

**Q: What is Bit-map and how do I use it?**

## ETS2k BIT-MAP

The bit-map feature of ETS2k locates failures on a memory device by generating either a logical or physical map of the device. Using the bit-map, two different maps can be created: a general map or a detailed map. A general map contains information about addresses, and a detailed map shows actual data failures. The general map displays one red spot for every address (every word). Detailed maps display a red spot for every bit, so the bitmap will be 8 times wider. Before performing any bitmapping, it is assumed that a valid setfile and vector file have been loaded, and that the memory test parameters have already been set in the Memory Test window.

For use of bitmap and permutations discussed in this document, the following Special Script must be enabled in the SpScript.txt file found in the ETSusb folder (remove the leading semicolon to enable the special script):

```
MemTestNewRowColWay
```

Click here on the FA Support sidebar to create bitmaps.



### **Physical Representation**

In order to display the physical representation of a device (i.e. a memory chip) as a bitmap, two different tools are available:

1. Scramble address bits. Define a permutation of address bits, according your specification of physical locations on the chip. Note that after this operation, all 8 bits (from a single location) will remain grouped together for display purposes.
2. Split data bits. Create up to 8 group of bits, for generating a real bit map of the device. For every row (page), the bits will be displayed according to these groups. For example:

for 8 data bits, define 4 groups: 0 1, 2 3, 7 6, 5 4.

In this case, the representation (for a detailed map) will display like this:

d0d1d0d1í d0d1d2d3d2d3í d2d3d7d6d7d6í .d7d6d5d4í d5d4  
 (i.e., data zero, data one, data zero, data oneí )  
 Every group is repeating ÷columnøtimes. Number of columns = page size.

### Setting Scramble and Group Information

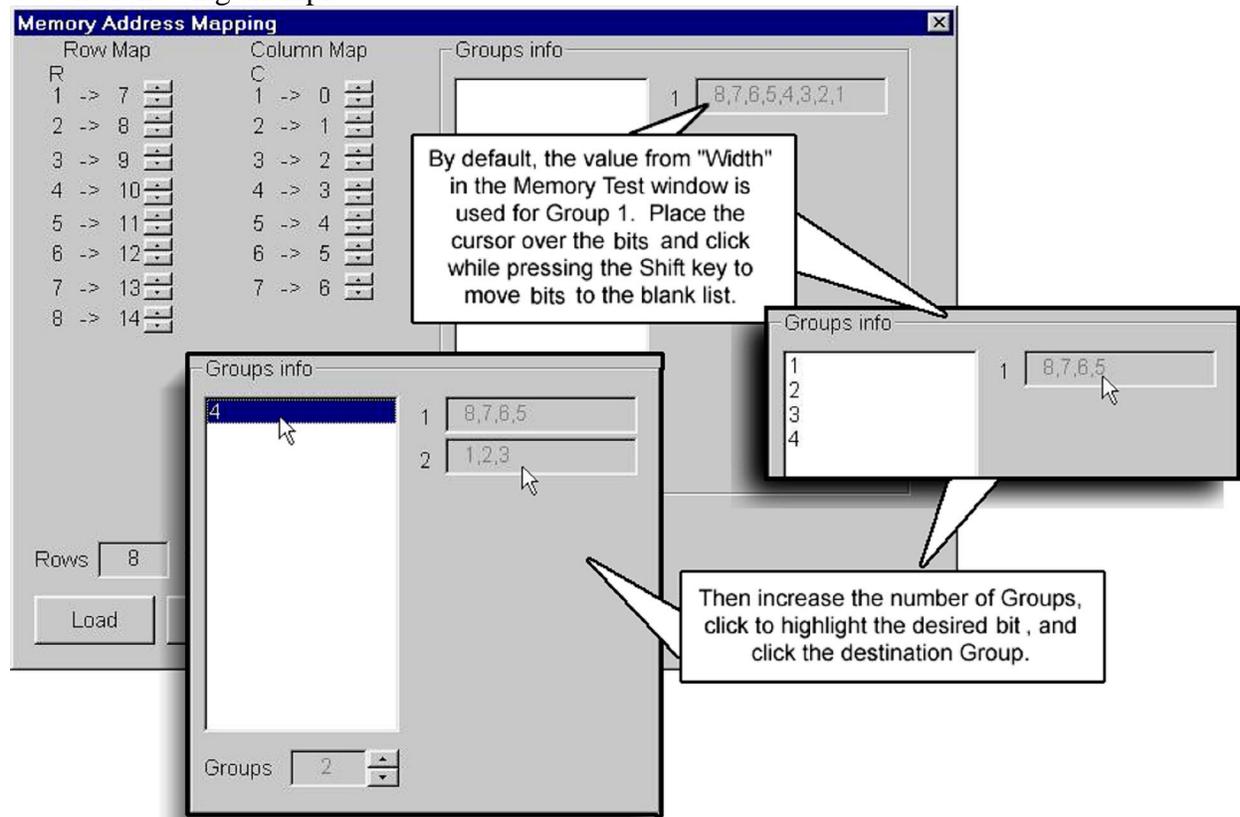
To set scramble and group information, go to the Memory Address Mapping window (from FA sidebar choose Memory FA). The window will use the Memory Test ÷Widthö value as the number of pins.

The left side of the window is used to define the permutation of address bits. The row and column information is imported from the Memory Test window.

The right side is used to define the bit groups. :

- add a new bit to a group by selecting the bit in the list and clicking on the group.
- delete a bit from a group by pressing SHIFT and clicking on the group. The last bit in the group is moved to the list.

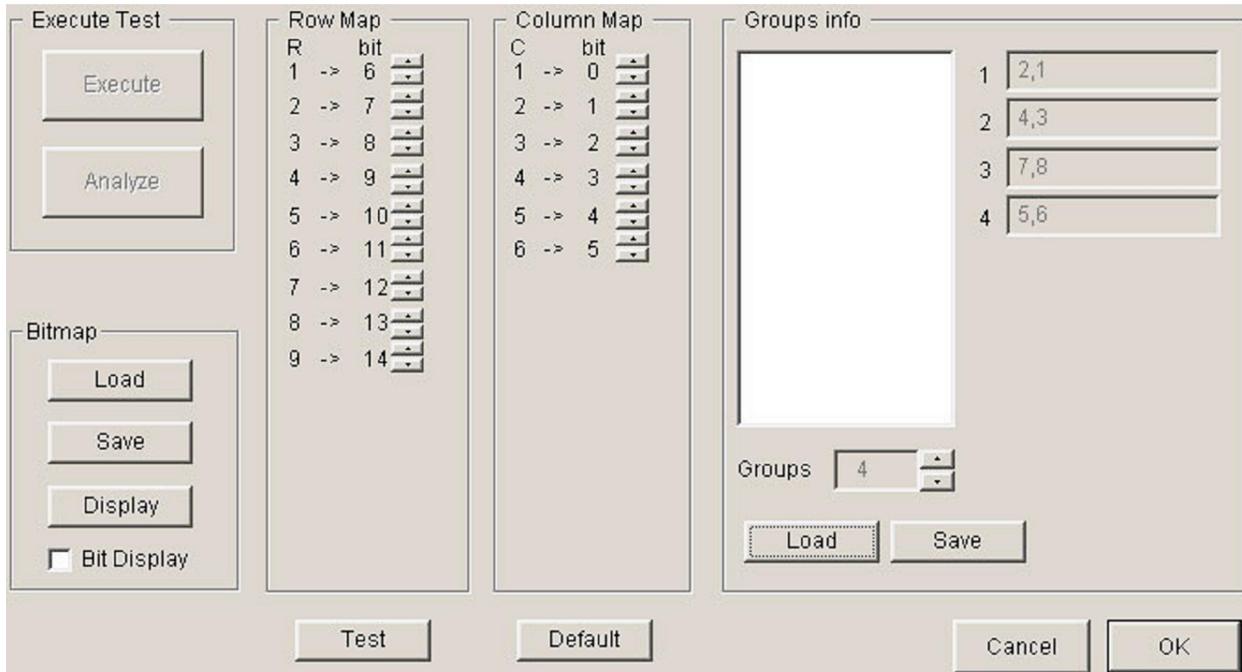
See the following example.



### **Working with the Memory FA Window**

Use the TEST button to check the permutation of address pins. All the information about scramble and groups can be saved in .grpö files by clicking the Save button. This is an ordinary ASCII text file, but the information seen in this file is of value only to the ETS2k software and is used only for saving and restoring the permutation definitions. (Do not edit this file.)

By loading the file ötest.grpö, our group permutations now look like this:



When OK is clicked in the scramble window, all information is loaded and used until the program is finished, or until another ögrpö file is loaded. This also closes the window and returns to the Memory Test window for execution and bitmapping.

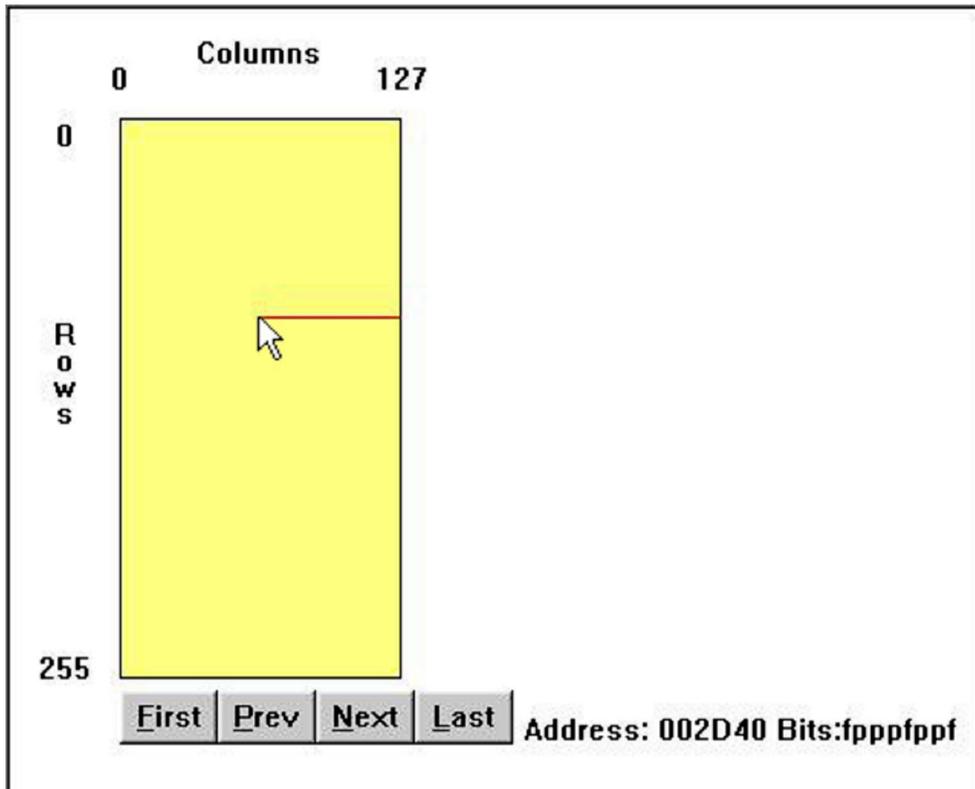
### **Creating the Bit-map**

Test the device by clicking the öExecuteö button in the Memory Test window. Press the öAnalyzeö button in order to generate data for the bit-map. This will create and open an ASCII file called öfailures.logö, which could be large and may take some time to generate. It is important to remember that each bitmap is created from the *last* öAnalyzeö event. Hereö an example of a failures.log file:

Failures for C:\ets\Memchip.set		
Address	Expecting	Actual
002D40	00000000	10001001
002D41	11111111	10101001
002D42	00000000	10101001
002D43	11111111	10101001
002D44	00000000	10101001
002D45	11111111	10101001
002D46	00000000	10101001
	.	
	.	

### Partial failures.log file

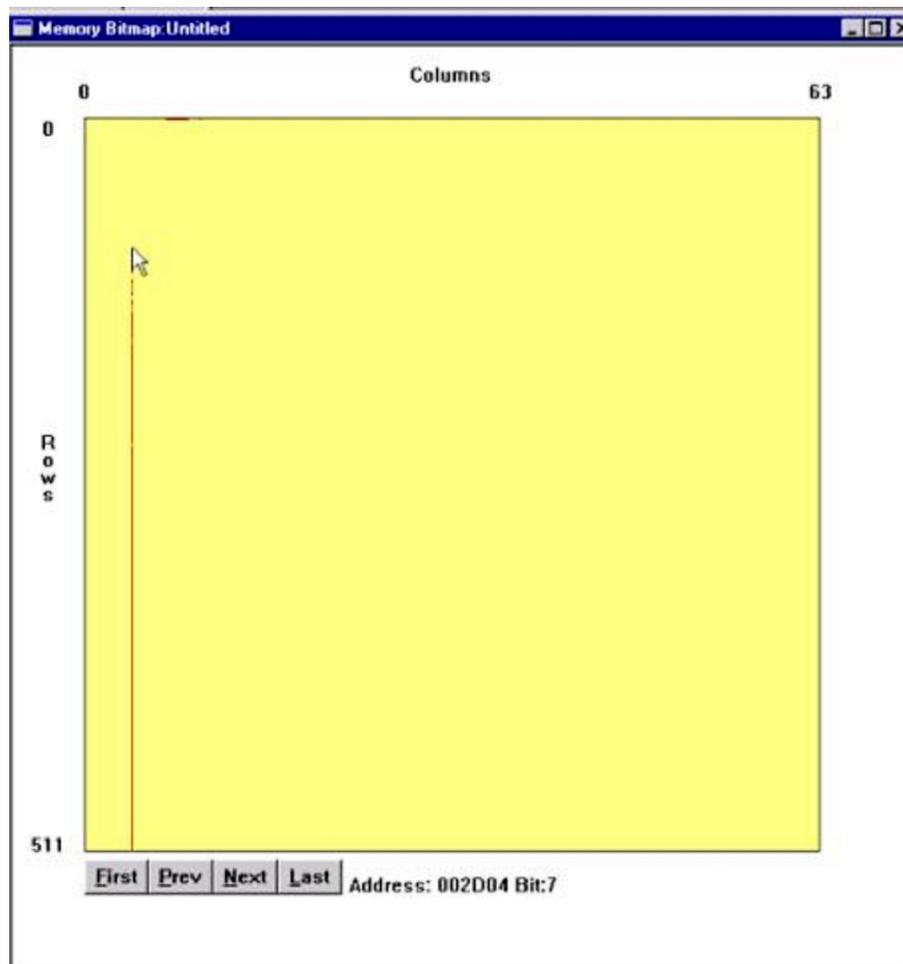
Failures.log is always saved in the ETSusb root directory, and is overwritten by each "Analyze" event. Clicking the BitMap button will display a graphic image using the data in this file (it does not matter if the file is open or closed). An example of a bitmap image using the example failures.log file (without the use of the "Bit info" option) appears below.



Bitmap of "Failures.log"

Navigate through errors by pressing **F** (first), **P** (previous), **N** (next) and **L** (last) keys, or by clicking on the **F**irst, **P**rev, **N**ext and **L**ast buttons. The mouse cursor will automatically move to the failure, and the window will also display specific Address and Bit information. In our example, the cursor is pointing to address 002D40 (hex), and the eight bits at this address are represented as **õppppppfõ** (where **f** = failed and **p** = passed). Click inside the yellow square and the cursor will move to the next failure according the current mouse position.

Use of the Bit info feature will generate a highly detailed bitmap as seen on the last page. It is not necessary to Execute and Analyze again if it has been done once prior to checking the Bit info box.



### Bitmap using the “Bit info” feature

Also See:

**Q'nApp #E4:** Memory Test

**Q'nApp #E36:** Memory Test Functional Description