LWA-SV Memo 2

LWA-SV F-engine control registers

D. Price   
28th January 2016

# Introduction

The LWA-SV digital signal processing system is known as the Advanced Data Processor (ADP). It consists of:

* 32x CASPER ADC16x250-8 digitizer cards (a total of 32x16=512 inputs).
* 16x CASPER ROACH2 FPGA processing boards (Xilinx Virtex-6 SX475T FPGA)
* Mellanox SX1024 10/40GbE switch
* 6x GPU servers (ASUS ESC4000 G3 server)

The digitization, channelization, and channel selection are done on the ROACH2 boards, the firmware of which is written using the CASPER / MATLAB / Simulink / Xilinx ISE toolflow[[1]](#footnote-1).

This memo gives a listing of the user-accessible registers and shared BRAMs that may be used for monitor and control of the LWA-SV F-engine.

Monitor and control is done via the KATCP protocol[[2]](#footnote-2). Python bindings for KATCP are provided by CASPER.

## /Users/dan/Desktop/Screen Shot 2016-01-28 at 12.10.55 PM.pngFirmware overview

The model consists of three top-level blocks:

* **ADC** – this contains the ADC16x250 digitizer yellow block, and reset / synchronization pulse logic.
* **FFT** – this contains the polyphase filterbank implementation for each of 32 inputs, and the post-channelization 4-bit requantization logic.
* **PKT** – this contains 10 GbE Ethernet packetization logic, including channel selection.

## ADC top-level block

|  |  |
| --- | --- |
| ADC control / GPIO registers |  |
| adc\_rst | Provides a reset line for ‘cores’ (i.e. logic blocks), and counter values. The *adc\_rst* line should be set high (e.g. 0b11) and then low (0b00) to trigger a reset. It is important to issue a reset after configuring values  LSB is for core reset, LSB+1 is counter. To turn both on, write a value 0b11 = 3. To reset on core only, write 0b01=1. To reset counters, write 0b10=2 |
| force\_sync | Force a sync pulse to be sent. Should only be used for debugging. |
| adc\_sync\_in | a GPIO connection that is connected to a pulse-per-second signal derived from GPS |
| adc\_sync\_pulse | stores how many timing sync pulses have been sent. |
| adc\_sync\_out | GPIO propagates the sync pulse to the GPIO output (this is not used at LWA-SV). |
| adc\_sync\_count | stores a count of how many PPS have occurred since the last counter reset. |

## FFT top-level block

|  |  |
| --- | --- |
| FFT control BRAM and registers |  |
| fft\_f0\_cg\_bpass\_bram  fft\_f1\_cg\_bpass\_bram  fft\_f2\_cg\_bpass\_bram  fft\_f3\_cg\_bpass\_bram | A 2^13 deep BRAM for storing 32-bit gain coefficients for bandpass correction, used in 4-bit requantization.  The 18-bit real and imag values are treated as Fix18\_17 values, multiplied through by this Ufix32\_17 coefficient, resulting in a Fix50\_34 value. This is then converted to a Fix4\_3 and then rounded to stay within [-7, 7].  The conversion *truncates* bits before the decimal point, and *rounds* after the decimal. So a gain value of 1 means the top bits of the 18\_17 are taken; a value of 8 would shift by 3 bits. Good values are likely to be >> 1. |
| fft\_f0\_fft\_shift fft\_f1\_fft\_shift fft\_f2\_fft\_shift  fft\_f3\_fft\_shift | Set the shift schedule in the FFT. The FFT has 13 stages; after each stage the signal may be bitshifted so as to avoid overflows.  To set an FFT stage to shift, write a binary 1.  For example, shifting every stage, write 0b1111111111111, to shift every second stage 0b1010101010101.  Default is to shift every stage. |

|  |  |
| --- | --- |
| FFT control BRAM and registers |  |
| fft\_f0\_rms\_mon\_4b  fft\_f1\_ rms\_mon\_4b  fft\_f2\_ rms\_mon\_4b fft\_f3\_ rms\_mon\_4b | A 2^12 deep BRAM which is essentially a 4-bit, single integration spectrometer.  This output is provided to help tuning the RMS at the 4-bit requantization level. |

## PKT top-level block

|  |  |
| --- | --- |
| Packetizer control registers |  |
| pkt\_roach\_id | Roach ID number. Should run 1-16, used to identify ROACH |
| pkt\_gbe0n\_chan\_per\_sub | Number of channels per subband for gbe0 packetizer.  Allowable values are 10-144 |
| pkt\_gbe1\_n\_chan\_per\_sub | Number of channels per subband for gbe1 packetizer.  Allowable values are 10-144 |
| pkt\_gbe0\_n\_subband | Number of subbands for gbe0 packetizer. Total number of channels sent will be n\_subband x n\_chan\_per\_sub.  Allowable values are 1-32 |
| pkt\_gbe1\_n\_subband | Number of subbands for gbe1 packetizer. Total number of channels sent will be n\_subband x n\_chan\_per\_sub.  Allowable values are 1-32 |
| pkt\_gbe0\_start\_chan | Start channel (lowest channel in range) for gbe0.  Allowable values 10-4000 |
| pkt\_gbe0\_start\_chan | Start channel (lowest channel in range) for gbe1.  Allowable values 10-4000 |
| pkt\_gbe0\_stop\_chan | Stop channel (highest channel in range) for gbe0.  Allowable values 20-4095 |
| pkt\_gbe1\_stop\_chan | Stop channel (highest channel in range) for gbe1.  Allowable values 20-4095 |
| pkt\_tx\_enable | Enable data flow. Controls data flow for both gbe0 and gbe1. LSB is for gbe0, LSB+1 is gbe1. To turn both on, write a value 0b11 = 3. To turn on gbe0 write 0b01=1 or for gbe1 0b10=2 |
| Packetizer control BRAMS |  |
| pkt\_gbe0\_ip\_addr\_bram pkt\_gbe1\_ip\_addr\_bram | List of IP address to send to. This should be the same length as the n\_subbands; each subband is sent to a corresponding IP in this list. |
| pkt\_gbe0\_ip\_port\_bram pkt\_gbe1\_ip\_port\_bram | List of IP ports to send to. This should be the same length as the ip\_addr\_bram. |

|  |  |
| --- | --- |
| Packetizer output registers |  |
| pkt\_gbe0\_oflow\_cnt pkt\_gbe1\_oflow\_cnt | This register will be >0 if there’s overflows happening in the 10GbE core. This probably means that your n\_subbands \* n\_chan\_per\_sub is too high. Only used in debugging. |
| pkt\_gbe0\_eof\_cnt pkt\_gbe1\_oeof\_cnt | Count of how many end of frame (EOF) have passed. More simply, the number of packets send out over the 10GbE link. Only used in debugging. |
| pkt\_fifo\_pc\_full | A fractional number showing what percent of the FIFO is full. Only used in debugging. |
| pkt\_gbe0\_linkup pkt\_gbe1\_linkup | A register that shows if the 10GbE core is configured and the link is up. Returns 1 if up, 0 if down. |
| pkt\_gbe0\_full pkt\_gbe1\_full | Returns 1 if the gbe0 or gbe1 blocks are filled. |

1. https://casper.berkeley.edu/wiki/Main\_Page [↑](#footnote-ref-1)
2. https://casper.berkeley.edu/wiki/KATCP [↑](#footnote-ref-2)