



10 Clever Lane .Tewksbury, MA 01876  
Ph: 978-455-3383 Fx: 978-926-3091

Email: [info@ultsol.com](mailto:info@ultsol.com) Web: [www.ultsol.com](http://www.ultsol.com)

## **Getting Started With Config Files**

Generating a proper config file is the most difficult part of starting out with the BDI2000/BDI3000. The config file is used to initialize your board up to the point where SDRAM and FLASH are usable, after this you can put an initial boot agent onto the board or replace the existing boot agent. The information you need to develop a successful config file is as follows:

CPU Documentation, usually the Programmer's Reference Guide or User's Manual  
Board documentation, including memory map and significant chip identification  
From board docs:

- Flash memory data sheets
- SDRAM chip data sheets
- Board Clocking info
- Board-level CSR Register definitions

The process is an engineering effort. Depending on the board and components used on the board, it can take anywhere from a day to a couple of weeks to get it right. Every processor and board combination is different. As an example, take a look at one of the example config files found in the distribution from Abatron for both 8xx and 4xx (i.e. a 405). Walk through each line in the [INIT] section with the corresponding processor user manual and other chip documentation and try to figure out just what it is doing to the CPU/board. That will give you a good head start on what you need to write one from scratch.

Remember that the sequence of events is usually critical, for example on the 8xx, you must set the IMMR before you can access anything else, and on many processors you have to disable the watchdog right away. Another example is on the 8245, where you have to set up the entire memory controller registers and then set the MEMGO bit as the final step in the sequence or it doesn't work.

So a sequence to creating a config file will be as follows:

1. Copy an existing config file from the unzipped bdi directory that most closely matches your CPU, then rename it as your own custom config file.
2. Comment out all the init sections in this config file that may not pertain to your particular CPU keeping only the IMMR section, shutting off the watchdog timer, the memory controller initialization and any sections pertaining to initializations of the Flash and SDRAM.
3. Use STARTUP RUN and let your boot loader bring up the board. At the BDI prompt, Type in "halt" to stop the target.
4. One by one read back the registers in the [INIT] section of the config file and update them with the values that are read back from the BDI. Use the md command to read back registers "md 0x0f010100".
5. Make sure you know which chip selects are needed to configure Flash for your board.

Bottom line: a developer has to have a very detailed knowledge of both his CPU, his board and usually the FLASH and SDRAM subsystems on the board in order to be successful.