

Subband Statistics Module

	Organisatie / Organization	Datum / Date
Auteur(s) / Author(s): Harm Jan Pepping	ASTRON	1 August 2013
Controle / Checked: Eric Kooistra	ASTRON	
Goedkeuring / Approval: Andre Gunst	ASTRON	
Autorisatie / Authorisation: Handtekening / Signature Andre Gunst	ASTRON	

© ASTRON 2011
All rights are reserved. Reproduction in whole or in part is prohibited without written consent of the copyright owner.

UniBoard

Doc.nr.: ASTRON-RP-1397
Rev.: 0.1
Date: 01-08-2013
Class.: Public

Distribution list:

Group:	Others:
Andre Gunst Eric Kooistra Daniel van der Schuur	Gijs Schoonderbeek Sjouke Zwier Harro Verkouter (JIVE) Jonathan Hargreaves (JIVE) Salvatore Pirruccio (JIVE)

Document history:

Revision	Date	Author	Modification / Change
0.1	2013-08-01	Harm Jan Pepping	Creation
0.2	2013-09-11	Harm Jan Pepping	Updated verification (Chapter 5)

Table of contents:

1	Introduction.....	5
1.1	Mode of operation.....	5
1.1.1	Auto-correlation mode	5
1.1.2	Cross-correlation mode	5
2	Firmware interface.....	6
2.1	Clock domains	6
2.2	Parameters	6
2.3	Interface signals	6
3	Software interface	7
3.1	Register span.....	7
3.2	RAM span	7
4	Implementation.....	8
5	Verification.....	8
6	Validation.....	8
7	Appendix – list of files.....	9
7.1	Firmware VHDL	9
7.2	Testbench	9

Terminology:

EOP	Start Of Packet
MM	Memory-Mapped
MOSI	Master Out Slave In
RAM	Random Access Memory
Signal Path	Time series signal
SISO	Source In Sink Out
SOP	Start Of Packet
SOSI	Source Out Sink In
Subband	Frequency signal

References:

1. \$UNB/Firmware/modules/Lofar/st/

1 Introduction

The statistics module (st module) is a streaming interface component that can calculate the auto-correlation and cross-correlation products of an incoming data stream. This data stream might contain subband data, beamlet data or any other type of data. When used for auto-correlations the st module is predominately used for monitoring purposes. In case cross-correlations are created in combination with a beamformer algorithm it can be used for calibration purposes as well.

1.1 Mode of operation

The st module can operate in two modes: auto-correlation mode and cross-correlation mode. Default the auto-correlation is enabled. The cross-correlation mode is enabled when a non-zero value is written to the threshold register.

1.1.1 Auto-correlation mode

In auto-correlation mode each incoming complex data value in a frame is multiplied with its own conjugated variant. The results are integrated over a number of frames by an accumulator. The accumulator stores the intermediate results in ram (this ram holds frame_size intermediate results). The integration period is marked by the sync signal. Once a sync signal is received the integrated data is copied to a dual port memory from where it is accessible for the user. Consider an input stream where the frame_size is 8 and the sync interval is set to every 4 frames. Where in $A_{x,y}$: x=sample number in frame, y = frame number

Frame 0: $A_{0,0}A_{1,0}A_{2,0}A_{3,0}A_{4,0}A_{5,0}A_{6,0}A_{7,0}$
 Frame 1: $A_{0,1}A_{1,1}A_{2,1}A_{3,1}A_{4,1}A_{5,1}A_{6,1}A_{7,1}$
 Frame 2: $A_{0,2}A_{1,2}A_{2,2}A_{3,2}A_{4,2}A_{5,2}A_{6,2}A_{7,2}$
 Frame 3: $A_{0,3}A_{1,3}A_{2,3}A_{3,3}A_{4,3}A_{5,3}A_{6,3}A_{7,3}$

These frames applied to the st module will result in 8 statistic values that are defined as follows:

Statistic 0: $A_{0,0} A_{0,0}^* + A_{0,1} A_{0,1}^* + A_{0,2} A_{0,2}^* + A_{0,3} A_{0,3}^*$
 Statistic 1: $A_{1,0} A_{1,0}^* + A_{1,1} A_{1,1}^* + A_{1,2} A_{1,2}^* + A_{1,3} A_{1,3}^*$
 Statistic 2: $A_{2,0} A_{2,0}^* + A_{2,1} A_{2,1}^* + A_{2,2} A_{2,2}^* + A_{2,3} A_{2,3}^*$
 Statistic 3: $A_{3,0} A_{3,0}^* + A_{3,1} A_{3,1}^* + A_{3,2} A_{3,2}^* + A_{3,3} A_{3,3}^*$
 Statistic 4: $A_{4,0} A_{4,0}^* + A_{4,1} A_{4,1}^* + A_{4,2} A_{4,2}^* + A_{4,3} A_{4,3}^*$
 Statistic 5: $A_{5,0} A_{5,0}^* + A_{5,1} A_{5,1}^* + A_{5,2} A_{5,2}^* + A_{5,3} A_{5,3}^*$
 Statistic 6: $A_{6,0} A_{6,0}^* + A_{6,1} A_{6,1}^* + A_{6,2} A_{6,2}^* + A_{6,3} A_{6,3}^*$
 Statistic 7: $A_{7,0} A_{7,0}^* + A_{7,1} A_{7,1}^* + A_{7,2} A_{7,2}^* + A_{7,3} A_{7,3}^*$

1.1.2 Cross-correlation mode

In cross-correlation mode the first complex data value from a frame will first be multiplied with its own conjugated variant. After that it will be registered. This registered value will then be multiplied with the conjugated variant of the following complex values in the frame. The registered sample will be updated every threshold sample and at the start of a new frame. Using the same input frames from the auto-correlation example and a threshold value of 3 will lead to the following 8 statistic values:

Statistic 0: $A_{0,0} A_{0,0}^* + A_{0,1} A_{0,1}^* + A_{0,2} A_{0,2}^* + A_{0,3} A_{0,3}^*$
 Statistic 1: $A_{0,0} A_{1,0}^* + A_{0,1} A_{1,1}^* + A_{0,2} A_{1,2}^* + A_{0,3} A_{1,3}^*$
 Statistic 2: $A_{0,0} A_{2,0}^* + A_{0,1} A_{2,1}^* + A_{0,2} A_{2,2}^* + A_{0,3} A_{2,3}^*$
 Statistic 3: $A_{3,0} A_{3,0}^* + A_{3,1} A_{3,1}^* + A_{3,2} A_{3,2}^* + A_{3,3} A_{3,3}^*$
 Statistic 4: $A_{3,0} A_{4,0}^* + A_{3,1} A_{4,1}^* + A_{3,2} A_{4,2}^* + A_{3,3} A_{4,3}^*$
 Statistic 5: $A_{3,0} A_{5,0}^* + A_{3,1} A_{5,1}^* + A_{3,2} A_{5,2}^* + A_{3,3} A_{5,3}^*$
 Statistic 6: $A_{6,0} A_{6,0}^* + A_{6,1} A_{6,1}^* + A_{6,2} A_{6,2}^* + A_{6,3} A_{6,3}^*$
 Statistic 7: $A_{6,0} A_{7,0}^* + A_{6,1} A_{7,1}^* + A_{6,2} A_{7,2}^* + A_{6,3} A_{7,3}^*$

2 Firmware interface

This chapter covers all firmware interface related topics of the st module. It describes the functionality of the in- and output ports and the parameters that configure the st module.

2.1 Clock domains

There are two clock domains used in the st module: the mm_clk and the dp_clk domain.

2.2 Parameters

The parameters that define an instantiation of the st module are listed in Table 1.

Generic	Type	Value	Description
nof_stat	NATURAL	512	Specifies the number of statistics to be made. It basically indicates the framesize of the incoming datastream.
xst_enable	BOOLEAN	FALSE	When set to TRUE, an extra memory is instantiated to hold the imaginary part for the cross-correlation results. When set to FALSE the results are considered non-complex.
in_data_w	NATURAL	18	Specifies the width on the incoming data.
stat_data_w	NATURAL	56	Width of the output of the st module. This value must be high enough to accommodate the highest possible bitgrowth in the st module. Highest integration period is set to one second.
stat_data_sz	NATURAL	2	This number specifies how many 32-bit registers are required to read out one accumulated value.

Table 1: bf parameters

2.3 Interface signals

The interface signals of the st_sst module are shown in Figure 1 and Table 2 lists the general specifications of these interfaces.

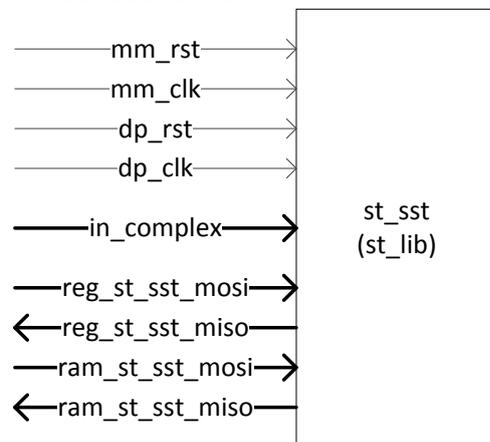


Figure 1: interface signals

Interface	Type	Size or Span	Description
in_complex	t_dp_sosi	na	Input streams that holds the data for statistical processing.
reg_st_sst_mosi	t_mem_mosi	1	Mosi interface to the threshold register
reg_st_sst_miso	t_mem_miso	1	Miso interface to the threshold register

ram_st_sst_mosi	t_mem_mosi	nof_stat * stat_data_sz * 2	A mosi interface to read out the statistics data.
ram_st_sst_miso	t_mem_miso	nof_stat * stat_data_sz * 2	A miso interface to read out the statistics data.
dp_clk	std_logic	na	Datapath clock
dp_rst	std_logic	na	Datapath reset
mm_clk	std_logic	na	Memory mapped interface clock
mm_rst	std_logic	na	Memory mapped interface reset

Table 2: interface signals

3 Software interface

This chapter describes the software interface for the st module.

3.1 Register span

The register span of the st module consists of a single register

Name	Address (words)	Size (bits)	Read/Write	Description
threshold	0x0	32	r/w	Register that holds the threshold value for the cross-correlation mode. When set to 0x0 the st module works in auto-correlation mode.

Table 3 statistics REG span

3.2 RAM span

Table 4 shows the register span for the st module with the following parameters: nof_stat=512, xst_enable=TRUE, in_data_w=18, stat_data_w=56, stat_data_sz=2. Note that not all registers are listed in the table in order to save space.

Name	Address (words)	Size (bits)	Read/Write	Description
stat_0_real_lsb	0x0	32	r/w	Least significant part of the real part of statistic 0
stat_0_real_msb	0x1	24	r/w	Most significant part of the real part of statistic 0
stat_1_real_lsb	0x2	32	r/w	Least significant part of the real part of statistic 1
stat_1_real_msb	0x3	24	r/w	Most significant part of the real part of statistic 1
-----	-----	---	----	-----
stat_510_real_lsb	0x3FC	32	r/w	Least significant part of the real part of statistic 510
stat_510_real_msb	0x3FD	24	r/w	Most significant part of the real part of statistic 510
stat_511_real_lsb	0x3FE	32	r/w	Least significant part of the real part of statistic 511
stat_511_real_msb	0x3FF	24	r/w	Most significant part of the real part of statistic 511
stat_0_imag_lsb	0x400	32	r/w	Least significant part of the imag part of statistic 0
stat_0_imag_msb	0x401	24	r/w	Most significant part of the imag part of statistic 0
stat_1_imag_lsb	0x402	32	r/w	Least significant part of the imag part of statistic 1
stat_1_imag_msb	0x403	24	r/w	Most significant part of the imag part of statistic 1
-----	-----	---	----	-----
stat_510_imag_lsb	0x7FC	32	r/w	Least significant part of the imag part of statistic 510
stat_510_imag_msb	0x7FD	24	r/w	Most significant part of the imag part of statistic 510
stat_511_imag_lsb	0x7FE	32	r/w	Least significant part of the imag part of statistic 511
stat_511_imag_msb	0x7FF	24	r/w	Most significant part of the imag part of statistic 511

Table 4 statistics RAM span

4 Implementation

This chapter gives a brief description of the implementation of the st module. The st module was originally designed in the Lofar project where only the auto-correlation was supported. The cross-correlation mode is added later. The original st module consisted only of the st_calc unit and the dualport_ram. The cross-correlation mode extension is displayed in Figure 2. The value in the treshold register defines the counters' maximum value. When the output of the counter is equal to zero input register a is updated.

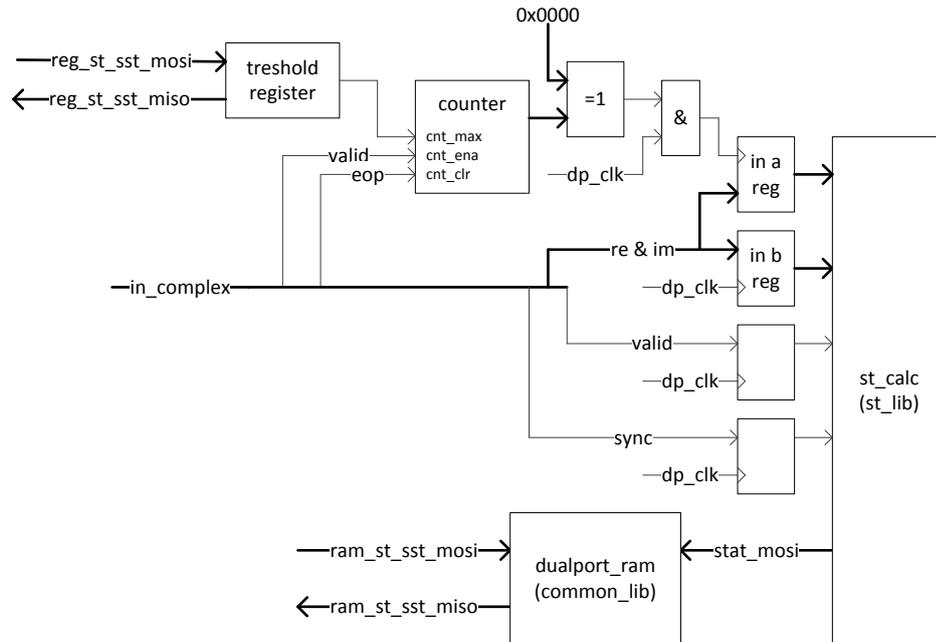


Figure 2 Implementation of the cross-correlation mode

5 Verification

Verification of the st module is based on co-simulation between Modelsim and python scripting. The VHDL testbench can be found in [1] and is called `tb_mmf_st_sst.vhd`. The accompanying python test case can be found in [1] and is called `tc_sst.py`. The simulation is started with the auto simulation file `st_sst.py` that is also available in [1]. The script should be called as follows:

```
python st_sst.py --hold
```

The calculated statistics are automatically verified by the script.

6 Validation

Validation of the st module is performed in the Apertif beamformer design. In both the `bn_filterbank` image and the `fn_beamformer` image the st module is instantiated.

7 Appendix – list of files

7.1 Firmware VHDL

All VHDL source files that are used for the st module can be found in the following directory:

\$UNB/Firmware/modules/Lofar/st/src//vhdl

The next table gives an overview of the VHDL source files:

VHDL File	Description
st_acc.vhd	This file contains the accumulator algorithm.
st_calc.vhd	Multiplication and accumulation is done in here.
st_ctrl.vhd	This file contains the controller for the complete statistics algorithm.
st_sst.vhd	Top level entity.

7.2 Testbench

The testbench files for simulation are in the following directory:

\$UNB/Firmware/modules/Lofar/st/tb/