2012

Feedback Control for Electron Beam Lithography

Yugu Yang
University of Kentucky, yuguyang@gmail.com

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained and attached hereto needed written permission statements(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine).

I hereby grant to The University of Kentucky and its agents the non-exclusive license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless a preapproved embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student’s advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student’s dissertation including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Yugu Yang, Student

Dr. J. Todd Hastings, Major Professor

Dr. Zhi David Chen, Director of Graduate Studies

FEEDBACK CONTROL FOR ELECTRON BEAM LITHOGRAPHY

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Engineering at the
University of Kentucky

By
Yugu Yang
Lexington, Kentucky

Director: Dr. J. Todd Hastings, Associate professor of Electrical and Computer
Engineering
Lexington, Kentucky 2012

ABSTRACT OF DISSERTATION

FEEDBACK CONTROL FOR ELECTRON BEAM LITHOGRAPHY

Scanning-electron-beam lithography (SEBL) is the primary technology to generate arbitrary features at the nano-scale. However, pattern placement accuracy still remains poor compared to its resolution due to the open-loop nature of SEBL systems. Vibration, stray electromagnetic fields, deflection distortion and hysteresis, substrate charging, and other factors prevent the electron-beam from reaching its target position and one has no way to determine the actual beam position during patterning with conventional systems. To improve the pattern placement accuracy, spatial-phase-locked electron-beam lithography (SPLEBL) provides feedback control of electron-beam position by monitoring the secondary electron signal from electron-transparent fiducial grids on the substrate. While scanning the electron beam over the fiducial grids, the phase of the grid signal is analyzed to estimate the electron-beam position error; then the estimates are sent back to beam deflection system to correct the position error. In this way, closed-loop control is provided to ensure pattern placement accuracy. The implementation of spatial-phase-locking on high speed field-programmable gate array (FPGA) provides a low-cost method to create a nano-manufacturing platform with 1 nm precision and significantly improved throughput.

Shot-to-shot, or pixel-to-pixel, dose variation during EBL is a significant practical and fundamental problem. Dose variations associated with charging, electron source instability, optical system drift, and ultimately shot noise in the beam itself conspire to increase critical dimension variability and line width roughness and to limit the throughput. It would be an important improvement to e-beam patterning technology if real-time feedback control of electron-dose were provided to improve pattern quality and throughput even beyond the shot noise limit. A novel approach is proposed in this document to achieve the real-time dose control based on the measurement of electron arrival at the sample to be patterned, rather than from the source or another point in the electron-optical system. A dose control algorithm, implementation on FPGA, and initial experiment results for the real-time feedback dose control on the e-beam patterning tool is also presented.

KEYWORDS: Electron Beam Lithography, feedback control, pattern placement error, dose variation

Author's signature: Yugu Yang

Date: September 26, 2012

FEEDBACK CONTROL FOR ELECTRON BEAM LITHOGRAPHY

By
Yugu Yang

Director of Dissertation: J. Todd Hastings

Director of Graduate Studies: Zhi David Chen

Date: September 26, 2012

To my parents.

ACKNOWLEDGMENTS

I cannot imagine completing this thesis without the guidance and encouragement from my adviser, Professor Todd Hastings. His endless enthusiasm has always been inspirational to me, and his constructive suggestions with regard to technical issues and research methods have helped me overcome countless difficulties through this work. His willingness to give his time so generously, to either troubleshooting or to proofreading this and many other documents, has been very much appreciated. I consider myself extraordinarily lucky and privileged to be his student and to have had the opportunity to work in his lab.

It has been a great experience to work with the members of UK's Center for Nanoscale Science and Engineering. Dr. Chuck May has provided many useful insights into developing recipes for fabricating gratings. I'd also like to thank him for acquiring the PEK-130A6 photoresist for short wavelength lithography. Brian Wajdyk taught me how to use most the instruments necessary for the successful completion of this work. His diligence in maintaining the equipment has made this process a smooth one.

I've been fortunate to work alongside many talented people in our group. Dr. Chandan Samantaray laid the foundations of the interference lithography setup, which accelerated the process of adapting it to my project. Dr. Eugene Donev helped me develop and improve my skills at aligning optics. I also had many productive discussions with him concerning electron beam lithography. Last but not least, Neha Nehru has been a great friend, with whom I could share my experiences as a graduate student, both good and bad.

I am very grateful to my parents for their unconditional support and love. Their influence has carried me through the toughest moments in the past years. I don't know how to thank my loving husband Donnie Keathley for being patient and understanding, and helping me maintain a positive outlook.

Finally, I'd like to thank the National Science Foundation for supporting the work presented in this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The invention of integrated circuit (IC) in mid-20th-century revolutionized the industry of electronics. It can be found virtually in any electronic component today, and its absence from our modern, digital life is almost unimaginable. This is made possible because of the transition from manually assembling discrete electronic elements to mass production of highly compact circuits, leading to significantly reduced production cost per elementary function as well as increased performance to cost ratio. From the very first version of integrated circuit consisting of only a few transistors, to several billions on the surface of a thin substrate of semiconductor material today, miniaturization of circuitry by scaling down the transistor has followed the trend, known as "Moore's Law", stated by Gordon Moore in his paper published in 1965 [1]. This trend has become the principal driver for the International Technology Roadmap for Semiconductors (ITRS), and the engine of the virtuous cycle of the semiconductor industry.

Planar fabrication technology, developed by Jean Hoerni [2, 3], has been the standard process of manufacturing the integrated circuit. By projecting a circuit on a two-dimentional plane, the concept of photographic process is adopted such that the transistors and the interconnections are built up vertically from the semiconductor substrate with layers of materials aligned with each other, as shown in Figure 1.1. Therefore, the functionality of each element and the entire circuitry is determined by the geometric patterns in each layer and the relative placement between layers, both

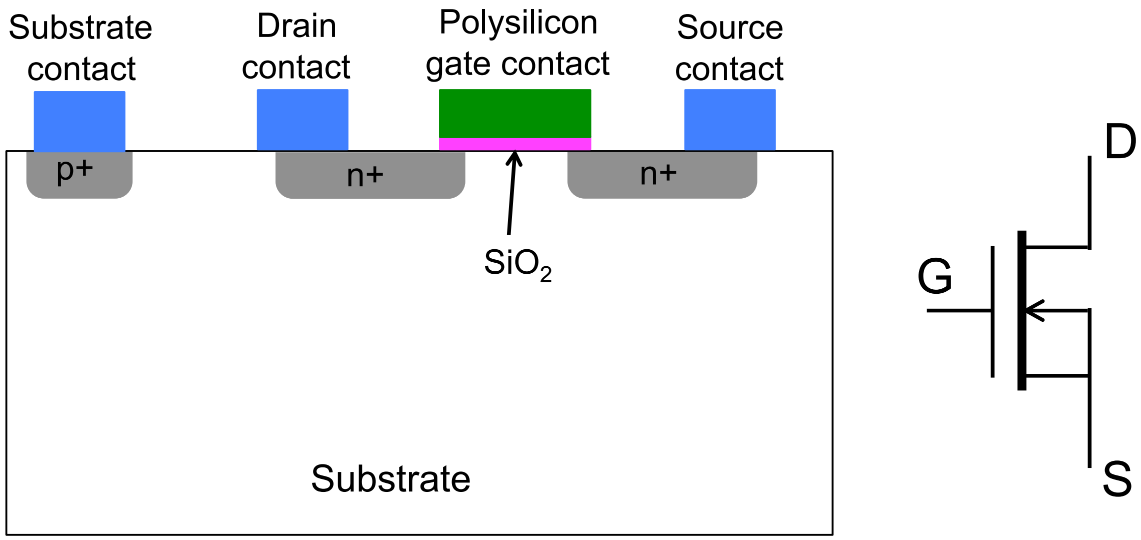


Figure 1.1: The cross section of an NMOS transistor fabricated by planar technology on the surface of a thin layer of semicondct substrate.

of which are defined by lithography. Lithography shares some fundamental principles with photography in that the pattern is created by transferring energy from a source to a layer of material that goes through certain chemical changes upon the absorption of energy.

For mass production of the electronic devices today, optical lithography (also called photolithography) still remains an essential step because of its extremely high throughput. However, to maintain the trend of continuously shrinking minimum half-pitch [4], it is facing the near-term challenge of transitioning from a single optical exposure to either double and multiple patterning or patterning using EUV light. Although other lithography technologies have been developed for nanostructure fabrication, such as electron beam lithography (EBL), x-ray lithography and ion beam lithography, they are not viable for cost-effective semiconductor manufacturing and mostly used for production of low-volume devices and research/development. Although EBL has not been used to pattern directly on Si substrate for IC production, it has found wide usage in writing photolithography mask, thus also faces the challenges of continuously tracking the requirements.

As the patterned feature migrates to smaller size, the accuracy of patterning in a layer of material with respect to the existing ones has to be maintained within a certain fraction of the critical dimension (CD). The roadmap projects image placement in an optical mask with 2.7 nm error (3 sigma) in 2015, but the manufactural solution still remains unknown at the time of writing of this document [4]. However, the development of Spatial-Phase-Locked Electron Beam Lithography (SPLEBL) offers a promising solution to the problem.

The pattern placement error arises from the fundamental open-loop nature of EBL. During patterning, the electron beam is focused onto the target substrate and the feature is exposed pixel by pixel, or shot by shot. Despite the extensive effort invested in the development of individual components, one has no means of knowing the actual electron beam position. By detecting a reference signal from the substrate, processing the signal to calculate the deviation of the beam position, and feeding back the correction signal, SPLEBL monitors and controls the beam with a feedback loop, thereby improving the pattern placement accuracy. The work reported in Hastings' paper [5] using this technology has demonstrated 1 nm (1 sigma) pattern placement error, which achieved the benchmark set for year 2014.

Another aspect of the issue associated with miniaturization is poor CD control and increased line edge roughness (LER). According to ITRS's report on optical masks updated in 2011, a manufacturing solution is still not available for the CD control of transistor gate and contact till now. And the same problem for half pitch of dynamic random access memory (DRAM) also needs to be addressed soon in order to reach the benchmark in year 2014 [4]. The line edge roughness, sometimes referred to as line width roughness, is especially critical for fabricating optical waveguides as the roughness can lead to energy loss.

This lack of pattern fidelity is caused by the dosage variation of EBL, which is due to the inherent statistical nature of electrons arriving at the substrate and the system imperfection. Dosage, often referred to as dose, is defined as the electric charges delivered to the substrate per unit area, and is directly related to the total number of electrons arriving at the sample. When the exposure of each pixel is restricted to a very few electrons, so that nanometer scale features can be obtained rapidly, the uncertainty of the total number of electrons involved will become prominent, resulting in worsened variation in size and geometry. We propose to address this issue by detecting a dose related signal from the substrate and establishing a feedback control loop for the dose, which will be highlighted in Chapter 6.

1.1 Electron Beam Lithography Overview

Electron Beam lithography is the primary technology for writing arbitrary patterns with sub-10-nm resolution. A beam of electrons is emitted and scanned across a layer of electron sensitive resist material on the substrate in a patterned fashion. Chemical properties of the resist change upon the incidence of the electrons, which leads to the resist being hardened or removed in the process of development to achieve pattern transfer.

An EBL system can be classified according to the size and shape of the e-beam. A scanning beam system focuses the electrons to a Gaussian-shaped spot and exposes the pattern pixel by pixel. A shaped beam, either fixed [6] or variable [7], projects the electron image of an aperture onto the substrate and exposes a segment of the pattern. This method helps reduce exposure time and improve field coverage compared to the scanning concept, but the resolution is highly dependent on the geometry of the pattern. Electron projection lithography combines the optical lithography approach with an electron source. With a broad beam of electrons projected onto a physical mask, the entire pattern is defined by a single exposure. Although the limitation of low throughput encountered by the scanning and shaped EBL is overcome by using this method, the major challenge becomes the design of mask with minimized Coulomb interaction between electrons and thereby the distortion of the final pattern. Since scanning EBL is mostly suitable for implementation with spatial-phase locking, the remainder of this document will exclusively focus on the scanning-beam system.

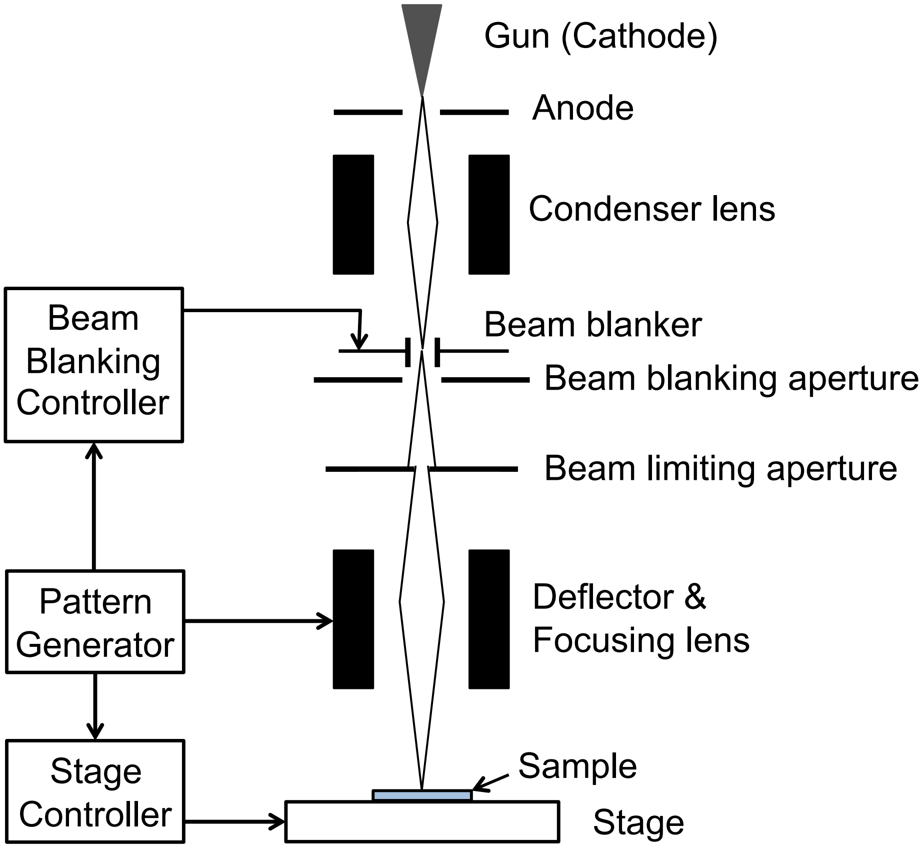


Figure 1.2: Schematic of the scanning electron-beam lithography system with major components.

1.1.1 Components of Electron Beam Lithogaphy System

The major components of the scanning electron beam lithography (SEBL) system include an electron source, electron-optical column, mechanical positioning stage, and a microprocessor that generates the pattern as illustrated in Figure 1.2.

Good current stability and sufficient brightness are the two most important criteria when choosing the electron source for the lithographic tool. For a thermionic emitter, electrons gain thermal energy from a heated conducting tip and overcome the work function of the metal. This type of source is used mostly in low resolution systems due to large electron energy spread and low current density, but it offers high stability over typical writing times of several hours and is less sensitive to gases in the environment. Although increased temperature delivers more current, the emitter's lifetime will be reduced exponentially. In applications requiring higher resolution, a field electron emitter is more desirable, as the typical energy spread of a field emitter is under one electron volt (eV), and the virtual source size is on the scale of nanometers [8]. Field emission occurs when the applied electric field is sufficiently strong so that electrons can tunnel through the barrier between the the emission source and vacuum. Due to the drawback of the short term fluctuations and long term drift of emission current, cold field emission source has seen little use in EBL. However, heating up the emitter tip, know as Schottky emitter, helps reduce the current instability without compromising the resolution dramatically.

While the emission source constantly supplies electrons, an electric field at specific potential is applied to accelerate emitted electrons. The typical primary electron energy used in lithography lies in the range between 1 KeV and 100 KeV. When electrons are accelerated with extremely high energy (at least 100 KeV), they gain enough energy to penetrate deep into the substrate material. Transmission electron microscopy makes use of this phenomenon when the substrate is thin enough to let electrons pass through, while high-aspect-ratio and high resolution features are fabricated by taking the advantage of electrons' high penetration [9, 10, 11].

Intensive studies on lithography at low e-beam energy (below 5 KeV), motivated by the benefits of significantly reduced proximity effect and thereby improved resolution, began in 1990s [12, 13, 14, 15]. In addition, electron beam microcolumns for high throughput patterning require to operate at no more than few electron volts due to the miniaturized electron sources [16, 17, 18, 19, 20]. Because the penetration depth is typically well below 100 nm at low primary electron energy, a thin resist with high sensitivity is required to achieve proper exposure. The requirement on the resist's conductivity becomes especially stringent as the severe surface charging can lead to pattern distortion [21].

After the electrons have been accelerated away from the emitter, they go through a series of apertures, electron-optical lenses, and deflectors before arriving at the substrate. Electron-optical lenses focus electrons either by applying electrostatic force or magnetic force. The term "optical" is used in this context because electrons can be manipulated by the magnetic or electrostatic filed in a similar manner to a light beam through an optical lens. However, the aberration of electron lens is far worse than an optical lens, which means the electrons with different energies are focused at different points along the electron-optic axis. The electrostatic lens is usually positioned adjacent to the source as a condenser lens for its worse aberration compared to a magnetic lens. In addition, the electrostatic lens helps pull out electrons from cathode. Since the final lens determines the performance of the electron-optic column, an electron lens with the highest quality should be placed closest to the substrate.

An aperture is basically a small hole through which electron beam passes along the electron-optic axis. A beam blanking aperture works together with a high-speed electrostatic deflector to interrupt the electron flow toward the substrate without shutting off the emitter. While not writing, the electron beam is swept away from the electron optic axis by the electrostatic deflector, and intercepted by the aperture. The beam limiting aperture truncates a portion of the beam to set the convergence angle through which the electrons pass through the column, and thus the beam spot size and current.

The beam deflection system is for scanning the electron beam across the substrate during microscopy and exposure. As the beam is deflected off the optical axis, additional aberration is introduced, resulting in pattern distortion. Therefore the deflection range with acceptable aberration is highly dependent on the optical column and varies from system to system. That being said, the reliable deflection range sets the upper limit of the dimension of the deflection field, thus the throughput of the system. Deflecting the electron beam can be done either electrostatically or magnetically. The magnetic deflector allows for a longer range of deflection than the electrostatic deflector, but its frequency response is limited due to the inductance of the magnetic coils and the eddy current introduced by the magnetic field.

Since the deflection of beam is limited over a small area, precise mechanical stage positioning needs to be combined with beam deflection to pattern large features by exposing multiple deflection fields and piecing them together. This is made possible by using two laser interferometers to measure the stage position in x- and y-direction. Two split laser beams are directed to a reference mirror and a mirror attached to the stage in each direction, then the interferometers compare the position of the stage mirror to that of the reference mirror to detect and correct stage position errors.

The pattern to be exposed is first drawn out using computer aided design (CAD) program, and then the pattern data is translated into two sequences of signals by the pattern generation system for EBL. One signal contains the information of the pixel addresses in the pattern coordinate, and is converted to an analog signal by a digital-to-analog converter (DAC), and eventually sent to the beam deflection system. The other signal sequence controls the beam blanking hardware, which is synchronized with the position signal to ensure that the beam is switched on and off at the desired location according to the pattern design. The smallest pixel size is determined by the beam deflection range and the DAC's resolution. The pixel size and dwell time at each pixel are chosen to achieve a desired exposure condition. To obtain the best result, the pixel size should be kept small, usually a fraction of the beam spot size, so that smooth feature edges can be obtained by overlaying each pixel.

1.1.2 Exposure Strategy

Since the beam cannot be reliably deflected more than a few millimeters due to the aberration of electron optics, while the pattern's dimension can be easily larger than that, the system must break up the patterns into segments and expose them one at a time. Depending on whether every single pixel is addressed within the segment during exposure, SEBL can be classified as vector-scan or raster-scan.

For a raster-scan system, the beam addresses every single pixel, and is switched on and off as required by the pattern design. This strategy is mostly used for high-throughput mask making when the stage is moved continuously in the direction perpendicular to beam raster scan direction. In this way, the limitation of small beam deflection range only presents itself in one direction, and is removed in the other by the stage movement. Therefore, patterns are built up with long strips as shown in Figure 1.3.

The vector-scan system only deflects the beam to where the pattern resides, and fills in the segmented shapes of the pattern as depicted in Figure 1.4. In this strategy,

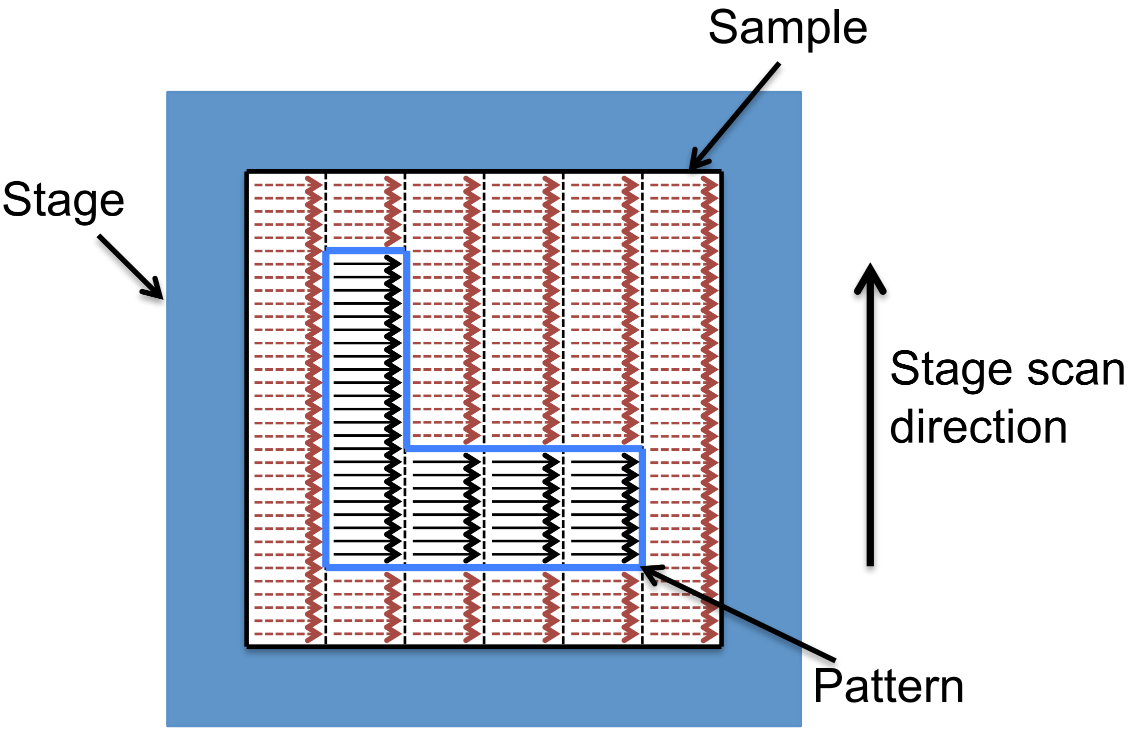


Figure 1.3: In raster-scan system, the beam is deflected in one direction to address every pixel address on the sample. E-beam is only switched on at where patterns reside. If the stage scans in the direction perpendicular to the beam scanning direction, pattern is stitched with long strips as illustrated here. Red arrows show the deflection with beam switched off, while blues ones with beam switched on.

the stage is kept stationary while the beam fills in the shape, and then moved to the next location. In addition to the strategy that fills the shape line by line in one direction, referred to as flyback, as illustrated in Figure 1.4, two other methods, known as meander and spiral, are also adopted by the SEBL system as depicted in Figure 1.5(b) and (c). The benefit of using either the meander or spiral method is that the overhead time is comparably less without the effort of settling the beam between lines, and distortion introduced by the dramatic change of beam movement direction is almost eliminated. Some systems also include circular mode as shown in Figure 1.5(d) to account for round shapes.

A vector-scan system is most efficient for writing sparse patterns, and it allows to introduce dose variation from shape to shape for correcting proximity effect. However,

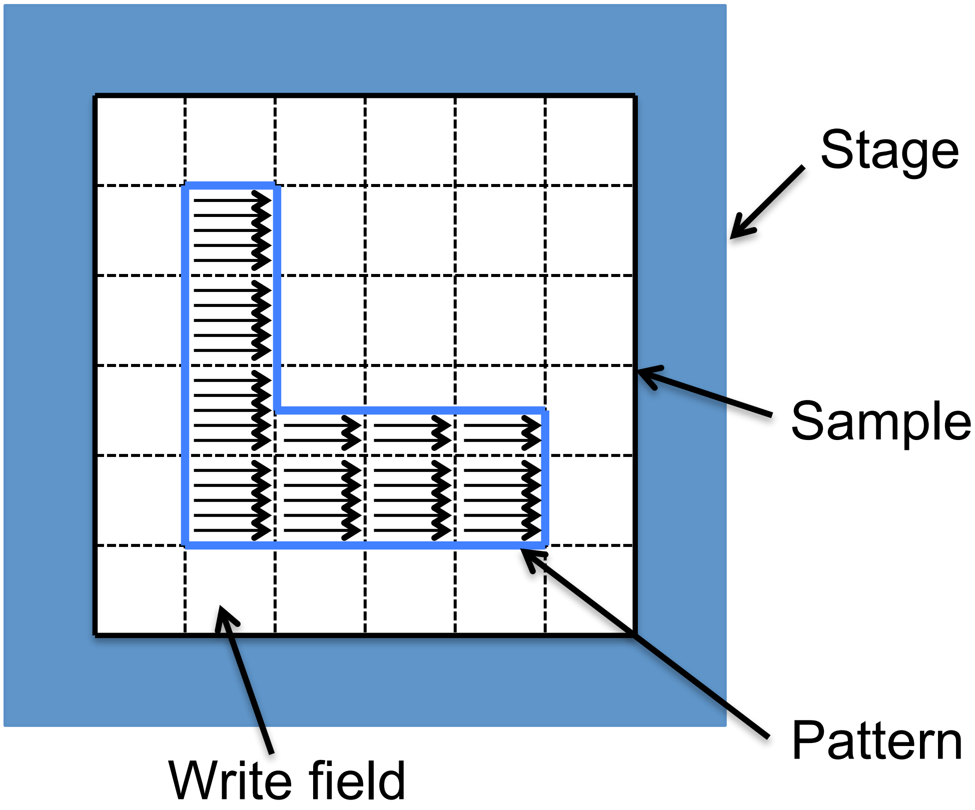


Figure 1.4: In vector-scan system, the beam is only deflected to where pattern is located, and the stage is fixed while the e-beam is scanning in the write filed. Only the flyback filling method is demonstrated in the schematics.

the overhead can be significant for dense patterns because of increased stage settling time before starting the exposure in each field. A vector-scan system is most suitable for research and prototyping, and sometimes for fabricating high-resolution, low-volumn devices.

1.1.3 Electron Resist Interactions

Patterning in electron sensitive resist is through the process of transferring energy from incident electrons to the polymer, in which either scission or crosslinking of the polymer chain occurs. The process of development after exposure dissolves lighter molecular weight polymers, revealing the pattern defined by e-beam writings. Although the change in chemical properties of the polymer is the direct cause of imaging, the physical interaction between the incident electrons and the resist/substrate

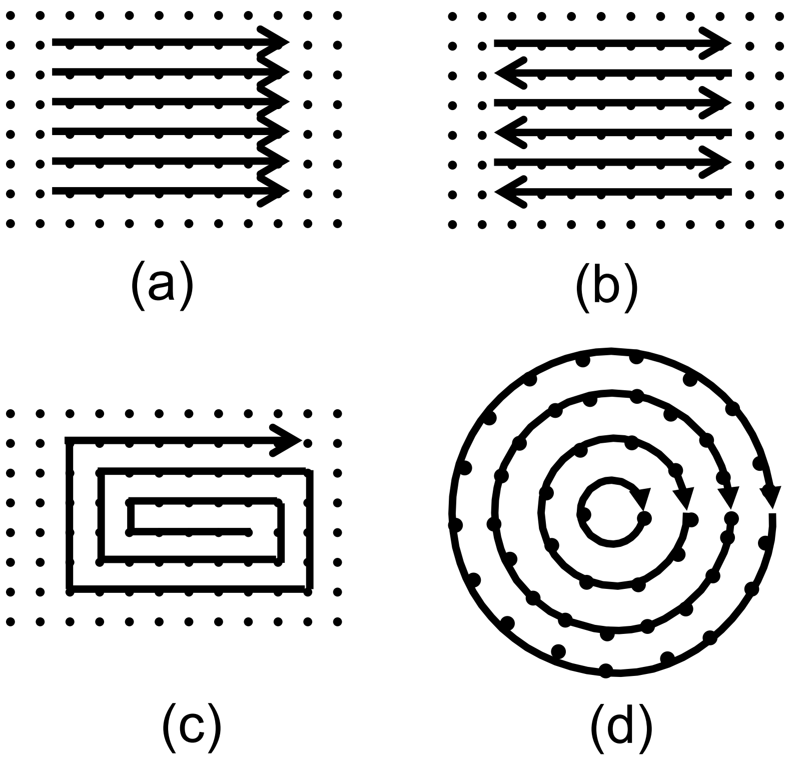


Figure 1.5: Filling method used in vector-scan system. (a) Flyback. (b) Meander. (c) Spiral. (d) Circular.

dictates the energy distribution, thus the chemical reaction in the polymer.

As a primary electron penetrates through resist, and travels further into substrate material, it experiences both elastic and inelastic scattering from the material atoms and electrons before it completely loses its energy. The average distance traveled by the electrons before they stop is defined as range, which is a very important parameter for characterizing the electron profile in the bulk material.

Forward scattering, corresponding to many small angle scatterings, contributes to the depth of electron penetration into material as well as the beam profile broadened at the bottom of the resist compared to that on the surface. To reduce the effect of beam expansion, one can increase the accelerating voltage and spin coat thinner resist. In some instances, though, it is made use of to tailor the resist sidewall angle.

When large angle scattering occurs, also referred to as backscattering, electrons travel laterally further from the incident location, and sometimes can return back through the resist, resulting in additional exposure. This undesired exposure is the reason of what is known as proximity effect in EBL system. The mean free path of backscattered electrons is determined by the incident energy and material, and therefore, reducing accelerating voltage and choosing the material with low atomic number can effectively reduce the proximity effect.

Both events are accompanied, in a continuous manner, by both energy loss and excitation of secondary electrons (SE) with energies less than 1 KeV. The secondary electron is the largest component of free electron population in the bulk of the material, and is responsible for the actual resist exposure. Because of its low energy, the range of the secondary electrons is typically a few nanometers, which accounts for the minimum practical resolution of the final pattern after development.

Charging in the sample can cause considerable distortion due to the unexpected deflection of the e-beam away or toward the charged area, if the electrons can not quickly gain access to a path to ground. For lithography using high primary electron energy (greater than 10 KeV) on a Si substrate, virtually all electrons stop in the bulk of the substrate and are eventually grounded. However, if exposure is performed on an insulating substrate, such as photomask, the embedded electrons will take much longer time to be dissipated. The most effective solution to dissipate charges during this high energy exposure is to deposit a thin layer of metal either above or below resist. For low energy exposure (about 10 KeV or lower), positive charging of the surface due to the secondary electron emission to the vacuum has more effect. In this case, a charge dissipation layer, a type of conductive polymer, is often incorporated with the resist to minimize the scattering for high pattern fidelity [22, 23, 24].

The sensitivity of electron beam resist is determined by the dose required during exposure, measured as the amount of charges per unit area that ensures the resist film to be dissolved (positive) or retained (negative) in the process of development. It is related to the exposure parameters, i.e., beam current I, dwell time τ and pixel size l, as given by

$$Dose = \frac{I \cdot \tau}{l^2}. \tag{1.1}$$

With the selected parameters, the dose can be calculated and compared to the critical value of the specific resist, and therefore necessary adjustments can be made to achieve proper exposure.

1.2 Pattern Placement Errors

In either raster or vector-scan system, pattern placement is achieve by reliable beam deflection together with precise stage control. Although careful engineering of electron-optical lenses can reduce the deflection distortion caused by aberration, and the laser-controlled stage can be positioned with nanometer resolution, precise pattern placement is still impossible to retain unless other causes of error are addressed. If the sources of the error are static or quasistatic, such as an aberration of the electron-optics, imperfection of interferometer and non-linearity of deflection axes, intermittent calibration procedures can be implemented to minimize such error. As for errors introduced by dynamic sources, such as charging of sample and system, thermal expansion, vibration and stray magnetic field, if frequent calibration were to be performed, the amount of overhead added to exposure time would be prohibitive.

All pattern placement errors can be classified as intrafield or interfiled. The intrafield error refers to that within the write field as depicted in Figure 1.6(a), while the interfield error is what causes the mismatch between adjacent fields as in Figure 1.6(b). Figure 1.7 shows some interfield errors observed in the experiment of exposure.

1.3 Spatial Phase Locked Electron Beam Lithography

Different from the other methods of improving pattern placement precision, in which the efforts are put in directly addressing the sources that cause the error, SPLEBL seeks to “close the loop” by monitoring electron beam location and correct the deviation. As illustrated in Figure 1.8, this approach relies on detecting the secondary electron signal from a fiducial grid placed on the substrate that does not adversely affect the normal exposure. With the periodic grid signal, the e-beam position error

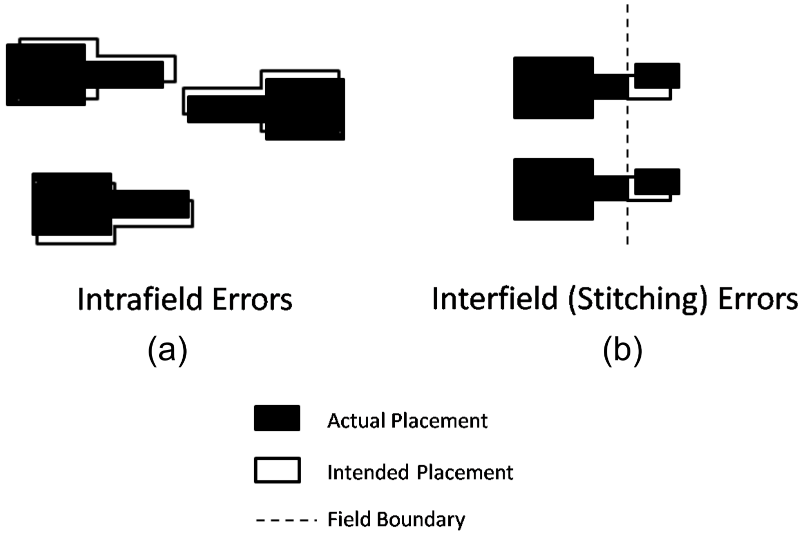


Figure 1.6: Illustration of two types of pattern placement errors.

referenced to the grid is estimated by external electronics, which is then sent back to the beam deflection system to be corrected.

Since the idea was first proposed [25], it has been implemented in different strategies for both raster- and vector-scan EBL. The early implementation employed discontinuous feedback mode for vector-scan system. The fiducial grids or gratings are examined before exposure, then Fourier transform is performed to extract the spatial phase and finally the correction is applied to the beam deflection system during exposure.

If segmented grid/gratings are used, referring to the reference pattern written in the unused portion of write field, highly scattering material are chosen to fabricate the reference pattern and sampling is performed with high dose level. Therefore, precise phase estimation can be achieved with high signal-to-noise ratio (SNR). Integrated-optical devices have been patterned using this technique, and the best result demonstrates the interfield error with 0.3 nm standard deviation [26, 27]. While with global fiducial grid/gratings patterned on the substrate, examining the grid/gratings is conducted at discrete locations within the write field using sub-exposure threshold dose. As a result, phase locking with lower precision is expected due to the reduced SNR. The standard deviation of 5 nm has been achieved with gratings [28], and 12 nm with

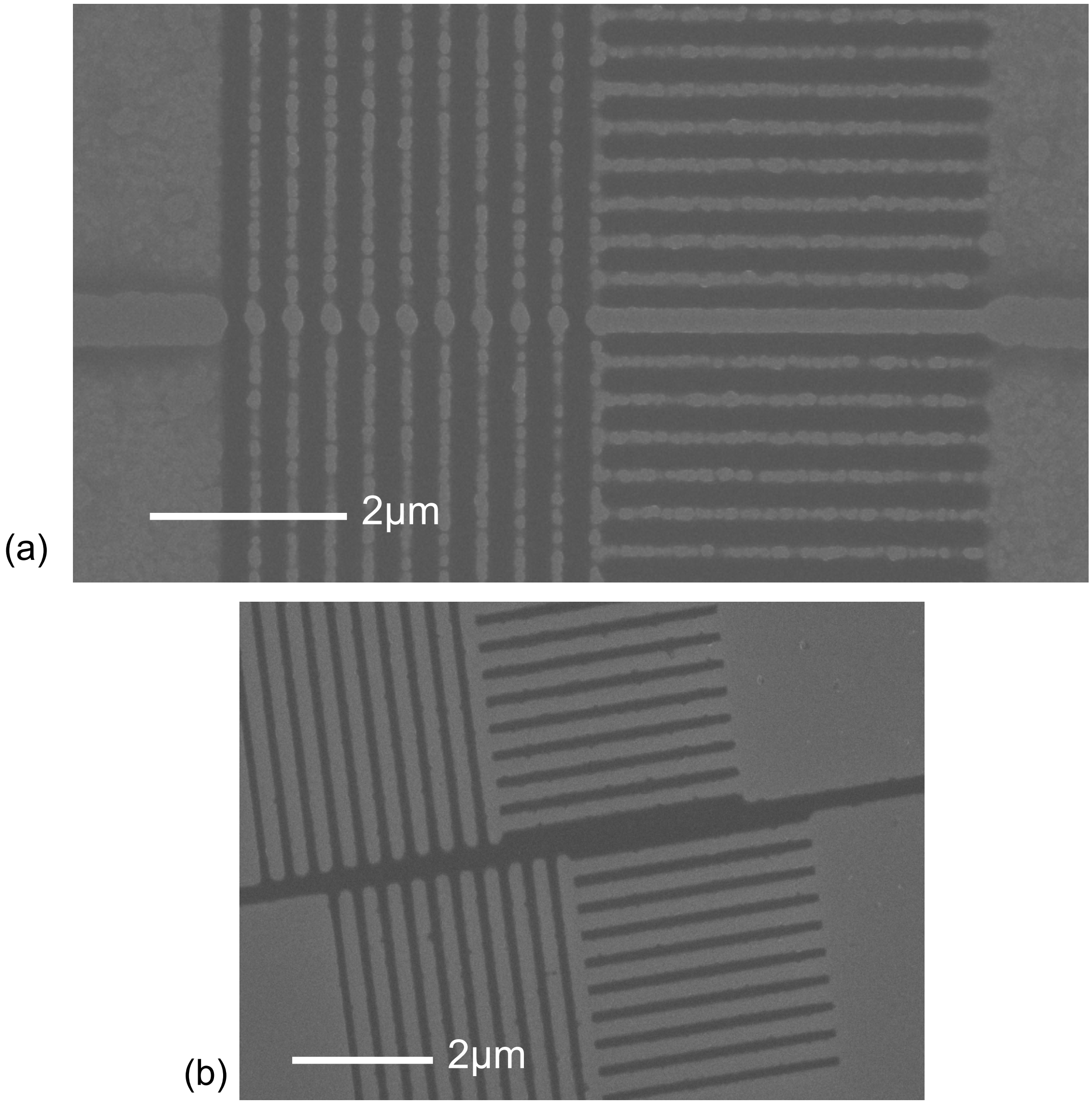


Figure 1.7: Stitching errors at the write field boundary observed in the experiment. (a) Bulges at the boundary due to overlapped exposure. (b) Discontinuity and offset at the field boundary.

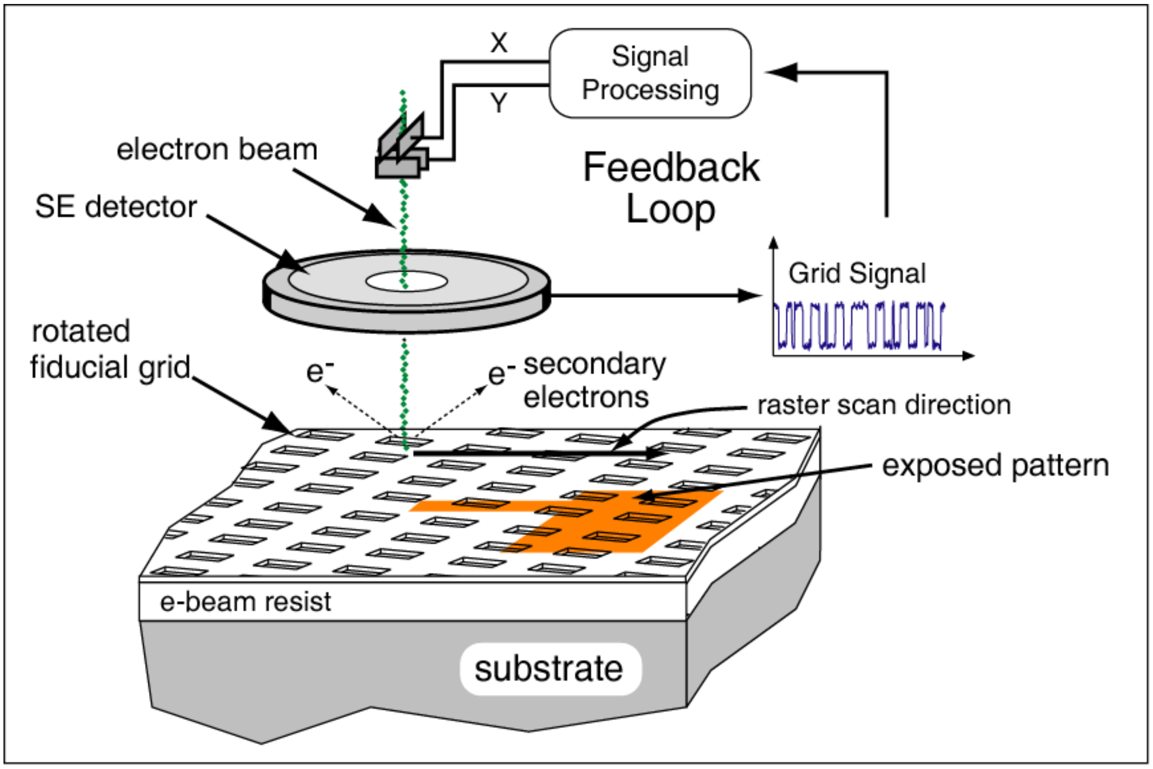


Figure 1.8: Spatial-phase-locked electron-beam lithography (SPLEBL) provides feedback control of pattern placement by estimating the beam position error using detected secondary electron signal from an electron-transparent fiducial grid.

a fiducial grid [29].

The continuous feedback mode has been implemented for raster-scan system. By conducting phase locking with full dose level during exposure and reduced dose while e-beam addressing the unpatterned area, the overhead time required for examining the grid before exposure is greatly reduced, and some rapidly occurring errors are also effectively corrected during exposure. The reported exposure experiment demonstrates sub-1-nm pattern placement precision [5].

1.4 Shot Noise and Dose Variation

To obtain the desired electrical characteristics of devices as the lithographic resolution scales down to sub-100 nm, variation of the critical dimension and the line edge/width roughness (LER/LWR) needs to be reduced accordingly. As stated in ITRS's report, on a 40 nm half pitch mask for 193 nm wavelength single exposure, 3σ variation of 1.7 nm for gate width and 1.9 nm for DRAM half pitch is projected for year 2013 to achieve the critical dimension of 27 nm and 28 nm respectively. The stochastic nature of the exposure process using either electron or ion-beam lithography has profound impact in this regime, and many experiments have demonstrated this effect [30, 31, 32, 33, 34]. Through numerical simulations, the effect of shot noise, resist properties and process conditions on CD control and LER has also been analyzed [35, 36, 37].

Shot noise arises from the random arrival events of electrons. It directly contributes to the dose variation, either pixel-to-pixel or shot-to-shot, and places a lower limit on CD uniformity and LER/LWR. Its statistical property can be described by the model derived from the Poisson distribution [38], implying that the number of electrons delivered to the resist in a fixed interval of time is uncertain. This statistical variation is often averaged out by exposing less sensitive resist with extended time. However, in the case of low dose exposure, such as writing densely packed high-resolution lines, the shot noise induced geometrical or size variation becomes more prominent. Also, if more sensitive resist is used to achieve high throughput, the same effect can be observed due to significantly reduced exposure time. This suggests a tradeoff between throughput and statistical consideration.

1.5 Exposure Dose control for Electron Beam Lithography

Currently, for single pixel/shot exposure, the desired dose is obtained by measuring the beam current density and setting the dwell time. The current measurement is taken before exposure and can be checked periodically during exposure to correct for slow drifts. Otherwise, this technique assumes that the beam current is perfectly stable during exposure and neglects the inherently stochastic properties of e-beam sources. A method that reduces variations in electron dose and ultimately breaks through the shot noise limit would be an important improvement to both mask writing and emerging maskless technologies.

Previous efforts to provide feedback dose control measure the current from a portion of the beam blocked by an aperture in the electron optical column. This technique has been proved to improve exposure throughput for lithography and sensitivity for inspection [39, 40, 41, 42]; however, the statistical variation of electrons arriving at the sample cannot be addressed in this approach. Control of current at the emitter by integrating a control circuit with the field emission source has also been investigated [43, 44, 45, 46, 47, 48]. This technique is effective for correcting flicker noise from field emission source and for detecting high frequency beam current fluctuation, but does not provide the capability of detecting the number of electrons impinging the substrate to compensate the shot noise.

Single-ion control for ion-beam based system has been developed by either detecting the burst of secondary electrons from each ion impact [49] or counting ions passing through an opening pore on the substrate with a single-ion detector on the back side of the sample [50, 51, 52]. The former approach is impractical for EBL because SE yield is typically less than one and thus the shot-noise associated with SE production dominates the detected signal.

A new approach, that provides real-time feedback control of dose for each exposed pixel based on the signal from the sample to be patterned, will be described in Chapter 6. This strategy monitors the number of electrons deposited in the resist during exposure by introducing a layer of scintillating material on the substrate. The photon signal is detected and used to estimate the current exposure status at real-time. Once the desired dose level is achieved, a feedback signal is sent back to EBL to terminate the exposure. With direct monitoring electrons delivered to the resist, this approach holds the potential to ultimately overcome the shot noise.

Chapter 2

Fiducial Grid and Interference Lithography

2.1 Fiducial Grid

The fiducial grid plays a key role in Spatial-Phase Locked Electron Beam Lithography. It provides the absolute reference with respect to which the e-beam position is determined, thus a fiducial grid with high quality detectable signal offers a good foundation for beam position control. A good fiducial grid should meet the following requirements: (1) The fabrication process should be simple and the grid should not interfere with the normal EBL procedure. (2) The grid should not worsen the CD control of the e-beam writer significantly, i.e. extra electron scattering due to the introduction of the fiducial grid should be minimized; (3) The quality of the grid signal should allow to obtain broad bandwidth performance in real-time system. (4) The pattern should possess spatial coherence over the entire area to be patterned.

As an electron impinges on a material, electrons or/and photons are produced retaining different energies. According to the type of the detected particle and the energy it carries, the fiducial grid can be categorized into three groups: backscattered electron grid, scintillating grid and secondary electron grid.

Backscattering happens when an electron experiences the elastic interaction with the nucleus of an atom, and comes back without losing much kinetic energy. As the size of the atom nucleus increases, the probability of producing elastic collision increases. Therefore, elements with large atomic number are always chosen to make the backscattered electron grid in order to produce high contrast signal. Due to the fact that high-metallic-number metal causes strong scattering and reduces penetration of the primary electrons, which results in deteriorated pattern resolution, a backscattered electron grid is not ideal for real-time position control. However, since such grid yields high SNR signal, it is always a good candidate for algorithm demonstration or evaluation, such as the Au fiducial gratings in references [53, 28], and 1 μm Au fiducial grid in [29].

A scintillating fiducial grid is fabricated by incorporating the scintillator into the e-beam resist, and photons are detected as the grid signal for SPLEBL [54, 29, 55]. Because of the scintillating component, the resolution and sensitivity of the e-beam resist are greatly compromised, and the grid is expected to be damaged quickly upon exposure to the e-beam. Furthermore, the SNR reduces as the grid period scales down, thus spatial phase locking with broad bandwidth is usually unattainable if it is performed on such grid.

The secondary electron fiducial grid relies on the difference of the secondary electron yield between the grid and the underlying material. Low atomic-number metal is often chosen to form the grid due to the benefits of (1) high secondary-electron yield, (2) easy evaporation, (3) preventing the sample from charging and thus providing stable SE signal [56]. Because of its low energy, usually less than 5 eV, the secondary electron does not escape from the surface unless originated within the depth of a few nanometers; thus, only a very thin layer of metal ($<$ 10 nm) needs to be evaporated on the very top of the stack. To improve the contrast of the SE signal, a layer of SiO_x, which has low SE yield, is usually put in between the e-beam resist and the metallic grid. A 8 nm-thick Al lattice was used to achieve sub-nm pattern placement precision for raster-scan EBL [5].

In the effort to improve the throughput and reduce the cost of EBL, e-beam microcolumn lithography was introduced in late 1980s [16, 57], which led to the development of the arrayed e-beam microcolumn lithography system for parallel patterning [58, 18, 19]. Low primary e-beam energies are commonly used in the microcolumn system, typically between 1 KeV and 5 KeV. This brings in other advantages like reduced proximity effect from backscattering and fast exposure on resist of high sensitivity. To explore the potentials of combining SPLEBL with the e-beam microcolumn system, fiducial grids that generate SE signal with sufficient contrast at low primary e-beam energy were investigated [59, 60]. By microcontact printing a very thin layer of organic compound onto 5 nm-think Al coated polymethyl methacrylate (PMMA), a 400 nm period grid was fabricated, and the SNR was measured to be 0.4 at primary beam energy of 2 KeV [60].

2.2 Interference Lithography Overview

Interference Lithography (IL) is the standard technique to fabricate the fiducial grids and gratings. The interference of two coherent light waves produces the standing wave, and the alternating maximum and minimum light intensities cause the variation of exposure dosage in the light-sensitive material. A 1-dimensional (1D) periodic pattern is obtained in a single flash, while 2D pattern involves rotating substrate by certain angle between exposures. To obtain high coherent illumination, laser is always chosen as the light source, which is split to two beams and then directed to the sample to form interference.

One of the common IL setups is modified from Mach-Zehnder interferometer as depicted in Figure 2.1. Because of the imperfect or damaged optics, and the variation in the laser gain medium itself, unwanted varying intensity noise at the output of a laser system is often inevitable. Therefore, a spatial filter needs to be incorporated to filter the high spatial-frequency noise to produce a “clean” laser beam with Gaussian intensity distribution. The spatial filter is typically composed of an aspheric lens to focus the beam and an aperture, known as “pinhole”, to allow only the fundamental Gaussian mode to pass through.

Since the pinhole corresponds to a good approximation of a point source, the light wavefront on the flat substrate can be approximated as spherical. The spherical

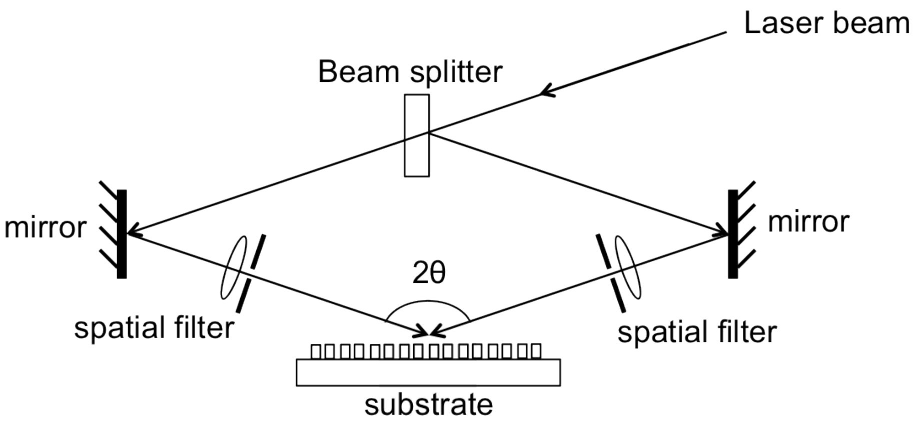


Figure 2.1: Schematics of interference lithography in Mach-Zehnder style. The laser beam is split into two arms which ideally should be symmetrical about an axis normal to the substrate.

wave interference produces gratings with non-linear spatial phase progression which was modeled and measured in reference [61]. Although using collimating lens after the spatial filter can planarize the wavefront to achieve almost linear spatial phase progression over exposed area, the dose uniformity is greatly compromised due to the imperfection and scattering of the lens, as well as the area of exposed region due to the limited lens size. Another method to reduce the hyperbolic phase effect, proposed by Walsh and Smith, is to conduct the exposure on spherical concave surface [62]. However, the implementation of more complex spherical surface correction requires highly elastic substrate, which prevents this technique from achieving the optimum result.

Scanning-beam interference lithography was developed to provide a different approach to address the hyperbolic-phase progression issue by bringing two collimated beams of small radius to form a standing wave image. By scanning the substrate under the image at a constant velocity in one direction, and stepping the stage between scanning lines, one can obtain much larger area of interference gratings with minimized hyperbolic phase effect [63, 64]. To ensure good stitching, The stage steps over an integer number of fringe periods between the adjacent scans, as depicted in Figure 2.2. Although linear spatial-phase and low distortion is almost guaranteed

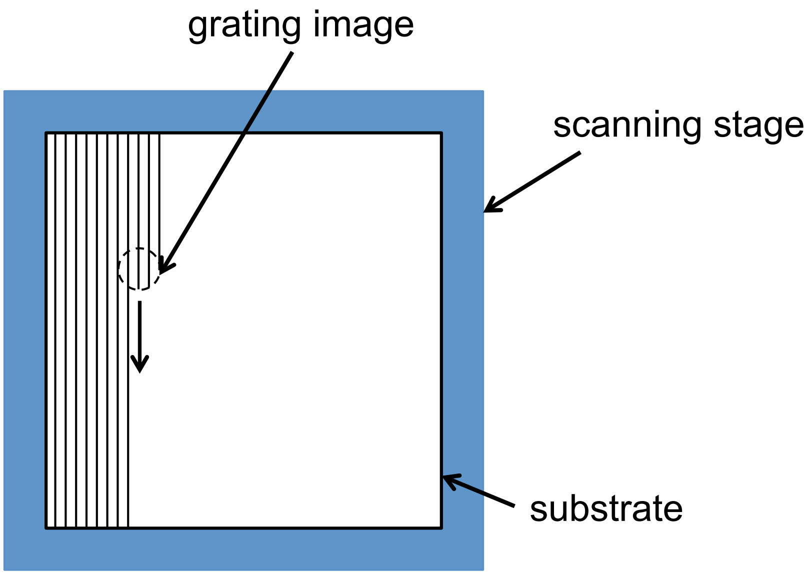


Figure 2.2: Schematics of scanning beam interference lithography. The long range fringes are formed by moving the stage under the interference image, and the adjacent lines are overlapped by integer number of fringe periods to ensure good stitching.

within each "small" grating image, phase-locking the gratings to a moving substrate requires a great deal of effort.

A rather simpler setup of interference lithography is illustrated as in Figure 2.3, known as the Lloyd's mirror interferometer. The mirror is perpendicular to the sample to be exposed, and the interference of the incident light directly from the source and the light reflected off the mirror forms the standing wave patterns on the sample. As the mirror serves as the second light source, its quality, such as flatness, cleanness and reflectivity, is critical to ensure the desired pattern, as the defect of the mirror will result in the pattern distortion or/and poor contrast.

To obtain the maximum area of grids or gratings, one can expand the beam by placing the rotation stage as far away from the spatial filter as the optical bench allows, but longer exposure time is expected due to the reduced light intensity on the sample. There are other benefits associated to the enlarged beam diameter, such as more uniform intensity distribution over the exposed area as well as the approximation to planar wave on the surface of the sample. As the diameter of the beam spot increases, the size of the mirror should be scaled up accordingly, so that

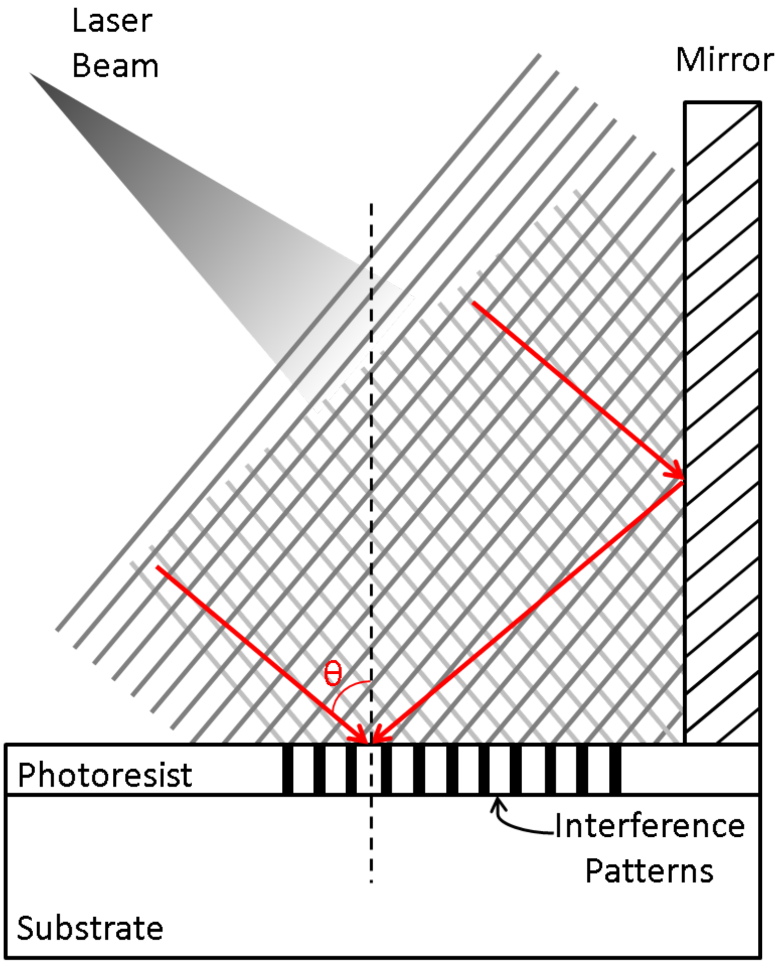


Figure 2.3: Schematics of interference lithography in Lloyd's mirror configuration. The periodic patterns are produced by the incident light on the sample interfering with the light reflected from the mirror.

the effect of the scattering from the edges of the mirror on the exposed pattern is eliminated.

For both Mach-Zehnder and Lloyd's mirror IL setups described above, the period of the gratings P is determined by the angle of incident light θ on the sample and the wavelength of light source λ:

$$P = \frac{\lambda}{2\sin\theta}. \tag{2.1}$$

Once the laser source is chosen, the incident angle is adjusted to obtain the interference pattern with the desired period. The simplicity of the Lloyd's mirror style allows for easy adjustment by rotating the interferometer, while physically moving and realigning the optics in both arms is involved for Mach-Zehnder's style.

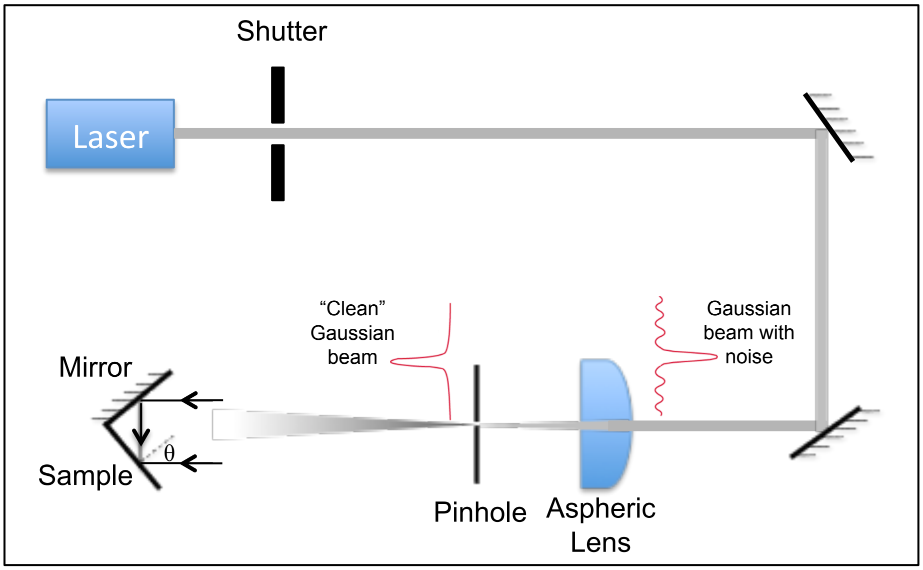
Extreme ultraviolet (EUV) laser is also incorporated with the traditional interference lithography to fabricate patterns with 100 nm period or less. The most recently reported works demonstrate that gratings with period of 100 nm and feature size of sub-20 nm [65], and also that with period of sub-50 nm [66] were successfully fabricated using 13.4 nm wavelength laser.

Although the pattern period shrinks as light wavelength reduces, it is still challenging to achieve long range phase coherence as EUV laser with high spatial or/and temporal coherence is not readily available. To overcome this problem, achromatic interference lithography (AIL) was developed, which employs transmission phase gratings for beam splitting and recombining [67, 68, 69, 70]. In this case, the period of the interference pattern is independent of the wavelength of the light, rather relies on that of the parent gratings, resulting in the challenge of fabricating high quality transmission phase gratings.

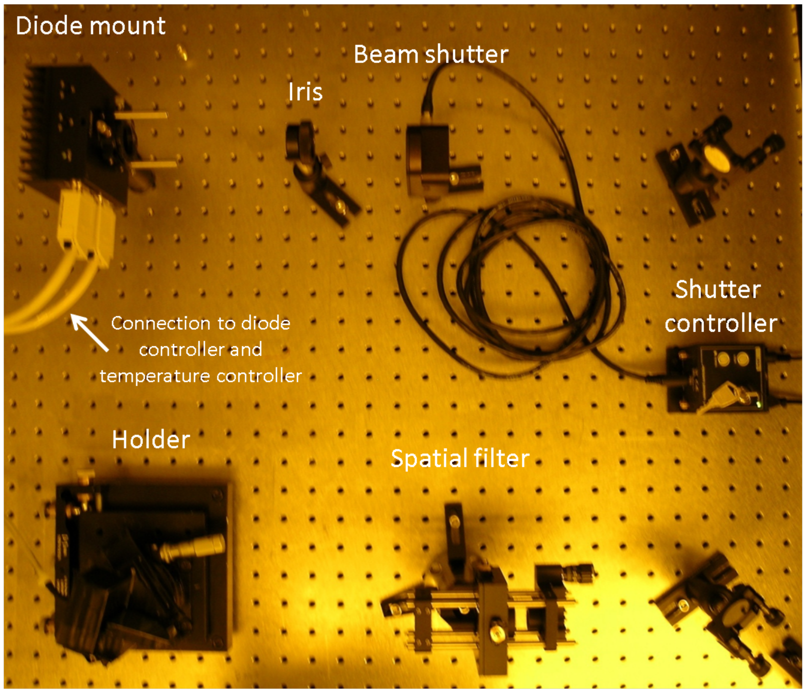
2.3 Implementation of interference lithography at UK

We chose the Lloyd's mirror style interference lithography system to be implemented at Center for Nanoscale Science and Engineering. All components, including laser source, spatial filter, rotation stage and other optical elements, are situated on an optical bench in the fluorescent light filtered cleanroom as shown in Figure 2.4(b). The light is redirected by two mirrors to achieve the maximum beam expansion for the available space accommodated by the optical bench.

Spatial filter, which in our implementation consists of a positive lens with focal length of F and a pinhole with diameter of D_P, is to minimize the spatial intensity noise in a laser beam of a Gaussian profile as illustrated in Figure 2.5. Passing the Gaussian beam through the converging lens is analogous to performing the Fourier transform in the domain of spatial frequency along radial axis, thereby the image



(a)



(b)

Figure 2.4: Interference lithography setup in Lloyds mirror configuration. (a) Schematics of the experimental setup. (b) The experimental setup on a vibration isolated optical table in the cleanroom. The light source shown here is a 405 nm wavelength laser diode.

of the imperfect Gaussian beam formed at the lens' focal point is also a Gaussian distribution but with fringes on the sides, which corresponds to the intensity noise at high spatial frequency. By placing a pinhole with the appropriate diameter size, the fringes can be effectively blocked. The diffraction limited spot size D at the 99% contour of a perfect Gaussian beam is given by

$$D = \frac{F \cdot \lambda}{\tau}, \tag{2.2}$$

where F is the focal length of the lens, λ is the wavelength of the laser and τ is the radius of laser beam. The pinhole's diameter D_p should be no smaller than this value to ensure most of the beam energy passes through but noise is eliminated. The fraction of power passed by the pinhole can be calculated by [71]

$$1 - \exp\big[-\frac{1}{2}(\frac{\pi\tau D_p}{\lambda F})^2\big]. \tag{2.3}$$

For easy alignment, the lens is held by a Z-transistor and the pinhole is mounted in a XY-translator. The Z-translator allows to adjust the distance between the lens and the pinhole to the focal length, and XY-translator to center the focused beam on the pinhole, after which the power is measured by the optical power meter.

A vacuum-backed holder was designed with two plates, as shown schematically in Figure 2.6, for the sample and the mirror respectively. Holes are drilled on the sample plate so that suction can be used to hold firmly both the sample and the edge of the mirror plate during exposure. Due to the heating from laser cutter, a beveled edge on the mirror plate was formed, which prevents the two plates from being perpendicular. The small angular misalignment will lead to a slight discrepancy between the period of exposed pattern and the expected value, which is, however, independent of the light incident angle [72]. The holder is mounted on a rotation base in a way that the intersection of the two plates is oriented along the base's axis of rotation, which prevents the error of pattern's period from varying with the light incident angle. Detailed analysis on the effect of rotational axis misalignment is reported in reference [72].

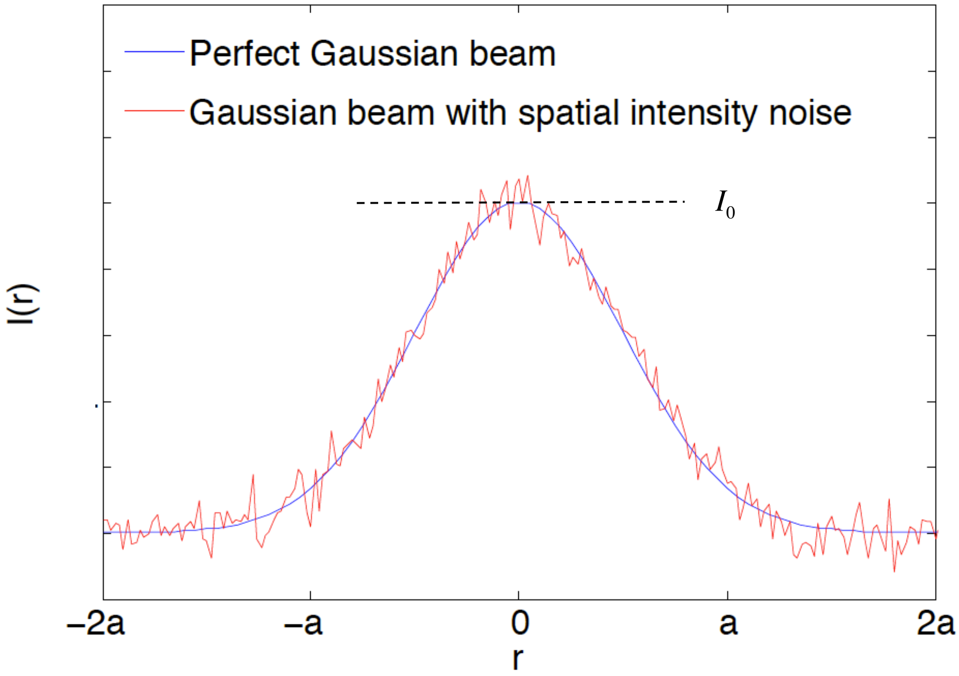


Figure 2.5: Intensity profile of a perfect (blue) and a noisy (red) Gaussian beam versus the radial distance from the center of the beam. r represents the radial distance from the center of the beam. a is the radius at which the light intensity drops to $1/e^2$ of I_0.

2.4 Lithography Experiment

2.4.1 Patterning with 266 nm wavelength laser

The Coherent Azure deep-UV laser system is used to output a 266 nm wavelength Gaussian beam, and the incident light intensity at the sample was measured to be 3.2 mW/cm^2. To pattern 200 nm period gratings, the stage was rotated so that the incident angle θ equals 42°.

The interference lithography experiment was first conducted with positive photoresist Shipley 1813. The resist was diluted (Shipley 1813 : Thinner = 1:5) before it was spin coated on the Si substrate, so that a very thin layer of coating can be obtained to reduce the standing-wave effect and to prevent collapse of high-aspect-ratio fringes during development. A 10-second exposure was performed on 100 nm thick

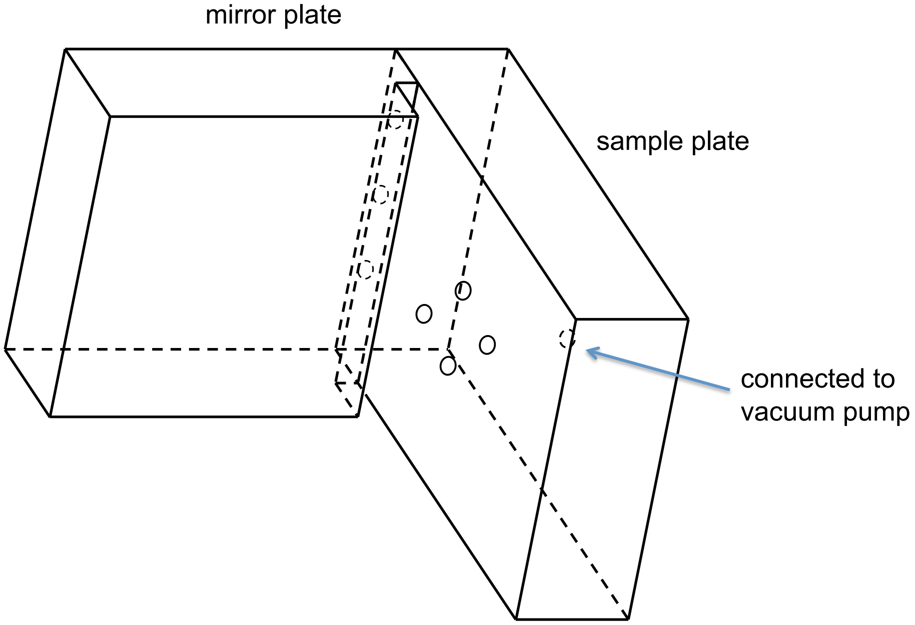


Figure 2.6: Schematics of the mirror/sample holder for interference lithography. When the vacuum pump is activated, the mirror plate and the sample are held firmly against the sample plate by suction.

resist, which was followed by development in the mixture of MF-319 and deionized (DI) water (MF-319 : DI water = 1:4). The secondary electron microscopy (SEM) image of the gratings, as shown in Figure 2.7, indicates the resist's poor absorption contrast between the exposed and unexposed region at light wavelengths shorter than 300 nm.

A positive resist PEK-130A6 with higher contrast at short light wavelengths was used to pattern using the same light source. The adhesion promoter hexamethyldisilazane (HMDS) was applied to the substrate before spin coating a layer of 100 nm thick resist, which is diluted by 2-methylbutyl acetate to a solution of 50% by mass. Then the sample was baked at 110° for 30 s before a 15 s exposure, which was followed by post-exposure baking at 90° for 30 s. Finally, the sample was developed in a solution of MIF-319 and DI water (1:1) for 40 s. The obtained gratings are well defined as shown in the SEM images in Figure 2.8, which promises a fiducial grid with good SNR for spatial-phase locking if the experiment is continued in this direction. However, due to the unexpected issue with the laser system, we were forced to pursue

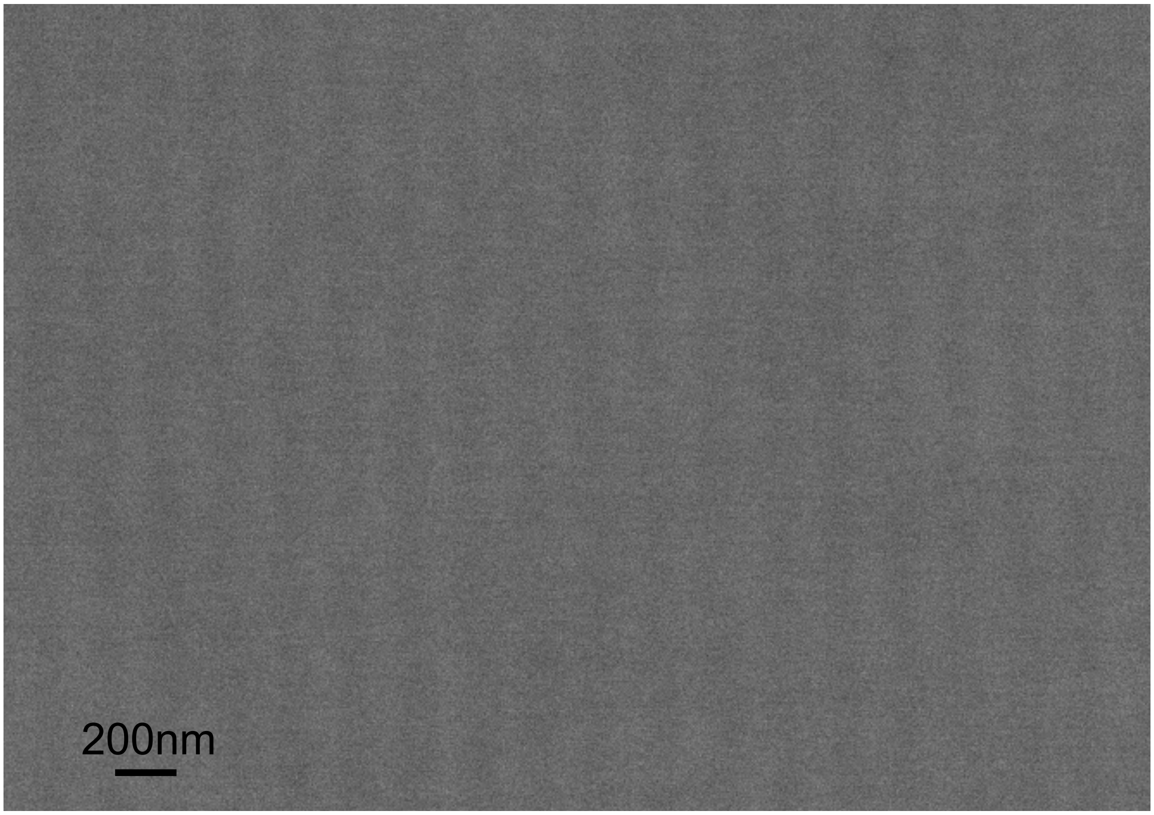


Figure 2.7: SEM images of the gratings patterned on 100 nm thick Shipley 1813 resist using 266 nm wavelength laser.

the investigation using a different laser source.

2.4.2 Patterning with 405 nm wavelength laser

The Ondaxs 405 nm wavelength laser diode was used to perform the IL experiment on photoresist Shipley 1813. The laser diode outputs a collimated single frequency elliptical beam with size of 0.8 mm × 0.4 mm. The laser diode was mounted in the Thorlabs TCLDM9 diode mount, and controlled by Thorlabs laser diode controller TLD001 and temperature controller TTC001. Since the laser diode is not paired with a photodiode (PD) in the package, the PD pin on the mount was grounded and the constant current mode on the controller was selected to drive the laser diode with a current of 40.85 mA. The light intensity at the sample was measured to be 1.9 mW/cm^2. The experimental setup with the laser diode is shown in Figure 2.4(b).

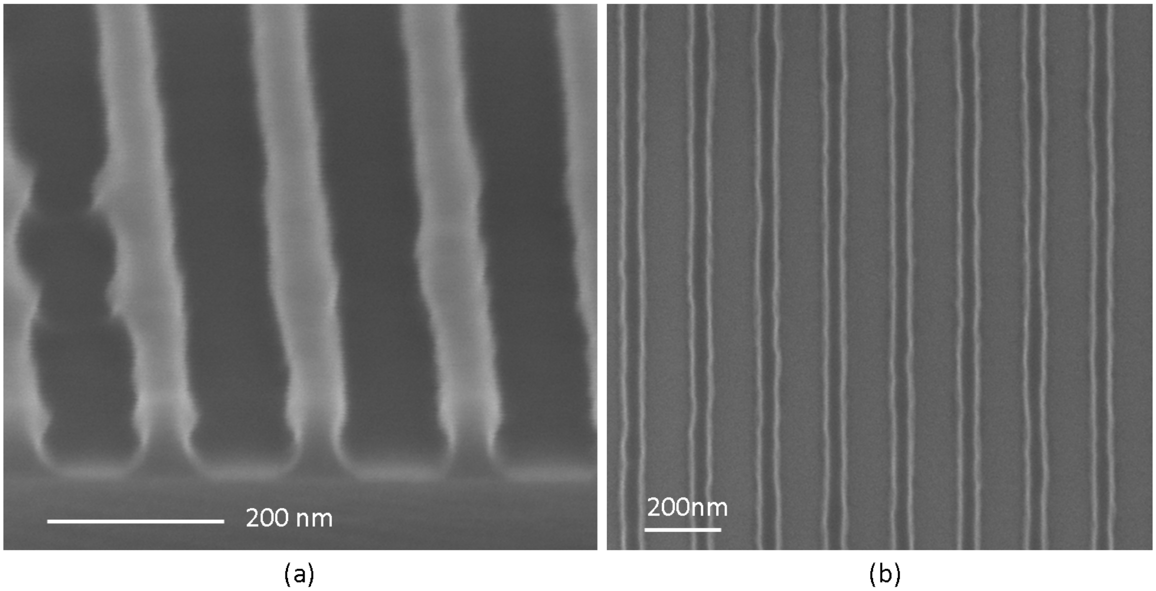


Figure 2.8: SEM images of the gratings patterned on PEK-130A6 using 266 nm wavelength laser. The period of the gratings is slightly less than 200 nm. This is very likely caused by the angular misalignment of the mirror due to the curved cutting edge of the acrylic plastic sheet from heating by the laser cutter. (a) Cross-sectional view of the gratings. (b) Top view of the gratings.

Figure 2.9 shows the SEM images of 250 nm period gratings exposed on 45 nm thick resist for 50 s at light incident angle θ of 54°. HMDS was coated on the Si substrate before applying resist to promote adhesion. Although the absorption of the resist has good contrast between exposed and unexposed region at this wavelength, the gratings are hardly resolved by the developer. The cross-sectional view reveals a smooth transition in the thickness of the resist between exposed and unexposed region. This may be due to the limited resolution offered by Shipley 1800 series photoresist, which is designed mostly suitable to pattern features of micrometer size.

To verify the hypothesis, the incident angel θ was adjusted to 18° to pattern 1 μm period gratings on the same resist. The exposure was carried out for 42 s. The gratings are as seen in the top view SEM image in Figure 2.10(c). However, the cross-sectional view, as shown in Figure 2.10(a) and (b), reveals that the ridges as wide as 200 nm with gentle slopes formed in the unexposed region, suggesting that the severe

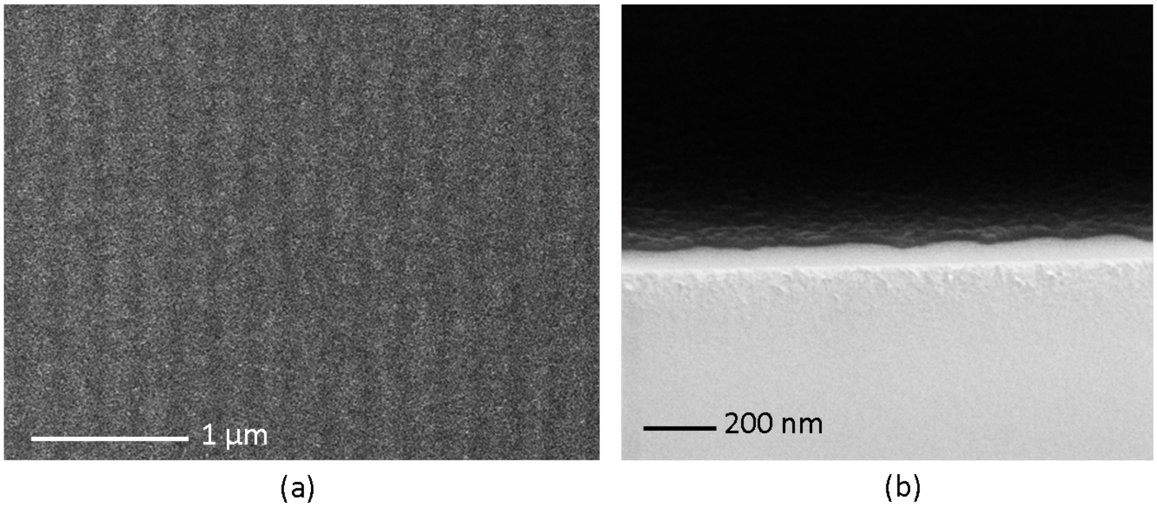


Figure 2.9: SEM images of gratings patterned on 45 nm thick Shipley 1813 using 405 nm wavelength laser. The period of the gratings is about 250 nm as expected. (a) Top view image of the gratings. (b) Cross-sectional view of the gratgins.

line edge roughness will prevent finer gratings from being resolved. Although, one can use high normality developer to improve signal-to-noise ratio by dissolving the residual resist in the trench while maintaining the profile of the unexposed region, the properties of the resist still set the lower limit on the pattern's resolution, as well as the period of dense features.

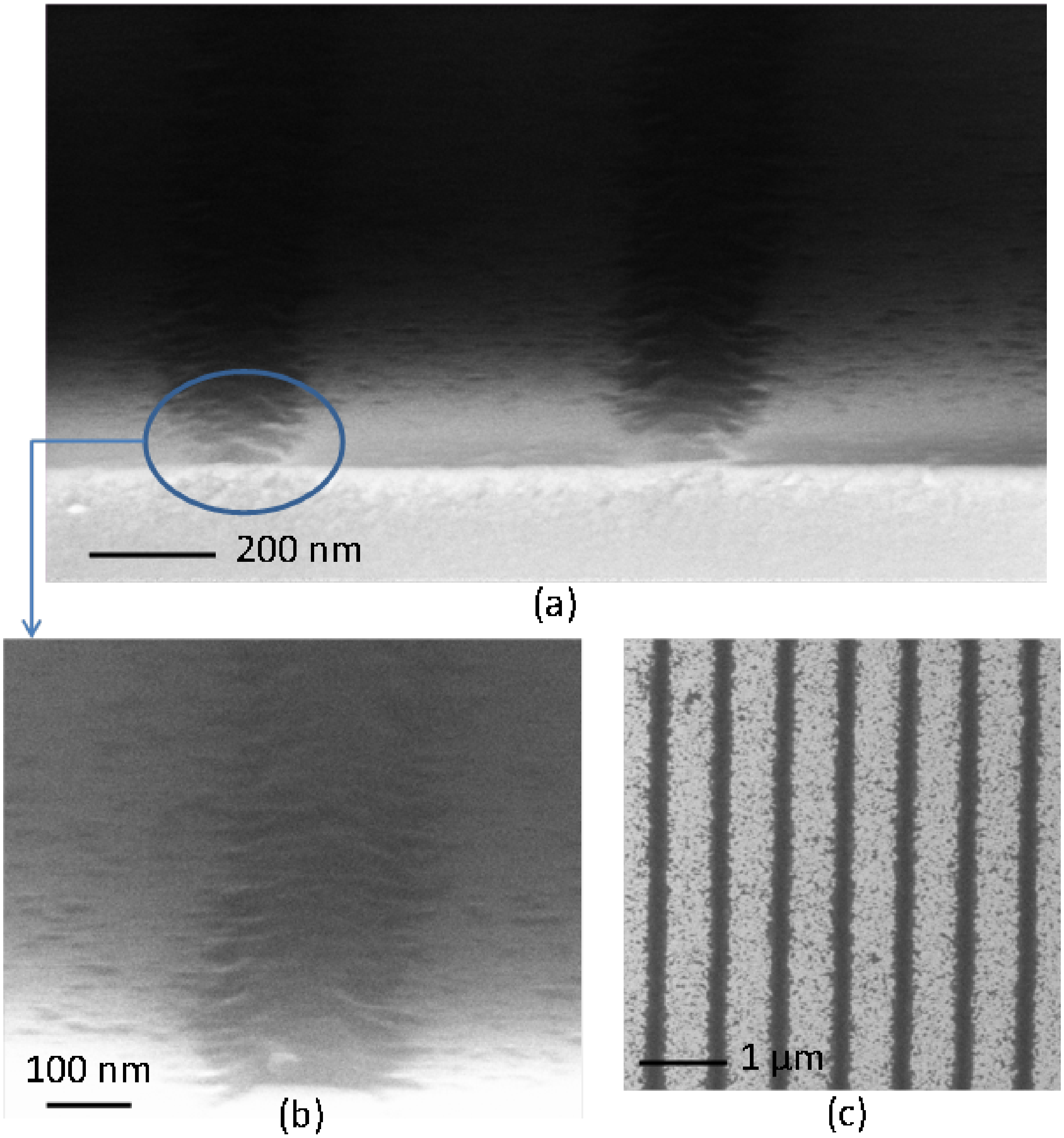


Figure 2.10: SEM images of 1 μm period gratings patterned on 45 nm Shipley 1813 with 405 nm wavelength laser. (a) Cross-sectional view. (b) Zoomed-in cross-sectional view on one fringe. (c) Top view.

Chapter 3

Spatial Phase Locking for Vector-scan Electron Beam Lithography

3.1 Phase Estimation from Discrete Signal

The Spatial Phase Locked Electron Beam Lithography is based on the well established method used for estimating the parameters of a periodic signal, such as frequency, phase and amplitude [73, 74, 75, 76]. The performance of the estimator with the available data, known as Cramer-Rao (CR) bound, has also been investigated to provide a design benchmark.

Considering a continuous-time single-tone periodic signal in the presence of a noise ν with zero mean and variance σ^2 described as:

$$s(t) = a\cos(\omega t + \phi) + \nu[0; \sigma^2], \tag{3.1}$$

in which a is the amplitude, ω is the angular frequency and ϕ is the phase. If this signal is sampled at a constant sampling frequency of $1/T$ with the first sample taken at $t = t_0 = n_0 T$, the discrete-time representation of this signal with N samples is given by:

$$s[n] = a\cos(\omega n_0 T + \omega n T + \phi) + \nu\left[0; \sigma^2\right], \qquad n = 0, 1, \ldots, N-1. \tag{3.2}$$

According to the maximum-likelihood (ML) estimation algorithm described in reference [75], the estimated phase $\hat{\phi}$ is given by:

$$\hat{\phi} = \angle\Big[\exp(-j\hat{\omega}t_0)A(\hat{\omega})\Big], \tag{3.3}$$

where $\hat{\omega}$ is the estimated frequency in the interval $[-\pi, \pi]$ at where the maximum magnitude of

$$A(\omega) = \frac{1}{N}\sum_{n=0}^{N-1} s[n]\exp(-j\omega nT) \tag{3.4}$$

occurs.

The same estimation algorithm can be applied to perform on the spatially periodic signal, such as the secondary electron signal from the fiducial grid used in SPLEBL, to estimate the position shift. For a one-dimensional single-tone periodic signal with period of λ, the angular spatial frequency is defined as $k = 2\pi/\lambda$. The discrete signal sampled with sampling space of δx is given by

$$sig[n] = a\cos(k\cdot\delta x\cdot n + \phi_x) + \nu\left[0;\sigma^2\right], \qquad n = 0, 1, \ldots, N-1, \tag{3.5}$$

in which the first sample is assumed to be taken at location $x = 0$. If the spatial frequency is known or can be measured beforehand, the spatial phase can be calculated by applying $k = 2\pi/\lambda$ in the formula, thus:

$$\hat{\phi_x} = \angle\left[\sum_{n=0}^{N-1} sig[n]\exp(-jk\cdot\delta x\cdot n)\right]. \tag{3.6}$$

By introducing the intended position x_P, estimated position x_B and position error Δx, the phase estimate $\hat{\phi_x}$ can be related to Δx as

$$x_B = x_P + \Delta x = \frac{\lambda}{2\pi}\hat{\phi_x}. \tag{3.7}$$

If one starts sampling at where $\phi_x = 0$, equation 3.7 can be simplified as

$$\Delta x = \frac{\lambda}{2\pi}\hat{\phi_x} = \frac{\hat{\phi_x}}{k}. \tag{3.8}$$

Although such sampling condition is not always guaranteed during the exposure with SPLEBL, the initial beam position shift x_P, which is sent back to the beam deflection system as part of the estimated position, only causes an overall pattern offset that is smaller than half period of the reference pattern.

With the estimation strategy established above, it is important to understand the best achievable performance under the specific circumstances. Based on the Rife and Boorstyn's investigation on Cramer-Rao bound of the ML estimator [75], Krishnamurthy, Namepalli and Hastings [77] derived the lower bound of the variance of the estimated spatial phase with known frequency as given by

$$var\left\{\hat{\phi_x}\right\} \geq \frac{2\sigma^2}{a^2N} = \frac{2}{\gamma N}, \tag{3.9}$$

where γ is the signal-to-noise ratio (SNR) defined as $\gamma = a^2/\sigma^2$. Therefore, the variance of position error estimate can be calculated according to

$$var\left\{\Delta x\right\} = \frac{1}{k^2} \cdot var\left\{\hat{\phi_x}\right\} = \frac{\lambda^2}{4\pi^2}\frac{2}{\gamma N}. \tag{3.10}$$

It is known that the non-linear estimator suffers the threshold effect at low SNR, which is referred to as the threshold region, and the variance of the estimate parameter increases dramatically. In the region where the variance follows the CR bound and decreases as SNR improves, the estimation system is considered to be limited by the noise. The threshold is determined by calculating the probability of the occurrence of outliers, and the detailed calculation and simulation results can be found in references [78, 75, 76]. Besides, the performance of the spatial-phase estimator for SPLEBL in a large range of SNRs was evaluated by conducting Monte Carlo simulation [77], and it is suggested that operating spatial phase estimation at SNRs below the threshold should be avoided as the phase cannot be determined within the range of $[-\pi, \pi]$, resulting in uncertainty of beam position with respect to the grid.

According to equation 3.10, the performance of spatial-phase estimator can be improved by (1) using finer fiducial pattern, (2) estimating over more samples and (3) increasing the signal's SNR. Since the variance of estimated position errors is proportional to λ^2, reducing the period of the reference pattern has the greatest effect. However, a compromise often has to be made between the grid period and SNR acceptable for rapid phase locking. Besides, the fabrication conditions prevent the period from being reduced arbitrarily. On the other hand, increasing the number of samples reduces the bandwidth of spatial-phase locking in a real-time system, and the available hardware resources can restrain this approach from being fully implemented.

3.2 Vector-scan Spatial-phase Locking Algorithm

Spatial-phase locked Electron Beam Lithography demonstrated sub-1 nm (one standard deviation) pattern placement precision with respect to a 246 nm period Al fiducial grid using raster-scan exposure strategy on a Gaussian beam tool [5]. However, SPLEBL for vector-scan system has been limited to "look-then-writ" modes, in which the grid is imaged, the beam deflection is corrected, and then the pattern is written open loop [53, 26, 29, 55, 79, 80]. This approach has produced excellent results for patterning gratings, but requires the system to be stable while writing. In addition, the discontinuous control of beam position does not allow for correction of deflection dependent placement errors, and the throughput is significantly reduced due to the overhead time required to examine the grid before exposure. Therefore, it will be advantageous to provide real-time feedback control for the vector-scan system.

In this section, a new spatial-phase locking algorithm that provides real-time feedback for vector-scan EBL using a global, electron-transparent fiducial grid is evaluated. During a vector-scan exposure, the grid signal is sampled in a complicated, but deterministic sequence. Thus, with an appropriate signal processing algorithm, one can extract the beam position information from this signal.

The noise-free fiducial grid with period λ_G is modeled as sum of two spatially orthogonal sinusoids to which the Heaviside step function has been applied to create a binary signal. The spatial frequency of the grid is $k_0 = 2\pi/\lambda_G$ in radians per unit length. The grid is rotated by angle θ with respect to the beam deflection axes. By rotating the grid, information about both the x- and y-placement errors can be obtained more rapidly than if the grid is orthogonal to the beam deflection axes. In this case, the 2D grid signal S can be described as

$$S\left(x_B, y_B\right) = H\Big\{\cos\left[k_0\left(x_B\cos\theta + y_B\sin\theta\right)\right] + \cos\left[k_0\left(y_B\cos\theta - x_B\sin\theta\right)\right] + C\Big\}, \tag{3.11}$$

where x_B and y_B are the coordinates of actual beam position. The heaviside step function H is defined by

$$H(z) = \begin{cases} 1, & z > 0 \\ 0, & z < 0. \end{cases} \tag{3.12}$$

The constant C varies between -2 and 2 to create various grids such as an array of bright dots($C < 0$), a checkerboard ($C = 0$), or an array of dark dots ($C > 0$). A simulated, noise-free grid signal is shown in Figure 3.1(a). The pattern does not appear binary because it has been sampled by a Gaussian beam of finite radius.

Figure 3.1(b) shows the two fundamental spatial frequency components (k_{x1}, k_{y1}) and (k_{x2}, k_{y2}) in 2D Fourier space. The phases of these components contain information about the x- and y-positions of electron beam with respect to the grid. In addition, the phases of higher order spatial frequency components could be used to increase position estimation precision, but this has not yet been implemented. If the grid axes are orthogonal to the deflection axes of the SEBL system, then the fundamental spatial frequencies lie along the principal spatial frequency axes. if the grid is rotated with respect to the deflection axes, as shown in Figure 3.1(a), then the spatial frequency components, $(k_{x1}, k_{y1}) = (k_0\cos\theta, k_0\sin\theta)$ and $(k_{x2}, k_{y2}) = (-k_0\sin\theta, k_0\cos\theta)$, are rotated by the same angle. If we assume additive Gaussian white noise, then the signal-to-noise ratio (SNR) of the grid is given by a^2/σ^2, where a is the amplitude of grid signal at the fundamental frequency components and σ^2 is the variance of the noise component.

Define x_P and y_P the intended beam position and the two coordinates are related by position error Δx and Δy such that

$$\begin{aligned} x_B &= x_P + \Delta x \\ y_B &= y_P + \Delta y. \end{aligned} \tag{3.13}$$

The phase of each fundamental frequency component can be determined by substituting equation 3.13 into equation 3.11 and taking 2D Fourier transform with

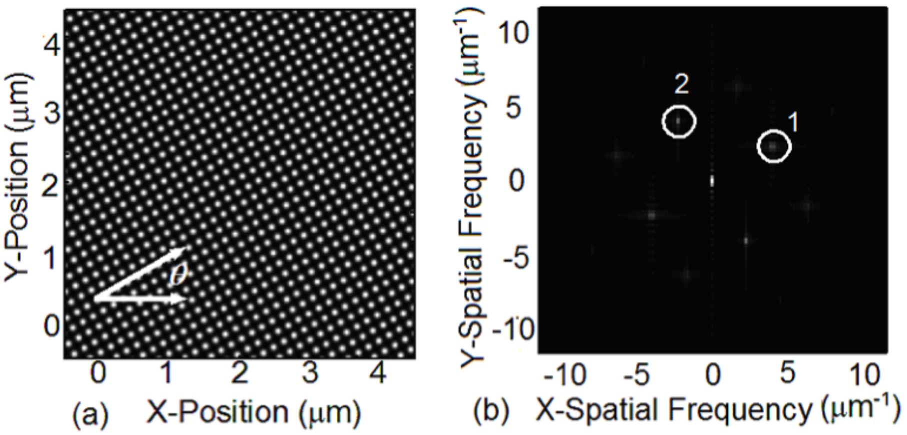


Figure 3.1: (a) Simulated 2D signal from a 200 nm period fiducial grid rotated by an angle θ with respect to the scan direction. (b) 2D Fourier transforms of the grid signal after sampling 5000 pixels. Circle 1 and 2 represent spatial frequencies (k_{x1}, k_{y1}) and (k_{x2}, k_{y2}) respectively.

respect to x_P or y_P. However, the spatial frequencies are known in this case, so it is more efficient to multiply the grid signal by the appropriate complex references, $\exp(-jk_{x1}x_P - k_{y1}y_P)$ and $\exp(-jk_{x2}x_P - k_{y2}y_P)$, and sum over the sampled data. Therefore, the phase errors are give by

$$\Delta\phi_1 = \angle\left[\sum_{n=0}^{N-1} S(x_P + \Delta x, y_P + \Delta y)\exp(-jk_{x1}x_P - jk_{y1}y_P)\right] \tag{3.14}$$

$$\Delta\phi_2 = \angle\left[\sum_{n=0}^{N-1} S(x_P + \Delta x, y_P + \Delta y)\exp(-jk_{x2}x_P - jk_{y2}y_P)\right], \tag{3.15}$$

where N is the number of samples accumulated to estimate one phase shift.

Once the phase error is determined, the beam position error can be determined by adjusting for the grid rotation as given by:

$$\Delta x = (\Delta\phi_1 \cos\theta - \Delta\phi_2 \sin\theta)/k_0 \tag{3.16}$$

$$\Delta y = (\Delta\phi_1 \sin\theta + \Delta\phi_2 \cos\theta)/k_0. \tag{3.17}$$

The equation governing the phase of the signal are similar to those used for raster-scan spatial-phase locking [5] and also for phase-based multilevel alignment [77].

However, it is important to note that, in this case, we use a 2D signal that is not sequentially sampled; whereas, both of these previous efforts focused on using a sequentially sampled 1D signals. The phase locking algorithm described here makes no assumption about the order in which the pixels are acquired. As a result, one can, in principle, phase lock while scanning the electron beam in an arbitrary manner.

By combining equation 3.9 with equations 3.16 and 3.17, and assuming the estimation errors for the two phases are independent [77], the variance of the estimated position error Δx and Δy is determined by:

$$var\left\{\Delta x\right\} = var\left\{\Delta y\right\} = \frac{2}{\gamma N k_0^2}. \tag{3.18}$$

3.3 Simulation of Spatial-phase Estimation for Vector-scan System

A series of numerical simulations was performed to better understand the influence of various system parameters on the speed and accuracy of vector-scan SPLEBL. For these simulations, the figures of merit considered were the number of data points required for the position estimate to converge to a given precision and the mean residual error for a fixed number of data points. Thus, one can ensure that the position error estimate converged rapidly enough for feedback control and a small enough residual error to allow sub-nanometer patterning accuracy.

To study the effect of the scanning manner on the algorithm's performance, simulations were conducted in both flyback and spiral filling strategies. The simulation result obtained at a specific condition, i.e. specific system parameter, only demonstrates the estimated position errors in real-time as electron beam scanning over a single 100 × 50 pixels feature with no feedback loop control, so an initial beam position shift can be passed through and the large residual mean error at the beginning of the simulation may be observed. Figure 3.2 shows a simulation conducted with rotation angle $\theta = 20°$, grid period $\lambda = 200$ nm, sample spacing $ss = 40$ nm and $SNR = 10^9$, and each data point represents the estimated position error with the

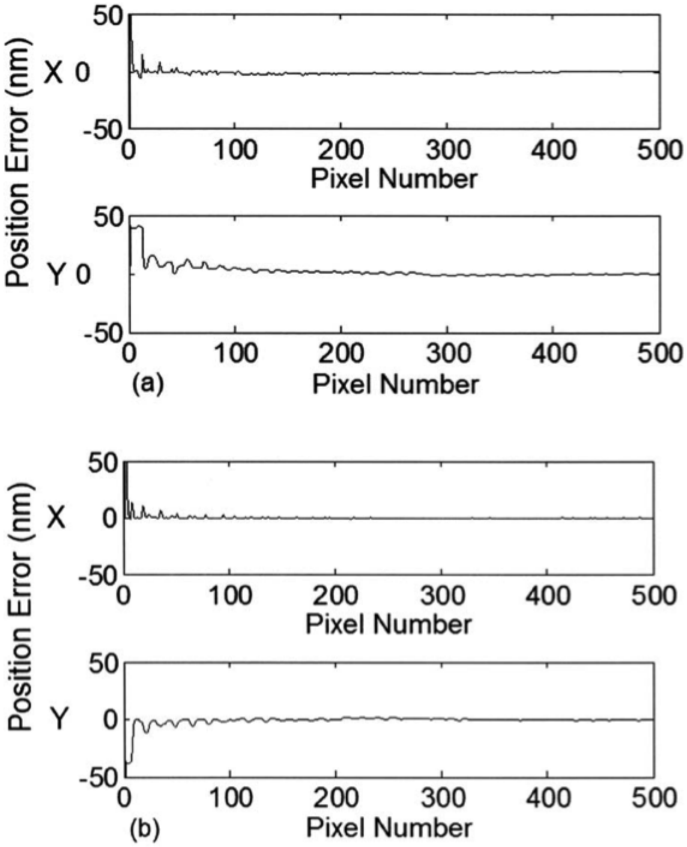


Figure 3.2: Estimated x- and y-beam position errors vs. pixel number from simulation (rotation angle = 20°). (a) Spiral fill. (b) Flyback fill.

information extracted from all the previous and the current pixels.

To investigate the effect of rotation angle with respect to the scanning direction on the algorithm, 46 angles spanning 0°-45° were simulated for both flyback and spiral filling strategies on a 200 nm fiducial grid sampled by 40 nm spacing. The SNR was set to 10^9 to eliminate the influence of noise. To determine the numer of pixels required to achieve convergence of 0.5 nm (one standard deviation), standard deviation of every 50 error estimates is calculated, and then the reciprocals of each variance are plotted versus the number of sampled data for a linear fit. The number of pixels required for spatial-phase locking at different rotation angles is shown in Figure 3.3. The filling strategy does not strongly affect the spatial-phase locking speed and less than 1000 samples are required for spatial-phase locking for all rotation angles between 0°-45°. However, the best performance is obtained between 15° and 35° rotation angles, while

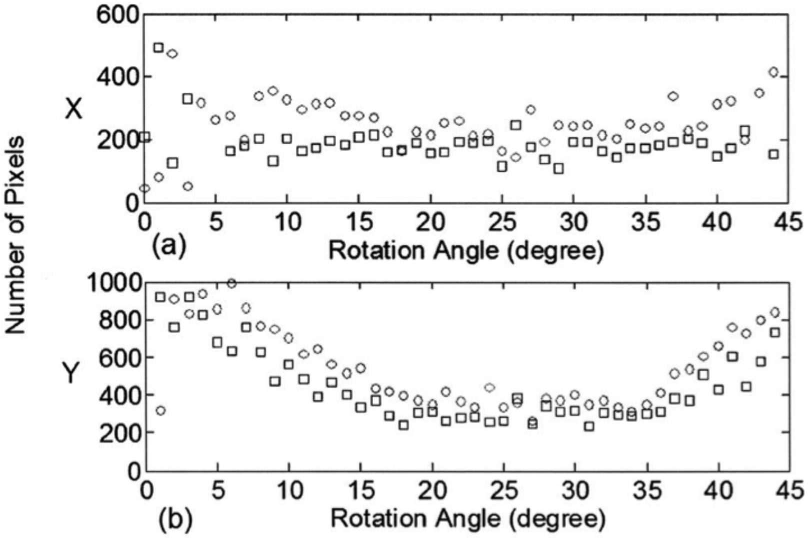


Figure 3.3: Calculated number of pixels required for 0.5 nm convergence as a function of rotation angle. (◯: spiral fill; □: flyback fill). A large range of angles provide good performance for spatial-phase locking).

angles near 0° and 45° should be avoided. It is interesting to note that the selection of an appropriate angle is less critical than when one uses a 1D signal [77] because of reduced interference from harmonics.

The effect of noise was studied by changing the SNR value in the numerical simulation and the results are plotted in Figure 3.4(a). As the noise increases, or SNR decreases, more samples are required to lock-in the beam position. In the region of typical SNRs (< 1), the performance of the algorithm is limited by the noise. Although not typical in electron microscopy or normal lithographic alignment algorithms, SNRs less than 1 are suitable for spatial-phase locking because accurate phase estimation can be performed after acquiring a suitable number of samples. At higher SNRs, which are not usually attainable at normal exposure doses, the algorithm is limited by interference from harmonic frequency components.

By changing sample spacing while keeping the grid period fixed, simulations with various ratios of grid period to sample spacing were performed and the result is plotted in Figure 3.5(a). The sampling frequency must be at least twice the grid spatial frequency according to Nyquist-Shannon sampling theorem, and the best performance of the algorithm is achieved when the ratio is just slightly larger than 2. Further

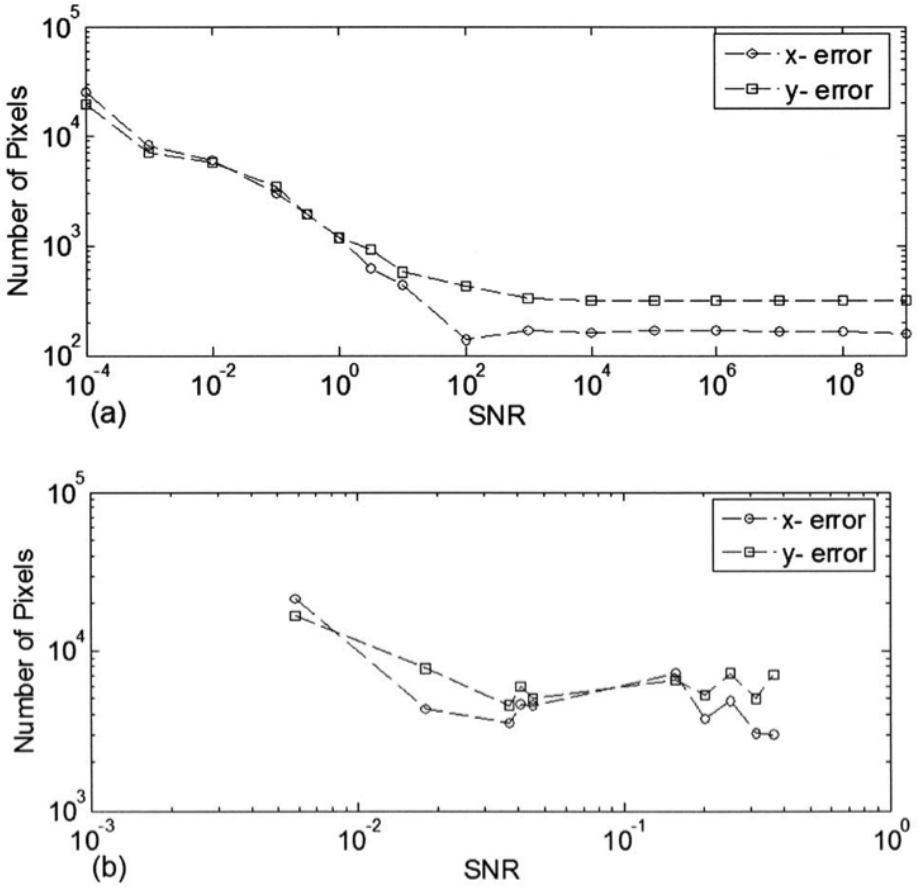
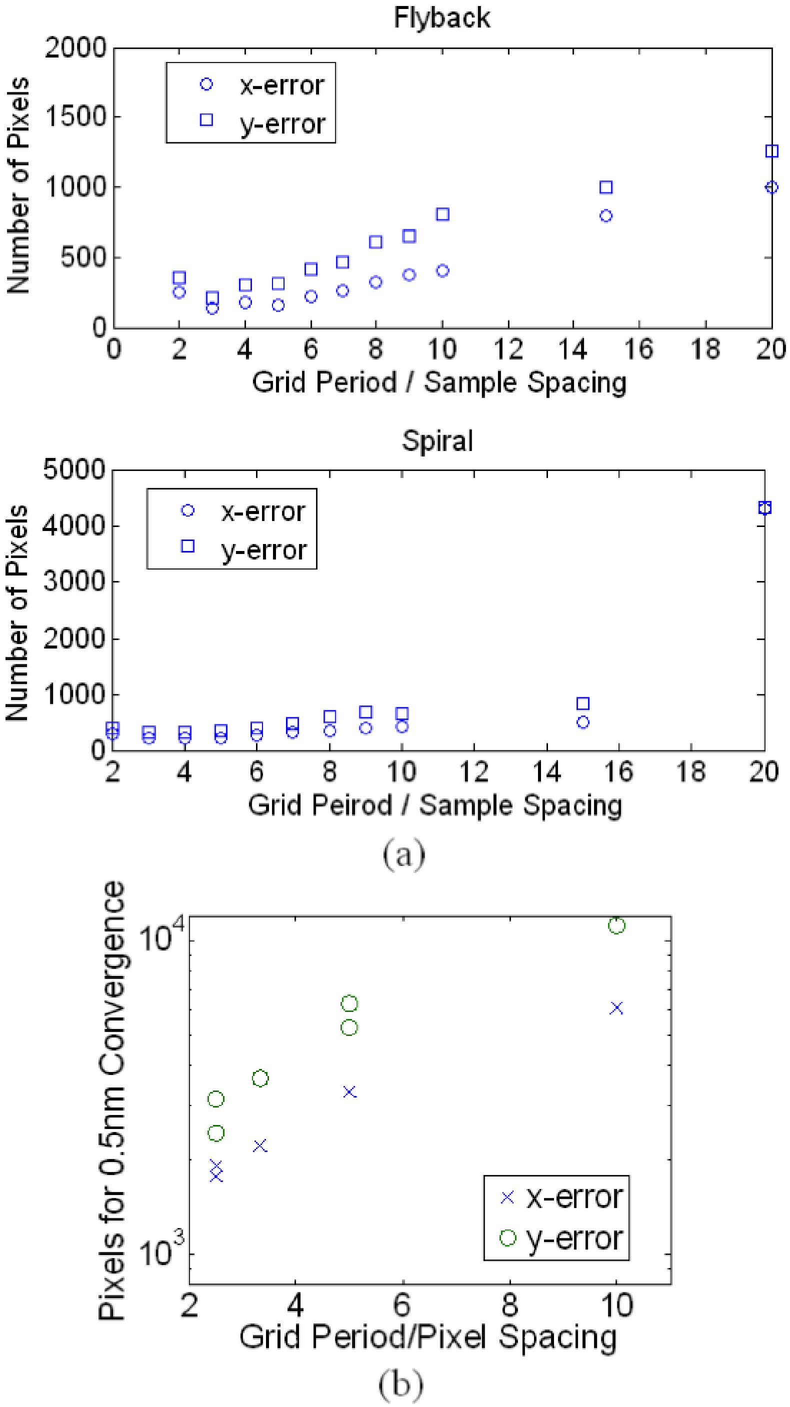


Figure 3.4: Number of pixels required for 0.5 nm convergence vs. SNR for flyback fill strategy. (a) Simulation. (b) Experiment.

increasing the number of samples per grid period deteriorates the performance due to the enhanced interference from harmonic frequency components. In the situation of exposure, the sample spacing, also referred to as step size, cannot be chosen arbitrarily due to the requirements of reasonable dose and beam speed. Therefore, for the best available fiducial grid, the bandwidth of the estimator may have to be sacrificed to ensure proper exposure under optimal conditions.

According to the simulations, the scanning strategy has little effect on the speed and accuracy of the spatial-phase locking algorithm. However, in the actual experiment of exposure, spiral filling strategy is preferred as the phase locking can be achieved in the noncritical portion of the feature and the edges will be patterned as accurately as possible.



(a)

(b)

Figure 3.5: Number of pixels required for 0.5 nm convergence vs. ratio of grid period to pixel sample spacing for flyback fill strategy. (a) Simulation with 200 nm period grid, SNR = 10^9. (b) Experimental result. Measurements were taken on a 200 nm period grid with SNR ≈ 0.4. Increasing pixel spacing beyond the minimum (1/2 grid period) tends to degrade performance slightly.

3.4 Experimental Verification of Vector-scan Spatial Phase Estimation

Two different fiducial grids, fabricated using Lloyds mirror interferometric lithography by MIT's NanoStructures Laboratory, were used for algorithm verification. A 400 nm period SiO_2 on Si grid provides higher SNR, while a 200 nm period photoresist grid on a SiO_2-antireflection coating stack provides lower SNR testing. Since the fiducial grids were used to generate secondary electron signal for algorithm evaluation in this case, they were patterned directly on the substrate rather than on e-beam resist. The procedures for patterning fiducial grids on e-beam resist can be found in the references [5, 56].

The University of Kentucky's Raith 50 SEBL system was used to image the fiducial grid. The primary beam energy was set to 30 KeV and an Everhart-Thornley detector was used to acquire the secondary electron signal while e-beam is scanning over various patterns. The grid was manually rotated at an angle of 20° with respect to the stage axes (X, Y) and the system coordinates (U, V) were mapped on the grid axes, which follows the procedure of assigning coordinates, corresponding to the exact rotation angle, to three well separated points on the grid. In this way, the grid's rotation angle with respect to the deflection axes is ensured to be 20°. Figure 3.6 illustrates the relationship between the fiducial grid and the EBL coordinates. Then the deflection field was calibrated using the modified coordinate system to compensate for the beam deflection error. The flyback exposure mode was used for all experiments because spiral scan is not yet available on this SEBL system.

To evaluate the algorithm performance, features larger than the grid period were scanned with varying SNRs. SNR was adjusted by changing the bias on the detector collection grid between -100 and 300 V. For each feature, SNR, given by a^2/σ^2, was calculated by taking discrete Fourier transform of the grid signal, determining the amplitude of the fundamental Fourier components a, and finally summing the power of all the noise components $(\sigma_1^2 + \sigma_2^2 + \cdots)$. The x- and y-position errors were estimated using the algorithm described in the previous section. Figure 3.7 shows the

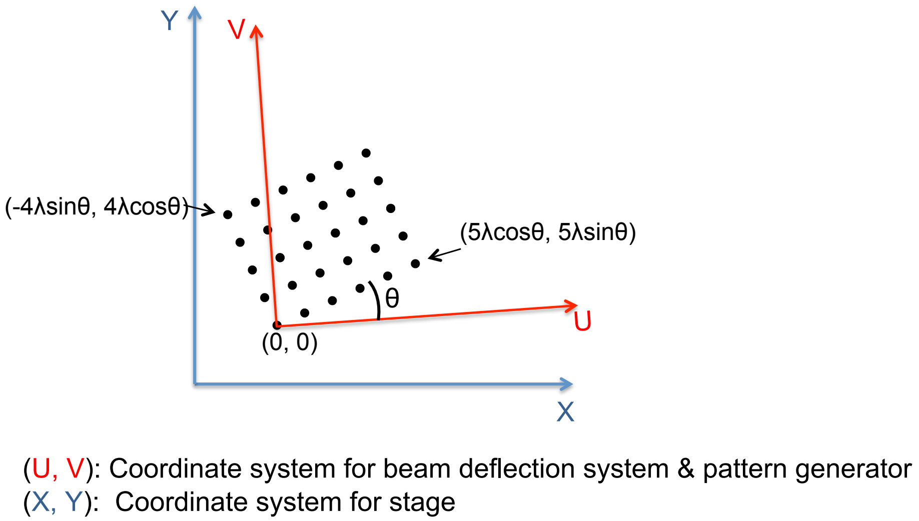

Y
V
(-4λsinθ, 4λcosθ)
(5λcosθ, 5λsinθ)
θ
U
(0, 0)
X
(U, V): Coordinate system for beam deflection system & pattern generator
(X, Y): Coordinate system for stage

Figure 3.6: Illustration of rotated fiducial grid with respect to the EBL's stage and deflection coordinate systems after mapping the deflection axes on the grid. The angular misalignment between the two coordinate system is due to the error introduced by manually rotating the grid. The three sets of coordinates assigned to the grid only shows an example of the mapping procedure. Further apart points are usually used to minimize the error introduced by localized grid defects.

secondary electron signal from 200 and 400 nm period fiducial grids along with their 2D Fourier transforms. Even though the grids are somewhat difficult to see from the secondary electron signal, the spatial frequency components are distinctly visible in the Fourier transform.

To determine how quickly the algorithm converges, we mapped the position error versus the number of pixels sampled. As observed in the simulations, the error decreased as reciprocal of the number of pixels acquired, and the number of pixels required was determined by the point at which the best fit curve fell below 0.25% of the grid period. Figure 3.4(b) shows the experimental data for the number of pixels required for convergence versus SNR. In all cases, the pixel spacing was set to 1/5 the grid period, and an average dose of 20 μC/cm^2 was used.

In agreement with the simulations, the experimental data show noise limited behavior at low SNR, but stabilizes at a somewhat larger number of pixels than predicted. This is most likely the result of increased interference from harmonic frequencies. Comparing the 2D Fourier transforms, the harmonic frequency components are

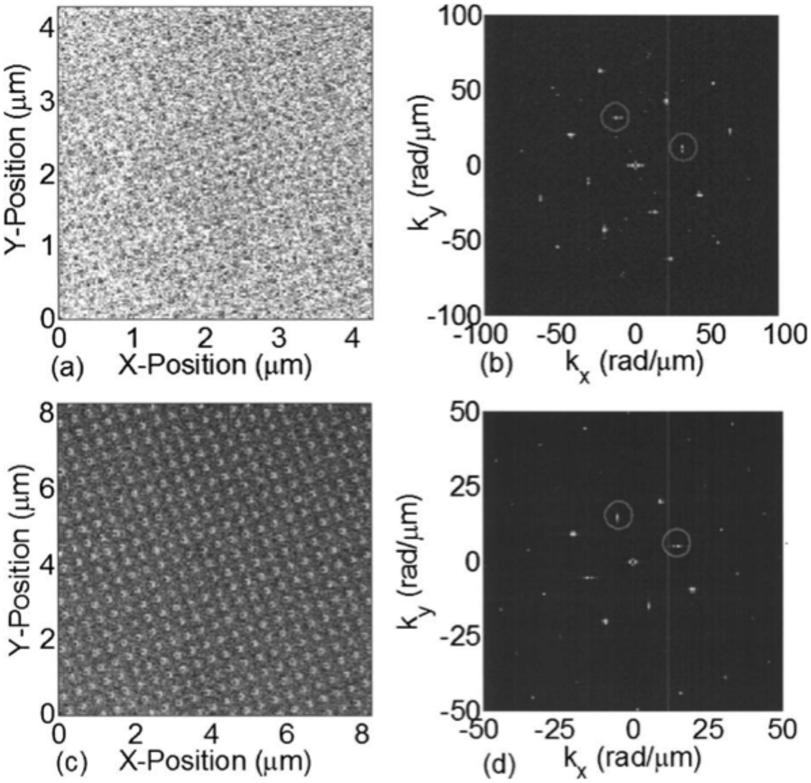


Figure 3.7: (a) Experimental secondary electron signal from the 200 nm period photoresist on SiO_2 grid. SNR = 0.04. (b) 2D Fourier transform of the grid signal after sampling 12000 pixels. The circled are the fundamental frequency components. [(c) and (d)] Experimental secondary electron signal and Fourier transform from the 400 nm period SiO_2 on Si grid. SNR = 0.39.

stronger in the real data than the simulated data. This is a result of a smaller beam diameter and enhanced secondary electron emission from the edges of the experimental grids. Both effects tend to enhance high frequency components, and, as a result, increase interference. This is especially problematic when only one or two y-axis positions have been sampled, and the overlap between fundamental and harmonic components is largest. This effect could potentially be reduced using an appropriate window function with the data, or by choosing a slightly different rotation angle. Regardless, locking still occurs within a practical data length for real grid signals.

The precision with which one can maintained spatial-phase locking after convergence to an initial error estimate was also measured. In this case, the position error was calculated 200 times during the course of scanning a feature with 600 samples per

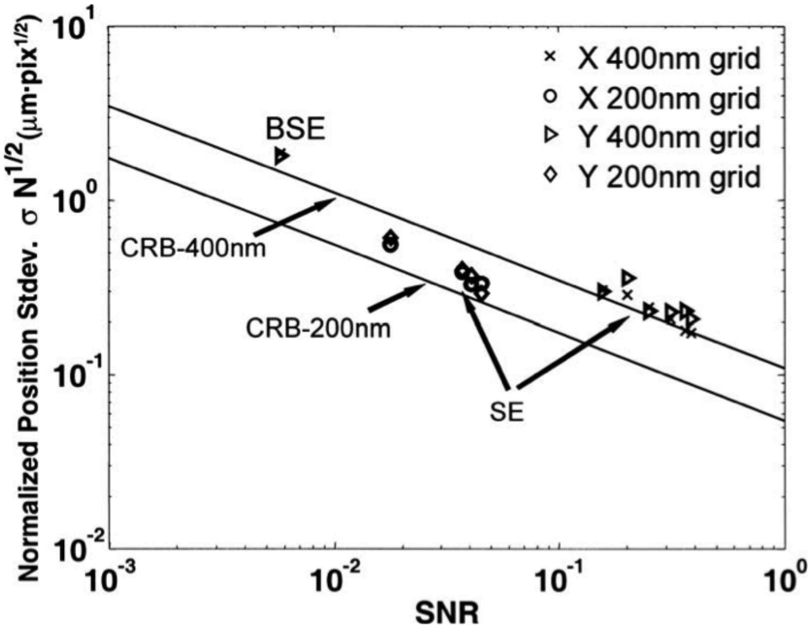


Figure 3.8: Normalized standard deviation (200 trials per data point) of experimental x- and y-position errors vs. SNR for both 200 nm and 400 nm fiducial grids rotated by 20°. Regions where the signal was primarily composed of secondary electrons (SE) or backscattered electrons (BSE) are indicated. Also shown are the Cramer-Rao bounds based on phase estimation in Gaussian white noise.

measurement. In order to reduce interference, Hann window function was applied to the time sequenced data. In Figure 3.8, each experimental data point represents the normalized standard deviation of the 200 error estimates. The standard deviation is normalized to the number of pixels used for estimating one position error, which is 600 in this case, because it is necessary to change this number to adjust the feedback bandwidth of the phase locking system. Here, the estimation variance is divided by the square root of the number of acquired pixels to find the normalized value. The straight line in Figure 3.8 corresponds to the theoretical normalized minimized variance determined by the Cramer-Rao bound for phase estimation of a single sinusoid in Gaussian white noise [75, 77]. This lower bound has been increased by a factor of 1.5 to account for the use of Hann window [81]. The experimentally measured variance is only slightly larger than the fundamental limit and clearly shows the same improvement with increasing SNR. Thus, the phase locking algorithm is nearly a maximum likelihood estimator, and the CRB can be used as a guide to the performance.

Given this, one estimates that the number of pixels required for 1 nm precision with a 200 nm period grid and the SNR of 0.1 is approximately 40,000. However, if a slower resist such as PMMA is considered, then the dose would increase by a factor of about 7 and the SNR would improve correspondingly. As a result, the number of pixels required would decrease to about 5000. This indicates that feedback bandwidth in the range from hundreds of hertz to a few kilohertz should be achievable with typical vector-scan pattern generation rates (10-50 Mpixels/s).

Fianlly, the effect of sample spacing on algorithm performance was evaluated using sample spacing of 20, 40, and 80 nm. Figure 3.5(b) plots the number of pixels required for 0.5 nm convergence versus the ratio of grid period to pixel size. In agreement to the numerical simulation, the best performance is achieved at about twice the spatial frequency. Oversampling results in slower spatial-phase locking because fewer periods are sampled and interference with harmonic components increases. This suggests that, all other things being equal, one should use the largest pixel spacing possible; however, this is often limited by other exposure parameters.

The simulations and experiments indicate that vector-scan spatial-phase locking has the potential to provide sub-nanometer pattern placement accuracy for electron beam lithography. Acceptable performance was obtained with different fill strategies (flyback and spiral), and the vector-scan SPLEBL algorithm allows flexibility in choice of exposure parameters. Sub-1-nm beam position estimation can be achieved with practical SNRs, reasonable data lengths, and commonly used exposure doses.

For features with at least one dimension large with respect to the grid period, phase locking can be achieve within a single feature. This suggests that a spiral fill strategy allowing locking in the center of the feature while precisely writing the edges will be most effective. Alternatively, for features with both dimensions smaller compared to the grid period, it will be necessary to accumulate data over several features before accurate phase locking can be achieved. This requires more complex, but still feasible exposure strategies where the less critical or larger features could be patterned first, followed by more critical and smaller features. Similarly, the interior of several features could be exposed first followed by the edges of the features. Finally, one could expose features in unused areas of the field at a sub-exposure dose to initially achieve phase locking.

Chapter 4

Implementation of Vector-scan Spatial-phase Locking on Field-Programmable Gate Array

The previous implementation of real-time spatial-phase locking used a general purpose microprocessor, a real-time operating system, and PCI bus connected analog-to-digital converter (ADC) and digital-to-analog converter (DAC) boards. This work demonstrated sub-1-nm (one standard deviation) pattern placement precision for raster-scan electron-beam lithography [5]. However, it only supports e-beam lithography systems with writing speed of less than 10 Mpixels/s, and the computational rate for estimating beam position error on microprocessor is limited to several kilohertz with millisecond delay. In addition, the complexity and high cost of this configuration make it difficult to be parallelized on multi-beam or multi-column system. In order to meet the requirements of higher pixel rates used by many current and emerging e-beam lithography tools as well as straightforward parallelization for multicolumn system, Field-Programmable Gate Array (FPGA) is chosen to implement the spatial-phase locking algorithm.

4.1 Hardware for Implementation

Implementing spatial phase locking algorithm for vector-scan EBL requires not only a FPGA board that provides sufficient hardware resources and computational rate, but also peripherals, such as ADCs/DACs for data acquisition and feeding the correction signal back to EBL, as well as memories for more complex beam deflection calibration.

With all the requirements in mind, we chose the XtremeDSP Development Kit-IV as our development platform as shown in Figure 4.1 [82]. The development board consists of a Virtex-4 User FPGA, 2 independent 14-bits ADC channels, 2 independent 14-bits DAC channels, two banks of 133 MHz ZBT-SRAM with 512K×32-bits per bank. One of the ADCs will be used to route the grid signal from the SE detector to the Virtex-4 FPGA, at where the position errors are calculated according to the spatial phase estimation algorithm. The error estimates in x- and y-direction will be directed separately to the DACs, and the output of the converters will be connected to the beam deflection system of EBL for correction. The Virtex-4 FPGA (XC4VSX35-10FF668) supports clock frequency up to 350MHz, and both ADCs and DACs can convert data at the maximum rate of 105 mega samples per second (MSPS).

The memory will be used to store the estimated errors during scanning e-beam over the fiducial grid within the write field. The saved errors can be used to characterize the higher order deflection distortion, which will be corrected before conducting the real-time feedback control. Considering the often used write field size of 100 μm × 100 μm, if a 200 nm period grid is sampled every 40 nm, one will collect 6,250,000 samples of the grid signal within the write field. The simulation suggests the position control with 1 nm precision requires the number of samples for one beam position estimate to be in the range of several thousands. Therefore, at most 6,250 error estimates in each of x- and y-direction will need to be recorded, which is well within the memory's capacity.

In addition, the development board provides total of 42 user accessible digital I/O pins. One of the user I/O pins will be connected to the beam blanking signal from EBL, which will serve as the control signal for the estimation.

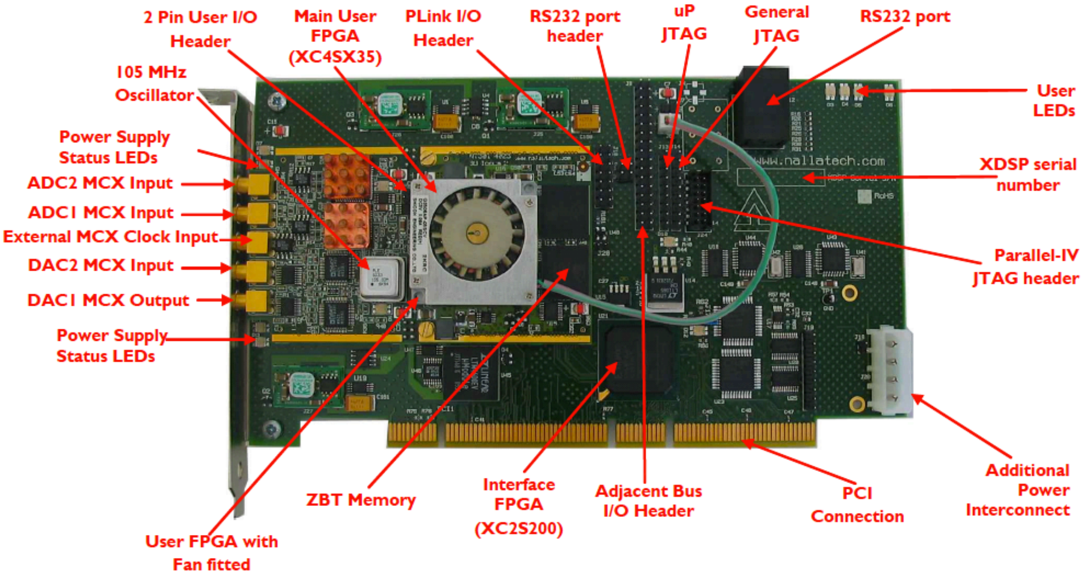


Figure 4.1: Front view of the development board physical layout.

Although the sampling and computational rate supported by the development board exceeds the patterning speed of the Raith e_LiNE system which the hardware design will be tested on, it offers the enhanced flexibility that allows for the implementation to be incorporated with the higher-speed commercially available lithography systems. The Leica's Ebpg 5000 and JEOL's JBX-9300FS system support high energy exposure up to 100 KeV and the installed pattern generation system runs up to 25 MHz and 50 MHz respectively. If the estimation is performed at 50 MHz over 50,000 samples on a 200nm period fiducial grid, one can provide feedback position control at bandwidth of almost up to 5 KHz, with the consideration of using a loop filter to attenuate the errors introduced by the estimator itself. This will lead to 50 times improvement on the bandwidth compared to the implementation on microprocessor, thus more rapidly varied position errors can be compensated during exposure.

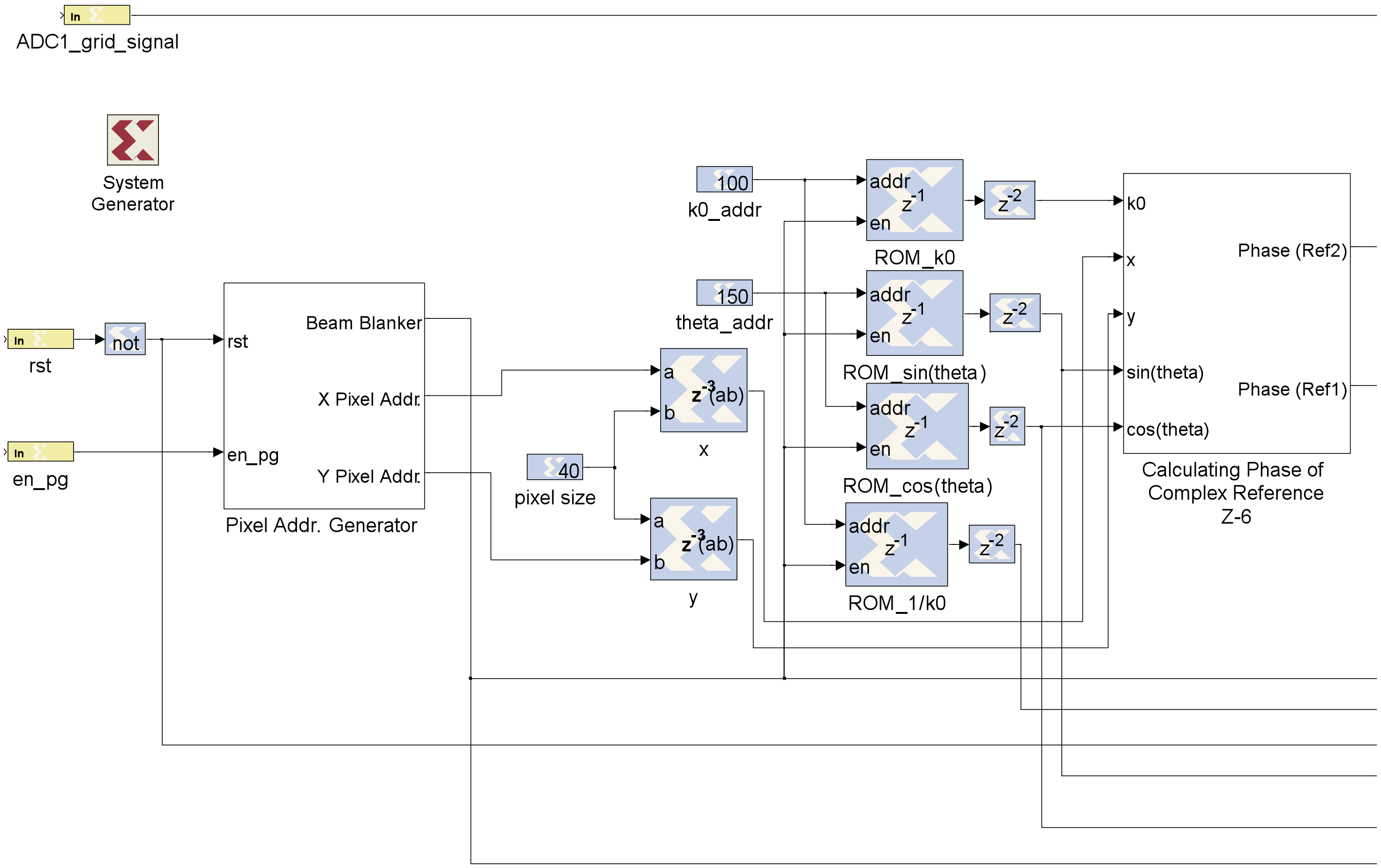
For the low-voltage electron beam microcolumn system that supports pixel rate of up to 100 MHz [20], a sophisticated hardware design on this development board holds the promise to meet the need of high speed performance. However, the feedback bandwidth may not scale up linearly, if not narrower, because much worse SNR is expected at low primary electron energy, thus more samples have to be acquired to ascertain desired estimation precision.

4.2 Digital Signal Processing (DSP) Design for Spatial-Phase Estimation Algorithm

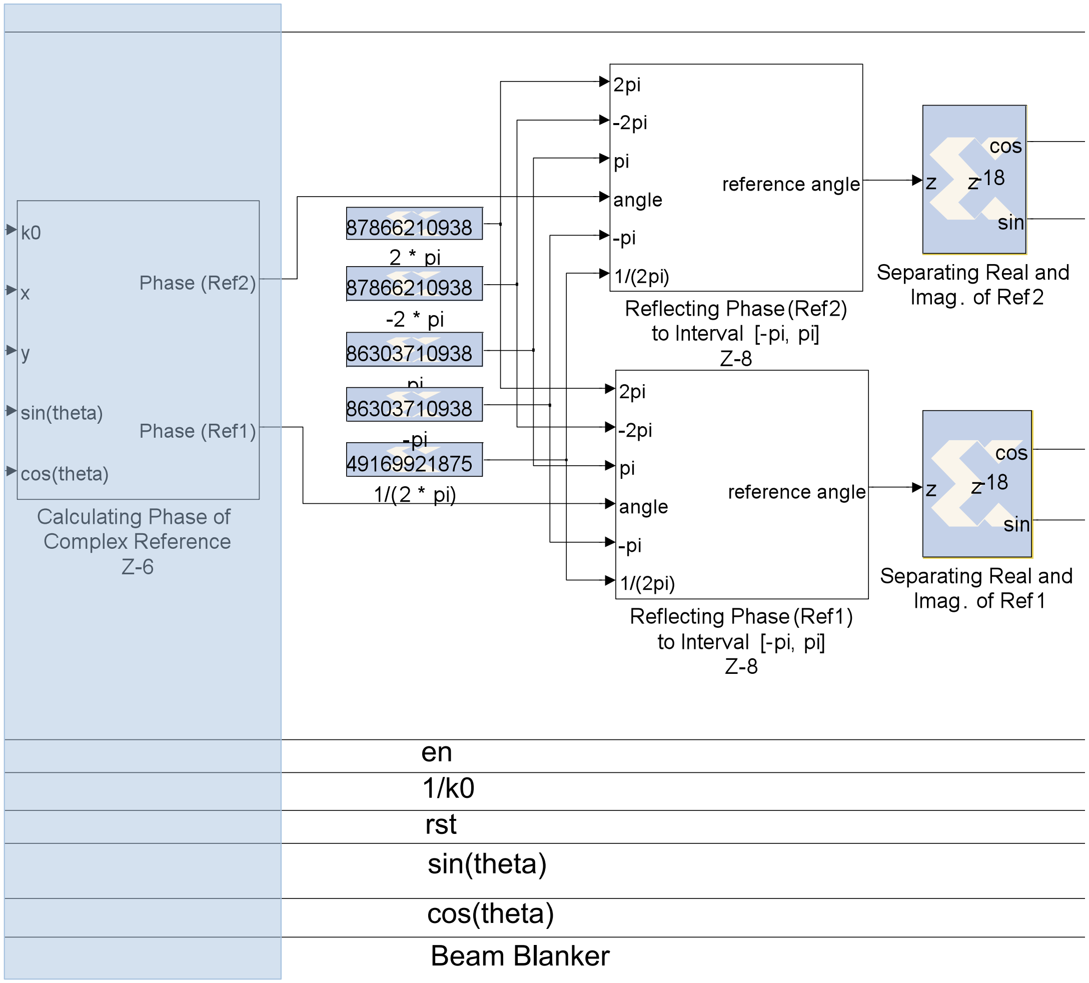
4.2.1 Algorithm Translation into DSP System Using System Generator

With the MATLAB model of the spatial phase estimation algorithm been verified through simulations, a digital signal processing (DSP) counterpart with fixed-point numerical precision and hardware details needs to be created so that the algorithm can to be executed by FPGA. Xilinx System Generator (SG) was used to build the DSP design with Xilinx DSP blockset in the Mathworks design environment Simulink. In the SG design, the hardware implementation details are specified for the Xilinx devices. Also, the DSP blockset, Simulink, MATLAB and hardware description language (HDL) can be brought together for the simulation and implementation of the design.

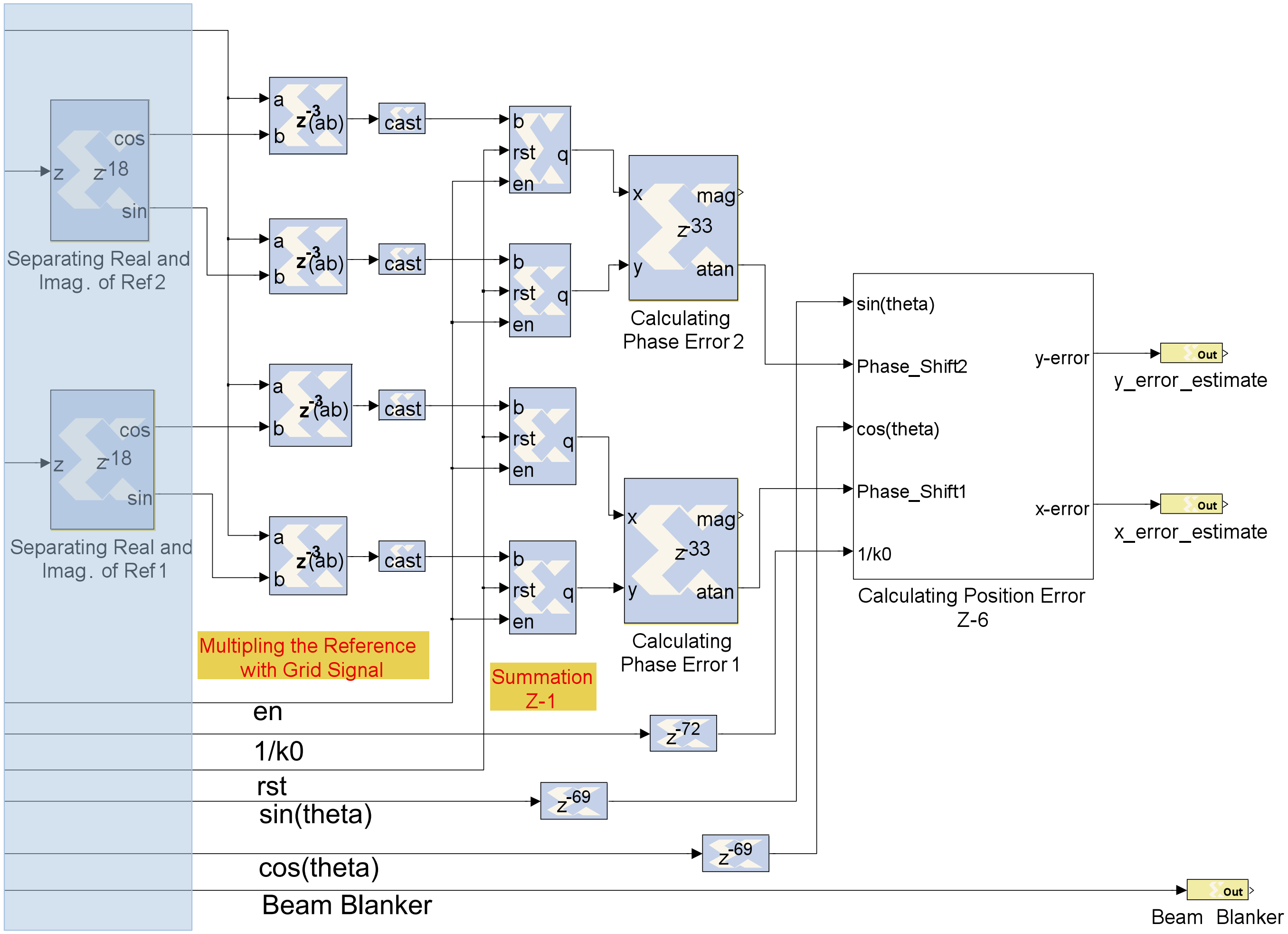
The DSP blockset can be accessed via Simulink Library Browser once System Generator is installed with the compatible MATLAB. In addition to the common building blocks, such as adder, multiplier and register, more complex blocks, such as filter, memory and FFTs, are also included. The very essential blocks for any DSP design in SG are "Gateway In/Out" and "System Generator" token. The "Gateway In/Out" defines the boundary of the SG design for FPGA. The "Gateway In" converts the floating point input to a fixed-point number, i.e., Boolean, signed and unsigned. Since the DSP blocks are bit-accurate, meaning the value produced by them matches that produced in hardware, it is important to set appropriate bit width and the location of the binary point for a numerical value, so that the binary representation corresponds to the desired range and resolution. Therefore, any DSP block that involves arithmetic operation needs to be specified in the same manner. “System Generator” token is not connected to anything in the DSP design, but serves to drive FPGA implementation process as well as to identify target netlist, device, performance target and system period. The standard Simulink blocks are not supported for use within the boundary but can be useful for simulation to verify the functionality of the DSP design.


In
ADC1_grid_signal
System Generator
In
rst
not
In
en_pg
rst
en_pg
Beam Blanker
X Pixel Addr.
Y Pixel Addr.
Pixel Addr. Generator
40
pixel size
a
b
z^{-3}(ab)
x
a
b
z^{-3}(ab)
y
100
k0_addr
150
theta_addr
addr
z^{-1}
en
ROM_k0
z^{-2}
addr
z^{-1}
en
ROM_sin(theta)
z^{-2}
addr
z^{-1}
en
ROM_cos(theta)
z^{-2}
addr
z^{-1}
en
ROM_1/k0
z^{-2}
k0
x
y
sin(theta)
cos(theta)
Phase (Ref2)
Phase (Ref1)
Calculating Phase of Complex Reference Z-6

(a)


k0
x
y
sin(theta)
cos(theta)
Phase (Ref2)
Phase (Ref1)
Calculating Phase of Complex Reference
Z-6
87866210938
2 * pi
87866210938
-2 * pi
86303710938
pi
86303710938
-pi
49169921875
1/(2 * pi)
2pi
-2pi
pi
angle
-pi
1/(2pi)
reference angle
Reflecting Phase (Ref2) to Interval [-pi, pi]
Z-8
Reflecting Phase (Ref1) to Interval [-pi, pi]
Z-8
z
z-18
cos
sin
Separating Real and Imag. of Ref2
Separating Real and Imag. of Ref1
en
1/k0
rst
sin(theta)
cos(theta)
Beam Blanker

(b)



(c)

Figure 4.2: Schematics of the DSP design with self-generated digital beam position in System Generator using Xilinx blocks. The lower level schematics of the subsystems can be found in Appendix A.1.

To achieve the synchronization between the FPGA and the lithography system, the DSP system is designed to self-generate the intended beam position once the write field scan starts as demonstrated in Figure 4.2. If the intended digital beam positions are accessed directly from EBL, the pixel address generator can simply be replaced with two "Gateway In" blocks to import positions in x- and y-direction respectively as depicted in Figure 4.3. The input port for grid signal "ADC1_grid_signal" is defined as signed 14-bit fixed-point arithmetic with 13 fractional bits to accommodate the -1 V to $+1$ V input range of the ADC. The "reset" signal is included to clear and reset the counting and accumulating blocks after scanning each write field, and is asserted through user interface software. The "en_pg" is connected to the beam blanking signal from EBL's Pattern Generator to activate pixel address generation once the scanning starts. A new beam blanking signal "Beam Blanker" will be routed back to the Beam Blank Controller of EBL to physically switch e-beam on and off.

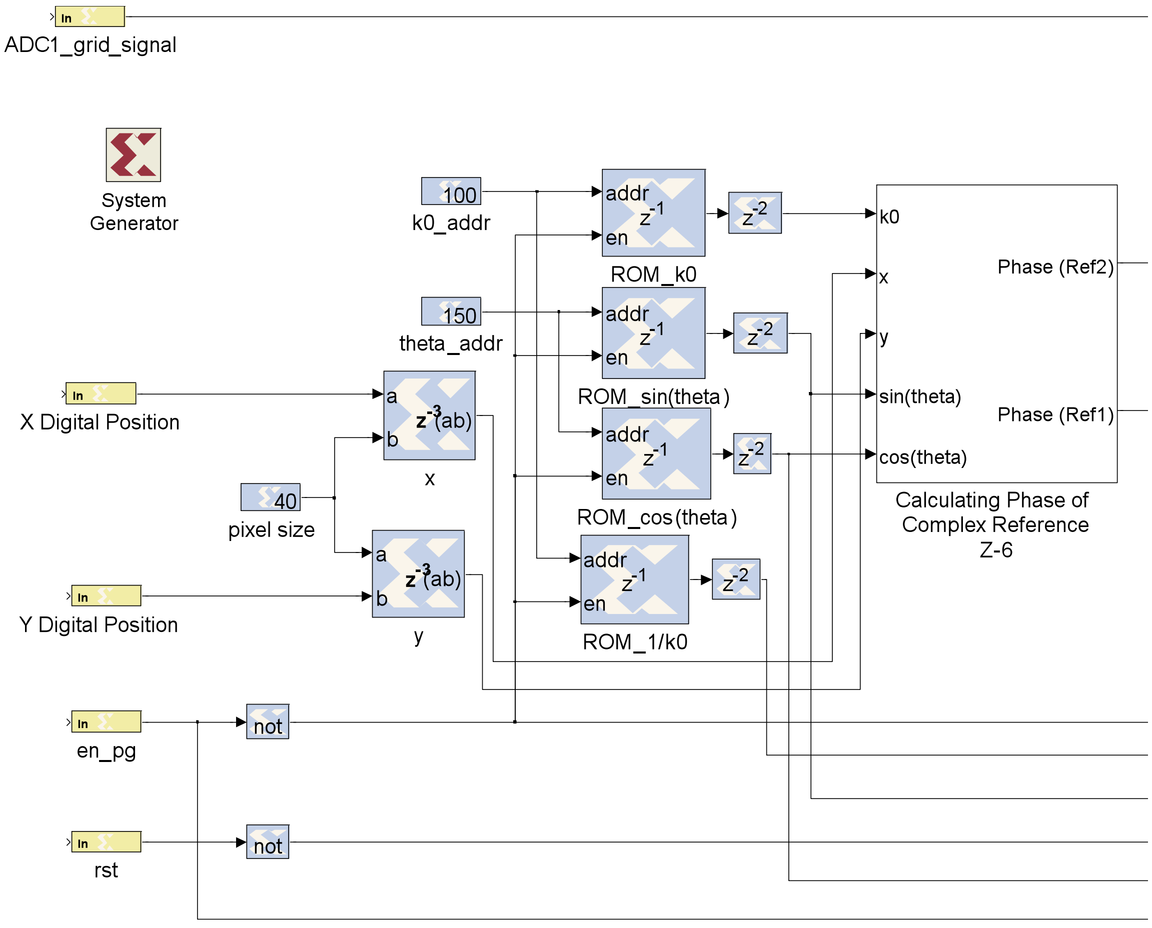


Figure 4.3: Schematics of DSP design in System Generator for accessing digital beam position directly from EBL. Only the input part of the design is shown, and the rest is identical to that shown in Figure 4.2.

To synchronize the pixel address generation with the beam deflection when the "flyback" filling strategy is used, the "Pixel Addr. Generator" subsystem is designed to execute the following actions: (1) The counter corresponding to digital addresses in x-direction updates the count when the beam steps over to the next pixel. (2) The counter for x-direction counts to a limit while the beam is scanning along the line, then stops and resets when the beam is deflected back for scanning the next one. (3) The counter for y-direction updates the count when x-direction counter resets, but holds the value duing line scan. The count limit is determined by the write field dimension and pixel size, and the physical beam position is obtained by multiplying the signal "pixel_size" with the outputs of the "Pixel Addr. Generator" subsystem.

To calculate the complex references, $\exp(-jk_{x1}x_P - k_{y1}y_P)$ and $\exp(-jk_{x2}x_P - k_{y2}y_P)$, one needs the information of grid period and rotation angle in addition to the position of the beam. A range of values for grid spatial frequency and rotation angle are given to the initial value vector of the Xilinx real-only memory (ROM) block, and the desired value is accessed at the output by specifying its location at the input port.

The phases of the two complex references for each beam position are calculated by "Calculating Phase of Complex Reference" subsystem, and then the real and imaginary part of the complex number are calculated by "CORDIC SINCOS" block from Xilinx reference blockset. The equivalent phase angle of the complex reference in the interval $[-\pi, \pi]$ is determined by "Reflecting phase to Interval $[-\pi, \pi]$" subsystem because the "CORDIC SINCOS" block only converges for angles in that range. The grid signal acquired at the specific location is multiplied to the real and imaginary part of the corresponding complex reference respectively, which is followed by summation performed over certain number of samples.

For the DSP system depicted in Figure 4.2, the accumulator continuously adds the input to the sum of all the previous values until the write field scan finishes. Therefore the output of this implementation demonstrates the estimated beam position as the number of samples increases. Although, a more sophisticated control scheme for the summation is required for feedback control, which resets the accumulator once the desired number of samples is reached, the design described here can help verify the functionality of the DSP design and also determine the number of samples required for acceptable estimation precision under experimental condition.

Finally, the accumulated real and imaginary part of the complex number are brought to the inputs of the "CORDIC ATAN" block to calculate the spatial phase in the interval $[-\pi, \pi]$, which is then used by the "Calculating Position Error" subsystem to estimate the beam position.

As the operation gets more complex, more pipeline stages are involved, and thereby more clock cycles are required to yield the result. For example, a multiplication takes 3 clock cycles to complete, while "CORDIC SINCOS" block needs 4 clock cycles or more according to the desired resolution. In order to synchronize the dataflow, the "delay" block, denoted as "Z-n" (n is the specified number of clock cycles), is placed in the paths with less pipeline stages to wait on more complicated calculations to be carried out in the others.

Timing Analysis tool provided by SG was used to determine the shortest clock cycle for the DSP design to be implemented by the hardware. For the current design, 25 ns FPGA clock period or 40 MHz computation rate is required to accommodate the slowest path.

Simulation of the DSP design was conducted in Simulink modeling environment. A noise-free (SNR $=10^9$) signal, sampled by a Gaussian beam of 40 nm radius from a 200 nm period grid, was generated in Matlab and imported into the DSP design sequentially in the way that mimics the beam scanning over a feature of 100×50 pixels with sample spacing of 40 nm using "flyback" filling strategy. The simulation result is plotted in Figure 4.4. The trances indicate that the beam position error converges quickly within less than 500 clock cycles, corresponding to about 400 pixels, at about -40 nm and 0 nm in x- and y-direction, respectively. The residual beam position error is due to an initial electron-beam shift with respect to defined origin on the grid. The constant output before the position error starts fluctuating is due to the pipeline latency of the processing elements in the DSP design.

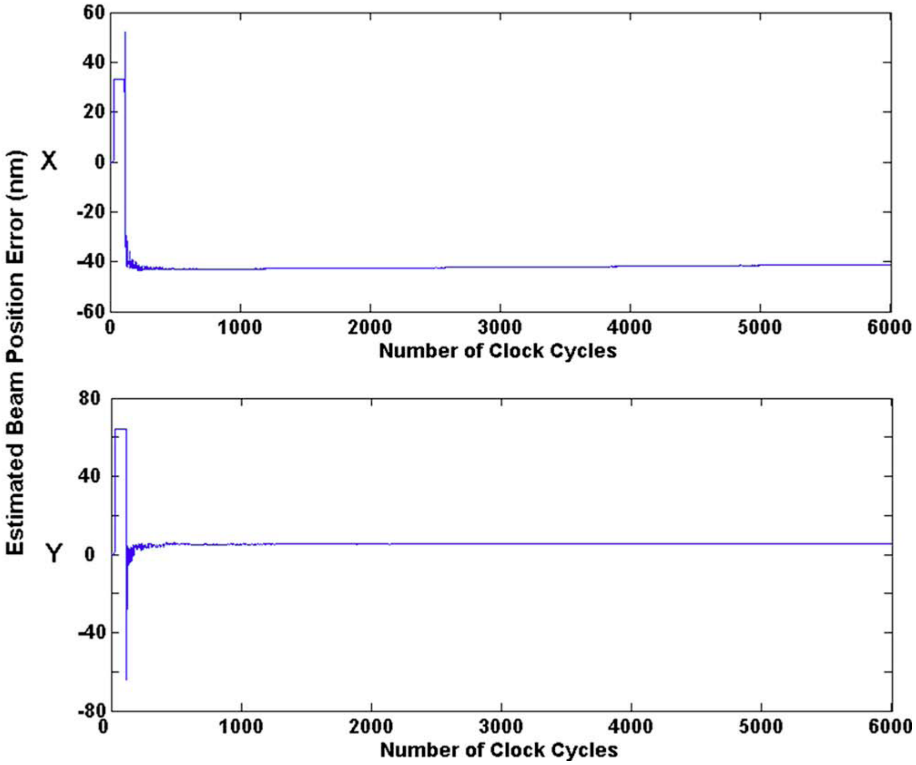


Figure 4.4: Estimated x- and y-beam position errors vs. number of FPGA clock cycles from DSP design simulation in Simulink. (rotation angle: 20°; grid period: 200 nm; pixel size: 40 nm; sample area: 100×50 pixel feature; SNR: 10^9).

4.2.2 Top-level Hardware Design Using Hardware Description Language (HDL)

Now with the SPL algorithm being successfully translated into a DSP system, a top-level HDL model is created to handle the external interface issue. To incorporate the System Generator design into the HDL design, NGC Netlist compilation was launched to encapsulate the SG design into a binary module, which contains both logical design data and constraints. Then the SG design is instantiated as the "black box" in the top level HDL entity. The HDL entity targeted on Virtex-4 user FPGA is programmed to (1) connect input/outputs of the SG module to the signals associated with the FPGA onboard pins and ADC/DACs' I/Os, (2) specify one of the programmable oscillators as the system clock signal that is controlled through user interface software, (3) pass the clock signal to Virtex-II clock FPGA, and (4) define the ADC/DACs' control signals to achieve the best performance at the desired clock rate.

The Virtex-II clock FPGA on the motherboard is designated to manage the clock signals for both the main FPGA and the ADCs/DACs as demonstrated in Figure 4.5. In the HDL entity (refer to Appendix B.1 for details) targeted on Virtex-II clock FPGA, the clock signal routed from the user FPGA is forwarded to ADC/DACs, and at the same time, a feedback clock signal is sent to main FPGA so that the synchronization between data acquisition and signal processing is assured. A user constraint file associated with each HDL entity assigns the onboard pin numbers to the signals, as well as the clock period of 25 ns, according to the timing analysis, to the clock input pin. After both HDL entities were verified through simulation, hardware compatible bitstream files were generated and downloaded on the specific FPGA.

4.2.3 Experiment Verification of DSP Design Functionality

The University of Kentucky's Raith e_LiNE SEBL system was used to test the FPGA implementation of the spatial phase estimation. The primary beam energy was 10 KeV and the aperture was 30 μm, providing a beam current of $203.7pA$. The in-lens detector was used to acquire the secondary electron signal, and the detector's brightness and contrast were adjusted so that the output signal from the detector varied between -1 V and $+1$ V. To verify the implementation, 1 μm spatial period gold grid patterned directly on Si substrate rather than on top of e-beam resist was used in the experiment. Although such grid differs from the $<$ 250 nm period, ultra-thin Al or Cu grids used in the exposures, it allows to conveniently test the implementation at high signal-to-noise ratio. The grid was manually rotated at an angle of 20° with respect to the stage axes and the deflection system coordinates were mapped to the grid axes. The deflection field was calibrated using the modified coordinate system to compensate for any mechanical alignment error. Flyback exposure mode was used for the experiments discussed in the rest of this section.

The experimental setup is demonstrated schematically in Figure 4.6. The Xilinx

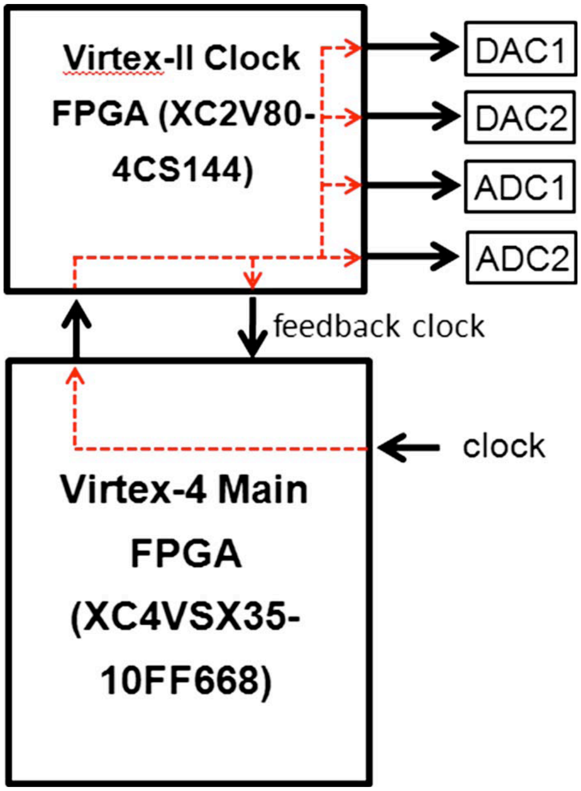

Virtex-II Clock FPGA (XC2V80-4CS144)
DAC1
DAC2
ADC1
ADC2
feedback clock
clock
Virtex-4 Main FPGA (XC4VSX35-10FF668)

Figure 4.5: Schematic of clock signal routing on the FPGA. A programmable oscillator controlled by FUSE system software was chosen as the system clock signal. It is passed to clock FPGA to drive ADC/DACs and fed back to main FPGA to achieve synchronization.

XtremeDSP development board is installed on a host computer and controlled by Nallatech FUSE system software. The secondary electron signal is sampled by the ADC driven by 40 MHz clock and then sent directly to Virtext-4 main FPGA where the spatial phase estimation algorithm is implemented. Since the maximum pixel rate supported by the SEBL system is 10 Mpixel/s, the ADC takes four samples at each pixel. The estimated beam position error is converted to analog signal by two DACs driven by the same clock signal, and captured on Tektronix TDS 380 oscilloscope. The experiments were conducted without laser-interferometer feedback control of the stage position.

The first test of the FPGA implementation was done by scanning the electron beam over a 100×100 μm^2 write field with sample spacing of 200 nm and dwell time of 100 ns. The corresponding dose was 0.051 μC/cm^2. The experimental data shown

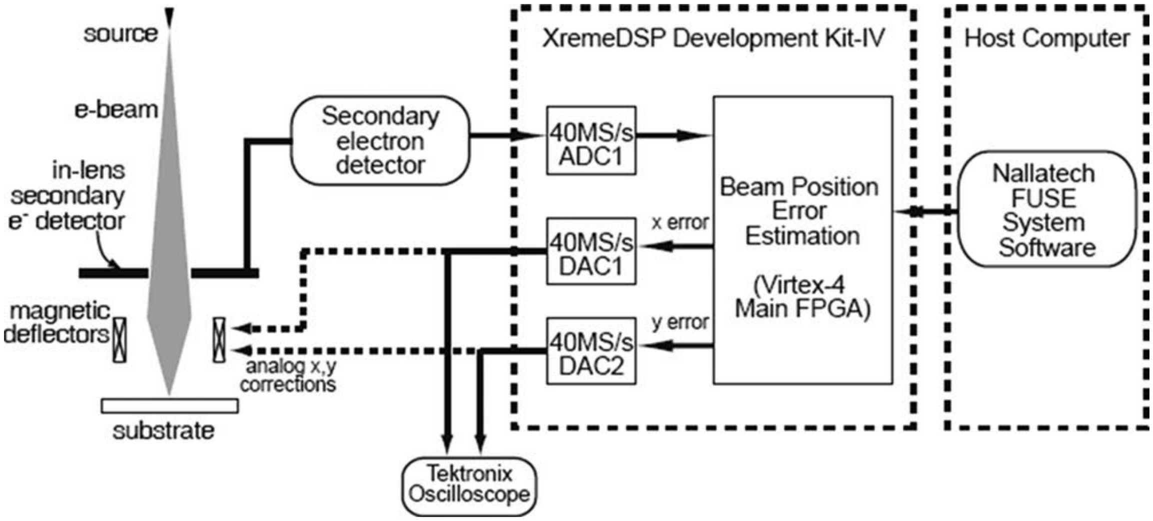


Figure 4.6: Schematics of experimental setup. The grid signal from the in-lens secondary electron detector is routed to a high-speed ADC on the FPGA board. For actual exposure, the correction signal is routed to the deflection system (dashed lines). For validation purpose, the feedback correction signal is acquired on an oscilloscope (solid lines). The FPGA board is programmed and controlled through a host computer.

in Figure 4.7 indicate that the beam position error fluctuates for 3 ms, or about 20,000 pixels, and then converges. Considering commonly used e-beam resist such as PMMA, the dose would increase by a factor of 2000 for proper exposure, resulting in a significant SNR improvement. However, the lower SNR and finer periods of real electron-transparent grids would also influence system performance. In addition, the field was not corrected for higher (second and third) order distortions in this case. These beam deflection errors occur too quickly for the phase-estimating system to track, and thus lengthen the time required for the system to converge. This is particularly pronounced in a flyback exposure of a 100 × 100 μm^2 write field using a scanning electron microscopy column.

The implementation was also tested by acquiring samples every 40 nm from a feature of 20×20 μm^2 square in the center of the write field, where the beam deflection error is expected to be the minimum. The same dwell time was used in this case, and the corresponding dose was 0.127 $\mu\text{C/cm}^2$. This is still much lower than the typical exposure dose level, and the performance of the algorithm is limited by the noise in

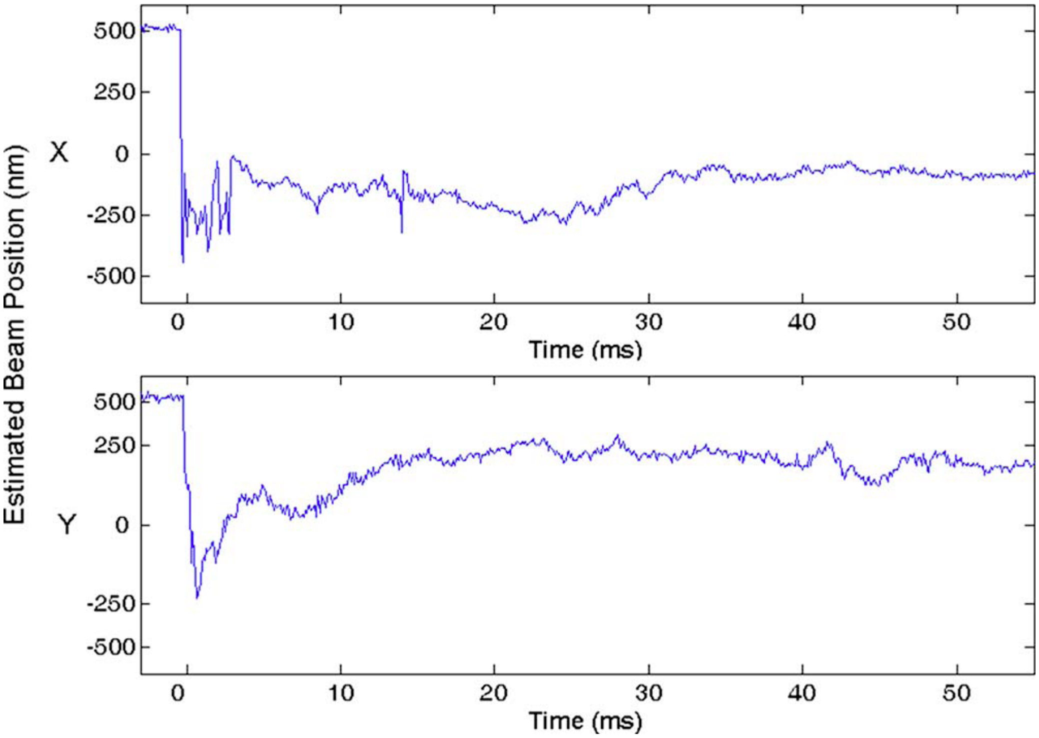


Figure 4.7: Experimental results of estimated x- and y-beam position errors vs. time. Measurements were taken on 1 μm period gold grid in a 100 × 100 μm^2 area with area dose of 0.051 $\mu\text{C/cm}^2$ at sampling speed of 40 MHz.

the secondary electron signal. The experimental data are plotted in Figure 4.8. The beam position error changes dramatically for about 17 ms, or 80,000 pixels, before it is confined to small oscillations.

Compared to traces obtained in the experiment with larger pixel size, longer times or more pixels are required to achieve spatial-phase locking even with slightly higher dose level. This is in agreement with the numerical simulations addressed in section 3.3. According to Nyquist-Shannon sampling theorem, the sampling frequency must be at least twice the grid spatial frequency; however, increasing the number of samples per spatial period further reduces the performance of the algorithm because fewer periods of the fiducial grids are sampled, leading to enhanced interference from the harmonic frequency components. This reemphasizes that, aside from using larger pixels when sampling, which is usually limited by other exposure parameters, finer fiducial grids greatly improve performance.

The experimental results demonstrate that a FPGA implementation of SPL al-

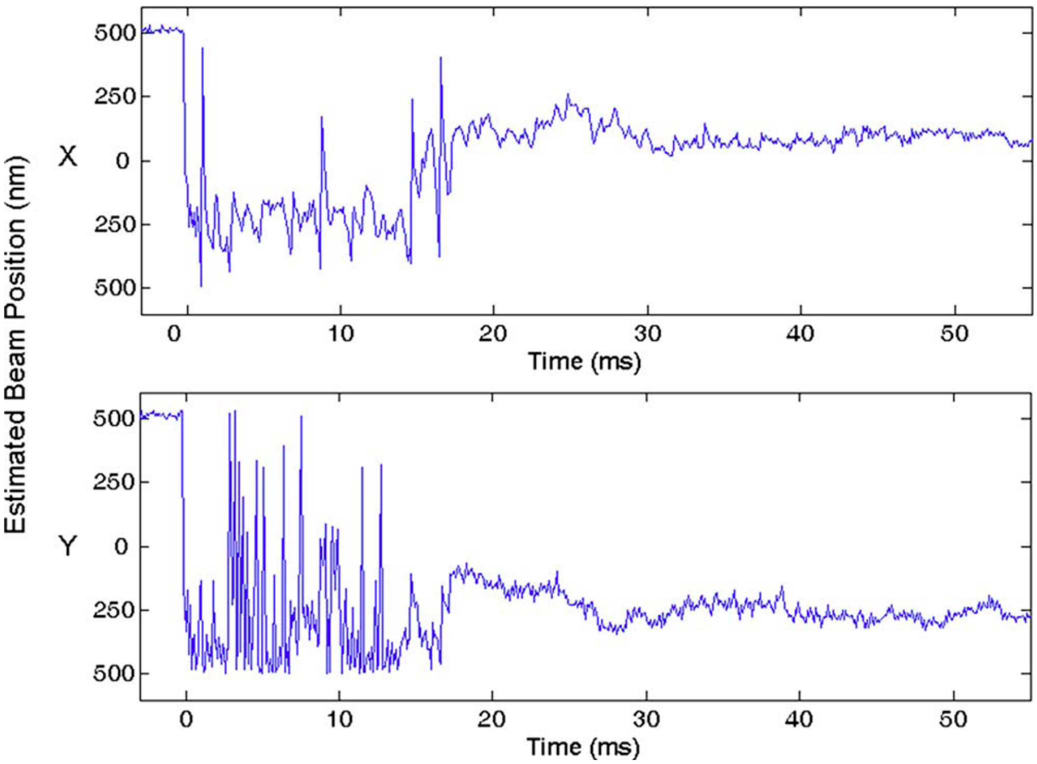


Figure 4.8: Experimental results of estimated x- and y-beam position errors vs. time. Measurements were taken on $1\mu m$ period gold grid in a 20×20 μm^2 area with area dose of $0.127\mu C/cm^2$ at sampling speed of $40MHz$.

gorithm provides real-time beam position estimation with greatly improved speed. The grid signal is sampled at 40 MHz, which is four times faster than the previous implementation on microprocessor. Electron beam position error is calculated also at 40 MHz with only a few μs initial delay, while the previous work only achieved 2.5 KHz with more than 1 ms delay. Optimization of the FPGA design promises further acceleration so that the implementation will accommodate e-beam lithography tools with even higher pixel rate.

Considering the exposure conditions under which the experimental data were obtained, the performance of the algorithm, in terms of number of pixels required to lock in, is expected to improve significantly when it is applied in patterning process. Using typical exposure doses, more than 100 times larger than the dose used here, will provide a grid signal with higher SNR and consequently faster convergence. For a particular lithography process defined by the exposure rate, resist speed, and grid

SNR, a low-pass loop filter can be included after position error estimation to balance feedback bandwidth and position precision.

4.3 FPGA Implementation of Pattern Generation

Because of the pipeline and path delay in the hardware design, synchronization between the position estimation and the exposure can not be assured if the original beam blanking signal, routed from the EBL Pattern Generator, controls the beam blanking electrode in the electron-optical column. Therefore, we designed a pattern generating component on the FPGA, which derives a new beam blanking signal based on the FPGA-generated addresses, in order to align the exposure with the estimation. This new FPGA beam blanking signal, consisting of a series of logic ‘1’s and ‘0’s according to the desired pattern, will be in charge of switching the beam on and off.

An M-function was programmed in MATLAB language to create a test pattern as illustrated in Figure 4.9. This function is imported to the SG design by “MCode” block, which allows its input values to be applied to the input arguments of the M-function and the output ports export the values determined by the output arguments. In the DSP design for pattern generation, as shown in Figure 4.10, the “MCode” block, denoted as “pattern_gen_para” , applies the pixel address, generated by the ‘ Pixel Addr. Generator” subsystem, to the input arguments of the M-function “pattern_gen_para”, then a logic ‘1’ or ‘0’ appears at the output port at that specific pixel location according to the pattern definition. Although the M-function was originally designed to perform the exposure with pixel size of 40 nm, the input “ratio”, determined by dividing 40 nm by the desired pixel size, allows to choose various pixel sizes to expose the same test pattern. As the pixel size and dwell time change for proper exposure, the count limits of each counter in the design need to be adjusted accordingly.

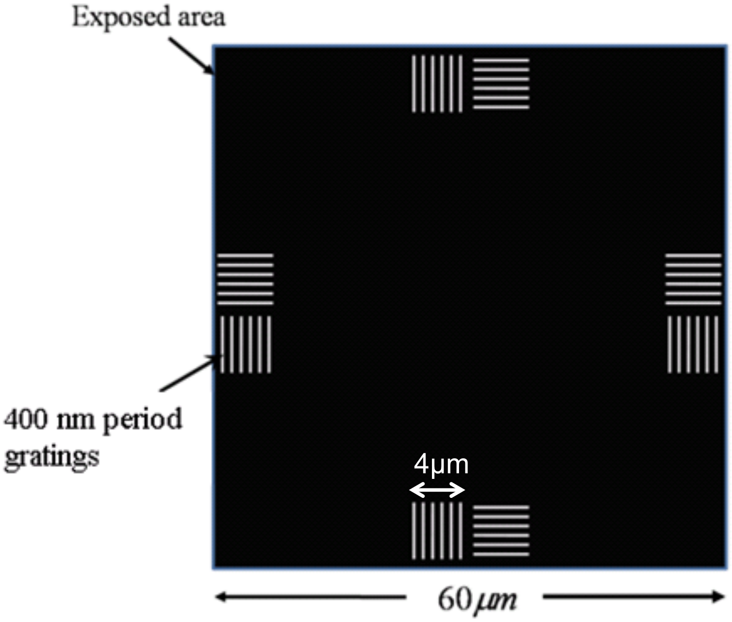
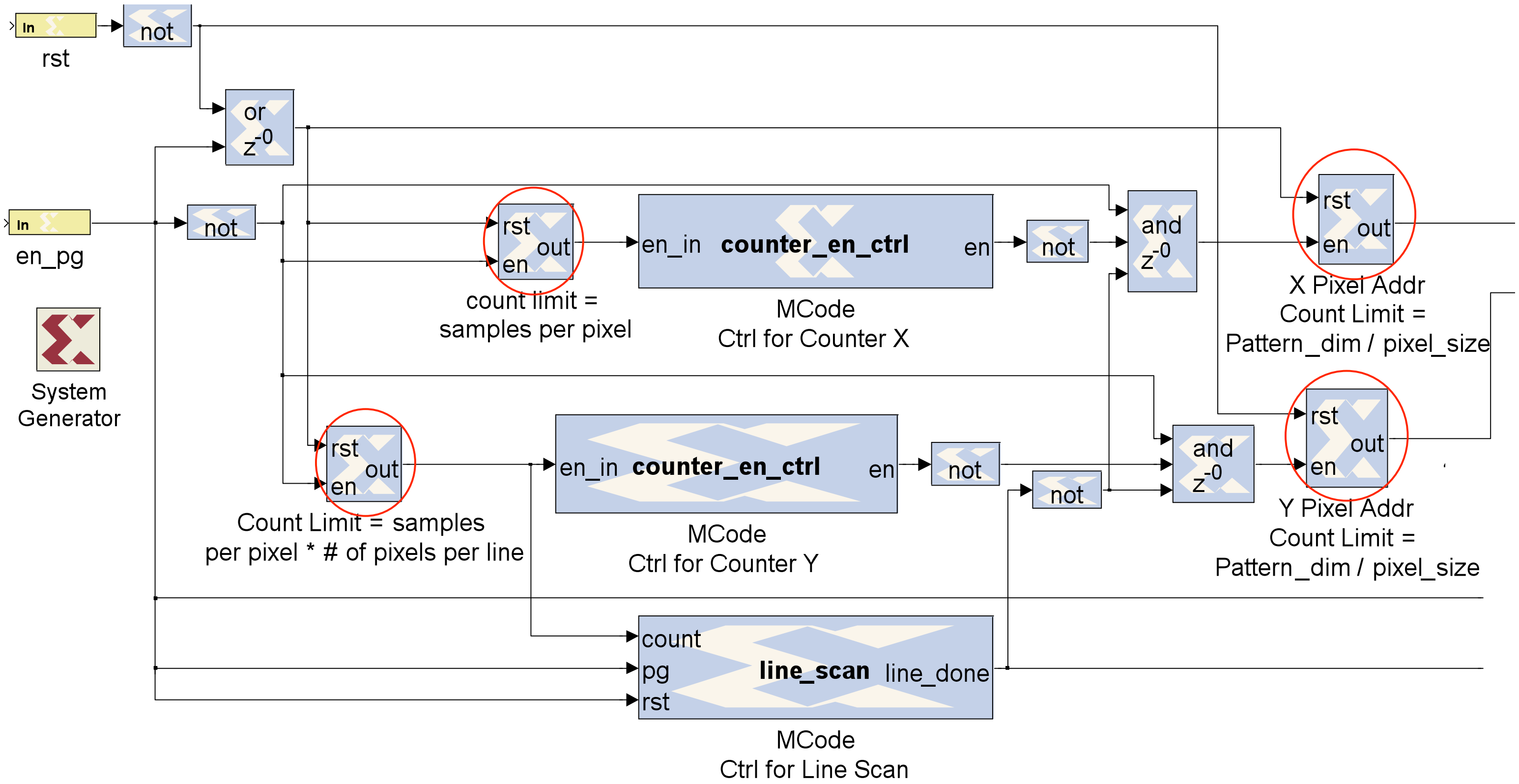
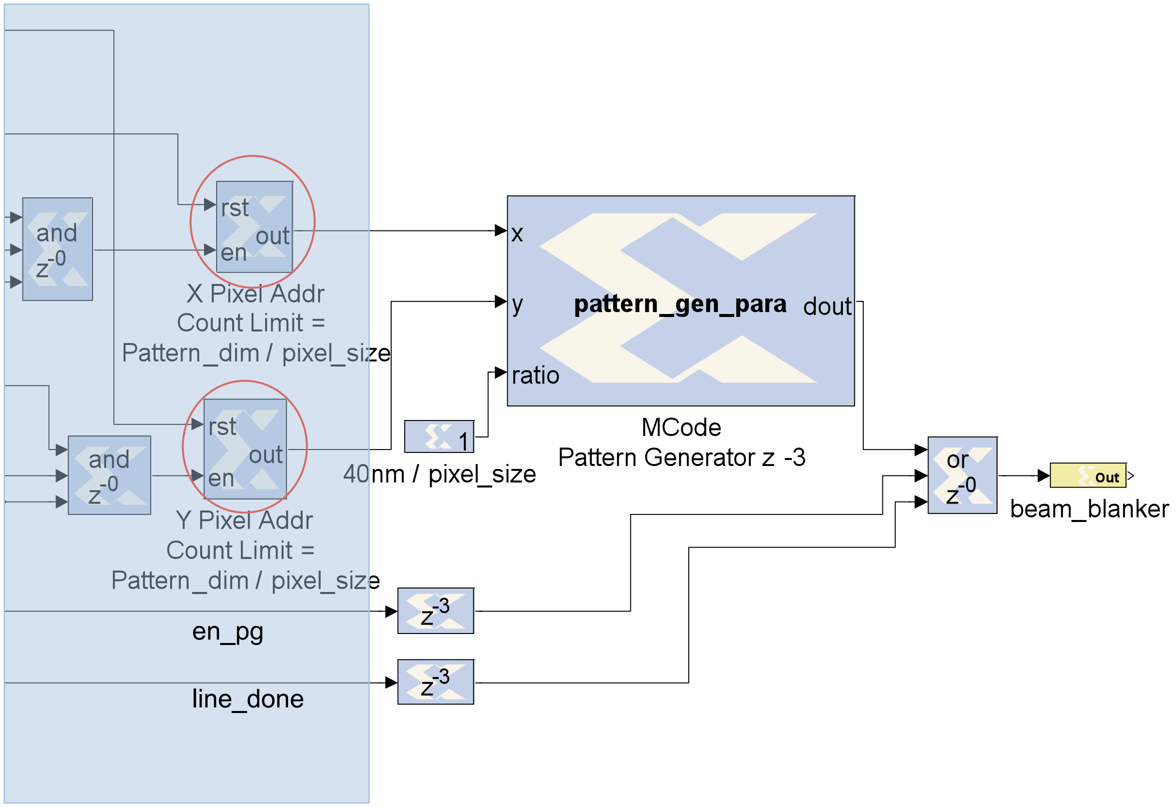


Figure 4.9: Illustration of the design pattern programmed in FPGA pattern generator. E-beam will be blanked when writing half pitch of the 400 nm period gratings with 50% duty cycle, illustrated as bright lines, and the dark area will be exposed.


In
rst
not
or
z-0
In
en_pg
not
rst
out
en
count limit =
samples per pixel
en_in
counter_en_ctrl
en
MCode
Ctrl for Counter X
not
and
z-0
rst
out
en
X Pixel Addr
Count Limit =
Pattern_dim / pixel_size
System
Generator
rst
out
en
Count Limit = samples
per pixel * # of pixels per line
en_in
counter_en_ctrl
en
MCode
Ctrl for Counter Y
not
not
and
z-0
rst
out
en
Y Pixel Addr
Count Limit =
Pattern_dim / pixel_size
count
pg
rst
line_scan
line_done
MCode
Ctrl for Line Scan

(a)



(b)

Figure 4.10: Schematics of System Generator design for generating test pattern by FPGA. Changing the count limits of the four counters (circled blocks) and the value applied to input “ratio” of “pattern_gen_para” block enables conducting exposure with various pixel sizes and dwell times.

Similar to the SG design for spatial-phase estimation, the beam blanking signal from Raith, programmed to expose the entire write field, is connected to the input "en_pg" to initiate the FPGA pattern generator and to synchronize the "Pixel Add. Generator" with the beam deflection. The output "beam_blanker" is routed to Raith Beam Blank Controller to switch the beam off at specific addresses to form the desired pattern.

The test pattern was exposed with dose of 99 μC/cm^2 at the primary electron beam energy of 10 KeV on 100 nm thick PMMA with various pixel sizes. The experimental setup is illustrated in Figure 4.11. Figure 4.12(a) shows the whole pattern exposed with pixel size of 20 nm and dwell time of 2 μs. To avoid large distortion at the left edge of the write field introduced by flying the beam back between each pixel line [83], the pattern is placed in the region of 30 μm $\leqslant x \leqslant$ 90 μm and 20 μm $\leqslant x \leqslant$ 80 μm. Identical patterns were exposed with pixel sizes of 40 nm, 10 nm and 5 nm, and dwell time of 8 μs, 500 ns and 125 ns respectively.

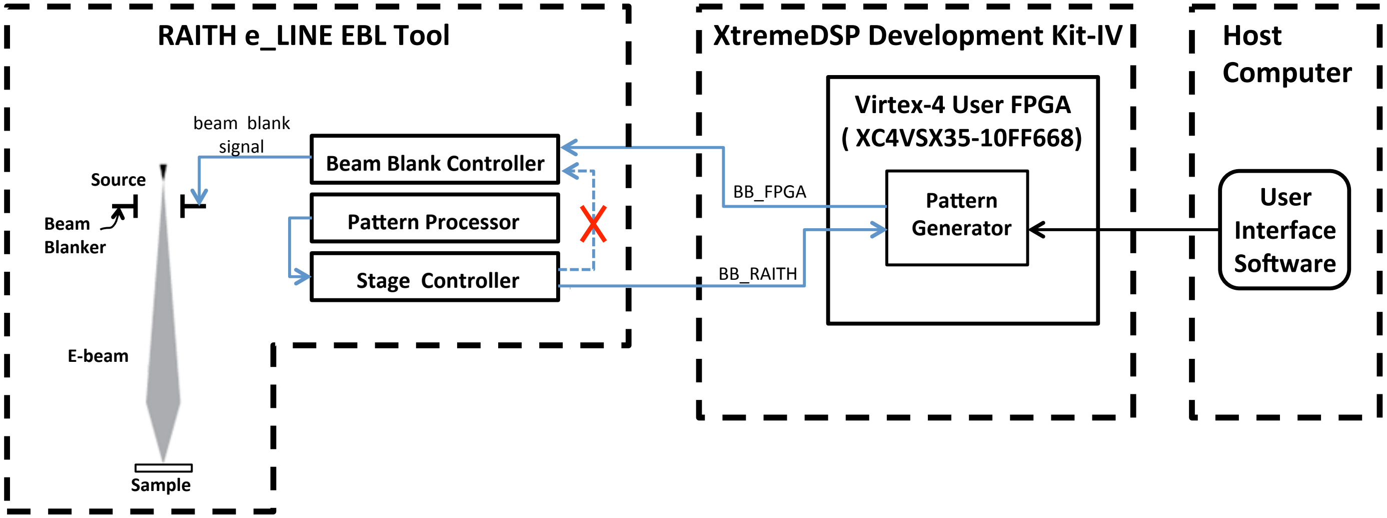


Figure 4.11: Schematics of experimental setup for exposing the test pattern with FPGA pattern generator.

There's a 5 μm difference in measured dimensions between x- and y-direction, suggesting the presence of a significant scaling error. This needs to be manually calibrated to allow the algorithm to accurately estimate the position error with respect to the grid period. Besides, an extra half period of the horizontal grating was exposed at the bottom of the pattern as shown in Figure 4.11(c), suggesting that the Raith software chose to expose more pixel lines than the theoretically calculated value using the specified pixel size. Although this is not fully understood, its effect on the FPGA design can be eliminated, and will be addressed in the following section. Despite the discrepancy, the exposed pattern indicates that the FPGA pattern generating component is performing as expected.

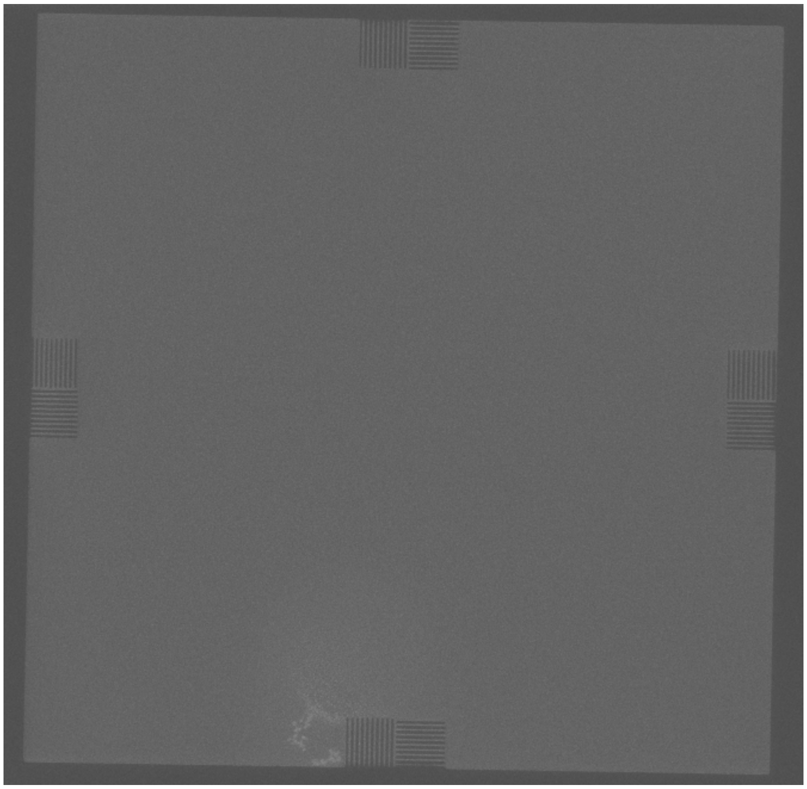
4.4 Memory Configuration for Distortion Correction

The limitations of the electron optics and beam deflection system cause the discrepancy between the actual and expected beam deflection across the write field, and the beam position error related to this can occur at a rate faster than the spatial-phase estimation can track. Typically, the field distortion is constant from field to field during exposure, therefore, one can calibrate it before conducting spatial phase estimation to obtain high precision beam position control.

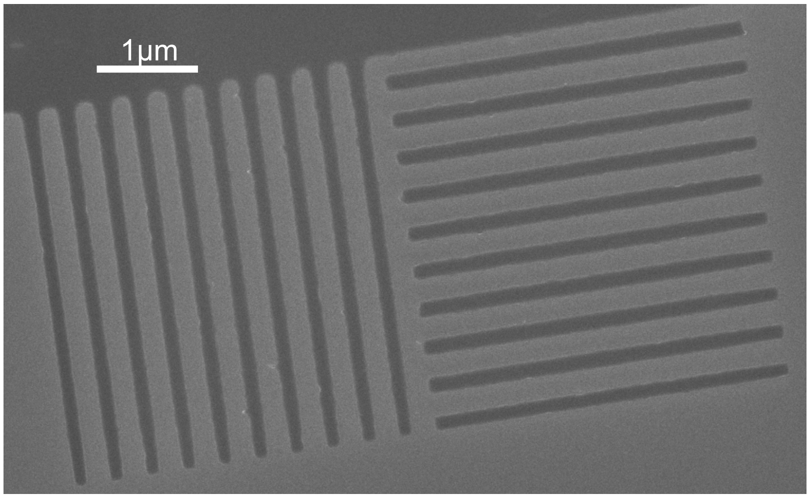
The first order field calibration, shift, scale and rotation, is often provided by electron beam lithography system. A feature on the sample is driven with the stage to three different locations within the write field, and SEM images are taken at each location by deflecting the beam by the same distance. With the measured positions of the feature x_B and y_B, and the intended position x_P and y_P, the distortion coefficients can be calculated according to the following equations:

$$\begin{aligned} x_B &= a_0 + a_1 x_P + a_2 y_P \\ y_B &= b_0 + b_1 x_P + b_2 y_P, \end{aligned} \tag{4.1}$$

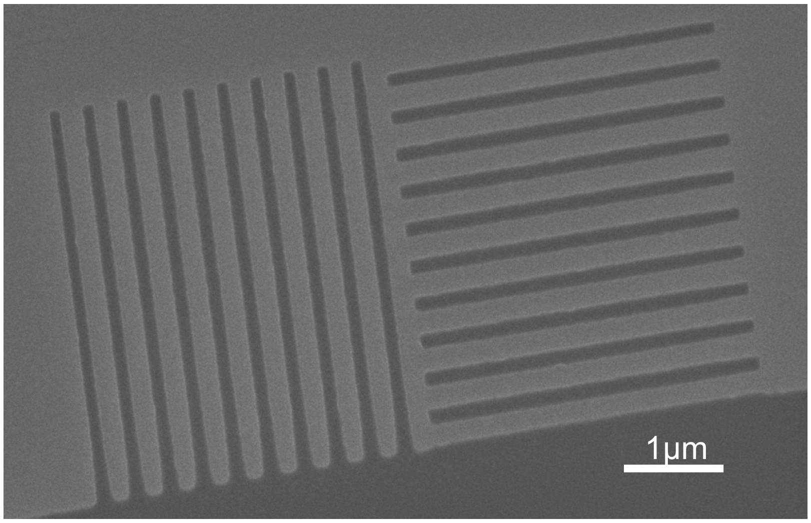
where coefficients a_0, a_1 and a_2 represent the shift, scale and rotation errors of x-axis



(a)



(b)



(c)

Figure 4.12: (a) SEM image of the test pattern exposed on 100 nm PMMA at beam energy of 10 KeV. pixel size = 20 nm; dwell time = 2 μs; beam current = 198 pA; area dose = 99 μC/cm^2. (b) SEM image of gratings on the top edge of the pattern exposed with pixel size of 40 nm and dwell time of 8 μs. The duty cycle can be adjusted by changing the exposure dose. (c) SEM image of gratings on the bottom edge of the pattern exposed under the same conditions as (b).

respectively, and b_0, b_1 and b_2 correspond to y-axis. Then appropriate correction can be applied to beam deflection system to correct the deviation.

The approach as described above can be extended to correct higher order field distortion if more measurements of beam positions are acquired. Instead of manually moving the stage to more locations and measuring the deviation of the feature from it's expected position in each SEM image, the beam position errors estimated by the FPGA, Δx and Δy, can be used to calculate the actual beam position given by equation 3.13, and thus the distortion coefficients. Therefore, the DSP design is modified to allow the estimated beam position errors to be stored in the memory, and a hardware design for retrieving the data from the memory is also developed.

4.4.1 HDL Design Entity for Memory Access

The two banks of Zero Bus Turnaround (ZBT) SRAMs on the development board are used to store the estimated x- and y-position errors respectively. Since the memory chips are driven exclusively by the Virtex-4 user FPGA, saving the estimated position errors is very straightforward (refer to Appendix B.2 for the HDL design). The outputs of the SG design for the position errors in x- and y-direction are connected to the two 32-bit data buses of the memories, and the FPGA generated memory addresses are assigned to the 19-bit address buses. The memory control signals, also generated by user FPGA, ensure that the estimates are loaded by the memory to the specific locations when they are ready.

The data transfer between the host computer and Virtex-4 user FPGA is through Spartan-II Interface FPGA. The Interface Communications Bus, consisting of a 32-bit Local Bus and a 7-bit Adjacent OUT bus, directly couples the interface FPGA to the user FPGA. In order to load the saved position errors from the memory to the host computer, a HDL entity, targeted for Virtex-4 user FPGA, is developed to bridge between the two banks of ZBT SRAMs and the Spartan-II Interface FPGA, thus to establish the communication between the memory and host computer.

The PCI-to-user-FPGA interface core, provided by Nallatech, is used to deal with the protocol over the Interface Communications Bus. The interface core supports two modes of data transfer: memory map interface for reading/writing registers; and direct memory access (DMA) for burst large amount of data across the PCI bus. The memory map interface can be useful for applying various parameters of spatial-phase estimation from the host computer to the user FPGA, in stead of compiling, synthesizing and implementing with the parameters initialized in the SG design. The memory map interface mode will be discussed in detail in Section 5.2. The remaining section will focus on information transfer to/from memory via DMA channel.

To better handle the signal latency among control, address and data signals required for memory's desired functionality, Nallatech ZBT controller is incorporated with the interface core. Figure 4.13 illustrates the signal routing specified in the HDL entity for user FPGA (refer to Appendix B.3 for the HDL design), as well as that among the FPGAs and memory. There are two registers associated to each IP core, control/status register and count register. For the interface core, the two registers (DMA_CSR and DAM_COUNT) are defined internally, while for the ZBT controller core, they are instantiated in the HDL design as ZBT_CSR and ZBT_COUNT. The value assigned to the control/status register determines the data transfer direction in the IP core, and the number of data to be transferred is loaded to the count register. Once the count reaches zero, the data transfer stops momentarily.

When writing to the memory, each address followed by the data to be written are sent to the ZBT controller via the DMA channel. While for reading data from the memory, the addresses are written across the DAM channel to the memory's address bus before the data transfer.

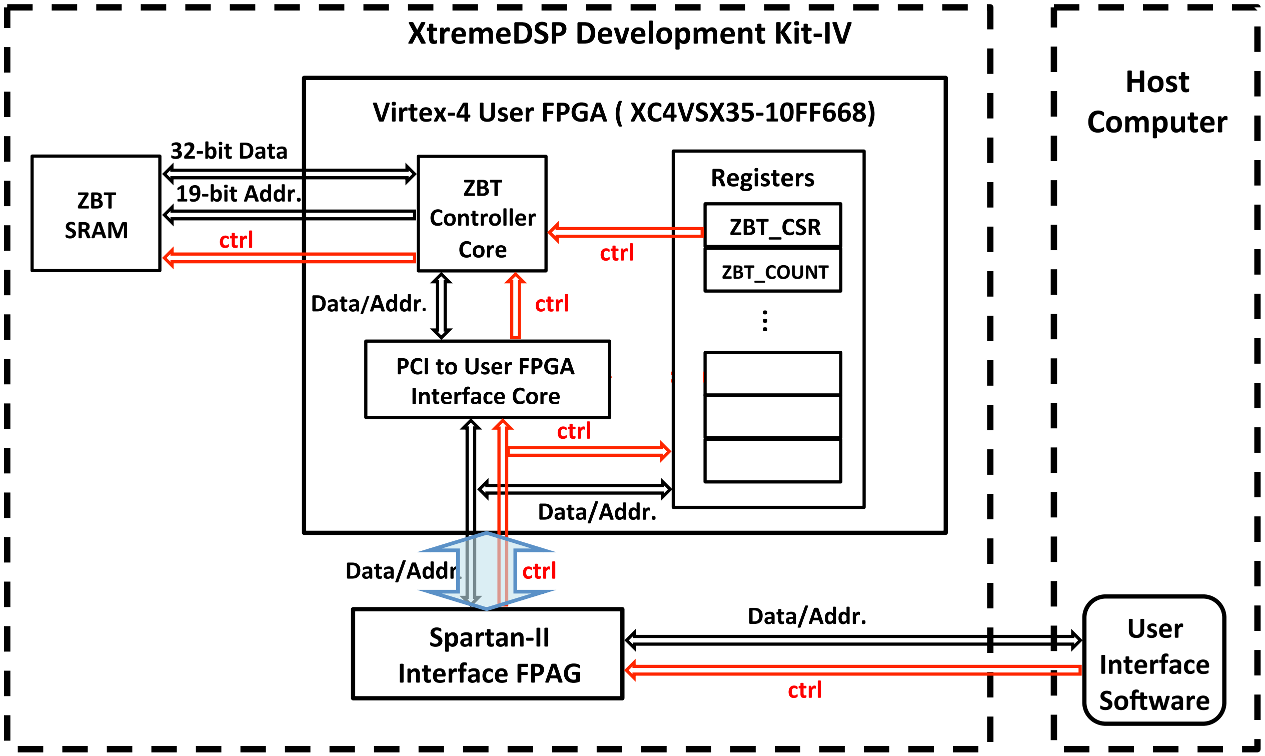


Figure 4.13: Schematics of the communication among interface FPGA, user FPAG and ZBT SRAM. Only one bank of SRAM is shown here, while both are included in the experiment and the second bank of memory is connected to the user FPGA in the same manner. The blue arrow between interface FPGA and Virtex-4 user FPGA represents the Local Bus, which is shared by the “ctrl” and “Data/Addr” signal for both interface core and ZBT controller core.

A script, programmed in DIMEscript language, is executed to control the modules on the motherboard and to define the IP cores' actions by writing specific values to the four registers. The data or/and addresses to be written across the DMA channel to the memory are saved in a text file beforehand, and then are loaded into a defined data area by the script. The data from the memory are also loaded in a data area, which are then saved in a designated text file.

4.4.2 Experimental Verification of Memory design

In order to verify the hardware design that establishes the communication between the host computer and the memories, we conducted the experiments to record the secondary electron signal from the fiducial grids and reconstruct the grid using the saved data from the memory. In addition, with the reconstructed signal, the grid's parameters, such as SNR, grid period and rotation angle, can be analyzed.

The SG model as shown in Figure 4.14 is designed to utilize the full capacity of both ZBT banks to store grid signal. The subsystem "Calculating Ave. Sig. per Pixel" serves to calculate the average grid signal from each pixel when the ADC samples faster than EBL Pattern Generator pixel rate. The beam blanking signal "Beam_Blanker", synchronized with and derived from the FPGA generated pixel address, controls the rest of the design and will be routed to the beam blanking electrode to switch the beam on and off. Therefore, the signal acquisition, calculating the average grid signal, and writing to the memory are all aligned. To prevent the problem of overwriting to the memory due to the extra lines of exposure at the bottom of the write field as discussed in Section 4.3, a finite state machine is included in the "Generating Pixel Addr." subsystem (Appendix A.2.2) to bring "Beam_Blanker" to logic high once every FPGA-generated pixel has been addressed, and to maintain the state until the system is reset

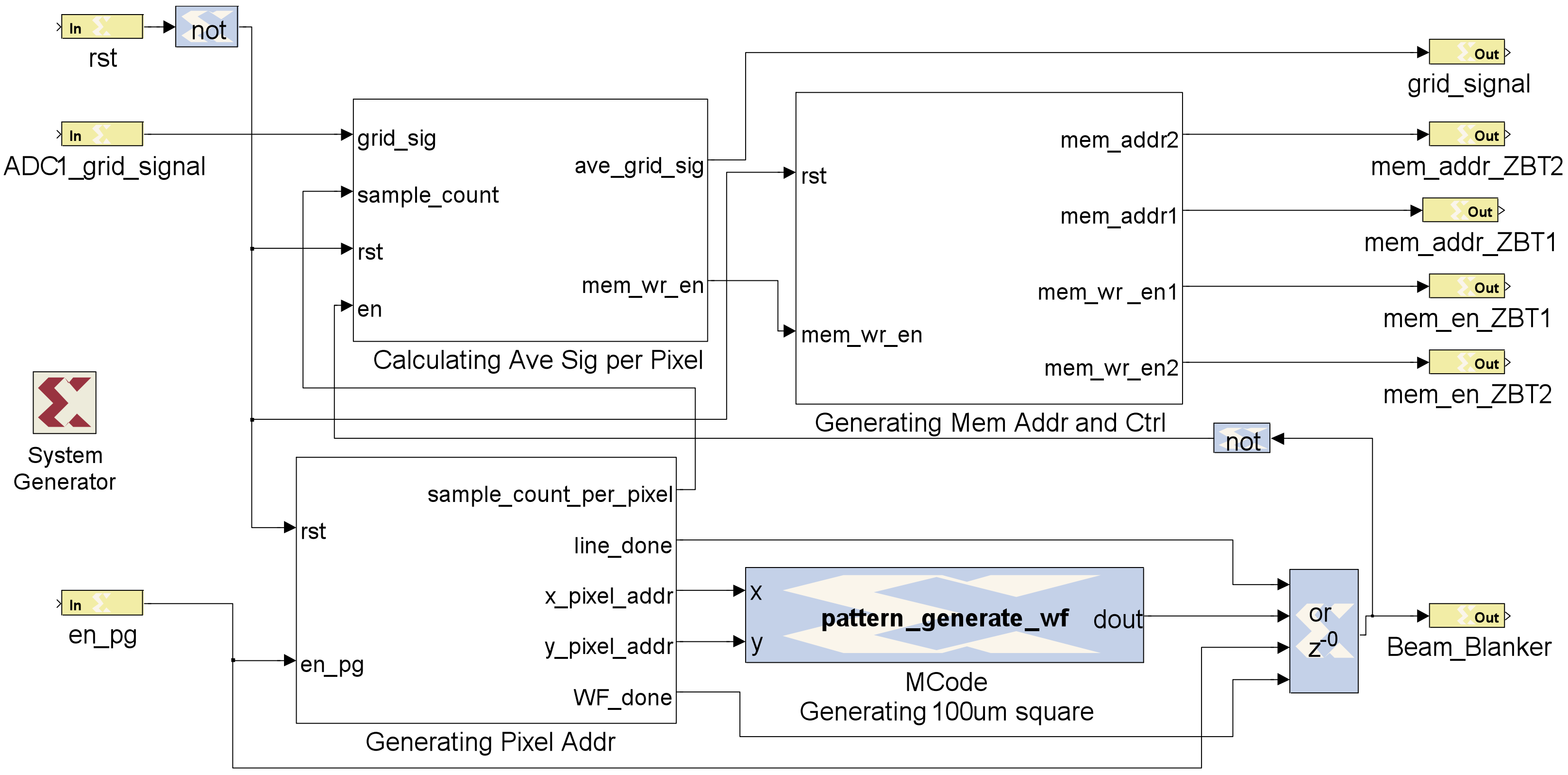


Figure 4.14: System Generator design for saving grid signal in the ZBT SRAMs during scanning over the write field. Refer to Appendix A.2 for subsystem details.

Figure 4.15 demonstrates the experimental setup for acquiring and saving grid signal while scanning the e-beam within a write field. A 200 nm period SiO_2 fiducial grid on Si was rotated by 25° and was sampled with the pixel size of 10 nm at the pixel rate of 10 MHz. The ADC samples at 40 MHz and four samples were acquired at each pixel address, however only the average value was saved to the memory. Due to the limited memory volume, the average grid signals from the first 104 scanning lines were saved, and Figure 4.16 shows the 2D secondary electron signal of 500×100 pixels captured at the top left corner of the field. The black strip indicates the delay between the beam blanking signal issued by the FPGA and the secondary electron emission upon the incidence of primary electrons, which can be attributed to the signal path delay on FPGA and the rise time of the beam blanker in the electron-optic column.

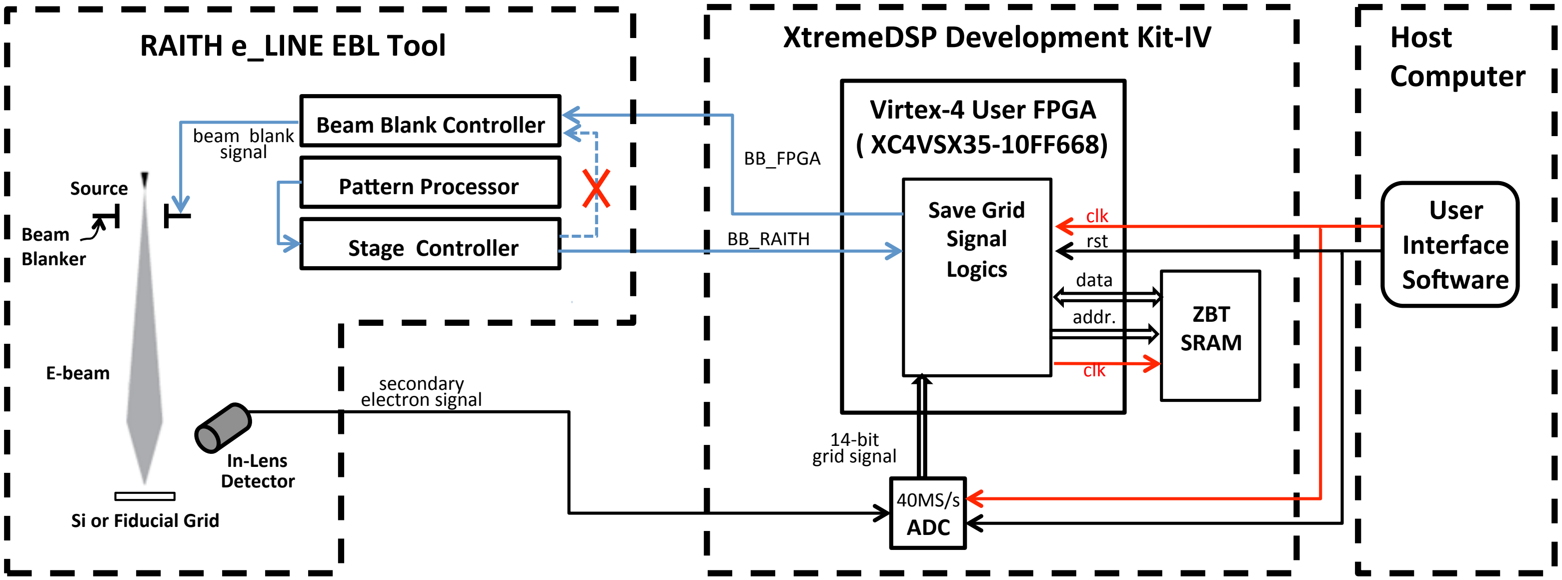


Figure 4.15: Schematics of the experimental setup for saving the grid signal to the memory. Only one ZBT bank is shown in the schematics, while both are included in the experiment.

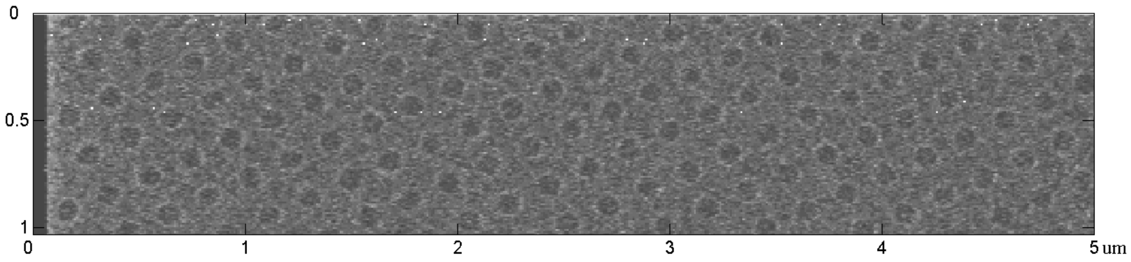


Figure 4.16: Secondary electron signal captured by ADC and saved to two ZBT SRAMs while electron beam is scanning over the write field. The data are read from the memory to the host computer and plotted in MATLAB. Only the top left corner of the write field is plotted to show the details.

To characterize the delay, an experiment in the same setup was conducted to collect the secondary electron signal at 50 MHz from bare Si. The saved signal at the beginning of one of the scanning lines is plotted in Figure 4.17, indicating the secondary electron emission didn't occur immediately after a logic '0' is asserted to the FPGA beam blanking signal. To characterize the delay time, we define the e-beam to be fully switched on when the amplitude reaches 90% of the maximum. The average delay time calculated over 20 scanned lines is 660 ns, corresponding to 26 clock cycles for 25 ns FPGA clock period. To compensate the delay, the SG design was modified as depicted in Figure 4.18, and the 2D signal sampled on the same grid is shown in Figure 4.19, which indicates that sampling grid signal is synchronized with the secondary electron emission. It is important to note that the enhanced secondary electron emission on the edge of the grids could introduce significant interference with fundamental frequency components, leading to reduced performance of the spatial phase estimation.

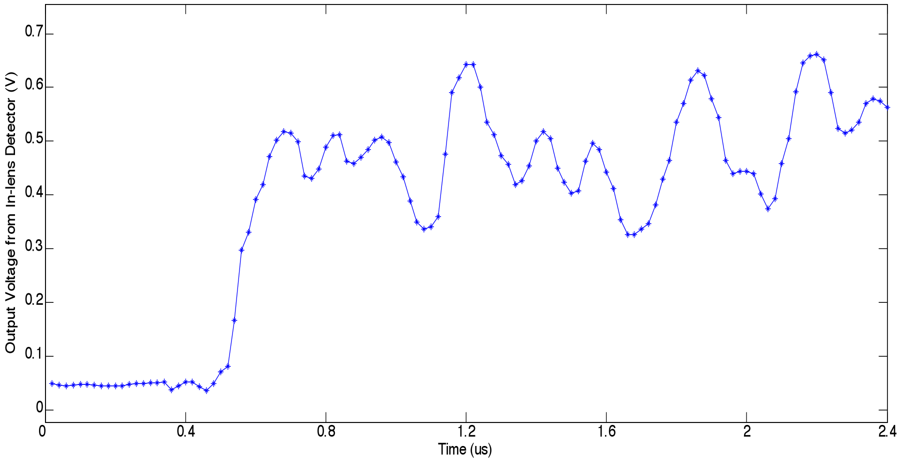


Figure 4.17: Sampled secondary electron signal on Si vs. time at the beginning of the scanning line along x-axis. The sample rate is 5 times of the EBL Pattern Generator pixel rate, resulting in 5 samples acquired at each pixel address.

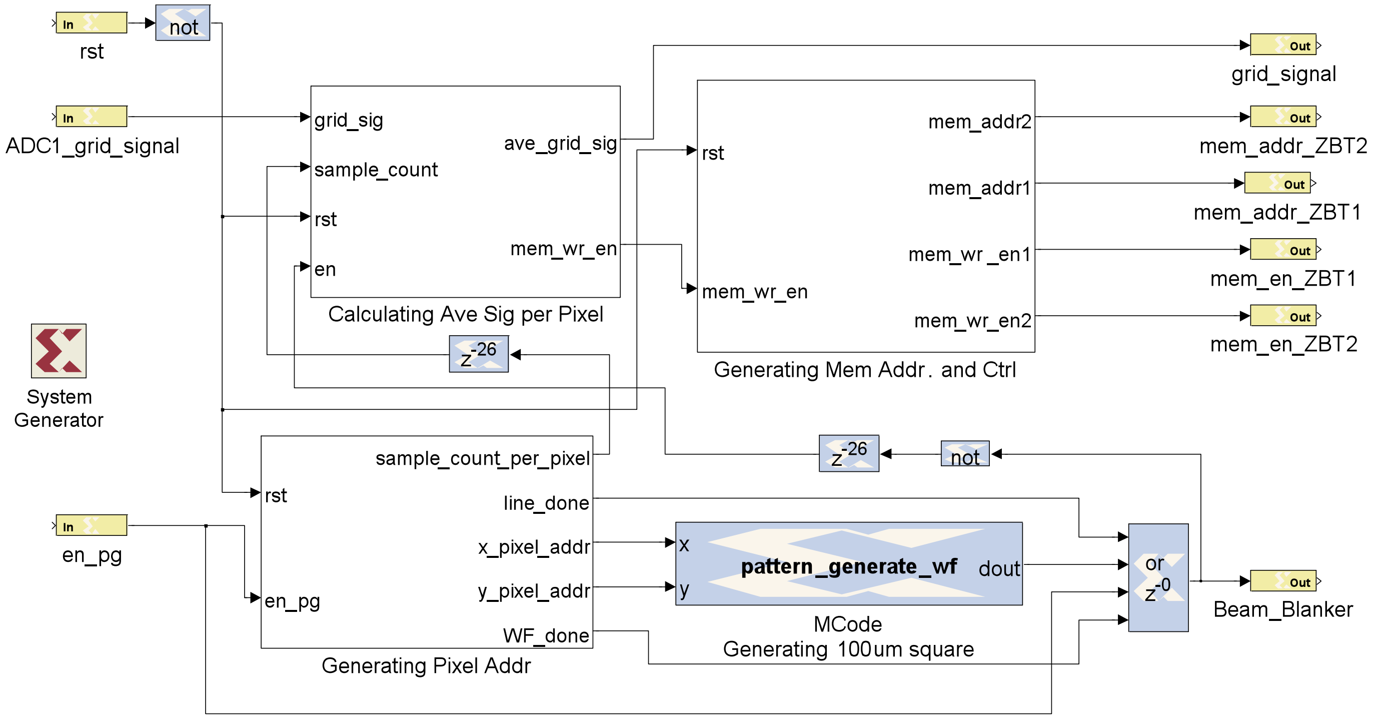


Figure 4.18: System Generator design for saving grid signal to the ZBT SRAMs during scanning over the write field with compensation for delay.

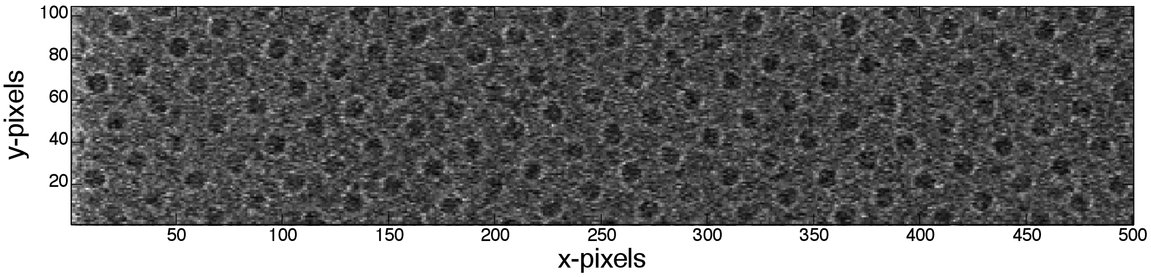


Figure 4.19: Secondary electron signal captured from 200 nm period SiO_2 grids by ADC and saved to two ZBT SRAMs while e-beam is scanning over the write field with the hardware delay compensated.

Chapter 5

Real-time Electron Beam Position Estimation

Before closing the feedback loop for reliable e-beam position control with desired precision, the FPGA implementation of the spatial phase estimation needs to be quantitatively evaluated in real time. For the evaluation, the fiducial grid needs to be investigated first to calibrate the estimation parameters, such as grid period and rotation angle, with the measured values. Secondly, a field distortion map needs to be constructed using the estimated beam position errors, so that high order distortion coefficients can be calculated and appropriate correction can be applied to the beam deflection system before conducting estimation. Finally, a filter, that attenuates the error introduced by the estimator but passes through that caused by physical disturbance, needs to be included after the estimation.

5.1 Analysis of Fiducial Grid

With the saved fiducial grid signal as shown in Figure 4.19, one can take the 2D Fourier transform as shown in Figure 5.1. A Hann window function was applied to the grid signal to reduce the interference from the harmonics. However, the 2nd and higher order harmonic frequency components are still much more significant compared to the fundamental frequencies, and SNR of the grid signal is calculated to be 0.02.

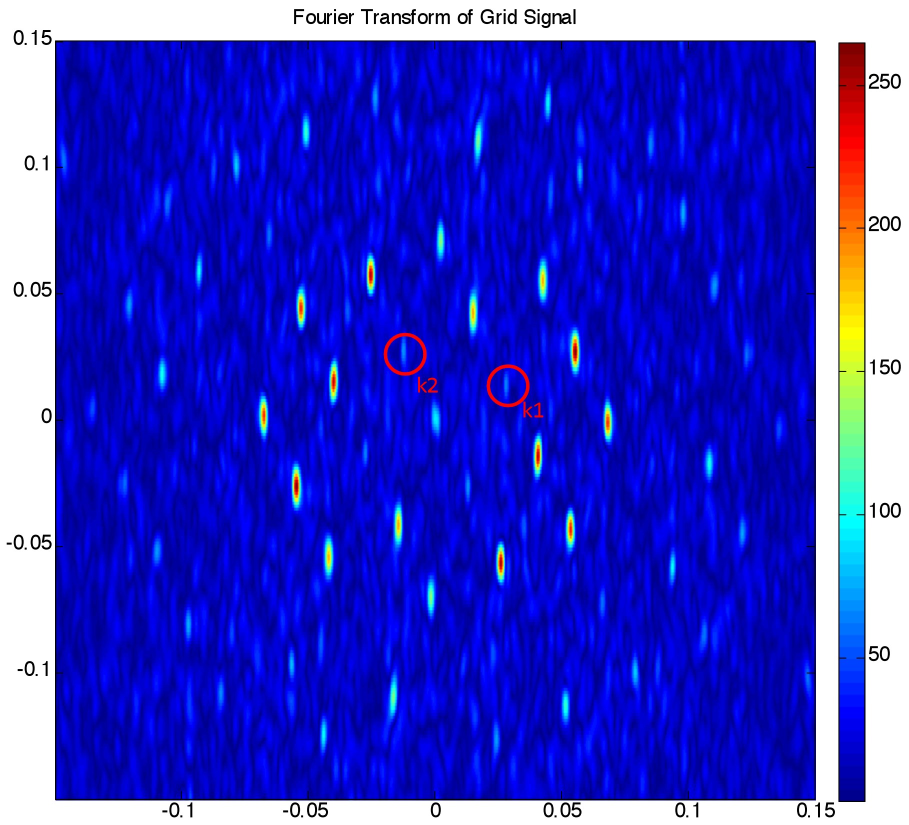


Figure 5.1: 2D Fourier transform of the grid signal as in shown Figure 4.19. A 2D-Hann window was applied to the signal to attenuate the interference from the harmonics. The circled are the two fundamental frequency components: $k_1 = (k_0 \cos\theta, k_0 \sin\theta)$ and $k_2 = (-k_0 \sin\theta, k_0 \cos\theta)$. Strong interference from 2nd and higher order harmonics will significantly reduce the performance of the spatial phase estimator.

This suggests that the estimation algorithm described in Section 3.2 and implemented in Section 4.2 will not be efficient if such grid is used as the reference, as the spatial phase is estimated at the fundamental frequencies.

Figure 5.2 shows the secondary electron signal obtained from a 200 nm period PMMA grid on Si with rotation angle of 25°. It is patterned on 70 nm thick PMMA using electron beam lithography. An array of single pixel dots in the area of 150 μm $\times$ 150 μm was exposed in the center of a 200 μm write field, which eliminates the severe distortion on the edge of the write field.

By taking the Fourier transform on 1000 samples and calculating the average amplitudes in the spatial frequency domain, the spatial-frequency spectrum of the grid

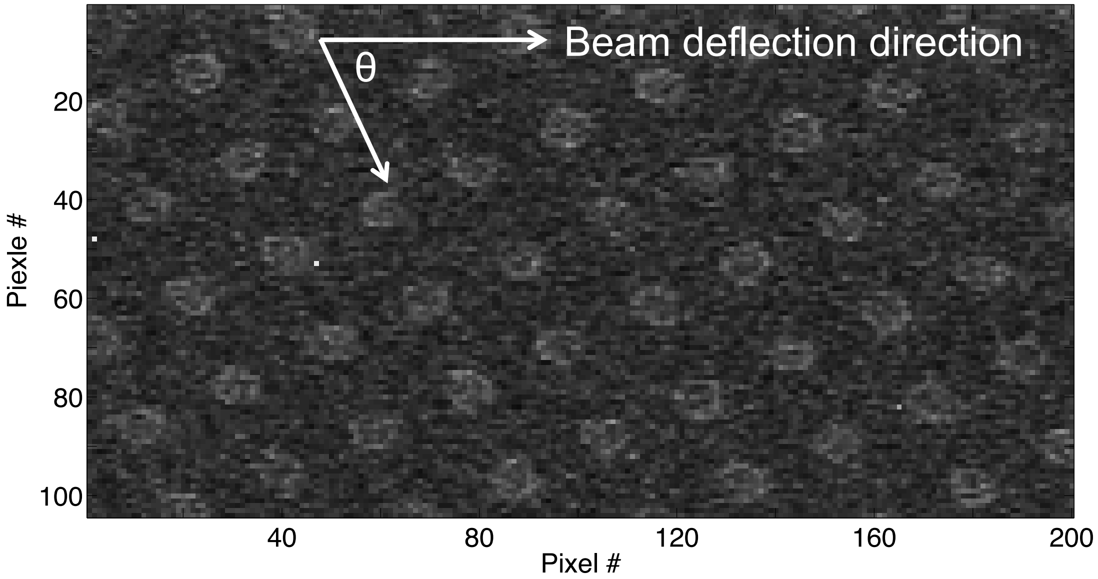


Figure 5.2: Secondary electron signal from 200 nm period PMMA grid. Dose of 0.016pC was used to expose an array of single pixel dots. 10000 × 104 pixels were sampled and stored in the memory, the 2D image shown here is reconstructed by a section of the samples.

signal is obtained as plotted in Figure 5.3. The two fundamental frequency components k_{LO} and k_{HI} are easily distinguished from the base noise and the harmonics, and the signal to noise ratio was calculated to be 0.25, promising the improved performance if the spatial phase estimation performed on such grid. However, the proximity of harmonic frequency component $k_{HI} - k_{LO}$ to the fundamental frequency k_{LO} could cause increased variance of estimation. Therefore, a window function should be applied to the data to eliminate the interference.

Once the two fundamental frequencies are identified on the spatial frequency spectrum, the grid's rotation angle and the period are determined to be 65° and $205.1nm$ according to:

$$k_{LO} = \frac{2\pi}{\lambda} \cdot \cos\theta \tag{5.1}$$

$$k_{HI} = \frac{2\pi}{\lambda} \cdot \sin\theta. \tag{5.2}$$

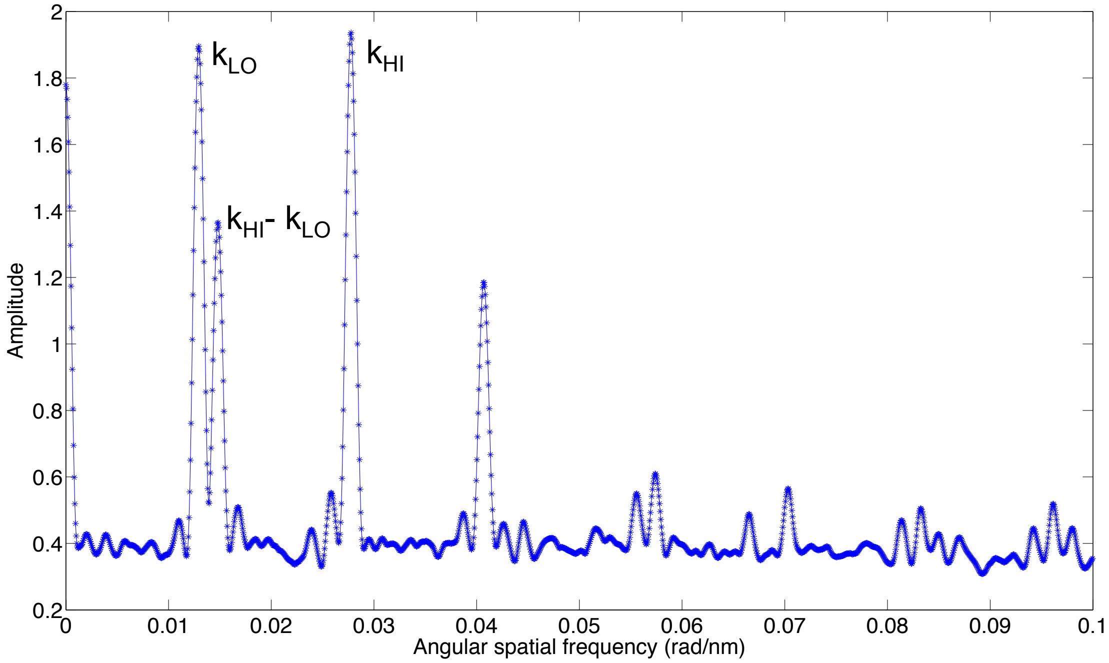
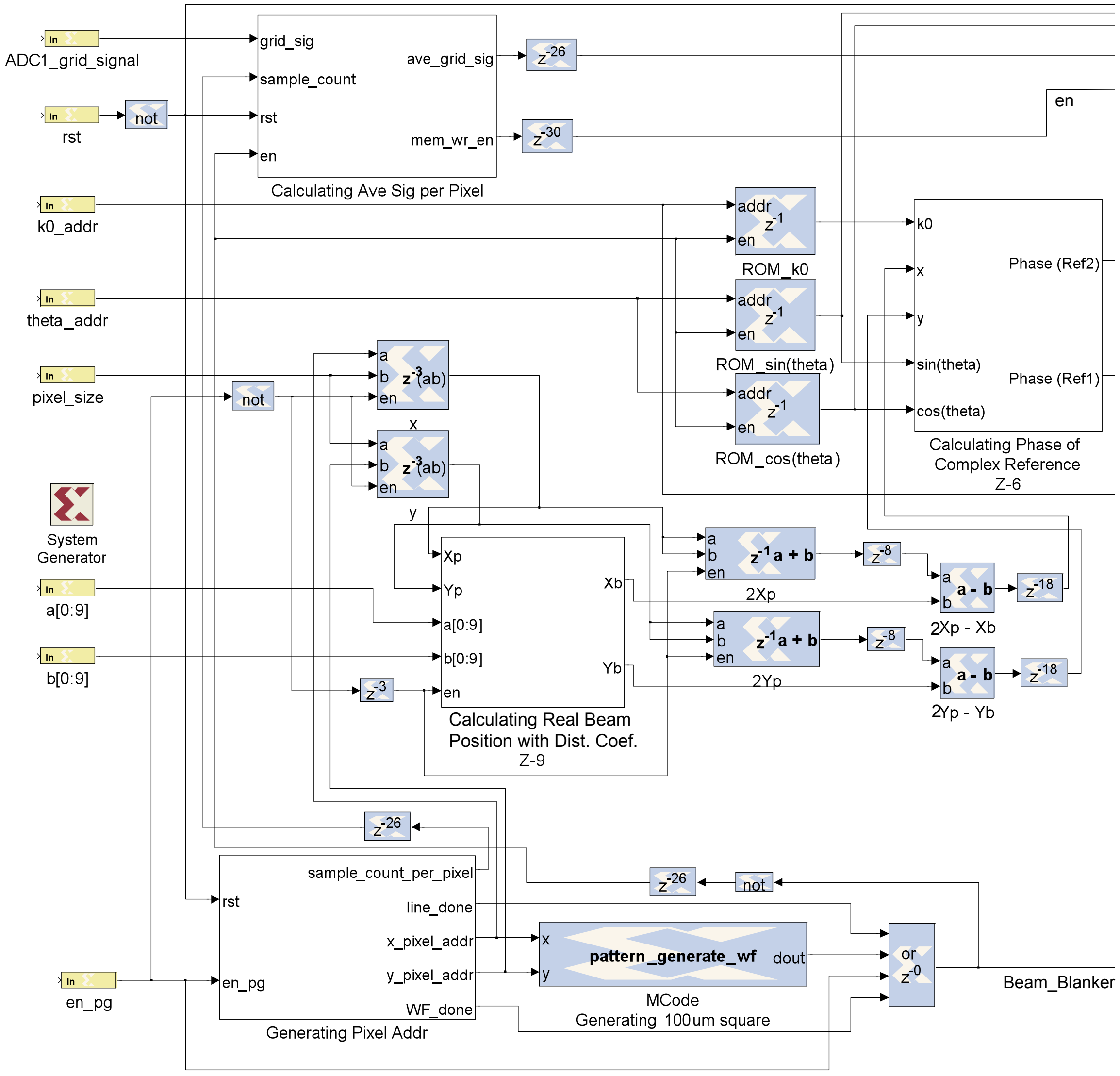


Figure 5.3: Average spatial-frequency spectrum of the grid signal. The Fourier transform is taken over 1000 samples and the average amplitudes over 104 lines are plotted out.

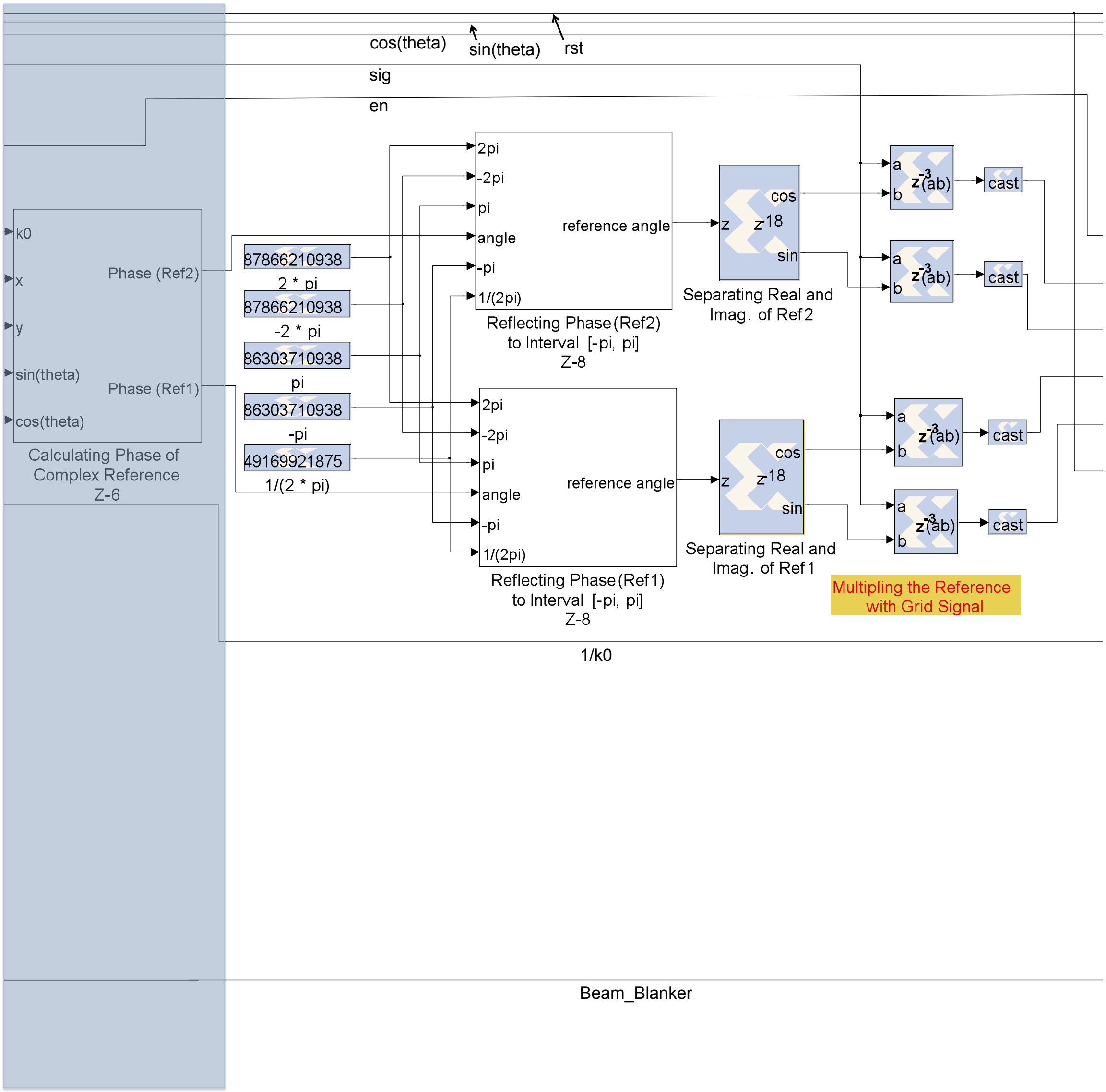
It is important to note that although the grid was physically loaded by 25° with respect to the deflection direction, the complementary angle 65°, as calculated above, should be used in the estimation as the origin of the algorithm's coordinates is defined at the top left corner. Besides, the discrepancy between the expected grid period and the measured value can lead to inaccuracy of estimated position error. In this case, 5 nm difference can cause error of estimation up to 40 nm across a 100 μm field. Therefore, the parameters to be used in the FPGA implemented estimator should be calibrated using the measured values.

5.2 Field Distortion Correction

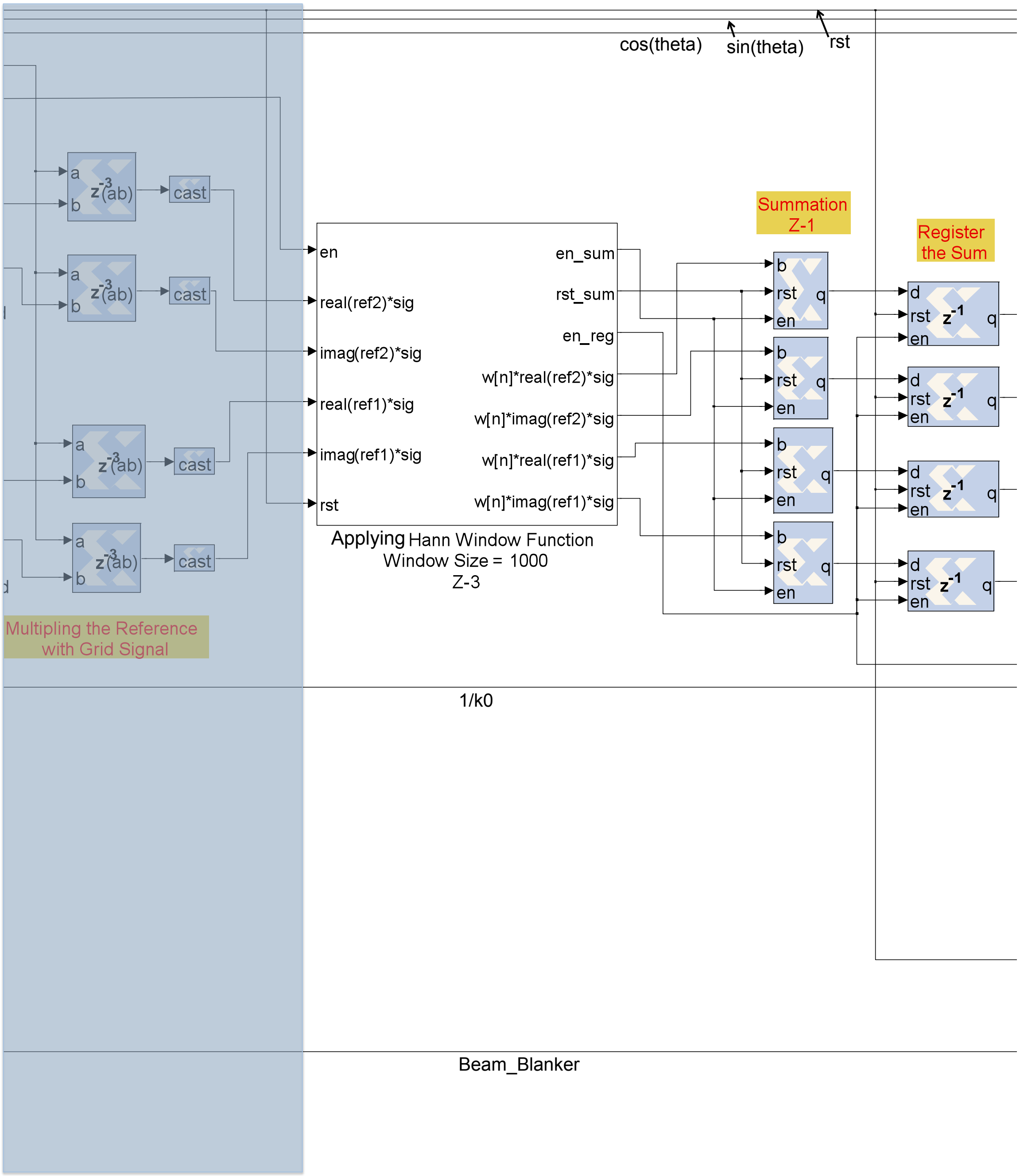
Several modifications in the hardware design were made to characterize the higher order distortion of the beam deflection system. The block diagram of the modified SG design is shown in Figure 5.4. The subsystem that calculates the average grid signal at each pixel is included and the average grid signal is used for estimating the position error. A subsystem that applies a window function to the samples is added, denoted as “Applying Hann Window Fucntion”, in order to minimize the interference from the harmonics. The reset signal, derived from the window function subsystem, is issued to the accumulator so that the integration is performed over the data of the same length as the window function. Once the accumulator is finished with one cycle of integration, the register captures the sum before the accumulator is reset, and stores the value until the next cycle is over. The memory addresses and control signals are generated in a manner analogous to the reset signal to ensure the synchronization with the estimated position errors.


ADC1_grid_signal
rst
not
k0_addr
theta_addr
pixel_size
System Generator
a[0:9]
b[0:9]
en_pg
grid_sig
sample_count
rst
en
ave_grid_sig
mem_wr_en
Calculating Ave Sig per Pixel
z-26
z-30
en
addr
z-1
en
ROM_k0
ROM_sin(theta)
ROM_cos(theta)
k0
x
y
sin(theta)
cos(theta)
Phase (Ref2)
Phase (Ref1)
Calculating Phase of Complex Reference Z-6
a
b
en
z-3(ab)
x
y
Xp
Yp
a[0:9]
b[0:9]
en
Xb
Yb
z-3
Calculating Real Beam Position with Dist. Coef. Z-9
z-1 a + b
2Xp
2Yp
z-8
a - b
2Xp - Xb
2Yp - Yb
z-18
z-26
not
sample_count_per_pixel
rst
line_done
x_pixel_addr
y_pixel_addr
en_pg
WF_done
Generating Pixel Addr
pattern_generate_wf
dout
MCode
Generating 100um square
or
z-0
Beam_Blanker

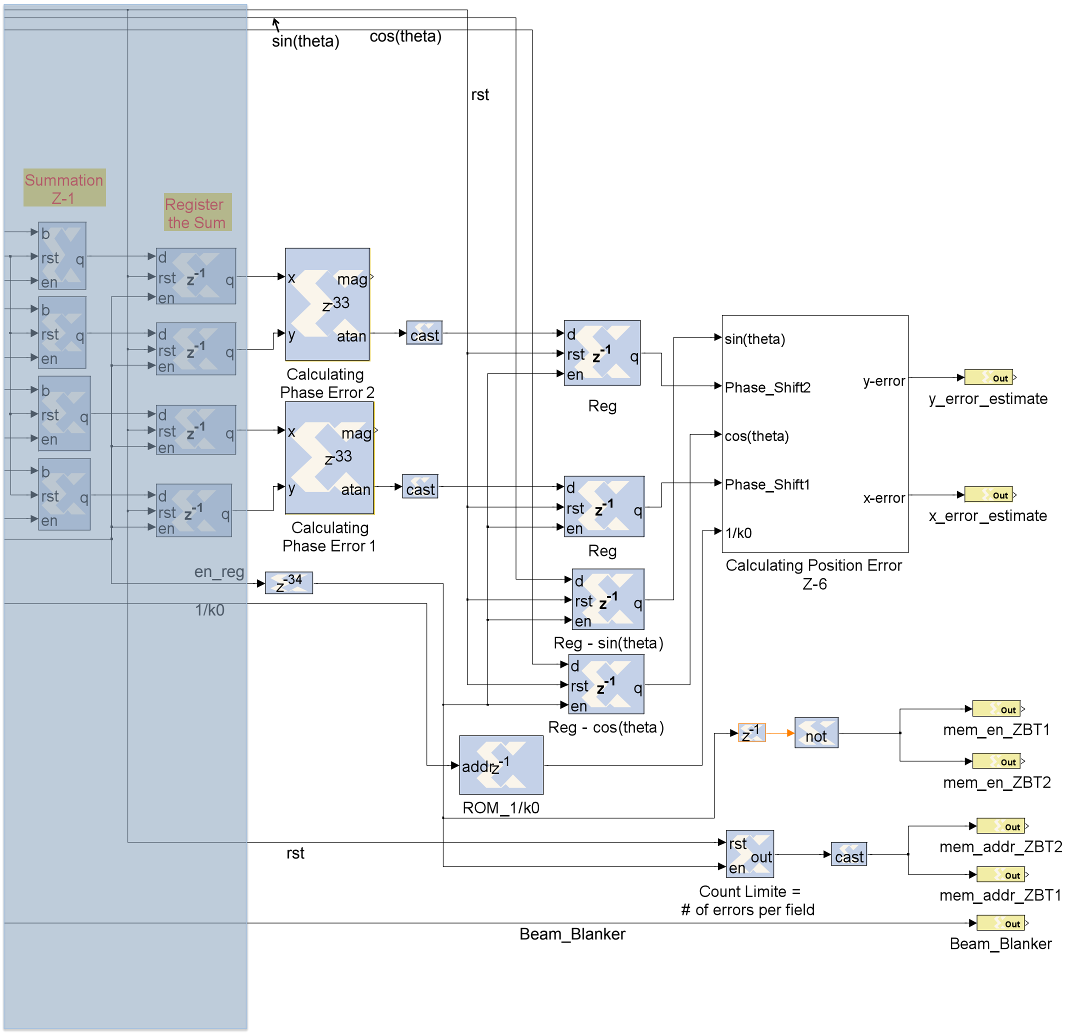
(a)


cos(theta)
sin(theta)
rst
sig
en
k0
x
y
sin(theta)
cos(theta)
Phase (Ref2)
Phase (Ref1)
Calculating Phase of Complex Reference
Z-6
87866210938
2 * pi
87866210938
-2 * pi
86303710938
pi
86303710938
-pi
49169921875
1/(2 * pi)
2pi
-2pi
pi
angle
-pi
1/(2pi)
reference angle
Reflecting Phase (Ref2) to Interval [-pi, pi]
Z-8
Reflecting Phase (Ref1) to Interval [-pi, pi]
Z-8
z
cos
z-18
sin
Separating Real and Imag. of Ref 2
Separating Real and Imag. of Ref 1
a
b
z-3(ab)
cast
Multipling the Reference with Grid Signal
1/k0
Beam_Blanker

(b)


cos(theta)
sin(theta)
rst
a
b
z-3 (ab)
cast
Summation Z-1
Register the Sum
en
real(ref2)*sig
imag(ref2)*sig
real(ref1)*sig
imag(ref1)*sig
rst
en_sum
rst_sum
en_reg
w[n]*real(ref2)*sig
w[n]*imag(ref2)*sig
w[n]*real(ref1)*sig
w[n]*imag(ref1)*sig
Applying Hann Window Function
Window Size = 1000
Z-3
b
rst
en
q
d
z-1
Multipling the Reference with Grid Signal
1/k0
Beam_Blanker

(c)



(d)

Figure 5.4: Schematics of System Generator design that estimates position errors and saves them to the memory. The saved position errors will be combined with the intended beam position to calculate the field distortion coefficients off-line. Refer to Appendix A.3 for details of subsystem “Applying Hann Window Fucntion”.

With the estimated position errors in x- and y-direction from the entire write field, one can randomly pick 10 pairs and calculate the second and third distortion coefficients given by [83]:

$$\begin{aligned} x_B =& a_0 + a_1 x_P + a_1 y_P + a_3 x_P y_P + a_4 x_P^2 + a_5 y_P^2 \\ & + a_6 x_P^3 + a_7 y_P^3 + a_8 x_P^2 y_P + a_9 x_P y_P^2 \end{aligned} \tag{5.3}$$

$$\begin{aligned} y_B =& b_0 + b_1 x_P + b_1 y_P + b_3 x_P y_P + b_4 x_P^2 + b_5 y_P^2 \\ & + b_6 x_P^3 + b_7 y_P^3 + b_8 x_P^2 y_P + b_9 x_P y_P^2. \end{aligned} \tag{5.4}$$

However, the error of estimation introduced by the local defect in the fiducial grid can lead to unreliable characterization of the field distortion. Therefore, the strategy described in Hastings' report on raster-scan SPLEBL [5] is adopted, in which the field is divided into 10 × 10 blocks, the average position error in each block is determined, and the least squares solution to the distortion coefficients is calculated.

In the top-level HDL entity for spatial phase estimation, the interface core is included to update the grid period and rotation angle on the FPGA with the measured values from the host computer. For each parameter, a register is instantiated in the HDL entity, and the interface core loads the data from the interface communication bus to the register using the memory map interface.

The 200 nm period PMMA grid on Si is loaded into Raith e_LiNE SEBL system, and the rotation angle and grid period are measure to be 65° and 205.1 nm respectively. The experimental setup is similar to that as illustrated in Figure 4.15 except that the hardware design for spatial phase estimation is downloaded to the FPGA. The grid signal is sampled at pixel rate of 40 MHz with pixel size of 10 nm, and the estimation is performed over 1000 samples for each pair of x- and y-direction errors. The first order write field alignment is performed using the procedure provided by Raith software, and the 2D estimated position errors are shown in Figure 5.5. The position error in x-direction suffers steep change when the beam is deflected to the edge of the field, while goes through a relatively smooth transition in the center. The extreme distortion on the edge is largely due to the "flyback" filling strategy. The

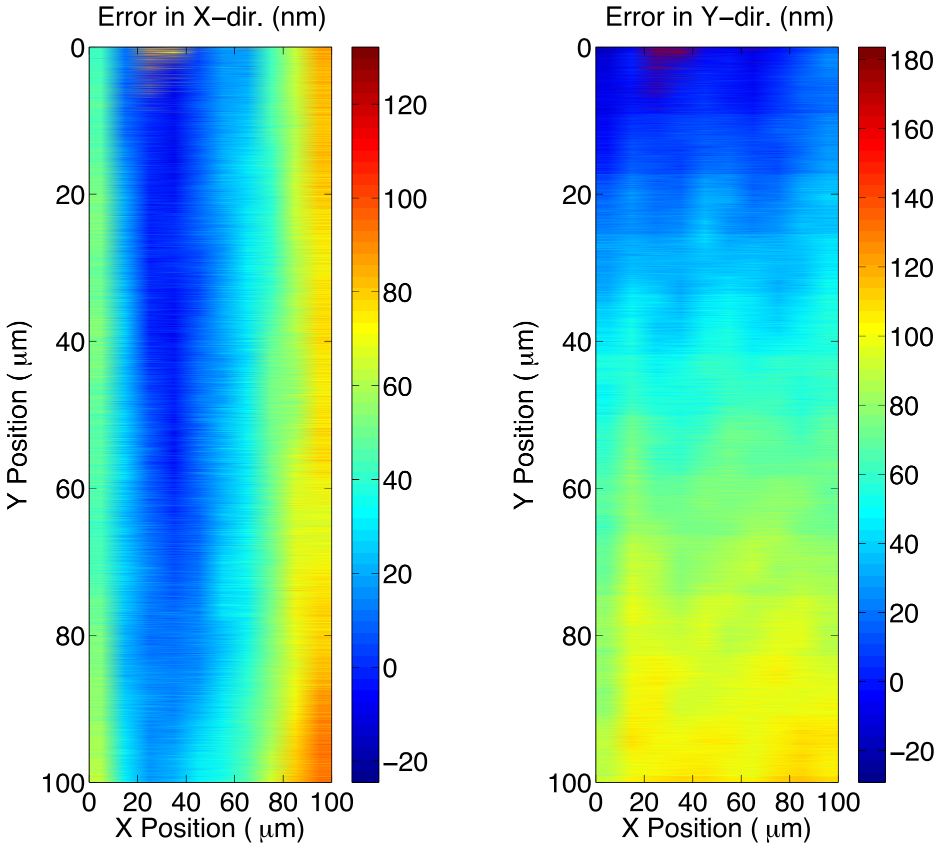


Figure 5.5: The 2D estimated position errors in x- and y-direction.

smooth change in the center indicates the presence of a scale error of the deflection along x-axis. As for position error in y-direction, it changes almost linearly as the scan proceeds from the top of the field to the bottom. This is more obvious as shown in Figure 5.6, in which the position error in y-direction is plotted as a function of the absolute beam position along y-axis of the deflection coordinates. The map of the estimated position errors, plotted in Figure 5.7(a), reveals the residual field distortion after the calibration for first-order distortion by Raith, which requires further correction for the control of the position with high precision.

Combining the estimated beam positions with the ideal ones, the 10-term correction coefficients of each axis are calculated and are given in Table 5.1. If a proper signal conversion from the FPGA board to Raith deflection system is established, one

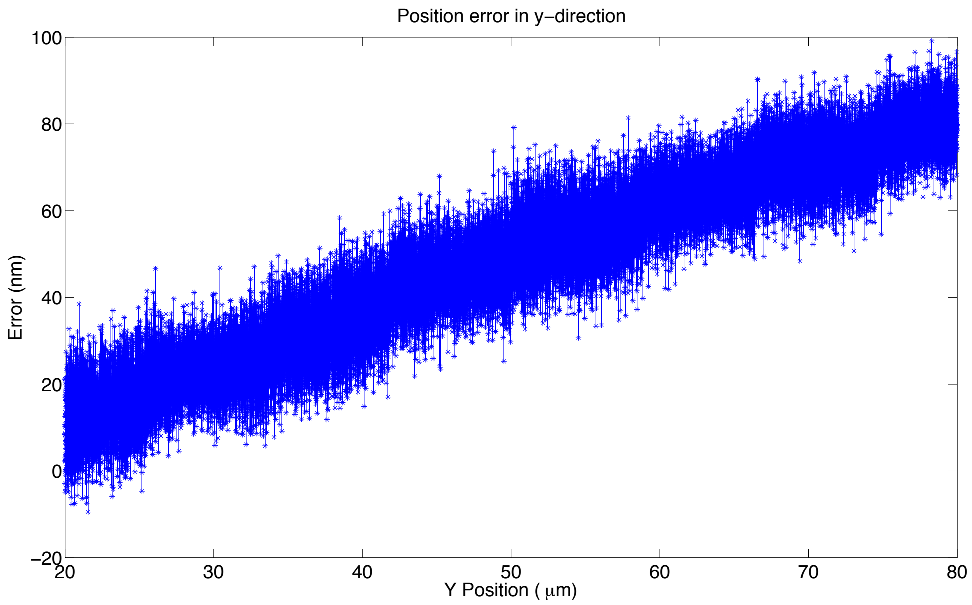


Figure 5.6: Position error in y-direction vs. absolute beam position along y-axis of deflection coordinates. Only the data from a 60 μm $\times$ 60 μm area in the center of the write filed are shown in the plot.

can subtract the deterministic errors characterized by the coefficients at each pixel location to compensate for the field distortion. Alternatively, this is demonstrated by correcting the coordinates of the estimation algorithm on the hardware as shown in Figure 5.4(a) when the feedback connection is not in place.

It is important to note that the coefficients for second and third order distortion, a_3 - a_9 and b_3 - b_9, are too small to be fully interpreted by the input ports that are assigned with a fixed data width. In the current implementation, each coefficient input port is allocated with 12 bits inclusive of the sign bit, resulting in the resolution of $2^{-11} = 4.9E - 4$. Therefore, the fitting is performed with the coefficients of shift, scale and rotation to be applied, and the calculated values are given in Table 5.2.

Then the position estimation was conducted with the 1st-order correction applied and other conditions remained the same. Comparing the field distortion maps as shown in Figure 5.7, which is constructed with the average position errors before and after applying the 1st-order correction, the distortion error has significantly reduced in the 60 μm$\times$60 μm area in the center of the write field after the correction. However,

Table 5.1: 10-term correction coefficients for higher-order field distortion calculated from the estimated position errors as plotted in Figure 5.5.

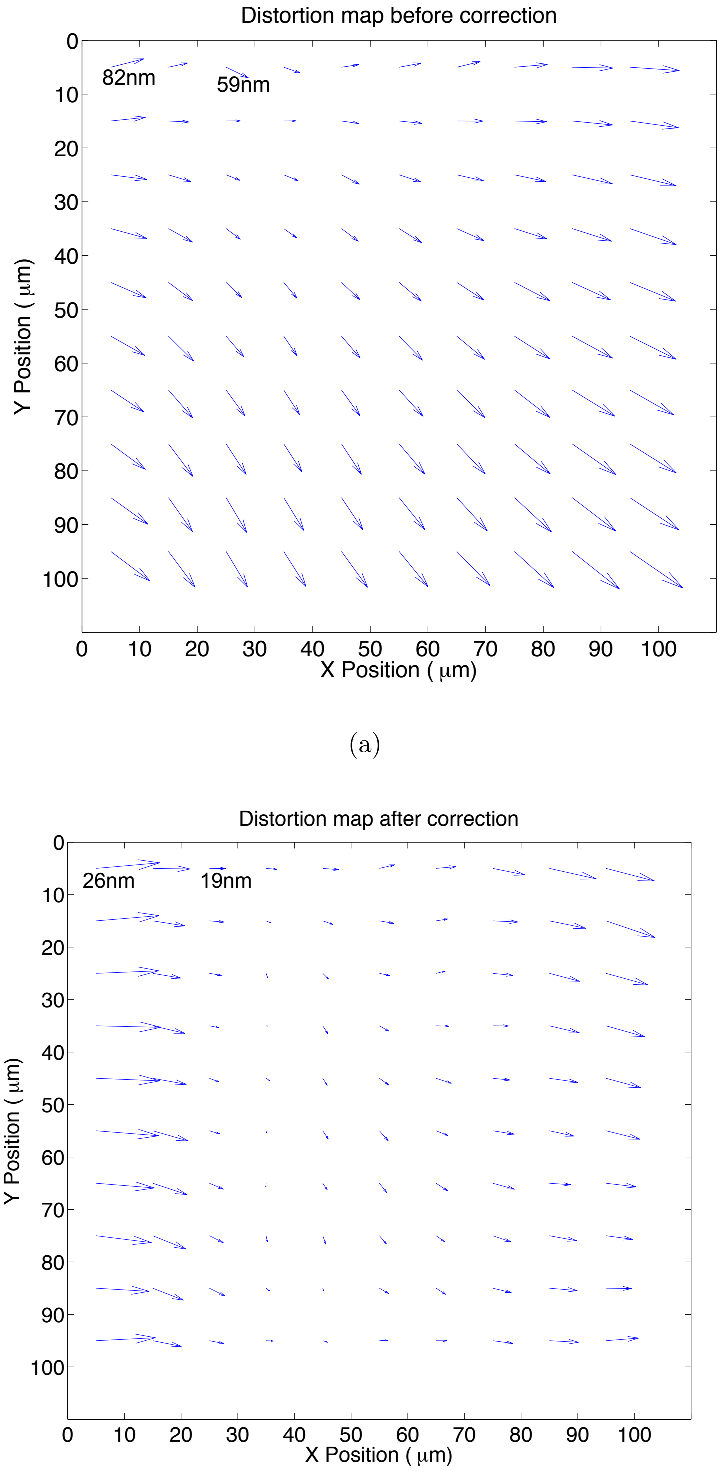
x-direction		y-direction	
a_0	$-4.71448E + 1$ nm	b_0	$-7.97487E + 1$ nm
a_1	$9.98172E - 1$	b_1	$5.42123E - 4$
a_2	$2.47500E - 3$	b_2	$1.00055E + 0$
a_3	$4.75603E - 8$ nm^{-1}	b_3	$5.72882E - 9$ nm^{-1}
a_4	$4.75603E - 8$ nm^{-1}	b_4	$-1.46935E - 8$ nm^{-1}
a_5	$-3.46433E - 8$ nm^{-1}	b_5	$1.49414E - 8$ nm^{-1}
a_6	$-2.08894E - 13$ nm^{-2}	b_6	$1.12545E - 13$ nm^{-2}
a_7	$1.69742E - 13$ nm^{-2}	b_7	$-1.18328E - 13$ nm^{-2}
a_8	$1.08672E - 13$ nm^{-2}	b_8	$-5.69044E - 14$ nm^{-2}
a_9	$1.96164E - 13$ nm^{-2}	b_9	$-1.48980E - 15$ nm^{-2}

Table 5.2: Correction coefficients for 1st-order field distortion calculated from estimated position errors as plotted in Figure 5.5.

x-direction		y-direction	
a_0	$-4.71448E + 1$ nm	b_0	$-7.97487E + 1$ nm
a_1	$1.00052E + 0$	b_1	$8.52598E - 5$
a_2	$1.44767E - 4$	b_2	$1.00108E + 0$

relatively large position errors in x-direction are still present in adjacent to the left and right edge of the field, which is attributed to the flyback exposure strategy. The standard deviation of the estimated position errors after the correction in x- and y-direction are calculated to be 8.8 nm and 6.7 nm respectively.

By substituting k_0, N and γ in Equation 3.18 with 205.1 nm, 1000 and 0.25 respectively, and multiplying a factor of 1.5 [81] to account for the increased variance associated with the window function, the Cramer-Rao bound is calculated to be 4.4 nm at the given experimental conditions. In addition to the uncorrected 2nd- and 3rd-order distortion, the errors introduced by the estimator also contribute to the discrepancy in the variance of the estimated position errors between the theoretical bound and the experimental result. Therefore, a low pass filter needs to be included in the hardware design so that the errors introduced by the estimator can be attenuated before the correcting signal is sent back to the beam deflection system. On the other hand, if 1 nm precision is to be achieved using the same grid, a compromise between



(a)

(b)

Figure 5.7: Field distortion map constructed with the average position errors calculated in each 10 μm × 10 μm area. (a) Before 1st order correction applied. (b) After 1st order correction applied.

the precision and the system's bandwidth has to be made as 1 nm precision estimation requires to be performed over more samples.

5.3 Loop Filter

The digital filter is expected to effectively minimize the error introduced by the estimation, while passing through the position error caused by the physical disturbance, which will be sent back to the beam deflection system to be corrected. Therefore, the physical disturbance error is compensated by the feedback loop and the estimation error is eliminated by the filter. For the updated hardware implementation of the spatial-phase estimation as discussed in Section 5.2, the grid signal updates at the rate of 10 MHz, each position error is calculated with 1000 samples and thereby updates at the rate of 10 KHz. If the hardware design is modified to estimate with more samples to achieve 1 nm precision, the update rate of estimates will reduce to 1.1 KHz. This calls for a digital filter that compensates the estimation error at the frequencies of 1.1 KHz and higher. While for the physical disturbance error, a wide attenuation band at low frequencies, at least several hundred hertz, is desired to be obtained by the feedback system.

Figure 5.8 shows the part of the SG design with the filters included after the position error estimation, and the rest of the design is the same as shown in Figure 5.4(a), (b) and (c). The filter is designed based on Xilinx "MAC FIR Filter" block, which loads the digital filter's transfer function coefficients, specified by "FDATool" block, into a ROM and applies them to the filter's input. The customized additional input ports for control signals, denoted as "rst", "en_asr" and "en" in Figure 5.8, ensure (1) that the filter is disabled when the position errors are being calculated and also when the estimator is terminated between exposure lines, and (2) that it is reset after finishing a write field. The details of the filter design can be found in Appendix A.4.

Under the same experimental conditions as described in Section 5.2, the beam position error estimation is performed with a 50-tap Hann low pass filter (refer to Appendix A.4.1 for the filter coefficients), and with the distortion correction char-

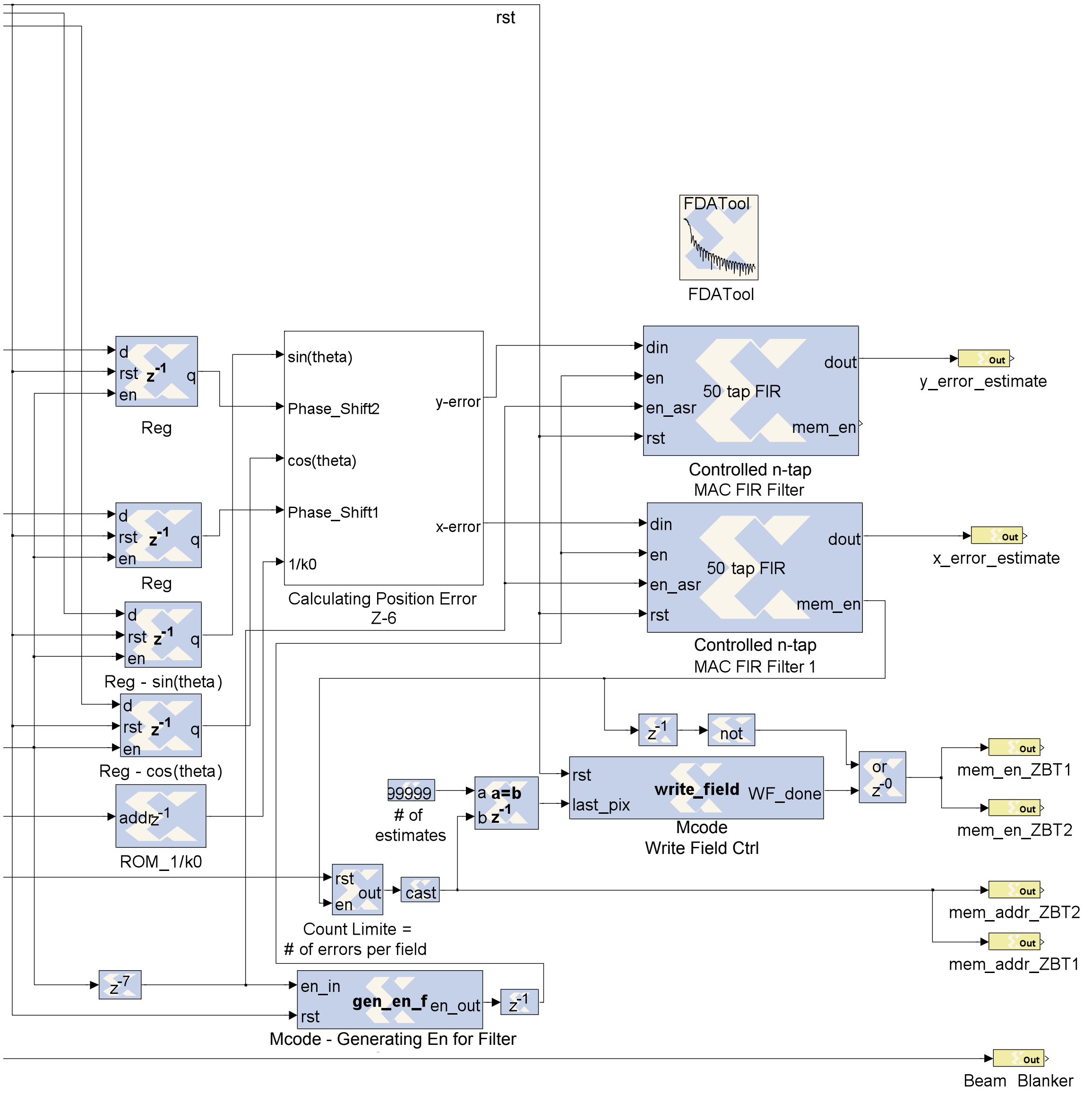


Figure 5.8: Schematics of the System Generator design with filter added before outputting estimated position error. Only the modified part is shown in schematics. Refer to Figure 5.4(a), (b) and (c) for the rest part of the design.

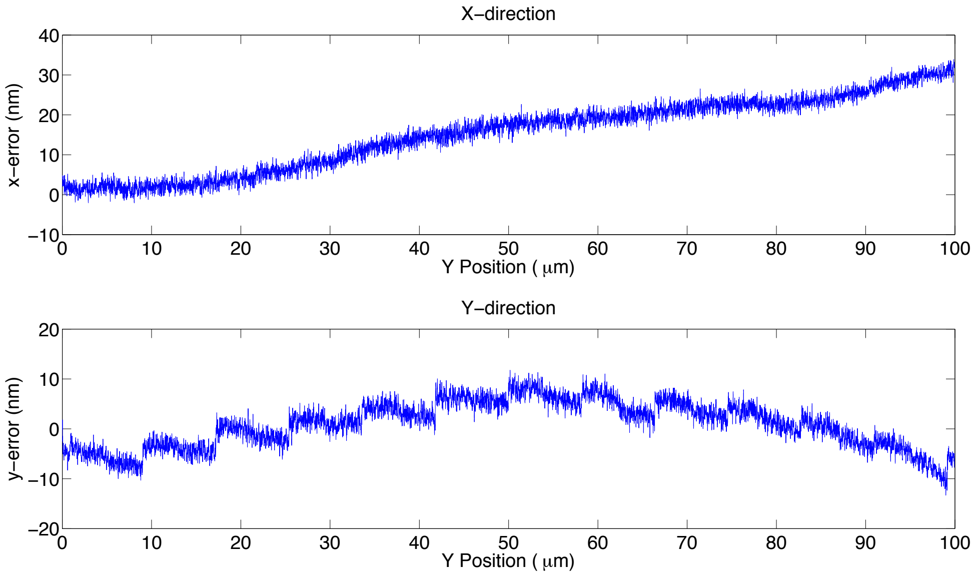


Figure 5.9: Estimated beam position errors vs. absolute position in y-direction with a low-pass filter included in the hardware design and 1st order distortion correction applied.

acterized by the coefficients from Table 5.2 applied. The position errors in x- and y-direction are plotted in Figure 5.9. Compare to Figure 5.6, the local fluctuation due to estimation error is successfully minimized by the filter, and the residual deterministic error attributed to the higher order field distortion is revealed in the much cleaner data. The x-direction error increases gradually as the beam scans horizontally from top of the field to the bottom, indicating a rotation error of the deflection in x-axis. The change of y-direction error resembles a quadratic curve, which requires at least a 2nd-order polynomial fitting. The step-like behavior of y-direction error could be caused by the instability of the least-significant-bit of the DAC that drives the y-axis deflector.

Following the same strategy as described in Section 5.2, the 1st-order distortion correction coefficients are calculated with the filtered data and shown in Table 5.3. With the corrections applied, the rotation error in x-direction is reduced from the range between 0 and 30 nm, as shown in Figure 5.9, to that between -10 nm and 5 nm according to Figure 5.10. To quantify the residual position error for comparison,

Table 5.3: Correction coefficients for 1st-order field distortion calculated from position errors as plotted in Figure 5.9.

x-direction		y-direction	
a_0	$-7.30448E+1$ nm	b_0	$-3.03912E+1$ nm
a_1	$1.00000E+0$	b_1	$-2.30642E-8$
a_2	$3.05880E-4$	b_2	$1.00003E+0$

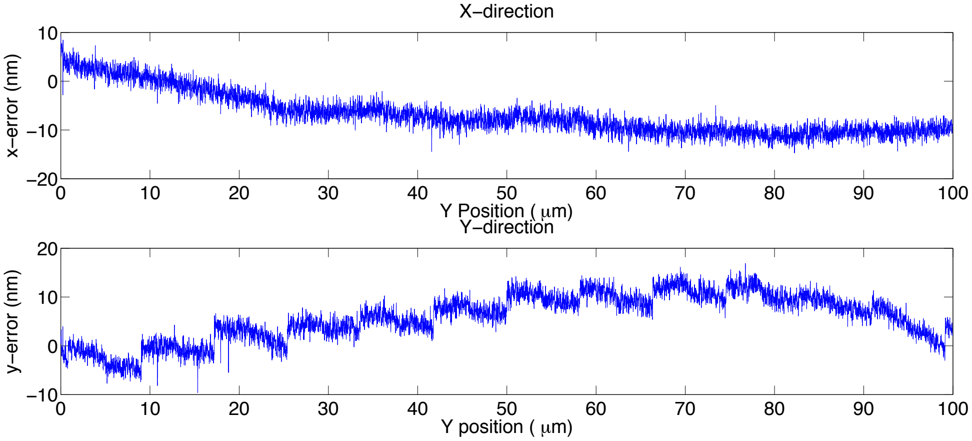


Figure 5.10: Beam position errors vs. absolute position in y-direction.

despite the fact that it's systematic, the standard deviation is calculated to be 4.3 nm and 4.8 nm in x- and y-direction, suggesting that the estimation error has been significantly reduced by the filter.

With the estimator and filter incorporated in the feedback control loop for the beam position, the system is depicted schematically with the input of zero position error in a simplified form [83] as shown in Figure 5.11. The transfer function of the position error estimator H_E with inherent error component D_E is given by [83]

$$H_E(z) = \frac{1}{N}\left(\frac{1+z^{-N}}{1-z^{-1}}\right), \tag{5.5}$$

where N is the number of samples that the estimation is performed on. ΔX is the position error, attributed to the physical disturbance D_P from the imperfection of EBL system, and estimation error D_E from the feedback signal. The transfer function of the digital filter H_F is determined by the filter coefficients a_n specified by

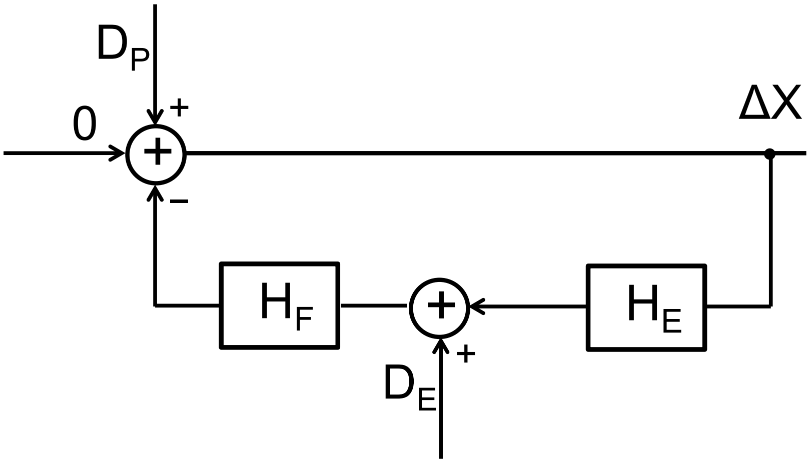


Figure 5.11: Schematics of the feedback control loop.

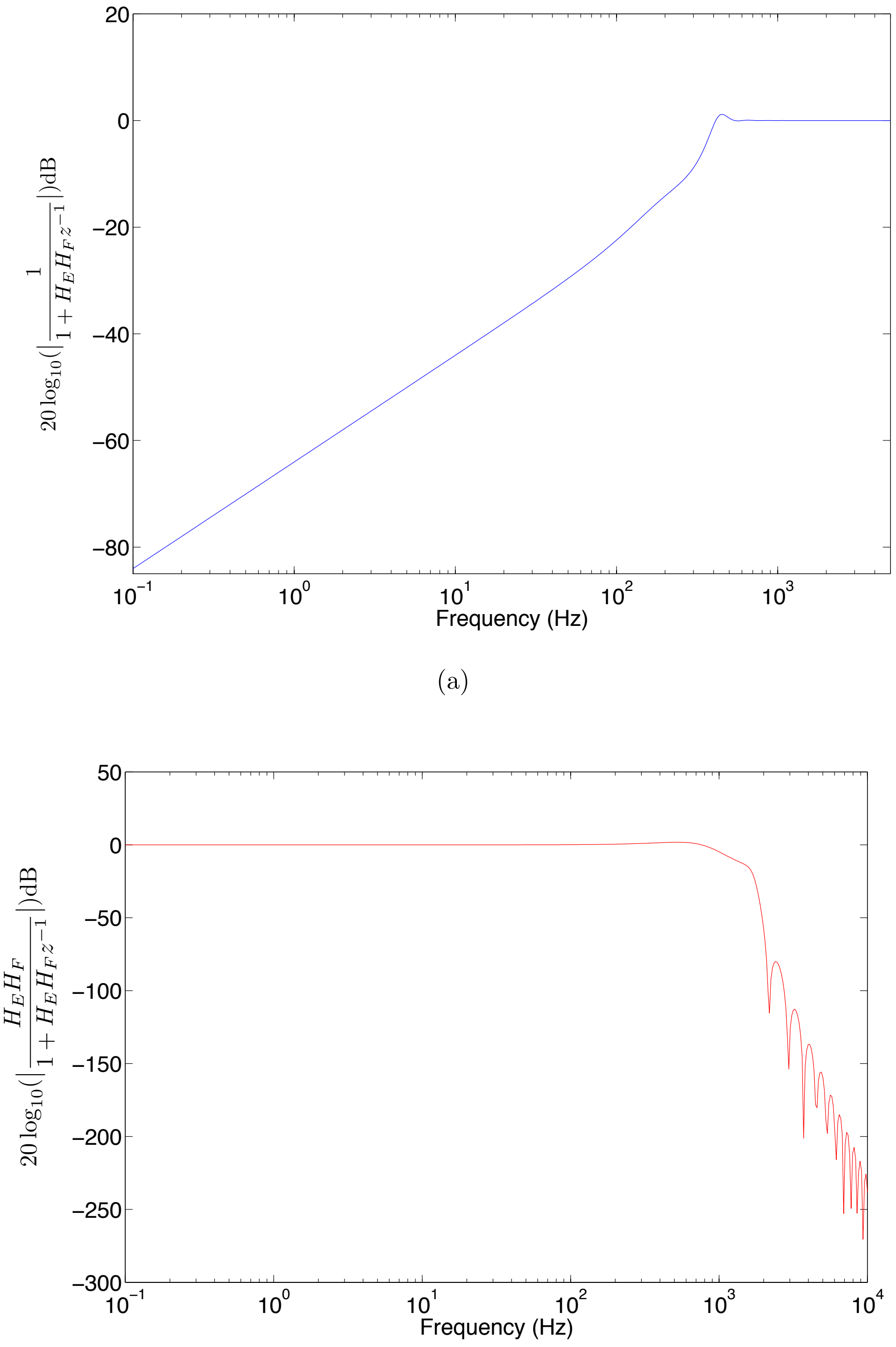
the "FDATool" block as

$$H_F = a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n-1} z^{-(n-1)}, \tag{5.6}$$

where n is the order of the filter. The position error ΔX derived from the schematics is given by

$$\Delta X = D_P \frac{1}{1 + H_E H_F z^{-1}} - D_E \frac{H_E H_F}{1 + H_E H_F z^{-1}}. \tag{5.7}$$

By plugging in the parameters of the FPGA implemented estimator and digital filter, the simulated response of the feedback system to the errors is plotted in Figure 5.12(a). For the errors introduced by the physical disturbance D_P, a -3dB attenuation occurs at 380 Hz; while the estimation error D_E that occurs above 930 Hz is filtered out in the loop. The system responses for both estimation and physical disturbance errors indicate that the digital filter meets the design criteria.



(a)

(b)

Figure 5.12: (a) System response for errors introduced by physical disturbance. 50-tap low pass filter is used to achieve low-frequency cutoff of 380 Hz. (b) System response for errors introduced by estimator with high-frequency cutoff at 930 Hz.

Chapter 6

Real-time Exposure Dose Control for Electron Beam Lithography

Unlike these previous methods to control electron dose, the approach presented here introduces a scintillating layer in the resist stack that emits photons on the arrival of primary electrons. As a result, the actual dose delivered to a given pixel on the sample can be detected. The photons are collected and converted to an electrical signal, which is then routed to the hardware where the control algorithm is executed. The control electronics finally blank the beam once the desired dose is delivered to each pixel. Figure 6.1 illustrates the feedback system that relies on detecting the optical signal from the scintillator on the substrate.

6.1 Dose Control Algorithm

The dose control algorithm discussed here relies on the detection of photon emission from a layer of scintillating material on the incident of a primary electron. When electrons impinge on the scintillator, one should observe a spike due to the emitted photon signal captured by a sensor. The first algorithm is based on counting the current peaks at each pixel of exposure. Once the accumulated value reaches to a threshold, exposure is terminated by switching the beam off to prevent excessive dose deposited in the resist. Under exposure should also be compensated by forcing the

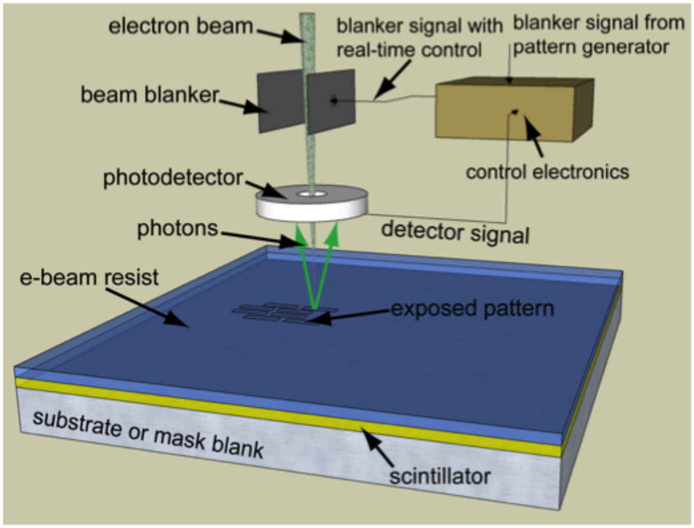


Figure 6.1: Illustration of feedback system for real-time dose control. The substrate to be patterned is coated with a scintillating layer that produces optical signal. The signal is detected and processed to determine whether each pixel has received sufficient dose so that the control electronics can stop the exposure.

beam to stay on until the current peaks count to the desired value.

Another more sophisticated algorithm not only monitors the incidence of signal spike, but also measures the peak value. If the photon yield is known at the specific primary electron energy, the number of electrons contributing to the peak can be determined accordingly. With the information of the actual number of electrons delivered to the resist, the shot noise can be addressed by the electron counting strategy. Maloney investigated both algorithms and the effect of the some parameters on the performance in his thesis [84].

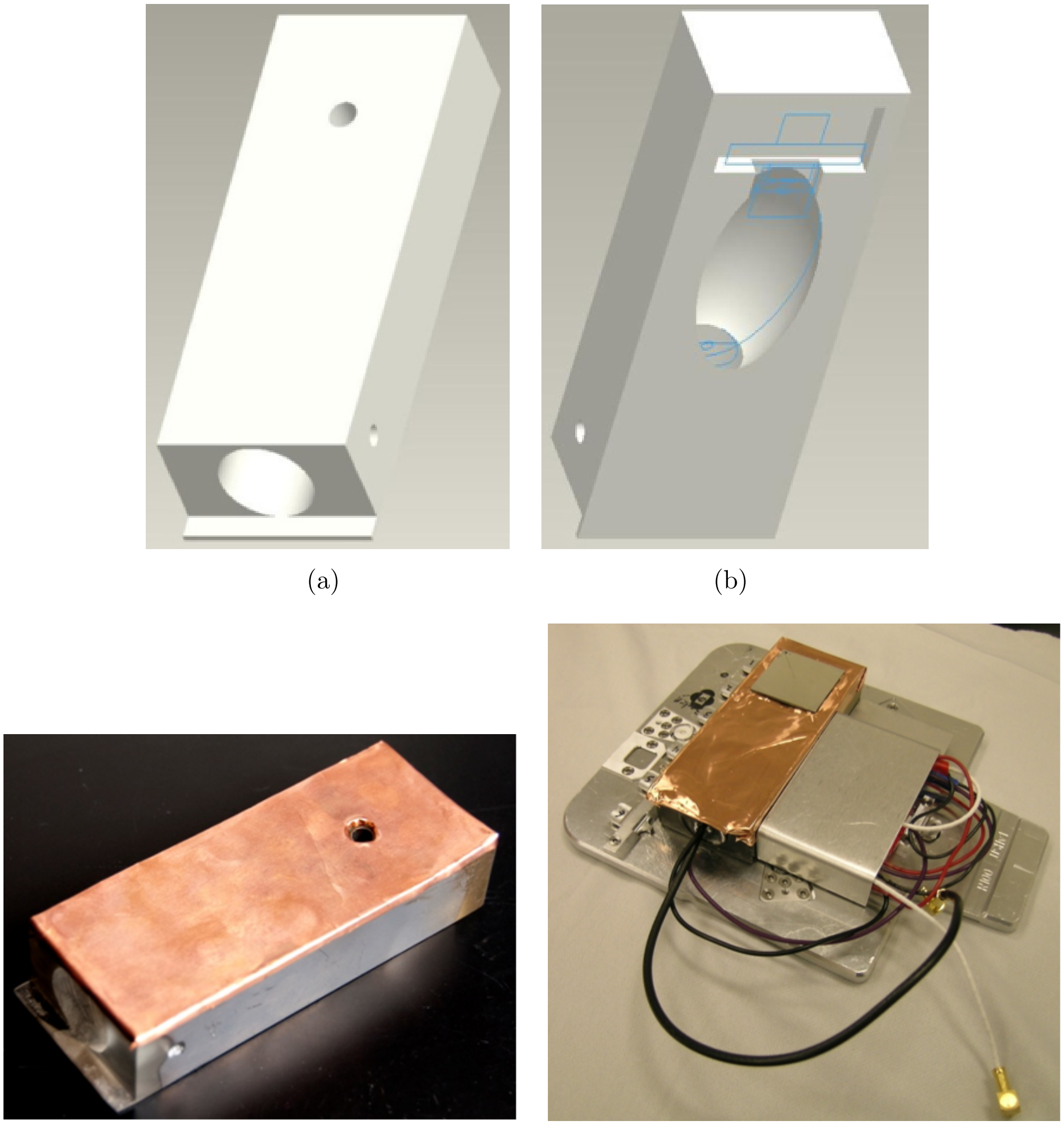
The major challenge for fully implementation of either algorithm is that the electron arrival rate for the lithography tools easily exceed 10^{10} electrons per second. This requires extremely fast scintillator, light detector and counting electronics to capture the event and respond. The experiment presented in the rest of this chapter implements a more conservative control strategy as a proof of concept demonstration. Instead of counting pulses, which will not be obtained at the output of the detector due to the limited bandwidth, the signal is integrated over time until a threshold level is reached. This will not fully compensate for the shot noise effect, but holds the potential to reduce the dose variation related to sample charging and instability of electron optical column.

6.2 Light Detection and Collection

Hamamatsu R7400U photomultiplier tube (PMT) was used as the light detector. Besides the fact that it is one of the smallest PMTs available, it has peak sensitivity through most of the visible spectrum and offers gain up to 3×10^5. It is powered by a Hamamatsu C4900 high voltage power supply, which converts input voltage of 15 V to the range of 0 to -1250 V. An ELPAC external power supply of 15 VDC is connected through a feedthrough to the vacuum chamber as the input voltage when both PMT and power supply are loaded in EBL system. The output current signal from PMT is routed out from the chamber in the same way.

On the arrival of primary electrons, photons are emitted from the scintillator in random directions. It is important to design a light collecting system (LCS) to capture as much light as possible and direct it to the detector. Also, the LCS should contain mechanical support for a PMT to be placed inside. Maloney designed an ellipsoidal mirror cutout inside of a rectangular solid as shown in 6.2 [84]. This assembly, along with the power supply, will fit on the sample holder of the EBL system. The ellipsoid is tilted off the horizontal axis so that the point at which the beam strikes the sample, and where the light is emitted, will be at one focus and the detector at the other. The LCS was constructed with UV-curable photopolymer resin utilizing stereo lithography. To achieve the maximum reflectivity of the ellipsoidal mirror, the surface of cavity is polished, coated with standard PMMA solution, and finally coated with thermally evaporated aluminum.

To characterize the ability of the LCS for capturing light, we measured the output current from the PMT with and without the LCS. A white light source was guided to a diffuse reflectance standard (Labsphere), which replaced the sample and mimicked light emitting from scintillator. In the scenario without the LCS, the PMT was placed at the exact same location with respect to the diffuse reflectance standard as to that



(a)

(b)

(c)

(d)

Figure 6.2: (a,b) Solid model showing opposite sides of the light collection system. (a) shows the opening through which the e-beam passes and the mechanical support for the PMT. (b) shows the elliptical mirror and the recess in which the sample sits. (c) Side (a) of LCS is covered with copper foil, and the sample is placed underneath the small hole where the e-beam focuses. (d) Both the LCS and power supply are placed on the sample holder. The power supply is covered with a sheet of Al. In this case, side (b) of LCS faces upward, is covered with copper foil, and has the sample placed on top.

with the LCS. The light from the environment is blocked when taking measurements. The LCS used in the experiments reported here increased the amount of light captured by 12 times.

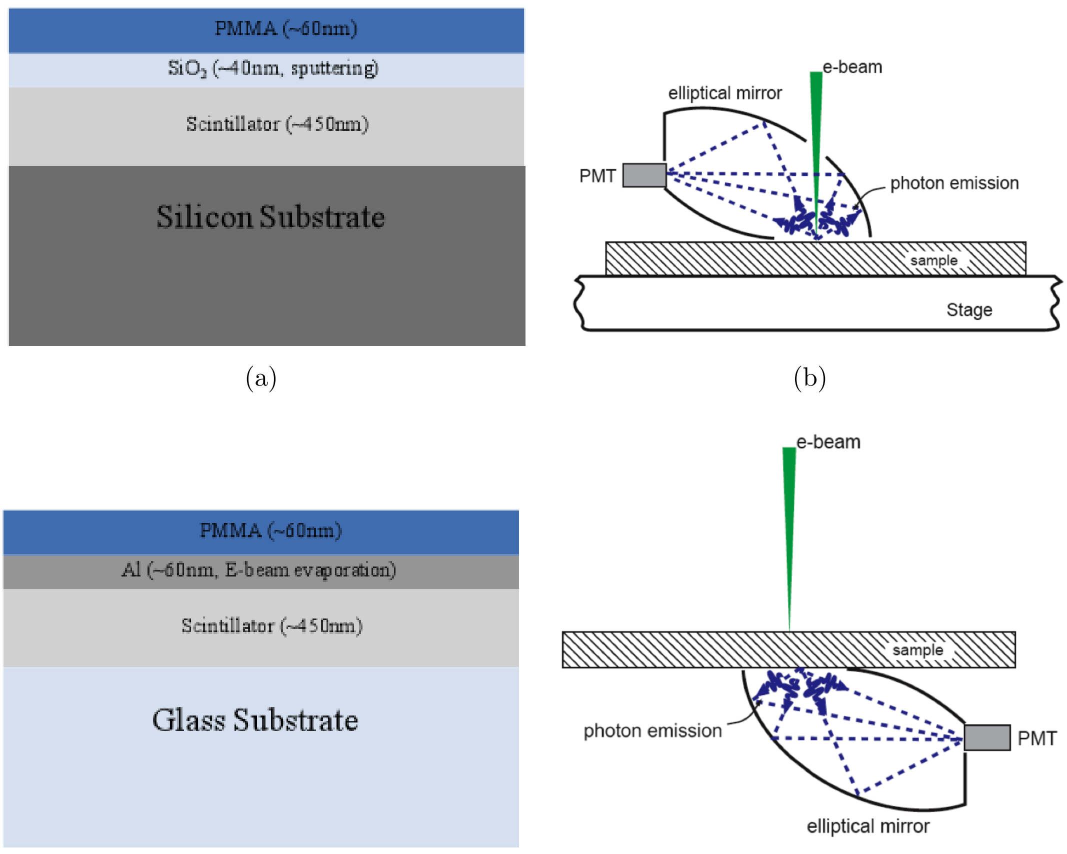
6.3 Scintillator and Sample Preparation

The scintillator is a key element of our dose control system and is composed of a primary scintillator, p-terphenyl, and a secondary scintillator, POPOP, in a polyvinyl toluene (PVT) polymer matrix. The secondary scintillator shifts the final emitted wavelength to 420 nm [85]. The scintillators and polymer are dissolved in chlorobenzene (C_6H_5Cl). The mixture was heated and stirred at 90°C for 24 hours to assure that all components were dissolved in the solution.

We designed two types of samples to accommodate the two setups as depicted in Figure 6.3: one with the elliptical mirror underneath the sample and the other above. For samples placed under the ellipsoidal mirror, a silicon substrate was spin-coated with scintillator mixture at 2000 rpm, resulting in a film thickness of approximately 450 nm. Then a layer of 40 nm thick SiO_2 was sputtered on the scintillator, followed by spin-coating a 60 nm thick PMMA layer on the top. For samples placed above the mirror, a glass substrate was used to let light pass through, and SiO_2 was replaced by 60 nm thick e-beam evaporated Al to reflect the light back to the mirror and prevent charging.

6.4 Implementation of Dose Control Algorithm on FPGA

The same FPGA board was used to implement the dose control algorithm. The DSP design in System Generator is shown as in Figure 6.4. The core component of the design is a finite state machine defined in MATLAB which uses persistent state variables to store the state. The beam blanking signal from Raith lithography system (denoted as "BB_RAITH" in Figure 6.4) is delayed by one clock cycle to be compared



(a) (b)

(c) (d)

Figure 6.3: (a,b) Illustration of scintillator coated Si substrate for which light is primarily emitted upwards. The LCS is placed on top of this type of sample. (c,d) Illustration of scintillator coated glass substrate. Light emitted from scintillator is reflected downwards by Al film. The LCS collects light from below with this type of sample.

with its own current value. Therefore the signal's transition behavior, both rise and fall, is monitored and used together with the current state of the finite state machine to define the new blanking signal (denoted as "BB_FPGA" in Figure 6.4) to be sent back to Raith. The falling edge of signal "BB_RAITH", which indicates the beam is switched on, triggers the accumulator to integrate the signal from ADC. If the accumulated value reaches to the desired level, accumulator is disabled and reset, and logic '1' is issued to BB_FPGA to terminate the exposure. On the other hand, if BB_RAITH's rising transition occurs before the integration achieves the threshold value, the finite state machine maintains the state of exposing until desired dose has been delivered to the pixel.

The threshold value for a proper exposure varies from exposure to exposure, due to the change of beam energy, beam current, and resist thickness. The scintillator's efficiency at the specific exposure condition also plays an important role. Therefore, a register is instantiated in the top level HDL entity to be accessed via interface core from host computer, in which the threshold value determined by the exposure conditions will be stored and then loaded to the SG design through input "threshold". Details on the signal routing can be found in Figure 6.5

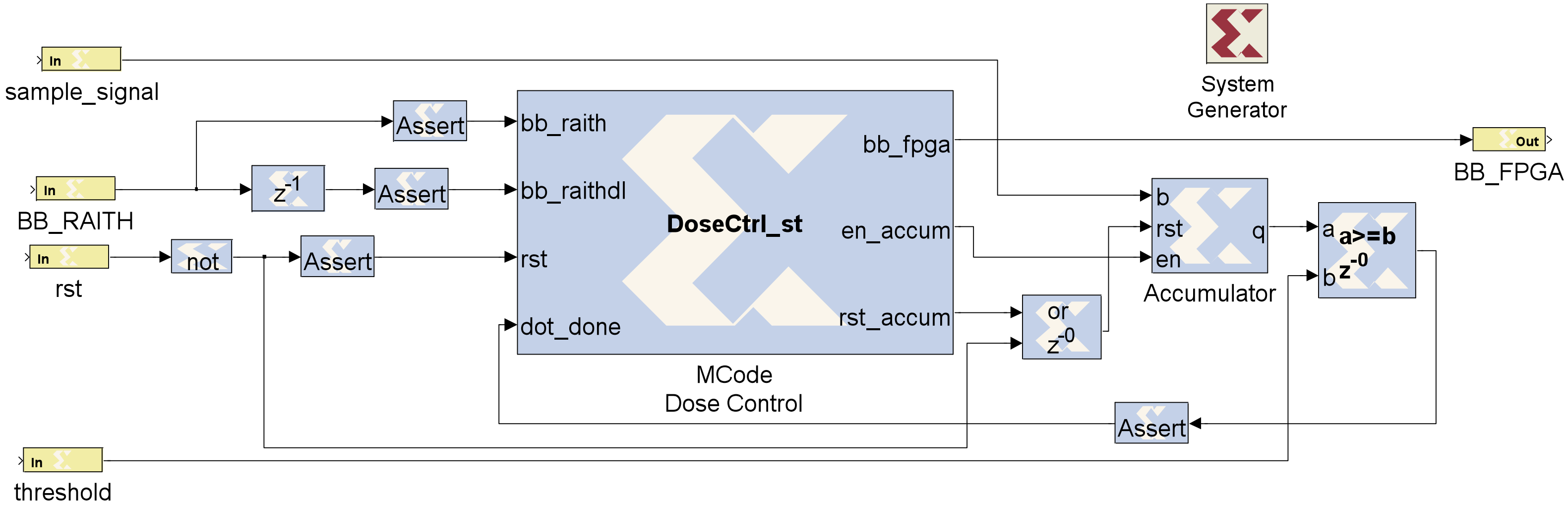


Figure 6.4: Block diagram of System Generator design for dose control. The optical signal is integrated and compared to a threshold value to determine whether desired dose level has reached. Refer to Appendix A.5 for MATLAB code of the finite state machine "DoseCtrl_st".

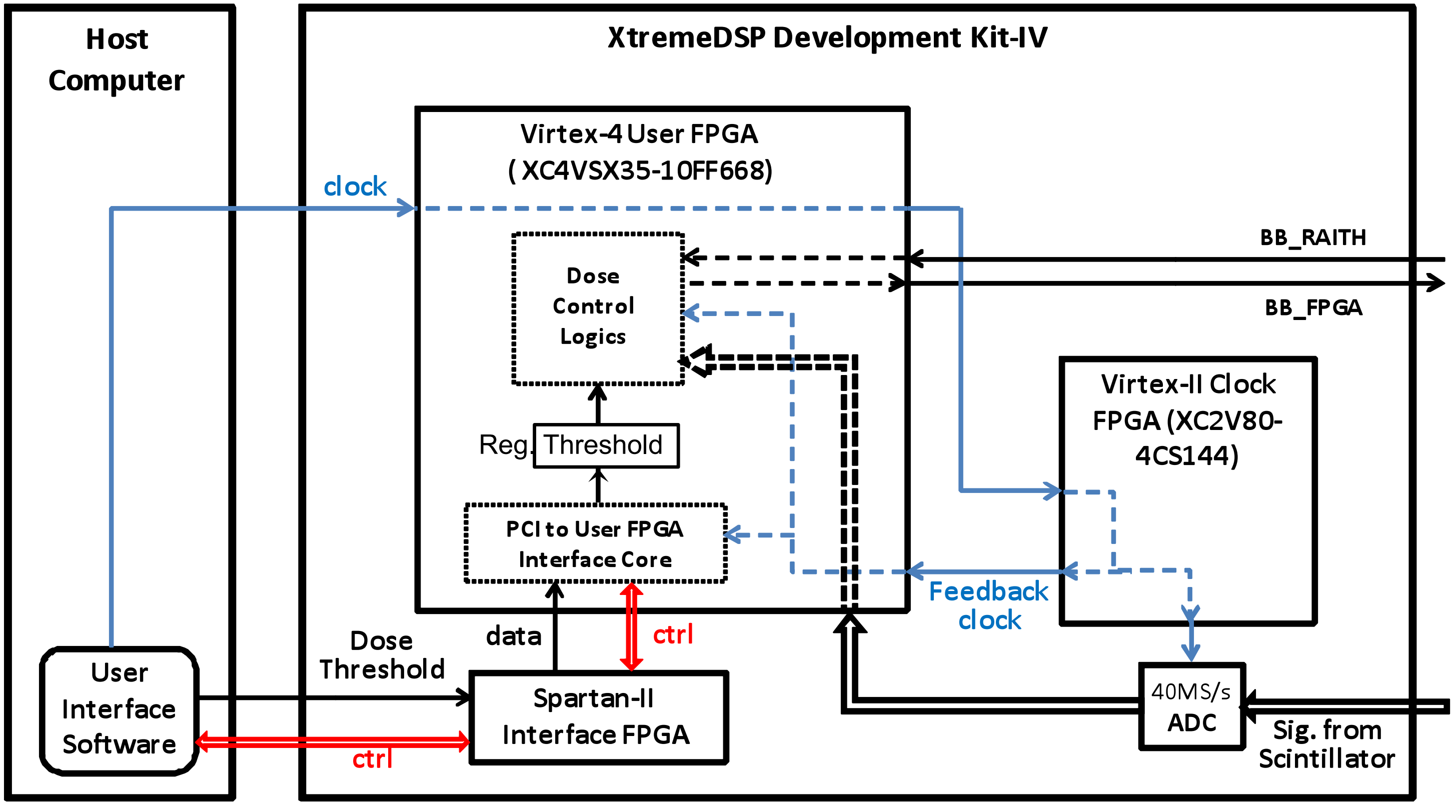


Figure 6.5: Schematic of signal routing among modules on the FPGA development board and the host computer for dose control.

6.5 Experiment Verification

6.5.1 Hardware Functionality Verification

The University of Kentucky's Raith e_LiNE SEBL system was used to test the functionality of FPGA implementation with the sample placed below the mirror. The side of LCS that faces upwards was covered with copper foil as shown in Figure 6.2(c), and the high voltage power supply with Al, to prevent electron charging. The e-beam passes through a hole in the LCS, which is lined with a small aluminum tube, and travels through the cavity to arrive at the surface of the sample. The primary beam energy was 10 KeV and the beam current was 210 pA. Due to the fact that the LCS was placed between the final lens and the sample, we had to use a large working distance of 24 mm. Although this is not optimal for writing patterns, it serves to verify the FPGA implementation.

As shown in the schematic of the experimental setup (Figure 6.6), the output current from PMT is routed to a trans-impedance amplifier before reaching the ADC on the board. The trans-impedance amplifier converts the negative current signal from PMT to 0-1 V voltage, which is the required input range of ADC. The beam blank signal from Raith Pattern Processor was disconnected from the Beam Blank Controller, and was routed to an I/O port on the board to control the dose control logics implemented on Virtex-4 as well as an oscilloscope. A new beam blank signal from FPGA, which switches off the beam before or after the Pattern Processor does depending on the actual accumulated signal, was routed to the input of Raith Beam Blank Controller. Since the scintillator only emits photons when the e-beam strikes the sample, we used the signal from PMT, which was also routed to the oscilloscope, to monitor the behavior of the new beam blank signal.

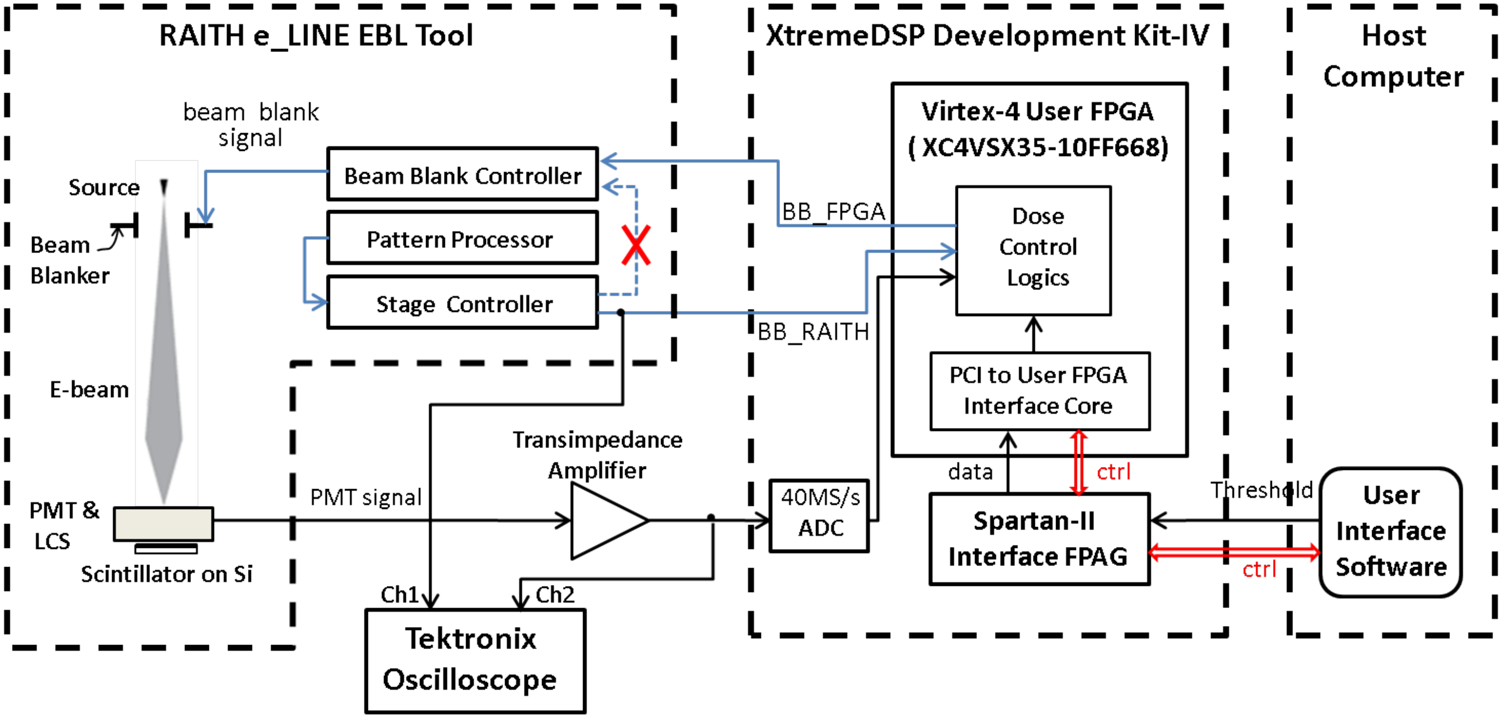


Figure 6.6: Schematic of the experimental setup for verifying the hardware design of dose control. The signal from PMT is routed through a trans-impedance amplifier and ADC, and finally processed by a Virtex-4 FPGA. The beam-blanking control signal from the EBL system (original connection shown with a red "X") now provides the control signal for the logic, and a new signal from the FPGA controls the beam blanker.

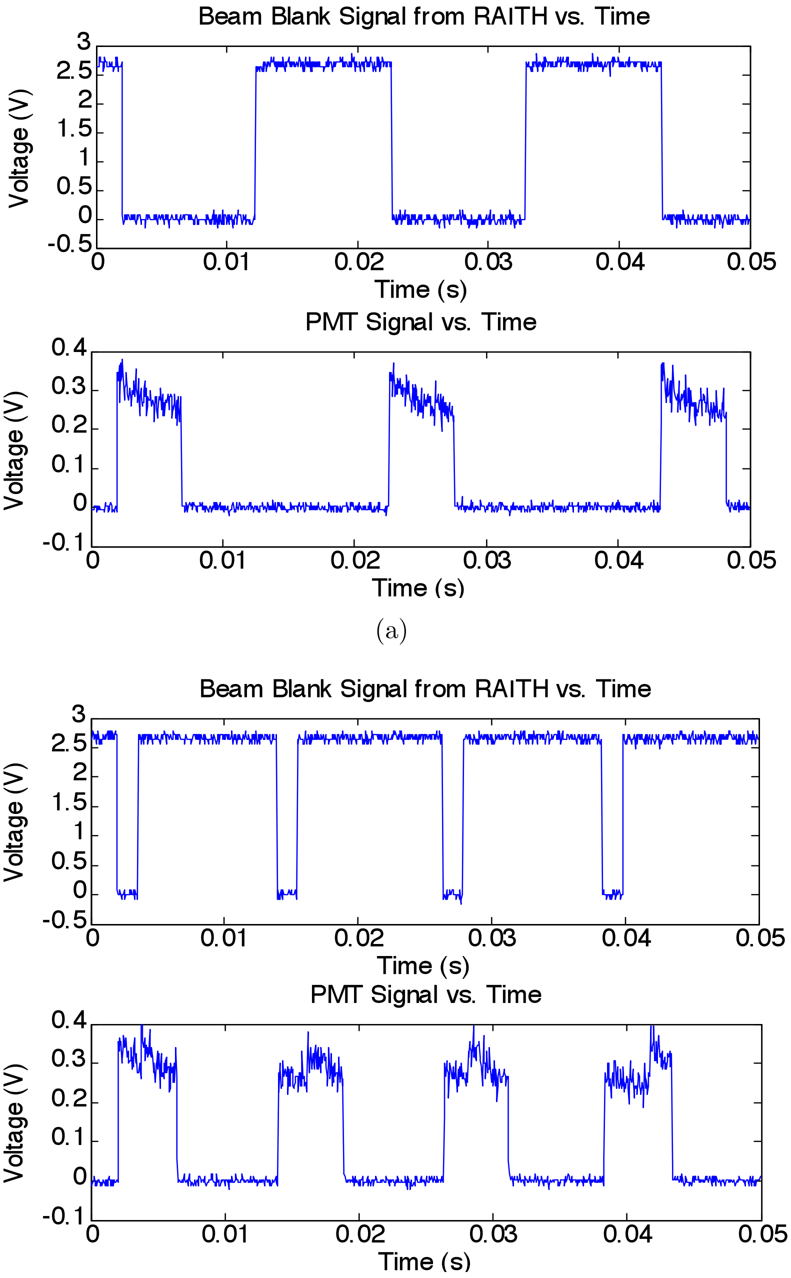
We chose the reference dose to be 629 fC and programmed the EBL tool to write series of single pixel dots with higher (2095 fC) or lower (314 fC) dose. Signal "BB_RAITH" captured on an oscilloscope is shown as upper traces in Figure 6.9, and the e-beam would be on in the absence of feedback control when this signal is 0 V. The voltage pulses in the lower traces (PMT signal) indicate the duration when the e-beam was actually turned on. It is obvious that the hardware is capable of adjusting the dwell time by either terminating the exposure early or extending the dwell time when required.

The system works most effectively when the programmed dwell time is longer than the expected dwell time with feedback control. In this case, one need not be concerned about premature beam-deflection by the system when a pixel is still being exposed. Clearly, a more sophisticated system could control both beam deflection and blanking based on the scintillator signal. In these experiments, the neighboring dots had to be separated by 900 nm to prevent the organic scintillator from being damaged by nearby exposures. More robust scintillators, both organic and inorganic, are being explored along with faster resists to mitigate this problem.

6.5.2 Exposure Experiment

For the exposure experiment, we placed the sample above the mirror. The side of LCS facing the final lens, as shown in Figure 6.2(b), was covered with copper foil, with a square cutout to allow sample to sit stably on the LCS. Figure 6.2(d) shows both the shielded LCS and the power supply placed on the EBL sample holder. This setup enables a much shorter working distance and eliminates the difficulty of optimizing beam focus encountered in the previous experiment. The same accelerating voltage and aperture were used, and the working distance was 5 mm.

A test pattern with built-in dose variation was created using MATLAB. The pattern consists of a 16-row by 6-column array of fields containing single pixel dots. There are 49 dots in each field in a 7 x 7 array. The distance between two dots, vertically and horizontally, was chosen to be 900 nm to ensure the scintillator degradation does not affect adjacent exposures. The same average base dose is applied to all 6



(a)

(b)

Figure 6.7: Comparison of EBL system beam blanker signal (upper traces) with scintillator signal from the PMT (lower traces). For the upper traces, a voltage of 0 indicates that the beam would be on in the absence of feedback control. (a) The dwell time was set to 10 ms (nominal dose 2095 fC) and the feedback system achieved the required dose (629 fC) by terminating the exposure early. (b) The dwell time was set to 1.5 ms (nominal dose 314 fC) and the feedback system extended the dwell times to achieve the desired dose.

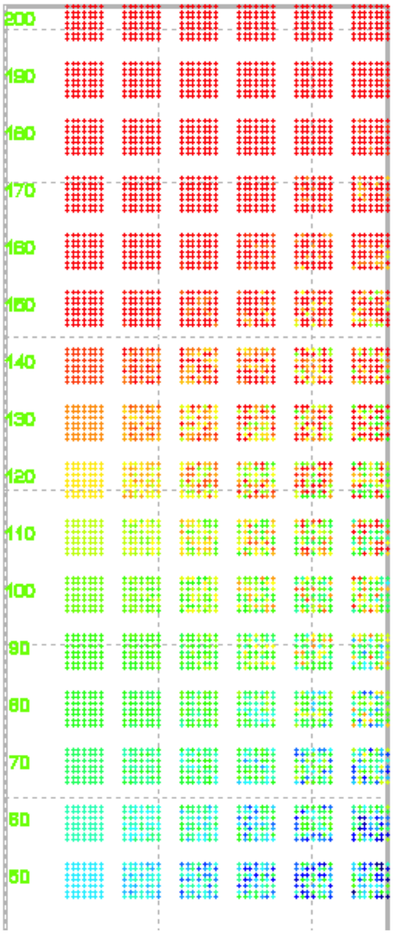


Figure 6.8: Dot array in Raith software for exposure experiment.

fields in the same row, and the base-dose factor increases from bottom to top from 50% to 200%. The same dose variance is intentionally introduced to each column, with the standard deviation ranging from 0% on the left to 29% of the base dose on the right. The test pattern as shown in Figure 6.8 was then imported into Raith software.

Exposure was performed with and without feedback control on the same substrate and under the same experimental environment. For exposure without dose control, a base dose of 0.07 pC/dot was used for 60 nm thick PMMA with a 233 pA beam current. Before exposing the test pattern with feedback, the dose threshold was determined so that patterning with feedback control would yield the same dose level. We turned the beam on over the sample close to the area to be exposed and captured

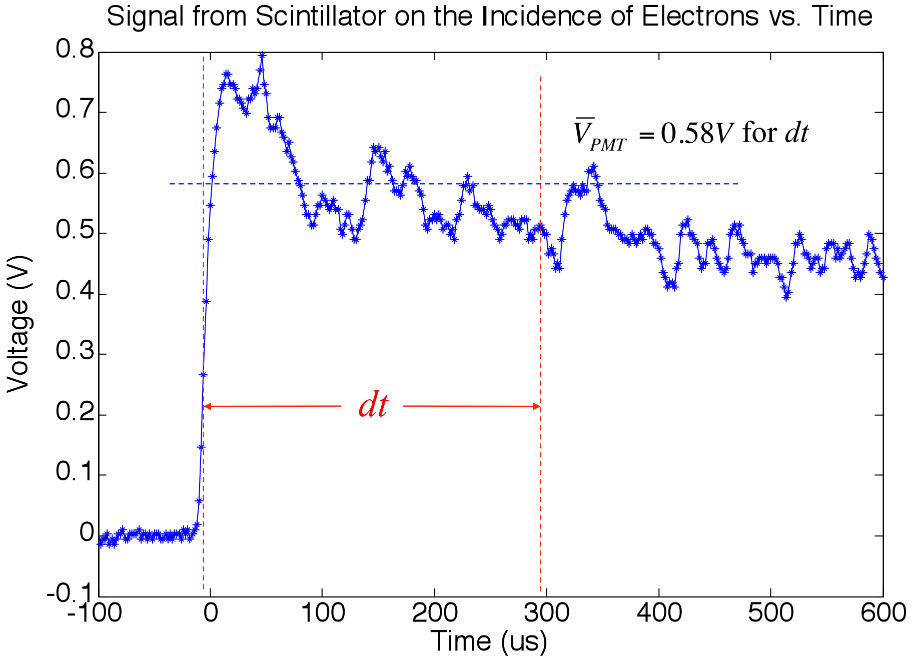
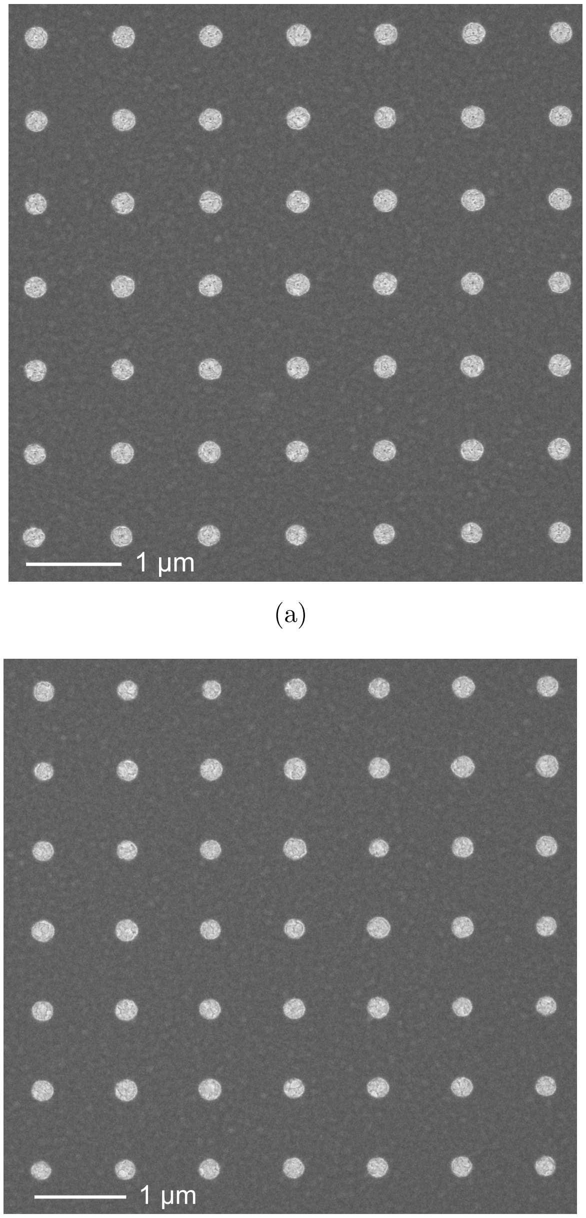


Figure 6.9: Signal from PMT vs. time after turning on the beam. A slight decay in scintillator emission is present. *dt* represents the dwell time, which is used to determine the threshold value before exposure.

the signal from PMT as shown in Figure 6.9. The average voltage value was first estimated between when the voltage starts rising and the expected dwell time (0.3 ms for dose of 0.07 pC in this case) elapses, and then this value was multiplied by the number of samples acquired by ADC, which is the dwell time divided by FPGA clock cycle (0.3 ms/25 ns = 12000). Finally, the threshold value was loaded to the hardware and was compared to the accumulated signal in each clock cycle to determine whether the desired dose had been reached or not. The nominal dose was set to 0.28 pC in Raith software and dot-settling time was set to 1 ms.

The exposed sample was developed in a solution composed of isopropyl alcohol (IPA) and methyl isobutyl ketone (MIBK) (3:1) for 30 seconds, and then loaded back into the EBL system for imaging. Figure 6.10 shows a SEM image of one field of dots that were exposed with and without dose control. We used ImageJ software to analyze the SEM images and to determine the mean and standard deviation of the dot diameters. The average dot diameter exposed with dose control is closest to that exposed with dose of 0.09 pC in the absence of dose control, which indicates that the dose that was actually delivered with dose control is about 0.09 pC rather than intended 0.07 pC. This is because the light signal emitted from scintillator and that collected by PMT is different between the exposed area and the region where the threshold was determined.

As shown in Figure 6.11, the variance in dot size becomes worse as the intentionally introduced dose variation increases when the exposure was performed without feedback. In contrast, the dots exposed with dose control present a constant variance. The feedback control system improves dimensional control for programmed dose variations with standard deviation greater than 14 fC. However, for variations below this level, the feedback system actually makes the dimensional variation worse. This indicates that the dose variation is dominated by scintillator signal noise in these cases. One could easily reduce this problem by introducing a loop filter that accounts for the dose over several dots. However, we believe there is extensive room for improvement in the scintillator, light collection system, and electronics that will allow pixel-by-pixel control.



(a)

(b)

Figure 6.10: SEM images of dots exposed with average dose of 0.09 pC. (a) Dose control was incorporated during exposure to achieve the desired dose level. The dwell time was set to 600 μs (nominal dose 0.28 pC, standard deviation 0.024 pC) in the Raith lithography software to avoid the effect of e-beam motion on the pattern shape. (b) Feedback control was not used during exposure. The dose standard deviation is 0.023 pC.

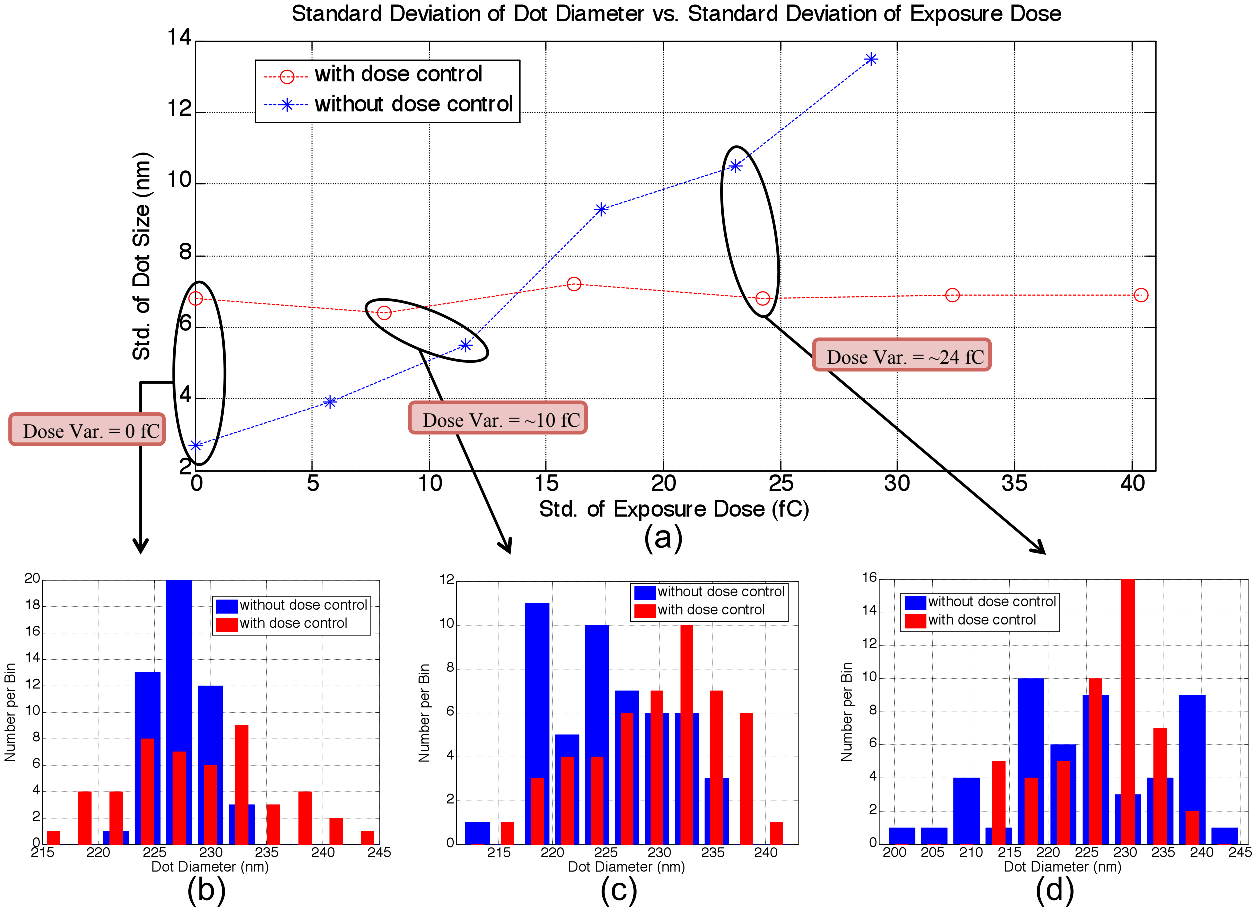
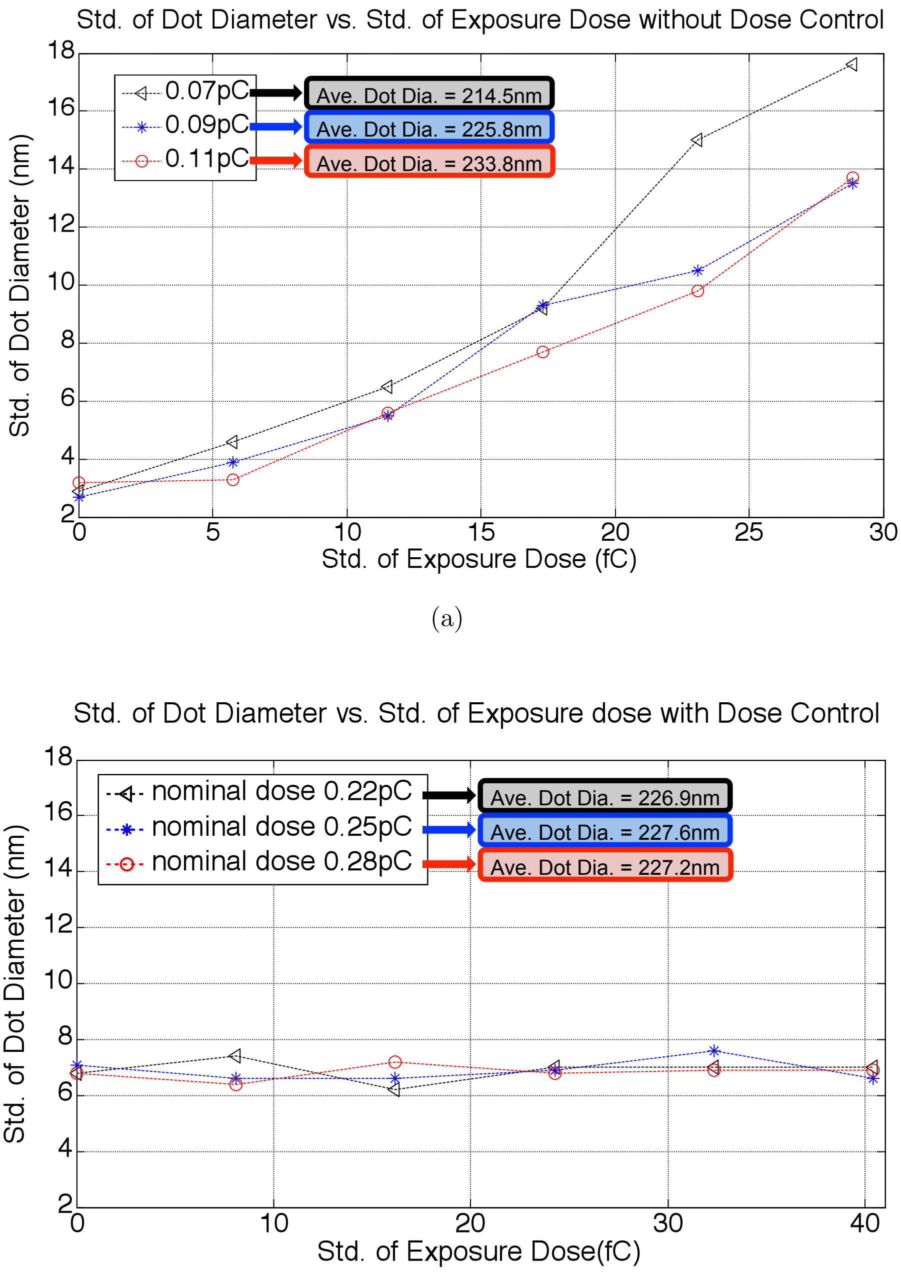


Figure 6.11: (a) Standard deviation (49 trials per data point) of dot diameter versus standard deviation of dose. Exposure was performed with average dose of 0.09 pC. For programmed standard deviations greater than 14 fC feedback control reduced the variance in dimension. (b-d) Distribution of dot sizes for three example dose variations when patterned with (red bars) and without (blue bars) feedback for dose control.

The same trend in standard deviation of dot size when dose control is not incorporated is observed when exposure was conducted with different average doses, and this is shown in Figure 6.12(a). However, the change in nominal dose and dose variance affects neither the average nor the variance of dot size when exposure was performed with dose control, as shown in Figure 6.12(b).



(a)

(b)

Figure 6.12: (a) Standard deviation of dot diameter versus standard deviation of dose in the absence of dose control. (b) Standard deviation of dot diameter versus standard deviation of dose with feedback control.

Chapter 7

Conclusions

Pattern placement, critical dimension control and throughput have been the major challenges when manufacturing sub-100-nm resolution pattern with nanometer-precision using electron beam lithography. This document has reported investigations on solutions to address these issues in two aspects: continuous estimation of electron beam position with respect to a reference pattern that can be used to correct beam deflection error, and real-time feedback control of exposure dosage using electron emitted photons from the sample.

The spatial phase estimation algorithm has been extended to be applied to the vector-scan electron beam lithography system. The simulation and experimental results have demonstrated that sub-nanometer beam position estimation can be achieved by a vector-scan system under typical exposure conditions using a rotated 2D periodic signal, and spatial-phase locking of beam position can be potentially obtained within a single feature that has dimension larger than the fiducial grid period.

With the current FPGA implementation of the spatial-phase estimation and the filter, the beam position error is computed over 1000 samples at rate of 40 MHz, and the experiment has demonstrated 4.3 nm standard deviation of estimated position error, with a feedback system attenuation bandwidth of 380 Hz as suggested by simulations. Although the hardware design updates estimated position error at the rate of 10 MHz, the maximum pixel rate supported by Raith e_LiNE, it is readily compatible with electron beam lithography systems that supports pixel rate up to

40 MHz. In this case, it is necessary to expand the bandwidth of the feedback system accordingly, which only involves the digital filter redesign in the System Generator model.

The hardware design characterizes the deterministic deflection error with a 3rd-order polynomial, but application of correction coefficients is limited to the 1st-order. Although increasing the bit-width of the input ports for distortion coefficients can address this issue without changing the hardware design, it will result in prohibitive consumption of hardware resources. Alternatively, the design can be modified to use the digital beam positions to characterize the distortion, and the existing input port width would be sufficient to represent the new set of coefficients. If the exposure is conducted with the feedback control of beam position at the bandwidth of 380 Hz along with higher-order field distortion corrected, it appears possible to achieve sub-1 nm pattern placement precision with the FPGA implemented SPLEBL.

In order for a high pixel rate electron beam microcolumn system to exploit this technology, and potentially for high speed mask writers, increasing the computational rate of position error estimation is always desirable. Modification of the hardware design with more efficient building blocks, which implement multiple slower operations using time-multiplexing method, holds promise for speeding up the computation using current devices. However, one can simply update the hardware with more powerful electronics and download the same core design to the FPGA to achieve overall speed enhancement. This is unarguably a more economical solution in contrast to frequent calibration between exposures or complex measures to stabilize the system.

The experimental result of dose control for electron beam lithography has shown that controlling the exposure dosage by monitoring a signal from the interaction of the electron and substrate is plausible. This approach requires no modification to the electron optical system, but rather relies on an optical detector and appropriate control electronics. To substantiate the concept, an algorithm of accumulating optical signal was implemented on FPGA, a fabrication procedure was developed to integrate the scintillator with e-beam resist, and exposure experiments were conducted with the feedback control.

Single-pixel exposures with intentionally varied dose are well controlled and yield feature sizes of constant mean and variance according to the statistical analysis of the exposed patterns. Currently the feedback control system offers improvement only when the intentional dose variation exceeds 14 fC/pixel. Although there still remain many opportunities for improvement using the current accumulation algorithm, such as choosing different accelerating voltage and aperture for high SNR, implementing the algorithm at higher rate, etc., it will still require detection of each optical pulse by the light detector and accurate counting of the pulses by the data acquisition and processing electronics, in order to ultimately achieve dose control beyond the shot noise limit.

With regard to the sample preparation, a more resilient scintillator that offers a fast response to electron excitation is critical for the production of reliable optical signals which are used to determine the number of delivered electrons for exposure. Finally, separating the resist from the scintillator will improve the pattern dimension control by eliminating the variation introduced by interlayer surface roughness. To pursue the ideas, Leontsev has fabricated a sample with the resist suspended over a small window of Si substrate, while Jackson redesigned the elliptical mirror to accommodate for a circular scintillator disc to be placed underneath the substrate. The author has recently been involved in a collaborative effort to integrate the new sample/scintillator setup with the pulse counting strategy.

Appendix A

DSP Designs in System Generator

A.1 Figure 4.2 Low Level Schematics of Subsystem and M-function script for MCode Block

A.1.1 Pixel Address Generator Subsystem

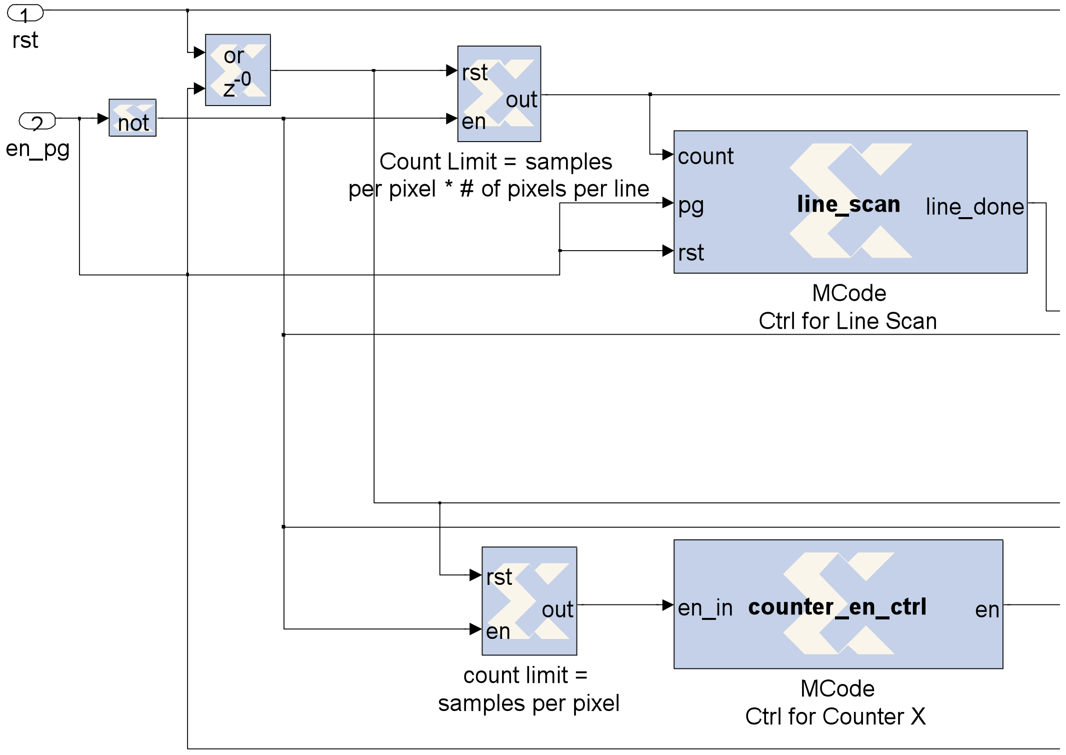
A.1.1.1 M-function for “counter_en_ctrl” block

```
function en=counter_en_ctrl(en_in)

   const1=xfix({xlUnsigned,3,0},0);

   en = en_in > const1;
```

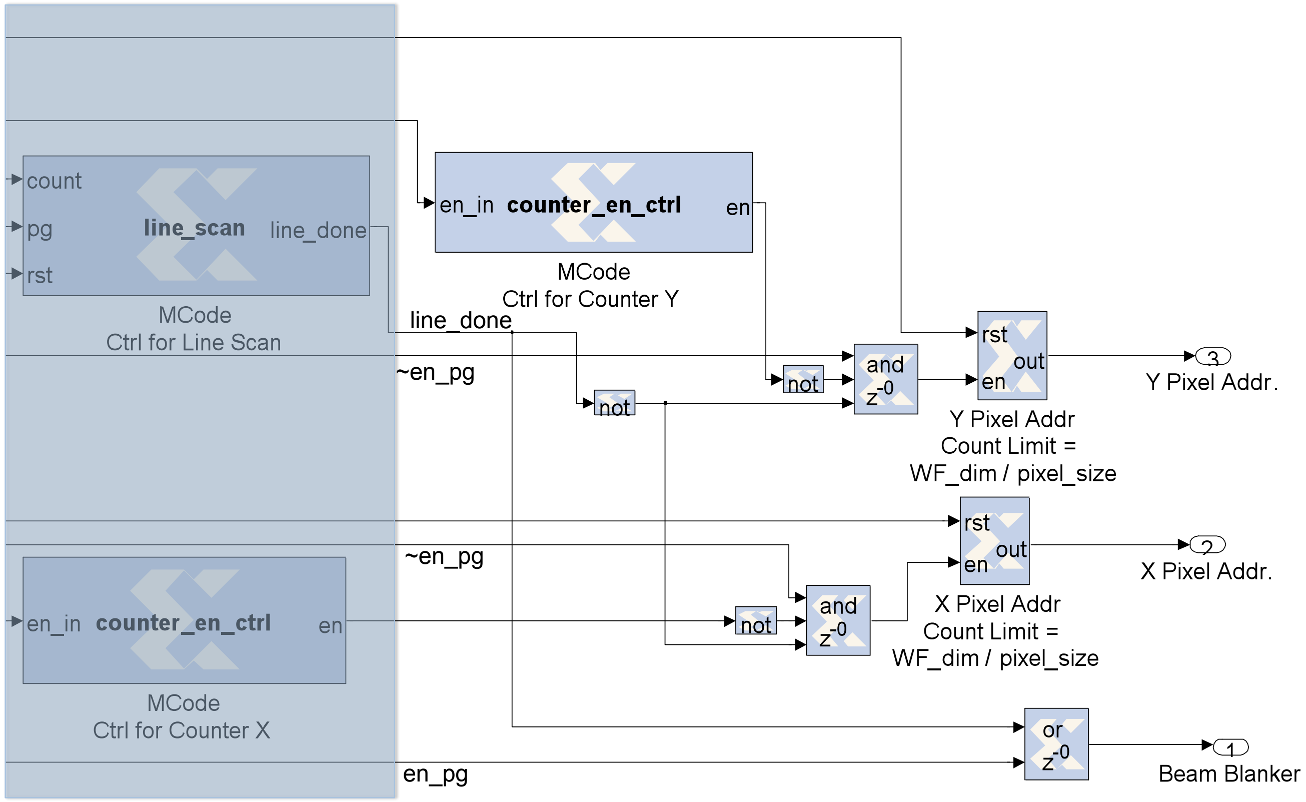
A.1.1.2 M-function for “line_scan” block

```
% Once the last pixel address in the scan line is issued into this
% block, a logic ‘1’ is asserted to the output signal "line_done",
% and the state is maintained until next line scan is initialized.

function line_done = line_scan(count, pg, c_nbits, c_binpt, rst)

    prec = {xlUnsigned, c_nbits, c_binpt, xlRound, xlWrap};
    persistent state, state = xl_state(0,prec);

    line_done = false;

    if rst
        state = 0;
    elseif pg
        line_done = false;
```

```
else
    switch state
        case 0
            if count > 0
                state = 0;
                line_done = false;
            else
                state = 1;
            end
        case 1
            line_done = true;
    end
end
```


1
rst
2
en_pg
not
or
z-0
rst
out
en
Count Limit = samples
per pixel * # of pixels per line
count
pg
rst
line_scan
line_done
MCode
Ctrl for Line Scan
rst
out
en
count limit =
samples per pixel
en_in
counter_en_ctrl
en
MCode
Ctrl for Counter X

(a)



(b)

Figure A.1: Subsystem for generating pixel address.

A.1.2 Subsystem for Calculating Phase of Complex Reference

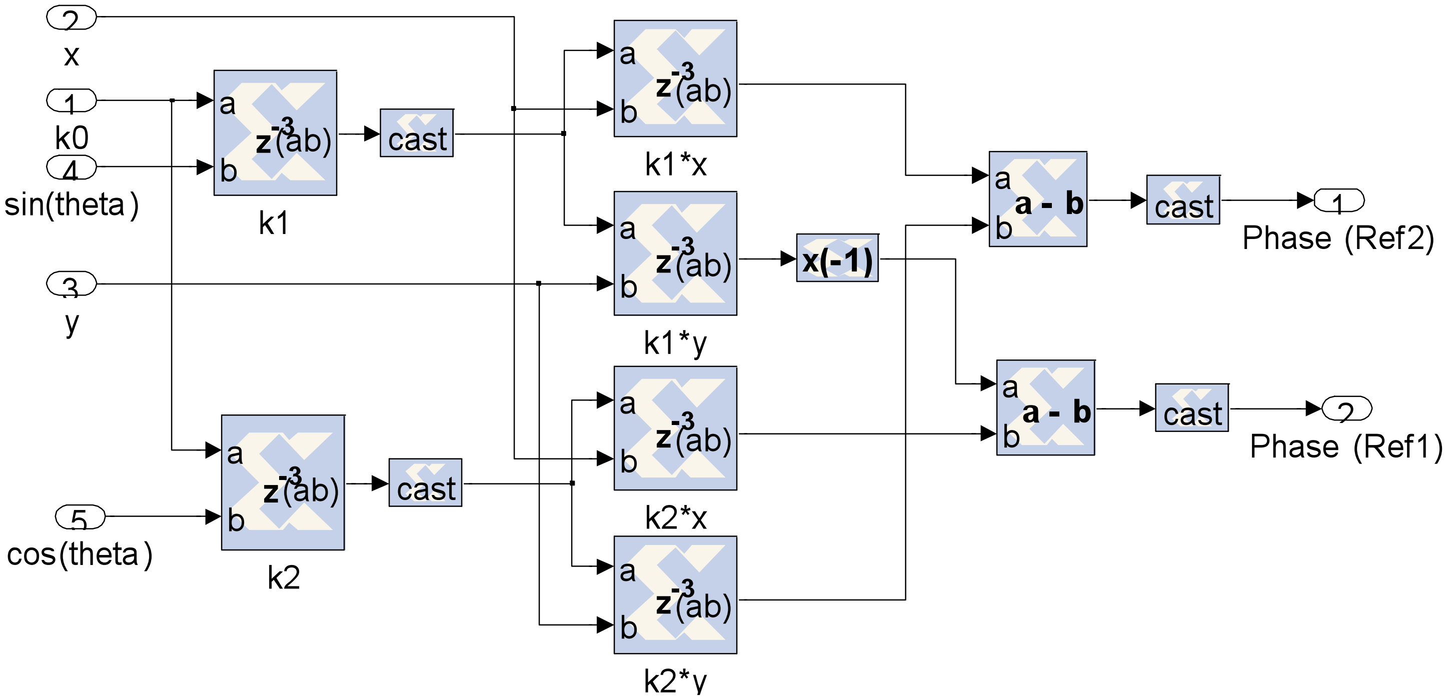
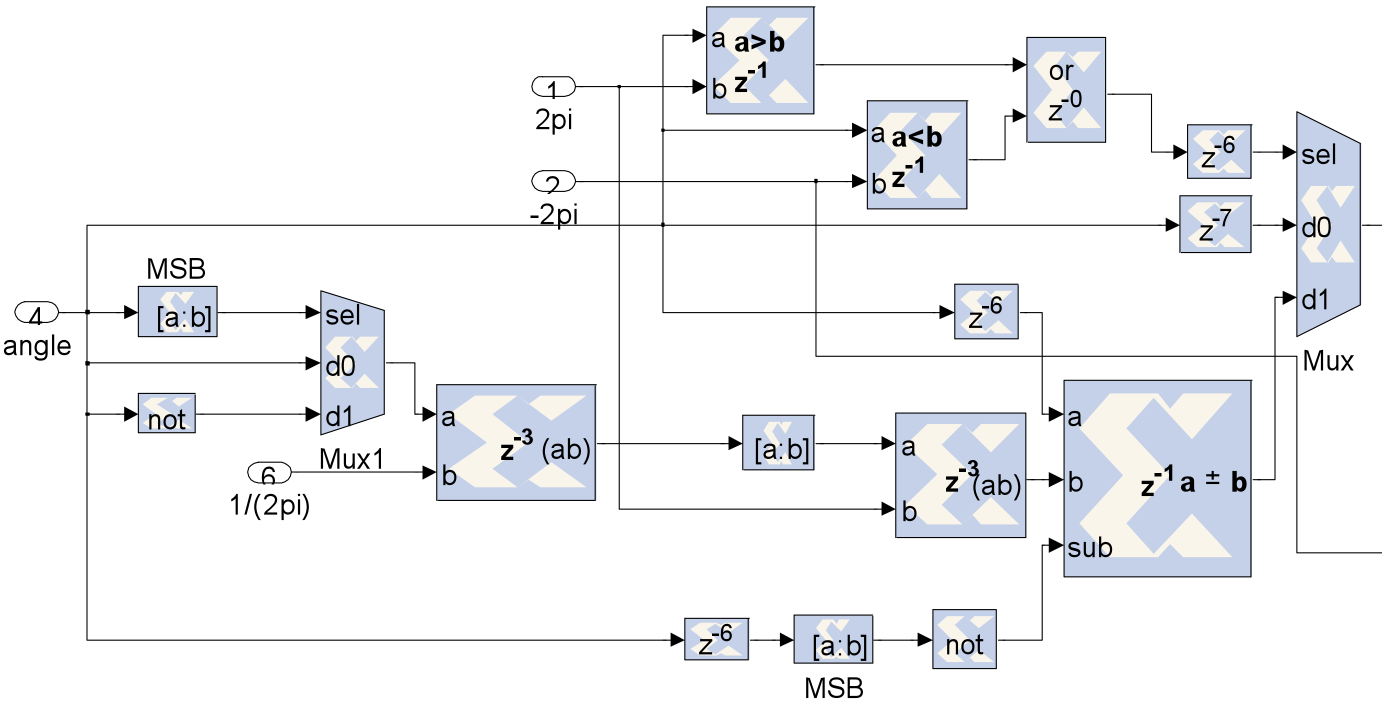
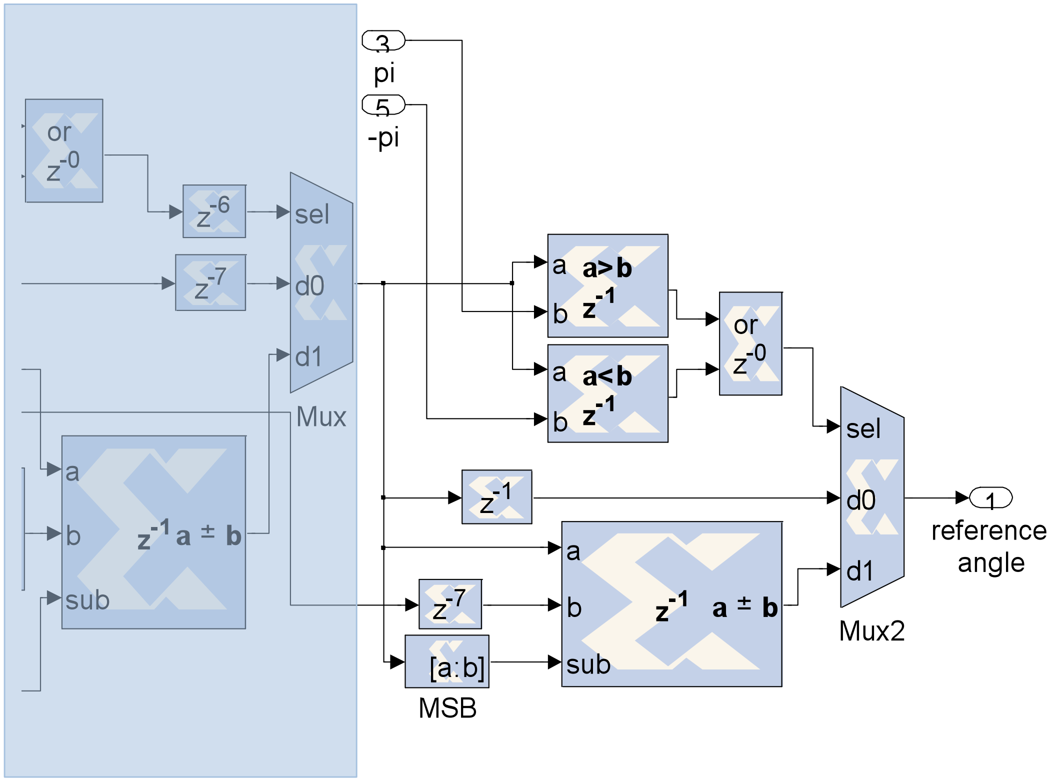


Figure A.2: Subsystem for calculating phase of the complex reference.

A.1.3 Subsystem for Reflecting Phase to Interval $[-\pi, \pi]$


a a>b z-1
b
1
2pi
or z-0
a a<b z-1
b
2
-2pi
z-6
z-7
sel
d0
d1
Mux
MSB
[a:b]
4
angle
sel
d0
d1
Mux1
not
a
z-3 (ab)
b
6
1/(2pi)
z-6
[a:b]
a
z-3 (ab)
b
a
b
z-1 a ± b
sub
z-6
[a:b]
MSB
not

(a)



(b)

Figure A.3: Subsystem for reflecting phase to interval $[-\pi, \pi]$.

A.1.4 Subsystem for Calculating Position Error

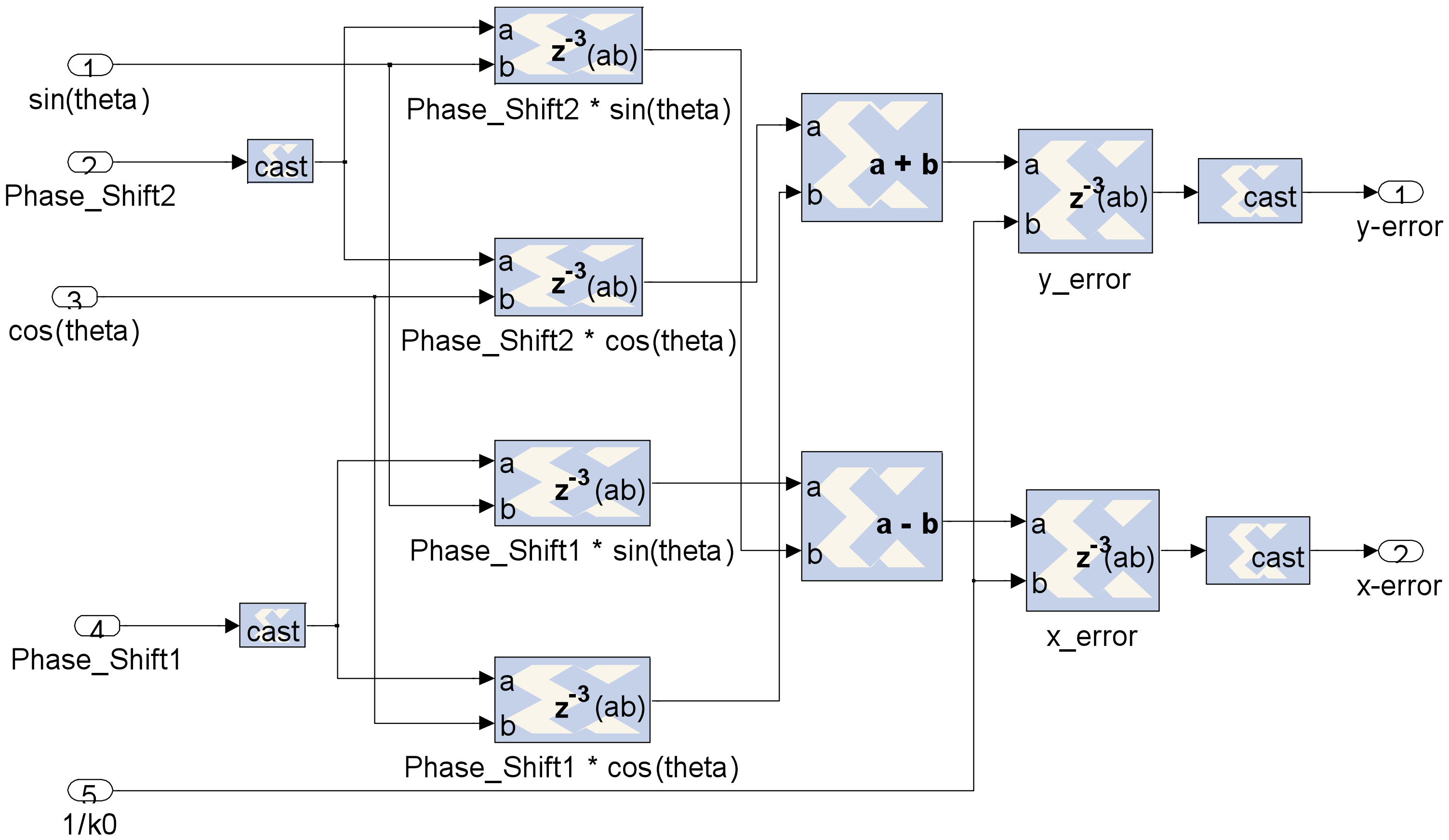


Figure A.4: Subsystem for calculating position error

A.2 Figure 4.14 Low Level Schematics of Subsystem and M-function Script for MCode Block

A.2.1 Subsystem for Calculating Average Signal per Pixel

A.2.1.1 M-function script for "accum_sample" block

```
% The output from the counter, which counts the number of samples
% acquired at each pixel, is compared to 0. The accumulated value is
% available at output when the count equals 0, and a pulse is
% asserted to signal "accum_sample_done", and the register is reset
$ to 0.

function [dout,accum_sample_done]=accum_sample(sample,count,rst,...
en,num_sample_pixel)

    persistent reg, reg = xl_state(0,{xlSigned,24,13});
    num_sample_xfix = xfix({xlUnsigned,10,0},num_sample_pixel);
    const0 = xfix({xlUnsigned,1,0},0);
    count_max = num_sample_xfix - const1;

    if rst
        reg = 0;
        dout = const0;
        accum_sample_done = false;
    elseif en
        if count == count_max
            reg = sample;
            dout = 0;
            accum_sample_done = false;
        elseif count > const0
            reg = reg + sample;
            dout = 0;
            accum_sample_done = false;
        else
            dout = reg + sample;
            reg = 0;
            accum_sample_done = true;
        end
    else
        reg = 0;
        dout = const0;
        accum_sample_done = false;
```

```
end
```

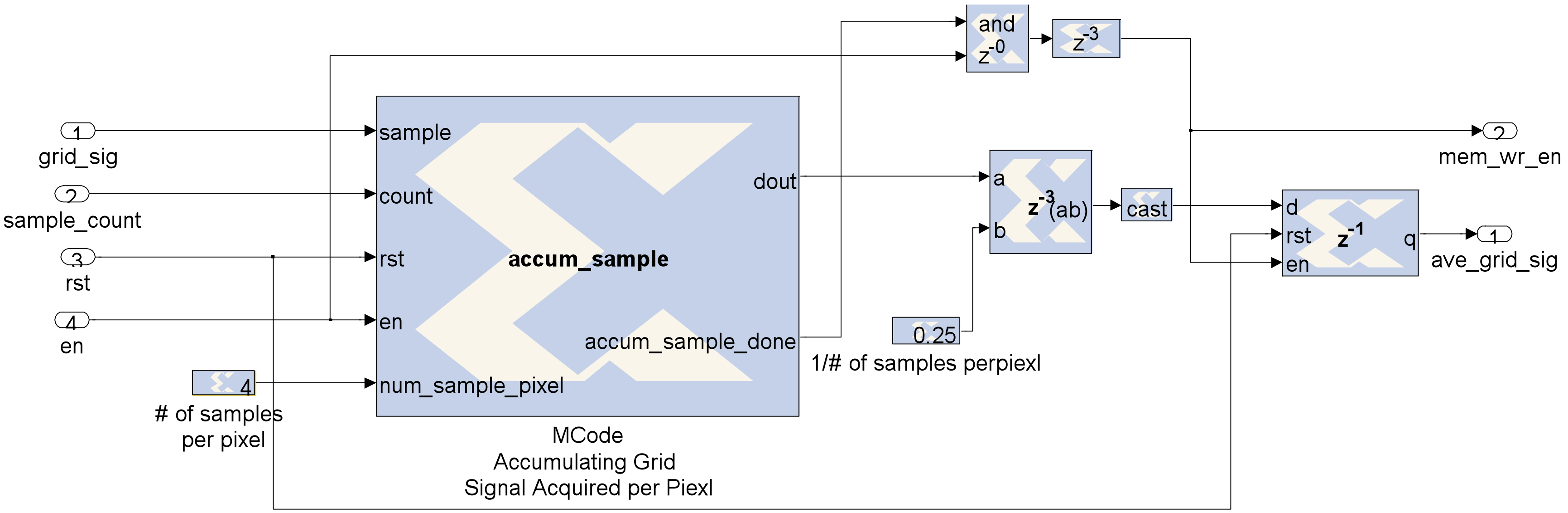
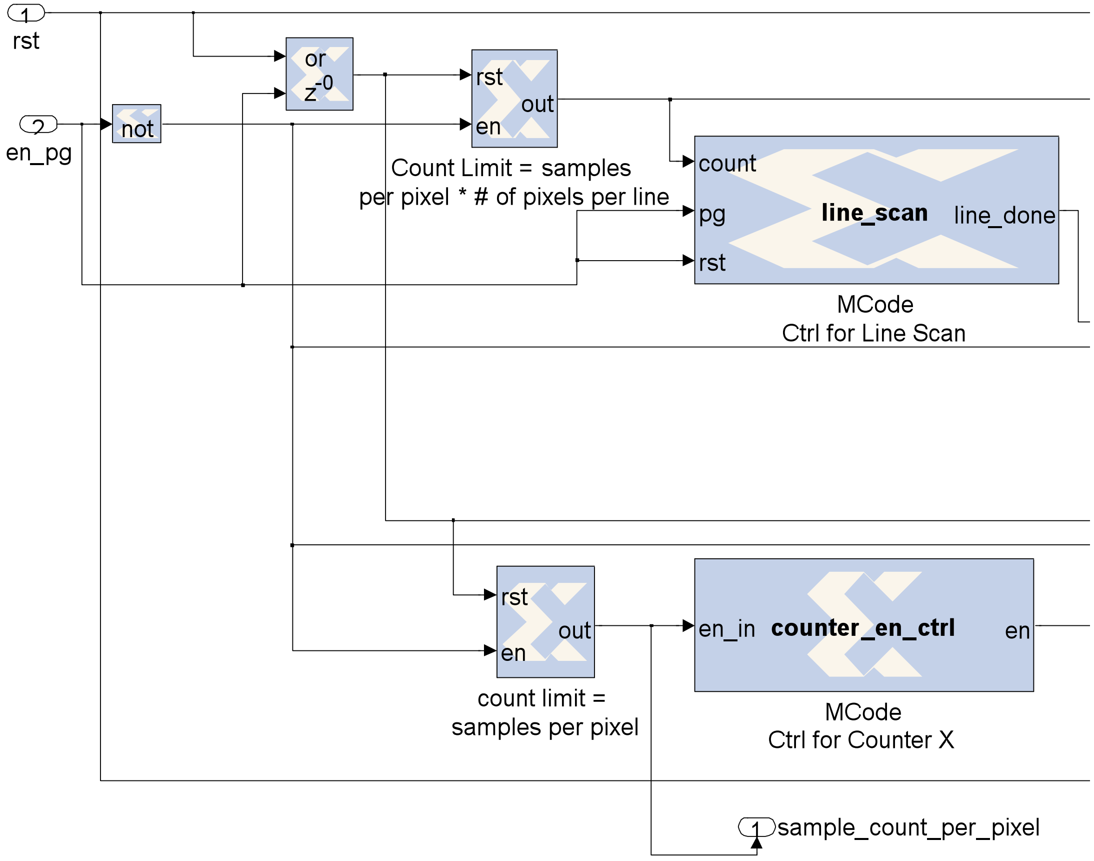


Figure A.5: Subsystem for calculating average signal per pixel

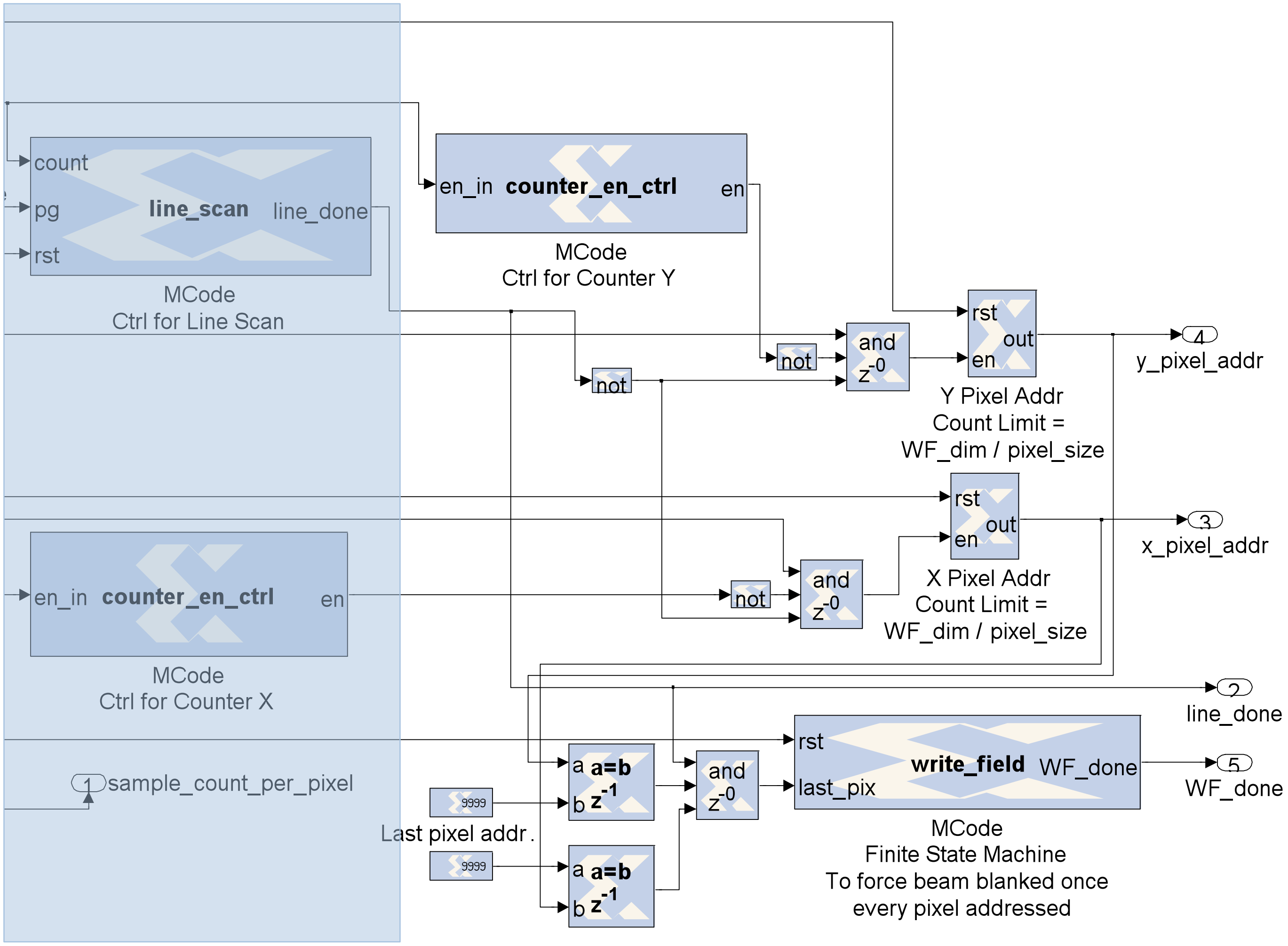
A.2.2 Subsystem for Generating Pixel Address with Write Field Control

A.2.2.1 M-function script for "write_field" block

```
% When a pulse is asserted to "last_pix",  suggesting every pixel
% in the field has been addressed, a logic ‘1’ is issued to switch
% off the beam and the state is maintained upon reset.

function WF_done = write_field(rst,last_pix)

    persistent state, state = xl_state(1,{xlUnsigned,1,0});

     WF_done = false;

    if rst
        state = 1;
        WF_done = false;
    else
        switch state
            case 0
                WF_done = true;
                state = 0;
            case 1
                if last_pix
                    state = 0;
                    WF_done = true;
                else
                    state = 1;
                end
        end
    end
```


1
rst
2
en_pg
not
or
z^{-0}
rst
out
en
Count Limit = samples
per pixel * # of pixels per line
count
pg
rst
line_scan
line_done
MCode
Ctrl for Line Scan
rst
out
en
count limit =
samples per pixel
en_in
counter_en_ctrl
en
MCode
Ctrl for Counter X
1
sample_count_per_pixel

(a)



(b)

Figure A.6: Subsystem for generating pixel address with write field control

A.2.3 Subsystem for Generating Memory Address and Control Signal

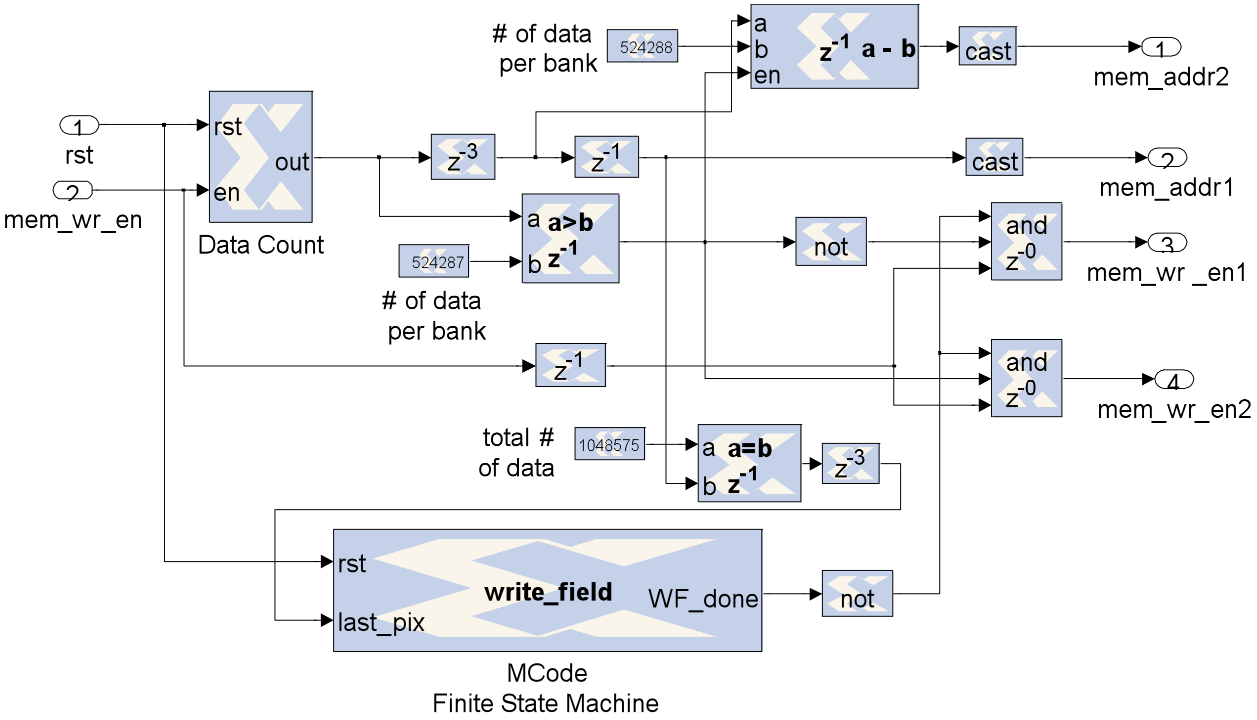


Figure A.7: Subsystem for generating memory address and control signal.

A.3 Figure 5.4 Subsystem for Applying Window Function

The window function block denoted as "1000 coef" in Figure A.8 is a "masked" subsystem, which allows user defined parameters to be used within the scope of the subsystem. In this case, 1000 coefficients are specified in the mask parameter editor and are loaded in the ROM within the subsystem. Figure A.9 shows what is under the "masked" subsystem, consisting a ROM to store the coefficients, a counter to generate addresses for the ROM and multipliers.

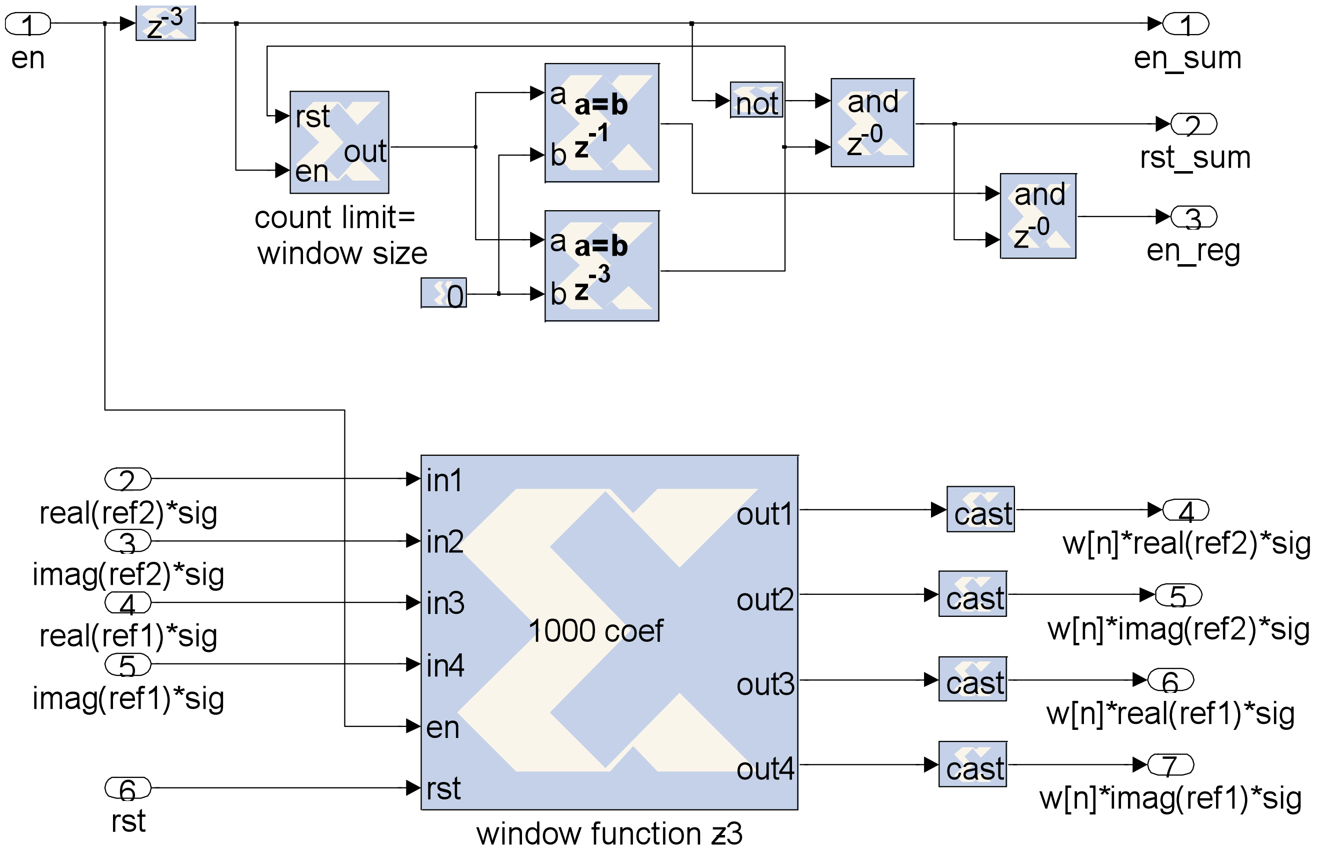


Figure A.8: Subsystem for applying window function.

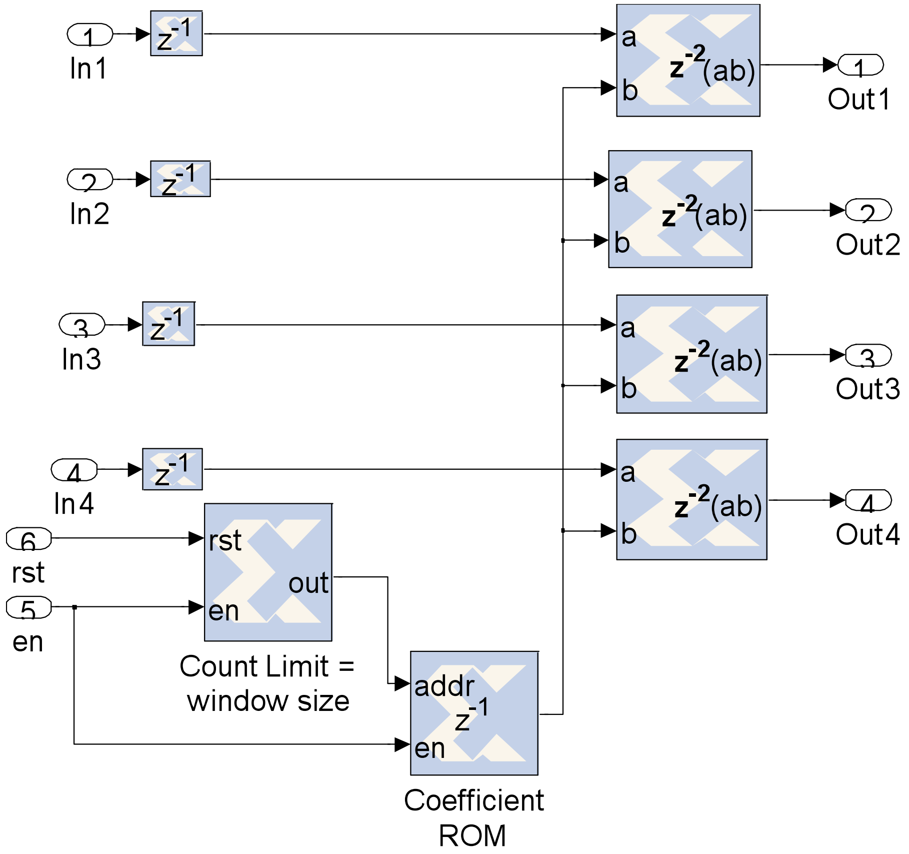


Figure A.9: Schematics of "masked" subsystem of applying user defined coefficients to data.

A.4 Figure 5.8 Subsystem of “Controlled n-tap MAC FIR Filter” and M-function Script for “gen_en_f” Block

A.4.1 Masked Subsystem of “Controlled n-tap MAC FIR Filter”

The input signal “en_asr” is derived from the spatial phase estimator, and the pulse is asserted after an error estimate is calculated. The pulse is then extended for n (determined by the order of the filter) clock cycles by the “gen_en_f” block (shown in the upper level block diagram in Figure 5.8), and routed to “en” input port of the filter block. When there’s an error estimate ready, ASR loads it into the first register according to “en_asr”, which is followed by the counter generating n addresses for “Coefficients ROM” and ASR for n clock cycles according to “en”. The ASR reads the data from the addressed register (always the latest one), drives it to the output and then shifts it to make room for the new data’. Therefore, it is ensured that the error estimates are loaded into filter only when they are ready and only the error estimates are processed by the filter. The “en_asr” signal is also used to generate memory control signal and addresses, so that only the filtered estimates are saved in ZBT SRAM. This control strategy is not supported by Xilinx “MAC FIR Filter” reference block, but is very important for our implementation so that false data can be prevented from being processed while beam is flying back.

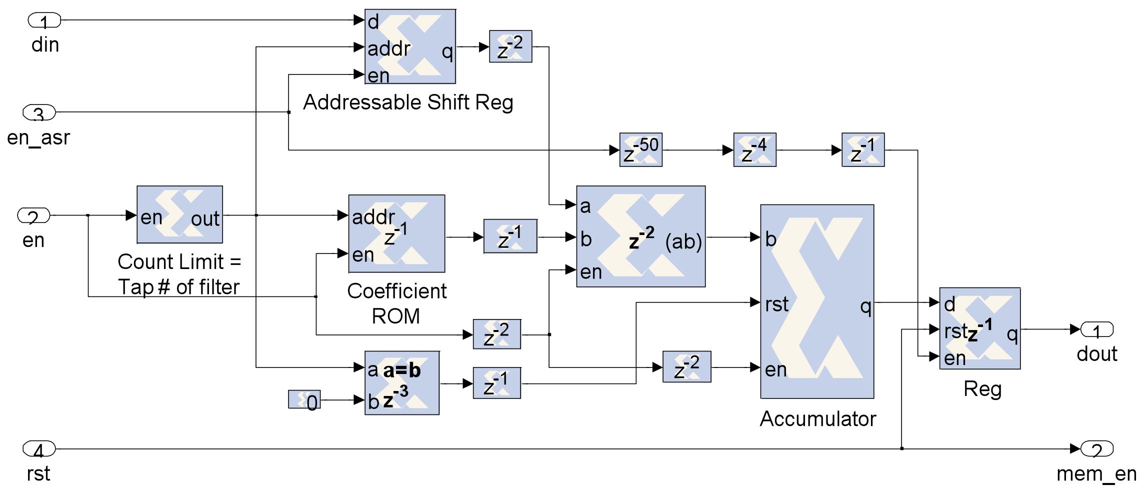


Figure A.10: Schematics of “Controlled n-tap MAC FIR Filter” subsystem.

The coefficients of 50-tap Hann low pass filter:

```
          0  9.4485e-003  4.3269e-002  4.0358e-002  7.0586e-003
1.2942e-005  1.2210e-002  4.5718e-002  3.7077e-002  5.0567e-003
8.8801e-005  1.5309e-002  4.7628e-002  3.3526e-002  3.4411e-003
2.8613e-004  1.8693e-002  4.8937e-002  2.9809e-002  2.1941e-003
6.6494e-004  2.2295e-002  4.9603e-002  2.6031e-002  1.2833e-003
1.2833e-003  2.6031e-002  4.9603e-002  2.2295e-002  6.6494e-004
2.1941e-003  2.9809e-002  4.8937e-002  1.8693e-002  2.8613e-004
3.4411e-003  3.3526e-002  4.7628e-002  1.5309e-002  8.8801e-005
5.0567e-003  3.7077e-002  4.5718e-002  1.2210e-002  1.2942e-005
7.0586e-003  4.0358e-002  4.3269e-002  9.4485e-003            0
```

The coefficients are read from top to bottom and then left to right.

A.4.2 M-function Script for "gen_en_f" Block

```
% The code only shows an example of extending a 1-clock-cycle wide
% positive pulse to a 5-clock-cycle wide pulse. Increase the number
% of states to achiver longer extension.

function [en_out]=gen_en_f(en_in,rst)

    persistent state, state = xl_state(0,{xlUnsigned,6,0});
    const0=xfix({xlUnsigned,1,0},0);
    en_out=xfix({xlBoolean,1,0},0);

    if rst
        state = const0;
        en_out = false;
    else
        switch state
            case 0
                if ~en_in
                    state = 0;
                    en_out = false;
                else
                    state = 1;
                    en_out = true;
                end
            case 1
                if ~en_in
                    state = 2;
                    en_out = true;
                else
                    state = 0;
                    en_out = false;
                end
            case 2
                if ~en_in
                    state = 3;
                    en_out = true;
                else
                    state = 0;
                    en_out = false;
                end
            case 3
                if ~en_in
                    state = 4;
                    en_out = true;
                else
                    state = 0;
                    en_out = false;
                end
            case 4
                if ~en_in
                    state = 5;
                    en_out = true;
                else
                    state = 0;
                    en_out = false;
                end
              case 5
                if ~en_in
                    state = 0;
                    en_out = false;
                else
                    state = 0;
                    en_out = false;
                end
        end
    end
```

A.5 Figure 6.4 M-function Script for “DoseCtrl_st” Block

```
function [bb_fpga, en_accum, rst_accum]=DoseCtrl_st(bb_raith, ...
bb_raithdl, rst, dot_done)

   persistent state, state = xl_state(0,{xlUnsigned,1,0});
   bb_fpga = true;
   en_accum = false;
   rst_accum = true;


    if rst
        state = 0;
        bb_fpga = true;
        en_accum = false;
        rst_accum = true;
    else
        switch state
            case 0
              %Before exposing the new dot or Exposure is finished%
                if bb_raith && bb_raithdl
                    state = 0;
                    bb_fpga = true;
                elseif ~bb_raith && bb_raithdl
                    state = 1;
                    en_accum = true;
                    bb_fpga = dot_done;
                    rst_accum = false;
                else
                    state = 0;
                end
            case 1
                %Exposing the dot%
                if ~dot_done
                    state = 1;
                    en_accum = true;
                    bb_fpga = dot_done;
                    rst_accum = false;
                else
                    state = 0;
                    bb_fpga = true;
                    en_accum = false;
                    rst_accum = true;
            end
        end
    end
```

Appendix B

HDL Design Entity

B.1 HDL Design for Virtex-II Clock FPGA

```
-- This HDL entity defines the clock signal routing on Virtex-II
-- clock FPGA on the development board.
-- The script is based on the example provided by Nallatech,
-- which has been modified to suit for our implementation.

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity GEN_CLKB is
    Port ( GEN_CLKB : in std_logic;
           DAC0_CLKp : out std_logic;
           DAC0_CLKn : out std_logic;
           DAC1_CLKp : out std_logic;
           DAC1_CLKn : out std_logic;
           ADC0_CLKp : out std_logic;
           ADC0_CLKn : out std_logic;
           ADC1_CLKp : out std_logic;
           ADC1_CLKn : out std_logic;
           CLK3_FB : out std_logic
   );
end GEN_CLKB;

architecture Behavioral of GEN_CLKB is

component BUFG
port (  I: in std_logic;
O: out std_logic
);
end component;

component OBUFDS_LVPECL_33
port ( O: out std_logic;
OB: out std_logic;
I: in std_logic
);
end component;

component OBUF
port (  I: in std_logic;
    O: out std_logic
);
end component;

signal CLKB_OUT: std_logic;
signal CLKB_OUTl: std_logic;

begin

-- Input software programmed clock signal through Virtex-4 user FPGA
H6: BUFG port map (I => GEN_CLKB, O => CLKB_OUT);

-- Generating clock signal for ADCs/DACs.
H1: OBUFDS_LVPECL_33
port map (I => CLKB_OUT, O => DAC0_CLKp, OB => DAC0_CLKn);
H2: OBUFDS_LVPECL_33
port map (I => CLKB_OUT, O => DAC1_CLKp, OB => DAC1_CLKn);
H3: OBUFDS_LVPECL_33
port map (I => CLKB_OUT, O => ADC0_CLKp, OB => ADC0_CLKn);
H4: OBUFDS_LVPECL_33
port map (I => CLKB_OUT, O => ADC1_CLKp, OB => ADC1_CLKn);

-- Output clock signal to Virtex-4 user FPGA so that the skews on
-- signal traveling down the nets to ADCs/DACs and Virtex-4 are
-- matched.
H5: OBUF port map (I => CLKB_OUT, O => CLK3_FB);

end Behavioral;
```

B.2 HDL Design for SPL on Virtex-4 User FPGA

```
------------------------------------------------------------
-- This HDL entity (1) instantiates the System Generator design of
-- position error estimation as "blackbox", (2) connects the ADC to
-- estimator’s input for grid signal, and (3) routes the estimated
-- position errors and memory addresses to the ZBT data and address
-- bus respectively. In order to load the estimator parameters (grid
-- rotation angle, grid period and pixel size) and distortion
-- correction coefficients into the estimation "blackbox", interface
-- core provided by Nallatech is instantiated in the entity also.
-- This information is stored in the host computer in advance,
-- and loaded into the registers and eventually to the estimation
--  "blackbox".
------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

library UNISIM;
use UNISIM.VComponents.all;

entity SPL_distortion_correction_input_if is
generic(
COEF_WIDTHg : integer range 1 to 32 := 12
);
port(
--- System signal
CLK3_FB : in std_logic;
CLKB : in std_logic;
GEN_CLKB : out std_logic;
PP0LK_10 : in std_logic;
PP0LK_11 : out std_logic;
RESETl : in std_logic;
CONFIG_DONE : out std_logic;
ADC1_D :  in std_logic_vector(13 downto 0);
--- Interface core I/O
EMPTY :  in std_logic;
BUSY :  in std_logic;
AS_DSl :  in std_logic;
RDl_WR :  out std_logic;
RENl_WENl :  out std_logic;
INTl :  out std_logic;
ADIO :  inout std_logic_vector(31 downto 0);
--- ZBT signal, both banks are used to store x- and y-position
--- errors respectively
ZBTA_CLK : out std_logic;
ZBTA_CLK_FB_OUT : out std_logic;
ZBTA_CLK_FB_IN : in std_logic;
ZBTA_ADV : out std_logic;
ZBTA_CLKEl : out std_logic;
ZBTA_CSl : out std_logic_vector(1 downto 0);
ZBTA_WEl : out std_logic;
ZBTA_OEl : out std_logic;
ZBTA_A : out std_logic_vector(18 downto 0);
ZBTA_D : out std_logic_vector(31 downto 0);
ZBTB_CLK : out std_logic;
ZBTB_CLK_FB_OUT : out std_logic;
ZBTB_CLK_FB_IN : in std_logic;
ZBTB_ADV : out std_logic;
ZBTB_CLKEl : out std_logic;
ZBTB_CSl : out std_logic_vector(1 downto 0);
ZBTB_WEl : out std_logic;
ZBTB_OEl : out std_logic;
ZBTB_A : out std_logic_vector(18 downto 0);
ZBTB_D : out std_logic_vector(31 downto 0)
--- DAC control signal, disconnected from the estimation "blackbox"
-- for now, but will be connected to the output of estimated position
--- errors when feedback control loop is closed.
--DAC1_D : out std_logic_vector(13 downto 0);
--DAC2_D : out std_logic_vector(13 downto 0);
--DAC1_RESET   : out std_logic;
--DAC2_RESET   :  out std_logic;
--DAC1_MOD0 : out std_logic;
--DAC1_MOD1 : out std_logic;
--DAC2_MOD0 : out std_logic;
--DAC2_MOD1 : out std_logic;
--DAC1_DIV0    :  out std_logic;
--DAC1_DIV1    :  out std_logic;
--DAC2_DIV0   :  out std_logic;
--DAC2_DIV1    : out std_logic
);
end SPL_distortion_correction_input_if;

architecture Structural of SPL_distortion_correction_input_if is

attribute syn_black_box : boolean;
attribute box_type : string;

component IBUFG
port(
I : in std_logic;
O : out std_logic
    );
end component;
attribute box_type of IBUFG : component is "black_box";
attribute syn_black_box of IBUFG : component is true;

component BUFG
port(
I : in std_logic;
O : out std_logic
    );
end component;
attribute box_type of BUFG : component is "black_box";
attribute syn_black_box of BUFG : component is true;

component OBUF
generic(
CAPACITANCE : string : ="DONT_CARE";
DRIVE     : integer : =12;
IOSTANDARD  : string : ="LVTTL";
SLEW     : string : ="SLOW"
);
port(
I : in std_logic;
O : out std_logic
    );
end component;
attribute box_type of OBUF : component is "black_box";
attribute syn_black_box of OBUF : component is true;

component IBUF
generic(
CAPACITANCE : string := "DONT_CARE";
DRIVE     : integer := 12;
IOSTANDARD  : string := "LVDCI_33";
SLEW     : string := "SLOW"
);
port(
I : in std_logic;
O : out std_logic
```

```
    );
end component;
attribute box_type of IBUF : component is "black_box";
attribute syn_black_box of IBUF : component is true;

-- Instantiate VHDL clock module.
-- This design was released by Xilinx for Virtex-4 silicon to
-- handle the new clocking constraints for stopping the clock
-- for more than 100ms and for holding the design in reset
-- for more than 10 sec.
component dcm_standby is
port (
CLK0        : OUT std_logic;
CLK180      : OUT std_logic;
CLK270      : OUT std_logic;
CLK2X       : OUT std_logic;
CLK2X180    : OUT std_logic;
CLK90       : OUT std_logic;
CLKDV       : OUT std_logic;
CLKFX       : OUT std_logic;
CLKFX180    : OUT std_logic;
DO          : OUT std_logic_vector(15 DOWNTO 0);
LOCKED      : OUT std_logic;
PSDONE      : OUT std_logic;
CLKFB       : IN std_logic;
CLKIN       : IN std_logic;
PSCLK       : IN std_logic;
PSEN        : IN std_logic;
PSINCDEC    : IN std_logic;
RST         : IN std_logic
    );
end component;

-- Instantiate the interface core provided by Nallatech
component SV_IFACE
    generic (
     NUM_BLOCKSg : integer range 1 to 29;-- := 4;
     BLOCK_SIZEg : integer range 1 to 29;--:= 8;
     NUM_REGSg : integer range 1 to 29-- := 4
     );
     port (
   CLK: in std_logic;
         RST1: in std_logic;
         BUSY: in std_logic;
         EMPTY: in std_logic;
         AS_DSl: in std_logic;
         DMA_WEN: in std_logic;
         DMA_REN: in std_logic;
         INT: in std_logic;
         RENl_WENl: out std_logic;
         RDl_WR: out std_logic;
         INTl: out std_logic;
         ADDRESS: out std_logic_vector(30 downto 0);
         WRITE_STROBE: out std_logic;
         READ_STROBE: out std_logic;
         COUNT: out std_logic_vector(31 downto 0);
         DMA_ENABLE: out std_logic;
         DMA_DIRECTION: out std_logic;
         DMA_SEL: out std_logic_vector(3 downto 0);
         DMA_RDY: out std_logic;
         DMA_DATA_AVAILABLE: out std_logic;
         RST: out std_logic;
         SYNC_RESET: out std_logic;
         DMA_RESET: out std_logic;
         ADIO: inout std_logic_vector(31 downto 0);
         DATA: inout std_logic_vector(31 downto 0);
         DMA_DATA: inout std_logic_vector(31 downto 0)
      );
end component;

-- Instantiate the System Generator design for position error
-- estimation as "blackbox".
component distortion_correction_pg1_coef_offset_Filter_delay_cw is
port(
 ce    : in std_logic := ’1’;
 clk    : in std_logic;
 rst    : in std_logic;
 en_pg    : in std_logic;
 ADC1_grid_signal1: in std_logic_vector(13 downto 0);
 theta_addr    : in std_logic_vector(7 downto 0);
 k0_addr    : in std_logic_vector(7 downto 0);
 pixel_size    : in std_logic_vector(3 downto 0);
 distortion_coef_x0 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x1 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x2 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x3 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x4 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x5 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x6 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x7 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x8 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_x9 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y0 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y1 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y2 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y3 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y4 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y5 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y6 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y7 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y8 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 distortion_coef_y9 : in std_logic_vector((COEF_WIDTHg-1) downto 0);
 y_error_estimate   : out std_logic_vector(31 downto 0);
 x_error_estimate   : out std_logic_vector(31 downto 0);
 mem_addr_x      : out std_logic_vector(18 downto 0);
 mem_addr_y         : out std_logic_vector(18 downto 0);
 wr_en_x         : out std_logic;
 wr_en_y      : out std_logic;
 Beam_Blanker      : out std_logic
 );
end component;
attribute box_type of
distortion_correction_pg1_coef_offset_Filter_delay_cw :
component is "black_box";

attribute syn_black_box of
distortion_correction_pg1_coef_offset_Filter_delay_cw :
component is true;


signal COEF_X0, COEF_X1, COEF_X2, COEF_X3, COEF_X4, COEF_X5,
  COEF_X6, COEF_X7, COEF_X8, COEF_X9 :
  std_logic_vector((COEF_WIDTHg-1) downto 0);

signal COEF_Y0, COEF_Y1, COEF_Y2, COEF_Y3, COEF_Y4, COEF_Y5,
  COEF_Y6, COEF_Y7, COEF_Y8, COEF_Y9 :
  std_logic_vector((COEF_WIDTHg-1) downto 0);

signal CLKIN_B, CLKFB_B, CLK_B, ENABLE, CLKB_GEN, BB,
  LOCKED1 : std_logic;
signal RESET, RST1, RST : std_logic;
signal GND : std_logic;
signal INT : std_logic;
signal DMA_REN, DMA_WEN : std_logic;
signal DMA_ENABLE, DMA_DIRECTION, DMA_DATA_AVAIL,
  DMA_RDY : std_logic;
signal WR_STROBE, RD_STROBE : std_logic;
signal ADDRESS : std_logic_vector(30 downto 0);
signal DATA, DMA_DATA : std_logic_vector(31 downto 0);

-- Instantiate registers and define control signals for estimation
-- parameters and distortion correction coefficients.
signal REG_THETA : std_logic_vector(7 downto 0);
signal REG_THETA_WR, REG_THETA_RD : std_logic;
signal REG_K0 : std_logic_vector(7 downto 0);
signal REG_K0_WR, REG_K0_RD : std_logic;
signal REG_PIXEL_SIZE : std_logic_vector(3 downto 0);
signal REG_PIXEL_SIZE_WR, REG_PIXEL_SIZE_RD : std_logic;
signal REG_COEF_X0, REG_COEF_X1, REG_COEF_X2, REG_COEF_X3,
  REG_COEF_X4, REG_COEF_X5, REG_COEF_X6, REG_COEF_X7,
  REG_COEF_X8, REG_COEF_X9 :
  std_logic_vector((COEF_WIDTHg-1) downto 0);
signal REG_COEF_Y0, REG_COEF_Y1, REG_COEF_Y2, REG_COEF_Y3,
  REG_COEF_Y4, REG_COEF_Y5, REG_COEF_Y6, REG_COEF_Y7,
  REG_COEF_Y8, REG_COEF_Y9 :
  std_logic_vector((COEF_WIDTHg-1) downto 0);
signal REG_COEF_X0_WR, REG_COEF_X0_RD : std_logic;
signal REG_COEF_X1_WR, REG_COEF_X1_RD : std_logic;
signal REG_COEF_X2_WR, REG_COEF_X2_RD : std_logic;
signal REG_COEF_X3_WR, REG_COEF_X3_RD : std_logic;
signal REG_COEF_X4_WR, REG_COEF_X4_RD : std_logic;
signal REG_COEF_X5_WR, REG_COEF_X5_RD : std_logic;
signal REG_COEF_X6_WR, REG_COEF_X6_RD : std_logic;
signal REG_COEF_X7_WR, REG_COEF_X7_RD : std_logic;
signal REG_COEF_X8_WR, REG_COEF_X8_RD : std_logic;
signal REG_COEF_X9_WR, REG_COEF_X9_RD : std_logic;
signal REG_COEF_Y0_WR, REG_COEF_Y0_RD : std_logic;
signal REG_COEF_Y1_WR, REG_COEF_Y1_RD : std_logic;
signal REG_COEF_Y2_WR, REG_COEF_Y2_RD : std_logic;
signal REG_COEF_Y3_WR, REG_COEF_Y3_RD : std_logic;
signal REG_COEF_Y4_WR, REG_COEF_Y4_RD : std_logic;
signal REG_COEF_Y5_WR, REG_COEF_Y5_RD : std_logic;
signal REG_COEF_Y6_WR, REG_COEF_Y6_RD : std_logic;
signal REG_COEF_Y7_WR, REG_COEF_Y7_RD : std_logic;
signal REG_COEF_Y8_WR, REG_COEF_Y8_RD : std_logic;
signal REG_COEF_Y9_WR, REG_COEF_Y9_RD : std_logic;

signal REN, WEN : std_logic;
signal DMA_READ, DMA_WRITE : std_logic;
signal THETA : std_logic_vector(7 downto 0);
signal K0 : std_logic_vector(7 downto 0);
signal PIXEL_SIZE : std_logic_vector(3 downto 0);
signal OFFSET_X, OFFSET_Y : std_logic_vector(9 downto 0);
signal ADC1, x_error, y_error : std_logic_vector(13 downto 0);

signal ZBTA_CLK_OUT, ZBTA_CLK_IN : std_logic;
signal ZBTA_WE, CLKEl_A : std_logic;
signal CSl_A, CSl_B : std_logic_vector(1 downto 0);
signal ZBTA_X_D : std_logic_vector(31 downto 0);
signal ZBTA_X_A : std_logic_vector(18 downto 0);

signal ZBTB_CLK_OUT, ZBTB_CLK_IN : std_logic;
signal ZBTB_WE, CLKEl_B : std_logic;
signal ZBTB_Y_D : std_logic_vector(31 downto 0);
signal ZBTB_Y_A : std_logic_vector(18 downto 0);

constant ZEROS : std_logic_vector(23 downto 0) := (others => ’0’);
constant ZEROS1 : std_logic_vector(27 downto 0) := (others => ’0’);
constant ONE : std_logic_vector ((COEF_WIDTHg-1) downto 0)
:= ( 10 => ’1’, others => ’0’);

begin

CONFIG_DONE <= ’0’;

-- Control signals for DACs to ensure desired behavior.
-- Set low pass filter response and no zero stuffing for both DACs
--   DAC1_MOD0 <= ’0’;
--   DAC1_MOD1 <= ’0’;
--   DAC2_MOD0 <= ’0’;
--   DAC2_MOD1 <= ’0’;
--
---- Disable resets for DACs
--   DAC1_RESET <= ’0’;
--   DAC2_RESET <= ’0’;
--
---- Optimum settings for sampling rate
--   DAC1_DIV0 <= ’1’;
--   DAC1_DIV1 <= ’0’;
--   DAC2_DIV0 <= ’1’;
--   DAC2_DIV1 <= ’0’;

GND <= ’0’;
RESET <= not RESETl;
```

```
U0_IBUFG : IBUFG
port map(
I => CLK3_FB,
O => CLKIN_B
);

U1_IBUFG : IBUFG
port map(
I => ZBTA_CLK_FB_IN,
O => ZBTA_CLK_IN
);

U2_IBUFG : IBUFG
port map(
I => ZBTB_CLK_FB_IN,
O => ZBTB_CLK_IN
);

U0_BUFG : BUFG
port map(
I => CLKB,
O => CLKB_GEN
);

GEN_CLKB <= CLKB_GEN;

U1_BUFG : BUFG
port map(
I => ZBTA_CLK_OUT,
O => ZBTA_CLK
);

U2_BUFG : BUFG
port map(
I => ZBTB_CLK_OUT,
O => ZBTB_CLK
);

U0_OBUF : OBUF
port map(
I => ZBTA_CLK_OUT,
O => ZBTA_CLK_FB_OUT
);
```

```
U1_OBUF : OBUF
port map(
I => ZBTB_CLK_OUT,
O => ZBTB_CLK_FB_OUT
);
CLK_B <= CLKFB_B;

-- Clock signal for the interface core and estimation "blackbox"
H0_SYSCLK_DCM : dcm_standby
port map(
CLKIN => CLKIN_B,
CLKFB => CLK_B,
PSINCDEC => GND,
PSEN => GND,
PSCLK => GND,
RST => RESET,
CLK0 => CLKFB_B,
LOCKED => RST1
);

-- Clock signal for ZBTA
H1_ZBTACLK_DCM : dcm_standby
port map(
CLKIN => CLKFB_B,
CLKFB => ZBTA_CLK_IN,
RST => RESET,
PSINCDEC => GND,
PSEN => GND,
PSCLK => GND,
CLK0 => ZBTA_CLK_OUT
);

-- Clock signal for ZBTB
H2_ZBTBCLK_DCM : dcm_standby
port map(
CLKIN => CLKFB_B,
CLKFB => ZBTB_CLK_IN,
RST => RESET,
PSINCDEC => GND,
PSEN => GND,
PSCLK => GND,
CLK0 => ZBTB_CLK_OUT
);

H1_PCICOMMS : SV_IFACE
```

```
    generic map (
     NUM_BLOCKSg => 1,
     BLOCK_SIZEg => 5,
     NUM_REGSg => 5
    )
     port map (
   CLK => CLK_B,
         RSTl => RSTl,
         BUSY => BUSY,
         EMPTY => EMPTY,
         AS_DSl => AS_DSl,
         DMA_WEN => DMA_WEN,
         DMA_REN => DMA_REN,
         INT => INT,
         RENl_WENl => RENl_WENl,
        RDl_WR => RDl_WR,
         INTl => INTl,
         ADDRESS => ADDRESS,
         WRITE_STROBE => WR_STROBE,
         READ_STROBE => RD_STROBE,
         COUNT => open,
         DMA_ENABLE => DMA_ENABLE,
         DMA_DIRECTION => DMA_DIRECTION,
         DMA_SEL => open,
         DMA_RDY => DMA_RDY,
         DMA_DATA_AVAILABLE => DMA_DATA_AVAIL,
         RST => RST,
         SYNC_RESET => open,
         DMA_RESET => open,
         ADIO => ADIO,
         DATA => DATA,
         DMA_DATA => DMA_DATA
);

spatial_phase_estimation :
distortion_correction_pg1_coef_offset_Filter_delay_cw
port map(
ce => ’1’,
clk => CLKFB_B,
rst => RSTl,
en_pg => ENABLE,
ADC1_grid_signal1 => ADC1,
theta_addr => THETA,
k0_addr => K0,
pixel_size => PIXEL_SIZE,
distortion_coef_x0 => COEF_X0,
distortion_coef_x1 => COEF_X1,
distortion_coef_x2 => COEF_X2,
distortion_coef_x3 => COEF_X3,
distortion_coef_x4 => COEF_X4,
distortion_coef_x5 => COEF_X5,
distortion_coef_x6 => COEF_X6,
distortion_coef_x7 => COEF_X7,
distortion_coef_x8 => COEF_X8,
distortion_coef_x9 => COEF_X9,
distortion_coef_y0 => COEF_Y0,
distortion_coef_y1 => COEF_Y1,
distortion_coef_y2 => COEF_Y2,
distortion_coef_y3 => COEF_Y3,
distortion_coef_y4 => COEF_Y4,
distortion_coef_y5 => COEF_Y5,
distortion_coef_y6 => COEF_Y6,
distortion_coef_y7 => COEF_Y7,
distortion_coef_y8 => COEF_Y8,
distortion_coef_y9 => COEF_Y9,
x_error_estimate => ZBTA_X_D,
y_error_estimate => ZBTB_Y_D,
mem_addr_x => ZBTA_X_A,
mem_addr_y => ZBTB_Y_A,
wr_en_x => ZBTA_WE,
wr_en_y => ZBTB_WE,
Beam_Blanker => BB
);

ZBTA_CSl <= CSl_A;
ZBTA_CLKEl <= CLKEl_A;
ZBTA_ADV <= GND;
ZBTA_OEl <= ’1’;

ZBTB_CSl <= CSl_B;
ZBTB_CLKEl <= CLKEl_B;
ZBTB_ADV <= GND;
ZBTB_OEl <= ’1’;

-- Define address and read/write control signal for each register
REG_THETA_WR <= ’1’
when WR_STROBE = ’1’ and ADDRESS(4 downto 0) = "00010" else ’0’;
REG_THETA_RD <= ’1’
when RD_STROBE = ’1’ and ADDRESS(4 downto 0) = "00010" else ’0’;
REG_COEF_X0_WR <= ’1’
```

```
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "00011" else '0';
REG_COEF_X0_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "00011" else '0';

REG_COEF_X1_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "00100" else '0';
REG_COEF_X1_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "00100" else '0';

REG_COEF_X2_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "00101" else '0';
REG_COEF_X2_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "00101" else '0';

REG_COEF_X3_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "00110" else '0';
REG_COEF_X3_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "00110" else '0';

REG_COEF_X4_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "00111" else '0';
REG_COEF_X4_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "00111" else '0';

REG_COEF_X5_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01000" else '0';
REG_COEF_X5_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01000" else '0';

REG_COEF_X6_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01001" else '0';
REG_COEF_X6_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01001" else '0';

REG_COEF_X7_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01010" else '0';
REG_COEF_X7_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01010" else '0';

REG_COEF_X8_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01011" else '0';
REG_COEF_X8_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01011" else '0';

REG_COEF_X9_WR <= '1'
```

```
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01100" else '0';
REG_COEF_X9_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01100" else '0';

REG_COEF_Y0_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01101" else '0';
REG_COEF_Y0_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01101" else '0';

REG_COEF_Y1_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01110" else '0';
REG_COEF_Y1_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01110" else '0';

REG_COEF_Y2_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "01111" else '0';
REG_COEF_Y2_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "01111" else '0';

REG_COEF_Y3_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10000" else '0';
REG_COEF_Y3_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10000" else '0';

REG_COEF_Y4_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10001" else '0';
REG_COEF_Y4_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10001" else '0';

REG_COEF_Y5_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10010" else '0';
REG_COEF_Y5_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10010" else '0';

REG_COEF_Y6_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10011" else '0';
REG_COEF_Y6_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10011" else '0';

REG_COEF_Y7_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10100" else '0';
REG_COEF_Y7_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10100" else '0';

REG_COEF_Y8_WR <= '1'
```

```
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10101" else '0';
REG_COEF_Y8_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10101" else '0';

REG_COEF_Y9_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10110" else '0';
REG_COEF_Y9_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10110" else '0';

REG_K0_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "10111" else '0';
REG_K0_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "10111" else '0';

REG_PIXEL_SIZE_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "11000" else '0';
REG_PIXEL_SIZE_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "11000" else '0';

REG_X_OFFSET_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "11001" else '0';
REG_X_OFFSET_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "11001" else '0';

REG_Y_OFFSET_WR <= '1'
when WR_STROBE = '1' and ADDRESS(4 downto 0) = "11010" else '0';
REG_Y_OFFSET_RD <= '1'
when RD_STROBE = '1' and ADDRESS(4 downto 0) = "11010" else '0';

-- Write parameters and coefficients from registers to the 32-bit
-- Local Bus, which is connected to Spartan-II interface FPGA
DATA <= ZEROS & REG_THETA
when REG_THETA_RD = '1' else (others => 'Z');
DATA <= ZEROS1(23 downto 0) & REG_K0
when REG_K0_RD = '1' else (others => 'Z');
DATA <= ZEROS1 & REG_PIXEL_SIZE
when REG_PIXEL_SIZE_RD = '1' else (others => 'Z');
DATA <= ZEROS(21 downto 0) & REG_X_OFFSET
when REG_X_OFFSET_RD = '1' else (others => 'Z');
DATA <= ZEROS(21 downto 0) & REG_Y_OFFSET
when REG_Y_OFFSET_RD = '1' else (others => 'Z');

DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X0
when REG_COEF_X0_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X1
```

```
when REG_COEF_X1_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X2
when REG_COEF_X2_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X3
when REG_COEF_X3_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X4
when REG_COEF_X4_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X5
when REG_COEF_X5_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X6
when REG_COEF_X6_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X7
when REG_COEF_X7_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X8
when REG_COEF_X8_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_X9
when REG_COEF_X9_RD = '1' else (others => 'Z');

DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y0
when REG_COEF_Y0_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y1
when REG_COEF_Y1_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y2
when REG_COEF_Y2_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y3
when REG_COEF_Y3_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y4
when REG_COEF_Y4_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y5
when REG_COEF_Y5_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y6
when REG_COEF_Y6_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y7
when REG_COEF_Y7_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y8
when REG_COEF_Y8_RD = '1' else (others => 'Z');
DATA <= ZEROS((31-COEF_WIDTHg) downto 0) & REG_COEF_Y9
when REG_COEF_Y9_RD = '1' else (others => 'Z');

-- Read parameters and coefficients to the specific registers from
-- the 32-bit Local Bus
process (RST, CLK_B)
begin
if RST = '1' then
REG_THETA <= (others => '0');
```

```
REG_K0 <= (others => ’0’);
REG_PIXEL_SIZE <= (others => ’0’);
REG_X_OFFSET <= "0100011000";
REG_Y_OFFSET <= "0100011000";
REG_COEF_X0 <= (others => ’0’);
REG_COEF_X1 <= ONE;
REG_COEF_X2 <= (others => ’0’);
REG_COEF_X3 <= (others => ’0’);
REG_COEF_X4 <= (others => ’0’);
REG_COEF_X5 <= (others => ’0’);
REG_COEF_X6 <= (others => ’0’);
REG_COEF_X7 <= (others => ’0’);
REG_COEF_X8 <= (others => ’0’);
REG_COEF_X9 <= (others => ’0’);
REG_COEF_Y0 <= (others => ’0’);
REG_COEF_Y1 <= ONE;
REG_COEF_Y2 <= (others => ’0’);
REG_COEF_Y3 <= (others => ’0’);
REG_COEF_Y4 <= (others => ’0’);
REG_COEF_Y5 <= (others => ’0’);
REG_COEF_Y6 <= (others => ’0’);
REG_COEF_Y7 <= (others => ’0’);
REG_COEF_Y8 <= (others => ’0’);
REG_COEF_Y9 <= (others => ’0’);
elsif CLK_B’event and CLK_B = ’1’ then
if REG_THETA_WR = ’1’ then
REG_THETA <= DATA(7 downto 0);
end if;
if REG_K0_WR = ’1’ then
REG_K0 <= DATA(7 downto 0);
end if;
if REG_PIXEL_SIZE_WR = ’1’ then
REG_PIXEL_SIZE <= DATA(3 downto 0);
end if;
if REG_X_OFFSET_WR = ’1’ then
REG_X_OFFSET <= DATA(9 downto 0);
end if;
if REG_Y_OFFSET_WR = ’1’ then
REG_Y_OFFSET <= DATA(9 downto 0);
end if;
if REG_COEF_X0_WR = ’1’ then
REG_COEF_X0 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X1_WR = ’1’ then
REG_COEF_X1 <= DATA((COEF_WIDTHg-1) downto 0);
```

```
end if;
if REG_COEF_X2_WR = ’1’ then
REG_COEF_X2 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X3_WR = ’1’ then
REG_COEF_X3 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X4_WR = ’1’ then
REG_COEF_X4 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X5_WR = ’1’ then
REG_COEF_X5 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X6_WR = ’1’ then
REG_COEF_X6 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X7_WR = ’1’ then
REG_COEF_X7 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X8_WR = ’1’ then
REG_COEF_X8 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_X9_WR = ’1’ then
REG_COEF_X9 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y0_WR = ’1’ then
REG_COEF_Y0 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y1_WR = ’1’ then
REG_COEF_Y1 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y2_WR = ’1’ then
REG_COEF_Y2 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y3_WR = ’1’ then
REG_COEF_Y3 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y4_WR = ’1’ then
REG_COEF_Y4 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y5_WR = ’1’ then
REG_COEF_Y5 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y6_WR = ’1’ then
REG_COEF_Y6 <= DATA((COEF_WIDTHg-1) downto 0);
```

```
end if;
if REG_COEF_Y7_WR = ’1’ then
REG_COEF_Y7 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y8_WR = ’1’ then
REG_COEF_Y8 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
if REG_COEF_Y9_WR = ’1’ then
REG_COEF_Y9 <= DATA((COEF_WIDTHg-1) downto 0);
end if;
end if;
end process;

-- Connect the registers to the estimation "blackbox" inputs to load
-- the parameters and coefficients to the DSP design.
THETA <= REG_THETA;
K0 <= REG_K0;
PIXEL_SIZE <= REG_PIXEL_SIZE;
OFFSET_X <= REG_X_OFFSET;
OFFSET_Y <= REG_Y_OFFSET;

COEF_X0 <= REG_COEF_X0;
COEF_X1 <= REG_COEF_X1;
COEF_X2 <= REG_COEF_X2;
COEF_X3 <= REG_COEF_X3;
COEF_X4 <= REG_COEF_X4;
COEF_X5 <= REG_COEF_X5;
COEF_X6 <= REG_COEF_X6;
COEF_X7 <= REG_COEF_X7;
COEF_X8 <= REG_COEF_X8;
COEF_X9 <= REG_COEF_X9;

COEF_Y0 <= REG_COEF_Y0;
COEF_Y1 <= REG_COEF_Y1;
COEF_Y2 <= REG_COEF_Y2;
COEF_Y3 <= REG_COEF_Y3;
COEF_Y4 <= REG_COEF_Y4;
COEF_Y5 <= REG_COEF_Y5;
COEF_Y6 <= REG_COEF_Y6;
COEF_Y7 <= REG_COEF_Y7;
COEF_Y8 <= REG_COEF_Y8;
COEF_Y9 <= REG_COEF_Y9;


-- connect DAC output to grid signal input of estimation "blackbox",
-- estimated position error and memory address outputs from
-- estimation "blackbox" to ZBTs’ data and address ports.
DisplacementCaculate : process(CLK_B, RSTl)
begin
if RSTl = ’0’ then
ENABLE <= ’1’;
ADC1 <= (others => ’0’);
ZBTA_D <= (others => ’0’);
ZBTA_A <= (others => ’1’);
ZBTA_WEl <= ’1’;
ZBTB_D <= (others => ’0’);
ZBTB_A <= (others => ’1’);
ZBTB_WEl <= ’1’;
PP0LK_11 <= ’1’;
elsif CLK_B = ’1’ and CLK_B’event then
ENABLE <= PP0LK_10;
ADC1 <= ADC1_D;
-- There is an inverting op-amp on the output of DACs prior to being
-- output via MCX connector. Proper conversion is necessary to
-- ensure output data to be matched with the voltage.
--DAC1_D <= not (not x_error(13) & x_error(12 downto 0));
--DAC2_D <= not (not y_error(13) & y_error(12 downto 0));

PP0LK_11 <= BB;
ZBTA_D <= ZBTA_X_D;
ZBTA_A <= ZBTA_X_A;
ZBTA_WEl <= ZBTA_WE;
ZBTB_D <= ZBTB_Y_D;
ZBTB_A <= ZBTB_Y_A;
ZBTB_WEl <= ZBTB_WE;
end if;
end process;

-- Define memory control signal.
process(RSTl)
begin
if RSTl = ’0’ then
CSl_A <= "11";
CSl_B <= "11";
CLKEl_A <= ’1’;
CLKEl_B <= ’1’;
else
CSl_A <= "00";
CSl_B <= "00";
CLKEl_A <= ’0’;
```

```
CLKE1_B <= '0';
end if;
end process;

end Structural;
```

B.3 HDL Design for communication between Memory and Host Computer

```
-------------------------------------------------------------------
-- Written by Paul Dunn
-- Copyright Nallatech 2002
-- Modified by Yugu Yang to allow 32-bit wide data transfer properly
-- between ZBTs and host computer.
-- The code here only shows the signal connection for ZBT1. To
-- bring the connection to ZBT2, follow the same manner.
-------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ZBT_TEST is
    Port (
             CLKC : in std_logic;
          CLKB : in std_logic;
    CLK_ZBT_IN : in std_logic;
    CLK_ZBT_OUT : out std_logic;
     ZBT_CLK : out std_logic;
            RSTl : in std_logic;
             EMPTY : in std_logic;
             BUSY : in std_logic;
             AS_DSl : in std_logic;
             RDl_WR : out std_logic;
             RENl_WENl : out std_logic;
             INTl : out std_logic;
             ADIO : inout std_logic_vector(31 downto 0);
             ZBT1_ADDR : out std_logic_vector(18 downto 0);
             ZBT1_DATA : inout std_logic_vector(31 downto 0);
             ZBT1_CEl : out std_logic_vector(1 downto 0);
             ZBT1_CENl : out std_logic;
             ZBT1_ADV_LDl : out std_logic;
             ZBT1_RD_WRl : out std_logic;
         ZBT1_OEl : out std_logic;
         CONFIG_DONE : out std_logic
   );
end ZBT_TEST;

architecture ZBT_TEST_arch of ZBT_TEST is
```

```
component BUFGDLL
port (
I: in std_logic;
O: out std_logic
);
end component;

component IBUFG
port (
I: in std_logic;
O: out std_logic
);
end component;

component OBUF
port (
I: in std_logic;
O: out std_logic
);
end component;

component BUFG
port (
I: in std_logic;
O: out std_logic
);
end component;

component DCM
port (
CLKIN     : in  std_logic;
CLKFB     : in  std_logic;
DSSEN     : in  std_logic;
PSINCDEC  : in  std_logic;
PSEN      : in  std_logic;
PSCLK     : in  std_logic;
RST       : in  std_logic;
CLK0      : out std_logic;
CLK90     : out std_logic;
CLK180    : out std_logic;
CLK270    : out std_logic;
CLK2X     : out std_logic;
CLK2X180  : out std_logic;
CLKDV     : out std_logic;
CLKFX     : out std_logic;
CLKFX180  : out std_logic;
LOCKED    : out std_logic;
PSDONE    : out std_logic;
STATUS    : out std_logic_vector(7 downto 0)
);
end component;

component SV_IFACE
    generic (
     NUM_BLOCKSg : integer range 1 to 29;-- := 4;
     BLOCK_SIZEg : integer range 1 to 29;--:= 8;
     NUM_REGSg : integer range 1 to 29-- := 4
     );
    port (
     CLK: in STD_LOGIC;
        RSTl: in STD_LOGIC;
        BUSY: in STD_LOGIC;
        EMPTY: in STD_LOGIC;
        AS_DSl: in STD_LOGIC;
        DMA_WEN: in STD_LOGIC;
        DMA_REN: in STD_LOGIC;
        INT: in STD_LOGIC;
        RENl_WENl: out STD_LOGIC;
        RDl_WR: out STD_LOGIC;
        INTl: out STD_LOGIC;
        ADDRESS: out STD_LOGIC_VECTOR (30 downto 0);
        WRITE_STROBE: out STD_LOGIC;
        READ_STROBE: out STD_LOGIC;
        COUNT: out STD_LOGIC_VECTOR (31 downto 0);
        DMA_ENABLE: out STD_LOGIC;
        DMA_DIRECTION: out STD_LOGIC;
        DMA_SEL: out STD_LOGIC_VECTOR (3 downto 0);
        DMA_RDY: out STD_LOGIC;
        DMA_DATA_AVAILABLE: out STD_LOGIC;
        RST: out STD_LOGIC;
        SYNC_RESET: out STD_LOGIC;
        DMA_RESET: out STD_LOGIC;
        ADIO: inout STD_LOGIC_VECTOR (31 downto 0);
        DATA: inout STD_LOGIC_VECTOR (31 downto 0);
        DMA_DATA: inout STD_LOGIC_VECTOR (31 downto 0)
    );
end component;

component ZBT_BLK
```

```
    Port (
        DATACLK : in std_logic;
    SYSCLK : in std_logic;
           ZBTCLK : in std_logic;
           RST : in std_logic;
      SYNC_RST : in std_logic;
       WR_STROBES : in std_logic_vector(0 downto 0);
       ENABLE : in std_logic;
       COMPLETE : out std_logic;
       READ_WRITEl : in std_logic;
       DATA : in std_logic_vector(31 downto 0);
           DATAIN : in std_logic_vector(31 downto 0);
           DATA_WEN : in std_logic;
           DATA_AFULL : out std_logic;
           DATAOUT : out std_logic_vector(31 downto 0);
           DATA_REN : in std_logic;
           DATA_EMPTY : out std_logic;
           ZBT_ADDR : out std_logic_vector(18 downto 0);
           ZBT_DATA : inout std_logic_vector(31 downto 0);
           ZBT_CEl : out std_logic_vector(1 downto 0);
           ZBT_CENl : out std_logic;
           ZBT_ADV_LDl : out std_logic;
           ZBT_RD_WRl : out std_logic;
   ZBT_OEl : out std_logic
   );
end component;

-- Assign addresses to ZBT control/status and count register
constant ZBT_CSR_ADDRc : std_logic_vector(2 downto 0) := "010";
constant ZBT1_COUNT_ADDRc : std_logic_vector(2 downto 0) := "011";

signal SYSCLKin, SYSCLKdll, SYSCLKi: std_logic;
signal SYSCLK2xdll, SYSCLK2xi: std_logic := '1';
signal DSPCLKin, DSPCLKdll, DSPCLKi: std_logic;
signal ZBTCLKin, ZBTCLKdll: std_logic;

signal LOCKED1, LOCKED2, LOCKED3: std_logic;

signal RESET, RESETl: std_logic;
signal RST, SYNC_RST: std_logic;

signal ADDRESS: std_logic_vector(30 downto 0);
signal WR_STROBE, RD_STROBE: std_logic;

signal WR_STROBES: std_logic_vector(1 downto 0);
signal ZBT_CSR: std_logic_vector(31 downto 0);
signal ZBT_CSR_EN, ZBT_CSR_WR, ZBT_CSR_RD: std_logic;

signal COMPLETE: std_logic_vector(1 downto 0);

signal DMA_WRITE, DMA_READ: std_logic;
signal DMA_WEN, DMA_REN: std_logic;
signal DMA_ENABLE, DMA_DIRECTION: std_logic;
signal DMA_SEL: std_logic_vector(3 downto 0);

signal DMA_COUNT_EN, DMA_COUNT_WR: std_logic;
signal COUNT1, COUNT2: std_logic_vector(31 downto 0);

signal DMA_DATA, DATA: std_logic_vector(31 downto 0);
signal DMA_DATA_AVAIL, DMA_RDY: std_logic;

signal REN, WEN: std_logic;
signal DMA_WEN1: std_logic;
signal DMA_REN1: std_logic;
signal DATAIN1: std_logic_vector(31 downto 0);
signal DATAOUT1: std_logic_vector(31 downto 0);
signal DATA_AFULL1: std_logic;
signal DATA_EMPTY1: std_logic;

signal INT: std_logic;

signal ZBT1_COUNT_EN : std_logic;
signal ZBT1_COUNT_WR : std_logic;

signal ZEROS: std_logic_vector(31 downto 0);
signal ZERO: std_logic;

signal REG_DATA : std_logic_vector(31 downto 0);

signal COUNTER1, COUNTER2 : std_logic_vector(31 downto 0);

begin

CONFIG_DONE <= '0';

process (RST, DSPCLKi)
begin
if RST='1' then
COUNTER1 <= (others => '0');
elsif DSPCLKi'event and DSPCLKi='1' then
```

```
COUNTER1 <= COUNTER1+1;
end if;
end process;

process (RST, SYSCLKi)
begin
if RST='1' then
        COUNTER2 <= (others => '1');
elsif SYSCLKi'event and SYSCLKi='1' then
COUNTER2 <= COUNTER2+1;
end if;
end process;


ZEROS <= (others => '0');
ZERO <= '0';

RESET <= not RST1;

H1_SYSCLK_IBUFG : IBUFG
port map (
I => CLKC,
O => SYSCLKin
);

H2_SYSCLK_DCM : DCM
port map (
CLKIN => SYSCLKin,
CLKFB => SYSCLKi,
DSSEN => ZERO,
PSINCDEC => ZERO,
PSEN => ZERO,
PSCLK => ZERO,
RST => RESET,
CLK0 => SYSCLKdll,
CLK2X => SYSCLK2xdll,
LOCKED => LOCKED1
);

H3_SYSCLK_BUFG : BUFG
port map (
I => SYSCLKdll,
O => SYSCLKi
);
```

```
H4_ZBTCLK_IBUFG : IBUFG
port map (
I => CLK_ZBT_IN,
O => ZBTCLKin
);

H5_ZBTCLK_DCM : DCM
port map (
CLKIN => SYSCLKin,
CLKFB => ZBTCLKin,
DSSEN => ZERO,
PSINCDEC => ZERO,
PSEN => ZERO,
PSCLK => ZERO,
RST => RESET,
CLK0 => ZBTCLKdll,
LOCKED => LOCKED2
);

H6_ZBTCLK_OBUF : OBUF
port map (
I => ZBTCLKdll,
O => CLK_ZBT_OUT
);

H7_DSPCLK_IBUFG : IBUFG
port map (
I => CLKB,
O => DSPCLKin
);

H8_DSPCLK_DCM : DCM
port map (
CLKIN => DSPCLKin,
CLKFB => DSPCLKi,
DSSEN => ZERO,
PSINCDEC => ZERO,
PSEN => ZERO,
PSCLK => ZERO,
RST => RESET,
CLK0 => DSPCLKdll,
LOCKED => LOCKED3
);

H9_DSPCLK_BUFG : BUFG
```

```
port map (
I => DSPCLKdll,
O => DSPCLKi
);

H10_ZBTCLK_OBUF : OBUF
port map (
I => ZBTCLKdll,
O => ZBT_CLK
);

RESETl <= not RESET;

H1_PCICOMMS : SV_IFACE
    generic map (
     NUM_BLOCKSg => 1,
     BLOCK_SIZEg => 4,
     NUM_REGSg => 3
     )
    port map (
     CLK => DSPCLKi,
        RSTl => RESETl,
        BUSY => BUSY,
        EMPTY => EMPTY,
        AS_DSl => AS_DSl,
        DMA_WEN => DMA_WEN,
        DMA_REN => DMA_REN,
        INT => INT,
        RENl_WENl => RENl_WENl,
        RDl_WR => RDl_WR,
        INTl => INTl,
        ADDRESS => ADDRESS,
        WRITE_STROBE => WR_STROBE,
        READ_STROBE => RD_STROBE,
        COUNT => COUNT1,
        DMA_ENABLE => DMA_ENABLE,
        DMA_DIRECTION => DMA_DIRECTION,
        DMA_SEL => DMA_SEL,
        DMA_RDY => DMA_RDY,
        DMA_DATA_AVAILABLE => DMA_DATA_AVAIL,
        RST => RST,
        SYNC_RESET => SYNC_RST,
        DMA_RESET => open,
        ADIO => ADIO,
        DATA => DATA,
        DMA_DATA => DMA_DATA
    );

REG_DATA <= DATA;

H2_ZBTBLK1 : ZBT_BLK
    Port Map (
        DATACLK => DSPCLKi,
       SYSCLK => SYSCLKi,
           ZBTCLK => SYSCLKi,
           RST => RST,
       SYNC_RST => SYNC_RST,
       WR_STROBES => WR_STROBES(0 downto 0),
       ENABLE => ZBT_CSR(0),
       COMPLETE => COMPLETE(0),
       READ_WRITEl => ZBT_CSR(1),
       DATA => REG_DATA,
           DATAIN => DATAIN1,
           DATA_WEN => DMA_REN1,
           DATA_AFULL => DATA_AFULL1,
           DATAOUT => DATAOUT1,
           DATA_REN => DMA_WEN1,
           DATA_EMPTY => DATA_EMPTY1,
           ZBT_ADDR => ZBT1_ADDR,
           ZBT_DATA => ZBT1_DATA,
           ZBT_CEl => ZBT1_CEl,
           ZBT_CENl => ZBT1_CENl,
           ZBT_ADV_LDl => ZBT1_ADV_LDl,
           ZBT_RD_WRl => ZBT1_RD_WRl,
       ZBT_OEl => ZBT1_OEl
       );

--Address Decode
DMA_COUNT_EN <= ’1’ when ADDRESS(4)=’0’ and
ADDRESS(2 downto 0)="001" else ’0’;
ZBT_CSR_EN <= ’1’ when ADDRESS(4)=’0’ and
ADDRESS(2 downto 0)=ZBT_CSR_ADDRc else ’0’;
ZBT1_COUNT_EN <= ’1’ when ADDRESS(4)=’0’ and
ADDRESS(2 downto 0)=ZBT1_COUNT_ADDRc else ’0’;

DMA_COUNT_WR <= ’1’ when DMA_COUNT_EN=’1’ and WR_STROBE=’1’
else ’0’;

ZBT_CSR_WR <= ’1’ when ZBT_CSR_EN=’1’ and WR_STROBE=’1’ else ’0’;
ZBT_CSR_RD <= ’1’ when ZBT_CSR_EN=’1’ and RD_STROBE=’1’ else ’0’;
```

```
ZBT1_COUNT_WR <= '1'
when ZBT1_COUNT_EN='1' and WR_STROBE='1' else '0';

WR_STROBES(0) <= ZBT1_COUNT_WR;

process (RST, DSPCLKi)
begin
if RST='1' then
ZBT_CSR <= (others => '0');
COUNT2 <= (others => '0');
      elsif DSPCLKi'event and DSPCLKi='1' then
ZBT_CSR(4) <= COMPLETE(0);
ZBT_CSR(5) <= COMPLETE(1);
if ZBT_CSR_WR='1' then
ZBT_CSR(3 downto 0) <= DATA(3 downto 0);
end if;
if DMA_COUNT_WR='1' then
COUNT2 <= DATA;
elsif REN='1' then
COUNT2 <= COUNT2-1;
end if;
  end if;
end process;

DATA <= ZBT_CSR when ZBT_CSR_RD='1' else (others => 'Z');

WEN <= DMA_REN1 when DMA_SEL="0000" else
   DMA_REN2 when DMA_SEL="0001" else
   '0';

DMA_REN <= WEN;

REN <= DMA_WEN1 when DMA_SEL="0000" else
       DMA_WEN2 when DMA_SEL="0001" else
       '0';

process (RST, DSPCLKi)
begin
if RST='1' then
DMA_WEN <= '0';
elsif DSPCLKi'event and DSPCLKi='1' then
DMA_WEN <= REN;
        end if;
end process;
```

```
DMA_READ <= '1' when (DMA_ENABLE='1') and (DMA_DIRECTION='0') and
(COUNT1 /= ZEROS) else '0';
DMA_WRITE <= '1' when (DMA_ENABLE='1') and (DMA_DIRECTION='1') and
(COUNT2 /= ZEROS) else '0';

DMA_REN1 <= '1' when DMA_READ='1' and DMA_SEL="0000" and
DMA_DATA_AVAIL='1' and DATA_AFULL1='0' else '0';

DMA_WEN1 <= '1' when DMA_WRITE='1' and DMA_SEL="0000" and
DMA_RDY='1' and DATA_EMPTY1='0' else '0';

DATAIN1 <= DMA_DATA;

DMA_DATA <= DATAOUT1 when DMA_DIRECTION='1' and DMA_SEL="0000"
else (others => 'Z');

end ZBT_TEST_arch;
```

Appendix C

DIMEScript Code

C.1 DIMEScript Code for Loading Estimation Parameters and Distortion Correction Coefficients to Virtex-4 User FPGA

```
### This script defines the following actions: (1) download the bit
### files to the specific FPGA, (2) load the estimation parameters
### and distortion correction coefficients in the registers
### instantiated in the HDL entity for Virtex-4 user FPGA.
opencard pci

printconfig

sysreset e

fpgareset e

# Clock frequency in KHz
clockset 2 40000 # Set DSP Clock

# Download the bitfile to the specific FPGA
config 0 1 spl_distortion_correction_input_if.bit
config 0 0 gen_clkb.bit

# Module 0 - BenADDA DIME-II
# Module 1 - components in the JTAG chain on BenONE-Kit motherboard
# Device 0 - Clock FPGA
# Device 1 - Main user FPGA

pcireset
```

```
fpgareset d

sysreset d

# Define the size of the data area
dsize D0 20
dsize D1 3

# Load parameters and distortion correction coefficients into
# the data area
load D0 coef_reinterpret.csv
load D1 para.txt

# Set the base for output routines to decimal
base d

# Write the parameters from the data area to specific registers.
print write k0
writeaddr 0x00000017
writedata 1 D1 0

print write pixel_size
writeaddr 0x00000018
writedata 1 D1 1

print write theta
writeaddr 0x00000002
writedata 1 D1 2


# Write distortion correction coefficients from the data area
# to specific registers.
print write coef_x0
writeaddr 0x00000003
writedata 1 D0 0

print write coef_x1
writeaddr 0x00000004
writedata 1 D0 1

print write coef_x2
writeaddr 0x00000005
writedata 1 D0 2

print write coef_x3
```

```
writeaddr 0x00000006
writedata 1 D0 3

print write coef_x4
writeaddr 0x00000007
writedata 1 D0 4

print write coef_x5
writeaddr 0x00000008
writedata 1 D0 5

print write coef_x6
writeaddr 0x00000009
writedata 1 D0 6

print write coef_x7
writeaddr 0x0000000A
writedata 1 D0 7

print write coef_x8
writeaddr 0x0000000B
writedata 1 D0 8

print write coef_x9
writeaddr 0x0000000C
writedata 1 D0 9


print write coef_y0
writeaddr 0x0000000D
writedata 1 D0 10

print write coef_y1
writeaddr 0x0000000E
writedata 1 D0 11

print write coef_y2
writeaddr 0x0000000F
writedata 1 D0 12

print write coef_y3
writeaddr 0x00000010
writedata 1 D0 13

print write coef_y4
```

```
writeaddr 0x00000011
writedata 1 D0 14

print write coef_y5
writeaddr 0x00000012
writedata 1 D0 15

print write coef_y6
writeaddr 0x00000013
writedata 1 D0 16

print write coef_y7
writeaddr 0x00000014
writedata 1 D0 17

print write coef_y8
writeaddr 0x00000015
writedata 1 D0 18

print write coef_y9
writeaddr 0x00000016
writedata 1 D0 19

# Read out the parameters and coefficients from register to display
print read k0
writeaddr 0x80000017
readdata 1 1

print read pixel_size
writeaddr 0x80000018
readdata 1 1

print read theta
writeaddr 0x80000002
readdata 1 1

print read coef_x0
writeaddr 0x80000003
readdata 1 1

print read coef_x1
writeaddr 0x80000004
readdata 1 1

print read coef_x2
```

```
writeaddr 0x80000005
readdata 1 1

print read coef_x3
writeaddr 0x80000006
readdata 1 1

print read coef_x4
writeaddr 0x80000007
readdata 1 1

print read coef_x5
writeaddr 0x80000008
readdata 1 1

print read coef_x6
writeaddr 0x80000009
readdata 1 1

print read coef_x7
writeaddr 0x8000000A
readdata 1 1

print read coef_x8
writeaddr 0x8000000B
readdata 1 1

print read coef_x9
writeaddr 0x8000000C
readdata 1 1


print read coef_y0
writeaddr 0x8000000D
readdata 1 1

print read coef_y1
writeaddr 0x8000000E
readdata 1 1

print read coef_y2
writeaddr 0x8000000F
readdata 1 1

print read coef_y3
```

```
writeaddr 0x80000010
readdata 1 1

print read coef_y4
writeaddr 0x80000011
readdata 1 1

print read coef_y5
writeaddr 0x80000012
readdata 1 1

print read coef_y6
writeaddr 0x80000013
readdata 1 1

print read coef_y7
writeaddr 0x80000014
readdata 1 1

print read coef_y8
writeaddr 0x80000015
readdata 1 1

print read coef_y9
writeaddr 0x80000016
readdata 1 1


closecard pci
```

C.2 DIMEScript Code for Reading Data from ZBT Memory

```
### This script defines the following actions: (1) download bit file
###  to Virtex-4 FPGA , (2) define data areas for addresses and
### data respectively, (3) load addresses from a text file to one of
### the data area, (4) setup a loop and call another script to read
### data from memory.
opencard pci

sysreset e
fpgareset e

clockset 2 40000 # Set DSP Clock
clockset 1 100000 # Set SYS Clock

config 0 1 zbt_test.bit

pcireset
fpgareset d
sysreset d

dsize D0 65536
dsize D1 65536

# Load memory addresses from text file "readmem.txt" to data area
load D0 readmem.txt
base d

# Disable ZBT control/status register
writeaddr 2
writedata 1 0

# Disable DMA control/status register
writeaddr 0
writedata 1 0

print Setting Up Memory Read

# Set loop iteration number, and loop variable !0 is used within
# the loop
loop 16
# Set internal variable %1 to be loop variable multiplied with 4096,
# which is used in the following "readback.dsc" file
```

```
setvar 1 !0*4096.
# Call another script file
call readback.dsc
endloop

# Save the data to a text file.
print Saving Data
save D1 mem_data.txt n 1

closecard

### This script is called by the previous script, and defines the
### following actions: (1) load addresses from data area to ZBT
### controller, (2) read data from memory to a different data area
### through ZBT controller and DMA, (3) save the data in a text
### file.
writeaddr 3
writedata 1 4096

print ZBT CSR
# Setup ZBT controller to read addresses
writeaddr 2
writedata 1 3

print DMA Count
# Load number of addresses to be transferred in DMA count register
writeaddr 1
writedata 1 4096

print DMA CSR
# Setup DMA to read addresses
writeaddr 0
writedata 1 1

# Write the addresses to ZBT controller through DMA
writedata 4096 D0 %1

# Poll DMA count, if count = 0, data transfer stops
writeaddr 0x80000001
readdata 1 1

print Disabling Registers
# Disable DMA control/status register
writeaddr 0
writedata 1 0
```

```
print Setting Up DMA Count
# load number of data to be transferred in DMA control/status
# register
writeaddr 1
writedata 1 4096

print Enable DMA Read
# Set DMA in read mode
writeaddr 0
writedata 1 3

print Reading Data

# Load data from memory to data area through ZBT controller and DMA
readdata 4096 D1 %1
```

Bibliography

[1] G. E. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

[2] J. A. Hoerni, “Method of manufacturing semiconductor devices,” U.S. Patent 3,025,589, 1959.

[3] ——, “Semiconductor device,” U.S. Patent 3,064,167, 1960.

[4] I. T. R. for Semiconductors. (2011) Lithography. [Online]. Available: http://www.itrs.net

[5] J. T. Hastings, F. Zhang, and H. I. Smith, “Nanometer-level stitching in raster-scanning electron-beam lithography using spatial-phase locking,” *Journal of Vacuum Science & Technology B*, vol. 21, pp. 2650–2656, 2003.

[6] H. C. Pfeiffer, “New imaging and deflection concept for probe-forming microfabrication systems,” *Journal of Vacuum Science & Technology*, vol. 12, pp. 1170–1173, 1975.

[7] ——, “Variable spot shaping for electron-beam lithography,” *Journal of Vacuum Science & Technology*, vol. 15, pp. 887–890, 1977.

[8] M. J. R. Mark A. McCord, *SPIE Handbook of Microlithography, Micromachining and Microfabrication*, P. Rai-Choudhury, Ed., vol. 1.

[9] W. Chen, “Fabrication of 5-7 nm wide etched lines in silicon using 100 KeV electron-beam lithography and polymethylmethacrylate resist,” *Applied Physics Letters*, vol. 62, pp. 1499–1501, 1993.

[10] A. Grigorescua, M. van der Krogtb, C. Hagena, and P. Kruita, “10 nm lines and spaces written in HSQ, using electron beam lithographty,” *Microelectronic Engineering*, vol. 84, pp. 822–824, 2007.

[11] J. Vila-Comamala, S. Gorelick, V. A. Guzenko, E. Färm, M. Ritala, and C. David, “Dense high aspect ratio hydrogen silsesquioxane nanostructures by 100 KeV electron beam lithography,” *Nanotechnology*, vol. 21, p. 285305, 2010.

[12] D. M. Tanenbaum, C. W. Lo, M. Isaacson, H. G. Craighead, M. J. Rooks, K. Y. Lee, W. S. Huang, and T. H. P. Chang, “High resolution electron beam

lithographt using zep-520 and krs resists at low voltage," *Journal of Vacuum Science & Technology B*, vol. 14, pp. 3829–3833, 1996.

[13] M. A. McCord and T. H. Newman, "Low voltage, high resolution studies of electron beam resist exposure and proximity effect," *Journal of Vacuum Science & Technology B*, vol. 10, pp. 3083–3088, 1992.

[14] C. W. Lo, M. J. Rooks, W. K. Lo, M. Isaacson, and H. G. Craighead, "Resists and processes for 1 KV electron beam microcolumn lithography," *Journal of Vacuum Science & Technology B*, vol. 13, pp. 812–820, 1995.

[15] N. scale patterning of polystyrene resists in low-voltage electron beam lithography, "S. manako and j. fujita and y. ochiai and e. nomura and s. matsui," *Japanese Journal of Applied Physics*, vol. 36, pp. 7773–7776, 1997.

[16] M. A. McCord, T. H. P. Chang, D. P. Kern, and J. L. Speidell, "A novel scanning tunneling microscope controlled field emission microlens electron source," *Journal of Vacuum Science & Technology B*, vol. 7, pp. 1851–1854, 1989.

[17] K. Kratschmer, H. S. Kim, M. G. R. Thomson, K. Y. Lee, S. A. Rishton, M. L. Yu, and T. H. P. Chang, "An electron-beam microcolumn with improved resolution, beam current, and stability," *Journal of Vacuum Science & Technology B*, vol. 13, no. 2498-2503, 1995.

[18] J. P. Spallas, C. S. Silver, and L. P. Muray, "Arrayed miniature electron beam columns for mask making," *Journal of Vacuum Science & Technology B*, vol. 24, pp. 2892–2896, 2006.

[19] C. S. Silver, J. P. Spallas, and L. P. Muray, "Multiple beam sub-80-nm lithography with miniature electron beam column arrays," *Journal of Vacuum Science & Technology B*, vol. 25, pp. 2258–2265, 2007.

[20] L. P. Muray, C. S. Silver, and J. P. Spallas, "Sub-100-nm lithography with miniature electron beam columns," *Journal of Vacuum Science & Technology B*, vol. 24, pp. 2945–2950, 2006.

[21] K. M. Satyalakshmi, A. Olkhovets, M. G. Metzler, C. K. Harnett, D. M. Tanenbaum, and H. G. Craighead, "Charge induced pattern distortion in low energy electron beam lithography," *Journal of Vacuum Science & Technology B*, vol. 18, pp. 3122–3125, 2000.

[22] R. Abargues, U. Nickel, and P. J. Rodriguez-Canto, "Charge dissipation in e-beam lithography with Novolak-based conducting polymer films," *Nanotechnology*, vol. 19, p. 125302, 2008.

[23] D. O. Patterson, Ed., *A submicron e-beam lithography process using an overcoating conducting polymer for the reduction of beam charging effects on lithium niobate and quartz*, vol. 2194. SPIE, 1994.

[24] R. Dylewicz, S. Lis, R. D. L. Rue, and F. Rahman, "Charge dissipation layer based on conductive polymer for electron-beam patterning of bulk zinc oxide," *Electronics Letters*, vol. 46, pp. 1025–1027, 2010.

[25] H. I. Smith, S. D. Hector, M. L. Schattenburg, and E. H. Anderson, "A new approach to high fidelity e-beam and ion-beam lithography based on an in situ global-fiducial grid," *Journal of Vacuum Science & Technology B*, vol. 9, pp. 2992–2995, 1991.

[26] V. V. Wong, J. Ferrera, J. N. Damask, J. M. Carter, E. E. Moon, H. A. Haus, and H. I. Smith, "Spatial-phase-locked electron-beam lithography and X-ray lithography for fabricating first-order gratings on rib waveguides," *Journal of Vacuum Science & Technology B*, vol. 12, no. 3741-3745, 1994.

[27] V. V. Wong, J. Ferrera, J. N. Damask, T. E. Murphy, H. I. Smith, and H. A. Haus, "Distributed Bragg grating integrated-optical filters: synthesis and fabrication," *Journal of Vacuum Science & Technology B*, vol. 13, pp. 2859–2864, 1995.

[28] J. Goodberlet, J. Ferrera, and H. I. Smith, "Spatial-phase-locked electron-beam lithography with a delay-locked loop," *Journal of Vacuum Science & Technology B*, vol. 15, pp. 2293–2297, 1997.

[29] J. T. Hastings, F. Zhang, M. A. Finlayson, J. G. Goodberlet, and H. I. Smith, "Two-dimensional spatial-phase-locked electron-beam lithography via sparse sampling," *Journal of Vacuum Science & Technology B*, vol. 18, pp. 3268–3271, 2000.

[30] B. Icard, L. Pain, V. Arnal, S. Manakli, J.-C. Le-Denmat, P. Brun, M. Vincent, C. Soonkindt, B. Minghetti, and T. Matsumiya, "45 nm node back end of the line yield evaluation on ultrahigh density interconnect structures using electron beam direct write lithography," *Journal of Vacuum Science & Technology B*, vol. 25, pp. 124–129, 2007.

[31] P. Kruit, S. Steenbrink, and M. Wieland, "Predicted effect of shot noise on contact hole dimension in e-beam lithography," *Journal of Vacuum Science & Technology B*, vol. 24, pp. 2931–2935, 2006.

[32] N. Rau, F. Stratton, C. Fields, T. Ogawa1, A. Neureuther, R. Kubena, and G. Willson, "Shot-noise and edge roughness effects in resists patterned at 10 nm exposure," *Journal of Vacuum Science & Technology B*, vol. 16, pp. 3784–3788, 1998.

[33] D. Rio, C. Constancias, M. Saied, B. Icard, and L. Pain, "Study on line edge roughness for electron beam acceleration voltages from 50 to 5 KeV," *Journal of Vacuum Science & Technology B*, vol. 27, pp. 2512–2517, 2009.

[34] K. Hosono, Ed., *Mask writing time explosion and its effect on CD control in e-beam lithographty*, vol. 7748. SPIE, 2010.

[35] G. P. Patsis, N. Tsikrikas, D. Drygiannakis, and I. Raptis, "Simulation of shot noise effect on CD and LER of electron-beam lithography in 32 nm designs," *Microelectronic Engineering*, vol. 87, pp. 1575–1578, 2009.

[36] H. I. Smith, "A statistical-analysis of ultraviolet, X-ray, and charged-particle lithographies," *Journal of Vacuum Science & Technology B*, vol. 4, pp. 148–153, 1986.

[37] L. H. A. Leunissen, M. Ercken, and G. P. Patsis, "Determining the impact of statistical fluctuations on resist line edge roughness," *Microelectronic Engineering*, vol. 78-79, pp. 2–10, 2005.

[38] *A numerical model of electron beam shot noise.* Particle Accelerator Conference, 2003.

[39] A. Yamada and Y. Oae, "Electron beam exposure system capable of detecting failure of exposure," U.S. Patent 5,449,915, 1995.

[40] Y. Fujikura, "Electron beam lithography appratus including a beam blank device utilizing a reference comparator," U.S. Patent 4,937458, 1990.

[41] M. A. McCord and A. D. Brodie, "Method and apparatus for beam current fluctuation correction," U.S. Patent 7,091,486, 2006.

[42] M. Mankos, T. P. Chang, M. Y. Lee, and M. Lee, "Suppression of emission noise for microcolumn application in electron beam inspection," U.S. Patent 6,555,830, 2003.

[43] S. K. Islam, C. Durisety, R. Vijayaraghavan, B. J. Blalock, T. Grundman, L. R. Baylor, and W. L. Gardner, "All-inverter complementary metal oxide semiconductor based dose control circuit for using vertically aligned carbon nanofibers in maskless lithography," *Journal of Vacuum Science & Technology B*, vol. 24, pp. 1026–1029, 2006.

[44] L. R. Baylor, D. H. Lowndes, M. L. Simpson, C. E. Thomas, M. A. Guillorn, V. I. Merkulov, J. H. Whealton, E. D. Ellis, D. K. Hensley, and A. V. Melechko, "Digital electrostatic electron-beam array lithography," *Journal of Vacuum Science & Technology B*, vol. 20, pp. 2646–2650, 2002.

[45] C. S. A. Durisety, R. Vijayraghavan, L. Seshan, S. K. Islam, and B. J. Blalock, "Dose control circuit for digital electrostatic electron-beam array lithography," *Analog Integrated Circuit and Signal Processing*, vol. 48, pp. 143–150, 2006.

[46] T. Rahman, S. K. Islam, R. Vijayaraghavan, T. Gundman, S. A. Eliza, A. Hossain, B. Blalock, S. J. Randolph, L. R. Baylor, T. S. Bigelow, W. L. Gardner, M. N. Ericson, and J. A. Moore, "Integration of a dose control circuit with a vertically aligned nanofiber field emission device," *Journal of Vacuum Science & Technology B*, vol. 25, pp. 655–660, 2007.

[47] K. Wilder and C. F. Quate, "Scanning probe lithography using a cantilever with integrated transistor for on-chip control of the exposing current," *Journal of Vacuum Science & Technology B*, vol. 17, pp. 3256–3261, 1999.

[48] L. H. Veneklasen, "Field emission gun with noise compensation," U.S. Patent 4,337,422, 2003.

[49] D. N. Jamieson, C. Yang, T. Hopf, S. M. Hearne, C. I. Pakes, S. Prawer, M. Mitic, E. Gauja, S. E. Andresen, F. E. Hudson, A. S. Dzurak, and R. G. Clark, "Controlled shallow single-ion implantation in silicon using an active substrate for sub-20-KeV ion," *Applied Physics Letters*, vol. 86, p. 202101, 2005.

[50] J. Li, D. Stein, C. McMullan, D. Branton, M. J. Aziz, and J. A. Golovchenko, "Ion-beam sculpting at nanometer length scale," *Nature*, vol. 412, pp. 166–169, 2001.

[51] N. Patterson, D. P. Adams, V. C. Hodges, M. J. Vasile, J. R. Michael, and P. G. Kotula, "Controlled fabrication of nanopores using a direct focused ion beam approach with back face particle detection," *Nanotechnology*, vol. 19, p. 235304, 2008.

[52] D. M. Stein, C. J. Mcmullan, J. L. Li, and J. A. Golovchenko, "Feedback-controlled ion beam sculpting apparatus," *Review of Scientific Instruments*, vol. 75, pp. 900–905, 2004.

[53] J. Ferrera, V. V. Wong, S. Rishton, V. Boegli, E. H. Anderson, D. P. Kern, and H. I. Smith, "Spatial-phase-locked electron-beam lithography," *Journal of Vacuum Science & Technology B*, vol. 11, pp. 2342–2345, 1993.

[54] J. Goodberlet, J. Carter, and H. I. Smith, "Scintillating global-fiducial grid for electron-beam lithography," *Journal of Vacuum Science & Technology B*, vol. 16, pp. 3672–3675, 1998.

[55] J. T. Hastings, F. Zhang, J. G. Goodberlet, and H. I. Smith, "Improved pattern-placement accuracy in e-beam lithography via sparse-sample spatial-phase locking," *Microelectronic Engineering*, vol. 53, pp. 361–364, 2000.

[56] F. Zhang, H. I. Smith, and J. F. Dai, "Fabrication of high-secondary-electron-yield grids for spatial-phase-locked electron-beam lithography," *Journal of Vacuum Science & Technology B*, vol. 23, pp. 3061–3064, 2005.

[57] T. H. P. Chang, D. P. Kern, and M. A. McCord, "Electron optical performance of a scanning tunneling microscope controlled field emission microlens system," *Journal of Vacuum Science & Technology B*, vol. 7, pp. 1855–1861, 1989.

[58] T. H. P. Chang, M. G. R. Thomson, E. Kratschmer, H. S. Kim, M. L. Yu, K. Y. Lee, S. A. Rishton, B. W. Hussey, and S. Zolgharnain, "Electron-beam microcolumns for lithography and related application," *Journal of Vacuum Science & Technology B*, vol. 14, pp. 3774–3781, 1996.

[59] C. B. Samantaray and J. T. Hastings, “Self-assumbled monolayer fiducial grids for spatial-phase-locked electron-beam lithography,” *Journal of Vacuum Science & Technology B*, vol. 26, pp. 2351–2355, 2008.

[60] ——, “Amino-propyl-triethoxy-silane on aluminum fiducial grids for spatial-phase-locked electron-beam lithography,” *Journal of Vacuum Science & Technology B*, vol. 27, pp. 2558–2562, 2009.

[61] J. Ferrera, M. L. Schattenburg, and H. I. Smith, “Analysis of distortion in interferometric lithography,” *Journal of Vacuum Science & Technology B*, vol. 14, pp. 4009–4014, 1996.

[62] M. E. Walsh and H. I. Smith, “Method for reducing hyperbolic phase in interference lithography,” *Journal of Vacuum Science & Technology B*, vol. 19, pp. 2347–2352, 2001.

[63] C. G. Chen, “Beam alignment and image metrology for scanning beam interference lithography - fabricating gratings with nanometer phase accuracy,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[64] M. Schattenburg and P. N. Everett, *Method and system for interference lithgraphy utilizing phase-locked scanning beams*, us 6882477 b1 ed., Massachusetts Institute of Technology, Cambridge, MA, 2005.

[65] L. Wang, B. Terhalle, M. Hojeij, V. A. Guzenko, and Y. Ekinci, “High-resolution nanopatterning by achromatic spatial frequency multiplication with electroplated grating structures,” *Journal of Vacuum Science & Technology B*, vol. 30, pp. 031 603–031 608, 2012.

[66] H. H. Solak, C. David, J. Gobrecht, V. Golovkina, F. Cerrina, S. O. Kim, and P. F. Nealey, “Sub-50nm period patterns with EUV interference lithography,” *Microelectronic Engineering*, vol. 67-68, pp. 56–62, 2003.

[67] E. H. Anderson, K. Komatsu, and H. I. Smith, “Achromatic holographic lithography in the deep ultravoilet,” *Journal of Vacuum Science & Technology B*, vol. 6, pp. 216–218, 1988.

[68] A. Yen, E. H. Anderson, R. A. Ghanbari, M. L. Schattenburg, and H. I. Smith, “Achromatic holographic configuration for 100-nm-period lithography,” *Applied Optics*, vol. 31, pp. 4540–4545, 1992.

[69] T. A. Savas, S. N. Shah, M. L. Schattenburg, J. M. Carter, and H. I. Smith, “Achromatic interferometric lithography for 100-nm-period gratings and grids,” *Journal of Vacuum Science & Technology B*, vol. 13, pp. 2372–2375, 1995.

[70] T. A. Savas, M. L. Schattenburg, J. M. Carter, and H. I. Smith, “Large-area achromatic interferometric lithography for 100nm period gratings and grids,” *Journal of Vacuum Science & Technology B*, vol. 14, pp. 4167–4170, 1996.

[71] N. Corporation. (2012) Spatial filter. [Online]. Available: http://www.newport.com/Spatial-Filters/144910/1033/content.aspx

[72] M. E. Walsh, "On the design of lithographic interferometers and their application," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[73] D. Slepian, "Estimation of signal parameters in the presence of noise," *IRE Transaction on Information Theory*, vol. 3, pp. 68–89, 1957.

[74] D. C. Rife and G. A. Vincent, "Use of the discrete Fourer transform in the measurement of frequencies and levels of tones," *Bell System Technical Journal*, vol. 49, pp. 197–228, Feburary 1970.

[75] D. C. Rife and R. R. Boorstyn, "Single-tone parameter estimation from discrete-time observations," *IEEE Transaction on Information Theory*, vol. 20, pp. 591–598, 1974.

[76] L. C. Palmer, "Coarse frequency estimation using the discrete Fourier transform," *IEEE Transaction on Information Theory*, vol. 20, pp. 104–109, 1974.

[77] A. V. Krishnamurthy, R. V. Namepalli, and J. T. Hastings, "Subpixel alignment for scanning-beam lithography using one-dimensional, phase-based mark detection," *Journal of Vacuum Science & Technology B*, vol. 23, pp. 3037–3042, 2005.

[78] H. L. V. Trees, *Detection, Estimation, and Modulation Theory - Part 1 - Detection, Estimation, and Linear Modulation theory.* John Wiley & Sons, 2001.

[79] J. T. Hastings, M. H. Lim, J. G. Goodberlet, and H. I. Smith, "Optical waveguides with apodized sidewall gratings via spatial-phase-locked electron-beam lithography," *Journal of Vacuum Science & Technology B*, vol. 20, pp. 2753–2757, 2002.

[80] C. Moormann, J. Bolten, and H. Kurz, "Spatial phase-locked combination lithography for photonic crystal devices," *Microelectronic Engineering*, vol. 73-73, pp. 417–422, 2004.

[81] D. C. Rife and R. R. Boorstyn, "Multiple tone parameter estimation from discrete-time observations," *Bell System Technical Journal*, vol. 22, pp. 1389–1410, 1976.

[82] N. Inc, *XtremeDSP Development Kit-IV Reference Guide*, 3rd ed., Nallatech Inc, Camarillo, CA, July 2007.

[83] J. T. Hastings, "Nanometer-precision electron-beam lithography with applications in integrated optics," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[84] S. A. Maloney, “Dosage control for electron beam lithography,” Master’s thesis, University of Kentucky, Lexington, KY, 2010.

[85] M. A. Finlayson, “Development of a scintillating reference grid for spatial-phase-locked electron-beam lithography,” Master’s thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2001.

Vita

Author Name: Yugu Yang

Education:

Bachelor of Microelectronic Science and Technology
University of Electronic Science and Technology of China
May, 2005

Professional Positions:

Graduate Research Assistant
University of Kentucky
January, 2007 - December, 2011

Graduate Teaching Assistant
University of Kentucky
August, 2005 - December, 2006

Publications:

Y. Yang and J. T. Hastings, FPGA Implementation of Real-time Spatial-Phase Locking for Electron Beam Lithography, Journal of Vacuum Science & Technology B., 26, pp. 2316-2321 (2008).

Y. Yang and J. T. Hastings, Real-time Spatial Phase Locking for Vector-Scan Electron Beam Lithography, Journal of Vacuum Science & Technology B., 25, pp. 2072-2076 (2007).

Conferences/Presentations:

Y. Yang, S. Maloney and J. T. Hastings, Real-time Dose Control for Electron Beam Lithography, Presented at the International Conference on Electron, Ion, and Photon Beam Technology and Nanofabrication, Las Vegas, Nevada (2011).

Y. Yang and J. T. Hastings, FPGA Implementation of Real-time Spatial-Phase Locking for Electron Beam Lithography, Presented at the International Conference on Electron, Ion, and Photon Beam Technology and Nanofabrication, Portland, Oregon (2008).