FABRI-TEK® INC.
COMPUTER SYSTEMS

MP12

MICROPROCESSOR

Reference Manual

402-1001-00

©Copyright 1974 by FABRI-TEK, INC.
Printed in the United States of America

SECTION I

PREFACE

Section I of this manual describes the organization, features, and operation of the FABRI-TEK MP12 Microprocessor.

SECTION I
TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER I
INTRODUCTION

The FABRI-TEK MP12 Microprocessor is a general purpose, 12-bit computer designed primarily for OEM applications. The basic system, contained in a 2.0 in. wide, 15.0 in. deep, and 9.5 in. high enclosure, consists of the following elements:

- Parallel-logic processor.
- 4096 x 12 random access magnetic core memory.
- Input-Output interface.
- Hardware interrupt facility.
- Power-fail restart facility.
- Operating console.

Optional FABRI-TEK MP12 features includes a full-duplex, asynchronous, communications interface, up to 2048 words of bipolar read-only memory, +5V power supply module, chassis assembly, and wire-wrap assembly.

Standard FABRI-TEK MP12 software includes an assembler, loader, source edit utility, debugging utility, and diagnostics.

PROCESSOR

The FABRI-TEK MP12 Microprocessor is mounted on a single 9.5 in. by 15.0 in. printed circuit card that contains all logic circuitry necessary for processor operation and interfacing.

MEMORY

The standard FABRI-TEK MP12 memory is a 4K x 12, coincident current magnetic core memory. The memory is mounted on a single 14.6 in. by 9.25 in. printed circuit card that contains all necessary memory electronics and interface circuitry.

INPUT-OUTPUT INTERFACE

The input-output interface incorporates two input-output channels, a processor input-output (PIO) channel, and a direct memory access (DMA) channel. The PIO channel interfaces with the processor via the data input bus and provides simplex character-oriented data transfer capability, at rates of up to 66,000 words-per-second. The DMA channel interfaces directly with the memory, via the data input bus, and provides high-speed, record-oriented data transfer capability at rates of up to 666,666 words-per-second.

The combinations of power-fail, auto-restart, and read-only memory enable the FABRI-TEK MP12 to function effectively in unattended operating environments.

POWER SUPPLY

The optional power supply provides a regulated DC source of 5 volts at 20 amps to operate the processor and peripheral interface logic. The power supply also includes circuitry to detect a low AC input condition and to provide a power low interrupt signal to the processor. An additional feature of the power supply is a line frequency clock interrupt circuit that provides an interrupt signal to the processor at 16 2/3 ms intervals. The power supply is designed to permit the regulator section to operate with 12-volt battery input; this feature permits operation in a mobil vehicle. The power supply is housed in a double width enclosure (2 X processor enclosure size) that plugs into the chassis assembly without any wiring other than the AC line cord.

CHASSIS ASSEMBLY

The optional chassis assembly mounts in a standard 19 inch rack and provides convenient mounting locations for the processor, power supply, and up to 15 peripheral interface cards. The chassis assembly is designed using a printed circuit backplane for all interconnecting wiring.

WIRE-WRAP ASSEMBLY

Input-output signals, available at the FABRI-TEK MP12 input-output interface connector, are arranged to permit the use of a printed circuit backplane for interconnecting signals between the processor and external input-output device controllers. An optional wire-wrap assembly, having the same physical dimensions as the processor enclosure, accommodates a printed circuit card suitable for mounting up to 140 14-pin, 16-pin, or 24-pin packages. This card connects directly to the printed circuit backplane and may be used to host peripheral interface logic.

SOFTWARE

Standard software for the FABRI-TEK MP12 includes an assembler, loader, debugging utility, source edit utility, and diagnostic programs. The assembler translates symbolic assembly language programs into executable machine programs. The loader loads object tapes produced by the assembler or debugging utility. The debugging utility aids program checkout and features multiple breakpoints, instruction trace, and several other standard functions. The source edit utility is used to generate new assembly language source tapes or modify existing source tapes. The diagnostics are used to verify MP12 Microprocessor operation.

CHAPTER II
ORGANIZATION

INTRODUCTION

This chapter describes the internal organization of the FABRI-TEK MP12 Microprocessor. Chapter topics include the Data Input Bus, Processor, Processor Registers, Memory, Memory Registers, Input-Output Interface, and Interrupt Facility.

DATA INPUT BUS (DIB)

The FABRI-TEK MP12 features a single bus structure; the processor, memory, and input-output channels all share a common Data Input Bus (DIB). The Data Input Bus is the mechanism whereby address information and data are transferred between the switch register (SR) and the processor, between the processor and the memory, between the memory and the input-output interface, and between the processor and the input-output interface. The system block diagram, Figure 2-1, illustrates the single bus organization of the FABRI-TEK MP12 .

PROCESSOR

The MP12 Microprocessor performs control, input-output, arithmetic, and logical operations by executing instructions obtained from the memory. All instructions use a single 12-bit machine word. Depending on the number of separate memory accesses required, the processor may require one, two, or three memory cycles to complete execution of an instruction.

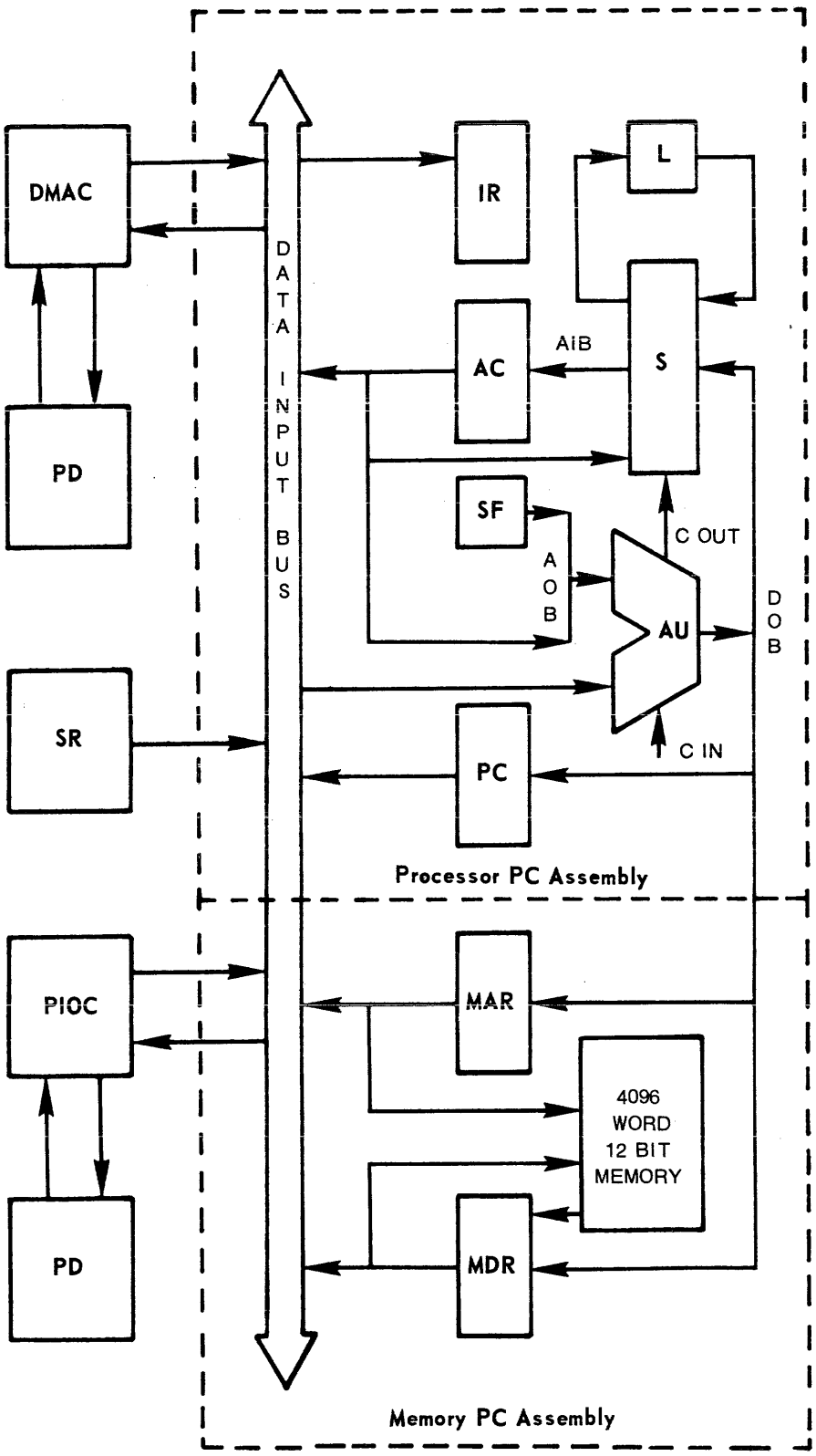
The FABRI-TEK MP12 Microprocessor incorporates four hardware registers and the logic circuitry necessary to perform control, arithmetic, and logical operations with respect to instructions and data stored in the memory. The processor logic includes a 12-bit parallel arithmetic unit (AU) that performs two's complement arithmetic operations, and a parallel shifter unit (S) that performs logical and shift operations.

ACCUMULATOR (AC)

The accumulator is a 12-bit register that functions as a holding register for arithmetic, logical, and input-output operations.

LINK REGISTER (L)

The link register is a 1-bit register that functions as an extension of the accumulator for arithmetic and logical operations.



AC	Accumulator
AIB	Accumulator In Bus
AOB	Accumulator Out Bus
AU	Arithmetic Unit
C IN	Carry In
C OUT	Carry Out
DMAC	Direct Memory Access Controller
DOB	Data Out Bus
IR	Instruction Register
L	Link
MAR	Memory Address Register
MDR	Memory Data Register
PC	Program Counter
PD	Peripheral Device
PIOC	Processor Input/Output Controller
S	Shifter
SF	Skip Flag
SR	Switch Register

Figure 2-1
MP12 BLOCK DIAGRAM

PROGRAM COUNTER (PC)

The program counter is a 12-bit register that holds the memory address of the next instruction to be processed. The execution of each instruction causes the program counter to be loaded with the address of the next instruction to be executed.

INSTRUCTION REGISTER (IR)

The instruction register is a 12-bit register that is used to hold the instruction currently being executed by the processor.

SKIP FLAG (SF)

The skip flag is a 1-bit register that represents a true/false skip condition with respect to the instruction being executed by the processor.

MEMORY

The FABRI-TEK MP12 memory has a capacity of 4096 12-bit words and has a read/write cycle time of 1.5 microseconds. The memory is non-volatile; if power is removed, data stored in memory is not lost.

The processor and input-output interface communicate with the memory by way of the data input bus. Two hardware registers are used to hold memory address information and data received via the bus: the memory address register and the memory data register.

MEMORY ADDRESS REGISTER (MAR)

The memory address register is a 12-bit register that is used to hold the address of a data word to be read from, or written into, memory.

MEMORY DATA REGISTER (MDR)

The memory data register is a 12-bit register that holds the last data word read from, or written into, the memory location addressed by the contents of the memory address register.

INPUT-OUTPUT INTERFACE

The FABRI-TEK MP12 input-output interface incorporates two input-output channels, a processor input-output channel, and a direct memory access input-output channel.

PROCESSOR INPUT-OUTPUT (PIO) CHANNEL

The processor input-output channel enables data transfer between the accumulator and a selected input-output controller and device, as directed by the execution of series of input-output transfer (IOT) instructions.

DIRECT MEMORY ACCESS (DMA) CHANNEL

The direct memory access channel functions as an independent data path to the memory. For a DMA transfer, control and address information are transmitted to a selected DMA controller via the processor input-output channel. The DMA controller then initiates and controls the transfer of data between the memory and a specified input-output device.

INTERRUPT FACILITY

The FABRI-TEK MP12 interrupt facility provides a processor interrupt when an input-output device is ready to receive or send data, or when a primary power failure is detected. The interrupt facility may be enabled or disabled using the interrupt on (ION) and interrupt off (IOF) instructions. If the interrupt facility is enabled when an interrupt occurs, the processor disables the interrupt facility, stores the contents of the program counter in memory location 0, and executes the instruction at memory location 1. If the interrupt system is disabled when an interrupt occurs, the interrupt is remembered by the processor and will remain active until cleared. When the ION instruction is executed, the interrupt facility is enabled after the instruction that follows the ION instruction is executed.

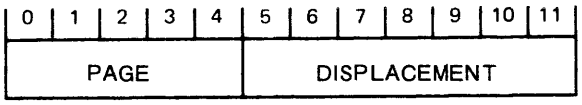
CHAPTER III
FUNCTIONAL DESCRIPTION

INTRODUCTION

This chapter describes the functional characteristics of the FABRI-TEK MP12 computer. It discusses addressing techniques, including effective address generation and auto-index addressing; describes the formats used to represent various types of data internally; and concludes with a detailed description of the FABRI-TEK MP12 instruction set.

ADDRESS STRUCTURE

The FABRI-TEK MP12 possesses a contiguous memory address space of 4096 12-bit words. Any location within memory is accessable by way of a 12-bit address. This 12-bit address may be interpreted as: (1) a 5-bit page address field which specifies one of 32 pages of 128 words each, and (2) a 7-bit displacement address field which specifies one of 128 locations within the specified page.

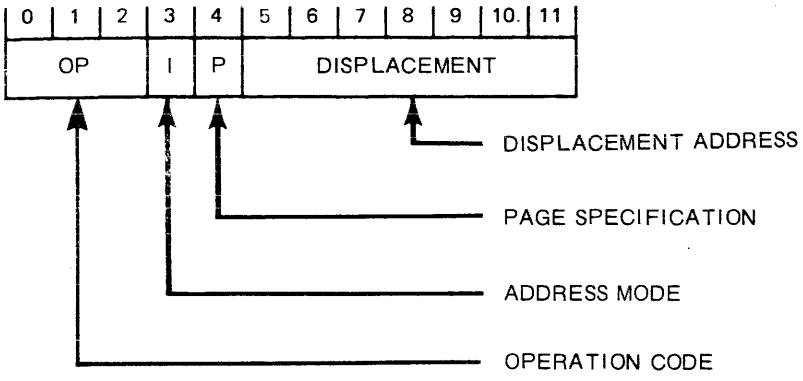


12-BIT ADDRESS FORMAT

Pages are assigned consecutive addresses in the range 0 to 31 (0_8-37_8) with page 0, the first 128 locations of memory, referred to as the base page. Displacement addresses range from 0 to 127 (0_8-177_8).

ADDRESSING TECHNIQUES

Each FABRI-TEK MP12 memory reference instruction dedicates three bits for operation code, one bit for address mode, one bit for page specification, and seven bits for displacement address.



MEMORY REFERENCE INSTRUCTION FORMAT

The effective address of a memory reference instruction operand is computed using the displacement address and the mode and page specification bits. The effective address is then used to access the required location in memory.

EFFECTIVE ADDRESS GENERATION

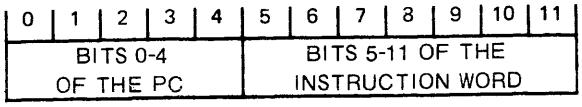
The effective address of a memory reference instruction operand is generated in the following manner:

1. A 12-bit primary address is generated from bit 4, the page specification bit, and the displacement address, bits 5 through 11.

2. A 12-bit effective address is generated from bit 3, the address mode bit, and the 12-bit primary address.

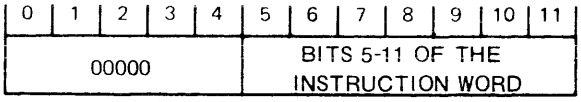
Primary Address

Bit 4, the page specification bit, controls the selection of the page address. When set to one, this bit indicates that the specified memory location lies within the current page; the page containing the instruction itself. In this case, a 12-bit primary address is obtained by combining bits 0 through 4 of the program counter with the displacement address of the instruction word as illustrated below.



CURRENT PAGE ADDRESS

A zero in the page specification bit position indicates that the specified memory location lies within the base page. In this case, as illustrated below, a 12-bit primary address is obtained directly from the displacement address.



BASE PAGE ADDRESS

Effective Address

Bit 3, the address mode bit, controls effective address generation. When zero, this bit indicates the direct address mode. When the direct address mode is specified, the primary address is interpreted as the effective operand address.

When set to one, the address mode bit indicates the indirect address mode. In this case, the contents of the primary address are interpreted as the effective operand address.

AUTO-INDEX ADDRESS

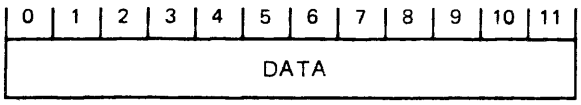
Base page locations 10_8 through 17_8 are referred to as auto-index locations. When any of these locations are indirectly addressed, the contents are read, incremented by one, rewritten back in the same location, and then used as an effective address.

DATA FORMATS

Data is logically represented in three internal binary formats: single word, double word, and floating point.

SINGLE WORD DATA

Single word data uses a single machine word to represent a 12-bit binary number in the following manner:

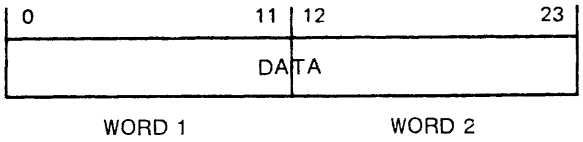


SINGLE WORD DATA

The data is right justified, in bit positions 0 through 11. Negative data is represented in two's complement form. The numerical range of signed data, which may be represented in this format, is $-2048_{10} \leq i \leq 2047_{10}$ ($4000_8 \leq i \leq 3777_8$). The numerical range of absolute (unsigned) data is $0_{10} \leq i \leq 4095_{10}$ ($0_8 \leq i \leq 7777_8$).

DOUBLE WORD DATA

Double word data, illustrated below, uses two consecutive machine words to represent a 24-bit signed binary number.

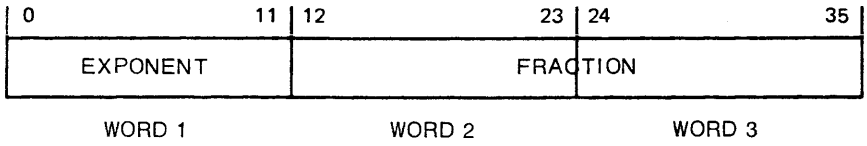


DOUBLE WORD DATA

The data is right justified, in bit positions 0 through 23. Negative data is represented in two's complement form. The numerical range of signed data, which may be represented in this format, is $-8388608_{10} \leq i \leq 8388607_{10}$ ($40000000_8 \leq i \leq 37777777_8$).

FLOATING POINT DATA

Floating point data is represented in three consecutive machine words as depicted below:



FLOATING POINT DATA

The 12-bit two's complement exponent occupies bits 0 through 11 of word one. The 24-bit two's complement fraction occupies bit positions 12 through 35 of words two and three. The radix point of the number is located immediately to the right of the high order fraction bit. Six-plus significant digits are representable in this format within an absolute numerical range of approximately $10^{\pm 616}$.

INSTRUCTION SET

The MP12 instruction set is organized into three instruction classes: Memory Reference, Operate, and Input-Output. The following sections describe each class by format and instruction functions.

MEMORY REFERENCE INSTRUCTIONS

The MP12 instruction set includes six basic memory reference instructions. Each instruction occupies a single 12-bit machine word and consists of a 3-bit operation code, a 2-bit address modification field, and a 7-bit displacement address as illustrated in Figure 3-1.

AND (Octal Code 0) Logical AND

The AND instruction results in a bit-by-bit Boolean AND operation between the contents of the accumulator and the memory data word addressed by the instruction. The result of the operation is retained in the accumulator. The contents of the addressed memory word are not altered and the link register is not affected.

TAD (Octal Code 1) Two's Complement Add

The TAD instruction performs a binary addition between the addressed data word and the contents of the accumulator. The result of the addition is retained in the accumulator. If a carry from the most significant bit of the accumulator occurs, the link register is complemented. The contents of the addressed memory word are not altered.

ISZ (Octal Code 2) Increment, Skip on Zero

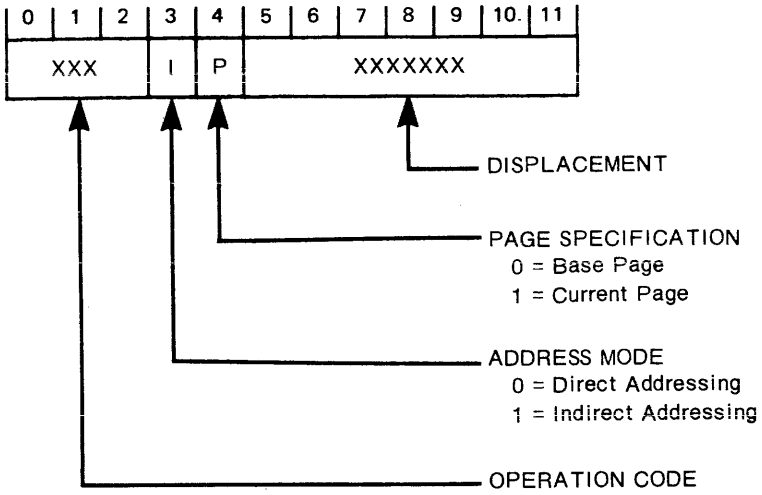
The ISZ instruction increments the contents of the addressed memory location and examines the result. If the result is zero, the next instruction in sequence is skipped and the following instruction is executed. If the result is non-zero, the next instruction in sequence is executed. In either case, the incremented result replaces the original data word in memory. Neither the accumulator nor the link register is affected.

DCA (Octal Code 3) Deposit and Clear Accumulator

The DCA instruction stores the contents of the accumulator in the addressed memory location, replacing the original contents of that location. The accumulator is then set to zero. The link register is not affected.

JMS (Octal Code 4) Jump to Subroutine

The JMS instruction causes the contents of the PC, the address of the JMS instruction plus one, to be stored in the addressed memory location, replacing the original contents. The PC is then set to the address of this location plus one; thus, the next instruction executed is the one following the location at which the PC was stored by the JMS instruction. Neither the accumulator nor the link register is affected.



Mnemonic	Op Code	Instruction
AND	000	Logical AND
TAD	001	Two's Complement Add
ISZ	010	Increment, skip on zero.
DCA	011	Deposit and clear accumulator
JMS	100	Jump to subroutine
JMP	101	Jump

Figure 3-1
MEMORY REFERENCE INSTRUCTION FORMAT

JMP (Octal Code 5) Jump

The JMP instruction causes the PC to be loaded with the address specified by the instruction word, resulting in a transfer of control to this location. The contents of the accumulator and link register are not affected.

OPERATE INSTRUCTIONS

The MP12 operate instructions enable the manipulation and testing of data located in the accumulator and link register. The operate instructions are separated into two functional groupings, referred to as Groups I and II. Each operate instruction occupies a single 12-bit machine word and consists of a 3-bit operation code, a group specification bit, and eight instruction specification bits.

Group I Operate Instructions

The Group I operate instructions manipulate the accumulator and link register. Figure 3-2 illustrates the format of the Group I operate instructions. The operation of each individual instruction is described below.

CLA — CLEAR ACCUMULATOR. Each bit in the accumulator is set to zero.

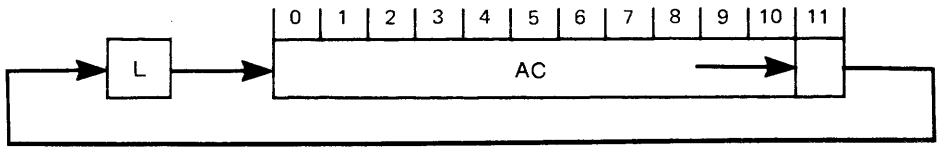
CLL — CLEAR LINK REGISTER. The link register is set to zero.

CMA — COMPLEMENT ACCUMULATOR. Each bit in the accumulator is complemented.

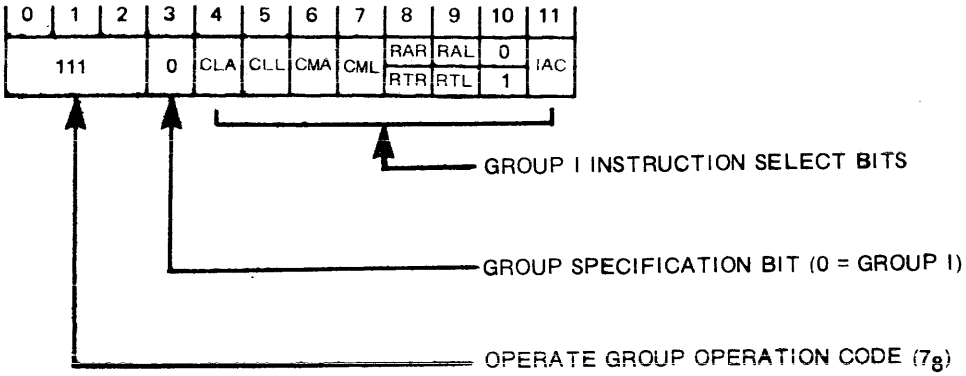
CML — COMPLEMENT LINK. The link register is complemented.

IAC — INCREMENT ACCUMULATOR. The accumulator is incremented by one. The link register is complemented in case a carry occurs from the most significant bit of the accumulator.

RAR — ROTATE ACCUMULATOR AND LINK RIGHT. The link register and accumulator are rotated one position to the right as illustrated below:



RTR — ROTATE ACCUMULATOR AND LINK RIGHT TWICE. The link register and accumulator are rotated two positions to the right.



Mnemonic	Octal Code	Instruction
CLA	7200	Clear accumulator
CLL	7100	Clear link register
CMA	7040	Complement accumulator
CML	7020	Complement link
IAC	7001	Increment accumulator
RAR	7010	Rotate accumulator and link right
RTR	7012	Rotate accumulator and link right twice
RAL	7004	Rotate accumulator left
RTL	7006	Rotate accumulator and link left twice
NOP	7000	No operation

COMBINING GROUP I INSTRUCTIONS

1. Only one of the shift instructions RAR, RTR, RAL, RTL may appear in a combined instruction.

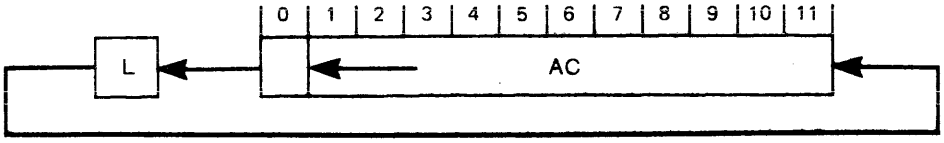
2. In a combined instruction, CLA and CLL, if specified, are executed first; CMA and CML, if specified, are executed next; IAC, if specified, is executed next; one of RAR, RTR, RAL, RTL, if specified, is executed last.

COMBINED INSTRUCTION EXAMPLES:

CMA IAC	Negate accumulator
CLA CMA	Load accumulator with -1
CLA CLL CMA RAL	Load accumulator with -2
CLA CML	Clear accumulator and complement link
CLA CLL CML IAC RTR	Load accumulator with -1024
CLA CLL CML RTL	Load accumulator with +2
NOP	No operation

Figure 3-2
GROUP I OPERATE INSTRUCTION FORMAT

RAL — ROTATE ACCUMULATOR AND LINK LEFT. The link register and accumulator are rotated one position to the left as illustrated below:



RTL — ROTATE ACCUMULATOR AND LINK LEFT TWICE. The link register and accumulator are rotated two positions to the left.

NOP — NO OPERATION. No operation is performed.

Combining Group I Instructions

Group I operate instructions may be combined into a composite instruction subject to the following rules:

1. NOP is not combinable.

2. Only one of the shift instructions RAR, RTR, RAL, RTL may appear in a combined instruction.

3. The execution sequence for a composite Group I instruction is defined as follows:

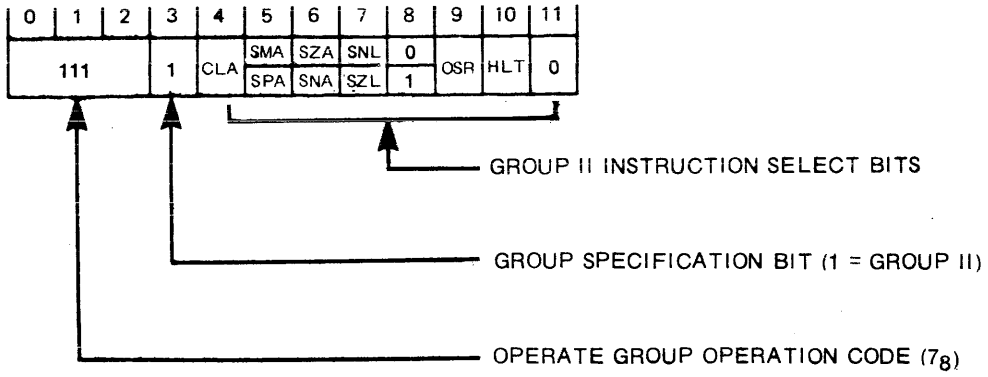
   a. CLA and CLL, if specified, are executed first.
   b. CMA and CML, if specified, are executed next.
   c. IAC, if specified, is executed next.
   d. One of RAR, RTR, RAL, or RTL, if specified, is executed last.

Group II Operate Instructions

Group II operate instructions provide test and skip capability based on the contents of the accumulator and link register. The format of the Group II operate instructions is illustrated in Figure 3-3. The following information describes the operation of each individual instruction.

CLA — CLEAR ACCUMULATOR. Each bit in the accumulator is set to zero.

SMA — SKIP ON MINUS ACCUMULATOR. If the contents of the accumulator are less than zero, the next instruction in sequence is skipped.



Mnemonic	Octal Code	Instruction
CLA	7600	Clear accumulator
SMA	7500	Skip on minus accumulator
SZA	7440	Skip on zero accumulator
SNL	7420	Skip on non-zero link
SPA	7510	Skip on positive accumulator
SNA	7450	Skip on non-zero accumulator
SZL	7430	Skip on zero link
OSR	7404	Inclusive "OR" switch register with accumulator
HLT	7402	Halt
SKP	7410	Unconditional skip

COMBINING GROUP II INSTRUCTIONS

1. For the skip group SMA, SZA, and SNL, a combination of these instructions will result in a skip only when at least one of the specified skip conditions is true.

2. For the skip group SPA, SNA, and SZL, a combination of these instructions will result in a skip only when all specified skip conditions are true.

3. Only members of one skip group may appear in a combined instruction.

4. SKP is combinable only with CLA, OSR, and HLT.

5. In a combined instruction, skip instructions are executed first; CLA, if specified, is executed next; OSR, if specified, is executed next; and HLT, if specified, is executed last.

COMBINED INSTRUCTION EXAMPLES:

SMA SZA – Skip if accumulator is negative or zero
CLA OSR – Transfer switch register into accumulator
SZA CLA – Skip if accumulator is zero and clear accumulator
CLA HLT – Clear accumulator and halt

Figure 3-3
GROUP II OPERATE INSTRUCTION FORMAT

SPA	SKIP ON POSITIVE ACCUMULATOR. If the contents of the accumulator are greater than or equal to zero, the next instruction in sequence is skipped.
SZA	SKIP ON ZERO ACCUMULATOR. If the contents of the accumulator are zero, the next instruction in sequence is skipped.
SNA	SKIP ON NON-ZERO ACCUMULATOR. If the contents of the accumulator are non-zero, the next instruction in sequence is skipped.
SNL	SKIP ON NON-ZERO LINK. If the link register does not contain a zero, the next instruction in sequence is skipped.
SZL	SKIP ON ZERO LINK. If the link register contains a zero, the next instruction in sequence is skipped.
SKP	UNCONDITIONAL SKIP. The next instruction in sequence is skipped.
OSR	INCLUSIVE "OR" SWITCH REGISTER WITH ACCUMULATOR. The console switch register is inclusive OR'ed with the contents of the accumulator and the result is retained in the accumulator.
HLT	HALT. Causes the computer to halt at the conclusion of the current machine cycle.

Combining Group II Instructions

Group II operate instructions, subject to the following rules, may be combined into a composite instruction.

1. Skip group instructions SPA, SNA, and SZL. A combination of these instructions will result in a skip only when all specified skip conditions are true.

2. Skip group instructions SMA, SZA, and SNL. A combination of these instructions will result in a skip only when at least one of the specified skip conditions is true.

3. Only members of one skip group may appear in a combined instruction.

4. SKP is combinable only with CLA OSR, and HLT.

5. The execution sequence for a composite Group II operate instruction is defined as follows:

   a. Skip instructions are executed first.
   b. CLA, if specified, is executed next.

c. OSR, if specified, is executed next.
d. HLT, if specified, is executed last.

INPUT-OUTPUT INSTRUCTIONS

The MP12 input-output instructions enable data transfer between the computer and peripheral units by way of the accumulator. Each input-output instruction consists of a 3-bit operation code, a 6-bit device address, and a 3-bit function code. Input-output instructions are described in Chapter V, "Input-Output Interface."

CHAPTER IV
OPERATION

INTRODUCTION

This chapter describes the operation of the FABRI-TEK MP12 Microprocessor. It discusses the layout of the operating console, console controls and indicators and the functions which may be performed at the operating console.

OPERATING CONSOLE

The operating console, mounted on the front of the processor enclosure, contains all controls and indicators necessary for the operation of the processor. The console frame measures 9.5 in. by 2.125 in. Figure 4-1 depicts the layout of the FABRI-TEK MP12 operating console.

CONTROLS AND INDICATORS

The following information describes the controls and indicators on the FABRI-TEK MP12 operating console.

SWITCH REGISTER	The Switch Register consists of twelve data entry switches that are used to manually alter the contents of the accumulator, program counter, or memory data register. The switch register can also be read under program control.
DISPLAY REGISTER	The Display Register consists of twelve indicators that display the contents of the register selected by the Display switch.
RUN SWITCH	The Run switch, when toggled, causes the processor to commence instruction execution beginning at the address contained in the program counter.
HALT SWITCH	The Halt switch, when toggled, causes the processor to stop instruction execution; the program counter contains the address of the next instruction to be executed. When the processor is halted, toggling the Halt switch causes the instruction located at the address contained in the program counter to be executed.

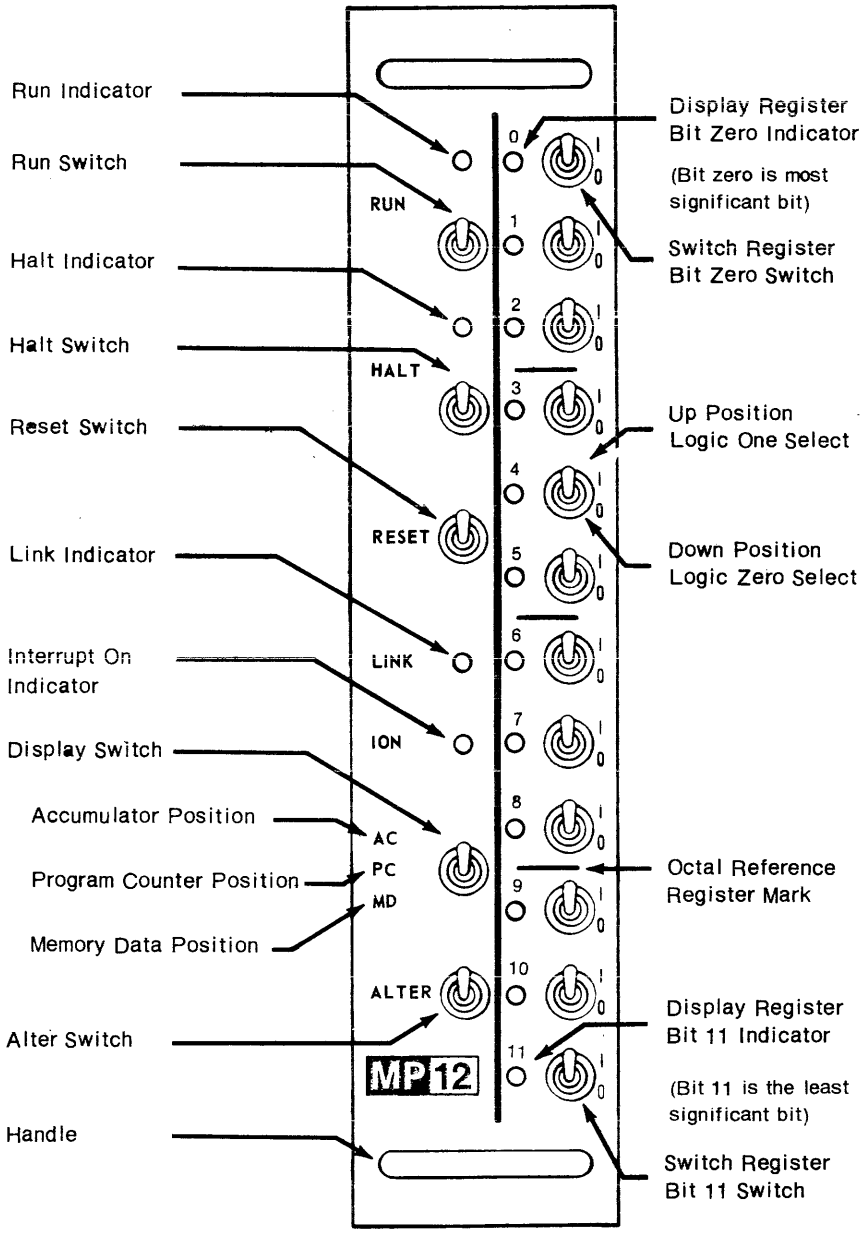


Figure 4-1
MP12 OPERATING CONSOLE

RESET SWITCH	The Reset switch, when toggled, generates a master reset condition. The processor is halted, all internal registers are set to zero, the interrupt facility is disabled, the input-output interface is initialized, and the program counter is set to 200_8. The Reset switch also functions as an indicator test in that all indicators are illuminated when the Reset switch is toggled.
RUN INDICATOR	The Run indicator is illuminated whenever the processor is in the RUN mode.
HALT INDICATOR	The Halt indicator is illuminated whenever the processor is in the HALT mode.
LINK INDICATOR	The Link indicator displays the content of the 1-bit link register.
ION INDICATOR	The ION indicator is illuminated when the interrupt facility is enabled.
DISPLAY SWITCH	The Display switch selects the register to be displayed in the display register. The accumulator (AC), program counter (PC) or memory data register (MD) can be selected. AC POSITION. When the display switch is set to AC, the contents of the accumulator are displayed in the display register. PC POSITION. When the display switch is set to PC, the contents of the program counter are displayed in the display register. MD POSITION. When the display switch is set to MD, the contents of memory at the address contained in the program counter are displayed in the display register.
ALTER SWITCH	The Alter switch, when toggled, causes the contents of the switch register to be copied into the register selected by the display switch, or the memory location contained in the program counter if the display switch is set to MD.

REMOTE SIGNAL SOURCES

Sources for the run, halt, and reset signals may be remotely located up to 100 feet from the system enclosure. Leads for these signals are present at the input-

output interface connector. An additional signal lead, LOAD, is also provided for use in conjunction with a ROM-installed loader program. The load signal causes RESET to occur, the program counter to be set to 7777_8, and RUN to be executed.

CHAPTER V
INPUT-OUTPUT INTERFACE

INTRODUCTION

This chapter describes the FABRI-TEK MP12 input-output interface in general. It discusses input-output channels, input-output controllers and devices, input-output instructions and functions, and concludes with a description of the FABRI-TEK MP12 interrupt facility.

INPUT-OUTPUT CHANNELS

The FABRI-TEK MP12 input-output interface includes two input-output channels, a processor input-output (PIO) channel, and a direct memory access (DMA) input-output channel. The processor communicates with the PIO channel and the DMA channel communicates with the memory via the data input bus. Communications between the processor and the PIO channel are established by executing a series of input-output transfer (IOT) instructions which address a specified input-output controller and device. Addresses range from 01_8 to 77_8 and permit a maximum of 63 separate input-output addresses to be specified.

PROCESSOR INPUT-OUTPUT (PIO) CHANNEL

The processor input-output channel is used to communicate with low-speed, character-oriented devices which are asynchronous in nature. Each item of data is transferred to or from an addressed device, via the accumulator, by executing a separate IOT instruction for each transfer. IOT instructions, in addition to transferring data, are also used to test the status of a device and to initiate input or output operations. The PIO channel is capable of accommodating devices with transfer rates of up to 66,000 words-per-second.

DIRECT MEMORY ACCESS (DMA) CHANNEL

The direct memory access input-output channel is used to communicate with high-speed, record-oriented devices such as disk units and magnetic tape equipment. DMA input-output requests require control and address information to be transmitted to a selected DMA controller via the PIO channel. A series of IOT commands are executed to consumate the transfer of information. Once started, a DMA input-output operation proceeds to completion independently of the processor. The DMA channel is capable of sustaining burst transfer rates of up to 666,666 words-per-second.

INPUT-OUTPUT DEVICE CONTROLLERS

Input-output controllers consist of the necessary logic circuitry required to interconnect one or more peripheral devices with the input-output interface. Each input-output controller functions as either a PIO or a DMA controller depending upon which channel is interfaced. An input-output controller is normally identified with a single device; however, certain types of controllers may accommodate multiple devices of the same physical type.

DIRECT MEMORY ACCESS DEVICE CONTROLLERS

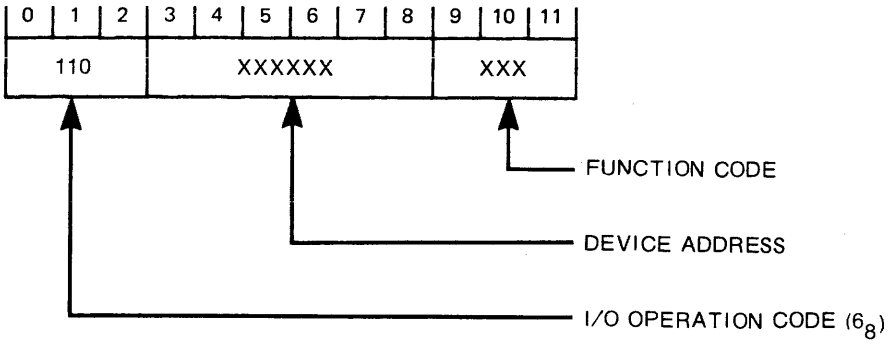
DMA controllers interface one or more devices with the DMA channel and communicate with memory via the data input bus, and with the processor via the PIO channel.

PROCESSOR INPUT-OUTPUT DEVICE CONTROLLERS

PIO controllers interface one or more devices with the PIO channel and communicate with the processor via input-output transfer (IOT) instructions.

INPUT-OUTPUT INSTRUCTIONS

Input-output instructions are used to communicate with a selected input-output controller and device via the processor input-output channel. Each input-output instruction consists of a 3-bit operation code, a 6-bit device address, and a 3-bit function code as illustrated.



INPUT-OUTPUT TRANSFER (IOT) INSTRUCTION FORMAT

The function code, as specified in bit positions 9 through 11 of the input-output instruction word, is interpreted with respect to the following device states:

STATE	INTERPRETATION
AVAILABLE	A device is AVAILABLE provided that it is powered, on-line, properly enabled, and otherwise capable of operation.
READY	A device is READY provided that it is available and not in the process of performing a previously ordered input or output operation.
DONE	A device is DONE if it has generated an interrupt request to the processor, indicating that a previously ordered input or output operation has been completed. For input devices, the state DONE implies that data is present in the device data buffer. A DONE device is always READY, but not conversely.

For the standard FABRI–TEK teletype and high speed paper tape reader/punch interfaces, the operations performed by each input–output function are described below. The function code is decoded in the manner shown in Figure 5–1. Note that bits 9 and 10 of the function code result in the clearing of an interrupt request as follows:

Bit 9: When an input device is addressed and bit 9 of the function code is set to one, the interrupt request is cleared if the device is DONE.

Bit 10: When an input or output device is addressed and bit 10 of the function code is set to one, the interrupt request is cleared if the device is DONE.

FUNCTION CODE	INTERPRETATION
1_8	SKIP IF READY. If the addressed device is READY, the next instruction in sequence is skipped. If the device is not READY, no skip occurs.
2_8	START OPERATION. If the addressed device is READY, the next input or output operation is started. If the device is not READY, no operation is performed.
3_8	SKIP IF READY AND START OPERATION. If the addressed device is READY, the next instruction in sequence is skipped and the next input or output operation is started. If the device is not READY, no skip occurs and no operation is performed.

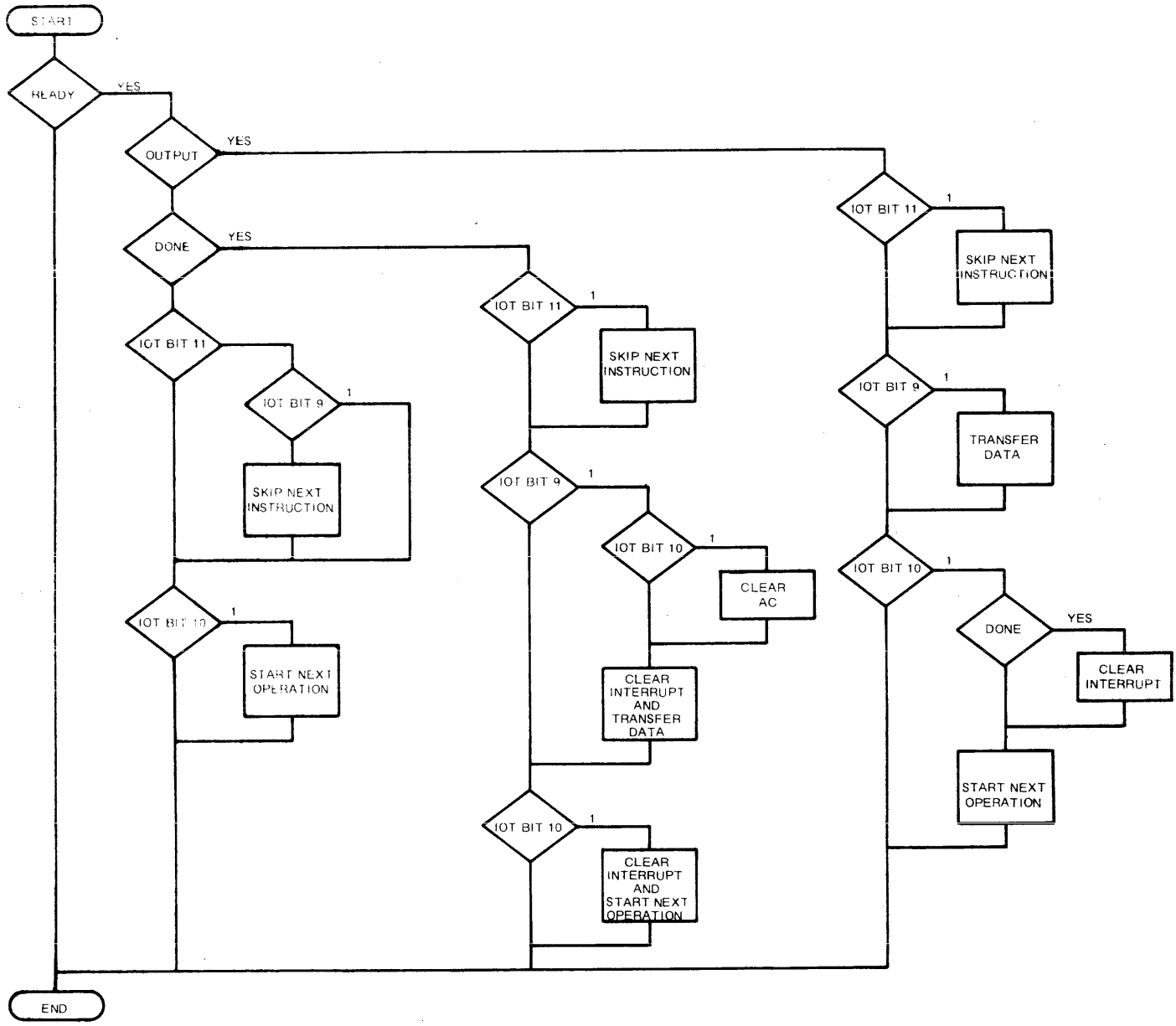


Figure 5-1
IOT FUNCTION DECODE

FUNCTION CODE	INTERPRETATION
4_8	TRANSFER DATA. For input devices, if the addressed device is DONE, the contents of the device data buffer are inclusive OR'ed with the accumulator and the result retained in the accumulator. If the device is not DONE, no operation is performed. For output devices, if the device is READY, the contents of the accumulator are transferred to the device data buffer. If the device is not READY, no operation is performed.
5_8	SKIP IF DEVICE READY AND TRANSFER DATA. For input devices, if the addressed device is DONE, the next instruction in sequence is skipped, the contents of the device buffer are inclusive OR'ed with the accumulator, and the result is retained in the accumulator. If the device is not DONE, no skip occurs and no operation is performed. For output devices, if the addressed device is READY, the next instruction in sequence is skipped and the contents of the accumulator are transferred to the device data buffer. If the device is not READY, no skip occurs and no operation is performed.
6_8	TRANSFER DATA AND START NEXT OPERATION. For input devices, if the addressed device is DONE, the accumulator is cleared, the contents of the device data buffer are inclusive OR'ed with the accumulator, and the result is retained in the accumulator. If the device is either READY or DONE, the next input operation is started. If the device is not READY, no operation is performed. For output devices, if the addressed device is READY, the contents of the accumulator are transferred to the device data buffer and the next output operation is started. If the device is not READY, no operation is performed.
7_8	SKIP IF DEVICE READY, TRANSFER DATA, AND START NEXT OPERATION. For input devices, if the addressed device is DONE, the next instruction in sequence is skipped, the accumulator is cleared, the contents of the device data buffer are inclusive OR'ed with the accumulator, and the result is retained in the accu-

mulator. If the device is READY and not DONE, no skip occurs but the next input operation is started. If the device is not READY, no skip occurs and no operation is performed.

For output devices, if the addressed device is READY, the next instruction in sequence is skipped, the contents of the accumulator are transferred to the device data buffer, and the next output operation is started. If the device is not READY, no skip occurs and no operation is performed.

The following examples illustrate the use of the FABRI-TEK MP12 input-output (IOT) instructions, and do not utilize the interrupt facility. The examples are formulated in symbolic assembly language notation as described in Chapter VI, "Assembly Language."

Example 1: Print the letter "A" on the teletype (Teletype printer/punch address = 4).

```
IØT   041   /SKIP IF PRINTER READY
JMP   .-1   /NØT READY, TRY AGAIN
TAD   LET   /READY, FETCH THE LETTER "A"
IØT   046   /TRANSFER TØ PRINTER AND
            /START PRINT ØPERATIØN
---
LET,0301    /ASCII "A"
```

Example 2: The same operation performed in Example 1 may be performed with a single input-output instruction using the function code 7 as follows:

```
TAD   LET   /FETCH THE LETTER "A"
IØT   047   /SKIP IF READY AND TRANSFER
JMP   .-1   /NØT READY, TRY AGAIN
---         /PRINT ØPERATIØN STARTED
---
LET,0301    /ASCII "A"
```

Example 3: Read a character from the teletype keyboard (Keyboard/reader address = 3).

```
IØT   032   /START KEYBØARD
---
---
IØT   031   /SKIP IF KEYBØARD READY
JMP   .-1   /NØT READY, TRY AGAIN
IØT   036   /READ CHARACTER AND
            /START NEXT READ ØPERATIØN
```

Example 4: The same operation performed in Example 3 may be performed with a single input-output instruction using the function code 7 as follows:

```
-------------------------------------------
.  IØT   037  /READ WHEN READY AND SKIP    .
.  JMP   .-1  /NØT READY, TRY AGAIN        .
.  ---        /CHARACTER READ AND THE NEXT .
.             /INPUT ØPERATIØN STARTED     .
-------------------------------------------
```

INTERRUPT FACILITY

The FABRI-TEK MP12 interrupt facility provides a processor interrupt when an input-output device is ready to send or receive data, or a power failure is detected. If the interrupt facility is enabled when an interrupt occurs, the processor disables the interrupt facility, stores the contents of the program counter in location 0, and executes location 1. The following instructions are used to control the FABRI-TEK MP12 interrupt facility.

MNEMONIC	INSTRUCTION
ION (Octal Code 6001)	TURN INTERRUPT SYSTEM ON. Enables the interrupt system after a one instruction delay.
IOF (Octal Code 6002)	TURN INTERRUPT SYSTEM OFF. Disables the interrupt system. No interrupts can occur until the interrupt system is enabled.
SPL (Octal Code 6004)	SKIP ON POWER LOW. The next instruction in sequence is skipped if power is low. This instruction is used to identify interrupts originated by the automatic power fail detection circuitry in the optional FABRI-TEK MP12 power supply module.
CON (Octal Code 6774)	TURN LINE FREQUENCY CLOCK ON. Enables the line frequency clock in the optional power supply module. When enabled, the clock will generate a processor interrupt each 16 2/3 milliseconds until disabled.
COF (Octal Code 6772)	TURN LINE FREQUENCY CLOCK OFF. Disables the line frequency clock.
SCD (Octal Code 6771)	SKIP ON CLOCK DONE AND CLEAR INTERRUPT. The next instruction in sequence is skipped if the line frequency clock has generated an interrupt request.

CHAPTER VI
ASSEMBLY LANGUAGE

INTRODUCTION

This chapter describes the FABRI-TEK MP12 assembly language. It discusses the characteristics of symbolic assembly language programs and describes the mechanism by which such programs are translated into executable machine programs. Chapter topics include the Assembler, the Assembly Language Character Set, Statements, Expressions, Machine Instructions, Error Processing, and the Assembly Listing.

ASSEMBLER

Programs written in FABRI-TEK MP12 assembly language are translated by an assembler program into executable machine programs. The assembly process is basically one of converting symbolic instructions into binary machine instructions, generating data, assigning storage locations for machine instructions and data, and performing auxiliary functions necessary to produce an executable machine program.

An assembly language program consists of a series of symbolic statements which are normally written on coding forms and later transcribed to paper tape for input to the assembler. The assembler reads the source tape containing the symbolic program and produces a printed listing which contains the machine code resulting from each statement, and a punched paper tape, or object tape, containing the machine program. The object tape can then be loaded into the computer and the program executed. Three separate passes or readings of the source program are required to complete the assembly process. The function of each assembly pass is described below:

Pass 1 – The assembler reads the source tape and constructs an internal symbol table which records the value of each symbol in the program.

Pass 2 – The assembler punches an object tape containing the assembled machine program.

Pass 3 – The assembler prints a listing of the assembled machine program.

ASSEMBLY LANGUAGE CHARACTER SET

Program statements are constructed with characters taken from the following character set:

Letters: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Digits: 0 1 2 3 4 5 6 7 8 9

Special Characters: space ! '' # % & ' ( ) + , - : < > ? @ [/] ↑ ←

All other characters, except for the following, are ignored.

Slash (/):	Indicates the start of a comment string.
Carriage-Return:	Indicates the end of a symbolic statement.
Semicolon (;):	Same as carriage-return unless appearing within a comment string. Allows multiple statements to be coded on the same physical source line.
Equal sign (=):	Used to define equality symbols.
Asterisk (*):	Used to specify location counter value.
Rubout:	Ignored. May be used to overpunch preparation errors.
Dollar-sign ($):	The dollar sign ($) is used to indicate the last physical statement of the program. It must appear as the first non-blank character of the last statement.
Apostrophe ('):	Indicates a character string.

STATEMENTS

The statement is the basic unit used to construct assembly language programs. Each statement begins in character position one of a source line and is terminated by a carriage-return, or semi-colon (;). Use of the semicolon enables multiple statements to be coded on the same physical source line. If a statement extends past character position 72, the assembler ignores all succeeding characters until a carriage-return is encountered.

The text of a statement may be preceeded by one or more blank positions. The first non-blank position may then contain any one of the following characters:

Slash (/)
Apostrophe (')
Letter (A-Z)
Digit (0-9), plus (+), or minus (-)
Asterisk (*)

The treatment of each of these characters is described below.

SLASH

The appearance of a slash in the first non-blank statement position signifies a comment string. No action is taken except to reproduce the statement on the program listing.

```
/THIS IS A COMMENT STATEMENT
  /  THE ASSEMBLER IGNORES
/      COMMENT STATEMENTS
 / COMMENTS ARE PRECEDED BY A SLASH (/)
```

APOSTROPHE

The appearance of an apostrophe (') in the first non-blank statement position indicates the start of a character string. The ASCII value of each successive character following the initial apostrophe is output as a data word until a closing apostrophe is encountered. All characters following the closing apostrophe are ignored until a carriage-return or semicolon is encountered.

```
'A'       /LETTER "A"
'XYZ 123'
'CHARACTER STRING'
```

LETTER

The appearance of a letter in the first non-blank statement position signifies the presence of a label, an equality symbol, an assembler mnemonic, or an arithmetic expression.

LABEL. A label consists of at most eight letters and digits beginning with a letter and followed by a comma (,). Each label is assigned a value, during assembly pass 1, equal to the value of the program location counter at the time it is encountered. Refer to the discussion of the location counter in the "Asterisk" statement text. The first non-blank character following the comma can be a semicolon (;), carriage-return, or one of the characters listed above. If one of these characters is present, it is processed exactly as though it was the first non-blank statement character.

```
   TEMP,
LABEL151,
   A12345,
      X15B24,
```

EQUALITY SYMBOL. An equality symbol consists of at most eight letters and digits beginning with a letter and followed by an equal sign (=). An expression must appear to the right of the equal sign. The assembler evaluates the expression and assigns the value to the symbol on the left of the equal sign. All expression terms must be previously defined, and the expression must be terminated by a slash (/), semicolon (;), or carriage-return. Refer to the "Expressions" text for a discussion of expression formation and evaluation.

```
TEN = 10
TWELVE = TEN + 2
   TWENTY = TEN + TEN
NEG = 07041    /NEGATE OPERATOR
```

ASSEMBLER MNEMONIC. An assembler mnemonic consists of at most eight letters followed by a blank, slash (/), carriage-return, or semi-colon (;). If more than four letters are specified, only the first four are used in processing the mnemonic. Mnemonics are assigned for each FABRI-TEK MP12 instruction and are described in the "Machine Instructions" text. The remaining assembler mnemonics are described below.

MNEMONIC	MEANING
DECIMAL	Set decimal conversion mode. Normally, all numeric program data is treated in the following manner: Numbers preceded by a zero (0) are treated as octal while those not preceeded by a zero are treated as decimal. The DECIMAL mnemonic directs the assembler to regard all subsequent numeric data as decimal data.
OCTAL	Set octal conversion mode. The OCTAL mnemonic directs the assembler to regard all subsequent numeric data as octal data.

All characters following a DECIMAL or OCTAL mnemonic are ignored until a carriage-return or semicolon (;) is encountered.

```
-------------------------------------------
.  ØCTAL  /DECLARE ØCTAL CØNVERSIØN       .
.   DECIMAL;128;-512;+1024 /DECIMAL DATA  .
. ØCTAL;77;777;DECIMAL;99;999             .
.                                         .
-------------------------------------------
```

ARITHMETIC EXPRESSION. If a label, equality symbol, or assembler mnemonic is not present, the assembler assumes an expression is specified and attempts to evaluate it. If the evaluation is successful, the value of the expression is output as a data word. The expression must be terminated by a slash (/), semicolon (;), or carriage-return.

```
-------------------------------------------
.  LØØP + 6                               .
.     START+0200                          .
.   BUFF2 - BUFF1 + 1                     .
.  DATA3, DATA + 2/ LABELED EXPRESSIØN    .
-------------------------------------------
```

DIGIT, PLUS, OR MINUS

The appearance of a digit (0-9), a plus sign (+), or a minus sign (-) signifies the presence of an expression which is evaluated and the value output as a data word. The expression must be terminated by a slash (/), semicolon (;) or carriage-return.

```
-------------------------------------------
.  -2047                                  .
.      +999                               .
.   -SWITCH+3                             .
. 1 - TAG                                 .
-------------------------------------------
```

ASTERISK

The asterisk character controls the setting of the program location counter. The location counter is used by the assembler to assign machine instructions and data into consecutive memory addresses. The value of the location counter represents the physical memory address into which any data generated by the current statement is to reside when the machine program is loaded. The assembler increments the location counter by one for each instruction or data item assembled. Statement labels are assigned the value of the location counter, during pass 1 of the assembly, at the time they are encountered. The assembler initially sets the location counter to octal 200.

The asterisk must be followed by an expression. The expression is evaluated and the value assigned to the program location counter. Each term of the expression must be previously defined, and the expression must be terminated by a slash (/), semicolon (;), or carriage-return. Refer to the ''Expressions'' text for a discussion of expression formation and evaluation.

```
*0400   /SET LØCATIØN
   START, *0200
*START+128
BUFF,*.+72  /RESERVE 72 LØCATIØNS FØR
            /BUFFER AREA
```

Note that if a label appears in conjunction with an asterisk, it is assigned the value of the location counter prior to establishing the new value of the location counter. For example, if the value of the location counter was 0763 prior to the statement BUFF, *01000 then the value assigned to the label ''BUFF'' would be 0763.

EXPRESSIONS

Expressions are formed by combining ''terms'' from left to right using plus (+) and (-) signs. Blanks may appear before, between, and after terms; however, terms may not contain imbedded blanks. An expression must be terminated by a slash (/), semicolon (;) or carriage-return.

TERMS

Terms are the basic units used in constructing expressions. The following types of terms are defined:

PERIOD (.). The period is a term, which in statement context, represents the current value of the program location counter.

NUMERIC CONSTANT. A numeric constant is a self-defining term which is treated as an octal or decimal number depending upon the conversion mode in affect at the time the term is encountered. Initially, numeric terms beginning with a zero are treated as octal numbers. Those not beginning with a zero are treated as decimal numbers. The DECIMAL and OCTAL directives may, subsequently, be used to declare strict decimal or octal conversions. All numeric terms are converted modulo 4096. Octal numbers may not contain the digits 8 or 9.

```
.  0100      /ØCTAL 100                          .
.  2769      /DECIMAL 2769                       .
. 07776      /ØCTAL 7776                         .
.  4099      /3 [4099 MØD(4096) = 3]             .
.  8192      /0 [8192 MØD(4096) = 0]             .
```

SYMBOL. A symbol consists of up to eight letters and digits beginning with a letter. Symbols are defined by their appearance as statement labels or equality symbols. The value of a symbol, defined as a label, is the value of the location counter at the time the label was encountered. The value of a symbol, defined by equality, is the value of the expression appearing on the right of the equal sign.

```
.  SYMB     /FØUR LETTER SYMBØL                 .
.  P1234    /ØNE LETTER, FØUR DIGITS            .
.  P1QR23   /MIXED LETTERS AND DIGITS           .
.  2SYM     /ILLEGAL, FIRST CHARACTER IS        .
.           /NØT A LETTER                       .
.  X.15     /ILLEGAL, PERIØD IS NØT A           .
.           /LETTER ØR DIGIT                    .
.  Y12 Z    /ILLEGAL, IMBEDDED BLANK            .
```

EXPRESSION EVALUATION

Expressions are evaluated from left to right by combining the terms as indicated.

```
.  .+6            /VALUE ØF LØCATIØN CØUNTER .
.                 /PLUS SIX                  .
.  LØØP - 3       /VALUE ØF SYMBØL "LØØP"    .
.                 /MINUS THREE               .
.   -128          /TWØ'S CØMPLEMENT ØF       .
.                 /DECIMAL 128               .
.  A-B+C          /VALUE ØF SYMBØL "A" MINUS .
.                 /VALUE ØF SYMBØL "B" PLUS  .
.                 /VALUE ØF SYMBØL "C"       .
.  LAST - .       /VALUE ØF SYMBØL "LAST"    .
.                 /MINUS CURRENT VALUE ØF    .
.                 /LØCATIØN CØUNTER          .
```

MACHINE INSTRUCTIONS

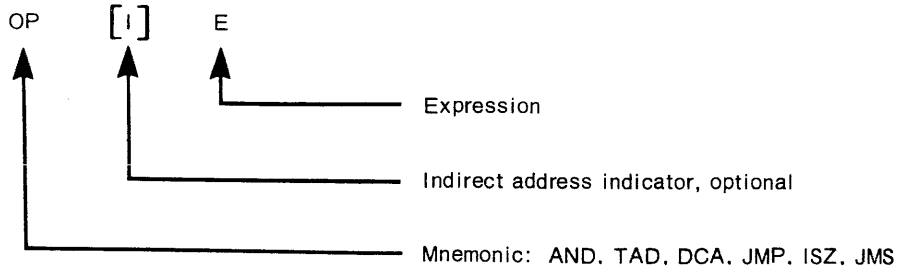
Each FABRI-TEK MP12 machine instruction is identified by a symbolic instruction mnemonic. The assembler recognizes each mnemonic and generates a binary machine instruction which corresponds to the symbolic instruction. In the following instruction descriptions, optional statement parameters are enclosed in square brackets ([ ]). All instruction statements may contain a label preceeding the instruction mnemonic, and a comment string preceeded by a slash (/). Labels and comment strings are not depicted in the following general instruction format descriptions.

MEMORY REFERENCE INSTRUCTIONS

Memory reference instructions consist of the following six instructions:

MNEMONIC	INSTRUCTION
AND	Logical ''AND''
TAD	Two's complement Add
DCA	Deposit and clear accumulator
JMP	Jump
ISZ	Increment and skip if zero
JMS	Jump to subroutine

Each memory reference instruction is coded in the format:



At least one blank position must separate each of the above fields. The value of the expression E represents the primary address of the instruction operand. If the I indicator is present, the assembler sets the address mode bit in the instruction to 1 to specify indirect addressing. If the I parameter is not present, the address mode bit is set to zero to specify direct addressing. The expression E is evaluated and if the value lies within the same page as the location counter, the page specification bit in the instruction is set to one; specifying current page addressing. If the value lies within the first 128 memory locations, the page specification bit is set to zero to specify base page addressing. In either the base or current page specification mode, the least significant seven bits of the value of the expression are inserted into bit positions 5 through 11 of the instruction word.

```
        AND MASK    /MASK ØFF CERTAIN BITS
FETCH, TAD LØC       /FETCH DATA
  DCA  SAVE          /SAVE DATA
 ISZ I 010    /INCREMENT        PØINTER
   JMS  SUB     /JUMP TØ SUBRØUTINE
 JMP I SUB  /RETURN FRØM SUBRØUTINE
```

OPERATE INSTRUCTIONS, GROUP I

The Group I operate instructions consist of the following:

MNEMONIC	INSTRUCTION
CLA	Clear accumulator
CLL	Clear link
CMA	Complement accumulator
CML	Complement link
IAC	Increment accumulator
RAR	Rotate accumulator and link right
RAL	Rotate accumulator and link left
RTR	Rotate accumulator and link right twice
RTL	Rotate accumulator and link left twice
NOP	No operation

The Group I operate instructions are coded in the format:

$$OP_1\ [OP_2]\ldots[OP_K]$$

OP_1 through OP_K represent Group I mnemonics which are combinable to form a composite Group I instruction. Each of the mnemonics must be separated by at least one blank position. The rules of combination for Group I instructions are summarized below.

1. NOP is not combinable.

2. Only one of the shift instructions RAR, RTR, RAL, RTL may appear in a combined instruction.

3. The execution sequence for a composite Group I instruction is defined as follows:

   a. CLA and CLL, if specified, are executed first.
   b. CMA and CML, if specified, are executed next.
   c. IAC, if specified, is executed next.
   d. One of RAR, RTR, RAL, RTL, if specified, is executed last.

```
.  CLA     /CLEAR ACCUMULATØR                .
.   CLA CLL CML /CLEAR AC AND SET LINK       .
. CMA IAC  /NEGATE ACCUMULATØR               .
.  CLA CLL CML RTL /LØAD AC WITH +2          .
.    NØP    /NØ ØPERATIØN                    .
.  INIT, CLA CMA  /LØAD AC WITH 7777         .
```

OPERATE INSTRUCTIONS, GROUP II

The Group II operate instructions are listed below.

MNEMONIC	INSTRUCTION
CLA	Clear accumulator
SMA	Skip on minus accumulator
SPA	Skip on positive accumulator
SZA	Skip on zero accumulator
SNA	Skip on non-zero accumulator
SNL	Skip on non-zero link
SZL	Skip on zero link
SKP	Skip
OSR	Inclusive ''OR'' switch register with accumulator
HLT	Halt

The Group II operate instructions are coded in the format:

$$OP_1\ [OP_2]\ldots[OP_K]$$

OP_1 through OP_K represent Group II mnemonics which are combinable to form a composite Group II instruction. Each of the mnemonics must be separated by at least one blank position. The rules of combination for Group II instructions are summarized below.

1. For the skip group SPA, SNA, and SZL, a combination of these instructions will result in a skip only when all specified skip conditions are true.

2. For the skip group SMA, SZA, and SNL, a combination of these instructions will result in a skip only when at least one of the specified skip conditions is true.

3. Only members of one skip group may appear in a combined instruction.

4. SKP is combinable only with CLA, OSR, and HLT.

5. The execution sequence for a combined Group II operate instruction is defined as follows:

   a. Skip instructions are executed first.
   b. CLA, if specified, is executed next.

c. OSR, if specified, is executed next.
d. HLT, if specified, is executed last.

```
SMA SZA    /SKIP IF AC NEGATIVE ØR ZERØ
 TEST, SNA SZL  /SKIP IF AC IS NØN-ZERØ
                /AND LINK IS ZERØ
 CLA  ØSR  /TRANSFER SWITCH REGISTER
           /CØNTENTS TØ ACCUMULATØR
  SNA CLA   /SKIP IF AC IS NØN-ZERØ AND
            /CLEAR AC
HALT, CLA HLT  /CLEAR AC AND HALT
```

INPUT/OUTPUT AND INTERRUPT INSTRUCTIONS

The Input/Output and interrupt instructions consist of the following:

MNEMONIC	INSTRUCTION
IOT	I/O Transfer
ION	Enable interrupt system
IOF	Disable interrupt system
SPL	Skip if power is low
CON	Turn clock on
COF	Turn clock off
SCD	Skip if clock done

The IOT instruction is coded in the format:

```
IOT  E
```

E is an expression that must be separated from the IOT mnemonic by at least one blank position. The expression E is evaluated, and the least significant nine bits of the value inserted into bit positions 3 through 11 of the instruction. Bits 3-8 represent a device address, and bits 9-11 represent one of seven input-output function codes as described in the Chapter V Input-Output information.

The interrupt instructions ION, IOF, SPL, CON, COF, and SCD are coded in the format:

```
OP
```

where OP is one of the above mnemonics. All characters following the mnemonic OP are ignored until a semicolon (;) or carriage-return is encountered.

```
 IØT 031  /SKIP IF KEYBØARD/READER READY
  XFER, IØT 046  /PRINT CHARACTER
   IØN  /ENABLE INTERRUPT FACILITY
    IØF  /TURN INTERRUPTS ØFF
TEST, SPL /SKIP IF PØWER IS LØW
```

ERROR PROCESSING

Errors detected during the assembly process result in an error flag being printed to the left of the statement which originated the error. The following information describes the error flags.

ERROR FLAG	MEANING
S	STATEMENT ERROR. An illegal or unexpected character was encountered in processing the current statement.
P	PAGE ERROR. A memory reference instruction operand does not lie within the current page or the base page, as required. The instruction cannot be assembled in its present form. A current page address of 0177_8 is assumed.
I	ILLEGAL COMBINATION. An illegal instruction combination has been specified in the current statement.
D	DOUBLY DEFINED SYMBOL. A statement symbol has been previously defined. The value assigned at the first definition is used. This error is only indicated during assembly pass 1.
U	UNDEFINED SYMBOL. A program symbol has not been defined in any statement. The value 0 is assigned.
F	SYMBOL TABLE FULL. No further symbols are stored.

If multiple errors result from the same statement, only the last error detected is indicated.

ASSEMBLY LISTING

The assembly listing is produced during pass 3. Each statement is printed in the following format:

```
-----------------------------------------------
.  PRINT PØSITIØN -     1: ERRØR FLAG          .
.                       2: BLANK               .
.                    3 - 6: LØCATIØN (ØCTAL)   .
.                       7: BLANK               .
.                    8 -11: DATA (ØCTAL)       .
.                      12: BLANK                .
.                    13-16: STATEMENT NUMBER    .
.                      17: BLANK                .
.                    18-72: PRØGRAM STATEMENT   .
-----------------------------------------------
```

The listing is printed with 52 lines per page; each page is numbered in decimal. Eleven inch page separation marks, consisting of six dashed lines, are printed to aid manual page separation. A sample listing is provided below.

```
PAGE  1


                    1 /------------------------------------------------------
                    2 /                                                      .
                    3 /      BINARY TØ ØCTAL CØNVERSIØN SUBRØUTINE           .
                    4 /                                                      .
                    5 /      CALLING SEQUENCE:                               .
                    6 /                                                      .
                    7 /      TAD   ...        /AC = NUMBER TØ BE CØNVERTED   .
                    8 /      JMS   BØC        /CALL CØNVERSIØN RØUTINE       .
                    9 /      ...              /ADDRESS ØF STØRAGE AREA TØ    .
                   10 /                       /RECEIVE 4 ØCTAL CHARACTERS    .
                   11 /      ...              /RETURN PØINT                  .
                   12 /                                                      .
                   13 /------------------------------------------------------
0200 0000          14 BØC,   0
0201 3231          15        DCA   T0         /STØRE VALUE
0202 1600          16        TAD I BØC        /FETCH ADDRESS ØF STØRAGE AREA
0203 3232          17        DCA   T1         /STØRE
0204 2200          18        ISZ   BØC
0205 1226          19        TAD   M4         /INITIALIZE CØUNT
0206 3233          20        DCA   T2
0207 1231          21 BØC2,  TAD   T0         /FETCH VALUE
0210 7006          22        RTL              /EXTRACT ØCTAL DIGIT
0211 7006          23        RTL
0212 3234          24        DCA   T3
0213 1234          25        TAD   T3
0214 7010          26        RAR
0215 3231          27        DCA   T0
0216 1234          28        TAD   T3
0217 0227          29        AND   07
0220 1230          30        TAD   0260       /CØNVERT TØ ASCII CHARACTER
0221 3632          31        DCA I T1         /STØRE CHARACTER
0222 2232          32        ISZ   T1         /INCREMENT ADDRESS
0223 2233          33        ISZ   T2         /INCREMENT CØUNT, FINISHED
0224 5207          34        JMP   BØC2       /NØ, CØNTINUE
0225 5600          35        JMP I BØC        /YES, RETURN
                   36 /------------------------------------------------------
0226 7774          37 M4,    -4
0227 0007          38 07,    7
0230 0260          39 0260,  0260
0231 0000          40 T0,    0                /VALUE
0232 0000          41 T1,    0                /ADDRESS
0233 0000          42 T2,    0                /CØUNT
0234 0000          43 T3,    0                /TEMP
```

APPENDIX A
FABRI-TEK MP12 INSTRUCTION SET

MEMORY REFERENCE INSTRUCTIONS

MNEMONIC SYMBOL	OPERATION CODE	EXECUTION TIME*	OPERATION DESCRIPTION
AND Y	0	Direct-3.0 Indirect-4.5	LOGICAL AND. This instruction generates the logical product of the contents of memory location Y and the contents of the accumulator. The result replaces the previous contents of the accumulator. The bit stored in the link register is not affected by the LOGICAL AND operation.
TAD Y	1	Direct-3.0 Indirect-4.5	TWO'S COMPLEMENT ADD. This instruction generates the arithmetic sum of the contents of memory location Y and the contents of the accumulator. The result replaces the previous contents of the accumulator. If the operation produces a carry from the most significant bit position, the link bit is complemented.
ISZ Y	2	Direct-3.0 Indirect-4.5	INCREMENT AND SKIP IF ZERO. This instruction adds one to the contents of memory location Y. If the result is zero the next instruction in sequence is skipped. The contents of the link register and accumulator are not affected by the INCREMENT AND SKIP IF ZERO operation.
DCA Y	3	Direct-3.0 Indirect-4.5	DEPOSIT AND CLEAR ACCUMULATOR. This instruction copies the contents of the accumulator into memory location Y and then clears the accumulator to zero. The bit stored in the link register is not affected by the DEPOSIT AND CLEAR ACCUMULATOR operation.
JMS Y	4	Direct-3.0 Indirect-4.5	JUMP TO SUBROUTINE. This instruction copies the address of the next instruction in sequence into memory location Y and transfers program control to location Y+1. The contents of the link register and accumulator are not affected by JUMP TO SUBROUTINE operation.
JMP Y	5	Direct-1.5 Indirect-3.0	JUMP. This instruction transfers program control to location Y. The contents of the link register and accumulator are not affected by the JUMP operation.

*Time referenced in microseconds.

GROUP I OPERATE MICROINSTRUCTIONS

MNEMONIC SYMBOL	OCTAL CODE	EXECUTION TIME*	OPERATION DESCRIPTION
NOP	7000	3.0	NO OPERATION. This instruction performs no operation.
IAC	7001	3.0	INCREMENT ACCUMULATOR. This instruction adds one to the contents of the accumulator and replaces the previous contents of the accumulator with the result. If the operation produces a carry from the most significant bit position, the link bit is complemented.
RAL	7004	3.0	ROTATE ACCUMULATOR AND LINK LEFT. This instruction shifts the contents of the accumulator left one bit position. The bit shifted out of bit position 00 is shifted into the link register and the previous content of the link register is shifted into bit position 11.
RAR	7010	3.0	ROTATE ACCUMULATOR AND LINK RIGHT. This instruction shifts the contents of the accumulator right one bit position. The bit shifted out of bit position 11 is shifted into the link register and the previous content of the link register is shifted into bit position 0.
RTL	7006	3.0	ROTATE ACCUMULATOR AND LINK TWICE LEFT. This instruction is equivalent to two sequential RAL instructions.
RTR	7012	3.0	ROTATE ACCUMULATOR AND LINK TWICE RIGHT. This instruction is equivalent to two sequential RAR instructions.
CML	7020	3.0	COMPLEMENT LINK. This instruction complements the bit stored in the link register. The contents of the accumulator are not affected by the COMPLEMENT LINK operation.
CMA	7040	3.0	COMPLEMENT ACCUMULATOR. This instruction generates the one's complement of the contents of the accumulator. The result replaces the previous contents of the accumulator. The bit stored in the link register is not affected by the COMPLEMENT ACCUMULATOR operation.
CLL	7100	3.0	CLEAR LINK. This instruction sets the content of the link register to zero. The contents of the accumulator are not affected by the CLEAR LINK operation.
CLA	7200	3.0	CLEAR ACCUMULATOR. This instruction sets the contents of the accumulator to zero. The bit stored in the link register is not affected by the CLEAR ACCUMULATOR operation.

*Time referenced in microseconds.

GROUP II OPERATE MICROINSTRUCTIONS

MNEMONIC SYMBOL	OCTAL CODE	EXECUTION TIME*	OPERATION DESCRIPTION
HLT	7402	3.0	HALT. This instruction stops instruction execution. If the HALT instruction is combined with other Group II microinstructions, the HALT instruction is the last operation to be performed.
OSR	7404	3.0	OR SWITCH REGISTER. This instruction generates the logical sum of the contents of the accumulator and the value set in the control panel switches. The result replaces the previous contents of the accumulator. The bit stored in the link register is not affected by the OR SWITCH REGISTER operation.
SKP	7410	3.0	Skip.
SNL	7420	3.0	Skip of L $\neq$ 0.
SZL	7430	3.0	Skip if L = 0.
SZA	7440	3.0	Skip if AC = 0.
SNA	7450	3.0	Skip if AC $\neq$ 0.
SZA SNL	7460	3.0	Skip if AC = 0, or L = 1, or both.
SNA SZL	7470	3.0	Skip if AC $\neq$ and L = 0.
SMA	7500	3.0	Skip if AC $<$ 0.
SPA	7510	3.0	Skip if AC $\geq$ 0.
SMA SNL	7520	3.0	Skip if AC $<$0, or L = 1, or both.
SPA SZL	7530	3.0	Skip if AC $\geq$ 0 and L = 0.
SMA SZA	7540	3.0	Skip if AC $\leq$ 0.
SPA SNA	7550	3.0	Skip if AC $\geq$ 0.
CLA	7600	3.0	Clear AC.
SZA CLA	7640	3.0	Skip if AC = 0, then clear AC.
SNA CLA	7650	3.0	Skip if AC $\neq$ 0, then clear AC.
SMA CLA	7700	3.0	Skip if AC $<$ 0, then clear AC.
SPA CLA	7710	3.0	Skip if AC $\geq$ 0, then clear AC.

*Time referenced in microseconds.

INPUT-OUTPUT INSTRUCTIONS

MNEMONIC SYMBOL	OPER. CODE	EXECUTION TIME*	OPERATION DESCRIPTION
IOT Y	6	3.0	INPUT-OUTPUT TRANSFER. The input-output function specified by the least significant three bits of Y is performed with respect to the device addressed by the most significant 6 bits of Y ($000_8 \leq Y \leq 777_8$).

INTERRUPT INSTRUCTIONS

MNEMONIC SYMBOL	OCTAL CODE	EXECUTION TIME*	OPERATION DESCRIPTION
ION	6001	3.0	TURN INTERRUPTS ON. This instruction enables the interrupt system after a one instruction delay.
IOF	6002	3.0	TURN INTERRUPTS OFF. This instruction disables the interrupt system.
SPL	6004	3.0	SKIP IF POWER LOW. This instruction causes the next instruction in sequence to be skipped if the power low signal from the power supply is asserted.
CON	6774	3.0	TURN LINE FREQUENCY CLOCK ON. Enables the line frequency clock in the optional power supply module. When enabled, the clock will generate a processor interrupt each 16 2/3 milliseconds until disabled.
COF	6772	3.0	TURN LINE FREQUENCY CLOCK OFF. Disables the line frequency clock.
SCD	6771	3.0	SKIP ON CLOCK DONE AND CLEAR INTERRUPT. The next instruction in sequence is skipped if the line frequency clock has generated an interrupt request.

*Time referenced in microseconds.

APPENDIX B
USASCII CHARACTER SET

OCTAL CODE	CHARACTER NAME	ASCII CHARACTER	TELETYPE CHARACTER	KEY OR KEY COMBINATIONS
220	Null/Idle	NULL	---	CTRL @
201	Start of Message	SOM	---	CTRL A
202	End of Address	EOA	---	CTRL B
203	End of Message	EOM	---	CTRL C
204	End of Transmission	EOT	---	CTRL D
205	Who Are You	WRU	---	CTRL E
206	Are You	RU	---	CTRL F
207	Audible Signal	BELL	---	CTRL G
210	Format Effector	FE	---	CTRL H
211	Horizontal Tabulation	H TAB	---	CTRL I
212	Line Feed	LF	---	CTRL J
213	Vertical Tabulation	V TAB	---	CTRL K
214	Form Feed	FF	---	CTRL L
215	Carriage Return	CR	---	CTRL M
216	Shift Out	SO	---	CTRL N
217	Shift In	SI	---	CTRL O
220	Device Control Reversed for Data Line Escape	DC0	---	CTRL P
221	Device Control ON	DC1	---	CTRL Q
222	Device Control (TAPE) ON	DC2	---	CTRL R
223	Device Control OFF	DC3	---	CTRL S
224	Device Control (TAPE) OFF	DC4	---	CTRL T
225	Error	ERR	---	CTRL U
226	Synchronous Idle	SYNC	---	CTRL V
227	Logical End of Media	LEM	---	CTRL W
230	Separator, Information	S0	---	CTRL X
231	Separator, Data Delimiters	S1	---	CTRL Y
232	Separator, Words	S2	---	CTRL Z
233	Separator, Groups	S3	---	SHIFT CTRL K
234	Separator, Records	S4	---	SHIFT CTRL L
235	Separator, Files	S5	---	SHIFT CTRL M
236	Separator, Misc.	S6	---	SHIFT CTRL N
237	Separator, Misc.	S7	---	SHIFT CTRL O
240	Space	SP	Space	Space Bar
241	Exclamation Point	!	!	SHIFT !
242	Quotation Marks	"	"	SHIFT "
243	Number Sign	#	#	SHIFT #
244	Dollar Sign	$	$	SHIFT $

(CONTINUED)

OCTAL CODE	CHARACTER NAME	ASCII CHARACTER	TELETYPE CHARACTER	KEY OR KEY COMBINATIONS
245	Percent Sign	%	%	SHIFT %
246	Ampersand	&	&	SHIFT &
247	Apostrophe	'	'	SHIFT '
250	Parenthesis, Beginning	(	(	SHIFT (
251	Parenthesis, Ending	)	)	SHIFT)
252	Asterisk	*	*	SHIFT *
253	Plus Sign	+	+	SHIFT +
254	Comma	,	,	,
255	Hyphen	-	-	-
256	Period	.	.	.
257	Virgule	/	/	/
260	Numeral 0	0	0	0
261	Numeral 1	1	1	1
262	Numeral 2	2	2	2
263	Numeral 3	3	3	3
264	Numeral 4	4	4	4
265	Numeral 5	5	5	5
266	Numeral 6	6	6	6
267	Numeral 7	7	7	7
270	Numeral 8	8	8	8
271	Numeral 9	9	9	9
272	Colon	:	:	:
273	Semicolon	;	;	;
274	Less Than	<	<	SHIFT <
275	Equals	=	=	SHIFT =
276	Greater Than	>	>	SHIFT >
277	Interrogation Point	?	?	SHIFT ?
300	At	@	@	SHIFT @
301	Letter A	A	A	A
302	Letter B	B	B	B
303	Letter C	C	C	C
304	Letter D	D	D	D
305	Letter E	E	E	E
306	Letter F	F	F	F
307	Letter G	G	G	G
310	Letter H	H	H	H
311	Letter I	I	I	I
312	Letter J	J	J	J
313	Letter K	K	K	K
314	Letter L	L	L	L
315	Letter M	M	M	M
316	Letter N	N	N	N
317	Letter O	O	O	O

(CONTINUED)

OCTAL CODE	CHARACTER NAME	ASCII CHARACTER	TELETYPE CHARACTER	KEY OR KEY COMBINATIONS
320	Letter P	P	P	P
321	Letter Q	Q	Q	Q
322	Letter R	R	R	R
323	Letter S	S	S	S
324	Letter T	T	T	T
325	Letter U	U	U	U
326	Letter V	V	V	V
327	Letter W	W	W	W
330	Letter X	X	X	X
331	Letter Y	Y	Y	Y
332	Letter Z	Z	Z	Z
333	Bracket, Left	[	[	SHIFT K
334	Reverse Virgule	\	\	SHIFT L
335	Bracket, Right	]	]	SHIFT M
336	Up Arrow	↑	↑	SHIFT
337	Left Arrow	←	←	SHIFT
340 through 374 are not available				
375	Unassigned Control	1	---	ALT MODE
376	Not Available			
377	Rub Out	DEL	---	RUB OUT

APPENDIX C
USASCII CHARACTER CODES

CHARACTER	8-BIT CODE (IN OCTAL)	CHARACTER	8-BIT CODE (IN OCTAL)
A	301	!	241
B	302	''	242
C	303	#	243
D	304	$	244
E	305	%	245
F	306	&	246
G	307	'	247
H	310	(	250
I	311	)	251
J	312	*	252
K	313	+	253
L	314	,	254
M	315	-	255
N	316	.	256
O	317	/	257
P	320	:	272
Q	321	;	273
R	322	<	274
S	323	=	275
T	324	>	276
U	325	?	277
V	326	@	300
W	327	[	333
X	330	\	334
Y	331	]	335
Z	332	↑	336
		←	337
0	260		
1	261	Leader/Trailer	200
2	262	Line-Feed	212
3	263	Carriage-Return	215
4	264	Space	240
5	265	Rub-out	377
6	266	Null	000
7	267	alt-mode	375
8	270	escape	233
9	271		

APPENDIX D
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
68 719 476 736	36	0 000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0 000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0 000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0 000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0 000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0 000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0 000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0 000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0 000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0 000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0 000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0 000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0 000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 949 953 421 312	49	0 000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
1 125 899 906 842 624	50	0 000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 685 248	51	0 000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
4 503 599 627 370 496	52	0 000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
9 007 199 254 740 992	53	0 000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
18 014 398 509 481 984	54	0 000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
36 028 797 018 963 968	55	0 000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5

APPENDIX E
OCTAL-DECIMAL CONVERSION TABLE

0000 (Octal)	0000 (Decimal)
to	to
0777 (Octal)	0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 (Octal)	0512 (Decimal)
to	to
1777 (Octal)	1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

(CONTINUED)

2000 to 2777 (Octal)	1024 to 1535 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

3000 to 3777 (Octal)	1536 to 2047 (Decimal)

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
8600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

(CONTINUED)

4000 to 4777 (Octal)	2048 to 2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 to 5777 (Octal)	2560 to 3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

(CONTINUED)

6000 to 6777 (Octal)	3072 to 3583 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

7000 to 7777 (Octal)	3584 to 4095 (Decimal)

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

APPENDIX F
OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

(CONTINUED)

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

(CONTINUED)

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

COMMENT SHEET

Manual Title: _____

Publication Number: _____ Title Page Revision Letter: _____

From: Name _____

Business Address _____

_____

_____

_____

FABRI-TEK welcomes your comments and evaluation of this publication. Use this postage paid mailer to record errors, to note deficient areas, and to make general comments. Reference specific subjects and page numbers whenever possible.