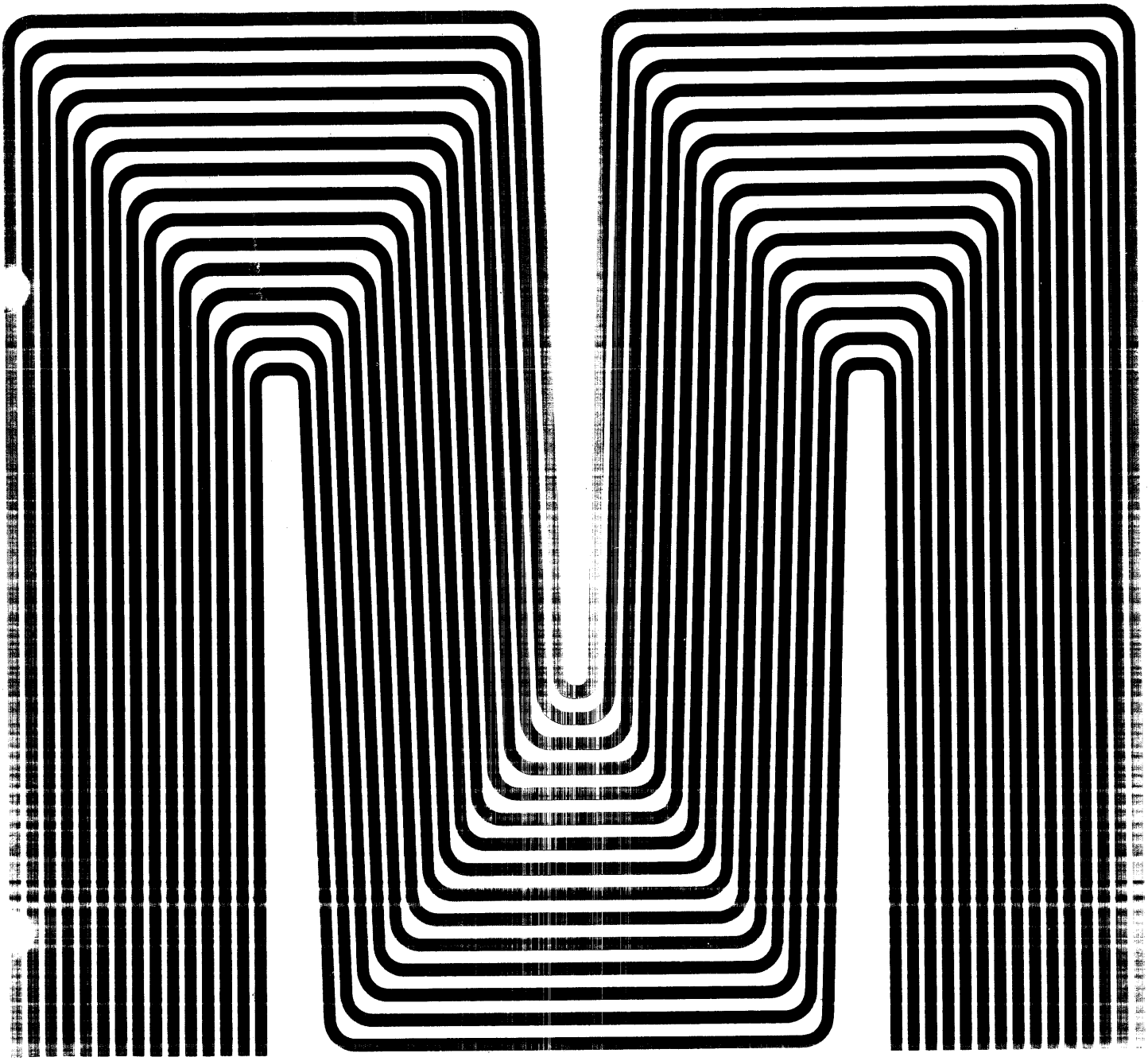
Microdata

COMPUTER REFERENCE MANUAL

Micro 1600/21
Micro 821



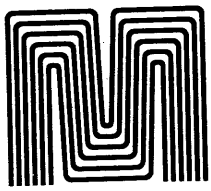
COMPUTER REFERENCE MANUAL

Micro 1600/21
Micro 821

71-1-821-001

August 1971

Microdata



Microdata Corporation
644 East Young Street
Santa Ana, California 92705

© 1971 Microdata Corporation

Printed in USA

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

APPENDIXES

TABLES

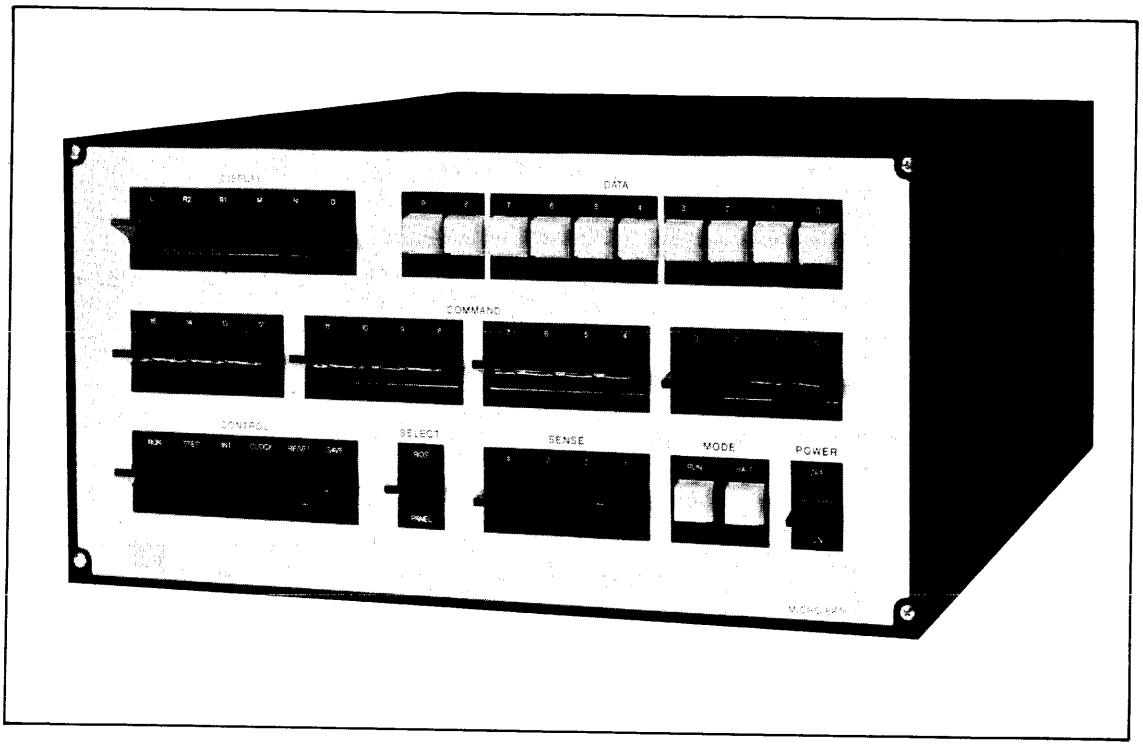


Figure 1. MICRO 821 Computer With Systems Panel

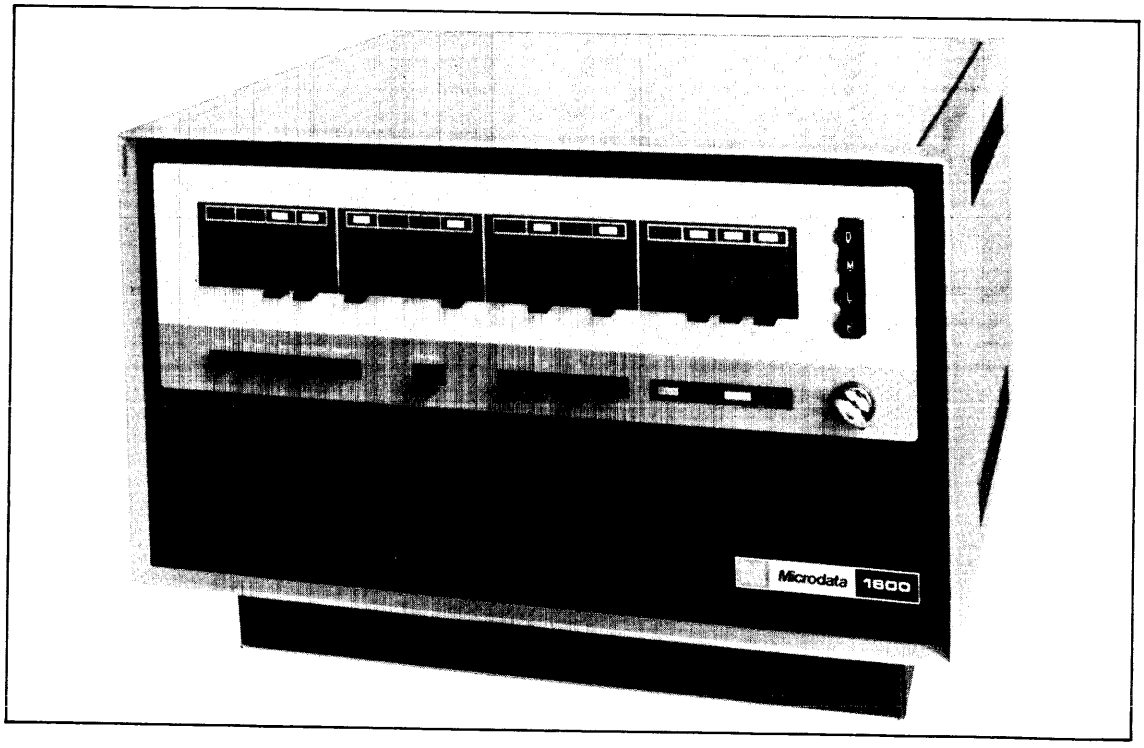


Figure 2. MICRO 1600/21 Computer With Systems Panel

SYSTEM DESIGN FEATURES

The MICRO 1600/21 and MICRO 821 are high-speed microprogrammed general-purpose computers which provide a comprehensive instruction repertoire and powerful input/output facilities.

The two computers are directly program compatible, with the major difference being that the MICRO 1600/21 features about a 10 percent faster execution time than the MICRO 821.

System architecture of both computers is byte-oriented, allowing precision operations and character manipulation to be highly efficient in speed and memory utilization.

Superior price/performance of the MICRO 821 and 1600/21 in terms of efficient core utilization and high throughput is achieved through the availability of powerful macro instructions. Both systems use TTL monolithic integrated circuits, including a large number of medium and large-scale integration types. The use of read-only memories for control greatly reduces the number of circuits that otherwise would be required to provide comparable functions.

Modular design of core memory, read-only memory, processor options and input/output elements permits inexpensive system expansion within the compact basic enclosure.

Basic models of the MICRO 800 and 1600 series of computers differ mainly in mechanical configuration (see Figures 1 and 2).

The MICRO 800 features flexibility, functional modularity and system-oriented packaging which make it ideally suited for dedicated volume applications and permit the computer to be expanded or reduced to the exact configuration needed for any application.

The MICRO 1600 family is the newest and most advanced Microdata product, designed as a companion line to the MICRO 800 and featuring improvements in speed and function. Both the 1600 and 800 are functionally compatible, enabling established MICRO 800 users to use the 1600 directly without redevelopment of firmware, software or system peripherals or interfaces. However, new and revised firmware can achieve significant performance improvements in the MICRO 1600 at both the micro and macro levels of programming.

GENERAL CHARACTERISTICS

The features and characteristics of the MICRO 1600/21 and MICRO 821 include:

- Variable precision operations
- Character/string manipulation

- Stack processing
- Memory addressing to 32,768 bytes

  4096 and 8192 byte plug-in memory modules

  32,768 bytes of memory in basic enclosure

  1 microsecond memory cycle time (1.1 microseconds for the 821)
- Six operational registers

  Accumulator (A) – 16 bits

  Auxiliary accumulator (B) – 16 bits

  Index register (X) – 16 bits

  Program counter (P) – 15 bits

  Overflow (O) – 1 bit

  Word length control (W) – 2 bits
- Extensive, powerful instruction set including 107 individual operations:

  Control (16)

  Multi-bit arithmetic and logical shifts (12)

  Conditional jumps (17)

  Input/Output (6)

  Inter-register (19)

  Stack control (8)

  Character/string manipulation (5)

  Multiply/Divide (2)

  Decimal arithmetic (add and subtract instructions) (2)

  Memory reference including jump, compare and variable word length operations (20)
- Eight operand addressing modes including:

  Direct to page 0 (first 256 bytes)

  Direct relative to P (± 128 bytes)

Indirect to page 0 (first 256 bytes)

Indirect relative to P (±128 bytes)

Indexed (to 32,768 bytes)

Indexed with bias (to 32,768 bytes)

Extended address (to 32,768 bytes)

Literal

- Multi-precision 1, 2, 3, or 4-byte load, store, and arithmetic operations
- Flexible I/O facilities including:

  Programmed transfers to/from A register, B register and memory

  Concurrent buffered I/O

  Direct memory access
- Expandable priority interrupt system
- Processor options including:

  Real-time clock

  Power-fail detect and automatic restart (standard on 1600/21)
- Built-in bootstrap loader in non-volatile read-only memory
- Standard supplied software including:

  loaders

  teletype debug and operating system

  two-pass assembler

  text editor

  diagnostics
- TTL integrated circuitry
- Power: 115/230 vac, 50-60 cycle, 380 watts
- Environment: 0-50° C

SYSTEM ORGANIZATION

Basic elements of both computers include the operational registers, core memory, interrupt system, input/output system, and control console. A group of processor options is also available to meet a broad range of special system requirements.

REGISTERS

Both computers contain six operational registers which are accessible to the programmer. These operational registers occupy nine of the 16 file registers in the basic microprogrammable hardware; the remaining seven file registers are used for internal operation and are not accessible to the programmer. The assignment of the file registers is given in Appendix A.

A Register

The 16-bit A register is the accumulator with which most operations are performed. The A register holds the upper portion of 24- or 32-bit data words and all of 8- and 16-bit data words. The A register may be shifted by itself or in conjunction with the B register.

B Register

The 16-bit B register is the auxiliary accumulator and is used mainly as an extension of the accumulator to hold the lower 16 bits of 24- and 32-bit data. The B register may be shifted by itself or in conjunction with the A register.

X Register

The 16-bit X register is an index register used in address modification. It can communicate directly with memory, be operated on arithmetically, and compared with the A register.

P Register

The 15-bit P register is the program counter which holds the address of next memory instruction to be executed.

W Register

The 2-bit W register holds the word length mode. It is loaded by a control instruction and sets the byte length of the operand for all variable word length instructions.

O Register

The one-bit O register holds the overflow flag. The overflow is set by arithmetic instructions when an overflow occurs, or by execution of a Control instruction. It may be reset by execution of a Control instruction or by a Conditional Jump instruction that tests for an overflow condition.

CORE MEMORY

The magnetic core memory is organized into pluggable modules of 4096 or 8192 bytes. The memory is byte addressable. Each byte contains eight information bits.

The core memory may be expanded up to 32,768 bytes (four 8192 byte modules) within the basic enclosure. The memory cycle time is 1 and 1.1 microseconds respectively on the 1600/21 and 821.

The direct memory access (DMA) selector channel option allows for interfacing peripheral devices directly with the memory to provide peak transfer rates of up to 1,000,000 and 909,000 bytes per second respectively.

INTERRUPTS

The priority interrupt system provides for internal processor interrupts, I/O peripheral device interrupts, and groups of individual external interrupts, each with its own unique interrupt memory address and priority assignment.

Internal Interrupts

Internal interrupts include those that are supplied as part of the basic system as well as optional features. The internal interrupts have priority over external interrupts and are listed below in order of their priority, with the lowest listed first.

Console. The standard console interrupt is triggered by a switch on the console, allowing an operator to exert control. This interrupt routine also is used by the trap instruction.

DMA Termination. The DMA termination interrupt occurs when a direct memory access channel has reached a terminal condition and is requesting software attention.

Real-Time Clock. The real-time clock interrupt occurs when a preset clock count in a unique memory location is incremented to zero. The clock count location is automatically advanced at each clock time. The real-time clock interrupt is enabled and disabled under program control.

Power-Fail. The power-fail interrupt provides an interrupt when a loss of primary power is detected. A minimum of one millisecond of computer operation is assured after the interrupt.

Programming Note: The following three instructions must be the first instructions of any power-fail subroutine. This will remove a microprogram set, interrupt lock-out flag from the push-down stack. Failure to remove it would inhibit the recognition of any interrupt following a power restart.

```
PWR     LDA*    X'8C'       Pick Up φv/w
        ANA=    X'7FFF'     Remove Flag
        STA*    X'8C'       Put Back
```

Power-Restart. Power-restart interrupt occurs when the power is applied and is up to normal operating levels and the processor placed in the run mode.

External Interrupts

External interrupts may be associated with peripheral devices or may be individual lines not associated with devices on the I/O bus. The device interrupts are used to indicate such conditions as data ready, error and end of operation conditions in the device. These interrupts are enabled by sending a function code to the device controllers. The memory location containing the interrupt routine address is 100_{16} plus twice the device address.

Individual interrupts may be handled by an external interrupt module which provides for arming/disarming individual interrupts and enabling/disabling recognition of interrupts in the group. Standard external interrupt cards containing 8 priority interrupt lines are available. A total of 64 external interrupts can be implemented.

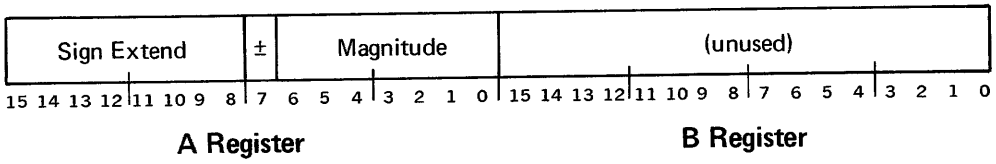
INFORMATION FORMAT

The basic element of information is an 8-bit byte in which the bit positions are numbered from 7 through 0, left to right. Both instructions and data occupy a variable number of bytes for maximum storage efficiency. A word is a 16-bit element of information consisting of two bytes. The accumulator and index register both hold a 16-bit word.

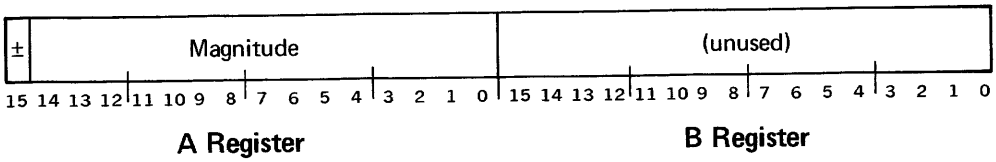
Data Format

Data is variable precision of 8, 16, 24, or 32-bit length. Negative numbers are represented in 2's complement form. The range of magnitude and data format in the A and B registers for the four data lengths is shown as follows:

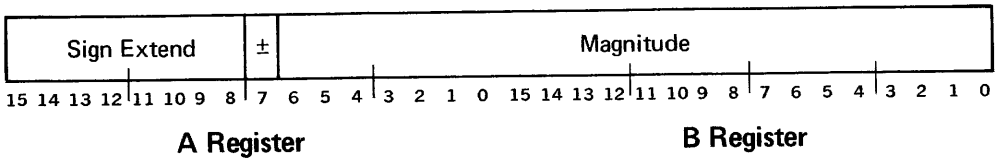
8 Bits (1 Byte) — Range: $+2^7-1$ to -2^7



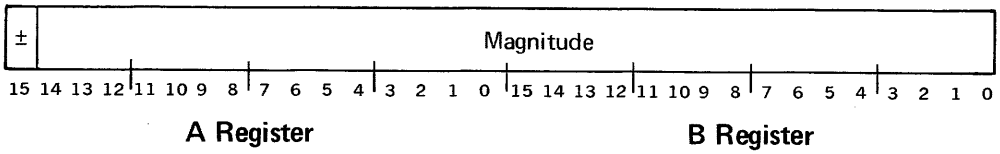
16 Bits (2 Bytes) — Range: $+2^{15}-1$ to -2^{15}



24 Bits (3 Bytes) — Range: $+2^{23}-1$ to -2^{23}

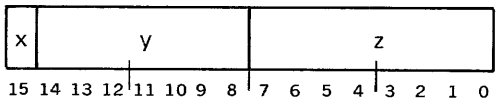


32 Bits (4 Bytes) — Range: $+2^{31}-1$ to -2^{31}



Address Word Format

A 16-bit address word contains a 15-bit memory address and an index flag as shown below. The address may be direct or indirect address as dictated by the instruction operation code. The value of the address word is equal to the contents of bits 14-0 and is equal to the contents of bits 14-0 plus the contents of the X register if bit 15 is a 1-bit.



Instruction Format

Instruction formats are one to five bytes, but in all cases the first contains an eight-bit operation code which defines the operation class, the sub-operation code, and any modifiers. Succeeding byte(s) contain such information as:

Single byte absolute or relative address

Double byte address word

Single byte shift count

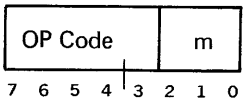
Single byte I/O function and device address

1, 2, 3, or 4 byte literal data.

OPERAND ADDRESSING MODES

The memory reference instructions defined in the following section each have eight possible modes of addressing an operand in memory. The number of bytes in the instruction format varies with the mode. The additional bytes of the instruction contain addresses, partial addresses, or data (literals).

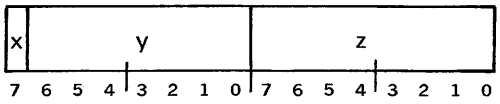
The basic memory reference instruction is one byte containing two fields as follows:



The 5-bit operation code defines the basic instruction; the 3-bit m field specifies the address mode. Additional bytes contain the address of an operand, an indirect address, a base address, or a literal depending upon the addressing mode. The effective operand address is the memory location specified after all indirect and/or index modifications have been performed.

When an indirect address mode is specified, the location of the indirect address word is the first byte of a two-byte word having the format shown below:

Indirect Address Word Format



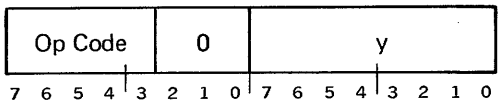
Bit 7 of the first byte (x) defines whether or not the indirect address word will be modified by the contents of the index register:

If x = 0, the 15-bit number formed by y and z is the effective operand address.

If x = 1, the 15-bit number formed by y and z is a base address to which is added the contents of the X register. The result is the effective operand address.

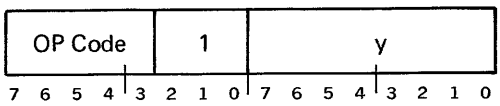
The individual addressing modes and the memory reference instruction format for that mode are defined below.

Direct Page 0 (m=0)



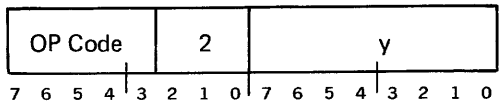
The effective operand address is given by the contents of the second byte of the instruction (y) with seven high order zero bits appended. This mode provides direct addressing of operands in the first 256 memory locations.

Direct Relative (m=1)



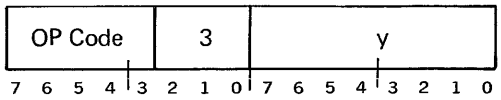
The effective operand address is given by the sum of the contents of the second byte (y) with its high order sign bit (bit 7) extended and the contents of the P register. The contents of the P register at the time the addition is performed is the address of the memory location following y. This mode provides for addressing from 127 locations ahead to 128 locations behind the memory location of the next instruction.

Indirect Page 0 (m=2)



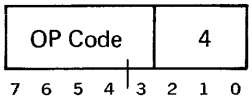
An indirect address word is specified by the contents of the second byte (y) of the instruction with seven high order zero bits appended. The 2-byte indirect address word addressed is located in the first 256 memory locations. The effective operand address is given by the contents of the indirect address word if the index flag (bit 15) is a 0-bit, or by the sum of the contents of the indirect address word and the X register if the index flag (bit 15) is a 1-bit.

Indirect Relative (m=3)



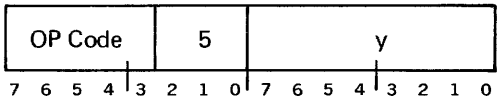
A relative indirect address word is specified by the sum of the contents of the second byte (y) with its high order bit (bit 7) extended and the contents of the P register. The contents of the P register at the time the addition is performed is the address of the memory location following y. The effective operand address is given by the contents of indirect address word if the index flag (bit 15) is a 0-bit or by the sum of the contents of the indirect address word and the X register if the index flag (bit 15) is a 1-bit.

Indexed (m=4)



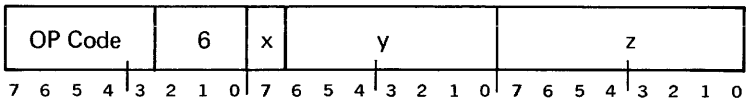
The effective operand address is given by the contents of the X register.

Indexed With Bias (m=5)



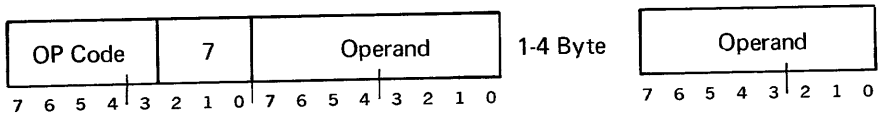
The effective operand address is given by the sum of the contents of the X register and the contents of the second byte (y) of the instruction.

Extended Address (m=6)



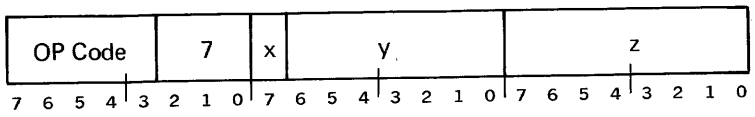
A 16-bit address word is located in the second and third byte of the instruction. The effective operand address is given by the contents of the address word if the index flag bit in bit 15 is a 0-bit, or by the sum of the contents of the address word and the X register if the index flag is a 1-bit.

Literal (m=7)



The effective operand address is given by the contents of the P register. The operand is located in from 1-4 bytes following the first byte of the instruction, depending upon the operand precision. The P register is incremented for each operand byte accessed. The Jump and Return Jump memory referencing instructions do not have a literal mode.

Jump/Return Jump Indirect Extended Address (m=7)



A 16-bit direct address word is located in the second and third bytes of the instruction. This word addresses an indirect address word located at the address given by the contents of the second and third bytes if bit 15 of the address word is a 0-bit or by the sum of the contents of the second and third bytes and the X register is the index flag bit in bit 15 is a 1-bit.

The effective jump address is given by the contents of the indirect address word if the index flag in bit 15 of the indirect address word is a 0-bit, or by the sum of the contents of the indirect word and the X register if the index flag bit in bit 15 of the indirect address word is a 1-bit.

Table 1. Effective Address Computation

M	Effective Address		Mode
0		y	Direct Page 0
1		y+(P)	Direct Relative
2		(y)	Indirect Page 0
3		(y+(P))	Indirect Relative
4		(X)	Indexed
5		y+(X)	Indexed with Bias
6	x=0: x=1:	y,z y,z+(X)	Extended Address Extended Address Indexed
7		(P)	Literal
7	x=0:	(y,z)	Indirect Extended Address (Jump and Return Jump only)
	x=1:	(y,z+(X))	Indirect Extended Address Indexed (Jump and Return Jump only)

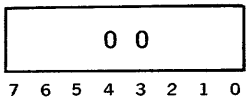
INSTRUCTION REPERTOIRE

This section contains descriptions of all instructions except input/output, described later. With each description is a diagram showing the format of the instruction and its operation code, normally given in hexadecimal. Above each diagram are the mnemonic code and the name of the instruction, followed by a list of the registers and indicators that can be affected by the instruction. The timing of each instruction is given in Appendixes C and D.

CONTROL

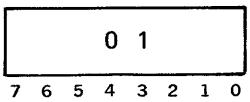
The control group of instructions are single byte instructions which provide specific control functions.

HLT Halt



The processor, and concurrent I/O are halted. The contents of the P register will be the address of the halt instruction plus one. Depressing the console run or step switches will cause the next instruction to be executed. Interrupts or concurrent I/O requests cannot be recognized before the execution of the next instruction.

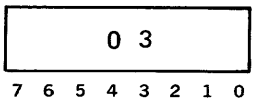
TRP Trap



The contents of the P register are stored at the two-byte memory location specified by the two-byte address word at location 80_{16}. Subsequently, the two-byte address word (at 80_{16}) plus two replaces the original contents of the P register. Execution of this instruction is the same as depressing the console interrupt switch. Interrupts or concurrent I/O requests cannot be recognized before the execution of the next instruction.

Affected: P, Memory.

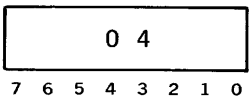
ESW **Enter Sense Switches**



The status of the four console sense switches is placed in bits 15-12 of the A register. If the sense switch is on, the corresponding bit in the A register will be set to one. Bits 8-11 of the A register are set to one and bits 0-7 are unaltered.

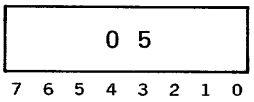
Affected: A (high order 8 bits)

DIN **Disable Interrupt System**



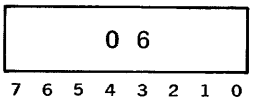
All external interrupts are disabled, preventing the processor from recognizing an external interrupt request. Interrupts are saved in the disabled state. Internal interrupts or concurrent I/O requests cannot be recognized before the execution of the next instruction.

EIN **Enable Interrupt System**



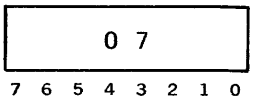
All external interrupts are enabled, allowing the processor to recognize an external interrupt. Interrupts or concurrent I/O requests cannot be recognized before the execution of the next instruction.

DRT **Disable Real-Time Clock**



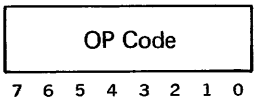
The updating of the real-time clock memory location and the generation of real-time clock interrupts are inhibited. Interrupts or concurrent I/O requests cannot be recognized before the execution of the next instruction.

ERT **Enable Real-Time Clock**



The updating of the real-time clock memory location and the generation of real-time clock interrupts are enabled. Interrupts or concurrent I/O requests cannot be recognized before the execution of the next instruction.

Reset Overflow and Set Word Length

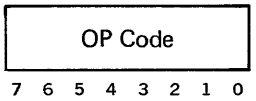


The Overflow register is reset and the variable precision mode (byte length) is placed in the W register. The four instructions are as follows:

OP Code	Mnemonic	Instructions
08	RO1	– RESET OVERFLOW AND SET WORD LENGTH TO 1
09	RO2	– RESET OVERFLOW AND SET WORD LENGTH TO 2
0A	RO3	– RESET OVERFLOW AND SET WORD LENGTH TO 3
0B	RO4	– RESET OVERFLOW AND SET WORD LENGTH TO 4

Affected: O, W

Set Overflow and Set Word Length

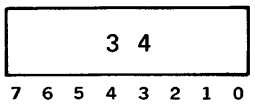


The overflow register is set to one and the variable precision mode (byte length) is placed in the W register. The four instructions are as follows:

OP Code	Mnemonic	Instructions
0C	SO1	– SET OVERFLOW AND SET WORD LENGTH TO 1
0D	SO2	– SET OVERFLOW AND SET WORD LENGTH TO 2
0E	SO3	– SET OVERFLOW AND SET WORD LENGTH TO 3
0F	SO4	– SET OVERFLOW AND SET WORD LENGTH TO 4

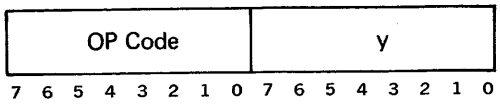
Affected: O, W

NOP No Operation



This instruction performs no operation.

Conditional Jumps



The conditional jump instructions are a two byte format. The first byte provides the operation code which includes the condition being tested (bits 2-0) and whether the jump will be made on the condition being true or false (bit 3). The second byte contains an 8-bit signed value, y, which specifies a jump location relative to P.

If the test condition is met, the sum of the contents of the second byte (y) with its high order bit extended and the current contents of the P register are placed in the P register; otherwise the P register remains unaltered and the next instruction in sequence is accessed. The contents of the P register at the time of addition is the address of the next instruction. The instructions which test the overflow condition also reset the overflow register.

The conditional jump instructions, their operation codes and mnemonics follows:

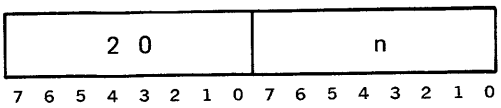
OP Code	Mnemonic	Instructions
10	JOV	JUMP IF OVERFLOW SET
11	JAZ	JUMP IF A EQUAL TO ZERO
12	JBZ	JUMP IF B EQUAL TO ZERO
13	JXZ	JUMP IF X EQUAL TO ZERO
14	JAN	JUMP IF A NEGATIVE
15	JXN	JUMP IF X NEGATIVE
16	JAB	JUMP IF A EQUALS B
17	JAX	JUMP IF A EQUALS X
18	NOV	JUMP IF OVERFLOW NOT SET
19	NAZ	JUMP IF A NOT EQUAL TO ZERO
1A	NBZ	JUMP IF B NOT EQUAL TO ZERO
1B	NXZ	JUMP IF X NOT EQUAL TO ZERO
1C	NAN	JUMP IF A NOT NEGATIVE
1D	NXN	JUMP IF X NOT NEGATIVE
1E	NAB	JUMP IF A NOT EQUAL TO B
1F	NAX	JUMP IF A NOT EQUAL TO X
5A	JEP	JUMP IF EVEN PARITY (the A Register contains an even number of "1" Bits)

Affected: P, O

SHIFTS

The shift group of instructions provides both arithmetic and logic shifts of A register, B register and A and B registers together. A signed shift count is specified in the second byte of the instructions. The shift count is any positive number from 0 to 127; if negative, a no operation results. A concurrent input/output request is acknowledged between bit shifts of all shift instructions. However, normal interrupts are not recognized until the end of the complete shift instruction. In addition, the response time to an external request should be considered when programming long bit shifts.

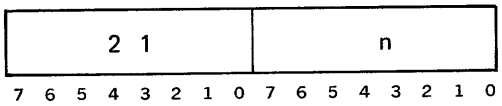
LLA Logical Left A



The contents of the A register are shifted n bits to the left. Bits shifted out of A_{15} are shifted into A_0.

Affected: A

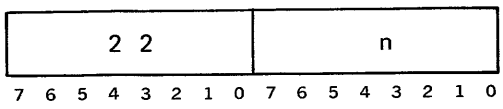
LLB Logical Left B



The contents of the B register are shifted n bits to the left. Bits shifted out of B_{15} are shifted into B_0.

Affected: B

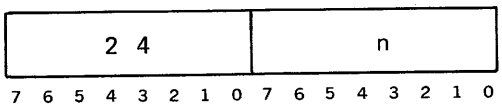
LLL Logical Left Long



The contents of the A and B registers are shifted n bits to the left. Bits shifted out of A_{15} are shifted into B_0. Bits shifted out of B_{15} are shifted into A_0.

Affected: A, B

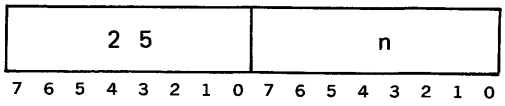
LRA Logical Right A



The contents of the A register are shifted n bits to the right. Zeros are shifted into A_{15}, and bits shifted out of A_0 are lost.

Affected: A

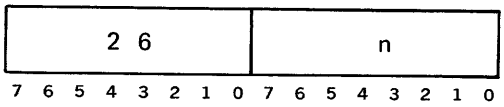
LRB Logical Right B



The contents of the B register are shifted n bits to the right. Zeros are shifted into B_{15}, and bits shifted out of B_0 are lost.

Affected: B

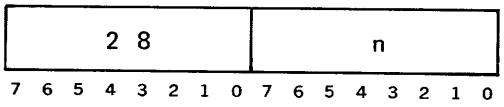
LRL Logical Right Long



The contents of the A and B registers are shifted n bits to the right. Zeros are shifted into A_{15}. Bits shifted out of A_0 are shifted into B_{15}, and bits shifted out of B_0 are lost.

Affected: A, B

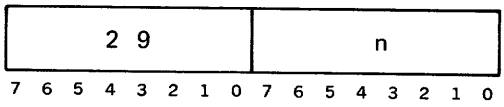
ALA Arithmetic Left A



The contents of the A register are shifted n bits to the left. Bits shifted out of A_{15} are lost. Zeros are shifted into A_0.

Affected: A

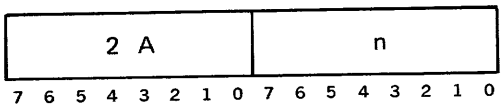
ALB Arithmetic Left B



The contents of the B register are shifted n bits to the left. Bits shifted out of B_{15} are lost. Zeros are shifted into B_0.

Affected: B

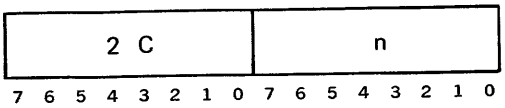
ALL **Arithmetic Left Long**



The contents of the A and B register are shifted n bits to the left. Bits shifted out of A_{15} are lost. Bits shifted out of B_{15} are shifted into A_0. Zeros are shifted into B_0.

Affected: A, B

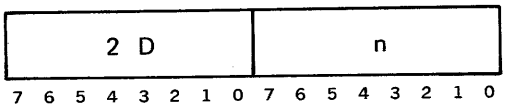
ARA **Arithmetic Right A**



The contents of the A register are shifted n bits to the right. The sign bit in A_{15} is copied into vacated high order bits. Bits shifted out of A_0 are lost.

Affected: A

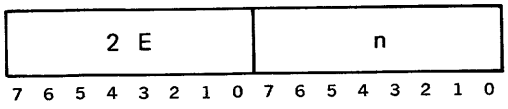
ARB **Arithmetic Right B**



The contents of the B register are shifted n bits to the right. The sign bit in B_{15} is copied into vacated high order bits. Bits shifted out of B_0 are lost.

Affected: B

ARL **Arithmetic Right Long**



The contents of the A and B registers are shifted n bits to the right. The sign bit in A_{15} is copies into vacated high order bits. Bits shifted out of A_0 are shifted into B_{15}, and bits shifted out of B_0 are lost.

Affected: A, B

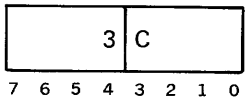
EXTENDED ARITHMETIC

Decimal numbers are strings of ASCII characters from 1 to 16 digits in length. The decimal digits zero to 9 are represented by the hexadecimal values B0 to B9. Hexadecimal values of A0 (Blank) or 00 will be treated as B0. The rightmost, or units position, digit of the number contains the sign of the number. If this digit is in the range of B0 to B9, the number is positive. When this digit is in the range of D0 to D9, the number is considered to be negative.

When performing decimal arithmetic operations, the B and X registers point to the leftmost, or high-order, digit of each operand. The lower eight bits of the A register contains two four bit values indicating the number of digits to the right that each operand extends. Bits 7-4 contain the field length for the B register and bits 3-0 contain the field length for the X register. The memory address, formed by the sum of a register and its initial field length, points to the units position of that operand. During decimal arithmetic operations, bit 15 of the B register is set to zero. This should be of no concern, since all valid memory addresses would have this bit set to zero anyway.

When the operation is an add, with signs opposite, or a subtract, with signs alike, and a carryout of the high-order digit does not occur, the result is not in true form. This initiates a recomplement cycle to tens complement the sum or difference.

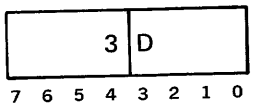
DAD Decimal Add



The variable length operand at the memory location given by the contents of the B register (ADDEND), is added decimally by digit (bytes) to the variable length operand at the memory location given by the contents of the X register (AUGEND) and the sum replaces the augend. If the addend is shorter than the augend, high-order zero digits are supplied. When the addend is longer than the augend, the sum will be correct if the extra high-order addend digits are zero. If the magnitude of the sum exceeds the field length to contain it, the overflow register will be set, otherwise it will be reset. The sign of the result is determined by the rules of algebra and is attached to the units position of the sum. A zero sum is always positive. When a high-order digit is lost because of an overflow, a zero result has the sign of the correct sum. After the operation, the content of the A register will be set to minus one, zero, or plus 255 to indicate the condition of the result as negative, zero, or positive. Interrupts and concurrent I/O requests may be serviced during the execution of this instruction.

Affected: A, O, Memory

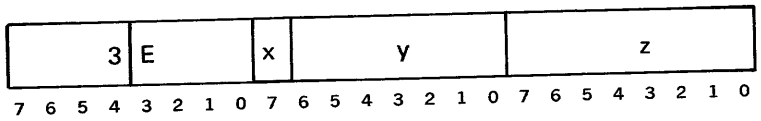
DSB **Decimal Subtract**



The variable length operand at the memory location given by the contents of the B register (SUBTRAHEND), is subtracted decimally by digit (byte) from the variable length operand at the memory location given by the contents of the X register (MINUEND) and the difference replaces the minuend. If the subtrahend is shorter than the minuend, high-order zero digits are supplied. When the subtrahend is longer than the minuend, the difference will be correct if the extra high-order subtrahend digits are zero. If the magnitude of the difference exceeds the field length to contain it, the overflow register will be set, otherwise it will be reset. The sign of the result is determined by the rules of algebra and is attached to the units position of the difference. A zero difference is always positive. When a high-order digit is lost because of an overflow, a zero result has the sign of the correct difference. After the operation, the content of the A register will be set to minus one, zero, or plus 255 to indicate the condition of the result as negative, zero, or positive. Interrupts and concurrent I/O requests may be serviced during the execution of this instruction.

Affected: A, O, Memory

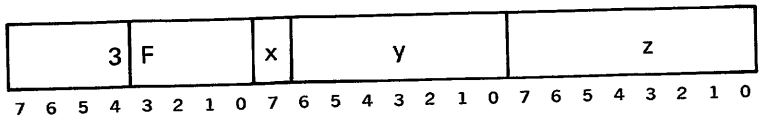
MUL **Multiply (Binary)**



The two-byte operand located at the effective memory address is multiplied by the contents of the A register and the product is placed in the A-B register. The multiply is an integer type and the 30 bit resultant magnitude occupies the 30 low order bits of A-B register and a double sign bit occupies the two high order bits. A concurrent I/O request can be serviced during the instruction execution.

Affected: A, B

DIV **Divide (Binary)**



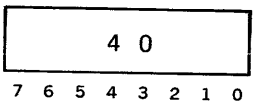
The contents of the A-B register is divided by the two byte operand located at the effective memory address. The signed quotient is placed in the B register and the signed remainder is placed in the A register. The remainder will have the same sign as the original dividend unless the remainder is zero. The divide is an integer type operation. If the relative magnitude of the original contents of the A-B register (dividend) and the operand (divisor) is such that the quotient would be greater than $2^{15}-1$ or less than -2^{15}, the overflow register is set. A concurrent I/O request can be serviced during the instruction execution.

Affected: A, B, O

REGISTER OPERATE

The register operate group of instructions provides for special arithmetic and logical operations on individual registers and between registers.

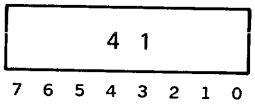
ORA OR B With A



The logical inclusive-OR of the contents of the A register and the contents of the B register are placed in the A register.

Affected: A

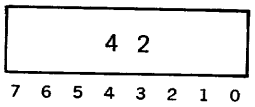
XRA Exclusive-OR B With A



The logical exclusive-OR of the contents of the A register and the contents of the B register are placed in the A register.

Affected: A

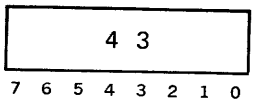
ORB OR A With B



The logical inclusive-OR of the contents of the A register and the contents of the B register are placed in the B register.

Affected: B

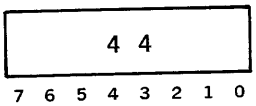
XRB Exclusive-OR A With B



The logical exclusive-OR of the contents of the A register and the contents of the B register are placed in the B register.

Affected: B

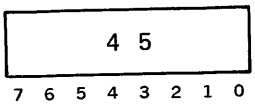
INX **Increment X**



The contents of the X register plus one replaces the contents of the X register. If the result is greater than $2^{15}-1$, the overflow register is set.

Affected: X, O

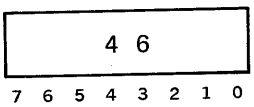
DCX **Decrement X**



The contents of the X register minus one replaces the contents of the X register. If the result is less than -2^{15}, the overflow register is set.

Affected: X, O

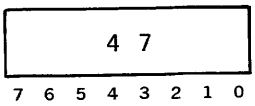
AWX **Add Word Length to X**



The contents of the W register plus one is added to the contents of the X register and the sum is placed in the X register. If the sum is greater than $2^{15}-1$, the overflow register is set.

Affected: X, O

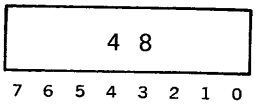
SWX **Subtract Word Length from X**



The contents of the W register plus one is subtracted from the contents of the X register and the difference is placed in the X register. If the difference is less than -2^{15}, the overflow register is set.

Affected: X, O

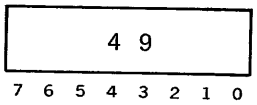
INA **Increment A**



The contents of the A register plus one replaces the contents of the A register. If the sum is greater than $2^{15}-1$, the overflow register is set.

Affected: A, O

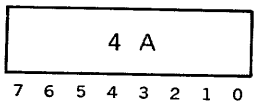
INB **Increment B**



The contents of the B register plus one replaces the contents of the B register. If the sum is greater than $2^{15}-1$, the overflow register is set.

Affected: B, O

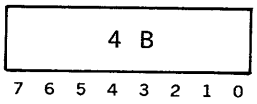
OCA **One's Complement A**



The one's complement of the contents of the A register replaces the contents of the A register.

Affected: A

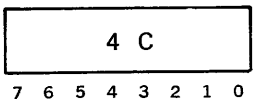
OCB **One's Complement B**



The one's complement of the contents of the B register replaces the contents of the B register.

Affected: B

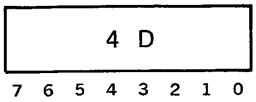
TAX Transfer A to X



The contents of the A register are placed in the X register.

Affected: X

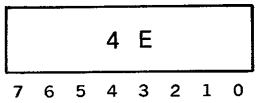
TBX Transfer B to X



The contents of the B register are placed in the X register.

Affected: X

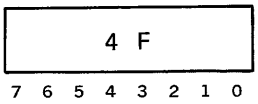
TXA Transfer X to A



The contents of the X register are placed in the A register.

Affected: A

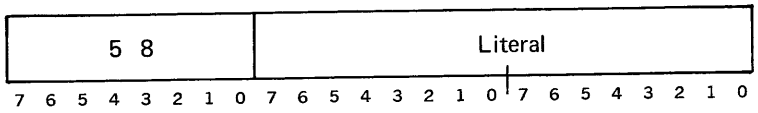
TXB Transfer X to B



The contents of the X register are placed in the B register.

Affected: B

MST Multiply Step

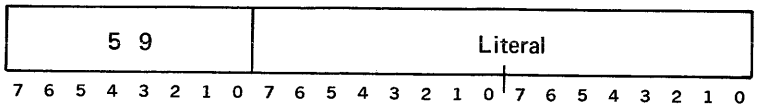


If the low order bit of the B register is a 1-bit, the 16-bit literal contained in the second and third bytes of the instruction is added to the contents of the A register and the contents of the A and B registers are shifted one bit to the right. If the low order bit of the B register is a 0-bit, the contents of the A and B registers are shifted one bit to the right without the

addition. Bits shifted out of A_0 are shifted into B_{15}. Bits shifted out of B_0 are lost. The sign bit in A_{15} is copied into the vacated high order bit. Overflow cannot occur on the addition since the result is shifted one bit to the right as the addition takes place.

Affected: A, B

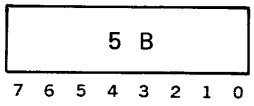
ADX Add to X



The 16-bit literal, contained in the second and third bytes of the instruction is added to contents of the X register. If the result is greater than $2^{15}-1$, or less than -2^{15}, the overflow register is set.

Affected: X, O

EBX Exchange B and X



The contents of the B and X registers are interchanged.

Affected: B, X

STACK CONTROL

The Stack Control group of instructions provides for CPU context switching of all active registers to and from a designated stack. The stacking capability of the MICRO 1600/21 and MICRO 821 is extremely efficient in processing multiple external interrupts and in employing reentrant coding techniques.

Push-Down/Pull-Up Operation

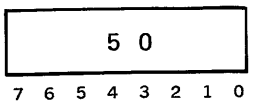
The push-down stack is a reserved area of memory (stack) into which registers are pushed (stored) and from which registers are pulled (loaded) on a last-in, first-out basis. Instructions are provided for pushing and pulling the A, B and X registers individually, or all the registers together. Also all internal and external interrupts except the console, power restart, and stack overflow interrupts cause all operational registers to be pushed into the stack and a jump to be made to the appropriate service routine.

The push-down stack has a maximum size of 255 bytes and is fully contained in any single 256 byte page of memory. Control of the current stack location is performed by a stack pointer word located at memory location $8C_{16}$. This pointer, which is the address of the last byte stored or the next byte to be loaded from the top of the stack, is decremented before each byte is pushed into the stack and is incremented after each byte is pulled from the stack. When a register or group of registers is to be pushed into the stack, a check is made to see if the registers can be stored without causing the stack to fill the page. If there is not sufficient storage available, the stack pointer will be unaltered and the system will

perform a return jump to the address contained in the stack overflow pointer located at memory location 88_{16}. There is no error indication if the stack pointer is incremented, (pulled) past the upper limit of the page.

A maximum stack size of 255 bytes may be obtained by initializing the stack pointer equal to the first byte of the page. This will permit stacking to start in location FF_{16} of that page, since the stack pointer is decremented before storing and arithmetic is performed only on the low order 8-bits of the address. For proper operation, pulling operations should not be performed without previous pushing operations, and over a given period of time, all pushing and pulling must be of equal occurrence.

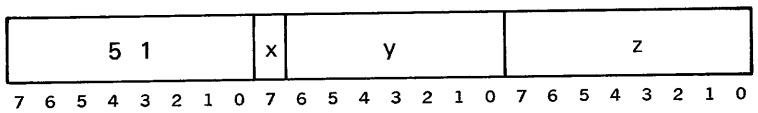
RTN Return



The O and W, P, B, A, and X registers, in this order, are loaded from the nine bytes at the top of the stack and the stack pointer is incremented by nine. The next instruction is obtained at the address loaded into the P register from the stack.

Affected: P, Stack Pointer, A, B, X, O and W

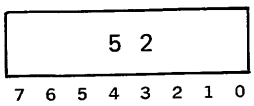
CAL Call



The X, A, B, P, O and W registers, in this order, are pushed into the stack and the effective address replaces the contents of the P register. The stack pointer is decremented by nine and points to the memory location containing the overflow and word length. Interrupts or concurrent I/O requests cannot be recognized before the execution of the next instruction.

Affected: P, Memory, Stack Pointer

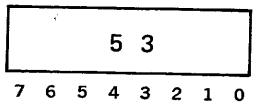
PLX Pull X



The two-byte operand located at the top of the push-down stack is placed in the X register and the stack pointer is incremented by two.

Affected: X, Stack Pointer

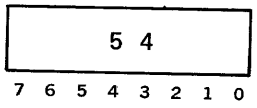
PSX **Push X**



The contents of the X register are stored in memory at the top of the push-down stack and the stack pointer is decremented by two.

Affected: Memory, Stack Pointer

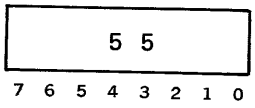
PLA **Pull A**



The two-byte operand located at the top of the push-down stack is placed in the A register and the stack pointer is incremented by two.

Affected: A, Stack Pointer

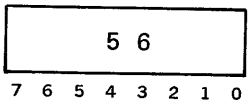
PSA **Push A**



The contents of the A register are stored in memory at the top of the push-down stack and the stack pointer is decremented by two.

Affected: Memory, Stack Pointer

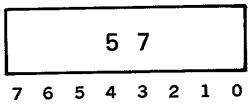
PLB **Pull B**



The two byte operand located at the top of the push-down stack is placed in the B register and the stack pointer is incremented by two.

Affected: B, Stack Pointer

PSB Push B



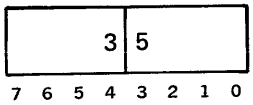
The contents of the B register are stored in memory at the top of the push-down stack and the stack pointer is decremented by two.

Affected: Memory, Stack Pointer

CHARACTER/STRING MANIPULATION

The character/string manipulation group of instructions provide the capability to process both individual characters and strings of characters in a manner compatible to common Input-Output operations and communications processing.

CLC Compare Logical

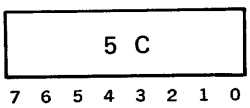


The byte operand located at the memory location given by the contents of the X register is compared with the byte operand given by the contents of the A register and then the contents of the A and X registers are incremented by one. If the two operands were equal and the modified contents of the X register is less than or equal to the contents of the B register the instruction is executed again. If the operand given by the contents of the X register was less than the operand given by the contents of the A register, the following two byte instruction is executed. If the two operands were equal and the modified contents of the X register was greater than the contents of the B register, the next two bytes are skipped and the following two byte instruction is executed. If the operand given by the contents of the X register was greater than the operand given by the contents of the A register, the next four bytes are skipped. Comparison is binary on any 8-bit value and proceeds from left to right. Interrupts and concurrent I/O requests may be serviced after each byte is compared.

Affected: A, X, P

Programming Note: The repeated execution of this instruction compares a string of characters starting with the address contained in the X register and ending with the address contained in the B register, to the string of characters starting with the address contained in the A register and indicating a less than, equal to, or greater than result.

MOV Move



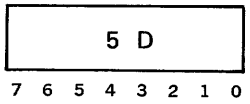
The byte operand located at the memory location given by the contents of the X register is stored at the memory location given by the contents of the A register and then the contents

of the A and X registers are incremented by one. If the modified contents of the X register is less than or equal to the contents of the B register, the instruction is executed again; otherwise, the following instruction is executed next. Interrupts and concurrent I/O requests may be serviced after each byte has been moved.

Affected: A, X, Memory

Programming Note: The repeated execution of this instruction causes the block of memory starting with the address contained in the X register and ending with the address contained in the B register to be moved to the memory area starting with the address contained in the A register.

GCC Generate Cyclic Code

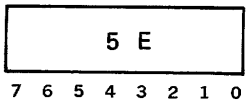


The byte operand located at the memory location given by the contents of the X register is entered into the 16-bit cyclic code contained in the A register and the contents of the X register is incremented by one. The polynomial used for the cyclic code is $X^{16} + X^{15} + X^2 + 1$. If the modified contents of the X register are less than or equal to the contents of the B register, the instruction is executed again; otherwise, the next instruction is executed. Interrupts and concurrent I/O requests may be serviced after each byte is processed.

Affected: A, X

Programming Note: This instruction is used to encode a block of eight bit characters starting with the address contained in the X register and ending with the address contained in the B register. This type of cyclic redundancy code (CRC) is used with the IBM Binary Synchronous Communication System (BSC), and assumes that each byte is transmitted low order bit first. Since the sixteen bit CRC, which is accumulated in the A register, will be transmitted as two bytes, the A register must be rotated eight-bits before attaching it to a message.

SCH Search

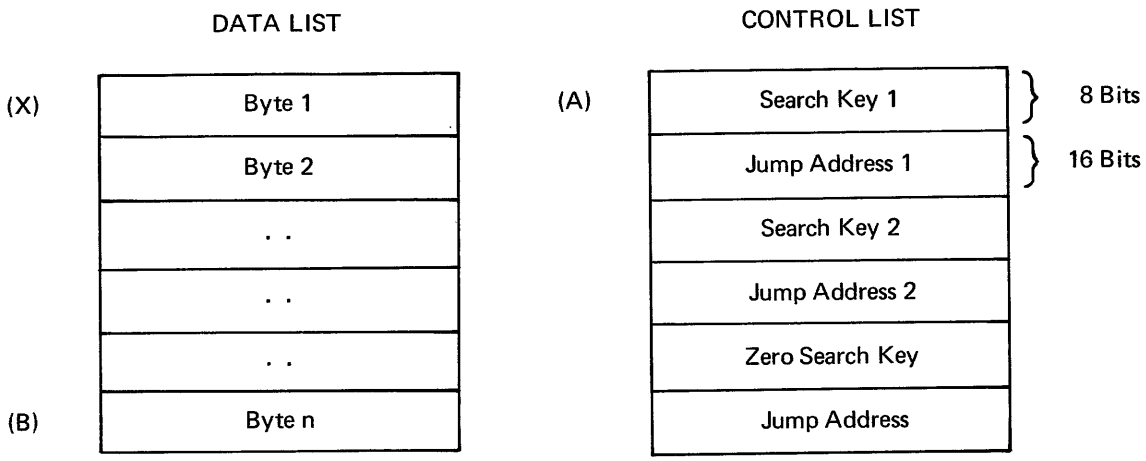


The byte operand located at the memory location given by the contents of the X register is compared with the Search key in a control list whose address is given by the contents of the A register. The control list contains one or more Search key bytes, each of which is followed by a two byte jump address. The list is terminated with a zero value Search key and jump address. If the operand is equal to one of the Search keys, the associated jump address from the control list replaces the contents of the P register. If the operand is not equal to any of the Search keys in the control list, the contents of the X register is incremented by one. If the modified contents of the X register is less than or equal to the contents of the B register, the instruction is executed again; otherwise, the following instruction is executed next.

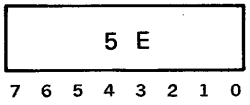
Interrupts or concurrent I/O requests will not be serviced until a match is found and the jump is performed, or until the complete control list is checked and the instruction is re-executed.

Afected: P, X

Programming Note: The repeated execution of the instruction compares the bytes in the block of memory starting with the address contained in the X register and ending with the address contained in the B register with the Search keys in a control list. When a match is found, a jump is made to the address contained in the word following the matched byte in the control list. When the jump is made, the X register contains the address of the byte in the data list which compared with the byte in the control list.



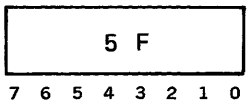
SCH Search Not



When the control list address contained in the A register has its index bit set to one, the byte operands in the data list are only compared with the first search key in the control list and a jump is made only when a match is not found. Following each successful match, the X register is incremented by one. If the modified contents of the X register is less than or equal to the contents of the B register, the instruction is executed again; otherwise, the following instruction is executed next. Interrupts and concurrent I/O requests may be serviced after each successful match before the instruction is re-executed, or after an unsuccessful match and the jump is performed.

Affected: P, X

GAP Generate ASCII Parity



The byte operand located at the memory location given by the contents of the X register is given a high order parity bit which makes an odd number 1-bits in the byte, and the modified operand is exclusive-ORed with the low order eight bits of the A register. Subsequently, the contents of the X register are incremented and if the modified contents are less than or equal to the contents of the B register, the instruction is executed again; otherwise, the next instruction is executed. Interrupts and concurrent I/O requests may be serviced after each byte is processed.

Affected: A (low order 8 bits), Memory

Programming Note: The repeated execution of this instruction will generate and attach an odd parity bit (VRC) for each character and will generate a block longitudinal parity (LRC) for all the characters starting with the address contained in the X register and ending with the address contained in the B register.

MEMORY REFERENCE

The 20 instructions of the memory reference group obtain their operands from memory. The operand memory location is addressed by one of eight modes as explained in Section 2. The number of bytes required for the instruction depends on the addressing mode and, for the literal mode, the length of the operand.

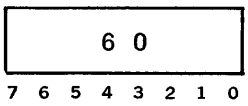
In the following instruction descriptions, only the first byte of the instruction which contains the basic operation code and the addressing mode is shown. The two-digit hexadecimal code given is for an operand addressing mode of 0 (m=0). For another addressing mode, the value of m must be added to the low order digit; i.e., for the Jump instruction, the code is:

$$(60_{16} + m).$$

For example, if the addressing mode is indirect to page 0 (m = 2), the hexadecimal value of the operation code is:

$$60_{16} + 2 = 62_{16}.$$

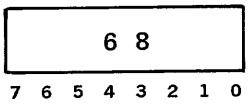
JMP Jump



The effective address replaces the contents of the P register causing the next instruction to be accessed at that location. Interrupts or concurrent I/O requests are not recognized before the execution of the next instruction.

Affected: P

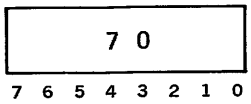
RTJ Return Jump



The current contents of the P register are stored in memory at the two-byte location specified by the effective address, and the effective address plus two replaces the original contents of the P register causing the next instruction to be accessed at that location. Interrupts or concurrent I/O requests are not recognized before the execution of the next instruction.

Affected: P, Memory

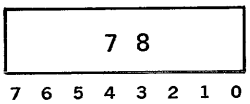
IWM Increment Word in Memory



The two-byte word in memory at the location specified by the effective address is incremented by one. If the result is greater than $2^{15}-1$, the overflow register is set.

Affected: O, Memory

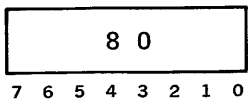
DWM Decrement Word in Memory



The two-byte word in memory at the location specified by the effective address is decremented by one. If the result is less than -2^{15} the overflow register is set.

Affected: O, Memory

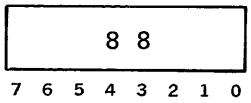
LDX Load X



The two-byte operand located at the effective memory location replaces the contents of the X register.

Affected: X

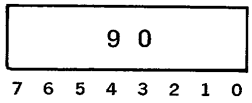
STX Store X



The contents of the X register are stored in memory at the two-byte location specified by the effective address.

Affected: Memory

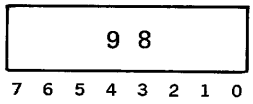
LDB Load B



The two-byte operand located at the effective memory location replaces the contents of the B register.

Affected: B

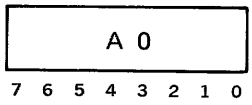
STB Store B



The contents of the B register are stored in memory at the two-byte location specified by the effective address.

Affected: Memory

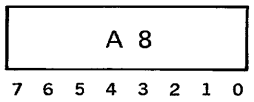
ADA Add to A



The two-byte operand located at the effective memory location is added to the contents of the A register and the sum is placed in the A register. If the sum is greater than $2^{15}-1$, or less than -2^{15}, the overflow register is set.

Affected: A, O

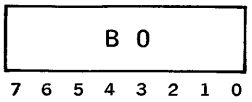
ADV Add Variable



The variable length operand located at the effective memory location is added to the contents of the A or A-B register and the sum is placed in the A or A-B register. If the magnitude of the sum is greater than can be contained in A or A-B for the specified word length, the overflow register is set.

Affected: A, B, O

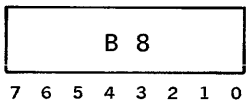
SBA Subtract from A



The two-byte operand located at the effective memory location is subtracted from the contents of the A register and the result is placed in the A register. If the result is greater than $2^{15}-1$, or less than -2^{15}, the overflow register is set.

Affected: A, O

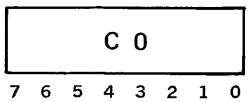
SBV Subtract Variable



The variable length operand located at the effective memory location is subtracted from the contents of the A or A-B register and the result is placed in the A or A-B register. If the magnitude of the difference is greater than can be contained in A or A-B for the specified word length, the overflow register is set.

Afected: A, B, O

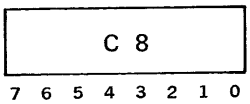
CPA Compare A (Less Than, Equal To, Greater Than)



The contents of the A register is compared with the two-byte operand at the effective memory location and the result determines the address of the next instruction to be executed. If the contents of the A register is less than the operand, the following two byte instruction is executed. If the contents of the A register is equal to the operand, the next two bytes are skipped and the following two byte instruction is executed. If the contents of the A register is greater than the operand the next four bytes are skipped.

Affected: P

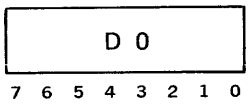
CPV Compare Variable (Less Than, Equal To, Greater Than)



The contents of the A or A-B register is compared with the operand at the effective memory location and the result determines the address of the next instruction to be executed. If the contents of the A or A-B register is less than the operand, the following two byte instruction is executed next. If the contents of the A or A-B register is equal to the operand, the next two bytes are skipped and the following two byte instruction is executed. If the contents of the A or A-B register is greater than the operand, the next four bytes are skipped.

Affected: P

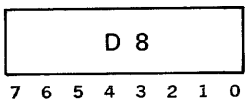
ANA AND



The two-byte operand located at the effective memory locations is logically ANDed with the contents of the A register and the result is placed in the A register.

Affected: A

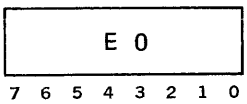
ANV AND Variable



The variable length operand located at the effective memory location is logically ANDed with the contents of the A or A-B register and the result is placed in the A or A-B register.

Affected: A, B

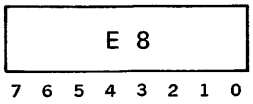
LDA Load A



The two-byte operand located at the effective memory location replaces the contents of the A register.

Affected: A

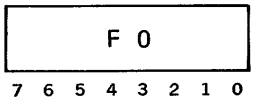
LDV Load Variable



The variable length operand located at the effective memory location replaces the contents of the A or A-B register.

Affected: A, B

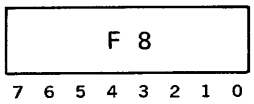
STA Store A



The contents of the A register are stored in memory at the two-byte location specified by the effective address.

Affected: Memory

STV Store Variable



The contents of the A or A-B register are stored in memory at the effective address.

Affected: Memory

INPUT/OUTPUT OPERATIONS

The MICRO 1600/21 and MICRO 821 provide three types of input/output:

Program-controlled transfer of data bytes via the Byte Input/Output Bus

Buffered concurrent transfer of data bytes via the Byte Input/Output Bus

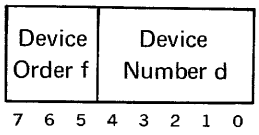
Direct transfer to memory via the direct memory access (DMA) channel

The Byte I/O Bus provides a path for transfer of data, control, and status between the processor and external peripheral devices. The direct memory access (DMA) channel communicates directly with memory.

BYTE INPUT/OUTPUT INSTRUCTIONS

Byte programmed input/output operations provides transfers of data, control, and status over the Byte I/O channel. This multiplex channel permits intermixed program and concurrent I/O transfers. More than one device on the bus may be operating in a concurrent block transfer mode at the same time. A maximum of 32 devices may normally be addressed on the Byte I/O bus.

The second byte of the instruction is a control byte which provides a three-bit device order and a five-bit device number as follows:



Byte input/output is basically a two phase operation. First, the control byte is placed on the output bus prior to the actual transfer of data. All devices examine the transmitted device number. The device, whose assigned number is the same as contained in the control word, accepts the control byte and performs the input or output of a single byte. When a device order does not require a data transfer, the second byte is disregarded by the device controller.

Device Address

Each device on the Byte I/O bus is assigned a unique five-bit device number. The numbers are assigned by means of selectively placed jumper wires on the printed circuit board of the device controller. The assigned device number is used by the device controller to compare against the device number of the control byte to determine if it is being addressed, and for identifying the device to the processor when requesting an interrupt or concurrent I/O transfer. Device number zero is always assigned to the parallel teletype interface.

Device Orders

The 3-bit device order specifies the type of I/O operation which will be performed. The device order accompanies the device number and is sent prior to each programmed transfer or to start a concurrent transfer.

Standard device orders designate the operations given in Table 2. Order codes 2, 3, 5, 6, and 7 are shown with their standard assignment, but they may be changed, depending on individual interface requirements.

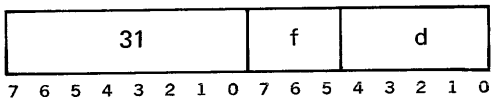
Status Bytes

The eight-bit status byte input as the result of a status order has four bits which are common to all devices and four which are device dependent. This byte is input to the A or B register or to memory by an input instruction with device order 1. The meaning of the status bits is given in Table 3.

INSTRUCTIONS

Three input and three output instructions provide for byte transfers between external devices and the A register, B register, or memory. When the transfer is to or from the A or B registers, only the eight low order bits are used. Interrupts or concurrent I/O requests are not recognized immediately following the execution of all Byte I/O instructions except, input byte to memory.

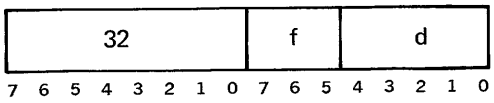
IBA Input Byte to A



The order code, f, is sent to the device designated by d. An eight-bit data byte is input from the device and placed in the low order bits of A. The eight high order bits of A remain unchanged.

Affected: A (low order 8-bits)

IBB Input Byte to B



The order code, f, is sent to the device designated by d. An eight-bit data byte is input from the device and placed in the eight low order bits of B. The eight high order bits of B remain unchanged.

Affected: B (low order 8-bits)

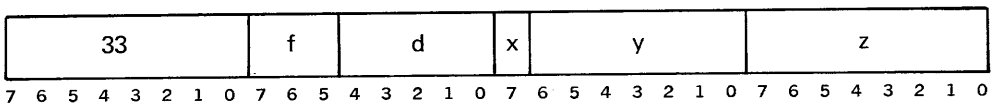
Table 2. Device Orders

Order Number	Operation	Description
0	Data Transfer	A data byte will be transferred between the addressed device and the processor. Direction of the transfer will depend on whether the instruction is an input or an output.
1	Status/Function	A status byte will be input from the addressed device or a function byte will be output to the addressed device, depending on whether the instruction is an input or an output.
2	Block Input/INT	The addressed device will start a concurrent block input to memory and will generate an external interrupt at the conclusion of the transfer unless the interrupt has been subsequently disarmed. This order should be sent by an output instruction.
3	Arm Interrupt	Permits the addressed device to make an external interrupt request upon the satisfaction of an interrupt condition. This order should be sent by an output instruction.
4	Disconnect	The block transfer in progress by the addressed device is stopped and end of block interrupt will occur unless the interrupt has been disarmed. This order should be sent by an output instruction.
5	Disarm Interrupt	Inhibits the addressed device from making an external interrupt request under any condition. This order should be sent by an output instruction.
6	Block Output/INT	The addressed device will start a concurrent block output from memory and will generate an external interrupt at the conclusion of the transfer unless the interrupt has been subsequently disarmed. This order should be sent by an output instruction.
7	Unassigned	This order, if assigned, may perform any required function as interpreted by the individual interface. If a byte transfer is desired the order may be sent by an input or an output instruction.

Table 3. Status Bytes Definition

Bit Number	Status	Description
0	Ready	This bit is a 1-bit when the external device is in a ready state.
1	Input Flag	This bit is a 1-bit when the external device has a byte ready for input to the computer.
2	Output Flag	This bit is a 1-bit when the external device is ready to receive a byte from the computer.
3	Error	This bit is a 1-bit when an error has occurred during a transfer. Errors may be timing, or device malfunction. This bit is cleared when the status byte is input.
4-7		Device dependent

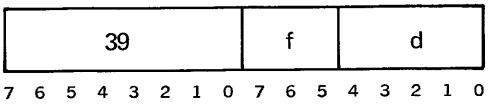
IBM Input Byte to Memory



The order code, f, is sent to the device designated by d. An eight-bit byte is input from the device and stored in memory at the effective memory address.

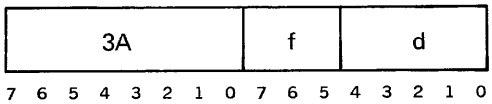
Affected: Memory

OBA Output Byte from A



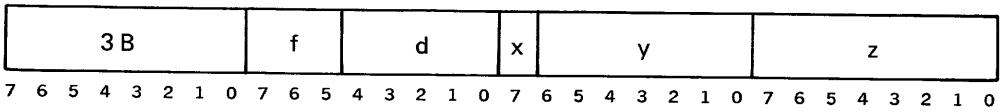
The order code, f, is sent to the device designated by d. The contents of the eight low order bits of A are output to the device. The contents of A remain unchanged.

OBB Output Byte from B



The order code, f, is sent to the device designated by d. The contents of the eight low order bits of B are output to the device. The contents of B remain unchanged.

OBM **Output Byte from Memory**



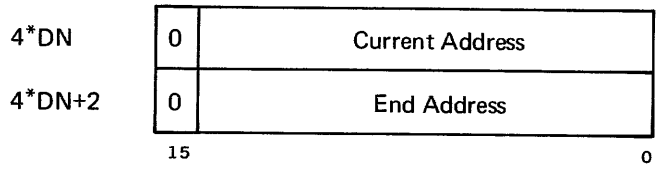
The order code, f, is sent to the device designated by d. The contents of the eight bit byte at the effective memory address are sent to the device. The contents of memory remain unchanged.

CONCURRENT INPUT/OUTPUT

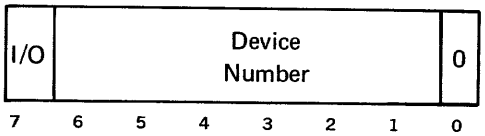
The concurrent I/O allows for block transfers between the external device on the Byte I/O bus and memory, at a maximum rate of 20,000 bytes per second. The transfers are fully automatic, and once started, proceed without program intervention. Concurrent I/O takes priority over instruction execution and forces momentary sequence breaks during executions of long instructions to insure that concurrent I/O delays are not excessive.

Address Control

Concurrent I/O addresses for each external device are controlled by a pair of two-byte address words. These two words are located in memory starting at an address equal to four times the device number. The first word is the current address (CA) and contains the address of the next memory byte to be used for the transfer. The second word is the End Address (EA) and contains the address of the last byte of the block. The first 128 locations in memory are reserved for storing concurrent I/O addresses for control of up to 32 external devices. The four bytes for each device have the following format:



When the processor detects a request for concurrent I/O, it inputs an externally supplied address (ESA) from the requesting device. This byte must contain a device address in bits 5-1, zeros in bits 0 and 6, and an output flag in bit 7. When bit 7 is a one, it signifies that the device is requesting an output transfer; otherwise an input is performed. The ESA is used by the processor to define the type of concurrent I/O operation requested and to locate the appropriate address control words. The ESA has the following format:



Concurrent Operations

Concurrent I/O operations are started by executing byte I/O instructions with the proper device order codes. These codes are given in Table 2. A block transfer can be performed with or without an interrupt at the end of the transfer. After a concurrent I/O operation is initiated by a processor instruction, byte transfers proceed automatically until the last byte of the block is transferred. Following each transfer, the processor increments the current

address. When the current address (CA) is greater than the end address, the processor automatically sends a disconnect order code to the device. This order code causes the concurrent I/O operation to cease and a device interrupt to be generated, unless it was previously disabled.

EXTERNAL INTERRUPTS

External interrupts originate with device controllers or interrupt modules on the Byte I/O bus. An interrupt module provides control of eight external interrupt signals. Device controllers may also generate interrupts to signify individual data transfers, end of operation, or error conditions.

The external interrupt system contains a single interrupt line, a priority line, and a select line. A device may initiate an interrupt request only if priority has been received from higher level interrupts on the priority chain. Devices not requiring interrupt service will propagate priority to the next device in line.

When the processor recognizes the interrupt signal, it enables the select line for the interrupt system. Each device in order will interrogate the select line and, if not requesting, will propagate this signal to the next device in line. Once the select signal has been propagated by a device, it will be locked out from acknowledging this signal until the select is removed. When the select signal is received by the requesting device, it will input its address on the data in bus. This ESA address may be six bits, (bits 6-1) since interrupt modules may take on interrupt addresses in the range of 32 to 63. The ESA address is used to locate the interrupt subroutine address located in memory starting at location 100_{16}. The processor reads this subroutine address and performs a call to the specified address. This entails storing all the operational registers into the push down stack and performing a jump to the subroutine address. Interrupts or concurrent I/O requests cannot be recognized before the execution of the instruction located at the subroutine address.

MICRO 1600/21 OPERATOR CONTROLS

Basic and system consoles, differing in number of displays and controls, are available with the MICRO 1600/21 computer. Choice of console can be based on the user's needs to meet control and display capability required for specific applications, and he can tailor the cost accordingly.

All console panels are pluggable and fully interchangeable without modification to the computer.

An optional parallel Teletype controller, physically contained within the control consoles, may be specified.

SYSTEM CONSOLE

The system console (Figure 3) provides control plus a selectable display of all hardware registers in the machine including the files. It is designed for maintenance operations and for installations where system development and firmware checkout is being performed.

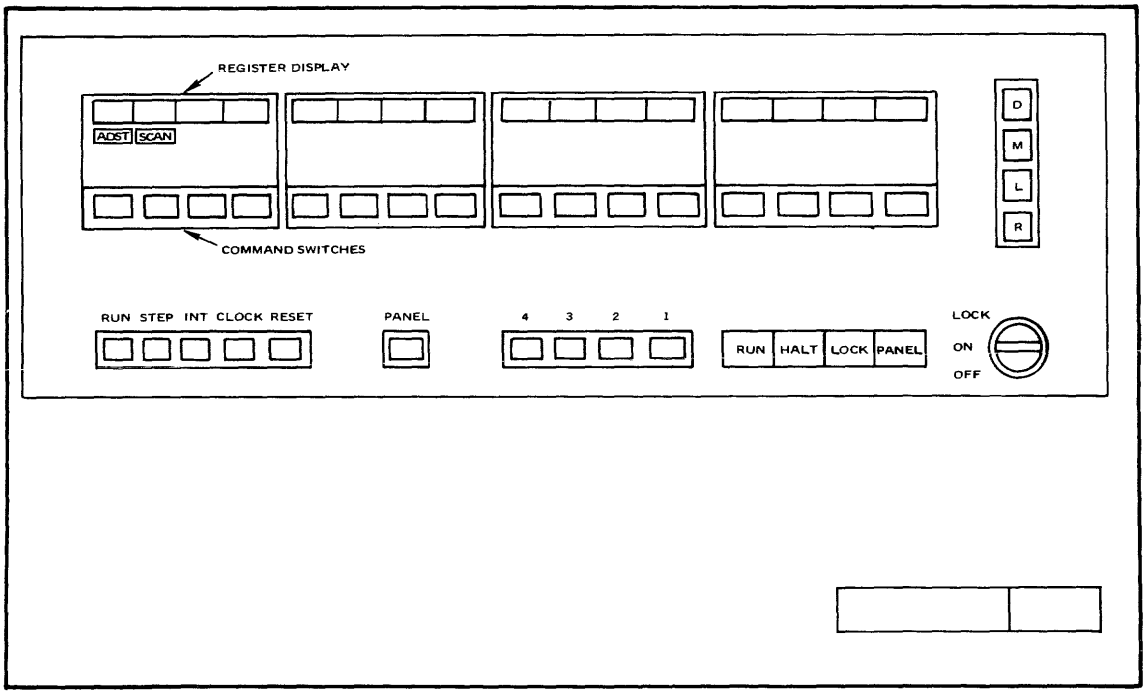


Figure 3. MICRO 1600/21 System Console

DISPLAYS

Data Display

Illuminated lamps (16 on system console), display the output of the A Bus, Memory Address, R Register input and the "L" counter from the processor.

Run Lamp

Run lamp is illuminated when processor is in run mode.

Halt Lamp

Halt lamp is illuminated when processor is in halt mode. Instruction step, clock step, reset switches, and data display are active only in this mode.

Display Selector (D, M, C, L)

Display selection includes: four alternate action switches which select the file registers or the other hardware registers (when the processor is in the halt mode) as follows:

D – A Bus Data (8 bits)

M – Memory Address (16 bits)

C – Control Memory Output (16 bits)

L – Microprogram location counter (15 bits)

SWITCHES

Sense Switches (4)

Four alternate action switches which can be entered and tested by microcommand, permitting implementation of various conditional sense switch macro instructions.

Address Stop: The 16th switch enables an address stop mode if L is selected on the display switch. The lower 15 command switches are used to select the address stop location. This facility provides a console breakpoint operation for the microprogram and is useful for troubleshooting and firmware debugging.

Address Sync: A sync jack is mounted on the rear of the front panel for maintenance purposes. If "L" is selected on the display switches and the 16th command switch is not depressed, a sync will occur for the address set by the lower 15 command switches.

Command Switches (16)

Sixteen alternate action switches which provide manual input of microcommand word for execution. Switches are enabled only when the panel select switch is in the down position.

Run Switch

Momentary contact switch places processor in run mode.

Halt/Step Switch

Momentary contact switch places processor in the halt mode from the run mode. In the halt mode, depressing the switch causes execution of a single macro instruction step from core memory.

Clock Step Switch

Momentary contact switch which executes a single micro clock step in the halt mode.

Master Reset Switch

Three-position switch: up lock, down momentary contact. Sets the processor to the halt mode from the run mode, clears the microprogram location counter (L), overflow indicator, and all internal status flags. Also generates a master reset signal over the I/O bus. Placing switch in the up position before turning power off will provide a memory data save function.

Interrupt Switch

Momentary contact switch which generates micro level interrupt.

Panel Select Switch

Used primarily for maintenance purposes, the alternate action switch selects the microprogram control store as microcommand source in the up position. When the switch is in the down position, microcommands are executed from the 16 console command switches.

Power On/Off/Lock

A three-position key lock switch applies dc power to the system. A Master Reset is generated and the halt lamp will be illuminated when power is first applied. The lock position inhibits panel control switches except sense switches but leaves power applied to the system.

BASIC CONSOLE

The basic console provides a minimal control facility. The control switches (run, halt/step, clock step, reset, and interrupt) permit a basic ability to sequence the machine.

MICRO 821 OPERATOR CONTROLS

CONSOLES

Two control consoles are available: system console and basic console. These consoles differ in their number of displays and controls. This range of consoles permits the user to tailor the cost to meet the control and display capability required for a particular application. The systems control console is shown in Figure 4.

System Console

The system console provides complete control and display facilities. It is primarily used for maintenance, system and firmware checkout. This console provides for display of the registers in addition to the functions of the operator console. The features include:

Run and halt indicators

Display of A-bus

Display of M, N, and L registers

Display of output of read-only memory

Four sense switches

Six control switches including:

- Run
- Step
- Interrupt
- Clock
- Reset
- Save

Manual Command execution

Power on/off

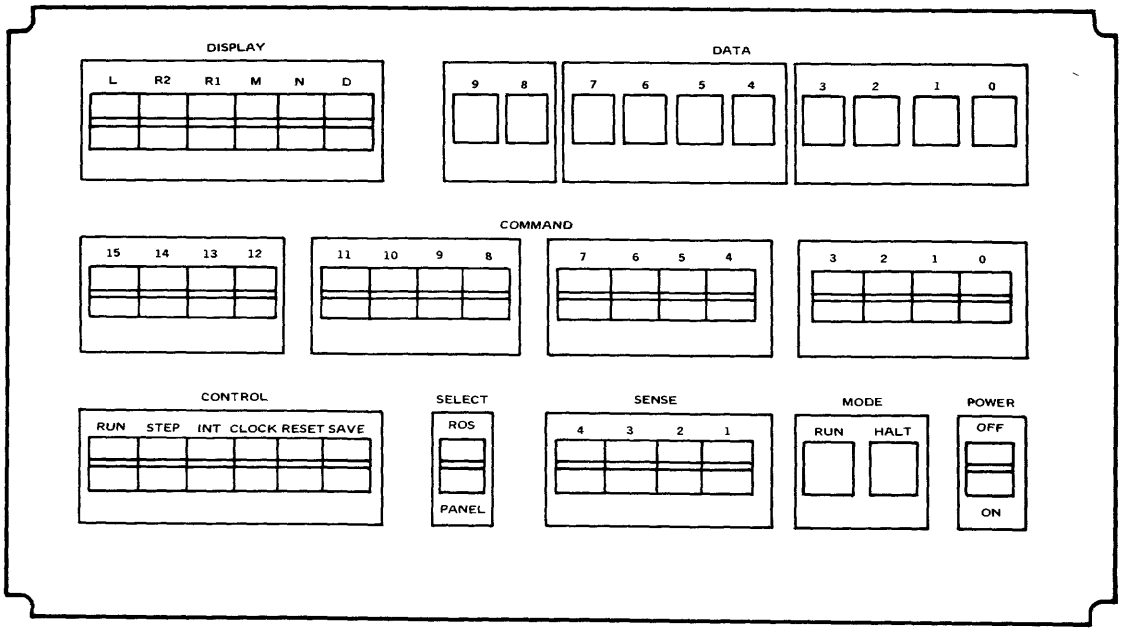


Figure 4. MICRO 821 System Console

Basic Console

The basic console provides minimal control capability and is designed for dedicated system applications where operator control is not required. The features include:

Run and halt indicators

Four sense switches

Six control switches including:

Run
Step
Interrupt
Clock
Reset
Save

Power on/off

DISPLAYS

Run Lamp

The run lamp is illuminated when the processor is running.

Halt Lamp

The halt lamp is illuminated when the power is on and the process is not running.

Data Display

On the operator console, eight lamps display the data which is on the A bus of the processor. When the processor is halted, the contents of a file register or the T register can be displayed by setting the proper command in the command switches and enabling the switches by placing the select switch in the panel position. The hexadecimal commands used for display are:

File Register f	-Cf00
T Register	-B020
Link Register	-B080

On the system console, a set of display selector switches select the data to be displayed on a set of 10 lamps. (See Data Selector Switches).

SWITCHES

Display Selector

These seven interlocked switches select the register or bus to be displayed on the system console. The displays which can be selected are: L register, M register, N register, eight high order bits of the read-only-storage output, eight low order bits of the read-only-storage and the A-bus. When the machine is halted, the output of the read-only-storage is the same as the contents of the R register, and is the next command to be executed.

Command

These 16 alternate action switches are substituted for the read only storage on the system and operator consoles when the select switch is in the panel position. Depressing the clock switch causes the command set on the switches to be executed. The command may also be executed repeatedly by depressing the run switch. These switches are used to gate registers to the A bus display and for entering data into the file and registers.

Select

This alternate action switch selects the console panel command switches (panel) or the read only memory (ROM) as the command to be executed next. This switch is not available on the basic console.

Sense

The four alternate action sense switches are available on all consoles. The state of these switches may be transferred to the A register by the enter sense switch instruction to provide manual control over program execution.

Run

This momentary contact switch places the processor in the run mode causing it to run.

Step

This momentary contact switch causes the execution of one Micro 821 instruction and also halts the machine at the end of the current instruction execution, if it is running. If the instruction executed will not permit recognition of interrupts following it, at least one more instruction will be executed.

Interrupt

This momentary contact switch places the processor in the run mode and causes a console interrupt.

Clock

This momentary contact switch causes the processor to execute a single microcommand. If the processor is running at the time the switch is depressed, the processor will come to a forced halt following the current microcommand execution.

Reset

This momentary contact switch instantly halts the processor and clears the L register, I/O control register and other control flip-flops. The reset is made available to I/O devices. Since the current microcommand execution will not be completed, the computer should not be stopped by this switch. Starting the computer after a reset causes it to start instruction execution at memory location 0.

Save

This alternate action switch is the same as the reset switch but can be set on providing a continuous reset. If this switch is on at the time the power is turned on or off, the contents of the memory will not be lost or altered. This switch need not be used when proper power fail/restart software is resident in core memory.

APPENDIX A. FILE REGISTER ASSIGNMENTS

The 16 file registers of the MICRO 800 and 1600 are used for temporary storage and for the operational registers of the MICRO 821 and MICRO 1600/21 as shown below:

File Register	Use
0	Condition Flags
1	Instruction Register
2	Lower Byte of X Register
3	Upper Byte of X Register
4	Lower Byte of A Register
5	Upper Byte of A Register
6	Lower Byte of B Register
7	Upper Byte of B Register
8	Lower Byte of P Register
9	Upper Byte of P Register
A (Bit 1-0)	W Register
A (Bit 2)	O Register
B	Temporary Storage
C	Temporary Storage
D	Temporary Storage
E	Lower Byte of Operand Address
F	Upper Byte of Operand Address

Note: The MICRO 1600 has a secondary bank of file registers which are unused in the implementation of the MICRO 1600/21.

APPENDIX B. DEDICATED MEMORY

Hex Address	Assignment
000-001	Device 0 CA
002-003	Device 0 EA
004-005	Device 1 CA
006-007	Device 1 EA
.	.
.	.
058	DMA Channel 1 Status
059	DMA Channel 2 Status
.	
.	
060-061	DMA Channel 1, Buffer 1 SA
062-063	DMA Channel 1, Buffer 1 EA
.	
.	
06C-06D	DMA Channel 1, Buffer 4 SA
06E-06F	DMA Channel 1, Buffer 4 EA
070-071	DMA Channel 2, Buffer 1 SA
072-073	DMA Channel 2, Buffer 1 EA
.	
.	
07C-07D	DMA Channel 2, Buffer 4 SA or, Device 31 CA
07E-07F	DMA Channel 2, Buffer 4 EA or, Device 31 EA
080-081	Console Interrupt
082-083	DMA Channel Interrupt
084-085	Real-Time Clock Counter
086-087	Real-Time Clock Interrupt
088-089	Stack Overflow Interrupt
08A-08B	Memory Parity Interrupt
08C-08D	Push Down Stack Pointer
08E-08F	Power Fail Interrupt
090-091	Power Restart Interrupt
092	DMA Umbrella Cell

Hex Address	Assignment
097	
.	
.	Undedicated Page 0
.	
0FF	
100-101	Device 0 Interrupt
102-103	Device 1 Interrupt
.	.
.	.
.	.
13E-13F	Device 31 Interrupt
140-141	External Interrupt 32
142-143	External Interrupt 33
.	.
.	.
.	.
17E-17F	External Interrupt 63

APPENDIX C. MICRO 1600/21 EXECUTION TIMES

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
0 0	HLT		5.2	
0 1	TRP		15.4	Includes Return Jump
0 2	ESW		4.4	
0 4	DIN		4.4	
0 5	EIN		4.4	
0 6	DRT		4.4	
0 7	ERT		4.0	
0 8	RO1		4.8	
0 9	RO2		4.8	
0 A	RO3		4.8	
0 B	RO4		4.8	
0 C	SO1		4.8	
0 D	SO2		4.8	
0 E	SO3		4.8	
0 F	SO4		4.8	
1 0	JOV	Jump	7.8	Add .2 if displacement negative
		No Jump	6.2	
1 1	JAZ	Jump	7.6	Add .2 if displacement negative
		No Jump	6.8	
1 2	JBZ	Jump	7.4	Add .2 if displacement negative
		No Jump	6.6	
1 3	JXZ	Jump	7.2	Add .2 if displacement negative
		No Jump	6.4	
1 4	JAN	Jump	7.6	Add .2 if displacement negative
		No Jump	6.8	
1 5	JXN	Jump	7.4	Add .2 if displacement negative
		No Jump	6.6	
1 6	JAB	Jump	7.8	Add .2 if displacement negative
		No Jump	7.0	
1 7	JAX	Jump	7.6	Add .2 if displacement negative
		No Jump	6.8	
1 8	NOV	Jump	7.0	Add .2 if displacement negative
		No Jump	7.0	
1 9	NAZ	Jump	7.6	Add .2 if displacement negative
		No Jump	6.8	
1 A	NBZ	Jump	7.4	Add .2 if displacement negative
		No Jump	6.6	

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
1 B	NXZ	Jump	7.2	Add .2 if displacement negative
		No Jump	6.4	
1 C	NAN	Jump	7.6	Add .2 if displacement negative
		No Jump	6.8	
1 D	NXN	Jump	7.4	Add .2 if displacement negative
		No Jump	6.6	
1 E	NAB	Jump	7.8	Add .20 if displacement negative
		No Jump	7.0	
1 F	NAX	Jump	7.6	Add .20 if displacement negative
		No Jump	6.8	
2 0	LLA		5.8	Add 3.2 for each bit position shifted
2 1	LLB		5.8	Add 3.2 for each bit position shifted
2 2	LLL		5.8	Add 3.4 for each bit position shifted
2 4	LRA		5.8	Add 3.0 for each bit position shifted
2 5	LRB		5.8	Add 3.0 for each bit position shifted
2 6	LRL		5.8	Add 3.6 for each bit position shifted
2 8	ALA		5.8	Add 3.2 for each bit position shifted
2 9	ALB		5.8	Add 3.2 for each bit position shifted
2 A	ALL		5.8	Add 3.4 for each bit position shifted
2 C	ARA		5.8	Add 3.0 for each bit position shifted
2 D	ARB		5.8	Add 3.0 for each bit position shifted
2 E	ARL		5.8	Add 3.6 for each bit position shifted
3 1	IBA		7.6	
3 2	IBB		8.0	
3 3	IBM		13.0	Add 1.2 if indexed
3 4	NOP		4.0	
3 5	CLC		9.6	Per byte, if equal
			10.0	For last byte, if less than
			11.0	For last byte, if equal
			10.8	For last byte, if greater than
3 C	DAD	(Average)	21.0	Units position digit
		(Average)	17.4	Per non-units position digit
		(Average)	17.0	Per digit for re-complementing
		(Average)	20.6	High order digit
				Add 1.2 if result overflows
3 D	DSB	(Average)	21.0	Units position digit
		(Average)	17.4	Per non-units position digit
		(Average)	17.0	Per digit for re-complementing
		(Average)	20.6	High order digit
				Add 1.2 if result overflows

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
3 E	MUL	(Minimum)	59.8	Multiplier (A) equals zero Add 1.2 if indexed
		(Maximum)	73.2	Multiplier (A) $-2^{15}+1$ Add 1.2 if indexed
3 F	DIV	(Minimum)	90.0	No remainder, quotient equal to 2^n Add 1.2 if indexed
		(Maximum)	95.4	Negative divident, quotient equal to $215-2$ Add 1.2 if indexed
Concurrent I/O				
	During Multiply		14.4	Add 4.4 if end of block occurs
	During Divide		13.8	Add 4.4 if end of block occurs
3 9	OBA		7.6	
3 A	OBB		8.4	
3 B	OBM		13.2	Add 1.2 if indexed
4 0	ORA		5.8	
4 1	XRA		5.8	
4 2	ORB		6.0	
4 3	XRB		6.0	
4 4	INX		6.4	Add .60 if result overflows
4 5	DCX		6.4	Add .60 if result overflows
4 6	AWX		6.4	Add .60 if result overflows
4 7	SWX		6.4	Add .60 if result overflows
4 8	INA		6.4	Add .60 if result overflows
4 9	INB		6.4	Add .60 if result overflows
4 A	OCA		6.0	
4 B	OCB		6.0	
4 C	TAX		6.4	
4 D	TBX		6.4	
4 E	TXA		6.6	
4 F	TXB		6.6	
5 0	RTN		29.6	
5 1	CAL		31.4	Add 1.2 if indexed
5 2	PLX		13.8	
5 3	PSX		12.8	
5 4	PLA		13.8	
5 5	PSA		12.8	
5 6	PLB		13.8	
5 7	PSB		12.8	
5 8	MST		8.2	Add .60 if B register is odd
5 9	ADX		8.0	Add .60 if overflow occurs

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
5 A	JEP	(Minimum)		
		Jump (A=0)	8.6	Add .20 if displacement negative
		No Jump (A=1)	7.8	
		(Maximum)		
		Jump (A<0)	29.6	Add .20 if displacement negative
		No Jump (A<0)	28.8	
5 B	EBX		6.4	
5 C	MOV		9.2	Per Byte, less .60 for termination
5 D	GCC	(Minimum)	27.0	Per Byte, less .60 for termination
		(Maximum)	31.8	Per Byte, less .60 for termination
5 E	SCH		7.2	General overhead per data byte Add 3.00 for each non-zero, unmatched, key checked Add 4.8 for zero key (no match) Add 4.8 to perform jump (any match) Less .6 for termination (no match)

Example: Two byte data list, three byte control list, data byte two is matched with search key two.
(7.2 + (2X3.0) + 4.8) + (7.2 + 3.0 + 4.8) = 33.0

5 E	SCH	(Not)		
		Match, No Jump	11.4	Per data byte
		No Match, Jump	12.0	Less .6 for termination (all matched)
5 F	GAP	(Data = 0)	9.8	Per byte, less .6 for termination
		(Data <32)	14.6	Per byte, less .6 for termination
		(Data <64)	15.8	Per byte, less .6 for termination
		(Data <128)	17.0	Per byte, less .6 for termination
		(Data <0)	18.2	Per byte, less .6 for termination

ADDRESSING MODES – Time to be added to memory referencing instructions

Direct Page 0	4.8	
Direct Relative	5.8	Add .6 if displacement negative
Indirect Page 0	7.8	Add 1.2 if post indexed
Indirect Relative	8.8	Add 1.2 if post indexed Add .6 if displacement negative
Indexed	4.8	
Indexed with Bias	5.6	
Extended	6.2	Add 1.2 if indexed
Literal		
Fixed Length	7.4	
Two Byte with A	7.8	
Variable	7.4	
Indirect Jumps	10.4	Add 1.2 if indexed Add 1.2 if post indexed

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
MEMORY REFERENCING INSTRUCTION				
6 0	JMP		3.2	
6 8	RTJ		5.8	
7 0	IWM		5.4	Add .6 if result overflows
7 8	DWM		5.4	
8 0	LDX		5.4	
8 8	STX		5.4	
9 0	LDB		5.4	
9 8	STB		5.8	
A 0	ADA		4.8	Add .60 if result overflows
A 8	ADV	(1 Byte)	6.2	Add .60 if result overflows
		(2 Bytes)	5.8	Add .60 if result overflows
		(3 Bytes)	8.6	Add .60 if result overflows
		(4 Bytes)	8.2	Add .60 if result overflows
B 0	SBA		5.2	Add .60 if result overflows
B 8	SBV	(1 Byte)	6.6	Add .60 if result overflows
		(2 Bytes)	6.2	Add .60 if result overflows
		(3 Bytes)	9.0	Add .60 if result overflows
		(4 Bytes)	8.6	Add .60 if result overflows
C 0	CPA		4.8	Add .80 if A $\geqslant$ memory
C 8	CPV	(1 Byte)	4.8	Add .80 if A $\geqslant$ memory
		(2 Bytes)	5.6	Add .80 if A,B $\geqslant$ memory
		(3 Bytes)	7.2	Add .80 if A,B $\geqslant$ memory
		(4 Bytes)	8.2	Add .80 if A,B $\geqslant$ memory
D 0	ANA		5.2	
D 8	ANV	(1 Byte)	6.6	
		(2 Bytes)	6.2	
		(3 Bytes)	9.0	
		(4 Bytes)	8.6	
E 0	LDA		5.2	
E 8	LDV	(1 Byte)	6.6	
		(2 Bytes)	6.2	
		(3 Bytes)	9.0	
		(4 Bytes)	8.6	
F 0	STA		4.2	
F 8	STV	(1 Byte)	3.4	
		(2 Bytes)	4.6	
		(3 Bytes)	8.0	
		(4 Bytes)	9.2	
INTERRUPTS				
	Console Interrupt		13.8	Includes Return Jump
	DMA Termination		34.2	Includes Call, Operation
	Real Time Clock (Increment)		9.8	Add 27.4 if result is zero, to perform Call, Operation

Hex	Mnemonic	Time (micro-seconds)	Additions or Conditions
	Stack Overflow	12.8	Includes Return Jump
	Memory Parity	34.2	Includes Call, Operation
	Console Halt	6.8	
	Power Fail	33.8	Includes Call, Operation
	Power Restart	10.2	Includes Return Jump
	External Interrupt	32.0	Includes Call, Operation
INPUT OUTPUT			
	Concurrent I/O		
	Between Instructions	15.8	Add 4.4 if end of block occurs
	During Shift	13.0	Add 4.4 if end of block occurs

APPENDIX D. MICRO 821 EXECUTION TIMES

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
0 0	HLT		5.72	
0 1	TRP		16.94	Includes Return Jump
0 2	ESW		4.84	
0 4	DIN		4.84	
0 5	EIN		4.84	
0 6	DRT		4.84	
0 7	ERT		4.40	
0 8	R01		5.28	
0 9	R02		5.28	
0 A	R03		5.28	
0 B	R04		5.28	
0 C	S01		5.28	
0 D	S02		5.28	
0 E	S03		5.28	
0 F	S04		5.28	
1 0	JOV	Jump	8.58	Add .22 if displacement negative
		No Jump	6.82	
1 1	JAZ	Jump	8.36	Add .22 if displacement negative
		No Jump	7.48	
1 2	JBZ	Jump	8.14	Add .22 if displacement negative
		No Jump	7.26	
1 3	JXZ	Jump	7.92	Add .22 if displacement negative
		No Jump	7.04	
1 4	JAN	Jump	8.36	Add .22 if displacement negative
		No Jump	7.48	
1 5	JXN	Jump	8.14	Add .22 if displacement negative
		No Jump	7.26	
1 6	JAB	Jump	8.58	Add .22 if displacement negative
		No Jump	7.70	
1 7	JAX	Jump	8.36	Add .22 if displacement negative
		No Jump	7.48	
1 8	NOV	Jump	7.70	Add .22 if displacement negative
		No Jump	7.70	
1 9	NAZ	Jump	8.36	Add .22 if displacement negative
		No Jump	7.48	
1 A	NBZ	Jump	8.14	Add .22 if displacement negative
		No Jump	7.26	
1 B	NXZ	Jump	7.92	Add .22 if displacement negative
		No Jump	7.04	

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
1 C	NAN	Jump	8.36	Add .22 if displacement negative
		No Jump	7.48	
1 D	NXN	Jump	8.14	Add .22 if displacement negative
		No Jump	7.26	
1 E	NAB	Jump	8.58	Add .22 if displacement negative
		No Jump	7.70	
1 F	NAX	Jump	8.36	Add .22 if displacement negative
		No Jump	7.48	
2 0	LLA		6.38	Add 3.52 for each bit position shifted
2 1	LLB		6.38	Add 3.52 for each bit position shifted
2 2	LLL		6.38	Add 3.74 for each bit position shifted
2 4	LRA		6.38	Add 3.30 for each bit position shifted
2 5	LRB		6.38	Add 3.30 for each bit position shifted
2 6	LRL		6.38	Add 3.96 for each bit position shifted
2 8	ALA		6.38	Add 3.52 for each bit position shifted
2 9	ALB		6.38	Add 3.52 for each bit position shifted
2 A	ALL		6.38	Add 3.74 for each bit position shifted
2 C	ARA		6.38	Add 3.30 for each bit position shifted
2 D	ARB		6.38	Add 3.30 for each bit position shifted
2 E	ARL		6.38	Add 3.96 for each bit position shifted
3 1	IBA		8.36	
3 2	IBB		8.80	
3 3	IBM		14.30	Add 1.32 if indexed
3 4	NOP		4.40	
3 5	CLC		10.56	Per byte, if equal
			11.00	For last byte, if less than
			12.10	For last byte, if equal
			11.88	For last byte, if greater than
3 C	DAD	(Average)	23.10	Units position digit
		(Average)	19.14	Per non-units position digit
		(Average)	18.70	Per digit for re-complementing
		(Average)	22.66	High order digit
				Add 1.32 if result overflows
3 D	DSB	(Average)	23.10	Units position digit
		(Average)	19.14	Per non-units position digit
		(Average)	18.70	Per digit for re-complementing
		(Average)	22.66	High order digit
				Add 1.32 if result overflows
3 E	MUL	(Minimum)	65.78	Multiplier (A) equals zero
				Add 1.32 if indexed
		(Maximum)	80.52	Multiplier (A) $-2^{15}+1$
				Add 1.32 if indexed

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
3 F	DIV	(Minimum)	99.00	No remainder, quotient equal to 2^n Add 1.32 if indexed
		(Maximum)	104.94	Negative divident, quotient equal to 215 -2 Add 1.32 if indexed
Concurrent I/O				
	During Multiply		15.84	Add 4.84 if end of block occurs
	During Divide		15.18	Add 4.84 if end of block occurs
3 9	OBA		8.36	
3 A	OBB		9.24	
3 B	OBM		14.52	Add 1.32 if indexed
4 0	ORA		6.38	
4 1	XRA		6.38	
4 2	ORB		6.60	
4 3	XRB		6.60	
4 4	INX		7.04	Add .66 if result overflows
4 5	DCX		7.04	Add .66 if result overflows
4 6	AWX		7.04	Add .66 if result overflows
4 7	SWX		7.04	Add .66 if result overflows
4 8	INA		7.04	Add .66 if result overflows
4 9	INB		7.04	Add .66 if result overflows
4 A	OCA		6.60	
4 B	OCB		6.60	
4 C	TAX		7.04	
4 D	TBX		7.04	
4 E	TXA		7.26	
4 F	TXB		7.26	
5 0	RTN		32.56	
5 1	CAL		34.54	Add 1.32 if indexed
5 2	PLX		15.18	
5 3	PSX		14.08	
5 4	PLA		15.18	
5 5	PSA		14.08	
5 6	PLB		15.18	
5 7	PSB		14.08	
5 8	MST		9.02	Add .66 if B register is odd
5 9	ADX		8.80	Add .66 if overflow occurs
5 A	JEP	(Minimum)		
		Jump (A=0)	9.46	Add .22 if displacement negative
		No Jump (A=1)	8.58	
		(Maximum)		
		Jump (A<0)	32.56	Add .22 if displacement negative
		No Jump (A<0)	31.68	

Hex	Mnemonic		Time (micro-seconds)	Additions or Conditions
5 B	EBX		7.04	
5 C	MOV		10.12	Per Byte, less .66 for termination
5 D	GCC	(Minimum)	29.70	Per Byte, less .66 for termination
		(Maximum)	34.98	Per Byte, less .66 for termination
5 E	SCH		7.92	General overhead per data byte Add 3.30 for each non-zero, unmatched, key checked Add 5.28 for zero key (No Match) Add 5.28 to perform jump (Any Match) Less .66 for termination (no match)
	Example: Two byte data list, three byte control list, data byte two is matched with search key two. (7.92 + (2X3.30) + 5.28) + (7.92 + 3.30 + 5.28) = 36.30			
5 E	SCH	(Not)		
		Match, No Jump	12.54	Per data byte
		No Match, Jump	13.20	Less .66 for termination (all matched)
5 F	GAP	(Data = 0)	10.78	Per byte, less .66 for termination
		(Data <32)	16.06	Per byte, less .66 for termination
		(Data <64)	17.38	Per byte, less .66 for termination
		(Data <128)	18.70	Per byte, less .66 for termination
		(Data <0)	20.02	Per byte, less .66 for termination
ADDRESSING MODES — Time to be added to memory referencing instructions				
	Direct Page 0		5.28	
	Direct Relative		6.38	Add .66 if displacement negative
	Indirect Page 0		8.58	Add 1.32 if post indexed
	Indirect Relative		9.68	Add 1.32 if post indexed Add .66 if displacement negative
	Indexed		5.28	
	Indexed with Bias		6.16	
	Extended		6.82	Add 1.32 if indexed
	Literal			
	Fixed Length		8.14	
	Two Byte with A		8.58	
	Variable		8.14	
	Indirect Jumps		11.44	Add 1.32 if indexed Add 1.32 if post indexed
MEMORY REFERENCING INSTRUCTION				
6 0	JMP		3.52	
6 8	RTJ		6.38	
7 0	IWM		5.94	Add .66 if result overflows
7 8	DWM		5.94	Add .66 if result overflows

Hex	Mnemonic		Time (micro-seconds)	Additions or Changes
8 0	LDX		5.94	
8 8	STX		5.94	
9 0	LDB		5.94	
9 8	STB		6.38	
A 0	ADA		5.28	Add .66 if result overflows
A 8	ADV	(1 Byte)	6.82	Add .66 if result overflows
		(2 Bytes)	6.38	Add .66 if result overflows
		(3 Bytes)	9.46	Add .66 if result overflows
		(4 Bytes)	9.02	Add .66 if result overflows
B 0	SBA		5.72	Add .66 if result overflows
B 8	SBV	(1 Byte)	7.26	Add .66 if result overflows
		(2 Bytes)	6.82	Add .66 if result overflows
		(3 Bytes)	9.90	Add .66 if result overflows
		(4 Bytes)	9.46	Add .66 if result overflows
C 0	CPA		5.28	Add .88 if A $\geqslant$ memory
C 8	CPV	(1 Byte)	5.28	Add .88 if A $\geqslant$ memory
		(2 Bytes)	6.38	Add .88 if A,B $\geqslant$ memory
		(3 Bytes)	7.92	Add .88 if A,B $\geqslant$ memory
		(4 Bytes)	9.02	Add .88 if A,B $\geqslant$ memory
D 0	ANA		5.72	
D 8	ANV	(1 Byte)	7.26	
		(2 Bytes)	6.82	
		(3 Bytes)	9.90	
		(4 Bytes)	9.46	
E 0	LDA		5.72	
E 8	LDV	(1 Byte)	7.26	
		(2 Bytes)	6.82	
		(3 Bytes)	9.90	
		(4 Bytes)	9.46	
F 0	STA		4.62	
F 8	STV	(1 Byte)	3.74	
		(2 Bytes)	5.06	
		(3 Bytes)	8.80	
		(4 Bytes)	10.12	

INTERRUPTS

Console Interrupt	15.18	Includes Return Jump
DMA Termination	37.62	Includes Call, Operation
Real Time Clock (Increment)	10.78	Add 30.14 if result is zero, to perform Call, perform Call, Operation
Stack Overflow	14.08	Includes Return Jump
Memory Parity	37.62	Includes Call, Operation
Console Halt	7.48	
Power Fail	37.18	Includes Call, Operation
Power Restart	11.22	Includes Return Jump
External Interrupt	35.20	Includes Call, Operation

Hex	Mnemonic	Time (micro-seconds)	Additions or Conditions
INPUT OUTPUT			
	Concurrent I/O		
	Between Instructions	17.38	Add 4.84 if end of block occurs
	During Shift	14.30	Add 4.84 if end of block occurs

APPENDIX E. STANDARD CHARACTER CODES

SYMBOL	ASCII (HEX)	EBCDIC (HEX)	HOLLERITH (029)	(026)
blank	A0	40	blank	
!	A1	5A	11-8-2	
"	A2	7F	8-7	0-8-5
#	A3	7B	8-3	0-8-7
$	A4	5B	11-8-3	
%	A5	6C	0-8-4	11-8-7
&	A6	50	12	12-8-7
'	A7	7D	8-5	8-4
(	A8	4D	12-8-5	0-8-4
)	A9	5D	11-8-5	12-8-4
*	AA	5C	11-8-4	
+	AB	4E	12-8-6	12
,	AC	6B	0-8-3	
–	AD	60	11	
.	AE	4B	12-8-3	
/	AF	61	0-1	
0	B0	F0	0	
1	B1	F1	1	
2	B2	F2	2	
3	B3	F3	3	
4	B4	F4	4	
5	B5	F5	5	
6	B6	F6	6	
7	B7	F7	7	
8	B8	F8	8	
9	B9	F9	9	
:	BA	7A	8-2	8-5
;	BB	5E	11-8-6	
<	BC	4C	12-8-4	12-8-6
=	BD	7E	8-6	8-3
>	BE	6E	0-8-6	8-6
?	BF	6F	0-8-7	12-8-2

SYMBOL	ASCII (HEX)	EBCDIC (HEX)	HOLLERITH (029)	(026)
@	C0	7C	8-4	0-8-2
A	C1	C1	12-1	
B	C2	C2	12-2	
C	C3	C3	12-3	
D	C4	C4	12-4	
E	C5	C5	12-5	
F	C6	C6	12-6	
G	C7	C7	12-7	
H	C8	C8	12-8	
I	C9	C9	12-9	
J	CA	D1	11-1	
K	CB	D2	11-2	
L	CC	D3	11-3	
M	CD	D4	11-4	
N	CE	D5	11-5	
O	CF	D6	11-6	
P	D0	D7	11-7	
Q	D1	D8	11-8	
R	D2	D9	11-9	
S	D3	E2	0-2	
T	D4	E3	0-3	
U	D5	E4	0-4	
V	D6	E5	0-5	
W	D7	E6	0-6	
X	D8	E7	0-7	
Y	D9	E8	0-8	
Z	DA	E9	0-9	
[	DB	4F	12-8-7	12-8-5
\	DC	4A	12-8-2	0-8-6
]	DD	5F	11-8-7	11-8-5
↑	DE	6D	0-8-5	8-7
←	DF	6A	0-8-2	8-2

APPENDIX F. TELETYPE CONTROL AND TRANSMISSION CODES

FUNCTION	ASCII
NULL	80
SOM (Print on)	81
EAO	82
EOM	83
EOT (Print off)	84
WRU	85
RU	86
BELL	87
FEO	88
H.TAB	89
LINE FEED	8A
V.TAB	8B
FORM	8C
CARRIAGE RETURN	8D
SO	8E
SI	8F
DCO	90
X-ON (Reader on)	91
TAPE (Punch on)	92
X-OFF (Reader off)	93
TAPE OFF (Punch off)	94
ERROR	95
SYNC	96
LEM	97
S0	98
S1	99
S2	9A
S3	9B
S4	9C
S5	9D
S6	9E
S7	9F

APPENDIX G. TABLE OF POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25

APPENDIX H. HEXADECIMAL = DECIMAL INTEGER CONVERSION TABLES

The following tables aid in converting hexadecimal values to decimal values, or the reverse.

Direct Conversion Table

This table provides direct conversion of decimal and hexadecimal numbers in these ranges:

HEXADECIMAL	DECIMAL
000 to FFF	0000 to 4095

For numbers outside the range of the table, add the following values to the table figures:

HEXADECIMAL	DECIMAL
1000	4096
2000	8192
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B000	45056
C000	49152
D000	53248
E000	57344
F000	61440

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00_	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01_	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02_	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03_	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04_	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05_	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06_	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07_	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08_	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09_	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A_	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B_	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C_	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D_	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E_	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F_	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
10_	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11_	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12_	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13_	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14_	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15_	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16_	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17_	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18_	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19_	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A_	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B_	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C_	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D_	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E_	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F_	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20_	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21_	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22_	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23_	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24_	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25_	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26_	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27_	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28_	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29_	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A_	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B_	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C_	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D_	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E_	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F_	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30_	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31_	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32_	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33_	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34_	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35_	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36_	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37_	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38_	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39_	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A_	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B_	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C_	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D_	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E_	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F_	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
40_	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
41_	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
42_	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
43_	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
44_	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
45_	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
46_	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
47_	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
48_	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
49_	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A_	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B_	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C_	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D_	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E_	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F_	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
50_	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
51_	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
52_	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
53_	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
54_	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
55_	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
56_	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
57_	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
58_	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
59_	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A_	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B_	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C_	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D_	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E_	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F_	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
60_	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
61_	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
62_	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
63_	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
64_	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
65_	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
66_	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
67_	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
68_	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
69_	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A_	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B_	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C_	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D_	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E_	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F_	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
70_	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
71_	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
72_	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
73_	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
74_	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
75_	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
76_	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
77_	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
78_	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
79_	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A_	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B_	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C_	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D_	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E_	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F_	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

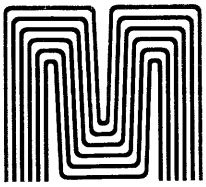
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80_	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
81_	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
82_	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
83_	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
84_	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
85_	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
86_	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
87_	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
88_	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
89_	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A_	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B_	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C_	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D_	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E_	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F_	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
90_	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
91_	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
92_	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
93_	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
94_	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
95_	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
96_	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
97_	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
98_	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
99_	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A_	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B_	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C_	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D_	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E_	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F_	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0_	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A1_	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A2_	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A3_	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A4_	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A5_	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A6_	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A7_	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A8_	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A9_	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA_	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB_	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC_	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD_	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE_	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF_	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B0_	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B1_	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B2_	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B3_	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B4_	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B5_	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B6_	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B7_	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B8_	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B9_	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA_	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB_	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC_	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD_	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE_	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF_	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0_	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C1_	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C2_	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C3_	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C4_	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C5_	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C6_	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C7_	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C8_	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C9_	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA_	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB_	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC_	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD_	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE_	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF_	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D0_	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D1_	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D2_	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D3_	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D4_	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D5_	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D6_	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D7_	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D8_	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D9_	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA_	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB_	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC_	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD_	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE_	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF_	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E0_	3584	3585	3586	3587	3583	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1_	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2_	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3_	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4_	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5_	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6_	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7_	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8_	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9_	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA_	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB_	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC_	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED_	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE_	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF_	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0_	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1_	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2_	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3_	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4_	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5_	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6_	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7_	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8_	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9_	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA_	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB_	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC_	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD_	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE_	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF_	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

Microdata



Microdata Corporation
644 East Young Street
Santa Ana, California 92705
Telephone: (714) 540-6730
TWX: 910-595-1764