Application Note

SCANBRK

How to ensure that a harness is completely tested when the previous harness is prematurely removed from the fixture.

Dynalab
TEST SYSTEMS

SCANBRK

This document explains how to use the PASS 6.0 software to program the Dynalab Analyzer to ensure that a harness is completely tested when the previous harness is prematurely removed from the test fixture.

This document contains the following main sections:

1. a list of assumptions – knowledge required to perform the tasks outlined in this document
2. an explanation of the problem
3. an explanation of the solution approach
4. an overview of the Dynalab solution to the problem
5. an example Sequence
6. detailed descriptions of the Sequence items required for the solution
7. advanced concepts

Assumptions

To successfully use this document, the following knowledge is required:

- basic knowledge of how to enter harness data using PASS 6.0
- knowledge of how to use the Sequence table to create a Sequence

For assistance on how to use features of PASS 6.0, see the PASS 6.0 Help file.

Requirements

The SCANBRK Sequence item requires that the Analyzer be equipped with EPROM Version 7.032 or later.

Warning: Attempts to execute a Sequence item on an Analyzer equipped with an older EPROM version that does not support the Sequence item will result in non-execution of the item – it is ignored in Sequence execution. This may have undesirable consequences, including the passing of defective harnesses.

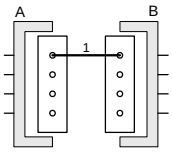
Problem

If an operator removes a harness before all the tests have been completed, and the program is not manually restarted, the next harness will only be partially tested. (The test of the next harness starts where the test of the prematurely removed harness left off.) This can be especially confusing, since the Analyzer will display the "Assembly OK" message at the end of the test, and the operator will assume that the second harness has passed all tests when in fact, the second harness has only been partially tested.

This problem is illustrated below, using a simple 4-wire harness as the example.

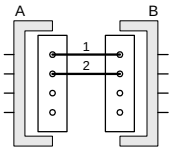
Operator presses START button to start the test. The Analyzer displays the first connection to be made.

```
OPEN
A-1
B-1
```



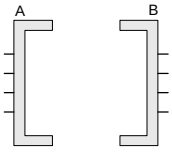
Operator installs wire 1. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-2
B-2
```



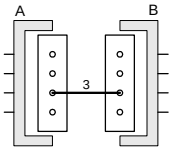
Operator installs wire 2. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-3
B-3
```



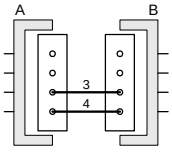
Operator removes partial harness assembly. The Analyzer is still looking for wire 3 to be connected.

```
OPEN
A-3
B-3
```



Operator starts to connect a new harness, starting with wire 3. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-4
B-4
```



Operator installs wire 4. The Analyzer responds by indicating the harness has passed all tests, even though the harness is only partially assembled.

```
ASSEMBLY OK
```

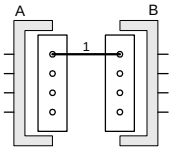
Solution

A solution to this problem is to have the Analyzer detect when a harness is removed before all the tests are completed, and to have the Analyzer force the tests to start over when this condition is detected. This ensures that the testing of the next harness is complete.

This solution is illustrated below, using a simple 4-wire harness as the example.

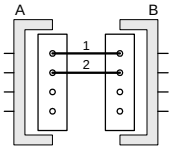
Operator presses START button to start the test. The Analyzer displays the first connection to be made.

```
OPEN
A-1
B-1
```



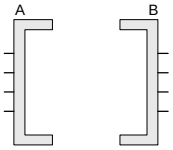
Operator installs wire 1. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-2
B-2
```



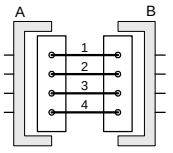
Operator installs wire 2. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-3
B-3
```



Operator removes partial harness assembly. The Analyzer detects that the harness has been prematurely removed, and the test starts over. The Analyzer displays the first connection to be made.

```
OPEN
A-1
B-1
```



Operator must connect all wires for the Analyzer to indicate that the harness has passed all tests.

```
ASSEMBLY OK
```

Solution Overview

What is a break condition ?

A break condition occurs when the harness is removed from the test fixture before all tests are completed.

How is a break condition detected ?

To detect a break, the Analyzer must be capable of scanning for two conditions:

1) **The harness is connected (or at least partially connected) to the test fixture**.

   The Analyzer considers a harness to be at least partially connected to the test fixture if a least 2 wires have been simultaneously connected.

2) **The harness is completely removed from the test fixture before the tests are completed**.

   The Analyzer considers a harness to be completely removed from the test fixture if all connections defined for the harness are OPEN.

So, the Analyzer decides that a break condition has occurred if the harness is completely removed from the test fixture BEFORE the tests are complete and AFTER at least two wires have had continuity.

What is needed for a scan break function ?

Three things are needed for a scan break function:

1) Enable scanning for a break condition

SCANBRK - this sequence item enables scanning for a break condition

2) Define where testing resumes if a break condition occurs

BBRK - this sequence item defines where testing resumes if a break condition occurs

3) Disable scanning for a break condition

SCANBRKOFF - this sequence item disables scanning for a break condition

Items 1 and 3 are required because there are times in a sequence when we do not want the Analyzer to scan for a break condition – such as after testing is completed. Therefore, we need to have the capability to turn the scanning for a break condition on and off.

Item 2 is needed so that we can tell the Analyzer what to do if it finds a break condition.

Example Sequence

This example shows a very simple sequence that uses SCANBRK. This sequence enables scanning for a break condition before testing begins. It sets the sequence location to start over in the event that a break condition occurs. It then tests the harness. Scanning for a break condition is disabled after the harness tests are finished. If the harness passed all the tests, the Analyzer will display the message "Assembly OK". If errors were logged during the tests, the Analyzer will display the errors. The Analyzer then waits for the operator to push the START button. When the operator pushes the START button, the sequence repeats.

Example Sequence using SCANBRK

Line	Sequence Item	Parameter	Description
1	**SCANBRK**	MAIN	Enables scanning for a break condition
2	**BBRK**	3	Specifies the sequence line number to branch to if harness is removed before testing is complete
3	TEST	MAIN	Performs a complete Netlist scan of the harness.
4	**SCANBRKOFF**		Disables scanning for a break condition
5	REPORT		Displays "Assembly OK" if harness passes all tests, or displays error information
6	KWAIT		Waits for the operator to push the Start button
7	REPEAT		Goes to line 1 to repeat Sequence execution

Line 1 The **SCANBRK** item specifies MAIN as its parameter. This means that the Analyzer will do a background scan of the MAIN Netlist. This has the effect of scanning for removal of the entire harness. If the harness is removed after at least 2 wires have been connected, the break flag will be set, and execution resumes at the next **BBRK** sequence item after **SCANBRK**.

Line 2 The **BBRK** item specifies 3 as its parameter. This means that if the break flag is set, execution will branch to line 3.

Line 3 TEST performs a series of scans on the Netlist specified by the Parameter, in this case, MAIN.

Line 4 The **SCANBRKOFF** item disables scanning for a break condition. This allows the harness to be removed after the TEST sequence item has completed.

Line 5 REPORT displays a summary report.

Line 6 KWAIT waits for the operator to press the START button.

Line 7 REPEAT instructs the Analyzer to go to Line 1 and repeat execution of the sequence.

Sequence Items

As mentioned above, SCANBRK, SCANBRKOFF and BBRK are the sequence items used to implement a scan break capability. This section presents a detailed description of each of these sequence items.

SCANBRK

The presence of SCANBRK in a sequence enables the option to scan for a break condition. When enabled, the Analyzer will perform a background continuity scan on the specified Netlist. This background continuity scan takes place when stopped on an error, between sequence items, and while waiting for the START button to be pressed during execution of the KMESSAGE, KWAIT and KFLASH sequence items. SCANBRK requires the name of a Netlist as its parameter. If all connections in the specified Netlist are OPEN, and if a previous scan of the specified Netlist resulted in at least two connections having had continuity, then the break flag is set, and execution resumes at the line number specified by the first BBRK sequence item which appears after SCANBRK.

SCANBRKOFF

The presence of SCANBRKOFF in a sequence disables the option to scan for a break condition

BBRK

BBRK causes branching on a break condition. BBRK requires a sequence line number as its parameter. If the break flag is set, execution branches to the specified line number.

Advanced Concepts

Effect of momentary continuity

When testing a harness, it is possible for the operator to start the test before attaching the harness to the fixture, connect one wire at a time, and then disconnect each wire before connecting the next wire. This results in momentary continuity – each wire establishes continuity between two scan points for a short time. Although the Analyzer will sense continuity as each wire is momentarily connected, this method of testing is not good practice because it prevents SCANBRK from detecting a break condition. Therefore, when SCANBRKOFF is reached in the sequence, the Analyzer will cause execution to branch to the line number specified by the first BBRK sequence item which appears after SCANBRK, as though a scan break had been detected. If BBRK specifies the line number associated with the start of the test (usual condition), this has the effect of forcing a retest, requiring the operator to properly connect the harness to the fixture.

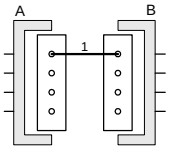
This is illustrated below, using a simple 4-wire harness as the example.

This example uses the sequence shown below. SCANBRK is set at the beginning of the sequence. BBRK is set to branch to line 3, forcing a re-test in the case of a break condition.

Line	Sequence Item	Parameter	Description
1	SCANBRK	MAIN	Enables scanning for a break condition
2	BBRK	3	Specifies the sequence line number to branch to if harness is removed before testing is complete
3	TEST	MAIN	Performs a complete Netlist scan of the harness.
4	SCANBRKOFF		Disables scanning for a break condition
5	REPORT		Displays “Assembly OK” if harness passes all tests, or displays error information
6	KWAIT		Waits for the operator to push the Start button
7	REPEAT		Goes to line 1 to repeat Sequence execution

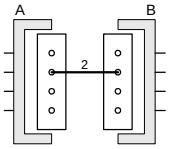
Operator presses START button to start the test. The Analyzer displays the first connection to be made.

```
OPEN
A-1
B-1
```



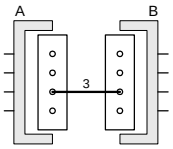
Operator installs wire 1. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-2
B-2
```



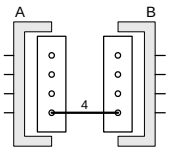
Operator removes wire 1 and installs wire 2. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-3
B-3
```



Operator removes wire 2 and installs wire 3. The Analyzer responds with an audible beep, and displays the next connection to be made.

```
OPEN
A-4
B-4
```



Operator removes wire 3 and installs wire 4. The Analyzer determines that two wires have not been simultaneously connected, and responds by branching to line 3, causing the test to start over. The Analyzer then displays the first connection to be made.

```
OPEN
A-1
B-1
```

Effect of multiple BBRK sequence items

SCANBRK requires a corresponding BBRK sequence item to appear in the sequence. Only a single BBRK item is required. If, however, a sequence is incorrectly programmed with more than one BBRK, it is important to understand that the first BBRK to appear after SCANBRK is the right one. Any other BBRK's are ignored.

For example, in the sequence shown below, the BBRK on line 2 is the sequence item that is operational, because it is the first BBRK to appear after the SCANBRK sequence item. The BBRK that appears on line 4 will be ignored.

Line	Sequence Item	Parameter	Description
1	SCANBRK	MAIN	Enables scanning for a break condition
2	**BBRK**	3	Branches to line 3 if harness is removed before testing is complete
3	TEST	MAIN	Performs a complete Netlist scan of the harness.
4	**BBRK**	1	Branches to line 1 if harness is removed before testing is complete
5	SCANBRKOFF		Disables scanning for a break condition
6	REPORT		Displays "Assembly OK" if harness passes all tests, or displays error information
7	KWAIT		Waits for the operator to push the Start button
8	REPEAT		Goes to line 1 to repeat Sequence execution