**DOS**

The DOS should provide a number of commands that allow the user to access files and other data from an interface. The DOS should provide a fixed number of channels, each channel being able to have its own state. The state includes the current path, having a file open, etc.. The DOS commands will work in a command/response manner.

Examples of the commands are:

* Global commands:
  + Identify
  + Initialize
  + Enter 1541U Freezer
  + Get RTC time
  + Set RTC time
  + …
* Channel related commands:
  + GetPath
  + SetPath
  + File Create
  + File Open
  + File Close
  + File Seek
  + File Read (with DMA option)
  + File Write (with DMA option)
  + File Sync
  + Directory Open
  + Directory Get Entry
  + Mount File on Drive
  + Read Block from Image
  + Write Block to Image
  + …

**Interface**

The interface provides the means to communicate with the DOS eventually. It consists of two sub-layers:

* The actual hardware interface
* The command/response translator

An example of a command/response translator is the “IEC” translator, that makes this DOS look like a CBM drive.

**Hardware interface – I/O bus**

* Consists of 4 visible registers from the C64 side:
  + $00 Status Register (RO, Write to clear sticky bits)
  + $01 Command Data (W) / Command Status (R)
  + $02 Command/Data handshake(W)
  + $03 Data Channel (R/W)
* The status register shows the state of the low level interface, like: ready to accept command, command status available, ready to accept data, data available.
* The command data register is the register to pass command data, of one of more bytes.
* The command status data register (R) is used to provide status data, error codes, error strings from the command interpreter.
* The command/data handshake register helps in grouping command/data bytes together. This register has bits like “command complete”, “data complete”, and signals to the interface layer that the bytes now in the queue belong together and should be processed as one packet.
* The data channel register is used to pass data to/from the interface. In case of DMA transfers, this register is not used.

These registers could be located anywhere in DE00-DFFF range, obviously in steps of 4 bytes. The most likely location would be $DFF0-$DFF3; so that it would not interfere with REU or RR. Other locations might be needed for other cartridges.