



Jamieson T. Olsen
Engineering Note

DFE Crate Controller (DFEC) Command Structure and 1553 Interface Specification

Engineering Note 2001-04-16a

Revised 16 January 2002

Summary

The DFE Crate Controller (DFEC) is a 6U custom board that resides in slot 1 of a DFE or Mixer crate. It's primary functions are to store device configuration files in non-volatile CompactFlash memory, execute a list of commands written to it over the 1553 network, and provide up to date status information collected from the boards in the DFE crate. ***This specification describes how to talk to the DFEC over the 1553 network; it also defines the commands and their arguments.***

This document replaces the following Engineering Notes:

**2000-06-29a,
2000-06-29b,
2000-08-07a,
and 2000-08-11b.**

1553 Interface

A 1553 bus supports up to 31 Remote Terminals (RTs). Each RT supports up to 30 subaddresses. Up to 31 16-bit words can be read or written to any given subaddress. The DFEC requires one RT address.

Unlike other 1553 devices where each subaddress has a unique and specific function, the DFEC uses a fewer number of subaddresses to access a block of memory that is shared between the 1553 RT slave circuitry and the DFEC microcontroller. The 1553 host writes command(s) and data into specific locations within this block of memory and then tells the DFEC to execute the command(s). The status of the DFEC can be accessed as well. There are three subaddresses used to access the DFEC:

subaddr	function		words
	read	write	
16	address pointer	address pointer	1
17	data register	data register	1-31
18	DFEC Status Word	execute command(s)	1

The address pointer is accessed through subaddr 16. This pointer can be read or written to at any time. It's purpose is to specify where in the block of memory the next data operation will take place. The address pointer is a single 16-bit word. If more than 1 word is written to this subaddr, the pointer will take on the value of the LAST word written.

The data register is where the contents of the block of memory are accessed. Up to 31 words can be read or written to the block of memory with each 1553 transaction. The address pointer is automatically incremented after each data word is read or written to this register. By using the address pointer and data register memory locations anywhere in the block of memory can be read or written to.

The final subaddr (18) is used to tell the microcontroller on the DFEC to execute a list of commands stored in a specific area in memory. Writing a word (data doesn't matter) to subaddr 18 will interrupt the microcontroller and tell it to begin executing commands. Reading from subaddr 18 will return the DFEC Status Word, which is described below:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	IDLE	HALT									CMR	DTE	CFR	FNF	FUL

The DFEC can be in only ONE of THREE states:

Bit 15: Busy. Set when the DFEC is busy processing a command. Do not write to the block of memory when this bit is set or you'll risk overwriting commands.

Bit 14: Idle. Set after the DFEC has finished processing the its list of commands and there were no errors. Look for this bit to be set before writing new commands into the block of memory.

Bit 13: Halted. Set if an unknown command is encountered or if one of the command arguments is bad. When this happens the DFEC stops executing commands, sets the appropriate status bits in the lower byte of the DFEC STATUS WORD, sets the HALT bit, and awaits new commands.

If the DFEC is HALTED, check these bits to figure out what went wrong...

Bit 4: CMR. Command Error. Set when the DFEC finds an unknown command or argument(s) in the DFEC Command Buffer area of memory. This bit is also set if the "End of Command" marker is not found in the Command Buffer.

Bit 3: DTE. DTACK Error. The DFEC tried to talk to a slot in the backplane where there was no board, or the board was not responding properly and the DFEC did not receive a DTACK signal back from it.

- Bit 2: CFR. CompactFlash Error Bit. Set when the DFEC cannot initialize the CompactFlash card, or if it is otherwise unresponsive and the DFEC times out trying to access it. Also set if the CompactFlash is present but NOT formatted as a MS-DOS FAT16 partition.
- Bit 1: FNF. File Not Found Bit. Set if a command references a FileID that is not found on the CompactFlash card.
- Bit 0: FUL. CompactFlash Full Bit. Set if there is no more space left on the CompactFlash card.

If the DFEC is in BUSY or IDLE mode, the lower 5 bits of the DFEC STATUS WORD are guaranteed to be zero.

1553 Memory Map

The 1553 “RT Slave” circuitry supports accessing a full 16-bits of memory, or 128k bytes arranged as 65536 16-bit words. In reality this block of memory is much smaller, only 512 16-bit words. Valid addresses are from 0000h to 01FFh. Do not read or write to addresses outside of this range. The memory map is defined as follows:

	UPPER BYTE								LOWER BYTE							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0000h	DFEC status word															
0001h	command buffer															
:::	:::															
:::	:::															
007Fh	:::															
0080h	board status block															
:::	:::															
:::	:::															
00F9h	:::															
00FAh	general purpose memory															
:::	:::															
00FFh	:::															
0100h	sector buffer															
:::	:::															
:::	:::															
01FFh	:::															

The DFEC status word is located at address 0000h. *The preferred method of accessing the DFEC status word is to simply read a single word from subaddr 18.*

Next is the command buffer. It’s a block of 127 16-bit words that are used for storing lists of commands and arguments. When the DFEC is told to begin executing a list of commands, it begins at address 0001h and continues until it finishes the list of commands. The number of commands in a list is variable, as is the length of commands. (More on this later)

The board status block consists of 120 words that describe the status of the rest of the boards in the DFE crate. This block of memory is continually updated and requires no special commands to be executed on the DFEC. It can be read *at any time*. ***Refer to my engineering note 2000-01-31a for more information about this section of memory.***

The Sector Buffer is used when files are transferred to and from the CompactFlash card sector by sector. Each sector is 512 bytes or (256 16-bit words) long. The first byte of a sector goes in the lower half of memory location 0100h; the last byte of a sector goes in the upper half of memory location 01FFh.

Command Summary

The DFEC supports a mixture of high level and low level commands. Commands are variable length, but always an integer number of words long. They can be divided into high level and low level commands.

Low Level Commands

The low level commands are generally intended for debugging and maintenance. They are not intended to be used on an everyday basis. Low level commands bypass safeguards imposed by the DFEC and therefore can really screw up the system if used improperly.

1. Write a byte to backplane.
2. Read a byte from the backplane.
3. Transfer a CompactFlash sector to the Sector Buffer.
4. Transfer the contents of the Sector Buffer to the CompactFlash card.
5. Transfer a file from the CompactFlash to the backplane.

High Level Commands

These commands are essentially macros created out of many commonly used low level commands. These are designed to be used most often. These commands are:

6. Delete a file from the CompactFlash card.
7. Append the Sector Buffer to a file on the CompactFlash card.
8. Generate the checksum of a file on the CompactFlash card.
9. Reset a board.
10. Set the clock select bits for a board.
11. Configure a device from a file on the CompactFlash card.
12. Return the firmware revision byte for a device.
13. No Operation (NOP)
14. Get File Size.
15. End of List marker.

Lists of Commands

The DFEC is designed to be list processor that deals with variable length lists of commands. Furthermore, the commands themselves are of different lengths. The DFEC must know when to stop processing the list of commands so that it does not try to interpret leftover garbage in the command buffer as a valid command. This is the purpose of the `End_Of_List` command. This one byte command tells the DFEC that it has reached the end of the list of commands and it can return to IDLE mode.

File System Information

The file system used on the CompactFlash card is MS-DOS FAT16. This means that the CompactFlash card can be removed from the DFEC and inserted into any commercially available CompactFlash card reader in any PC. Files can then be copied, moved, renamed, and deleted using Microsoft Windows Explorer.

FAT16 supports cluster sizes from 1k to 32k, however the DFEC is limited to supporting cluster sizes from 2k to 16k. Thus the user needs to be careful when formatting the CompactFlash card on the PC – some manufacturers default to formatting the CompactFlash card as old FAT12, which the DFEC doesn't understand. Contact me before formatting any CompactFlash card that is to be used with the DFEC.

DFEC commands expect files to be named with a four character string consisting of "hex" characters 0-9 and a-f. When a file is referenced in a command this four character hex string is represented with a single 16-bit word. For example, the 16-bit word 12E8h references the file named "12E8????.???" (where '?' is a wildcard character) on the CompactFlash card. Files created by the DFEC will be named "nnnn_DFE.BIN". Note that these files are BINARY and DO NOT have an End of File character.

Chances are that a file will never be exactly an integer number of clusters long – therefore wasted space will almost always occur. When a file is transferred down to a device the entire file – including the cluster waste at the end – will be sent. All devices shall ignore this extra data after configuring.

Low Level Commands

1. Write Byte to Backplane

This two word command takes three arguments: the slot number, the memory offset and the data byte. This low level command allows the user to write a single byte to DFE motherboards or Mixer boards. For more information on these addresses, refer to my engineering note 2000-01-28a. The format of this command is as follows:

1	1	1	1	1	1												
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		
0	0	0	1	0	0	1	0	data byte								first word	
offset												slot				second word	

If the backplane address is valid – that is, if there is a functioning board that responds to the address – then that board will accept the data byte and reply with a DTACK signal. If the DFEC does not get this DTACK signal back, then it assumes that the address was not valid, sets the DTE bit in the DFEC STATUS WORD, and halts command execution.

4. Sector Buffer to CompactFlash

This command takes the contents of the Sector Buffer and writes it into the specified sector of the CompactFlash. Any data contained in the specified sector of the CompactFlash is overwritten when this command is executed. **WARNING: this command has the potential to really mess up the FAT16 file system, so use with care!**

1	1	1	1	1	1												
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		
0	1	0	0	0	0	1	0	lba high byte								first word	
lba mid byte								lba low byte								second word	

If the CompactFlash device is not responding or if the LBA address points to a sector past the capacity of the CompactFlash card, the DFEC will set the CFR bit in the DFEC STATUS WORD and halt command execution.

5. Copy File to Backplane

This three byte command transfers a file from CompactFlash to a backplane address. Chances are that the specified file will not be an integral number of clusters long; the resultant unused space (“cluster waste”) between the end of the file and the end of the last cluster WILL be transferred to the backplane address when this command is executed. These value of extra bytes is not known, but it isn’t a problem since the Xilinx FPGAs will ignore the extra bytes.

1	1	1	1	1	1												
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		
0	1	0	1	0	0	1	1									first word	
offset												slot				second word	
file name																third word	

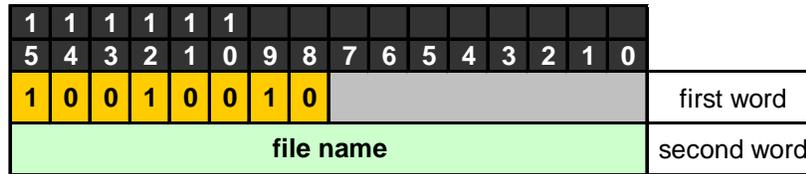
No handshaking between the board and DFEC is used when this command is executed – bytes are pulled out of the CompactFlash and written to the backplane address as quickly as possible. The DFEC does not check to make sure that the specified backplane address makes sense – it just dumps the file to that address.

The following DFEC STATUS WORD bits could be set if there is a problem when this command is executed: DTE (DTACK error), CFR (CompactFlash Error), and/or FNF (File Not Found).

High Level Commands

6. Delete File

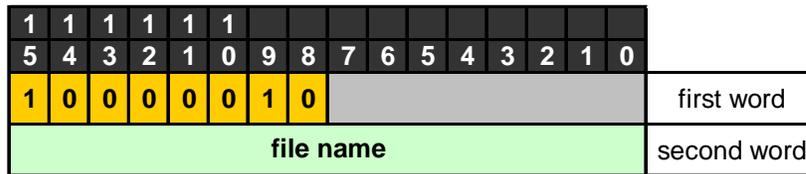
This two word command deletes a file from the CompactFlash memory.



If the file is not found on the CompactFlash memory the DFEC will set the FNF bit in the DFEC STATUS WORD and halt. If the CompactFlash is not responding properly the DFEC will set the CFR bit in the DFEC STATUS WORD and halt.

7. Append Sector to File

This two word command is the preferred method of transferring a file from the online system to the CompactFlash card.

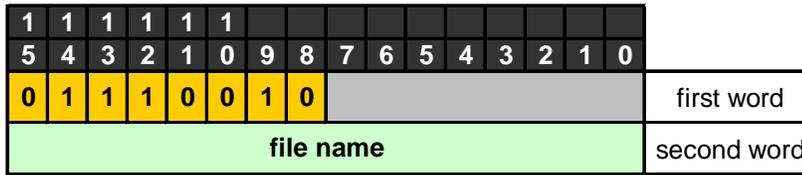


When this command is executed the data contained in the Sector Buffer will be tacked on to the end of the specified file. If the file name does not exist, it will be created at the time of command execution.

If the CompactFlash is not present (or is not responding properly) when this command is executed the DFEC will set the CFR bit in the DFEC STATUS WORD and halt. If the CompactFlash is full the DFEC will set the FUL bit in the DFEC STATUS WORD and halt without writing the contents of the Sector Buffer.

8. Generate File Checksum

This command performs a 16-bit checksum on the specified file. (This command is used instead of a lengthy file read back command.) This command takes approximately 1 minute for a 1.6Mbyte file.



The 16-bit value is returned to memory location 00FCh as shown below:



The algorithm used is “checksum-16” which adds all of the bytes in the file and returns the lower 16 bits of the final sum. The “cluster waste” is ignored in this calculation.

This command can set the following bits in the DFEC STATUS WORD and halt:

- ~~CFR~~ CFR – CompactFlash not present or is not responding
- ~~FNF~~ FNF – File not found

9. Reset Board

The one word command performs a hardware reset on the board located in the specified slot of the backplane. Valid slot numbers are 2 (00010) through 21 (10101). Resetting a board does not cause devices located on that board to erase their configuration memory.



If the board does not respond with a DTACK pulse the DFEC will set the DTE bit in the DFEC STATUS WORD and halt.

10. Set Clock Bits

This one word command sets the clock selection bits on the specified board. The clock selection bits determine the source of the master clock that will be distributed to all devices on the board. For more information about the meaning of the clock selection bits, refer to board specific documentation.

1	1	1	1	1	1												
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		
1	1	0	0	0	0	0	1			clk	slot number					first word	

If the board does not respond with a DTACK pulse the DFEC will set the DTE bit in the DFEC STATUS WORD and halt.

11. Configure Device

This three byte command is the preferred method of configuring a device. There are four arguments for this command: the slot number and device fields uniquely identify a device in the crate. The firmware rev byte is a user defined value used to identify the firmware revision number. Lastly, the file name is specified.

1	1	1	1	1	1												
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		
0	1	1	0	0	0	1	1			slot number					first word		
				device				firmware rev byte					second word				
file name															third word		

Valid slot numbers are 2 (00010) through 21 (10101).

Executing this command involves talking to the CompactFlash and the backplane, so there are some situations that could result in the DFEC halting on this command and setting the following DFEC STATUS WORD bits:

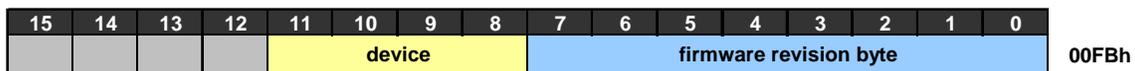
- ~~CFR~~ CFR – CompactFlash not present or is not responding
- ~~FNF~~ FNF – File not found
- ~~DTE~~ DTE – Backplane write error / board not responding

12. Get Firmware Revision

This two word command queries the specified device on the board in the specified slot and returns the firmware revision byte.



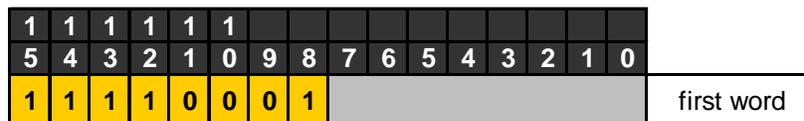
The firmware rev byte is written into the lower half of memory location 00FBh. The upper half of memory location 00FBh contains the device number, as shown below:



This command does not check to see if memory location 00FB contains data from a previous command – it simply overwrites it.

13. NOP Command

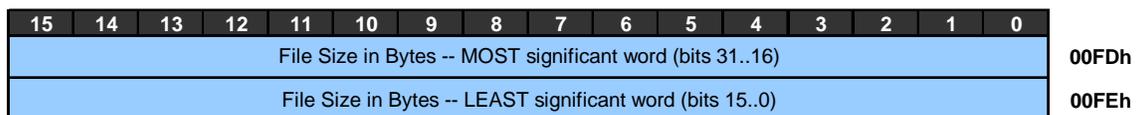
This command does absolutely nothing. Stefano Rapisarda wanted it for some reason or other.



14. Get File Size



Get File Size checks to see if the specified file is present on the CompactFlash card. If the file is present, the size of the file (in bytes) is written into memory locations 00FDh and 00FEh as a 32-bit unsigned integer:



If the specified file is not found, these two memory locations are zeroed out and the DFEC will HALT with the File Not Found bit set.

15. End of List

This word marks the end of the list of commands that is present in the DFEC's Command Buffer section of memory. This marker **MUST** follow the last command since it tells the DFEC to stop executing commands and return to idle mode.

1	1	1	1	1	1														
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0				
1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

first word

If this marker is not present, the DFEC will continue working through the command buffer memory looking for the next command. Chances are that if this happens the DFEC will hit garbage and halt execution – BUT there is always the chance that the DFEC will interpret the junk in memory as a valid command and execute it.

Don't get in this situation! Always place this marker in the command buffer immediately following the last command in the list of commands. Don't fill up the entire Command Buffer, either.

Command Examples

1. Download a file to the CompactFlash

This example demonstrates how to transfer a ASCII *.HEX file from the host system to the DFEC. The file name will be "ABCD" on the CompactFlash card.

1. Read the DFEC STATUS WORD and verify that it is in idle mode. (read subaddr 18)
2. On the host computer: Read 1024 characters from the HEX file and put them into an array. Compress the "HEX ASCII" characters into 512 unsigned bytes. For example, if the HEX file begins with the 8 character string "FF5A669F..." it will be compacted into a 4 byte array beginning with {FFh, 5Ah, 66h, 9Fh}
3. Set the Address Pointer to the beginning of the Sector Buffer (write 0100h to subaddr 16)
4. Write the contents of the array into the data register (subaddress 17). The first byte of the array (FFh) goes in the lower byte of the first data word, and the next byte (5Ah) goes in the upper byte of the first data word. Thus the first data word of the 1553 transaction will be 5AFFh followed by 9F66h, etc. Continue writing the contents of the array until all 256 words have been written into the sector buffer.
5. Now the Sector Buffer is full and the **Append Sector to File** command can be entered into the Command Buffer. Set the Address Pointer to the beginning of the Command Buffer by writing 0001h to subaddr 16.
6. The **Append Sector to File** command only has one argument: the filename. Since we want to call it "ABCD" the second word of the command will be ABCDh.
7. Write 8200h, ABCDh, and A100h to the data register (subaddr 17). The first word is the opcode for **Append Sector to File**, the second word is the filename, and the third word is the End of List marker.
8. Now write 1 word to subaddr 18 – the data doesn't matter. This will cause the DFEC to execute the commands you just wrote into the command buffer. The DFEC will first check to see if the CompactFlash is ready. Then it will check to see if the file "ABCD" is present. It isn't, so the DFEC will create the file and transfer the contents of the Sector Buffer into the new file.
9. Repeat steps 1-8 until the HEX file has been transferred over. Pad the remainder of the last sector with zeros.

2. Calculate the Checksum of a File

1. Read the DFEC STATUS WORD and verify that it is in idle mode. (read subaddr 18)
2. Set the Address Pointer to the beginning of the Command Buffer (write 0001h to subaddr 16)
3. Write 7200h, A67Ch, and A100h to the data register (subaddr 17). The first word is the opcode for the **Generate File Checksum**, the second word is the filename "A67C", and the third word is the **End of List** marker.
4. Now write 1 word to subaddr 18 – the data doesn't matter. This will cause the DFEC to execute the commands you just wrote into the Command Buffer. The DFEC will first check to see if the CompactFlash is ready. Then it will check to see if the file "A67C????.???" exists. If so, it will begin to generate the checksum. This can take a few minutes, depending on the size of the file.
5. Poll the DFEC STATUS WORD by reading from subaddr 18. Wait until the BUSY bit is cleared, then check the HALT bit. If the DFEC is halted, check the lower 5 bits of the DFEC STATUS WORD to figure out why it crapped out. If the DFEC is idle, then the command was successful, continue to step 6.
6. The 16-bit checksum has been calculated and it is stored in memory location 00FCh. Set the Address Pointer to the first word of the checksum by writing 00FCh to subaddr 16.
7. Read 1 word from the data register. This is the "checksum-16" value for the specified file.

3. Configure device and verify the Firmware Revision of that Device

Preliminaries: the target firmware file “76A4????.???” already exists on the CompactFlash. It is going into device 2 of the board in slot 15. The firmware revision byte for this device is 8Fh. This is a list of two commands being executed one after another.

1. Read the DFEC STATUS WORD and verify that it is in idle mode. (read subaddr 18)
2. Set the Address Pointer to the beginning of the Command Buffer (write 0001h to subaddr 16)
3. Write in the command for **Configure Device** followed by its arguments: 630Fh, 028Fh, 76A4h. The first word is the **Configure Device** opcode and slot number, the second word is the device number and firmware revision byte, the third word is the filename.
4. Now write in the command for the **Get Firmware Revision** command followed by its arguments: B20Fh, 0200h, A100h. The first word is the **Get Firmware Revision** opcode and slot number, the second word is the device number, and the last word is the **End of List** marker. The Command Buffer should now look like this:

address	data	comment
0001h	630F	Configure Device Command
0002h	028F	
0003h	76A4	
0004h	B20F	Get Firmware Rev Byte
0005h	0200	
0006h	A100	End of List
0007h	X	X

5. Write 1 word to subaddr 18 to execute the commands. Poll the DFEC STATUS WORD until the DFEC is done with the command.
6. When the DFEC returns to IDLE mode, go get the Firmware Rev Byte located at memory location 00FBh. Write 00FB to the Address Pointer (subaddr 16).
7. Read 1 word from the Data Register (subaddr 17). It should be 028Fh. The 02 is the device number, and the 8F is the firmware revision byte.

Other Stuff...

- ?? To check if a file exists on the CompactFlash, use the **Generate File Checksum** command. If the file exists, the command will execute and the DFEC will return to IDLE mode. If the file does not exist, the DFEC will halt with the FNF bit in the DFEC STATUS WORD set.

- ?? Files can be transferred to the CompactFlash directly from a PC using a flash card reader. Don't use the **Append Sector To File** command on these files. Differences in the way the length of the file is reported could result in corruption of the file when the **Append Sector To File** command is executed on a file that was written to the CompactFlash via a card reader.

Summary of Changes

- 16 April 2001: Create this new document to replace multiple DFEC engineering notes.
- 17 August 2001: Commands 1, 2, and 5: replace 16-bit BACKPLANE ADDRESS field with SLOT and OFFSET fields. FAT16 now supports multiple cluster sizes, not just 32kB. Updated file name notes on page 5. Clarify that the file checksum algorithm does not consider the undefined "cluster waste" at the end of the last cluster of a file.
- 29 August 2001: Change from "checksum-32" to "checksum-16" instruction. Remove the DNS status bit and replace it with the CMR status bit.
- 22 Oct 2001: Add NOP and GET_FILE_SIZE commands.
- 16 January 2002: Expand the OFFSET field to 8-bits for the "Write Byte to Backplane", "Read Byte from Backplane" and "Copy File to Backplane" commands.