

VMES User's Manual

VMES Group
State University of New York at Buffalo

November 29, 1989

1 Introduction

VMES (versatile maintenance expert system) is a fault diagnosis system which assists maintenance technicians in troubleshooting electronic circuits. Since a major objective of VMES is its versatility in diagnosing a wide range of faults in different devices, it takes the device-model-based approach. Given the model i.e., structure and function of a device and the symptoms of its malfunction, VMES reasons on the model of the device to come up with the correct diagnosis. This approach is as opposed to diagnosis using shallow rules of association of symptom with cause, traditionally used in medical diagnosis.

VMES uses the design model of the diagnosed device and the observed symptom to locate the faulty objects at the selected maintenance level. This usually requires additional information and VMES is capable of suggesting places for the technician to probe. According to the reported measurements, candidates are reordered and some are eliminated. An enhancement is also made to diagnose a sequential circuit by utilizing fault characteristics for candidate generation.

In fact, the current software maintains two versions of VMES system, which we call VMES I and VMES II. VMES I makes use of SNePS knowledge representation and reasoning system to represent the logical and physical descriptions of the device by semantic nets, and to diagnose the device using those information. VMES I does the diagnosis for combinational circuits. VMES II incorporates the following features :

- Utilization of *fault characteristics* for candidate generation.
- Sequential Circuit Diagnosis
- Explanation Generation
- Working with Multiple Symptoms

The following changes have been affected in the representation scheme:

- Decoupling of logical hierarchy from physical hierarchy. This makes it possible to have several more levels of reasoning in the device.
- No representation of wires. This saves more than 50% of the space. Further the diagnostic power afforded by the representation of wires can be summed up in a single rule.

Currently, VMES I is used to diagnose a half adder circuit, and VMES II is able to diagnose a sequential multiplier and a M3A2 circuit. Therefore, activation of a specific version of VMES depends upon the application.

2 How to run the programs

2.1 Load source files

Host machine name and the top directory name can be different from machine to machine, so before you run the “load” command you have to do a little house-keeping job such as changing or adding logical path-name for host machine. If you load this software to other host machine, look at the file “myload.lisp” in the top directory and try to modify the function “fs:add-logical-pathname-host”. After doing that, all you have to do is to execute one load command.

```
(load "hostname:directory-spec;myload.lisp")
```

2.2 Run the program

Call the top-level function “vmes” as follows :

```
(vmes)
```

This will create 2 graphics windows and 1 lisp listener window as shown in Fig. 1 in the Appendix. Top window is for drawing the circuit under test, and the second window shows the current suspect list which is dynamically changed as the diagnosis continues. Third window is a normal listener window in which you can type the commands, and see the error messages if something goes wrong.

After creating 3 windows, the system provides a menu so that you can choose a circuit you want to diagnose, or go back to listener, or quit the diagnosis.

If you type 'a', 'b', or 'c', then what the system does is to draw the corresponding circuit on the top screen, and start the diagnosis by asking a series of questions. If you type 'b', you will be under VMES I, and if you type 'a' or 'c', you are running VMES II. You can notice two different components by different messages on the screen.

2.3 Running VMES I

At the beginning of a diagnostic session, the user is asked to set a couple of system parameters: `VMES.IML` and `VMES.ConnCheck`. `VMES.IML`, the system parameter for intended maintenance level, can be set to one of the two maintenance levels, "field" or "depot", with "field" being higher than "depot". `VMES.ConnCheck` is the system parameter which allows a user to make the assumption that all wires and POCOns (point of contact) are intact.

After these two parameters are set, VMES requests the initial symptom of the device by asking the user for the values of its primary inputs and outputs. The default number system of VMES is decimal and binary numbers can be entered with a prefix "B". Violations are derived using the functional descriptions of the device and candidates are ordered according to their relationships with the violations. Throughout a diagnostic session, VMES maintains an ordered list of candidates which contains potential faulty components. When the list is not empty, VMES suggests the user to check the first candidate by measuring its inputs and outputs. The reported values are used to reorder and eliminate candidates. This process continues until the list becomes empty or all the remaining candidates are in the same physical component at the intended maintenance level. In the latter case, the user is asked if the diagnosis can be terminated since it may or may not be necessary to distinguish faults among the remaining candidates.

Since diagnostic reasoning is carried out on the logical model of a device, VMES always requests the value of a logical port which may correspond to several otherwise unrelated physical pins of various chips. Through the use of cross-links between logical and physical structures, VMES is able to inform the user which physical ports should be measured for a logical port.

The physical structure of a device is also used for repair suggestions. At the end of a diagnostic session, VMES suggests a repair plan to the user according to the type of the faulty object. If the faulty object is a common components, VMES simply suggests the user to replace its corresponding physical part. If it is a wire, the corresponding physical wires are identified for repair. Note that a logical wire may correspond to several physical wires, e.g., a 4-bit logical wire is realized by four wires on a printed circuit board. If it is a POCOn, the location of the bad contact point is given to the user by referring to the involved physical port of the non-wire component.

2.4 Running VMES II

VMES II has been implemented in Common Lisp without using SNePS. This has been dictated by a necessity to use re-assignable variables to store the values of sequential circuits. A result of this change is that VMES II is much faster in its responses than VMES.

Associated with each port of every device is a list of measured, expected

and test values. All the model information and port values are stored as property lists attached to the name of the device. This results in fast access of information. However, it also necessitates clean-up after each run.

VMES II expects all its values in decimal form. It automatically converts the values to binary form and back. Though this conversion adds considerably to the execution time, it is deemed a worthwhile feature, from the point of view of ease of use.

The reasoning code of VMES II is completely embedded in the recursive function `deep-reason()`. It checks for violation of the output of the device and justification of the unexpected output with the help of measured inputs. If neither of these succeed, the algorithm goes down to the level of the subdevices (if they exist) of the device, or declares the device faulty. Some of the other prominent code segments are for suspect-generation by backward propagation of the violated outputs and forward propagation of input values for simulation purposes.

Usage details are described as follows. The system asks if explanation generator should be activated and whether the user has multiple symptoms to proffer. After this, the system asks for the input and output values from the user. If the system determines that the subdevices of the device should be brought up for questioning, the procedure recurses over each of the subdevices, that occur in the ordered candidate list.

Two aspects need further elaboration. This explanation generator puts out canned messages into a file named `explain.lsp`. The sequence of these messages suggests a logical explanation for the diagnosis performed. The explanation is concatenated onto the screen at the end of the session. Answering the question about explanation essentially switches this feature on or off.

The other question about multiple symptoms works as follows. If the user has many symptoms (complete input-output sets) to provide for the device, he answers 'yes' to this question. Thereupon, the system proceeds to ask for the first set of symptoms. After all the input and output values have been asked for the device, the user is prompted to come up with the second set of symptoms and so on. If the user had initially responded that he did not have multiple symptoms, the system takes up diagnosis after asking for the first set of values.

3 Sample Runs

```
> (vmes)
  Select a circuit you want to diagnose :

a) M3A2   b) Half Adder   c) Sequential Multiplier   d) test circuit
e) goto listener f) goto listener without clearing windows   q) quit
```

Type [a-f], or q (default a): a

Drawing M3A2...

Diagnosing M3A2...

Do you want a summary/explanation of diagnosis?

n

Do you want to use multiple symptoms?

n

What is the MEASURED value of the port OUT1 of the device M3A2-1

19

What is the MEASURED value of the port OUT2 of the device M3A2-1

19

What is the MEASURED value of the port IW1 of the device M3A2-1

3

What is the MEASURED value of the port IW2 of the device M3A2-1

3

What is the MEASURED value of the port IW3 of the device M3A2-1

3

Type any key to continue:

Type any key to continue:

Type any key to continue:

What is the MEASURED value of the port OUT1 of the device MULT-2

10

What is the MEASURED value of the port IW1 of the device MULT-2

3

What is the MEASURED value of the port IW2 of the device MULT-2

3

VMES DIAGNOSES : MULT-2 is faulty.

VMES DIAGNOSES : M3A2-1 is faulty.

Type any key to go back to top-level diagnosis:

Select a circuit you want to diagnose :

a) M3A2 b) Half Adder c) Sequential Multiplier d) test circuit

e) goto listener f) goto listener without clearing windows q) quit

Type [a-f], or q (default a): b

Drawing a half adder....

Diagnosing a half adder....

Please select IML (intended maintenance level):

D(epot
F(ield

* D/F? d
VMES.IML set to DEPOT level

Assume all wires/POCOMs are intact?
* y/n? y
VMES.ConnCheck set to nil (False)

@@@@ diagnose H001

What's the value of port OUT1 of H001
Equivalent Physical Port: pin 3 of H001
* [value]/nil? 1
What's the value of port OUT2 of H001
Equivalent Physical Port: pin 4 of H001
* [value]/nil? 1
What's the value of port IN1 of H001
Equivalent Physical Port: pin 1 of H001
* [value]/nil? 1
What's the value of port IN2 of H001
Equivalent Physical Port: pin 2 of H001
* [value]/nil? 1
 @@ searching vio-expct ...
@@@@ vio-outps found: ports (OUT1) of H001

@@@@ getting suspects for H001 ...
@@@@ suspects created:
 @@ (H001-01 H001-W1 H001-A2 H001-A1)

Type any key to continue:

@@@@ diagnose H001-01

What's the value of port OUT of H001-01
Equivalent Physical Port: pin 3 of H001-U2
* [value]/nil? 1
@@@@ H001-01 shows no problem

@@@@ diagnose H001-W1

What's the value of port OUT of H001-W1
Equivalent Physical Port: pin 2 of H001-U3
* [value]/nil? 1
What's the value of port IN of H001-W1

Equivalent Physical Port: pin 1 of H001-U3
* [value]/nil? 1
 @@ searching vio-expct ...
@@@@@ vio-outps found: ports (OUT) of H001-M1

@@@@@ H001-M1 is faulty
@@ by vio-expct & unit-at-IML
@@ part to be replaced: H001-U3

Terminate the diagnosis?
* y/n? y
VMES manually terminated

>>>> I GOT THE FAULTY PARTS AS >>>>

\$\$ Repair Order: replace H001-U3 (type: 54F04)

Type any key to go back to top-level diagnosis:

Select a circuit you want to diagnose :

- a) M3A2 b) Half Adder c) Sequential Multiplier d) test circuit
- e) goto listener f) goto listener without clearing windows q) quit

Type [a-f], or q (default a): b

Drawing a half adder....

Diagnosing a half adder....

Please select IML (intended maintenance level):

D(epot
F(ield

* D/F? d
VMES.IML set to DEPOT level

Assume all wires/POCOWs are intact?
* y/n? n
VMES.ConnCheck set to t (True)

@@@@@ diagnose H001

```
What's the value of port OUT1 of H001
Equivalent Physical Port: pin 3 of H001
* [value]/nil? 0
What's the value of port OUT2 of H001
Equivalent Physical Port: pin 4 of H001
* [value]/nil? 0
What's the value of port IN1 of H001
Equivalent Physical Port: pin 1 of H001
* [value]/nil? 1
What's the value of port IN2 of H001
Equivalent Physical Port: pin 2 of H001
* [value]/nil? 0
  @@ searching vio-expct ...
@@@@@ vio-outps found: ports (OUT1) of H001

@@@@@ getting suspects for H001 ...
@@@@@ suspects created:
  @@ (H001-W2 H001-W1 H001-W6 H001-W5 H001-W4 H001-O1 H001-N1 H001-A2 H001-W3 H001-A1)
```

Type any key to continue:

```
@@@@@ diagnose wire: H001-W2
```

```
What's the value of port 3 of H001-W2
The WIRE-END connected to: pin 2 of H001-U1
* [value]/nil? 0
What's the value of port 2 of H001-W2
The WIRE-END connected to: pin 2 of H001-U2
* [value]/nil? 0
@@@@@ H001-W2 shows no problem
```

```
@@@@@ diagnose wire: H001-W1
```

```
What's the value of port 2 of H001-W1
The WIRE-END connected to: pin 1 of H001-U2
* [value]/nil? 1
What's the value of port 3 of H001-W1
The WIRE-END connected to: pin 1 of H001-U1
* [value]/nil? 1
@@@@@ H001-W1 shows no problem
```

```
@@@@@ diagnose wire: H001-W6
```

```
What's the value of port 1 of H001-W6
The WIRE-END connected to: pin 6 of H001-U1
* [value]/nil? 0
@@@@@ H001-W6 shows no problem
```


@@@@ diagnose wire: H001-W5

What's the value of port 2 of H001-W5
The WIRE-END connected to: pin 5 of H001-U1
* [value]/nil? 1
@@@@ H001-W5 shows no problem

@@@@ suspects eliminated
@@ eliminated suspects: (H001-W1 H001-W3 H001-A1)
@@ remaining suspects: (H001-W4 H001-01 H001-A2)

Type any key to continue:

@@@@ diagnose wire: H001-W4

What's the value of port 2 of H001-W4
The WIRE-END connected to: pin 4 of H001-U1
* [value]/nil? 0
What's the value of port 1 of H001-W4
The WIRE-END connected to: pin 3 of H001-U2
* [value]/nil? 1
@@@@ suspects reordered:
@@ (H001-A2 H001-01)

Type any key to continue:

@@@@ wire H001-W4 is faulty
@@ by showing different values at wire ends

Terminate the diagnosis?
* y/n? y
VMES manually terminated

>>>> I GOT THE FAULTY PARTS AS >>>>

\$\$ Repair Order: fix the wire connecting
port 4 of H001-U1
port 3 of H001-U2

Type any key to go back to top-level diagnosis:

Select a circuit you want to diagnose :

- a) M3A2 b) Half Adder c) Sequential Multiplier d) test circuit
- e) goto listener f) goto listener without clearing windows q) quit

Type [a-f], or q (default a): c

Drawing a sequential multiplier....

Diagnosing a sequential multiplier....

Do you want a summary/explanation of diagnosis?

n

Do you want to use multiple symptoms?

n

What is the MEASURED value of the port OUT1 of the device SMULT-1

49

What is the MEASURED value of the port IN1 of the device SMULT-1

6

What is the MEASURED value of the port IN2 of the device SMULT-1

8

Type any key to continue:

Type any key to continue:

What is the MEASURED value of the port OUT1 of the device QREG8-1

49

What is the MEASURED value of the port IN1 of the device QREG8-1

49

VMES DIAGNOSES : QREG8-1 is not faulty.

What is the MEASURED value of the port OUT1 of the device SADDER-1

49

What is the MEASURED value of the port IN1 of the device SADDER-1

0

What is the MEASURED value of the port IN2 of the device SADDER-1

48

"Interconnection Fault Detected"

VMES DIAGNOSES : SADDER-1 is faulty.

VMES DIAGNOSES : SMULT-1 is faulty.

Type any key to go back to top-level diagnosis:

Select a circuit you want to diagnose :

a) M3A2 b) Half Adder c) Sequential Multiplier d) test circuit

e) goto listener f) goto listener without clearing windows q) quit

Type [a-f], or q (default a): q

Quitting the diagnosis....

NIL

>

4 Figure illustration

Some TI snapshots are attached in the following pages, and brief explanations for them are given here.

Figure 1.

Initial setup for 3 windows. Topmost window is for drawing the circuit under test, and the current suspect list is displayed on the second screen. Bottom window is a lisp listener which shows a menu for user to choose a circuit.

Figure 2.

A snapshot during the diagnosis of M3A2 after a suspect list is generated by the observed valued of output port 1.

Figure 3.

A snapshot during the diagnosis of M3A2 after mult-2 component is found faulty. Notice that mult-2 is highlighted by a bold line on the top window.

Figure 4.

A snapshot during the diagnosis of a half adder after an initial suspect list is generated.

Figure 5.

A snapshot during the diagnosis of a sequential multiplier after the current suspect list is generated by observed output value.

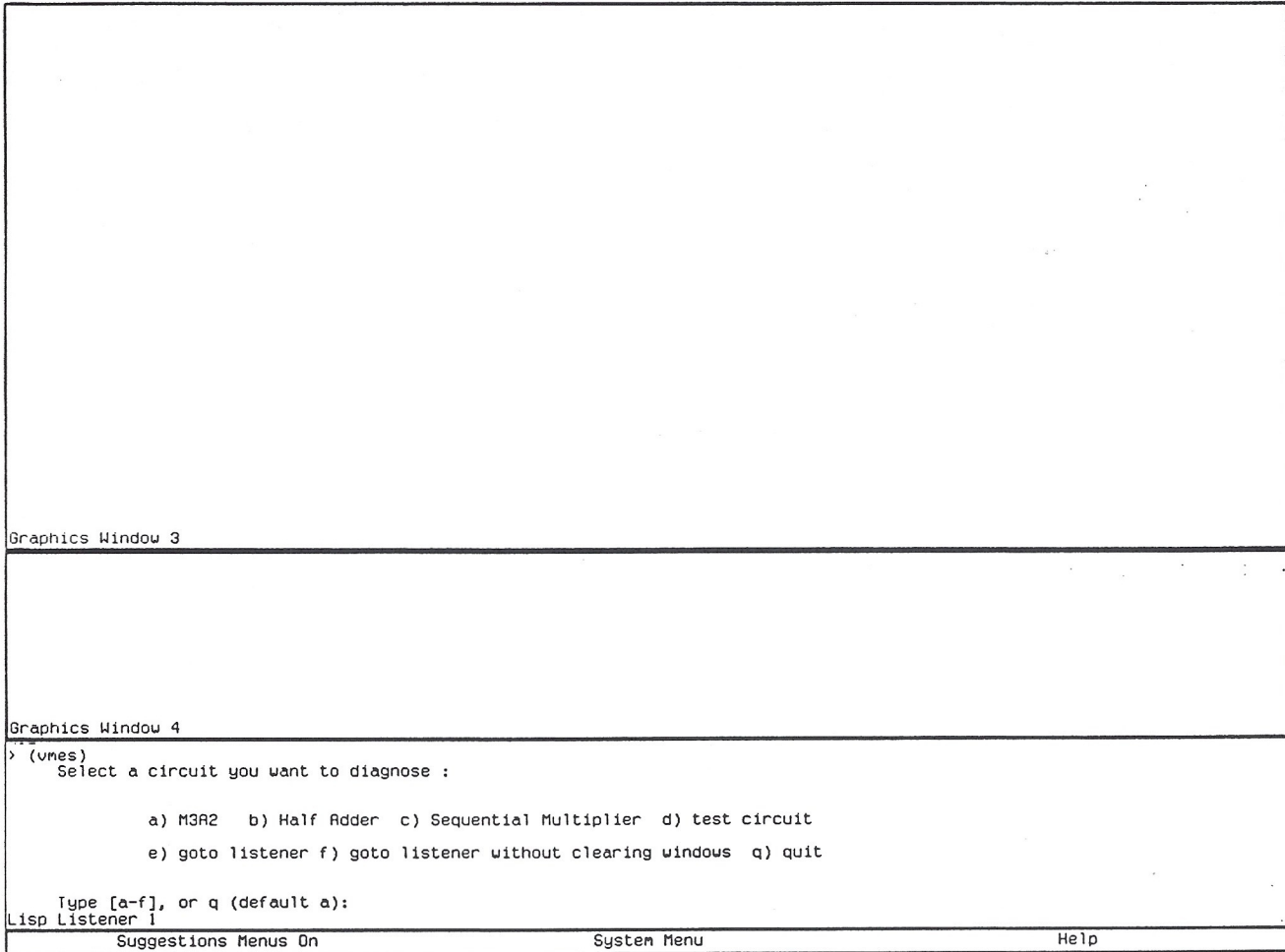


Figure 1.

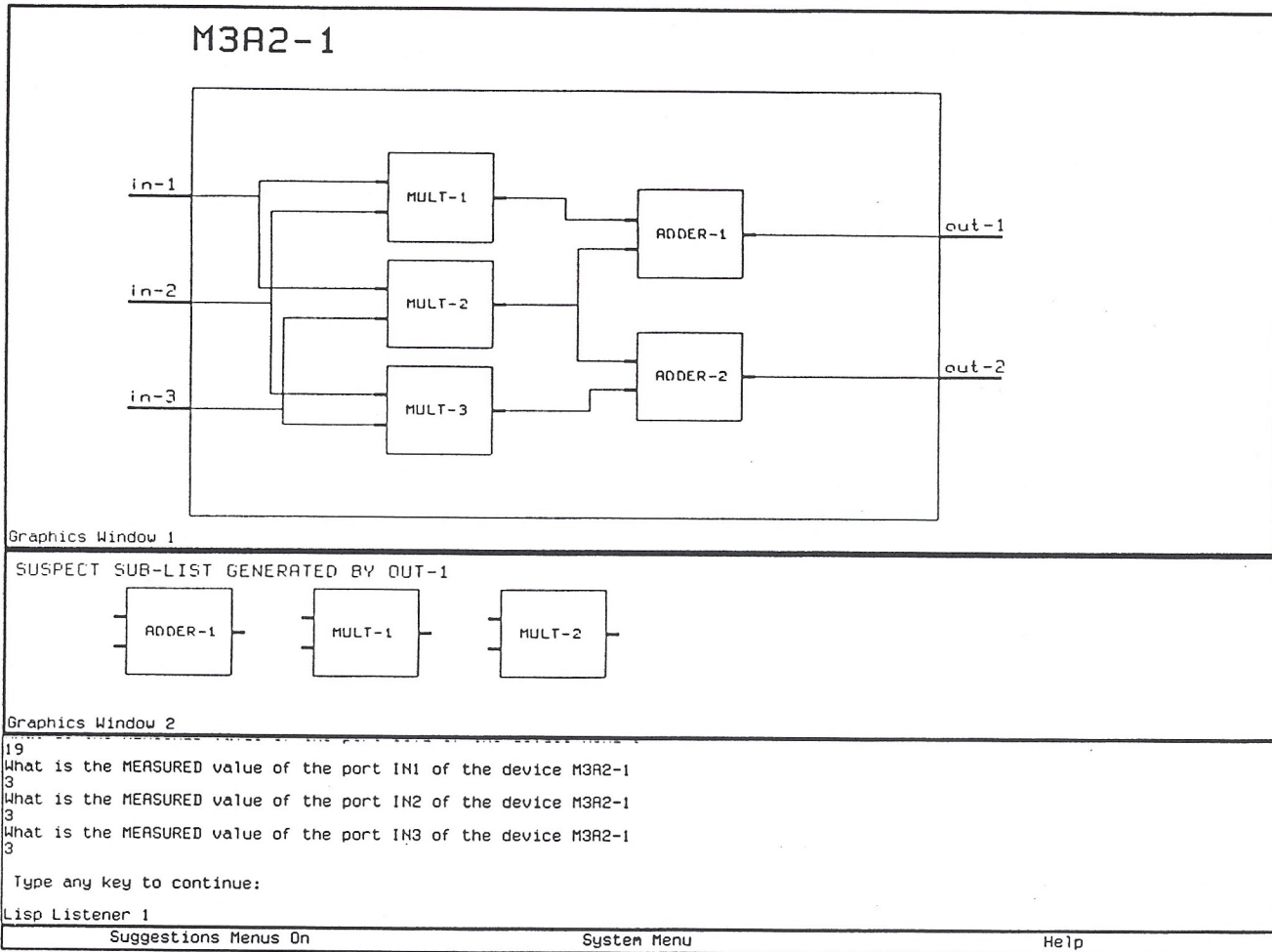
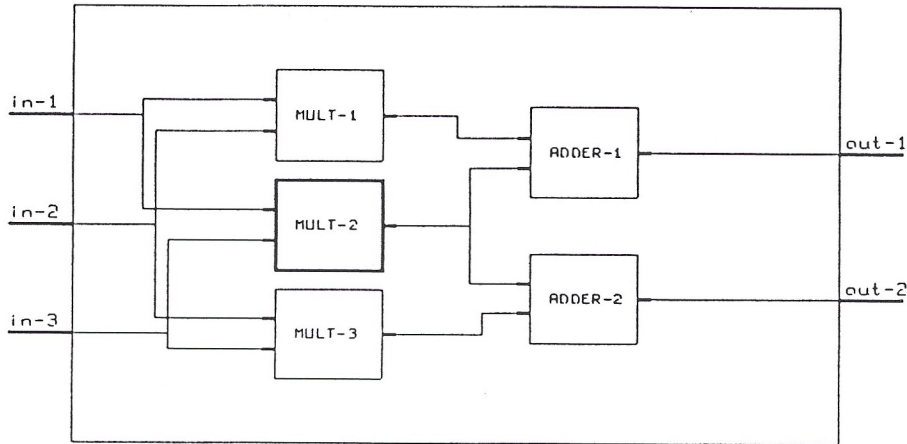


Figure 2.

M3A2-1



Graphics Window 3

CURRENT SUSPECT LIST



Graphics Window 4

What is the MEASURED value of the port IN1 of the device MULTI-2

3

What is the MEASURED value of the port IN2 of the device MULTI-2

3

VMES DIAGNOSES : MULTI-2 is faulty.

VMES DIAGNOSES : M3A2-1 is faulty.

Type any key to go back to top-level diagnosis:

Lisp Listener 1

Suggestions Menu On

System Menu

Help

Figure 3.

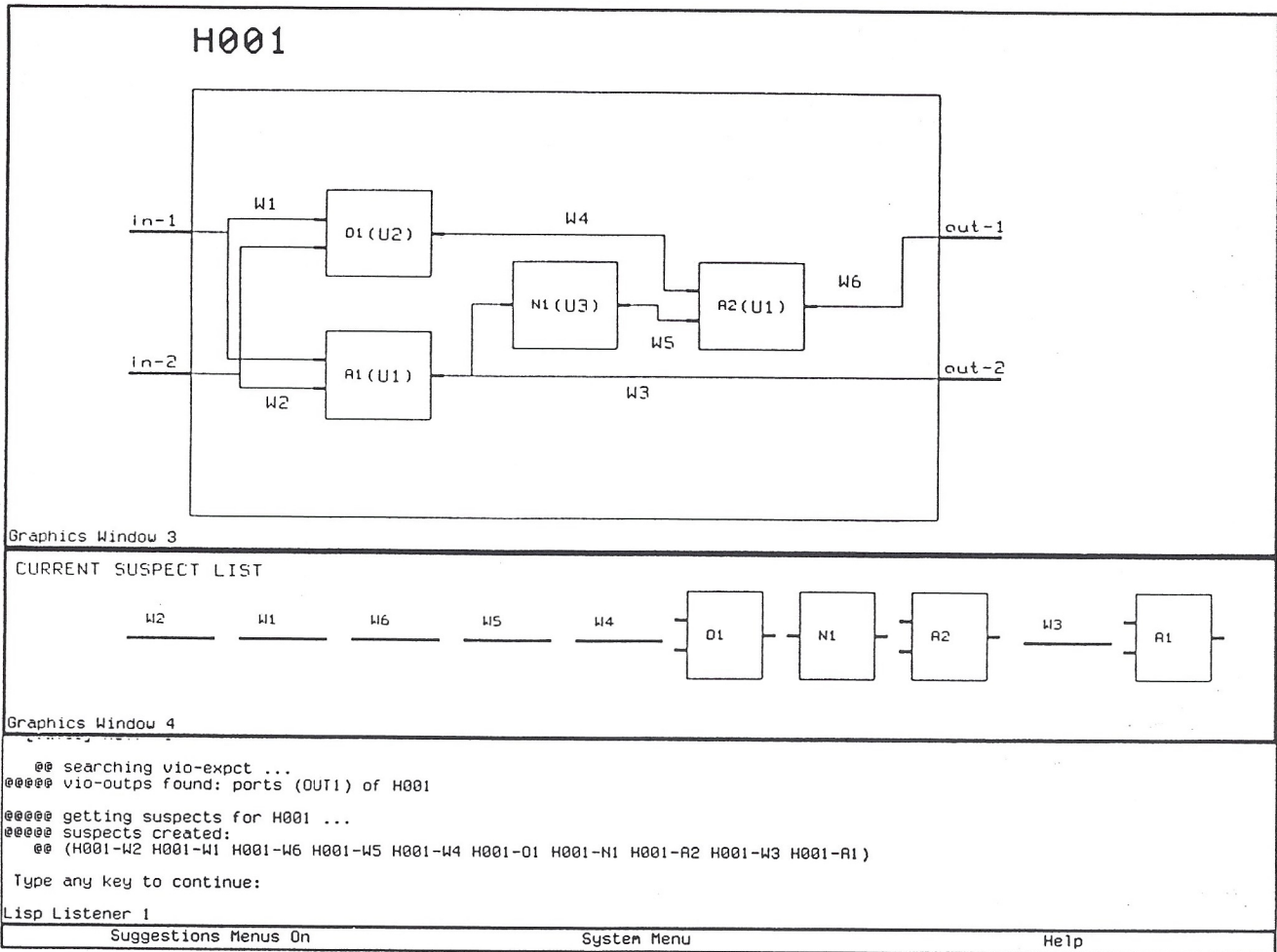
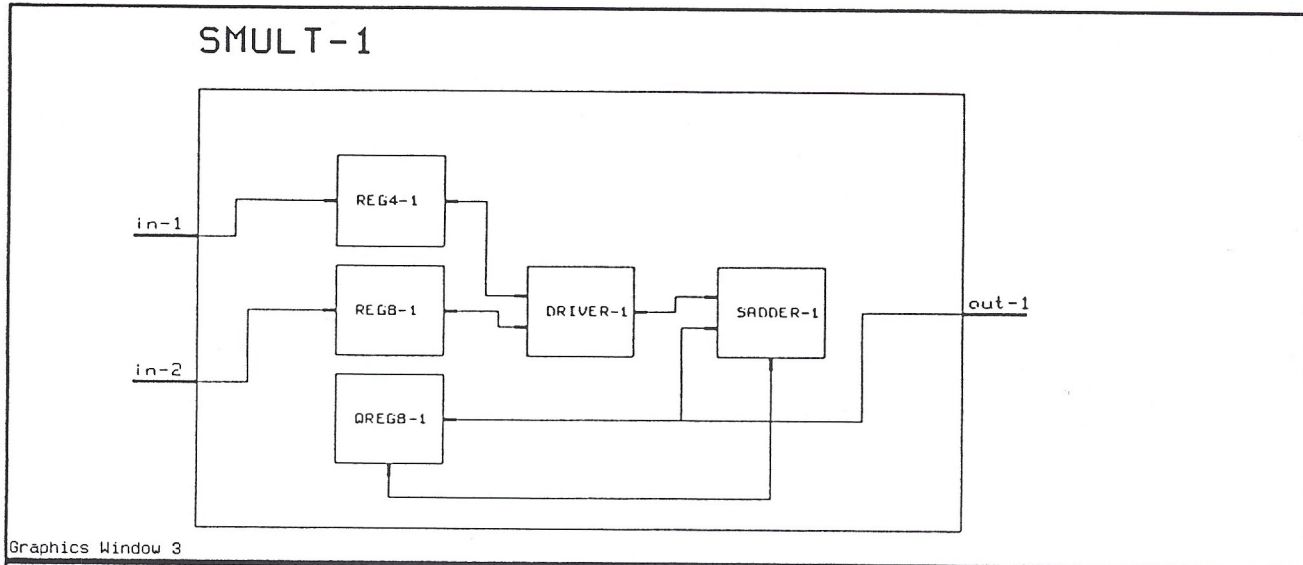


Figure 4.



Graphics Window 3



Graphics Window 4

49
 What is the MEASURED value of the port IN1 of the device SMULT-1
 6
 What is the MEASURED value of the port IN2 of the device SMULT-1
 8

Type any key to continue:

Type any key to continue:

Lisp Listener 1

Suggestions Menu On

System Menu

Help

Figure 5.