ARM and Thumb Mode

About ARM and Thumb Mode

ARM and Thumb are two different instruction sets supported by ARM cores with a "T" in their name. For instance, ARM7 TDMI supports Thumb mode. ARM instructions are 32 bits wide, and Thumb instructions are 16 wide. Thumb mode allows for code to be smaller, and can potentially be faster if the target has slow memory. Please consult documentation from your MCU vendor, or documentation from ARM, for additional information.

Detecting Thumb Mode

EMUL-ARM supports both ARM and Thumb mode. To do this, it must know which memory regions that are Thumb mode, which is automatically detected when loading a program. However, this is dependent on the output from the compiler/linker. Different compiler vendors give different amount of information. Thumb mode detection should work on the c/c++ level.

Memory not detected as Thumb are considered ARM. For details regarding Thumb mode detection for your compiler, please check the "ARM_Compilers" document. Note that thumb mode is not detected when:

- There is program memory on your target for which code is not loaded.
- The load file does not support detection – such as Intel Hex or Motorola S-Record.

Note - if EMUL-ARM does not know which memory regions that are Thumb:

- Source stepping and breakpoints will no longer work. (However, HW breaks do work.)
- The disassembly will not be correct.

The easiest way to check if an address is considered to be ARM or Thumb is to check the disassembly. If the instruction is 32 bits wide – it is ARM, if it is 16 bits wide it is Thumb.

Manual Configuration

If needed, Thumb mode regions can be manually configured by selecting menu "Config|Emulator" and go to the "Map Config" tab. When Thumb regions are configured manually, EMUL-ARM will consider an address range to be Thumb, if it is automatically detected or if manually detected. There is no provision for manually override a region automatically detected as Thumb to be treated as ARM.

Detecting Thumb Mode in Run Time

At entry into "debug state" (i.e. after Step or Break), it is detected if Thumb mode is active. If it is, assembly stepping will work independently of automatic detection and manual configuration. However, source stepping will not work.

When setting the execution point (writing to the PC), the new location is checked for ARM/Thumb status and the internal configuration is updated accordingly. This is based on Automatic detection and manual configuration. PC must align to even 32 bits address, unless in a Thumb region.

Modifying the Thumb bit in the CPSR (Current Program Status Register) in the Seehau Register Window has no affect. That bit is checked for and writes to it will be removed before the contents being passed to the ARM core.