TENET, Inc.
International Science Center
Sunnyvale, California

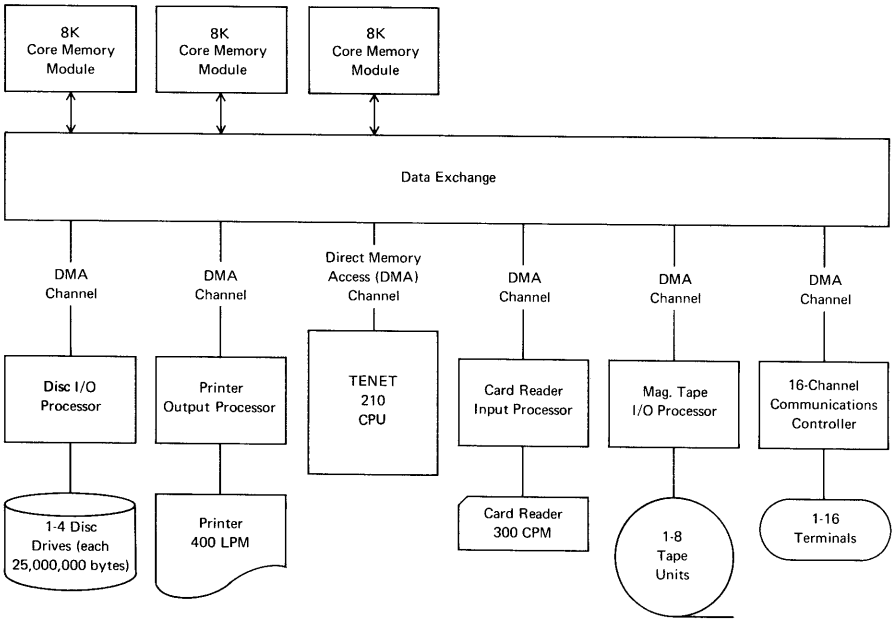
TENET 210

TENET 210

The TENET 200 series is a computer family designed for interactive time-share operations. It features multiple central processors, direct memory accessing, LSI (large-scale integrated circuit) memory mapping, 800-nanosecond memory, and memory interleaving.

Each memory module is 8K or 16K words of 33 bits (K = 1024). The 200 series architecture provides 20 direct channels between the memory modules and processors (central and I/O). Hardware priority provides 200-nanosecond time slices to serve each processor.

The TENET 210 Computer System is the first of the family to be made available. It is a single-processor system, and typically would include (in addition to the CPU) 32K words of core memory, two single-spindle disc drives (50 million bytes), one magnetic tape transport, and ports for 32 simultaneous users.



Memory may be expanded modularly to provide up to 128K words of storage. The CPU has 8 general registers and 8 control registers. The Data Exchange provides 20 bi-directional DMA channels with access priority, and 20 levels of nested interrupts, each level expandable to 16 sublevels.

Data Exchange

The TENET 210 architecture is simple and symmetrical; it is centered around a Data Exchange which combines advanced system concepts with unique electronics techniques to produce a system with unimpeded data flow. The addition of system modules such as memory, CPU's, and I/O Processors increases system throughput without channel congestion. Features of the Data Exchange include:

- **20 processor ports**
- **8 memory ports**
- **4 simultaneous memory access channels**
- **20 levels of channel priority**
- **20 levels of program interrupt priority**

Up to four processors can access memory within one memory cycle, provided each requires access to a different memory module, and channel priorities can be arbitrarily assigned to each processor.

The Data Exchange also provides direct communication between processors for exchange of commands, data, program interrupt requests, and status.

Resolving program interrupt priorities is another function of the Data Exchange. Twenty levels of program interrupt are assignable independent of channel priority. Program interrupts are nested by the CPU when an incomplete interrupt request is interrupted by a higher priority. Interrupt requests can be temporarily suppressed; I/O processors can also be disarmed so that they will not issue interrupt requests.

Core Memory

Core Memory Modules are 8K or 16K each, 128K maximum (K = 1024). Word length is 33 bits (32 data plus one parity). Access time is 325 nsec with a full cycle time of 800 nsec. Memory modules are interleaved to decrease the effective cycle time to 400 nsec. Memory band width is 5 million words per second.

Central Processor

The CPU (the 3210) is a general purpose processor designed for time-sharing; it includes provisions for scientific data processing and real-time applications. Features are:

- Word-oriented operation
- Expandable core to 128K words
- Direct addressing from primary instruction
- Memory mapping, with memory protect
- Indirect addressing with pre- or post-indexing
- 8 high-speed general registers (7 for indexing)
- 8 addressable high-speed control registers
- Nested real-time interrupts
- Automatic interrupt identification
- Master/Slave modes
- Automatic trap on error condition
- Programmable instructions
- Extensive instruction set:
  - Bit manipulation and testing
  - Word, double-word, field
  - Program status exchange and stacking
  - Move
  - Register to Memory
  - Immediate to Register
  - Immediate to Memory
- Fast instruction execution times:
  - Multiply 3.4μ Typical
  - Divide 7.8μ Maximum
  - Branch 0.8μ Maximum
  - Register to register word 1.0μ
  - Memory to register word 1.2μ (with interleaving)

Floating Point

The 210 employs software floating point. Single-precision floating point provides 7+ significant figures in the range 10^{-19} to 10^{19}. Double precision provides 16+ significant figures in the range 10^{-76} to 10^{76}. Double precision is standard, with single precision available when specified by the user.

When using double precision, a simulated multiply or divide requires approximately 135 microseconds and an add or subtract requires approximately 95 microseconds. Single precision adds about 20 microseconds to these operations.

Hardware floating point will be available as an option in 1971.

Memory Mapping

A Memory Map is available in segments of 32 pages (a page is 512 words), up to 256 pages (128K words).

In memory mapping, the 128K virtual addressing space is treated as 256 pages of 512 words each. Every virtual page is represented in the map by the following:

an eight-bit register containing the actual memory page address. A zero value indicates an unassigned virtual page; instructions referencing unassigned pages are trapped.

a page status register consisting of two bits. One bit is a write protect flag; if set, the page is considered read-only. (In this mode, an instruction which tries to modify the contents of the page is trapped. Master mode programs may override page protection.) The other bit is the page-altered flag; it is set whenever the page is written on by the CPU.

In two of the four machine modes—Slave/Mapped and Master/Mapped—all memory references are mapped automatically. In the Master/Unmapped mode, all memory references are unmapped. In the Master/Selective mode, all instruction fetches are unmapped, while data references are conditionally mapped.

Programmed Instructions (PINS)

Certain CPU operation codes are executed as Programmed Instructions. These operation codes cause an automatic transfer of control to a program that interprets them, thus providing a very efficient single-argument subroutine capability. *Global* PINS are used for user/system interface; there are 16 of these. *Local* PINS are used for system/system and user/user interface; user and system have 48 each.

Trap Capability

The system's trap capability is designed to permit re-execution of trapped instructions; the user avoids space- and time-consuming instruction analysis and register reconstruction. Conditions resulting in traps include:

- **Unimplemented op code**
- **Privileged in slave mode**
- **EXEC of an EXEC instruction**
- **Unimplemented memory**
- **Write-protect violation**
- **Unassigned virtual page**
- **Unassigned op code**
- **Instruction fetch from register**
- **Floating overflow, underflow**
- **Floating divide by zero**
- **Stack overflow**
- **Parity error**
- **Power failure**
- **Unassigned IOC command field**

TEST & TEST REGISTER BIT	TB	Test Bit
	STB	Store Bit
	TBS	Test Bit and Set
	TBR	Test Bit and Reset
	TRB	Test Register-designated Bit
	STRB	Store Register-designated Bit
	TRBS	Test Register-designated Bit and Set
	TRBR	Test Register-designated Bit and Reset
	FRB	Find and Reset Bit Single Word
	FRBD	Find and Reset Bit Double Word
FIELD	DFP	Decrement Field Pointer
	IFP	Increment Field Pointer
	AFA	Add Field Arithmetic
	AF	Add Field Logical
	SFA	Subtract Field Arithmetic
	SF	Subtract Field Logical
	ISFA	Inverse Subtract Field Arithmetic
	ISF	Inverse Subtract Field Logical
	LF	Load Field Logical
	LFI	Load Field Logical after Incrementing
	LFA	Load Field Arithmetic
	LFAI	Load Field Arithmetic after Incrementing
	STF	Store Field
	STFI	Store Field after Incrementing
	STIF	Store Immediate to Field
	CF	Compare Field Logical
	CFI	Compare Field Logical after Incrementing
	CFA	Compare Field Arithmetic
	CFAI	Compare Field Arithmetic after Incrementing
	CIF	Compare Immediate to Field
WORD	LW	Load Word
	STW	Store Word
	AW	Add Word
	SW	Subtract Word
	ISW	Inverse Subtract Word
	MW	Multiply Word
	DW	Divide Word
	ANDW	AND Word
	ORW	OR Word
	EORW	Exclusive OR Word
	CW	Compare Word
	XW	Exchange Word
	LWD	Load Word Doubled
IMMEDIATE	LI	Load Immediate
	AI	Add Immediate
	MI	Multiply Immediate
	DI	Divide Immediate
	ANDI	AND Immediate
	ORI	OR Immediate
	EORI	Exclusive OR Immediate
	CI	Compare Immediate
	ISI	Inverse Subtract Immediate
MEMORY	AM	Add to Memory
	SM	Subtract from Memory
	ANDM	AND to Memory
	ORM	OR to Memory
	ILW	Increment and Load Word
	EORM	Exclusive OR to Memory
	CAIM	Complement and Add Immediate to Memory
	AIM	Add Immediate to Memory
	SIM	Subtract Immediate from Memory
	STIM	Store Immediate to Memory
	CIM	Compare Immediate with Memory

DOUBLE WORD	LD	Load Double Word
	STD	Store Double Word
	AD	Add Double Word
	SD	Subtract Double Word
	CD	Compare Double Word
	DD	Divide Double Word
	MD	Multiply Double Word
SHIFT	SAL	Shift Arithmetic Left Single
	SAR	Shift Arithmetic Right Single
	SALD	Shift Arithmetic Left Double
	SARD	Shift Arithmetic Right Double
	SLL	Shift Logical Left Single
	SLR	Shift Logical Right Single
	SLLD	Shift Logical Left Double
	SLRD	Shift Logical Right Double
	SCL	Shift Circular Left Single
	SCR	Shift Circular Right Single
	SCLD	Shift Circular Left Double
	SCRD	Shift Circular Right Double
	SUN	Shift Until Normalized Single
	SUND	Shift Until Normalized Double
FLOATING POINT	FSS	Floating Subtract Short
	FSL	Floating Subtract Long
	FISS	Floating Inverse Subtract Short
	FAL	Floating Add Long
	FMS	Floating Multiply Short
	FML	Floating Multiply Long
	FDS	Floating Divide Short
	FDL	Floating Divide Long
	FIDS	Floating Inverse Divide Short
	FIDL	Floating Inverse Divide Long
BRANCH	B	Branch
	BOV	Branch if Overflow
	BNO	Branch if No Overflow
	BCRY	Branch if Carry
	BNC	Branch if No Carry
	NOP	No operation
	BNCO	Branch if No Carry and No Overflow
	BCO	Branch if No Carry and No Overflow
	BL	Branch if Less
	BG	Branch if Greater
	BNE	Branch if Not Equal
	BMO	Branch if Any Matching Ones
	BIOCB	Branch if I/O Controller Busy
	BIOB	Branch if I/O Busy
	BIOA	Branch if I/O Accepted
	BBS	Branch if Bit Set
	BGE	Branch if Greater or Equal
	BE	Branch if Equal
	BNMO	Branch if No Matching Ones
	BBR	Branch if Bit Reset
	BIODB	Branch if I/O Device Busy
	BIOI	Branch if I/O Inoperative
	BIOR	Branch if I/O Rejected
	BRL	Branch if Register Less Than Zero
	BRG	Branch if Register Greater Than Zero
	BRNE	Branch if Register Not Equal Zero
	BROD	Branch if Register Odd
	BRGE	Branch if Register Greater or Equal Zero
	BRLE	Branch if Register Less Than or Equal Zero
	BRE	Branch if Register Equal Zero
	BREV	Branch if Register Even

BRANCH (Cont.)

BIR	Branch on Incrementing Register
BDR	Branch on Decrementing Register
BCS	Branch if Condition Set
BCR	Branch if Condition Reset
BRCS	Branch if Register Condition Set
BRCR	Branch if Register Condition Reset
BAC	Branch on Arithmetic Conditions
BAL	Branch and Link

INPUT/OUTPUT

SIO	Start IO—record oriented devices
SI	Start Input character devices
SO	Start Output character devices
TIO	Test IO—record oriented devices
TI	Test Input character devices
TO	Test Output character devices
CION	Controller Interrupts On
CIOF	Controller Interrupts Off
IXIT	Interrupt routine exit
ISIO	Interrupt on SIO acceptable
RIO	Reset controller
HIO	Halt IO—reset device
PON	Power On
POFF	Power Off
AIO	Acknowledge IO interrupt
TDV	Test Device

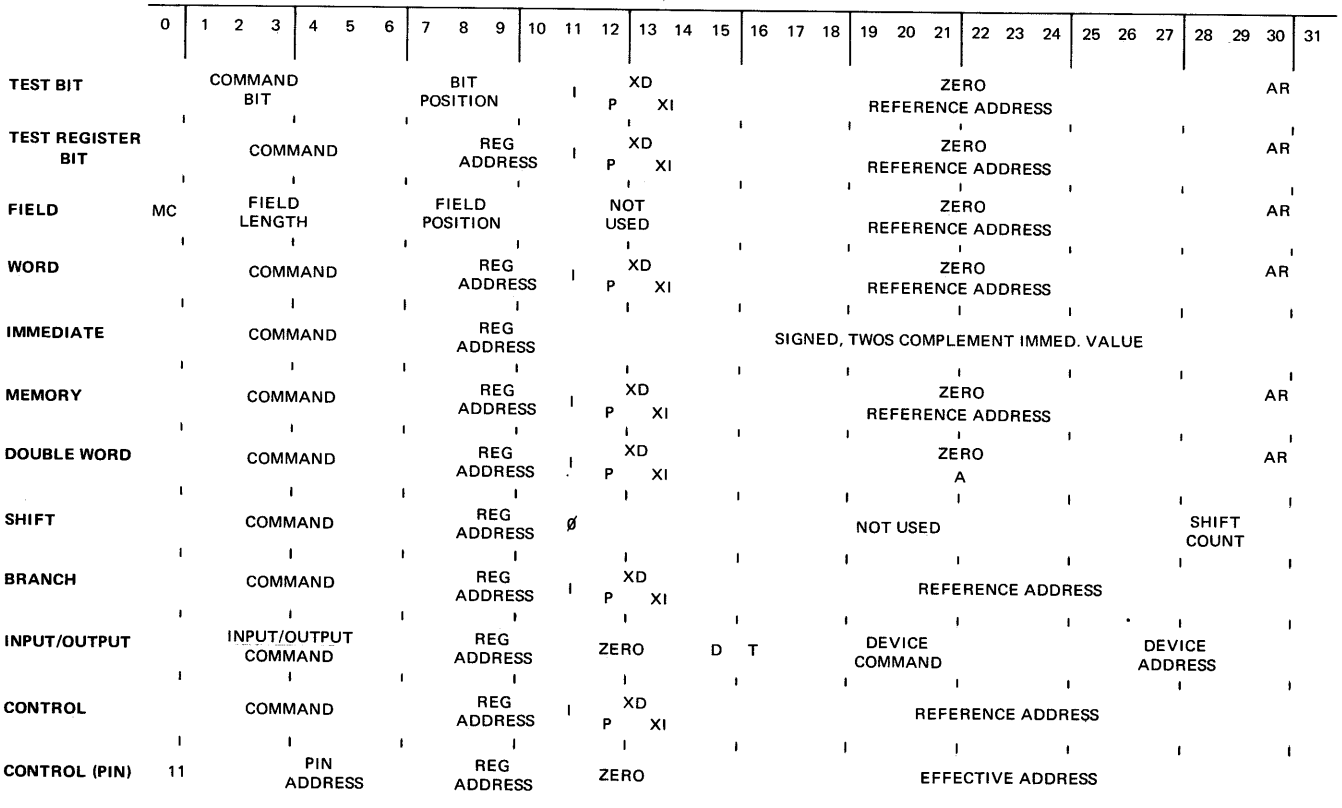
CONTROLS Privileged

LPS	Load Program Status
XPS	Exchange Program Status
LCR	Load Control Register
PPS	Push Program Status
STCR	Store Control Register
POPS	Pop Program Status
PAUS	Pause
ION	System Interrupts On
IOFF	System Interrupts Off
LM	Load Map
LMS	Load Map Status
LAA	Load Actual Address

CONTROLS Unprivileged

STMS	Store Map Status
STSR	Store Switch Register
STPS	Store Program Status
LCI	Load Condition Indicators
EXEC	Execute
LG	Load Register Group
STG	Store Register Group
MOVI	Move with Incrementing Pointer
MOVD	Move with Decrementing Pointer
LAS	Load and Set
LVA	Load Virtual Address
PIN	Programmed Instruction

Formats



AR	Register Reference Address	XI	Index Register Address—Indirect Addressing	D	Register Transfer Direction
I	Indirect Address Flag	P	Pre-Indexing Flag	T	Register Transfer Flag
XD	Index Register Address—Direct Addressing	MS	Master/Slave	MC	Mapping Control

Input / Output

Terminals

The interrupt structure of the Data Exchange permits any level of interaction between the internal operating programs and the independent IOP terminal subsystems. The Direct Memory Access channels provide each IOP direct access to memory for transfer of both data and control information. The IOC instruction allows the CPU to transfer information to and from each IOP. This combination of features—direct memory access, program interrupts, and a general I/O instruction—can service very simple terminal IOP's as well as complex ones.

The first terminal IOP to be offered with the TENET 210 supports the Model 33 Teletype and other units compatible with the M-33. Options include direct wire connection or access via the public telephone network. Sixteen full-duplex channels are provided by each IOP. Direct connected terminals have teletype On/Off power features under program control.

Peripherals

The CPU communicates with peripheral processors through the Data Exchange. The IOC instruction, when executed by the CPU, selects a peripheral processor and passes a command to the controller. If the command requires a data transfer, from 1 to 32 bits of data are transferred in either direction between a CPU general register and the selected peripheral processor. This IOC implementation permits substantial flexibility in the communications between the CPU's and the IOP's.

All Direct Memory Access IOP's feature (1) Bi-directional access to all of core memory; (2) Data and order chaining, and (3) Interrupt control in each IO order of a data chain. Each of the following peripherals has direct access to memory:

Disc Memory:

- **Each disc IOP controls one to four 25-million-byte disc drives**
- **Simultaneous seek operations on all disc drives**
- **Data transfer to or from a disc drive with seeks on other drives**
- **128-word sectors; hardware verification of sector address**

Magnetic Tape Unit

- **IBM-compatible, nine-channel, 800 BPI**
- **36K bytes/second**

Card Reader

- **Binary, USASI, and Data Dependent reading mode (USASI or binary card sensitive)**
- **300 or 600 CPM**

Line Printer

- **400 LPM**
- **132 column/line**
- **USASI character set (64 printing characters)**
- **Fully buffered**

Conversational Languages

Software for the TENET 210 provides simultaneous (time-shared) access by 32 or more users, allows real-time data acquisition, and includes a BASIC and FORTRAN compiler.

BASIC

BASIC is recognized by TENET as a key time-sharing language of the seventies, and every effort has been made to make TENET BASIC the most powerful one in the industry.

Some of the features of TENET BASIC are:

Run time interaction

Programmer defined sequence control

String manipulation

Four-character identifiers

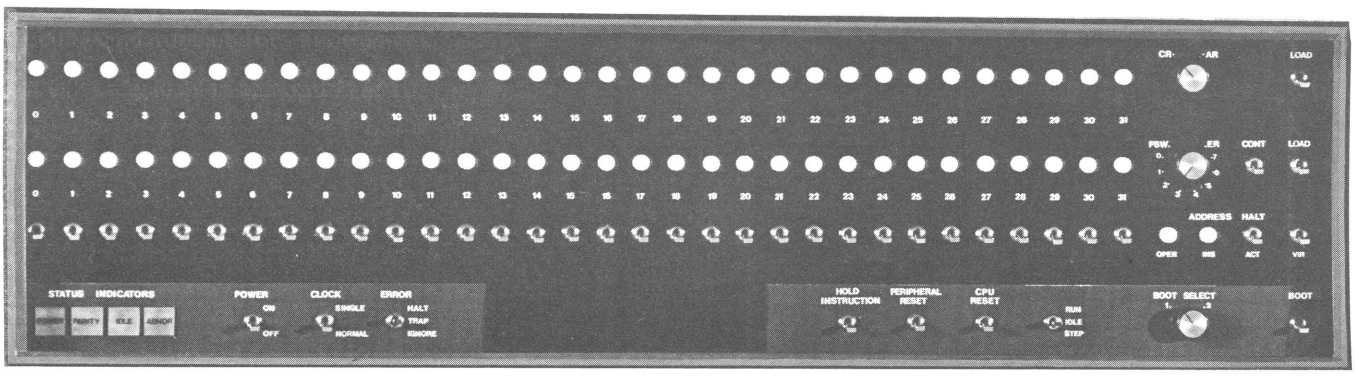
User-defined multiline functions

Subroutine capability

Simultaneous access of 8 files

FORTRAN

TENET FORTRAN IV is an interactive level-H version of the language. It allows compilation and execution of programs of 2000 lines and even larger. The compiler contains comprehensive editing and debugging facilities for on-line checkout of programs. The random file feature allows creation and processing of large files for special situations. Object code is directly generated for fast, efficient program execution.



Software

The TENET 210 system software includes a one-pass Meta Assembler with special, user-tailored capabilities, plus a relocatable linking loader, a basic and a symbolic debugger, an extensive mathematical library, and comprehensive test diagnostics.

Meta Assembler

The Meta Assembler accepts source language input and generates relocatable binary object code and an assembly listing. Included in its single pass is full capability for updating the assembly source. In addition to conventional assembler functions, the Meta Assembler enables the user to define his own programming language and thus program his problem in a language more suitable to his needs. This is made possible by an extremely powerful form of macro facility.

ASSEMBLY CONTROL

DO/ELSE/FIN	Loop control and conditional assembly	DSECT/ISECT/ ASECT	Identifies section where code is to be generated
END	Terminates the assembly	ORG/LOC	Sets load and/or execution location counter(s)
RES/ZERO	Reserves a block of undefined or zero words	MASTER	Enables assembly of privileged instructions
BOUND	Advances location counter to designated boundary		

LISTING CONTROL

TITLE	Establishes heading for listing pages	LPP	Designates desired number of output lines/page
PAGE	Advances listing to next page	LIST	Controls suppression/resumption of listing
SPACE	Spaces lines	ERROR	Generates error indicator on listing

DATA GENERATION

DATA	Generates data values	ADR/MADR	Generates address word
GEN	Generates packed data values	FIELD/MFIELD	Generates field specification for use by field instructions
TEXT/TEXTC	Generates textual information		

SYMBOL MANIPULATION

EQU	Equates a non-resettable symbol to a value	REF/SREF	Declares external references
SET	Sets a resettable symbol to a value	CMN	Declares COMMON symbols
LOCAL	Defines the beginning of a local symbol region	PIN	Defines user PIN instructions
DEF	Declares external definitions		

PROCEDURES

CNAME	Declares a command name for a user procedure	DISP	Controls the listing of procedure expansions
FNAME	Delcares a function name for a user procedure	LBLEQU	Equates the label on the invoking procedure reference line to a value
PROC	Defines the beginning of a procedure	LBLSET	Sets the label on the invoking procedure reference line to a value
PEND	Defines the end of a procedure		

Relocatable Linking Loader

The loader loads one or more binary object modules and links their external symbols. The loader accepts control cards which provide the following facilities.

- Alter one or more memory locations
- Define a value for external definition name
- Specify a load boundary for the next object module
- Specify a load origin for the next object module
- Override the starting address
- Produce an external symbol map on the printer or teletype

Debug

Debug provides the user with a compact, yet comprehensive debugging aid. The debugger accepts commands from the teletype or the card reader and provides the user with the following facilities:

- Alter memory or registers
- Type or print contents of designated memory cells or registers (in hexadecimal)
- Branch to designated memory locations
- Set up to twenty different breakpoints
- Selectively reset any or all of the currently active breakpoints

Super Debug

The symbolic debugger is a very powerful tool which enables the user to debug his program more efficiently. The user may refer to locations within his program symbolically by using external symbol names in addition to absolute hexadecimal memory locations. The features provided by the symbolic debugger include:

- Printing or typing designated memory cells or registers in several formats. The permissible formats are: hexadecimal, USASI, mnemonic instructions, floating point short, floating point long, or a user specified field grouping
- Conditional snapshots
- Expression evaluation
- Masked memory searches
- Symbol creation
- Mnemonic instruction insertions
- Memory or register alteration (may be specified in hexadecimal, floating point, symbolic or mnemonic instruction format)
- Listing all of the currently active snapshots
- Selective removal of any or all of the currently active snapshots
- Addresses may be either virtual or actual (mapped or unmapped)

Mathematical Library

An extensive floating point library is available to the user.

Single and Double Precision

ATANH	ASIN/ACOS	SIN/COS
POWER	TAN	EXP
RNDM	SINH/COSH	LOG
MATRIX INVERSE	TANH	ATAN/ATAN2
MATRIX TRANSPOSE	ASINH/ACOSH	SQRT

Single and Double Precision Complex

SIN/COS	LOG
SQRT	ATAN
EXP	SINH/COSH
	ATANH

In addition, there are library routines for floating point input and output conversions.

Comprehensive Test Diagnostics

Thorough diagnostics are provided for testing the full instruction repertoire, the peripherals, the map, the interrupt system, and for exercising the memory system under a variety of conditions.

T E N E T, Inc. 927 Thompson Place / Sunnyvale, California 94086 / (408) 245 - 8751

PRICE LIST

	Purchase Price	Monthly Maintenance
PROCESSING UNIT & MAIN STORAGE (Notes 1 & 2)		
	Single CPU	
32K Words (128K Bytes)	$220,000	$ 820
48K Words (192K Bytes)	270,000	870
64K Words (256K Bytes)	315,000	915
96K Words (384K Bytes)	400,000	1000
128K Words (512K Bytes)	475,000	1075
	Dual CPU (Note 3)	
32K Words (128K Bytes)	270,000	870
48K Words (192K Bytes)	320,000	920
64K Words (256K Bytes)	365,000	965
96K Words (384K Bytes)	450,000	1050
128K Words (512K Bytes)	525,000	1125
PERIPHERALS & I/O PROCESSORS		
Card Reader, 600 CPM, including Processor	15,000	80
Line Printer, 400 LPM, including Processor	23,500	150
Magnetic Tape Transport, 45 ips, 9 track, 800 bpi	15,000	80
Magnetic Tape Transport IOP (Note 4)	11,000	30
Disc Drive 25M-Byte Capacity, not including Processor	19,500	70
Disc Drive IOP (Note 5)	21,500	50
COMMUNICATIONS		
Communications Controller (Note 6)	4,000	15
Quad Adaptor (Note 6)	2,000	10
SOFTWARE		
Primary Package (Note 7)	1,000	100

Note 1: Processing Unit includes: Data Exchange, Control Panel, Power Fail-Safe and Data Save, Real Time Clock and Interval Timer, Interleaving, Cabinet, and Memory Map.

Note 2: K = 1024, M = 1,000,000 and Byte = 8 bits.

Note 3: The dual CPU includes two Processing Units each with a Control Panel and Memory Map.

Note 4: Services 1-4 Tape Transports.

Note 5: Services 1-4 25M-Byte Disc Drives.

Note 6: The Controller is designed for 0-300 bps terminals. Each Controller supports 1-4 Quad Adaptors. Each adaptor supports 1-4 terminals (16 terminals maximum per Controller). Terminals may be direct- or acoustic-coupled type, but all terminals connected to any one adaptor must be the same type.

Note 7: The primary software package includes: Interactive FORTRAN, Interactive BASIC, Time-sharing Operating System, Statistical Math Package, Meta Assembler, Math Library, Loader, and Debug.

Note 8: The cash purchase prices as shown include a one-year warranty on parts.

Note 9: A 48-month non-cancellable lease plan is offered at a rate of 2.5% of purchase price per month. A purchase credit of 70% of all lease payments will apply to purchase. Lease prices do not include maintenance charges.

Note 10: An installment purchase plan provides for a down payment of 5% of list price and 60 monthly payments of 2% of list price per month.

Note 11: These prices are effective 1 September 1970 and are subject to change without notice.

TENET INC. 927 THOMPSON PLACE, SUNNYVALE, CALIFORNIA 94086, TELEPHONE (408) 245-8751