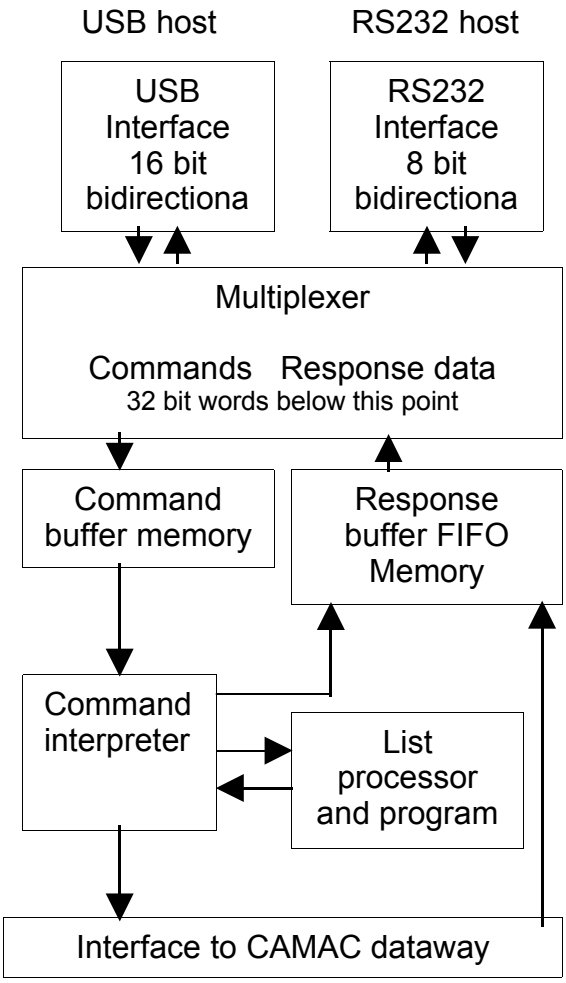
CMC100 FastCamac USB Crate Controller

The CMC100 contains two host interfaces, USB 2,0 and RS232. Incoming commands and data from either source are assembled into 32 bit words and sent to the command interpreter via a FIFO buffer. The total buffer space available is about 1000 words. Commands for the crate controller are executed immediately and the result is sent to the response buffer memory, also organized as a FIFO. Commands for the dataway are executed by the dataway interface and the response sent to the response buffer. Commands and instructions intended for the program store are sent to the list processor and stored.

Both host interfaces (USB and RS232) can operate simultaneously, commands are flagged with the source information. The response data is examined by the multiplexer and directed to the host that supplied the incoming command.

The list processor program memory is 512 words, and can contain multiple short programs. There are two registers, a 20 bit counter and a 32 bit accumulator (accum). Programs can be executed in 3 ways, a LAM, an external NIM pulse or a command from the host. The LAM executes the instruction in location 1 and the pulse executes location 0. These should contain jump instructions to the appropriate location. The command from the host can begin execution at any location.



CAMAC Modules

*Fast*Camac is implemented for both level 1 and level 2, up to the maximum rates (one data transfer on the dataway every 100 nS). All features of *Fast*Camac are implemented except *Fast*Camac writes and wide (48 bits) reads. Read instructions are limited to the normal 24 read lines.

The response buffer memory is organized as a FIFO and can contain about 1,048,000 Camac words.

The CMC100 can read data from Camac modules at up to 30 Mbytes/sec. The 24 bit data words are expanded to 32 bits (adding a byte containing X, Q, Lam and crate #) and sent to the FIFO buffer. The rate into the USB interface is 40 Mbytes/sec. The USB host must pull the data from controller. The USB 2.0 interface transmits data to and from the host in 512 byte blocks, at about 11 microseconds per block. The USB interface can also operate as USB 1.1, with a reduced data transfer rate.

The USB Interface

The CMC100 requires no power from the USB cable, it is a self powered USB device. It is compatible with both USB 1.1 and USB 2.0. There are six USB endpoints implemented in the CMC100.

- Endpoint 0 is the bi-directional control endpoint and is used by the Operating System to initialize the USB. Endpoint 1, 2, 6 and 8 are used to control and communicate with the crate controller using USB bulk transfers.
- Endpoint 1, bulk in (data from USB chip in the crate controller to the host), 64 byte buffer size. Reads the unit number (the switch on the front panel), the Lam status and the most significant 2 bits of the word count in the 1Mword buffer memory.
- Endpoint 1, bulk out (data from host to crate controller), 64 byte buffer size. Writing anything to this endpoint will reset the crate controller by forcing the FPGAs to reload from the flash memory chip. This is the only function of this endpoint.
- Endpoint 2, bulk out, 512 byte buffer size, quad buffered. Commands and data from the host are sent to this endpoint, in blocks of 32 bit words.
- Endpoint 6, bulk in, 512 byte buffer size, quad buffered. Response words from the crate controller to the host, via the 1M fifo, are read from this endpoint as blocks of 32 bit words.
- Endpoint 8, bulk in, 512 byte buffer size, double buffered. Response words from the crate controller to the host, bypassing the 1M fifo, are read from this endpoint as blocks of 32 bit words.

For maximum performance, a High Speed USB 2.0 host is required. If a USB 1.1 host is used, all buffers are reduced to 64 bytes. This is usually transparent to the user, and is handled by the operating system. All data transfers will be much slower of course, and the maximum rate will be about 1 Mbyte per second.

The USB buffers are normally available to be read by the host only when full (they are automatically committed to the USB when full). The flush (or commit to USB) command will cause a partially full buffer to be committed to the USB and available for transmission to the host. It is important when requesting a USB read operation to specify a block size that is an integral number of the buffer size (128 words). If less than a full block is requested, and the next block to be sent is larger than that, the extra data will be discarded by the driver. The USB read operation will terminate when the word count is satisfied, or a short block or a zero length block is encountered.

When sending data to the controller, the block can be any size, up to the limits imposed by the operating system and USB driver (1 Mbyte for Windows), but a practical limit is about 2000 words, limited by buffer space in the controller. The USB system automatically breaks the data into appropriate size blocks.

USB 2.0 cables are limited to about 15 feet by time delay considerations. However, this can be extended by using USB hubs. Up to 5 hubs (or active extension cables) can be used, with each adding 15 feet to the length of the link (total > 90 feet). The use of hubs does not substantially affect the data transfer speed.