DS-1 DISPLAY SYSTEM

Introduction

This document describes a proposed modular graphic/alphanumeric
display system containing a 512 x 512 line, 60 line per inch plasma
display/memory panel and a mini- processor. It is intended to
combine the advantages of display memory and local processing power
to operate in three general modes as follows:

1. As an "intelligent terminal" operating on data received from the
   network or a local host to perform text editing, picture
   processing, etc.

2. As a passive terminal in which the mini-processor translates
   existing display lists, command strings and data structure for
   storage tube terminals or other devices into the proper form to
   operate the plasma display. In particular, a software module for
   simulation of the ARDS 100A is provided.

3. As an offline system, where the processor is operated as a stand-
   along mini-computer for debugging, editing, and general display
   work.

The DS-1 consists of a display module, a processor module, and
keyboard (see Figure 1). The display model is a DIGIVUE
display/memory unit, model 512-60, produced by Owens-Illinois, Inc.,
containing the plasma panel and associated drive circuitry. The
processor module was specially designed and built for the DS-1
application by the Raytheon Company.

A modem is enclosed in the processor module, and is described in
later sections. An alternative RS 232 interface is also available
for connection to a TIP or teletype compatible system.

[Figure 1 DS-1 Display System*]

In addition to the display module and the processor, the DS-1 has a
modem for data transmission, and Ascii keyboard, and an I/0 interface
to support numerous external devices. The mechanical design of the
DS-1 emphasizes flexibility, so that both the keyboard and display
module can be oriented independently of the processor module.

Software will be supplied with the DS-1, for such functions as text editing, vector generation, data management and various I/0 routines.

The DS-1 is intended principally to operate as an online terminal; the offline mode is used for programming and data preparing chores which do not require access to the host computer. In describing system operation, therefore, offline operations are simply a subset of online operations in which the only I/0 functions are local. The I/0 operations are accomplished by "direct" input/output instructions, a special feature of the 700 series machines developed at the Raytheon Company. A single instruction (DIN for input, DOT for output) enables the external device addressed by the instruction to accept the data presently in the accumulator (DOT) or to transmit data to the accumulator (DIN) over the data buss.

The plasma display X and Y address registers are seen as output devices by the processor; the other basic output device is the modem. The keyboard and input modem are the basic input devices, in addition to the optional cassette tape system for program loading. These devices, and the interfacing of other peripherals are described in later sections.

Figure 2 is a block diagram of the basic DS-1 configurations, emphasizing the I/0 structure. The principal role of the processor module is in the restructuring of input data to an appropriate output form, either from modem-to-display (computer to operator) or keyboard-to-modem (operator to computer). Such processing consists of handling communication chores,

[Figure 2 US-1 I/0 Configuration*]

string and stack manipulation, character suppression or translation, vector construction from endpoint data or character-encoded line drawing commands, and data stream protocol and management, so that the input/output character stream over the modem or channel remains compatible with the host computer, while the keyboard inputs and display outputs are being effectively and efficiently handled.

The real significance of the DS-1 as a new display terminal stems from its use of the plasma display/memory unit. Because of its inherent memory and selective erase capability, it can be addressed asynchronously, requires no special "refresh" or access to buffer memory, and is indistinguishable from any other output medium, such as tape, etc. Seen from the operator's point of view, it has very desirable human factors, such as high contrast and "crisp" line dimensions, no jitter, flicker, or distortion, and the capability for rear projection of pictorial or tabular data from slides or microfiche, for example.

Both the display module and the processor are described in details in
subsequent sections, as well as the mechanical construction,
communication interfaces, keyboard, and the system software.

Section 2. Display Module

The DS-1 display module is an Owens-Illinois DIGIVUE_ display/memory
unit Model 512-60. It contains the plasma panel, drive electronics,
and display logic. These three assemblies are described in the
following paragraphs.

PANEL

The DIGIVUE_ display panel (or plasma panel) is a matrix device. It
is constructed from two pieces of 1/4" plate glass, upon which very
fine gold electrode lines are deposited. These electrodes are then
completely covered by a dielectric film. These two plates are then
sealed together with a gap of a few thousandths of an inch between
them, and with their electrode patterns orthogonally oriented.

This "sandwich" is then baked out and pumped down to high vacuum,
backfilled with a gas mixture consisting largely of Neon, and sealed
off. The addition of flexible ribbon cable connectors completes the
fabrication of the panel. The device is fat, roughly _" thick, and
virtually transparent.

The panel is operated in such a way that any location (i.e., any x, y
intersection) may be individually turned "on" or "off" as a source of
visible light. Further, such a location has a "memory" and is
sustained in the 'on" or "off" state until a new selection signal
changes it.

An AC voltage is applied between all x and all y electrodes in
parallel, so that an AC signal less than the breakdown potential is
applied across the gas at all times. This is called the sustaining
voltage, VS. When a single x,y point is to be addressed, a voltage
pulse, VP, is applied to the appropriate x and y lines, such that the
total addressing voltage at their intersection is given by

$$V\ ADD = VS + 2VP,$$

which exceeds the breakdown potential of the gas, initiation a
discharge.

Since the electrodes are covered by a dielectric, no discharge
current can flow in the external circuit. Rather, the ions and
electrons produced by the discharge are carried by the applied field
and are deposited as stored charge on the dielectric surfaces, until

the resultant net field is nearly zero, quenching the discharge after
about one microsecond. As the AC sustaining voltage changes
polarity, however, this stored charge constitutes a bias voltage,
adding to the sustaining signal and producing a new discharge, which
results in stored charge causing another discharge on the next half
cycle, etc. Thus, once addressed, a single x,y location continues to
produce a discharge twice in every sustainer period, or about 100
thousand times per second. Naturally, this appears as a continuous
glow to the eye. A sustained location may be turned off by
addressing in such a way that the stored charge is allowed to return
to zero. Thus, a location, or 'cell", may be turned on or off in a
bistable, random access manner.

The plasma display has a number of extremely useful visual
characteristics. Its transparency permits the use of rear-projected
data from microfilm, color slides, maps, forms, etc., to be "mixed"
with the dynamic, computer generated data to be displayed
(characters, plots, graphics, etc.). In addition to the high
brightness of the gas discharge, the contrast ration is extremely
nigh. The panel used in the model 512-60 has 512 x 512 lines at 60
lines per inch. This high resolution and large size allows highly
readable characters and graphics to be mixed, without distortion or
jitter of the data. Dot matrix characters are especially attractive,
since the "dots" of the discharge sites have very clean, crisp
dimensions.

A significant system characteristic of the plasma display matrix is
its random access memory property. Unlike storage tube displays, it
may be selectively erased as well as written, changing a single
point, curve, or character without altering the contents of other
locations on the panel. Large quantities of text, time histories of
measured data, or complex memory. This allows more efficient use of
the processor and simpler data management within computer memory.

The 512-60 panel used in the DS-1 application will have an active
area of 8.5 x 8.5 inches, containing 512 x 512 lines. There will be
an additional 4 line peripheral border around the addressable area to
provide conditioning of the discharge sites.

The back of the panel will be frosted with an optically diffusing
surface suitable for the projection of optical images. A
circularly-polarizing neutral density filter with anti-glare coating
will be provided on the front of the display. The brightness and
contrast will be sufficient for use in a normal lighted laboratory
area.

Drive Electronics

The Drive electronics for the display module will be the standard
Owens-Illinois circuits requri3ed to operate a 512 x 512 line panel
as part of our model 512-60 standard display unit. They consist of a
sustaining generator, which maintains cells, once addressed, in the
appropriate state ("on" or "off") after removal of the addressing
signal, and pulse-forming circuits for the generation and appropriate
mixing of addressing pulses at the selected panel sites.

The sustainer will operate with a basic period of 20 microseconds,
with a peak voltage of approximately 130 volts. Exact voltage
levels, frequency and waveshape are adjusted for optimal performance
subject to power supply constraints and data rate requirements.

The address circuits are multiplexed to reduce the number of active
circuits required and employ a non-linear mixing scheme for line
selection. A pulse of 100 volts is applied with the proper time
phase to achieve writing and erasing. The pulse is disabled in all
but the selected line pair.

Provision is also made for bulk erasure in which the entire panel is
erased in two sustainer periods. A power supply of high density
design will provide the necessary sustaining, addressing, and logical
level voltages. The design will employ high-frequency invertors and
switching regulation for maximum efficiency and reduced size. It
requires about 250 watts, and will be housed inside the processor
module.

Display Logic

The logic included in the display module provides the timing and
control required for the sustaining generator and the address pulses,
and also provides the logic level interface to the processor module.
The interface is TTL compatible. It contains 2 nine-bit address
ports X0-X8 and Y0-Y8, which [not legible*], in absolute binary code
the x,y intersection which is to be addressed. These addresses are
accepted when given a write command or an erase command by the
control lines W or E, respectively. These inputs must all be held
steady until acceptance of the data and completion of the operation
has been indicated by the status line, a logic level output from the
display unit. Ad additional control line, B, is used in conjunction
with the H line to cause a bulk erasure of the entire screen.

The model 512-60 display unit proposed in this document is of the serial address type, and thus can write or erase a single point each 20 u/seconds. This permits vectors to be drawn at more than 800 inches/second. Characters in a 5x7 dot matrix can be written at a rate of about 1400 characters per second.

A parallel address type display unit is presently being developed at Owens-Illinois. This module, when available, will be compatible with the DS-1 system with no modification to the processor required. This display will address 16 lines in parallel, which will allow more than 6,000 characters per second to be displayed, or a complete page to be generated in .330 seconds.

Section 3 Processor Module

The Processor is a minicomputer (Figure 1) that accepts ASCII encoded commands and generates outputs to control the writing of information by the display module. The Processor performs the functions of storing information, data and instructions and regulating the flow of information between the terminal's Keyboard, Modems, Display Module, and Program Loading Device (Figure 2).

In a typical operation, the processor calls up input commands from the modem and the keyboard and generates instructions to the display module driver circuitry to produce the required characters, vectors, or editing programs.

Instructions and data are stored in 16 bit words. Arithmetic functions are performed in 2's complement form. The basic machine cycle time is 1.6 microseconds.

The major functional units of the processor are: (Figures 3 & 4)

o Memory

o Memory Address Register

o Program Counter

o Instruction Register

o Accumulator

o Instruction Processing Synchronizer

o Terminal I/O Circuitry

These units are briefly described next.

Memory

An operational memory of 2048 - 16 bit words plus bootstrap memory of 64-16 bit words are provided. The 2038 word memory is a Random Address Memory (RAM) and is used to store the character generator, the vector generator, and the editing programs needed to provide the specific operational requirements of the terminal. Using the RAM type memory provides the flexibility

[Figure 3 Processor Data Flow*]

[Figure 4 Instruction Processor Synchronizer*]

to alter instruction routines and to tailor the terminal for the application. Since the terminal uses RAM, an operational program loading device (a magnetic tape cassette read/write recorder) is proposed so that the operational program can be loaded when the terminal is turned on.

The RAM size is sufficient to permit storage of 128 characters, a vector generation routine and an instruction set.

The bootstrap memory is provided for storing a terminal start up program that puts the terminal in a state for reading instruction data into its operational memory from external sources. Read Only Memory (ROM) devices are used for the bootstrap memory. Both the RAM and the ROM memories are made up entirely of MOS.

Memory Address Register

The Memory Address Register (MA) is a twelve bit register that can be loaded with the program counter or the address portion of the memory data.

Program Counter

The Program Counter (PC) is a 12 bit counter-register that can be loaded with the address portion of the instruction register.

Instruction Register

The Instruction Register (IR) is a 16 bit register of which 4 bits contain the instruction and 12 bits the data associated with the instruction. This can be loaded with a 16 bit memory word.

Accumulator (AC)

The Accumulator (AC) is a 16 bit shift register that can be loaded
with the contents of the program counter, the data inputs, or the
arithmetic unit.

Instruction Processing Synchronizer (IPS)

The Instruction Processing Synchronizer controls the timing and data
flow in the processor. Clocked by its own oscillator, the IPS calls
up instructions from the memory, which then dictates the sequence to
be followed.

Terminal I/O Circuitry

The processor communicates with the other elements of the terminal
either directly on the I/O bus or via one of three device
controllers. The controllers are required to make the I/O devices
compatible with the processor. Separate controllers are provided for
Keyboard, Display Module, and Serial I/O devices.

The I/O Bus includes:

o 16 Parallel Input Data Lines

o 16 Parallel Output Data Lines

o 4 lines for selecting the I/O device to communicate with the
   processor

o 4 Lines for specifying the function to be performed

o 16 sense lines for monitoring the status of each I/O device

o 2 strobe lines to initiate the transfer of data to and from the
   I/O devices

o 1 system clock line

Each of these lines are available at the I/O Bus Connector. The
lines required by each controller varies with the function performed
by the controller (Figure 4). Three ports are provided with the
Serial I/O Channel Controller, two asynchronous and one synchronous.
The latter is for use with an optional cassette tape recorder that
can be either supplied as an option with the terminal or purchased at
a later date. The two asynchronous ports are for interface with a

teletypewriter modem (not supplied) and a telephone modem that is supplied with the terminal. Direct connection with a data processing machine ca be accomplished at the I/O bus connector.

The processor provides a single level interrupt that causes the processor to transfer control to a designated sub-program while automatically storing the contents of all appropriate registers and the return linkage. The mask and unmask instructions cause the interrupt line to be gated according to the state of an inhibit flip-flop. The last instruction (Interrupt Return) restores the program counter, accumulator, and overflow indicator to the condition existing at the time of interrupting.

Software

The desired terminal operations, that is, the logical capabilities of the display terminal will be implemented in software. This approach provides a degree of flexibility and versatility to the terminal's editing and display writing capability. In the latter area, the generation and display of characters, symbols, and graphics are limited in variety only by the resolution (60 dots per inch) and the size (8-1/2" x 8-1/2") of the display screen. The flexible nature of this approach is enhanced by the expandability of the memory to 4K (32K optional).

The software is divided into two segments: (1) servicing routines for the input modem, the output modem, the keyboard, and the display module and (2), specific function routines for line generation, editing, point generation, and character generation.

Typical software processing starts with the program in an idle loop checking the input sense line. When a bit is ready in the modem, the sense line goes true, the processor skips and the program inputs the bit. The sense line resets. The program counts the bit, saves it, and returns to an idle loop waiting for subsequent bits. As each bit is input, it is stored in proper sequence. When a word is entirely memory, it is moved to another memory location where it can be decoded. The program resets the input, counts for the next word, and transfers to the task routine which analyzes the command and determines which segment of the program is to perform the function.

For this case, assume that the program was instructed to plot a character. The program computes a table from the value of the character code. The table contains a compacted image of the dots representing the character. The image is unpacked from the table into a temporary hold area. Then, starting at the top left corner, the program outputs the coordinates and the one or zero value of the image. The next point is output in the next lower Y position. The

process is repeated until 14 bits have outputted (one column). The
program resets the Y coordinate to the original Y and increments the
X. The next column is output the same as the first.

When nine columns have been output, the program returns to the idle
loop.

Instruction Description

Instructions and data are in the form of 16 bit words. Arithmetic
functions are performed in 2's complement form. The following word
formats are applicable:

```
Instruction                0  1  3  4      15
                           I    OP     Address

Address Data               0  1            15
                           I      Address

Data                       0 1             15
                           S       Data
```

The instruction format provides 1 bit to indicate indirect
addressing, 3 bits for the operations code (OOP) and 12 bits for the
address. The address data format contains 15 bits of address and 1
bit for indicating indirect addressing. The data word format
includes 1 sign bit and 15 bits for data.

Logical control of data flow in the display processor is centered
about the accumulator, with the majority of the operations involving
it, or the sensing of data conditions involving the accumulator. The
19 processor operations represent 6 types of instructions:

o Data Movement (Through the accumulator)

o Logical operations (on accumulator contents)

o Condition Checking

o Input/Output

o Jump Control

o Indexing

Each of these instruction types are briefly discussed next. Refer to
Appendix I for a complete description of the instruction set.

Data Movement Instructions - CLA, LDA, ADD, STC

These four instructions: enable the accumulator to be cleared or to
be loaded with data; effect the addition of a word to the
accumulator's contents; and enable the accumulator contents to be
stored in memory.

Logical Operations - AND, CMP, RAL, SAL

These four instructions: enable a word in memory to be logically
"anded" to the accumulator; permit the 2's complement of the
accumulator's contents to be formed; enable a left rotation of the
accumulator contents, and permit a left shift of the accumulator's
contents to be made.

Condition Checking - SAZ, SNZ, SAC, SNA, SNO

These five instructions allow for various condition checks. SAZ and
SNZ permit the accumulator to be checked for zero or non-zero
contents. SAG and SNA enable any bit position in the accumulator to
be checked for a 1 or 0 setting. SNO provides a check for
accumulator overflow.

Input/Output - SS, DIN, DOT

The SS instruction provides a means of determining whether devices
which interface with the processor (e.g., modem display module) are
ready for data transfers. Actual transfer of data occurs through the
accumulator, one word at a time. DIN causes one 16-bit word to be
transferred from an external device to the accumulator. DOT
indicates the transfer of one 16-bit word from the accumulator to an
external device. The particular device DIN or DOT address is
indicated by the value in the device address field of the
instruction; the instructions enable controller I/O with a variety of
peripheral units.

Jump Control - JMP, JSR

These instructions provide the means for altering the flow of program
logic. JMP is an unconditional transfer of control to a designated
memory location. JSR provides a subroutine capability to the
program. The execution of a JSR instruction involves saving the
return address to the jump at a designated location. By designating
the store for the return address as not being in line with executed
code, the program which is in execution can be held in read only
memory.

Indexing - IAS

These instructions provide an indexing facility to the order code.
The indexing occurs in the memory location specified in the address
field of the IAS instruction; thus, multiple indexing is possible.

The above instruction types provide a fundamental but general
programming capability. A basic symbolic assembly language has been
built about this instruction set and is used to write programs which
do character, assembler program; the assembler is written in FORTRAN
language.

Modem

The display terminal is supplied with a modem for interfacing the I/O
controller with a telephone line. The modem operates asynchronously
through one of the three I/O controller ports (Figure 5). It is an
FSK unit that operates at data rates from 0 to 1800 bps in a four
wire full-duplex mode and from 0 to 1200 bps in a half-duplex mode.
In the four wire full-duplex mode, both the local and remote modem
operate fully independently. A data transfer sequence may be
originated from either model at any time by activating the Request To
Send signal. The data transfer sequence is terminated by dropping
the RTS signal at the originating modem. In the half-duplex mode,
the modem operates with any data format. Selection between transmit
and receive modem identity is determined by the status of the Request
To Send signal from the processor.

The modem may be used over a dial-up network using the half-duplex
mode. Operation in this manner requires the use of the November,
1968, Data Access Arrangement, number F-57951, supplied by the Bell
System. In this operating mode, a data call is originated by the
phone set associated with the Data Access Arrangement. At the remote
location, the call is manually answered and the data key on the phone
se is placed in the data position.

Keyboard

A multimode keyboard with standard ASCII character-to-key assignments
is proposed. The keyboard includes codes for upper and lower case
alphanumerics and special function keys and codes for editing
operations.

[Figure 5 DS-1 I/O Data Lines*]

Section 4 Mechanical Considerations

The DS-1 consists of three separate self-contained sub-units: the
display module, the processor module, and the keyboard.

Display Module - The Display Module will contain the plasma panel and the driver electronics. It's overall size will 14" W x 14" H x 61/2" D, and will weight approximately 25 pounds. An 8-1/2" x 8-1/2" viewing area will be provided.

Interconnections to this unit will be provided at the rear of the assembly.

Processor Module - The processor enclosure will contain circuit board assemblies, three separate power supplies, and a cassette tape recorder (optional). The overall size of this unit will be approximately 20" W x 6-1/2" H x 26" D and will weigh approximately 60 pounds.

Input/Output connectors will be provided at the rear of the enclosure to accommodate the units with which it must interface.

Keyboard - This unit will provide a full alphanumeric keyboard housed in a desk top enclosure approximately 15" W x 3" H x 7" D. The unit will include:

o The entire displayable character set in the standard ASCII layout.

o Special Function keys for editing and I/O control.

Keyboard operation is similar in touch characteristics to an electric typewriter and suitable for use by an inexperienced operator.

Appendix I

DS-1 Instruction Set The following instructions are provided;

The execution item for instructions having indirect addressing (optional with instruction marked*) requires one cycle for each indirect level.

```
LDA, A, I Load Word       0  1  2  3  4       15
                          I  0  0  0       A
```

This instruction loads the contents of address A into the accumulator. The previous contents are lost. The program counter is indexed by 1.

```
                        Time = 1 cycle*
                        0  1  2  3  4       15
JMP, A, I, Jump         I  0  0  1       A
```

This instruction causes program control to be transferred to the
address A. The contents of the accumulator are not altered.

```
                         Time = 1 cycle*
                         0 1 2 3 4      15
JSR, A, I, Jump & Save   I 0 1 1      A
            Return
```

This instruction stores the present address plus 1 at the location
specified by the contents of memory location A and transfers program
control to location A+1. The accumulator is cleared.

```
                         Time = 3 cycles*
                         0 1 2 3 4      15
ADD, A, I, Add           I 0 1 0      A
```

This instruction adds the contents of memory at location A to the
accumulator. The overflow indicator is set to the appropriate state
and remains set until the next ADD, IAS, or CLA instruction is
processed.

```
                         Time = 2 cycles*
                         0 1 2 3 4      15
AND, A, I And            I 1 0 0      A
```

The instruction ands logically the contents of the accumulator with
the data stored in Location A. The results is stored in the
accumulator.

```
                         Time = 2 cycles*
                         0 1 2 3 4      15
STA, A, I Store          I 1 0 1      A
        Accumulator
```

This instruction stores the contents of the accumulator in memory
Location A.

```
                         Time = 2 cycles*
IAS, A, I Index & Skip   0 1 2 3 4      15
         If Zero         I 1 1 0      A
```

This instruction reads the contents of memory location A and adds 1
to it. The results are then stored back in location A of the memory.
If the results of the addition is zero (i.e., an overflow generated)
the next instruction is skipped. Otherwise, the next instruction is
executed. The accumulator contains the contents of A plus 1.

```
                          Time = 3 cycles*
                          0  1  2  3  4  5  6  7  8    11  12  15
     SAL, N Shift         0  1  1  1  0  0  0  0    n
     Accumulator Left
```

This instruction shifts the contents of the accumulator left by n
bits, the least significant bits become zero.

```
                          Time = 1 cycle*
                          0  1  2  3  4  5  6  7  8    11  12  15
     RAL, N Rotate        1  1  1  1  0  0  0  0    n
     Accumulator Left
```

This instruction rotates the contents of the accumulator left by n
bits. The most significant bit becomes the initial least significant
bit.

```
                          Time = cycle*
                          0  1  2  3  4  5  6  7  8     15
     HLT, Halt            0  1  1  1  1  0  0  1
```

This instruction causes program execution to stop. Used primarily
for debugging purposes, depression of the RUN button or SINGLE CYCLE
button causes continued execution either continuous or one cycle
respectively.

```
                          Time = 1 cycle*
                          0  1  2  3  4  5  6  7  8     15
     SAZ Skip Accumulator 0  1  1  1  0  1  0  0
     Zero
```

This instruction causes the next instruction to be skipped if the
accumulator is zero.

```
                          Time = 1 cycle*
                          0  1  2  3  4  5  6  7  8     15
     SNZ Skip Accumulator 1  1  1  1  0  1  0  0
     Non-Zero
```

This instruction causes the next instruction to be skipped if the
accumulator is non-zero.

```
                          Time = 1 cycle*
                          0  1  2  3  4  5  6  7  8  11  12  15
     SAC, B Skip          1  1  1  1  0  1  0  1    B
     Accumulator
```

This instruction causes the next instruction to be skipped if bit
number B is one (1). The most significant bit is zero and the least
significant is 15.

```
                              Time = 1 cycle*
                         0  1  2  3  4  5  6  7  8     15
SNO Skip No              1  1  1  0  0  0  0
Overflow
```

This instruction causes the skipping of the next instruction if the
overflow is not set.

```
                              Time = 1 cycle*
                         0  1  2  3  4  5  6  7  8     15
CMP Complement           1  1  1  0  0  1  1
Accumulator
```

This instruction stores the 2's complement of the accumulator in the
accumulator.

```
                              Time = 1 cycle*
                         0  1  2  3  4  5  6  7  8  11  12  15
DOT Data Output          1  1  1  0  0  0  1   Add    Func.
```

This instruction causes the contents of the accumulator to be gated
to the DIO Data buss and a Data Output Strobe (DOS) pulse to be
transmitted to the peripheral device. This pulse is used by the
device to store the contents of the Standard I/O Buss. The device
Address and Function data is transmitted simultaneous with data.

```
                              Time = 1 cycle*
                         0  1  2  3  4  5  6  7  8  11  12  15
DIN Data Input           1  1  1  1  1  0  0   Add    Func.
```

This instruction causes the device address and function data to be
transmitted as in the DOT instruction. Upon detection of this
address, function and DATA Input Strobe (DIS), the device gates its
data onto the Standard I/O Buss. The deactivation of the Data Input
Strobe indicates that the data has been received and stored.

```
                              Time = 1 cycle*
                      0  1  2  3  4  5  6  7  8  9  10  11  13  14  15
INR Interrupt         0  1  1  1  1  1  1  1  0  0  0       L   0   0
Return
```

This instruction causes the Program Counter, the Accumulator, and the Overflow indicator to be restored to the conditions prior to servicing the present active level. Parameter L must be set to the level being processed for correct return linkage. The interrupt is returned to the idle state. See pages 3-3 and 3-4.

```
                        Time = 4 cycles*
                        0  1  2  3  4  5  6  7  8     15
MSK Mask Interrupts     1  1  1  1  1  1  1  0
```

This instruction causes the interrupt system to be inhibited from causing any interrupt to be processed. Interrupts waiting or received while interrupts are _masked_ will be processed when they are unmasked. See pages 3-3 and 3-4.

```
                        Time = 1 cycle*
                        0  1  2  3  4  5  6  7  8     15
UNM Unmask Interrupts   0  1  1  1  1  1  1  0
```

This instruction causes the interrupt system to be processed in the normal manner. See pages 3-3 and 3-4.

Time = 1 cycle*

* Please see the PDF file for figures and missing text (not legible in the original).