A 256-channel (element) Correlator Design Based on an FPGA for X-ray Photon Correlation Spectroscopy

Li, Zhi Yong, Durga Misra, D. Peter Siddons, Andrei Fluerasu, Trevor Tyson

Abstract— In this work we have demonstrated a 256-channel(element) correlator array design based on a Virtex 6 FPGA for X-ray Photon Correlation Spectroscopy that uses Dynamic Light Scattering to probe nanometer scale structures, where the channel here means one correlator element which could provide auto correlation functions for 36 lags. This design is incorporated along with a design consisting of a 64x64 pixel silicon detector array and a custom 3D integrated circuit called vertically integrated pixel imaging chip (VIPIC). The challenge was how to handle the large amounts of data from VIPIC while performing correlation analysis in a real time. Our design introduces an autocorrelator per pixel to address the per pixel momentum transfer. Initially the VIPIC collects the X-ray photon arrival events from the 64x64 pixels and transfers it over 16 serial buses to the data acquisition system. Timing information is provided by reading the detector contents every 10 μs. Readout can be in one of two modes, a sparsified readout mode for low intensity applications where many pixels acquire no events, and an imaging mode in which all pixels are read out sequentially. The correlator design described here can handle either operating mode. This design provides many advantages over the existing commercial correlators as they are not able to meet the requirements of VIPIC. The commercial ones are limited to low intensity light correlation and a few simultaneous inputs. Besides, the large physical size of the commercial correlators makes the implementation of a system with several channels impractical. A multi-τ correlator design based on virtex 2 was addressed earlier but the practical realization was only ended up for one element of such muti-τ correlator. The correlator design discussed in this paper will provide a correlation dynamic range from 10μs to 10.24ms for each of its 256 correlator elements. Our target for the design is to reach the minimum physical size, resource and power consumption costs based on the approach of using an 8 stage multi-τ design for each element and barrel shift DSPs and pipelined RAM block for the elements. We are planning to implement this design as the basis for a custom integrated circuit which implements 4x256 element correlator elements.

I. Introduction

X-ray Photon Correlation Spectroscopy [1-3] is similar to traditional Dynamic Light Scattering, but because it uses the shorter wavelengths of X-rays, it is able to probe length scales down to nanometers. A design consists of a 64x64 pixel silicon detector array and a custom 3D integrated circuit called vertically integrated pixel imaging chip (VIPIC) [4] used in the detector system that collects the X-ray photon arrival events from the 64x64 pixels and transfers it over 16 serial buses to the data acquisition system. Time information is provided by reading the detector contents every 10 μs. Readout can be in one of two modes, a sparsified readout mode for low intensity applications where many pixels acquire no events, and an imaging mode in which all pixels are read out sequentially. The main challenge in the system is how to handle the large amounts of data from VIPIC while performing correlation analysis in a real time.

The light or X-ray intensity correlation is very important analysis tool to understand the interplay between light or X-ray coming out from the subject under study and its original source in the coherent scattering research area, which could reveal the quantum nature of subject under study. It is required to calculate the second-order degree of coherence, which is defined as follows [2].

$$g^{(2)}(\tau) = \frac{\langle I(t)I(t+\tau)\rangle}{\langle I(t)\rangle^2}$$
$$= \frac{\langle E^*(t)E(t)E^*(t+\tau)E(t+\tau)\rangle}{\langle E^*(t)E(t)\rangle^2}$$

$$(1)$$

Where $g^{(2)}(\tau)$ is the second order temporal coherence of the light, $I(t)$ and $E(t)$ are the intensity and electric field of light at time t, $I(t+\tau)$ and $E(t+\tau)$ are the intensity and electric field at time t+τ, E^* is the conjugate of E.

There are 3 types of correlator which are linear correlator, exponential correlator and multi-τ correlators [5]. A design of 8 stage multi-τ correlator as discussed in [5] is intended for the low intensity light correlation which means the intensity count is either 1 or 0. A more profound correlator design is discussed in [6] but the description of its function and architecture is very unclear. The correlator in this paper will deal with a high intensity (0-31counts/10μs for each pixel) counts and the correlation range from 10us to 10.24ms at the lowest resource cost.

Each correlator element in this paper needs to calculate the auto correlation function as follows:

$$G(n) = \sum_{i=1}^{m} x(i) \times x(i+n)$$

$$(2)$$

where $x(i)$ is the ith data, $x(i+n)$ is the (i+n)[th] data, the interval between n[th] and (n+1)[th] is one τ, $G(n)$ is the n-τ auto correlation function of x(i), m is the total number of the data.

The correlator element is also required to provide the Intp(n), Intf(n) and m for the purpose of symmetric normalization g(n) of G(n) as follows:

$$g(n) = \frac{G(n)}{Intp(n) * Intf(n)} \times (m - n)$$

(3)

(3) where $Intp(n) = \sum_{i=1}^{m-n} x(i)$ and $Intf(n) = \sum_{i=n+1}^{m} x(i)$

(4)

Such a method is used frequently in X-ray intensity fluctuation spectroscopy (XIFS) to derive the drifting time constant by calculating the auto correlation functions of intensity. An example is found in paper [3] to measure the equilibrium dynamics of colloidal suspensions of charge-stabilized Sb2O5.

Scientists in Fermi Lab and BNL have developed the X-ray silicon detector and corresponding ASIC chip which is called VIPIC [4] as described earlier to collect the intensity counts of X-ray. The VIPIC chip is optimized for monochromatic photon beam of 8 keV. The prototype consists of a 64x64 pixel silicon detector array. The physical dimension of an individual pixel is 80×80 μm^2. The total active area of the chip is 5120×5120 μm^2. The VIPIC chip will be bonded to the detector using the Direct Bonding Interconnect (DBI) technique. It is designed to yield 10 μs frame readout time at the mean occupancy of 3.8×10^8 ph/cm^2/s at the serial readout clock frequency of 100 MHz. This is achieved by dividing the whole pixel array in 16 readout groups of pixels. Each group is read out through an individual output port [4]. And each group handles 62 hit pixels every 10 μs with 3 bits of start signal, 5 bits of photon counts, 8 bits of pixel address for every pixel.

In order to derive the intensity auto correlation information from 64x64 pixels, a 16x256 auto correlator array system is needed. Each correlator is required to have the time resolution of 10 μs and correlation dynamic range of 10.24 ms. The 16x256 correlator array system is intended to be placed after the VIPIC and before the microprocessor unit as shown in Fig. 1. The correlator will be programmed into a multi FPGA chips set, the Virtex 6 family of Xilinx. The entire detector system structure is demonstrated in Fig. 1, which has the 64x64 X-ray silicon semiconductor Sensor, VIPIC ASIC and FGPA chips which contain 256 correlator arrays each.

The 256-element correlator system design discussed here needs to extract the photon counts and the corresponding pixel address from one of 16 serial bus of read out groups in VIPIC at 10 ns rate for data bit and 10 μs frame for all the hits out of 256 pixels. The photon counts from each pixel needs to be pipelined and propagated in the correlator system at the range of 1τ to 128τ for the symmetric normalized autocorrelation function g(n) with lags which are from 1 to 8 τ with increasing step of 1 τ and $2^{m+1}+2^{m-1}$ to 2^{m+2} τ with increasing step of 2^{m-1} τ where m is from 2 to 8, total 36 different lags with 10μs τ. The memory depth to store the calculation results needs to be 64 bits for running long time auto correlation. The g(n) for all

pixels should be read out by external clock on 64 bit parallel bus sequentially. The objectives of the work is to design with the VHDL the 36 lags correlator element and 256 of such a
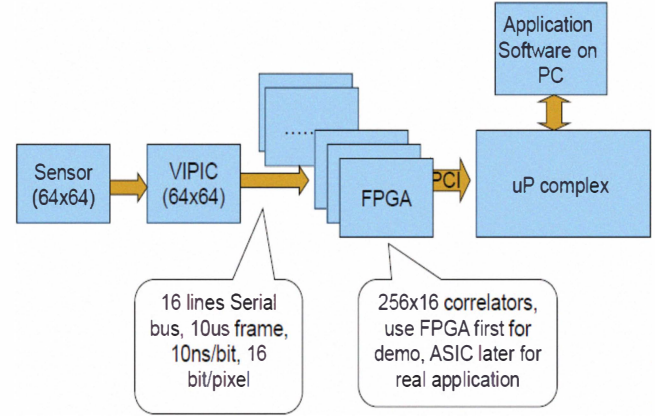

Fig. 1. X-ray Photon Correlation Spectroscopy detector system architecture.

system with minimum logic elements by utllizing the pipelined RAM, barrel shifter DSP structure. The design process includes the architecture, VHDL description, simulation and synthesis in the Xilinx tools to emulate in FPGA chips, where XC5VLX330 is used for 36 lags correlator element synthesis at the beginning and XC6VLX550T, XC6VLX240T are used for 256-channel(element) correlator system synthesis latterly to demonstrate how the logic elements is reduces with optimized hardware architecture.

II. ARCHITECTURE OF ONE CORRELATOR ELEMENT

Since the correlator needs to satisfy 1 to 1024 τ dynamic range, and each τ requires one sum of products (SOP) unit. 36 lags multi-τ correlator structure depicted as follows is adopted to satisfy the required dynamic range with minimize SOP resource at beginning and later a more centralized SOP structure is used.

In the 36 lags correlator system, auto correlation function of 36 different τ is calculated simultaneously by parallel structure DSPs. The whole system is divided in to 8 stages.

The first stage, which is called 8_lags_correlator in Fig. 2, is to calculate the autocorrelation functions from 1 to 8 τ with increasing step of 1 τ. The stages from stage 2 to stage 8, called 4_lags_correlator in Fig. 3, are to calculate the autocorrelation functions from $2^{n+1}+2^{n-1}$ τ to 2^{n+2} τ with increasing step of 2^{n-1} τ where n = stage number.

At each stage, the incoming data is propagated with the 2^{n-1} τ delay from pipe node to node. In the first stage (8_lags_correlator), 8 SOP units are paralleled to calculate the auto correlation function for each lag at the same time. The rest stages, 4 SOP units at each stage are used for 4 lags. The parallel DSP structure could be further optimized by using barrel shifter DSP structure to reduce the number of DSP, memory and other logic resource, which is described later.
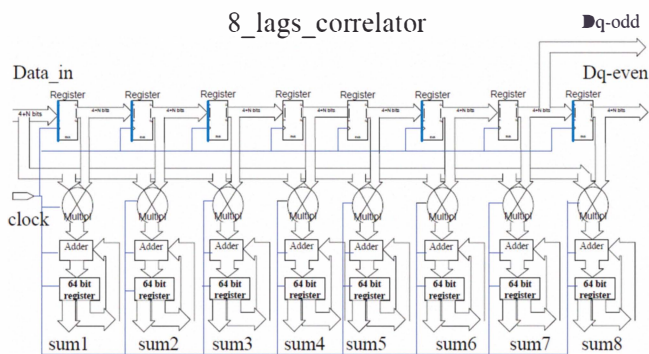
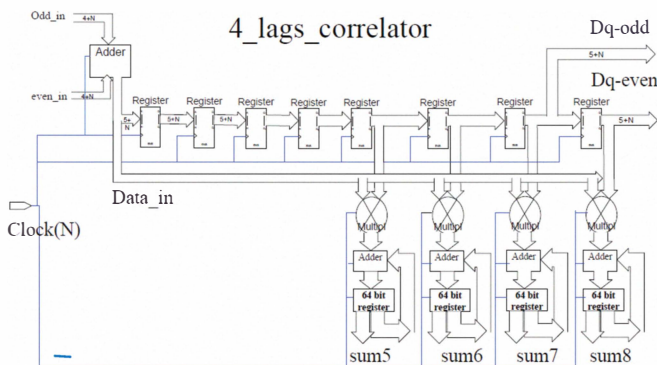Fig. 2. 8 lags correlator system diagram for first stage.

Fig. 3. 4 lags correlator system diagram for stage 2 to 8.

The serial data input to the first stage is 5 bit width with 10us clock period rate. The data is branched into "Dq_odd" and "Dq_even" when passes to the next stage, which is the 2 adjacent numbers (odd and even) of the input data pipelined. The "Dq_odd" and "Dq_even" is summed up in the following stage and feeds to pipelined 8 registers at that stage.

When it is required to stop the correlator at the end of beam line experiment, to avoid generating the statistic noise, one challenge is how to stop the data flowing properly in the pipes without flushing the useful data away which is taking into count in the previous stage. Therefore, the enable/disable signal for each stage should be designed as token signal passed to next stage with proper delay. Fig. 4 displays such mechanism. When en_in, the signal to start the auto correlation, is set to 1, all 8 stages can be enabled at same time since all the registers of pipelines are latched at 0 which will not affect the results of lags. But if en_in is set to 0, which means to stop, all 8 stages should be disabled sequentially in a method such that the feeded data in the pipe of each stage is fully dumped one after another. The process is demonstrated in

Fig. 4, where en_in is properly delayed to each stage until all the useful data in pipe of its stage is "used up".
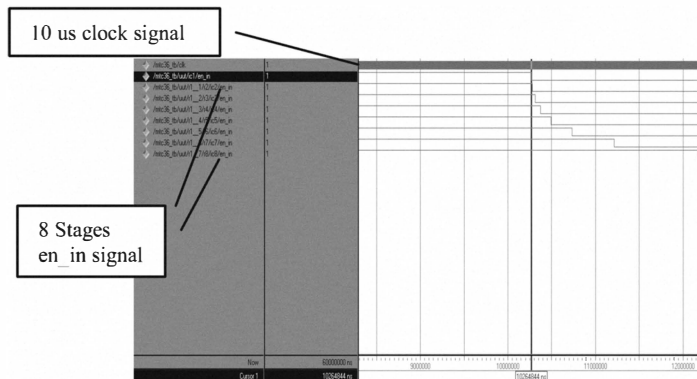

Fig. 4. enable (1 is enable, 0 is disable) signal (en_in) for each stage (ic1 to ic8). When en_in at ic1 (stage 1) is set to 0, en_in sginals at the rest stages are sequentially turned to 0 when the data in its pipe is fully dumped.

III. VHDL CODE DESIGN FOR ONE CORRELATOR ELEMENT

The VHDL code program structure for 36 lags correlator system including test bench program is displayed in Fig. 5.

MTC36_system.vhd is the major program for 36 lags correlator which contains 1 MTC8_core component (ic1) and 7 MTC4_core components (ic2 to ic8). The Ram.vhd is the 64 bit width memory to store the 36 lags, 36 sets of "intf" and 36 sets of "intp" data and the total number of feed in data and its summation obtained from the correlator. The data type and components of VHDL code is defined in the package file cor_package.vhd.

The MTC8_core.vhd defines the finite stage machine (FSM) and pipeline circuit for 8 lags correlator shown in Fig. 2. The MTC4_core.vhd defines the circuit for 4 lags correlator shown in Fig. 3. MTC_all_core_readout.vhd links the model of "MTC36_system.vhd" and "Ram.vhd" together. Test_bench_mem.vhd is the test bench program for MTC_all_core_readout.vhd to feed the stimulus to MTC_all_core_readout module and load the symmetric normalized auto correlation function data required for equation 3 from Ram.vhd to the external text file. More details will be elaborated the next section.

IV. THE CORRELATOR ELEMENT CIRCUIT IMPLEMENTED IN VIRTEX 5

Since MTC36_system.vhd program has too many I/O ports, which causes the I/O assignment problem when implemented in virtex 5, MTC_all_core.vhd program is designed to reduce the I/O ports by storing these 36 lags, 36 sets of intf, intp, one set each for the total number and sum of data to RAM module instead of putting them on the I/O.

The MTC_all_core.vhd program is synthesized and implemented on XC5vlx330 by ISE version 9.2 tool. The table 1 and Fig. 6 display the resource cost, speed and register transfer level (RTL) schematic map for the module.
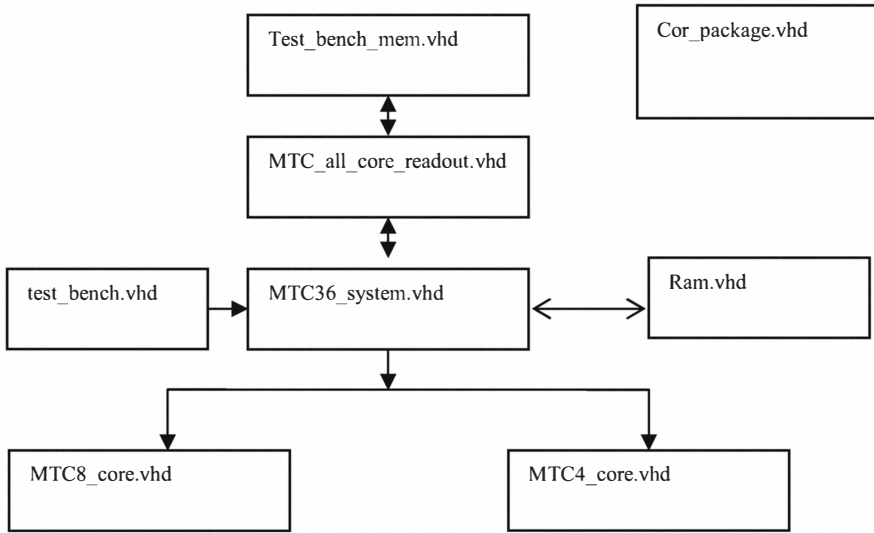
Fig.5. Software architecture of the correlator element.

Table 1. Resource cost of signal correlator element and max. delay

No. of slice register used:	*2956*	*Utilization: 1%*	
No. of slice LUTs used:	*3643*	*Utilization: 1%*	
No. of DSP48Es used:	*24*	*Utilization: 12%*	

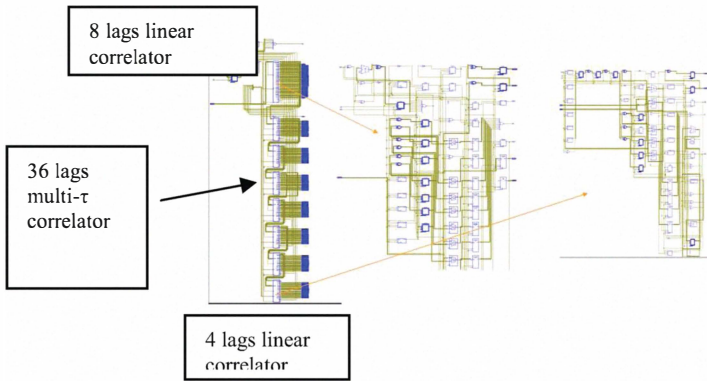*Max. delay: clk to IC1 PSUM signal: 10.190ns*

Fig. 6. 36 lags multi-τ correlator system on virtex 5.

V. FUNCTION VERIFICATION

The following steps were taken to verify the function of 36 lags correlator element.

Step1: Design a program in Matlab which can simulate the 36 lags correlator function specified earlier in the paper. Then use the sine ware as the stimulus and get the expected correlation results from this program. The results were plotted with red on Fig. 9 and Fig. 10. This will be used as expected results for VHDL simulations in the next steps.

Step 2: This step is for Modsim function simulation. Firstly design a test bench program in VHDL which is called "test_bench.vhd" and could test MTC36_system.vhd with the same sine wave used in step 1 by getting the data to form the

sine wave from external text file, then run the simulation on Modsim and put the results obtained from 36 lags correlator element to another text file, from which the Matlab program gets the corresponding lag data to do the symmetric normalization and then plot the results on the same figure of step 1.

Step 3: This step is for post synthesis simulation, where we use the Test bench file "Test_bench_mem.vhd" to test in ISE the "MTC_all_core_readout.vhd" module which contains the modules of 36 lags correlator and lags memory. The test bench module reads out the results from lags RAM of post synthesis module and writes to the external text file. Finally, the Matlab works out the symmetric normalization on the data from the text file and plot the results on the same figure of step 1.

Fig. 7. Modsim simulation result of MTC36_tb_mem.

In step 2 and 3, both the test bench files of VHDL are designed to use the "textio" library of standard VHDL to read the data from text file which contains all the data for sine wave stimulus. The data is loaded on the serial bus with clock rate of 10 us and sent to the correlator element under test. When it reaches the end of text file, the test bench program will send a signal to stop the correlator and then collect the results of auto correlation function from it.

The tests are run with 2 types of sine wave stimulus, one with period of 62.8 τ, the other with 628 τ. The stimulus signal expressed in 15*(1+sin(x)) is feed to "data_in" signal in Fig. 2, where we could see if the x increases from 0 to 307.1 at step 0.1, it gives the period of 62.8 τ. If x increases from 0 to 30.71 at step 0.01, the period will be 628 τ.

When data_in is 15*(1+sin(x)) with T=62.8τ, the test results from Modsim are displayed in Fig. 7 where "lag" array stores the 36x64 bits of auto correlation results for 36 lags. "Intf*" and "intp*" are the data sets for symmetric normalization for 8 stages such as "Int1" is for the first stage which has 8 numbers etc. "data_in" signal are the sine wave stimulus, "term" are the total number of stimulus data (3072), "intc" are the sum of these stimulus data.

These data sets are stored in 128*64 bits ram in the order (from address 0 to 127) of 36*64 bit of "lag" first, then 8*64 bit of "intp1" plus 8*64 bit of "intf1" for the 1st stage and 4*64 bit of "intp2" plus 4*64 bit of "intf4" …… plus 4*64 bit of "intp8" plus 4*64 bit of "intf8" for the rest 4 lags stages, 64 bit of "term", 64 bit of "inc", which is displayed in Fig. 8.
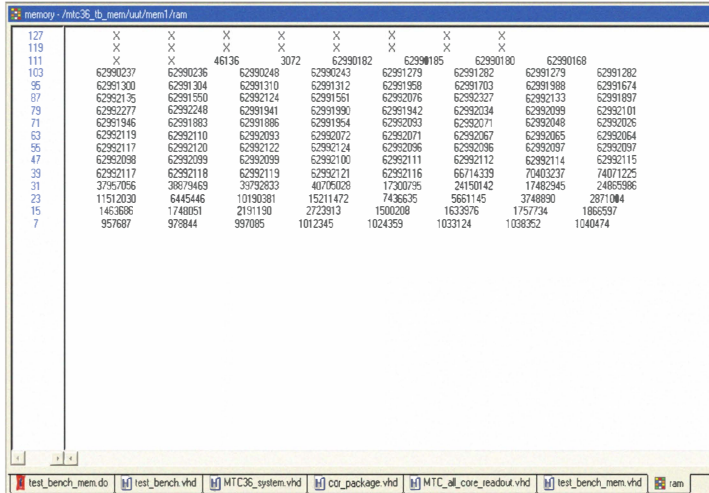

*Fig. 8. Simulation results of MTC36_tb_mem are stored in 128*64 bit RAM.*

The simulation results of 36 lags multi-τ correlator for the final post synthesis module for the sine wave stimulus with T=62.8τ and T=628τ are plotted in Fig. 9 and Fig. 10. The red curve on these figures is the Matlab program simulation result (step 1), VHDL post synthesis simulation results on ISE (step 3) is displayed in black curve. Also, the 1024 τ linear correlator simulation results of Matlab are displayed for the purpose of comparison with multi-τ approach, which shows that both methods are very close to each other before the lag comes to the period of sine wave stimulus.

From Fig. 9 and Fig. 10, we could see clearly that the simulation results of VHDL are perfectly overlapped with Matlab program. The feed in data is maximum correlated at T=62.8τ for Fig. 9 and 628τ for Fig. 10 and minimum correlated at the half T, which demonstrated the natural of sine wave. This proved that the function of VHDL correlator is correct.

VI. DESIGN WITH 256 ELEMENTS OF SUCH 36 LAGS CORRELATOR SYSTEM ON ONE CHIP

256 elements of such 36 lags correlator are integrated with the structure displayed in Fig. 11 at beginning, where 36 sets of DSP module, which utilize 24 sets of DSP block of Vertex chips of Xilinx, are separated from each correlator element and centrialized as global DSP for 256 sets of data input at different stages, a 256 more times faster clock period than $2^{n-1}\tau$ synchronizes DSP block to chop the data from "data_in" signal properly within $2^{n-1}\tau$ period at each stage where again n is the stage number. Such a DSP structure to reduce the number of DSP blocks used in the design is called the barrel shifter DSP structure in this paper.
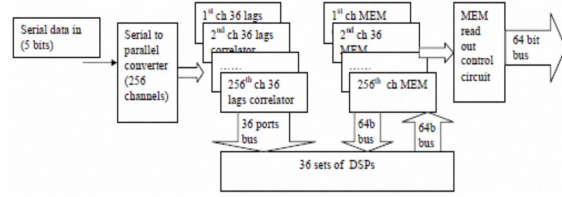

Fig. 11. 256 channel correlator system, each channel is with 36 lags.

The resource cost of XC6VLX550T with such a system by using the barrel shifter DSP structure is listed in table 2.

Table 2. Resource cost of XC6VLX550t

Device Utilization Summary (estimated values)				[-]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	366882	687360	53%	
Number of Slice LUTs	307661	343680	89%	
Number of fully used LUT-FF pairs	189972	484571	39%	
Number of bonded IOBs	72	840	8%	
Number of BUFG/BUFGCTRLs	17	32	53%	
Number of DSP48E1s	24	864	2%	

From table 2, it is obvious that the design still requires to build the pipe lines lots of slice registers and look up tables (LUT) resources from Vertex chip which is more than that XC6VLX240T has; Therefore architecture of 256-element correlator system needs to be further optimized to by utilizing the more integrated DSP structure, pipelined RAM, and IO buffer resource of FPGA. In the further optimized design, 50MHZ clk rate is selected to chop the input data for DSP and pipeline the RAM blocks, such that, for slowly stage such as stage 3 which has 40 us delay time from pipe to pipe, one DSP module is able to calculate the autocorrelation function for 4x256 inputs within one pipeline period of 40 us, by doing this we can reduce 36 sets of DSP blocks to 13, which is shown in table 3. Also, for each stage, the SRAM blocks is added and pipelined up with selected clock which is 256 times fast than clock(N) displayed in Fig. 3 to replace the registers used in the pipe line. This will save seas of registers for each stage. Finally, the I/O buffer resource is used for the DSP, RAM output to reduce the number of multiplexer. The planned new resource cost per stage for the optimized design is listed in table 3.

The synthesis results on ISE with the further optimized design on XC6VLX240T-3FF1156 are listed in table 4.

By analyzing the data from table1, 2, 4 we could conclude that the further optimized design work the best to utilize less resources and reach the smaller physical dimension for ASIC.

VII. CONCLUSION

From the simulation, the function of 36 lags correlator element is demonstrated. It meets the expectation described in the specifications. Array of 256 of such correlator element is possible to be realized on one single Chip of Vertex 6. The synthesis result of further optimized design shows that it could even fix the size of XC6VLX240T. The future work is to
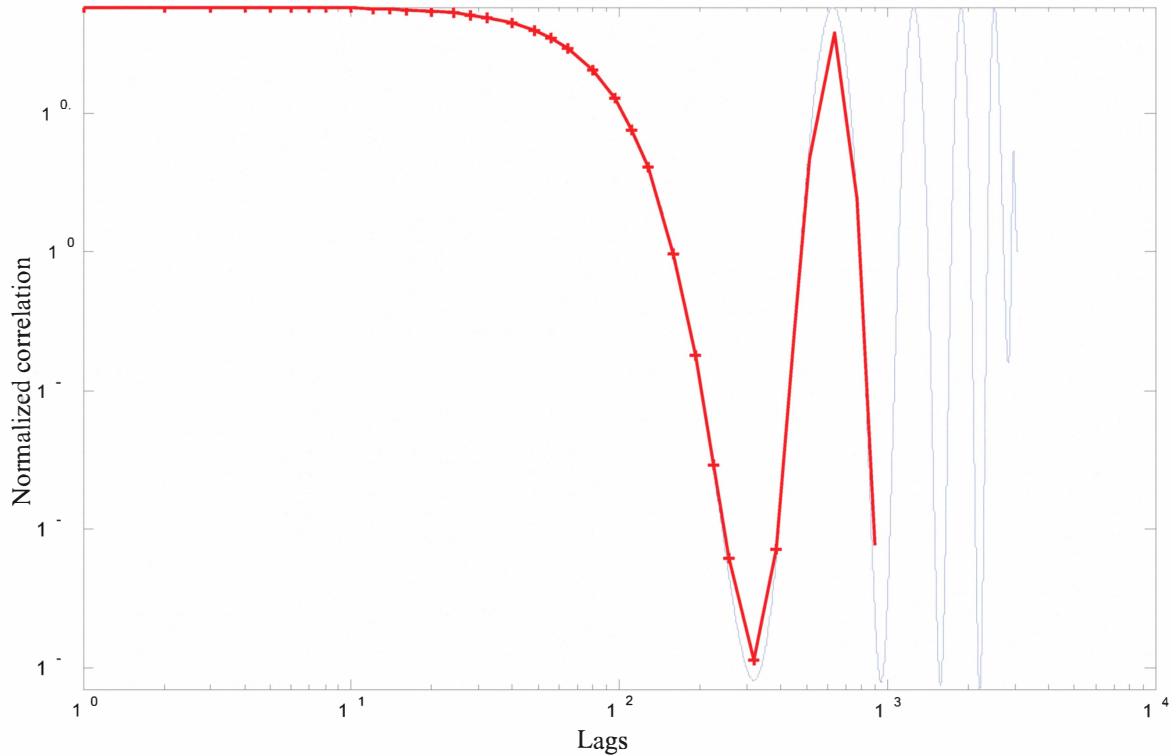
Plot of g$^{(2)}$ in logarithm scale

Fig. 9. *When the stimulus is T=62.8τ sine wave, the function simulation result(red) on matlab and post synthesis simulation result(black) for 36 lags multi-τ correlator and function simulation result(blue) on matlab for 1024 lags linear correlator is displayed here.*
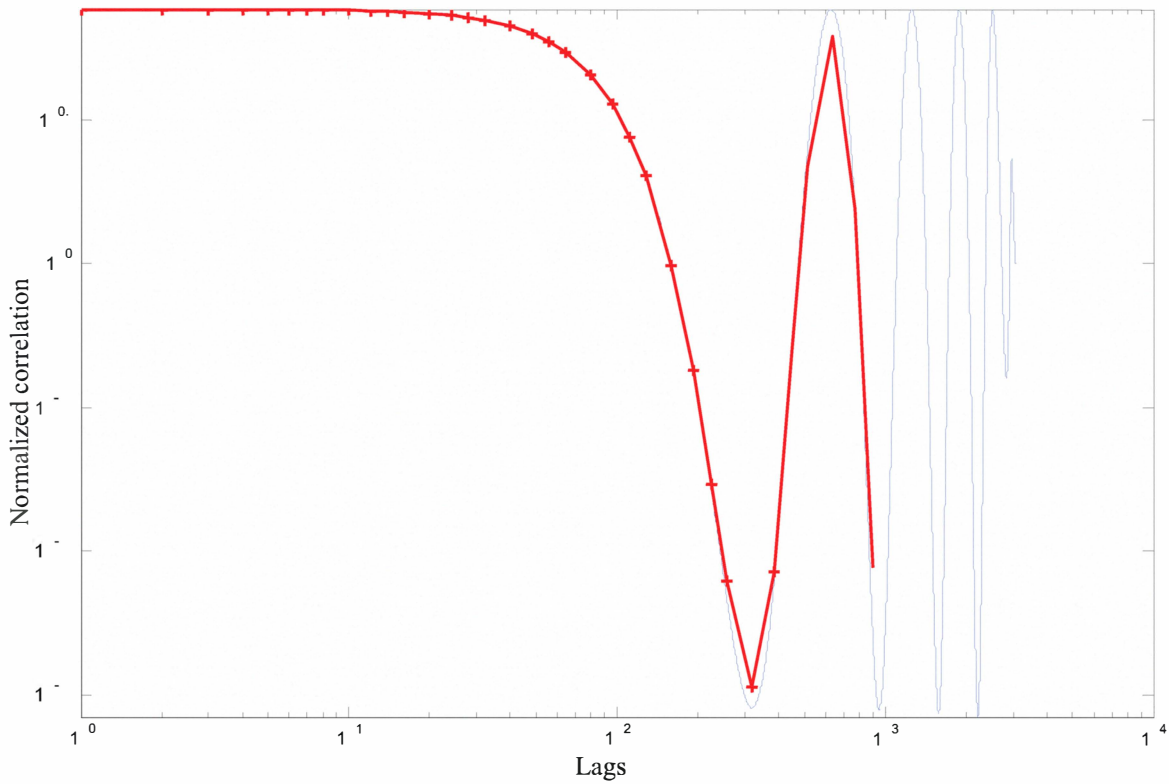
Plot of g$^{(2)}$ in logarithm scale

Fig. 10. *When the stimulus is T=628τ sine wave, the function simulation result(red) on matlab and post synthesis simulation result(black) for 36 lags multi-τ correlator on modsim and function simulation result(blue) on matlab for 1024 lags linear correlator is displayed here.*

Table3. Planned resource cost for optimized design with more integrated

Stage	Data width (bit)	Lags	Pipe lined RAM	DSP needed	RAM for lags	RAM for normalization data	IO port needed
1	5	8	8 of 256x5	8	8 of 256x64	8 of 512x5	64
2	6	4	8 of 256x6	2	2 of 512x64	2 of 1024x6	64
3	7	4	8 of 256x7	1	1 of 1024x64	1 of 2048x7	64
4	8	4	8 of 256x8	1	1 of 1024x64	1 of 2048x8	64
5	9	4	8 of 256x9	1	1 of 4096x64	1 of 4096x12	64
6	10	4	8 of 256x10				
7	11	4	8 of 256x11				
8	12	4	8 of 256x12				

Table 4. Actual resource cost for optimized design by utilizing more integrated DSP, RAM and I/O buffer on XC6VLX240T

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	6114	301440	2%
Number of Slice LUTs	8137	150720	5%
Number of fully used LUT-FF pairs	4839	9412	51%
Number of bonded IOBs	72	600	12%
Number of Block RAM/FIFO	74	416	17%
Number of BUFG/BUFGCTRLs	9	32	28%
Number of DSP48E1s	3	768	0%

develop the test system to verify the function of a 256 correlator element system on XC6VLX240T evaluation board. The development activities should include the utilization of real data from beam lines as stimulus, design of transmitter module, communication modules for unit under test, GUI on computer to plot the curve of g $^{(2)}$ vs. lags in real time. After the VHDL design is verified on FPGA, we will come to the ASIC design for such a correlator system with 90 nm technology finally.

REFERENCES

[1] K. Schatzel, "Correlation techniques in dynamic light scattering", Appl. Phys. B 42, 193-213 (1987).
[2] Mark Sutton, "A review of X-ray intensity fluctuation spectroscopy" C. R. Physique 9 (2008) 657–667.
[3] O. K. C. Tsui and S. G. J. Mochrie, "Dynamics of concentrated colloidal suspensions probed by x-ray correlation spectroscopy", PHYSICAL REVIEW E VOLUME 57,NUMBER 2, FEBRUARY 1998.
[4] Grzegorz Deptuch, "VIPIC, vertically integrated pixel imaging chip".
[5] Wei Liu, Jin Shen, "Design of multiple-τ photon correlation implemented by FPGA", DOI 10.1109/ICESS.2008.7.
[6] B.Hoppe, H.Meuth, M.Engels and R.Peters, "Design of digital correlation systems for low-intensity precision photon spectroscopic measurements", IEE Proc.-Circuits Devices Syst., Vol. 148. No. 5. October 2001.