Basic Triggering and Data Capture with the FuturePlus Systems FS4300

This application note discusses how to use the FS4300 labels and symbols to create specific trigger conditions and to selectively capture data on a Fibre Channel link.

Introduction to Triggering with the FS4300

The principle behind specifying trigger conditions for the FS4300 is to use the **WrdTyp** label and its associated symbols to define the general type of word to be detected (e.g. *primitive*, *header*, *payload*, *SOF*, *EOF*, etc.) and then to use other labels to specify the specific instance of that type. The following simple example illustrates this principle:

Let's say that we want to trigger on the loop initialization primitive set to acquire a valid AL_PA (LIP F7, F7.) To do this we define a trigger pattern (which we'll call LIPF7,F7) with the general word type as primitive and a specific primitive value of LIP acq. The pattern is defined by specifying that it contains **WrdTyp** with the value *PRIMITIVE* (selected from the list of symbol values associated with the label **WrdTyp**) and the specific instance selected from the list of **primtve** symbols to be *LIP acq*. This is shown in Figure 1.

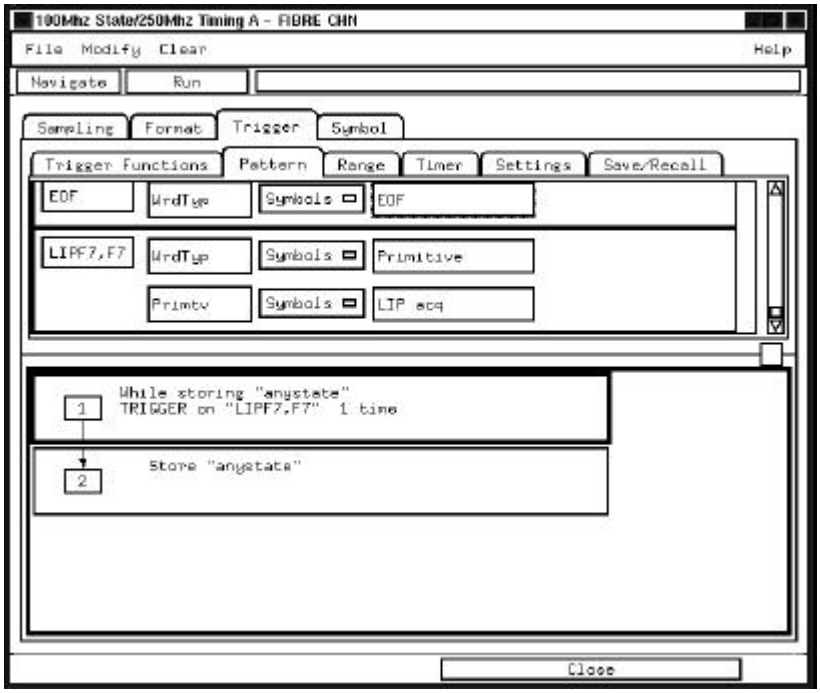

Figure 1: Triggering on a Specific Primitive

Remember that in general it is necessary to specify both a **WrdTyp** and a specific value for that type when defining trigger conditions. Bearing that in mind let's look at some more complex examples.

Capturing Packets between 2 specific Nodes.

If the traffic of interest is always between two particular nodes then a combination of store qualification and trigger specification can be used to only capture packets that originate on one node and are destined for another. Figure 2 below shows an example of this. A trigger pattern called D_S_ID is created that specifies the condition of a **WrdTyp** of *SOF* and a Destination ID of 555555 and a Source ID of AAAAAA. Note that the D_ID and S_ID are only valid when **WrdTyp** is *SOF* so it is necessary to specify a general word type of SOF as a qualifier to the D_ID and S_ID labels. See the section on the Format menu in the FS4300 Users Guide for a table showing when each label is valid. A pattern called EOF is also created (but is not shown in the figure) that is defined as a **WrdTyp** of *EOF* (see Figure 1or Figure 4 for the definition of this pattern).
The trigger function and store qualification is then set to specify that the analyzer only capture states that match the D_S_ID pattern and to trigger on that pattern but to then capture all states until the EOF pattern is detected. The analyzer will trigger on a frame with the required source and destination Ids and store all the states that represent the frame up to the end of frame. After the EOF no further information will be stored until the next frame that matches the pattern (and therefore the specified source and destination ID). To ensure that maximum memory is available the trigger position should be set to the start of memory.

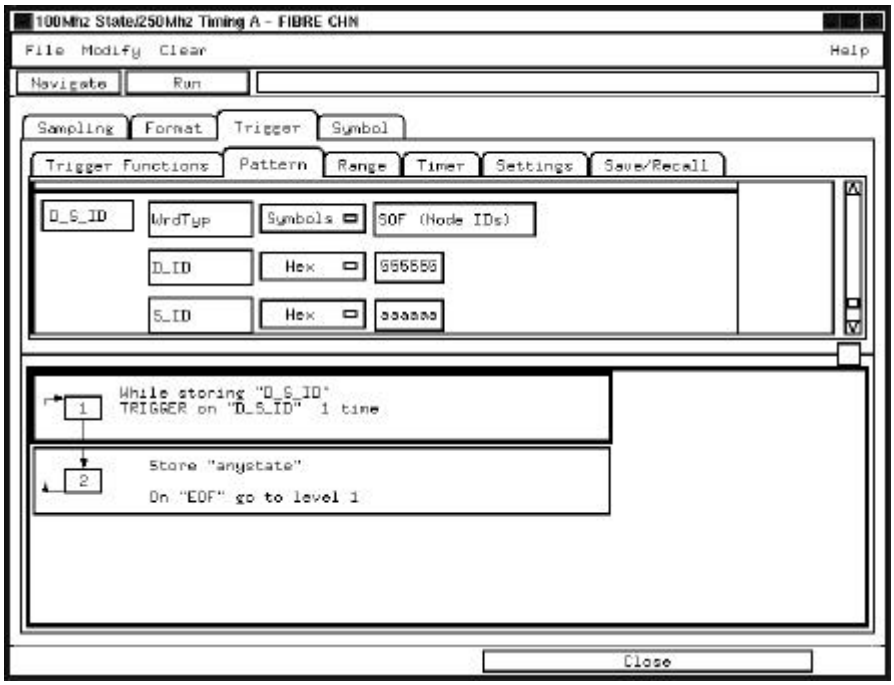

Figure 2: Capturing Traffic between Specific Nodes

Using the FrmTyp Label

Here is another example of using a combination of triggering and store qualification to only capture a particular type of frame and so significantly improve the utilization of the logic analyzer's memory. However, this example uses the label **FrmTyp** as the qualifier instead of **WrdTyp**. The label **FrmTyp** has a number of associated symbols that represent common types of frames (Appendix 1 in the FS4300 Users Guide provides a list of defined frame types). Using **FrmTyp** as a trigger or storage qualifier specification saves the need for more complex trigger patterns that look across a number of fields in a frame header.

In the example we want to only capture frames that represent FCP commands that are sourced from node 2 and destined for node 4. The **FrmTyp** label is valid throughout a frame but has the value 0 in between

frames. Therefore, in the example shown in Figure 3 the level 1 statement specifies that the logic analyzer is to look for and trigger on data that meets the frame description, i.e. a FCP command from node 2 to node 4. Once the trigger is found the level 2 statement instructs the analyzer to capture all data until the End of Frame is encountered and then to look for the next frame that meets the requirements. Note that again the trigger specification uses a trigger pattern called EOF that is defined to be **WrdTyp** of *EOF* (see Figure 1 or Figure 4 to see how this pattern is defined). Of course if the requirement is to capture FCP commands between any nodes then the S_ID and D_ID fields in the trigger pattern are omitted so that the pattern only specifies a **FrmTyp** of *FCP_CMD*.

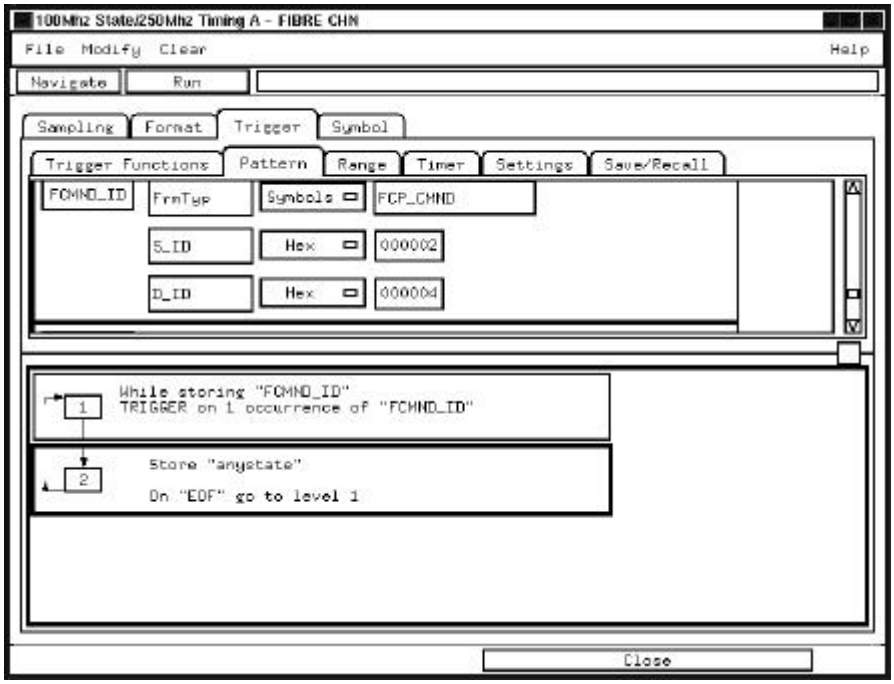

Figure 3: Capturing Frames of a Particular Type

Capturing Frame Headers

A third example shows another use of the store qualifier mechanism to significantly save capture memory and hence allow a longer period of Fibre Channel traffic to be stored. In this case the requirement is to only capture frame header information. Although the FS4300 provides a mechanism to filter data payload in FCP packets the payload of other frame types is not filtered. However, by creating a store qualifier and trigger specification as shown in Figure 4 only the header information is stored and payload for any frame type is filtered. In this case trigger patterns for Start of Frame, End of Frame, and a **WrdTyp** of *Header* are created. The first level of the trigger and store qualification specification instructs the logic analyzer to only store states representing SOF or part of a frame header and to trigger on the first occurrence of a header word. The second part of the specification instructs the analyzer to store all words with a **WrdTyp** of *Header* and then to return to level 1 when a word other than a header word is detected. Level 1 specifies that both SOF and header words are to be stored so that the SOF word is seen in the listing display and can be used to delimit different frames

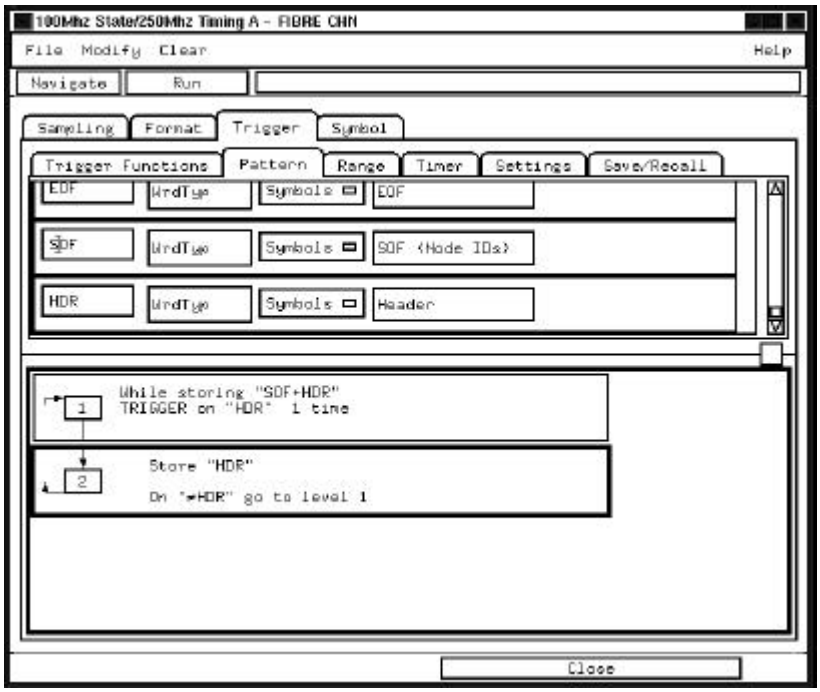

Figure 4: Capturing Frame Headers Only

Figure 5 below shows the result of applying this trigger and store qualifier combination. It can be seen that only frame headers and SOF words have been captured.

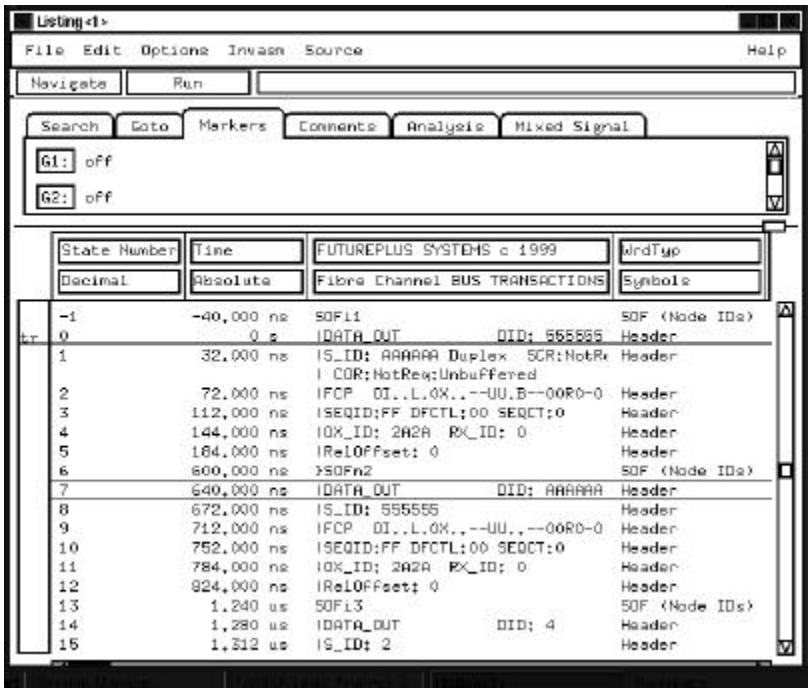

Figure 5: Result of Capturing Frame Headers Only

If you also wanted to capture the EOF word you could modify the second level to store “HDR” and “EOF” and to return to level 1 on EOF. The analyzer will only store words with **WrdTyp** of *Header* or *EOF* so payload will not be stored. This is shown in Figure 6 and the resulting listing display in Figure 7. Comparing Figure 5 and Figure 7 it can be seen that the EOF word has also been included.

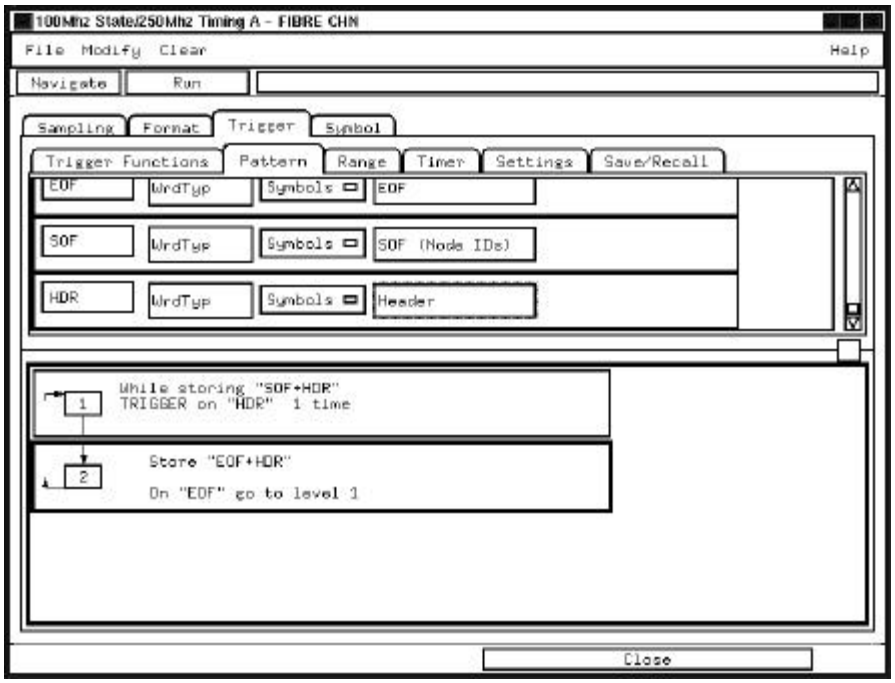

Figure 6: Capturing only SOF, the Frame Header and EOF

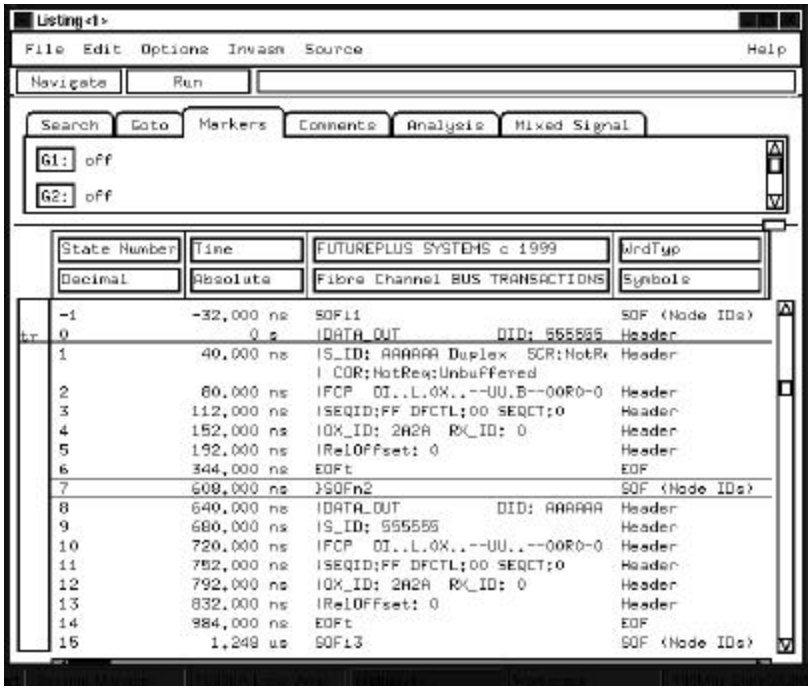

Figure 7: Listing showing capture of SOF, Frame Header and EOF

FuturePlus Systems Corporation
2790 N. Academy Blvd. Suite 307
Colorado Springs, CO 80917
Phone: 719 380-7321
FAX: 719 380-7362
Email: sales @futureplus.com
Web: www.futureplus.com