Using ARM ETB with TI CCS

CCS 3.3 with SR9 on TMS320DM6446
ETB Usage Brief Tutorial

1. Setup CCS setup configuration to include the ETB.
2. Connect to the target (including the ETB)
3. Select the ETB as the Receiver type in the XDS560 Trace system control panel. (Tools→XDS560 Trace→Control)
4. Turn the trace system ON by going to the Unified Breakpoint Manager (Debug→Breakpoints) and create a “Trace” operation. Select the action to be “Trace ON” and check the box for “Program Address” and “Data”. Enable the action and submit it. The trace system is ready!
5. Run your program to collect trace. If you know your point of failure, halt the core at that point. By default, the trace system will start/stop with the core. This means that every Run, Step, and Halt operation will refresh the Trace Display with trace information.
Make sure your CCS Setup includes the ETB
Parallel Debug Manager (PDM) on startup
Parallel Debug Manager (PDM) on startup

1. Select the target.
2. Choose “Connect”
3. Open the ARM processor CCS View
Parallel Debug Manager (PDM) on startup

1. Select the target.
2. Choose “Connect”
3. Open the ARM processor CCS View
Parallel Debug Manager (PDM) on startup

1. Select the target.
2. Choose “Connect”
3. Open the ARM processor CCS View
Choose Trace Control to select ETB as “Receiver”
Select Trace Receiver to be ETB

Choose “Receiver” to select Trace collection method

Select ETB
Select “Breakpoints” to turn the Trace system on. Can also do this through Tools→XDS560 Trace→Setup
The Unified Breakpoint Manager (UBM) is now available. Choose “New”
Select “Trace”
The Trace properties page is available. Click the “+” sign to choose when trace action (Start/Stop) as well as the type of tracing (program, data, etc.)
Select “+” on what to trace to select Program and Data

Note that different start/stop events can be used.
Select Program Trace

Check the “Enable” box.
Click the ‘Submit’ button to submit this trace job to the system.
Trace with ETB

◆ At this point, the ETB system is setup and is collecting trace.
◆ By default, the trace system will start/stop when Code Composer Studio™ (CCS) starts and stops the CPU.
◆ It is possible to choose different start/stop events for tracing, as these are based on the Advanced Event Triggering capabilities of the device (AET).
Trace Display is available under Tools ➔ XDS560 Trace ➔ Trace Display
Once you halt the processor, the Trace display will refresh with trace information. Choose Fields to select what is displayed. Right click-and choose “source” to setup the source code correlation.

One display shows the current line, and the second display shows the C file for that line.
By filtering for source code only, we can see the C code that the processor executed.
With Data trace enabled, we can see the processor load address and values. This is useful for checking whether the address is correct (pointer problems) and to validate the data values.
The trace information can be saved for later viewing and analysis.

In this case, I am looking at the data accesses for a particular function.