ITD
T-49-17-32

10486

WHETSTONE®

90 MHz, 32 Bit Microprocessor

- 32 Bit, 30 MIPS MPU
- 33 Ns memory cycle time
- On-chip memory mapping; directly supports up to 1 MB of memory
- Dual data buses for fast RAM access
- ECL internal logic, TTL I/O
- Single +5 volt power supply
- 149 PGA with integral heatsink
- Mature development tools and O.S.
- MIL-STD-883C, Class B and Class S versions can be fabricated as required

High Performance

The 10486 microprocessor provides an unprecedented 90 MHz performance as the result of its ECL implementation. It is designed to directly drive up to one megabyte of high-speed static RAM, providing up to 60 megabytes per second of available memory bandwidth.

Applications

10486 is the solution for power-demanding applications including; CAD, CAM, CAE, ATE, high resolution graphics, optical/voice recognition, AI, imbedded controllers, robotics, realtime synthesis and micromainframes.

Long Term Solution

Unlike traditional microprocessors, the performance of the 10486 is limited by the speed of memory. As RAM speeds increase, MPU execution speed tracks with it up to 30 million instructions per second (MIPS). Memory cycle time can be as short as RAM access time plus 13 nanoseconds. Designs using the 10486 can be upgraded by the substitution of faster memories.

Development Support

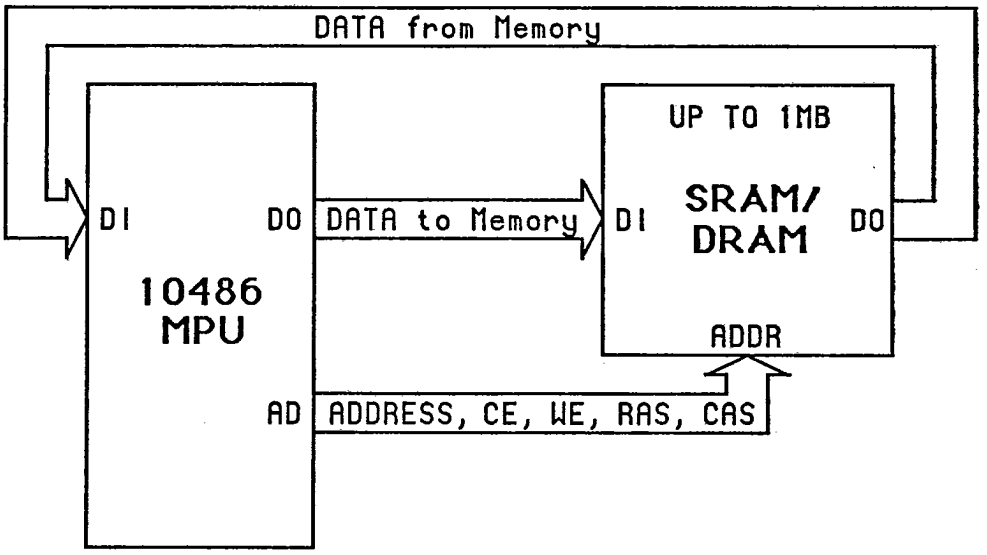
Mature software tools, development systems and a multi-user, multi-tasking O.S. which is optimized for the microprocessor's architecture and written in a compiled high-level development language to ensure efficient, maintainable structured programming. The 10486 supports BASIC, C, COBOL, FORTH, FORTRAN and PASCAL.

Convenient Packaging

The cavity-down pin grid array (PGA) with integral heat sink provides the shortest thermal path to cool the chip. Because of the low thermal resistance, only conventional air-cooling is necessary. An on-chip thermal diode is externally accessable to allow junction temperature measurement.

Preliminary Specification 5/89

8003-3239 Editor SA



Unique Three-Bus Architecture Directly Drives Memory Chips

Absolute Maximum Ratings

PARAMETER	MAXIMUM
Supply Voltage	7.0 V (Vcc-VEE)
TTL Input Voltage	5.5 V (Input-VEE)
Power Consumption	10 Watts
Operating Temperature	-55 °C (ambient) to +125 °C (case)
Operating Junction Temp.	+150 °C
Storage Temperature	-65 °C to +150 °C

Recommended Operating Conditions

PARAMETER	MIN	TYP	MAX	UNIT
Vcc (VEE=0)	4,5	5,0	5,5	V
TTL Output Current	-	-	20	mA
Operating Junction Temp.	-55	130	150	°C

Thermal Characteristics

PARAMETER	SYM	TYP	MAX	UNIT
Power Dissipation	Pd	6,8	10	Watts
Junction Temperature	Tj	130	150	°C
Propagation Derating	ΔTpd	-	0,37	% of Max. Tpd/°C of Tj≥130°C
Thermal Resistance	Øjc	3	5	°C/W (to case only)
" "	Øja	21	23	°C/W in still air.
" "	Øja	17	19	°C/W @ 100 LFPM
" "	Øja	14	16	°C/W @ 200 LFPM
" "	Øja	12	14	°C/W @ 300 LFPM
" "	Øja	10	12	°C/W @ 400 LFPM
" "	Øja	8	10	°C/W @ 600 LFPM
		TYP	W/C	
Ambient Temperature	Ta	14	-40	°C @ 100 LFPM
" "	Ta	35	-10	°C @ 200 LFPM
" "	Ta	48	+10	°C @ 300 LFPM
" "	Ta	62	30	°C @ 400 LFPM
" "	Ta	76	50	°C @ 600 LFPM

TTL Input/Output DC Characteristics

SYM	PARAMETER	DC TEST CONDITIONS	MIN	NOM	MAX	UNIT
V_{IH}	Input HIGH voltage	Guaranteed HIGH	2,0	-	-	V
V_{IL}	Input LOW voltage	Guaranteed LOW	-	-	0,8	V
V_{IK}	Input clamp voltage	V_{CC}=Min, I_{IN} = -18mA	-	-,8	-1,2	V
V_{OH}	Output HIGH voltage	V_{CC}=Min, I_{OH}= -1mA	2,7	3,4		V
V_{OL}	Output LOW voltage	V_{CC}=Min, I_{OL}=4mA	-	-	0,4	V
		V_{CC}=Min, I_{OL} =20mA	-	-	0,5	V
I_{OZ}	Output OFF current	V_{CC}=Max, 0.4≤V_{OUT}≤2.4	-50	-	50	µA
I_{IH}	Input HIGH current	V_{CC}=Max, V_{IN}=2.7V	-	-	50	µA
I_I	Input HIGH current	V_{CC}=Max, V_{IN} =5.5V	-	-	1,0	mA
I_{IL}	Input LOW current	V_{CC}=Max, V_{IN} =0.5V	-	-	-0,4	mA
I_{OS}	Output short current	V_{CC}=Max, V_{OUT}=0V	-25	-	-100	mA
I_{CC}	Supply current	V_{CC}=Max	-	1355	1900	mA

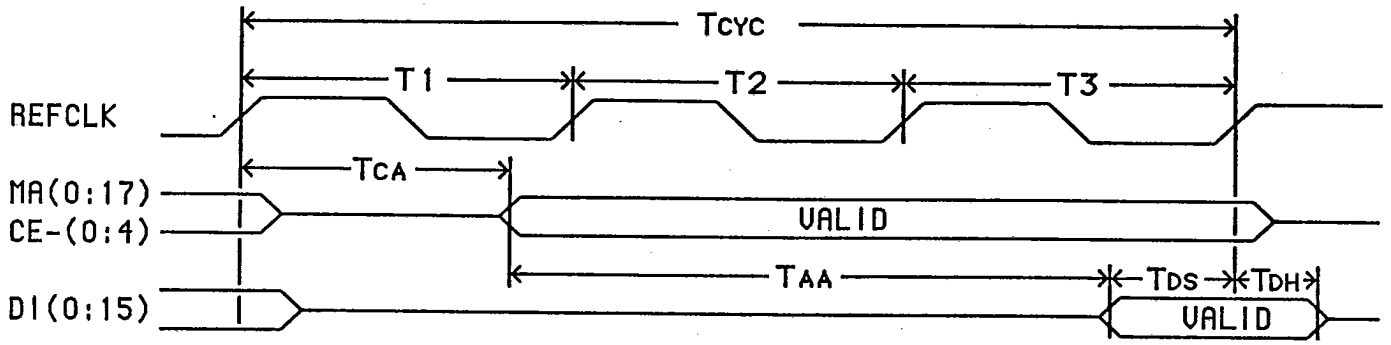
AC Characteristics: MEMORY CYCLE TIMING

(0°C≤Ta≤70°C; Tj≤130°C)

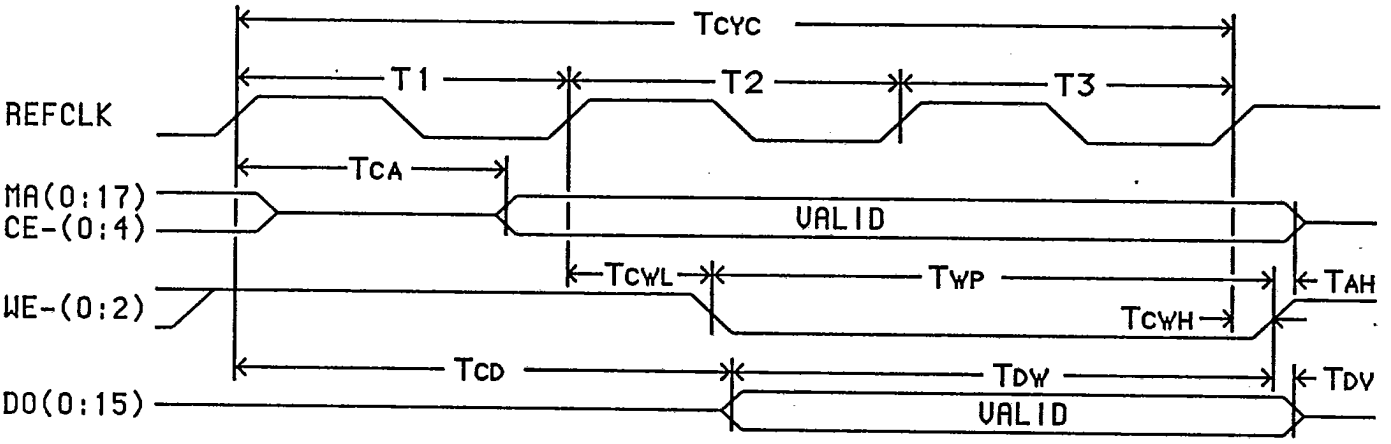
SYM	PARAMETER	MIN	TYP	MAX	UNIT
f_{MAX}	Maximum Clock frequency	90	-	-	MHz
T_{CLK}	Clock Cycle	11	-	-	ns
T_{CYC}	Memory Cycle (limited by T_{AA})	33	-	-	ns
T_{CA}	Clock high to valid Address	-	8	11	ns
T_{AA}*	Memory Access Time from Address $T_{AA}=T_{CYC}-T_{CAmax}-T_{DSmin}=T_{CYC}-10ns$	20	-	-	ns
T_{CWL}	Clock to Write Enable low	-	-	7	ns
T_{CWH}	Clock to Write Enable high	5	-	-	ns
T_{WP}*	Write Enable pulse width $T_{WP}=2T_{CLK}-T_{CWLmax}+T_{CWHmin}$	20	-	-	ns
T_{CD}	Clock to Data Out valid	-	15	20	ns
T_{DW}*	Data Out valid to end of Write Enable $T_{DW}=T_{CYC}-T_{CDmax}+T_{CWHmin}$	10	-	-	ns
T_{DV}	Data Out valid after end of Write Enable	0	-	-	ns
T_{DS}	Data Input setup to Clock	2	-	-	ns
T_{DH}	Data Input hold after Clock	0	-	-	ns
T_{AH}	Address hold after end of Write Enable	0	-	-	ns
T_{AW}*	Valid Address to Wait- Low $T_{AW}=2.5T_{CLK}-T_{CAmax}-T_{WSmin}$	-	-	10	ns
T_{WS}	Wait- Setup to Clock	2	-	-	ns
T_{WH}	Wait- Hold time after Clock	0	-	-	ns
T_{CH}	Clock Low to Hold	-	-	9	ns
T_{RC}	Ready Setup to Clock†	2	-	-	ns
T_{RH}	Ready Hold time after Clock†	0	-	-	ns
ΔT_{CL}	Derating of *all outputs* for capacitive loading	-	72	-	ps/pf

†Note: Ready can be asynchronous.
*Note: Assumes T_{CYC}=25 ns.
Note: Refclk is an *internal* clock that is for reference only.

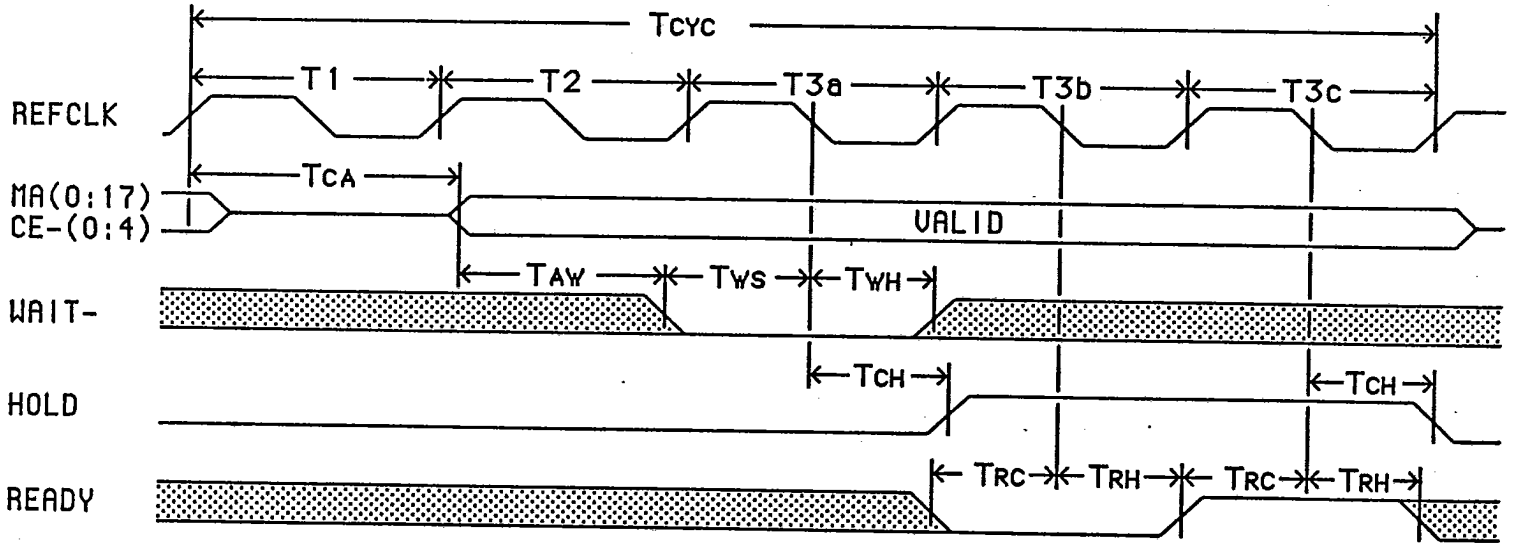
4



Memory Read Timing



Memory Write Timing



Delayed Memory Cycle Timing

AC Characteristics: PROGRAMMED I/O INSTRUCTIONS (0°C≤Ta≤70°C; Tj≤130°C)

SYM	PARAMETER	MIN	TYP	MAX	UNIT
T_{DRL}	I/O Acknowledge to I/O Request (Min=T_{CYC})	33	-	-	ns
T_{DRH}	I/O Acknowledge to I/O Request	33	-	80	ns
T_{MR}	Instruction valid prior to I/O Request	15	33	-	ns
T_{MH}	Instruction valid after I/O Acknowledge	10	33	-	ns
T_{DH}	I/O Acknowledge Hold time after I/O Request	0	-	-	ns
T_{RD}	I/O Request to Data Out	0	6	10	ns
T_{IE}	I/O Request to Data In Enabled	0	-	-	ns
T_{IS}	Data In setup to I/O Acknowledge Low	10	-	-	ns
T_{IH}	Data In hold after I/O Acknowledge Low	40	-	-	ns
T_{IZ}	I/O Request to Data In Disabled	0	-	10	ns
T_{SS}	I/O Skip setup to I/O Acknowledge High	10	-	-	ns
T_{SH}	I/O Skip hold after I/O Acknowledge High	40	-	-	ns

AC Characteristics: DIRECT MEMORY ACCESS (0°C≤Ta≤70°C; Tj≤130°C)

SYM	PARAMETER	MIN	TYP	MAX	UNIT
T_{RZ}	Bus Request to High Impedance Buses	33	-	80	ns
T_{ZA}	Buses released to Bus Acknowledge	10	-	33	ns
T_{RA}	Bus Request to Bus Acknowledge	75	-	105	ns
T_{AV}	Bus Acknowledge to Buses Active	33	-	80	ns

Signal Descriptions: PROCESSOR INTERFACE

Processor Control and State Signals

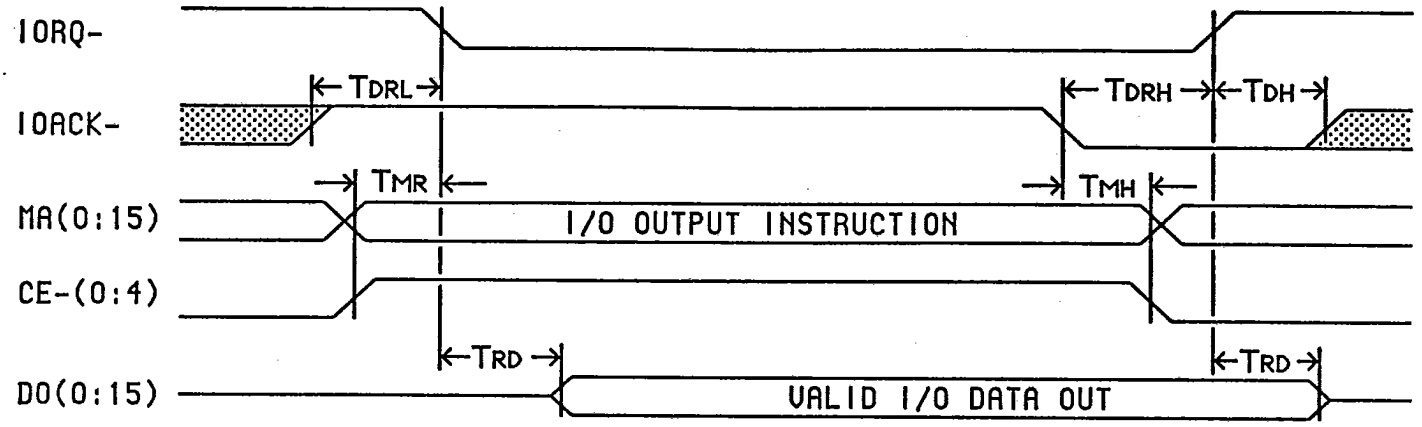
RST-	I	1B	Processor power-up reset
STOP-	I	2C	Switch input to force execution into shadow memory
HALT-	I	2D	Forces execution into shadow memory for Halt instruction
TRAP	I	1H	Extended instruction set trap enable
CLKI, CLKI-	I	1E,2E	ECL level balanced clock inputs
RUN-	O	10B	Processor run indicator (not executing in shadow)
VBB	O	1G	External ECL reference voltage
CARRY	O	2N	Carry flip-flop state

External Programmed I/O Handshake

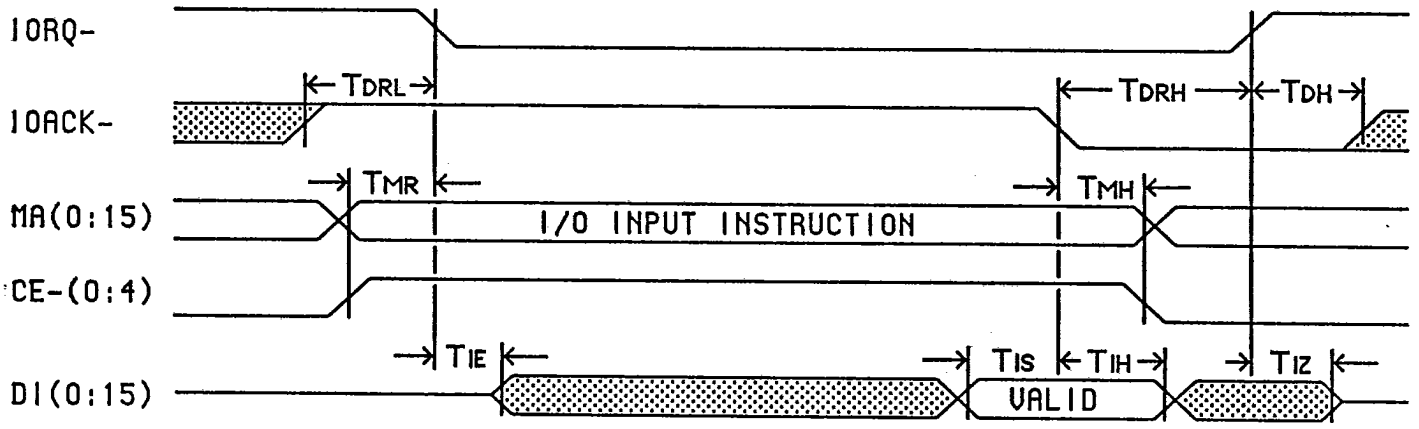
IORQ-	O	9B	External programmed I/O instruction
IOACK-	I	8A	Programmed I/O instruction completed
SKIP-	I	9A	Programmed I/O skip condition

Processor Interrupt

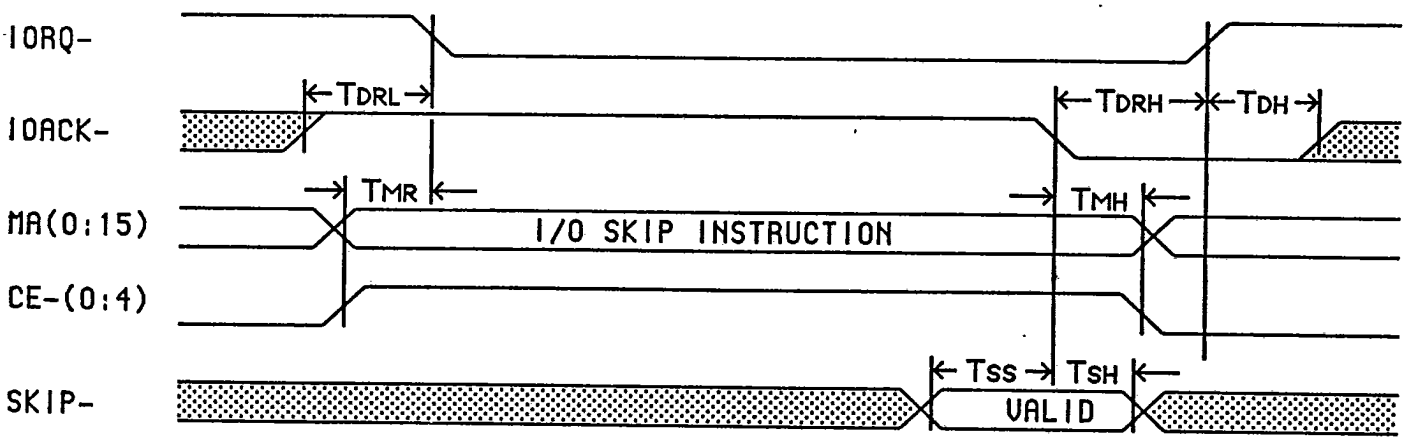
INT-	I	13C	Interrupt request (asynchronous)
DINT-	I	14E	Disable interrupt strobe
INTD-	O	10A	Interrupt disable strobe



I/O Output Instruction Timing



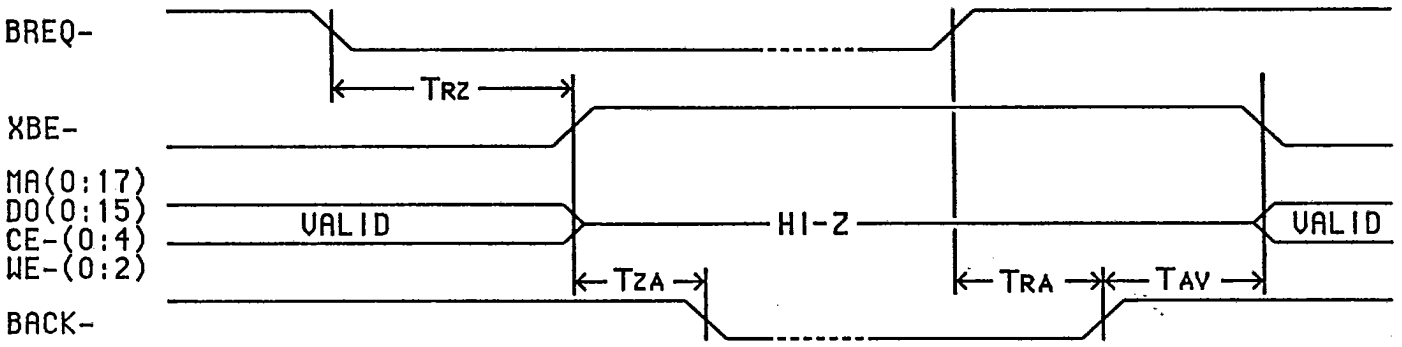
I/O Input Instruction Timing



I/O Skip Instruction Timing

Signal Descriptions: MEMORY INTERFACE

Memory Driving Signals			
DI(0:15)	I	--	Data input from memory
DO(0:15)	O	--	Data output to memory
MA(0:17)	O	--	Memory, I/O address output
CE0-, CE1-	O	14C,K	64K SRAM chip enable, page 0, 1
CE2-, CE3-	O	14L,M	256K SRAM chip enable, page 0, 1
CE4-	O	14N	Shadow RAM/ROM chip enable
WE0-, WE1-	O	14G,H	Write enable, RAM page 0, 1
WE2-	O	14J	Shadow RAM write enable
RAS-	O	1K	DRAM row address strobe
CAS-	O	3C	DRAM column address strobe
XBE-	O	2F	External buffer enable (memory drivers)
Memory Mapping			
MAP(0:7)	I	--	Mapped page selects (0:3, 4:7)
MAPE-	I	10R	Mapping enable strobe
MAPD-	I	10P	Mapping disable strobe
LA15	O	13L	Unmapped address bit 15
Asynchronous Memory Handshake			
WAIT-	I	8B	Delayed memory cycle request
HOLD	O	7A	Memory cycle suspended
READY	I	7B	Continue suspended memory cycle (asynchronous)
Direct Memory Access Handshake			
BREQ-	I	6A	DMA request (asynchronous)
BACK-	O	6B	DMA acknowlege (outputs disabled)
Miscellaneous Signals			
TP(0:3)	O	--	Internal test points - do not use
ANOD	X	1N	Anode of thermal diode
CATH	X	1P	Cathode of thermal diode
GND	X	--	System ground
+5V	X	--	System +5 volt power supply
N/C	X	--	Not internally connected pins - reserved for future use



DMA Handshake Timing

10486 PINOUTS

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
15		MA17	MA0	MA1	MA2	MA3	N/C	MA4	MA5	MA6	MA7	GND	N/C	GND		15
14	DO0	N/C	CE0-/ MA16	GND	DINT-	GND	WE0-	WE1-	WE2-	CE1-	CE2-	CE3-	CE4-	DI4	DO4	14
13	DO1	DI0	INT-	N/C	N/C	+5V	+5V	GND	GND	+5V	LA15	N/C	GND	DI5	DO5	13
12	DO2	DI1	N/C		GND						+5V		N/C	N/C	DO6	12
11	DO3	DI3	DI2	GND								GND	DI6	DI7	DO7	11
10	INTD-	RUN-	+5V										GND	MAPD-	MAPE-	10
9	SKIP-	IORQ-	GND										+5V	MAP0	MAP1	9
8	IOACK-	WAIT-	GND										GND	MAP2	MAP3	8
7	HOLD	READY	+5V										GND	MAP4	MAP5	7
6	BREQ-	BACK-	GND										+5V	MAP6	MAP7	6
5	DO8	DI8	N/C	GND								GND	N/C	DI12	DO12	5
4	DO9	DI9	N/C	N/C	+5V						GND		N/C	DI13	DO13	4
3	DO10	DI10	CAS-	TP0	TP1	+5V	GND	GND	+5V	+5V	TP2	TP3	GND	DI14	DO14	3
2	DO11	DI11	STOP-	HALT-	CLKI-	XBE-	MA10	MA8	MA12	MA13	MA14	MA15	CARRY	DI15	DO15	2
1		RST-	MA11	MA9	CLKI	GND	VBB	TRAP	GND	RAS-	GND	GND	ANOD	CATH		1
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	

PC BOARD COMPONENT-SIDE VIEW

Preliminary Specification 5/89

/ D

10486 Instruction Set

The 10486 microprocessor is a 32 bit machine; instructions are 16 or 32 bits long and data is addressed on 16-bit "word" boundaries. There is no native byte addressing, only word addressing. Arithmetic instructions provide a "byte swap" function for byte manipulation. The processor has four programmable registers (R0-R3), a carry flip-flop (CY) and a program counter (PC). These registers comprise all of the processor context other than external memory mapping.

The logical address space is 64K words (2^{16}), or 128K bytes. The physical address space can be expanded to one megabyte using the on-chip memory mapping logic, or to any size using external mapping logic. The MPU has a Von Neumann architecture permitting instructions and data to be intermixed freely within memory under software control.

The 10486 is designed to support an I/O bus that is separate from the memory bus. This was implemented because of the significant speed disparity between very-high-speed memory and slow I/O devices. The MPU does not actually execute I/O instructions by itself, but rather hands them off to an external automaton which can control an I/O bus at the prescribed speed. As a result of having a separate I/O automaton the I/O execution time can often parallel other MPU instruction execution. This allows the MPU to continue further program execution during relatively slow I/O bus transfers.

The 10486 instruction set is designed for maximum performance and simplicity. There are three basic instruction types; memory reference, arithmetic-logic and PIO (programmed input and output). The instructions are described below by type.

1.1 Memory Reference Instructions

The eight memory reference instructions provide the only program access to main memory. They function as follows:

JMP Directs program execution to the specified address, which changes PC.

JSR Same as JMP, but also saves the return address in R3. The return address is the location after the JSR instruction.

INC Adds one to the value stored at the specified absolute address. No registers are altered by this instruction.

ISZ Increment and Skip if Zero; adds one to the value stored at the specified address and skips the next instruction if the resulting value is zero. No registers are altered by this instruction.

DEC Subtracts one from the value stored at the specified absolute address. No registers are altered by this instruction.

DSZ Decrement and Skip if Zero; subtracts one from the value stored at the specified address and skips the following instruction if the resulting value is zero. No registers are altered by this instruction.

LDA Loads the specified register (R0-R3) with the value stored at the specified address in memory.

11

1.1 Memory Reference Instructions (cont.)

STA Stores the value of the specified register (R0-R3) into the specified address in memory.

PRIMARY ADDRESSING

In all memory reference instructions, the primary address is specified by the mode bits 6 and 7, plus the displacement field, as explained below:

ZP Zero-page addressing; the 8 bit address field specifies an address between zero and 255_{10} (377_8).

REL Relative addressing; the displacement field specifies a signed value from -128_{10} to $+127_{10}$ (-200_8 to $+177_8$) which is added to the value in PC to produce an effective address.

R2 Indexed addressing; adds the signed displacement (-128 to +127) to the value in register 2 to produce an effective address.

R3 Indexed addressing; adds the signed displacement (-128 to +127) to the value in register 3 to produce an effective address.

INDIRECT ADDRESSING

In memory reference instructions the indirect bit (5) indicates that the primary address points to an indirect address rather than directly to the data. The indirect address in turn points to the actual data to be accessed. All indirect addresses are 16 bit absolute.

AUTO-INCREMENT & DECREMENT CELLS

The processor provides automatic increment cells at Zero-page locations 20-27, and automatic decrement cells at locations 30-37. These cells are incremented or decremented by one when specified as an indirect address in a memory reference instruction. When accessed, the value in an "automatic" cell is read and modified, then used as an indirect address. These cells are only recognized when accessed using absolute (base page) indirect addressing. Relative and indexed indirect addressing of these cells does not modify them.

SELF-MODIFYING CODE

Supporting self-modifying code in a pipelined MPU provides important compatibility to the Von Neumann philosophy. The 10486 supports software that modifies itself using any addressing mode (e.g. relative addressing of PC+1). When the program modifies an instruction that is already loaded in the pipeline, the processor recognizes the instruction address and reloads the pipe at the same time as the memory location is changed. This can occur with STA, ISZ and DSZ instructions because they modify memory.

1.2 Arithmetic Instructions

Arithmetic instructions combine functions of one or two registers with shifting, carry control and test branching. The functions are:

ADC Adds the compliment of the value in the source register to the value in the destination register and stores the result into the destination register.

ADD Adds the value in the source register to the value in the destination register and stores the sum into the destination register.

AND Performs a logical AND (bit by bit) between the value in the source register and the value in the destination register and stores the result into the destination register.

INC Increments the value of the source register and stores it into the destination register.

MOV Moves the value from the source register into the destination register.

NEG Negates the value of the source register and stores it into the destination register.

NOT Logical inverse of the value of the source register and stores it into the destination register (logical invert).

SUB Subtracts the value in the source register from the value in the destination register and stores the difference into the destination register.

SHIFT

The shift field, bits 8 and 9, allows the result of any arithmetic function to be shifted before it is stored into the destination register and the carry flip/flop. The choices are:

- \- No shift.
- L Shift left one bit through carry.
- R Shift right one bit through carry.
- S Swap bytes within the word, no effect on carry.

CARRY

The carry field (bits 10 and 11) controls the *next state* of the carry flip-flop during an arithmetic function. In addition, if an arithmetic function causes a carry-out, it will complement the carry state specified by this field. The resulting value will then be stored into the carry flip-flop, unless a shift or no-load option is specified. The four choices are:

- \- Leave carry as it is.
- Z Clear carry to zero.
- O Set carry to one.
- C Complement carry's present value.

1.2 Arithmetic Instructions (cont.)

NO LOAD

The no-load bit (12) prevents the result of any arithmetic instruction from being stored into the destination register or the carry flip-flop. It is often used to perform non-destructive tests in conjunction with the skip function below.

SKIP

The skip field (bits 13-15) permits skipping the next instruction based on the resulting data or carry; after the arithmetic function, shifting and carry control have been performed. The eight options include:

--	Do not skip.
SKP	Always skip.
SZC	Skip if the carry result is zero.
SNC	Skip if the carry result is one.
SZR	Skip if the data result is zero.
SNR	Skip if the data result is not zero.
SEZ	Skip if either the data or carry result is zero.
SBN	Skip if both the data and carry results are not zero.

1.3 Programmed Input/Output

The eight PIO instructions provide the only program access to the I/O devices. The MPU provides 256 I/O device addresses from the eight least-significant bits of the instruction. In addition, the instructions imply the selection of a register within any I/O device. Bits 9 and 10 can be used as an extension of the I/O address if required, with the exception of the SKP instruction. Bit 8 generally determines whether it is an input or output instruction (except SKP which does not alter any register).

The MPU does not actually execute I/O instructions by itself, it expects an external automaton to control the interface and timing to any I/O devices. The MPU puts the actual instruction onto the MA bus and asserts IORQ to initiate the external I/O logic control of the I/O transfer. When completed external logic must assert IOACK.

The I/O instructions are as follows:

DOA	Transfers data from the specified register to the A register of the specified I/O device.
DIA	Transfers data from the A register on the specified I/O device to the specified register.
DOB	Transfers data from the specified register to the B register of the specified I/O device.
DIB	Transfers data from the B register on the specified I/O device to the specified register.

1.3 Programmed Input/Output (cont.)

DOC Transfers data from the specified register to the C register of the specified I/O device.

DIC Transfers data from the C register on the specified I/O device to the specified register.

DOD Transfers data from the specified register to the C register of the specified I/O device.

SKP Selectively skips the following instruction depending on the SKIP input to the MPU. No registers are modified.

Note: It is the responsibility of external logic to assert the SKIP input only during PIO instructions in which you want a skip function. It will cause an instruction skip when asserted during any PIO.

1.4 Programmed MPU Control

We recommend that a an I/O address be reserved for MPU control. The following list provides an overview of useful MPU control functions. They may be implemented in a variety of ways using I/O instructions that are convenient to your application.

Halt, branch into shadow memory.

Software interrupt.

Read/write the interrupt mask.

Enable/disable interrupts.

Read/write the map word.

Clear all I/O devices.

Skip if interrupts are enabled/disabled.

Skip if a power failure is detected.

2.0 Memory Mapping

The 10486 provides an extremely fast internal memory map to allow access to up to one megabyte of memory without external logic. This method can be externally extended to provide fast access to a larger memory area. Alternately any other memory mapping method can be implemented externally as required by the application.

2.1 Internal Memory Mapping

The 10486 provides onchip memory mapping that provides program access to any size main memory. Memory is divided into 64 KB physical pages. The MPU can access any two 64 KB pages of memory at a time. Because of the memory mapping provided on the 10486 there is *no speed penalty* to directly access up to 1 MB (one megabyte) of main memory. The 10486 provides the chip enable strobes (CE-), write enable strobes (WE-) and addresses (MA) to access up to 1 MB of 64K or 256K static RAMs. It also provides the RAS- and CAS- strobes to access dynamic RAMs.

2.2 External Memory Mapping

The LA15 output signal can be used as a selector to extend the internal memory mapping scheme to access over one megabyte of memory. There is a speed penalty when accessing over one megabyte of static RAM. There is generally no penalty when accessing slower dymanic RAM.

When a PIO instruction changes the external map register so that it maps-out the executing memory page (the page that the PIO instruction was fetched from), the very-next instruction is already in the pipe, and will be executed next. That instruction's memory reference and all subsequent instructions will be read from the newly mapped-in page. Example: the I/O instruction (which changes the map) followed by a JMP @60 will change the map, then jump to the address stored in location 60 of the newly selected page. If the MPU map is disabled (following an interrupt) the program can change the map word, but mapping will not take effect again until after it is reenabled.

3.1 Processor Interrupts

The INT- input to the 10486 is a non-maskable interrupt to the MPU, but may be masked externally as required. Traditionally this is done using PIO instructions.

When an interrupt occurs the MPU map is internally disabled, forcing the MPU to access physical pages zero and one. The address of the next instruction that would have been executed is stored at location zero in physical page zero in memory. The interrupt vector is read from location one in physical page zero. Program execution resumes at the location specified by the interrupt vector and further interrupts are ignored until interrupts are reenabled, typically by a specific PIO instruction.

The external map register can be read and saved or rewritten at any time whether mapping is enabled or disabled. If the MPU map selects other than page zero & one when mapping is reenabled , the next instruction will be executed; but its memory reference and any subsequent instructions will reference the newly mapped-in page(s).

10486 INSTRUCTION SET

DEFINITIONS:

R	CPU register 0, 1, 2, or 3
I	CPU register 2 or 3; used as an index
AA	an 8 bit memory address or I/O address
±AA	an 8 bit signed displacement (-128 to +127)
AAAA	a 16 bit absolute memory address
XX	don't care

ADC Add logical compliment

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	ADC R,R	84XX	2	1

ADD Binary Addition

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	ADD R,R	86XX	2	1

AND Logical AND

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	AND R,R	87XX	2	1

DEC Decrement memory

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
MEMORY ABSOLUTE	DEC AAAA	1D01	4	4

DI- Input from I/O device

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
I/O ADDRESS	DIA R,AA	61XX	2	*
I/O ADDRESS	DIB R,AA	63XX	2	*
I/O ADDRESS	DIC R,AA	65XX	2	*

*Note: Number of cycles for I/O depends upon external I/O timing.

DO- Output to I/O device

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
I/O ADDRESS	DOA R,AA	60XX	2	*
I/O ADDRESS	DOB R,AA	62XX	2	*
I/O ADDRESS	DOC R,AA	64XX	2	*
I/O ADDRESS	DOD R,AA	66XX	2	*

*Note: Number of cycles for I/O depends upon external I/O timing.

DSZ Decrement and Skip if Zero result

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
BASE PAGE	DSZ AA	18XX	2	3
PROGRAM RELATIVE	DSZ .±AA	19XX	2	3
INDEXED RELATIVE	DSZ ±AA,I	1AXX	2	3
BASE PAGE INDIRECT	DSZ @AA	1CXX	2	4
RELATIVE INDIRECT	DSZ @.±AA	1DXX	2	4
INDEXED INDIRECT	DSZ @±AA,I	1EXX	2	4

10486 INSTRUCTION SET

INC Add one (Increment)

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	INC R,R	83XX	2	1
MEMORY ABSOLUTE	INC AAAA	1501	4	4

ISZ Increment and Skip if Zero result

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
BASE PAGE	ISZ AA	10XX	2	3
PROGRAM RELATIVE	ISZ .±AA	11XX	2	3
INDEXED RELATIVE	ISZ ±AA,I	12XX	2	3
BASE PAGE INDIRECT	ISZ @AA	14XX	2	4
RELATIVE INDIRECT	ISZ @.±AA	15XX	2	4
INDEXED INDIRECT	ISZ @±AA,I	16XX	2	4

JMP Jump (program branch)

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
BASE PAGE	JMP AA	00XX	2	2
PROGRAM RELATIVE	JMP .±AA	01XX	2	2
INDEXED RELATIVE	JMP ±AA,I	02XX	2	2
BASE PAGE INDIRECT	JMP @AA	04XX	2	3
RELATIVE INDIRECT	JMP @.±AA	05XX	2	3
INDEXED INDIRECT	JMP @±AA,I	06XX	2	3
ABSOLUTE	JMP AAAA	0501	4	3

JSR Jump to Subroutine (save return address in R3)

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
BASE PAGE	JSR AA	08XX	2	2
PROGRAM RELATIVE	JSR .±AA	09XX	2	2
INDEXED RELATIVE	JSR ±AA,I	0AXX	2	2
BASE PAGE INDIRECT	JSR @AA	0CXX	2	3
RELATIVE INDIRECT	JSR @.±AA	0DXX	2	3
INDEXED INDIRECT	JSR @±AA,I	0EXX	2	3

LDA Load register from memory

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
BASE PAGE	LDA R,AA	20XX	2	2
PROGRAM RELATIVE	LDA R,.±AA	21XX	2	2
INDEXED RELATIVE	LDA R,±AA,I	22XX	2	2
BASE PAGE INDIRECT	LDA R,@AA	24XX	2	3
RELATIVE INDIRECT	LDA R,@.±AA	25XX	2	3
INDEXED INDIRECT	LDA R,@±AA,I	26XX	2	3
ABSOLUTE	LDA R,AAAA	2501	4	3

MOV Move word

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	MOV R,R	82XX	2	1

10486 INSTRUCTION SET

NEG Negate (twos compliment)

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	NEG R,R	81XX	2	1

NOT Logical invert

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	NOT R,R	80XX	2	1

SKP Skip on I/O condition

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
I/O ADDRESS	SKP AA	67XX	2	*

*Note: Number of cycles for I/O depends upon external I/O timing.

STA Store register into memory

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
BASE PAGE	STA R,AA	40XX	2	2
PROGRAM RELATIVE	STA R,.±AA	41XX	2	2
INDEXED RELATIVE	STA R,±AA,I	42XX	2	2
BASE PAGE INDIRECT	STA R,@AA	44XX	2	3
RELATIVE INDIRECT	STA R,@.±AA	45XX	2	3
INDEXED INDIRECT	STA R,@±AA,I	46XX	2	3
ABSOLUTE	STA R,AAAA	4501	4	3

SUB Subtract

ADDRESSING MODE	*MNEMONIC*	*OP CODE*	*BYTES*	*CYCLES*
REGISTER-TO-REGISTER	SUB R,R	85XX	2	1

REGISTER-TO-REGISTER INSTRUCTION MODIFIERS

SKIP	*FUNCTION*	*MNEMONIC*	*OP CODE*	*(BINARY OP CODE)*
	SKIP NEXT INSTRUCTION	SKP	XXX1	(XXXX.XXXX.XXXX.X001)
	SKIP IF CARRY IS ZERO	SZC	XXX2	(XXXX.XXXX.XXXX.X010)
	SKIP IF CARRY IS ONE	SNC	XXX3	(XXXX.XXXX.XXXX.X011)
	SKIP IF RESULT IS ZERO	SZR	XXX4	(XXXX.XXXX.XXXX.X100)
	SKIP IF RESULT IS NOT ZERO	SNR	XXX5	(XXXX.XXXX.XXXX.X101)
	SKIP IF RESULT OR CARRY=0	SEZ	XXX6	(XXXX.XXXX.XXXX.X110)
	SKIP IF RESULT & CARRY≠0	SBN	XXX7	(XXXX.XXXX.XXXX.X111)
TEST	TEST ONLY - without modifying the registers or the carry flip-flop.	#	XXX8	(XXXX.XXXX.XXXX.1XXX)

CARRY	*FUNCTION*	*MNEMONIC*	*OP CODE*	*(BINARY OP CODE)*
	PRE-ZERO CARRY	Z	XX1X	(XXXX.XXXX.XX01.XXXX)
	PRESET CARRY	O	XX2X	(XXXX.XXXX.XX10.XXXX)
	PRE-COMPLIMENT CARRY	C	XX3X	(XXXX.XXXX.XX11.XXXX)

ROTATE	*FUNCTION*	*MNEMONIC*	*OP CODE*	*(BINARY OP CODE)*
	ROTATE LEFT THRU CARRY	L	XX4X	(XXXX.XXXX.01XX.XXXX)
	ROTATE RIGHT THRU CARRY	R	XX8X	(XXXX.XXXX.10XX.XXXX)
	SWAP TWO BYTES IN WORD	S	XXCX	(XXXX.XXXX.11XX.XXXX)