Programming the Helicity Decoder Module

Ed Jastrzembski
FE-DAQ Group
8/17/2021

1 Using the Helicity Decoder Module (*** V5.0 ***)

1.1 Controlling the Module

Communication with the module is by standard VME bus protocols. All registers and memory locations are defined to be 4-byte entities. The VME slave module has two distinct address ranges.

A24 – The base address of this range is set by a 16-element DIP switch on the board. It occupies 256 bytes of VME address space, organized in 64 32-bit words. Relative to the base address, this space is utilized as follows:

00-FF – Register space to control and monitor the module

A32 - The base address of this range is programmed into register ADR32. It occupies 256 Kbytes of VME address space, organized in 64K 32-bit words. A read of any address in this range will yield the next data word from the module. Even though the module is a FIFO, the expanded address range allows the VME master to increment the address during block transfers. This address range can participate in single cycle, 32-bit block, and 64-bit block reads. The only valid write to this address range is the data value 0x80000000 which re-enables the module to generate interrupts (after one has occurred). The address range must be enabled by setting ADR32[0] = 1.

1.2 Module Operation

After a reset of the module (CSR[31] = 1), the system clock source is set (CTRL_1[2..0]). The sync_reset source is set (CTRL_1[6..5]). The BLOCK SIZE register is loaded with the number of events (i.e. triggers) that constitute a *block*. The INTERRUPT register may be loaded with the interrupt ID and level if the module is to initiate an interrupt when the defined *block* of data is available for readout. The address for data access is loaded (ADR32). The event level interrupt (CTRL_1[16] = 1) is enabled if interrupt generation is desired. The BERR response is enabled (CTRL_1[17] = 1) if the module is required to indicate when the complete *block* of data has been read out. Select the trigger source (CTRL_1[4..3]). Enable the decoder function (CTRL_2[0] = 1) and event building function (CTRL_2[2] = 1). A sync_reset signal is generated (CSR[28] = 1) or delivered to the module. Enable the GO bit (CTRL_2[1] = 1) to allow triggers to be accepted by the module.

When the programmed number of triggers has been received, the Block Ready Flag (CSR[3]) will be set and an interrupt will be generated if enabled. The user should initiate a DMA block read (32 or 64-bit) from the address in stored in ADR32. The length of the block read should be programmed to be the expected number of words in the block if

termination by BERR is not selected (CTRL_1[17]). Otherwise, the length of the block read should be chosen larger than the expected size of the data block. In this case the module will terminate the DMA transfer by issuing BERR when all data from the *block* has been transferred. Interrupt generation must be re-enabled by writing 0x80000000 to the address in ADR32.

A trigger accepted by the module will generate an event fragment that consists of **18** data words (32-bits each). Block Header and Trailer words enclose the event data for the block, so the smallest block has **20** words. Current data acquisition software allows for blocks as large as 255 events (4592 words). The 64K word on-board storage (FPGA memory) thus allows for 14 maximum size blocks to be stored on board (3570 events).

1.3 Module Registers

VERSION – board/firmware revision (0x00) **(reg 0)**

[7…0] – (R) – firmware revision

[15…8] – (R) – board revision

[31…16] – (R) – board type (“DEC0”)

CSR – Control/Status (0x04) **(reg 1)**

0 – (R) – System clock PLL locked status (1 = locked)

1 – (R) – Module clock PLL locked status (1 = locked)

2 – (R) – Block of Events Accepted (Event Level Flag)

3 – (R) – Block of Events Ready for readout

4 – (R) – Events on board empty flag status (1 = no events)

5 – (R) – BERR status (1 = module asserted BERR)

6 – (R) – BUSY status (current)

7 – (R) – latched BUSY status (‘1’ means at least one occurrence). Clear latched status by writing ‘1’ to this bit.

8 – (R) – Internal Buffer #0 empty flag

9 – (R) – Internal Buffer #1 empty flag

10 – (R) – Helicity sequence error

11 – (R/W) – TRIGGER TIME WORD ERROR – a mismatch of build event number and trigger number from time word occurred. Clear latched status by writing ‘1’ to this bit.

[12…15] – Spare (read as zero)

16 – (W) – FORCE BLOCK TRAILER INSERTION – will be successful only if there are NO triggers waiting to be processed

17 – (R) – Last FORCE BLOCK TRAILER INSERTION Successful

18 – (R) – Last FORCE BLOCK TRAILER INSERTION Failed

[19…27] – Spare (read as zero)

28 – (W) – Pulse software generated SYNC_RESET (if CTRL_1[7] = 1)

29 – (W) – Pulse software generated TRIGGER (if CTRL_1[7] = 1)

30 – (W) – Pulse soft reset

31 – (W) – Pulse hard reset

CTRL 1 – (0x08) **(reg 2)**

[1…0] – (R/W) – System clock select (0 = P0, 1 = FP 1, 2 = FP 2, 3 = internal) (FP = front panel)

2 – (R/W) – Internal system clock chip enable (0 = OFF, 1 = ON)

[4…3] – (R/W) – TRIGGER source (0 = P0, 1 = FP 1, 2 = FP 2, 3 = soft)

[6…5] – (R/W) – SYNC_RESET source (0 = P0, 1 = FP 1, 2 = FP 2, 3 = soft)

7 – (R/W) – Enable Soft control signals (SYNC_RESET, TRIGGER)

[8…15] – (R/W) – Spare

16 – (R/W) – Enable Interrupt

17 – (R/W) – Enable BERR response

18 – (R/W) – ‘1’ – use internal helicity generator signals for processing
‘0’ – use external helicity input signals for processing

19 – (R/W) – ‘1’ – select copper cable helicity inputs as external inputs used
‘0’ – select fiber optic helicity inputs as external inputs used

20 – (R/W) – ‘1’ – route internal generator signals to helicity front panel outputs
‘0’ – route helicity signals used for processing to front panel outputs

[21…31] – (R/W) – Spare

CTRL 2 – (0x0C) **(reg 3)**

0 – (R/W) – Enable decoder

1 – (R/W) – GO – Enable triggers

2 – (R/W) – Enable event build

[3…7] – (R/W) – Spare

8 – (R/W) – Enable internal helicity generator (set after Helicity CONFIG 1, CONFIG 2, CONFIG 3 are written)

9 – (R/W) – Force BUSY output

[10…31] – (R/W) – Spare

ADR32 – Address for data access (0x10) **(reg 4)**

0 – (R/W) – Enable 32-bit address decoding

1 – 6 – (not used – read as 0)

[15…7] – (R/W) – Base Address for 32-bit addressing mode (8 Mbyte total)

INTERRUPT (0x14) **(reg 5)**

[7…0] – (R/W) – Interrupt ID (vector)

[10…8] – (R/W) – Interrupt Level [2..0]. Valid values = 1,..,7.

11 - 15 – (not used)

[20…16] – (R) – Geographic Address (slot number) in VME64x chassis.

21 – 22 – (not used – read as ‘0’)

23 – (R) – Parity Error in Geographic Address.

24 – 31 – (not used – read as ‘0’)

BLOCK SIZE (0x18) **(reg 6)**

[15…0] - (R/W) – Number of events defining a block

[31…16] – (not used = read – as '0')

TRIGGER LATENCY and DATA DELAY (0x1C) **(reg 7)**
(This register must always be programmed.)

[11…0] – (R/W) – latency value – **must** be non-zero (1 count = 8 ns)

[14…12] – (not used = read – as '0')

15 – (R) – latency configured

[27…16] – (R/W) – data delay value – **must** be non-zero (1 count = 8 ns)

[30…28] – (not used = read – as '0')

31 – (R) – data delay configured

==========

Internal Helicity generator configuration registers: CONFIG 1 and CONFIG 2 must be written before CONFIG 3. (The write to CONFIG 3 loads values into generator.) After writing CONFIG registers enable generator by setting CTRL_2[8] = 1.

HELICITY CONFIG 1 (0x20) (internal generator) **(reg 8)**

[1…0] – (R/W) – Pattern mode: 0 = pair, 1 = quartet, 2 = octet, 3 = toggle

[7…2] – (not used = read – as '0')

[15…8] – (R/W) – Helicity delay in windows

[31…16] – (R/W) – Helicity settle time (1 count = 40 ns)

HELICITY CONFIG 2 (0x24) (internal generator) **(reg 9)**

[27…0] – (R/W) – Helicity stable time (1 count = 40 ns)

[31…28] – (not used = read – as '0')

HELICITY CONFIG 3 (0x28) (internal generator) **(reg 10)**

[29…0] – (R/W) – Initial pseudorandom sequence seed

[31…30] – (not used = read – as '0')

==========

TEST REGISTER (0x2C) **(reg 11)**

[31…0] – (R/W) – test value (used to confirm data integrity to/from module)

TRIGGER 1 SCALER – (0x30) **(reg 12)**

[31…0] – (R) – total event count

TRIGGER 2 SCALER – (0x34) **(reg 13)**

[31…0] – (R) – total TRIGGER 2 count

SYNC_RESET SCALER – (0x38) **(reg 14)**

[31…0] – (R) – total SYNC_RESET count

EVENTS ON BOARD (0x3C) – Number of events currently stored on board **(reg 15)**

[23…0] - (R) – event count[23…0]

[31…24] – (not used – read as '0')

BLOCKS ON BOARD – (0x40) **(reg 16)**

[31…20] – not used

[19…0] – (R) - number of event BLOCKS on board (non-zero → CSR[4] = 1).

HELICITY SCALER 1 – (0x44) **(reg 17)**

[31…0] – (R) – count of rising edge T_SETTLE

HELICITY SCALER 2 – (0x48) **(reg 18)**

[31…0] – (R) – count of falling edge T_SETTLE (latched on read of HELICITY SCALER 1)

HELICITY SCALER 3 – (0x4C) **(reg 19)**

[31…0] – (R) – count of PATTERN_SYNC (latched on read of HELICITY SCALER 1)

HELICITY SCALER 4 – (0x50) **(reg 20)**

[31…0] – (R) – count of PAIR_SYNC (latched on read of HELICITY SCALER 1)

==========

Processing Clock Test Register: The processing clock (125 MHz) that may be derived from an external source is validated. A write of any data to the register initiates a 10.24 us time interval over which the processing clock periods are counted. The time interval is generated by a fixed free-running on-board clock (50 MHz). A subsequent read of the register should yield a value of **1280** counts if the processing clock frequency is correct.

PROCESSING CLOCK TEST REGISTER – (0x54) **(reg 21)**

[31…0] – (R/W) – count of processing clock periods in a10.24 us time interval

==========

RECOVERED SHIFT REGISTER VALUE (0x58) **(reg 22)**

[29…0] – (R/W) – Current recovered value

[31…30] – (not used = read – as ‘0’)

GENERATOR SHIFT REGISTER VALUE (0x5C) (internal generator) **(reg 23)**
(Note: this value is latched when the RECOVERED value is read)

[29…0] – (R/W) – Current internal generator value

[31…30] – (not used = read – as ‘0’)

==========

Delay Confirmation Registers: The two digital delays programmed in the TRIGGER LATENCY and DATA DELAY register (0x1C) are implemented with dual-port memory structures. The difference between the WRITE address (wraddr) and the READ address (rdaddr) is the dynamic measure of delay in counts (1 count = 8 ns). To be precise:

if (wraddr > rdaddr) delay = wraddr – rdaddr
else delay = 4096 + wraddr – rdaddr

The computed delay values should **always** match the values programmed into the TRIGGER LATENCY and DATA DELAY register. A read of the TRIGGER LATENCY CONFIRMATION REGISTER latches data for both confirmation registers and thus should be read first.

TRIGGER LATENCY CONFIRMATION REGISTER (0x60) **(reg 24)**

[11…0] – (R) – Memory READ address

[15…12] – (not used = read – as '0')

[27…16] – (R) – Memory WRITE address

[31…28] – (not used = read – as '0')

DATA DELAY CONFIRMATION REGISTER (0x64) **(reg 25)**

[11…0] – (R) – Memory READ address

[15…12] – (not used = read – as '0')

[27…16] – (R) – Memory WRITE address

[31…28] – (not used = read – as '0')

==========

==========

Helicity History Registers: Bit 0 is the most recent value. A read of REGISTER 1 latches data for all 4 registers and thus should be read first.

HISTORY REGISTER 1 (0x68) **(reg 26)**

[31…0] – (R) – Last 32 windows of PATTERN_SYNC

HISTORY REGISTER 2 (0x6C) **(reg 27)**

[31…0] – (R) – Last 32 windows of PAIR_SYNC

HISTORY REGISTER 3 (0x70) **(reg 28)**

[31…0] – (R) – Last 32 windows of reported HELICITY

HISTORY REGISTER 4 (0x74) **(reg 29)**

[31…0] – (R) – Last 32 values of reported HELICITY at PATTERN_SYNC

==========

SPARE REGISTER (0x78) **(reg 30)**

2 Data Format

2.1 Data Word Categories

Data words from the module are divided into two categories: Data Type Defining (bit 31 = 1) and Data Type Continuation (bit 31 = 0). Data Type Defining words contain a 4-bit data type tag (bits 30 - 27) along with a type dependent data payload (bits 26 - 0). Data Type Continuation words provide additional data payload (bits 30 – 0) for the *last defined data type*. Continuation words permit data payloads to span multiple words and utilize bits 30 - 27 as data. Any number of Data Type Continuation words may follow a Data Type Defining word. The decoder data header type is an exception. It specifies the number of 32-bit data words that follow.

2.2 Data Type List

0 – block header
1 – block trailer
2 – event header
3 – trigger time

8 – decoder data header

14 – data not valid (empty module)
15 – filler (non-data) word

2.3 Data Types

Block Header (0) – Word 1 indicates the beginning of a block of events.
(31) = 1
(30 – 27) = 0
(26 – 22) = slot number (set by VME64x backplane)
(21 – 18) = module ID ('D' for Helicity Decoder)
(17 – 8) = event block number
(7 – 0) = number of events in block

Block Trailer (1) – indicates the end of a block of events.
(31) = 1
(30 – 27) = 1
(26 – 22) = slot number (set by VME64x backplane)
(21 – 0) = total number of words in block of events

Event Header (2) – indicates the start an event.

(31) = 1
(30 – 27) = 2
(26 – 22) = slot number (set by VME64x backplane)
(21 – 12) = trigger time (bits 9 – 0 (see below))
(11 – 0) = trigger number

Trigger Time (3) – time of trigger occurrence relative to the most recent global reset. Time is measured by a 48-bit counter that is clocked by the 125 MHz system clock. The six bytes of the trigger time

$$\text{Time} = T_A\ T_B\ T_C\ T_D\ T_E\ T_F$$

are reported in two words (Type Defining + Type Continuation).

Word 1:

(31) = 1
(30 – 27) = 3
(26 – 24) = T_C bits 2 – 0 (duplicated in Word 2)
(23 – 16) = T_D
(15 – 8) = T_E
(7 – 0) = T_F

Word 2:

(31) = 0
(30 – 24) = reserved (read as 0)
(23 – 16) = T_A
(15 – 8) = T_B
(7 – 0) = T_C

Decoder Data Header (8) – indicates the beginning of a block of Decoder Data words. The number of 32 bit data words that will immediately follow it is provided in the header. The fixed order of the data words is used to identify them (see below). Currently there are 16 words reported for each event.

(31) = 1
(30 – 27) = 8
(26 – 6) = reserved (read as 0)
(5 – 0) = number of decoder data words to follow (14 = current)

Data Not Valid (14) – module has no valid data available for read out.

(31) = 1
(30 – 27) = 14
(26 – 22) = slot number (set by VME64x backplane)
(21 – 0) = undefined

Filler Word (15) – non-data word appended to the block of events. Forces the total number of 32-bit words read out of a module to be a multiple of 2 or 4 when 64-bit VME transfers are used. **This word should be ignored**.

(31) = 1
(30 – 27) = 15
(26 – 22) = slot number (set by VME64x backplane)
(21 – 0) = undefined

2.4 Decoder Data Words

1. Helicity seed
   (31 – 30) = 0
   (29 – 0) = recovered seed

2. Count of rising edge T_STABLE (32 bits)

3. Count of falling edge T_STABLE (32 bits)

4. Count of PATTERN_SYNC (32 bits)

5. Count of PAIR_SYNC (32-bits)

6. Time of trigger since start of T_STABLE (32 bits, 1 count = 8 ns)

7. Time of trigger since end of T_STABLE (32 bits, 1 count = 8 ns)

8. Time duration of last complete T_STABLE period (32 bits, 1 count = 8 ns)

9. Time duration of last complete T_SETTLE period (32 bits, 1 count = 8 ns)

10. Status at trigger time
    (0) – T_SETTLE state
    (1) – PATTERN_SYNC state
    (2) – PAIR_SYNC state
    (3) – HELICITY state
    (4) – HELICITY state at pattern start
    (5) – Event Polarity (XOR [HELICITY, HELICITY at pattern start])
    (7 – 6) – ‘0’
    (11 – 8) – Pattern phase count
    (31 – 12) – ‘0’

11. Last 32 windows of PATTERN_SYNC (32 bits)

12. Last 32 windows of PAIR_SYNC (32 bits)

13. Last 32 windows of HELICITY_STATE (32 bits)

14. Last 32 values of HELICITY_STATE at start of pattern (32 bits)