Procedure for generating and downloading a FINDER FPGA design.

The design in the EPC16 device can be downloaded to the Finder FPGA using a command initiated through VME or upon power cycle of the PC board.
Each of the two Finder FPGAs on the Stereo Finder board has two EPC16 devices associated with it to allow for a quick change from one algorithm to another(each EPC16 device can contain a different Finder designs). The EPC16 devices can be loaded via a JTAG chain or accessed using the VME path. Since there are two Finder FPGAs and each has two EPC16 devices there are 4 EPC16 devices that can be downloaded with Finder designs.
The term "load" refers to the loading of data to the EPC16 devices and the term "download" refers the the transferring of the data from the EPC16 device to the FPGA.

Upon a power cycle the design in the EPC16 device that is labeled as "B" is downloaded to the Finder FPGA. (At the moment the two Finder FPGA's are labeled as "A"(cell 1-18) and "B"(cell 19-36) and the two EPC16 devices with each FPGA are labeled "A" and "B". So Finder FPGA "A" has an EPC16 "A" and a EPC16 "B" and Finder FPGA "B" has an EPC16 "A" and EPC16 "B". We should consider renaming the four EPC16 devices to 1-4).

To load the EPC16 devices using the JTAG chain use the ALTERA Quartus software - load the appropriate file into the location in the JTAG chain that is designated for the appropriate EPC16 device and hit the program button. The Finder FPGA's can also be loaded directly using the JTAG chain, as can the other FPGA and configuration devices on the PC board.

For the VME load of the EPC16 devices use the code that has been written by Rod Klein. The source code is in the SXFTv2 section of the CDF ONLINE CVS repository at CDF. The procedure to load the EPC16s requires a number of control bits that need to be set or reset as detailed below. The commands in the SXFTv2 code are in blue text.

1. In the Finder Board "Control/Status" register(YY00 0004) set the appropriate bit 25 or 26 that disables downloading of the EPC16 contents to the Finder FPGA. This needs to be set so that data isn't transferred while the EPC16 is being loaded by VME. Turning both bits on will not affect the loading of only one of the EPC16 devices.(Option 2 get you to the Control Register in the SXFTV2 code and then option 6 and 7 turn on the bits)

2. Select the appropriate EPC16 device that is being loaded in the Finder Board "Flash RAM Bank Select" register(YY00 0010). Only one Bank should be selected at a time. (Option 6 get you to the Flash RAM Bank Select Register in the SXFTV2 code and then option 1-4 selects the EPC16 device, option 5 will turn all bits off)

3. Erase the contents of the EPC16 device that has been selected. The erase procedure takes a few minutes.(Option 18 gets you to the a menu that allows

erasing, writing and reading of the EPC16 device – code is already written for these procedures)

4. Select the .hexout file to load into the EPC16 device that has been selected. The EPC16 device is 16M bits and will take a while to load. The .hexout file is always the same size, the actual number of bits that are "on" varies with the Finder design. The number of writes to the EPC16 device is hardcoded in the SXFTv2 code. If the design size increases/decreases the number of writes can be changed to minimize EPC16 loading time. (Option 18 gets you to the a menu that allows erasing, writing and reading of the EPC16 device – code is already written for these procedures)

5. After the EPC16 has been loaded bits 25 and/or 26 of the Control register should be set back to "0". .(Option 2 get you to the Control Register in the SXFTV2 code and then option 6 and 7 turn off the bits)

6. The Flash RAM Bank Select register should be cleared. (Option 6 gets you to the Flash RAM Bank Select Register in the SXFTV2 code and option 5 will clear all bits to off)

7. To download the design in the EPC16 device to the Finder FPGA, access the Control Register and set bit 24 "on" or "off" depending on which EPC16 device contains the design that should be downloaded. (Option 2 get you to the Control Register in the SXFTV2 code and then option 8 controls bit 24 of the register)

8. To force the design to be downloaded from the EPC16 to the Finder FPGA toggle bit 29 or 30 for the appropriate Finder FPGA(A or B). (Option 2 gets you to the Control Register in the SXFTV2 code and then option 2(A) or 3(B) forces the download from EPC16 to Finder FPGA)

9. The download time between the EPC16 and Finder FPGA is minimal. Control register bits 9 or 10 represent a configuration done bit from the FPGA. (Option 2 gets you to the Control Register in the SXFTV2 code and bits 9 and 10 are represented on the screen as to whether the FPGA has been configured)

10. To verify that the Finder FPGA has the correct design loaded - access the Finder FPGA Firmware ID register(YY20 0000 for FGPA A and YY30 0000 for FPGA B). (Option 14 get you to the Finder FPGA Internal Registers and option 3 in that menu reads back the Firmware register to the screen, option 1 allows the switching between FPGA A or FPGA B)

Method to generate the ".pof" and ".hexout" files to load to the EPC16 devices.

The files that are generated automatically by the ALTERA Quartus software for a design when it is compiled have the extension ".sof" and ".pof". These files are found in the design directory and have the same file name as the design. The ".sof" file can be loaded directly to the Finder FPGA using the JTAG chain.
The file that can be loaded directly to the EPC16 devices using the JTAG chain is a ".pof" file but not the ".pof" that is automatically generated as mentioned above.

To make the EPC16 ".pof" file to load using the JTAG chain select the Convert Programming File option under the File pull down tab. A window opens that allows the generation of the file. The procedure for generating this file are:

1. The Programming file type should say "Programmer Object File (.pof)".
2. Under the Mode pull down - select "Fast Passive Parallel".
3. Select the "Options.." block and another window will open. In that window check the "Compression Mode" box. Close that window with the "OK" tab.
4. Select an output file name – suggest that you name the file with the version and revision number. i.e. FinderAv1r5.pof – the file will have the .pof extension.
5. In the section "Input file to convert" select the text "SOF Data" and it will highlight. Click the "Add File…" button and a window will pop up, select the .sof design you want converted. It will be added to the "Input File to convert" window when you hit the open tab.
6. Hit the "OK" button and the file you named in item 3 above will be generated to the directory you named in item 3.

The file that can be loaded to the EPC16 devices using the VME path is a ".hexout" file. The ".hexout" file is converted from the ".pof" file that was made for the JTAG downloading and not made from the ".pof" that is automatically generated.
To make the EPC16 ".hexout" file select the Convert Programming File option under the File pull down tab. A window opens that allows the generation of the file. The procedure for making this file is:

7. Change the Programming file type to say "Hexadecimal(Intel Format) Output File for EPC4/8/16 (.hexout)".
8. Select an output file name – suggest that you name the file with the version and revision number. i.e. FinderAv1r5.hexout – the file will have the .hexout extension.
9. In the section "Input file to convert" select the text "POF Data" and it will highlight. Click the "Add File…" button and a window will pop up, select the .pof design you want converted. It will be added to the "Input File to convert" window when you hit the open tab. **The .pof that is added should be the <u>one</u> that was converted form the .sof in steps 1-6 above. Using the .pof that is automatically generated creates a .hexout file also – it can be downloaded but the FPGA doesn't seem to configure. Looking at the contents of the two .pof**

files shows different data and a different number of bits 'on' within the file(? For ALTERA).

10. Hit the "OK" button and the file you named in item 8 above will be generated to the directory you named in item 8.
11. The .hexout file that is generated will need to be in a location that it can be accessed by the SXFTv2 code.