FEBEX4A Trigger Matrix V3

Summary:

This document describes the functionality and interface of the Trigger Matrix V3 module implemented on the FEBEX4A. The trigger matrix modifies which traces are read out in response to triggers that are aggregated over 1 to 65535 clock cycles (10 nS to 655.36 uS). This reduces the data per trigger allowing a higher event rate. It is envisaged that

Features:

- 19 trigger sources (16 CFDs on ADC channels + 2 external global triggers + dummy register trigger).
- 17 trace memories which can selectively be read out in response to triggers (16 Channels + Dummy register).
- Each of the 19 trigger sources has a 17 bit long row in the 17*19 trigger matrix. Setting a bit in the trigger matrix causes the corresponding trace memory to be read out when the trigger source is received in the trigger window.
- Acquisition window allows aggregation of triggers up to 65535 clock cycles (16 bit clock cycle length value). This is for the circumstance that triggers arrive not at the same clock cycle. I.E an implant close to a readout wire and one further away at the same time instance but the charge transport taking longer from the further away implant. In such circumstance attempting to trigger twice is undesirable.
    - The information on whether a trigger is accepted is obtained on the 6th clock cycle after the trigger so this window can be shorter or longer than the acknowledgement time.
    - Setting window value to 0 disables aggregation (only triggers on the same clock cycle used to select memories)
    - Window length is common to all trigger sources
- For the trigger matrix to reduce the number of channels to readout the data_reduction register (0xFFFFB0) must be written to with the value 0x1FFFF. This enables data reduction functionality on all 17 memories. E.G gosipcmd -w -x 1 $card 0xFFFFB0 0x1ffff

Diagram/Example:

		D16	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
		T15	**T14**	**T13**	**T12**	**T11**	**T10**	**T9**	**T8**	**T7**	**T6**	**T5**	**T4**	**T3**	**T2**	**T1**	**T0**	**Du**
R18	**CD**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R17	**NAQU**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R16	**AQU**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R15	**C15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R14	**C14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R13	**C13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R12	**C12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R11	**C11**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R10	**C10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R9	**C9**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R8	**C8**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R7	**C7**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R6	**C6**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R5	**C5**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R4	**C4**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R3	**C3**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R2	**C2**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R1	**C1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R0	**C0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

R(18 DOWNTO 0) = register of row (use channel bits to access)
D(16 DOWNTO 0)= data bit of row (the data bits written to the registers that control which traces are selected for readout)
C(15 DOWNTO 0) = CFD from channel trace trigger source
NAQU = global don’t acquire trace trigger (RC3) (not anticipated to be used)
AQU = global acquire trace trigger (RC1) (used as global trigger line)
Du = Dummy memory (used for aggregated events with data format). To operate in fast aggregate mode set only R18, D0 so that the dummy register controls the trigger.

In this example a trigger matrix has been configured to ignore NAQU and all channel triggers apart from C0 and C8 which when triggered read out just the ‘dummy’ memory. Additionally the global AQU trigger has been set to cause readout of all memories.

If aggregation is set and multiple triggers are received during the window, at the end of the window the channels selected for readout is the OR of all memories selected by each trigger.

Addressing:

Only one register is used to communicate with the peripheral at GOSSIP address: 0x20002C. The writes to this register are 32 bits with the most significant nibble used to distinguish which sub register the write is for. The byte below this used to select the channel, the number of channels has increased to 18 with NAQ and AQU trigger sources occupying channel 17 and 16 respectively. The least significant bits are used for the data payload which varies between 18 and 1 bit in length depending on the parameter. Right justification is used throughout.

If settings are needed to be read back the same address is written to but with the most significant bit set (E.G Store most significant nibble becomes 1001). To read a write should be undertaken and then a subsequent read of the register 0x20002C will have the correct data payload. (similar to how the SPI read/write is performed (see memory map)). The read data does not contain the channel number only the contents of the register for that channel.

When writing to store trigger matrix row data is right justified. “0000 0000” is the dummy channel trace memory with “0000 0001” CH0 (first physical channel) trace memory, with a total of 17 trace memories. Inside the FEBEX there are a total of 34 trace memories ordered into 17 in buffer 0 and 17 in buffer 1 which are alternatively read and written to. However the application of the memflags to each buffer does not need to be considered by the software.

Parameter	**Data word (32bit, x = don’t care)**
Store trigger matrix row	0001 [chan] [chan] xxx[data] [data] [data] [data] [data]
Read trigger matrix row	1001 [chan] [chan] xxxx xxxx xxxx xxxx xxxx
Store Window length*	0010 xxxx xxxx xxxx xxxx [data] [data] [data] [data]
Read Window length	1010 xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx

*A zero length window has the same result as a window length of 1 but will generate the internal trigger signal one clock cycle faster.

Example:

```
#Trigger matrix setup to only readout first 3 channels from CH0 trigger
#set window to 5 samples
BA=0x20002C #Address of trigger matrix register
echo Trigger Matrix: $BA
gosipcmd -w -x 1 0 $BA 0x10000007 #first 3 chs
#read out trigger matrix
gosipcmd -w -x 1 0 $BA 0x90000000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90100000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90200000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90300000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90400000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90500000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90600000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90700000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90800000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90900000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90A00000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90B00000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90C00000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90D00000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90E00000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x90F00000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x91000000
gosipcmd -r -x 1 0 $BA
gosipcmd -w -x 1 0 $BA 0x92000000
gosipcmd -r -x 1 0 $BA
#Write and readout window
gosipcmd -w -x 1 0 $BA 0x20000005
gosipcmd -w -x 1 0 $BA 0xA0000005
echo Window Setting: $BA
gosipcmd -r -x 1 0 $BA
```