: If needed, this field contains the 16-bits of immediate data the UT1750AR requires for the long-immediate instruction.

Operand Addressing Modes
The UT1750AR's RISC instruction set supports four basic addressing modes. All RISC instructions require a source operand and a destination operand. The destination operand is a data register (RDn or XRDn) for all RISC 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 RISC 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 UT1750AR uses during the current instruction execution. This type of memory addressing is only used with the Load (LR), Store (STR), PUSH, and POP RISC instructions.

Destination Addressing Mode
The destination operand is given explicitly for all UT1750AR RISC instructions. The UT1750AR 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

In 1750 emulation mode register pairs XR8, XR10 and XR12 have a special meaning. Register XR8 is a pointer to the MIL-STD-1750A destination register (defined as RA). Register pair XR10 is a pointer to the next register, RA+1. Register pair XR12 is a pointer to the source register.

Source Addressing Modes
The UT1750AR 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 UT1750AR uses the Data Register Direct mode, the source operand is one of the data registers. The data register is explicitly stated for all RISC instructions. The UT1750AR 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

In 1750 emulation mode register pairs XR8, XR10 and XR12 have a special meaning. Register XR8 is a pointer to the MIL-STD-1750A destination register (defined as RA). Register pair XR10 is a pointer to the next register, RA+1. Register pair XR12 is a pointer to the source register.

Literal
When the UT1750AR uses the Literal mode, the source operand is a 5-bit literal data value. The UT1750AR explicitly states this literal data value for the RISC instructions. The UT1750AR 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 UT1750AR uses the Immediate Long mode, the source operand is a 16-bit data value. The UT1750AR explicitly states this data for all RISC instructions and encodes the 16-bit data in a second 16-bit instruction word (figure 32). The UT1750AR 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 UT1750AR also supports three modes of indirect addressing: (1) Data Register Indirect; (2) Stack Pointer Indirect; and (3) Absolute.

Data Register Indirect
When the UT1750AR 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 RISC instructions. This mode is only available on the LR, STR, INR, and STR instructions. The UT1750AR 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

Stack Pointer Indirect
When the UT1750AR 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 UT1750AR 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 UT1750AR 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.

Data Movement Operations
The UT1750AR 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 UT1750AR handles the data movement for all RISC instructions.

When a RISC instruction specifies a word destination, a 16-bit result is always stored in the destination. If the RISC instruction specifies a 5-bit literal source operand, then the UT1750AR sign-extends this source data to produce a 16-bit operand. If the RISC instruction specifies a word-length source operand, there is no manipulation of the source data. If the RISC instruction specifies a long-word source operand, the UT1750AR only retains the least significant 16 bits of the result. The UT1750AR truncates the most significant 16 bits of the result.

When a RISC instruction specifies a long-word destination, a 32-bit result is always stored in the destination. If the RISC instruction specifies a 5-bit literal source operand, then the UT1750AR sign-extends this source data to produce a 32-bit operand. If the RISC instruction specifies a word-length source operand, then the UT1750AR also sign-extends this source data to produce a 32-bit operand. If the RISC 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 UT1750AR only stores eight bits of the result regardless of whether the RISC instruction specifies a word or long-word destination register.

Operation Code Matrix

The UT1750AR performs 30 basic operations, each with its own operation code. All the UT1750AR's operations are explicit, and are encoded in bits 14 through 10 of the RISC instruction (figure 32; see page 35). A list of the UT1750AR's opcodes are in table 6.

Instruction Clock Cycles

The number of processor clock cycles the UT1750AR requires to execute each of its instructions is in table 7. Table 7 specifies, for each instruction, the execution time for the three instruction types (Register-to-Register, Register-Literal, and Register-to-Long Immediate) where applicable.

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}+0.3	V
I_I	DC input current	±10	mA
T_{STG}	Storage temperature	-65 to +150	℃
I_{LU}	Latchup immunity (2)	±150	mA
P_D	Maximum power dissipation	600	mW
T_J	Maximum junction temperature	+175	℃
Θ_{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 43).
3. Test per MIL-STD-883, Method 1012.

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	℃
V_{IN}	DC input voltage	0 to V_{DD}	V

Table 6. UT1750AR Operation Code Matrix

OPCODE	MNEMONIC	DESCRIPTION
00000	MOV	Move Data
00001	LR	Load Data From Data Memory
00001	LRI	Load from RISC Instruction Memory
00001	POP	Pop from Stack
00010	STR	Store to Data Memory
00010	STRI	Store to Instruction Memory
00010	PUSH	Push to Stack
00011	CALL	Call Routine
00100	MOVC	Move and Set Condition Flags
00101	INR	Input Register
00110	OTR	Output Register
00111	--	Spare - Not Used
01000	ADD	Add
01001	ADDC	Add with Carry
01010	AB	Add Byte
01011	ADDU	Add Unsigned
01100	SUB	Subtract
01101	SUBB	Subtract with Borrow
01110	SB	Subtract Byte
01111	CMP	Compare
10000	AND	AND Logic
10001	OR	OR Logic
10010	XOR	XOR Logic
10011	NOT	NOT Logic
10100	RBR	Reset Bit
10101	SBR	Set Bit
10110	TBR	Test Bit
10111	--	Spare - Not Used
11000	SLR	Shift Logic
11001	SAR	Shift Arithmetic
11010	SCR	Shift Cyclic
11011	MULS	Signed Multiply
11100	MOVB	Move Byte
11101	SWAB	Swap Bytes
11110	DIVS	Signed Divide
11111	JC	Jump Conditionally
11111	BR	Branch Conditionally

Table 7. Execution Times for the UT1750AR RISC Instructions

MNEMONIC	UT1750AR Instruction Execution Clock Cycles REGISTER-TO-REGISTER	REGISTER-TO-LITERAL	REGISTER-TO-LONG IMMEDIATE
MOV	2	2	4
LR	3+W	N/A	4+W
LRI	N/A	4	N/A
POP	3+W	N/A	N/A
STR	3+W	N/A	4+W
STRI	N/A	4	N/A
PUSH	3+W	N/A	N/A
CALL	4	N/A	4
MOVC	2	2	4
INR	3+W	Special	4+W
OTR	3+W	2	4+W
ADD	2	2	4
ADDC	2	2	4
AB	2	2	4
ADDU	2	2	4
SUB	2	2	4
SUBB	2	2	4
SB	2	2	4
CMP	2	2	4
AND	2	2	4
OR	2	2	4
XOR	2	2	4
NOT	2	2	4
RBR	2	2	4
SBR	2	2	4
TBR	2	2	4
SLR	3+N	3+M	4+N
SAR	3+N	3+M	4+N
SCR	3+N	3+M	4+N
MULS	3+K	3+K	4+K
MOVB	2	2	4
SWAB	2	2	4
DIVS	36 OR 68	36 OR 68	37 OR 69
JC	2	N/A	4
BR	N/A	2	N/A

Where: W = Wait state(s)
M = Number of shifts where 1 < M < 16
N = Number of shifts where 1 <N < 32
J = Varies by operation
K = Between 16 and 32 if destination register is 16 bits, and between 32 and 64 if destination register is 32 bits.
N/A = Not Applicable

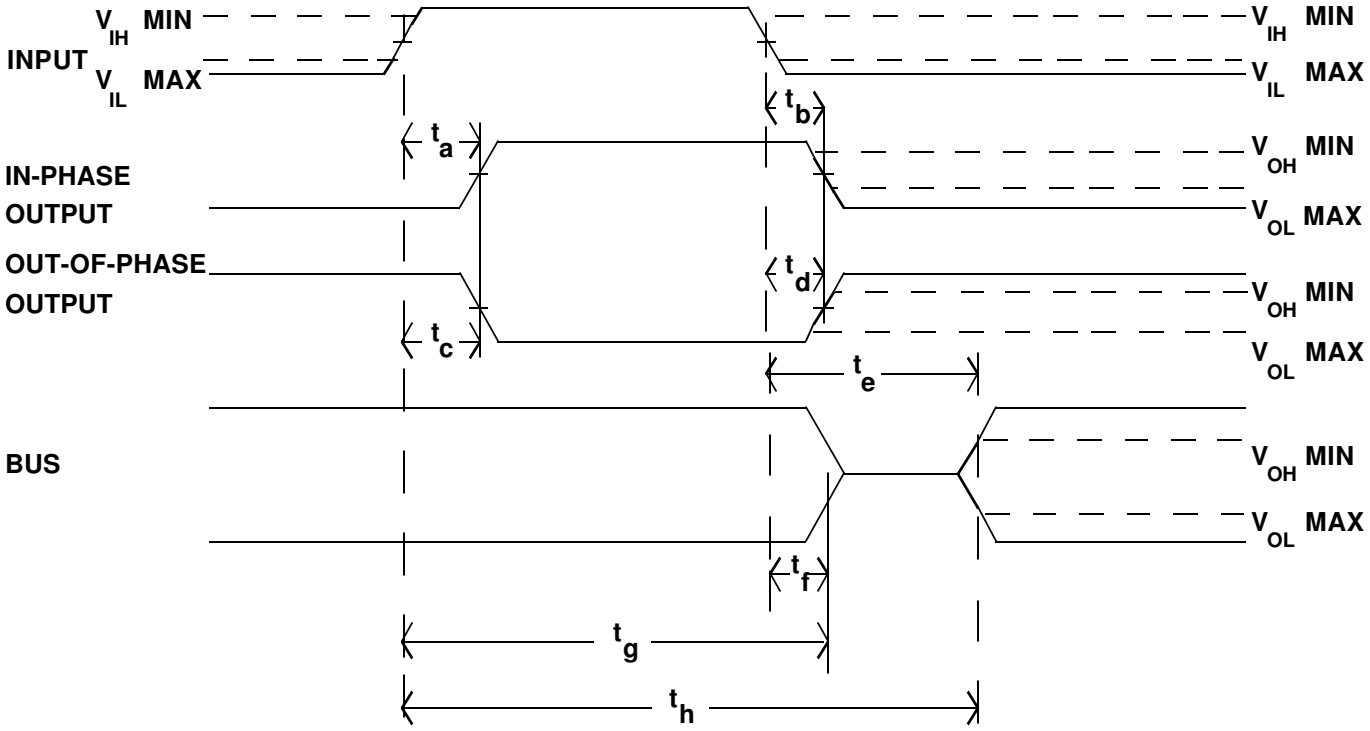
ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0V±10%; -55°C < Tc < +125°C

SYMBOL	PARAMETER	CONDITION	MINIMUM	MAXIMUM	UNIT
V_{IL}[6]	Low-level input voltage OSC inputs TTL inputs			 1.2 0.8	 V V
V_{IH}[6,7]	High-level input voltage OSC inputs TTL inputs		 3.6 2.0		 V V
I_{IN}	Input leakage current Inputs without resisters 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 80 -900	 10 900 -80	 µA µA µA
V_{OL}	Low-level output voltage TTL outputs OSC outputs	I_{OL} = 3.2mA I_{OL} = 6.4mA Note 5 I_{OL} = 100µA		0.4 0.4 1.0	V V V
V_{OH}	High-level output voltage TTL outputs OSC outputs	I_{OH} = -400µA I_{OH} = -800µA Note 5 I_{OH} = -100µA	2.4 2.4 3.5		V V V
I_{OZ}	Three-state output leakage current	$V_O = V_{DD}$ or V_{SS}	-10 -20 Note 5	+10 +20 Note 5	µA µA
I_{OS}[1,2]	Short-circuit output current	V_{DD} = 5.5V, V_O = 0V to V_{DD}	-100 -200 Note 5	+100 +200 Note 5	mA mA
C_{IN}	Input capacitance	F = 1MHz @ 0V		10	pF
C_{OUT}	Output capacitance	F = 1MHz @ 0V		15	pF
C_{IO}	Bidirectional I/O capacitance	F = 1MHz @ 0V		20	pF
I_{DD}[1, 4]	Average operating current	F = 12MHz, C_L = 50pF F = 16MHz, C_L = 50p		50 75	mA
Q_{IDD}	Quiescent current	Note 3		1	mA

Notes:

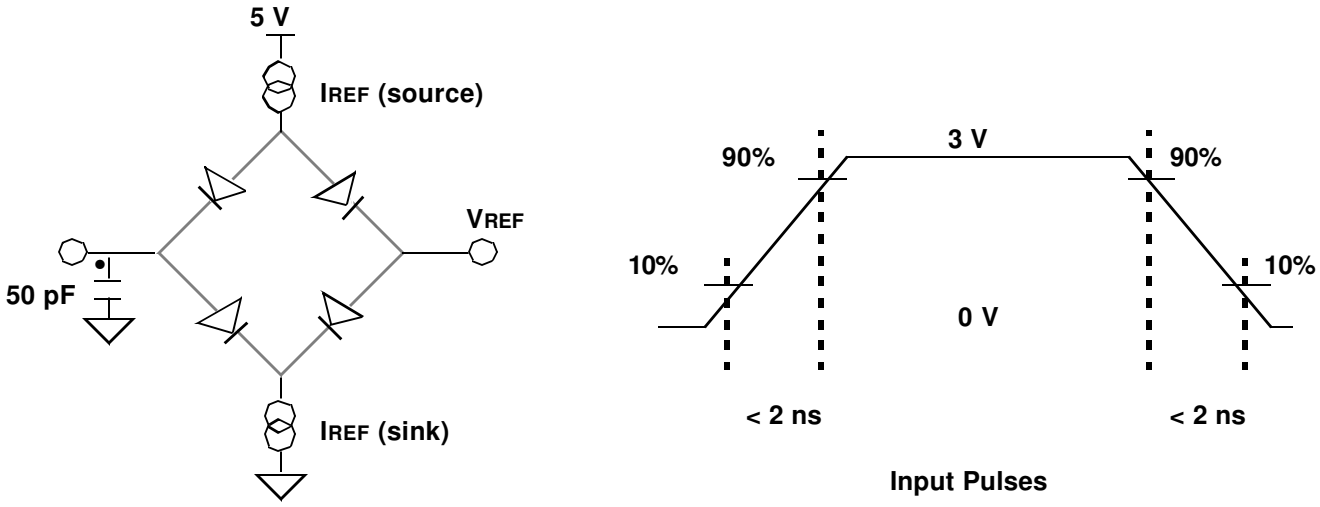
1. Supplied as a design limit but not guaranteed or tested.
2. Not more than one output may be shorted at a time for maximum duration of one second.
3. All inputs with internal pull-ups or pull-downs should be left open circuit, all other inputs tied low or high. $\overline{\text{TEST}}$ input pin asserted.
4. 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.
5. Double buffer output pins (i.e., DS, R/WR, M/IO, OP/I, AS).
6. Functional tests are conducted in accordance with MIL-STD-883 with the following input test conditions: $V_{IH} = V_{IH}$ (min) + 20%, -0%; $V_{IL} = V_{IL}$ (max) +0%, -50%, as specified herein, for TTL and CMOS compatible inputs. Devices may be tested using any input voltage within the above specified range, but are guaranteed to V_{IH} (min) and V_{IL} (max).
7. Radiation-hardened technology shall have a V_{IH} pre-irradiation of 2.2V.



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

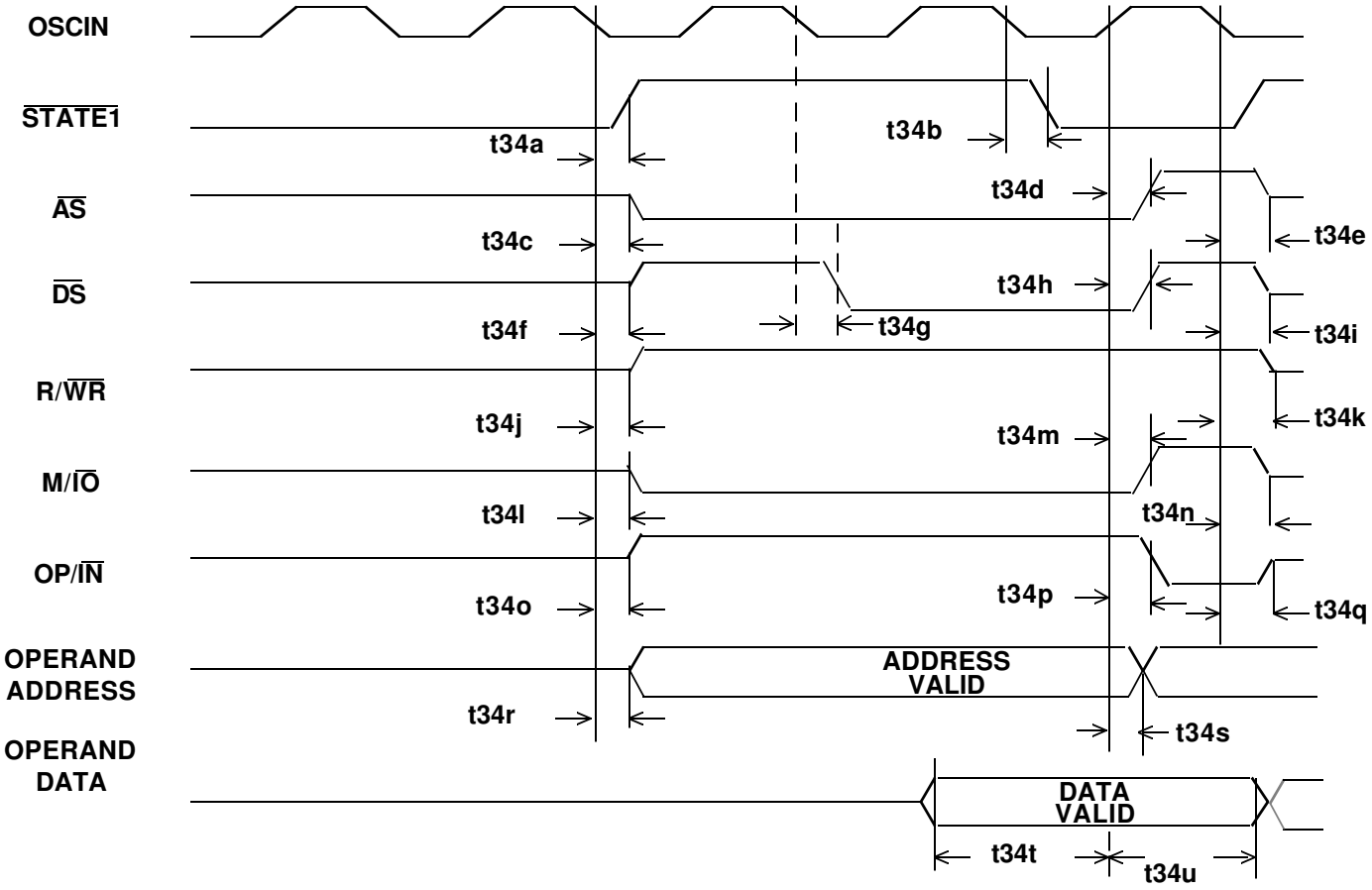
*Unless otherwise noted, all AC electrical characteristics are guaranteed by design or characterization.

Figure 33a. Typical Timing Measurements



NOTE:
50pF including scope probe and test socket.

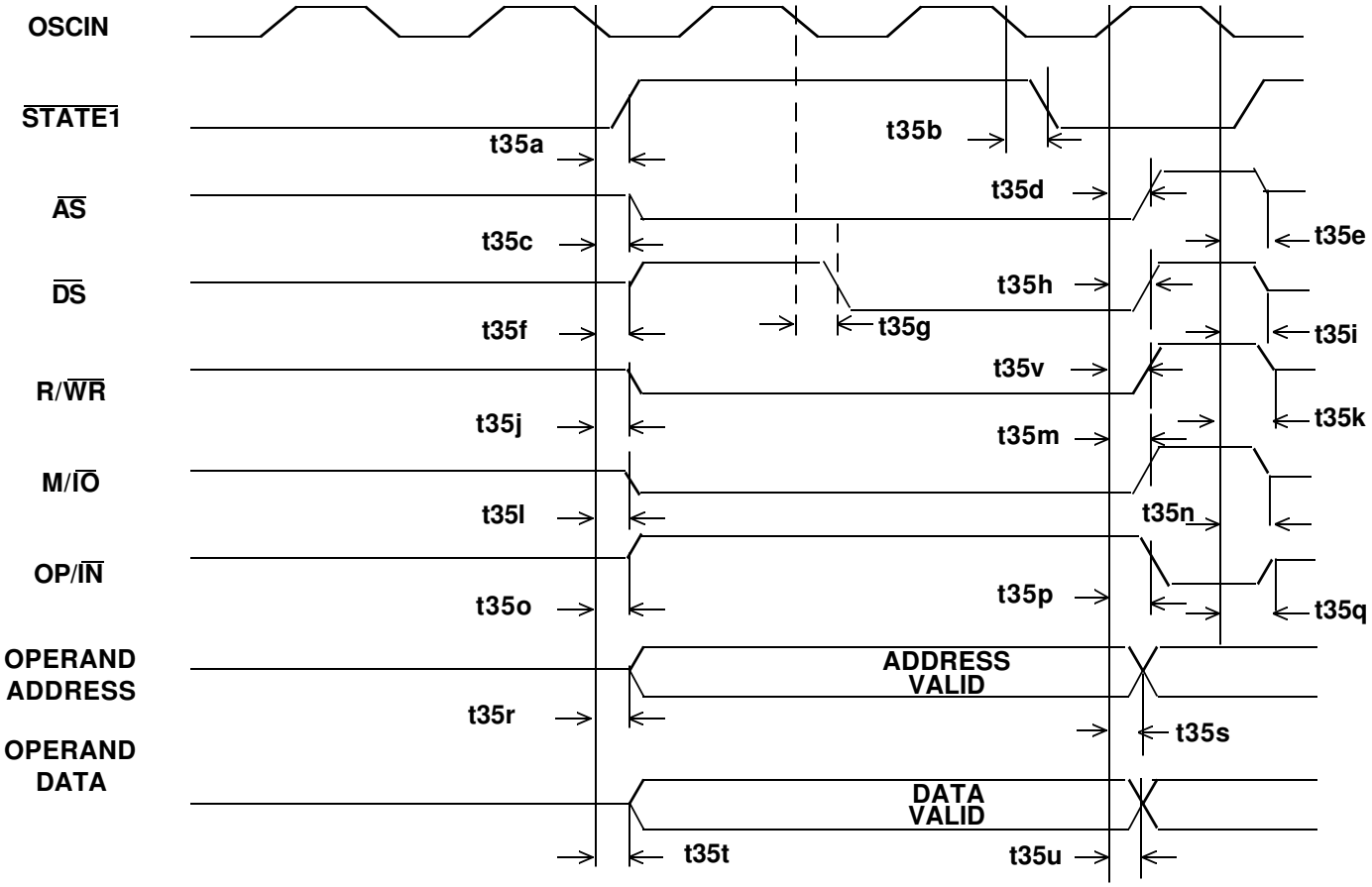
Figure 33b. AC Test Loads and Input Waveforms



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
t34c *	OSCIN low to AS active	0	51	0	42	ns
t34d *	OSCIN high to AS inactive	0	50	0	38	ns
t34e	OSCIN low to AS high Z	--	50	--	38	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
t34o *	OSCIN low to OP/IN high	0	54	0	41	ns
t34p *	OSCIN high to OP/IN low	0	71	0	53	ns
t34q	OSCIN low to OP/IN high Z	--	53	--	40	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.

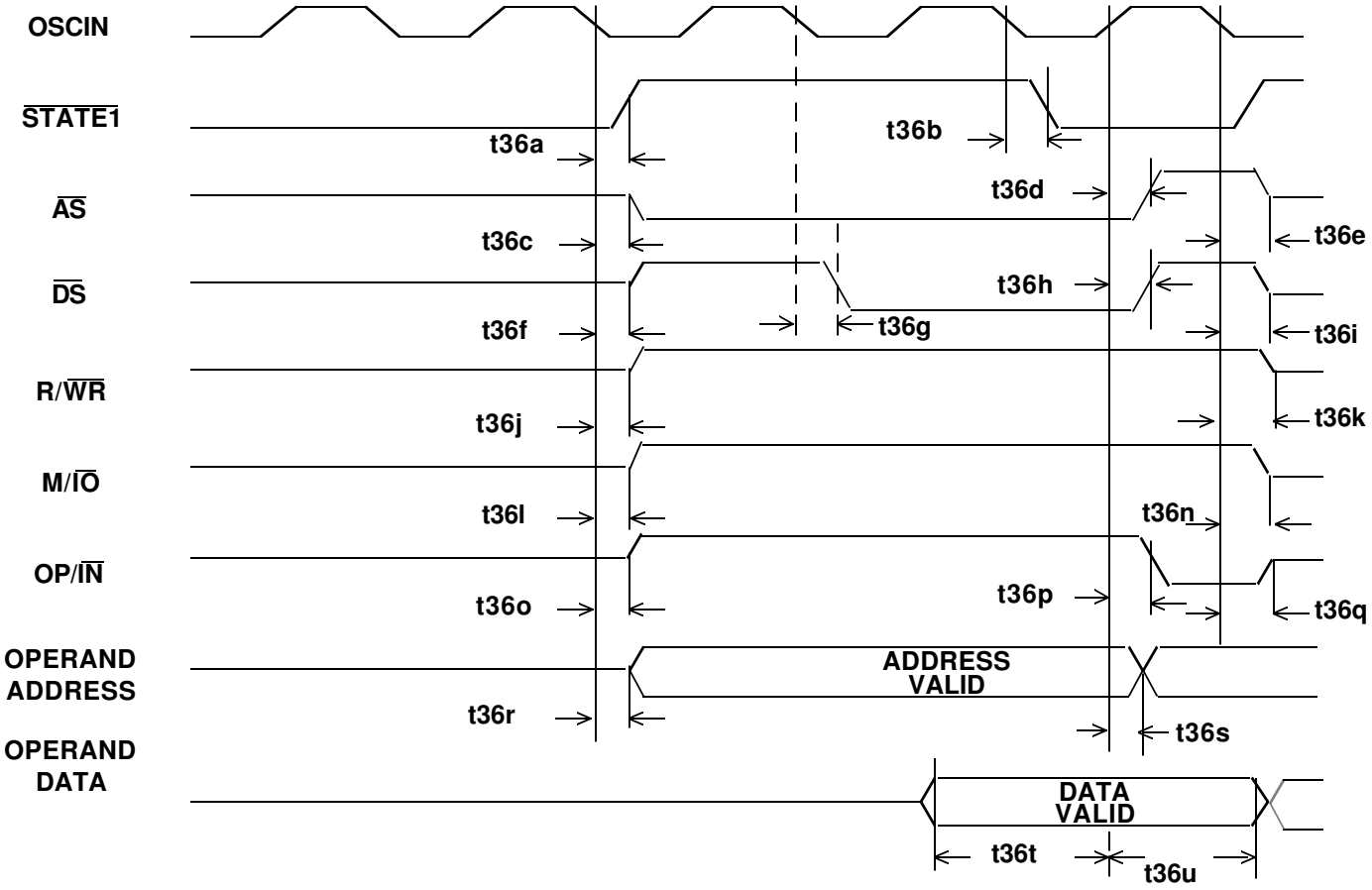
Figure 34. I/O Read Cycle



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
t35c *	OSCIN low to AS active	0	51	0	42	ns
t35d *	OSCIN high to AS inactive	0	50	0	38	ns
t35e	OSCIN low to AS high Z	--	50	--	38	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 active	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
t35o *	OSCIN low to OP/IN high	0	54	0	41	ns
t35p *	OSCIN high to OP/IN low	0	71	0	53	ns
t35q	OSCIN low to OP/IN high Z	--	53	--	40	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

Note:
*Guaranteed by test.

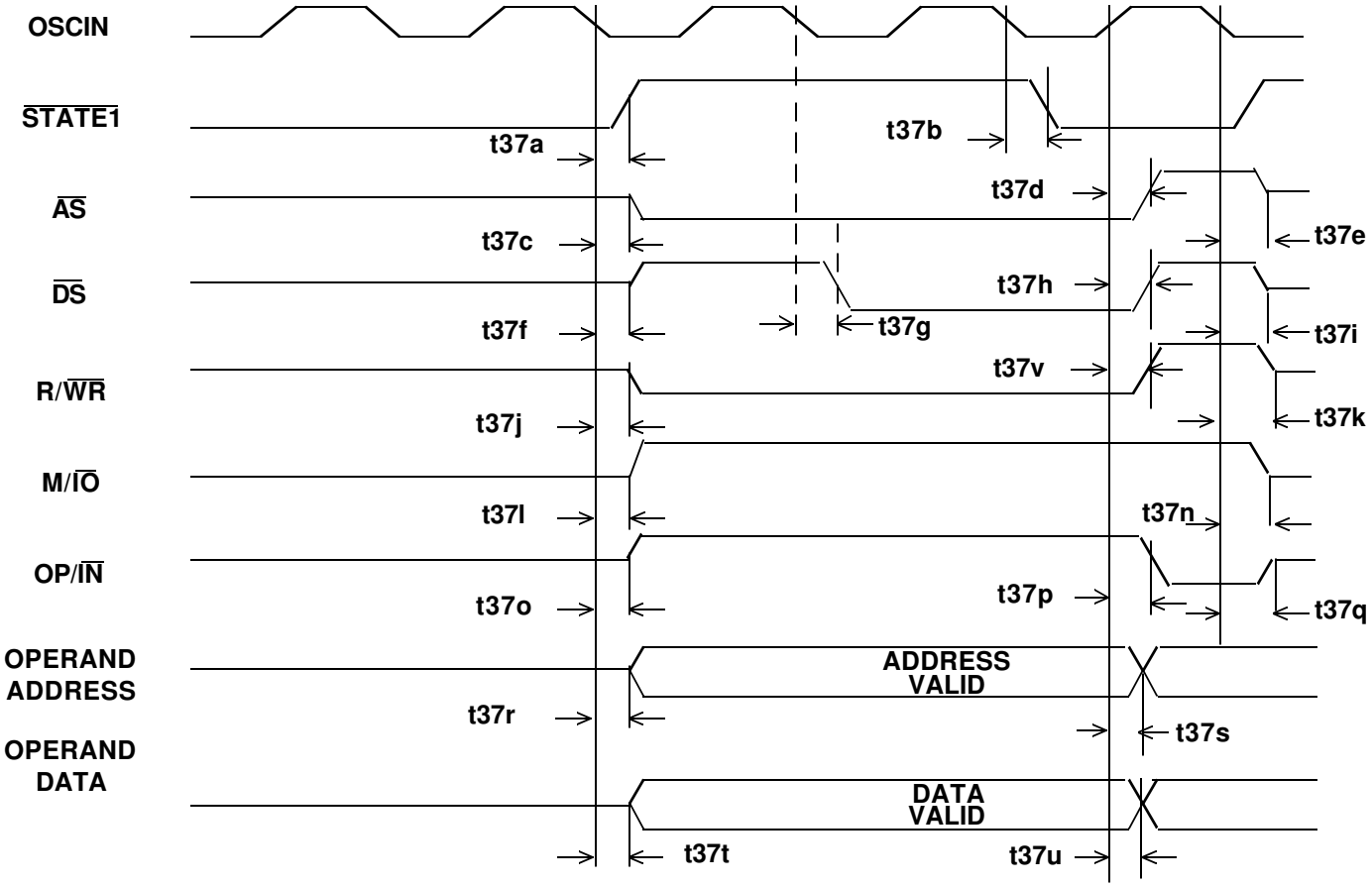
Figure 35. I/O Write Cycle



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
t36c *	OSCIN low to AS active	0	51	0	42	ns
t36d *	OSCIN high to AS inactive	0	50	0	38	ns
t36e	OSCIN low to AS high Z	--	50	--	38	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 active	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
t36o *	OSCIN low to OP/IN high	0	54	0	41	ns
t36p *	OSCIN high to OP/IN low	0	71	0	53	ns
t36q	OSCIN low to OP/IN high Z	--	53	--	40	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.

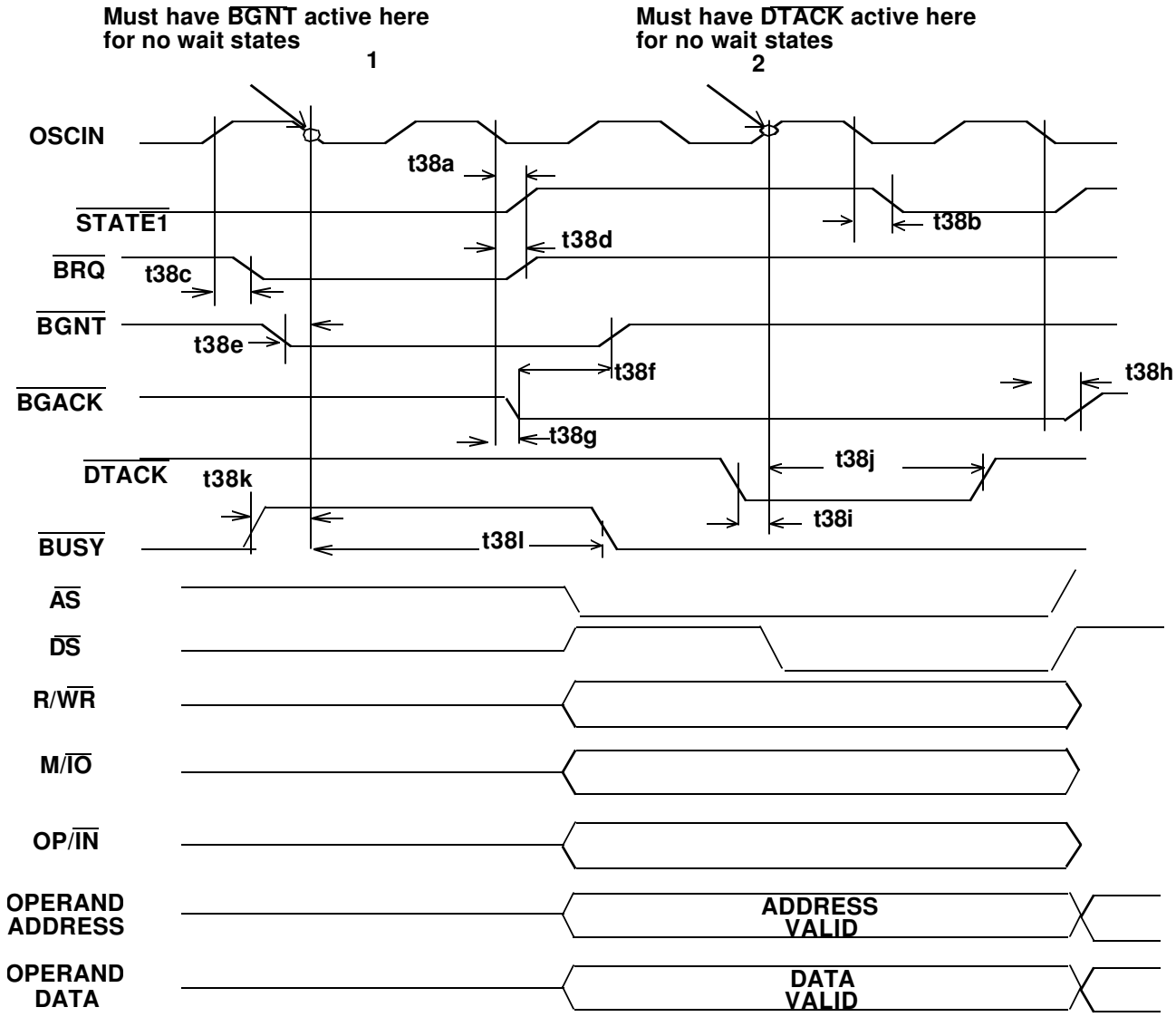
Figure 36. MEM Read Cycle



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
t37c *	OSCIN low to AS active	0	51	0	42	ns
t37d *	OSCIN high to AS inactive	0	50	0	38	ns
t37e	OSCIN low to AS high Z	--	50	--	38	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
t37o *	OSCIN low to OP/IN high	0	54	0	41	ns
t37p *	OSCIN high to OP/IN low	0	71	0	53	ns
t37q	OSCIN low to OP/IN high Z	--	53	--	40	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.

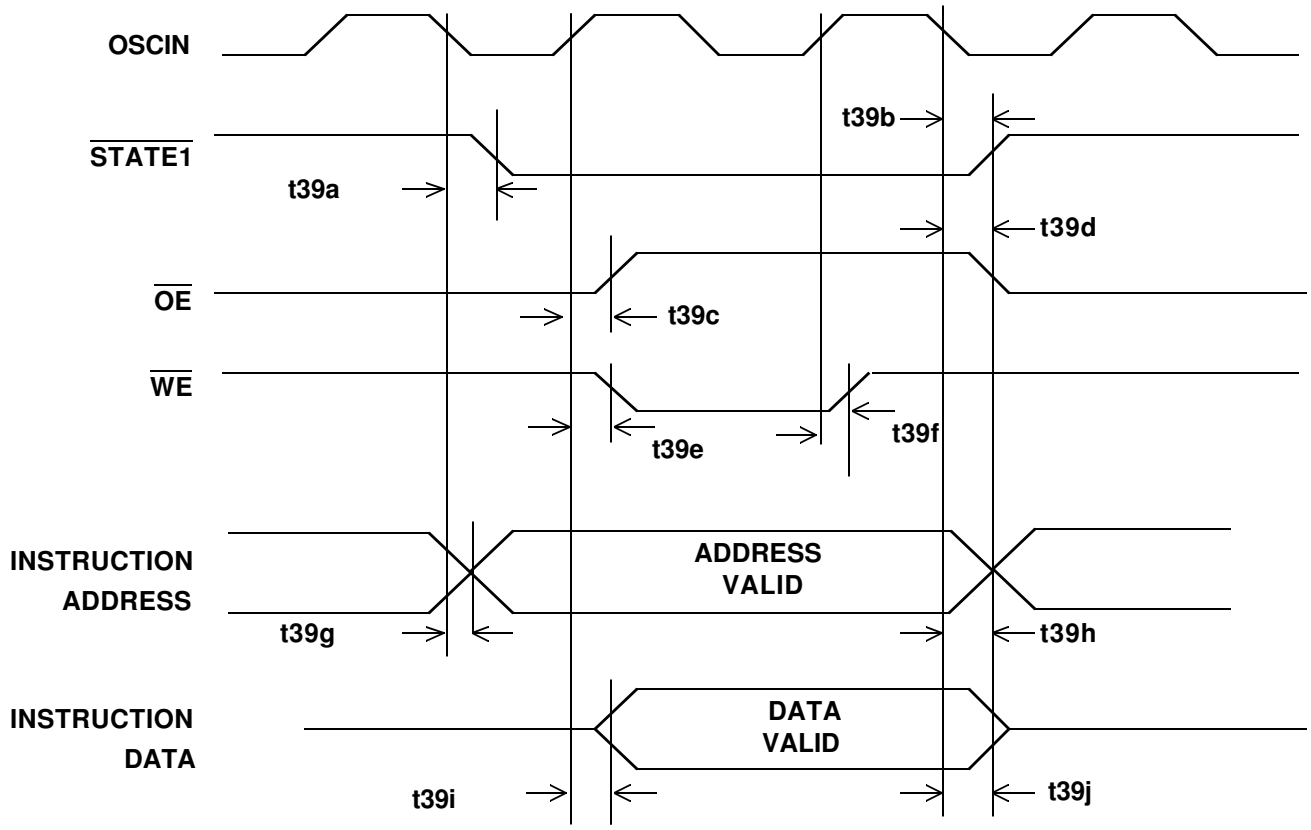
Figure 37. MEM Write Cycle



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	BGNT setup time	15	--	15	--	ns
t38f	BGNT 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.

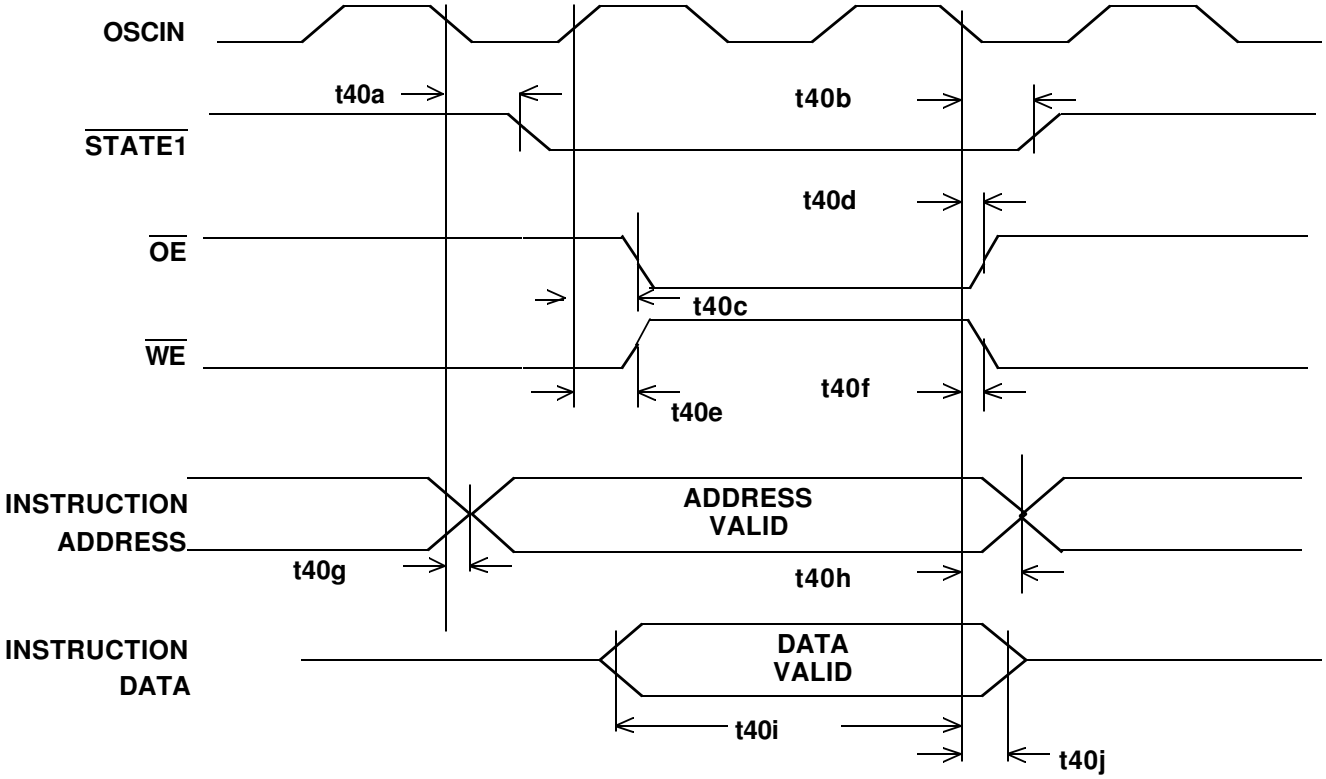
Figure 38. DMA No Wait State



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.

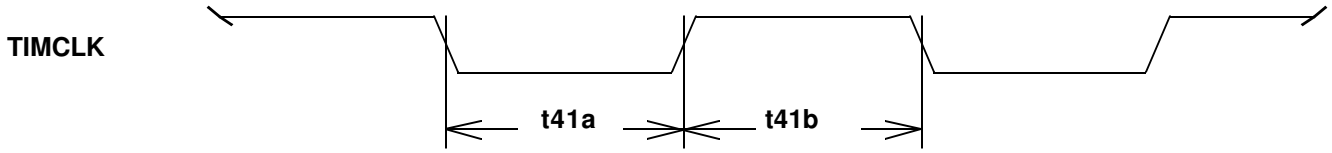
Figure 39. STRI Command, RISC Write Timing



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

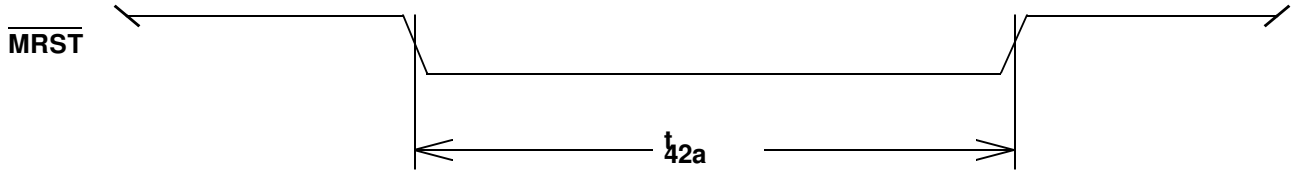
Note:
*Guaranteed by test.

Figure 40. LRI Command RISC Read Timing



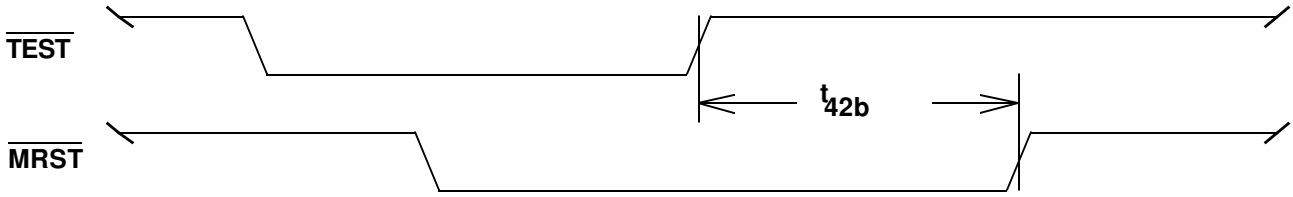
SYMBOL	PARAMETER	12 MHz		16 MHz		UNITS
		MIN	MAX	MIN	MAX	
t41a	TIMCLK low time	32	--	24	--	ns
t41b	TIMCLK high time	--	50	--	38	ns

Figure 41. 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

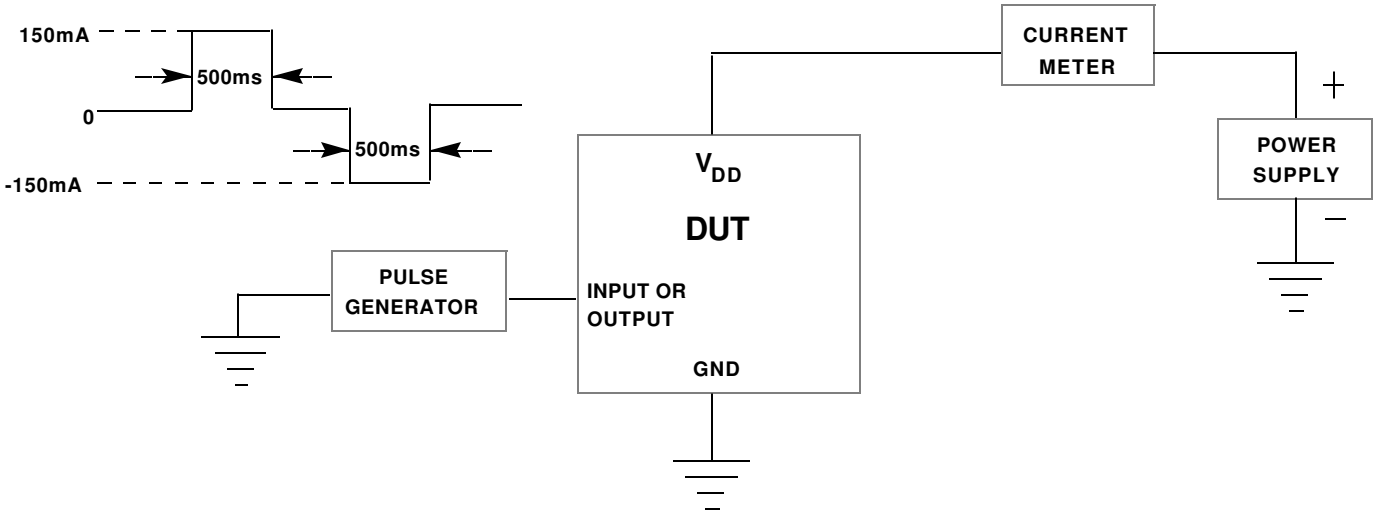
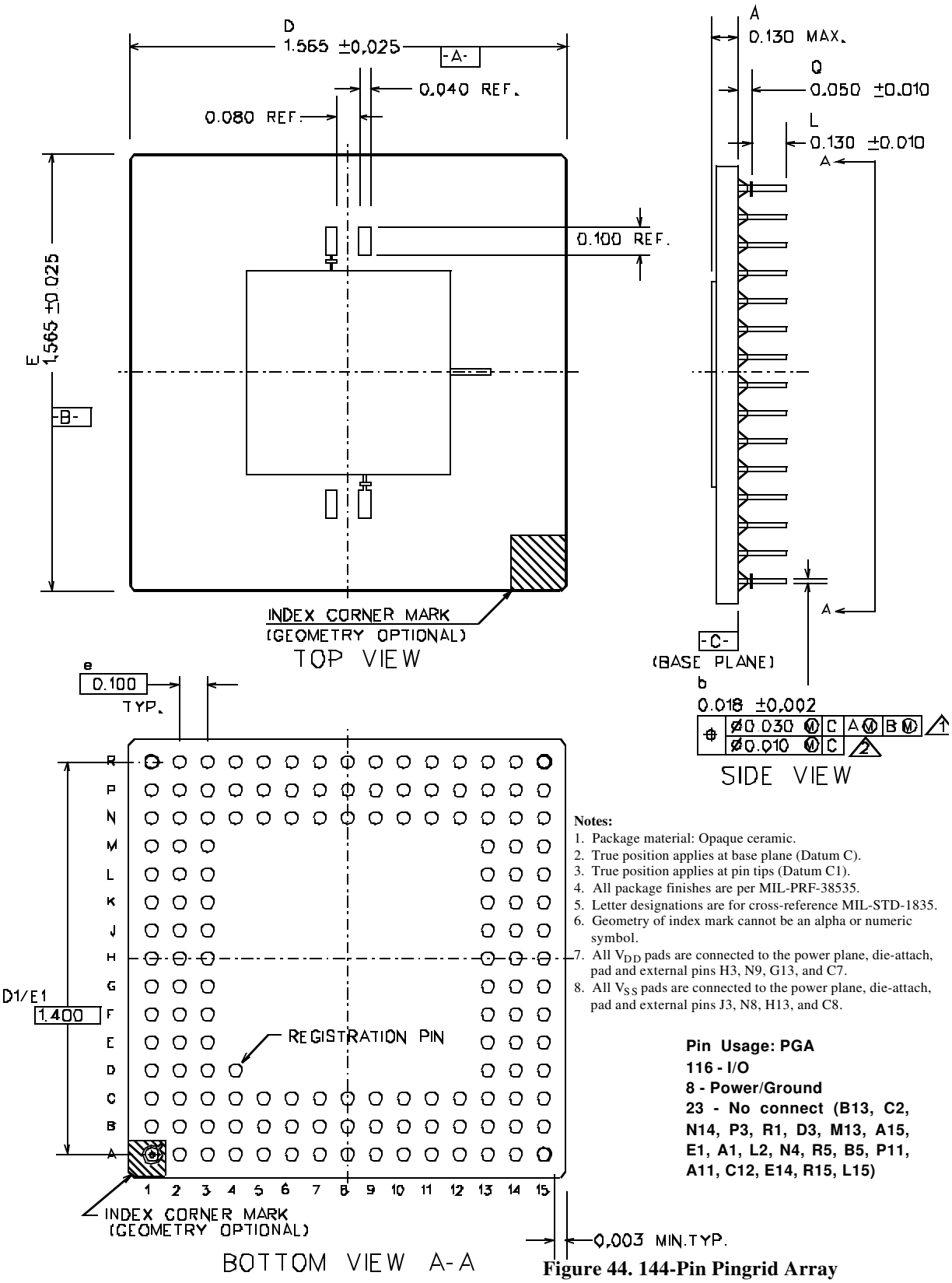


Figure 43. Latchup Test

LATCHUP TEST CONFIGURATION

Figure 43 shows the latchup test. V_{DD} holds at +5.5 V_{DC}, and V_{SS} 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 UT1750AR has latchup immunity in excess of +150 mA for 500 ms.

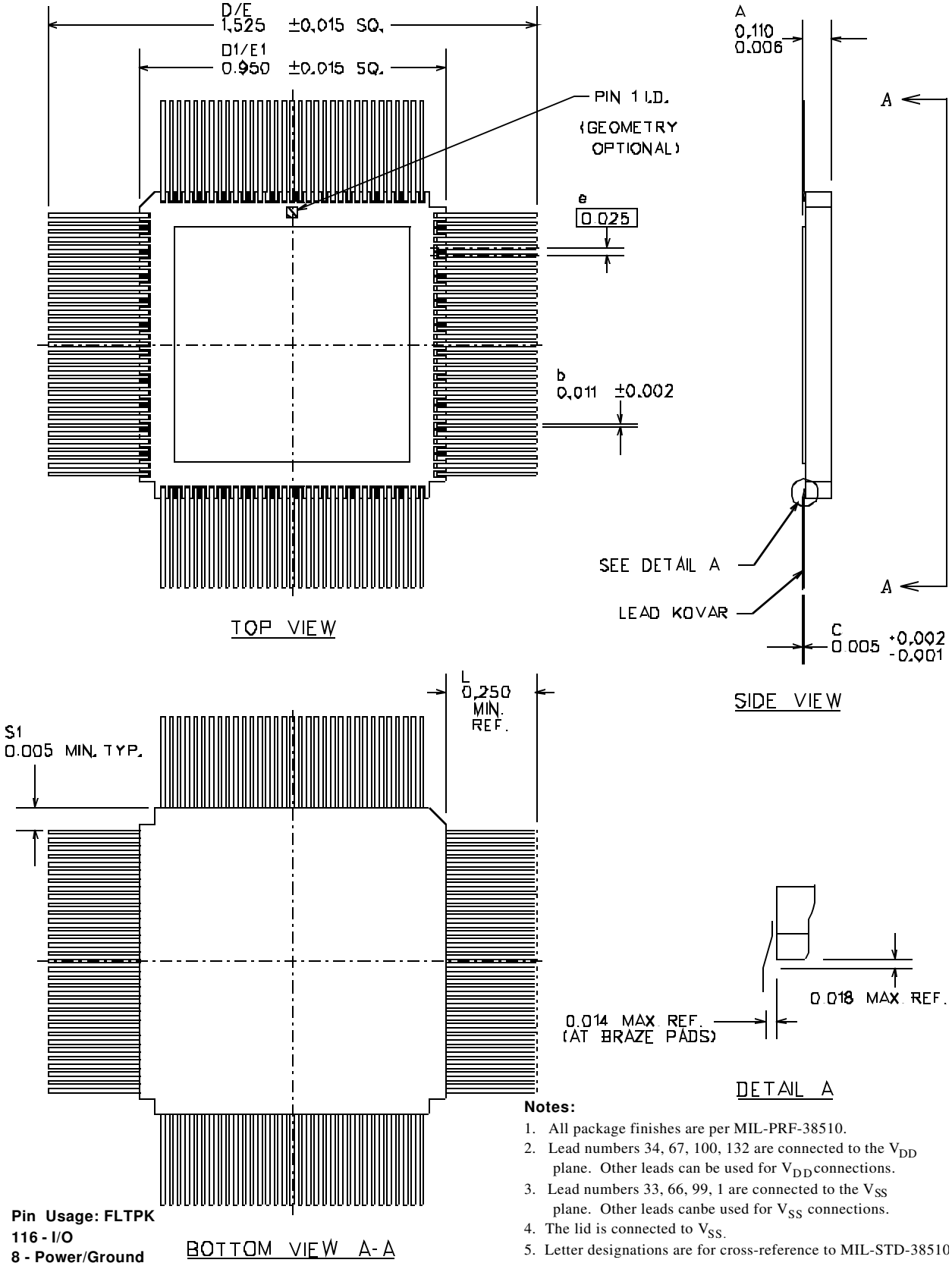


Notes:

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 finishes are per MIL-PRF-38535.
5. Letter designations are for cross-reference MIL-STD-1835.
6. Geometry of index mark cannot be an alpha or numeric symbol.
7. All V_{DD} pads are connected to the power plane, die-attach, pad and external pins H3, N9, G13, and C7.
8. All V_{SS} pads are connected to the power plane, die-attach, pad and external pins J3, N8, H13, and C8.

Pin Usage: PGA
116 - I/O
8 - Power/Ground
23 - No connect (B13, C2, N14, P3, R1, D3, M13, A15, E1, A1, L2, N4, R5, B5, P11, A11, C12, E14, R15, L15)

Figure 44. 144-Pin Pingrid Array



Pin Usage: FLTPK
116 - I/O
8 - Power/Ground
8 - No connect (2, 32, 35, 65, 68, 98, 101, 131)

Notes:

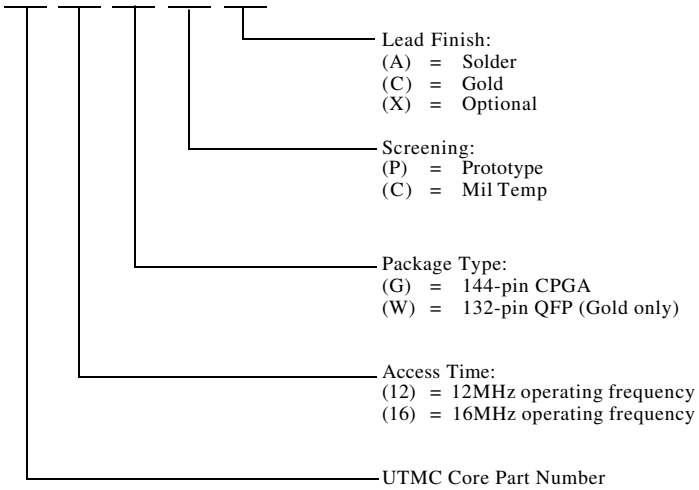
1. All package finishes are per MIL-PRF-38510.
2. Lead numbers 34, 67, 100, 132 are connected to the V_{DD} plane. Other leads can be used for V_{DD} connections.
3. Lead numbers 33, 66, 99, 1 are connected to the V_{SS} plane. Other leads canbe used for V_{SS} connections.
4. The lid is connected to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

Figure 45. 132-Lead Flatpack (Unformed Leads)

ORDERING INFORMATION

1750AR RISC Microprocessor

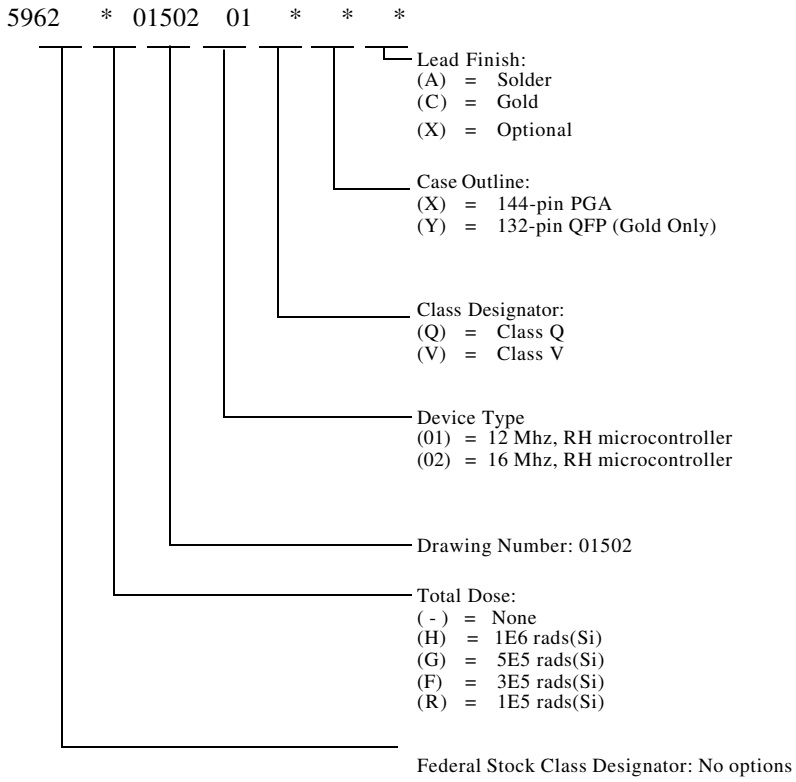
UT1750AR * * * * *



Notes:

1. Lead finish (A, C, or X) must be specified.
2. If an "X" is specified when ordering, part marking will match the lead finish and will be either "A" (solder) or "C" (gold).
3. Mil Temp range flow per UTMC's manufacturing flows document. Devices are tested at -55C, room temp, and 125C. Radiation neither tested nor guaranteed.
4. Prototype flow per UTMC's document manufacturing flows and are tested at 25C only. Lead finish is gold only. Radiation neither tested nor guarateed.

1750 RISC Microcontroller: SMD



Notes:

1. Lead finish (A, C, or X) must be specified.
2. If an “X” is specified when ordering, part number will match the lead finish and will be either “A” (solder) or “C” (gold).
3. 132 FP (package designator "Y") only available with gold lead finish.

Aeroflex Colorado Springs - Datasheet Definition

Advanced Datasheet - Product In Development

Preliminary Datasheet - Shipping Prototype

Datasheet - Shipping QML & Reduced Hi-Rel

COLORADO
Toll Free: 800-645-8862
Fax: 719-594-8468

INTERNATIONAL
Tel: 805-778-9229
Fax: 805-778-1980

NORTHEAST
Tel: 603-888-3975
Fax: 603-888-4585

SE AND MID-ATLANTIC
Tel: 321-951-4164
Fax: 321-951-4254

WEST COAST
Tel: 949-362-2260
Fax: 949-362-2266

CENTRAL
Tel: 719-594-8017
Fax: 719-594-8468

www.aeroflex.com/RadHard
info-ams@aeroflex.com

AEROFLEX
A passion for performance.

Our passion for performance is defined by three attributes represented by these three icons: solution-minded, performance-driven and customer-focused