Advanced Debugging for Cortex-M Microcontrollers

Javier Orensanz
Agenda

- CoreSight™ Debug Technology for Cortex™-M MCUs
- Using Debug and Trace
CoreSight Debug (Cortex-M)

- Start, Stop, and Single-step User Program
- 8 Hardware Breakpoints
- On-the-Fly read/write access
- JTAG (5-pin) or Serial Wire (2-pin + 1 trace pin)
- Instruction Trace Stream
- Application Trace Information: Debug printf, ITM, DWT, ETM
- Data Trace or Access Breakpoints for 4 Variables
- ITM, DWT, ETM Output via 4 trace data pins + 1 clock pin
- Trace (ETM, ITM, DWT) not available on Cortex-M0

Core-M3 Processor

- Run Control
- Breakpoint Unit
- Memory Access Unit
- Debug Interface
- Cortex Debug 10-pin or ARM JTAG 20-pin Connector
- ITM, DWT Output via 1 serial trace data pin (UART or Manchester Mode)
- Cortex Debug + ETM 20-pin Connector (optional)
- ITM Instruction Trace
- Instruction Trace Stream
- DWT Data Watchpoint & Trace Unit
- CPU & Interrupt Events
- Trace Port Interface
- Serial Wire Viewer
- JTAG or Serial Wire Debug

The Architecture for the Digital World®
Debug and Trace Connectors

20-pin (0.1”) or 10-pin (0.05”) Connector
- Identical Debugging capabilities

Support 2 Operating Modes:
- Standard 5-pin JTAG mode (device chaining)
- Serial CoreSight mode
  - 2-pin **Serial Wire Debug** (SWD)
  - 1-pin **Serial Wire Trace Output** (SWO) for Data Trace at minimum system cost

20-pin (0.05”) Debug+ETM Connector
- Superset of 10-pin 0.05” Connector
  - Adds 4 (trace data) +1 (trace clock) pins for high-speed Data + Instruction Trace in any operating mode (JTAG or SWD)

Debug and Trace Adapters

ULINK2: Debug + Serial Wire Trace
- Flash Programming + Run-Control
- Memory + Breakpoint (access while running)
- Serial Wire Trace Capturing up to 1Mbit/sec (UART mode)

ULINKpro: adds ETM Streaming Trace
- Cortex-M processors running up to 200MHz
  - 50MHz JTAG clock speed
  - Serial Wire Trace Capturing up to 100Mbit/sec (Manchester Mode)
  - ETM Trace Capturing up to 800Mbit/sec
- Virtually unlimited Trace Buffer
  - Streaming Trace allows 100% Code Coverage and Performance Analysis
What is Streaming Trace?

- Trace data transferred in real-time to debug host
- Capture size only limited by host resources (hard-disk)
- Trace for minutes, hours, or longer
- Required for full code-coverage and timing analysis
- Today’s workstations can present trace data instantly

The Architecture for the Digital World®
Using Debug and Trace
Run-Stop Debugging has Limitations

- Stopping code execution changes system behaviour
  - Execution timing cannot be analyzed
- Many practical problems result from a run-stop debugging
  - Communications systems get into timeout state
  - Motor controllers freeze in high current state
CoreSight Offers Simple Solutions

- **#1**: Direct memory access to running system
  - Read and write memory and variables
  - Breakpoints can be set while system running
  - No software overhead, no extra hardware, works with any Cortex-M device!

- **#2**: printf-style output via an ITM Channel
  - Output details to a debug console
  - Uses CMSIS standard interface
Trace Records (DWT + ITM)

- Trace Records display program flow
  - Capture timestamp, PC sample, and Read/Write accesses
  - Time delay and lost cycles are noted
- Raw trace data from all trace sources
  - Filter window to refine the view
- Updated while target system is running
Instrumented Trace (ITM)

- 32 ITM channels: write to memory location creates trace data
  - Channel 0: for printf-style debug information
  - Channel 31: for RTX event viewer
  - Remaining ITM channels for user data output

```c
// Output 32-bit variable to ITM channel 1
ITM->PORT[1].u32 = value;
```
Detailed Code Analysis (ITM)

- **Parasoft C++ Test™**
- **Complete C/C++ quality solution for:**
  - Static code analysis and coding policy enforcement
  - Automated code review
  - Automated unit and regression testing
  - Host and target test execution
  - Coverage analysis

- **Integrated support with MDK-ARM**
  - Based on ULINKpro streaming trace
  - Annotated code uses ITM channel for unit test result feedback

- **More information:**
  - www.parasoft.com
Logic Analyzer (DWT)

- Allows signals to be monitored graphically
  - Monitor variables in the application
- Accurate timing
  - Easy, fast analysis of signal timing with access to source code
  - View delta changes from cursor to current location
- Code analysis
  - View instruction that caused variable change
Instruction Trace (ETM)

- Execution history of all executed instructions
  - Instruction Trace window displays: cycle count (timing) and assembly code synchronized to the C source code.

- Instruction Trace is useful to analyze sporadic problems
  - Data corruption by incorrect interrupt/thread protection
  - Incorrect timing caused by interrupt/thread nesting
void Alarm (void) {                      // Alarm Function
  if (clock.min != 59 ||                 // Validate Time
      clock.hour != 12)  {
    debug_printf ("System Should never be there");
  }
}

void CheckAlert (void) {                 // check for alarm at 12:59
  if (clock.min == 59 &&                  // check minute for 59
      clock.hour == 12)  {                // check hour for 12
    Alarm ();                             // call Alarm Function
  :

Instruction Trace shows Interrupt Execution within the compare statement
Code Coverage (ETM)

- Complete software validation requires code coverage
  - Required for industry standards such as IEC61508……..
- ETM enabled devices provide complete instruction stream
  - Non-intrusive - use final, optimized code at full speed
- Feedback provided directly in the debugger window
  - Source & disassembly view
- Log File Support
  - Coverage information can be saved for documentation
Execution Profiling and Analysis (ETM)

- Instruction Trace provides timing information
  - Identify where most time is spent in your application

- Isolate problems by finding which C statements take longer than expected to execute
Demonstration

Demonstration will show

- Working with Cortex-M3 and CoreSight components
- Using Streaming ETM Instruction Trace
- Code Coverage and Performance Analysis

Please visit AMMOS and ESS booths for the demonstration.
Summary

- CoreSight debug technology enables
  - Unique debug and trace capabilities for modern embedded systems
  - Helps speed development
  - Provides better verification methods
Thank You

Please visit www.arm.com for ARM related technical details

For any queries contact < Salesinfo-IN@arm.com >