EDINBURGH UNIVERSITY COMPUTER SCIENCE DEPARTMENT

EDINBURGH MULTI-ACCESS PROJECT

This document is HIGHLY CONFIDENTIAL, and circulation is restricted to the Technical Committee and Project members only. It is a working paper written solely for the purpose of attract-:ing comment and criticism from within the Project, and does not necessarily represent the views of the Project as a whole.

HARDWARE SUMMARY

D. J. Rees.

HIGHLY CONFIDENTIAL

CONTENTS.

4-75 CENTRAL PROCESSOR

Introduction

The 4-75 is fully described in PS4.10.70 and numbering in parts A to D of this document tallies with that used in the Product Specification. This document purports to describe the programmer interface with the hardware, however some operational aspects are included.

The main features of the 4-75 are as follows:

1.4.2 The Central Control Unit, which obeys instructions stored in the Main Core Store and control I/O operations. This is described in Section A.

1.4.3 The Scratchpad. This is a 72 word high speed (300 ns cycle time) store and is used by the Central Control Unit, Peripheral Control, and it can be addressed by program.

1.4.4 Processor States. The Central Control Unit can operate in one of 4 states in each of which the behaviour and scope of the Control Unit may be different from when operating in the other states.

1.4.5 Interrupt System. It is possible to cause a Processor State to be interrupted and another started whenever one of a set of machine conditions occurs, or whenever an external signal is received (e.g. from a peripheral device). Section B deals with the Interrupt System.

1.4.6 Main Store. This is a core-store of cycle time 1 microsecond. The unit of access is one 32 bit word or 4 bytes. Its size can be 65,536 bytes or any multiple of this up to 1, 048, 576 bytes. It is 2-way interleaved. (Section C.)

SECTION A: THE CENTRAL CONTROL UNIT

2.1 The Central Control Unit Registers

The 4-75 is in fact a multiprocessor machine, the processors being the Central Control Unit and from 1 to 15 Channel Control Units which control the peripherals. The Central Control Unit has control over the Channel Control Units.

The Central Control Unit has a set of registers built into it which are not directly accessible. They are set on a change of Processor State, a description of Processor States follows.

2.3.1 The Processor may operate in one of four Processor States numbered 1 to 4. These are:

State 1 Program State. User programs operate in this state.

State 2 Interrupt Response State. Most of supervisor is in this state.

State 3 Interrupt Control State. When Control leaves the Program state this state is entered. The part of supervisor which establishes why the supervisor has been entered is in this state.

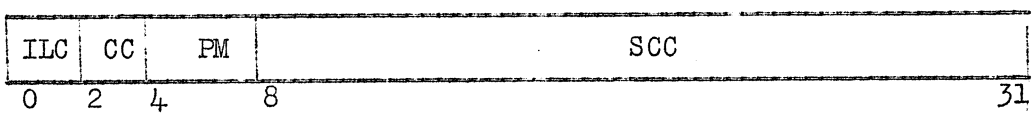
State 4 Machine Fault Control State.

Note

The above describes the software use of the four states. The hardware properties of the states will be described subsequently. From a hardware point of view, the properties of states 1 and 2 are identical.

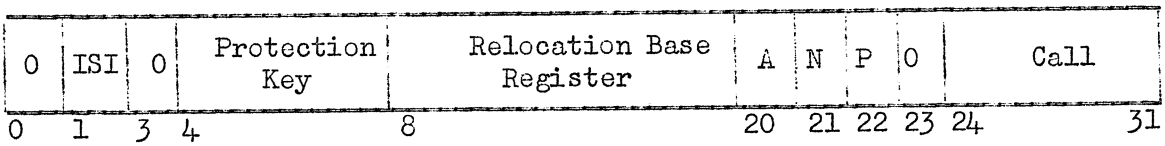
2.4.1 Associated with each Processor State there are two words in the Scratchpad, which are used by the hardware to initialise and store the Central Control registers. These two words are:-

The Program Counter (PC) whose format is:-



The/

The Interrupt Status Register (ISR) :-



The Central Control Unit has a register corresponding to each of the fields in the above two words, except the Call field. On change of program state (from Old to New), the Central Control register ILC, CC, PM, SCC * are stored in the Program Counter of the old state and the register CC, PM, SCC, Pro-:tection Key, Relocation Base Register, A, N, P are initialized from the Program Counter and Interrupt Status Register of the new state. In addition the ISI is stored in the Interrupt Status Register of the New State. Action on change of Program State is shown in diagrammatic form on
The Central Control registers and corresponding fields in the PC and ISR are described below.

2.5.2.1 I.S.I. Interrupt State Identifier. This register determines what Processor state the Control Unit is operating in. The representations of states 1, 2, 3, 4 is the ISI field of the ISR are respectively (in binary) 11, 10, 01, 00.

2.5.3 S.C.C. Sequence Control Counter. This is a 24 bit register and is the core byte address of the next instruction to be obeyed by the Control Unit. It is automatically updated by the Control unit between instructions.

2.5.4 Relocation Base Register. The relocation facility is not used in the 4-75 software since the Paging facility described later makes it redundant. This register should always be set to zero.

---

* If the change of state is caused by a Program Control instruction, the SCC is not stored. Instead the instruction specifies what is to be stored in the SCC field of the Program Counter of the Old state.

2.5.5/

2.5.5 I.L.C. Instruction Length Code. This register is 2 bits long and in it is stored the length of the current instruction, whether 2, 4 or 6 bytes long. When this is stored in the corresponding field of the ISR, the length is specified as the number of 2 byte units, i.e. the number 1, 2, or 3 is stored.

2.5.6 Protection Key. Every block of 128 words of Main Store has a 4 bit Reservation Key. The Protection Key is a 4 bit Central Control Unit register which, whenever an attempt is made by the Control Unit to access the store, is compared with the Reservation Key of the block being accessed. If the two 4 bit registers are equal, or if one is zero then the access is permitted to proceed.

2.5.7 C.C. Condition Code. This is a 2 bit register, which is set by certain instructions. The contents of the register can be tested by 'branch on condition' instructions.

2.5.8 Mode Indicator (A.N.P.)

A. Data Code Indicator.

If this bit is zero the internal code is EBCDIC, if 1, then ASCII.

N. Privilege Mode Indicator.

When set to 1, privileged instructions are inhibited, otherwise all instructions can be obeyed.

P. Paging Indicator.

When set to 1 and the processor is in state 1 or 2, then Main Store accesses are done through the segmentation-Paging mechanism, otherwise accesses are done direct. See section C for a description of Paging.

2.5.9 P.M. Program Mask Register.

This register is concerned with the Interrupt System, which is described in Section B. The register is 4 bits long, each bit corresponding to an arithmetic fault (Significance error, Exponent underflow, Decimal overflow, Fixed/

Fixed point overflow), and if a bit is set when the corresponding fault occurs a reason for interrupt is set up. The description of what happens when a reason for interrupt occurs is in section B and a logical flowchart will be found at the end of section B.

Instructions:- Load Scratchpad, Store Scratchpad (both privileged), allow transfer of words between the Scratchpad and Main Store, thus allowing any of the Program Counters or Interrupt Status Registers to be set by Program.

Instructions to cause change of program state are Supervisor Call (allows change to State 3 only) and Program Control with which one can change to any state. (This last is privileged)

In general the central control registers are not accessible directly, they are set on change of Processor State. The Program Mask Register and Condition Code Register, however, can be set directly by means of the Set Program Mask instruction.

2.5.10. The Timer.

This consists of a Clock register of length 32 bits and a Time Interrupt Register (8 bits). The Clock Register is incremented by 1 every $6\frac{1}{2}$ micro-:seconds and cannot be set by program, it is read - only. The Time Interrupt Register is set by program, and its value is unchanged otherwise. When bits 16 to 23 of the Clock Register (3rd byte) contain the same pattern as the Time Interrupt Register, the Clock reason for interrupt is set. (see section B for a description of the interrupt system.)

The clock register appears as though it is in the main store at addresses 80 to 83*. It is extracted by means of a Load Register instruction using core address/

*In fact both the Clock and Time Interrupt Registers are special registers separate from core. The fact that byte address 82 is both the Time In-:terrupt Register and part of the Clock Register does not cause hardware confusion. On a write this address refers to the Time Interrupt Register and on a read it refers to the third byte of the Clock Register.

address 80. However, it must not be accessed using Store Character instruc-:tions since parity will not be checked. The Time Interrupt Register appears to be at Main Store address 82 and can be accessed using any instruction..

SECTION B: THE INTERRUPT CONTROL SYSTEM

5. Introduction. Processor State can be changed in two ways.

1. Under program control (by means of the SVC or PC instructions)
2. Whenever one of a set of Processor or Peripheral conditions occurs (a so-called Interrupt Condition). A change of Processor State because of 2 is called an Interrupt. Interrupts can only cause a change of state to states 3 and 4 and the existence of an interrupt condition does not necessarily cause an interrupt. The set of Interrupt conditions that at any time will cause an interrupt to occur is under program control.

5.3.3 The Interrupt Flag Register (IFR). This is a 32 bit register occupying a word of the scratchpad. There are 32 possible interrupt conditions, and there corresponds a bit in the IFR for each of these. See the table on this page for a list of interrupt conditions and the corresponding bit in the IFR. If an interrupt condition exists, then the corresponding bit is set to 1, otherwise it is zero. It is cleared when that interrupt condition causes an interrupt.

5.3.3.3 Interrupt Conditions and positions of the corresponding flag bits.

Interrupt Condition	Bit position of flag in IFR
Power Failure	31
Machine Check	30
External Signal 0-5 or Channel No. 0-5	29 - 24
Channel 6-15	23 - 14
Clock	13
Console Interrupt Request	12
Supervisor Call Instruction	11
Privileged Operation	10
Op-Code Trap	9
Address Error	8
Data Error	7
Exponent Overflow	6
Divide Error	5

Significance/

Interrupt Condition	Bit position of flag in IFR
Significance Error	4
Exponent Underflow	3
Decimal Overflow	2
Fixed Point Overflow	1
Debug Mode	0

Notes

1. These reasons for interrupt are arranged in priority order, those with highest priority coming first.
2. An interrupt causes a change of Processor State to State 3, unless the interrupt is one of the first two (power failure or Machine check) when State 4 is initiated.
3. The Weight number of an interrupt (See Weight Registers, 5.3.6 below) is $4(31-b)$ where b is the bit position of the bit in the RFI representing the reason for interrupt.
4. Any of the interrupts can be inhibited in any Processor State by means of the Interrupt Mask Register of the Processor State (see section 5.3.4), and in addition the four interrupts corresponding to bits 4 to 1 of the IFR can be cancelled using the Program Mask Register, a 4 bit Central Control Register (see section 5.3.5). This last register can be set using a non-privileged instruction, Set Program Mask.
5. Each of the bits 29 to 24 is set if either the corresponding channel or the corresponding external device (e.g. another computer) causes an interrupt.

5.3.4 Interrupt Mask Register. There is one of these per program state. It is a 32-bit register in the Scratchpad and the bits correspond to the 32 reasons for interrupt in the same way as for the IFR. If a bit is zero and the corresponding reason for interrupt gets set in the IFR, then no interrupt is executed (if however it is an interrupt corresponding to one of the bits 4-1 of the IFR, see 5.3.5 below).

5.3.5/

5.3.5 Program Mask Register. This is a 4 bit register in the Control Unit. The bits correspond to bits 4 to 1 of the IFR. This register can be set using Set Program Mask, so that a programmer if he wishes can cause normal calculation to proceed even though an abnormal arithmetic condition has occurred. If one of the four reasons for interrupt occurs, and the corresponding bit in the Program Mask Register is zero the reason for interrupt is not staticised (that is the corresponding IFR bit is not set) the effect is that the interrupt is ignored. If the bit in the Program Mask Register is 1, then the Interrupt Mask Register is interrogated in the same way as described in 5.3.4 above.

5.3.6 Weight Registers. Whenever an interrupt occurs, the software wishes to know what caused the interrupt. The Weight Registers contain this information. There are two Weight Registers, both in scratchpad, one each for program states 3 and 4. When an interrupt is executed, part of the hardware action is to put the Weight number (see section 5.3.3.3 note 3) in the Weight Register of the state that is about to be entered. The IFR bit corresponding to the interrupt is cleared at this time.

5.4 Action of the Interrupt Control System.

The Interrupt Control System can be activated at any time by certain interrupt conditions (machine fault, power failure and all interrupt conditions to do with Instruction error (bits 10 to 5 of IFR), while certain interrupt conditions (the rest) will only activate the Interrupt Control System at the end of the instruction that the interrupt condition occurred. Interrupt requests are processed by the hardware in three phases:-

1. Initial response and staticising of the request.
2. Selection from the list of pending interrupts one to execute.

3./

3. Execution of the interrupt.

1. If the interrupt request is one of the four maskable by program (corresponding to bits 4 to 1 of the IFR) then the corresponding bit of the interrupt Mask Register is interrogated. If zero the request is ignored and normal processing continues. Otherwise phase 2 below is executed.

2. The flags of the IFR are scanned in order of descending priority until one is found that is 1 which has the corresponding bit of the Interrupt Mask Register of the current Processor state set. This interrupt is executed as described below. If there is no interrupt to be executed, normal processing continues.

3. The execution of an interrupt is a change of program state usually to state 3 unless the interrupt is a machine check or power failure when it is to state 4. The interrupt weight is stored in the appropriate weight register and the appropriate bit is removed from the IFR. The contents of the various Control Unit registers are stored, and some of the registers are reset from the scratchpad as described in 2.4.1.

The action of the Interrupt System is described in detail in flow chart form at the end of this section. Individual interrupts are described in detail in PS 4A.6.5.

Interrupt logic flowchart will be circulated later pending more information on the interrupt system.

SECTION C: THE 4-75 STORE AND ADDRESSING SYSTEM

The 4-75 central processor is described in detail in PS 4.10.70. Unless otherwise stated, all references in this summary are to the decimal numbering system used in the product specification.

Terminology

Throughout this paper, the term main store will refer to the core store of the central processor. Each 8-bit byte of the main store can be specified by its absolute address. The highest absolute address that can be used clearly depends on the number of bytes of core storage actually available on the machine.

The term virtual memory will refer to the 2^{24} bytes of storage which are made available on demand to any user program. At any time, only the portions of this virtual memory which the user has already requested from the system have a physical existence; moreover, the storage space being used may be in the main store, or may equally well be, for example, on a disc or a drum. Each byte of the virtual memory has a virtual address in the range 0 to $2^{24}-1$.

As far as a user of the system is concerned, all programming is done in terms of virtual addresses, and it is the task of the system software to maintain a correspondence between virtual addresses and physical storage devices.

Some familiarity with the interrupt system of the 4-75 will be assumed. This system is described elsewhere. (Section B)

3.1 Main Store

Main store is made up of one or more modules of core store. A byte of store consists of 8 data bits and an associated parity bit, while a module holds some multiple of 65536 bytes. Each byte has an absolute address starting from 0 upwards. Store modules are made in two halves, in such a way that all bytes with an odd address are in one half of the module, while all bytes with an/

an even address are in the other. As the two halves of the module may be accessed concurrently, this 'interleaving' of the store will often help to reduce the necessary access time.

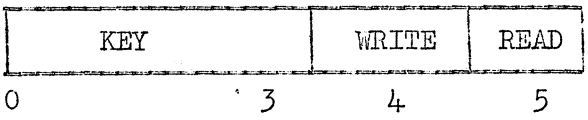
A word of store is 4 bytes, so that possible word addresses are 0, 4, 8, ... etc.; similarly a double word of store is 8 bytes, with address 0, 8, 16, ... etc.

The store may be accessed either by the Central Control Unit or by any of the Channel Control Units. (Up to 16 Channel Control Units are used to control input and output.) Access time is 450 nanoseconds, the cycle time being 1 microsecond, or 750 nanoseconds if the effects of interleaving are considered. The parity of all bytes entering or leaving the store is checked, a parity failure giving rise to a 'machine check' interrupt.

An attempt to access non-existent store will lead to an 'address error' interrupt. The trick described in 3.6.3.3 (wrap-around) should on no account be used.

3.7 and 102.4.4 Store Protection and Read/Write Markers

With each block of 128 words of main store there is associated a Store Protection register with the following format:



The register is stored or loaded by means of the privileged instructions ISK (insert storage key) and SSK (set storage key), and is used as described below.

Protection Areas of Main Store can be reserved in various ways by allocating to a program an associated Store Protection Key. When an attempt is made to write into store, this protection key is compared with the first four bits of the store protection register associated with the word addressed. The write operation is only completed if either both keys are identical, or if one or other of the keys is zero; otherwise an interrupt is caused.

Read/

Read/Write Markers Whenever a 'read' demand is made to an area of store, either by the Central Control Unit or by a Channel Control Unit, then the 'read marker' bit of the store protection register associated with that area will be set to 1. The 'write marker' bit operates similarly. In this way the supervisor system can tell how a given area of store has been used since the markers were last cleared to zero.

3.8 Reserved Store

The first part of main store contains a number of areas used for special purposes, which may therefore be found to be protected against overwriting. Up to 8192 bytes of store are liable to be reserved in this way.

4. The Scratchpad Store

The scratchpad is a 72 word high-speed store, having an access time of 120 nanoseconds and a cycle time of 300 nanoseconds. It is made up of four sets of sixteen registers, one set for each processor state, and of eight registers used in pairs to form four double-length floating point registers. Each byte of the scratchpad is parity checked.

Most scratchpad words have two addresses. Normal instructions specifying a stratchpad register address one of the 16 registers belonging to the pro-:cessor state in operation when the instruction is executed. However, the privileged instructions 'load stratchpad' (LSP) and 'store scratchpad' (SSP) may address any register in the scratchpad, using the scratchpad addresses shewn below.

Floating point registers may be addressed by floating point instructions in any state. They may also be addressed by LSP or SSP instructions.

The scratchpad is addressed by words: it is not possible to address individual bytes.

Many of the words on the scratchpad have special functions connected with the interrupt system. These functions, and the two addresses of each word, are given below.

State/

State	Normal address	Scratchpad address	Function
	0	0	Segment table base
	1 - 3	1 - 3	General purpose
	4	4	Addr. of page turning routine
	5	5	Page identity register
P4	6 - 11	6 - 11	General purpose
	12	12	Interrupt mask P4
	13	13	Interrupt status P4
	14	14	Program counter P4
	15	15	Weight P4
	0	32	Interrupt mask P1
	1	33	Interrupt status P1
	2	34	Program counter P1
	3	35	Interrupt flags
	4	36	Interrupt mask P2
P3	5	37	Interrupt status P2
	6	38	Program counter P2
	7	39	General purpose
	8	40	Interrupt mask P3
	9	41	Interrupt status P3
	10	42	Program counter P3
	11 - 14	43 - 46	General purpose
	15	47	Weight P3
P2	0 - 15	64 - 79	General purpose
P1	0 - 15	96 - 111	General purpose
	FP 0	112 and 113	floating point registers
	FP 2	114 and 115	
	FP 4	116 and 117	
	FP 6	118 and 119	

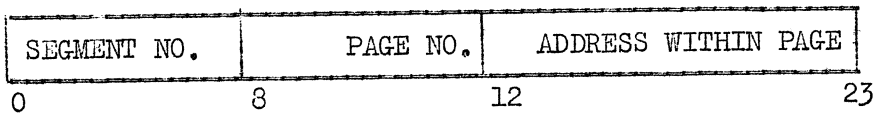
The associative memory (see below) may be addressed as scratchpad register 128.

101/

101 and 102 Virtual Memory and Paging

Virtual memory is addressed by a user program as if it were a large homogenous directly-addressable store. However, the system functions in such a way that only those parts of the virtual memory which are involved in current activities are actually in main store, the rest of virtual memory being in the backing store (usually disc or drum).

Each program's virtual memory is divided into units called pages. A page is the unit of storage for the transfer of data between main store and the backing store, so that any particular page of a program's virtual memory is either all in the main store, or all out on the backing store. A page is 4096 bytes of storage and the address of the first byte of a page in main store must always be a multiple of 4096. Sixteen pages form a segment, so each program's virtual memory consists at its maximum of 256 segments. Segments are numbered, pages are numbered within segments, and within a page the individual bytes are numbered, so that any byte of the virtual memory can be addressed by a 24 bit virtual address with the following format:



When a user program tries to execute an instruction which specifies a virtual address, then the paging system (which is part of the hardware of the 4-75) checks to see whether the virtual address refers to a byte of virtual memory which is currently being held in the main store. If so, then the instruction is allowed to proceed, the virtual address specified being replaced, however, by the absolute address of the relevant byte of main store. If the virtual address refers to a byte of virtual memory which is not in the main store, then the instruction is inhibited, and supervisor is entered. The necessary steps must now be taken by the software to bring the relevant byte of virtual memory into main store. The instruction can then be allowed to/

to proceed as above. The paging system is described in detail below.

A segment, or 65536 bytes, is the unit of virtual storage used in making allocations of space to a program. A user program has available to it only the number of segments (up to 256) which it has currently claimed from super-:visor. Typically, a program might use one segment to hold user-supplied instructions, one segment to hold library routines, one segment for work space, and one segment each for input and output files. If necessary, several con-:secutive segments can be grouped into an area.

101.5.3 Paged Address Mode and Standard Address Mode

The 4-75 processor can operate in either Paged Address Mode or Standard Address Mode. In paged address mode, each address specified by an instruction is inter-:preted as a virtual address, from which the corresponding absolute address is determined by the paging system. In standard address mode instructions are assumed to be referring to operands by their absolute addresses, so no conver-:sion is necessary. The mode in which the processor operates is determined by the following three criteria:

102.4.3 (i) there is an engineer's switch which may be set to inhibit the operation of the paging system completely.

(ii) there is a one-bit register in the Central Control Unit called the address mode (AM) register. If this register is set to 0, then the paging system is inhibited, and the processor operates in standard address mode.

(iii) if paging is not inhibited either by the engineer's switch or by the AM register, then in processor states P1 and P2 the pro-:cessor operates in paged address mode. In states P3 and P4 the processor always operates in standard address mode.

103.5.6 The AM register is loaded during a change of state with the contents of bit 22 of the Interrupt Status Register of the initiated state. The old value of the AM register is not stored.

102/

102.3 When the processor is operating in paged address mode, then the Pro-:gram Sequence Counter operates with virtual addresses, which are interpreted by the paging system. The contents of the Relocation Base register are added to virtual addresses, so to avoid confusion the RB register should normally be set to zero when in paged address mode. 'Branch and link' instructions store a virtual address when the processor is in paged address mode.

102.7 However, even in paged address mode, peripheral operations require that the Channel Control Units be presented with absolute addresses. The following conditions must be satisfied before an I/O command may be initiated:

(i) the addresses specified by the Channel Address Word and its Channel Command Words must be absolute and not paged addresses.

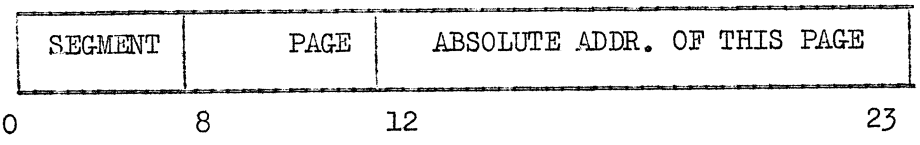
(ii) therefore the pages required for the operation must be in store.

(iii) if the specified I/O area is on more than one page, then either the required pages must be contiguous in the main store, or else data chaining must be performed appropriately.

103 The Paging System

103.5.3 Associative Memory

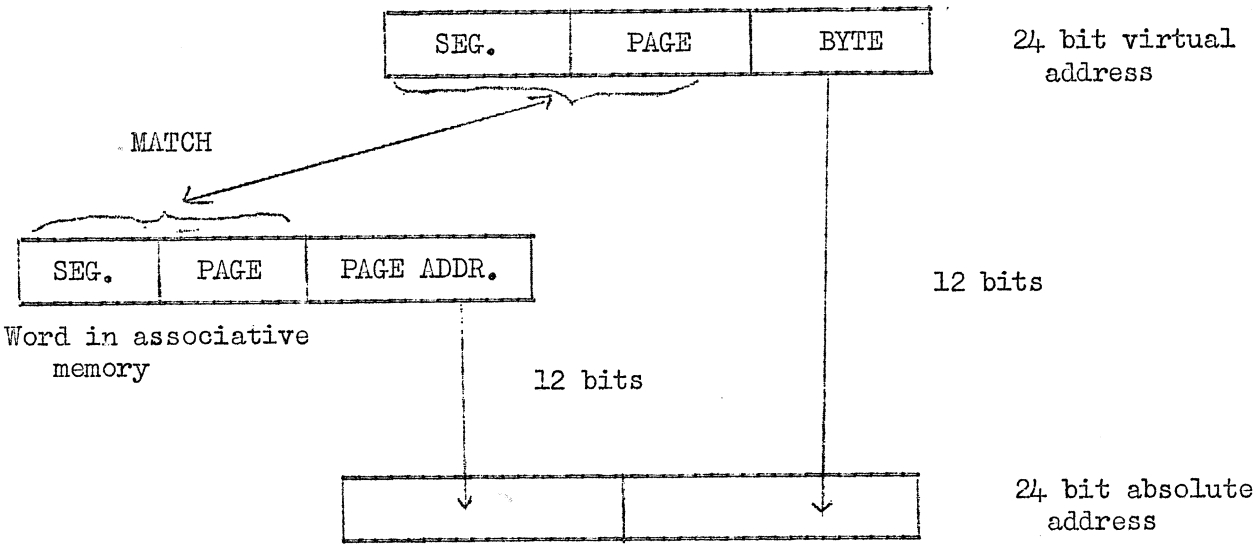
The associative memory is used by the paging system to give fast access to the eight most frequently used pages. The associative memory is eight words long, each word having the following format:



The first 12 bits of the 24-bit word provide the association key used by the paging system as described below. Each word in the associative memory is assigned a place in a queue, the entries being ordered according to the current activity of their associated pages. Whenever a word of the associative memory is referenced during conversion of a paged address, it is moved to the head of the queue; a new entry to the associative memory is/

is put at the head of the queue, the word at the tail of the queue being lost.

103.5.3.4 During the conversion of a virtual address to an absolute address, the most significant 12 bits of the virtual address (i.e. the segment and page number) are compared with the m.s. 12 bits of each word in the associative memory. These comparisons are performed simultaneously, and if there is a word in the memory which matches, then its least significant 12 bits hold the absolute address (divided by 4096) of the first byte of the referenced page. If these l.s. 12 bits are concatenated with the l.s. 12 bits of the virtual address, then the result is the absolute address corresponding to that virtual address. Diagramatically:



The word in associative memory where the "hit" was scored is moved to the head of the queue.

If none of the words of the associative memory matches the segment and page number of the required address, then the absolute address of the required page is determined from the segment and page tables described below.

103.5.3.5 The associative memory may be addressed by 'load scratchpad' (LSP) and 'store scratchpad' (SSP) instructions using scratchpad address 128. If a single word is loaded, it is put at the head of the queue, the word at the end of the queue being lost; if several words are loaded by a single instruction, the/

the words are loaded in ascending order of priority, the last word ending up at the head of the queue. If a single word of the associative memory is copied to store, it will be the word from the tail of the queue which is copied; if several words are copied, then they are copied in order starting with the word at the tail of the queue.

The associative memory must only be addressed when the processor is in state P3 or P4, otherwise entries may be altered which are needed to complete the current instruction.

Words entering and leaving the associative memory are not parity checked.

The supervisor will normally clear the associative memory whenever the current program is changed.

103.5.4 Segment Tables

Each program has a segment table, which is a list containing one entry for every segment so far allocated to the program. The format is as follows:

0	8	31
SPARE	ADDR. OF PAGE TABLE	0
SPARE	ADDR. OF PAGE TABLE	1
etc.		etc.

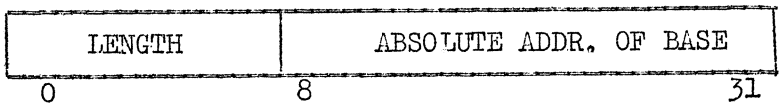
A segment table in main store must begin at a word boundary, and must consist of the required number of consecutive words. The first word holds the absolute address of the page table (see below) associated with Segment 0, the second word the absolute address of the page table of Segment 1, and so on. If a segment no longer exists for some reason, then its entry in the segment table must point to a dummy (all zero) page table. The spare field of each word is ignored, but should not be used by the Software.

When a program is running (rather than waiting) its segment table must clearly be in main store.

103.5.7/

103.5.7 The Segment Table Base Register (Register 0 of state P4)

The format of this 32-bit register is:



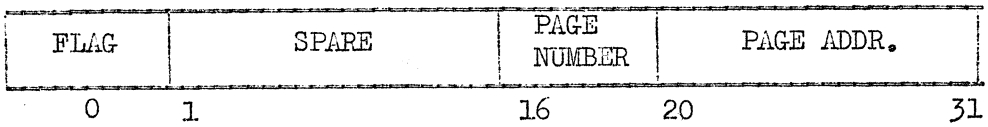
The number in the "length" field must be one less than the length in words of the current segment table. The "base" field holds the absolute address (in bytes) of the first entry in the current segment table.

Whenever the current program is changed, the supervisor must alter the segment table base register appropriately.

103.5.5 Page Tables

A page table is a 16 word list of the pages associated with a particular segment. Each entry in the table indicates whether or not a particular page is in main store, and, if the page is present, it specifies the page's address. The page tables for a program which is running must clearly be in main store themselves.

The absolute address of the first word of the page table, which must be aligned at a word boundary, is held in the corresponding entry of the segment table. The first word of the page table holds information about page 0 of the segment, the second word information about page 1, and so on. The format of each entry is

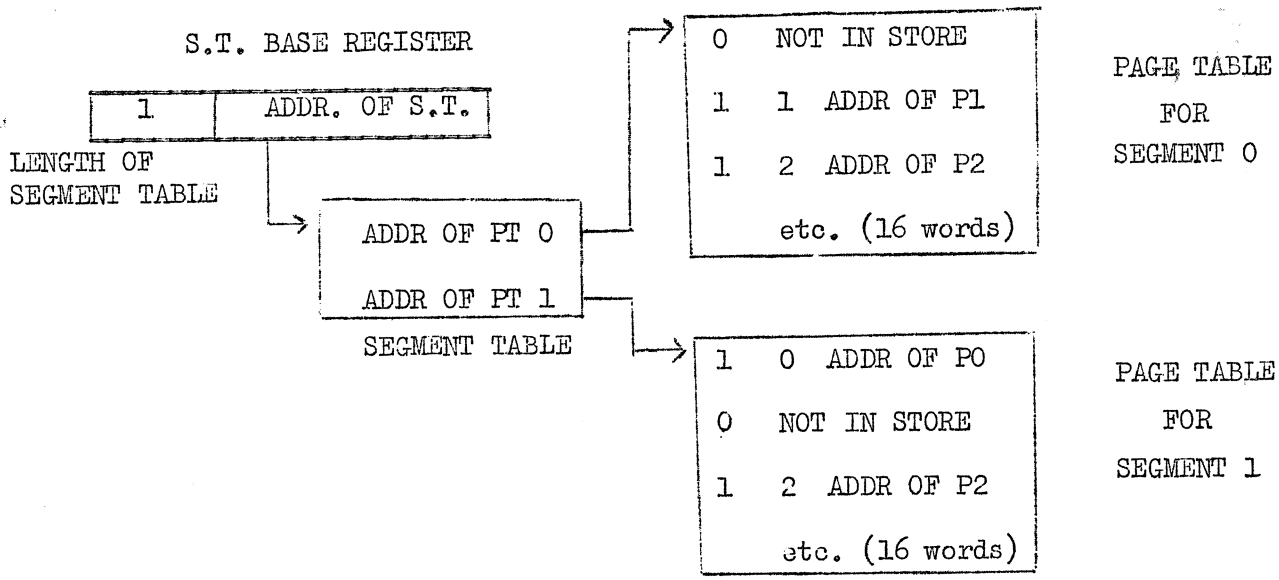


The flag bit must be set to 0 when the relevant page is not in store, and to 1 when the page is in store. If the flag is 0, then the rest of the word is disregarded by the paging hardware, and may contain any useful information. The spare field is always disregarded.

Bits 20 to 31 of an entry hold the 12 most significant bits of the page's absolute address in main store. Whenever an entry in the page table is addressed by the paging hardware, the "page number" of the entry is placed in the "page" field of a new entry in the associative memory and the "page address"/

address" is loaded to the "page address" field, while the "segment" field of the new entry is loaded with the segment number specified by the virtual address being converted.

The tables in store are thus held, diagramatically, as shewn:



It is the responsibility of the supervisor to keep these tables up-to-date. The paging hardware uses the tables, but does not alter them in any way. The associative memory is, however, altered by the paging hardware.

103.6 Addressing Mechanism

The process of converting a virtual address to an absolute address is therefore as follows:

(i) The associative memory is examined. If an entry with the required segment and page number is found, then the absolute address is formed as shewn above, and the conversion process is complete.

(ii) Otherwise, the specified segment number is compared to the "length" field of the S.T. base register; if the latter is less than the required segment number, an address error interrupt is caused.

(iii) The address of the page table for this segment is now determined from the appropriate entry in the segment table (using the specified segment number and the S.T. base register).

(iv)/

(iv) The required page word is now located in the page table (using the page table address from (iii) and the specified page number).

(v) The page word flag is examined to see whether the specified page is in store or not.

(a) if it is, then the contents of the least significant 16 bits of the page word are placed in the l.s. 16 bits of a new entry in the associative memory, and the 8-bit segment number specified by the instruction is placed in the 8 m.s. bits of this entry. The new entry is put at the head of the queue in the associative memory, the word at the end of the queue being lost.

When the associative memory has been loaded in this way, the virtual address specified is again presented to the address conversion system. This time, of course, stage (i) of the conversion process should succeed. If it does not, then an error interrupt is caused.

(b) if the required page is not in main store, then a 'page turning' interrupt is caused. It is then a software function to bring the required page into store, to update the page table, and to re-start the interrupted instruction.

102.6.3 The Page Turning Interrupt

A page turning interrupt is caused whenever the page specified by a virtual address is not present in main store. The PT interrupt is executed immedi-:ately, and does not have to wait its turn in the normal interrupt priority queue (see Section B). This interrupt has neither an interrupt flag nor a mask bit associated with it.

A PT interrupt is executed in the following manner:

(i) the current instruction is inhibited;

(ii) the central control storage registers of the interrupted state re-:ceive the information normally stored at a change of state. The Program Sequence Counter contains the address of the start of the current/

current instruction; the Instruction Length Code may be stored incorrectly;

(iii) the segment/page number of the missing page is placed in register 5 of state P4;

(iv) the Sequence Control Counter is loaded with the contents of register 4 of state P4, and normal processor operation commences in state P3, the Interrupt State Identifier being loaded appropriately.

104.2 Performance

The times taken to convert a virtual address to an absolute address are as follows:

(i) if a "hit" is scored in the associative memory - 120 nanosec.

(ii) if a new entry needs to be loaded to the associative memory - 4.75 microsec.

(iii) if a page-turning interrupt is necessary, execution of the interrupt takes about 5 microsec.

SECTION D: THE 4-75 PERIPHERAL CONTROL SYSTEM

This description of the peripheral control system is a condensation of Section 6 of product specification PS 4.10.70. Unless otherwise stated, the decimal numbering system used here refers to this product specification, which should be consulted if fuller details are required.

6.2 Introduction

The control of peripheral devices, and of the transfer of data between them and the main store, is the responsibility of Channel Control Units. The Central Control Unit obeys an I/O instruction by initiating the operation of the relevant Channel Control Unit, which thereafter proceeds autonomously. This autonomous operation of the Channel Control Unit is directed by instruc-:tions called Channel Command Words, which are held in the main store. When the autonomous action of a Channel Control Unit is finished, then a request for execution of a 'channel termination interrupt' is made to the Central Control Unit. Information is returned from a channel in the Condition Code, and in a special word called the Channel Status Word.

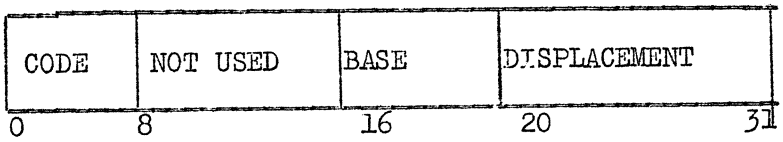
6.4.2.3 There are two types of channel, namely selector channels and multi-plexor channels. A selector channel, which is used to connect high-speed devices to the processor, can maintain one transfer at a time, and transfers information to and from the main store one word at a time. A multiplexor channel, which is used to connect slow peripheral devices to the processor, can maintain up to 256 transfers at a time, transferring information to and from 256 separate areas of main store one byte at a time.

6.7.3.1 Each individual device of the peripheral system is addressed by a unique combination of a channel number, in the range 0 - 15, and a device number, in the range 0 - 255. Information is returned from a device in a byte called the Standard Device Byte, and also in one or more extra bytes called the Secondary Information.

6.7.4/

6.7.4 The Input/Output Instructions

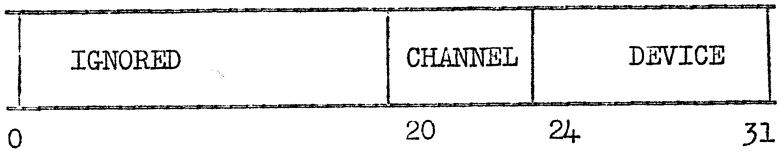
There are four I/O instructions, which are used to initiate all peripheral operations. These instructions are privileged, i.e. they may not be used in a normal program, but only in the supervisor system. Each instruction specifies a channel number, and all but one specify a device number. The instruction format is



the instruction codes (hexadecimal) being:

start device	(SDV)	9C
halt device	(HDV)	9E
test device	(TDV)	9D
check channel	(CKC)	9F

The contents of the specified base register are added to the specified displacement, and the resultant sum interpreted as



Start Device (SDV)

This instruction is used to initiate all peripheral operations. Its execution must be preceded by the setting-up of one or more Channel Command Words, and of a Channel Address Word (see 6.7.5 and 6.7.6 below). The Condition Code is set by this instruction.

Halt Device (HDV)

This instruction is used to terminate the operation of a particular device. If the device is operating on a selector channel, then no device number need be/

be specified. No Channel Command Word or Channel Address Word is needed. The Condition Code is set by this instruction.

Test Device (TDV)

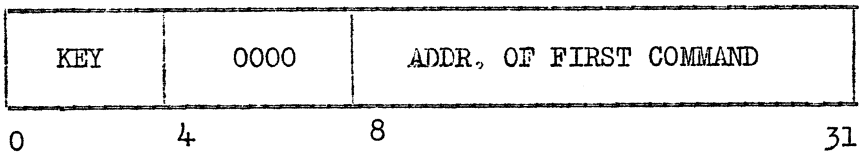
This instruction tests the availability of a particular device. No Channel Command Word or Channel Address Word is needed. The Condition Code is set.

Check Channel (CKC)

This instruction tests the status of a particular channel. No device number, Channel Command Word, or Channel Address Word is needed. The Condition Code is set.

6.7.5 Channel Address Word (CAW)

The CAW is a four-byte word with main store address 72 - 75 inclusive. It must be loaded before an SDV instruction is obeyed. Its format is as follows:

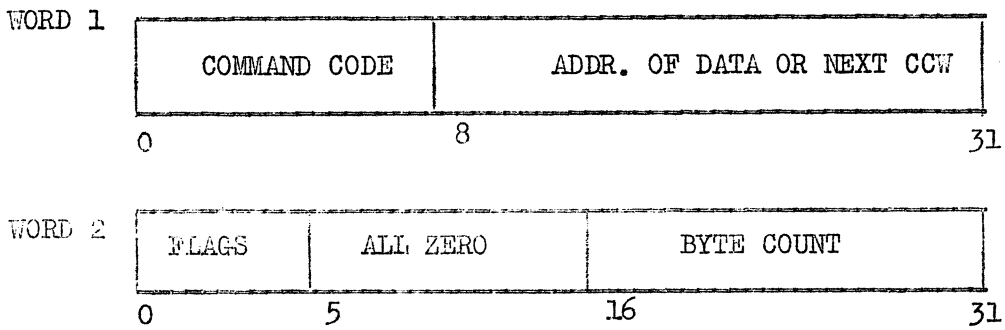


Bits 0 - 3 specify the store protection key (see 3.7.1) to be used in this I/O operation.

Bits 8 - 31 specify the absolute address of the first Channel Command Word which is to be obeyed.

6.7.6 Channel Command Words (CCW)

Channel Command Words supply information to direct the operation of a Channel Control Unit. A CCW is 8 bytes long, and must be positioned on a double-word boundary in main store. The format of a CCW is as follows:



The/

The purposes of the various fields are described below.

6.7.7 Command Codes

The first 8 bits of a CCW constitute the command code. The following codes are assigned:

M M M M	0 0 0 1	sense
M M M M	M 0 1 0	read reverse
M M M M	M 0 1 1	write
M M M M	M 1 0 0	write erase
M M M M	M 1 0 1	read
M M M M	M 1 1 1	write control
M M M M	1 0 0 1	transfer in channel

The bit positions marked M are designated for use by particular devices (see the appropriate device specification). A command code other than one of those given above will cause termination of the associated I/O operation.

Sense

Secondary status information is transferred from the device to main store. Further information is given in the relevant device specification.

Read Reverse

Information is read as it passes the peripheral in the reverse direction. The data address specified in the CCW should be that of the highest byte of the input area intended; successive bytes of information are put into main store from this high address downwards.

Write

Information is transferred from main store to the peripheral.

Write Erase

This code may be used only when addressing a magnetic tape device. The specified number of bytes are erased. Main store is accessed by 'read' demands,/

demands, even though the device control unit does not write the information it receives.

Read

Information is transferred from the peripheral to main store.

Write Control

Information is transferred from main store to the Device Control Unit of the peripheral, which responds in the manner described in the relevant device specification.

Transfer in Channel

This command is used to link non-consecutive Channel Command Word sequences. Bits 8 - 31 of word 1 of the CCW with the 'transfer in channel' command code specify the absolute address of the next CCW to be obeyed.

A 'transfer in channel' command may not specify the address of a CCW which is also a 'transfer in channel' command; also the first CCW to be obeyed following a SDV instruction may not be a 'transfer in channel' command.

6.5.7 and 6.5.8 Data or CCW Address

Bits 8 - 31 of word 1 of a CCW specify an absolute address.

If the command code of the CCW is 'transfer in channel', then this address specifies the location in the main store where the next CCW is to be found. Otherwise, this address specifies the location in the main store to or from which the next byte of information is to be transferred. Normally information is transferred starting at this address and working up the store. In the case of a 'read reverse' command, however, information is transferred starting at this address and working downwards.

6.5.6 Channel Command Flags

Bits 0 - 4 of word 2 of a CCW are used as flags to affect the action of a Channel Control Unit. The use of each flag is described below.

BO/

B0 - Chain Data Flag

This bit should be set when it is required to transfer data between a single unit of peripheral storage (e.g. a block on magnetic tape) and a number of non-consecutive areas of main store. For instance, to transfer 100 bytes from address 6000 upwards to magnetic tape, and then to transfer 50 bytes from address 7000 upwards into the same magnetic tape block, we might use the following two CCWs, which must be in consecutive store locations:

CCW 1:	command code	- write
	data address	- 6000
	flags	- chain data flag set
	byte count	- 100
CCW 2:	command code	- any (except 'transfer in channel')
	data address	- 7000
	flags	- none
	byte count	- 50

CCW 1 will be obeyed. When the required number of bytes have been transferred, CCW2 is used to provide a new data address, a new set of flags, and a new byte count. The command code of CCW2 is ignored unless it is 'transfer in channel'.

B1 Chain Command Flag

If this bit is set, then when the current operation terminates, the Channel Control Unit will proceed to obey the following CCW. Otherwise operation terminates when the current CCW has been obeyed.

If the Chain Data Flag (B0) is set, then the Chain Command Flag is ignored.

If the Chain Command Flag and the status modifier bit of the Standard Device Byte (see below, 6.6.6) are both set at the end of an operation, then the following CCW is skipped, and instead the next-but-one CCW is obeyed.

B2/

B2 Suppress Length Indicator Flag

When this flag is set, the Incorrect Length Indicator for the channel is inhibited. The flag should be set when it is expected that the byte count specified by a CCW may be too large. For instance, when reading variable length blocks from magnetic tape, the byte count will normally be set equal to the length of the input buffer, and reading a block shorter than this will not necessarily indicate an error.

If the Chain Data Flag (B0) is set, then the Suppress Length Indicator Flag is ignored.

B3 Skip Flag

If the Skip Flag is set on a 'read', 'read reverse', or 'sense' command, then the information read from the peripheral is not transferred into the main store. If this flag is used in conjunction with the Chain Data Flag (B0) then it is possible to select certain sections of a block for reading into store, while letting the rest pass unrecorded.

On any other command the Skip Flag is ignored.

B4 Program Controlled Interrupt Flag (PCI Flag)

The PCI flag is used to indicate to a program the progress of events within a channel command sequence. When a CCW with this flag set is obeyed by the Channel Control Unit, a request for an interrupt is signalled to the pro-:cessor in the normal way. On execution of the interrupt, the PCI bit of the Channel Status word (see below, 6.5.11) will be set.

The operation of the Channel Control Unit is not affected by this request for an interrupt. If a second CCW with the PCI flag set is obeyed before the first interrupt request has been noticed by the processor, then the first interrupt request will be lost. Further, if the Channel Control Unit finishes its sequence of operations before the interrupt request is noticed, then again only one interrupt will be executed. In this case both/

both the PCI bit and the Termination Interrupt bit will be set in the Channel Status word.

Channel and device status information stored during the execution of an interrupt refers to the status at the time the interrupt is executed, and not to the status at the time the interrupt was requested.

6.5.10 Byte Count

Bits 16 - 31 of the second word of a CCW specify the maximum number of bytes that are to be transferred during execution of the CCW. A count of zero is in general treated as a count of 65,536.

If, when the required number of bytes have been transferred, the end of a unit of peripheral storage (e.g. one block on magnetic tape) has not been reached, then in some cases the device will continue to operate until such an end is reached: see the relevant device specification.

If, on the other hand, the end of a unit of storage is reached before the required number of bytes have been transferred, then the Incorrect Length bit in the Channel Status byte will be set, and the operation terminated, unless the Suppress Length Indicator Flag (see 6.5.6 above) is set.

6.8 Peripheral System Status Storage

The status of components of the peripheral system is presented in three levels of detail. Information is presented in the following positions:

a/ the Condition Code register;

b/ the Channel Status word;

c/ Secondary Information in the main store.

6.8.7.3 The Condition Code Register is set during the execution of any I/O instruction, and may be tested by a subsequent 'branch on condition' instruc-:tion. Settings may be briefly summarised as follows:

Setting/

Setting	Instruction	Interpretation
CC = 3	All I/O instructions:	hardware failure, or non-existent device specified.
CC = 2	SDV, TDV:	the device or channel is busy, any new operation must wait.
	HDV:	the operation on this channel has finished, and an interrupt is pending.
	CKC:	the channel is busy.
CC = 1	SDV, TDV:	something is wrong, examine the Channel Status byte and the Standard Device byte to find out what.
	HDV:	if the channel is a selector channel, this CC setting indicates a hardware fault.
	CKC:	an interrupt is pending.
CC = 0	SDV:	channel operation has commenced (a SDV instruction specifying a 'sense' command may return CC = 0 even though the addressed device is inoperable.)
	TDV:	the device is available for an operation.
	HDV:	the addressed device or channel is not busy.
	CKC:	the channel is available (N.B. the device may not be!) Multiplexor channels always reply with CC = 0 except in the event of hardware failure.

For full details, see PS 4.10.70, section 6.8.

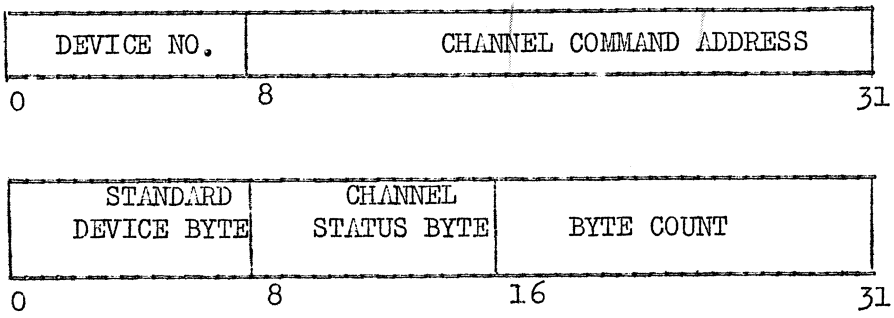
6.8.4 The Channel Status Word (CSW)

The Channel Status Word is used as a standard storage location for control information generated by Channel and Device Control Units. Information is stored/

stored into the word whenever one of the following occurs:

a/ An SDV, TDV or HDV instruction is executed which sets the Condition Code to 1

b/ A program control interrupt, a channel termination interrupt, or a manual request interrupt is executed. The CSW is a double-length word with main store address 88. Its format is as follows:



The fields of the CSW are described below.

Device Number

Identifies the device which caused information to be loaded. The channel number of the channel causing an interrupt must be determined from the P3 Weight Register - no further channel identification is given.

Channel Command Address

This field of the CSW normally holds the address of the next Channel Command Word to be obeyed, i.e. an address 8 bytes higher than the address of the last Channel Command Word to be executed. See, however, 6.8.4.5.

6.6.6 Standard Device Byte

The bits of this byte have the following significance:

B0 = 1 Manual Request bit: the device requires servicing. When this bit is set, further action by the Device Control Unit is inhibited.

B1 = 1 Termination Interrupt Pending. The action of the device has ended, and an interrupt request has been made, but not serviced yet.

B2 = 1 The device is busy.

B3/

B3 = 1 The Device Control Unit is busy.

B4 = 1 The device is available for a further operation.

B5 = 1 There is some secondary information available at the Device Control Unit. If this bit is set while a chain of commands is being obeyed, the operation ceases at the end of the current command. After an SDV instruction with a 'sense' command, this bit is reported zero even though secondary information is present.

B6 = 1 The device is inoperable.

B7 = 1 Status Modifier. This bit is set whenever some specified condition (see the appropriate device specification) occurs, and causes the Channel Control Unit to skip one Channel Command Word. See 'Chain Command Flag', 6.5.6 above.

6.5.11 Channel Status Byte

The bits of this byte have the following significance:

B8 = 1 A program controlled interrupt has been requested.

B9 = 1 Incorrect length: the operation has terminated before the required number of bytes were transferred.

B10 = 1 The program has sent faulty control information, e.g., an illegal command, to the Channel Control Unit. A list of errors which may set this bit can be found at 6.5.11.5.3.

B11 = 1 A store protection error has occurred.

B12 = 1 A parity failure was detected on a byte being transferred between main store and a peripheral device. A faulty byte entering main store is replaced by FF; outgoing bytes are not changed. Command chaining is terminated at the finish of the current operation.

B13 = 1 A parity failure has occured in the Channel Control Unit.

B14 = 1 The current operation on this channel is complete, and a termination interrupt has been requested but not executed yet.

B15/

B15 = 1 The last operation on this channel has been completed, and the associated termination interrupt has been executed. This bit is reset by an SDV instruction which sets the condition code to 0 or 1.

6.8.4.6 Byte Count

Bits 16 - 31 of the second word of the Channel Status Word normally indicate the number of bytes which remained to be transferred when the I/O operation was terminated. Under certain conditions this count may be meaningless.

6.8.6 Secondary Status Information.

The presence of secondary information in a Device Control Unit is indicated by the setting of B5 of the Standard Device byte (see 6.6.6 above). This information can be brought into a specified area of store by executing a 'sense' command. The interpretation of secondary information is given in the relevant device specification.

6.8.7.4 Channel Interrupt Analysis

Ten steps are given for analysing an interrupt following an SDV instruction. The most useful combinations of bits in the Standard Device Byte and Channel Status Byte are:

a/ B15 = 1 (Channel termination)
B4 = 1 (Device ended)
No fault indicators (e.g. B0 = 0, etc.)
An operation has terminated successfully.

b/ B8 = 1 (Program controlled interrupt)
B15 = 0 (Channel not finished)
No fault indicators
A program controlled interrupt has occurred.

All other combinations of bits indicate a fault condition.

6.9.1 Initial Program Loading is described at length.

SECTION G : THE REPLACEABLE DISC

This section describes the Replaceable Disc Drive emit and the Control Unit. More information will be found in P.S. 4.8.67 and 4.8.65, the numbering here refers to that used in these Product Specifications

Disc Drive Unit (P.S. 4.8.67)

2. Description

The disc stack consists of six discs fixed to a perpendicular shaft and is removeable from the Drive unit. Only the 10 inside surfaces are used for recording. There is one read/unit head per recording surface, all these heads are fixed to one positioning mechanism, which can move to 203 positions.

Definitions

A cylinder is that part of the unit that can be accessed without head movement.
A disc surface is that part that can be accessed by one head with head movement.
A track is the intersection of a disc surface and a cylinder, that is a track can be accessed by one head without head movement.

Addressing

The 203 cylinders are addressed from 0 to 202 (from outside to inside). The 10 heads are addressed from 0 to 9 (from top to bottom)

3. Performance

The maximum track data capacity is 3625 bytes. This is:

36,250 bytes per cylinder
735,875 bytes per disc surface
7,358,750 bytes per disc stack (the amount immediately available to a disc drive)

Average Latency is 12.5 m.s.

Switching/

Switching between heads on one cylinder - practically instantaneous.

Switching from one cylinder to another- from 30 m.s. (adjacent cylinder) to 145 m.s. (between outside and inside cylinders).

Transfer rate - 156K bytes/sec.

Time taken to run down drive, change disc pack and make operable is less than 50 seconds.

Error rate. Unrecoverable error occurs in less than one byte per 10^9 read.

Disc Control Unit (P.S. 4.8.65)

2.1 General

Each block of data is preceded by one or two control blocks which describe the data (e.g. its length and record number). The blocks are separated by gaps which are sufficiently large to allow a CCW command operating on a block to have been set up as a result of calculations using information from the previous block.

Every block of information on the disc has two bytes at the end, being a check of the validity of the information. This is the cyclic check. (CC).

2.2 Track Format

The format of a track is given diagrammatically, in fig. 1. A track consists of a physical index, which identifies the start of the track, a home address block, and the records.

The Physical Index This is built into the disc hardware (it is not a blip on the oxide surface) one per track and it indicates the start of a track. It is used by certain search instructions and for detecting attempts to overflow a track.

The Home Address This contains the address (cylinder and head number) of the track. This is used by the seek instructions to check that the correct track/

track has been found. This block also contains a 1 byte flag.

The Flag can be written and read by the software by means of special instructions operating on the Home Address. (The flags associated with the records are generated by hardware and cannot be set by program. However more below in the description of the Record format).

B0 and B1 must be zero

B2 to B5 are ignored by hardware

B6 and B7 are the Track Descriptor bits . B6 = 1 means that the track is defective and an alternative track is being used in its place. The address of the alternative track is stored in the Identifier pesition of the Track Descriptor Record (q.v. infra). B7 = 1 indicates that the track is being used as an alternative.

NOTE: The organization of the above is the responsibility of Software to a great extent. Software sets the value of B6 and B7, Hardware will detect that they are set and cause termination interrupts during the execution of certain commands, but software has to find the alternative track if an alternative track has to be used. Hardware action when these bits are set is described in the descriptions of the individual commands.

The Records On each track there is a Track Descriptor Record (numbered R0) and one or more data records. Each record has two or three blocks, being the Count, Key (optional) and Data Blocks. Record 0, the so-called Track Descriptor record has exactly the same format as the other records, except that it is not preceeded by an Address Marker. Its use, however, by the hardware is different and will be described in due course.

A The Count Block

This contains a Flag byte, an identifier and the lengths of the other two blocks of the record (Key and Data blocks). (Of course the Cyclic check as well).

(i) The Flag Byte This is generated by the hardware, however it can be read/

read by program.

B0 is zero for every alternate record and one for the others. B0 of R0 is zero. This is used by hardware to detect if any records are missing.

B1. This is normally zero, but indicates when the record overflows onto the next track in the cylinder. In this case B1 = 1. See Overflow facility in Section 8 below.

B2 to B5 are ignored

B6,B7 are the track condition bits and they are set by hardware to the values of B6,B7 in the flag byte of the Home Address

(ii) The Identifier This is five bytes long. As its name implies, it identifies the record. It can be written and read by software, however on a read the Device Control Unit checks that the track address is correct. The first two bytes give the Cylinder number in binary (0 to 202), the next two are the Head Address (0 to 9) and the last byte contains the record number. This can be any value, the record numbers on a track need not be sequential, monatonic or unique. (Record 0 must have zero in this position). In the case of Record 0, of a faulty track, the identifier could contain the address of the alternative track, which could be used by software.

(iii) The Key Lehgth Count. This is 1 byte long and gives the length of the key block of a record (excluding Cyclic check). If there is to be no key, then this count must be zero.

(iv) The Data Length Count This is 2 bytes in length and the length of the data block is contained in it.

B The Key Block

This block is optional, its length is specified in the Count block. The key can be used by software for identification of the corresponding record, by means of the Search Key commands.

C The Data Block

The Data Block's length is specified in the Count Block. The Data block of R0/

RO would probably contain information about the layout etc. of the track while that of the other records will contain the ordinary information of users and the system.

LAYOUT OF ONE TRACK

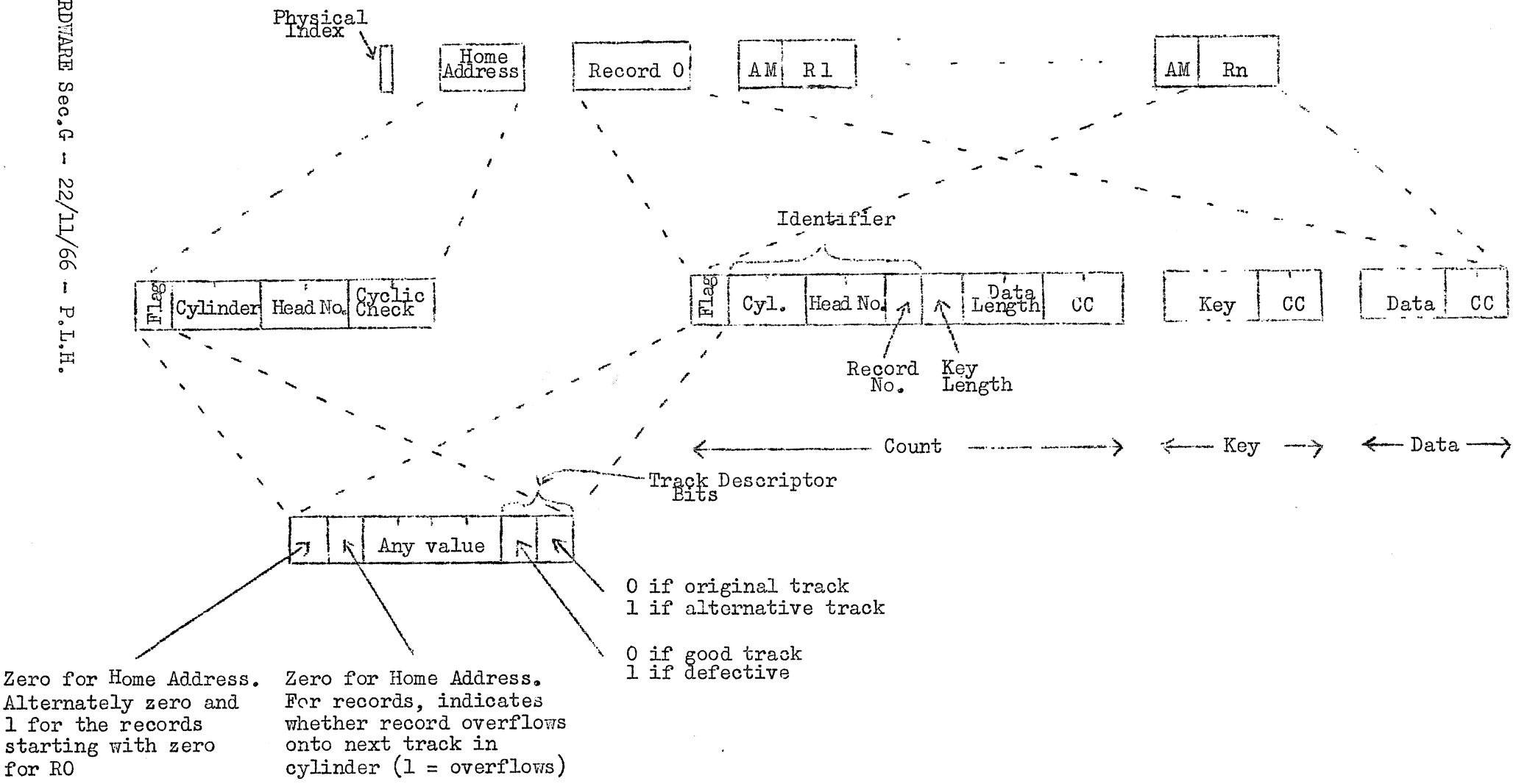


FIG. 1

2.3 COMMAND CODES

2.3.1 Summary of Commands

Operation	Hexadecimal
Seek Commands	
Seek Head	47
Seek Cylinder head	27
Restore	07
Search Commands	
Search Home Address Equal	33 (3B)
Search Identifier Equal	53 (5B)
Search Identifier High	73 (7B)
Search Identifier High or Equal	93 (9B)
Search Key Equal	B3 (BB)
Search Key High	D3 (DB)
Search Key High or Equal	F3 (FB)
Search Key and Data Equal	13 (1B)
Search Key and Data High	C3 (CB)
Search Key and Data High or Equal	E3 (EB)
Real Commands	
Read Initial Program Load	05 (0D)
Read Home Address	25 (2D)
Read Track Descriptor Record	45 (4D)
Read Count, Key Data	85 (8D)
Read Key, Data	65 (6D)
Read Count	E5 (ED)
Read Data	A5 (AD)
Write Commands	
Write Home Address	23
Write Track Descriptor Record	43
Write Count Key Data	83
Write Key Data	63
Write Data	A3
Write Special Count, Key, Data	03

Operation	Hexadecimal
Other Commands	
Erase	E7
Set File Mask	67
Space Record	87
Sense	01

NOTES : 1) Each Search Command and Read Command has two variants, the second (that corresponding to the hexadecimal number in brackets) causes in certain circumstances an automatic addition of one to the head address after the command has been obeyed and the track physical index has been detected. The first variant leaves the head address the same. It should be noted that the second variant is obtained from the first and vice versa by changing B4 of the command code.

2) Seek Cylinder, Head, and Restore are the only commands that cause head movement.

2.3.2. Reads

General If a read command is the first in the chain, its action is to locate the first Address Marker, and to read the relevant part or parts of the corresponding record. If the command is chained from another command, then it reads the next relevant parts that come under the heads.

There are two versions of each Read command. The second (2.3.1), which has B4 of the command code equal to 1, causes the Read/Write head address to be incremented by one when the command has been executed and the Index Marker is detected.

Read Initial Program Load This instruction is sent to the DCU by the processor when the operator pushes the IPL button on the console.

The/

The DCU selects cylinder and head zero, detects the index point (physical index), and then looks for the first Address Marker. The Data part of the foll-:owing record is then read into core.

Read Track Descriptor Record (RO) The DCU looks for Index point*. The Home Address is skipped over and the whole of the record (count, key if any, and data) is read into the core area specified by the command. The count occupies the first 9 core bytes, the key is read into byte 9 onwards (counting from 0) and the data is read into an area immediately following the key.

Read Count This locates the next record, and reads the Identifier, Key length and Data length from the count block into 8 bytes of core**.

To read the Count of a particular record, one would precede this command by another command in a Command chain. If the preceeding command is Search Home Address or Read Home Address, Read Count would read Record 0. If the preceeding command is any other command, or this is the first command of a chain, it searches for the first Address Marker and reads from the corresponding Count field.

Read Count Key Data The next AM is located and the corresponding record in its entirety is read into core.

Read Key Data This roughly reads the next key that come under the heads, and the corresponding data. However if this command is not in a chain, the Key and Data blocks corresponding to the first AM is read.

* If cmd chained from a Search Home Address or Read Home Address, Index Point is not searched for.

** Note the whole Count field is not read

Read Data This reads the next data block that comes under the heads, ex-:cept when the command is the first in a chain, when the data after the first detected AM is read.

Read Home Address The index point is located. In the second version of the command (H'2D') the head address is incremented by 1 and the Home Address is read.

For details of action on abnormal conditions see PS 4.8.65 Page 19 to 22.

2.3.3. Write Commands

General If the write command specifies a smaller byte count than the control unit requires, after the amount specified by the write command has been transferred, the control unit fills the remainder of the field with zeros.

If more bytes are specified by the write command than the control unit requires, transfer of information finishes when the control unit has what it requires, and a channel interrupt occurs with B7 of the first secondary status byte set (Command Code Reject)

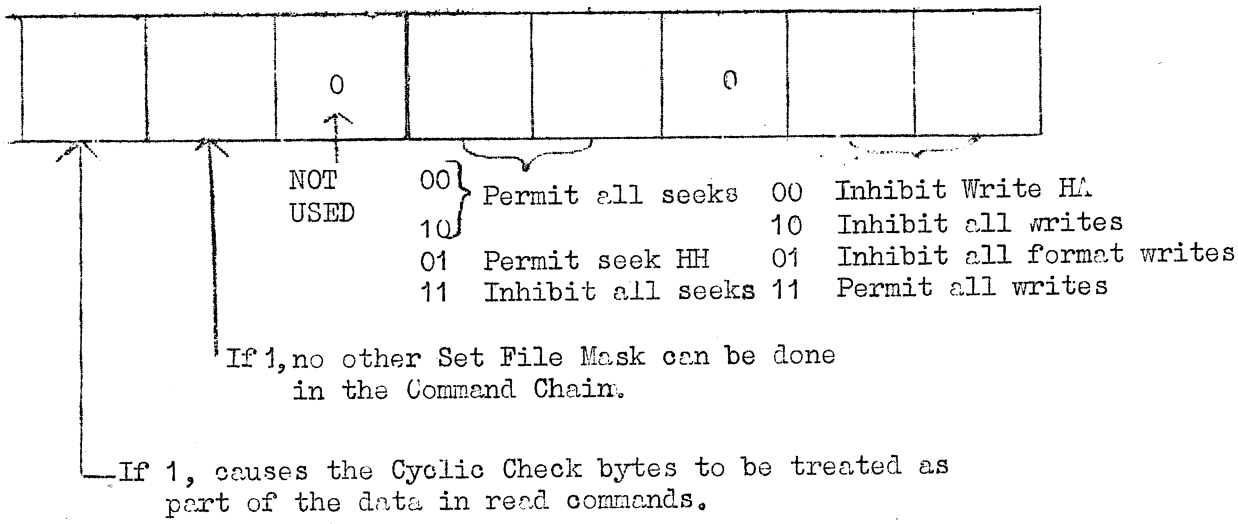
It is possible to overwrite the key and data fields or just the data fields in any record on a track, using the Write Key Data and Write Data commands. All other write commands called Format Writes, after having done the specified write operation cause the remainder of the track to be erased, unless another Format Write follows immediately in the chain.

Data-chaining within a block is dangerous unless complicated rules (not yet formulated) are adhered to.

An End of File mark can be written, using Write R0 or Write Count, Key Data command, by specifying a ZERO-length data block. In fact a 3-byte block is written, one zero byte and 2 cyclic check characters. During a read this block is treated as though it is ZERO length.

Set File Mask/

Set File Mask This is not a Write command, but controls what write commands can be done in a command chain. It causes one byte to be transferred from core to the DCU, which is interpreted as shown in the figure.



At the end of any chain, this byte reverts back to zero, so a Set File Command is only effective until the next Set File Command or at longest for the duration of a command chain.

It will be seen that in the absense of any Set File Command, one may do any seek and any write except write HA.

Attempts to violate the File Mask cause a termination interrupt and the Command Code and File Protected secondary indication are set.

Write Home Address. The Index point is searched for, and five bytes are transferred from the specified core address to the disc (two cyclic check characters are also written to disc).

NOTES : 1) A Set File Mask command must precede this command in the chain

2) Since this first block on a track is used by hardware subsequently for checking after a seek, the cylinder/head address should be that of the track. However, no checking is done during the Write HA command.

3) If another Format Write command (see above) does not immediately follow in the command chain then the rest of the track is erased.

Write RO (Track Descriptor Record). The DCU looks for the Index marker, reads the HA, remembering the flag byte, and writes RO as follows:

The flag byte is written as remembered from the HA, followed by the first 8 bytes from core forming the count block. Subsequent core bytes are transferred to the disc to form the key and data blocks, the lengths of which are specified in the information forming the count block. See Fig.1 in Section 2.2.

NOTES : 1) This is a Format Write, i.e. the rest of the track will be erased unless the command is immediately followed by another Format Write.

2) An End-of-File record can be written by specifying a zero data length.

3) If this command is chained from a Search/Read Home Address command, it will write RO immediately and not look for the Index Marker.

Write Count, Key Data This is similar in action to Write RO (q.v. above), (except that it does not search for the HA). It must be preceded by a command that leaves the head in the right place, i.e.

Any Search command except Search HA equal
Any Read command except Read HA
Write Count, Key, Data or Write RO

NOTES : 1) This is a Format Write Command

2) This command can be used to Write an End of File Record.

3) RO cannot be written using this command

Write Key,Data This must be chained from a Read Count or Search Identifier command (in order to get the heads in the right position and so that the DCU knows the lengths of the Key and Data blocks). Key and Data blocks are written using data from the specified core address, the lengths of the blocks are supplied by the DCU which has stored them from the preceding command.

This can be used for rewriting the Key and Data blocks of RO.

Write Data This must be chained from a Search Identifier, Search Key or Read Count command. The DCU obtains the data length from the identifier of the record and that number of bytes is transferred to the disc.

This command can be used for overwriting the Data field of RO.

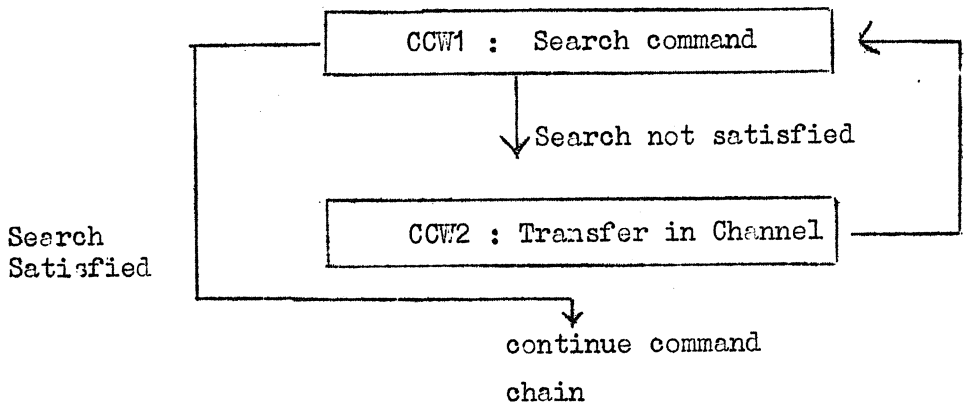
Write Special Count, Key Data This is exactly as Write Count, Key, Data except that B1 of the flag byte of the record is put equal to 1 by the DCU. It is used for writing Overflow Records. See Section 8 below.

For details of action on abnormal conditions, see PS 4.8.65. Pages 32 and 33.

2.3.4. Search Commands

When a Search Command is obeyed, data from the main store is compared with the specified data from the disc, one byte at a time.

If a search operation is satisfied the Status Modifier bit of the Standard Device Byte is set, (see the 4-75 Peripheral Control System which is Section D of this document, section 6.6.6). This tells the Peripheral Control System to skip the next Command and obey the next but one. If the search is not satisfied, the next command in sequence is obeyed. Thus software can by means of two CCWs search a whole track or cylinder for a record or a cylinder for a track thus:



On all search commands, the B4 bit of the command code can provide automatic/

automatic head switching at the end of a track. If B4 = 0, the search is done, and at the end of the track an interrupt is caused, and various bits in the sense bytes are set (See PS 4.8.65 Pages 40 to 42). If B4 = 1 then at the end of the track the next head is selected. If head 9 was initially the one selected then end of cylinder has been reached and a termination interrupt is caused with information for the CPU in the secondary status information.

If the Search command specifies more data than is required by the DCU, then the search can never be satisfied.

If it specifies less data than the DCU requires, the comparing takes place until all data has been transferred. The search can be satisfied.

In the case of the Search Key, Data commands, if a core byte is all ones the search is deemed satisfied for that byte. Thus one can mask out parts of a field for the purposes of a search.

Search Home Address Equal The Index Point is detected. The 4 bytes of the HA specifying the Cylinder/Head address of the track are compared with bytes entering from core. If the search is successful, the next but one command in the chain is then obeyed. If unsuccessful, and the B4 bit of the command code is zero, a termination interrupt occurs. If the B4 bit is 1, the head address is incremented by one, and the next command in the chain is executed.

Search Identifier Equal This looks for an AM and then compares the follow-:ing Identifier byte by byte with the specified core bytes. If chained from a Search/Read Home Address command, it operates on the identifier of R0.

Search Identifier High Same as Search Identifier Equal except that search is successful if the Identifier on the Disc is higher than that in core.

Search Identifier High or Equal See above two commands.

Search/

Search Key Equal The DCU searches for the first AM. It reads the count field of the following record and compares the key field with the specified core area. If successful, the next but one command is the next to be obeyed otherwise the next command is obeyed.

NOTES 1) If this command is chained from a Search Identifier command, then it operates on the key field of the record whose identifier satisfies the Search Identifier command (in this case an AM is not searched for)

2) If the command is chained from a Read/Search HA, it operates on the key of R0.

Search Key High Same as Search Key Equal, except that the search is successful if the Key on the Disc is higher than that in core.

Search Key High or Equal See above two commands.

Search Key and Data Equal

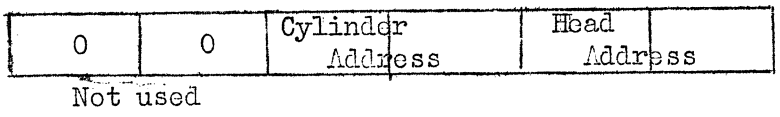
Search Key and Data High

Search Key and Data High or Equal These commands are exactly the same in operation as the three Search Key operations except that the search continues on into the data area (the key and data blocks are treated as one for the purposes of the search command). It is permissable to do these commands on a record with no key, in this case the command operates on the data block only.

Abnormal Conditions These are tabulated for Search commands on pages 40 to 42 of PS 4.8.65.

2.3.5 Seeks

Seek commands (except restore) use 6 core bytes as data which are interpreted as below:



Seek Head This seeks the specified head on the 'current' cylinder. This is obeyed almost instantaneously.

Seek Cylinder and Head This seeks the specified address, and will take as long to execute as it takes to mechanically move the heads.

Restore This requires no data and seeks Cylinder zero track zero.

Abnormal Conditions These are tabulated on Page 44 of PS 4.8.65.

2.3.6. Miscellaneous Commands

Erase This erases from the end of a complete Record to the end of a track. It is valid whenever Write CKD is valid. This is a Format Write command for protection purposes.

Set File Mask This is described in full near the start of 2.3.3 of this section.

Space Record The DCU looks for the next AM on the track, then allows the next command to be obeyed. If chained from the Read/Search HA the next command is obeyed immediately.

The effect of this command is to force the DCU to miss a record.

Sense This command is standard for all peripheral devices and causes Secondary Status information to be brought to core store. In the case of the Disc there are 4 bytes of such information. The next section describes the Secondary Status bytes in detail

2.3.7 Secondary Indicators

Byte 1

B0 Read Parity error. Cyclic check fails.

B1 Service Request Not Honoured. Set if CPU cannot keep up with data rate from or to the disc, or if a chained command comes too late to be executed.

B2 -

B2 Seek Check. Set if for any reason a seek command cannot be executed.

B3 Transmission Parity Error. (on a core byte)

B4 Track Check. Set if an attempt is made to do an automatic head switch from an alternative track, or if an automatic head switch has been done to an alternative or defective track.

B5 Automatic Head Switching error. Set if either i) The Home address of a track that is automatically switched to does not contain the correct head number, or ii) the track switched to is an alternative or defective, or iii) the track that it is attempted to switch from is an alternative track.

B6 End of File. Set if an attempt is made to write an End of File record with a command other than at Write R0 or Write Count, Key, Data. Also if an EOF is detected by a read other than Read HA.

B7 Command Code Reject. Set if i) Illegal command is attempted. ii) After automatic head switching, Cylinder/Head number in the new track's HA do not agree with actual track's address. iii) Violation of File Mask when a Seek or Write is attempted. iv) A write command specifies more data than required by the control unit. v) A seek command specifies less than six bytes.

Byte 2

B0 Count Field Data Error. Cyclic check error on reading count field.

B1 Overflow Incomplete. Set during Automatic Head switching if i) Head number of new track is not equal to that in the HA. ii) New track is defective or an alternative, or present track is an alternative. iii) File Mask prevents automatic head switching.

B2 Missing Address Markers. Set if the Index Marker is detected twice with no intervening AM, or if B0 of the flags of the Count field of two successive records have the same value.

This/

This bit is not set during execution of Search 1D or Read IPL.

B3 File Protected. An attempt is made to do an action prohibited by the File Mask.

B4 Not Found. A search command chain is not successful when the end of track or cylinder is detected as appropriate.

Also set if a Read HA or Read R0 is attempted and either B4 bit of the command is 0 and the end of the track is detected before the HA or R0, or B4 is 1 and the end of the cylinder is detected before the HA or R0.

B5 Invalid Sequence (of commands). These are summarised in the table of P51 of PS 4.8.65.

B6 End of Cylinder. End of cylinder is detected before a command or chain of commands is completed.

B7 Track End. Set if Index Marker is sensed during a Write R0 or Write Count, Key, Data command.

Bytes 3 and 4 These are for engineering purposes.

2.3.10 Track Capacity

Suppose that on a track there are n records and a Track Descriptor Record, p of these have a key and q do not ($p + q = n + 1$). Then it is possible to have any format which satisfies the following inequality.

$81p + 61q +$ (Sum of lengths of all Keys and Data blocks including R0) $+.049$ (Sum of lengths of all Keys and Data blocks excluding the last record on a track.)

≤ 3750

5. Reliability. 2000 hours MTBF

8. Record Overflow

It is possible for a record to overflow onto a second or perhaps more tracks. Overflowing is always to the next head on the same cylinder and one cannot/

cannot overflow onto other cylinders. The record if written in this way consists of a number of Segments one per track, each has the format of a record.

Only three instructions treat an overflow record as a normal record, Read Data, Read Key, Data and Read Count, Key Data. All other instructions treat each segment as a record.

Writing Each segment is written as a separate record. The Write Special Count, Key, Data command is used for all except the last segment. (This causes B1 of the flag of the HA to be put to 1), and the ordinary Write Count, Key, Data is used for the last. Each segment except the first must be the first record after R0 on the track.

Reading The three instructions Read Data, Read Key, Data and Read Count Key, Data will treat an overflow record exactly as one ordinary one. All other read and search instructions treat each segment as a record. For all segments other than the first, when reading using one of the above three reads, only the data parts of the records are brought into core.

SECTION L: MAGNETIC TAPE

The magnetic tape system is fully described in PS 4.8.50. Unless otherwise stated, all references in this summary are to the decimal numbering used in the product specification.

1. Introduction

Two types of magnetic tape unit will be available, one type for 7-track tape, and one for 9-track tape. In both cases the tape used is $\frac{1}{2}$" wide; tape reels may in general be 1200', 2400', or 3600' long.

On 7-track tape units the tape speed is 75 ins/sec, and the recording density may be 200, 556, or 800 char/in, giving a peak transfer rate of 60,000 char/sec.

On 9-track tape units the tape speed is 37.5 ins/sec, and the recording density 800 char/in, giving a peak transfer rate of 30,000 char/sec.

Fast rewinds can be performed at more than double the above speeds.

On 7-track units only, two forms of conversion are available:

a/ character conversion; three bytes of 8 bits may be automatically converted to four characters of 6 bits, and vice versa.

b/ code conversion; 8 bit EBCDIC characters may be automatically converted to 6 bit BCD characters, and vice versa.

2.2 Error Checking Facilities

One of the tracks of the tape is used to record a parity bit with each character. 7-track tape may be read or written with either odd or even parity; 9-track tape must always have odd parity.

At the end of each block an extra character is recorded to provide a longitudinal parity check on each track. This character will be such that each track has even parity.

On 9-track tape only, a cyclic redundancy check character (CRC) is also recorded as the penultimate character in each block. The use of this character/

character is described below.

Writing Check

During a write operation, the data just written to the tape is re-read and checked by the control unit to ensure that the parity of each character is correct, and that at the end of a block the parity of each track is correct. If any of the checks indicates a fault, an indicator is set. This indicator should be inspected by the program and the block re-written if necessary.

Reading Check

7-track tape: each character read is parity checked, and at the end of a block each track is parity checked. If a fault is discovered, then an indicator is set, so that the program may arrange to re-read the block or take any other necessary step.

9-track tape: checks are performed as for 7-track tape, but the cyclic redundancy check character is also used. If all the errors in a block have occurred in one track only (the commonest pattern of errors), the use of the CRC enables the faulty track number to be determined. The block may then be re-read, as described below, using an automatic error correction facility which will correct all the errors caused by the faulty track. Not all detectable errors are correctable, however (e.g. if two tracks are faulty), and in this case the faulty track number will be reported as zero. The program must then take appropriate action.

Compatibility

The error detection facilities are compatible with IBM systems.

2.5 Write Lockout

A special ring must be inserted by the operator on the hub of a tape reel which is to be written or erased. If this ring is not fitted, then an attempt to write or erase the tape is treated as an illegal operation.

4.2/

4.2 Programming

The following commands may be used:

sense	01	(hexadecimal)
read	05	
read reverse	02	
write	03	
erase	04	
write control	07	

(Note The two commands "send status" and "no op." mentioned in PS 4.8.50 are hardware-originated, and should not be used by the programmer.)

Sense

Two bytes of secondary status information are read into store from the designated tape unit. The format of these bytes is given at 4.6 below.

Read

Data is transferred from the tape to main store. The tape travels forward, and successive data bytes are placed in the store in ascending order of address. Transfer of data stops when the required number of bytes have been transferred, or when an interblock gap on the tape is encountered, whichever is the sooner. In the first case, data transfer terminates, but the tape unit continues until an interblock gap is reached, the entire block being parity checked in the normal way. In the second case, the incorrect length marker (see Section D) will be set. Any character which contains a parity error is transferred to store as FF.

Read Reverse

Data is transferred from the tape to main store. The tape travels backward, and successive data bytes are placed in the store in descending order of address. Otherwise this command is the same as 'read'.

Write/

Write

Data is transferred from main store to the tape. The tape travels forwards, and data is transferred from the store in ascending order of address. The tape is check-read immediately it has been written. When the specified number of bytes have been transferred, an interblock gap is generated on the tape.

Erase

A length of tape equivalent to the number of bytes specified in the command is erased, the tape travelling forwards.

NOTE Although no data is written to the tape, nevertheless data is transferred from main store to the device control unit just as if this were a 'write' command. The implications of this for, e.g., store protection should not be overlooked.

Write Control

Control data is transferred from the store to the device control unit. 9-track tapes use one or two bytes of control information, 7-track tapes use three bytes, the middle byte being ignored.

The significance of the bits in these control bytes is as follows:

i) First Control Byte (9-track and 7-track)

B0 = 1 instructs the device to write a 'Tape Mark' (a special one-character block; for its uses, see below)

B1 = 1 rewind to beginning-of-tape. Parity is not checked during the rewind.

B2 = 1 unload the tape. No parity check.

B3 = 1 rewind one block. No parity check.

B4 = 1 skip one block forwards. No parity check.

B5 = 1 (9-track tapes only) The next read from this tape is to be made with automatic error correction. (See 2.2 above.) The track to be corrected is specified in/

in the second control byte.

B6 = 1 rewind until either beginning-of-tape or a tape mark is discovered. No parity check.

B7 = 1 skip forward until a tape mark is discovered.

Not more than one bit should be set in this first control byte.

ii) Second Control Byte (9-track only; ignored by 7-track units)

When B5 of the first control byte is set, then B4 - B7 of the second control byte specify the track to be corrected. B3 is also set to signify whether the faulty block contained an even or odd number of characters. (See 4.6 below.)

iii) Third Control Byte (7-track only)

B0 = 1, B1 = 1 prepare to work at 800 char/in

B0 = 1, B1 = 0 prepare to work at 556 char/in

B0 = 0, B1 = 1 prepare to work at 200 char/in

B0 = 0, B1 = 0 continue working with the present packing density

B2 = 0 prepare to work with even parity tape

B2 = 1 prepare to work with odd parity tape

B3 = 0 set pack/unpack off.

B3 = 1 set pack/unpack on. Until another 'write control' command is given, the control unit will convert four 6-bit characters on tape to three 8-bit bytes in store and vice-versa, when reading/writing on the specified device. This option can be used on odd parity tapes only. See 9.7 for full details.

B4 = 0 set translator off.

B4/

B4 = 1 set translator on. Until another 'write control' command is given, the control until will convert 7-track BCD code to EBCDIC and vice-versa, when reading/writing on the specified device. See 9.8 for full details.

B5, B6, and B7 must all be zero.

Sensible combinations of bits may be used. However, if B3 = 1 then odd parity will be used, even if B2 = 0.

Options remain set for the device specified until another 'write control' command is issued to that device. A complete control byte must be specified whenever a mode change is made.

4.6 Secondary Information

Two bytes of secondary status information are read into store by a 'sense' command. The format of these bytes is as follows:

i) First 'Sense' Byte

B0 = 1 a parity error has occurred during a read operation, or during the check after a write operation.

B1 = 1 service request not honoured.

B2 = 1 a block on tape contained more bytes than were asked for in a 'read' command.

B3 = 1 parity error while writing (B0 will also be set).

B4 = 1 warning of a possible error. Set if

a/ a block of less than 12 characters is read or written (except a one-character tape mark).

b/ an interblock gap more than 25 feet long is found (a normal gap is about $\frac{3}{4}$ inch).

c/ a hardware malfunction is detected.

B5/

B5 = 1 set when

a/ beginning-of-tape is sensed, or

b/ write operations are performed at or beyond the end-of-tape marker.

B6 = 1 set when a tape mark is read, written, or passed during read, write, skip or rewind to tape mark, or skip or rewind one block operations.

B7 = 1 an illegal operation was attempted.

ii) Second 'sense' byte

On 9-track tape this byte contains a binary count indicating the faulty track and an even/odd marker for use with the automatic error correction facility. (See above.) On 7-track tape, B2 = 1 in this byte, if, in the pack/unpack mode of operation, a number of 8-bit bytes not a multiple of three or a number of 6-bit characters not a multiple of four was written/read. (See 9.7)

4.9 Bit and track correspondence

On 9-track tapes the following correspondence between bit positions in the store and tracks on tape is maintained:

Bit	0	1	2	3	4	5	6	7
Track	7	6	5	3	9	1	8	2

Track 4 is the parity track. This system maintains compatibility with IBM and RCA systems.

On 7-track tape, when neither pack/unpack nor BCD/EBCDIC translation is being used, the correspondence is:

Bit	0	1	2	3	4	5	6	7
Track	-	-	6	5	4	3	2	1

Track 7 is the parity track.

The pack/unpack and translate options are fully described at 9.7 and 9.8.

SECTION M : LINE PRINTER

This summary describes the Model 4561 medium-speed printer. All the printers available on System 4 are described in PS 4.8.70, to which document the decimal numbering system used in the summary will refer.

1. Introduction

The 4561 printer has a character set of 64 printable characters plus space. It will print lines of up to 132 characters at a rate of 600 lines/minute if the full character set is in use, and at a rate of 750 lines/minute under certain conditions in which not all the characters are used. Printing can be spaced at either 6 or 8 lines to the inch, selection being by a switch on the control panel.

Multiple line feeds, top-of-form, etc., can be controlled by means of a paper-tape loop; multiple line feeds up to a maximum of 15 can also be specified by program.

The printer can handle paper of width between 4" and 20", and multipart paper up to a maximum of 6 parts (tope and 5 copies).

4. Programming

The following command codes may be addressed to the printer:

sense	01	(hexadecimal)
write	03	
write control	07	

(The commands 'send status', 'no op.', etc., mentioned in PS 4.8.70 are hardware-originated and should not be used by the programmer.)

4.4.1 Sense

This command causes one byte of secondary status information to be input from the printer to main store. Bits of this byte have the following meanings:

B0 = 1 Channel 12 was used on the paper-tape loop (see below)

B1 = 1 Channel 9 do.

B2/

B2 Always zero

B3 = 1 Parity failure during a write operation. The current line has not been printed, and should if possible be re-transmitted.

B4 = 1 An illegal character code was sent to the printer. The offending character is replaced by a blank.

B5 = 1 Printer set on MANUAL.

B6 = 1 Low paper. 14" or less of paper are left in the printer.

B7 = 1 An illegal operation, e.g. 'read', was attempted.

4.4.2 Write

Data is transferred from ascending addresses in main store to the printer buffer. When the required number of bytes have been transferred, or when the printer buffer is full (whichever is the earlier), data transfer ends, and the line of characters in the buffer is printed. The buffer is then cleared.

The buffer of the 4561 printer holds 132 characters. If less data than this is sent, then the line is left-justified, right-hand character positions being left blank. Data transferred to the printer buffer is parity checked, and the line is not printed if a parity failure occurs. Characters not in the printer character set are replaced by blanks.

Normally, the printer will perform a single line feed when a line of characters from the buffer has been printed. However, this automatic action may be altered by a delayed 'write control' command. (See below.)

4.4.4 Write Control

This command causes the transfer of a single byte of control information from main store to the printer. The bits of this byte are used as described below:

B0 count / paper tape loop

B1 delay / immediate

B2)
B3) must be zero

B4-B7 specify a count from 0 to 15

a/

a/ If B0 = 0, then the printer will perform the number of line feeds specified by the count in bits B4-B7. If B0 = 1, then the printer will advance until a hole is sensed in the relevant track of the paper tape format loop, the relevant track being specified by bits B4-B7. Tracks 0, 9, and 12-15 should not be specified by a 'write control' command; a command specifying one of these tracks is treated as an illegal command, and no paper motion results. Track 1 of the paper tape loop will normally indicate top-of-form.

b/ If B1 = 0, then the specified paper advance is not made immediately, but instead it is kept pending until a 'write' command is obeyed. This 'write' operation will then be followed by the specified paper motion instead of by the usual automatic single line feed. (While a 'write control' action is held pending, a further 'write control' command will simply overwrite the previous one.) A delayed 'write control' command affects only one 'write' operation.

If B1 = 1, then the specified paper advance is performed immediately.

c/ Bits B4-B7 specify either a number of line feeds from 0 to 15, or a track number on the format tape. As noted at a/ above, tracks 0, 9, and 12-15 are illegal.

4.3/

4.3 The Printer Character Set

The EBCDIC codes of the printable characters are given below:

Second Hexadecimal digit

First Hexadecimal digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4	Sp										£	.	<	(	+	|
5	&										!	$	*	)	;	¬
6	-	/									^	,	%	...	>	?
7											:	#	@	'	=	"
8																
9																
A																
B																
C		A	B	C	D	E	F	G	H	I						
D		J	K	L	M	N	O	P	Q	R						
E	Bl		S	T	U	V	W	X	Y	Z						
F	0	1	2	3	4	5	6	7	8	9						◊

Neither blank (E0) nor space (40) causes printing of a character, but these two codes are both legal. Any code not listed above is illegal.

SECTION N: CARD READER

The card reader, type 4512, is fully described in PS 4.8.30. Unless otherwise stated, all references in this summary are to the decimal numbering used in the product specification.

1. Introduction

The unit will read 80-column cards at a rate of 800 cards per minute. Data may be read in System 4 card code or in binary image mode. When the standard card code is being used, a check is made for invalid characters, then the card code is converted by hardware to the standard internal code.

2. After being read, a card is directed into either the normal stacker or into the reject stacker. The selection may be program controlled, priority being as follows:

a/ the reader logic selects the normal stacker for all error-free documents.

b/ the reader logic selects the reject stacker in case of error. Possible errors are read error, invalid punching, service request not honoured, and all inoperable conditions (e.g. double feed. See 4.5.1.2.)

c/ program stacker selection has priority over reader logic selection except in the case of error conditions which render the device inoperable.

d/ program stacker selection is only effective if the "select" command is given within 36 msec. of completion of reading; otherwise the reader logic prevails. (See 3.0, and see also 4.4.3 below.)

4. Programming

4.2 Three commands are used, viz.,

Sense	01	(hexadecimal)
Read	05	
Write control	07	

(Note The two commands "send status" and "no op" mentioned in PS 4.8.30 are hardware-originated and should not be used by the programmer.)

4.3.1/

4.3.1 Standard mode

The 12-bit card code (See PS 4.6.2) is translated into 8-bit internal code characters. The punched code is checked for validity.

4.3.2 Binary Image Mode

Each 12-bit column is transferred as two characters; B0 and B1 are zero in both bytes, and card rows 12, 11, 0 - 3 are transferred in the first byte, rows 4 - 9 in the second. No checking is performed on the punched data.

4.4.1 Sense (See also 4.5)

The secondary status ~~character~~ byte is input. The bits of this ~~character~~ byte have the following meanings:

B0 = 1 A reader error has occurred.

B1 = 1 Service request not honoured.

B2 = 1 Stacker selection late. An attempt by the program to select a particular stacker was made too late (See 2d above) to take effect. This bit is cleared by a "sense" command.

B3 = 1 An invalid punch code was read (standard mode only). Any invalid character is transferred as FF, and the card is directed to the reject stacker.

B4 = 1 Card jam, double feed, etc.

B5 = 1 Reader set to MANUAL.

B6 Not used.

B7 = 1 An illegal operation (e.g. write) was attempted.

4.4.2 Read

Information is read from the card into store.

4.4.3/

4.4.3 Write Control

Causes a control character to be sent to the reader. The control character (hexadecimal) has the following meaning:

01	Select normal stacker
02	Select reject stacker
04	Set standard mode
08	Set binary image mode

Sensible combinations, e.g. 06 = set standard mode and select reject stacker, may also be used.

Once a data mode (standard or binary) has been set, all subse-:quent "read" commands will be executed in this mode until a new mode is set.

On the other hand, stacker selection applies only to the single card just read. As noted above, the selection must be made within 36 msec. of reading the card.

SECTION P: CARD PUNCH

The card punch is fully described in PS 4.8.40. Unless otherwise stated, all references in this summary are to the decimal numbering used in the product specification.

1. Introduction

The mechanism punches 80-column cards at 100 cards per minute continuously. Output data may be punched in System 4 card code or in a binary image code.

As each card is being punched, its predecessor is being check-read. If an error is thus detected, the faulty card and the card currently being punched will both be offset by approximately $\frac{1}{4}$" in the stacker. Clearly there must be two cards worth of data always available in store if this error condition is to be corrected. (See 1.5 and 4.6.8)

At "end of file" the program must perform two dummy punching operations to ensure that the last card punched containing information successfully reaches the output stacker.

2. Once a card cycle has been initiated, the machine cannot stop until the cycle is completed. The punch will stop if a write command is not received for 15 seconds, requiring about 1/5 second to restart.

4. Programming

4.1 Output to the punch may be from either a selector or a multiplexor channel.

4.2 Three commands are used, viz.,

Sense	01	(hexadecimal)
Write	03	
Write Control	07	

(Note The two commands "send status" and "no op" mentioned in PS 4.8.40 are hardware-originated, and should not be used by the pro-:grammer.)

4.2.1/

4.2.1 Sense (See also 4.6)

The secondary status character is input. The bits of this character have the following meanings:

B0=1 The card currently being check-read is faulty. Two cards (the current one and its predecessor) will be offset in the stacker.

B1=1 Parity error while punching. The current card will be offset when it reaches the stacker.

B2 Not used.

B3=1 Parity error while filling buffer. Nothing is punched; the write command is terminated.

B4=1 Intervention needed, for example,

a/ stacker full

b/ hopper empty

c/ chad box full

B5=1 Punch set on MANUAL.

B6 Not used.

B7=1 Illegal operation (e.g. reading) attempted.

4.2.2 Write

Data is transferred from store to the punch buffer. When the buffer is full, the punch cycle is initiated. Alternatively, if the data transferred is less than a buffer-full, the remainder of the buffer is filled with space characters, and the punch cycle initiated.

There are two punching modes determined by the Write Control character. (See below, 4.2.3)

Write Encode Mode

Characters are punched according to the card code given in PS 4.6.2.

Binary/

Binary Image Mode

B0 and B1 of each character transferred are discarded. The resulting 6-bit characters are punched on the card in the order ...

top of column 1

bottom of column 1

top of column 2

bottom of column 2 and so on.

4.2.3 Write Control

Causes the transfer of a control character to the punch. The control character (hexadecimal) should be

01 Set encode mode

or 02 Set binary image mode.

4.2.6 Halt Device

If sent during buffer loading time, the buffer is cleared and no punching takes place. Otherwise the current punch cycle is completed.

SECTION Q: THE PAPER TAPE READER

The paper tape reader is fully described in PS 4.8.10. Unless otherwise stated, the decimal numbering of this summary refers to the numbering used in the product specification.

1. Introduction

The paper tape reader will read paper tape of width 11/16", 7/8", or 1" (i.e. 5, 7, or 8 track tape) at a nominal rate of 1250 characters/second. 1" tape punched with 7-bit ISO characters and an even parity bit is regarded as standard for the system. Two reading modes are used, a standard mode in which ISO characters are automatically converted to EBCDIC, and a binary mode.

2.4 Reading Modes

The following switches are located on the control panel of the reader:

(i) standard / non-standard

(ii) parity check : odd / even / none

(iii) read all chars. / ignore blank / ignore delete / ignore blank and delete.

Standard mode: When switch (i) is set to 'standard', the device expects to read 1" tape punched in ISO 7-bit code, the eighth track being an even parity track. The tape is checked for even parity, switch (ii) being ignored. Blank and delete characters (00 and FF) are ignored, regardless of the setting of switch (iii). The ISO characters are converted to equivalent EBCDIC characters on input. Reading terminates either in the normal way, or else if a 'record separator' character is read.

Non-standard (binary) mode: When switch (i) is set to 'non-standard', then either 5, 7, or 8 track tape may be read. If tracks are numbered

8 7 6 5 4 S 3 2 1 (S = sprocket)

then each character read is transferred as one byte, with track 8 going to B0, track 7 to B1, and so on. When reading 5 or 7 track tape, the unused most significant bits are set to zero. The parity of the tape may be checked if required/

required, the check applied being selected by switch (ii). Furthermore, either blank or delete characters or both may be optionally skipped, depending on the setting of switch (iii).

In both standard and non-standard modes, after a tape has been loaded or reloaded, leading blank characters are ignored, the first character read being the first non-blank character.

The switches mentioned above are the only means of altering the reading mode. A program can neither alter the setting of the switches, nor even detect what position the switches are in.

4.2 Programming

The following commands may be addressed to the reader:

sense	01 (hexadecimal)
read (store backwards)	02
read	05

(The commands 'send status', etc., mentioned in PS 4.8.10 are hardware-originated, and should not be used by the programmer.)

Sense

One secondary status byte is input from the reader to store. The bits of this byte have the following significance:

B0 = 1 A wrong-parity character, or (in standard mode) an invalid character was read. The character will have been transferred as FF.

B1, B2, B3 Always zero.

B4 = 1 'No tape' or 'taut tape' was detected. The reader stops immediately either of these faults occurs.

B5 = 1 Reader set to MANUAL.

B6 Always zero.

B7 = 1 An illegal operation, e.g. 'write', was attempted.

Read/

Read

Data is transferred from the tape reader to main store. Successive characters are written into store in ascending order of address. If a wrong-parity character is read, or if (in standard mode) an invalid ISO character is read, it is transferred to store as FF. After tape loading or reloading, initial blanks are always ignored; otherwise blank and delete characters may be ignored as described in 2.4 above.

In standard mode only, when a 'record separator' character is read from tape, it is transferred to store, the 'read' operation then being terminated.

Read (Store Backwards)

Exactly the same as a 'read' operation, except that successive characters are written into store in descending, rather than ascending, order of address.

Needless to say, this command does not cause the paper tape to move backwards through the reader.

SECTION R: THE PAPER TAPE PUNCH

The paper tape punch is fully described in PS 4.8.20 Unless otherwise stated, the decimal numbering system used in this summary refers to the numbering used in the product specification.

1. Introduction

The paper tape punch for System 4 processors will punch tapes of nominal width $\frac{7}{8}$", 11/16", or 1", the latter tape being regarded as the standard tape for the system. Data may be punched at a nominal rate of 150 characters per second into 5, 7, or 8 tracks as required. No check is made of the punched data. Two punching modes are provided, a standard mode in which ISO characters are punched, and a binary mode.

2.4 Punch Mode

(i) Standard Mode: In the standard punching mode, 8-bit EBCDIC characters from main store are converted to 7-bit ISO characters, an even parity bit is added, and the resulting 8-bit character is punched in 1" tape. A check is made of the validity of the EBCDIC characters. Data transfer ends either in the normal way, or else when a "record separator" character is punched.

(ii) Binary Mode: The relevant bits of a byte from store are copied directly to the tape. If the tape tracks are numbered

8 7 6 5 4 S 3 2 1

where S represents the sprocket hole, then B0 of the byte is copied to track 8, B1 is copied to track 7, and so on. If 5- or 7-track tape is used, then B0-2, or B0 only are ignored.

The mode of punching in use is determined by a switch on the operators' control panel. The program has no means of selecting the mode of punching to be used. Moreover, the program cannot detect which punch mode is being used.

4./

4. Programming

The following commands may be used:

sense	01	(hexadecimal)
write	03	

(The commands "send status", etc., mentioned in PS 4.8.20 are hardware-originated and should not be used by the programmer.)

4.2.1 Sense

One secondary status byte is transferred from the punch to main store. The bits of this byte have the following significance:

B0 = 1 A parity error has occurred while data was being sent from main store to the device.

B1) B2) Always zero

B3 = 1 An invalid EBCDIC code was sent to the device while in standard mode. The invalid character will have been punched as FF.

B4 = 1 'No tape' or 'taut tape' was detected. Punching stops immed-:iately one of these conditions is found.

B5 = 1 The punch is set to MANUAL.

B6 = 1 Low tape warning. The punch is down to its last few feet of tape.

B7 = 1 An illegal operation, e.g. 'read', was attempted on the punch.

4.2.2 Write

Data is transferred from ascending addresses in main store to the punch. In standard mode only:

a/ the validity of each character transferred is checked, a faulty character being punched as FF;

b/ the transfer of data ends if a "record separator" character is sent to the punch. (This character will itself be punched however).

No check is made that the tape has been punched correctly. Punching ends if the punch runs out of tape.

SECTION S: MULTIPLEXOR COMMUNICATIONS CONTROL UNIT

More information on this will be coming; this section outlines the probable characteristics of the MCCU.

The transfer of information from and to devices attached to the MCCU is exactly similar to ordinary devices; it is done by means of Channel Command Words in the same way.

One of the MCCU lines, the Co-ordinations Message Line, is reserved for a special purpose and does not have a device attached to it. It is used for two purposes:

i) The MCCU allows to be attached to it a variety of devices which have different electrical interfaces with the MCCU. The MCCU must have stored within it for each device details of this interface. There is core store provided in the MCCU for this purpose, which must be primed by the CPU. This priming is done by means of Write commands along the Co-ordination Message Line.

ii) A device attached to the MCCU is different from an ordinary device in that it must be able to interrupt the CPU in certain circumstances. This is done as follows: the software arranges that a read is called on the Co-ordination Message Line (CML) by setting up a CCW with the Program Controlled Interrupt bit set. The MCCU determines when a device requires the CPU to be interrupted and completes the read on the CML.

It will be possible to build into the MCCU a set of up to 13 characters (or groups of up to 4 characters), for each device attached to it so that when the corresponding device inputs any one of these characters an interrupt to the CPU is caused. These 13 characters are built into the hardware. However, it is probable that the software will be able to decide that interrupts occur only when one of a subset of these 13 characters is input.

To allow interrupt of the CPU from a dormant device attached to the MCCU, the software could always have a one character read set up on the device. The input/

input of the character will cause an interrupt. However, the MCCU may be built so that whenever a character is input from a device upon which no read is called, it will cause the CPU to be interrupted by completing the read on the CML.