

EE442/EE592 Real-Time Digital Signal Processing

Laboratory 3: Advanced Debugging Techniques

1 Debugging

The following debugging techniques use Program 5 (FIR filter) in the text.

1.1 Code Tracing without Codec Interrupts

Stepping or tracing through code can be difficult if interrupts are enabled since these may occur much faster than your step speed. Constant servicing of the interrupts will divert you away from examining the code of interest. Interrupts from the codec can be disabled and you can step through code in the following way.

- Shutoff codec (`ADA_OFF equ 1` in **.DAT** file) so that you can trace through code without interrupts
- Run and stop the code in the usual way so as to mask off interrupts
- Change PC to start of FIR code in the `process_stereo` subroutine (examine code in Unassemble Window)
- Trace through one line of code (`Trace` in Command Window) and examine memory and registers
- Trace through three lines of code (`Trace 3` in Command Window) and examine memory and registers

1.2 Software Breakpoints

In order to halt the execution of code at a particular instruction, we set a “software breakpoint.” You can practice utilizing software breakpoints by doing the following:

- Shutoff codec (`ADA_OFF equ 1` in **.DAT** file) so that you can step through code without interrupts
- Scroll to start of FIR code (examine code in Unassemble Window)
- Double-click (in Unassemble Window) the first line of the FIR routine thereby setting a breakpoint
- Run DSP program and verify the code is halted at first line of FIR routine
- Step through code (Command Window) and examine memory and registers
- Scroll to start of FIR code and double click first line of FIR routine thereby clearing the breakpoint

The debugger supports many other features of software breakpoints. For more information, see Examples 6–17, p. 4-12 – 4-13 of *Debug-56K User's Manual*.

1.3 Hardware Breakpoints

In order to halt the execution of code when a specific hardware action takes place, such as writing to a particular memory address, we can use a “hardware breakpoint.”

- Change PC (in command window) to start of FIR code (examine code in Unassemble Window)

In the command window write the hardware break as shown in *Debug-56K User's Manual*. For example, in Program 5, if you want to break the execution after the first read operation from the queue (located at x:\$10),

```
break r xa x:000010
```

This instruction will break the execution of code just after the first read from x memory location 000010. Similarly there are break operations for write, read and write together. These operations can be extended to y memory as well.

See Examples 6–17, p. 4-12 – 4-13 of *Debug-56K User's Manual* for more information on hardware breakpoints.

1.4 Graphing/Plotting the Contents of Memory

Values in memory can be visualized by graphing or plotting the contents memory. This is useful in examining filter coefficients or a signal which has been stored in memory (queue).

- In the data window, go to the area in memory where the FIR filter coefficients are stored

Select the Graph Button

1.5 DSP Reset

Occasionally the DSP will get into “funny” state and cause problems. In this case, force a reset on the board with:

In the Command Window, type `FORCE R`

After reset, you must make sure Verify Memory Writes is Disabled, Com Baud is set to 56K, and most importantly rerun the pass code to configure the EVM properly—the Debugger has no idea that the EVM has external memory.

1.6 Miscellaneous Debug Commands

JUMP
SAVE
WATCH