# Model 632-0001*
# Series 32 Technical Bulletin
## Digital Computer System

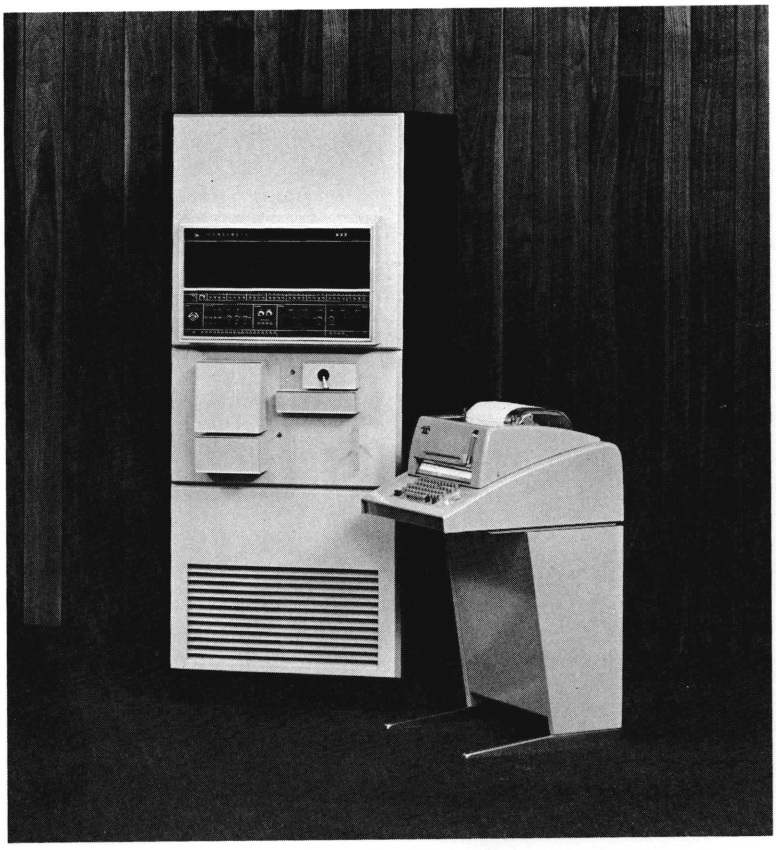
### INTRODUCTION

**General Description** — The H632, first member of the Honeywell Series 32 family of 32-bit real-time computer systems, is a fully integrated-circuit general-purpose digital computer system providing the performance, flexibility and reliability required for a wide variety of small-to-large-scale real-time scientific and control problems.

An advanced concept of modular system integration and coordination permits efficient use of specialized processor capabilities in multiprocessor / multiprogrammed configurations. The H632 is modular in design and construction to facilitate expansions and extensions of basic functions. This modularity permits the user to select an H632 configuration specifically suited for his particular requirements. It also allows for easy field expansion of the system as the requirements grow.

Outstanding features of the H632 system are:

- Unsurpassed system performance
- On-line/real-time operational environment
- Total systems orientation
- Extensive software package
- Complete line of peripheral equipment
- Multiprocessor/multiprogramming capability
- Flexible priority communication structure in systems modules
- Modular construction facilitates systems configurations and field expansion
- Reliability enhanced by extensive use of integrated circuits
- Asynchronous organization allows for the time sharing of systems modules
- Multiprocessing — Every H632 system is a multiprocessor system, with separate central processor of high computational capacity, and input/output processor with extensive input/output capabilities. In a single H632 system, processing capability may be enhanced by including up to four central processors and up to four input/output processors. Primary storage may be configured to permit simultaneous access of up to four independent memory banks, and the sharing of memory banks among processors. Conflicting access requirements can thus be minimized, permitting each of the processors to execute programs of near-maximum rate, while retaining the capability to access common programs and data when required. (See Figure 1 for maximum H632 processor configuration.)



- Multiprogramming — Facilitates the time-sharing of specialized processor capability among loosely related or totally unrelated real-time jobs. Both H632 central processors and input/output processors are multiprogrammable. Each of eight independent programs may either be a candidate for central processor execution, or be waiting for some external event to initiate its candidacy for execution. The highest priority candidate is automatically selected for execution. Similarly, each of eight concurrently active I/O channel programs may require program execution or data transfer; the I/O processor services these requirements in order of their urgency. Multiprogramming capability may be expanded with up to 40 central processor priority-selected programs, and up to 16 I/O processor concurrent channel programs.

*Model 632-0001 does not include punched paper tape reader and punch, as shown in photo above.

**Honeywell**

COMPUTER CONTROL DIVISION

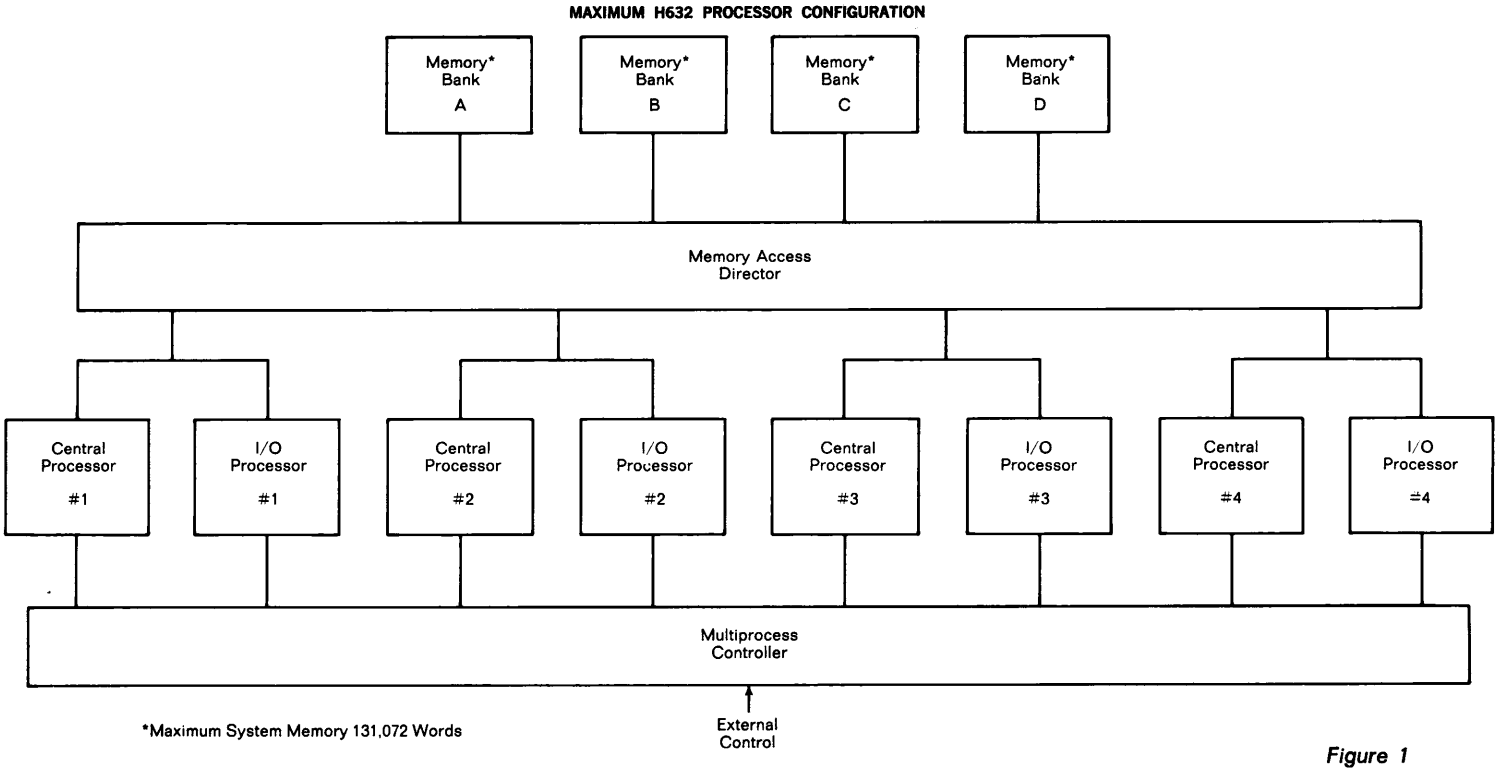



*Figure 1*

**System Organization** — The basic (Figure 2) H632 system, Model No. 632-0001, consists of 8,192 words of primary storage; a Central Processor (CP); an Input/Output Processor (IOP); a KSR-35 I/O typewriter connected to the IOP via a device controller; the system coordinating elements — Memory Access Director (MAD) and Multiprocess Controller (MPC); and a System Control Panel (SCP).

An H632 system is implemented by combining the five basic systems modules.

1. *Primary Storage* — The standard memory module for the H632 contains 8,192 words, each having 32 bits. Memory can be expanded from the basic 8,192 words to 131,072 words in 8,192-word modules. Depending on the Memory Access Director selected, memory can be organized in banks, each bank containing one or more 8,192-word modules and a separate bus system to which one or more processors may have access. Up to four banks are permitted on an H632 System, provided total memory does not exceed 131,072 words. Features are:

- Coincident-current, random access ferrite core
- 32-bit word
- Parity option
- 850-ns cycle time
- Expandable from the basic 8,192-word system to 131,072 words with pluggable memory modules
- All major functions performed with monolithic integrated circuits
- Low power consumption
- Standard product line ICM-500 memory system

2. *Memory Access Director (MAD)* — The Memory Access Director controls the access paths between core memory and the CP and IOP. The MAD resolves simultaneous memory access requests issued by two or more processors. By proper selection and connections of MAD units, shared and/or private memory banks may be achieved. Features are:

- Multiprocessor access to multimemory banks allows simultaneous operations
- Establishes common and/or private memory in multiprocessor/multimemory bank systems
- Resolves simultaneous access requests
- Allows for asynchronous CP, IOP and memory operations

3. *Central Processor (CP)* — The H632 Central Processor is a general-purpose, multiprogrammable, word-oriented, scientific data processor with limited optional I/O capability. There are eight standard program levels, optionally expandable to 40. The CP performs all data manipulation, arithmetic, logical, and comparison operations; and issues orders requesting changes in activity states of various program levels. Features are:

- 32-bit word-oriented
- Two's complement arithmetic
- Direct addressability of 65,536 32-bit words or 131,072 16-bit halfwords
- Bit, byte, halfword, word, doubleword, and immediate operand addressing
- Storage-to-Register, Register-to-Storage, Register-to-Register and Immediate Operand Operations
- 16 general-purpose high-speed, 32-bit flip-flop registers which may be used as:
  - accumulators
  - index registers (7)
  - mask register (1)
  - temporary storage
- Multilevel indexing and indirect addressing
- Instruction look-ahead
- Automatic trapping of arithmetic faults, unimplemented optional instructions, undefined instructions, and error conditions
- 144 high-speed instructions, including
  - Load and store operations
  - Logical operations
  - Fixed-point arithmetic operations
  - Compare operations
  - Shift operations
  - Bit-testing and bit-modification operations
  - Jump and Execute operations
  - Control operations
  - Floating point operations (optional)
  - Mask and byte operations (optional)
  - Direct read/write (optional)
- Typical Instruction Execution Times
  (Times include instruction and operand fetch and indexing if required.)
  - Load word — 1.7 $\mu$s
  - Add word — 1.7 $\mu$s
  - Multiply word — 8.1 $\mu$s
  - Floating add — 5.1 $\mu$s
  - Floating multiply — 9.4 $\mu$s
- Central Processor Options
  - Floating-point operations
  - Mask and byte operations
  - Additional interrupt levels
  - Direct read/write
  - Parallel input channels
  - Parallel output channels
  - Sense-line groups
  - Output-control pulse groups

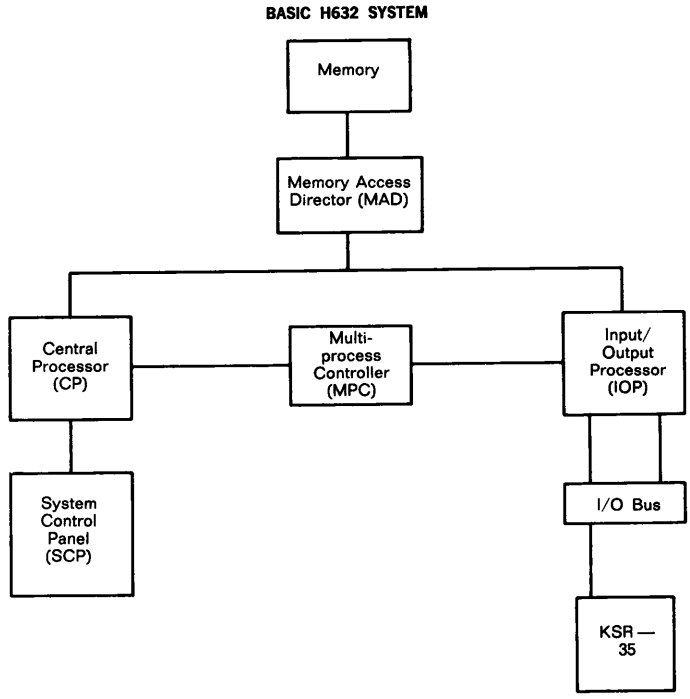



*Figure 2*

4. *Input/Output Processor (IOP)* — The H632 Input/Output Processor is an independent, multiprogrammable processor with highly developed I/O facilities and limited data processing capability. It is word oriented and provides direct interpretation of I/O commands fetched from memory concurrently with CP operation. Data is also directly transferred between memory and the selected device(s) without CP intervention. Modular I/O control capabilities of the H632 permit utilization of a full range of peripherals. The communication path to any device controller may be over any IOP channel. The standard IOP has eight I/O channels, optionally expandable to 16 channels. Features are:

- Simultaneous independent operation with the CP on a cycle-stealing basis
- Provision for a direct data path between memory and peripheral devices
- Programmable real-time I/O with command and data-chaining capability
- High-speed, word-oriented, up to 200 kHz word-transfer rate
- Eight channels standard, expandable to 16 channels
- Word, halfword, or byte transfers
- Dynamic channel/controller assignment
- Multiplexer/selector capability

5. *Multiprocess Controller (MPC)* — Each H632 processor (CP and IOP) can be time-shared among several programs. The MPC provides logical control over the interaction between CP, IOP, and external interrupts, and serves as a clearing house for interprocess and intraprocess state control in the system.

The activity states of all programs in all processors are recorded within the MPC. Orders to change the state of a process are transmitted to it. Simultaneous service requests for program execution are resolved by the MPC. The MPC identifies the active process of highest priority for each processor in the system. This information is then transmitted to the appropriate processor for action. Features are:

- Provides system coordination/control in multiprogram/multiprocessor environments
- Indicates current highest-priority task to the CP and IOP
- Resolves simultaneous service requests
- External interrupt and control (optional)

**System Peripheral Options** — All options have USASCII compatible operating modes.

*Seven-level magnetic tape*

36-ips, 200/556 bpi, 7.2-20.0 kHz character transfer rate
36-ips, 200/800 bpi, 7.2-28.8 kHz character transfer rate
36-ips, 556/800 bpi, 20.0-28.8 kHz character transfer rate
80-ips, 200/556 bpi, 16.0-44.4 kHz character transfer rate
80-ips, 200/800 bpi, 16.0-64.0 kHz character transfer rate
80-ips, 556/800 bpi, 44.4-64.0 kHz character transfer rate

*Card Reader*

400 cards per minute, 80-column card
800 cards per minute, 80-column card

*Card Punch*

100 to 400 cards per minute, 80-column card

*High-Speed Paper Tape*

300-character-per-second reader
110-character-per-second punch

*Mass Storage*

Moving head disc unit
1.8 million to 28.8 million words
Average latency — 12.5 ms
Access times — 1 track seek 30 ms
200 track seek 165 ms
Word transfer rate — 39 kHz

*Line Printer*

300 lines per minute, 120 columns/line

*Real-Time Subsystems*

Real-time clock
Watchdog timer
Power failure interrupt

**Custom Options**

- High-speed line printer
- High-speed magnetic tapes
- Nine-channel magnetic tapes
- Fixed head disc system
- Analog/digital and digital/analog converters
- Series 32/Series 16 adapter
- Graphic displays
- Alphanumeric displays
- Single-line communications controllers
- Multiline communications controllers

**Systems Software** — The extensive H632 software package consists of:

*Assembler*

- Nested macros
- Pseudo-operations for control, storage assignment, data definition, and conditional assemblies
- Absolute and relocatable coding
- Extensive error flagging
- Symbol table output including usage table
- Side-by-side listing

*Fortran IV*

- Standard ASA form including math library
- Extensions to include in-line code, multiple subscripts, implied Hollerith count, zero/negative subscripts and DO loop parameters, multiple subroutine entry points and nonstandard returns
- Extensive error flagging

*Loader*

- Automatic subroutine linkage
- Compatible for all source program translations
- Memory map
- Produces relocatable "load mode" programs

*Subroutine Library*

- Fixed-point and floating-point math libraries
- I/O library

*Debug Package*

*System Editor* — for editing standard library, user subprogram library, and user source files.

*OS-1* — for interpreting control commands which initiate execution of system and user programs.

*Unit Record Control* — provides for run-time assignment of peripheral I/O devices and device modes and the management of files of I/O information.

*Trap Package*

*Media Conversion*

*Test and Maintenance Routines*

**Technology** — Some of H632 technological highlights are:

- $\mu$-**PAC** monolithic integrated-circuit modules
- Machine-wired back planes
- Conventional manufacturing processes
- Cable connection of all systems modules
- H632 parallel machine organization permits use of moderate-speed circuitry and wide reliability performance margins
- High packing density — up to 32K-word system with CP, IOP, control console, paper tape reader (300 cps) and paper tape punch (110 cps) fit into a single rack.

**System Configurator** — (See Figure 3).

# H632 System Configuration

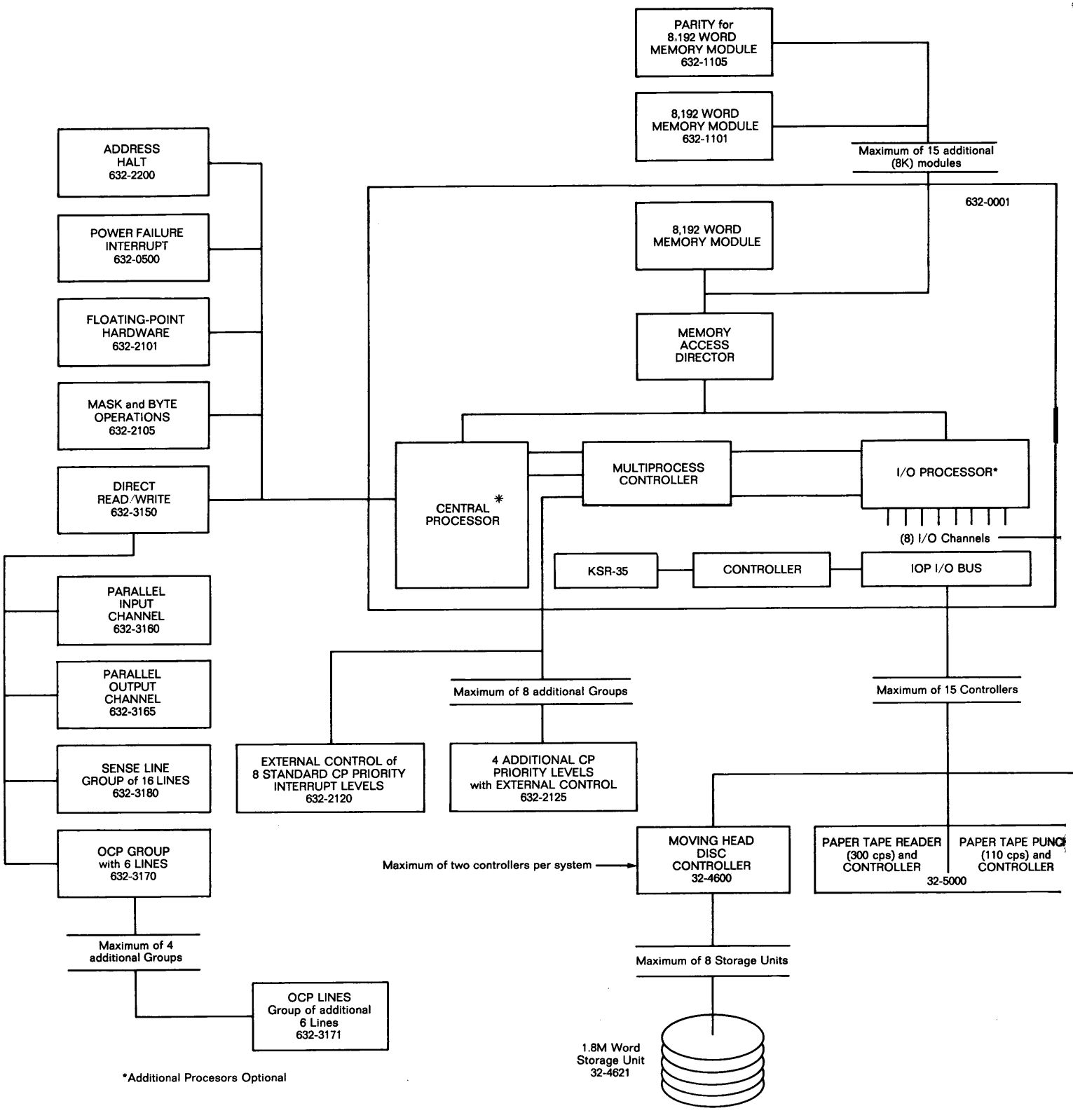
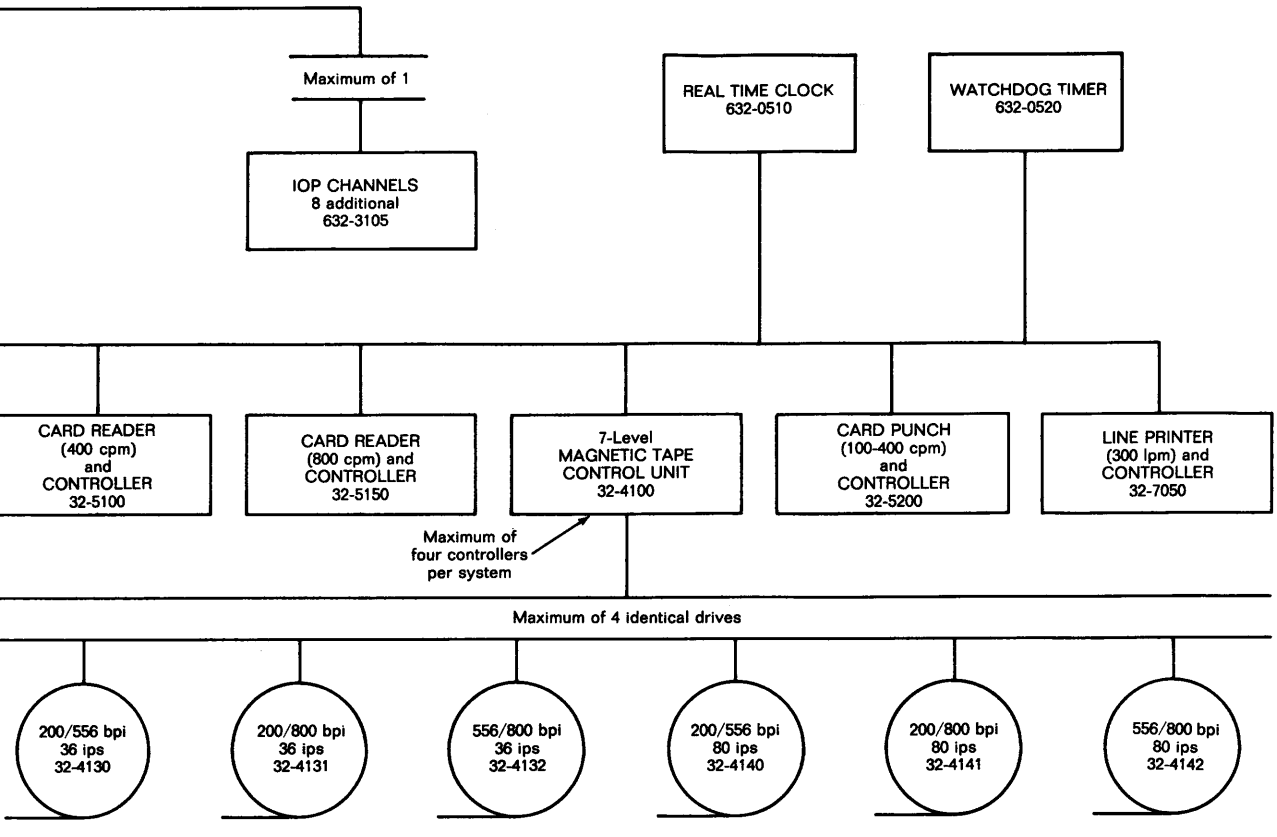



*Figure 3*


Maximum of 1
IOP CHANNELS
8 additional
632-3105
REAL TIME CLOCK
632-0510
WATCHDOG TIMER
632-0520
CARD READER
(400 cpm)
and
CONTROLLER
32-5100
CARD READER
(800 cpm) and
CONTROLLER
32-5150
7-Level
MAGNETIC TAPE
CONTROL UNIT
32-4100
CARD PUNCH
(100-400 cpm)
and
CONTROLLER
32-5200
LINE PRINTER
(300 lpm) and
CONTROLLER
32-7050
Maximum of
four controllers
per system
Maximum of 4 identical drives
200/556 bpi
36 ips
32-4130
200/800 bpi
36 ips
32-4131
556/800 bpi
36 ips
32-4132
200/556 bpi
80 ips
32-4140
200/800 bpi
80 ips
32-4141
556/800 bpi
80 ips
32-4142

## PRIMARY STORAGE

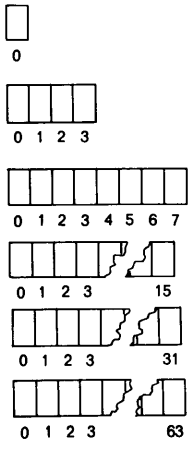
**General Description** — Primary storage is the high-speed random-access storage which is directly accessible to H632 System processors. The primary storage subsystem provides system processors with the accessibility of 8,192 to 131,072 32-bit storage cells. A small number of these cells (256) are dedicated to specific system storage requirements; the remainder may be occupied by programs and/or data.

Primary storage consists of from one to 16 core memory storage modules, each providing 8,192 individual 32-bit storage cells. Each 32-bit cell is identified by a unique 17-bit binary number. Four bits are used to select the module within which the cell resides; the remaining 13 bits identify the cell within the module.

**Operation** — Memory modules are cable-connected to the Memory Access Director (MAD), which controls memory accesses. A memory address is furnished to the MAD by the processor requesting a memory access. The MAD interprets this address as a module address and an address within the module.

The latter address is then transmitted to the selected module. Upon receipt of this address a memory access is performed and the content of the cell is transmitted or modified according to the operation being performed by the processor under consideration.

**Information Units** — Physically, H632 primary storage is simply an array of 32-bit storage cells. For processing purposes, it is convenient to further refine the structure of the stored data by definition of fixed-length information units, or *fields*. Explicitly recognized fields within the H632 System are:




The *one-bit field* is represented by a binary digit, or *bit*, either 0 or 1.

The *four-bit field* is represented by a hexadecimal digit, or *hex digit;* hex digits beyond 9 are : A,B,C,D,E, and F.

The *byte*, an eight-bit field is represented by two hex digits.

The *halfword*, a 16-bit field, is represented by four hex digits.

The *word*, a 32-bit field is represented by eight hex digits.

The *double word*, a 64-bit field, is represented by two words.

Bit positions within a field are assigned consecutive numbers, from left to right, starting with 0. When a word is to be explicitly recognized as such in storage, it is required to occupy a single storage cell.

A four-bit field, to be explicitly referenced in storage, is required to be in one of the eight nonoverlapping hex-digit positions of a word occupying a single storage cell. Similarly, a byte must be one of the four nonoverlapping bytes of a word occupying a single storage cell; a halfword must be one of the two nonoverlapping halfwords of a word occupying a single storage cell. Doublewords may occupy any two consecutively numbered storage cells.

## MEMORY ACCESS DIRECTOR (MAD)

**General Description** — The Memory Access Director (MAD) consists of interface logic which controls communications between processors and memory.

**Operation** — The communication between processors and memory is controlled by a request-response dialogue between the processor and the MAD. In order to access memory with the next available memory cycle, a processor sends an Access Request signal to the MAD. Upon the determination that memory is available for use by the processor, an Address Select signal is transmitted to the processor requesting access. Upon receipt of the Address Select signal the processor sends a word address and a Cycle Initiate signal to the memory via the MAD. Since the control of the Address and Cycle Initiate timing rests entirely with the processor, it regulates the memory data transfer rate. Cycle Request and Address Select signals are asynchronous with respect to one another.

Up to two processors may be connected to a MAD processor port. The processors sharing the port also time share a cable system on a memory cycle-stealing basis; therefore, only one of these processors may access memory at a time. Signals generated by the MAD inform the selected processor that it has exclusive use of the bus on which data is transmitted. Once an Access Request has been received and acted upon by the MAD, memory cannot be accessed by any other processor until the selected processor has initiated and completed an access.

In each MAD, the memory module addresses are specified independently and uniquely for each processor. Since two processors may share a MAD processor port, priority logic is required to resolve scheduling conflicts. This priority determination is made within the MAD. Processors sharing a port have fixed priorities relative to one another, with the IOP having higher priority than the CP.

## CENTRAL PROCESSOR (CP)

**General Description** — The H632 Central Processor is a modified single-address, 32-bit-word-organized, stored-program computer with 16 addressable high-speed registers. It performs operations — data transfers, arithmetic, logical, and control — according to instructions fetched from a list of instruction words (the program) residing in primary storage.

**Operation** — The CP issues memory access requests to the MAD. If memory is not being accessed by another processor (typically an IOP), the CP access request will be acknowledged. The contents of the addressed memory location are then presented to the CP for processing. If the memory word accessed was an instruction, the various subfields of the instruction word are processed and operations are performed according to instruction definition. After instruction decoding, the operand address, where needed, is formed. Memory is again accessed under control of the MAD and the instruction operand is presented to the CP for processing.

**Internal Organization** — Figure 4 is a simplified block diagram of the H632 Central Processor.

The CP contains 16 general registers, an arithmetic and logical unit (ALU), a sequencing and control unit (SCU), and two program stateword registers ($PS_1$ and $PS_2$). The ALU performs arithmetic, logical, and shift operations on data supplied from primary storage or retained in the 16 addressable registers. The SCU fetches and interprets instructions and controls the functions of the ALU throughout instruction execution.

*General Registers* — The 16 addressable 32-bit registers are numbered, in hexadecimal notation, 0 through F, and referred to as R0 through RF.

R0 is a special register, implicitly referenced in masking operations, reference to which is subject to certain restrictions.

R1 through RF are general registers, usable by all instructions as operand or result locations.

R1 through R7 can also serve as index registers in address formation.

*Shift Network* — Contains a parallel shift network which performs data transformations.

*Arithmetic Unit* — Contains four auxiliary registers which supply operands to the processing networks, retain partial results, and satisfy several interface requirements.

*Sequencing and Control Unit.* — Controls the interface of the CP with primary storage, retains and acts on the basic control information required for program sequencing and execution. The SCU is capable of preparatory interpretation of one instruction while controlling the execution of another. This "instruction look-ahead" feature permits high-speed operation with conservative speed circuitry.

CENTRAL PROCESSOR SIMPLIFIED BLOCK DIAGRAM

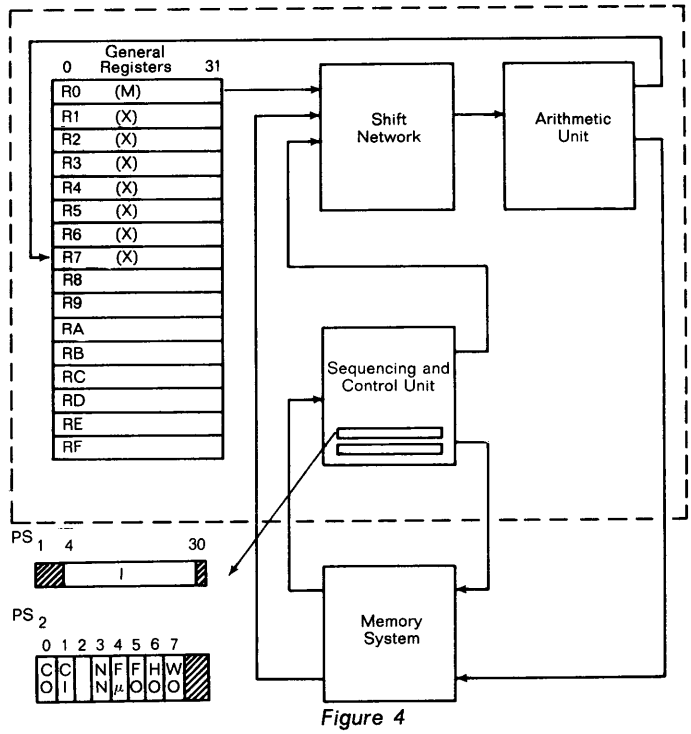



*Figure 4*

*Program Status* — Dynamically retained in two registers which contain all H632-defined portions of the Series 32 program status double word, as follows:

| | |
|---|---|
| I, | Instruction address, identifying the storage cell containing the next instruction to be fetched and executed |
| $C_0C_1$, | Condition codes, retaining an indication of the result of a compare operation |
| NN, FU, FO, | Floating-point mode controls |
| HO, WO, | Fixed-point mode controls |

*Instructions* — The H632 possesses a repertoire of 144 high-speed, systems-optimized instructions. With the H632 structure, the user can work directly with various data sizes: *words* (32 bits), *halfwords* (16 bits), *bytes* (8 bits), *hex digits* (4 bits), *single bits* and *doublewords* (64 bits).

The CP allows for:

*Storage-to-register operations* (e.g., add a word in memory to a word in a register and place the result back in the register), *register-to-storage operations* (e.g., add a word in memory to a word in register and place the result back in memory), *register-to-register operations*, and *immediate operand operations.*

The H632 CP offers 11 different classes of instructions:

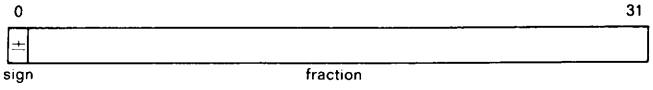
1 — Load/Store
2 — Logical
3 — Fixed-Point Arithmetic
4 — Compare
5 — Shift
6 — Bit
7 — Jump and Execute
8 — Control
9 — Floating Point (Optional)
10 — Mask and Byte (Optional)
11 — Direct Read/Write (Optional)

A complete listing of the standard instruction repertoire is shown in Table 1.

*Data Formats* — An operand fetched from memory is presented to the CP in the form of a 32-bit word. The whole word (32 bits), or some portion of the word (16 bits, eight bits, four bits, or a single bit) is then processed according to that particular instruction definition. Number representation is binary. Arithmetic processing is by two's complement arithmetic. In processing halfwords, sign extension is used. Thus, a 16-bit binary fraction is extended to 32 bits before any arithmetic operation is performed.

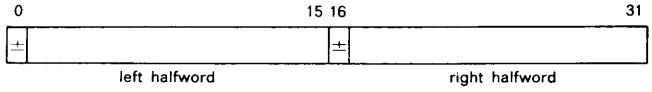
In some special cases, sign extension is inhibited or sign interpretation is ignored, enabling operations on unsigned numbers. The CP also has the ability to use immediate operands. In this type of instruction, the address field of the instruction (including bit 15) is interpreted as a 17-bit two's complement binary number. Sign extension is performed prior to any arithmetic operation, facilitating operations involving constants to be performed wthout using additional storage.

*Fixed-Point Single-Precision Fraction*

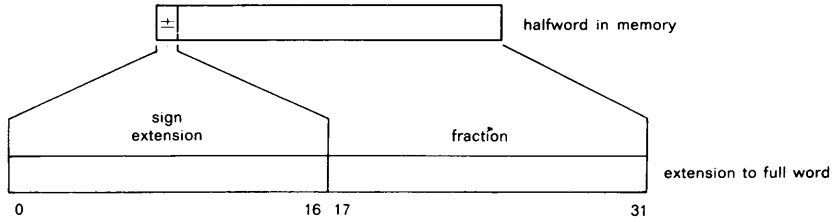



The range of numbers represented in this format is from $-1$ to $+1 - 2^{-31}$ inclusive.

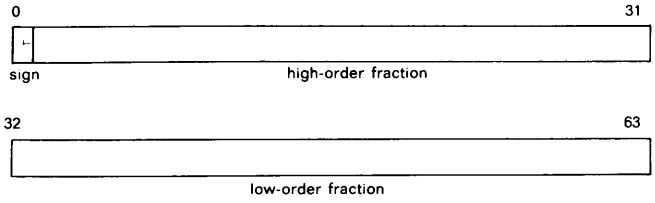
*Halfword Format*




Halfwords are extended to full words for processing.




*Doubleword Format*




The range of numbers represented in this format is from $-1$ to $+1 - 2^{-62}$.

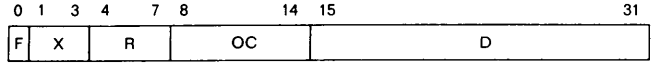
Bit 63 is ignored in operands.

*Instruction Formats* — There are five types of instruction word formats. They are:

Memory Reference
Shift
Bit
Immediate
Control Process

The formats for these classes of instructions are as follows:

*Memory Reference*



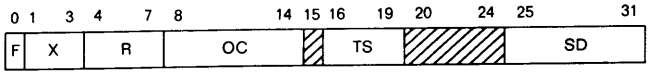

F — indirect address flag. When set to a 1, indirect addressing is performed.
X — index register specifier. This three-bit field indicates which of the general registers, R1 through R7, is to be used as an index register. If this field is zero, no indexing is performed. Indexing takes place before indirect addressing.
R — general register specifier. This four-bit field indicates which of the general registers, R0 through R15, is to be used in the execution of the instruction.
OC — operation code. This seven-bit field specifies the code of the instruction or operation to be performed.

*Displacement Address* — This 17-bit field allows for the direct addressability of 131,072 halfwords or 65,536 words. Word operations ignore bit 31 in address information.

When the effective halfword address is less than 16, a general register rather than a memory location is referenced, providing *register-to-register* operations.

When memory is expanded beyond 65,536 words, one must then consider the memory as being divided into sectors of 32,768 words each. A 131,072-word memory system will have four sectors. In this case bit 15 of the instruction no longer is considered to be part of the D field, but is defined as being a sector-indicator. If the sector bit (bit 15) of the current instruction is zero, reference is made to sector zero of the memory (first 32,768 words). If the sector bit of the current instruction is one, reference is made to the current program sector.

*Shift*




R — general register specifier. The R field specifies the general register on which the shift operation is to be performed. For doubleword shifts, left shifts are towards register R as specified, right shifts are towards register R + 1 (if R = 15, R + 1 is defined to be equal to R0).

OC — operation code. Indicates single- or doubleword shift operation.

TS — type of shift. This field indicates the type of shift to be performed:
  a. closed logical
  b. unpack
  c. open logical
  d. arithmetic

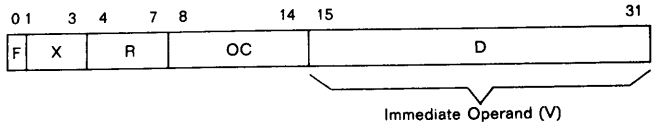
SD — shift distance and direction. This field is interpreted as a seven-bit two's complement integer defining the shift distance and direction. Positive shifts are to the right; negative shifts are to the left.

If indexing or indirect addressing are called for within a shift operation, bits 16-19 and 25-31 of the effective address are used as TS and SD specifiers.

*Bit* — For bit operations, the Memory Reference format is applicable with the exception of R-field interpretation. The R-field is used to refer to a single bit within the halfword pointed to by the effective halfword address, rather than being used as a register designator.

If the effective halfword address is greater than 0 and less than 16, the most significant 16 bits of the addressed general register may be tested and/or modified. If the effective halfword address is zero, the first 16 bits of Program Stateword two may be tested and/or modified. The sense switches may be tested when the effective halfword address equals zero and the R field is A, B, C or D.

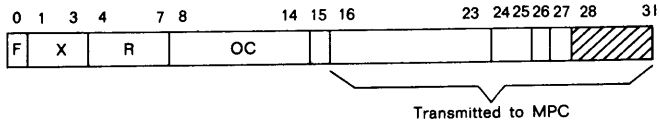
*Immediate*




Bits 15-31 of the effective address are used as an operand rather than as the address of the operand location in memory. The V field (bits 15-31) is sign extended (bit 15 copied into bits 0-14) before it is used as an operand.

If an immediate operation is performed using indexing or indirect addressing, sign extension is performed as follows: if bits 4-15 of the effective address are all zeros, bits 0-3 are made zeros; if bits 4-15 of the effective address are not all zeros, bits 0-15 are made all ones. The effective address is then used as the value of an operand.

*Control Process*

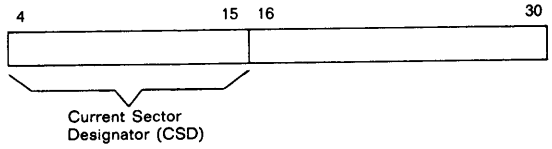



Bits 16-31 of the effective address are transmitted as an order to the MPC.

*Addressing* — The CP processes each instruction in two phases. The first phase, instruction fetch and address formation, is performed identically for all instructions. The second phase, instruction execution, is unique for each instruction. This section deals strictly with the first phase.

*Instruction Fetch* — An Instruction Address, I, maintained by the CP, points to a cell in main storage which contains the next instruction to be executed.

Format — Instruction Address Counter (I)



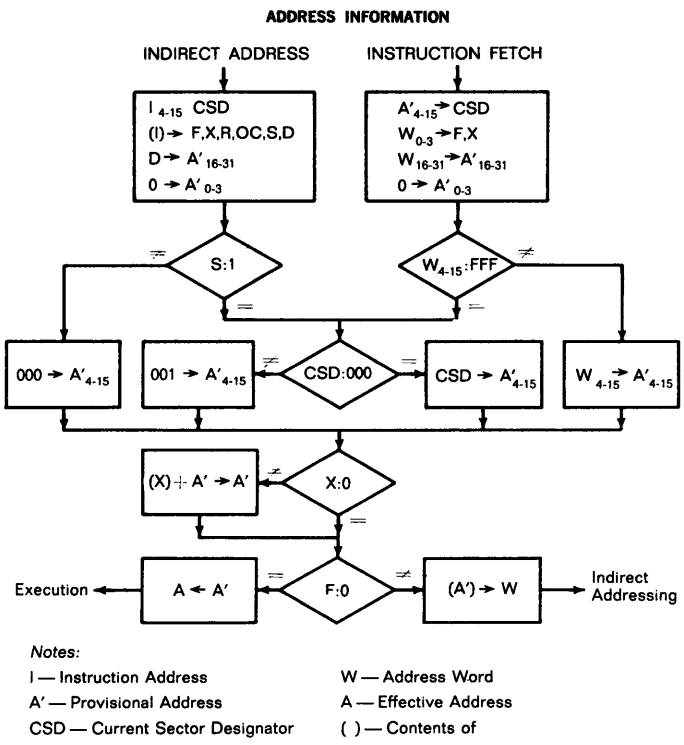



*Figure 5*

Since instructions are contained in full 32-bit words, bit 31 of the instruction address is always interpreted as being equal to zero. A portion of the instruction address may be used in the formation of the effective address of the operand of the instruction being processed. This portion is referred to as the Current Sector Designator, CSD, and indicates the sector, a block of 32,768 words of storage, which is currently being pointed to by I.

*Address Formation* — When execution of an instruction (including any trap-action by the processor) is complete, and any required interrupt action by the processor has been taken, the processor's next task is "sequential instruction-fetch." A sequentially fetched instruction is taken from the word whose address is *I*. Sequential fetch of an instruction includes the incrementation of *I* by 1 *after* effective address-formation is completed and before any further change in *I* required in the execution phase.

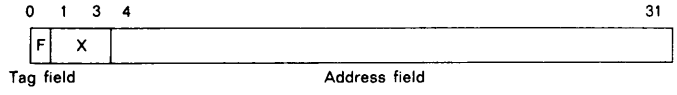
After an instruction has been fetched, a provisional address is formed (see Figure 5) using the D field of the instruction word and, conditionally, CSD. For systems with *65,536 words or less* of memory, one *need not* concern himself with the sector bit. For these systems, bits 15-31 of the instruction are used as a direct address field.

For systems with more than 65,536 words of memory, the user must think of his system as consisting of three or more (four maximum) sectors, with each sector being equal to 32,768 words. For these systems, the D field of the instruction references a particular word, either in sector zero (S-bit = 0) or in the current program sector (S-bit = 1).

The exception to this is when an instruction with the S-bit = 1 is being executed out of sector zero. In this case, the D field specifies a location within sector 1 and not within the current sector zero.

If the index specifier, field X, is non-zero, the contents of register No. X are added to the provisional address. If the X field is equal to zero, the provisional address is unmodified.

If the Flag-bit, F, is zero, the effective address A is used in the execution phase. If F is 1, the provisional address is used as an indirect address to address a word in main storage or in a general register, from which an address word, W, is fetched. A new provisional address is then formed from the address word and conditionally, a new CSD. This new provisional address may be indexed or used as an indirect address reference as discussed above. Indirect address words have this format:




Indexing occurs prior to indirect addressing in address formation. Multilevel indexing and/or indirect addressing is permitted.

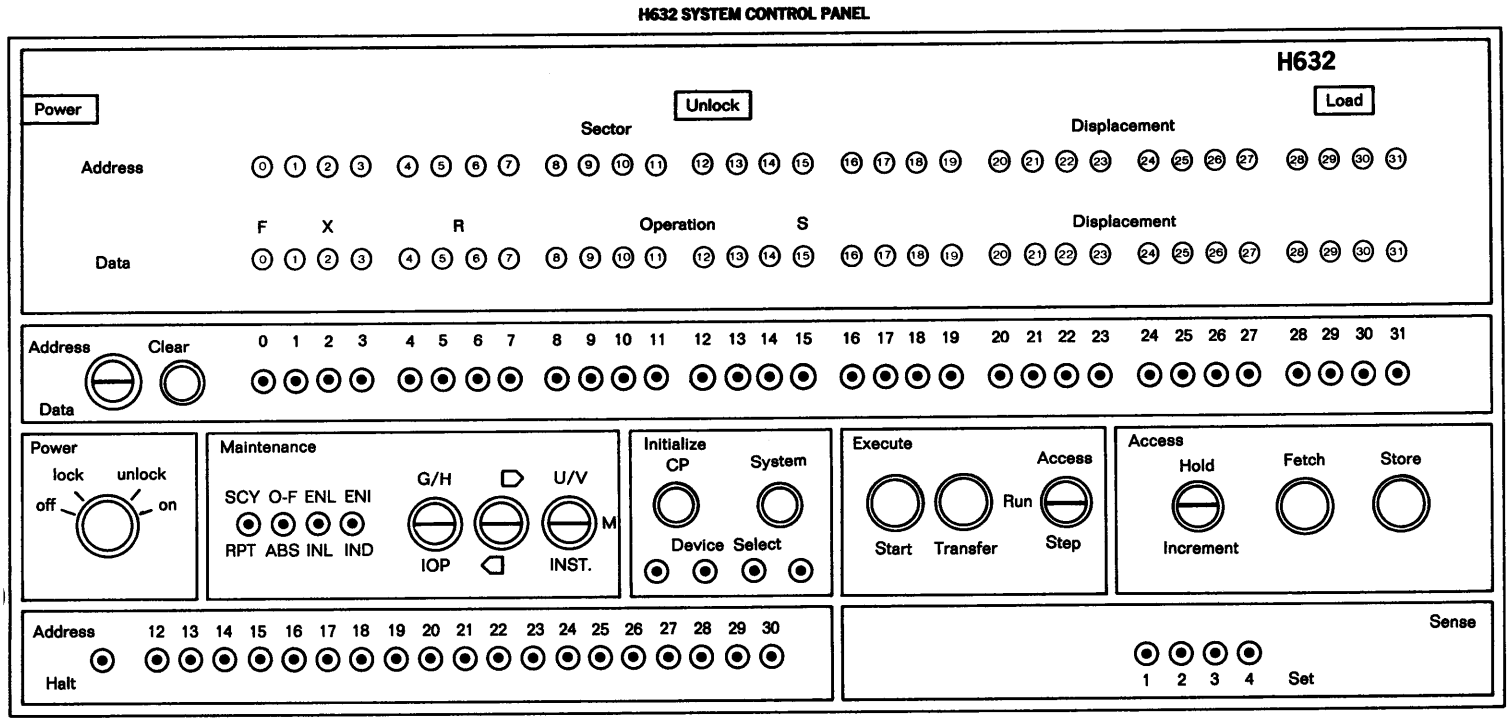



*Figure 6*

*Trap System* — Exceptional conditions, arising during instruction fetch and/or execution, cause trapping. The resulting change in program execution is dependent upon the condition that has caused trap action to occur. There are four general classes of trap conditions (a dedicated memory location is associated with each class):

Class
- I Arithmetic Exception
- II Unimplemented Instructions
- III Unimplemented Instructions
- IV Undefined Instructions

Class I traps occur when word or halfword overflow have occurred as the result of fixed-point arithmetic operation.

Class II and III traps occur when the execution of an unimplemented instruction is attempted. An example of this would be the attempted execution of a floating-point operation without the floating-point option. This class of trap may be used to simulate unimplemented instructions.

Class IV traps occur when the execution of an undefined instruction is attempted.

*System Control Panel* — The H632 System Control Panel (Figure 6) consists of a comprehensive set of indicators, displays, and controls used to indicate, and allow for the modification of, the state of the system.

The control panel provides these capabilities:

*State Indicators* — Indicate the state of the following system conditions:

a. A. C. Power
b. D. C. Power
c. Load
d. Run
e. CP Status
f. IOP Status
g. Single Cycle
h. Repeat
i. Access Memory Locations O-F
j. Inhibit Interrupt
k. Inhibit Look Ahead
l. Address Halt (Optional)

*Binary Displays* (upper and lower) — allow for the display of:

a. Memory Address, Memory Data Register
b. Program Stateword 1, Program Stateword 2
c. U- and V-networks
d. Instruction Register and Timing
e. CP Activity States
f. IOP Activity States

*System Control*

a. Power Switch
b. CP Initialize
c. System Initialize
d. Device Selection

*Display Entry — Data Entry*

a. Address/Data Switch
b. Clear Switch
c. Binary Entry Switches

*Execute Control*

a. Access/Run/Step
b. Start
c. Transfer Control

*Storage Access*

a. Fetch
b. Store
c. Hold/Increment

*Maintenance Display Control*

a. Group Select
b. Group I
   1. U- and V-networks
   2. Memory Address and Memory Data
   3. Instruction Register and Timing
c. Group II
   1. G and H Registers
   2. IOP Activity States
   3. CP Activity States

*Maintenance Function Control*

a. Look-Ahead
   enable/inhibit
b. Repeat
   single cycle/repeat
c. Interrupt
   enable/inhibit
d. Addressing
   0-F

*Address Halt Control and Switch Register* (Optional)

*Sense Switches* — Four are provided. Sense switches are tested by using bit testing instructions.

**Table I — H632 Central Processor Instruction Repertoire**

Notes

| | |
|---|---|
| + | Halfword overflow possible |
| ++ | Word overflow possible |
| EWA | Effective word address |
| EHA | Effective halfword address |
| EDA | Effective doubleword address |
| ( ) | Contents of |
| I | Instruction address |
| R | General Register specified in instruction |
| $\wedge$ | Logical AND |
| V | Logical OR |
| $\forall$ | Logical exclusive OR |
| $\rightarrow$ | Replaces |
| SE | Sign extended |
| ZE | Zeros extended |
| \| \| | Absolute value |
| $C_0C_1$ | Condition code indicators |
| V | Value (bits 15-31) of instruction address field |
| $(EHA)_R$ | Points to the bit, specified by the value of the R field in the instruction, in the halfword being addressed |
| EA | Effective address (word or halfword according to instruction) |

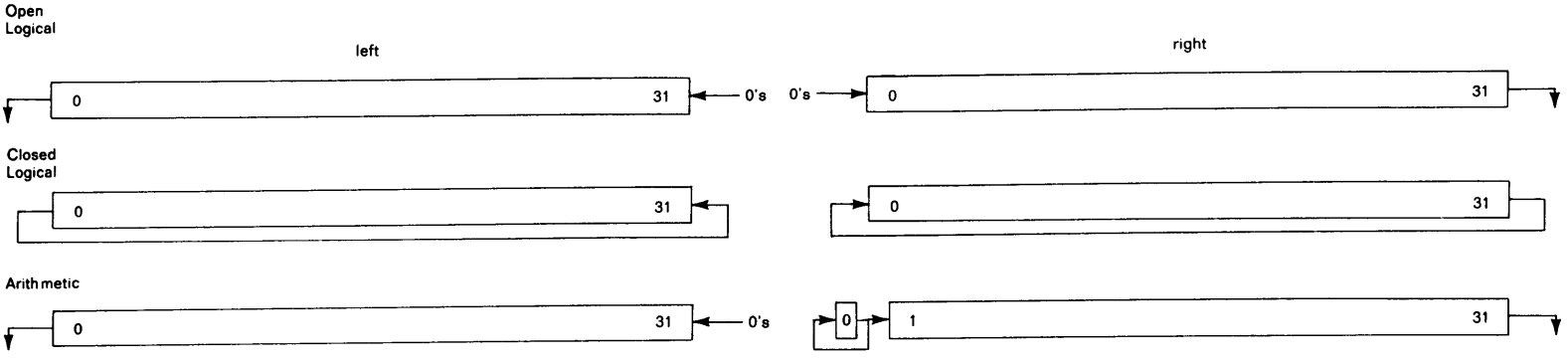
| OPERATION | MNEMONIC | FUNCTION |
|---|---|---|
| **Load/Store Operations** | | |
| Load Word | LDW | (EWA) → (R) |
| Load Halfword | LDH | $(EHA)_{SE}$ → (R) |
| Load Value | LDV | $V_{SE}$ → (R) |
| Store Word | STW | (R) → (EWA) |
| Store Halfword | STH | (R) → (EHA) + |
| Load Doubleword | LDD | (EDA) → (R,R+1) |
| Store Doubleword | STD | (R,R+1) → (EDA) |
| Load Unsigned Halfword | LDUH | $(EHA)_{ZE}$ → (R) |
| Store Unsigned Halfword | STUH | (R) → (EHA) |
| Swap Word | SWW | (R) ⇆ (EWA) |
| Swap Halfword | SWH | (R) ⇆ $(EHA)_{SE}$ + |
| Store Registers | STR | (R thru 15) → (EWA thru EWA +15−R) |
| Load Registers | LDR | (EWA thru EWA +15−R) → (R thru 15) |
| Load Address | LEA | (EWA) → (R) |
| **Logical Operations** | | |
| OR Word | ORW | (EWA) V (R) → (R) |
| OR Halfword | OHR | $(EHA)_{SE}$ V (R) → (R) |
| OR Value | ORV | $V_{SE}$ V (R) → (R) |
| OR Word to Storage | ORS | (EWA) V (R) → (EWA) |
| OR Halfword to Storage | ORHS | $(EHA)_{SE}$ V (R) → (EHA) + |
| Exclusive OR Word | XOW | (EWA) ⊻ (R) → (R) |
| Exclusive OR Halfword | XOH | $(EHA)_{SE}$ ⊻ (R) → (R) |
| Exclusive OR Value | XOV | $V_{SE}$ ⊻ (R) → (R) |
| Exclusive OR Word to Storage | XOS | (EWA) ⊻ (R) → (EWA) |
| Exclusive OR Halfword to Storage | XOHS | $(EHA)_{SE}$ ⊻ (R) → (EHA) + |
| AND Word | ANW | (EWA) ∧ (R) → (R) |
| AND Halfword | ANH | $(EHA)_{SE}$ ∧ (R) → (R) |
| AND Value | ANV | $V_{SE}$ ∧ (R) → (R) |
| AND Word to Storage | ANS | (EWA) ∧ (R) → EWA |
| AND Halfword to Storage | ANHS | $(EHA)_{SE}$ ∧ (R) → EHA + |
| Fill Word | FLW | 1's → (EWA) |
| Fill Halfword | FLH | 1's → (EHA) |
| Clear Word | CLW | 0's → (EWA) |
| Clear Halfword | CLH | 0's → (EHA) |
| Invert Word | IVW | One's complement (EWA) → (EWA) |
| Invert Halfword | IVH | One's complement $(EHA)_{SE}$ → (EHA) |

| OPERATION | MNEMONIC | FUNCTION |
|---|---|---|
| **Fixed-Point Arithmetic Operations** | | |
| Add Word | ADW | (EWA) + (R) → (R) ‡‡ |
| Add Halfword | ADH | $(EHA)_{SE}$ + (R) → (R) + |
| Add Value | ADV | $V_{SE}$ + (R) → (R) ‖ |
| Add Word to Storage | ADS | (EWA) + (R) → (EWA) ‡‡ |
| Add Halfword to Storage | ADHS | $(EHA)_{SE}$ (R) → (EHA) + |
| Subtract Word | SBW | (R) − (EWA) → (R) ‡‡ |
| Subtract Halfword | SBH | (R) − $(EHA)_{SE}$ → (R) + |
| Subtract Value | SBV | (R) − $V_{SE}$ → (R) ‡‡ |
| Subtract Word from Storage | SBS | (EWA) − (R) → (EWA) ‡‡ |
| Subtract Halfword from Storage | SBHS | $(EHA)_{SE}$ − (R) → (EHA) + |
| Difference Word | DFW | 1(R) − (EWA) 1 → (R) |
| Difference Halfword | DFH | 1(R) − $(EHA)_{SE}$ 1 → (R) |
| Difference Value | DFV | 1(R) − $V_{SE}$ 1 → (R) |
| Difference Word to Storage | DFS | 1(EWA) − (R) → (EWA) |
| Difference Halfword to Storage | DFHS | 1$(EHA)_{SE}$ − (R) 1 → (EHA) + |
| Multiply Word | MLW | (R) × (EWA) → (R,R+1) |
| Multiply Halfword | MLH | (R) × $(EHA)_{SE}$ → (R,R+1) — Low order in R+1 ‡‡ |
| Multiply Value | MLV | (R) × $V_{SE}$ → (R,R+1) |
| Divide Word | DVW | (R,R+1) ÷ (EWA) → (R,R+1) — quotient in R+1 ‡‡ |
| Divide Halfword | DVH | (R,R+1) ÷ $(EHA)_{SE}$ → (R,R+1) — remainder in R |
| Divide Value | DVV | (R,R+1) ÷ $V_{SE}$ → (R,R+1) |
| Augment Word | AGW | (EWA) +1 → (EWA) ‡‡ |
| Augment Halfword | AGH | $(EHA)_{SE}$ +1 → (EHA) + |
| Diminish Word | DMW | (EWA) −1 → (EWA) ‡‡ |
| Diminish Halfword | DMH | $(EHA)_{SE}$ −1 → (EHA) + |
| Negate Word | NGW | two's complement (EWA) → (EWA) |
| Negate Halfword | NGH | two's complement $(EHA)_{SE}$ → (EHA) + |

| OPERATION | MNEMONIC | | FUNCTION | $C_0$ | $C_1$ |
|---|---|---|---|---|---|
| **Compare Operations** * | | | | | |
| Compare Word | CMW | compare (EWA) with (R) | | | |
| Compare Halfword | CMH | compare $(EHA)_{SE}$ with (R) | | | |
| Compare Value | CMV | compare $V_{SE}$ with (R) | (EA)=(R) | 0 | 0 |
| Test Word | TSW | compare (EWA) with 0 | (EA)>(R) | 0 | 1 |
| Test Halfword | TSH | compare $(EHA)_{SE}$ with 0 | (EA)<(R) | 1 | 0 |
| Collate Word | UCM | compare (EWA) with (R) as unsigned nos. | | | |
| Collate Halfword | UCMH | compare $(EHA)_{ZE}$ with (R) as unsigned nos. | | | |
| Collate Value | UCMV | compare $V_{ZE}$ with (R) as unsigned nos. | | | |
| Compare Register with Word Interval | CIS | compare (R) with (EWA) and (EWA +1) and set condition code | (EA)≤ (R)≤(EA+1) | 0 | 0 |
| | | | (EA)≥ (R)≥(EA+1) | 0 | 0 |
| | | | (EA)< (R); (R)>(EA+1) | 0 | 1 |
| Compare Register with Halfword Interval | CISH | compare (R) with (EHA) and (EHA+1) and set condition code | (EA)> (R); (R)<(EA+1) | 1 | 0 |
| Compare Word in Storage with Interval | CIW | compare (EWA) with (R) and (R+1) and set condition | (R)≤(EA)≤(R+1) | 0 | 0 |
| | | | (R)≥(EA)≥ (R+1) | 0 | 0 |
| | | | (R)<(EA); (EA)>(R+1) | 0 | 1 |
| Compare Halfword in Storage with Interval | CIH | compare $(EHA)_{SE}$ with (R) and (R+1) and set condition | (R)>(EA); (EA)<(R+1) | 1 | 0 |

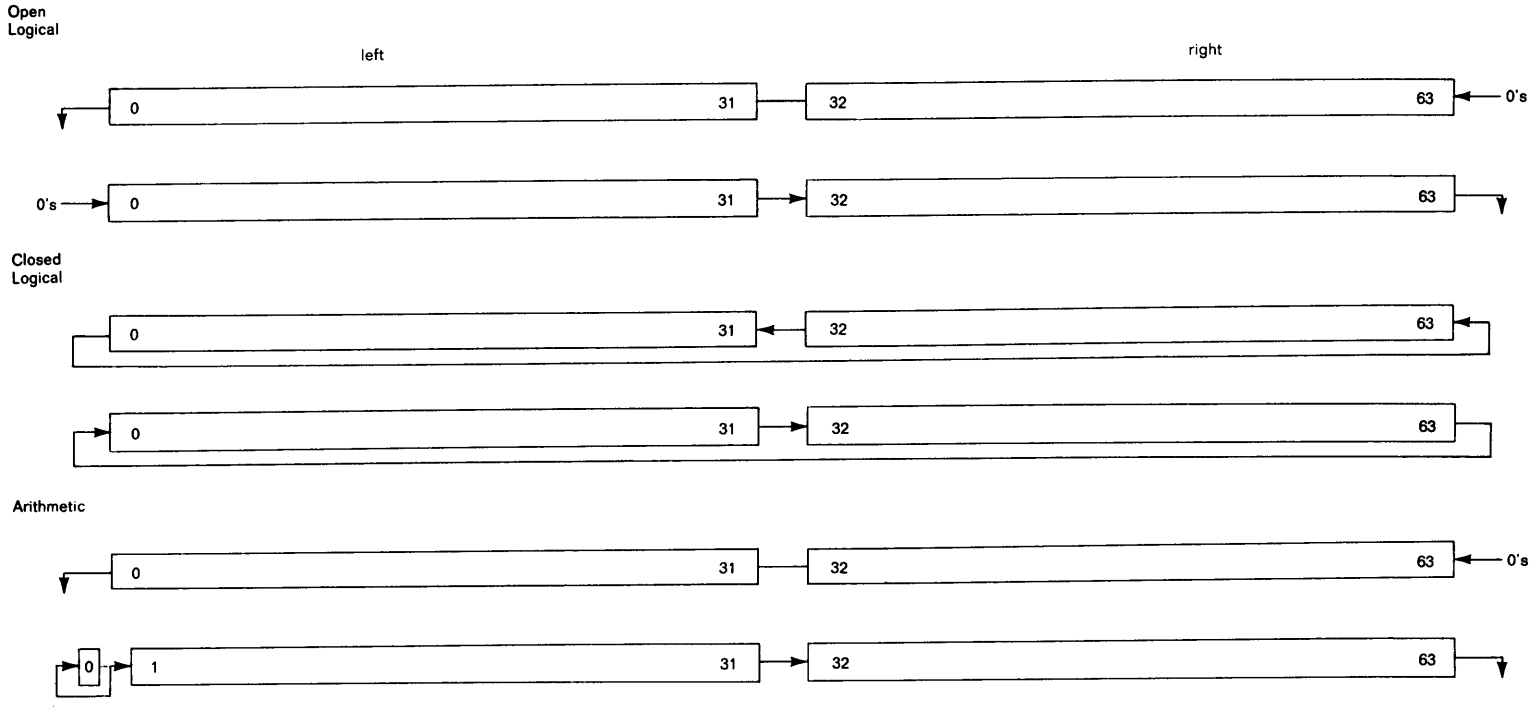
*for compare operations set conditions according to result

| OPERATION | MNEMONIC | FUNCTION |
|---|---|---|
| Shift Operations | | |
| Shift Word | SHF | Word shift performed on (R) |



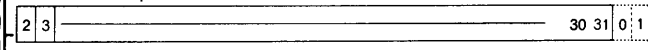

On arithmetic left shifts, if bit O changes Trap on WO.

| | | |
|---|---|---|
| Shift Doubleword | SHFD | Doubleword shifts performed on (R) and (R+1) |




On arithmetic left shifts, if bit O changes Trap on WO.

Unpack: clear the register towards which the shift is to occur to zero then perform a doubleword open logical shift.

| OPERATION | MNEMONIC | FUNCTION |
|---|---|---|
| **Bit Operations** | | |
| Clear Bit | CLB | $0 \rightarrow (EHA)_R$ |
| Fill Bit | FLB | $1 \rightarrow (EHA)_R$ |
| Invert Bit | IVB | one's complement of $(EHA)_R \rightarrow (EHA)_R$ |
| Skip on Zero | SKZ | if $(EHA)_R = 0$, $I+1 \rightarrow I$ |
| Skip on Zero and Zero Bit | SKZC | if $(EHA)_R = 0$, $I+1 \rightarrow I$; $0 \rightarrow (EHA)_R$ (regardless of outcome) |
| Skip on Zero and Fill Bit | SKZF | if $(EHA)_R = 0$, $I+1 \rightarrow I$; $1 \rightarrow (EHA)_R$ (regardless of outcome) |
| Skip on Zero and Invert Bit | SKZI | if $(EHA)_R = 0$, $I+1 \rightarrow I$; one's complement $(EHA)_R \rightarrow (EHA)_R$ (regardless of outcome) |
| Skip on Bit | SKB | if $(EHA)_R = 1$, $I+1 \rightarrow I$ |
| Skip on Bit and Zero Bit | SKBC | if $(EHA)_R = 1$, $I+1 \rightarrow I$; $0 \rightarrow (EHA)_R$ (regardless of outcome) |
| Skip on Bit and Fill Bit | SKBF | if $(EHA)_R = 1$, $I+1 \rightarrow I$; $1 \rightarrow (EHA)_R$ (regardless of outcome) |
| Skip on Bit and Invert Bit | SKBI | if $(EHA)_R = 1$, $I+1 \rightarrow I$; one's complement $(EHA)_R \rightarrow (EHA)_R$ (regardless of outcome) |
| **Jump and Execute Operations** | | |
| Jump if Equal to Zero | JEZ | if $(R) = 0$, EWA $\rightarrow I$ |
| Jump if Equal | JE | if $C_0\, C_1 = 00$, EWA $\rightarrow I$ |
| Jump if Not Equal to Zero | JNEZ | if $(R) \neq 0$, EWA $\rightarrow I$ |
| Jump if Not Equal | JNE | if $C_0\, C_1 \neq 00$, EWA $\rightarrow I$ |
| Jump if Greater than Zero | JGZ | if $(R) > 0$, EWA $\rightarrow I$ |
| Jump if Greater | JG | if $C_1 = 1$, EWA $\rightarrow I$ |
| Jump if Less than or Equal to Zero | JLEZ | if $(R) \leq 0$, EWA $\rightarrow I$ |
| Jump if Less than or Equal | JLE | if $C_1 = 0$, EWA $\rightarrow I$ |
| Jump if Less than Zero | JLZ | if $(R) < 0$, EWA $\rightarrow I$ |
| Jump if Less | JL | if $C_0 = 1$, EWA $\rightarrow I$ |
| Jump if Greater than or Equal to Zero | JGEZ | if $(R) \geq 0$, EWA $\rightarrow I$ |
| Jump if Greater than or Equal | JGE | if $C_0 = 0$ EWA $\rightarrow I$ |
| Jump if Even | JEV | if $(R)_{31} = 0$, EWA $\rightarrow I$ |
| Jump if Odd | JOD | if $(R)_{31} = 1$, EWA $\rightarrow I$ |
| Jump if Implication | JIM | if $C_0\, C_1 = 11$, EWA $\rightarrow I$ |
| Jump if No Implication | JNIM | if $C_0\, C_1 = 11$, EWA $\rightarrow I$ |
| Jump on Register Incremented by Halfword | JIH | $(R) + 1 \rightarrow (R)$; EWA $\rightarrow I$ unless $(R)_{15}$ changes |
| Jump on Register Decremented by Halfword | JDH | $(R) - 1 \rightarrow (R)$; EWA $\rightarrow I$ unless $(R)_{15}$ changes |
| Jump on Register Incremented by Word | JIW | $(R) + 2 \rightarrow (R)$; EWA $\rightarrow I$ unless $(R)_{15}$ changes |
| Jump on Register Decremented by Word | JDW | $(R) - 2 \rightarrow (R)$; EWA $\rightarrow I$ unless $(R)_{15}$ changes |
| Jump on Register Incremented by Doubleword | JID | $(R) + 4 \rightarrow (R)$; EWA $\rightarrow I$ unless $(R)_{15}$ changes |
| Jump on Register Decremented by Doubleword | JDD | $(R) - 4 \rightarrow (R)$; EWA $\rightarrow I$ unless $(R)_{15}$ changes |
| Jump on Register Incremented by Field | JIB | if $(R)_{2,3} = 0$, $(R) + 1 \rightarrow (R)$<br>if $(R)_{2,3} = 0$, $(R) + 2 \rightarrow (R)$ |
| Jump on Register Decremented by Field | JDB | if $(R)_{2,3} = 0$, $(R) - 1 \rightarrow (R)$<br>if $(R)_{2,3} = 0$, $(R) - 2 \rightarrow (R)$ |
| | | *R is interpreted as follows:*   jump unless $(R)_{30}$ changes if $(R)_{15}$ changes, $0 \rightarrow (R)_{0-31}$ |
| Link and Jump | LNJ | $I \rightarrow (R)$, EWA $\rightarrow I$ |
| Jump | JMP | EWA $\rightarrow I$ |
| Link and Execute | LNK | $I \rightarrow (R)$ then execute (EWA) |
| Execute | XEC | execute (EWA) |
| **Control Operations** | | |
| Load Program Stateword | LPS | (EWA, EWA+1) $\rightarrow$ PS1, PS2 |
| Exchange Program Stateword | SWS | (EWA, EWA+1) $\rightleftarrows$ PS1, PS2 |
| Control Process | CP | transmit EWA to Multiprocess Controller |

H632 SYSTEM INFORMATION FLOW

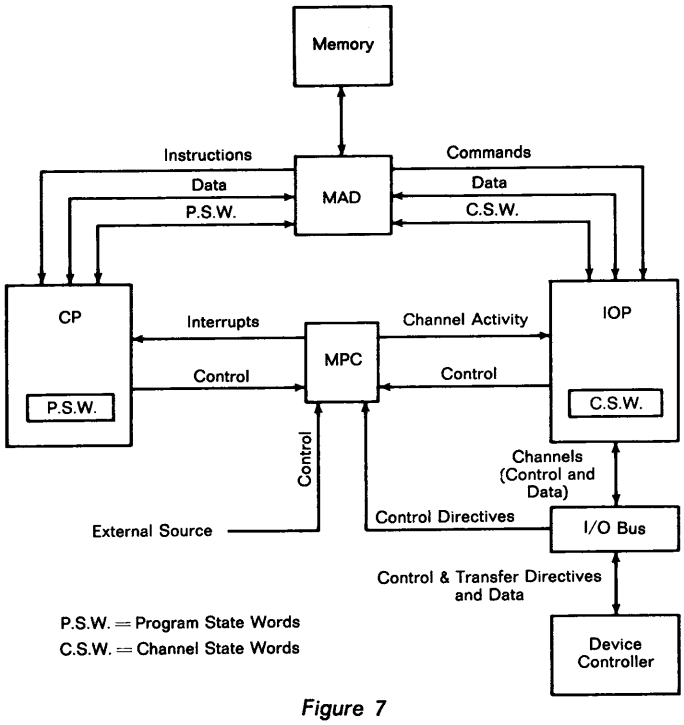



*Figure 7*

SIMPLIFIED IOP BLOCK DIAGRAM

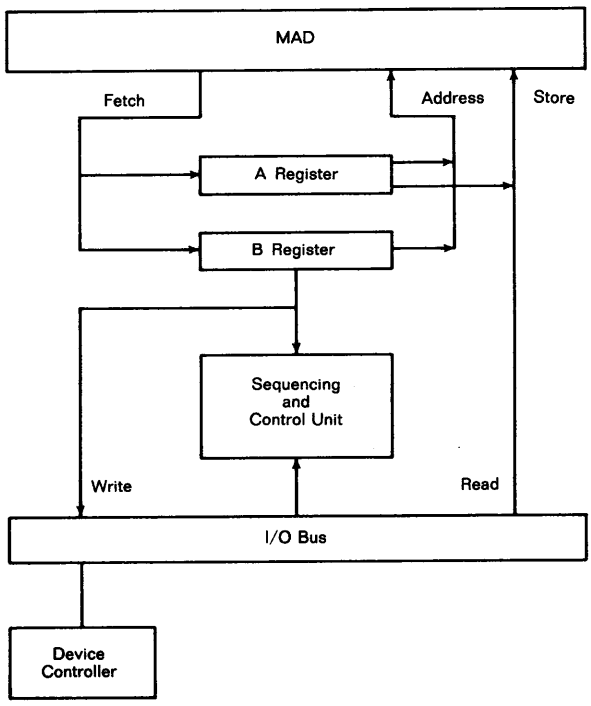



*Figure 8*

## H632 INPUT/OUTPUT PROCESSOR

**General Description** — The word-oriented H632 Input/Output Processor (IOP) has highly developed I/O facilities and limited arithmetic processing capabilities. The IOP performs all I/O operations in the basic H632 System. In the basic system, the IOP performs memory accesses, on a cycle-stealing basis, to a memory bank shared with the CP. Command fetch and execution, as well as the transfer of data between memory and a peripheral device, are accomplished without CP intervention.

**Operation** — The IOP issues memory access requests to the MAD. If memory is not being accessed by another processor, the IOP request will be acknowledged, and the contents of the addressed memory location will be presented to the IOP for processing. If the memory access was a command fetch, the various subfields of the command word are processed and operations are performed in accordance with command definition. If the memory access was a data-transfer request, a transfer is made under the direction of the controller initiating the request.

Figure 7 shows the lines of communications for instructions, commands, orders and other information in the basic H632 System. Note that the IOP and IOP I/O bus deal with information from the MPC as well as with information from memory and device controllers. The MPC issues service requests for channel program command execution on behalf of the highest priority sequencing channel. The device controllers issue control directives to the MPC and IOP to change program states and transfer device status. The IOP may also issue orders to the MPC change program state as does the CP.

**Internal Organization** — The IOP contains two 32-bit registers (A and D), a sequencing and control unit, and an I/O bus distribution system. Figure 8 shows a simplified block diagram of the IOP.

*A and D Registers*

The A and D registers are used in:

a. Command interpretation and execution
b. Output of data
c. Channel stateword modification

*Sequencing and Control Unit (SCU)* — The SCU controls the interface of the IOP with primary storage via the MAD, with the MPC, and with the various device controllers. It also retains and acts on the basic control information required for program sequencing and command execution.

Data transfers between memory and peripheral devices are accomplished over data channels. Associated with each IOP channel is a pair of Channel Statewords retained in dedicated memory locations. The standard H632 IOP provides eight channels, optionally expandable to 16 channels. Since the IOP contains independent I/O channels and corresponding channel programs, multiprogrammed/multichannel operations may be performed.

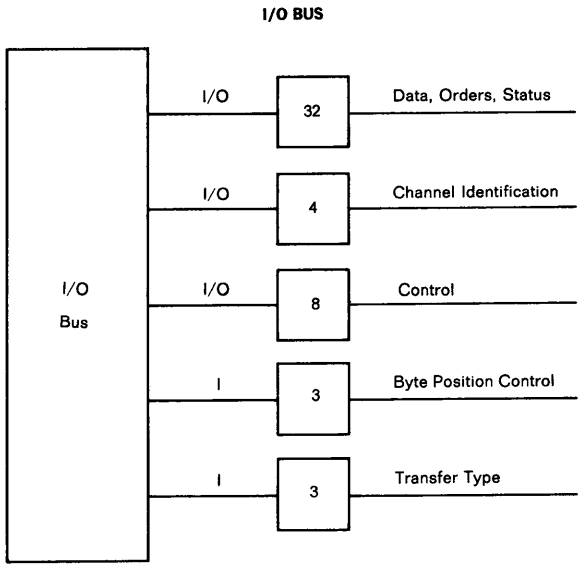
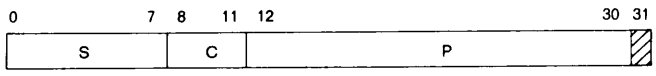



***Figure 9***

*Channel Statewords* — Each channel program is characterized by an activity state indicator retained by the MPC and by two channel statewords retained in memory.

Channel Stateword format:

CS1

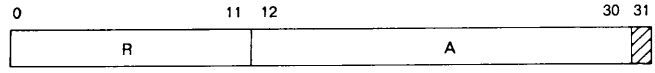



S — Status — Eight-bit field coded to represent the status of the device most recently serviced by this channel.

C — Count — Four-bit field used by the channel program to count iteration of program loops.

P — Pointer — 19-bit field which identifies the memory address of the next command in the associated channel program.

CS2




R —Range — 12-bit field used to count number of words transferred to or from memory.

A — Address — 19-bit field identifies the memory address of the next transfer to or from memory.

*I/O Bus System* — Provides the communication path for data and control between the IOP and device controllers.

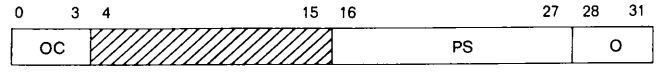
Figure 9 shows I/O bus signal lines. Device controllers are connected to this bus.

The various operations are performed as a result of controller responses to IOP commands and IOP responses to controller directives.

*Commands* — In performing I/O operations, a sequence of IOP commands, referred to as a channel program, will be executed. Commands may be stored in any contiguous area of memory. The Pointer field of CS1 points to the location of the next command to be executed in servicing the channel under consideration.

Commands are:

1. Control Process

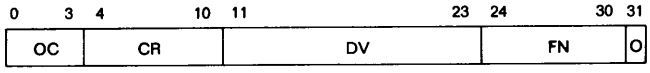



OC — Op code

PS — Process selection —

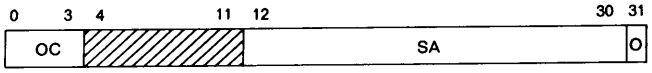
Effect of execution: The command word itself is transmitted to the IOP bus. PS is interpreted by the MPC as the process identification and the function to be performed on it.

2. Order Transmit



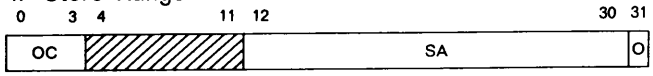

OC — Op code
CR — Controller identification
DV — Device
FN — Function to be performed
Effect of execution: The command word itself is transmitted to the IOP bus.

3. Store Status



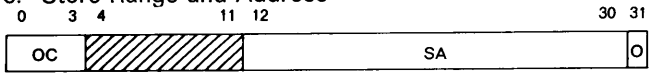

OC — Op code
SA — Storage address
Effect of execution: The status field of CS1 is stored in bits 0-7 of the word at SA. Bits 8-31 are not modified.

4. Store Range



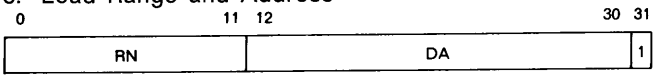

OC — Op code
SA — Storage address
Effect of execution: R of CS2 is stored in bits 0-11 of the word at SA. Bits 12-31 are not modified.

5. Store Range and Address



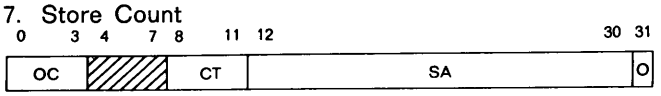

OC — Op code
SA — Storage address
Effect of execution: R and A of CS2 are stored in bits 0-30 of the word at SA; bit 31 of the word at SA is made a 1.

6. Load Range and Address



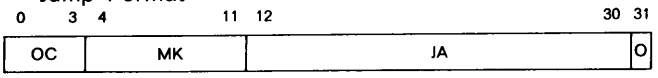

OC — Op code
RN — Range
DA — Data address
Effect of execution: The one's complement of RN replaces R, and DA replaces A, in CS2. Bit 31 of CS2 is set to 1. Note that bit 31 is a 1 in this command.

7. Store Count




OC — Op code
CT — Storage address
Effect of execution: The count field of CS1 is stored in bits 8-11 of the word at SA, leaving bits 0-7 and 12-31 unmodified. The count field in CS1 is then replaced with CT.

Jump Format




OC — Op code
MK — Mask
JA — Jump address

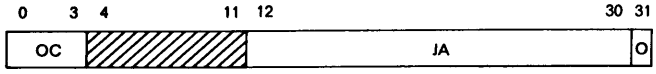
8. Jump on Status Reset
Effect of execution: JA replaces P in CS1 if the logical product of S and MK equals zero.

9. Jump on Status Set
Effect of execution: JA replaces P in CS1 if the logical product of MK and the one's complement of S equals zero.

10. Jump on Status not Reset
    Effect of execution: JA replaces P in CS1 if the logical product of S and MK is not equal to zero.

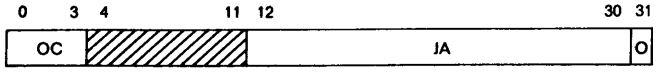
11. Jump on Status not Set
    Effect of execution: JA replaces P in CS1 if the logical product of MK and the one's complement of S is not equal to zero.

12. Jump on Count Zero




    OC — Op code
    Effect of execution: Increment the count field of CS1; if the count field is then equal to zero, JA replaces P in CS1.

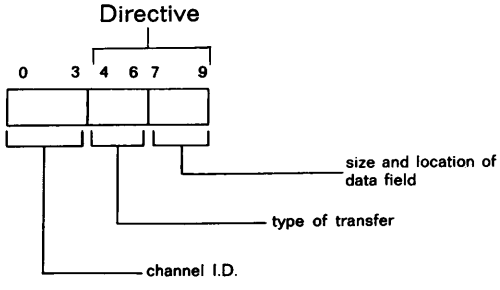
13. Jump on Count Non-Zero




    OC — Op code
    Effect of execution: Increment the count field of CS1; if the count field is then not equal to zero, JA replaces P in CS1.

Service Requests — In addition to executing commands, the IOP responds to service requests generated by device controllers and the MPC. Upon IOP receipt and acknowledgement of the service request, the originator of the request furnishes information containing a channel number and a directive. The IOP then performs the operation encoded in the directive for the identified channel.

Service request format




There are two types of directives—control and transfer:
*Control Directives* — This type of directive is related to the state of channels, the status of controllers, and command execution rather than to data transfers.

1. Execute Command
   P in CS1 of the identified channel is used as an address to fetch a command from memory; P is then incremented. The command fetched is executed as specified above. Should the command that was fetched not be executable (i.e., an order transmission command not accepted by a controller), P will be decremented, resulting in a value equal to the initial value. Both the MPC and device controller may issue this directive.

2. Change State
   This directive is ignored by the IOP, and is responded to by the MPC. A typical device controller would cause the assigned channel and all waiting channels to become sequencing when the controller is available for reassignment. This directive is issued by a device controller.

3. Change Status
   Status information supplied by the initiating controller is transferred to S in the CS1 of the identified channel. This directive could be issued by a controller that has detected some error condition that has occurred in the device it is serving. Note that the channel identification is supplied with this directive. This implies that controllers must retain the assigned channel number during engagement. This directive is issued by a device controller.

4. Change State and Status
   This directive combines the two directives listed above. A typical controller would generate this directive upon completion of an I/O transfer. Thus, the status of the device would be transmitted to the channel stateword and the state of the channel(s) could be altered.

*Transfer Directives* — Transfer directives, issued by device controllers, are concerned with data transfers. Encoded within the directive are the identification of the channel over which the transfer is to take place and the type of transfer to be performed. The directive also specifies the size and location within a computer word of the data field to be transferred. Eight-bit byte, 16-bit halfword, and 32-bit fullword transfers may be performed. The IOP uses fields A (data address) and R (range) from the identified channels CS2, in servicing requests for data transfer.

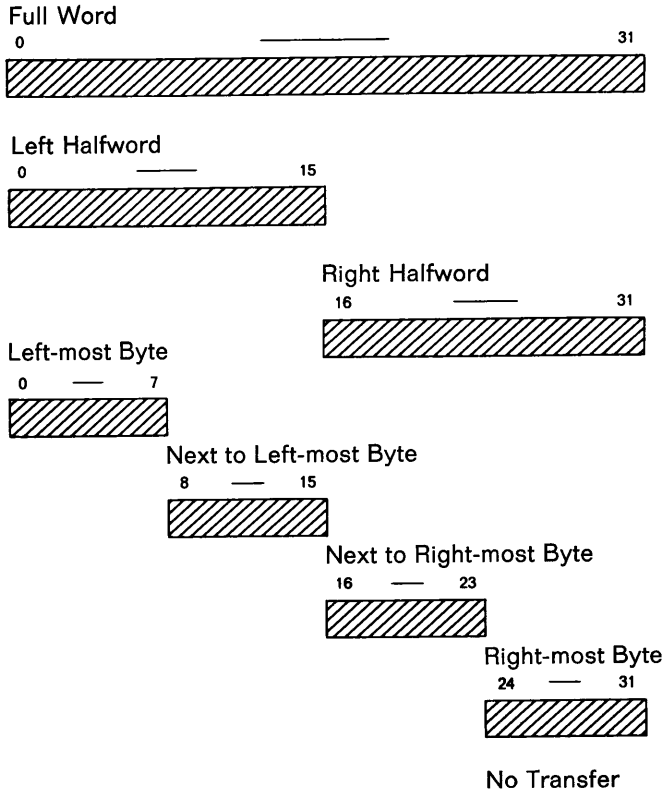
Transfer directives are:

1. Input
   Data supplied by the controller replaces the selected field in the word pointed to by A.

2. Output
   Data from the selected field of the word pointed to by A is transferred to the controller.
3. Input and Count Down
   Data supplied by the controller replaces the selected field in the word pointed to by A. A is then decremented by one and R is incremented by one. If R then contains all ones, an Execute Command operation is automatically performed for the identified channel.
4. Input and Count Up
   Data supplied by the controller replaces the selected field in the word pointed to by A. A and R are then incremented by one. If R then contains all ones, an Execute Command operation is automatically performed for the identified channel .
5. Output and Count Up
   Data from the selected field of the word pointed to by A, is transferred to the controller. A and R are then incremented by one. If R then contains all ones, An Execute Command operation is automatically performed for the identified channel.

As stated above the size and location of the data field to be transferred is specified in the transfer directive.

The IOP can perform transfers on these types of data:




*Channel Activity States* — There are four possible states that an IOP data channel may assume.

1. Engaged — In the engaged state, the channel is as-assigned to, and under the direction of, a specific device controller. Engagement can occur when an Order Transmit command is executed by the IOP and the order issued by the IOP is accepted by the controller to which it was directed. When engagement occurs, the device controller assumes the responsibility for issuing directives to effect data transfers and request subsequent command execution for the corresponding channel program.
2. Sequencing — In the sequencing state, the channel is not assigned to a controller and the corresponding channel program requires command execution. The MPC issues the service request for command execution on behalf of the highest priority sequencing channel.
3. Waiting — In the waiting state, a channel requires engagement to a controller that is temporarily busy. Transition to this state may occur when an Order Transmit is not accepted by a busy controller.
4. Stopped — A channel is in the stopped state if it is in none of the above states.

*Modes of Operation* — By issuing an MPC directed order, Control Process, control over the mode of IOP operations is achieved.

Selector — The MPC may direct all but one channel to assume the waiting or stopped state. The remaining channel and assigned controller can then operate in a selector mode.

Multiplexer — The MPC and controller may direct all channels to assume the sequencing state. Data transfers would then take place on a priority basis in a multiplexed mode.

## MULTIPROCESS CONTROLLER (MPC)

**General Description** — The Multiprocess Controller (MPC) serves as a clearing house for all interprocess and intraprocess state control requests in the H632 System. The MPC has the coordinating responsibility for the processes (programs) operating in the processors it administers.

**Operation** — Control of the states of the various processes (CP Program Levels and IOP Channel Activity) is accomplished through orders issued to the MPC either by IOP command, CP instructions or by signals received from device controllers. An external interrupt source may also be used to initiate/modify CP activity. (See Figure 10.)

*CP Program Levels* — can be in one of the following states:

Stopped — disabled

Enabled — conditioned to respond to an interrupt

Sequencing — program running if highest priority; pending running, if not

Committed — sequencing with a backed-up interrupt request

CP program states can be modified by any and all processors (CP and IOP) in the system, and optionally by some external source. The H632 CP can have a maximum of 40 program levels.

*IOP Program Levels* (channel programs) — can be in one of the following states:

Stopped — none of the below

Waiting — for a busy controller to become available

Sequencing — program running if highest priority, pending running, if not

Engaged — under the control of an I/O device controller

The standard IOP is capable of executing up to eight channel programs concurrently (optionally expandable to 16).

IOP program states can be modified by any and all processors (CP and IOP) in the system as well as by device controllers connected to the IOP.

**SIMPLIFIED MULTIPROCESS CONTROLLER (MPC) BLOCK DIAGRAM**

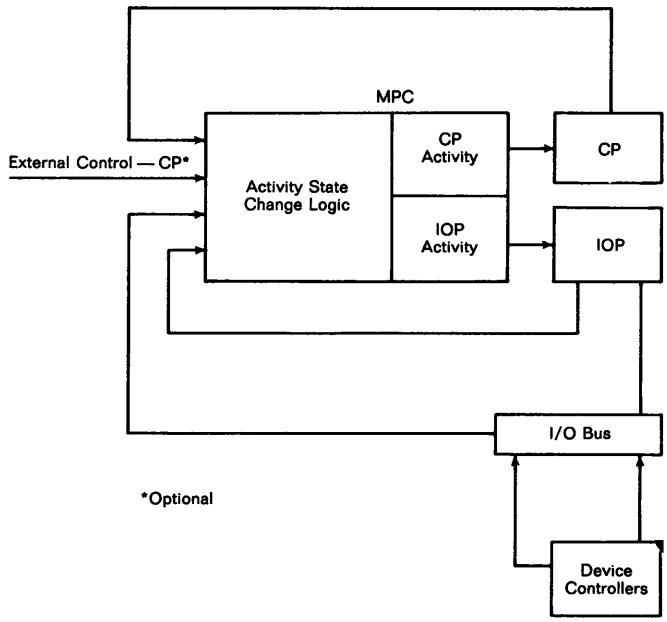
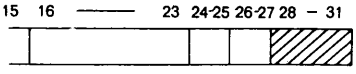



*Figure 10*

*Interprocess and Intraprocess Level Control* — Both the CP and the IOP can execute a Control Process instruction (command), the net effect of which is the transmittal of a 12-bit number to the MPC. This number is interpreted by the MPC as follows:




16-23 — identifies a target processor and process level (or channel) whose activity state is to be affected.

24 and 25 — *is an interprocess order* which defines the change to be made to the target process.

26 and 27 — *is an intraprocess order* which defines a change to be made to the activity state of the process which transmitted the order.

The MPC is responsible for determining the priorities of service requests and for furnishing each CP and IOP with a code identifying the sequencing process of highest priority.

The MPC also maintains indicators of the state of CP processes and IOP channel activity. These indicators can be displayed on the system control panel.

## PRINTER/KEYBOARD

**Introduction** — The character printer/keyboard provides the basic communication link between the operator and the H632 System. This feature consists of a KSR-35 character printer/keyboard and controller which can send and receive information in USASCII code at a maximum rate of ten characters per second.

The controller provides the necessary interface logic between the IOP and the KSR-35. It controls the sequencing of the KSR-35 and the data flow and format through the use of orders received from the IOP.

**Operation** — The KSR-35 controller executes these orders issued to it by the IOP:

Input from keyboard, packing 1 byte/word
Input from keyboard, packing 4 bytes/word
Output to printer, unpacking 1 byte/word
Output to printer, unpacking 4 bytes/word
Stop
Output without printing until stop

An output order allows the transfer of information directly from memory to the printer. The controller will continue to request service from the IOP until a stop order is received. An input order to the controller may be used to transfer information from the keyboard directly to memory. This order may be terminated by the depression of the carriage return key or by a stop order. In addition to directing the flow of information, the input and output orders specify character packing and unpacking (refer to Figure 11).

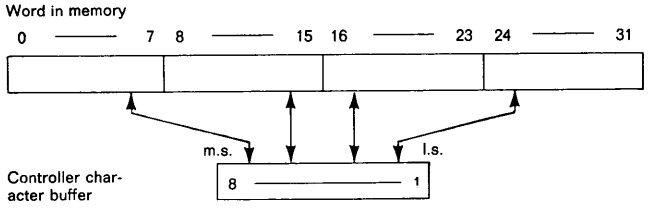



*Figure 11*

The stop order is used to terminate output activity and may be used to terminate input activity. The output without printing order is useful as a 100 ms timing interval for carriage control functions (i.e., carriage return, vertical and horizontal tab functions).

The KSR-35 controller maintains status codes to indicate these conditions to the IOP:

1. Device presence
2. Incorrect keyboard parity
3. IOP response failure within 13.6 ms of request
4. Premature termination to indicate that a stop order was used to terminate an input order prior to detection of a carriage return code
5. Keyboard interrupt
6. Position of last byte transferred
7. Normal device

When the printer or keyboard is operating at maximum speed, the controller will request service from the IOP every 100 ms. To maintain a printing rate of ten characters per second, a request for service initiated by the keyboard must be honored by the IOP within 13.6 ms. Failure to honor this request will result in loss of data. In the event that data is lost, an error status will be sent to the IOP at the end of the current order.

When the KSR-35 controller is non-busy (not assigned to a channel), hard copy will not be produced when a key is depressed. When the controller is in the output mode, hard copy will not be affected when a key is depressed. In either case, the keyboard interrupt status code will be set. The IOP will be notified of this when the next status transfer is made.

*Controller* — One character printer will be provided per controller. Mode of operation will be on-line only. Controller-to-IOP interface is half duplex; controller-to-device interface is full duplex. The controller is located within the basic main frame cabinet.

## 632-0001 SYSTEM SPECIFICATIONS SUMMARY

### CENTRAL PROCESSOR

| | |
|---|---|
| Type: | Parallel, binary, solid-state |
| Addressing: | Single address (modified), with multilevel indexing and indirect addressing |
| Numeric Representation: | Two's complement |
| Circuitry: | Integrated |
| Typical Execution Times: (including instruction and operand fetch with indexing) | Add — 1.7 $\mu$s<br>Subtract — 1.7 $\mu$s<br>Multiply — 8.1 $\mu$s<br>Divide — 28.05 $\mu$s |
| Instruction Complement: | 118 standard instructions<br>26 optional instructions |
| Word Length: | 32 bits |

### INPUT/OUTPUT PROCESSOR

| | |
|---|---|
| Word Size: | 32 bits |
| Throughput Rate: | 200,000 words per second, maximum |
| Commands: | 13 |
| Transfer Types: | 8 |
| Concurrent Operations: | 8 expandable to 16 |
| Maximum Number of Controllers: | 16 (including KSR-35) |

### MULTIPROCESS CONTROLLER

| | |
|---|---|
| Process Level Control: | 8 standard CP program levels<br>8 standard IOP program levels |

### PRIMARY STORAGE

| | |
|---|---|
| Capacity: | 8,192 words expandable to 131,072 words |
| Word Length: | 32 bits |
| Type: | Coincident-current random-access ferrite core |
| Cycle Time: | 850 ns |

### MEMORY ACCESS DIRECTOR

| | |
|---|---|
| Type: | 1 x 1 |
| Processor Ports: | One, shareable by one CP and one IOP |
| Memory Ports: | One, shareable with up to 16-8,192-word memory modules |

### PACKAGING

The 632-0001 System will be housed in a Main Frame Cabinet with these specifications:

| | |
|---|---|
| Dimensions: | 74″ high, 32″ wide, 30″ deep |
| Weight: | 1200 lbs. |

### ENVIRONMENT

| | |
|---|---|
| Room ambient for H632 System less I/O devices: Forced filtered air cooling | 0°C to 45°C |

### LOGIC LEVELS

| | |
|---|---|
| Logical one: | +6 volts |
| Logical zero: | 0 volts |
| Primary Power: | 30 amp, 220 volt, single phase, 50 or 60Hz; four-wire service outlet will be required for each rack of Series 32 equipment |

### KSR-35

| | |
|---|---|
| Printer Speed: | Ten characters per sec., max. |
| Keyboard Speed: | Ten characters per sec., max. |
| Character Set: | Standard KSR-35 character set of 63 characters |
| Character Code: | Standard USASCII Code |
| Paper Size: | 8½-inch wide, continuous fold business forms. Form length of 11 inches sprocket feed. |
| Horizontal Spacing: | Ten characters/inch<br>75 characters/line |
| Vertical Spacing: | Three or six lines/inch |

Specifications subject to change without notice.

# 632-0001

## HOW TO ORDER

To order the Basic H632 Computer System, specify model No. 632-0001. The 632-0001 system includes the following:

Hardware:

1 Central Processor (eight program levels)
1 Input/Output Processor (eight I/O Channels)
1 8,192-Word Memory Module
1 Memory Access Director (1x1)
1 Multiprocess Controller
1 Keyboard/Printer (KSR-35) and Controller

| | |
|---|---|
| Environment | Room ambient for H632 system less I/O devices: 0°C to 45°C |
| Cooling | Filtered forced air |
| Dimensions | 74" high, 32" wide, 30" deep |
| Weight | Main Frame — 1200 lbs. |
| Primary Power | 30 amp, 220 volt, single-phase, 50 or 60 Hz; four-wire service outlet will be required for each rack of Series 32 equipment |

Documentation, Software and Training — furnished without extra charge with the 632-0001:

## SOFTWARE

**General** — The software package for the H632 will have these properties:

All programs are completely specified by documents under standard engineering control procedures.

Program listings will be annotated in such a way that an English-language description of the logical flow of the program is provided.

Large programs ( Assembler, FORTRAN ) will be divided into small logical modules. The function of each logical module will be carefully described by comments in order to facilitate understanding program flow.

**Programs** — Each complete program in the software package consists of:

*Program Listings* — A product of the assembly process, and a graphic representation of the Source Program. Program Listings also depict the object code generated by the assembly process.

*Object Program* — The end product of the assembly process; represents the program in machine intelligible form.

*Program Documentation* — Complex programs such as FORTRAN require documentation in addition to the program listing. This documentation is usually in the form of a manual describing methods of operating the program. The program listing, however, is the basic vehicle for software documentation and will be all that is provided unless otherwise specified in the following list of programs.

**Program Listings and Software** — Program Listings will be published documents. The object programs will be supplied on a medium applicable to the first configuration to be shipped; for example: A customer having a system using a CCD-supplied Card Reader would get object programs on either punched cards or on punched paper tape but not both, whereas a customer having a minimum paper tape oriented machine configuration would get object programs on punched paper tape only.

**Software Package** — The contents of the H632 software package are:

OS-1
Loader
Assembler
Fortran IV Compiler
Unit Record Control
KSR-35 Driver
Math Routines
  Floating Point
  Fixed Point
Trap Package
Test and Maintenance Routines
Media Conversion
System Editor

## TRAINING

**Training Services** — Training support of the H632 is designed for maximum utilization of computer capabilities and for minimization of potential machine down time. Standard courses in Programming/Software and Logic/Maintenance are conducted at the Honeywell Computer Control Division Education Center, Framingham, Massachusetts.

**Programming/Software Training (Standard)** — A comprehensive two-week training course oriented to experienced programmers. Designed to review assembly language, input/output and real-time programming.

The effective use of Computer Control Division software package is stressed through student hands-on-computer experience in solving an applicable problem. Persons attending this course should have at least two years' assembly language programming experience. Computer Control Division can offer special computer programming fundamentals courses at Framingham or a customer specified site. A minimum of five students per course will be maintained.

**Logic/Maintenance Training** — A seven-week comprehensive logic/maintenace course conducted for experienced digital computer maintenance personnel. The first week includes an introduction to $\mu$-**PAC** and NAND logic, computer organization and operation as well as a general discussion of all the computer instructions. The standard input/output devices, basic memory cycle, and various methods of documentation are also covered. The remainder of the course is devoted to detailed logical analysis of computer operation and applicable options. Maintenance and troubleshooting techniques are reviewed, including machine time for troubleshooting and familiarization.

It is strongly recommended that those who will have direct maintenance responsibility attend the full course. Prerequisites for meaningful attendance at the standard course are at least two years of digital computer experience. Special computer indoctrination and pre-training courses can be offered at Framingham or a customer specified site. A minimum of five students in a course will be maintained.

**Free Registrations and Chargeable Training** — H632 customers of Computer Control Division will receive four free registrations at a regularly scheduled standard course in Programming/Software and two free registrations at a regularly scheduled standard course in Logic/Maintenace. In addition to training provided on a no-charge basis, Computer Control Division can provide instruction on a chargeable basis for additional personnel. Instruction will be at regularly scheduled standard courses held at Framingham, Massachusetts. Charges will not exceed $250.00 per week per student for either Programming/Software or Logic/Maintenace courses.

**On-Site Training: Chargeable** — Programming and Maintenance training courses can also be provided at the customer's site. The charges for on-site courses are $201.00 per day plus round trip travel to the customer's site billed at cost, and appropriate charges for manuals utilized.

| **DOCUMENTATION** | **Qty.** |
|---|---|
| Hardware | |
| Reference Manual | 1 |
| Installation Manual | 1 |
| Interface Manual | 1 |
| Operator's Manual | 1 |
| Software | |
| Fortran Manual | 1 |
| Assembler Manual | 1 |
| User's Guide Manual | 1 |
| Programmer's Reference Card | 10 |
| Maintenance | |
| CP Maintenance Manual Volume I | 1 |
| CP Maintenance Manual Volume II | 1 |
| CP Maintenance Manual Volume III | 1 |
| CP Maintenance Manual Volume IV | 1 |
| IOP Maintenance Manual Volume I | 1 |
| IOP Maintenance Manual Volume II | 1 |
| IOP Maintenance Manual Volume III | 1 |
| IOP Maintenance Manual Volume IV | 1 |
| Wire lists | 1 |
| Illustrated Parts Breakdown | 1 |
| Modular Products Manual | 1 |
| Program Maintenace Manual (listing) | 1 |

Additional Documentation

Additional documentation, drawings and wiring diagrams can be made available on an extra charge basis.

Options

One copy each of applicable wire lists and instruction manual will be furnished for each option (internal or peripheral) delivered.

# PRICE SCHEDULE

Effective April 9, 1968, subject to change without notice

| Model Number | Item | | |
|---|---|---|---|
| 632-0001 | H632 system including central processor with eight program levels (interrupts), I/O processor with eight channels, 8,192 word memory module, memory access director, multiprocess controller, and KSR-35 I/O typer | $ 97,000 | $ 2,700 |
| 632-0500 | Power failure interrupt | 1,000 | 30 |
| 632-0510 | Real time clock | 750 | 20 |
| 632-0520 | Watchdog timer | 1,000 | 30 |
| 632-1101 | 8,192 word memory module | 35,000 | 975 |
| 632-1105 | Parity for an 8,192 word memory module (not field expandable) | 2,400 | 70 |
| 632-2101 | Floating point hardware | 8,000 | 225 |
| 632-2105 | Mask & byte instructions | 2,000 | 60 |
| 632-2120 | External control of eight standard CP program levels (interrupts) | 1,500 | 40 |
| 632-2125 | Four additional CP program levels (interrupts) with external control | 1,500 | 40 |
| 632-2200 | Address halt feature for systems control panel | 500 | 15 |
| 632-3105 | Eight additional I/O processor channels | 4,000 | 110 |
| 632-3150 | Direct read/write | 2,500 | 70 |
| 632-3160 | Parallel input channel | 1,100 | 30 |
| 632-3165 | Parallel output channel | 1,100 | 30 |
| 632-3170 | Output control pulse group — six lines | 750 | 20 |
| 632-3171 | Six additional OCP lines — (maximum number of lines per group — 30) | 500 | 15 |
| 632-3180 | Sense line group — 16 lines | 1,000 | 30 |

## STANDARD SERIES 32 PERIPHERAL PRICE SCHEDULE

| | | | |
|---|---|---|---|
| 32-4100 | Magnetic tape control unit, seven level, controls up to four transports of similar speed and density | 12,000 | 335 |
| 32-4130 | 36 ips, Magnetic tape transport, 200/556 bpi | 13,225 | 370 |
| 32-4131 | 36 ips, Magnetic tape transport, 200/800 bpi | 16,900 | 470 |
| 32-4132 | 36 ips, Magnetic tape transport, 556/800 bpi | 16,900 | 470 |
| 32-4140 | 80 ips, Magnetic tape transport, 200/556 bpi | 21,150 | 590 |
| 32-4141 | 80 ips, Magnetic tape transport, 200/800 bpi | 25,300 | 700 |
| 32-4142 | 80 ips, Magnetic tape transport, 556/800 bpi | 25,300 | 700 |
| 32-4600 | Controller for High Capacity Disc Store for up to eight disc storage units | 14,400 | 400 |
| 32-4621 | Disc storage unit for 1.8 million words | 24,600 | 685 |
| 32-4622 | Additional disc pack for disc storage unit | 965 | 30 |
| 32-5000 | Paper tape reader (300 cps) & paper tape punch (110 cps) | 9,000 | 250 |
| 32-5025 | Low speed paper tape feature (not field expandable) | 2,000 | 60 |
| 32-5100 | Card reader, 400 cpm | 14,500 | 405 |
| 32-5150 | Card reader, 800 cpm | 21,000 | 585 |
| 32-5200 | Card punch, 100-400 cpm | 27,500 | 765 |
| 32-7050 | Line printer, 300 lpm | 25,000 | 695 |
| 32-9001 | Table, custom option (requires special quotation) | --- | --- |
| 32-9003 | Color option for KSR-35 (requires special quotation) | --- | --- |

* Based on one year lease. Four year lease terms also available.

**Honeywell**

COMPUTER CONTROL DIVISION