



---

# IceCube Software

---

---

## *DOM MainBoard Computing Environment*

---

**Chuck McParland, LBNL**

---

**1.0 Introduction**

---

**1.1 Purpose**

When operating, the DOM Main Board's behavior is determined by a combination of factors that include hardware, firmware and software. Given the level of complexity found in each of these areas, it is important to understand just how these architectural components interact. Furthermore, since several of these components will change and evolve over time, it is important to know just what elements constitute a complete DOM Main Board release. This document is intended to help clarify these issues.

**1.2 Scope**

This document's scope is limited to design components (hardware, firmware and software) located on the DOM Main Board. This document is not intended to give detailed explanations of particular component features or describe how one can interact with or configure these features. Those details will be found in other documents. It will however, describe the structures that govern how they interact and thus define some portion of the DOM's "configuration".

**1.3 Definitions, acronyms and abbreviations**

DOM MB: DOM Main Board

Excalibur: Altera chip that combines hardwired core processor and field programmable logic elements.

FPGA: Field programmable gate array portion of the Altera Excalibur.

CPLD: Programmable logic chip used to facilitate communications between CPU and a number of DOM MB elements (e.g. ADC's, DAC's).

BFD: IceCube structured software development system.

Configboot: Lowest level DOM MB bootstrap program (limited functionality)

Iceboot: Secondary DOM MB bootstrap program.

Stfserv: Framework used to configure and control execution of simple DOM MB tests.

DOMapp: DOM MB data taking application. Used for all data acquisition activities.

Jtag: hardware cable mechanism for loading each of the programmable parts on the DOM MB when physical access to the circuit board is available.

**1.4 References**

Iceboot users reference manual.

---

## **2.0 Major Software/Firmware Components**

---

For the purposes of this document, the DOM MB consists of the following major components:

Excalibur CPU and FPGA chip.

CPLD programmable logic chip.

Configuration bootstrap memory.

Flash memory arranged as a simple file system.

Main memory used for program execution and data storage.

Various files (see below) that define FPGA behaviors and contain executable Excalibur programs.

Misc. logic (DACs, ADCs, etc.) located on the DOM MB and addressable by Excalibur programs.

Single RS232 communications port accessible by Excalibur programs.

Twisted pair connection providing power and, if properly configured, communications.

---

## **3.0 Non Volatile Storage**

---

The DOM MB has three on board devices that are used for non volatile storage. Two of them, the CPLD device and the configuration serial boot memory, have specific well-defined functions. Both of these devices are programable by external cable through connectors located on the main board. Neither device can be reprogrammed by programs executing on the Excalibur CPU. Thus, once the glass sphere surrounding the main board is sealed, these devices are considered unalterable.

The third non volatile device, flash memory, is also programmable by external cable. However, unlike the other two on board storage devices, flash memory can be partially or completely rewritten under CPU control. Flash memory is primarily used in two ways. First, the Excalibur CPU, when properly restarted, can read and execute code contained in the first portion of its 8Mbyte address space (flash boot sector). Accordingly, this section of memory has been reserved for a copy of the IceBoot bootstrap program. Secondly, the remaining portion of flash memory is organized into a simple, single directory layer flash file system. All persistent storage of programs, text files, and fpga binary images is done in this portion of memory.

It should be pointed out that while the structure and format for data storage in the CPLD device, configuration serial boot memory, and the flash boot sector are determined by various portions of the hardware design, the area of flash memory dedicated to the flash file system is solely determined by software. Thus, any attempt to read, write or interpret files (e.g. programs, FPGAs images, etc.) stored in the flash file system must be done through software executing on the Excalibur CPU. In most cases, this is done within the IceBoot program.

---

## **4.0 Flash File System and File Naming Conventions**

---

Any binary image that can be downloaded to dynamic memory by the IceBoot program can be stored in the flash memory file system and be given a reference name. Other than name, byte length and starting address (in flash memory), files have no additional attributes or access protections. All files are bound to a single name (i.e. no aliases) of 16 characters maximum length. Although file names are treated as simple strings, file-names can contain extensions for the sake of clarity. For example, "abcd", "stf.exe", and "domapp.sbi" are all legal filenames.

We have adopted a naming convention for the base set of files contained in the flash file system. This base set of files is a subset of the DOM\_MB software release generated from the BFD IceCube development environment and contains all files necessary for normal DOM operations in both the STF testing and DAQ data acquisition environments. By convention, all FPGA file names end with the ".sbi" suffix. Presently, the following files and file names are part of the DOM MB distribution and are considered "reserved" names.

### **Program Files:**

iceboot (secondary DOM MB bootstrap program)

stfserv (simple test framework server)

domapp (DAQ data acquisition program)

### **FPGA Files:**

iceboot.sbi (FPGA code for use with iceboot)

stf.sbi (FPGA code loaded with stfserv)

domapp.sbi (FPGA code loaded with DOMapp)

### **Misc. Files:**

iceboot config (iceboot configuration information, internal use only)

startup.fs (supplemental forth commands loaded with iceboot)

Note that the configboot program does not appear in this list as it not stored in flash memory. Also note that, although the iceboot program does appear as just another file within the flash file system, it is handled and stored as a special image that occupies the boot sector of flash memory. Additionally, a file named "iceboot config" appears at the end of physical flash memory. Since this file name is created by the flash memory initialization code within iceboot and has no real contents. Therefore, it is not part of the DOM MB distribution. However, its name should considered reserved. Lastly, it is anticipated that additional stf server programs will be written to allow test programs to be collected into functionally related groups. All these programs will have names that begin with the string "stfserv" (e.g. stfserv1, stfservMonitor, etc.). Therefore all file names beginning with this character string should be considered reserved for future use.

---

## **5.0 Required FPGA Images**

---

Both boot operations and execute commands cause resets of the Excalibur CPU. And, since every Excalibur CPU reset operation clears FPGA memory, execution of any program on the DOM MB is typically preceded with an FPGA load operation. Furthermore, since all current applications are required to communicate over the twisted pair power cables, all program execution commands cause the FPGA to be loaded with a program containing a copy of the communications circuit. There may conceivably be exceptions to this rule. But they will be limited to special programs that either do not communicate with external systems or are only intended to communicate over the on board RS232 interface (possibly prior to sealing the enclosing glass sphere). At present, no such programs are foreseen.

Except for twisted pair communications support, programs will have varying requirements for FPGA features. Therefore, we have specified different FPGA files for each DOM MB application. Programs and iceboot command scripts will always load the appropriately named FPGA file prior to starting execution of a new program. In some unusual (and transient) cases, programs may use identical FPGA files. However the file will be duplicated in flash memory and be referenced by the appropriate file name.

---

## **6.0 ConfigBoot Environment**

---

Although the configboot program and its FPGA file follow the above role, due to Excalibur interface requirements, these two files are treated differently from others.

The FPGA file associated with the configboot program along with the program itself, is stored in the serial configuration flash memory chip. These two files are concatenated along with required header information. This is necessary to produce a properly formatted single byte stream that is compatible with the Excalibur's serial memory bootstrap loader. For this reason, neither the configboot program or FPGA file appear as individual files in the flash file system. The configboot FPGA program file is the simplest used on the DOM MB as it only need to support twisted pair communications functions.

---

## **7.0 IceBoot Environment**

---

As described earlier, the iceboot program is contained in the boot sector of flash memory. Its associated FPGA file resides in the flash file system under the name iceboot.sbi. At the start of iceboot execution, the FPGA is loaded with the contents of this file. This is necessary to allow iceboot to communicate over the twisted pair. At present, there are no other FPGA features required for iceboot operation, so iceboot.sbi should remain relatively simple.

Since iceboot is intended to operate as a low level diagnostic tool as well as a program loader, there will be occasions when users will manually instruct iceboot to load a new FPGA (e.g. iceboot "fpga" command). If continued communications over the twisted

pair are expected, this FPGA file must contain the required communications circuit. It is the users responsibility to insure that the image load contains the appropriate logic for the intended DOM MB configuration.

The iceboot program allows some command customization. Several common commands are contained in the startup.fs file found on the flash file system. Since, prior to accepting user commands, iceboot executes all command found in this file, this file contains any required iceboot customization. Since this file is part of the DOM MB distribution, it allows some customization of iceboot commands as part of the normal software release cycle. Within a given iceboot command session, further customization is possible. However, after any reboot operation, iceboot will return to the default configuration defined by the startup.fs file. This file is critical to the iceboot initialization process and, if it becomes corrupted, flash memory will need to be reprogrammed.

---

## **8.0 Stfserv Environment**

---

The Stfserv program environment assumes that the appropriate FPGA file has been loaded prior to start of execution. Elements of the communications library used by Stfserv determine the presence of a DOM MB jumper indicating which channel (RS232 or twisted pair) is to be used for command communications. After minimal initialization, Stfserv waits for command input and executes tests linked into its binary execution image (loaded from flash). In practice, individual Stfserv tests can access portions of the DOM MB through both CPLD and FPGA data paths and it is assumed that the appropriate firmware designs have been loaded into both the CPLD and FPGA portions of the DOM MB. Since program behavior is unpredictable when accessing firmware features that have not been loaded or implemented, reliable operation can only be insured by using a consistent set of DOM MB components -hardware, firmware and software. Although this is primarily the responsibility of the software release procedures found in BFD, consistency can be undermined by manual reconfiguration of either Jtag loaded parts or flash memory files located on the DOM MB. Therefore care should be taken when changing individual components on the DOM MB.

---

## **9.0 DOMapp Environment**

---

While different in purpose, most of the above comments also apply to the DOM MB when executing the DOMapp program. One minor exception is that, for upstream configuration reasons, it is not planned to execute DOMapp while communicating over the RS232 communications channel. Similar care is necessary when changing DOM MB components. Correct operation of both hardware and software can only be insured by having a consistent set of components properly installed on the DOM MB.