Programing Switching signal generator for VEGAS modes

D. Anish Roshi[1]
[1] National Radio Astronomy Observatory, Green Bank & Charlottesville

Dec 5, 2013
Version 1.0

A block diagram of the FPGA implementation of the Switching signal generator (SSG) developed by Srikanth, B. and Randy, M. is shown in Fig. 1. The blank signal (LOC_BLANK) and switching state signals (CAL: noise cal, SR0: LSB of sig/ref, SR1: MSB of sig/ref, ASR: Advanced sig/ref) form the 5 LSB bits of the 32 bit wide BRAM. The bit ordering is B0: BLANK, B1: CAL, B2: SR0, B3 SR1 and B4 ASR. The higher 27 bits are used to determine the duration over which the states of B0 to B4 to be present at the output of the BRAM – ie the higher bits determines how fast the address of BRAM is incremented. The address increment is controlled by *spec_tick*. Programing the SSG requires the duration of blank and state signals computed in units of *spec_tick* and loaded into the BRAM along with the state of bits B0 to B4. The value of the higher 27 bits of the BRAM will be loaded to the register *duration* whenever a location of the BRAM is selected. The total duration of both blank and state signals, which is essentially the sum of the numbers loaded in the BRAM as *duration* (ie is the higher 27 bits), has to be loaded in the register *ssg_length*. As described below there are restrictions for the minimum duration for blank and state signals due to the system design for different VEGAS modes and **therefore the minimum duration for these signals is not determined by the period of** *spec_tick* alone (see Table 1). *spec_tick* is a mode dependent parameter and is listed in Table 1.

Given below are the equations to compute the duration of blank and state signals. Definitions of terms used in this report are also listed below.

- **Definitions :**

  f_{valon} : Frequency of the valon synthesizer in Hz. The clock for the FPGA is generated using the valon synthesizer. Its dependence on VEGAS modes is listed in Table 1.

  Nchan : Number of spectral channels for each mode, listed in Table 1.

  f_{esamp} : The effective sampling frequency in Hz, listed in Table 1.

  spec_tick : Minimum duration in sec of blank and state signals due to the design of the SSG. Its relation to the valon frequency is listed in Table 1.

  t_{blank} : User requested blank period in sec.

  t_{state} : User requested state period in sec. The blank period needs to be subtracted from t_{state} to get the available state period.

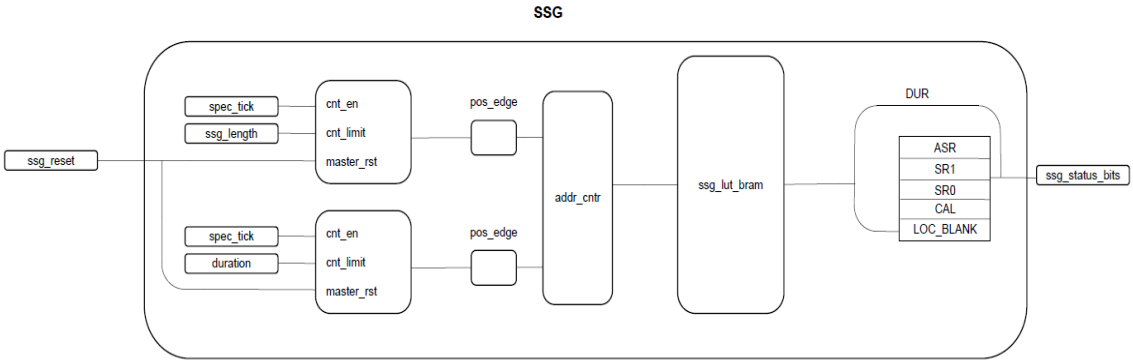

Figure 1: Block diagram of Switching signal generator developed by Srikanth, B. & Randy, M.

$t_{state-blank} = t_{state}$ - t_{blank} : The available state period in sec.

HWEXPOSR : Minimum integration supported by each mode in sec. These values are listed in the VEGAS mode table (www.gb.nrao.edu/vegas/modes) as the minimum integration time.

Blank_duration : Duration of blank signal in units of *spec_tick*

State_duration : Duration of state signal in units of *spec_tick*

CHAN_BW = f_{esamp}/2/Nchan : is the actual spectral resolution for each mode.

- **Equations to compute the actual Blank and State signal duration :**

Blank_duration = int(ceil(t_{blank}/HWEXPOSR) × HWEXPOSR/*spec_tick*); for mode 1 to 3
Blank_duration = int(ceil(t_{blank}× CHAN_BW) × 1/(*spec_tick* × CHAN_BW)) ; for mode 4 to 29

if $t_{blank} > 0$ and Blank_duration = 0 then set Blank_duration = 1, which forces one HWEXPOSR period as the blank time for mode 1 to 3 and at least one 1/CHAN_BW period for mode 4 to 29. For mode 4 to 29, since *spec_tick* is not aligned with FFT boundaries in the GPU, the blank time introduced by the hardware can be up to Blank_duration × 1/CHAN_BW + 1/CHAN_BW.

State_duration = int(ceil($t_{state-blank}$/HWEXPOSR)×HWEXPOSR/*spec_tick*); for all modes

Revision History

1. Dec 5, 2013 – Updated the values for HWEXPOSR for modes 2 and 3

Table 1: **List of parameters for programing SSG and their minimum value**

Mode Name	Valon freq (f_{valon}) (MHz)	Effective Samp freq (f_{esamp})	Nchan	*spec_tick*	HWEXPOSR (msec)	Min Blank period	Min Phase period
Single Sub-band mode							
Mode 1 (H1K/HBW)	1500	2xf_{valon}	1024	Nchan/f_{valon}	0.5[a]	HWEXPOSR	HWEXPOSR
Mode 2 (H16K/HBW)	1500	2xf_{valon}	16384	Nchan/f_{valon}	1.4[a]	HWEXPOSR	HWEXPOSR
Mode 3 (H16K/HBW)	1000	2xf_{valon}	16384	Nchan/f_{valon}	2.0[a]	HWEXPOSR	HWEXPOSR
Mode 4 (L1/LBW1)	1500	$f_{valon}/4$	32768	$1024/f_{valon}$	10	1/CHAN_BW	HWEXPOSR
Mode 5 (L1/LBW1)	1500	$f_{valon}/4$	65536	$1024/f_{valon}$	20	1/CHAN_BW	HWEXPOSR
Mode 6 (L1/LBW1)	1500	$f_{valon}/4$	131072	$1024/f_{valon}$	30	1/CHAN_BW	HWEXPOSR
Mode 7 (L1/LBW1)	800	$f_{valon}/4$	32768	$1024/f_{valon}$	10	1/CHAN_BW	HWEXPOSR
Mode 8 (L1/LBW1)	800	$f_{valon}/4$	65536	$1024/f_{valon}$	20	1/CHAN_BW	HWEXPOSR
Mode 9 (L1/LBW1)	800	$f_{valon}/4$	131072	$1024/f_{valon}$	30	1/CHAN_BW	HWEXPOSR
Mode 10 (L8/LBW1)	1500	$f_{valon}/32$	32768	$1024/f_{valon}$	5	1/CHAN_BW	HWEXPOSR
Mode 11 (L8/LBW1)	1500	$f_{valon}/32$	65536	$1024/f_{valon}$	10	1/CHAN_BW	HWEXPOSR
Mode 12 (L8/LBW1)	1500	$f_{valon}/32$	131072	$1024/f_{valon}$	30	1/CHAN_BW	HWEXPOSR
Mode 13 (L8/LBW1)	1500	$f_{valon}/32$	262144	$1024/f_{valon}$	40	1/CHAN_BW	HWEXPOSR
Mode 14 (L8/LBW1)	1500	$f_{valon}/32$	524288	$1024/f_{valon}$	75	1/CHAN_BW	HWEXPOSR
Mode 15 (L8/LBW1)	750	$f_{valon}/32$	32768	$1024/f_{valon}$	5	1/CHAN_BW	HWEXPOSR
Mode 16 (L8/LBW1)	750	$f_{valon}/32$	65536	$1024/f_{valon}$	10	1/CHAN_BW	HWEXPOSR
Mode 17 (L8/LBW1)	750	$f_{valon}/32$	131072	$1024/f_{valon}$	30	1/CHAN_BW	HWEXPOSR
Mode 18 (L8/LBW1)	750	$f_{valon}/32$	262144	$1024/f_{valon}$	40	1/CHAN_BW	HWEXPOSR
Mode 19 (L8/LBW1)	750	$f_{valon}/32$	524288	$1024/f_{valon}$	75	1/CHAN_BW	HWEXPOSR
8 Sub-band modes							
Mode 20 (L8/LBW8)	1500	$f_{valon}/32$	4096	$1024/f_{valon}$	5	1/CHAN_BW	HWEXPOSR
Mode 21 (L8/LBW8)	1500	$f_{valon}/32$	8192	$1024/f_{valon}$	10	1/CHAN_BW	HWEXPOSR
Mode 22 (L8/LBW8)	1500	$f_{valon}/32$	16384	$1024/f_{valon}$	30	1/CHAN_BW	HWEXPOSR
Mode 23 (L8/LBW8)	1500	$f_{valon}/32$	32768	$1024/f_{valon}$	40	1/CHAN_BW	HWEXPOSR
Mode 24 (L8/LBW8)	1500	$f_{valon}/32$	65536	$1024/f_{valon}$	75	1/CHAN_BW	HWEXPOSR
Mode 25 (L8/LBW8)	1000	$f_{valon}/32$	4096	$1024/f_{valon}$	7	1/CHAN_BW	HWEXPOSR
Mode 26 (L8/LBW8)	1000	$f_{valon}/32$	8192	$1024/f_{valon}$	14	1/CHAN_BW	HWEXPOSR
Mode 27 (L8/LBW8)	1000	$f_{valon}/32$	16384	$1024/f_{valon}$	40	1/CHAN_BW	HWEXPOSR
Mode 28 (L8/LBW8)	1000	$f_{valon}/32$	32768	$1024/f_{valon}$	55	1/CHAN_BW	HWEXPOSR
Mode 29 (L8/LBW8)	1000	$f_{valon}/32$	65536	$1024/f_{valon}$	100	1/CHAN_BW	HWEXPOSR

[a] For Modes 1 to 3, the HWEXPOSR is determined by the VEGAS manager parameter *acc_len*. For these modes HWEXPOSR = *acc_len* × *spec_tick*.