

## "PRELIMINARY" Q64 TEST PANEL

### HARDWARE

The Q64 Test Panel is a modified VT3 logic card. It must be installed in a VT3 monitor to be operational. The modifications required will result in the logic card remaining dedicated to this purpose ( the mods are permanent). The board will be shipped with mods in place. There are no current plans for modifications to be done in the field.

### INSTALLATION

Once the logic card is in place, the Q64TP is cabled directly to the AUX. CPU card through the "top" serial port. The port below is for indirect access and is not currently being used.

### TEST PANEL USE

Upon power-up, the the Q64TP will come up with the a display similar to Fig. 1. The processor is manipulated and interigated by the use of the commands listed in sections 4.0 thru 5.0. Commands are entered in the lower left corner. Note: some commands have not yet been implemented.

ALWAYS begin by entering HALT. The Q64TP can not execute commands if the CPU is already busy. Once halted commands may be entered by exact spelling or abbreviations. Typing HELP will get you a list of the acceptable abbreviations.

DISCLAIMER: This document is intended ONLY as an introduction to the Q64TP. The panel is still under development. The functions described here are subject to change. A final release document will accompany the final product.

## 1.0 INTRODUCTION

This document describes the Q64 Test Panel (Q64TP) for the Q64.

The design goals of the Q64 Test Panel are:

- . Provide a method to debug assembly language programs running on the Q64 processor;
- . Provide most of the features available on the Q29BS Test Panel for the Q64;
- . Provide some diagnostic capabilities to the programmer or field service person when the Q64 does not operate correctly;
- . Provide some "logic analyzer" features, such as tracing, counting, and histograms.

Whenever possible, the commands and features of DELTA, Qantel's Debugging Monitor, will be included. Ideas are also incorporated from the Q29BS Test Panel, the Q64 Micro Monitor, and DFULL32.

The Q64 Test Panel does not attempt to replace DELTA. It is designed to be used when DELTA is not sufficient, such as debugging the Operating System, I/O drivers, standalone programs, or ATPs.

It will only run with the Q64.

Using the Test Panel will impose extra processing on the Q64 microcode. This will be minimized as much as possible; however, it will slow the Q64 and change the timing of the macro execution. No time penalty is imposed when the Test Panel is idle.

## 2.0 COMPONENTS OF THE Q64 TEST PANEL

The Q64 Test Panel consists of a VT3 with specially-programmed ROMs. It will be connected via a standard RS232 cable to the serial port on the Q64 processor.

The Q64 supports the Test Panel with:

- 1) Serial port for transmitting and receiving asynchronous data at 19,200 bps;
- 2) Single Step hardware mode, which allows the processor to stop between macro instructions;
- 3) Q64 firmware which handles commands sent to it via the serial port.

## 2.1 TEST PANEL LIMITATIONS

A discussion of Test Panel limitations will be made here because they result from the lack of hardware in the system. Features, such as stopping the machine whenever an address is accessed, cannot be implemented because there is no hardware "MASP" function. The panel is also not capable of debugging or tracing the Q64 microcode, due to the nature of the hardware.

Advanced DELTA features, such as using the symbolic debugging, are not available because the Test Panel has no way of accessing the symbol table (nor the RAM space to keep it). Dis-assembly of REAL statements is also not feasible due to lack of ROM space. The Test Panel cannot generate hardcopy output because it has no access to a printer. Its only interfaces are the Q64 microcode, the display screen, and the keyboard.

The VT3 was selected over the VT4 for its availability. At a later time, the VT4 can be considered.

### 3.0 GENERAL DESCRIPTION

The Q64 Test Panel consists of a video display screen and a keyboard. The current state of the Q64 processor is indicated on the display screen. The operator communicates with the Q64TP by typing commands on the keyboard. These commands may be processed internally by the Q64TP or may result in messages being sent to the Q64. The results of all commands are displayed on the screen.

### 3.1 DISPLAY SCREEN

The display screen consists of 25 lines containing 80 characters each. The screen will be logically divided into two sections: fixed display area and operator area.

Lines 1 through 17 are the fixed display area which will display the current state of the Q64. This includes the macro code program counter (PC), the current macro instruction, the condition flags, the base registers, and the user breakpoints. All values are displayed in hexadecimal, except the flags which will be symbolic (NZ, MI, OV).

Lines 18 through 25 are used for operator input and responses from the Q64TP. This section of the screen is treated as a typewriter and will roll up with each new line. The Q64TP displays the prompt ":" on line 25, column 1, when it is awaiting operator input. Keys typed by the operator will be entered on line 25. A blinking cursor will indicate the next location for operator entry.

The display screen is formatted as shown on the next page.

FIGURE 1

(1)	BREAKPOINTS	Q64 TEST PANEL	(2)	BASE REGISTERS
	1 AT aaaa			BR0 aaaaaa
	2 IF aaaa rel bb			BR1 bbbbbb
	3 AT aaaa IF bbbb rel cc			BR2 cccccc
				BR3 dddddd
				BR4 eeeeeee
				BR5 fffffff
				BR6 gggggg
				BR7 hhhhhh
(3)	0-7: xxxxxxxxxxxxxxxx			BR8 iiiiii
	8-F: xxxxxxxxxxxxxxxx			BR9 jjjjjj
				BRA kkkkkk
(4)		HALT/RUN		BRB llllll
(5)	FLAGS NZ, MI, OV etc.			BRC mmmmmm
(6)	QPC aaaa: ccccccccccccccc			BRD nnnnnn
	pppppp			BRE oooooo
				BRF rrrrrr

: Operator enters data here

FIELDS DEFINED

- (1) BREAKPOINTS, debugging aids for determining proper program flow. Will cause the processor to halt if the criteria is met. See section 5.0 for determining criteria variables.
- (2) BASE REGISTERS, this is a display of the contents of the 16 registers.
- (3) ACCUMULATOR CONTENTS, scratch pad in memory.
- (4) HALT/RUN, will display either HALT or RUNNING.
- (5) FLAGS, will show what processor flags are set.
- (6) QANTEL PROGRAM COUNTER: aaaa= current "logical" memory address.  
cccccccccccccccc= current macro instruction from the FETCHER  
pppppp= current "physical" memory address

### 3.2 OPERATOR KEYBOARD ENTRY

The user communicates to the Q64TP by typing commands on the keyboard. As each character is typed, it is displayed at the cursor position on line 25. The cursor is incremented to the next location. At the end of line 25, the keys will overstrike, causing the audible alarm. The command is terminated by entering RETURN, TAB, or TRANSMIT.

The control keys available on the VT3 will be implemented. These are listed in the table below. All keys will repeat at approximately 10 cps after being depressed for 1 second.

#### 3.2.1 KEYBOARD CONTROL KEYS

BACKSPACE	Moves cursor right one location.
SHIFT/BACKSPACE	Moves cursor left one location.
CLEAR	Clears line 25 to blanks.
SHIFT/CLEAR	Clears line 25 from cursor to end of line.
INS CHAR	Inserts blank character at cursor position and shifts all characters to the end of the line one position right. The last character is lost.
SHIFT/INS CHAR	Deletes character at cursor position and shifts all characters to the right of the cursor one to the left. A blank is inserted at the end of the field.
LTRS	Allows upper and lower case alphabetic characters to be entered. When pressed a second time, it restores upper case mode.
RETURN	Terminates data entry. Indicates that command is ready to be executed by the Test Panel.
SHIFT/RETURN	Positions cursor to beginning of line 25.
TAB	Same as RETURN.
SHIFT/TAB	Same as SHIFT/RETURN.
TRANSMIT	Same as RETURN.
SHIFT/TRANSMIT	Same as SHIFT/RETURN.
F2	Single step Macros.
F3	Abort routine.

#### 4.0 COMMAND SYNTAX

The general format of a command to the Q64TP consists of a verb followed by zero, one, or more parameters.

The verb indicates what action is to be performed. The parameters specify the details of the action, such as what address should be displayed or modified.

#### 4.1 DATA ITEMS

The Test Panel recognizes the following data items:

- . Hexadecimal numbers - one or more digits 0-9 or A-F
- . Decimal numbers - one or more digits 0-9 followed by '.'
- . Symbolic names - an alphabetic character followed by zero to five alphanumeric characters
- . ASCII strings - zero or more characters enclosed in single or double quotes
- . '.' is the current Qantel macro program counter
- . '\$' is the last address entered by the user (MAR)

#### 4.2 ADDRESS EXPRESSIONS

An address expression is formed by combining one or more numbers, symbols, '\$', or '.' with the arithmetic operators '+', '-', '\*', or '/'. Expressions are evaluated left to right, with no operator precedence.

Normally the Test Panel will assume "logical addressing", i.e., addresses are 15 bits relative to the current base register setting. If the user requires a physical address, the address is prefaced with '#'. To access the privileged memory, preface the address with '%'. (Note: all addresses will be maintained internally as 24-bit absolute addresses.)

#### 4.3 LENGTHS

A length is a hexadecimal or decimal number.

#### 4.4 SYMBOLS

The user can define symbol names to reference certain values. Some symbols are predefined:

PC      Current Qantel macro program counter

BR0, BR1, ... BRF    The 16 base registers

MAR     Memory Address Register - last address entered by user.

#### 5.0 COMMANDS

Any command verb may be abbreviated to the minimum number of characters to make it nonambiguous.

A list of commands is available upon typing 'HELP'. The Test Panel will display '?' if it cannot recognize the command.



## 5.1 DISPLAY COMMANDS

### 5.1.1 DISPLAY MEMORY

DISP addr [,len] [\format]

This command displays the memory contents at a specified address with a given length and format. It sets '\$' to the current memory address. Addresses are left-hand addresses.

The length option sets the number of bytes to be displayed. If specified, it must be a constant (hex or decimal). It defaults to 16 if not specified.

The format option can be either 'H' for hexadecimal or 'A' for ASCII. Initially, it will be hexadecimal. Once a format has been explicitly set, it remains in effect until changed.

### 5.1.2 EXAMINE HEX

EH addr [,len]

Same as display in hexadecimal format.

### 5.1.3 EXAMINE ASCII

EA addr [,len]

Same as display in ASCII format.

### 5.1.4 DUMP MEMORY

DUMP addr [,len]

The Dump command displays one or more formatted lines containing 16 bytes of hexadecimal followed by the equivalent ASCII.

### 5.1.5 SHOW COMMAND

SHOW item

The Show command displays information about the Q64 or the debugging session. The items available to be shown are:

SHOW AT - list breakpoints set by AT command.  
SHOW DO - list commands saved by SAVE command.  
SHOW EQU - list all symbols defined by EQU command.  
SHOW IF - list all data breakpoints set by IF command.  
SHOW MICRO - list the contents of the 16 micro registers and other information pertaining to Q64 microcode.  
SHOW ST - list Status 0 of all 16 I/O controllers.

## 5.2 SET COMMANDS

The Set command group allow the user to modify the Q64's memory and other features.

### 5.2.1 SET MEMORY SM address,expr

This command changes the contents of a memory location to the expression specified. The expression may be a hexadecimal number or an ASCII string. The number of bytes changed is indicated by the expression.

### 5.2.2 MODIFY CURRENT MEMORY / expr

Same as Set Memory, except the address is assumed to be '\$'.

### 5.2.3 FILL MEMORY FM addr,len,expr

This command causes the memory specified to be filled with a single character indicated by expr.

### 5.2.4 MOVE MEMORY (NI) MOV addr1,len,addr2

The Move command moves the data specified at addr1 to addr2 for the number of bytes in len.

### 5.2.5 UPDATE MEMORY (NI) UM addr

The Update command displays 16 hexadecimal bytes of data at the address specified in foreground. It waits for the operator to change this data as desired. Then, it writes the updated data back to the Q64 memory. This is repeated until the operator enters an empty line.

## 5.2. SET COMMANDS - continued

### 5.2.6 EQUATE

EQU symbol [,addr]

The Equate command defines a debug-time symbol to the Test Panel. This symbol can be used in address calculations. If an address is specified, it is used as the symbol's value. If no address is specified, '\$' is used.

### 5.2.7 SET

SET item,expr

The Set command changes the value of certain elements of the Q64 machine.

SET BRn - sets base register n (n=0-F).  
SET FLAGS - sets the condition flags. (NI)  
SET PC - sets the macro program counter  
SET Rn - sets the micro register n (n=0-F) (NI)

### 5.3 EXECUTION CONTROL COMMANDS

These commands allow the user to control and monitor the Q64's execution of macro instructions.

#### 5.3.1 HALT HALT

The Halt command causes the Q64 to halt at the next macro instruction fetch. The display screen will be formatted with the current QPC, FLAGS, base register settings, etc.

#### 5.3.2 MICRO HALT HALT\*

This command should only be used if the normal HALT does not respond. HALT\* stops the Q64 in the middle of macro instruction execution (if possible) and allows the operator to investigate the machine state.

#### 5.3.3 START EXECUTION GO [addr]

The Go command causes the Q64 to start executing macro instructions at PC or the address specified.

#### 5.3.4 STEP STEP [addr]

The Step command causes the Q64 to execute one macro instruction at PC or at the address specified.

#### 5.3.5 INITIAL PROGRAM LOAD IPL

The IPL command causes the Q64 to branch to location 0 of the microcode and act as if IPL was pressed.

## 5.4 BREAKPOINT COMMANDS

The Breakpoint commands impose a tremendous processing overhead on the Q64. Don't be surprised at the slow machine.

### 5.4.1 SET INSTRUCTION BREAKPOINT            AT addr

This instruction sets an instruction breakpoint at the specified address. If the machine is running, it will halt upon executing a macro instruction starting at this address.

### 5.4.2 SET DATA BREAKPOINT            IF addr rel constant

A data breakpoint occurs when the address specified has the required relation to the constant. For example, when location 1000 is equal to 20. The length of compare is always 1 byte.

The relation may be EQ, NE, GT, GE, LT, or LE.

### 5.4.3 SET CONDITIONAL BREAKPOINT            AT addr1 IF addr2 rel constant

The conditional breakpoint sets an instruction breakpoint at addr1. When this occurs, it checks the data breakpoint specified. If the relation is true, the machine is halted; else, it continues execution.

## 5.5 MISCELLANEOUS HOUSEKEEPING COMMANDS

### 5.5.1 HELP HELP

The Help command displays all currently available to the user.

### 5.5.2 SAVE COMMAND SAVE text

Commands may be saved in an internal table via the SAVE command. The Test Panel will display a message of the form: 'SAVED IN nn'.

### 5.5.3 DO SAVED COMMAND DO num

The Do command activates a saved statement.

### 5.5.4 EVALUATE EVAL expr

The Evaluate command causes the expression to be evaluated and the result displayed on the screen.

### 5.5.4 KILL KILL item

The Kill command causes the Test Panel to remove the specified items from its internal tables.

KILL AT [addr] - discard all instruction and conditional breakpoints or the one whose address is specified.

KILL DO [num] - discard all saved commands or the one whose address is specified.

KILL EQU - discard all symbols defined by EQUATE commands.

KILL IF [addr] - discard all data breakpoints or the one whose address is specified.

5.6 FUTURE COMMANDS *Not Implemented*

These commands will be defined and implemented at a later time.

5.6.1 COMPARE MEMORY CMP addr1,len,addr2

5.6.2 FIND STRING FIND addr,len,expr

5.6.3 ACTIVATE HISTOGRAM HIST ON

5.6.4 DEACTIVATE HISTOGRAM HIST OFF

5.6.5 ACTIVATE COUNTING COUNT addr