VEGAS I-BANDxCHAN mode (BAND = 0800 MHz/ 1500 MHz and CHAN=0064 to 8192)

The latest firmware modules for VEGAS have been developed using Xilinx 14.7/MATLAB R2012b. The designs have been built for operations compatible with ROACH-2.

The VEGAS firmware building blocks, in the order of signal flow in ROACH-2, are as follows:

1) **ASIAA-ADCS**: Two interleaved ADCs, each clocked at 'BAND' MHz for the two polarizations P0 and P1, results in the input signal being digitally sampled at an effective rate of '2xBAND' MHz, satisfying the Nyquist sampling criterion. The resulting logic frequency of the Fabric is clocked at BAND/8 MHz. Each clock cycle yeilds 32 (16P0 + 16P1) samples, each 8 bits wide, in the offset-binary format. The signals are then converted to processible signed 2's-compliment format. In order to monitor the ADC outputs every clock cycle, the 16P0 and 16P1 8-bit streams are concatenated to form 128-bits P0 and P1 outputs from ADC0 and ADC1, respectively, and are logged using adcsnap0 and adcsnap1 blocks.

The ADCs also provide the framework with a 1 PPS, which is used along with a user-controlled ARM signal, to generate a Master-SYNC pulse, to synchronize the functioning of all the blocks in the design.

2) **Bus-expand**: The 128-bit P0 and P1 concatenated outputs from the ADC blocks are then split in order to re-obtain the 16P0 and 16P1 8-bit 2's compliment signals, to pass them on to further signal processing sub-systems.

3) **Polyphase Filter Bank**: The signal is then subjected to Fast-Fourier Transform (FFT) using the polyphase filterbank technique, instead of just a conventional Fourier transform operation, to alleviate leakage and scalloping losses. This is done by adding a PFB block, where initial signal conditioning takes place, before the FFT block, where the signal spectrum is generated for further processing/analysis stages. More information on this technique can be found in https://casper.berkeley.edu/wiki/The_Polyphase_Filter_Bank_Technique.

The 12-tap PFB block is set for 2*CHAN-points, and employs a Hamming window function, and a bin width scaling factor set to 0.95. The resulting output signal streams, each 14-bit wide now, are then fed to a 16-input 2*CHAN-point FFT blocks (FFT0 and FFT1), one each for the two polarizations. There are 8 outputs coming out of the FFT block, each 28-bit wide (14Real + 14Imaginary). It is to be noted that 8 channels of the total spectrum are being output, every clock cycle. This implies that it takes CHAN/8 clock cycles for one set of the entire spectra to be generated. A single FFT_SYNC pulse (also output by the FFT block) marks the start of the first spectra being output. Subsequent spectra generation occur continuously thereafter.

4) **STOKES Sub-system (STOKES_SS):** There are 8 STOKES SS blocks, to evaluate the Stokes parameters I, Q, U and V. Each of the 8 STOKES SS takes i'th of the 8 outputs from FFT0 and FFT1, every clock cycle, and generates I_i, Q_i, U_i and V_i, where i=0 to 7. Each of the outputs are now 29 bits wide.

5) **Scale-offset block**: The outputs of the STOKES_SS go into 4 Scale_Offset blocks, 1 each for one of the 4 Stokes parameters, where signal is conditioned using user-defined offset and scaling factor. For instance, Stokes_Offset_I takes in I_i (i = 0 to 7), along with user defined OFFSET_I and SCALE_I, as inputs. In the first step, I_i is "offset" using the input parameter, and is then multiplied by the scaling factor input. This results in 8 outputs every clock cycle, I_i_IN, each 32 bits wide. Similar operation and initialization takes place for STOKES_SS_Q/U/V.

6) **Vector Accumulator (Vacc)**: In this design, we have 4 Vacc blocks, one for each of the Stoke's parameters, and represented as x_Vacc, where x is I, Q, U and V. There are several functionalities offered the the Vacc blocks. Firstly, it accumulates the incoming vector values, whose length is specified by the user in ACC_LENGTH SW register. It is to be noted that the the value specified by the user must be one less than power of 2. Secondly, the Vacc blocks averages the accumulation over the accumulation period, and store them in ODD and EVEN Shared BRAM blocks for future readout by the Data Handling subsystem (DH-SS). Thirdly, it provides flags/valid signals when the BRAMs are full (x_BRF), and when valid data is accumulated (x_VALID), to let the succeeding subsytems know when to begin read operation from the BRAM. Therefore, each x_Vacc block has four output ports – x_EVEN_OUT, x_ODD_OUT, x_BRF and x_VALID.

7) **BRAM_FULL**: This block reads the 4 x_BRF signals from the x_Vacc outputs, and creates a single composite signal, to notify the DH_SS that the BRAMs are ready for readout.

8) **VACCs_VALID**: This block generates a single composite VALID signal, when all the x_VALID's output from x_Vacc are simultaneously true.

9) **DH_SS**: The DH_SS subsystem is primarily responsible for generating transmit packets for the 10Ge blocks, in the correct packet format. It starts its operation by first monitoring the BRAM_FULL and VALID signals to ensure readout at the appropriate time instance. Once the data is read, the Packer sub-system in the DH_SS forms packets for transmission, in the correct format. The packets for transmission are generated in the following format:

a) Frame count (FC): Appends an 8 Byte FC, with the 4 Bytes containing FC justified in the LSBs.
b) Data Payload: The data payload is formed by stacking the Stoke's parameters I_n, Q_n, U_n and V_n (n=0 to CHAN-1), in the following manner:

(1)

I_0, I_1, I_2……., I_7
.
.
……., I_(CHAN-1)
Q_0, Q_1, Q_2……., Q_7
.
.
……., I_(CHAN-1)

U_0, U_1, U_2……., U_7
.
.
……., U_(CHAN-1)

V_0, V_1, V_2……., V_7
.
.
……., V_(CHAN-1)

(2)

I_0, I_1, I_2……., I_7
.
.
……., I_(CHAN-1)
Q_0, Q_1, Q_2……., Q_7
.
.
……., I_(CHAN-1)

U_0, U_1, U_2……., U_7
.
.
……., U_(CHAN-1)

V_0, V_1, V_2……., V_7
.
.
……., V_(CHAN-1)

.

.
.
(8192/(4*CHAN))
I_0, I_1, I_2……., I_7
.
.
……., I_(CHAN-1)
Q_0, Q_1, Q_2……., Q_7
.
.
……., I_(CHAN-1)

U_0, U_1, U_2……., U_7
.
.
……., U_(CHAN-1)

V_0, V_1, V_2……., V_7
.
.
……., V_(CHAN-1)
Total size of the data payload is 8192 Bytes.

c) Frame Error Count (FEC): Appends an all 0's FEC, which is 8 Bytes wide.
d) Inter-Frame Gaps (IG0 and IG1): Two interframe gaps IG0 and IG1, each 8 Bytes, at the end of each packet (alternating 1's and 0's).

The DH_SS also generates STB and TX_EOF signals to control 10Ge transmit, and notify the end-of-frame, respectively. DH_SS also generates a TGE_SYNC pulse, synchronous with successful packet created. These three signals are used to generate a "runt packet" and force 10Ge block to transmit it, in order to clean up any garbage data from its buffers, for first legitimate packet transmission.

10) **10Ge block**: This block is responsible for transmitting the UDP packets, formed by wrapping data from the transmit buffer, to the user specified PORT and I/P Address of the destination.

User-specified Variables: Here is a list of variables that the user can initialize, to assign/control the functioning of a few blocks. These variables are realized using SW registers in the actual hardware implementation:

1) **arm** : This variable is used to generate a one clock-cycle wide Master-SYNC pulse, coincident with the rising edge of the first 1PPS signal. A complete 'arm' command is performed by first writing a value of 1, followed by a 0.

2) **fftshift**: This variable determines if the stage outputs in the FFT block have to be divided by 2 (right shift). This is done on a stage-by-stage basis. Usually the value of 'fftshift' is set to AAAAAAAA(hex) for observations with weak signals and FFFFFFFF (hex) for strongly correlated signals to avoid overflows.

3**) dest_ip**: Set to the destination IP address.

4) **dest_port**: Set to the destination port address.

5) **acc_len**: Sets the accumulator lengths for the x_Vacc subsystem. The value specified by the user must be one less than power of 2 (eg. 7, 15, etc.)

6) **offset_i/offset_q/offset_u/offset_v**: Used to set the offsets for I/Q/U/V components of the Stoke's parameters, in the Scale_Offset subsystem. The input values have a Signed 32_16 data-type.

7) **scale_i/scale_q/scale_u/scale_v**: Used to set the scaling factors for the I/Q/U/V components of the Stoke's parameters, in the Scale_Offset subsystem. The input values have an Unsigned 32_16 data-type

8) **reset**: This variable is asserted if the 10GE transmit buffer enters to an overflow condition (This can be monitored using the TXOF_STATUS SW register). The assignment is similar to 'arm' (write a value of 1 and then 0).

9) **trig**: This variable is asserted to trigger capture in the ADC snapblocks. The assignment is similar to 'arm' (write a value of 1 and then 0).

10) **n_chan**: This variable is initialized according to the number of channels in the design. Note that this variable is *not* to be initialized with the actual number of channels. Instead, it should be assigned with the value "k", where 2^k = CHAN. This variable is essential for the Switching Signal Subsystem.

11) **ssg_length**: This is used to denote the length of the switching signal (in terms of the number of spectral periods).

Real time monitoring subsystems: The following subsystems either allow real-time capture of signals within the ROACH-2, or provide us with Status signals to monitor the current state of the system. These sub-system can also be used for diagnostics and debugging of the firmware.

1) **adcsnap0/adcsnap1**: These snap blocks are used to capture 1024 concatenated 128-bit P0 and P1 ADC samples from the ASIAA_ADCs. This implies a capture of 16384 8-bit raw ADC samples from each of the polarizations. The capture begins once the user 'triggers' the blocks, by asserting the 'trig' SW register.

2) **Switching_Signal subsystem (SSB)**: This system returns the user with the current state the system is in, by either reading out of the SW register 'status_out', or monitoring the LED outputs of the ROACH (4:7).

Updates:

[1] DIBAS based raw ADC data-capture subsystem discontinued in future builds. Appropriate sections modified.
-- ASG (07/23/2015)

[2] Updated SW register names based on the new VPM Firmware-Manager instance naming convention.
-- ASG (07/23/2015)

[3] Updated Xilinx version to 14.7 .
-- ASG (07/23/2015)