Application Note
0380-0262-10 Rev 1.0

Embedded Performance, Inc.
606 Valley Way, Milpitas, CA 95035
Telephone: (408) 957-0350
FAX: (408) 957-0307
e-mail: support@epitools.com
web: www.epitools.com

MAJIC^{PLUS} Support For ARM/ETM

0380-0262-10 Rev 1.0

December 30, 2003

Introduction

This document provides application specific information on MAJIC^{PLUS} support for the ARM Embedded Trace Macrocell (ETM). While Chapter 6 of the *MAJIC User's Manual* covers the fundamental functionality of a MAJIC^{PLUS} probe, this application note is an addendum that provides specific information on:

- How to program the Embedded Trace Macrocell to emit trace data,
- How to enable and control MAJIC^{PLUS} trace acquisition,
- How various MAJIC^{PLUS} probe hardware configurations acquire ETM trace data.

Additional Documentation

Additional documentation for the EDT software package and MAJIC probes is installed as part of the EDT software package. On a Windows PC, use the **`[Start]/Programs...`** menu to browse into the **`EPI Tools/EDT`** folder and open **`EDT Documentation Index`**. On a Linux PC view **`./manuals/edtX_doc_index.html`** in your EDT installation.

MAJIC User's Manual	Complete information on configuration and operation of the MAJIC probe, and the MON command language.
EDB User's Manual	Details on using the Trace Display window to view traced instructions and/or data using EDB.
MAJIC Windows CE .NET eXDI User's Manual	Details on using the Trace Display window to view traced instructions and/or data using the Platform Builder Plug-In.
MAJIC Interface Specifications	Application note providing details on the 38-pin Mictor connector used for ETM trace data, and electrical specifications of different MAJIC^{PLUS} probe configurations.

Getting Support

Please do not hesitate to contact our technical support group if you have any questions or need assistance in configuring ETM or the MAJIC^{PLUS} probe. We recognize that these issues are complex, and are committed to making sure our tools work well for you.

The Embedded Trace Macrocell

During program execution, the Embedded Trace Macrocell emits a record or instructions that have been executed, and optionally data transfer addresses and even data values. This information is captured in a dedicated trace memory in the MAJIC^{PLUS} probe, and is subsequently uploaded to the debugger for display. The flow of ETM trace data is shown in Figure 1.

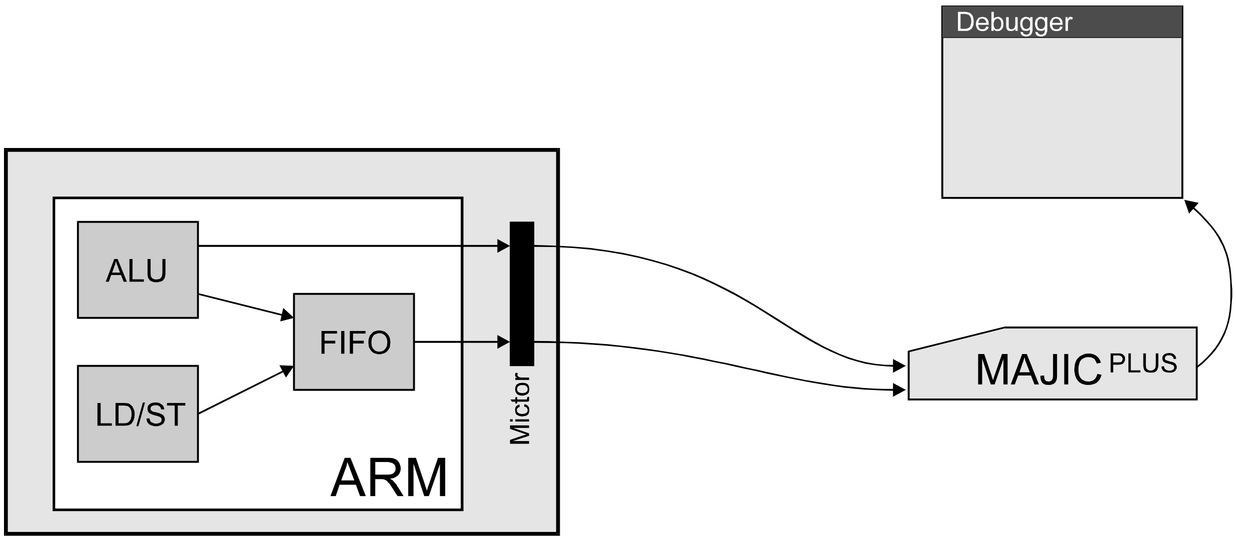


Figure 1: ETM trace data flow, from the CPU pipeline and load/store unit, through the MAJIC^{PLUS} probe, to the debugger.

The Embedded Trace Macrocell is a highly configurable execution trace interface. Some of its capabilities are configured by the silicon designer who incorporates ETM into your device, and some of the capabilities are programmable by the user. The start-up files in the `./targets` directories of your EDTA software installation take care of these details for the standard reference platforms, but you should check the technical details of the ETM implementation in your device before attempting to configure ETM for your target.

- What version of ETM is included in your ARM processor?
- How many "Trace Data Packet" (TRACEPKT) bits are connected from the FIFO to the Mictor connector in figure 1? ARM allows for 4-bit, 8-bit, or 16-bit mode. More trace data pins allow trace data to come off chip faster to avoid FIFO overruns, but costs extra pins.
- Are the ETM signals connected directly to dedicated output pins, as shown in figure 1? Or are the ETM signals multiplexed with other signals, and if so, how do you select ETM mode for those pins?
- Are there any special jumpers or switches on your board to control ETM output or connection to the 38-pin Mictor connector?
- What speed is your CPU pipeline? The trace clock is either the same rate, or half the pipeline speed in half-rate clock mode (also known as double data rate because data changes on both clock edges).

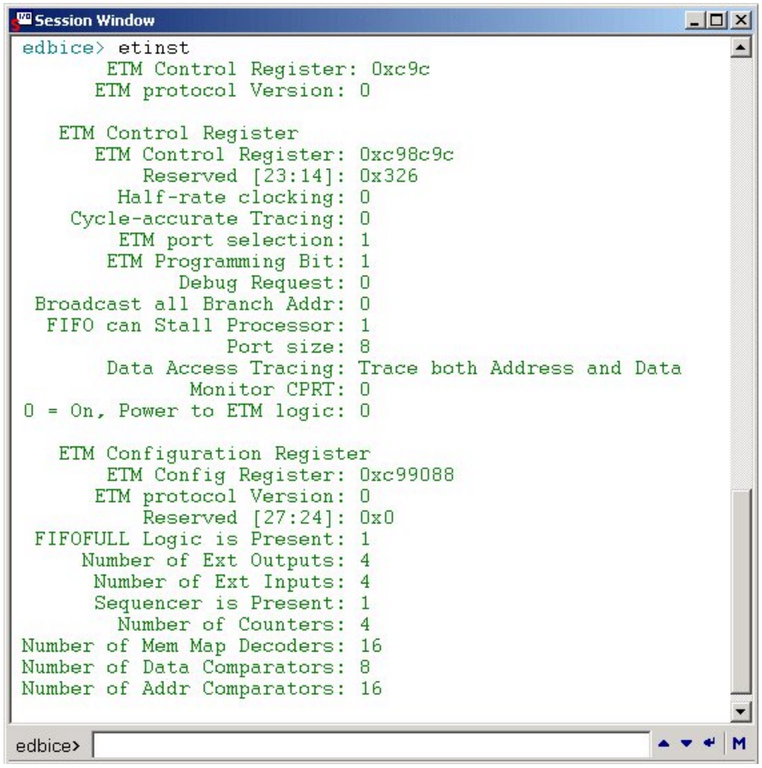
Configuring ETM

The Embedded Trace Macrocell is programmed by several command files included in the EDT software package installation. The default settings in these command files are good for the most typical case of wanting to trace program execution up to the next breakpoint, so most users simply need to run the desired command file(s) via the command alias(es) prior to start execution.

Note: The command files described below are located in the `./bin` directory of your EDT software installation. Many lower level command files are also located there, but most users do not need not worry those file. The command files described below are all that most users need to use (or possibly customize) in order to configure ETM.

Execution Tracing

The `etinst` command alias reads the `etrace.cmd` file, which configures ETM for tracing all instruction execution. This is the recommended configuration for most users who have at least 8 TRACEPKT pins (see notes below). After the `etrace.cmd` file configures ETM, it displays a table of ETM capabilities and settings. For example:



Notes:

- When using a MAJIC^PLUS/I with an Active Probe, the `etrace_ap.cmd` file should be used to configure execution tracing instead of `etrace.cmd`. You can either change the `etinst` alias definition in `startice.cmd` to read `etrace_ap.cmd`, or you can manually read it with the **`fr c etrace_ap`** command. By default, this configuration selects half-rate clock mode, so be sure the jumper on the MAJIC Adapter Module board is set for HALF.
- If your target has only four TRACEPKT pins (in addition to the TRACECLK, TRACESYNC and 3 PIPESTAT pins), then you must select 4-bit mode. Open `etrace.cmd` or `etrace_ap.cmd` with your preferred text editor, and change the **`$etm_Port_Size`** setting to **`1`**.

Data Tracing

The `etdata` alias reads the `eview.cmd` file, which is an adjunct to enable tracing of data transfers. You should always use `etinst` to configure basic tracing first, but if you want to trace data transfers as well program execution, then also use `etdata`.

Note: Turning on data tracing significantly increases the amount of information that ETM emits during program execution because the data address and value of each data transfer must be presented (unlike instruction execution, where only branch addresses are typically reported). This severely reduces the run length that can fit in the MAJIC^PLUS trace buffer.

ETM Trigger

The `etrace.cmd` and `etrace_ap.cmd` files define an ETM trigger if an instruction is executed within the range of 0..1F. The ETM trigger is referred to as a *Trace Point* and may be used to trigger MAJIC^PLUS trace acquisition, to set a marker in the trace data, or to trigger external test equipment.

If you want to use the ETM trigger to another address or address range, you should change the trigger address defined in the `etrace.cmd` or `etrace_ap.cmd`. Open the command file version that you use with your preferred text editor, scroll down to the section shown below, and change the start and end addresses indicated in **bold**.

```
//      *********************************************
//      *  USED FOR TRIGGER EVENT (FOR THIS EXAMPLE) *
//      *********************************************
//
if (@$etm_num_addr_comp < 0n10) { goto SKIP_TRIGGER_EVENT }
ew $etmX_ndx   = 4                   //Address Comparator Index (0-15) MUST be EVEN
ew $etmX_value = 0x0                 //Address Value [31:0]
ew $etmX_data  = @$A_DISABLE_DATA    //Select Data Comparator Enable or Disable
ew $etmX_mask  = @$A_MASK_NO         //Select Masking of Address A0 or (A0 and A1)
ew $etmX_type  = @$A_INST_FETCH      //Select Access Type (Fetch/Exec/Load/Store)
//
eacmp @$etmX_ndx  @$etmX_value @$etmX_data   @$etmX_mask   @$etmX_type
//
ew $etmX_ndx   = 5                   //MUST BE +1 of Start Address Range
ew $etmX_value = 0x1F                //MUST BE > Start Address Range
ew $etmX_data  = @$A_DISABLE_DATA    //MUST BE DISABLED for End Address Range
ew $etmX_mask  = @$A_MASK_NO         //MUST BE Set Identical to Start Address Range
ew $etmX_type  = @$A_INST_FETCH      //MUST BE Set Identical to Start Address Range
//
eacmp  @$etmX_ndx  @$etmX_value @$etmX_data   @$etmX_mask   @$etmX_type
:SKIP_TRIGGER_EVENT
```

Note: The default settings shown above generate an ETM trigger on the first instruction fetch from any address in the specified range. By changing both of the **`$etmX_type`** settings above to one of the following values, you can trigger on an instruction execution or data transfer instead: **`@$A_INST_EXECUTE`**, **`@$A_DATA_LOAD`**, **`@$A_DATA_STORE`**, **`@$A_DATA_LOAD_STORE`**.

MAJIC^PLUS Trace Acquisition

Once ETM is programmed to emit trace data, the next step is to enable the trace acquisition hardware in the MAJIC^PLUS probe. Normally, all that is required is to enter the **`+te`** command (perhaps in your board initialization file), or click the **`[+te]`** button in EDBICE or EPI's Platform Builder Plug-In.

On each active trace clock edge during program execution, all of the trace signals are captured in a dedicated trace buffer in the MAJIC^PLUS probe (subject to triggering and conditional tracing controls, explained below). Each captured trace cycle is called a *Trace Frame*. Starting and stopping program execution appends each runs' trace data, until you upload and display the trace data. Thereafter, a new acquisition is started. The width and depth of the trace buffer depends on your MAJIC^PLUS hardware configuration.

MAJIC^PLUS Model	Target Connection Option	Buffer Depth	Maximum TRACEPKT Width	Clock Modes
MAJIC^PLUS/I	CKP_ARM38/1 Passive Probe	512k	16	Normal (SDR)
MAJIC^PLUS/I	AP_ARM38/A Active Probe **OR** AP_ARM38+20 Active Probe	1024k	8	Normal (SDR) Half-Rate (DDR)
MAJIC^PLUS/II	AP2_ARM38/A Active Probe **OR** AP2_ARM38+20 Active Probe	1024k	16	Normal (SDR) Half-Rate (DDR)

Notes:

- The maximum TRACEPKT width is in addition to the TRACESYNC and three PIPESTAT signals.
- Refer to the *MAJIC Interface Specifications* application note for electrical specifications and connector details.
- Contact EPI technical support for information on multiplexed, demultiplexed, and dual trace modes, as these special modes are beyond the scope of this application note.

EDBICE, EPI's Platform Builder Plug-In, and other debug environments provide a trace window to display the trace data. Please refer to your debugger manual for information on displaying, searching, and filtering trace data. When there is no trace window, for example in MONICE or GDB, then you can use the **`DT`** command to display trace data. Use of the **`DT`** command is explained in Chapter 6 of the *MAJIC User Manual* and command syntax is shown in Chapter 5.

Triggers and Conditional Tracing

By default, trace acquisition is tied to program execution. The MAJIC^PLUS probe collects trace data in a circular buffer for as long as the target processor remains running. Once the buffer fills, the MAJIC^PLUS probe begins removing older trace frames from the buffer as necessary to make room for new information. When execution stops, the trace record includes the last N frames before the breakpoint, so you can see the events leading up to that point.

Alternatively, you can set the MAJIC^PLUS probe to capture the start of execution and automatically suspend acquisition when the buffer is full. That way you can see the start of program execution, in case you do not reach your expected breakpoint. Stop mode (also known as wrap mode) or Start mode trace acquisition is selected with the **`Trace_Trigger_Action`** configuration option. It is also possible to trigger start or stop of trace acquisition at a Trace Point (the ETM trigger) or the MAJIC^PLUS Trace Enable input, instead of execution status. The trigger source is set with the **`Trace_Trigger`** option.

In addition to the triggering facility, the MAJIC^PLUS probe can be set to automatically suspend trace acquisition if the processor enters an excessive stall condition, to prevent replacing useful trace history with stall cycles. Stall/Hang mode is particularly useful in cases where the trace CPU crashes hard, and you're not sure where or why. Trace acquisition can also be gated (qualified) by the MAJIC^PLUS Trace Enable input. These modes are controlled with the **`Trace_Gate`** option.

Note: Chapter 6 of the *MAJIC User Manual* describes these triggering and conditional trace features in more detail, and the configuration options used to control them.

Trace Acquisition Details

The triggering and conditional trace features mentioned above work essentially the same way for all MAJIC PLUS hardware configurations. However, the details of trace acquisition do vary between hardware models, as described below.

MAJIC PLUS/I With CKP_ARM38 Passive Probe

The DBGACK signal is used to qualify trace acquisition. When the processor is not in debug mode, tracing is enabled (subject to the triggering and conditional trace features described above). When the processor enters debug mode and asserts DBGACK, either because a breakpoint was hit or the user manually stopped execution, then tracing is automatically disabled.

Some ARM targets do not connect the DBGACK signal from the ARM core to the debug connector. If your target does not provide DBGACK then an active probe is recommended, although you can still capture trace data with a passive probe by making sure DBGACK on the debug connector is grounded. However, since the MAJIC PLUS probe cannot automatically shut down tracing in this case, you should make sure to select Stall/Hang mode with the **`Trace_Gate`** option. This suspends trace acquisition 512 cycles after execution stops by counting stall cycles, thereby preserving nearly all of the trace buffer.

MAJIC PLUS/I With AP_ARM38 Active Probe

When using an active probe, the DBGACK signal is ignored. Instead, trace acquisition is qualified by the pipeline status codes coming from ETM. Tracing is suspended after ETM reports a status of Trace Disabled (TD) for 256 consecutive trace clock cycles, and resumes when ETM reports a status other than TD. The TD status code means either that the processor has stalled long enough for the trace FIFO to clear, or that the processor is executing in an address range where tracing is disabled.

Some targets may stall for more than 256 cycles during normal operation, which can cause gaps in the trace data. The trace processing software must operate on contiguous trace cycles, and gaps can degrade the quality of the trace data. If your target can stall for over 256 cycles normally, then you should select Stall/Hang mode with the **`Trace_Gate`** option to suspend tracing after 1024 consecutive TD cycles instead of 256.

MAJIC PLUS/II With AP2_ARM38 Active Probe

When using a MAJIC PLUS/II probe, the DBGACK signal is ignored. Trace acquisition is qualified by the pipeline status codes coming from ETM. Normally tracing is suspended after ETM reports a status of Trace Disabled (TD) for 32 consecutive trace clock cycles, and resumes when ETM reports a status other than TD. The TD status code means either that the processor has stalled long enough for the trace FIFO to clear, or that the processor is executing in an address range where tracing is disabled.

Tracing is suspended in this way to minimize the buffer space consumed by stall cycles, and thereby maximize the trace history that can be viewed. If ETM is set for cycle accurate mode, though, then stall cycles are captured too so that the trace processing software can reconstruct accurate timestamp information. Tracing is still suspended when executing in a non-traceable address region, though, so timestamps can only be maintained within a traceable region.

When Stall/Hang mode is selected with the **`Trace_Gate`** option, tracing is enabled for all cycles until 2048 consecutive TD cycles have been detected, then tracing is disabled until the next non-TD cycle. Stall/Hang mode supercedes ETM cycle accurate mode, so tracing suspends after 2048 cycles even if the ETM cycle accurate indicator would normally have enabled tracing.

Notes:

- ETM cycle accurate mode is selected with the **`$etm_Cycle_Accurate`** parameter in `etrace.cmd` or `etrace_ap.cmd`.
- The **`Trace_Upload_Raw`** option must also be enabled for accurate timestamp reconstruction. Use the **`DOV TUR`** command for more information.