

**Development of a Three-Dimensional Trap for
Single-Molecule Studies with a Four-Focus
Confocal Fluorescence Microscope**

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Kapila Deshappriya Dissanayaka Dissanayaka Mudiyanse
May 2019

ACKNOWLEDGEMENTS

I would like to first thank my committee for their valuable insight, guidance and encouragement given to me to accomplish my research activities. I would especially like to thank my advisor, Dr. Lloyd M. Davis. His knowledge, experience, resource management and patience have provided the best guidance for the success of my research. I would also like to thank Dr. Brian Canfield and Mr. Alexander Terekhov for technical support provided both in and out of the lab. I feel that I'm so lucky to have worked with such talented staff members.

I really appreciate all the support and resources provided to me by the UTSI. Special thanks goes to the Center for Laser Applications (CLA). CLA has provided the necessary resources to conduct my research in a timely manner. Also, thank you very much to all the technical staff at the CLA for the support given to me in my research.

Last but not least, thank you to my family and friends. You all have always been there with me to share my thoughts.

ABSTRACT

This dissertation presents the development of an instrument based on a confocal fluorescence microscope for feedback-driven trapping of a single molecule or nanoparticle in three-dimensions as it undergoes Brownian diffusion within an aqueous medium. Such trapping enables prolonged observation of a molecule while untethered and free from collisions with surfaces, which is needed to improve various studies, such as investigations of protein folding dynamics, molecular heterogeneities, and interactions. In the experiment, a dilute solution (~ 100 pM) of fluorescent nano-objects is inserted into a microfluidic device, which achieves trapping by control of electroosmotic flows in two crossed channels. The geometry, which is designed using COMSOL Multiphysics, funnels the flows to achieve sufficient electroosmotic speed to counteract Brownian diffusion while maintaining a 4:1 width-to-depth for wide-angle light collection by the microscope objective from the center of the crossing region. A fluorescence excitation volume centered at this point is defined by four overlapping focused laser beams, each with ~ 0.5 μm beam waist but with centers offset in a tetrahedral arrangement. The beams are derived from a mode-locked laser using a series of beam splitters with the pulses in each beam delayed to provide pulse-interleaved excitation at 304 MHz. Fluorescence is collected through a pinhole and split to two single-photon detectors, which provide signals for an FPGA (Field Programmable Gate Array) for time-gated counting into four channels synchronous with the pulses in each of the laser beams. The FPGA also bins the

counts and applies an algorithm to estimate the direction of the position offset of the nano-object and to adjust four voltages. These are applied at the four fluid inlets of the microfluidic cross-channel to electroosmotically drive the fluid to keep the nano-object at the midpoint of the four foci. Movies of camera imaging of trapped nano-objects were acquired. Results show trapping of 40 nm FluoSpheres for ~4 minutes, 20 nm FluoSpheres for ~25 seconds, and 5 nm protein molecules of Streptavidin-Alexa Fluor™ 647 for ~1.5 seconds. In addition, Maximum Likelihood Estimation of positions from binned photons was conducted for the FluoSphere experiments to estimate effective spring constants of the trap.

TABLE OF CONTENTS

Chapter One: Introduction.....	1
Chapter Two: Background	8
Electrokinetic Manipulations.....	8
Microfluidic Cross-Channel Design Basics.....	11
Confocal Microscopy with One-Photon or Two-Photon Excitation	15
Feedback Control for Trapping.....	19
Applying Electroosmosis to Counteract Brownian Diffusion	21
Chapter Three: Microfluidic Device for Trapping.....	24
Microfluidic Cross-Channel Fabrication Overview	24
COMSOL Modelling and Results	26
Device Fabrication.....	33
Material Preparation	33
Spin Coating of PDMS Films	34
Femtosecond (fs) Laser Machining.....	39
PDMS Bonding	44
Device Assembly	48
Chapter Four: Optical Setup, Data Acquisition and Feedback.....	50
One-photon, Four-focus Microscope	50
Data Acquisition and Feedback.....	55
Chapter Five: Methods.....	59
Time-Gated Photon Counting.....	59
Trapping Algorithm	61
Position Estimation.....	69
Chapter Six: Experimental Results and Analysis	71
Evidence of Trapping	71
Measured Electroosmotic Speeds.....	72
Trapping 40 nm FluoSpheres.....	75
Image Data of Trapping for over 4 Minutes.....	75
Photon Count Data for Trapping of ~1 Minute	78
Trapping 20 nm FluoSpheres.....	82
Image Data of Trapping of ~25 Seconds	82
Photon Count Data of Trapping for over 5 Seconds	85
Trapping Streptavidin-Alexa 647 Protein Molecules.....	91
Image Data of Trapping of ~1.5 Seconds	91
Photon Count Data of Trapping for over 0.5 Seconds	94
Analysis of Direction of Fluid Flow within the ROI	97
Intensity Variations along Cartesian Axes.....	98
3D Analysis of Flow Directions	103
Discussion on Effective Potential of Trap.....	113
Chapter Seven: Conclusions and Future Improvements.....	115
List of References	118
Appendices	124

Appendix A. Schematics of the Perspex Hood	125
Appendix B. Schematics of the FPGA Program	128
Appendix C. User-defined Verilog Modules in FPGA	134
“counts_to_voltages_6_2” Module	134
“darpa_dac_2” Module	140
“usb_descriptor” Module	152
Appendix D. MATLAB Code for Position Estimations Using MLE	164
Appendix E. MATLAB Code for 3D Analysis of Flow Directions	167
Appendix F. MATLAB Code to Test Trapping Modes	176
Vita	179

LIST OF FIGURES

Figure 1.1. Feedback-driven trapping flowchart.	3
Figure 2.1. Three configurations for 3D electrokinetic manipulation of particles. 10	
Figure 2.2. Schematic of center part of the cross-channel (not to scale).	12
Figure 2.3. 2D view of a cross-channel, designed in COMSOL (units are in μm).	13
Figure 2.4. Defining the trapping region.	20
Figure 3.1. Tapered cross-channel formed in PDMS films on glass coverslips. .	25
Figure 3.2. Two microchannel geometries for COMSOL analysis.	28
Figure 3.3. Cross-channel design for fabrication.	30
Figure 3.4. COMSOL electroosmotic speed analyses for the cross-channel.	32
Figure 3.5. Curve fitting PDMS Thickness vs. Angular Speed to estimate c in Equation 15.	36
Figure 3.6. Extrapolation of PDMS Thickness vs. Angular Speed.	37
Figure 3.7. Laser machining setup for PDMS samples.	40
Figure 3.8. Optical micrograph (60x) of a fs laser machined microchannel.	43
Figure 3.9. Optical micrograph (60x) of an assembled cross-channel device.	46
Figure 3.10. Photograph of an assembled cross-channel device showed in Figure 3.9.	46
Figure 3.11. Ridge effect in spin coated PDMS films.	47
Figure 3.12. Cross-channel device (not clearly visible) interfaced with a Perspex hood.	47
Figure 3.13. Cross-channel device interfaced with Perspex hood (not to scale). 49	
Figure 4.1. Schematic of four-foci arranged in a tetrahedral pattern.	50
Figure 4.2. CCD Camera images of the four foci.	51
Figure 4.3. Double Michelson interferometer layout for four foci generation at 304 MHz [61].	53
Figure 4.4. Improved optical setup for four foci generation at 304 MHz.	54
Figure 4.5. Functional diagram of the trapping experiment.	56
Figure 4.6. Photograph of the experimental setup.	58
Figure 5.1. Schematic describing pulse-interleaved excitation of fluorescence. .	59
Figure 5.2. Flowchart of the FPGA board operation.	65
Figure 5.3. Timing diagram of voltage output by one DAC of the LTC2704.	68
Figure 6.1. CCD camera images to show evidence of trapping of a fluorescent particle.	71
Figure 6.2. Schematic of camera images to estimate electroosmotic speeds (not to scale).	74
Figure 6.3. Image data for trapping a 40 nm FluoSphere in distilled water for >4 minutes.	77
Figure 6.4. Trapping a 40 nm FluoSphere in distilled water.	79
Figure 6.5. MLE of positions vs. time for the data in Figure 6.4.	80
Figure 6.6. 3D plot of MLE positions of the 40 nm FluoSphere trapped for ~57 seconds.	81

Figure 6.7. Image data for trapping a 20 nm FluoSphere in distilled water for ~25 seconds.	84
Figure 6.8. Trapping a 20 nm FluoSphere in distilled water (I).	86
Figure 6.9. Trapping a 20 nm FluoSphere in distilled water (II).	87
Figure 6.10. MLE of positions vs. Time for the data in Figure 6.9.	89
Figure 6.11. 3D plot of MLE positions of a trapped 20 nm FluoSphere.....	90
Figure 6.12. Image data for trapping a Streptavidin-Alexa 647 molecule in distilled water for ~1.5 seconds.....	93
Figure 6.13. Trapping a single Streptavidin-Alexa 647 molecule in distilled water for ~0.5 seconds.	95
Figure 6.14. Trapping a single Streptavidin-Alexa 647 molecule in distilled water for ~1.5 seconds.	96
Figure 6.15. Intensity variations along the axes with $\Delta x = \Delta y = 760$ nm.	99
Figure 6.16. Intensity variations along the axes with $\Delta x = \Delta y = 686$ nm.	101
Figure 6.17 Intensity variations along the axes with $\Delta x = \Delta y = 660$ nm.	102
Figure 6.18. Flow directions in selected X-Y planes ($\Delta x = \Delta y = 760$ nm).....	104
Figure 6.19. Flow directions in selected X-Z planes ($\Delta x = \Delta y = 760$ nm).....	106
Figure 6.20. Flow directions in selected Y-Z planes ($\Delta x = \Delta y = 760$ nm).....	107
Figure 6.21. Testing for two-potential-well mode with zero diffusivity particle ($\Delta x = \Delta y = 760$ nm).	108
Figure 6.22. Flow directions in selected X-Y planes ($\Delta x = \Delta y = 660$ nm).	110
Figure 6.23. Flow directions in selected X-Z planes ($\Delta x = \Delta y = 660$ nm).....	111
Figure 6.24. Flow directions in selected Y-Z planes ($\Delta x = \Delta y = 660$ nm).....	112
Figure 6.25. Testing for one-potential-well mode with zero diffusivity particle ($\Delta x = \Delta y = 660$ nm).	113
Figure A.1. Schematic of the top of the Perspex hood.	126
Figure A.2. Schematic of the bottom of the Perspex hood.	127
Figure B.1. FPGA interfaced with SPAD 0,1 and 76 MHz external synchronous clock.....	129
Figure B.2. Photon detection circuitry and main triggering counters.	130
Figure B.3. Four time gates and four photon counters with buffers.	131
Figure B.4. Photon counts to voltage conversion.....	132
Figure B.5. Transfer photon counts data a(31...0) via USB.	133

LIST OF TABLES

Table 3.1. Properties and parameter values for COMSOL simulation.	29
Table 3.2. Key parameters used for fs laser machining of PDMS.	42
Table 5.1. Expected fluid flow directions.	62
Table 5.2. Flow directions vs. voltages V_1 , V_2 , V_3 and V_4	63
Table 6.1. Andor™ iXon 897 EM-CCD camera and SPAD settings.	72
Table 6.2. Experimentally measured and estimated electroosmotic speeds.	75

LIST OF ATTACHMENTS

Video 1. Trapping 40 nm FluoSphere.....Trapping_40nm_Fluosphere.mp4
Video 2. Trapping 20 nm FluoSphere.....Trapping_20nm_Fluosphere.mp4
Video 3. Trapping Streptavidin-Alexa647.....Trapping_Streptavidin-Alexa647.mp4

CHAPTER ONE:

INTRODUCTION

Prolonged observations of single molecules enable the understanding of chemical, biological and biophysical aspects [1, 2] on a much more detailed level than ensemble observations. Characterization of the dynamical properties of a single molecule, such as diffusion, visualization of protein folding and unfolding states in thermal equilibrium, probing the molecular heterogeneity of a mixture such as analyzing the protein composition of egg white [3] or analyzing the compositions of a cancer cell, determination of binding efficiency of two molecules such as proteins and ligands, and even visualization of the dynamics of cellular machines, are some of the instances that could be performed under prolonged observations of a freely diffusing single molecule [4-6]. However, the Brownian diffusion that is inherent for a single-molecule in an aqueous medium limits the exposure time available for making such measurements. Immobilization of a molecule by sticking or tethering it to a surface can obviously eliminate diffusion. However, it also introduces perturbations from the local environment [6] and barriers to the observation and measurement of the dynamical nature of the molecule, such as diffusion coefficients and conformational changes. Although the well-known trapping mechanisms such as far-field optical, magnetic, acoustic tweezers, and electrophoretic traps do not alter the immediate surroundings of the molecule, there are other factors to be considered in practice. For instance, the high intensities of light needed to trap molecular-level objects in far-field

optical tweezers can generate joule heating and photo damage the molecule [7]. In magnetic tweezers, a target molecule (such as a DNA strand) has to be attached to a micron-sized magnetic bead and the attachment of the bead causes perturbations in the molecular surroundings. Acoustic tweezers, which are suggested as favorable for biological samples [8], have not yet been able to trap nanoscale particles [9-11] at the low pressure nodes of the acoustic waves. Furthermore, this mechanism is yet to be developed for molecular-level trapping that incorporates GHz-level frequencies [9]. In electrophoresis-driven trapping, the target particles themselves have to be charged particles, such as cations. Thus the trapping of neutral, molecular-level particles is not possible with this method.

In contrast, electroosmotic manipulation coupled with the use of fluorescence detection can be very promising in trapping nano-objects dissolved in aqueous media for prolonged observations. Electroosmosis, in general, is the motion of liquid in response to a voltage applied along a capillary. Fast switching of fluid flow velocity and a flat fluid velocity profile across the channel are the key advantages of electroosmotic flow over a pressure-driven flow of fluid. With the incorporation of feedback mechanisms, as it is explained in Figure 1.1, the implementation of electroosmotic traps becomes possible as the fast switching of the liquid flow direction can effectively counteract the Brownian diffusion of a nano-object dissolved in the liquid.

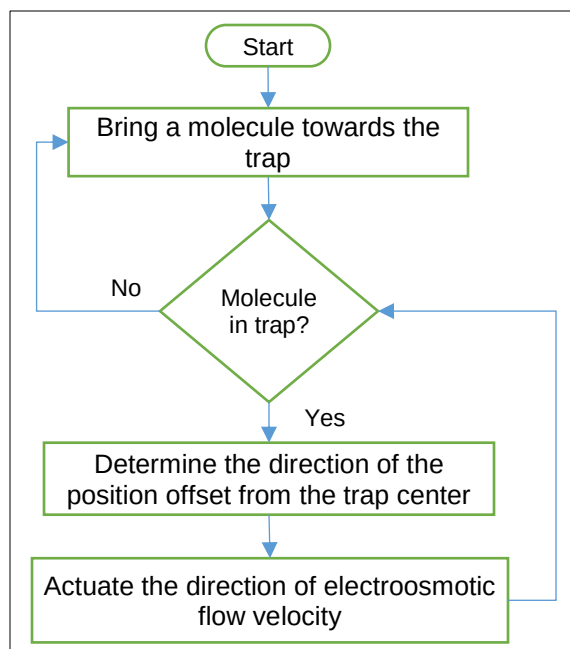


Figure 1.1. Feedback-driven trapping flowchart.

This principle, the feedback-driven trapping of a single fluorescent nano-object via electroosmosis (and more generally, electrokinesis), has been previously implemented in one-dimensional (1D) nanochannels [12-16] and in two-dimensional (2D) planar microfluidic devices known as Anti-Brownian Electrokinetic (ABEL) traps [17-22] for single-molecule and single-nanoparticle studies. Although nano-object manipulation can be implemented successfully in one or two dimensions by constraining the motion in the other dimensions, object sticking and rapid collisions with confining walls occur in 1D or 2D traps, which may perturb the trapped object. On the other hand, studies of completely unperturbed single molecules or nanoparticles could be performed in a three-dimensional (3D) ABEL trap by use of instrumentation such as that developed in

this research. To date, very few experimental results have been reported in this regard.

In the stream of development of 3D ABEL traps, a 3D electrokinetic trapping mechanism was first introduced by Jason K. King [23, 24]. He used astigmatic imaging and real-time processing of images acquired with a CCD (Charge Coupled Device) camera for the 3D position determination of a fluorescent nanoparticle. Repositioning of the nanoparticle within a 3D microvolume was controlled through adjustment of voltages, supplied by a National Instruments FPGA (Field Programmable Gate Array) board, applied across four electrodes arranged in a tetrahedral configuration. Trapping in 3D was demonstrated for fluorescently labeled nanoparticles ~40 nm in diameter dissolved in a high viscosity glycerol solution to slow diffusion. The main limitation of the setup was the slow speed for 3D position determination, which was limited by the CCD camera frame rate.

Soon after that work, another experiment for nanoparticle tracking in 3D was carried out by James A. Germann [25, 26]. He used alternately pulsed beams from four diode lasers focused at four spatially offset positions that were located at the vertices of a tetrahedron. Photon counts generated by a fluorescent nanoparticle due to the laser pulses at each vertex were used as the data for 3D position determination. The sample was mounted on a 3D piezo stage that was actuated to reposition the nanoparticle to the center of the tetrahedron following each position determination. Similar experiments were also

conducted by Wells et al. [27] and Perillo et al. [28] using a different strategy for 3D position determination based on collecting fluorescence through four spatially offset fiber optics, which acted as spatial filters. However, the 3D piezo stages used in such experiments have a limited range of motion, which limits the time for which the nanoparticle may be tracked.

In a subsequent recent report, a 3D ABEL trap was constructed using a two-layer geometry in polydimethylsiloxane (PDMS), and trapping of a fluorescent nanodiamond (FND)—with Nitrogen-vacancy (NV) defects—was demonstrated by using real-time image processing for axial position determination and laser beam scanning for transverse position determination [29]. In these experiments it was shown that a trapped FND with NV defects could be used as a nanoscale quantum sensor for external magnetic fields, thus a new direction for the use of 3D traps as sensors has emerged.

Although the 3D feedback-driven experiments highlighted above are remarkable achievements, each has its limitations. Foremost, the overall time lag between fluorescence signal detection and the actuation of position or fluid flow controllers is a notable issue. In references [23, 24], this time lag is ~40 ms, while in reference [29] it is ~15 ms, in each case due to the relatively slow and computationally intensive image processing. In references [25, 26], although the position determination of a particle could be performed much faster depending on the photon count rate, the main issue is the slow update rate of the piezo controller, which is limited due to the mechanical inertia of the stage. The fastest

possible update of the 3D position was in the 0.5 kHz range (minimum time between displacements ~ 1.86 ms).

In this dissertation research, a 3D ABEL trap is developed with a fast (low latency) feedback cycle for analyzing fluorescence signals and applying corrective motion to counteract diffusion every 13.5 μ s. The fast response was achieved by combining rapid 3D position control using four voltages, similar to that used by Jason King together with rapid 3D position determination using four offset laser foci, similar to that used by James Germann. However, the 3D position control uses electroosmosis in microfluidic channels rather than electrophoresis in an open volume; and the 3D position determination uses interleaved pulses from a single mode-locked laser and time-gated photon counting (TGPC) implemented in an FPGA circuit, rather than slower alternation of continuous beams from four laser diodes controlled by a counter-timer card. In the experimental setup, the four laser beams are obtained from the mode-locked laser using beam splitters and the beams are directed into the microscope objective so that they are partially overlapping with the centers of the beam foci arranged in a tetrahedron. The excitation volume defined by the foci of the four overlapping beams is imaged into a pinhole, which spatially filters (blocks) light from outside the volume, such as light scattered from the walls of the microfluidic device. Fluorescence photons generated in the excitation volume pass through the pinhole to single-photon avalanche diodes (SPADs) and TGPC is performed in an FPGA to give the input data for a feedback control algorithm for trapping. In

the feedback control, the FPGA applies voltages to the four electrodes of a microfluidic cross-channel to move a fluorescent nano-object via electroosmosis towards the midpoint of the four laser foci and thereby counteract Brownian diffusion. The optics are adjusted so that the midpoint coincides with the center of the cross-channel. The volume enclosed by the tetrahedron is considered as the trapping Region of Interest (ROI).

Also, before the pinhole, the collected fluorescence is split at a 50% beam splitter so that the fluorescent particle/molecule may be simultaneously imaged by a EM-CCD (Electron Multiplying Charge Coupled Device) camera, while photons passing the pinhole are detected by the SPADs to provide input for the FPGA. The time-gated photon counts are binned and used for the trapping control and are also recorded for post-process analysis. In particular, a Maximum Likelihood Estimation (MLE) algorithm is developed for estimating the 3D position of the particle as a function of time from the binned photons. Effective spring constants for the trap are calculated from the fluctuations of the estimated particle position under the approximation that the effective potential is harmonic.

CHAPTER TWO:

BACKGROUND

Electrokinetic Manipulations

There are many methods of manipulating single nanoparticles in aqueous media. From amongst them, electrokinetic manipulations—electrophoretic, dielectrophoretic, and electroosmotic—in microfluidic channels are preferable for feedback-driven trapping, mainly due to fast counteraction of Brownian diffusion of particles made possible by externally applied low-level direct current (DC) electric fields ($\leq \pm 10$ V). Under electrophoretic control, charged particles tend to move in a liquid medium in response to an externally applied electric field, and for a DC electric field the particles can attain a steady-state velocity that can be estimated by the Huckel model [30-32]. When it comes to dielectrophoresis, particles that exhibit a different polarizability compared to the liquid medium in which they are suspended, can only show movement for a non-uniform applied electric field [31, 33]. The governing equation for dielectrophoresis is given by equation 10 in [33]. In the case of electroosmosis, an electric double layer (EDL) facilitates the motion of a bulk fluid responding to an applied electric field. The EDL exists at the surface-fluid interface of a charged body (e.g., charged surface-fluid interface of a microchannel), and it creates the tightly bound first layer of ions attracted from the surrounding fluid, known as the Stern layer. Beyond the Stern layer, there will be a second, loosely bound layer of ions in solution known

as the diffuse layer, created by the existing residual potential from the charged body. In a microchannel, the diffuse layer is forced to move along the Stern layer in response to an applied electric potential along the channel, dragging the main fluid body with it and giving rise to a plug flow of the fluid in the microchannel [34, 35].

In the special case of manipulating a neutral nanoparticle, to which this dissertation is addressed, electroosmotic manipulation is preferred. In fact, under electroosmotic manipulation the motion of the nanoparticle is independent of its size, because the driving force acts to move the entire fluid medium rather than only the nanoparticle.

Irrespective of the type of electrokinetic manipulation, a particle in a 1D or 2D ABEL trap must be confined in a 1D nanochannel or 2D planar microfluidic channel and since it still undergoes Brownian diffusion it is subject to high collision rates with the walls of the channel, which introduce perturbations to the trapped particle. Also, other particles that have stuck to the walls lead to background fluorescence. Hence, 3D ABEL traps, for which the walls of the microfluidic device are far from the region of trapping, are preferable for studies of freely diffusing single nanoparticles or single molecules.

There are three configurations reported in the literature to date (see Figure 2.1) that use 3D electrokinetic manipulation of particles. The first one (Figure 2.1(a)) encompasses a volume controlled by four electrodes in a tetrahedral arrangement [23]. The second one (Figure 2.1(b)) consists of two

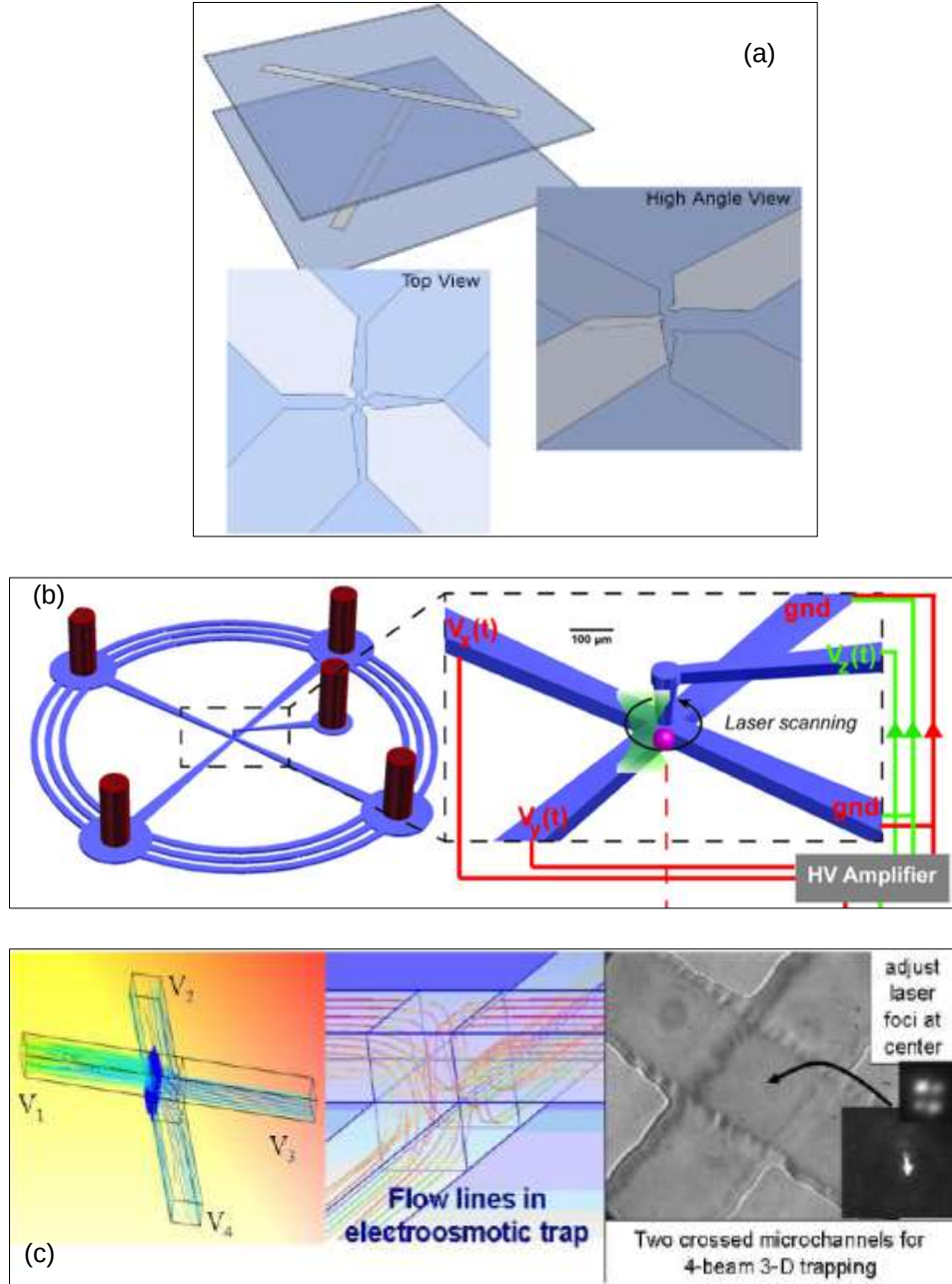


Figure 2.1. Three configurations for 3D electrokinetic manipulation of particles.

(a) Four electrodes in a tetrahedral arrangement [23]. (b) Two cross-channels in one plane and a third perpendicular channel that connects to the center where the planar channels intersect, and each end of a channel is connected to an electrode [29]. (c) Cross-channels stacked over one another and connected to two pairs of electrodes [36].

cross-channels in one plane and a third, perpendicular channel that connects to the center where the planar channels intersect, and each end of a channel is connected to an electrode to exert electrokinetic trapping [29]. The third one (Figure 2.1(c)) uses two microfluidic cross-channels stacked over one another and connected to two pairs of electrodes [36]. From amongst them, the theory, design, and construction of the third type mentioned—a 3D microfluidic cross-channel—which has been proposed in the literature but not yet implemented, is discussed in this dissertation.

Microfluidic Cross-Channel Design Basics

In designing a cross-channel, it is essential to consider two key factors: wide-angle fluorescence collection capability, and the ability to attain sufficiently fast electroosmotic speeds to counteract Brownian diffusion of molecules/nanoparticles.

In addressing wide-angle fluorescence collection, note that the fluorescence collection is limited by the numerical aperture of the microscope objective, NA . The numerical aperture defines the half-angle α that must be clear for light collection according to

$$\alpha = \arcsin(NA/n) \dots\dots\dots (1)$$

where n is the refractive index of the immersion medium. Figure 2.2 gives a schematic of the configuration for fluorescence collection from a cross-channel device.

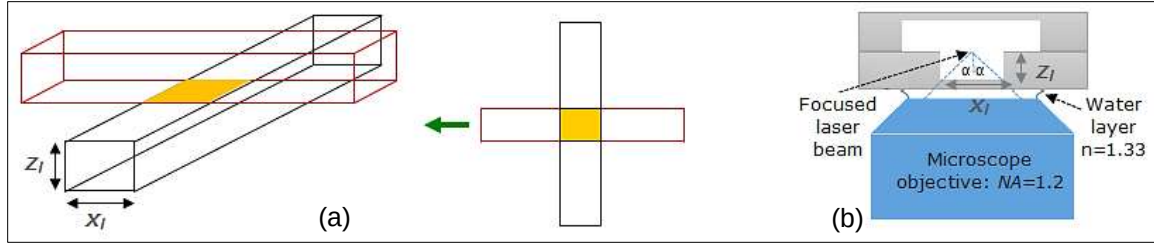


Figure 2.2. Schematic of center part of the cross-channel (not to scale).

(a) 3D view (left) and top view (right) of the center part of the cross-channel. (b) Side view of the channels and microscope objective.

For a water immersion objective with $NA = 1.2$ and medium of with $n = 1.33$, the-half angle α is about 64° and hence the aspect ratio of the rectangular channel cross-section must be at least $X_l : Z_l = 4:1$ to allow the microscope objective to collect the maximum fluorescence.

In addressing the ability to achieve fast electroosmotic speeds in cross-channels, computer simulations were carried out using COMSOL Multiphysics®. The channels were cut in polydimethylsiloxane (PDMS) and the dimensions of individual channels were changed during these simulations while maintaining the $X_l : Z_l = 4:1$ ratio. A 2D top view of a crossed channel microfluidic design is shown in Figure 2.3.

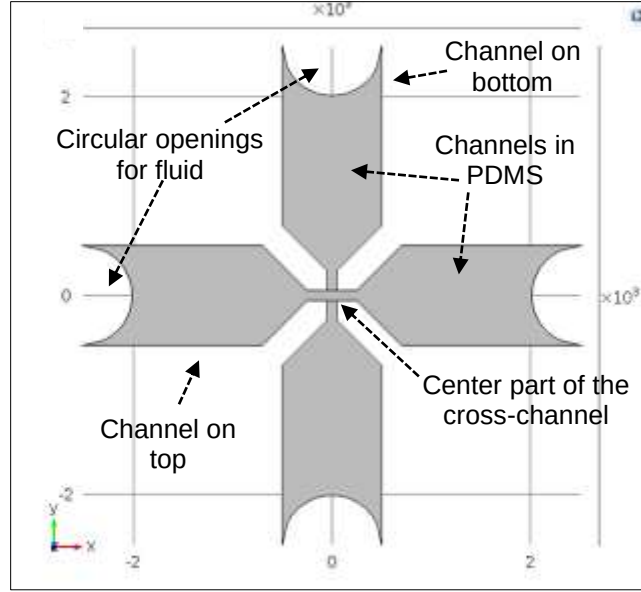


Figure 2.3. 2D view of a cross-channel, designed in COMSOL (units are in μm).

Note that only the boundary of the inner half of the circular openings were taken into account for the COMSOL analysis since that is where the electroosmotic fluid flow of a channel begins for an applied voltage along the channel.

In the COMSOL analysis, the main assumptions made were that the fluid is incompressible and works in the limit of laminar flow. Under these assumptions, the software numerically solves the Navier-Stokes equations from fluid dynamics together with the continuity equation. The software also numerically solves for the electric field in the cross-channel. The fluid dynamics equation to be solved then becomes the following in tensorial form [37]:

$$-\nabla \cdot p \mathbf{I} + \nabla \cdot (\mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) = 0 \quad \dots\dots\dots (2)$$

where $\nabla \cdot$ is the divergence operator, p is the fluid pressure, $\mathbf{I}$ is the identity tensor, μ is the fluid's dynamic viscosity, ∇ is the gradient operator, $\mathbf{u}$ is the fluid

velocity and T represents the transpose of the matrix. The equations to be solved for the electric field are:

$$\nabla \cdot \mathbf{J} = Q_i \dots\dots\dots (3)$$

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}_e \dots\dots\dots (4)$$

$$\mathbf{E} = -\nabla V \dots\dots\dots (5)$$

where $\mathbf{J}$, Q_i , σ , $\mathbf{E}$, $\mathbf{J}_e$, and V are the current density vector, free electrostatic charges, conductivity, electric field vector, free current density vector, and the voltage distribution in the micro-channel, respectively.

After solving for the electric field $\mathbf{E}$ in the cross-channel, COMSOL solves for the electroosmotic flow speed v_{eo} , according to the Helmholtz-Smoluchowski equation [30]:

$$v_{eo} = \epsilon \zeta E_t / \mu \dots\dots\dots (6)$$

where ϵ is the electric permittivity of the fluid, ζ is the zeta potential (see [34] for details) of the liquid-solid interface, and E_t is the electric field tangential to the surfaces of the microchannel [38]. The zeta potential ζ depends on the material from which the channel is fabricated and the ionic strength of the solution. For PDMS in the simulation ζ was taken as -70 mV [39]. This value of ζ is considered as an average value for glass and PDMS and ζ could take a different value in the experiment.

Confocal Microscopy with One-Photon or Two-Photon Excitation

In a confocal microscope, a pinhole is placed in the fluorescence collection path. The pinhole acts as a spatial filter, allowing most fluorescence photons that come from the center of the focal plane of the exciting laser beams to pass and rejecting most photons from outside the focal plane, thereby limiting the depth from which light is collected to provide sectioning capability. The size of the pinhole determines the level of discrimination, with a smaller hole sampling a shallower depth.

However, to maintain moderate throughput efficiency for fluorescence, the diameter of the pinhole must be at least that of the Airy disk that would be obtained from diffraction-limited imaging of fluorescence from a point in the focal plane, $M \times 1.22\lambda/NA$, where M and NA are the magnification and numerical aperture, and λ is the fluorescence wavelength. In the experiment, $M = 83.3$, $NA = 1.2$ and $\lambda \approx 0.7 \mu\text{m}$, giving a minimum required pinhole diameter of $60 \mu\text{m}$. For efficient throughput of fluorescence from an extended region of interest, the pinhole should be larger than the size of the image of that region. For the four laser foci with $1/e^2$ diameters of $\sim 1 \mu\text{m}$ offset by $\sim \pm 0.5 \mu\text{m}$, the image size is $\approx 83.3 \times 1.5 \mu\text{m} = 125 \mu\text{m}$, or about $\sqrt{125^2 + 60^2} \approx 140 \mu\text{m}$ when diffraction is considered. In the trapping experiments, a pinhole of $400 \mu\text{m}$ diameter was chosen to ensure high collection efficiency of fluorescence from throughout the $\sim \pm 1.6 \mu\text{m}$ depth of the excitation volume. In this case, the sectioning action of the

spatial filter primarily serves to block light scattered from the top and bottom walls of the microfluidic device, which are $\sim \pm 25 \mu\text{m}$ in depth from the middle of the ROI.

The initial trapping experiments in this research utilized a femtosecond (fs) laser providing 76 MHz pulses with $\sim 100 \text{ mW}$ average power at 800 nm excitation wavelength for two-photon excitation (2PE) of single Rhodamine-B dye molecules. Single-molecule detection of single Rhodamine-B in aqueous solution using 2PE has been previously demonstrated in [40]. The main motivation for using 2PE rather than one-photon excitation (1PE) for the trapping experiment was to utilize the improved sectioning capabilities of 2PE for the axial z-position determination of the trapped particle. However, the experiment had to be rebuilt to use 1PE with a different laser system due to failure of the pump laser for the fs laser system. Nevertheless, a comparison of 1PE and 2PE is given below as it is useful for ascertaining possible future improvements in the operation of the 3D trap.

In 1PE, one photon is absorbed by a molecule, which gets excited to an upper electronic energy state and then relaxes back to the ground state by emitting a photon in a short period of time (i.e., in nanoseconds) [41]. The emitted photon has less energy since some of the absorbed energy gets dissipated due to vibrational and/or rotational modes and the resultant shift between the excitation and the emission spectra is known as the Stokes shift. It is easy to isolate and study the fluorescence for comparatively large Stokes

shifts, but separating fluorescence and excitation can be challenging when the Stokes shift is small. In 2PE, two photons are absorbed simultaneously to excite the molecule to the first excited state and the molecule then relaxes to its ground state by emitting a photon with the energy equal to nearly the twice the energy of each of the two excitation photons. Hence the fluorescence is well separated from the excitation wavelength. According to the non-linear optical description, the 2PE process can be expressed as a third-order nonlinear optical phenomenon (governed by the third-order susceptibility, $\chi^{(3)}$) that is also proportional to the square of the intensity (I^2), while 1PE is a first-order, or linear, process (governed by the first-order susceptibility, $\chi^{(1)}$) and linearly proportional to the intensity (I) [42, 43].

To understand the differences in the sectioning capabilities of 1PE and 2PE, consider the intensity profile $I(x, y, z)$ of the laser beam focus approximated in the Gaussian-Lorentzian model [44]. In this approximation, the laser focus has a Gaussian intensity distribution in the focal plane (X-Y plane) and a Lorentzian distribution along the optical axis (Z-axis). In order to compare the dimensions of the excitation volumes for 1PE and 2PE, consider the following equations (Equation 7 and Equation 8) that represent the intensity profiles for 1PE and 2PE, respectively.

$$I(x,y,z) = \frac{I_0 w_0^2}{w^2(z)} \exp \left[\frac{-2 \left\{ (x-x_0)^2 + (y-y_0)^2 \right\}}{w^2(z)} \right] \dots\dots\dots (7)$$

$$I^2(x,y,z) = \left(\frac{I_0 w_0^2}{w^2(z)} \right)^2 \exp \left[\frac{-2 \{ (x-x_0)^2 + (y-y_0)^2 \}}{(w(z)/\sqrt{2})^2} \right] \dots\dots\dots (8)$$

where

$$w^2(z) = w_0^2 \left[1 + \left(\frac{z-z_0}{z_r} \right)^2 \right] \dots\dots\dots (9)$$

I_0 is the peak incident intensity, w_0 is the beam waist, and z_r is the Rayleigh range. The letters x , y , z represent the coordinates of the X -, Y -, Z -axes respectively. Also consider that for a fixed numerical aperture the waist of a focused laser beam is proportional to the wavelength, so $w_0^{1PE} \propto \lambda$. If the excitation wavelength λ were to remain unchanged, then w_0^{2PE} would be reduced by a factor of $1/\sqrt{2}$ compared to 1PE ($w_0^{2PE} = (1/\sqrt{2}) w_0^{1PE}$), whereas if λ doubled to keep the net excitation energy the same as that for 1PE, then w_0^{2PE} would be increased by a factor of $\sqrt{2}$ compared to 1PE ($w_0^{2PE} = \sqrt{2} w_0^{1PE}$) [45]. Also, considering the intensity variation along the Z -axis, it is possible to evaluate z_a for 1PE and 2PE where the intensity falls to $1/e^2$ times the peak value. The peak intensity falls to $1/e^2$ in 1PE (see Equation 10) when it satisfies $(1+(z/z_r)^2) = e^2$ leading to $z_a^{1PE} = z_r^{1PE} (\sqrt{e^2-1}) \approx 2.53 z_r^{1PE}$ whereas for 2PE (see Equation 11) this becomes $(1+(z/z_r)^2)^2 = e^2$ leading to $z_a^{2PE} = z_r^{1PE} (\sqrt{e-1}) \approx 1.31 z_r^{1PE}$, for

constant λ . If λ is doubled for 1PE (i.e., $\lambda' = 2\lambda$), then $(z_r^{1PE})_{2\lambda} = 2(z_r^{1PE})_\lambda$ (see

Equation 12) and leads to $(z_a^{2PE})_{2\lambda} = (z_r^{1PE})_{2\lambda} \cdot (\sqrt{e-1}) \approx 2 \times 1.31 (z_r^{1PE})_\lambda$.

$$I(0,0,z) = \frac{I(0,0,0)}{(1+(z/z_r)^2)} \dots\dots\dots (10)$$

$$I^2(0,0,z) = \frac{I^2(0,0,0)}{(1+(z/z_r)^2)^2} \dots\dots\dots (11)$$

$$(z_r^{1PE})_{2\lambda} = \left(\pi (w_0^{1PE})_{2\lambda}^2 / 2\lambda \right) = \left(\pi (2w_0^{1PE})_\lambda^2 / 2\lambda \right) = 2(z_r^{1PE})_\lambda \dots\dots\dots (12)$$

Hence we expect a larger radius of fluorescence excitation in 2PE (by a factor of $\sqrt{2}$), if the wavelength has been increased in 2PE to keep the same excitation energy as that was for 1PE. Also, we expect that the depth of excitation is slightly larger (by a factor of $2\sqrt{(e-1)/(e^2-1)} = 1.04$). It may be concluded that the depth/width ratio for 2PE is smaller than that for 1PE by a factor of $1.04/\sqrt{2} = 0.74$ and thus 2PE could be useful in defining 3D excitation volumes with relatively higher axial resolution.

Feedback Control for Trapping

The experiment for nanoparticle or molecule trapping relies on a feedback algorithm, which uses information in the detected photons to determine the direction that the particle should be moved in the microfluidic cross-channel to reposition it towards the center of the 3D trapping region. The voltages that are used to control the particle repositioning are applied to the four electrodes, which

are at the ends of the two crossed microchannels. The direction of motion is determined from the numbers of photons collected in four time bins, which correspond to the times of the laser pulses in the four laser beams. In the experiment, the feedback algorithm is executed by an FPGA module programmed using Altera® Quartus-II software [46-48]. A comprehensive explanation on the implementation of the feedback algorithm is given in Chapter Five.

A simplified understanding of the trapping algorithm is as follows: As shown in Figure 2.4, the 3D trapping region is divided into eight octants, defined as $(X+Y+Z+)$, $(X+Y+Z-)$, $(X+Y-Z+)$, $(X+Y-Z-)$, $(X-Y+Z+)$, $(X-Y+Z-)$, $(X-Y-Z+)$, and $(X-Y-Z-)$ regions. Four laser beams are focused to points centered on the opposing vertices of four of the octants, shown as Focus1,

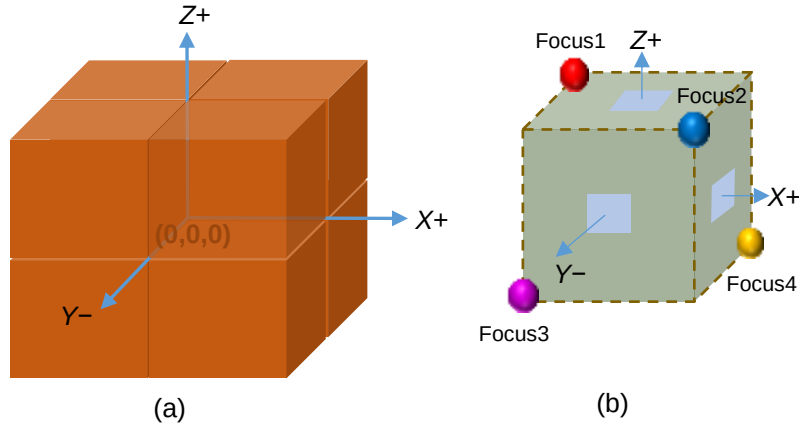


Figure 2.4. Defining the trapping region.

(a) Eight octants, defined by $(X+Y+Z+)$, $(X+Y+Z-)$, $(X+Y-Z+)$, $(X+Y-Z-)$, $(X-Y+Z+)$, $(X-Y+Z-)$, $(X-Y-Z+)$, and $(X-Y-Z-)$. (b) Centers of four laser foci forming the vertices of a tetrahedron centered at $(0,0,0)$.

Focus2, Focus3, and Focus4 in the figure. Pulse-interleaved excitation in the four beams and time-gated photon detection are used to collect fluorescence photons into four bins. When a particle diffuses into the trapping region, the numbers of photons in the four bins are used to estimate which octant the particle is in and hence to select the appropriate voltages needed to move the particle towards the origin. The implementation of the feedback algorithm is discussed in Chapter Five and the possibilities of the particle being on the boundaries between octants is also considered.

Applying Electroosmosis to Counteract Brownian Diffusion

A particle in a fluid undergoes Brownian diffusion with a diffusion coefficient that depend on its size, as well as the temperature and viscosity of the medium that it is dissolved in. The diffusion coefficient can be estimated by the Stokes-Einstein relation [49] shown in Equation 13:

$$D = \frac{k_B T}{6\pi\mu r} \dots\dots\dots (13)$$

where D is the diffusion coefficient of the particle, k_B is the Boltzmann constant, T is the absolute temperature in kelvin (K), μ is the viscosity of the suspending fluid, and r is the radius of the particle.

For a spherical particle having a radius of 20 nm, the diffusion coefficient in distilled water can be calculated using Equation 13 with $T = 293$ K (assuming a standard room temperature or 20 °C). At this temperature the viscosity of water

is about $8.9 \times 10^{-4} \text{ Pa}\cdot\text{s}$ [50] and the calculated diffusion coefficient becomes $12.1 \mu\text{m}^2\cdot\text{s}^{-1}$. The calculated diffusion coefficient may then be used to estimate the root-mean square distance σ_x that the particle moves in a given time interval t , accordingly to Equation 14 [51]:

$$\sigma_x = \sqrt{2Dt} \dots\dots\dots (14)$$

For trapping with a feedback cycle time of $t = 10 \mu\text{s}$ between corrections, the estimated value for σ_x (from Equation 14) becomes about 16 nm. We can then use these values of t and σ_x to estimate the average electroosmotic fluid speed (i.e., σ_x/t) needed to counteract Brownian diffusion in this interval, which gives about $1.6 \text{ mm}\cdot\text{s}^{-1}$.

The main purpose of the above calculation is to determine the electroosmotic speed needed for effective trapping with a given time interval for feedback and thereby set an approximate target for the flow velocity analysis in COMSOL. In the COMSOL simulations the dimensions of the cross-channel are altered until a satisfactory electroosmotic speed is achieved, while at the same time, the dimensions must be compatible with the optics and feasible for fabrication. In particular, the selection of dimensions of the cross-channel is limited by the physical limitations that would be encountered during laser machining and fabrication of the microfluidic device, which are discussed in Chapter Three. On the other hand, it should be possible to trap a particle of the given size even if the electroosmotic speed is lower than the target value, but the

dimensions of the trapping region will increase and hence the effective spring constants of the trap will decrease. A lower limit for the electroosmotic speed of $\sim 50 \mu\text{m}\cdot\text{s}^{-1}$ may be estimated by taking σ_x to be $\sim 0.5 \mu\text{m}$, which is the offset of laser foci from the trap center, which gives $t = \sigma_x^2 / (2D) \sim 0.01 \text{ s}$.

CHAPTER THREE:

MICROFLUIDIC DEVICE FOR TRAPPING

Microfluidic Cross-Channel Fabrication Overview

As mentioned in the previous chapter, the experiment employs a microfluidic cross-channel device for trapping a single nano-object. The main idea of using a cross-channel device is to exploit the phenomenon of electroosmosis in microfluidic channels and use it to control the position of a fluorescent nanoparticle dispersed in distilled water. The fabrication of such a cross-channel was done by bonding together two glass coverslips, each with a single channel in a film of PDMS at the surface, so that the channels overlay in a crossed configuration. The bonding of the PDMS surfaces is performed immediately after an air/oxygen-plasma activation process. The configuration is illustrated in Figure 3.1. After bonding two channels together, the middle section or crossing region becomes twice the depth of each single channel.

Since electroosmotic fluid flow must be voltage-driven, the cross-channel is interfaced with four platinum electrodes, which are placed at the ends of the channels within circular openings. By applying different potentials at the four electrodes, it is possible to control the electroosmotic flow in each microchannel and to orient the fluid flow at the center of the crossing region of the channels in any chosen direction in 3D [52]. As a result, a suspended fluorescent particle near the center of the crossing region can be moved by the electroosmotic flows

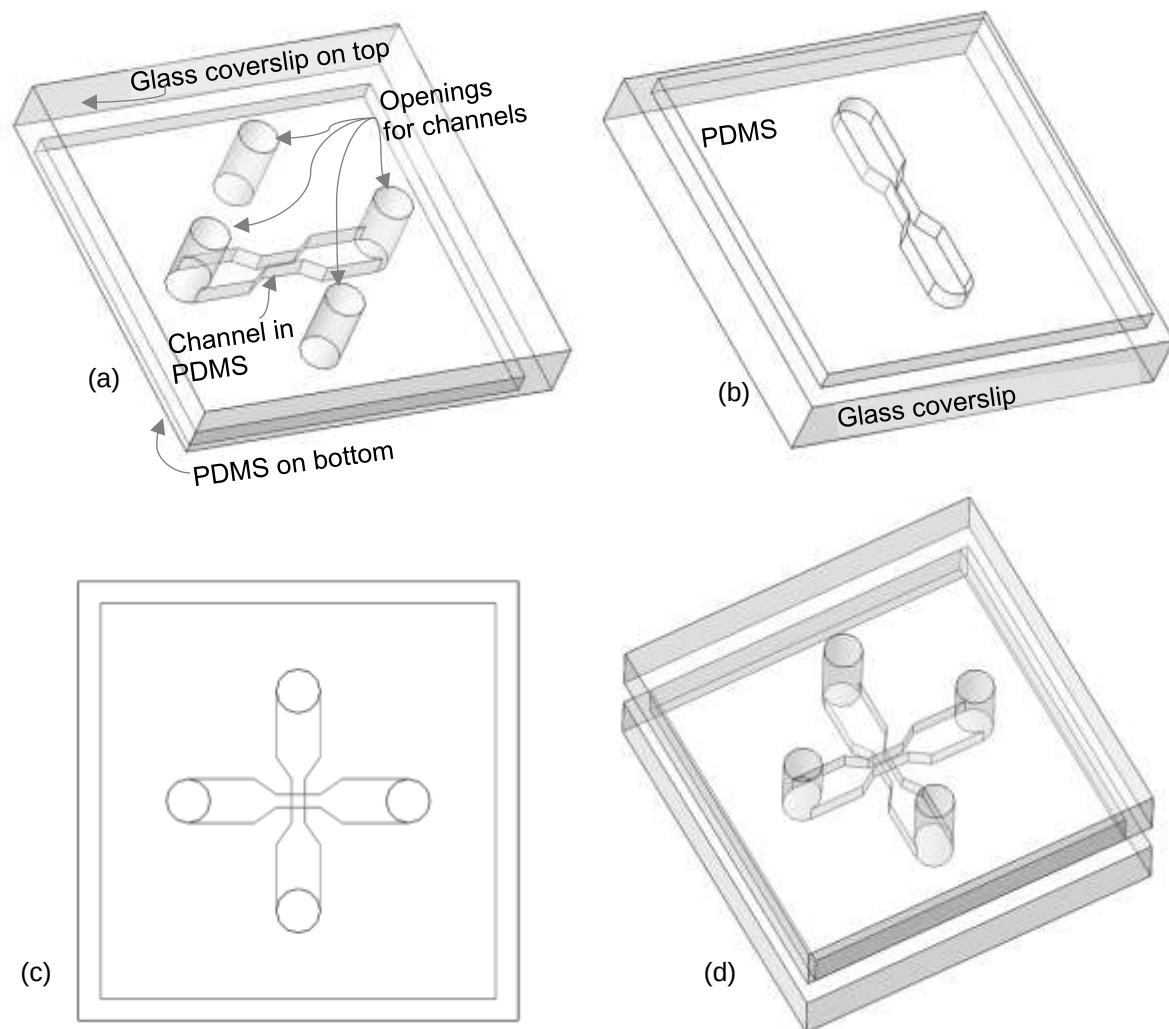


Figure 3.1. Tapered cross-channel formed in PDMS films on glass coverslips.

(a) Top piece: single channel with four openings made in the glass coverslip for fluid delivery. (b) Bottom piece: single channel. (c) 2D view of assembled cross-channel. (d) 3D perspective of assembled cross-channel. Note that the PDMS film and coverslip thicknesses are not to scale.

in any direction in 3D as desired, irrespective of its charge or size. In addition, if the particle is charged, there will be an extra electrophoretic flow velocity component that depends on the charge and size of the particle as well as the electric field at the center of the crossing region.

Microfluidic channels can be made directly in thin glass coverslips. However, the fabrication process is time-consuming and prone to various difficulties, such as the brittleness of glass itself and bonding issues. In bonding two glass pieces together, it is possible to apply an epoxy or resin layer in between the glass pieces and cure them [53] or use hydrofluoric acid bonding [54]. With microchannels directly laser-machined on the glass substrates, the success rate of getting a defect-free cross-channel is lower because the channels can be contaminated with the adhesive. Alternatively, it is very convenient to use fs laser machining of PDMS films deposited on glass coverslips [55, 56]. It does not require photolithography and etching techniques [57, 58], which need special laboratory environments and chemicals. Furthermore, the PDMS films are easily bonded together.

COMSOL Modelling and Results

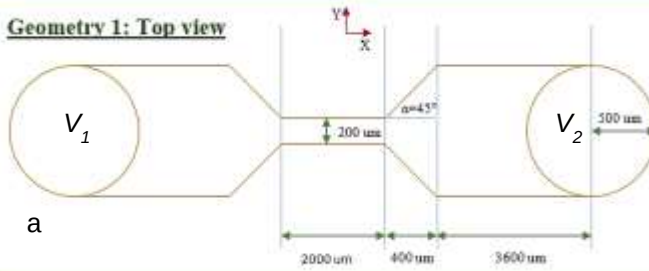
In designing the cross-channel, it is essential to consider several key factors. They are the dimensions that may be easily fabricated, the wide-angle fluorescence collection by the microscope objective, and the optimization of electroosmotic speeds to counteract molecular Brownian diffusion in distilled

water. The need for wide-angle fluorescence collection was introduced in Chapter Two. According to the microscope objective specifications (Olympus UPLSAPO 60X, $NA = 1.2$, water immersion) the width-to-depth of each channel near the point at which the particle is to be trapped must be at least $X_l : Z_l = 4:1$ (see Figure 2.2).

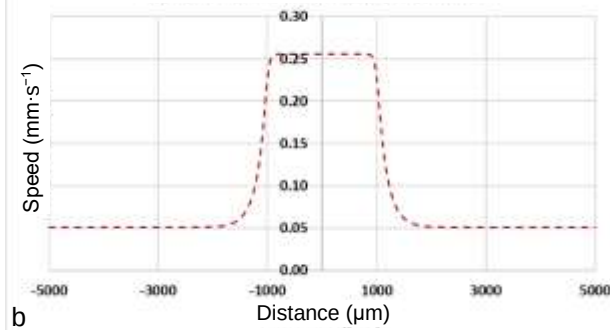
In order to optimize electroosmotic speeds in the cross-channel, the COMSOL Multiphysics® microfluidic analysis tool was used to conduct a fluid-dynamic analysis. For the analysis, it was necessary to design an efficient geometry for a single channel—one that is relatively simple to fabricate and also easy to analyze using COMSOL— and then determine if an electroosmotic speed of $\sim 1.6 \text{ mm}\cdot\text{s}^{-1}$ to counteract the Brownian diffusion of a nano-object could be obtained in that channel. In this context, two geometries were studied (Geometry 1 and Geometry 2 in Figure 3.2), in order to select one for further analysis towards making an operable cross-channel. The dimensions for the geometries are given in Figure 3.2, and the thickness of PDMS films was kept constant at $50 \text{ }\mu\text{m}$. The voltage along the channel (i.e., $|V_1 - V_2|$) was set to 20 VDC , since it is the maximum voltage that can be delivered using the available $\pm 10 \text{ V}$ outputs of the LTC2704 voltage-driving chip on the FPGA board.

The user-defined properties and parameters of the simulations are shown in Table 3.1. Temperature-dependent properties or parameters are automatically adjusted during the simulation, namely dynamic viscosity, density, ratio of specific heats, heat capacity at constant pressure, and thermal conductivity.

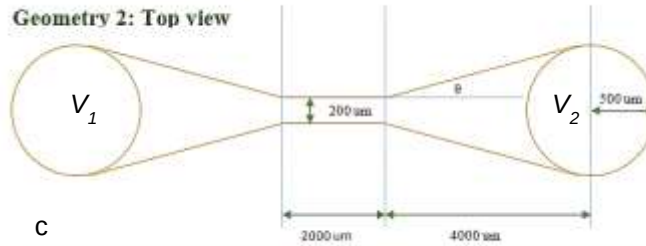
Geometry 1: Top view



Speed vs. Distance (along X-axis)



Geometry 2: Top view



Speed vs. Distance (along X-axis)

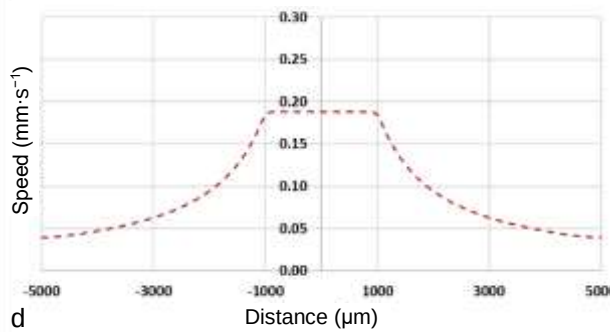


Figure 3.2. Two microchannel geometries for COMSOL analysis.

- (a) Geometry 1. (b) Electroosmotic speed along X-axis for Geometry 1. (c) Geometry 2. (d) Electroosmotic speed along X-axis for Geometry 2.

Table 3.1. Properties and parameter values for COMSOL simulation.

Property	Value	Unit
Temperature of water	293	K
Electrical conductivity of water	5.5×10^{-6}	$\text{S} \cdot \text{m}^{-1}$
Relative permittivity of water	80	–
Zeta potential of PDMS-water interface [39]	–70	mV

For the geometry in Figure 3.2(a), an electroosmotic speed along the channel of more than $0.25 \text{ mm} \cdot \text{s}^{-1}$ was determined, while for the geometry in Figure 3.2(c) the speed was less than $0.20 \text{ mm} \cdot \text{s}^{-1}$. Since the target is to get the highest possible fluid speed in the center part of the channel (i.e., the region between $-1000 \text{ }\mu\text{m}$ and $+1000 \text{ }\mu\text{m}$ in Figure 3.2(b) and Figure 3.2(d)) for a given geometry, we can choose Geometry 1 (Figure 3.2(a)) for further analysis. What we next consider is to build a cross-channel in a simulation and vary the total length of a channel (L_1), length of the center part (l_2), width of the center part (l_1) and the diameter of the fluid inlets (L_2) to optimize the fluid speed in the center part of the cross-channel, as needed to counteract Brownian diffusion. An improved design with electroosmotic speed in the center still $< 1 \text{ mm} \cdot \text{s}^{-1}$ with its dimensions is shown in Figure 3.3. This design was fabricated for the experiment.

The cross-channel dimensions chosen for fabrication were not theoretically expected to give the full desired flow speed but were chosen in part due to physical limitations of the fabrication and assembling processes. In particular, the minimum length of a channel is limited to about 5 mm in order to

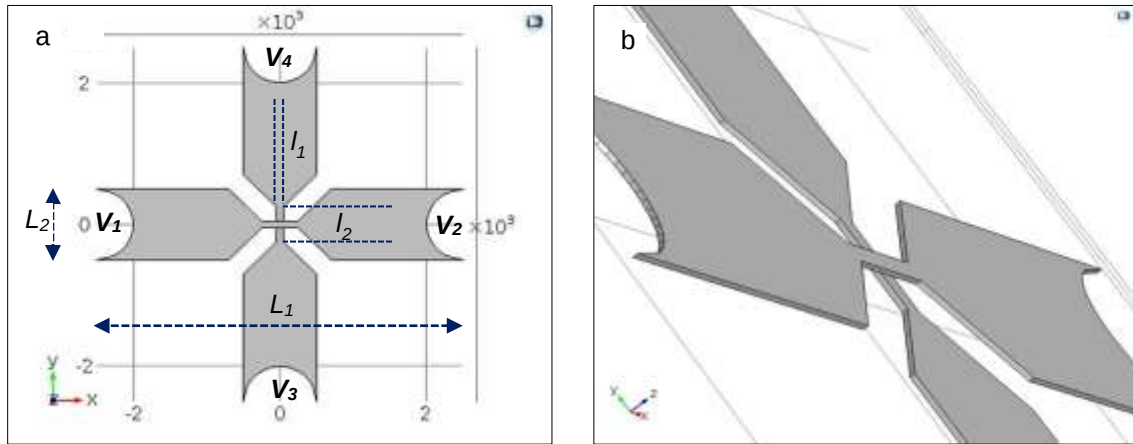


Figure 3.3. Cross-channel design for fabrication.

(a) 2D view: $l_1 = 100 \mu\text{m}$, $l_2 = 500 \mu\text{m}$, $L_1 = 5000 \mu\text{m}$ and $L_2 = 1000 \mu\text{m}$. The axes scales are in μm and V_1 , V_2 , V_3 , V_4 represent the voltage values for electrodes. (b) 3D view: thickness (Z in Figure 2.2) = $25 \mu\text{m}$ per channel (not to scale).

facilitate the fluid feed from fluid reservoirs that were machined in a Perspex hood. Four holes from the reservoirs are interfaced with the four inlets of the cross channel via four rubber O-rings with inner diameter 1 mm and outer diameter 3 mm. Keeping the overall channel length at about 5 mm makes it easier in interfacing the fluid reservoirs with the cross-channel, but does not fully utilize the opportunity of increasing the electroosmotic speeds by reducing the length of the channels. Also, the length l_2 was kept at $500 \mu\text{m}$ to allow a clear separation and good bonding of the channels during the bonding process so that no fluid leakage occurs in the middle part of the channels. Also, the chosen thickness of the PDMS films accounts for the strength of the bonded cross-channels. It was experimentally observed that for a thickness of about 20–30 μm , the bonding of channels could be very smooth and less prone to be broken

during handling. Hence, the thickness of the PDMS films on glass coverslips was set to be 25 μm . This value of thickness sets the width of the middle part of a channel to be about 100 μm leading to the $X_l : Z_l$ ratio of 4:1 needed for unobstructed fluorescence collection by the microscope objective as explained in Chapter Two.

In the COMSOL analysis, the key priority is given to the speeds along a channel in the X (or Y, which is assumed to be identical to X) and the Z directions. For the speed analysis along X+, V_1 and V_2 are set to +10 V and -10 V, respectively, while V_3 and V_4 are both set to 0 V. Since the X- and Y- axes have identical configurations, it is not necessary to conduct a separate analysis for the Y-axis. In the case of the speed along Z+, V_1 and V_2 are set to -10 V while V_3 and V_4 are set to +10 V to attain the maximum fluid speed. In order to study the speeds along negative directions, simply reversing the respective voltages suffices. Figure 3.4 shows the fluid speed at the center of the cross-channel for distilled water (outlined in red in Figure 3.4(a)) along X- or Y- (Figure 3.4(b)) and Z- (Figure 3.4(c)) axes. (The discontinuities in the curve in Figure 3.4(b) at $\pm 50 \mu\text{m}$ are not real but are due to numerical issues.)

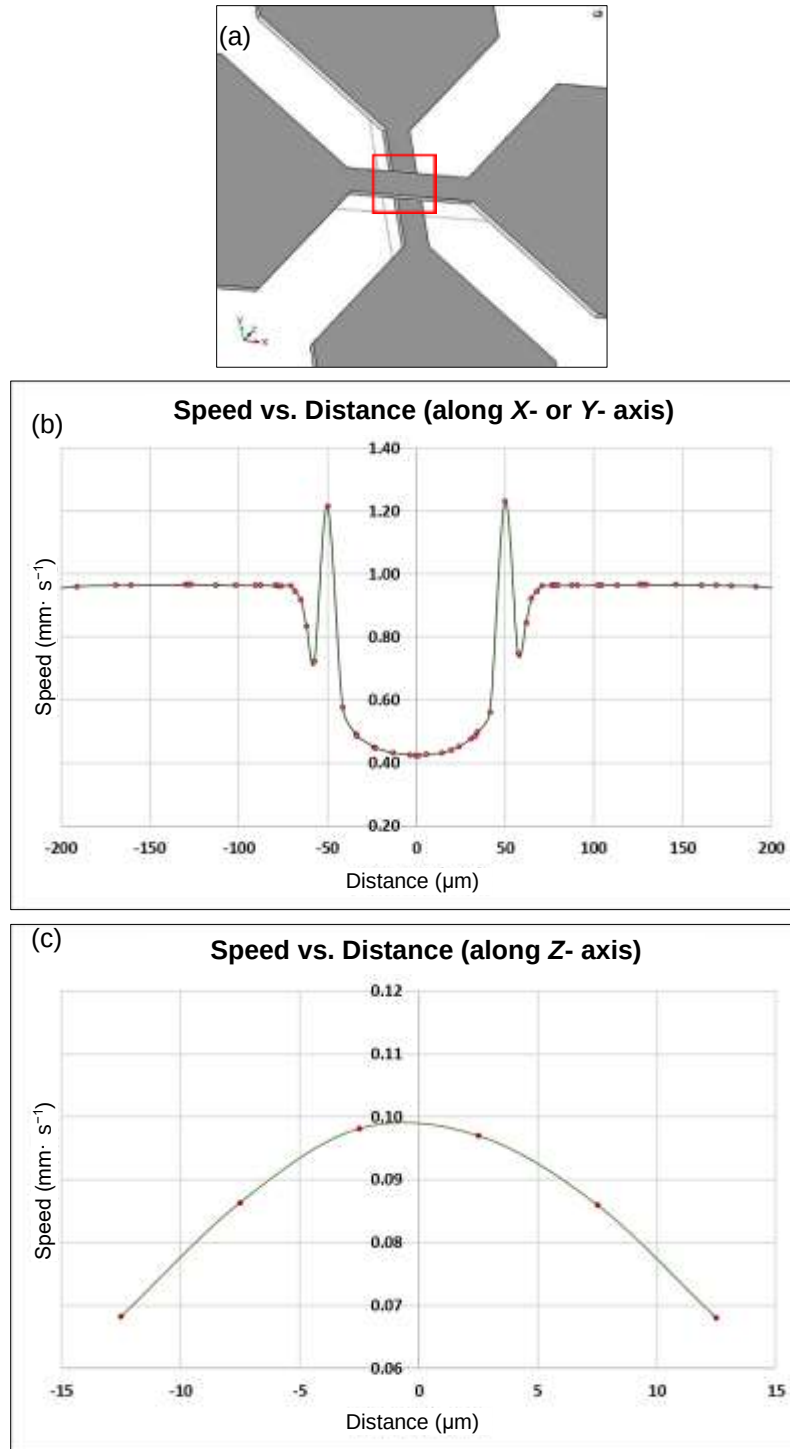


Figure 3.4. COMSOL electroosmotic speed analyses for the cross-channel.

(a) Center of the cross-channel. The red box covers the range $-200 \mu\text{m}$ to $+200 \mu\text{m}$ in the X-Y plane. (b) Speed along X- or Y-axis. (c) Speed along the Z-axis.

Device Fabrication

Material Preparation

The main materials used for fabrication of cross-channels are glass coverslips and PDMS. They were prepared using a repeatable protocol in a Class 1000 clean room where microprocessing of materials is conducted. The glass coverslips used were brand-new 180 μm thick Fisherbrand™ borosilicate cover glasses measuring 22 mm $\times$ 22 mm and were first prepared according to the following steps:

Step 1: Methanol bath with sonication for 1 minute

Step 2: Acetone bath with sonication for 2 minutes

Step 3: Methanol bath with sonication for 1 minute

Step 4: Distilled water bath with sonication for 2 minutes

Normally, the outcome should be dust- and oil-free glass coverslips. In case further cleaning appeared to be necessary, a repeat of Steps 2 to 4 is conducted. The cleaned coverslips were stored in closed petri dishes for later use.

The PDMS samples were made by mixing SYLGARD® 184 silicone elastomer base and curing agent in the weight ratio of 10:1 [59]. The components were thoroughly mixed for about five minutes until a uniform mixture was attained. The mixture gets uniformly bubbled (looks milky) when it is thoroughly mixed. It is necessary to let the mixture sit until the bubbles percolate out before spin coating (about 20–30 minutes but not more than 1 hour because the mixture will start to cure even without the application of heat). Curing makes the viscosity

change and it affects the fabrication processes such as in preparing thin PDMS films by spin coating as discussed in the next section.

Spin Coating of PDMS Films

Once PDMS and glass coverslips are ready, we can start making PDMS films through spin coating. The Chemat precision spin coater model KW-4A was used in this experiment. When using the spin coater, it is first necessary to center a coverslip on the spin coater's sample-holding vacuum chuck and then turn on the vacuum (the spin coater will not operate if the vacuum pump is not running.) Once the coverslip is placed properly, a drop of PDMS having a volume of about 1 ml is placed in the center of the coverslip. Next, flipping the Control switch on prepares the spin coater with the selected speed and time settings.

The spin coater uses a two-speed process consisting of lower- and higher-speed cycles. The low-speed cycle (Speed-I, Timer-I) is used to spread the material evenly over the sample surface, and the high-speed cycle (Speed-II, Timer-II) is used to achieve the desired film thickness. The settings used for the low-speed cycle were: Speed-I at 400 rpm, Timer-I at 9 seconds. For the higher-speed cycle, the Timer-II was set at 60 seconds while the speed was subjected to change to achieve different thickness PDMS films. Those settings were tested in prior spin coating experiments in our lab and had proven that the settings could provide very good results. Once the speed and time settings are complete, the lid must be placed over the spin coater to contain splashes. Pressing the Start button then engages the spin coater, which automatically stops after the

two cycles have completed. Once the spin coater has finished running, the Control switch should be turned off and the vacuum must be disengaged in order to remove the coated coverslip.

The COMSOL analyses showed that the thickness of the PDMS film is very critical in determining the electroosmotic fluid flow speeds in the tapered channels. In order to achieve a precise and reproducible film thickness during spin coating, Equation 15 was employed [60]:

$$h = \frac{h_0}{(1+c\omega^2 h_0^2 t)^{0.5}} \dots\dots\dots (15)$$

where h is final thickness, h_0 is initial thickness, ω is angular speed in rpm, t is spinning time in seconds, and $c = 4\rho/3\mu$, where ρ is density and μ is viscosity. (It is assumed here that the PDMS solution is Newtonian so that μ is independent of the shear rate.) The parameters h_0 and c were empirically determined to be $h_0 \approx 500 \mu\text{m}$ and $c \approx 5.01 \times 10^{-12} \text{ rpm}^{-2} \cdot \mu\text{m}^{-2} \cdot \text{s}^{-1}$. In determining c , it was necessary to conduct a nonlinear curve fit to the experimental data using OriginLab® as shown in Figure 3.5.

Knowing the parameters h_0 and c of the Equation 15, it is possible to extrapolate the PDMS thickness for ω values below 900 rpm and above 2000 rpm. This extrapolation helps in visualizing the thickness of a PDMS film for a given angular speed that can even go beyond 5000 rpm until it reaches the speed limit of the spin coater. The graph of Thickness vs. Angular Speed is shown in Figure 3.6 with experimental data highlighted in red. Error bars

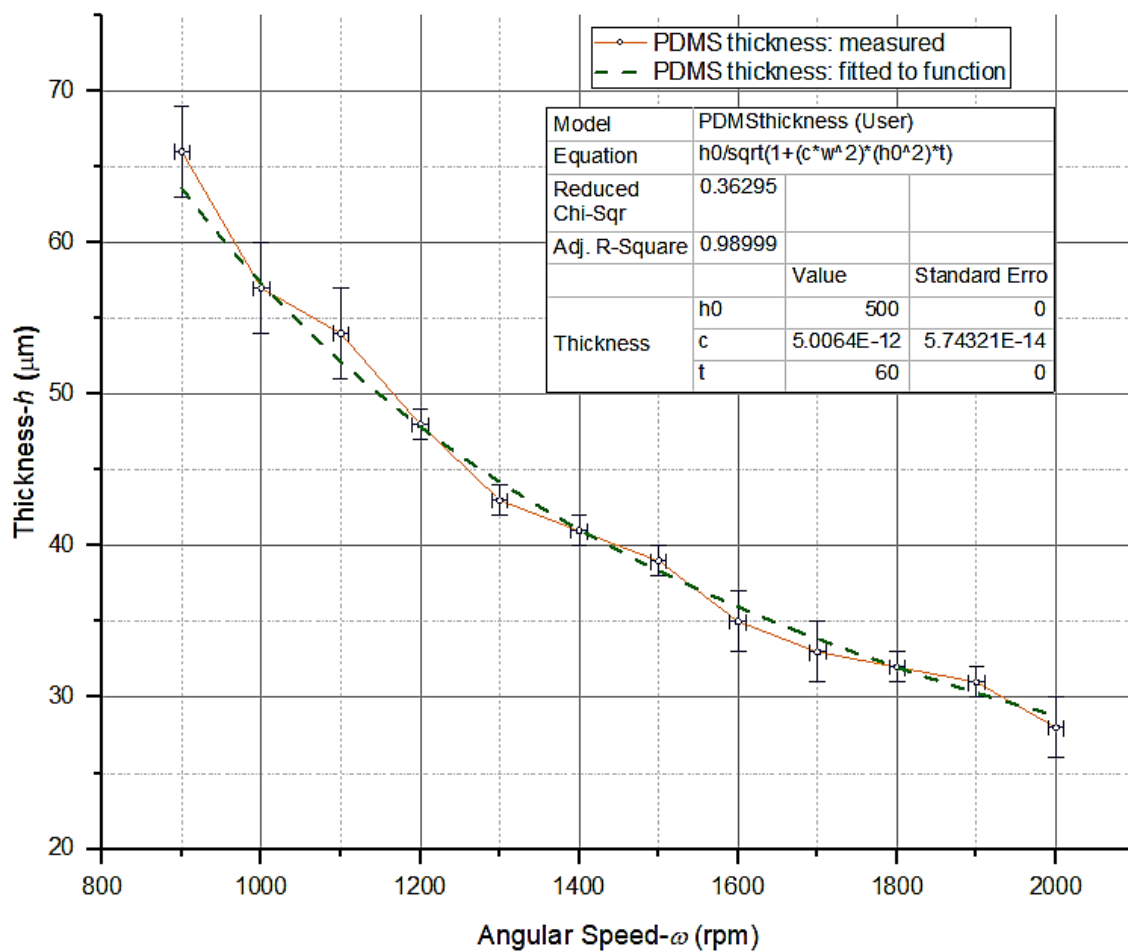


Figure 3.5. Curve fitting PDMS Thickness vs. Angular Speed to estimate C in Equation 15.

Note that the Angular Speed is the speed values that were set for the high-speed cycle. The low-speed cycle was only used to spread out PDMS all over the substrate coverslip so that higher-speed cycle can attain a very smooth PDMS film on the coverslip.

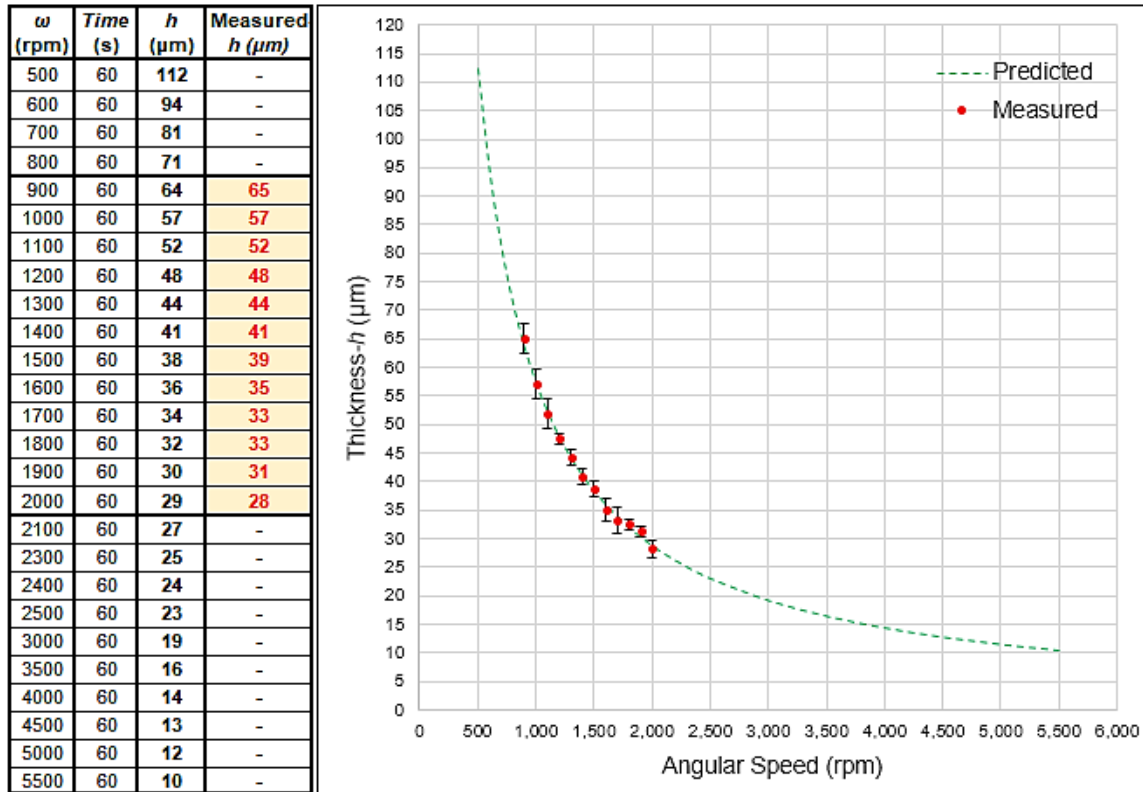


Figure 3.6. Extrapolation of PDMS Thickness vs. Angular Speed.

The experimental data are highlighted in red and the runtime for each data point is 60 s.

(standard deviations) are also shown for the experimental data to clarify the predictability of the theoretical model. From these results it was chosen that Speed-II has to be set at 2300 rpm with Timer-II at 60 s to produce a 25 μm thick PDMS film, which would be used for making the channels for the trapping experiment.

Note that the PDMS thickness predictions are for the center part of the coverslip. The edges of the PDMS film exhibit a ridged behavior causing thickness variations. This ridge effect probably occurs due to the discontinuation of the viscous flow of PDMS as a film near the edges of the coverslip. The PDMS that resides at the edges of the coverslip after the spin coating may tend to move inside the coverslip due to viscous forces creating ridges. It is necessary to remove this unevenness at the edges manually before going for the bonding process of PDMS films to have a perfectly bonded device.

Immediately after spin coating, the PDMS-coated coverslips were moved to the bottom of a plastic petri dish, and the dish was covered and placed on a hot plate at 80 $^{\circ}\text{C}$ with a protective sheet on it so that the PDMS films could be baked to cure rapidly. It was discovered that baking time had to be limited to about 30 minutes. Longer baking time overcures and makes the PDMS film very stiff, so it becomes very difficult to achieve a perfect bonding process after plasma treatment because the PDMS films cannot make good contact with one another.

Once the curing is finished, the samples could be used straight away to create tapered microchannels in them by fs laser machining. However, the ridged features at the coverslip edges were first manually removed by carefully scraping them off with a razor blade. This avoids damaging the coverslip and leaving debris on the remaining film before the bonding process. Good bonding results were observed when a strip of PDMS about 3 mm wide was removed from around the perimeter of the coverslips. The removal of the ridged features could be done before or after the fs laser machining stage, but completing it beforehand ensures that microscopic debris will not be left in the laser machined microchannels, and that possible accidental cracking of the fragile substrate during removal does not lead to wasted effort on machining the microchannels.

Femtosecond (fs) Laser Machining

The fs laser machining was carried out using a Coherent® RegA 9000 laser system that can provide up to 4 μJ , 160 fs pulses at 790 nm with a 250 kHz repetition rate. The basic machining apparatus employs a static microscope and motorized translation stages to move the sample under the focused beam to ablate patterns. An ultrafast dichroic laser mirror is used to direct the fs laser beam into the microscope objective used for machining. This mirror also allows simultaneous visual observation of the sample through the microscope eyepiece using the microscope's illumination source. (Although the source is white, the light transmitted through the ultrafast mirror appears green, as depicted at right in Figure 3.7.) The laser machining process is a fully computer-automated process

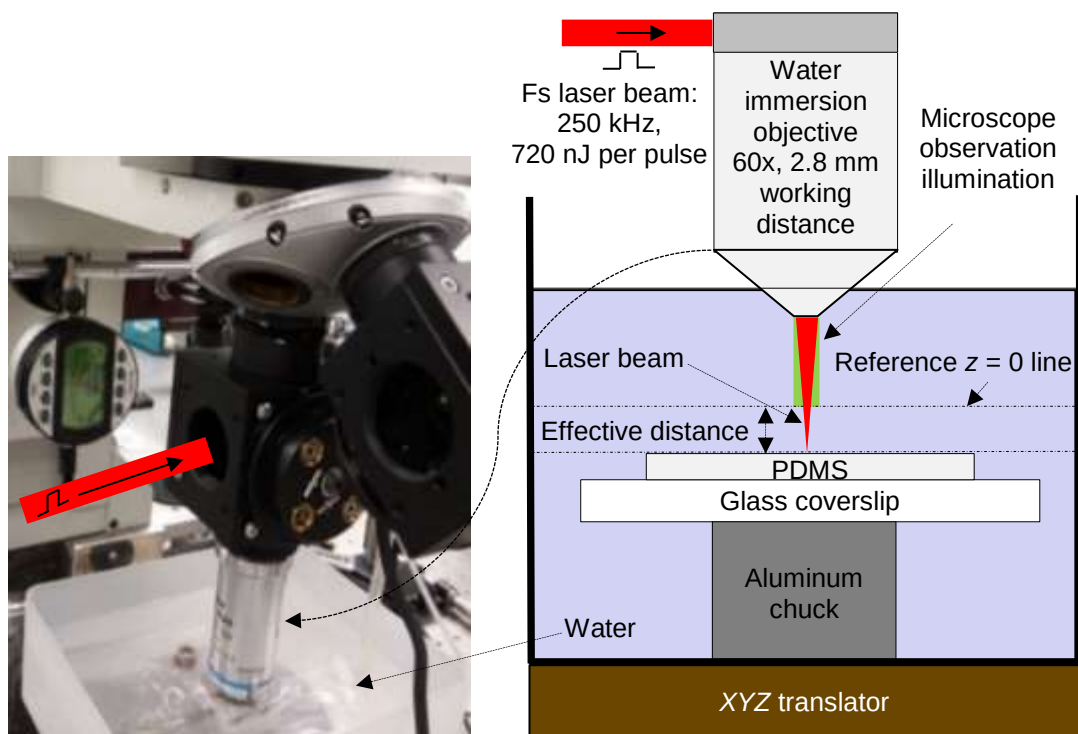


Figure 3.7. Laser machining setup for PDMS samples.

that is controlled through a custom-written LabVIEW program (developed by Dr. Brian Canfield), which allows constructing custom, complex patterns to be ablated in a sample.

When the program runs, it opens a shutter to pass the laser beam to the sample and actuates Aerotech ANT95XY and ANT95-3 Z-axis nanopositioning stages to trace the custom pattern set for ablation on a flat sample placed under a microscope. When machined in air, the PDMS can char and scatter debris. Moreover, the underlying coverslip glass may be damaged, which would affect both fluid flow and optical transparency in the microchannel. Best results for material removal are instead obtained by submerging the PDMS sample in water

and ablating material through cavitation, with a directed flow of water to remove debris from the focal region (see Figure 3.7). This method not only avoids charring the PDMS but also leaves clean, debris-free surfaces after rinsing.

In laser machining of PDMS films, it is essential to set a reference plane $z = 0$ for vertical movement of the translator. First, the sample must be leveled with respect to the X - Y plane of the translation stage and aligned for translation along X and Y . The laser is not fed to the microscope during these adjustment processes. Levelling and alignment is performed with the aid of a 2-axis tilt/rotation stage mounted atop the XYZ stages, on which the immersion tub in Figure 3.7 is mounted. By iteratively translating from edge to edge of the coverslip and checking the visual focal distance change with a linear stylus gauge (visible in the back at left in Figure 3.7), the sample can be leveled to maintain a constant focus over the entire coverslip to within $1\text{ }\mu\text{m}$ in both X - and Y -axes. Using the rotation feature of the stage, the coverslip edges can then be aligned parallel to the X - and Y - axes using a similar iterative procedure. In addition, a large XY translation stage underlies the nanopositioning stages and is used to reposition the coverslip for direct centralization of the XY coordinate origin to $x = 0$, $y = 0$ (displacements from this value can easily be handled by offsets in the LabVIEW program).

Once levelling and centering is finished, the reference $z = 0$ plane is set at the planar coordinate center ($x = 0$, $y = 0$) of the top of the PDMS film also by visual observation through the microscope. When laser machining, it is essential

to displace the sample away from the microscope objective along Z by a specified “effective distance” d , so that the focused laser starts ablating at the top of the PDMS film (this displacement is also handled programmatically through LabVIEW). The value of d depends on the XY translation speed, and the two most-used values are listed in Table 3.2.

It is essential to account for several key factors during the laser machining. The most important factor is the optimal pulse energy needed to ablate PDMS but not to cause any laser damage to the glass coverslips. It was concluded through empirical observation that ~180 mW average power provides the optimal pulse energy, 720 nJ at a repetition rate of 250 kHz, for the XY translation speeds in the range $5000 \mu\text{m}\cdot\text{s}^{-1}$ – $10,000 \mu\text{m}\cdot\text{s}^{-1}$. For these values, it was observed that a 5 μm translation in Z for the next ablation cycle yields a smooth, gradual removal of PDMS layer by layer. Thus, for a 25 μm thick PDMS film, 5 ablation cycles are needed to machine a single microchannel. In addition, it was preferable to use $10,000 \mu\text{m}\cdot\text{s}^{-1}$ speed for the final 5 μm layer removal to have a clean, undamaged glass surface at the bottom of the channel.

After forming the channels in this way, they were observed to be wider

Table 3.2. Key parameters used for fs laser machining of PDMS.

Energy per Pulse (nJ)	XY Translator Speed ($\mu\text{m}\cdot\text{s}^{-1}$)	Effective Distance- d (μm)
720	5000	60
	10000	40

than intended by about 10 μm at the bottom and about 20 μm at the top. The widening at the bottom can be compensated in the channel design, but the tapering is unavoidable with the present machining setup. Hence, the walls of laser-machined channels will not be perfectly trued at the microscopic level. An optical micrograph of a laser-machined single channel is shown in the Figure 3.8 and note the clean, undamaged glass quality at the bottom of the channel.

In order to drive fluid into a bonded cross-channel, there were four circular holes with 1 mm diameter in each, created by fs laser machining in the glass coverslip that goes as the top piece of the cross-channel device. The coordinates of the centers of the holes in the X-Y plane are (2.5,0), (0,2.5), (-2.5,0) and (0,-2.5) in millimeters. From these four holes, two holes are directly laser machined at the end of the tapered channel (see Figure 3.8) of the top piece, and the other two holes are to be aligned to the two ends of the tapered channel of the bottom piece of the cross-channel. The holes were laser machined with the same fs laser, but with full power of 1 W due to the rigidity of glass compared to PDMS. The XY translator in this situation was moved in a circle with 0.5 mm

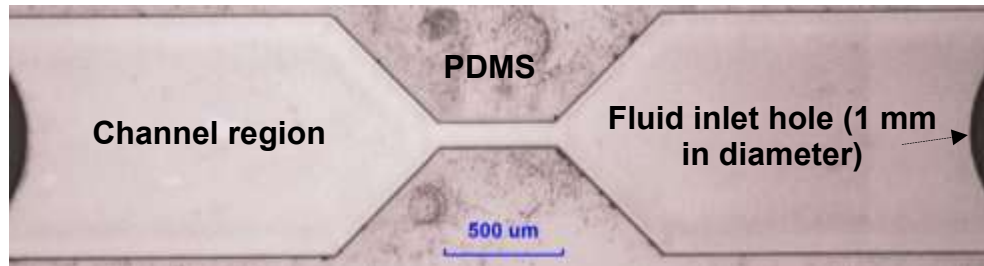


Figure 3.8. Optical micrograph (60x) of a fs laser machined microchannel.

radius centered at the coordinates mentioned above and the translation speed in the X-Y plane per axis was set to $5000\ \mu\text{m}\cdot\text{s}^{-1}$. The translator is moved up in Z once a cycle of ablation is completed. It was observed that about 10 μm translation in Z for the next cycle of ablation does a smooth ablation of the glass coverslip. Since the thickness of the glass coverslip is 180–200 μm , a hole can be ablated in about 20 cycles. Note that the laser cutting of holes in a coverslip was performed first to make sure there is no cracking in the glass before machining the microchannel in PDMS.

PDMS Bonding

The PDMS films with machined microchannels have to be cleaned rigorously before they can be bonded. In order to have a top-quality PDMS surface, it undergoes a specific cleaning routine as described below:

- Step 1: Acetone bath with sonication for 1 minute
- Step 2: Methanol bath with sonication for 1 minute
- Step 3: Water bath with sonication for 1 minute
- Step 4: Acetone bath with sonication for 15 minutes
- Step 5: Flush with distilled water for 20 seconds
- Step 6: Blow dry with inert gas (nitrogen/argon)

Once the cleaning is completed, it is essential to perform the plasma activation treatment of the PDMS surfaces as soon as possible, to avoid contamination by airborne particles, even in the clean room environment where all the processing occurs. For the plasma treatment, a Harrick Plasma cleaner PDC-32G model was used and the protocol for the process was strictly followed since the

requirement is just to activate the PDMS surfaces but not to damage them with overexposure. Activated PDMS films are supposed to be bonded immediately since the activation only lasts for a few minutes. The bonding of two films is accomplished with a micrometer-driven 3D translation stage while viewing through a stereoscopic microscope. One coverslip is held on a fixed arm, and the other is placed on the 3D stage and quickly maneuvered into proper alignment directly below the fixed slide, then raised into contact. Once the films contact, they must be gently but firmly pressed together (with a cotton-tipped swab) to ensure a complete seal around the microchannels so no fluid leakage will occur. Figure 3.9 shows an optical micrograph (top view) of a perfectly bonded cross-channel and Figure 3.10 shows a photograph of an assembled device with physical dimensions.

Figure 3.11 shows how the edge ridge feature can affect the bonding of PDMS films. The darker areas in each glass piece demonstrate perfect bonding, while ridged areas are whitish with either imperfect or no bonding. The ridges actually cause concavity around the central region, creating extra stress on the glass coverslips so that they may break when interfacing with the Perspex hooding (see Figure 3.12) that contains the fluid reservoirs and electrodes with platinum wires for driving the electroosmotic flow.

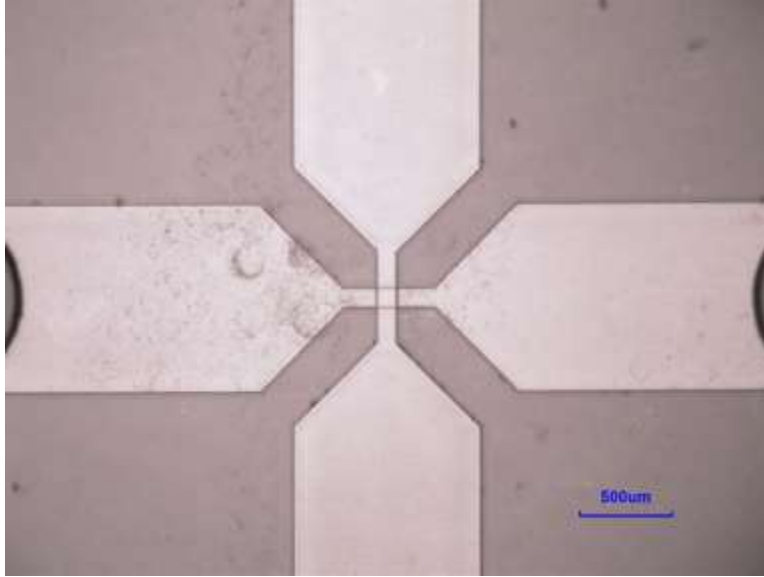


Figure 3.9. Optical micrograph (60x) of an assembled cross-channel device.

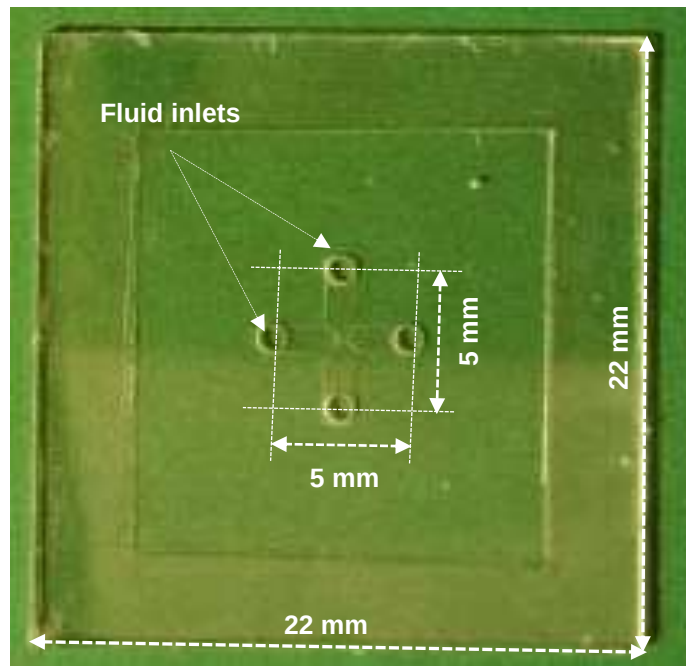


Figure 3.10. Photograph of an assembled cross-channel device showed in Figure 3.9.

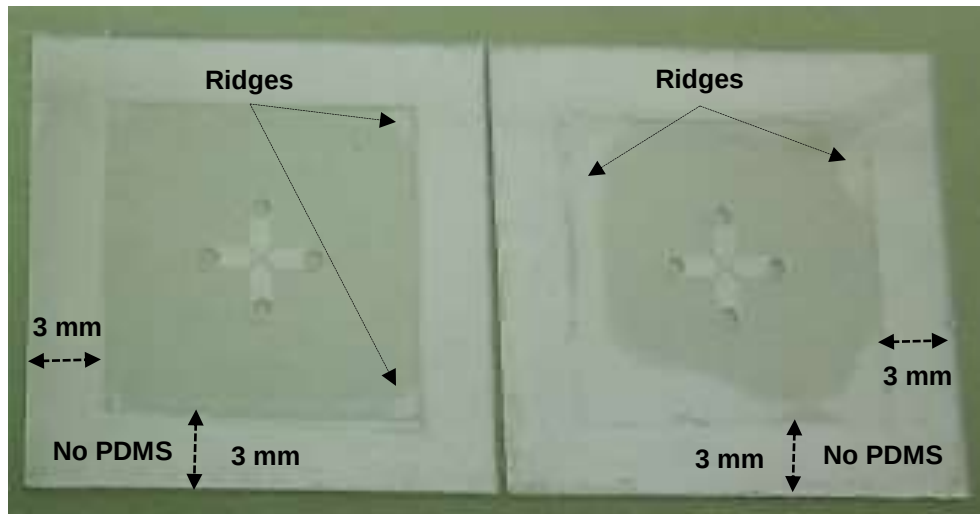


Figure 3.11. Ridge effect in spin coated PDMS films.

The device at left can be considered a better bonding result compared to the one at right with more ridges. However, both have sealed channel regions and are potentially viable.

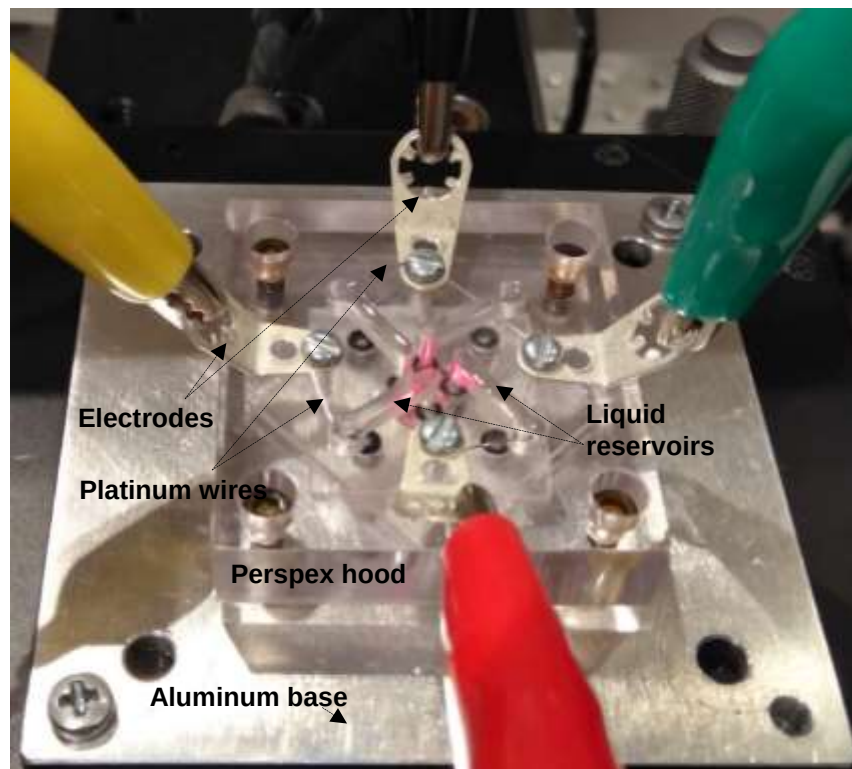


Figure 3.12. Cross-channel device (not clearly visible) interfaced with a Perspex hood.

Device Assembly

A cross channel is interfaced with a Perspex hood in order to connect with fluid reservoirs, as mentioned in the previous section. There are four cylindrical holes connected to four reservoirs, drilled through the Perspex hood to connect and feed the cross-channel device placed underneath. There are four Rubber O-rings (with inner diameter 1 mm, outer diameter 3 mm and 1 mm thickness) to connect the holes from the reservoirs and the inlets of the cross-channel device so that there will be no fluid leakage. Due to the O-rings, there is no direct physical contact of the Perspex hooding with the cross-channel piece underneath. Also, there are four more O-rings placed below the Perspex hood to smooth the suspension of the Perspex hood on the cross-channel device. Figure 3.13 shows how the top and bottom sections of the Perspex hood were machined (see Appendix A for the schematics with dimensions) so that they interface with the cross-channel. The cross-channel and Perspex hood are held together by bolting down the four corners of the Perspex hood to an aluminum base, as shown in Figure 3.13(e). The aluminum base has a 8 mm through-hole in the middle for viewing the cross-channel with the microscope objective.

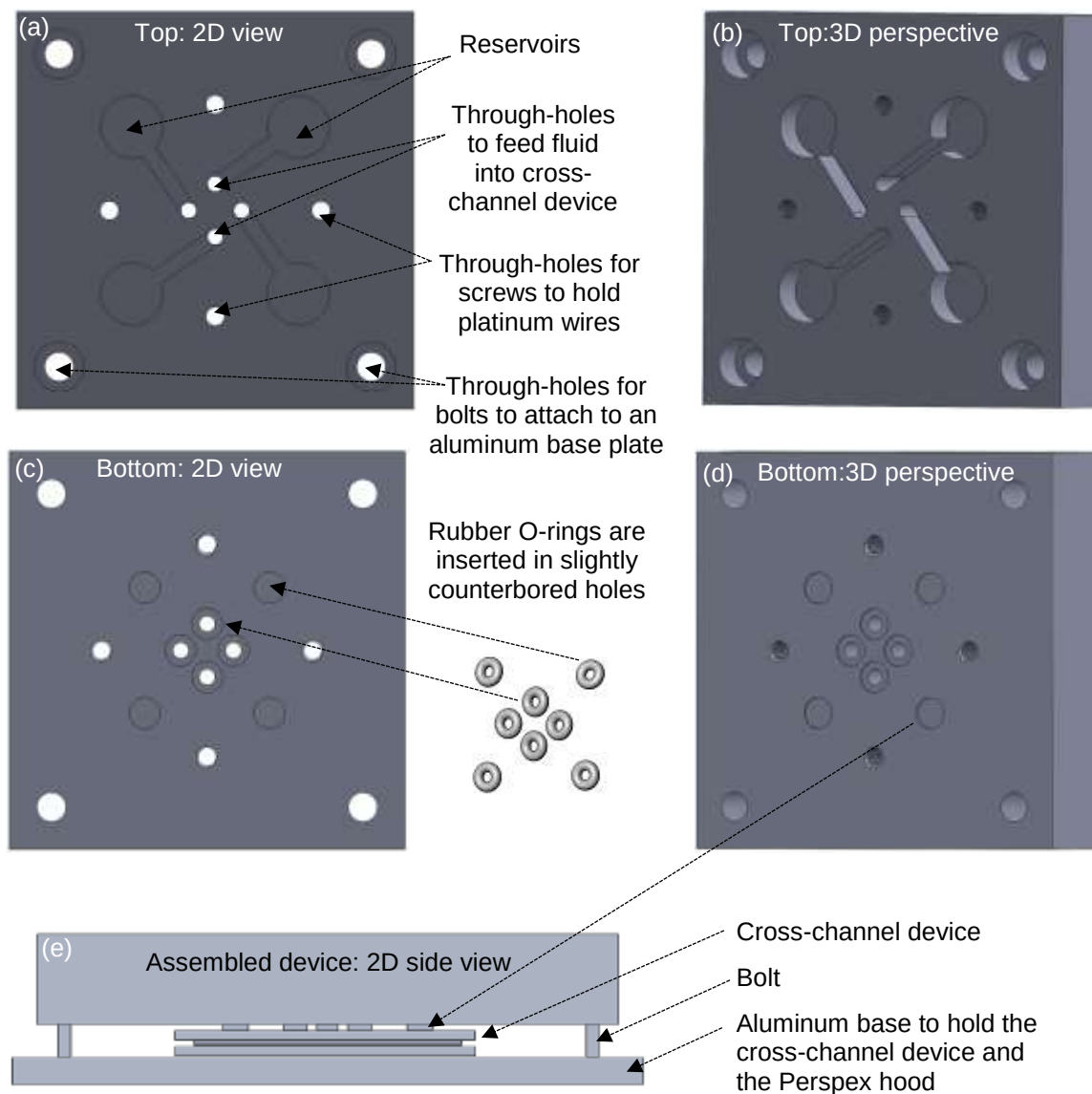


Figure 3.13. Cross-channel device interfaced with Perspex hood (not to scale).

Note that the schematic doesn't show how the electrodes were attached—they were attached with screws at the designated holes shown in the schematic, as shown in Figure 3.12.

CHAPTER FOUR:

OPTICAL SETUP, DATA ACQUISITION AND FEEDBACK

One-photon, Four-focus Microscope

As explained in Chapter One, the trapping region of the particles is determined by four partially overlapping laser beam foci with centers located at the vertices of a tetrahedral geometry. This geometry is illustrated in Figure 4.1(a). CCD camera images (where one pixel represents ~ 192 nm in object space— $16\text{ }\mu\text{m}$ pixel width divided by 83.3 image magnification) showing the experimental arrangement of the four foci, with each having a beam waist of about $0.5\text{ }\mu\text{m}$, are shown in Figure 4.2.

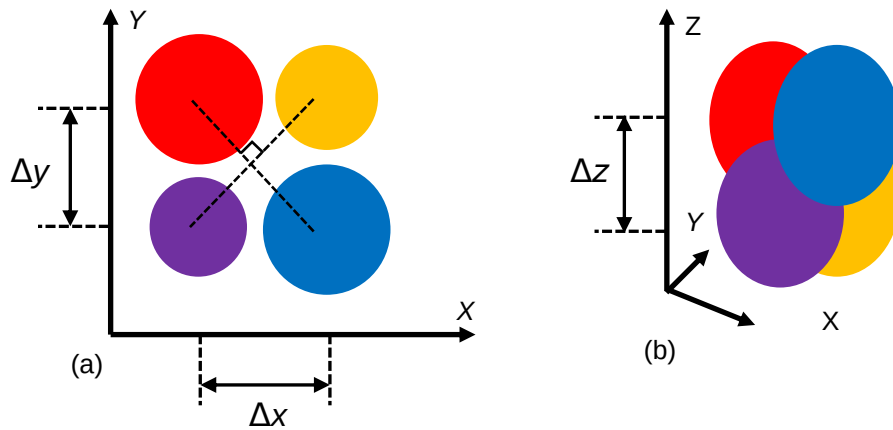


Figure 4.1. Schematic of four-foci arranged in a tetrahedral pattern.

(a) Viewed along the optic axis with the negative Z-direction going into the page. The separation between foci along the X- and Y-axes (Δx and Δy) is set at 760 nm . (b) 3D perspective of four foci arranged in a tetrahedral pattern. The separation between the two sets of foci along the optic axis (Δz) is set at 3200 nm in water.

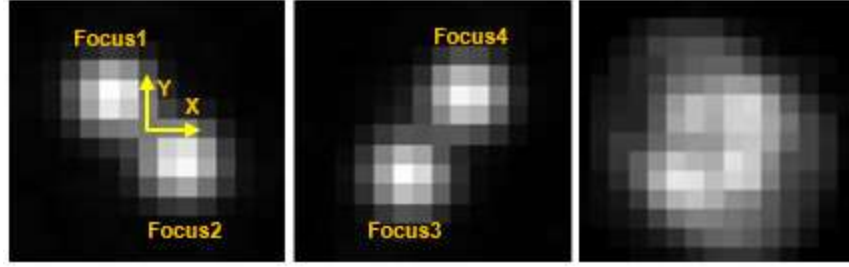


Figure 4.2. CCD Camera images of the four foci.

Left: foci Focus1 and Focus2 on top plane ($z = +1.6 \pm 0.1 \mu\text{m}$). Middle: foci Focus3 and Focus4 on bottom plane ($z = -1.6 \pm 0.1 \mu\text{m}$). Right: camera is focussed at $z = 0$ while all four beams are present.

All the images of the foci were taken by reflection of laser light at the glass-water interface (set horizontally in the X-Y plane) that comes after a microscope objective (set vertically along the Z-axis). A drop of water on a glass coverslip creates the glass-water interface and the coverslip is interfaced with a water immersion objective. Note that the coverslip is placed on an aluminum base with a through-hole in it (so that the objective can interface with the glass coverslip), which is mounted on an XYZ piezo translator stage.

The piezo stage is moved along the Z-axis so that the reflected light from a beam comes into focus at the CCD camera. The z-height that brings the beam into focus at the glass-water interface mentioned before was determined by recording the intensity of light in the CCD image as the z-height is scanned. The CCD image records a $32 \text{ pixel} \times 32 \text{ pixel}$ range and each pixel records a 16-bit value depending on the intensity to which the pixel was exposed. These pixel values are then summed column wise to generate an array of integer values (array-X) and the array undergoes a 1D Gaussian fit to estimate a peak intensity

value for the situation. The rows of the 2D image were also summed to generate another array of integer values (array-Y) and another 1D Gaussian fit was applied to estimate the orthogonal peak intensity value for the situation. These peak intensity values are then plotted against z-height and then fitted to a 1D Gaussian to estimate the center of the Gaussian in Z. The Gaussian centers taken from array-X and array-Y are then averaged to represent the z-height that corresponds to the maximum intensity of the reflected light of a laser beam.

All the z-heights were recorded corresponding to each of the four laser foci that are in a tetrahedral arrangement and are accurate to within $\sim 0.1 \mu\text{m}$, since the piezo stage was moved in $0.1 \mu\text{m}$ steps along the Z-axis. Note that the intensity measurement, piezo stage movement and Gaussian fitting were automated by a custom-made LabVIEW program. The X- and Y-coordinates were estimated just by counting the number of pixels to the center of the each focus from the center of the X-Y plane. Also, note that the beam waist per beam was also determined by recording each individual beam focus in the X-Y plane and performing 2D Gaussian fitting.

In the early experiments, where a two-photon, four-focus microscope was in operation, these four beams were achieved by splitting a single fs Ti-Sapphire laser beam (pulsing at 76 MHz) into four beams with an optical setup similar to a double Michelson interferometer [61] with appropriate delays in the separate arms, as shown in Figure 4.3. However, due to a power failure of the pump laser in the lasing system, the interferometer setup was changed to a one-photon,

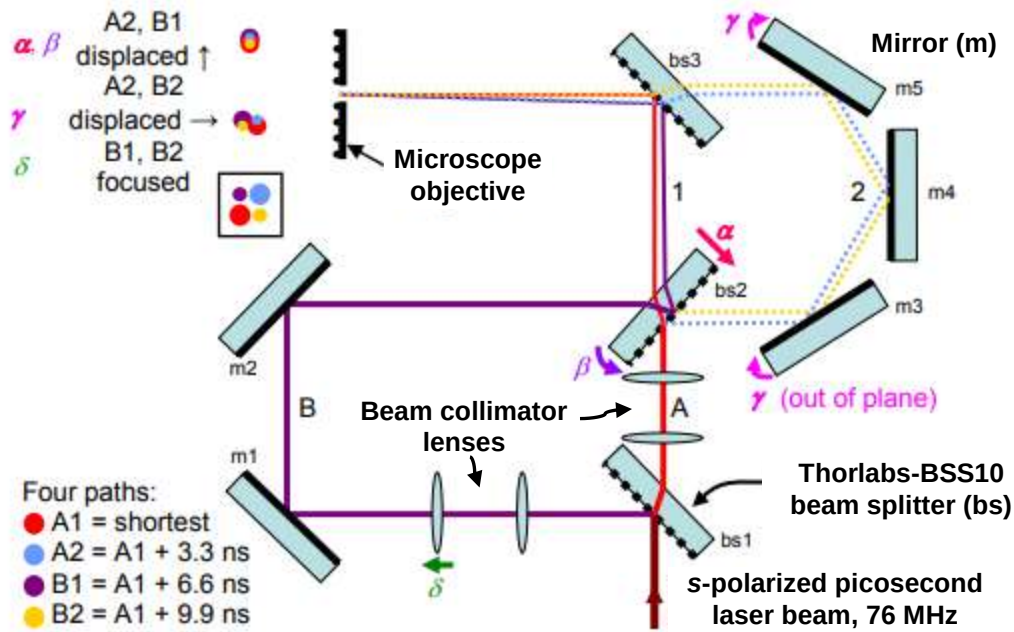


Figure 4.3. Double Michelson interferometer layout for four foci generation at 304 MHz [61].

four-focus microscope (Figure 4.4) that instead uses a picosecond mode-locked dye laser beam, also pulsing at 76 MHz and having 640 nm center excitation wavelength. These optical setups (either Figure 4.3 or Figure 4.4) both split a single pulse into a train of four pulses separated by the delay introduced by each arm. Each delay is set in such a way that the 2nd, 3rd, and 4th pulses arrive at the sample separated by 3.3 ns, 6.6 ns, and 9.9 ns, respectively, after the arrival of the 1st pulse. The 1st pulse of the next train then arrives 3.3 ns after the previous 4th pulse, completing the 13.2 ns interval corresponding to the base 76 MHz cycle timing. The new configuration shown in Figure 4.4 provides improved intensity control over each individual beam to equalize powers by changing Neutral Density (ND) filters.

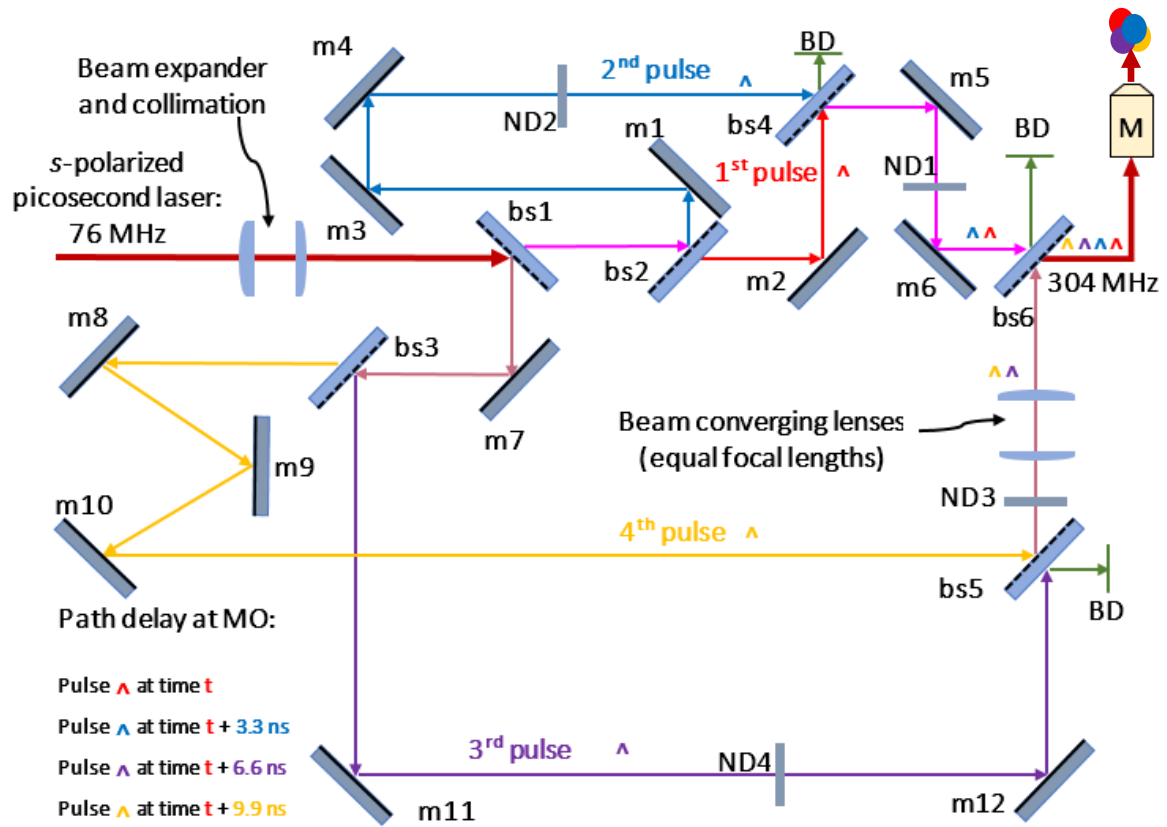


Figure 4.4. Improved optical setup for four foci generation at 304 MHz.

bs: Thorlabs BSS10 beam splitter, m: mirror, ND: neutral-density filter, BD: beam dump, M: microscope objective (water immersion Olympus UPLSAPO 60X, NA = 1.2).

In deriving the four beams, first, we have to make sure that the initial beam is expanded and collimated to have about 0.6 mm beam diameter using “Beam expander and collimation lenses” as shown in Figure 4.4. This collimated beam is then passed through a series of beam splitters to derive four laser beams with proper delays at the microscope objective. At this step of the alignment, the four beams have to be collinear with the optical axis (Z-axis) of the microscope objective so that they all focus to the same point after the objective. Then, placing the “Beam converging lenses” as shown in the Figure 4.4, and adjusting the separation between the lenses we make the Focus3 and Focus4 to be offset by 3.2 μm in Z-axis relative to the Focus1 and Focus2. All four beams should still be collinear and focusing in the Z-axis after going through the microscope objective. The lateral positions of Focus1, Focus2, Focus3, and Focus4 can then be set by adjusting the tilt angles of the mirrors m2, m4, m12, and m10, respectively, while viewing their images on the camera as shown in Figure 4.2.

Data Acquisition and Feedback

In the trapping experiment, the fluorescence generated by a nanoparticle in the four foci is collected by a custom-built confocal microscope that incorporates a pinhole to pass the fluorescence for single-photon counting (see Figure 4.5). The fluorescence that passes through the pinhole is then split at a polarizing beam splitter and focused onto two SPADs. The fast timing signals

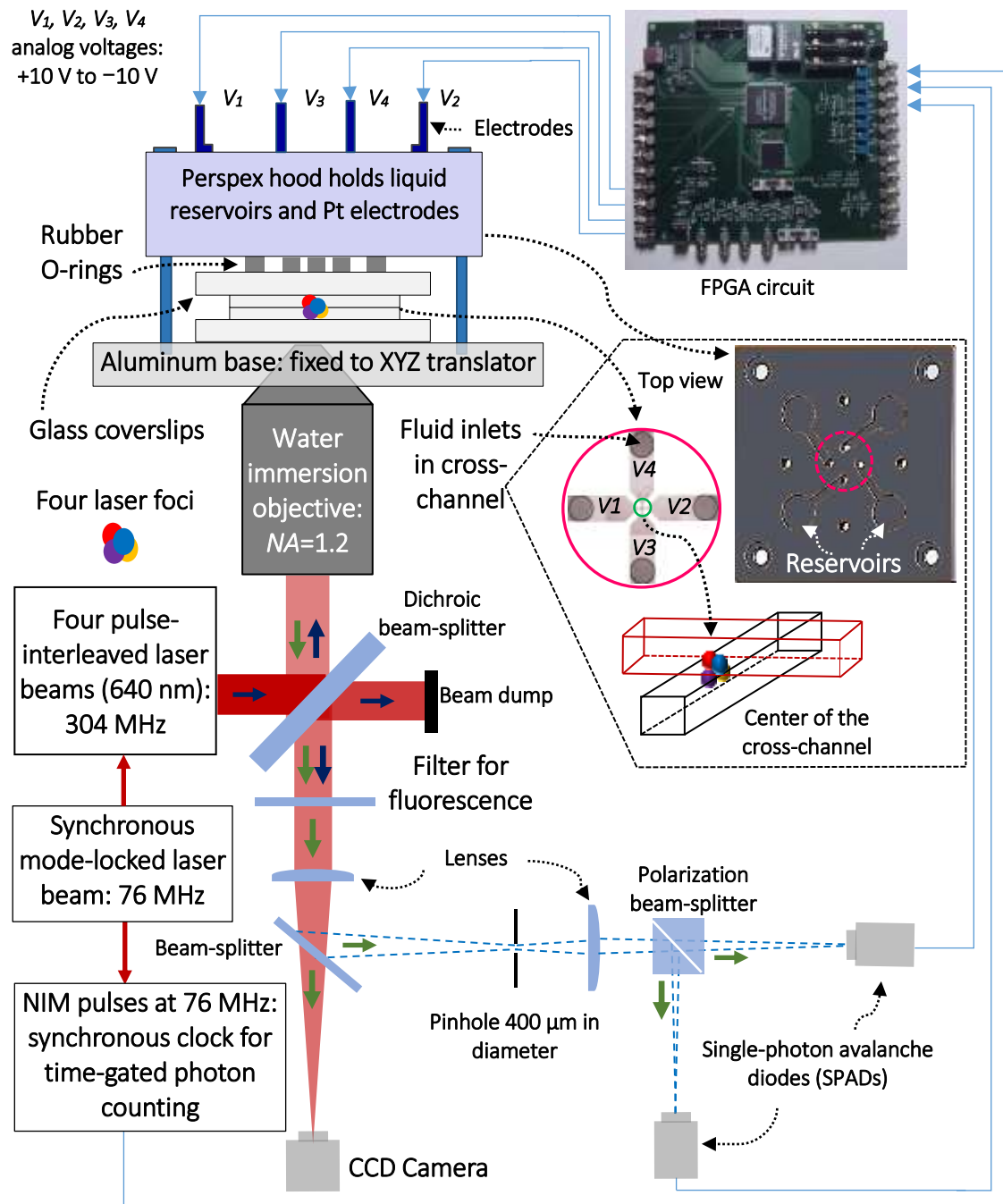


Figure 4.5. Functional diagram of the trapping experiment.

from the SPADs (i.e., negative NIM pulses: NIM is Nuclear Instrumentation Module and NIM pulses are a standard pulse of well-defined characteristics), and also 76 MHz NIM pulses derived from a photodiode detecting the laser pulses, are fed into a custom circuit board containing an FPGA module to conduct time-gated single-photon counting.

The FPGA board runs an algorithm to adjust voltages across the channels of the cross-channel to trap a molecule/particle. The algorithm for trapping is discussed in detail in Chapter Five. The feedback cycle in the FPGA updates every 13.5 μ s and the time-gated photon counts collected during this time are also recorded for further analysis, such as position estimation of a trapped particle. In addition, a beam splitter prior to the pinhole allows part of the signal to pass to an EM-CCD camera located parfocal to the pinhole to observe real-time movements of particles in the focal region (simultaneously with the trapping). Figure 4.6 shows captioned photographs of the complete experimental system.

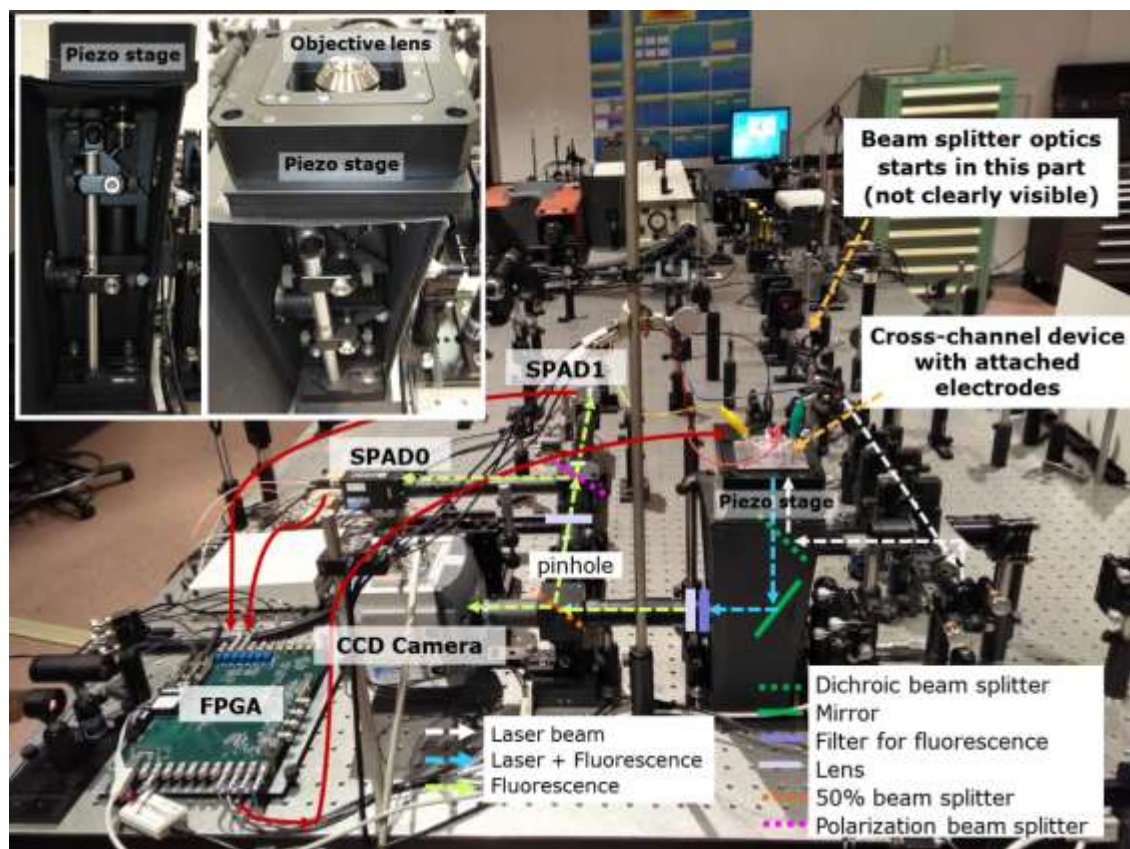


Figure 4.6. Photograph of the experimental setup.

CHAPTER FIVE:

METHODS

Time-Gated Photon Counting

A fluorescently labeled particle that is contained in the microfluidic cross-channel is excited within the tetrahedral excitation region (i.e., the ROI). Fluorescence from the excited particle is collected and used to determine how to manipulate fluid flow in the cross-channel to retain the particle in ROI. The first step in this process is TGPC. The particle undergoes pulse-interleaved laser excitation by four laser beams that are temporally offset by 3.3 ns. Emitted fluorescence photons are detected and sorted into four time bins, each with 3.3 ns width and corresponding to one of the four laser beams, as shown in Figure 5.1.

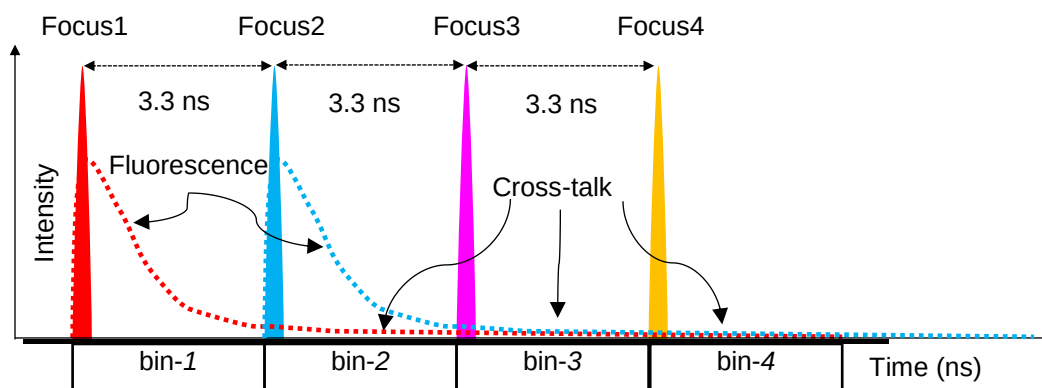


Figure 5.1. Schematic describing pulse-interleaved excitation of fluorescence.

This diagram depicts how the collected fluorescence is assigned into designated photon time bins per laser beam.

Since the laser beams are temporally spaced by 3.3 ns, it is preferred that the particle to be trapped have a fluorescence lifetime less than 3.3 ns, so that the laser beam that initiated the fluorescence photon is correctly identified. When the fluorescence lifetime is comparable to or longer than the time between pulses, a photon generated by one laser beam has a greater probability of being recorded in bins other than the correct one for that beam. This gives rise to cross-talk (see Figure 5.1) and can be estimated theoretically by using Equation 16.

$$p_{ij} = \exp\left[\frac{-t \times n_{ij}}{\tau_f}\right] - \exp\left[\frac{-(t \times (n_{ij} + 1))}{\tau_f}\right] \dots\dots\dots (16)$$

Here, p_{ij} is the probability of a photon generated by a laser pulse in bin- i being counted in bin- j , t is the width of the bins (3.3 ns), τ_f is the fluorescence lifetime, and n_{ij} is the number of bins since the laser pulse arrived ($n_{ij} = j - i$ modulo 4).

The equation shows that shorter lifetimes produce higher probabilities of detecting photons in the correct bins.

The implementation of TGPC is as follows: The photons are detected by two SPADs, which output a voltage pulse for each detected photon (a fast negative-NIM timing pulse of about -1 V into 50 Ω). The SPAD pulses are fed into an FPGA module. Also, the pulses from the laser are detected by a fast photodiode to generate a 76 MHz train of voltage pulses, which are conditioned by a module (Tennelec TC 454 Constant Fraction Discriminator) to give negative-NIM timing pulses. These are fed to the “Sync.” input of the FPGA. The FPGA is

programmed using the Quartus II software (see Appendix B for schematics). The program begins with a phase-locked-loop locked to the “Sync.” input to generate several internal TTL (Transistor-transistor Logic) clock signals. One clock signal is at $4 \times 76 \text{ MHz} = 304 \text{ MHz}$ and is synchronous with the 304 MHz pulse-interleaved excitation provided by the four laser beams. The FPGA is programmed with basic logic elements (AND, OR and NOT) to combine the 304 MHz internal clock with the internal TTL pulses from the SPADs to generate TTL pulses in four channels, thereby sorting or time-gating the signals from photons. The FPGA also generates a clock at $76 \text{ MHz} \div 1024 = 74.2 \text{ kHz}$ with a period of $\sim 13.5 \text{ } \mu\text{s}$. The TTL pulses in each of the four channels (from the time-gated photons) pass to 8-bit counters (range of 0 to 255) which are read and reset every $13.5 \text{ } \mu\text{s}$. The outputs from these four counters, which are the numbers of photons collected in each of the four bins over a period of $13.5 \text{ } \mu\text{s}$, provide the input for the trapping algorithm.

Trapping Algorithm

Every $13.5 \text{ } \mu\text{s}$, the trapping algorithm updates the voltages used for electroosmotic fluid flow on the basis of the number of photon counts collected in the four time bins, labeled as N_1 , N_2 , N_3 , and N_4 . These photon counts are run through a comparison process that is shown in the Table 5.1 to determine the direction of offset of the fluorescent particle from the center, and consequently, in which direction the fluorescent particle is to be moved via electroosmosis in the

Table 5.1. Expected fluid flow directions.

State No.	Expression	Expected Position	Expected Flow Direction (U)
1	$(N_1+N_3) > (N_2+N_4)$	X- region	X+ direction (U_{x+})
2	$(N_1+N_3) < (N_2+N_4)$	X+ region	X- direction (U_{x-})
3	$(N_1+N_3) = (N_2+N_4)$	At the origin	No movement along X (U_{x0})
4	$(N_2+N_3) > (N_1+N_4)$	Y- region	Y+ direction (U_{y+})
5	$(N_2+N_3) < (N_1+N_4)$	Y+ region	Y- direction (U_{y-})
6	$(N_2+N_3) = (N_1+N_4)$	At the origin	No movement along Y (U_{y0})
7	$(N_3+N_4) > (N_1+N_2)$	Z- region	Z+ direction (U_{z+})
8	$(N_3+N_4) < (N_1+N_2)$	Z+ region	Z- direction (U_{z-})
9	$(N_3+N_4) = (N_1+N_2)$	At the origin	No movement along Z (U_{z0})

next feedback cycle in order to direct it back towards the center of the trap. The directions $X_{\pm}$, $Y_{\pm}$, and $Z_{\pm}$ are the same as described in Figure 2.4.

For the 9 independent states that are shown in Table 5.1, there are 27 possible combinations to define the vectorial direction of fluid flow, such as (X+ direction, Y+ direction, Z+ direction), (X+ direction, Y+ direction, Z- direction), etc. All 27 combinations of flow directions and the related voltages V_1 , V_2 , V_3 , and V_4 are listed in Table 5.2. The physical connections of V_1 – V_4 with the cross-channel are depicted in Figure 3.3.

In order to perform rapid analysis on the photon counts, set voltage values to drive the fluid flow, and record data, a suite of software modules was developed using the Verilog programming language (via Quartus II) and implemented on the FPGA integrated chip (IC, Altera Cyclone II). First, the four 8-bit binned photon counts N_1 – N_4 are used to determine voltage values V_1 – V_4 by

Table 5.2. Flow directions vs. voltages V_1 , V_2 , V_3 and V_4 .

Flow Direction State No.	Flow Direction Combinations	V_1 (V)	V_2 (V)	V_3 (V)	V_4 (V)
1	(U_{x+} , U_{y+} , U_{z+})	0	-10	10	0
2	(U_{x+} , U_{y+} , U_{z-})	10	0	0	-10
3	(U_{x+} , U_{y+} , U_{z0})	10	-10	10	-10
4	(U_{x+} , U_{y-} , U_{z+})	0	-10	0	10
5	(U_{x+} , U_{y-} , U_{z-})	10	0	-10	0
6	(U_{x+} , U_{y-} , U_{z0})	10	-10	-10	10
7	(U_{x+} , U_{y0} , U_{z+})	0	-10	5	5
8	(U_{x+} , U_{y0} , U_{z-})	10	0	-5	-5
9	(U_{x+} , U_{y0} , U_{z0})	10	-10	0	0
10	(U_{x-} , U_{y+} , U_{z+})	-10	0	10	0
11	(U_{x-} , U_{y+} , U_{z-})	0	10	0	-10
12	(U_{x-} , U_{y+} , U_{z0})	-10	10	10	-10
13	(U_{x-} , U_{y-} , U_{z+})	-10	0	0	10
14	(U_{x-} , U_{y-} , U_{z-})	0	10	-10	0
15	(U_{x-} , U_{y-} , U_{z0})	-10	10	-10	10
16	(U_{x-} , U_{y0} , U_{z+})	-10	0	5	5
17	(U_{x-} , U_{y0} , U_{z-})	0	10	-5	-5
18	(U_{x-} , U_{y0} , U_{z0})	-10	10	0	0
19	(U_{x0} , U_{y+} , U_{z+})	-5	-5	10	0
20	(U_{x0} , U_{y+} , U_{z-})	5	5	0	-10
21	(U_{x0} , U_{y+} , U_{z0})	0	0	10	-10
22	(U_{x0} , U_{y-} , U_{z+})	-5	-5	0	10
23	(U_{x0} , U_{y-} , U_{z-})	5	5	-10	0
24	(U_{x0} , U_{y-} , U_{z0})	0	0	-10	10
25	(U_{x0} , U_{y0} , U_{z+})	-10	-10	10	10
26	(U_{x0} , U_{y0} , U_{z-})	10	10	-10	-10
27	(U_{x0} , U_{y0} , U_{z0})	0	0	0	0

the soft module “counts_to_voltages_6_2.” These values are then fed into another soft module named “darpa_dac_2” to command an external voltage-driving IC (LTC2704) to apply the voltages across the channels of the cross-channel to move the fluid accordingly.

Data recording is handled by the soft module named “usb_descriptor,” where the FPGA becomes the external master controlling the process of exporting the binned photon counts from the FPGA to a desktop computer through a USB 2.0 interface (CY7C68001). It should be noted that the Verilog programs for “darpa_dac_2” and “usb_descriptor” follow the digital command instructions that are mentioned in the respective IC manuals, which are available online [62, 63]. The Verilog code for these three soft modules can be found in Appendix C.

A flowchart of the FPGA operation is given in Figure 5.2. As all subsequent logic activity depends on it, TGPC is performed first. In order to interface the two SPAD signals and a reference external synchronous clock (76 MHz), a dedicated circuit that implements low-voltage differential signaling is configured in the FPGA (see Figure B.1 in Appendix B). Two clocks with frequencies 50.67 MHz and 304 MHz are derived from the base 76 MHz synchronous clock. The 50.67 MHz clock provides clock signals to the “darpa_dac_2” and “usb_descriptor” modules, while the 304 MHz clock becomes the main clock for all other components and modules.

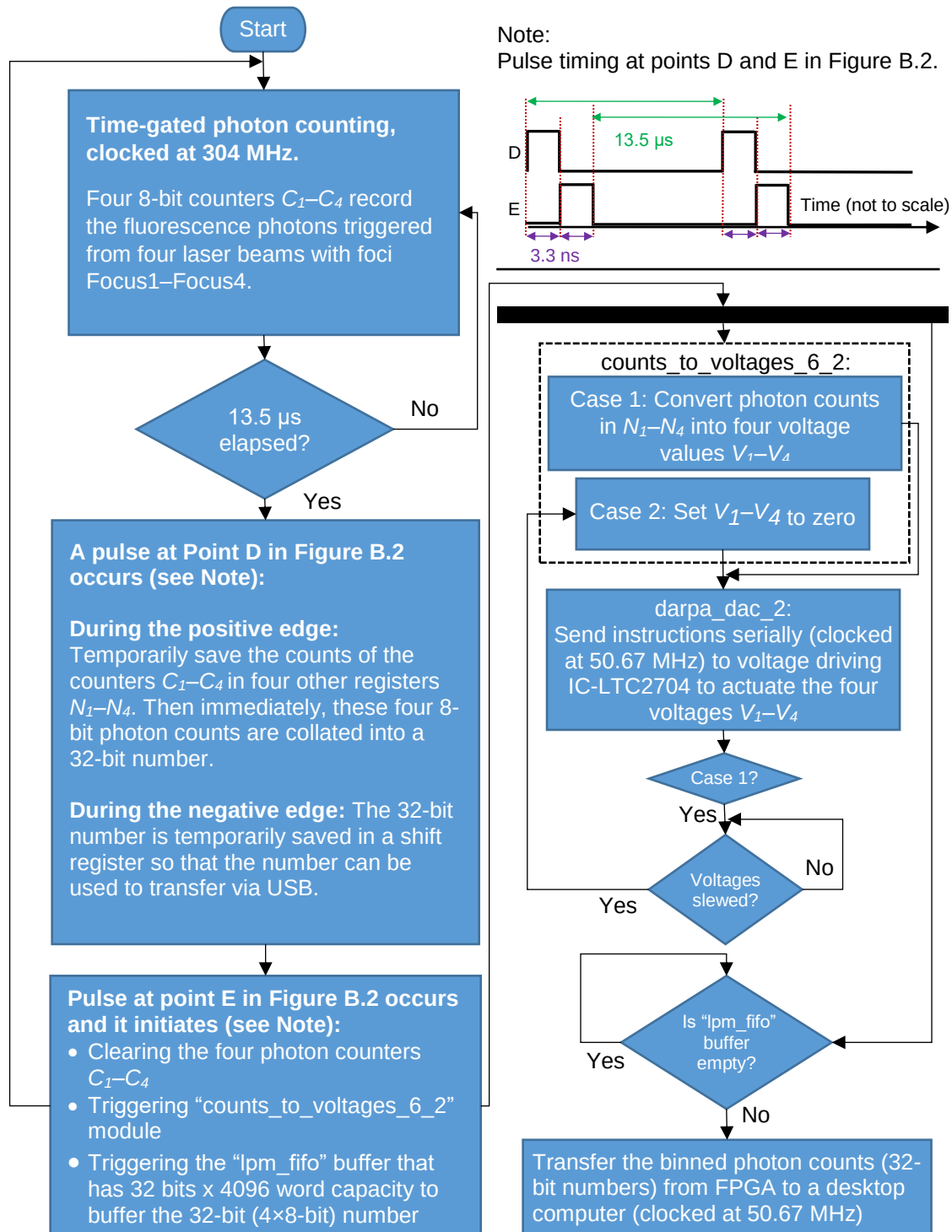


Figure 5.2. Flowchart of the FPGA board operation.

The main reason for deriving a clock at 304 MHz is to perform TGPC at the same frequency as photons are detected. The photon detection logic circuitry and the counters that run the four time gates and determine the feedback cycle time of 13.5 μ s are illustrated in Figure B.2. As shown in the flowchart in Figure 5.2, the primary function of the system is TGPC. This is done according to the logic diagram shown in Figure B.3 with four 8-bit counters C_1 – C_4 . Two-bit binary numbers “00,” “01,” “10,” and “11” trigger the four counters C_1 , C_2 , C_3 , and C_4 , respectively.

When the elapsed counting time reaches the feedback cycle time of 13.5 μ s, the values in the counters are transferred to four 8-bit D (“data”) flip-flops N_1 – N_4 (the values in these registers are also addressed by the same letters), and a 32-bit number is created from the four 8-bit values. The 32-bit number is then temporarily saved in a 32-bit D flip-flop during the negative going edge of the same pulse that initiated the creation of the 32-bit number. (This pulse is at point D in Figure B.2, and it occurs every 13.5 μ s and lasts for 3.3 ns. The initiation was triggered by the positive edge of the pulse.)

During the next pulse (this pulse occurs at point E in Figure B.2 and lasts 3.3 ns) that comes immediately after the pulse at the point D in Figure B.2, three main functions occur: clearing the four photon counters C_1 – C_4 , triggering the “counts_to_voltages_6_2” module, and triggering “lpm_fifo” (a buffer with the capacity of 32 bits x 4096 words) to hold the 32-bit numbers. From the next clock signal (a clock signal from the main 304 MHz clock) onwards, until the system

reaches 13.5 μs feedback cycle time, TGPC is performed during the next feedback cycle, voltages V_1 – V_4 dependent on the photon counts N_1 – N_4 are applied to the cross-channel, and the photon counts from the previous feedback cycle are transferred via USB interface to a desktop computer. Before applying the potentials, the “counts_to_voltages_6_2” module converts the photon counts into 16-bit voltage values and then triggers the “darpa_dac_2” module to communicate serially with the LTC2704 to apply them. If the total photon count is greater than two (i.e., $(N_1 + N_2 + N_3 + N_4) > 2$), a particle is assumed to be in the trap; otherwise, the voltages drive the fluid to bring particles towards the ROI. The “darpa_dac_2” module also commands the potential values to be reset to zero immediately after applying the required voltages derived from the binned photon counts per feedback cycle time. Resetting the voltages to be zero ensures that the voltages start rising/declining from zero in the next feedback cycle and it also reduces the chance of driving a trapped particle out from the trapping ROI.

The LTC2704 IC used in this process has four digital-to-analog (DAC) converters, namely DAC-A, DAC-B, DAC-C and DAC-D. The commands that are required to address those DACs have to be sent to the IC sequentially at a relatively low clock rate of 50.67 MHz (compared to the 304 MHz main clock). It takes $\sim 5 \mu\text{s}$ to load all the commands successfully onto the LTC2704. This IC also exhibits a slew rate of $\sim 10 \text{ V}$ per $3.4 \mu\text{s}$. Thus, altogether these timings ($2 \times$ setting instructions to DACs and time for slewing (see Figure 5.3)), add up to

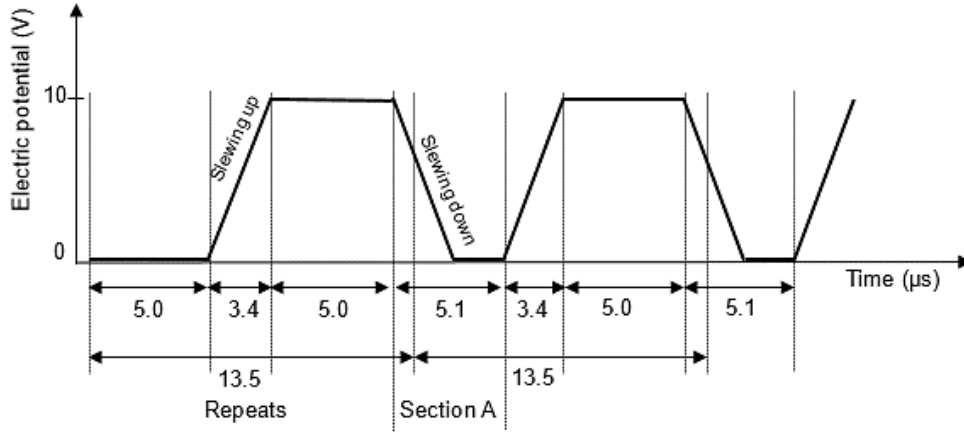


Figure 5.3. Timing diagram of voltage output by one DAC of the LTC2704.

Section A includes slewing down the potential to zero while loading the next set of instructions into the DACs of the LTC2704 IC via “darpa_dac_2”. The slewing of the potential is independent from loading the instructions into the DACs. All four DACs operate identically.

about 13.4 μs and enable us to define a digital clock for the feedback cycle duration using the 12th bit of a digital counter that triggers every 3.3 ns to have $\sim 13.5 \mu\text{s}$ cycle time (i.e., $2^{12} \times 3.3 \text{ ns} \approx 13.52 \mu\text{s}$).

Due to the fact that the slewing of the DACs becomes an independent task after receiving the commands from the FPGA, the feedback cycle time could be reduced, but the digital bit that can be selected from a counter to define a clock for the feedback cycle duration still becomes the 12th bit (i.e., if we select the 11th bit then the feedback cycle becomes $2^{11} \times 3.3 \text{ ns} \approx 6.76 \mu\text{s}$, which is less than 10 μs). Also, if we choose not to reset the voltages to zero in each feedback cycle, the feedback duration time could even be shortened down to 5 μs .

However, the number of photons counted during a 5 μs time duration would not be enough for the feedback algorithm to decide if the particle had escaped from

the trap. Considering all of those factors, it was decided that $\sim 13.5 \mu\text{s}$ would be an appropriate time duration for the feedback cycle.

Position Estimation

The trapping algorithm discussed so far only reacts according to the photon counts recorded in the four bins every $13.5 \mu\text{s}$. Position estimation of a trapped particle is not performed in real-time in the trapping algorithm due to the limitations of the FPGA used. In fact, the exact position of the trapped particle is not a key factor as long as trapping is in progress. However, post-process analysis of recorded photon counts from each bin can produce a set of position estimation data, so it is possible to derive a visualization of how the trapped particle moves within the ROI. In other words, the path the particle travelled while it was trapped, which includes Brownian diffusion and the electroosmotic corrections applied, can effectively be mapped.

In order to estimate the positions, an MLE (Maximum Likelihood Estimation) of positions within the limits imposed by a filter is performed via a MATLAB program (see Appendix D). The MLE uses a Poisson probability distribution approximation for the number of fluorescence counts in each bin and calculates the most probable position of the particle by taking the binned photons N_1 , N_2 , N_3 , and N_4 into account. The probability P of producing N_1 – N_4 photons when the particle is at position (x, y, z) is given by product of the Poisson probability distributions in Equation 17,

$$P(N_1+N_2+N_3+N_4|x,y,z) = \prod_{i=1}^4 \frac{\exp(-R_i) \times R_i^{N_i}}{N_i!} \dots\dots\dots (17)$$

with

$$R_i(x,y,z) = \left[\sum_{j=1}^4 p_{ij} F_j \right] \times C_0 + C_B \dots\dots\dots (18)$$

where $F_j(x,y,z)$ is the Gaussian-Lorentzian intensity distribution for beam j , C_0 is the maximum photon count per beam per binning time interval when the fluorescent particle is at the center of a beam, and C_B is the average background photon count per bin per binning interval.

The quantity usually maximized in MLE is the natural log of the probability, so for each iteration (i.e., each binning time period) the estimated position (x',y',z') is obtained from the domain of position values (x,y,z) that best satisfies Equation 19:

$$(x',y',z') = \operatorname{argmax} \left[\sum_i N_i \ln(R_i(x,y,z)) - R_i(x,y,z) \right] \dots\dots\dots (19)$$

The MLE positions (x',y',z') go through a limiting process similar to a Kalman filter so that the distance that the particle jumps in any one time step (Δx , Δy , or Δz) is within what diffusion permits (i.e., $\Delta x \leq \sqrt{2Dt}$; the same applies for Δy and Δz). This filtering prevents unexpected changes in the estimated position, which can occur due to various reasons, such as a second particle entering the ROI to produce photons from one or two beams.

CHAPTER SIX:

EXPERIMENTAL RESULTS AND ANALYSIS

Evidence of Trapping

The trapping of particles was observed using two methods: CCD camera imaging and fluorescence photon counting. When a particle is being trapped, a bright white spot on a black background can be seen within the camera (see Figure 6.1). In terms of photon counting, the fluorescence count rate suddenly jumps, from nearly zero to a considerably higher value, and it stays at the higher value as long as the particle is being trapped. In some experiments, only one observation method was set up to record data, although photon counts were more often recorded as this method allows tracking the motion of the particle in the 3D Cartesian coordinate system. The CCD movies show the X-Y coordinates of the trapped particle, but image analysis for determining the Z coordinate from the defocus of the image is not so straightforward and was not performed.



Figure 6.1. CCD camera images to show evidence of trapping of a fluorescent particle. Left: particle is trapped. Middle: particle is about to escape the trap (escape is along the Z-axis so that the image is defocused). Right: particle has fully escaped from the trapping region.

Table 6.1. Andor™ iXon 897 EM-CCD camera and SPAD settings.

Device	Property	Value	Unit
CCD Camera	Image magnification (M)	83.3	–
	Square pixel length (l)	16	μm
	EM gain range	1 to 1000	–
	Video frame rate: 1/exposure-time (t_e)	$1/(0.0291 \text{ s}) \approx 34.36$	Hz
SPAD	Photon detection efficiency	~ 65	%
	Timing jitter	127	ps
	Dead time	40	ns
	Afterpulsing probability	0.05	%
	Afterpulsing decay time	100	ns

During the video recordings, the CCD camera (Andor™ iXon 897 EM-CCD) was used with the settings shown in Table 6.1.

The object undergoes an optical magnification of about 83.3 before it is imaged onto CCD camera. The electron- multiplying (EM) gain setting was varied in the range from 3 to 120 depending on the gain needed for clear imaging. In taking videos in the dynamic mode of the camera, the frame rate was 34.36 Hz as determined from the exposure time setting. The movies were saved to files in “.sif” format (proprietary to Andor) and were converted to “.avi” and “.mp4” files with matched frame rate so that the converted videos run in real-time. Some important properties of the SPADs [64] are also shown in Table 6.1.

Measured Electroosmotic Speeds

For the fabricated cross-channel, electroosmotic speeds along the X-, Y- and Z-axes were predicted via COMSOL. However, these values will depend on

the actual channel dimensions and zeta potential, hence it is beneficial to measure the electroosmotic speeds experimentally. Since the particles are to be trapped in the center of the cross-channel, the measurements are performed only in this region of the cross-channel.

In order to measure the electroosmotic speeds along the X -axis, a different program was developed in the FPGA to switch the voltages V_1 and V_2 between zero and the values: $V_1 = +5$ V, $V_2 = -5$ V, while holding $V_3 = 0$, $V_4 = 0$ V. Note that V_1 and V_2 were set pulsing with a pulse width (t_w) of 36 μ s while the cycle time (t_c) is about 52 μ s. This pulsation provides +10 V across the channel that is placed along the X -axis and particles only tend to move along the $X+$ direction. The movement of particles in solution near the center of the cross-channel was captured using the iXon camera for an area enclosed by a square frame of 128 pixels per side (which corresponds to $128 \times 16 \mu\text{m} / 83.3 = 24.6 \mu\text{m}$ per side in object space).

Successive images of a particle that travelled along the $X+$ direction (in a horizontal line) for five frames were used for measuring the speed (see Figure 6.2). The total distance that the particle travelled (S_x) is taken as the product of the number of pixels traveled during five sequential frames (N_{px}) and pixel size (l_x) in microns, divided by the image magnification (M) (i.e., $S_x = (N_{px} \times l_x) / M$). The effective time ($T_{effective}$) that the particle was travelling from the first to the fifth frame is the time that the voltages were turned on during the last four frames and

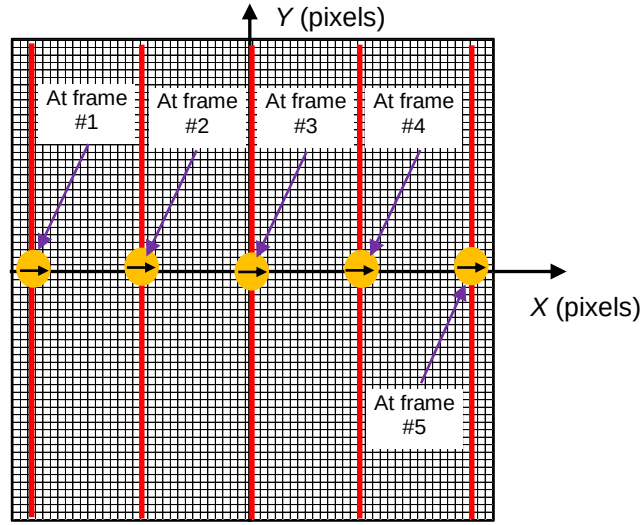


Figure 6.2. Schematic of camera images to estimate electroosmotic speeds (not to scale).

is estimated as $T_{effective} = (4 \times t_e) \times (t_w / t_c) = 80.6 \text{ ms}$. The electroosmotic speed in the X-direction $(v_{eo})_x$ for a 20 V voltage, as used in the trapping experiment and the COMSOL analysis, can then be calculated according to $(v_{eo})_x = 2 \times S_x / T_{effective}$, where the factor of 2 accounts for the use of only 10 V voltage in the measurement. We apply the same procedure to measure the speed along the Y-axis, but with voltage values $V_1 = 0 \text{ V}$, $V_2 = 0 \text{ V}$, $V_3 = +5 \text{ V}$, and $V_4 = -5 \text{ V}$. Note that the speed along the Z-axis is estimated to be one-fourth of the speed along the X-axis (or Y-axis), as found in the result of the COMSOL simulation. Table 6.2 summarizes the experimentally measured and estimated electroosmotic speeds at the center of the cross-channel along each of the axes. The quoted error ($\sim 1.7\%$) is that due to a one pixel error in the total distance moved and does not account for other factors.

Table 6.2. Experimentally measured and estimated electroosmotic speeds.

Axis	Pixels N_p	Distance (μm) $S = (N_p \times l) / 83.3$	Electroosmotic Speed ($\text{mm}\cdot\text{s}^{-1}$) $(v_{eo})_{x,y} = 2 \times S_{x,y} / T_{\text{effective}}$	Error ($\text{mm}\cdot\text{s}^{-1}$)
X	61	11.72	$(v_{eo})_x \approx 0.293$	0.005
Y	62	11.91	$(v_{eo})_y \approx 0.298$	0.005
Z	--	--	$(v_{eo})_z \approx ((v_{eo})_x + (v_{eo})_y) / 8 \approx 0.074$	0.001

Trapping 40 nm FluoSpheres

The developed ABEL trap has been tested for 40 nm diameter deep red FluoSpheres (fluorescence lifetime 2.2 ns and 660/680 nm excitation/emission) dispersed in distilled water (concentration ~ 100 pM), which is the medium considered for optimization of the electroosmotic speeds in the cross-channel. For 40 nm FluoSpheres with a diffusion coefficient of about $12.1 \mu\text{m}^2\cdot\text{s}^{-1}$ (at 293 K), it was observed that trapping could continue for more than one minute. During the trapping experiment, the laser power per beam—for 640 nm excitation wavelength—was set at $1 \mu\text{W}$ and the beams were kept at the separations shown in Figure 4.1.

Image Data of Trapping for over 4 Minutes

The results for 40 nm FluoSpheres gave clear evidence of trapping with a very significant increase of the residence time of the particle in the ROI. If the trap was turned off by interrupting the feedback (for example, by momentarily blocking the laser beams), the particle would immediately escape. Whenever a

40 nm particle was trapped, it stayed in the trap for considerably long times (>1 minute), indicating that the trap could be very useful for studying biological aggregates of similarly sizes. In one experiment, the Fluosphere remained in the trap for over four minutes (see Video 1), which is a remarkable result. This is to our knowledge the first report of such a long trapping time by a 3D ABEL trap for 40-nm-sized particles.

Figure 6.3 shows selected images taken from Video 1 as time elapses. The starting point of the video is marked as time = 0 and the selected frames show clear trapping of the particle for 4 minutes and 12 seconds. Note that the video recording was actually started sometime after the particle had already become trapped.

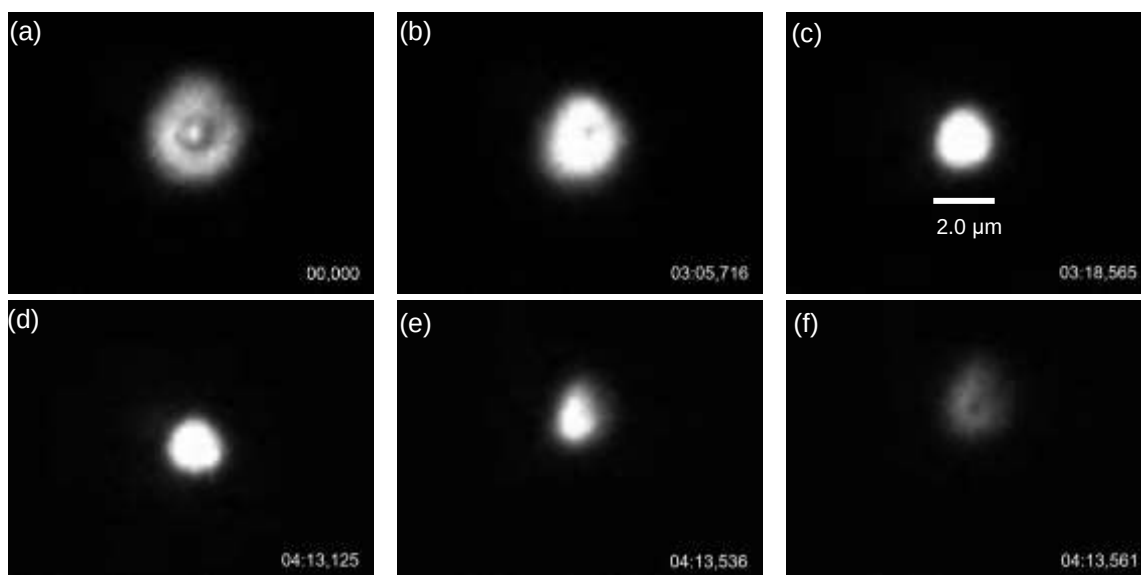


Figure 6.3. Image data for trapping a 40 nm FluoSphere in distilled water for >4 minutes.

Images (each having 64×49 square pixels) were taken from Video 1 and show the time elapsed in the image at lower right (format: 02:34,235 means two minutes, thirty-four seconds and two hundred thirty-five milliseconds). (a) CCD camera is out of focus to the particle. (b) CCD camera was being focused to the particle. (c) Camera focused to the particle. (d) Particle remains trapped. (e) Particle was about to escape from the trapping region. (f) Particle escaped from the trap.

Photon Count Data for Trapping of ~1 Minute

In a different trapping instance (i.e., not the result discussed in the section above) fluorescence photon counting data was collected for about 1 minute, as shown in Figure 6.4. The figure depicts the number of photons/100 ms vs. time in seconds for each of the four laser foci (Focus1–Focus4). Note that the photon recording was initiated for an already-trapped particle and hence the actual trapping time was longer than the time shown in the figure. While the particle is in the trapped state, it emits about 6×10^6 photons·s⁻¹. At about 57 s, the total count rate suddenly drops to close to the background rate of about 10^5 photons·s⁻¹, indicating that the particle has escaped from the trap. However, for a few seconds, there are still more photon counts from Focus3 and Focus4 than from Focus1 and Focus2, perhaps indicating that the particle is escaping in the Z-direction, and then the counts from all foci drop to the background levels, indicating that the escape is complete. The two states, trapped and escaped, can be clearly distinguished from the difference in the total photon counts.

The MLE of x , y , and z positions vs time for the data in Figure 6.4 is shown in Figure 6.5. Figure 6.6 below illustrates the 3D scatter of the estimated positions. Assuming that the harmonic potential well approximation and the equipartition theorem hold for this trap, it is possible to estimate the trap's effective spring constants— k_x , k_y , and k_z —from the standard deviations σ_i , $i = x, y, z$ of the histograms of the recorded positions, shown to the right in Figure 6.5. The effective spring constants are equal to $k_B T / \sigma_i^2$, where k_B is the

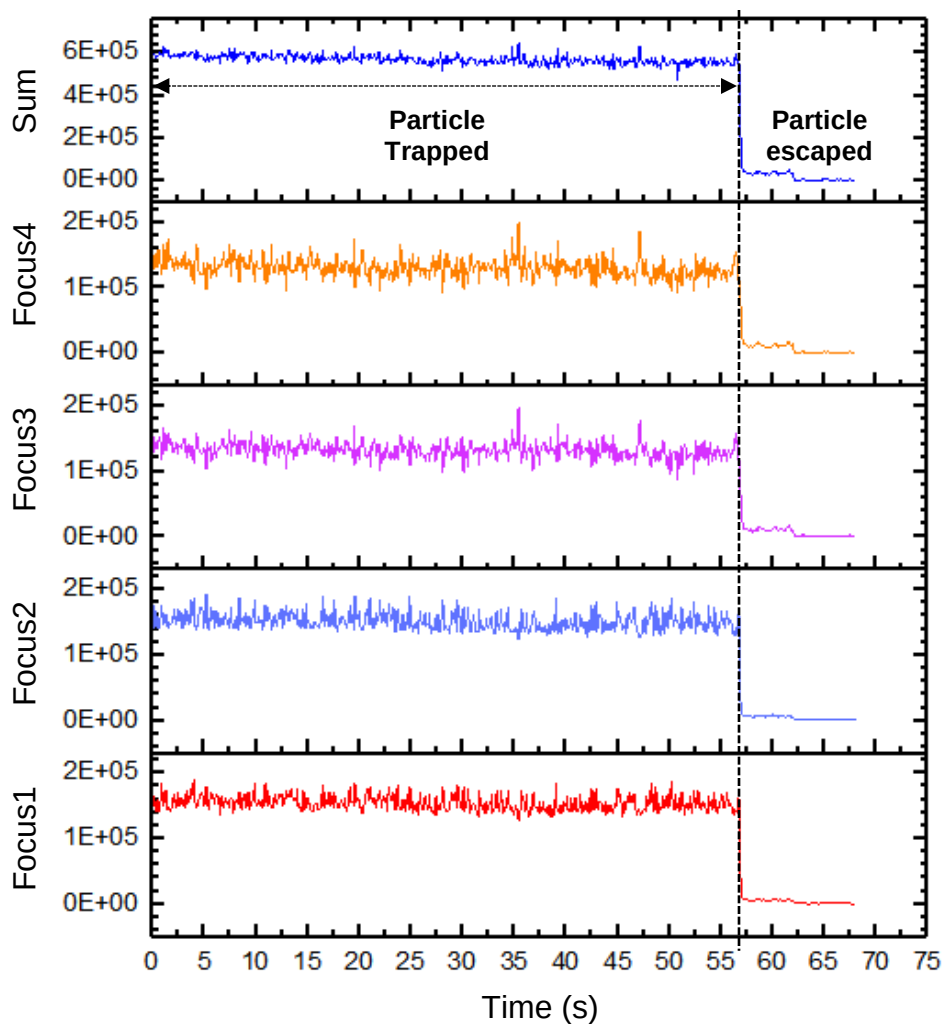


Figure 6.4. Trapping a 40 nm FluoSphere in distilled water.

Charts show photon counts (photons/100 ms) vs time for Focus1–Focus4 and the sum of counts from all four foci with 1 μ W power per beam. Note that it is clear when the particle escapes from the trap, as the counts drop suddenly (dashed vertical line at ~57 s).

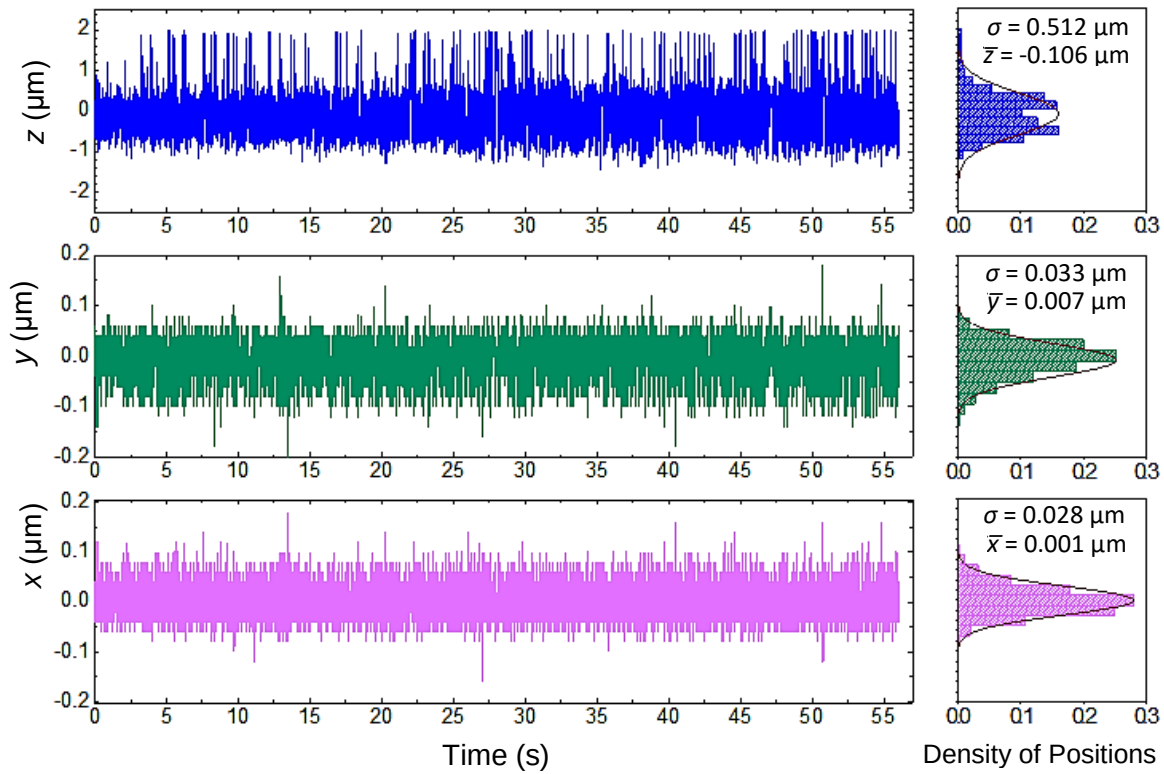


Figure 6.5. MLE of positions vs. time for the data in Figure 6.4.

The position variations only show about 57 seconds of position data for the trapped 40 nm FluoSphere shown in the Figure 6.4. A jumping average of 1 ms photon count data was taken for each position determination. The histograms at right show the probability density of positions with respective standard deviations σ and mean values $\bar{x}$, $\bar{y}$, $\bar{z}$. Note that the bin size of the histograms is 0.02 μm for x - and y -positions and 0.2 μm for z .

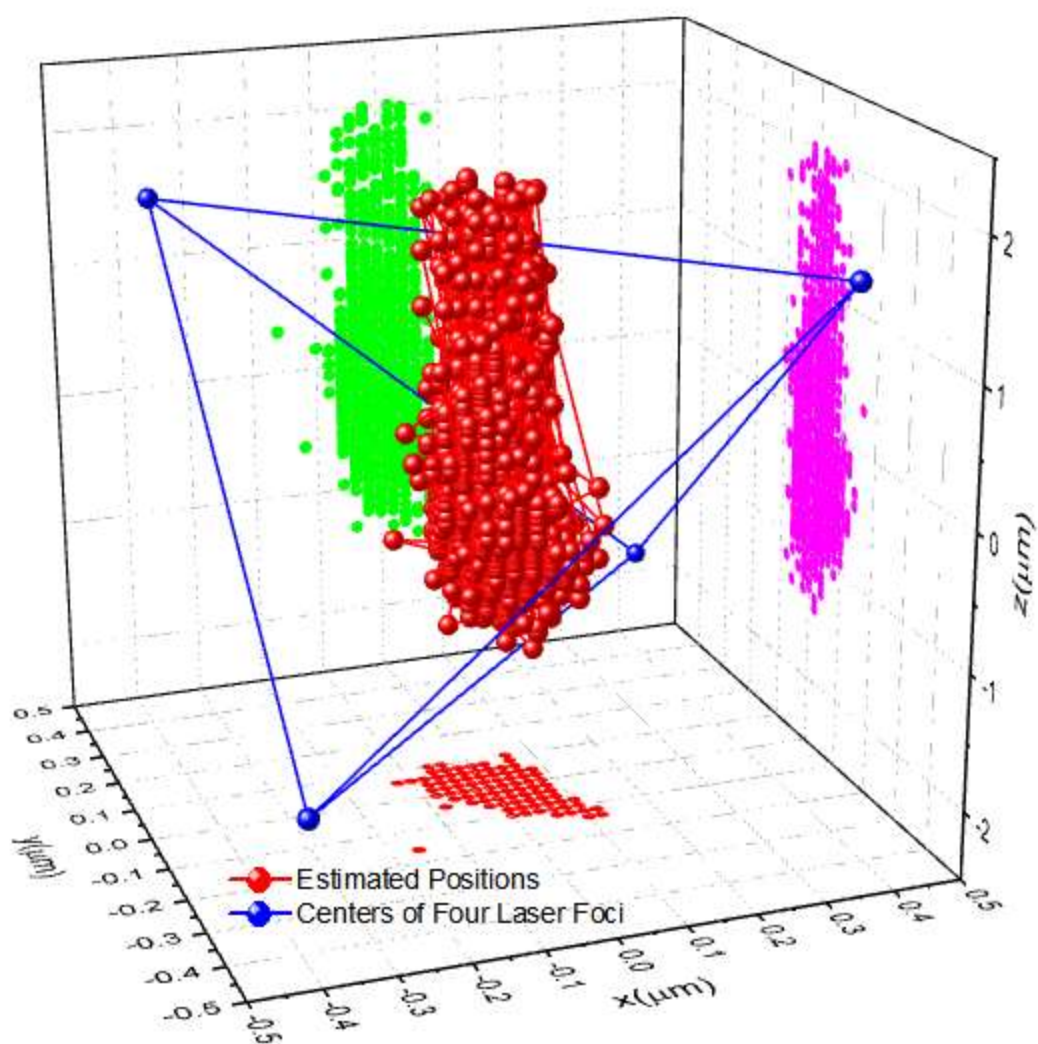


Figure 6.6. 3D plot of MLE positions of the 40 nm FluoSphere trapped for ~57 seconds.
(Data from Figure 6.5.)

Boltzmann constant and T is absolute temperature. The potential energy of a particle in a harmonic well with spring constant k is given by $k\sigma^2/2$, while according to the equipartition theorem, the energy of a particle per degree of motion is $k_B T/2$. By equating those energies and simplifying we obtain the equation $k_i = k_B T/\sigma_i^2$ for the effective spring constants for each Cartesian axis i . Then the estimated effective spring constants k_x, k_y , and k_z for a 40 nm particle become $5.0 \mu\text{N}\cdot\text{m}^{-1}$, $4.0 \mu\text{N}\cdot\text{m}^{-1}$ and $0.02 \mu\text{N}\cdot\text{m}^{-1}$, respectively.

Trapping 20 nm FluoSpheres

For a 20 nm FluoSphere the diffusion coefficient is about $24.1 \mu\text{m}^2\cdot\text{s}^{-1}$ (at 293 K) and it was observed experimentally that a 20 nm particle could be trapped for as long as ~25 seconds. However, because the 20 nm FluoSpheres are smaller and thus comparatively less bright than the 40 nm version, the trap was operated with ~12 μW excitation power per beam—for 640 nm excitation wavelength—in order to obtain a good Signal to Noise Ratio (SNR). The photon count data look similar to the one obtained for 40 nm FluoSpheres with slightly more fluctuating photon count rates.

Image Data of Trapping of ~25 Seconds

In the 20 nm FluoSphere trapping situation, the maximum trapping time recorded so far was about 25 seconds (see Video 2), which is an unprecedented result for the 20 nm scale. This particular result was recorded during an

experimental session that lasted for about 2 hours. During this session, there were several trapping instances observed with almost the same trapping time. The data has not been statistically analyzed to provide the mean trapping time as further improvements for the trapping device and the algorithm are possible. Figure 6.7 shows selected images from Video 2 as time elapses, including images within the first 1.8 seconds of the recording in which another particle also enters the trapping ROI and almost disrupts the trapping before diffusing away.

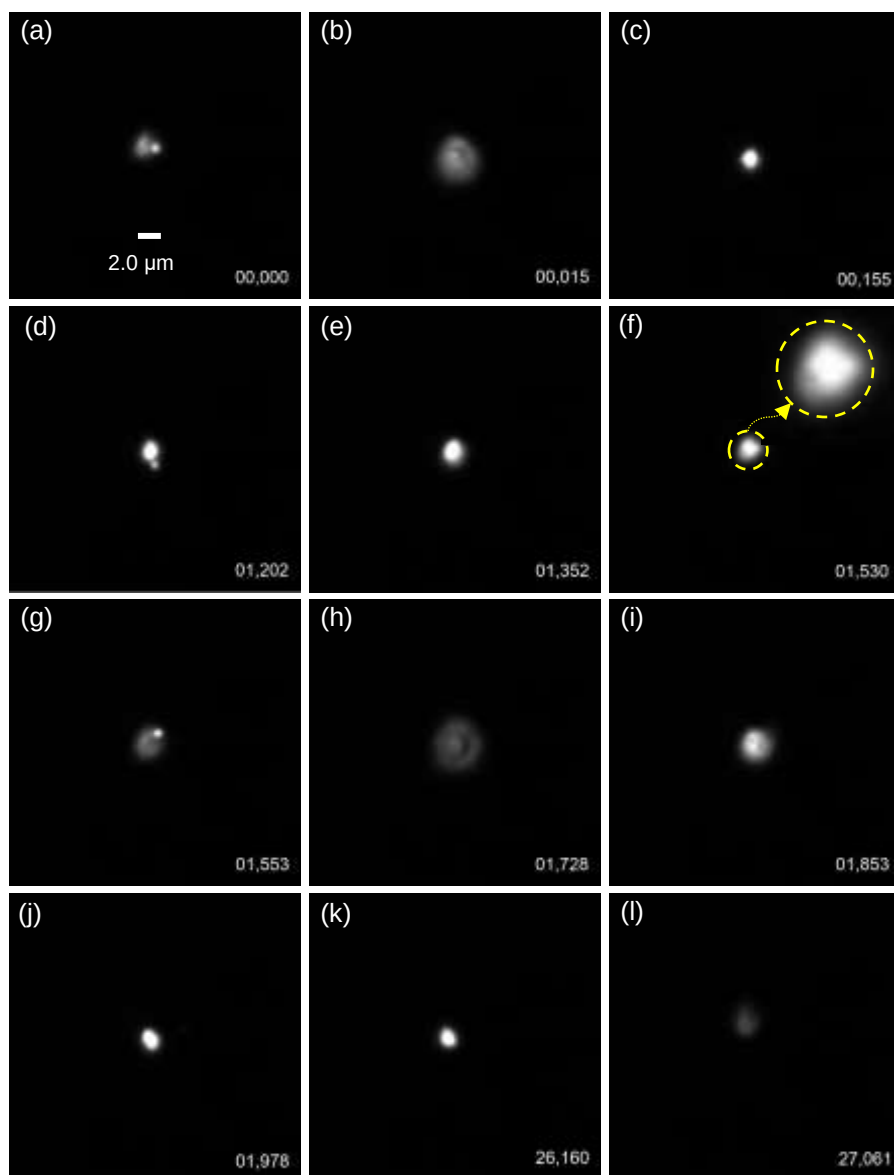


Figure 6.7. Image data for trapping a 20 nm FluoSphere in distilled water for ~25 seconds. Images (each having 128×125 square pixels) were taken from Video 2 and show the time elapsed in the image at lower right (format: 34,235 means thirty-four seconds and two hundred thirty-five milliseconds). (a) Two particles A, B; A is blurred and stays out of focus, B is brighter and stays behind A, close to focus. (b) A and B aligned along the Z-axis, but A is still closer to camera. (c) A comes into focus, B stays behind A in Z-axis. (d) A is in focus, but B moved to Y-. (e) A is in focus, B stays behind A in Z-axis; (f) A and B show a slight separation in X-Y plane, but B is still behind A. (g) A goes out of focus and B is still be seen behind A. (h) A and B aligned in Z-axis, A is still out of focus and B stays behind A. (i) A tends to be in focus and B has moved out from the laser excitation. (j) A is trapped perfectly. (k) A is still trapped, but begins to break away from the trap. (l) A escapes from the trap.

Photon Count Data of Trapping for over 5 Seconds

As was shown in Figure 6.7 above, during the process of trapping, it is possible that particles from the surrounding solution can enter the trapping region, either to the center or near the edges. Such occurrences, which we term invasions, depend on the concentration of the solution and occur infrequently for concentrations of ~ 100 pM and lower. An invasion event was captured clearly in the photon count data during the trapping of a 20 nm FluoSphere, as shown in Figure 6.8. At ~ 0.4 s, a spike in photon counts was recorded, indicating the temporary contribution from an invader FluoSphere. The invader approached closer to Focus3 and Focus4 (note that Focus1 and Focus2 do not show such clear evidence of the spike), and then quickly diffused away from the trap ROI, as revealed by the count rate rapidly dropping to a relatively stable level of roughly half the spike's peak value.

Although the probability of an invasion could be reduced by further lowering the concentration of particles in the solution, this would increase the waiting time for a particle to enter the trapping ROI when initially loading the trap, which is currently >1 minute. Long wait times are in fact a drawback of the presently implemented trapping algorithm, which relies on diffusion for a particle to initially enter the trap. Also, in another experimental run (see Figure 6.9), a trapped particle appears to momentarily escape from the trap causing the photon counts to decrease, although not evenly from all foci, and the trap then manages to re-capture it, highlighting some of the statistical features of the trap.

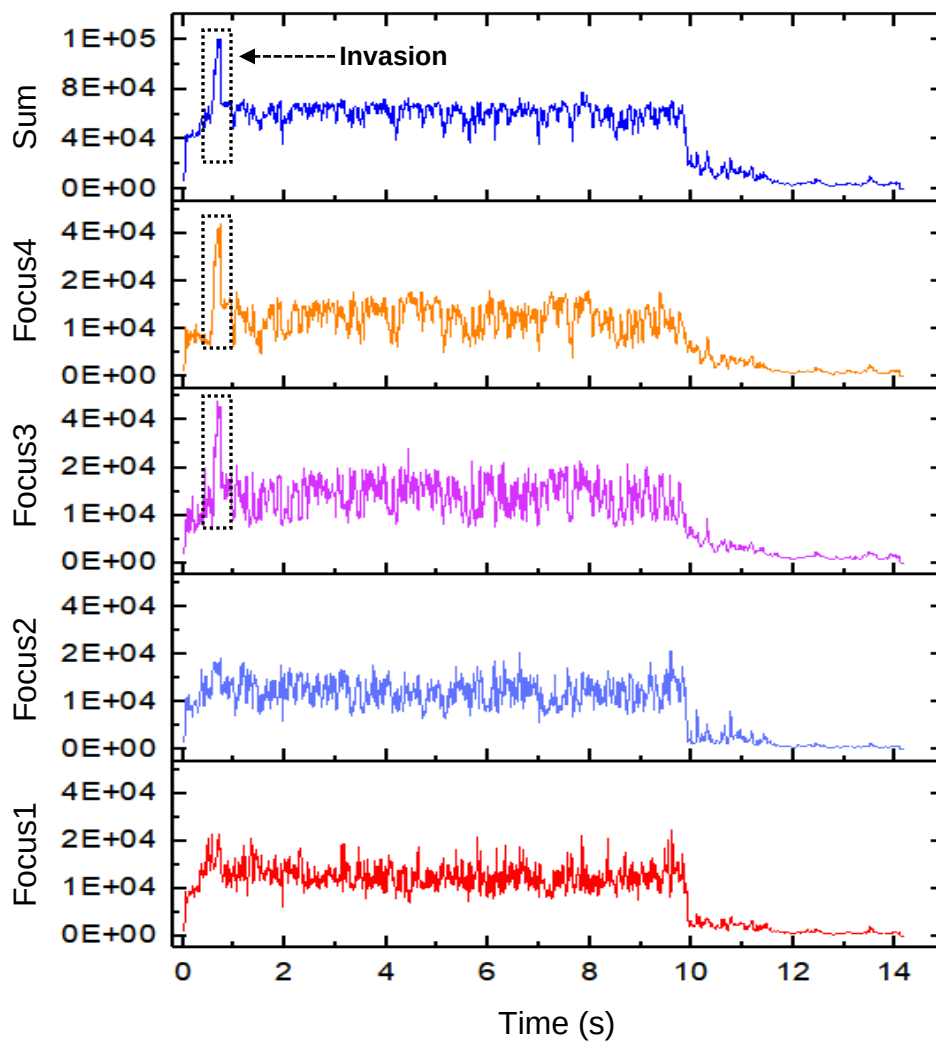


Figure 6.8. Trapping a 20 nm FluoSphere in distilled water (I).

Charts show photon counts (photons per 10 ms) vs time for Focus1–Focus4 and the sum of counts from all four foci with 9 μ W power per beam. The dotted boxes highlight a momentary invasion of second particle around Focus3 and Focus4. The original particle escapes at ~10 seconds.

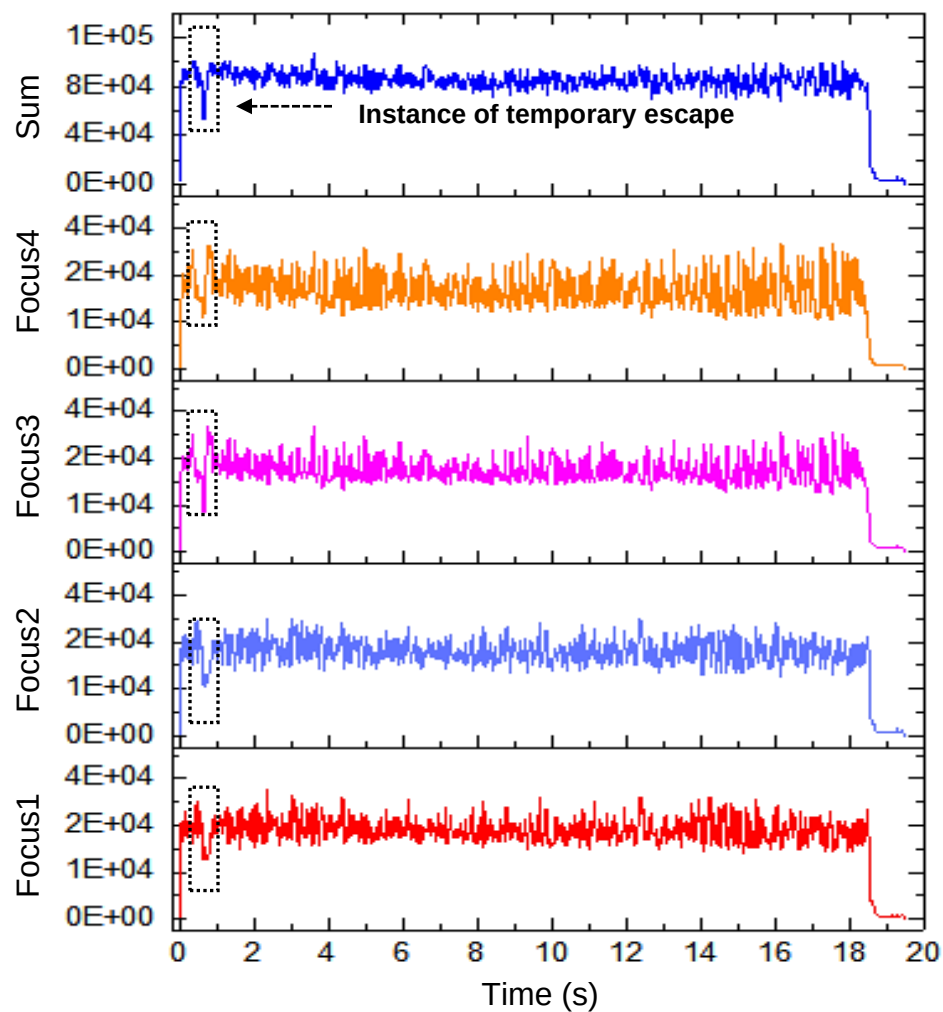


Figure 6.9. Trapping a 20 nm FluoSphere in distilled water (II).

Charts show photon counts (photons/10 ms) vs time for Focus1–Focus4 and the sum of counts from all four foci with 12 μ W power per beam. The dotted boxes highlight the time that the particle momentarily escapes from the trap, but the trap manages to recapture it.

An MLE of positions (see Figure 6.10) was performed for 20 nm FluoSphere trapping using some of the photon count data from Figure 6.9. The data for the 1–18 s range was chosen for this analysis (using the same averaging of photon counts per 1 ms, as was done for the 40 nm FluoSphere). Histograms of the particle positions are shown to the right and standard deviations are given beside their respective MLE position histograms, so it is again possible to estimate the effective spring constants k_x , k_y , and k_z for the 20 nm particle—about $8.0 \mu\text{N}\cdot\text{m}^{-1}$, $9.0 \mu\text{N}\cdot\text{m}^{-1}$, and $0.02 \mu\text{N}\cdot\text{m}^{-1}$, respectively. Although the histograms are fit to a Gaussian (in accord with modelling the trap by a 3D harmonic well effective trapping potential), the histogram for the z-positions of the particle actually has two peaks at about $z = \pm 0.4 \mu\text{m}$, which may indicate that the effective trapping potential is not harmonic in the Z-direction. (The z-histogram for trapping the 40 nm FluoSphere also had two peaks near $z = \pm 0.4 \mu\text{m}$, but the peaks were not clear and may have been statistical.)

Figure 6.11 shows the 3D scatterplot version of the data taken from Figure 6.10, which also indicates that the region in which the particle is trapped is elongated along the Z-direction.

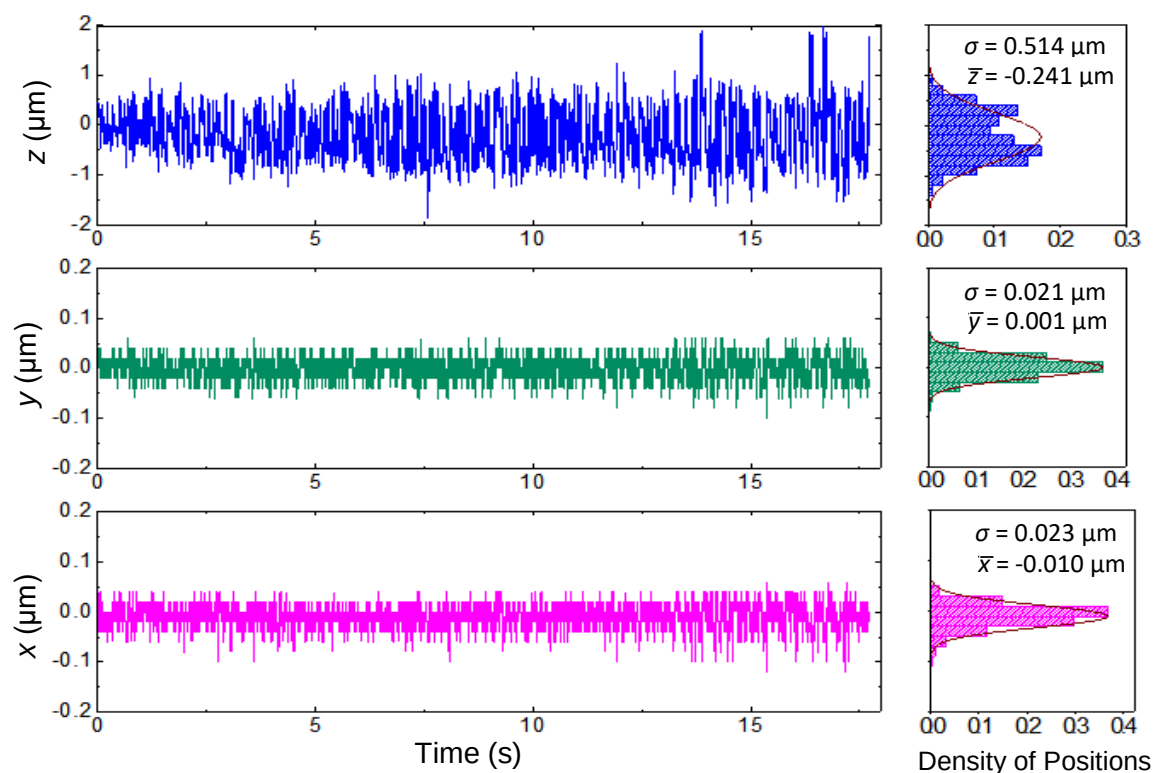


Figure 6.10. MLE of positions vs. Time for the data in Figure 6.9.

The position variations show only about 17 seconds (1–18 s) of position data for the trapped 20 nm FluoSphere shown in Figure 6.9. A jumping average of 1 ms photon count data was taken for one position determination. The histograms at right show the density of positions with respective standard deviations σ and mean values $\bar{x}$, $\bar{y}$, $\bar{z}$. Note that the bin size of the histograms is 0.02 μm for x - and y -positions and 0.2 μm for z .

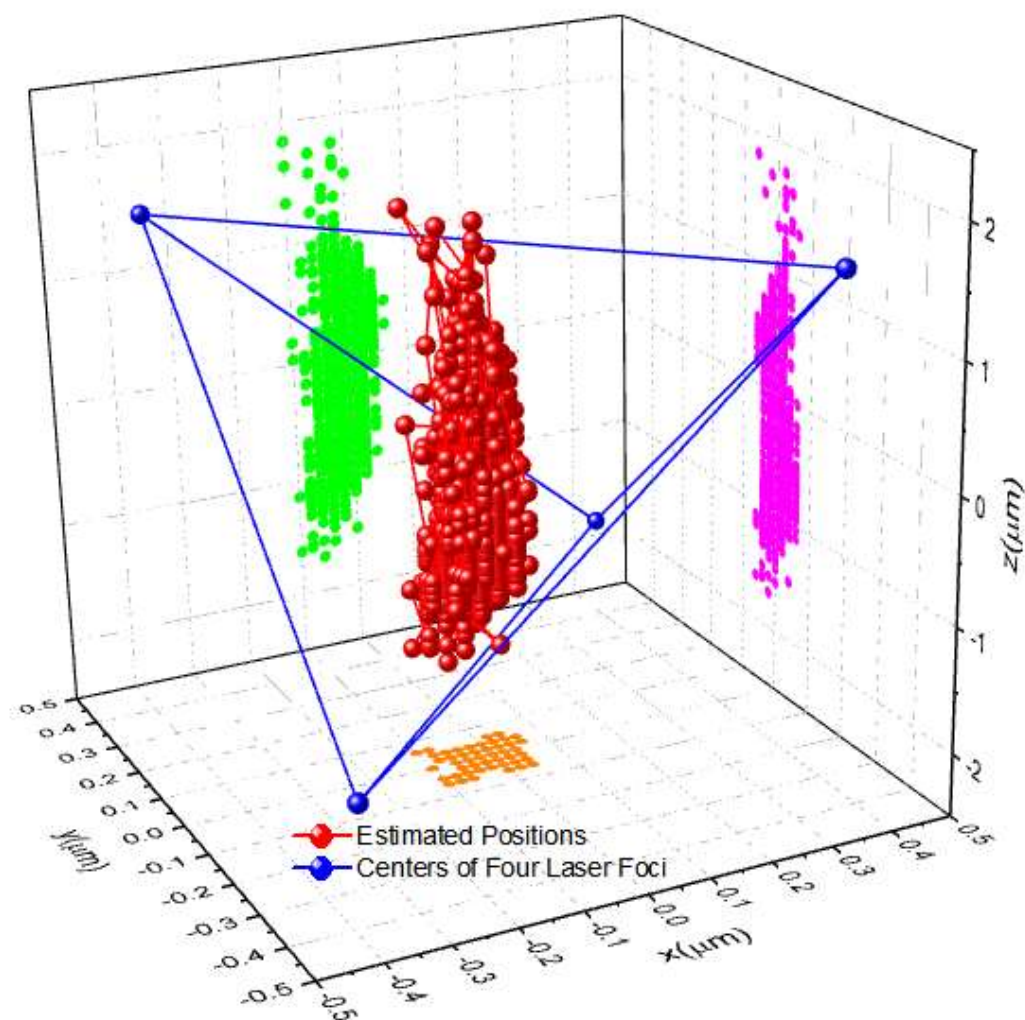


Figure 6.11. 3D plot of MLE positions of a trapped 20 nm FluoSphere.
(Data from Figure 6.10.)

Trapping Streptavidin-Alexa 647 Protein Molecules

Studies have also been conducted for Streptavidin-Alexa Fluor™ 647 molecules of ~5 nm in diameter [65]. Trapping times on the order of ~1 s have been observed. This much shorter time is to be expected compared to the larger FluoSpheres with lower diffusion coefficients and higher brightness. The estimated diffusion coefficient for the Streptavidin-Alexa conjugate is about $96.5 \mu\text{m}^2\cdot\text{s}^{-1}$ (at 293 K), which is factors of 4× and 8× faster compared to the 20 nm and 40 nm FluoSpheres, respectively. In addition, the approximated number of Alexa Fluor® 647 molecules per Streptavidin- Alexa Fluor® 647 conjugate molecule is about three [26] and this labelling is about two orders of magnitude lower than the fluorescent labelling of FluoSpheres compared with the brightness of a fluorescein molecule. (A 40 nm FluoSphere has a brightness equivalent to ~350 fluorescein molecules while a 20 nm FluoSphere has a brightness of ~180 fluorescein molecules [66].) Thus, the fluorescence response of the conjugate is much weaker than for the FluoSpheres, significantly reducing the SNR in the experiment. The laser power per beam therefore had to be increased up to ~30 μW (for 640 nm wavelength) to compensate for the low SNR and enable trapping.

Image Data of Trapping of ~1.5 Seconds

Trapping of a single-molecule for about 1.5 s was successfully captured as a video (see Video 3). Selected images taken from Video 3 are shown in

Figure 6.12 as time elapses. Note that the brightness of the trapped molecule is comparable to the background, so that the clarity of images is not as good as that of the FluoSphere images. In Video 3, several trapping events were observed, and the first trapping event is shown in Figure 6.12.

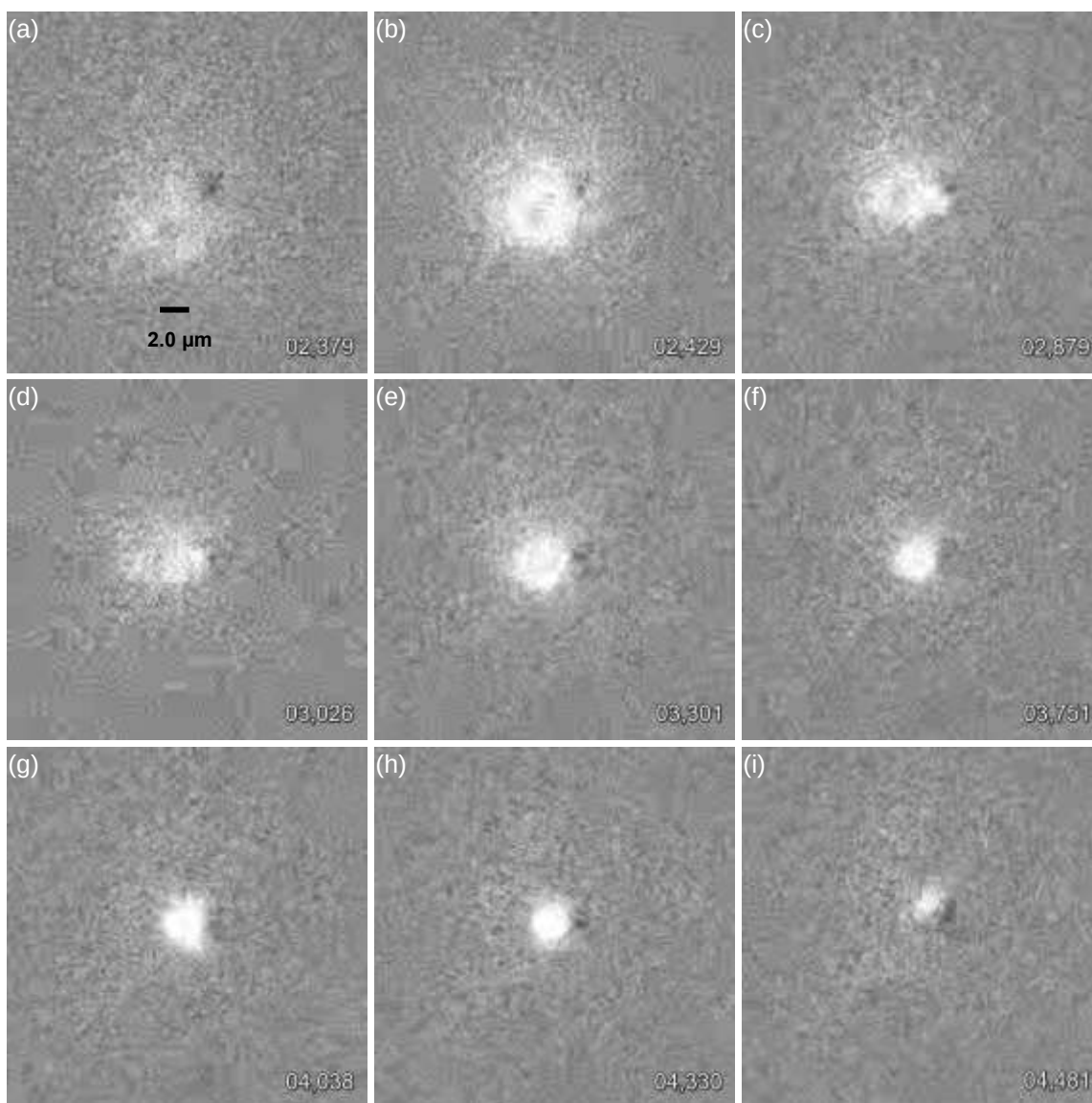


Figure 6.12. Image data for trapping a Streptavidin-Alexa 647 molecule in distilled water for ~1.5 seconds.

Images (each having 128×128 square pixels) were taken from Video 3 and show the time elapsed in the image at lower right (format: 34,235 means thirty-four seconds and two hundred thirty-five milliseconds). (a) A molecule enters the trapping region. (b) The molecule is trapped, but not in the center of the trap. (c)–(f) The molecule is being forced to move towards the center of the trap. (g), (h) The molecule is well-confined to the trap center. (i) The molecule escapes from the trap.

Photon Count Data of Trapping for over 0.5 Seconds

Figure 6.13 shows photon count data when trapping a single Streptavidin-Alexa molecule for about 0.5 s. During that time the trap has difficulty keeping the molecule in the ROI and the molecule eventually escapes. However, as shown in Figure 6.14, a trapping time of nearly 1.5 s may be achieved. In this case the trap was able to constrain the molecule more rigorously, and it can be seen that although the molecule almost diffuses away, it is returned to the center of the trap a second time (noted by the second peak in counts at ~1.3 s) before finally escaping. Even though the recorded trapping time is quite short compared to that for a 40 nm FluoSphere, it is a much longer amount of time than the mean diffusional residence time of about 1.3 ms for Streptavidin-Alexa molecules (according to Equation 14 with $\sigma_x = 0.5 \mu\text{m}$). Also, it should be noted that Alexa Fluor 647 dye has relatively high photobleaching resistance, so it should take more than 50 seconds to reduce its initial fluorescence intensity down to about 35% under an irradiance of $\sim 40 \mu\text{W}\cdot\mu\text{m}^{-2}$ [67]. Considering that the irradiance in the center of the trap is about $40 \mu\text{W}\cdot\mu\text{m}^{-2}$, it is more likely that the Streptavidin-Alexa molecule escapes from the trap not because of photobleaching but because of its high diffusivity.

MLE positions for the trapping of a Streptavidin molecules were not considered since the error in estimations would be large due to the low amount of fluorescence photons recorded per second compared to the cases of 40 nm or 20 nm FluoSpheres. Moreover, there is a significant increase in laser-light scatter

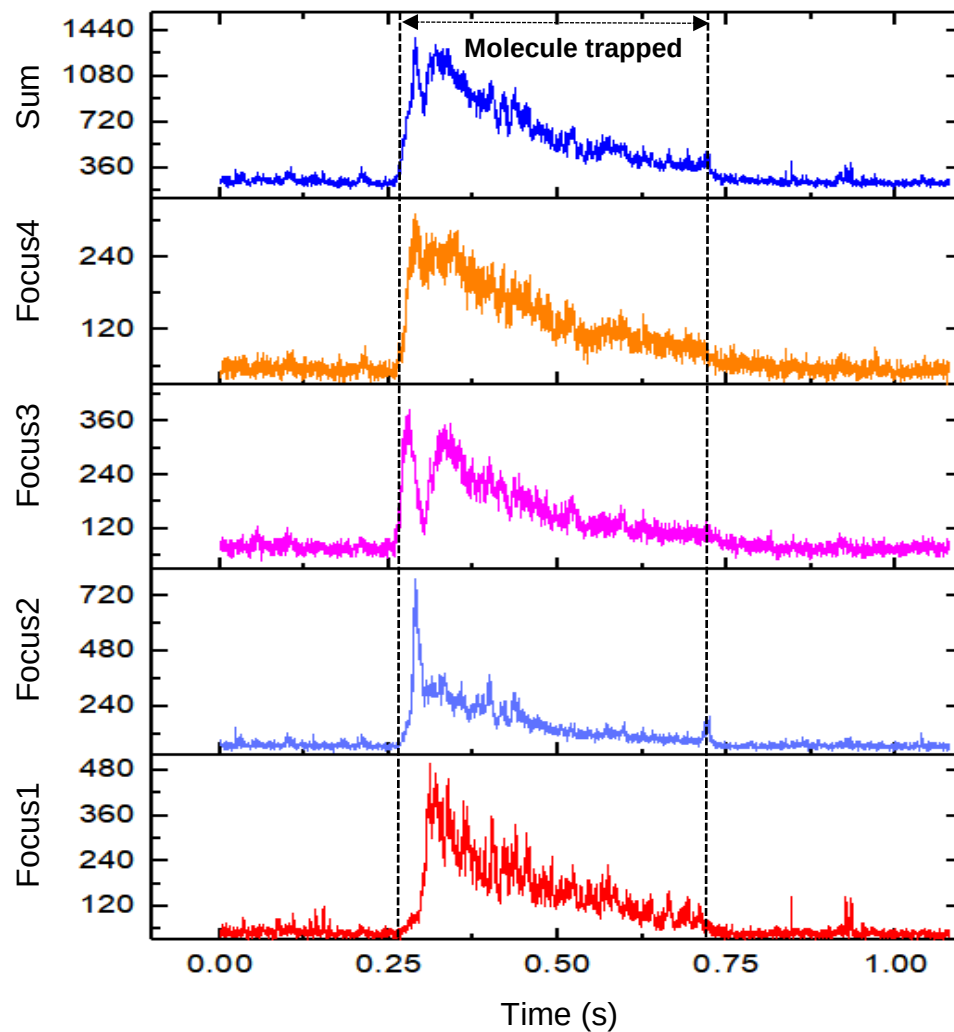


Figure 6.13. Trapping a single Streptavidin-Alexa 647 molecule in distilled water for ~0.5 seconds.

Charts show photon counts (photons/270 μ s) vs. time for Focus1–Focus4 and the sum of counts from all four foci with 30 μ W per beam.

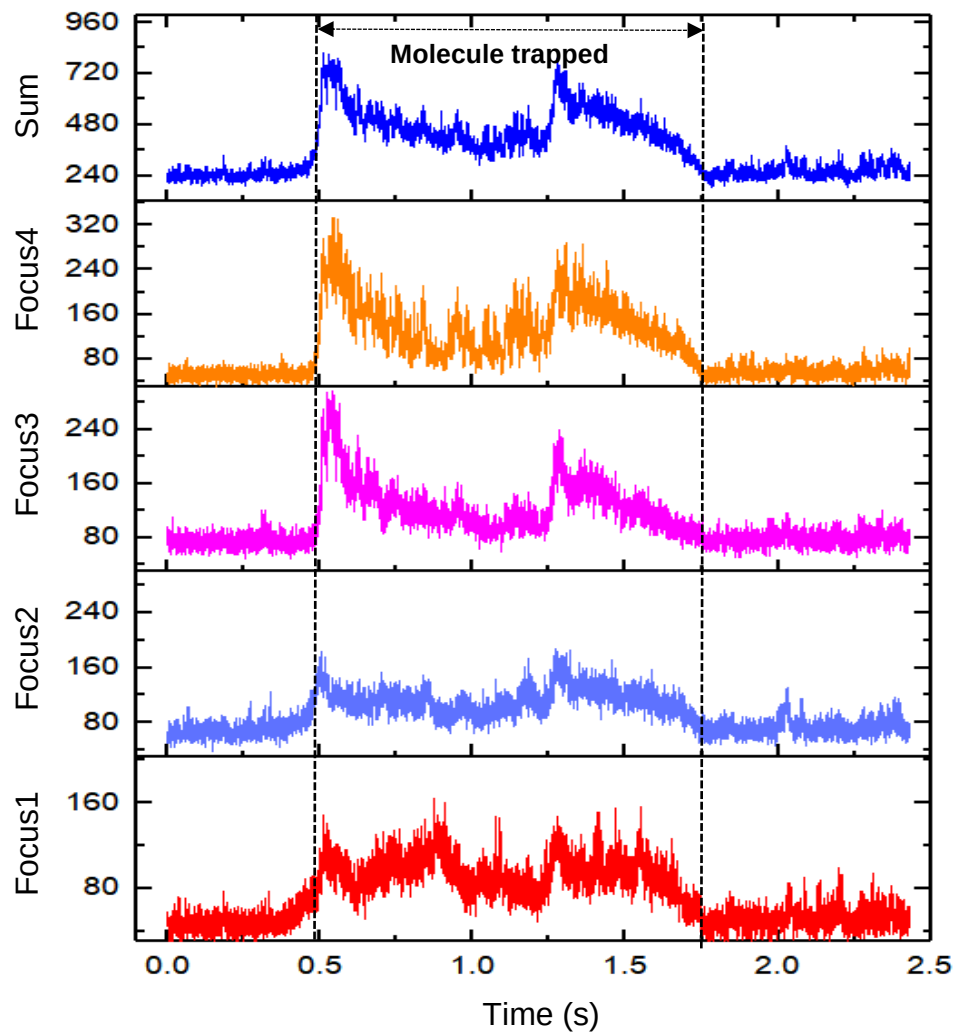


Figure 6.14. Trapping a single Streptavidin-Alexa 647 molecule in distilled water for ~1.5 seconds.

Charts show photon counts (photons per 270 μ s) vs. time for Focus1–Focus4 and the sum of counts from all four foci with 30 μ W per beam.

in the trapping region (due to higher powers of 30 μW per beam), and hence the SNR goes down to about 80:20, thereby lowering the trapping capability.

Analysis of Direction of Fluid Flow within the ROI

In this section, a theoretical estimation of the direction of the fluid flow within the tetrahedral laser excitation region is conducted to evaluate the trapping algorithm and the effective 3D trapping potential. For this estimation, it is sufficient to analyze the Gaussian-Lorentzian intensity profiles of the laser beams rather than using photon counts because the counts are directly proportional to the intensity profiles. For a 3D harmonic trapping potential, it is expected that the fluid flow always points towards the center of the trap, but the flow that results from the analysis of photon counts by the trapping algorithm is directed along 27 possible discrete directions, as were listed in Table 5.2.

The trapping algorithm was tested empirically through experiments and results show that it works (it achieves trapping), even though it is fairly simple. Applying the algorithm to photon counts expected from the intensity profiles of the four laser beams gives the expected fluid flow directions (if background and photon shot noise are negligible). An analysis determining the expected fluid flow throughout the ROI provides a guide to understanding how the trap works, as well as possible paths for future improvements. Therefore, intensity terms I_1 , I_2 , I_3 , and I_4 are defined for the four laser beam foci, Focus1, Focus2, Focus3, and Focus4, and analyzed as described in the following sections.

Intensity Variations along Cartesian Axes

The expected flow provided by the algorithm depends on comparing the summed intensity from various combinations of beams. In the experiments, the centers of the foci of the four laser beams are in a tetrahedral arrangement (see Figure 4.1) with 760 nm separation in X-Y plane ($\Delta x = \Delta y = 760$ nm) and with the two pairs of foci in two different planes in Z-axis separated by 3200 nm ($\Delta z = 3200$ nm). With these placements, the summed intensity from the various combinations of beams used by the algorithm for positions along the cartesian axes are shown in Figure 6.15. In the first column of the figure, at the top, I_1+I_2 is compared with I_3+I_4 for points along the Z-axis, as a similar comparison for photon counts is used by the algorithm in determining the flow along Z. In the middle, I_1+I_3 is compared with I_2+I_4 for points along the X-axis, as this comparison indicates the expected flow along X. At the bottom, I_2+I_3 is compared with I_1+I_4 for points along the Y-axis, as this indicates the expected flow along Y. In the second column of the figure, the differences are plotted: $((I_3+I_4) - (I_1+I_2))$ for points along the Z-axis; $((I_1+I_3) - (I_2+I_4))$ for points along the X-axis; and $((I_2+I_3) - (I_1+I_4))$ for points along the Y-axis.

The purpose of this analysis is to evaluate the way in which the trapping algorithm (shown in Table 5.1) behaves when a trapped particle is displaced from the origin. Assuming cross-talk effects are relatively small, the photon counts in each time gated channel, N_1, N_2, N_3 , and N_4 are approximately proportional to the intensities. For instance, if $((I_3+I_4) - (I_1+I_2)) > 0$, we expect $N_3+N_4 > N_1+N_2$, and

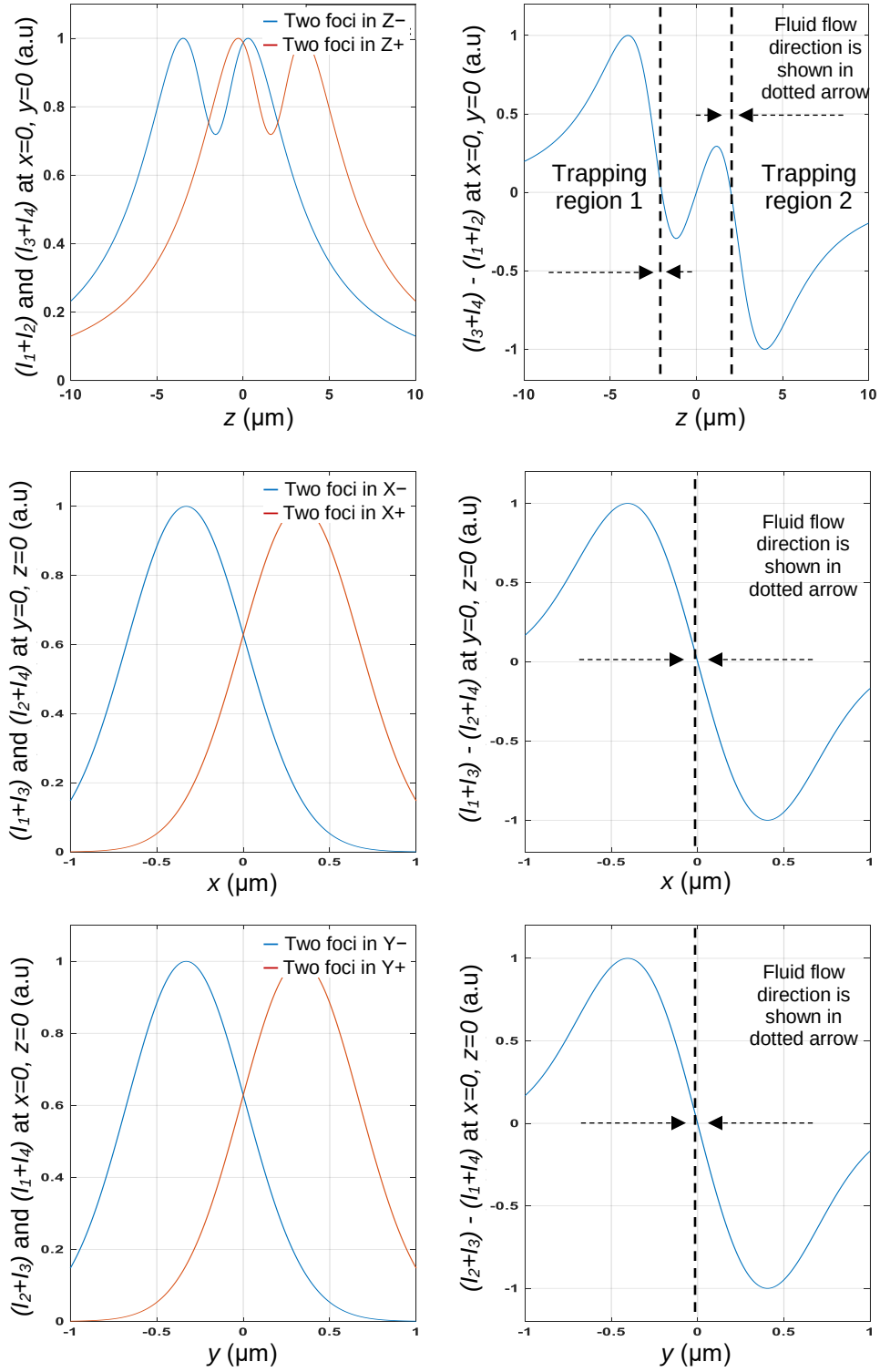


Figure 6.15. Intensity variations along the axes with $\Delta x = \Delta y = 760$ nm.

hence the algorithm would estimate that the trapped particle is in Z- region and it would set the fluid to flow along the Z+ direction. This makes sense, since Focus3 and Focus4 are in Z- region while Focus1 and Focus2 are in Z+ region. Similarly, if $((I_1+I_3) - (I_2+I_4)) > 0$, we expect the algorithm to set the fluid to flow along the X+ direction, since Focus1 and Focus3 are in X- region while Focus2 and Focus4 are in X+ region. Similar logic applies to $((I_2+I_3) - (I_1+I_4))$, with Focus2 and Focus3 are in Y- region while Focus1 and Focus4 are in Y+ region.

According to the lower two rows of Figure 6.15, the algorithm always moves the particle towards the center if it is displaced along the X- or Y-axis. However, from the top row of the figure, this is not necessarily the case for displacement along the Z-axis. If the z-displacement exceeds $\pm 2 \mu\text{m}$ (vertical dashed lines in Figure 6.15), the algorithm moves it towards the center, but if the z-displacement is less than $\pm 2 \mu\text{m}$, the algorithm moves it away from $z = 0$. This result gives a clue that the separation of the laser beam foci (i.e., Δx and Δy) can be set in such a way as to define different trapping modes: a one-potential-well trapping mode and a two-potential-well trapping mode.

In the one-potential-well mode, the well is centered at the origin (see Figure 6.16 or Figure 6.17) and the fluid flow is always towards the center. In the two-potential-well mode, the wells are offset along the Z-axis as shown in Figure 6.15. Figure 6.16, for which the beam separations are reduced to $\Delta x = \Delta y = 686 \text{ nm}$, shows the transition between the two modes and it is still a one-potential-well mode but with a broadened width. Figure 6.17, for which the beam

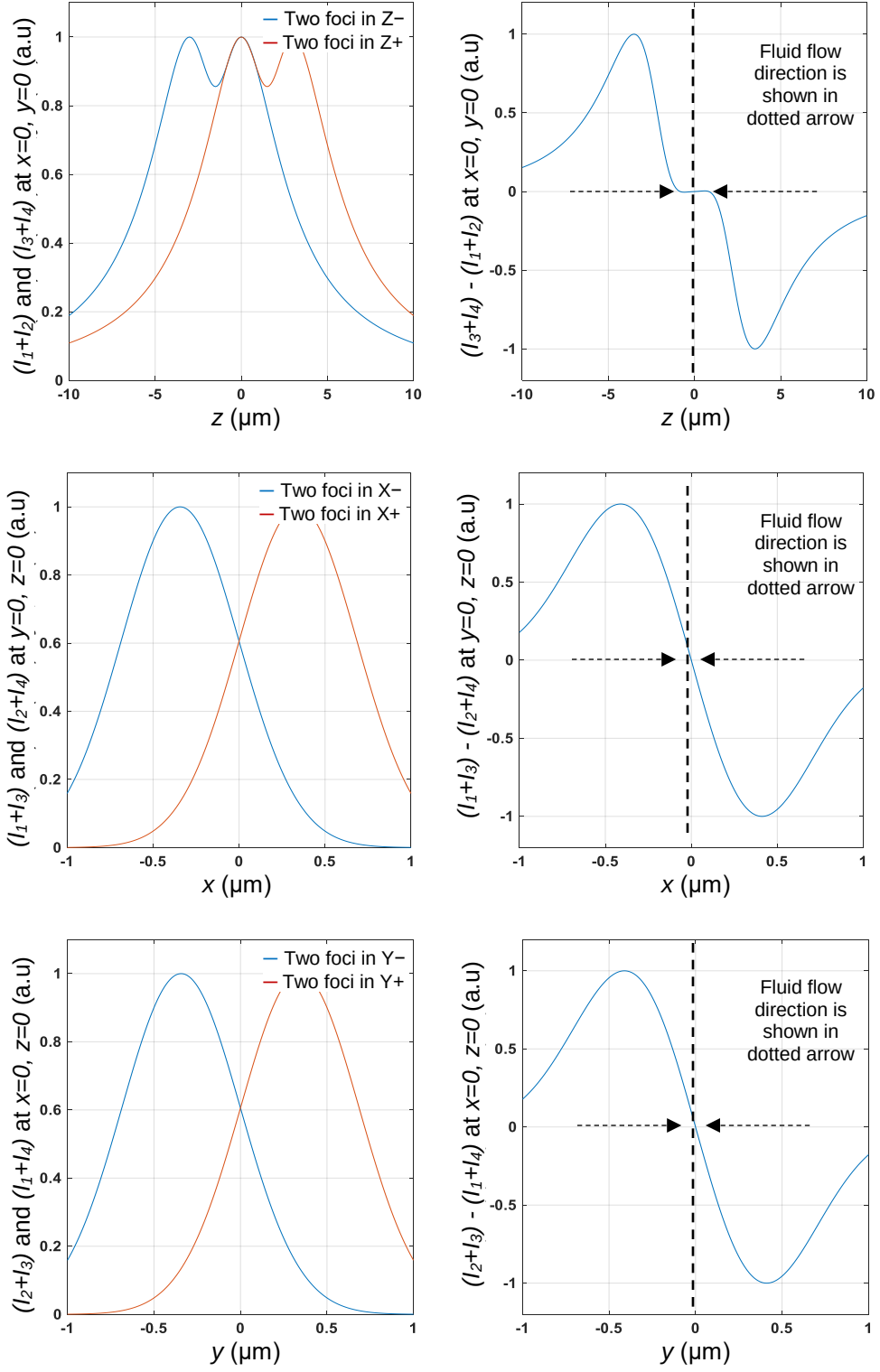


Figure 6.16. Intensity variations along the axes with $\Delta x = \Delta y = 686$ nm.

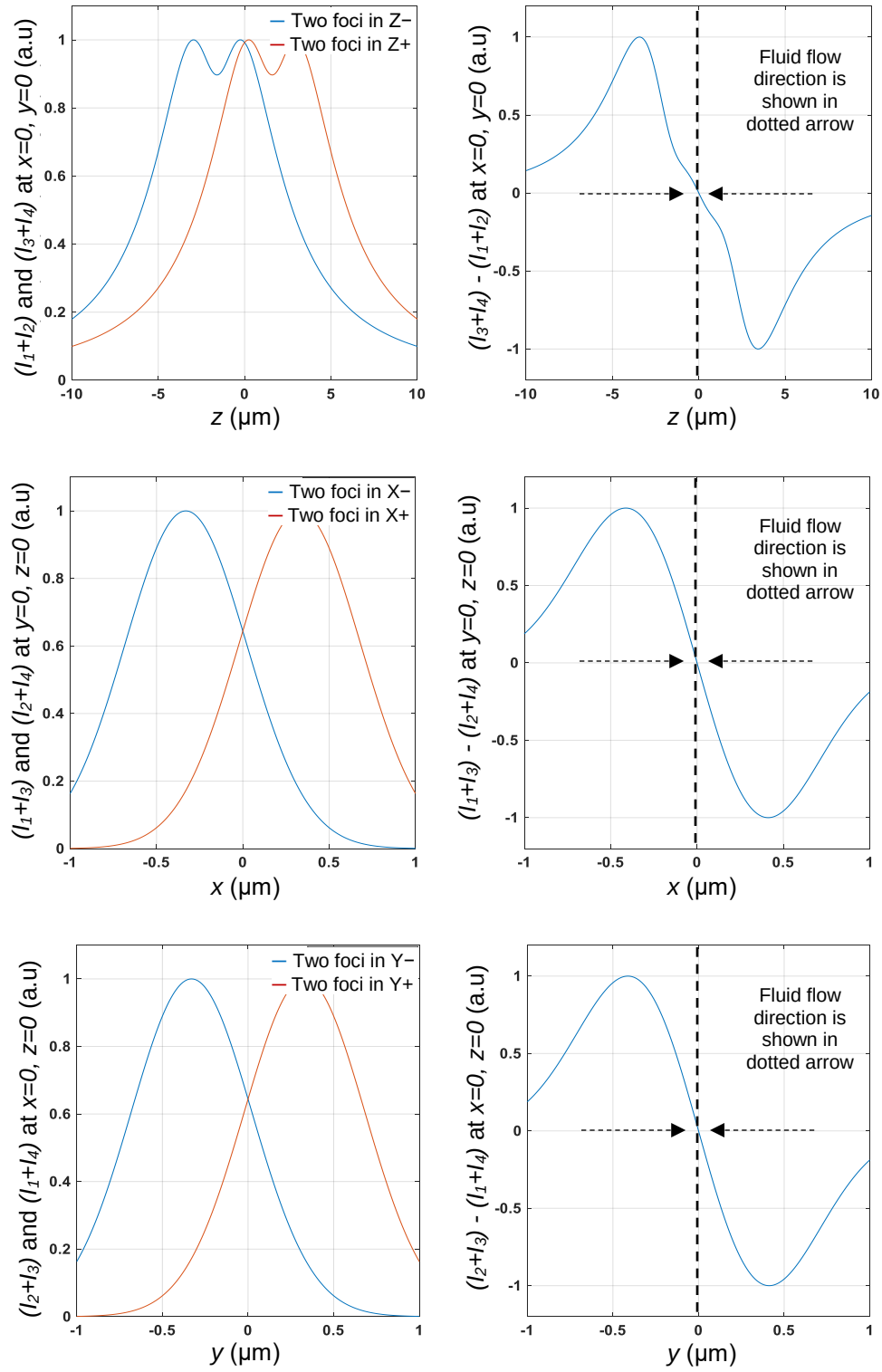


Figure 6.17 Intensity variations along the axes with $\Delta x = \Delta y = 660$ nm.

separations are further reduced to $\Delta x = \Delta y = 660$ nm, shows the case in which there is just a single potential well with a narrower width.

However, the above analysis only shows the case when the particle is displaced along the Cartesian axes. The potential well mode could differ when the particle is displaced from the center in three dimensions. To further investigate this possibility, the fluid flow direction analysis is performed for particle displacement in all three dimensions. This more in-depth analysis is reported below in the 3D Analysis of Flow Direction section. However, the findings do not alter and only reinforce what is expressed above, namely, if the separation of the beam foci is increased (to about $\Delta x = \Delta y = 760$ nm), the feedback trapping provided by the algorithm may result in a double-well effective potential along the Z-axis.

3D Analysis of Flow Directions

As pointed out in the previous section, in order to have an overall picture about the functionality of the trapping algorithm, an expected fluid flow direction analysis around the center of the coordinate system is conducted on a 3D grid of points (see Appendix E for the program written in MATLAB).

For the possible double-well case of $\Delta x = \Delta y = 760$ nm, Figure 6.18 shows the fluid directions in the X-Y plane for several selected Z-axis values (-1.8 μm to $+2.7$ μm). The red dots are points on the 3D grid and the blue arrows indicate the component in the plane for the direction of flow determined by applying the trapping algorithm to the combinations of the laser intensity fields, as discussed

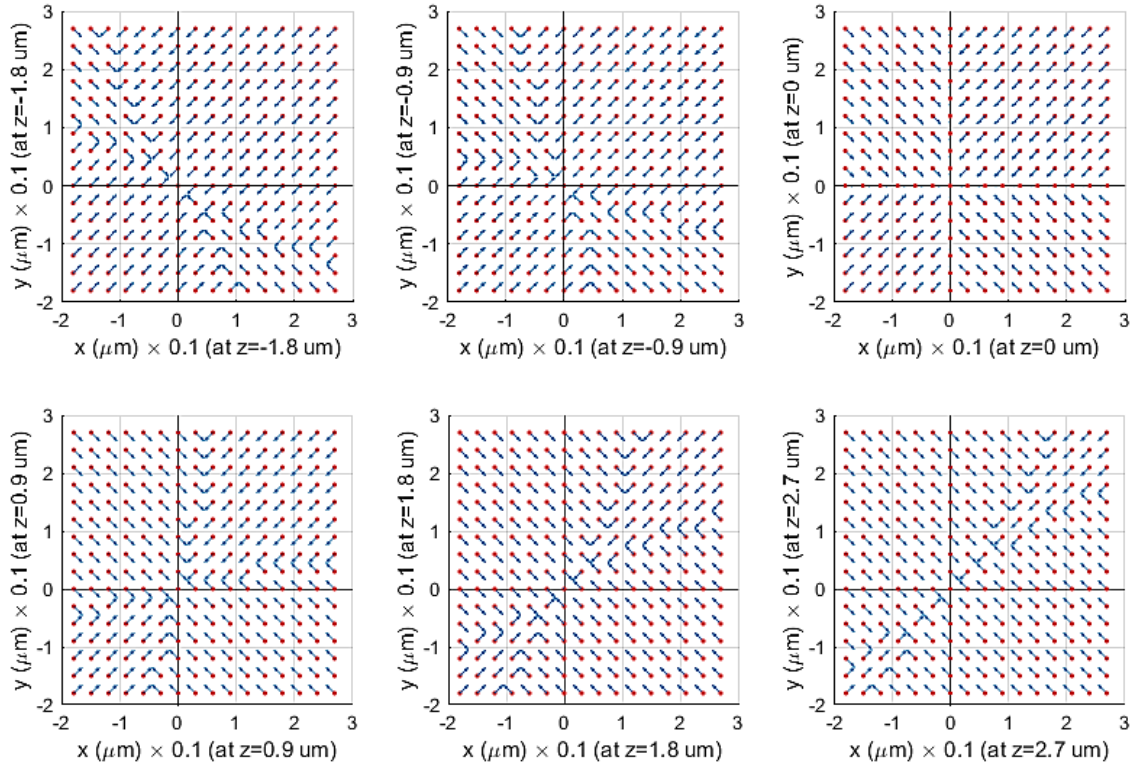


Figure 6.18. Flow directions in selected X-Y planes ($\Delta x = \Delta y = 760 \text{ nm}$).

in the previous section. The flow directions in each subdiagram are aligned towards the center of the X - Y plane, indicating that a particle should be moved towards the center of the X - Y plane.

For the same case of $\Delta x = \Delta y = 760$ nm, Figure 6.19 shows the fluid fields in the X - Z plane for selected Y -axis values (-0.18 μm to $+0.27$ μm). The fluid flow is not always pointing towards the center of the X - Z plane. In the X - Z plane for $y = 0$, we see the appearance of the two-potential well mode. The direction field is symmetric and also in agreement with the fluid direction that was illustrated in Figure 6.15. Notice that the symmetry of the direction field deteriorates when the planes move away from the origin (i.e., away from $y = 0$) when y reaches to ± 0.06 μm . Similar behaviour in the fluid flow directions is also seen for the direction fields in the Y - Z plane (Figure 6.20).

Even though the fluid direction field diagrams shown above indicate the existence of a two-potential-well mode, one may find it difficult to visualize and interpret the 3D fluid flow fields. In order to facilitate the interpretation, one can follow the motion of a hypothetical fluorescent particle with no diffusivity. The fluid flow moves the particle so that it eventually ends up at the center of the effective potential well, but if there is a double potential well, then different initial positions will result in two possible ending positions. Such a study was conducted for seven random initial positions for the particle using a MATLAB program (see Appendix F).

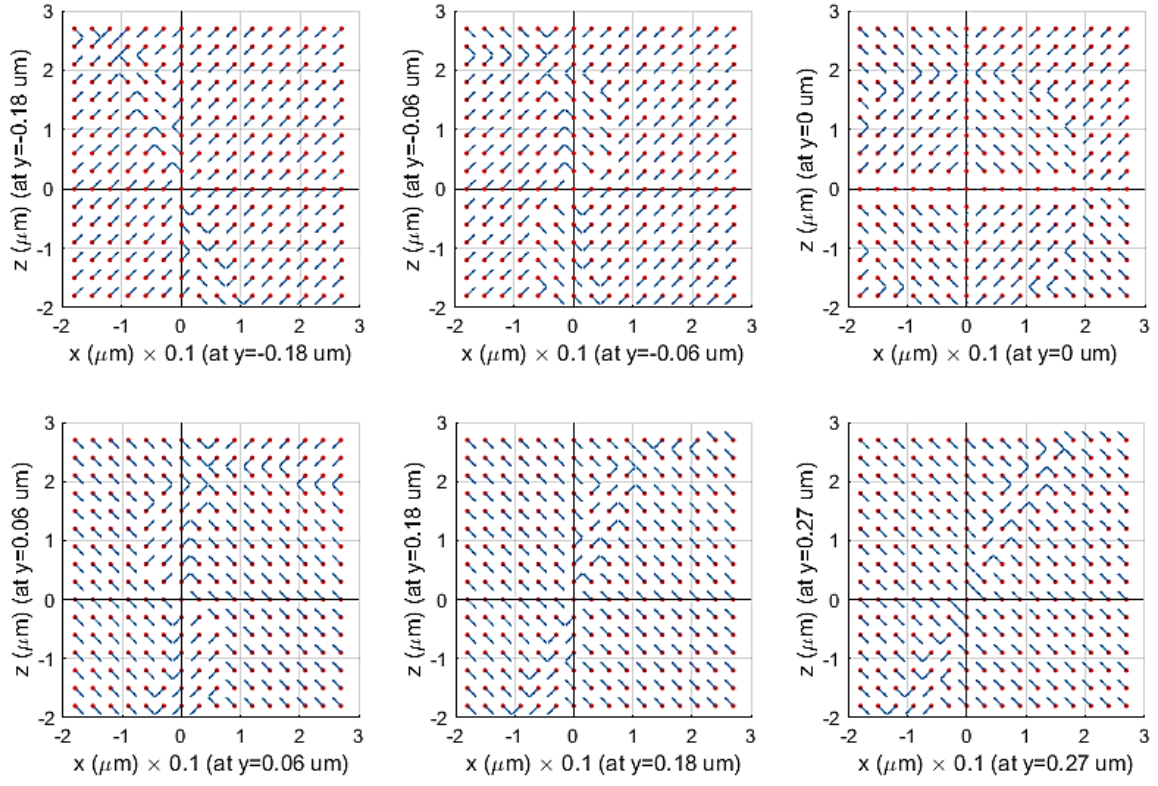


Figure 6.19. Flow directions in selected X-Z planes ($\Delta x = \Delta y = 760 \text{ nm}$).

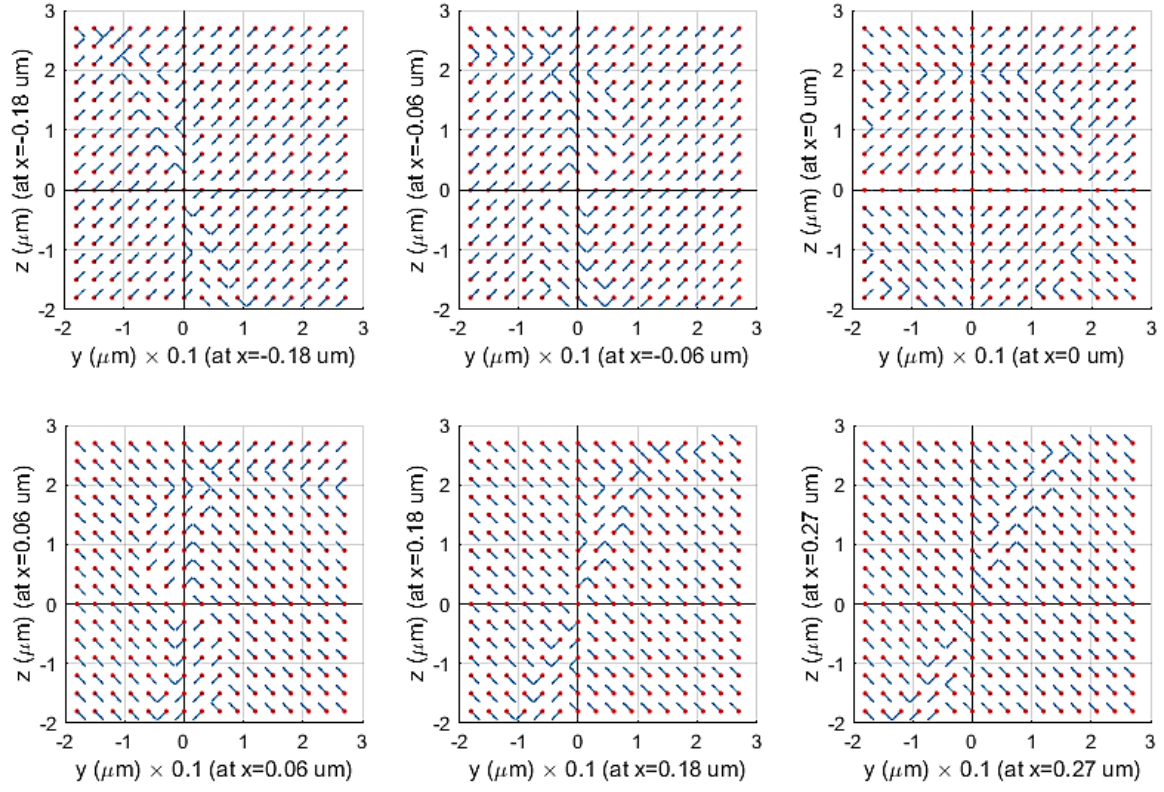


Figure 6.20. Flow directions in selected Y-Z planes ($\Delta x = \Delta y = 760 \text{ nm}$).

Figure 6.21 shows seven cases of Position vs. Iteration number starting with a random position on a 3D grid. For each position of the particle, the trapping algorithm is applied using the laser intensities at the particle position (as explained in the prior section of Intensity Field Variations Along Cartesian Axes) and the particle is moved to an adjacent grid point according to the direction of the flow. When the flow is zero, the particle has reached the center of the effective potential well. According to Figure 6.21, the particle always ends at $x = 0$ and $y = 0$ after a certain number of iterations. However, the positions along the Z-axis converge to $+2 \mu\text{m}$ or $-2 \mu\text{m}$. This confirms the interpretation of the simpler analysis of the previous section, that if the beam separation is $\Delta x = \Delta y = 760 \text{ nm}$, the trap may exhibit a double potential well mode along the Z-axis.

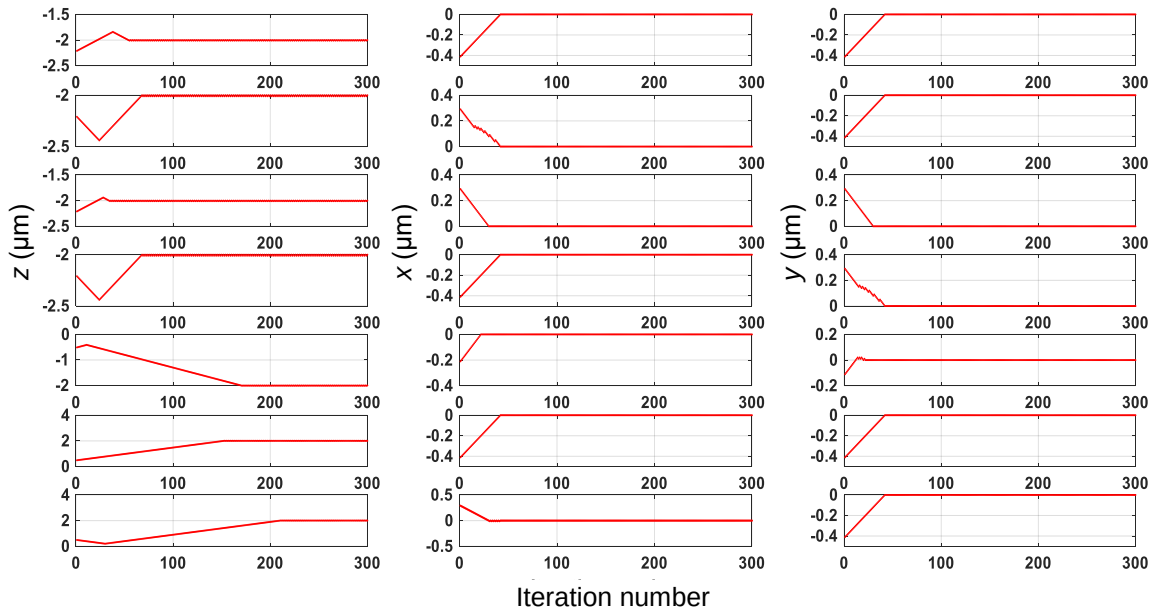


Figure 6.21. Testing for two-potential-well mode with zero diffusivity particle ($\Delta x = \Delta y = 760 \text{ nm}$).

The full 3D analysis was repeated for the other two beam separations considered in the simpler analysis of the section above. For brevity, the graphs are shown only the case of the closest beam separation, $\Delta x = \Delta y = 660$ nm. The fluid directions in the X-Y planes (see Figure 6.22) are quite similar to that for $\Delta x = \Delta y = 760$ nm. It again predicts an effective potential centered in the X-Y plane. However the fluid direction field in X-Z (Figure 6.23) and Y-Z (Figure 6.24) planes indicate that there is only a single potential well centered at $z = 0$.

To help interpret the above 3D flow directions, one can again follow the motion of a hypothetical particle with zero diffusivity. Results for seven random initial positions are shown in Figure 6.25 with each row represent the coordinates of the X- , Y-, and Z-axes. As expected, the figure shows that the positions in all the axes converge to zero confirming the simpler analysis that the effective potential has just a single well when the foci separations are $\Delta x = \Delta y = 660$ nm.

Overall, analysis of the 3D flow direction asserted by the trapping algorithm indicates that the trapping modes can be changed from one-potential-well mode to two-potential-well mode by increasing the separations of the foci, Δx and Δy .

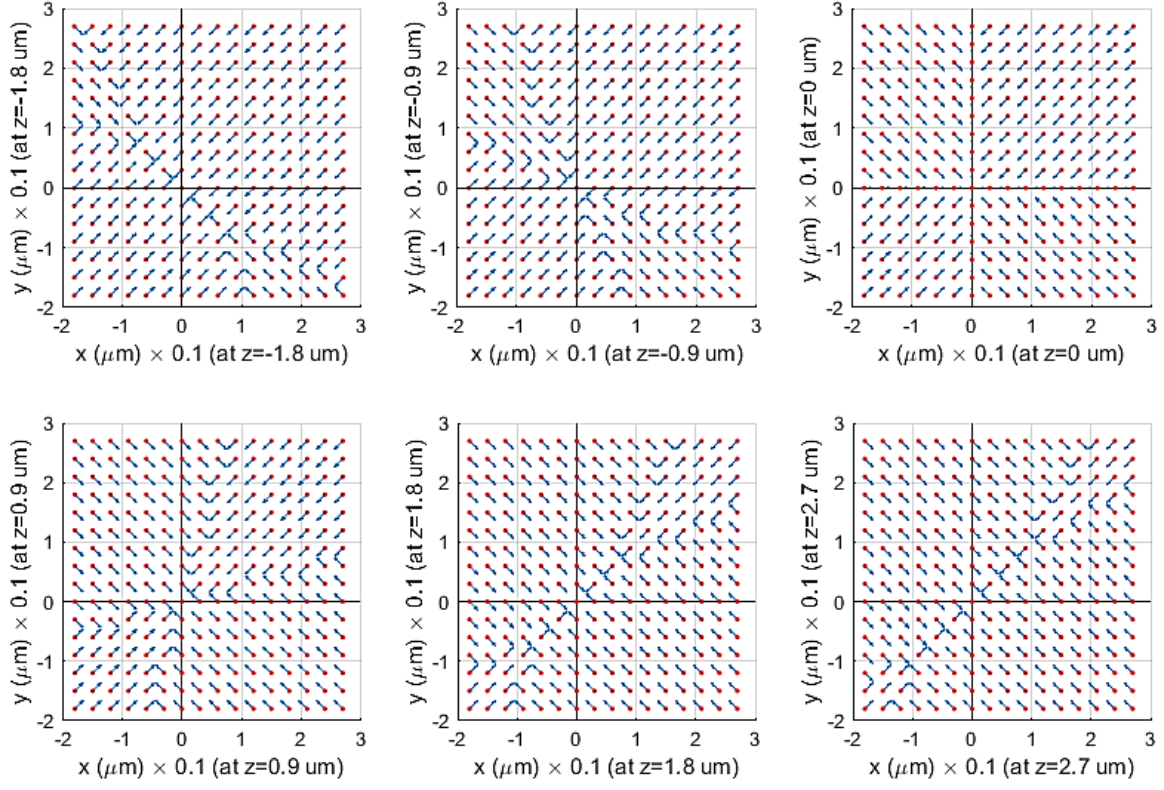


Figure 6.22. Flow directions in selected X-Y planes ($\Delta x = \Delta y = 660 \text{ nm}$).

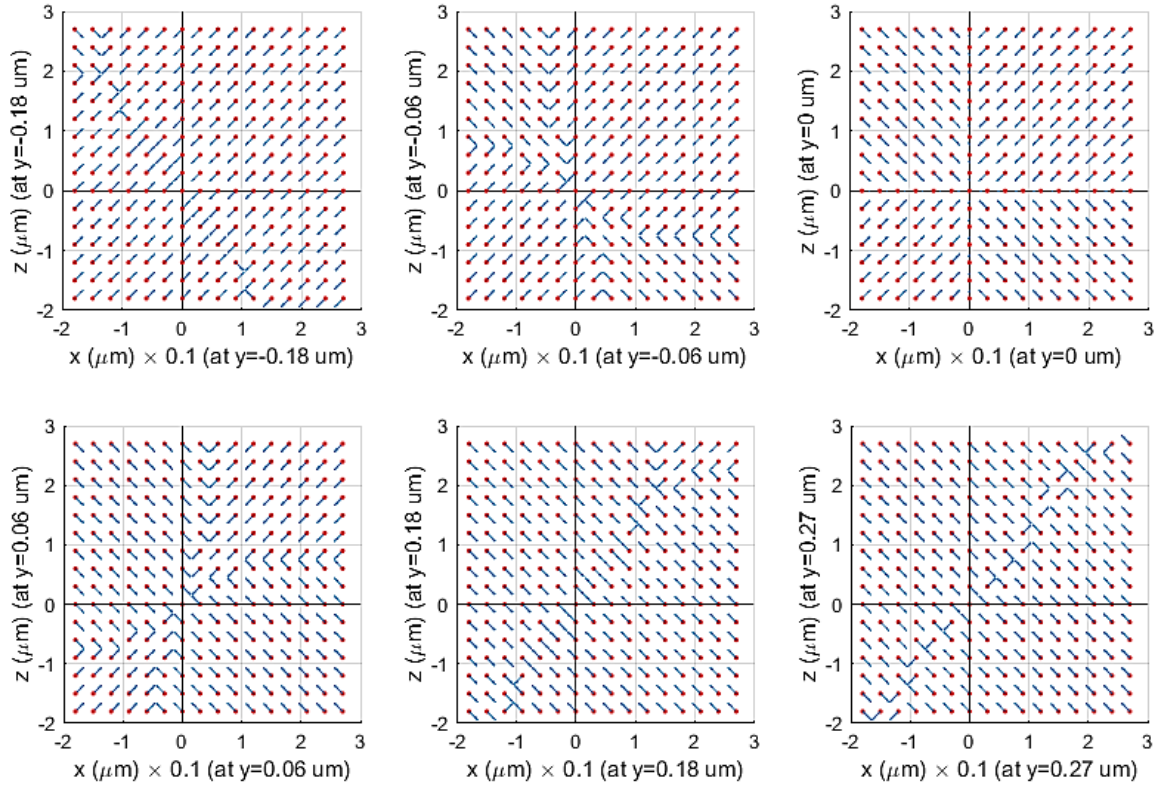


Figure 6.23. Flow directions in selected X-Z planes ($\Delta x = \Delta y = 660 \text{ nm}$).

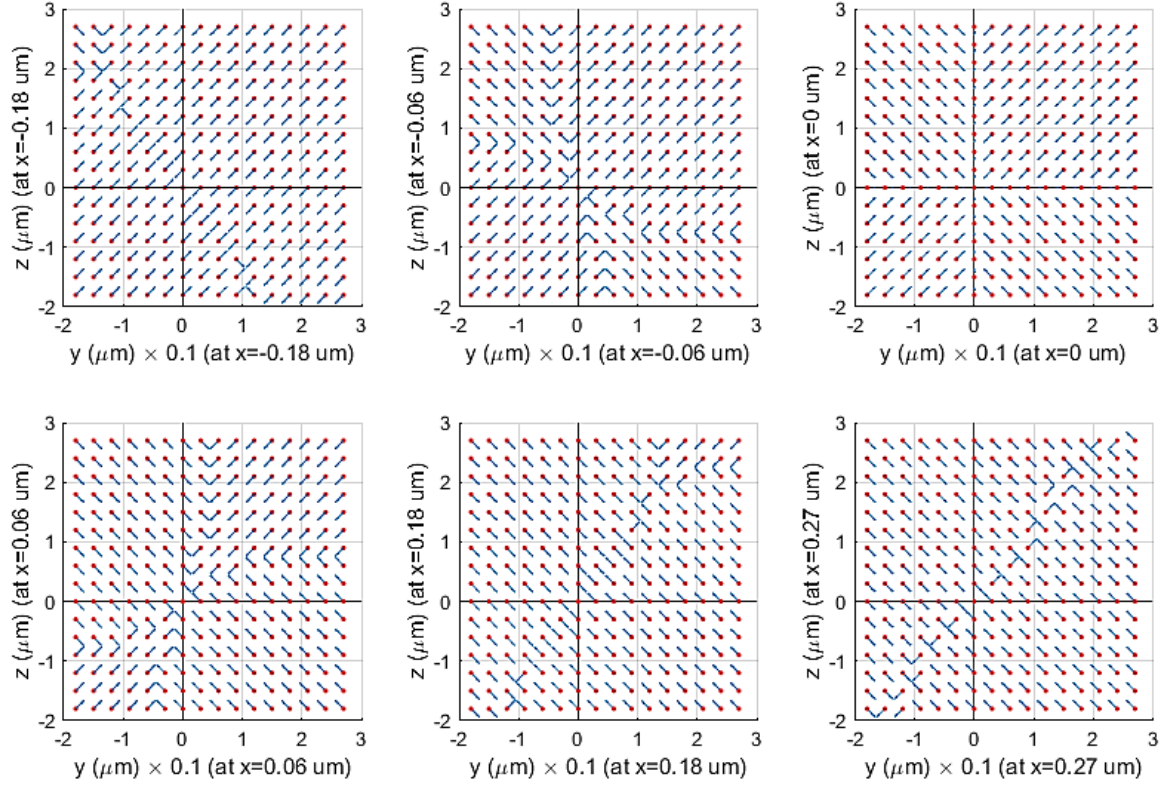


Figure 6.24. Flow directions in selected Y-Z planes ($\Delta x = \Delta y = 660$ nm).

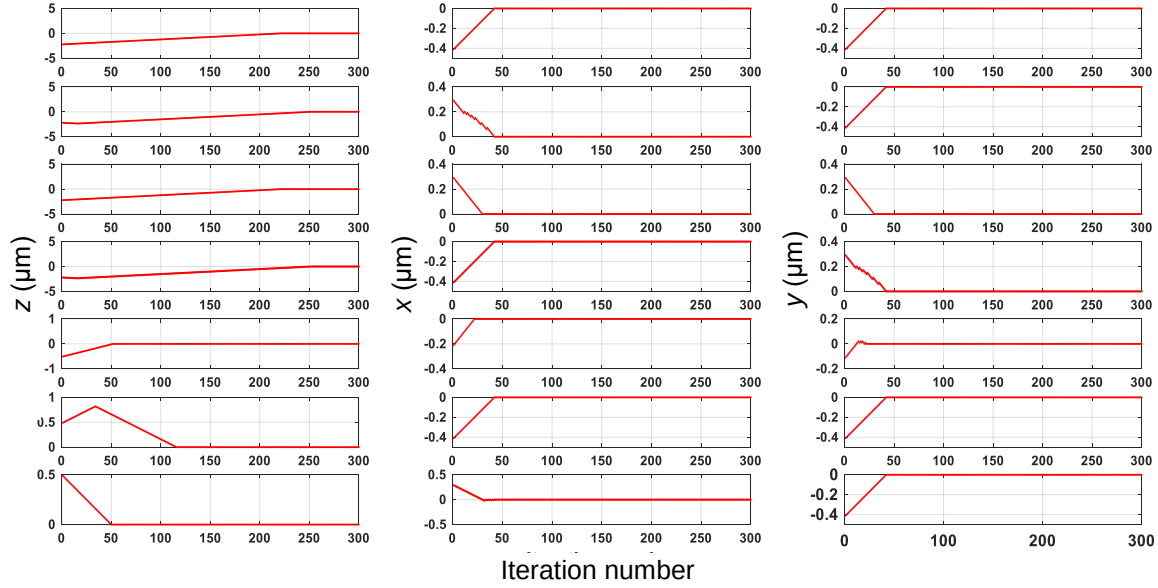


Figure 6.25. Testing for one-potential-well mode with zero diffusivity particle ($\Delta x = \Delta y = 660$ nm).

Discussion on Effective Potential of Trap

Although a two-potential well mode of trapping can exist for the case $\Delta x = \Delta y = 760$ nm, its effectiveness can be screened by the diffusion of a particle from one well to the other. This is possible because of the shallowness and narrow widths (within ± 0.06 μm in the X - Y plane) of the wells. The net motion of the particle, considering both the diffusion and the restoring electroosmotic motion, could be greater than the separation of the wells over the course of time. In other words, a trapped particle may move through the center of the ROI rather than being confined to one well or the other. To a first approximation, the effective trapping potential is then still harmonic near the center of the ROI but with a low effective spring constant.

Even if the energy of the particle is such that it remains for long periods within one well or the other, it may tunnel between the wells. A similar situation has been observed in a 2D ABEL trap, where a particular feedback algorithm was applied to deliberately cause a double-well effective potential to study such tunneling [21].

In general, the effectiveness of a feedback-driven trap is limited by the feedback latency and the restoring motion. The effective trapping potential has a finite depth and can be harmonic only near the center of the trap. Due to the finite depth of the effective potential, eventually the trapped particle will escape. (The particle can never escape from an infinite harmonic well potential.) For the cross-channel device, the electroosmotic restoring speed is limited, particularly in the Z-direction, and if the dimensions of the trapping ROI are decreased, the time before escape will also decrease. Although more tightly confined motion of a trapped particle could possibly be obtained by decreasing the separation of the foci to avoid the double-well mode, for a fixed electroosmotic speed, the time before escape will also decrease. These features were experimentally observed. Trapping experiments were attempted with decreased separation between the laser foci (about $\Delta x = \Delta y = 600$ nm), but the effectiveness of the trap was considerably decreased in that 40 nm Fluospheres could only be trapped for about 5 seconds.

CHAPTER SEVEN:

CONCLUSIONS AND FUTURE IMPROVEMENTS

The trapping system presented in this dissertation (four-focus confocal fluorescence microscopy, electroosmotic cross-channel device, and simple trapping algorithm having a fast feedback cycle of $\sim 13.5 \mu\text{s}$) enabled clear, empirical demonstration that prolonged observations of a single nanoparticle freely diffusing in aqueous solution is possible. The results demonstrate that trapping is possible for 40 nm diameter FluoSpheres, 20 nm FluoSpheres, and 5 nm Streptavidin-Alexa 647 single molecules in distilled water. Moreover, with the four laser foci offset by $\Delta x = \Delta y = 760 \text{ nm}$ in X - Y plane and $\Delta z = 3200 \text{ nm}$ in Z -axis, the trap could trap a 40 nm diameter FluoSphere for an unprecedented time exceeding 4 minutes. Since the results were obtained by using just distilled water (without slowing the rate of diffusion by adding glycerol to the solution), the system is quite promising, especially for biomolecular studies where aqueous solutions are highly preferred.

One key aspect of the trap is its use of a fast feedback cycle ($13.5 \mu\text{s}$) for electroosmotic control of fluid flow, with the same FPGA program used for trapping experiments on different particle sizes (i.e., 40 nm, 20 nm and 5 nm). It is possible to modify the programs and recompile the FPGA to adjust the duration of the feedback cycle, but the process requires a series of tests to make sure that other functionalities (i.e., TGPC and voltage control) still work properly. This testing is time-consuming and experience has shown that it is prone to

problems. Also, the FPGA logic is user-friendly only in regards to integers. Implementation of more complex algorithms that could increase trap effectiveness by incorporating fractional numbers is not easily programmed into the FPGA. Also, parameters used by the trapping algorithm cannot be dynamically changed during an experiment.

The main reason that the Altera Cyclone II FPGA was used is because it was the only device available that could be used to create four 3.3 ns-width time gates that are synchronous with the four laser beams used in the experiment. The FPGA was synchronized with the main laser beam that runs at 76 MHz to derive four laser beams at 304 MHz. Once this synchronization is properly established, it becomes natural to program the FPGA board to handle the rest of the control algorithm required for trapping as well. In the current situation the FPGA board controls the whole data collection and feedback cycle, but this setup offers limited flexibility in program modification.

As alternative to placing all the functionality on the Altera Cyclone II FPGA board, it is possible to use the Cyclone II FPGA board to obtain the digital signals of time-gated photon detection and interface these signals with a separate LabVIEW FPGA module (NI PCI-7833R) for the feedback algorithm and controlling the trapping voltages. Since the rate of time gates is 304 MHz but the 7833R can only run at 100 MHz maximum (and more importantly, cannot be synchronized to the 76 MHz laser pulse repetition rate), it would be required to program the Cyclone II FPGA board to output elongated digital signals for use by

the 7833R FPGA. This should be possible, considering that the output pulses of the FPGA board and input of the LabVIEW FPGA module are compatible (both use 3.3 V LVTTTL pulses).

The photon count data received by the 7833R could then be processed in a custom LabVIEW program operating in the LabVIEW real-time environment. The program could handle a trapping algorithm with greater complexity and more flexibility, since parameter control and tuning could be done in real time. Averaging photon counts, setting thresholds that activate the trapping algorithm, and real-time position estimation and feedback control with the use of digital filters such as a Kalman filter [20] (in contrast to the feedback control described in this dissertation) could be performed with relative ease. It is anticipated that such modifications and improvements could lead to enhanced trapping of single molecules of the order of 5 nm for much longer observation times than observed with the current 3D ABEL trap.

LIST OF REFERENCES

1. Moerner, W.E., *Nobel Lecture: Single-molecule spectroscopy, imaging, and photocontrol: Foundations for super-resolution microscopy*. Reviews of Modern Physics, 2015. **87**(4).
2. Moerner, W.E., *Single-molecule chemistry and biology special feature: New directions in single-molecule imaging and analysis (vol 104, pg 12596, 2007)*. Proceedings of the National Academy of Sciences of the United States of America, 2007. **104**(39): p. 15584-15584.
3. Hachohen, N., Ip, C.J.X., and Gordon, R., *Analysis of Egg White Protein Composition with Double Nanohole Optical Tweezers*. Acs Omega, 2018. **3**(5): p. 5266-5272.
4. Deniz, A.A., Mukhopadhyay, S., and Lemke, E.A., *Single-molecule biophysics: at the interface of biology, physics and chemistry*. Journal of the Royal Society Interface, 2008. **5**(18): p. 15-45.
5. Hoffmann, A., Nettels, D., Clark, J., Borgia, A., Radford, S.E., Clarke, J., and Schuler, B., *Quantifying heterogeneity and conformational dynamics from single molecule FRET of diffusing molecules: recurrence analysis of single particles (RASP)*. Physical Chemistry Chemical Physics, 2011. **13**(5): p. 1857-1871.
6. Cohen, A.E. and Fields, A.P., *The Cat That Caught the Canary: What To Do with Single-Molecule Trapping*. Acs Nano, 2011. **5**(7): p. 5296-5299.
7. Ashkin, A., Dziedzic, J.M., Bjorkholm, J.E., and Chu, S., *Observation of a single-beam gradient force optical trap for dielectric particles*. Optics Letters, 1986. **11**(5): p. 288-290.
8. Shi, J.J., Ahmed, D., Mao, X., Lin, S.C.S., Lawit, A., and Huang, T.J., *Acoustic tweezers: patterning cells and microparticles using standing surface acoustic waves (SSAW)*. Lab on a Chip, 2009. **9**(20): p. 2890-2895.
9. Hertz, H.M., *Standing-wave acoustic trap for nonintrusive positioning of microparticles*. Journal of Applied Physics, 1995. **78**(8): p. 4845-4849.
10. Huang, T.J. and Ieee, *Acoustic Tweezers: Manipulating Particles, Cells, and Organisms Using Standing Surface Acoustic Waves (SSAW)*, in 2013 Ieee Sensors. 2013. p. 1403-1403.
11. Ng, J.W., Devendran, C., and Neild, A., *Acoustic tweezing of particles using decaying opposing travelling surface acoustic waves (DOTSAW)*. Lab on a Chip, 2017. **17**(20): p. 3489-3497.
12. Davis, L.M., Canfield, B.K., Li, X., Hofmeister, W.H., Shen, G., Lescano-Mendoza, I.P., Bomar, B.W., Wikswo, J.P., Markov, D.A., Samson, P.C., Daniel, C., Sikorski, Z., and Robinson, W.N., *Electrokinetic Delivery of Single Fluorescent Biomolecules in Fluidic Nanochannels*, in Biosensing, M. Razeghi and H. Mohseni, Editors. 2008.
13. Lesoine, J.F., Venkataraman, P.A., Maloney, P.C., Dumont, M.E., and Novotny, L., *Nanochannel-Based Single Molecule Recycling*. Nano Letters, 2012. **12**(6): p. 3273-3278.

14. Canfield, B.K., King, J.K., Robinson, W.N., Hofmeister, W.H., and Davis, L.M., *Rapid, Single-Molecule Assays in Nano/Micro-Fluidic Chips with Arrays of Closely Spaced Parallel Channels Fabricated by Femtosecond Laser Machining*. Sensors, 2014. **14**(8): p. 15400-15414.
15. Hoang, H.T., Segers-Nolten, I.M., Berenschot, J.W., de Boer, M.J., Tas, N.R., Haneveld, J., and Elwenspoek, M.C., *Fabrication and interfacing of nanochannel devices for single-molecule studies*. Journal of Micromechanics and Microengineering, 2009. **19**(6).
16. Huh, D., Mills, K.L., Zhu, X.Y., Burns, M.A., Thouless, M.D., and Takayama, S., *Tuneable elastomeric nanochannels for nanofluidic manipulation*. Nature Materials, 2007. **6**(6): p. 424-428.
17. Cohen, A.E. and Moerner, W.E., *Method for trapping and manipulating nanoscale objects in solution*. Applied Physics Letters, 2005. **86**(9).
18. Cohen, A.E., *Control of nanoparticles with arbitrary two-dimensional force fields*. Physical Review Letters, 2005. **94**(11).
19. Cohen, A.E. and Moerner, W.E., *Controlling Brownian motion of single protein molecules and single fluorophores in aqueous buffer*. Optics Express, 2008. **16**(10): p. 6941-6956.
20. Kayci, M., Chang, H.C., and Radenovic, A., *Electron Spin Resonance of Nitrogen-Vacancy Defects Embedded in Single Nanodiamonds in an ABEL Trap (vol 14, pg 5335, 2014)*. Nano Letters, 2014. **14**(10): p. 6036-6036.
21. Jun, Y., Gavrilov, M., and Bechhoefer, J., *High-Precision Test of Landauer's Principle in a Feedback Trap*. Physical Review Letters, 2014. **113**(19).
22. Fields, A.P. and Cohen, A.E., *Electrokinetic trapping at the one nanometer limit*. Proceedings of the National Academy of Sciences of the United States of America, 2011. **108**(22): p. 8937-8942.
23. King, J.K., *Three-Dimensional Electrokinetic Trapping of a Single Fluorescent Nanoparticle in Solution*. 2013, University of Tennessee, Knoxville.
24. King, J.K., Canfield, B.K., and Davis, L.M., *Three-dimensional anti-Brownian electrokinetic trapping of a single nanoparticle in solution*. Applied Physics Letters, 2013. **103**(4).
25. Germann, J.A., *Tracking Real-Time Nanoparticle Positions and Measuring Three-Dimensional Solution Flow with a Four-Focus Confocal Microscope*. 2013, University of Tennessee, Knoxville.
26. Germann, J.A. and Davis, L.M., *Three-dimensional tracking of a single fluorescent nanoparticle using four-focus excitation in a confocal microscope*. Optics Express, 2014. **22**(5): p. 5641-5650.
27. Wells, N.P., Lessard, G.A., Goodwin, P.M., Phipps, M.E., Cutler, P.J., Lidke, D.S., Wilson, B.S., and Werner, J.H., *Time-Resolved Three-Dimensional Molecular Tracking in Live Cells*. Nano Letters, 2010. **10**(11): p. 4732-4737.

28. Perillo, E.P., Liu, Y.-L., Khang, H., Liu, C., Chou, C.-K., Hung, M.-C., Yeh, H.-C., and Dunn, A.K., *Deep and high-resolution three-dimensional tracking of single particles using nonlinear and multiplexed illumination*. Nature Communications, 2015. **6**.
29. Kayci, M. and Radenovic, A., *Single florescent nanodiamond in a three dimensional ABEL trap*. Scientific Reports, 2015. **5**.
30. Cohen, A.E., *Trapping and Manipulating Single Molecules in Solution*. 2006, Stanford University.
31. Wong, P.K., Wang, T.H., Deval, J.H., and Ho, C.M., *Electrokinetics in micro devices for biotechnology applications*. Ieee-Asme Transactions on Mechatronics, 2004. **9**(2): p. 366-376.
32. Precissi, J. *Capillary Electrophoresis*. [Web Page] 2017 [cited 2018 April 02]; Available from: http://chemwiki.ucdavis.edu/Core/Analytical_Chemistry/Instrumental_Analysis/Capillary_Electrophoresis.
33. CISM, *Electrokinetics and Electrohydrodynamics in Microsystems*. CISM Courses and Lectures, ed. A. Ramos. Vol. 530. 2011, New York: Springer Wien. 300.
34. Fairhurst, D. *An Overview of the Zeta Potential - Part 1: The Concept*. 2013 [cited 2019 January 20]; Available from: <https://www.americanpharmaceuticalreview.com/Featured-Articles/133232-An-Overview-of-the-Zeta-Potential-Part-1-The-Concept/>.
35. Tandon, V., Bhagavatula, S.K., Nelson, W.C., and Kirby, B.J., *Zeta potential and electroosmotic mobility in microfluidic devices fabricated from hydrophobic polymers: 1. The origins of charge*. Electrophoresis, 2008. **29**(5): p. 1092-1101.
36. Davis, L., Sikorski, Z., Robinson, W., Shen, G.Q., Li, X.X., Canfield, B., Lescano, I., Bomar, B., Hofmeister, W., Germann, J., King, J., White, Y., and Terekhov, A., *Maximum-likelihood position sensing and actively controlled electrokinetic transport for single-molecule trapping - art. no. 68620P*, in *Single Molecule Spectroscopy and Imaging*, J. Enderlein, Z.K. Gryczynski, and R. Erdmann, Editors. 2008. p. P8620-P8620.
37. COMSOL. *Navier-Stokes Equations*. [Web page] 2018 [cited 2018 June 10]; Available from: <https://www.comsol.com/multiphysics/navier-stokes-equations>.
38. Jorgenson, J.W. and Lukacs, K.D., *High-resolution separations based on Electrophoresis and electroosmosis*. Journal of Chromatography, 1981. **218**(1-3): p. 209-216.
39. Sze, A., Erickson, D., Ren, L.Q., and Li, D.Q., *Zeta-potential measurement using the Smoluchowski equation and the slope of the current-time relationship in electroosmotic flow*. Journal of Colloid and Interface Science, 2003. **261**(2): p. 402-410.
40. Mertz, J., Xu, C., and Webb, W.W., *Single-molecule detection by two-photon-excited fluorescence*. Optics Letters, 1995. **20**(24): p. 2532-2534.

41. Lakowicz, J.R., *Principles of Fluorescence Spectroscopy*. Third ed. 2006, New York: Springer. 954.
42. Hu, H., *Third Order Nonlinearity of Organic Molecules*, in *Optics and Photonics*. 2012, University of Central Florida: Orlando, Florida.
43. Boyd, R.W., *Nonlinear Optics*. Third ed. 2008: Academic Press. 613.
44. Novony, L. and Hecht, B., *Principles of Nano-Optics*. First ed. 2006, New York: Cambridge University Press. 539.
45. Shi, L.Y., Rodriguez-Contreras, A., and Alfano, R.R., *Gaussian beam in two-photon fluorescence imaging of rat brain microvessel*. *Journal of Biomedical Optics*, 2014. **19**(12).
46. Altera, *My First FPGA Design Tutorial*. 2008, San Jose, California: Altera Corporation. 52.
47. Altera, *Verilog HDL Basics*. 2011, Altera Corporation.
48. Cheung, P. *Digital Electronics 2*. [Web Page] 2018 [cited 2018 January 16]; Available from: http://www.ee.ic.ac.uk/pcheung/teaching/ee2_digital/.
49. Grier, D.D. *Diffusion Coefficients*. [Web Page] 1996 [cited 2017 June 21]; Available from: <http://physics.nyu.edu/grierlab/methods/node11.html>.
50. Korson, L., Drosthan.W, and Millero, F.J., *Viscosity of Water at Various Temperatures*. *Journal of Physical Chemistry*, 1969. **73**(1): p. 34-&.
51. Berg, H.C., *Diffusion: Microscopic Theory*, in *Random Walks in Biology*. 1993, Princeton University Press: New Jersey.
52. Probst, R. and Shapiro, B., *Three-dimensional electrokinetic tweezing: device design, modeling, and control algorithms*. *Journal of Micromechanics and Microengineering*, 2011. **21**(2).
53. Sayah, A., Solignac, D., Cueni, T., and Gijs, M.A.M., *Development of novel low temperature bonding technologies for microchip chemical analysis applications*. *Sensors and Actuators a-Physical*, 2000. **84**(1-2): p. 103-108.
54. Nakanishi, H., Nishimoto, T., Kanai, M., Saitoh, T., Nakamura, R., Yoshida, T., and Shoji, S., *Condition optimization, reliability evaluation of SiO₂-SiO₂HF bonding and its application for UV detection micro flow cell*. *Sensors and Actuators a-Physical*, 2000. **83**(1-3): p. 136-141.
55. Giridhar, M.S., Seong, K., Schulzgen, A., Khulbe, P., Peyghambarian, N., and Mansuripur, M., *Femtosecond pulsed laser micromachining of glass substrates with application to microfluidic devices*. *Applied Optics*, 2004. **43**(23): p. 4584-4589.
56. Huang, H. and Guo, Z.X., *Ultra-short pulsed laser PDMS thin-layer separation and micro-fabrication*. *Journal of Micromechanics and Microengineering*, 2009. **19**(5).
57. Jacobson, S.C., Hergenroder, R., Koutny, L.B., Warmack, R.J., and Ramsey, J.M., *Effects of injection schemes and column geometry on the performance of microchip electrophoresis devices*. *Analytical Chemistry*, 1994. **66**(7): p. 1107-1113.

58. Ceriotti, L., Weible, K., de Rooij, N.F., and Verpoorte, E., *Rectangular channels for lab-on-a-chip applications*. Microelectronic Engineering, 2003. **67-8**: p. 865-871.
59. Koschwanetz, J.H., Carlson, R.H., and Meldrum, D.R., *Thin PDMS Films Using Long Spin Times or Tert-Butyl Alcohol as a Solvent*. Plos One, 2009. **4**(2).
60. Emslie, A.G., Bonner, F.T., and Peck, L.G., *Flow of a viscous liquid on a rotating disk*. Journal of Applied Physics, 1958. **29**(5): p. 858-862.
61. Davis, L.M., Canfield, B.K., Germann, J.A., King, J.K., Robinson, W.N., Dukes, A.D., Rosenthal, S.J., Samson, P.C., and Wikswo, J.P., *Four-focus single-particle position determination in a confocal microscope*, in *Single Molecule Spectroscopy and Imaging Iii*, J. Enderlein, Z.K. Grycznski, and R. Erdmann, Editors. 2010.
62. LinearTechnology. *Quad 12-, 14- and 16-Bit Voltage Output SoftSpan DACs with Readback*. 2006 [cited 2017 October 10]; Available from: <https://www.analog.com/media/en/technical-documentation/data-sheets/2704fd.pdf>.
63. Cypress. *EZ-USB SX2™ High-Speed USB Interface Device*. 2017 [cited 2017 November 5]; Available from: <https://www.cypress.com/file/124346/download>.
64. Robinson, W.N. and Davis, L.M., *Simulation of single-molecule trapping in a nanochannel*. Journal of Biomedical Optics, 2010. **15**(4).
65. Williams, E.H., Davydov, A.V., Motayed, A., Sundaresan, S.G., Bocchini, P., Richter, L.J., Stan, G., Steffens, K., Zangmeister, R., Schreifels, J.A., and Rao, M.V., *Immobilization of streptavidin on 4H-SiC for biosensor development*. Applied Surface Science, 2012. **258**(16): p. 6056-6063.
66. ThermoFisher. *FluoSpheres® Fluorescent Microspheres*. 2005 October 25 [cited 2018 August 14]; Available from: <https://assets.thermofisher.com/TFS-Assets/LSG/manuals/mp05000.pdf>.
67. Duggal, D., Nagwekar, J., Rich, R., Midde, K., Fudala, R., Gryczynski, I., and Borejdo, J., *Phosphorylation of myosin regulatory light chain has minimal effect on kinetics and distribution of orientations of cross bridges of rabbit skeletal muscle*. American Journal of Physiology-Regulatory Integrative and Comparative Physiology, 2014. **306**(4): p. R222-R233.

APPENDICES

Appendix A. Schematics of the Perspex Hood

The following two figures (made using SolidWorks design software) show the dimensions and features of the hood used to interface fluid reservoirs to the cross-channel device. The hood is made from poly(methyl methacrylate), otherwise known by the brand name Perspex. Four cylindrical through-holes were drilled to facilitate the fluid flow from the reservoirs to the cross-channel placed underneath with four fluid inlets. There are four rubber O-rings sandwiched with a gentle press between the Perspex hood and the cross-channel to have a firm connection among them so that no fluid leakage occurs at the interface. Other than these four O-rings, there are another four O-rings placed on the bottom of the Perspex hood to balance and spread the pressure forces that are exerted on the cross-channel device by the Perspex hood.

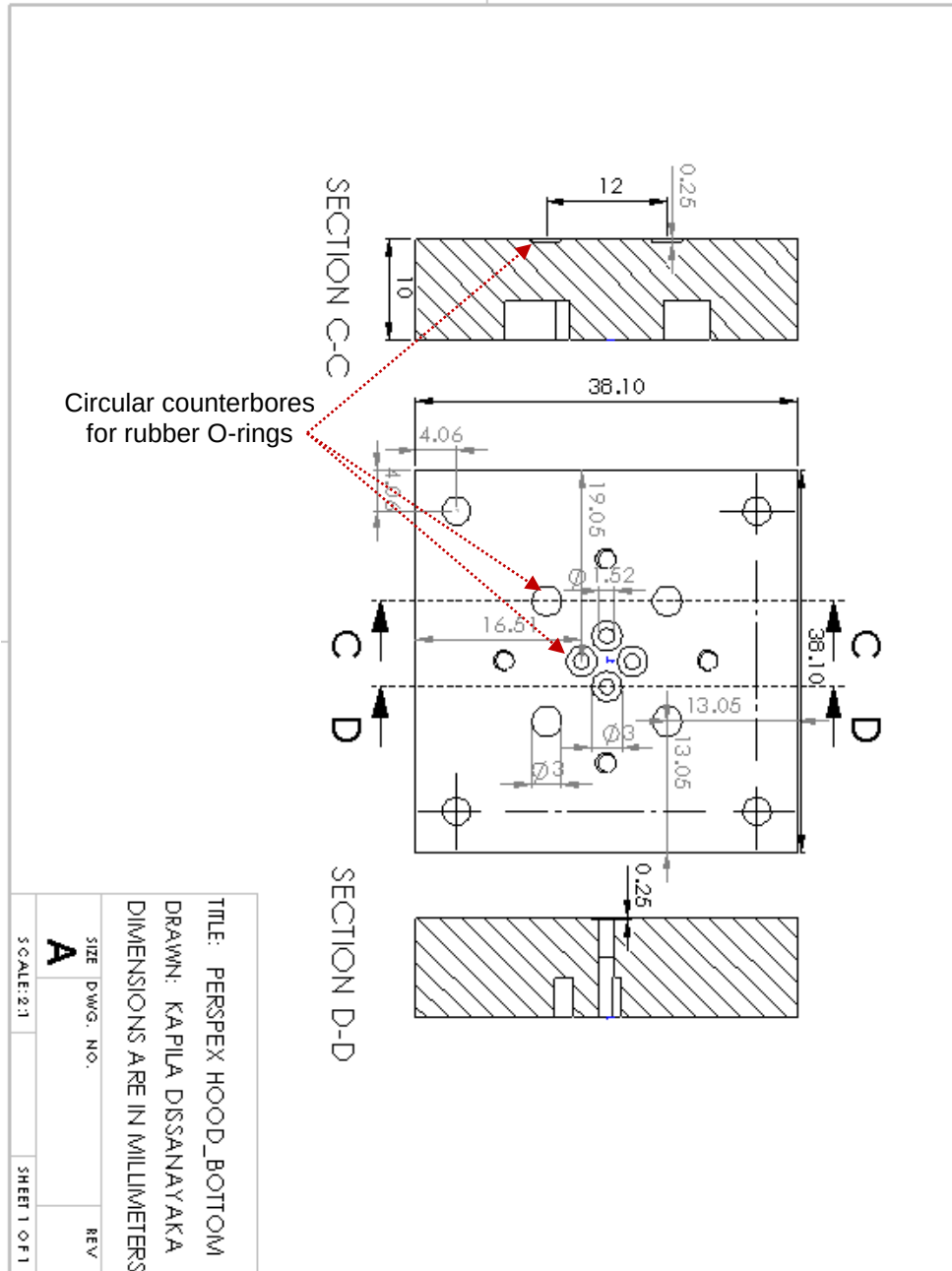


Figure A.2. Schematic of the bottom of the Perspex hood.

The bottom of the Perspex hood interfaces with the top of the cross-channel device using rubber O-rings.

Appendix B. Schematics of the FPGA Program

The schematic diagram of the total logical program that was created in the FPGA via Altera Quartus II is shown in the following diagrams. Each diagram is connected to another diagram with a connector (i.e., with an encircled letter) to show the continuity of the connections.

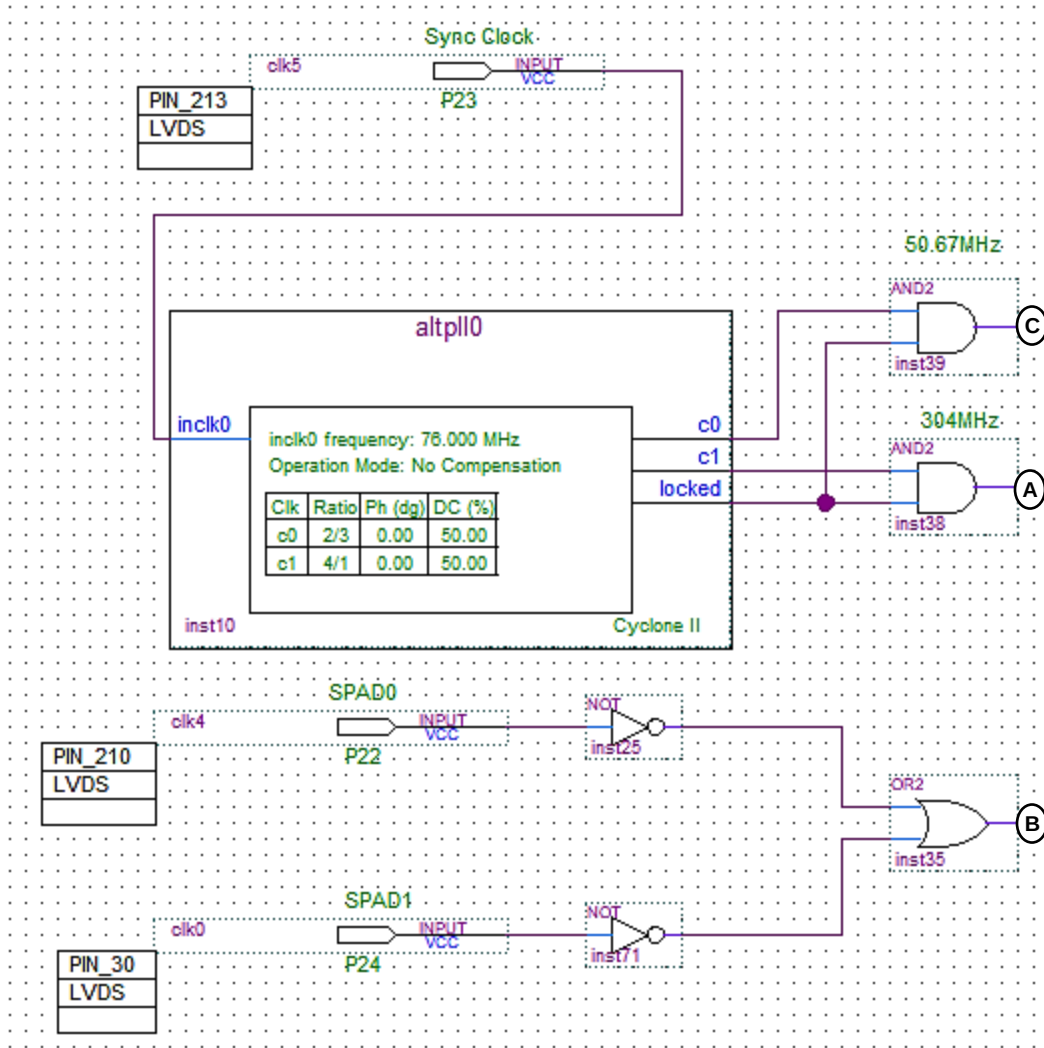


Figure B.1. FPGA interfaced with SPAD 0,1 and 76 MHz external synchronous clock.

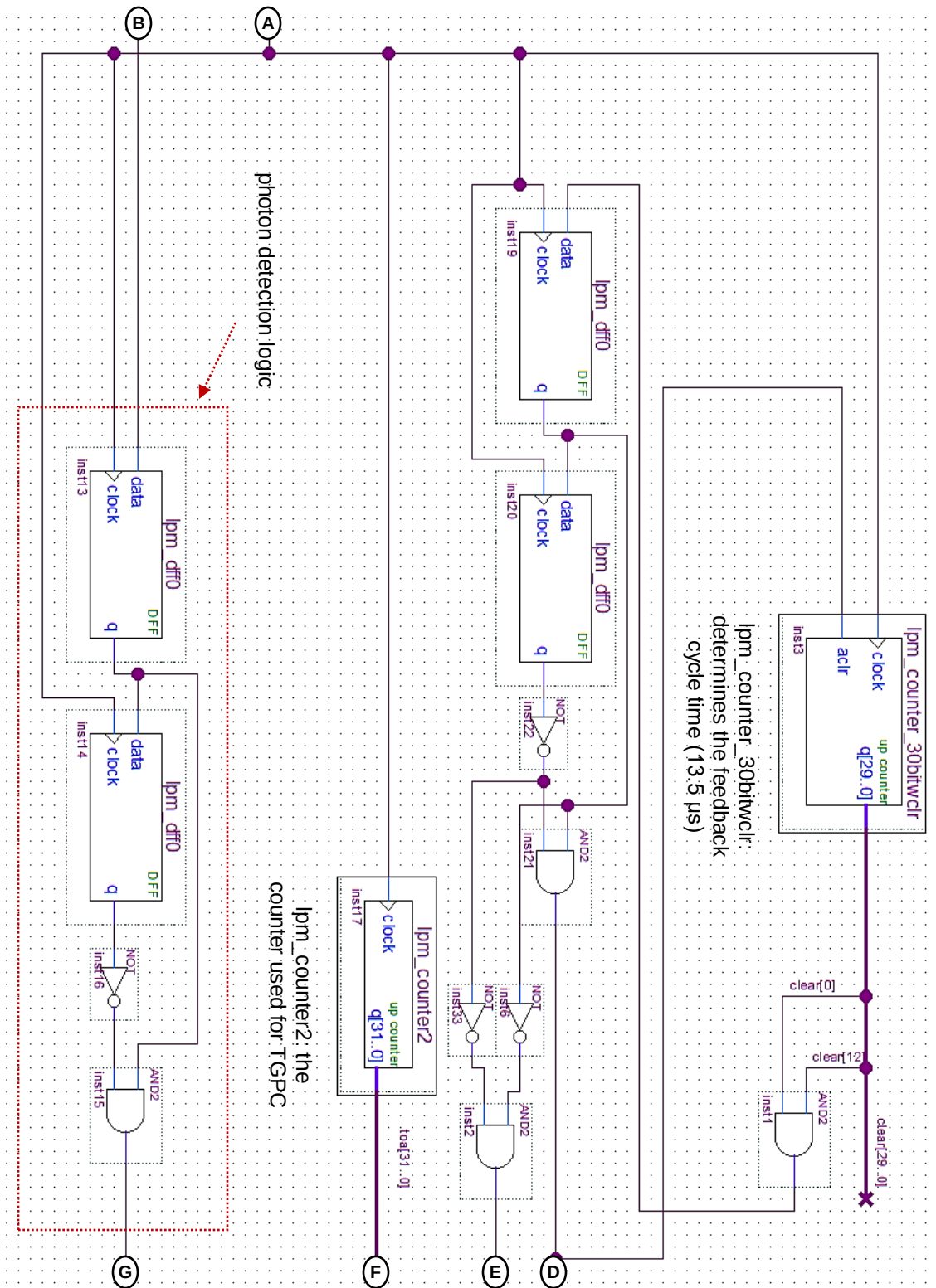


Figure B.2. Photon detection circuitry and main triggering counters.

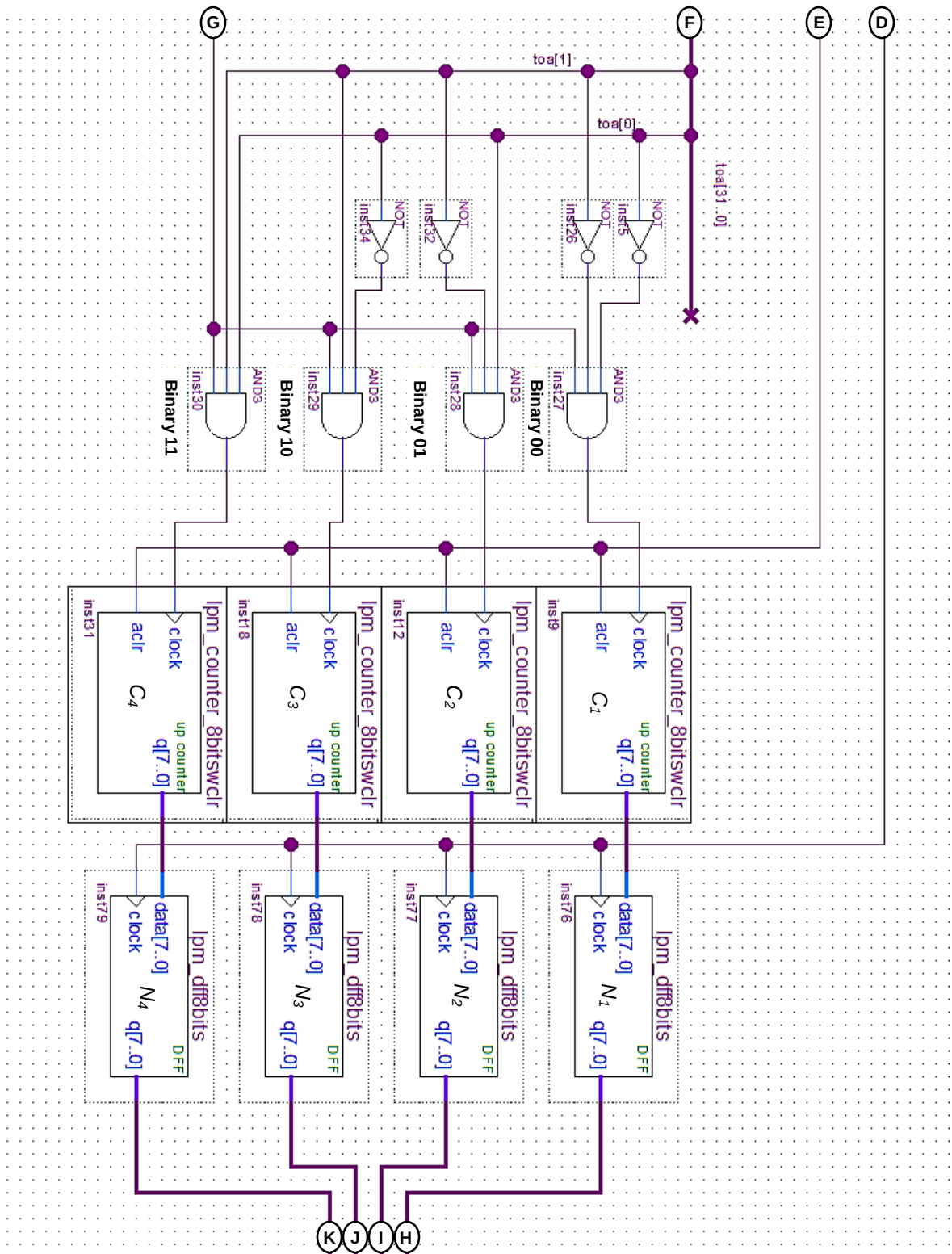


Figure B.3. Four time gates and four photon counters with buffers.

Appendix C. User-defined Verilog Modules in FPGA

“counts_to_voltages_6_2” Module

This Verilog module (clocked at 304 MHz) is dedicated to converting the binned photons N_1 – N_4 into the voltages used for trapping V_1 – V_4 . These voltage values are represented as 16-bit digital values and they are programmed to take only the values: ± 10 V, ± 5 V and 0 V. The voltage values are selected in such a way that they always sum to zero (i.e., $V_1 + V_2 + V_3 + V_4 = 0$). The module determines the voltage values once every feedback iteration ($2^{12}/304$ MHz = 13.5 μ s). After determining the voltage values, this module triggers the “darpa_dac_2” module to carry out the activation of the voltages. It should be noted that the “counts_to_voltages_6_2” module also makes sure that the voltage values go back to zero before the next feedback iteration.

The double forward slash symbol “//” is used to denote single line comments in Verilog. The “//-----//” is used to categorize the inputs, outputs, and variables.

```
module counts_to_voltages_6_2(count1,count2,count3,count4,clock,count_read,test,dac_iter,tristate_reain,
V1,V2,V3,V4,read_volts,check,dac_enable,tristate_renaout);

//-----inputs-----//
input clock,count_read,dac_iter,test,tristate_reain;
input [7:0] count1;input [7:0] count2;input [7:0] count3;input [7:0] count4;
//-----end inputs-----//

//-----outputs-----//
output [15:0] V1;output [15:0] V2;output [15:0] V3;output [15:0] V4;
output tristate_renaout,check,dac_enable,read_volts;
//-----end outputs-----//

//----regs for outputs-----//
reg [15:0] V1;reg [15:0] V2;reg [15:0] V3;reg [15:0] V4;
reg tristate_renaout,check,dac_enable,read_volts;
```

```

//----end regs for outputs-----//

//-----variables-----//
reg [7:0] temp_next;reg [7:0] next_state;reg [7:0] present_state;
reg [7:0] N1;//Path A1
reg [7:0] N2;//Path A2
reg [7:0] N3;//Path B1
reg [7:0] N4;//Path B2
reg [10:0] CA12; //N1+N2
reg [10:0] CA34; //N3+N4
reg [10:0] CA13; //N1+N3
reg [10:0] CA24; //N2+N4
reg [10:0] CA14; //N1+N4
reg [10:0] CA23; //N2+N3
reg [10:0] CA; //N1+N2+N3+N4
//-----end variables-----//

initial
begin
//-----outputs-----//
check = 0;dac_enable = 0;read_volts = 0;tristate_renaout=0;
//-----end outputs-----//

next_state = 8'b0000_0000;
present_state = 8'b0000_0000;
V1 = 16'b1000_0000_0000_0000; V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000; V4 = 16'b1000_0000_0000_0000;//V1=0,V2=0,V3=0,V4=0.
N1 = 8'b0000_0000; N2 = 8'b0000_0000; N3 = 8'b0000_0000; N4 = 8'b0000_0000;
end

always @ (posedge clock)
begin
    present_state = next_state;
end

always @ (posedge count_read)
begin
    N1 = count1;N2 = count2;N3 = count3;N4 = count4;
end

always @ (negedge count_read)
begin
    CA12 = N1+N2;CA13 = N1+N3;CA14 = N1+N4;CA23 = N2+N3;CA24 = N2+N4;
    CA34 = N3+N4;CA = N1+N2+N3+N4;
end

always @ (posedge tristate_retain)
begin
case (temp_next)
0:    begin
        if ((CA13 > CA24)&&(CA23 > CA14)&&(CA34 > CA12))
            begin
                //01//X-Y-Z-//Vx+,Vy+,Vz+
                V1 = 16'b1000_0000_0000_0000;V2 = 16'b0000_0000_0000_0000; //V1=0 V, V2 =-10 V
                V3 = 16'b1111_1111_1111_1111;V4 = 16'b1000_0000_0000_0000; //V3=10 V, V4 =0 V
            end
        else if ((CA13 > CA24)&&(CA23 > CA14)&&(CA34 < CA12))
            begin
                //02//X-Y-Z-//Vx+,Vy+,Vz-
                V1 = 16'b1111_1111_1111_1111;V2 = 16'b1000_0000_0000_0000;
                V3 = 16'b1000_0000_0000_0000;V4 = 16'b0000_0000_0000_0000;
            end
        else if ((CA13 > CA24)&&(CA23 > CA14)&&(CA34 == CA12))
            begin
                //03//X-Y-Z0//Vx+,Vy+,Vz0
                V1 = 16'b1111_1111_1111_1111;V2 = 16'b0000_0000_0000_0000;
            end
        end
    end
end

```

```

V3 = 16'b1111_1111_1111_1111;V4 = 16'b0000_0000_0000_0000;
end
else if ((CA13 > CA24)&&(CA23 < CA14)&&(CA34 > CA12))
begin
//04//X-Y+Z-//Vx+,Vy-,Vz+
V1 = 16'b1000_0000_0000_0000;V2 = 16'b0000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1111_1111_1111_1111;
end
else if ((CA13 > CA24)&&(CA23 < CA14)&&(CA34 < CA12))
begin
//05//X-Y+Z+//Vx+,Vy-,Vz-
V1 = 16'b1111_1111_1111_1111;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b0000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;
end
else if ((CA13 > CA24)&&(CA23 < CA14)&&(CA34 == CA12))
begin
//06//X-Y+Z0//Vx+,Vy-,Vz0
V1 = 16'b1111_1111_1111_1111;V2 = 16'b0000_0000_0000_0000;
V3 = 16'b0000_0000_0000_0000;V4 = 16'b1111_1111_1111_1111;
end
else if ((CA13 > CA24)&&(CA23 == CA14)&&(CA34 > CA12))
begin
//07//X-Y0Z-//Vx+,Vy0,Vz+
V1 = 16'b1000_0000_0000_0000;V2 = 16'b0000_0000_0000_0000; //V1=0 V, V2=-10 V
V3 = 16'b1100_0011_0101_0000;V4 = 16'b1100_0011_0101_0000; //V3=5 V, V4=5 V
end
else if ((CA13 > CA24)&&(CA23 == CA14)&&(CA34 < CA12))
begin
//08//X-Y0Z+//Vx+,Vy0,Vz-
V1 = 16'b1111_1111_1111_1111;V2 = 16'b1000_0000_0000_0000; //V1=10 V, V2=0 V
V3 = 16'b0011_1100_1011_0000;V4 = 16'b0011_1100_1011_0000; //V3=-5 V, V4=-5 V
end
else if ((CA13 > CA24)&&(CA23 == CA14)&&(CA34 == CA12))
begin
//09//X-Y0Z0//Vx+,Vy0,Vz0
V1 = 16'b1111_1111_1111_1111;V2 = 16'b0000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;
end
else if ((CA13 < CA24)&&(CA23 > CA14)&&(CA34 > CA12))
begin
//10//X+Y-Z-//Vx-,Vy+,Vz+
V1 = 16'b0000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1111_1111_1111_1111;V4 = 16'b1000_0000_0000_0000;
end
else if ((CA13 < CA24)&&(CA23 > CA14)&&(CA34 < CA12))
begin
//11//X+Y-Z+//Vx-,Vy+,Vz-
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1111_1111_1111_1111;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b0000_0000_0000_0000;
end
else if ((CA13 < CA24)&&(CA23 > CA14)&&(CA34 == CA12))
begin
//12//X+Y-Z0//Vx-,Vy+,Vz0
V1 = 16'b0000_0000_0000_0000;V2 = 16'b1111_1111_1111_1111;
V3 = 16'b1111_1111_1111_1111;V4 = 16'b0000_0000_0000_0000;
end
else if ((CA13 < CA24)&&(CA23 < CA14)&&(CA34 > CA12))
begin
//13//X+Y+Z-//Vx-,Vy-,Vz+
V1 = 16'b0000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1111_1111_1111_1111;
end
else if ((CA13 < CA24)&&(CA23 < CA14)&&(CA34 < CA12))
begin
//14//X+Y+Z+//Vx-,Vy-,Vz-
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1111_1111_1111_1111;

```

```

V3 = 16'b0000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;
end
else if ((CA13 < CA24)&&(CA23 < CA14)&&(CA34 == CA12))
begin
//15//X+Y+Z0//Vx-,Vy-,Vz0
V1 = 16'b0000_0000_0000_0000;V2 = 16'b1111_1111_1111_1111;
V3 = 16'b0000_0000_0000_0000;V4 = 16'b1111_1111_1111_1111;
end
else if ((CA13 < CA24)&&(CA23 == CA14)&&(CA34 > CA12))
begin
//16//X+Y0Z-//Vx-,Vy0,Vz+
V1 = 16'b0000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000; //V1=-10 V, V2 =0 V
V3 = 16'b1100_0011_0101_0000;V4 = 16'b1100_0011_0101_0000; //V3=5 V, V4 =5 V
end
else if ((CA13 < CA24)&&(CA23 == CA14)&&(CA34 < CA12))
begin
//17//X+Y0Z+//Vx-,Vy0,Vz-
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1111_1111_1111_1111; //V1=0 V, V2 =10 V
V3 = 16'b0011_1100_1011_0000;V4 = 16'b0011_1100_1011_0000; //V3=-5 V, V4 =-5 V
end
else if ((CA13 < CA24)&&(CA23 == CA14)&&(CA34 == CA12))
begin
//18//X+Y0Z0//Vx-,Vy0,Vz0
V1 = 16'b0000_0000_0000_0000;V2 = 16'b1111_1111_1111_1111;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;
end
else if ((CA13 == CA24)&&(CA23 > CA14)&&(CA34 > CA12))
begin
//19//X0Y-Z-//Vx0,Vy+,Vz+
V1 = 16'b0011_1100_1011_0000;V2 = 16'b0011_1100_1011_0000; //V1=-5 V, V2 =-5 V
V3 = 16'b1111_1111_1111_1111;V4 = 16'b1000_0000_0000_0000; //V3=10 V, V4 = 0 V
end
else if ((CA13 == CA24)&&(CA23 > CA14)&&(CA34 < CA12))
begin
//20//X0Y-Z+//Vx0,Vy+,Vz-
V1 = 16'b1100_0011_0101_0000;V2 = 16'b1100_0011_0101_0000; //V1=5 V, V2 =5 V
V3 = 16'b1000_0000_0000_0000;V4 = 16'b0000_0000_0000_0000; //V3=0 V, V4 =-10 V
end
else if ((CA13 == CA24)&&(CA23 > CA14)&&(CA34 == CA12))
begin
//21//X0Y-Z0//Vx0,Vy+,Vz0
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1111_1111_1111_1111;V4 = 16'b0000_0000_0000_0000;
end

else if ((CA13 == CA24)&&(CA23 < CA14)&&(CA34 > CA12))
begin
//22//X0Y+Z-//Vx0,Vy-,Vz+
V1 = 16'b0011_1100_1011_0000;V2 = 16'b0011_1100_1011_0000; //V1=-5 V, V2 =-5 V
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1111_1111_1111_1111; //V3=0 V, V4 =10 V
end
else if ((CA13 == CA24)&&(CA23 < CA14)&&(CA34 < CA12))
begin
//23//X0Y+Z+//Vx0,Vy-,Vz-
V1 = 16'b1100_0011_0101_0000;V2 = 16'b1100_0011_0101_0000; //V1=5 V, V2 =5 V
V3 = 16'b0000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000; //V3=-10 V, V4 =0 V
end
else if ((CA13 == CA24)&&(CA23 < CA14)&&(CA34 == CA12))
begin
//24//X0Y+Z0//Vx0,Vy-,Vz0
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b0000_0000_0000_0000;V4 = 16'b1111_1111_1111_1111;
end
else if ((CA13 == CA24)&&(CA23 == CA14)&&(CA34 > CA12))
begin
//25//X0Y0Z-//Vx0,Vy0,Vz+

```

```

V1 = 16'b0000_0000_0000_0000;V2 = 16'b0000_0000_0000_0000;
V3 = 16'b1111_1111_1111_1111;V4 = 16'b1111_1111_1111_1111;
end
else if ((CA13 == CA24)&&(CA23 == CA14)&&(CA34 < CA12))
begin
//26//X0Y0Z+/Vx0,Vy0,Vz-
V1 = 16'b1111_1111_1111_1111;V2 = 16'b1111_1111_1111_1111;
V3 = 16'b0000_0000_0000_0000;V4 = 16'b0000_0000_0000_0000;
end
else if ((CA13 == CA24)&&(CA23 == CA14)&&(CA34 == CA12))
begin
//27//X0Y0Z0/Vx0,Vy0,Vz0
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;
end
else
begin
//Default
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;
end
end
1: begin
// V1=0V,V2=0V,V3=0V,V4=0V
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;

end

2: begin
// V1=10V,V2=10V,V3=-10V,V4=-10V // move fluid from Z+ to Z-
V1 = 16'b1111_1111_1111_1111;V2 = 16'b1111_1111_1111_1111;
V3 = 16'b0000_0000_0000_0000;V4 = 16'b0000_0000_0000_0000;

end

default: begin
// V1=0V,V2=0V,V3=0V,V4=0V
V1 = 16'b1000_0000_0000_0000;V2 = 16'b1000_0000_0000_0000;
V3 = 16'b1000_0000_0000_0000;V4 = 16'b1000_0000_0000_0000;

end

endcase
end
//-----//
always @*
begin
case (present_state)
0: begin
temp_next=0; tristate_renaout=0; dac_enable=0;
next_state = 1;

end
1: begin
if (count_read)
begin
temp_next=0; tristate_renaout=0; dac_enable=0;
next_state = 3;

end
else
begin
temp_next=0; tristate_renaout=0; dac_enable=0;
next_state = 2;

end
end
2: begin
temp_next=0; tristate_renaout=0; dac_enable=0;
next_state = 1;

end
3: begin

```

```

        if (CA > 2)//CA = N1+N2+N3+N4;
            begin
                temp_next=0; tristate_renaout=0; dac_enable=0;
                next_state = 5;
            end
        else
            begin
                temp_next=2; tristate_renaout=0; dac_enable=0;
                next_state = 4;
            end
        end
4:    begin
        //Set voltages to move fluid to bring molecules towards the ROI
        temp_next=2; tristate_renaout=1; dac_enable=0;
        next_state = 6;//0
    end
5:begin
        temp_next=0; tristate_renaout=1; dac_enable=0;
        next_state = 6;
    end
6:begin
        temp_next=0; tristate_renaout=0; dac_enable=1;
        next_state = 7;
    end
7:begin
        temp_next=0; tristate_renaout=0; dac_enable=0;
        next_state = 8;
    end
8:begin
    if (dac_iter)
        begin
            temp_next=0; tristate_renaout=0; dac_enable=0;
            next_state = 10;
        end
    else
        begin
            temp_next=0; tristate_renaout=0; dac_enable=0;
            next_state = 9;
        end
    end
9:begin
        temp_next=0; tristate_renaout=0; dac_enable=0;
        next_state = 8;
    end
10:begin
    if (test)
        begin
            temp_next=1; tristate_renaout=0; dac_enable=0;
            next_state = 12;
        end
    else
        begin
            temp_next=0; tristate_renaout=0; dac_enable=0;
            next_state = 11;
        end
    end
11:begin
        temp_next=0; tristate_renaout=0; dac_enable=0;
        next_state = 10;
    end
12:begin
        temp_next=1; tristate_renaout=1; dac_enable=0;
        next_state = 13;
    end
13:begin
        temp_next=0; tristate_renaout=0; dac_enable=1;

```

```

        next_state = 14;
    end
14:begin
        temp_next=0; tristate_renaout=0; dac_enable=0;
        next_state = 0;
    end
default: begin
        temp_next=0; tristate_renaout=0; dac_enable=0;
        next_state = 0;
    end
endcase
end
endmodule

```

“darpa_dac_2” Module

This Verilog module (clocked at 50.67 MHz) is dedicated to activating the voltages V_1 – V_4 via the voltage driving IC LTC2704. The IC has four digital-to-analog converters (DAC)—DAC-A, DAC-B, DAC-C, and DAC-D—to output the four voltages. In order to write the relevant values in the DACs, the “darpa_dac_2” module has to send commands to the voltage driving IC serially. Once the writing of the values are completed, the “darpa_dac_2” modules sends the final command to the voltage driving IC to activate all four voltages simultaneously so that the 3D manipulation of particles via electroosmosis in the center of the cross-channel can take place.

An earlier version of the program for the “darpa_dac_2” module was originally coded by Mr. Isaac P. Lescano-Mendoza, who was a graduate student at UTSI (MS, Electrical Engineering, 2008). That code only addressed the functionality of DAC-A and DAC-B of the voltage driving IC (LTC2704) and was extended and further developed by Mr. Kapila D. Dissanayaka (the author of this dissertation) to add the functionality for DAC-C and DAC-D.

```
module darpa_dac_2(dac_datain1, dac_datain2, dac_datain3, dac_datain4, clock, sdodac, dacwrite,
nclrdac, nldac, ncsdac, sdidac, sckdac, dacwr_ack);
```

```
input clock, sdodac, dacwrite;
input [15:0] dac_datain1;
input [15:0] dac_datain2;
input [15:0] dac_datain3;
input [15:0] dac_datain4;
```

```
output nclrdac, nldac, ncsdac, sdidac, sckdac, dacwr_ack;
```

```
reg nclrdac, nldac, ncsdac, sdidac, sckdac, dacwr_ack=1;
```

```
reg [10:0] present_state, next_state;
```

```
reg [3:0] dac_control1 = 4'b0011;
reg [3:0] dac_control2 = 4'b0011;
reg [3:0] dac_control3 = 4'b0011;
reg [3:0] dac_control4 = 4'b0011;
reg [3:0] dac_address1 = 4'b0000; // address for DAC_A_Port 15
reg [3:0] dac_address2 = 4'b0010; // address for DAC_B_Port 16
reg [3:0] dac_address3 = 4'b0100; // address for DAC_C_Port 17
reg [3:0] dac_address4 = 4'b0110; // address for DAC_D_Port 18
reg [15:0] dac_code1 = 16'b1000_0000_0000_0000;
reg [15:0] dac_code2 = 16'b1000_0000_0000_0000;
reg [15:0] dac_code3 = 16'b1000_0000_0000_0000;
reg [15:0] dac_code4 = 16'b1000_0000_0000_0000;
```

```
parameter [23:0] SPAN = 24'b0010_1111_0000_0000_0000_0011;
```

```
parameter [10:0] s0 = 0; parameter [10:0] s1 = 1; parameter [10:0] s2 = 2; parameter [10:0] s3 = 3;
```

```
/*
```

The list of parameters for “s” go up to s257 and the definitions are exactly as shown above.

```
*/
```

```
always @(posedge clock)
```

```
begin
```

```
    present_state = next_state;
```

```
end
```

```
always @(posedge dacwrite )
```

```
begin
```

```
    dac_code1[15:0] = dac_datain1[15:0];
    dac_code2[15:0] = dac_datain2[15:0];
    dac_code3[15:0] = dac_datain3[15:0];
    dac_code4[15:0] = dac_datain4[15:0];
```

```
end
```

```
always @(*)
```

```
begin
```

```
case(present_state)
```

```
    s0: begin
```

```
        nclrdac=0;           //clear=LOW code and span B2 registers for all DACs
        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=0;
        next_state = s1;
    end
```

```
    s1: begin
```

```
        nclrdac=0;           //clear=LOW code and span B2 registers for all DACs
```

```

        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=0;
        next_state = s2;
    end

s2: begin
        nclrdac=0;           //clear=LOW code and span B2 registers for all DACs
        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=0;
        next_state = s3;
    end

s3: begin
        nclrdac=0;           //clear=LOW code and span B2 registers for all DACs
        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=0;
        next_state = s4;
    end

s4: begin
        nclrdac=0;           //clear=LOW code and span B2 registers for all DACs
        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=1;
        next_state = s5;
    end

s5: begin
        nclrdac=1;           //clear=HIGH (stop clear )
        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=1;
        next_state = s6;
    end

s6: begin
        nclrdac=1;
        nldac=1;
        ncsdac=0;           // chip select = LOW
        sdidac=0;
        sckdac=0;
        dacwr_ack=1;
        next_state = s7;
    end

//=====
//Writing to SPAN
//=====
s7: begin
        nclrdac=1;
        nldac=1;
        ncsdac=0;           // chip select = LOW
        sdidac=SPAN[23]; //write SDI data
        sckdac=0;
        dacwr_ack=1;
        next_state = s8;
    end

s8: begin
        nclrdac=1;

```

```

        nldac=1;
        ncsdac=0;           // chip select = LOW
        sdidac=SPAN[23];
        sckdac=1;           //clock dac positive edge to write serial data
        dacwr_ack=1;
        next_state = s9;
    end

/*
The cases sN with N=9,11,13,..53 vary as follows with x=23,22,21,20,...0:
sN: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;           // chip select = LOW
    sdidac=SPAN[x];     //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = sN+1;
    end

The cases sN+1 with N=9,11,13,..53 vary as follows with x=22,21,20,...0:
sN+1:begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;           // chip select = LOW
    sdidac=SPAN[x];
    sckdac=1;           //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = sN+2;
    end

*/

s55: begin
    nclrdac=1;
    nldac=1;
    ncsdac=1;           // chip select = HIGH load B1
    sdidac=0; //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s56;
    end

s56: begin
    nclrdac=1;
    nldac=0; //LDAC = active LOW load B2
    ncsdac=1;           // chip select = LOW
    sdidac=0;
    sckdac=0;           //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s57;
    end

//=====
//Writing to DAC A
//=====

s57: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;           // chip select = LOW
    sdidac=dac_control1[3]; //write SDI data

```

```

        sckdac=0;
        dacwr_ack=1;           //clear FF of dacwrite
        next_state = s58;
    end

s58: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;                 // chip select = LOW
    sdidac=dac_control1[3];
    sckdac=1;                 //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s59;
    end

s59: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;                 // chip select = LOW
    sdidac=dac_control1[2];   //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s60;
    end

s60: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;                 // chip select = LOW
    sdidac=dac_control1[2];
    sckdac=1;                 //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s61;
    end

s61: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;                 // chip select = LOW
    sdidac=dac_control1[1];   //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s62;
    end

s62: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;                 // chip select = LOW
    sdidac=dac_control1[1];
    sckdac=1;                 //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s63;
    end

s63: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;                 // chip select = LOW
    sdidac=dac_control1[0];   //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s64;
    end

s64: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;                 // chip select = LOW
    sdidac=dac_control1[0];
    sckdac=1;                 //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s65;

```

```

end
s65: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_address1[3]; //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s66;
end
s66: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_address1[3];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s67;
end
s67: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_address1[2]; //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s68;
end
s68: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_address1[2];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s69;
end
s69: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_address1[1]; //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s70;
end
s70: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_address1[1];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s71;
end
s71: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_address1[0]; //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s72;
end
s72: begin
    nclrdac=1;

```

```

nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_address1[0];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s73;
end

s73: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[15];    //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s74;
end

s74: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[15];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s75;
end

s75: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[14];    //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s76;
end

s76: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[14];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s77;
end

s77: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[13];    //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s78;
end

s78: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[13];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s79;
end

s79: begin
nclrdac=1;
nldac=1;

```

```

ncsdac=0;          // chip select = LOW

sddac=dac_code1[12];    //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s80;end

s80: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sddac=dac_code1[12];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s81;
end

s81: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sddac=dac_code1[11];    //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s82;
end

s82: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sddac=dac_code1[11];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s83;
end

s83: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sddac=dac_code1[10];    //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s84;
end

s84: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sddac=dac_code1[10];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s85;
end

s85: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sddac=dac_code1[9];    //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s86;
end

s86: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sddac=dac_code1[9];

```

```

sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s87;
end

s87: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[8]; //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s88;
end

s88: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[8];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s89;
end

s89: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[7]; //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s90;
end

s90: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[7];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s91;
end

s91: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[6]; //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s92;
end

s92: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[6];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s93;
end

s93: begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[5]; //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s94;
end

```

```

end
s94: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_code1[5];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s95;
end
s95: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_code1[4];    //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s96;
end
s96: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_code1[4];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s97;
end
s97: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_code1[3];    //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s98;
end
s98: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_code1[3];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s99;
end
s99: begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_code1[2];    //write SDI data
    sckdac=0;
    dacwr_ack=1;
    next_state = s100;
end
s100:
begin
    nclrdac=1;
    nldac=1;
    ncsdac=0;          // chip select = LOW
    sdidac=dac_code1[2];
    sckdac=1;          //clock dac positive edge to write serial data
    dacwr_ack=1;
    next_state = s101;
end
s101:
begin
    nclrdac=1;

```

```

nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[1]; //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s102;
end
s102:
begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[1];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s103;
end
s103:
begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[0]; //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s104;
end
s104:
begin
nclrdac=1;
nldac=1;
ncsdac=0;          // chip select = LOW
sdidac=dac_code1[0];
sckdac=1;          //clock dac positive edge to write serial data
dacwr_ack=1;
next_state = s105;
end
s105:
begin
nclrdac=1;
nldac=1;
ncsdac=1;          // chip select = HIGH load B1
sdidac=0; //write SDI data
sckdac=0;
dacwr_ack=1;
next_state = s106;
end

```

/*

Note: How to write into other DACs?:

*There are four DACs used in this code to drive V1, V2, V3 and V4.

*V1 is for DAC-A, V2 is for DAC-B, V3 is for DAC-C, V4 is for DAC-D.

*The codes for DAC-B,DAC-C and DAC-D are similar to the codes written for DAC A except the code for "sdidac". The "sdidac" should be assigned with the proper variables (dac_address, dac_code1,2,3,4) appropriately as defined in the beginning of the program.

* The next starting case is s106 and it ends at s251.

*The next_state variable in each case should be incremented by 1.

*/

```

end
s252:
begin
nclrdac=1;
nldac=1;

```

```

        ncsdac=1;          // chip select = HIGH load B1
        sdidac=0; //write SDI data
        sckdac=0;
        dacwr_ack=1;
        next_state = s253;
    end

//=====
//LOADING ALL DACS SIMULTANEOUSLY
//=====

    s253:        begin
        nclrdac=1;
        nldac=0; //LDAC = active LOW load B2
        ncsdac=1;
        sdidac=0;
        sckdac=0;          //clock dac positive edge to write serial data
        dacwr_ack=0;
        next_state = s254;
    end

//=====
//WAIT UNTIL THE DFF TRIGGERS THE DARPA_DAC
//=====

    s254:        begin
        nclrdac=1;
        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=1;

        if (dacwrite == 1)
            begin
                next_state = s57;
            end
        else
            begin
                next_state = s255;
            end
        end
    end

    s255:        begin
        nclrdac=1;
        nldac=1;
        ncsdac=1;
        sdidac=0;
        sckdac=0;
        dacwr_ack=1;
        next_state = s254;
    end

    default: next_state = s0;

endcase
end
endmodule

```

“usb_descriptor” Module

The “usb_descriptor” module (clocked at 50.67 MHz) is dedicated to export (every 13.5 μ s) the binned photons N_1 – N_4 to a desktop computer via a USB 2.0 interface (CY7C68001). The FPGA sends the commands serially to the CY7C68001 to configure the USB connection (as mentioned in the CY7C68001 manual). Once this one-time configuration is completed, the buffered binned photons are exported continuously.

The Verilog program for the “usb_descriptor” module was coded by Mr. Isaac P. Lescano-Mendoza and was embedded in the main FPGA program of the experiment that is discussed in this dissertation to export photon counts to a desktop computer.

```
module usb_descriptor(clock,ready,datain,fifoempty,nint,flaga_fixed,fifo_read, fifo_rdclk,
usbffoadr,slwr,slcs,pktend,nreset,fd_out,clk_rdreg,highspeed,slrd, testpoint1, testpoint2);
```

```
input clock, ready; input [31:0] datain; input fifoempty; input nint; input flaga_fixed;
```

```
output fifo_read, fifo_rdclk, slwr, pktend,clk_rdreg; output slcs = 0; output nreset = 1;
```

```
output [2:0] usbffoadr; output [15:0] fd_out; output highspeed; output slrd;
output testpoint1, testpoint2;
```

```
reg clk_rdreg=0;
reg nreset=1, slcs=1, pktend=1;
reg fifo_read, fifo_rdclk; reg highspeed=1;
```

```
reg testpoint1=0; reg testpoint2=0;
reg [2:0] usbffoadr= 3'b100;
reg slwr=1; reg slrd=1;
reg [15:0] fd_out=0; reg [9:0] ns; reg [9:0] cs; reg [15:0]counter;
```

```
parameter s0=10'd0;parameter s1=10'd1;parameter s2=10'd2;parameter s3=10'd3;
/*
```

Note:

The list of parameters to be defined are s0 to s49 and s390 to s414 and the definitions are exactly as shown above.

```
*/
```

```
always @(posedge clock)
```

```
begin
```

```
    counter = counter + 1;
```

```
end
```

```
always @ (posedge counter[2])
```

```

begin
    cs = ns;
end
always @(*)
begin
    case (cs)
        s0: begin
            fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
            usbffifoadr = 3'b100;
            pktend = 1; nreset = 1; slcs = 0; slrd = 1; slwr = 1; fd_out = 0;
            ns = s1;
        end
        s1: begin
            fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
            usbffifoadr = 3'b100;
            pktend = 1; nreset = 1; slcs = 0; slrd = 1; slwr = 1; fd_out = 0;

            if (ready == 0)
            begin
                ns = s1;
            end
            else
            begin
                ns = s2;
            end
            end
        s2: begin
            // load descriptor length=6 and VID,PID
            fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
            usbffifoadr = 3'b100;
            pktend = 1; nreset = 1; slcs = 0; slrd = 1; slwr = 0; fd_out = 8'hb0;
            ns = s3;
        end
        s3: begin
            fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
            usbffifoadr = 3'b100;
            pktend = 1; nreset = 1; slcs = 0; slrd = 1; slwr = 1;
            fd_out = 8'hb0;
            if (ready == 0)
            begin
                ns = s3;
            end
            else
            begin
                ns = s4;
            end
            end
        s4: begin
            fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
            usbffifoadr = 3'b100;
            pktend = 1; nreset = 1; slcs = 0; slrd = 1; slwr = 0;
            fd_out = 8'h00;
            ns = s5;
        end
        s5: begin
            fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
            usbffifoadr = 3'b100;
            nreset = 1; slcs = 0; slrd = 1; slwr = 1;
            fd_out = 8'h00;
            if (ready == 0)
            begin
                ns = s5;
            end
            else
            begin
                ns = s6;
            end
        end
    endcase
end

```

```

        end
    end
s6: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbfifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h06;
    ns = s7;
    end
s7: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbfifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h06;
    if (ready == 0)
        begin
            ns = s7;
        end
    else
        begin
            ns = s8;
        end
    end
s8: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbfifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h00;
    ns = s9;
    end
s9: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbfifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h00;
    if (ready == 0)
        begin
            ns = s9;
        end
    else
        begin
            ns = s10;
        end
    end
s10: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbfifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h00;
    ns = s11;
    end
s11: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbfifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h00;
    if (ready == 0)
        begin
            ns = s11;
        end
    else
        begin
            ns = s12;
        end
    end
end

```

```

s12: begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h04;
    ns = s13;
end
s13: begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h04;
    if (ready == 0)
        begin
            ns = s13;
        end
    else
        begin
            ns = s14;
        end
    end
s14: begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h07;
    ns = s15;
end
s15: begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h07;
    if (ready == 0)
        begin
            ns = s15;
        end
    else
        begin
            ns = s16;
        end
    end
s16: begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h00;
    ns = s17;
end
s17: begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h00;
    if (ready == 0)
        begin
            ns = s17;
        end
    else
        begin
            ns = s18;
        end
    end
s18: begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr= 3'b100;

```

```

        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
        fd_out = 8'h05;
        ns = s19;
        end
s19: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffioadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h05;
    if (ready == 0)
        begin
            ns = s19;
        end
    else
        begin
            ns = s20;
        end
    end
s20: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffioadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h00;
    ns = s21;
    end
s21: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffioadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h00;
    if (ready == 0)
        begin
            ns = s21;
        end
    else
        begin
            ns = s22;
        end
    end
s22: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffioadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h02;
    ns = s23;
    end
s23: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffioadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h02;
    if (ready == 0)
        begin
            ns = s23;
        end
    else
        begin
            ns = s24;
        end
    end
s24: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffioadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;

```

```

        fd_out = 8'h01;
        ns = s25;
    end
s25: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h01;
    if (ready == 0)
        begin
            ns = s25;
        end
    else
        begin
            ns = s26;
        end
    end
s26: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
    fd_out = 8'h00;
    ns = s27;
    end
s27: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h00;
    if (ready == 0)
        begin
            ns = s27;
        end
    else
        begin
            ns = s28;
        end
    end
s28: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;
    nreset = 1;
    slcs = 0;
    slrd =1;
    slwr =0;
    fd_out = 8'h00;
    ns = s29;
    end
s29: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
    fd_out = 8'h00;
    if (ready == 0)
        begin
            ns = s29;
        end
    else
        begin
            ns = s30;
        end
    end
s30: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;

```

```

        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
        fd_out = 8'h01;
        ns = s31;
        end
s31: begin
        fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
        usbffifoadr= 3'b100;
        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
        fd_out = 8'h01;
        if (ready == 0)
            begin
                ns = s31;
            end
        else
            begin
                ns = s32;
            end
        end
s32: begin
        fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
        usbffifoadr= 3'b100;
        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =0;
        fd_out = 8'h00;
        ns = s33;
        end
s33: begin
        fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
        usbffifoadr= 3'b100;
        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=1;
        fd_out = 8'h00;
        if (ready == 0)
            begin
                ns = s33;
            end
        else
            begin
                ns = s34;
            end
        end
s34: begin
        fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
        usbffifoadr= 3'b100;
        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=0;
        fd_out = 8'h00;
        ns = s35;
        end
s35: begin
        fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
        usbffifoadr= 3'b100;
        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
        fd_out = 8'h00;
        if (ready == 0)
            begin
                ns = s35;
            end
        else
            begin
                ns = s36;
            end
        end
s36: begin //just for delay
        fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
        usbffifoadr= 3'b100;
        pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
        fd_out = 8'h00;
        ns = s37;

```

```

end
s37: begin //just for delay and jumps to state s100
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
    fd_out = 8'h00;
    ns = s38;
end

s38: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=0;
    fd_out = 8'h82;
    ns = s39;
end

s39: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
    fd_out = 8'h82;
    if (ready == 0)
        begin
            ns = s39;
        end
    else
        begin
            ns = s40;
        end
    end

s40: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=0;
    fd_out = 8'h00;
    ns = s41;
end

s41: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
    fd_out = 8'h00;
    if (ready == 0)
        begin
            ns = s41;
        end
    else
        begin
            ns = s42;
        end
    end

s42: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=0;
    fd_out = 8'h06;
    ns = s43;
end

s43: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
    fd_out = 8'h06;
    if (ready == 0)
        begin
            ns = s43;
        end
    end

```

```

        end
    else
        begin
            ns = s44;
        end
    end
s44: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=0;
    fd_out = 8'h96;
    ns = s45;
end
s45:begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
    fd_out = 8'h96;
    if (ready == 0)
        begin
            ns = s45;
        end
    else
        begin
            ns = s46;
        end
    end
s46:begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=0;
    fd_out = 8'h08;
    ns = s47;
end
s47:begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;
    fd_out = 8'h08;
    if (ready == 0)
        begin
            ns = s47;
        end
    else
        begin
            ns = s48;
        end
    end
s48: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr=0;
    fd_out = 8'h09;
    ns = s49;
end
s49: begin
    fifo_read = 0;fifo_rdclk = 0;clk_rdreg = 0;
    usbffifoadr= 3'b100;
    pktend =1;nreset = 1;slcs = 0;slrd =1;slwr =1;fd_out = 8'h09;
    if (ready == 0)
        begin
            ns = s49;
        end
    else
        begin
            ns = s390;
        end
    end
end

```

```

        end
    end
    // jump back from checking fifoempty flag
s390:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        pktend = 1; nreset = 1; slcs = 0; slrd = 1; slwr = 1;
        fd_out = 0;
        ns = s391;
    end
s391:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; slrd = 1; fd_out = 0; pktend = 1;
        ns = s392;
    end
s392:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; nreset = 1; slcs = 0; slrd = 1; pktend = 1;
        fd_out = 0;
        ns = s393;
    end
s393:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; nreset = 1; slcs = 0; slrd = 1; pktend = 1;
        fd_out = 0;
        ns = s394;
    end
s394:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; nreset = 1; slcs = 0; slrd = 1;
        fd_out = 0; pktend = 1;
        ns = s395;
    end
s395:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; nreset = 1; slcs = 0; slrd = 1;
        fd_out = 0; pktend = 1;
        ns = s396;
    end
s396:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; nreset = 1; slcs = 0; slrd = 1;
        fd_out = 0; pktend = 1;
        ns = s397;
    end
s397:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; nreset = 1; slcs = 0; slrd = 1;
        fd_out = 0; pktend = 1;
        ns = s398;
    end
s398:    begin
        fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
        usbffifoadr = 3'b010; //delay
        slwr = 1; nreset = 1; slcs = 0; slrd = 1;
        fd_out = 0; pktend = 1;
        ns = s399;
    end
s399:    begin                                //check fifoempty=1?

```

```

fifo_read = 0; fifo_rdclock = 0; clk_rdreg = 0;
usbffifoaddr = 3'b010;
slwr = 1; nreset = 1; slcs = 0; slrd = 1;
fd_out = 16'h00; pktend = 1;
if (flaga_fixed == 1)
begin
ns = s390; //if fifoempty=1 means Empty go back to read data and flags again
end
else
begin
ns = s400; //if fifoempty=0 means no_empty
end
end
s400: begin //start write one word-packet sequence
fifo_read = 0; // turn HIGH to activate read_fifo
fifo_rdclock = 1; clk_rdreg = 0;
usbffifoaddr = 3'b010; //address FIFO EP2 (End point 2)
slwr = 1; nreset = 1; slcs = 0; slrd = 1;
fd_out = 16'h00; pktend = 1;
ns = s401;
end
s401: begin
fifo_read = 0;
fifo_rdclock = 0; // clock signal to read fifo
clk_rdreg = 1;
usbffifoaddr = 3'b010;
slwr = 1; nreset = 1; slcs = 0; slrd = 1;
fd_out = 16'h00; pktend = 1;
ns = s402;
end
s402: begin //check fifoempty=1?
fifo_read = 0; fifo_rdclock = 0; clk_rdreg = 0;
usbffifoaddr = 3'b010;
slwr = 1; nreset = 1; slcs = 0; slrd = 1;
fd_out = 16'h00; pktend = 1;

if (fifoempty == 1)
begin
ns = s390; //if fifoempty=1 means Empty go back to read data and flags again
end
else
begin
ns = s403; //if fifoempty=0 means no_empty
end
end
s403: begin //start write one word-packet sequence
fifo_read = 1; // turn HIGH to activate read_fifo
fifo_rdclock = 0; clk_rdreg = 0;
usbffifoaddr = 3'b010; //address FIFO EP2 (End point 2)
slwr = 1; nreset = 1; slcs = 0; slrd = 1;
fd_out = 16'h00; pktend = 1;
ns = s404;
end
s404: begin
fifo_read = 1; fifo_rdclock = 1; clk_rdreg = 0;
usbffifoaddr = 3'b010;
slwr = 1; nreset = 1; slcs = 0; slrd = 1;
fd_out = 16'h00; pktend = 1;
ns = s405;
end
s405: begin
fifo_read = 0; fifo_rdclock = 0; clk_rdreg = 1;
usbffifoaddr = 3'b010;
slwr = 1; nreset = 1; slcs = 0; slrd = 1;
fd_out = 16'h00; pktend = 1;
ns = s406;

```

```

end
s406:    begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010;
    slwr = 1; nreset = 1; slcs = 0; slrd = 1;
    fd_out = 16'h00;    pktend = 1;
    ns = s407;
end
s407:    begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010;
    slwr = 0; //slwr=LOW
    nreset = 1; slcs = 0; slrd = 1;
    fd_out = datain[15:0]; //write count value lower 16bits from register
    pktend = 1;
    ns = s408;
end
s408:    begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010;
    slwr = 1; //slwr=HIGH generates rising edge to write data
    nreset = 1; slcs = 0; slrd = 1;
    fd_out = datain[15:0]; //write count value lower 16bits from register
    pktend = 1;
    ns = s409;
end
s409:    begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010;
    slwr = 1; nreset = 1; slcs = 0; slrd = 1;
    fd_out = 0; pktend = 1;
    ns = s410;
end
s410:    begin //start write one word-packet sequence
    fifo_read = 1; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010; //address FIFO EP2 (End point 2)
    slwr = 1; nreset = 1; slcs = 0; slrd = 1;
    fd_out = 16'h00;    pktend = 1;
    ns = s411;
end
s411:    begin
    fifo_read = 1; fifo_rdclk = 1; clk_rdreg = 0;
    usbffifoadr = 3'b010;
    slwr = 1; nreset = 1; slcs = 0; slrd = 1;
    fd_out = 16'h00; pktend = 1;
    ns = s412;
end
s412:    begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010;
    slwr = 0; nreset = 1; slcs = 0; slrd = 1;
    fd_out = datain[31:16]; //write count value upper 16bits from register
    pktend = 1;
    ns = s413;
end
s413:    begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010;
    slwr = 1; nreset = 1; slcs = 0; slrd = 1;
    fd_out = datain[31:16]; //write count value upper 16bits from register
    pktend = 1;
    ns = s414;
end
s414:    begin
    fifo_read = 0; fifo_rdclk = 0; clk_rdreg = 0;
    usbffifoadr = 3'b010;

```

```

        slwr =1;   nreset = 1;slcs = 0;slrd =1;
        fd_out = 0;pktend =1;
        ns = s399;
        end
        default: ns = s0;
    endcase
end
endmodule

```

Appendix D. MATLAB Code for Position Estimations Using MLE

This is a MATLAB program that uses Maximum Likelihood Estimation (MLE) to find the most probable x , y , z positions of a trapped particle based on the time-gated photon counts, which are binned for $\sim 13.5 \mu\text{s}$ and transferred from the FPGA to the PC via USB 2.0. The binned photons, N_1-N_4 , are summed over consecutive 1 ms time periods before being used to estimate the positions. These estimations are conducted while taking into account the experimentally measured time-gating cross-talk ($p_1=0.777$; $p_2=0.173$; $p_3=0.039$; $p_4=0.011$) as well as the background photon counts. Note that the MLE positions undergo a filter to avoid undue jumps that would be unlikely by diffusion according to the diffusional root-mean square distance per axis.

The percentage symbol “%” is used to denote single line comments in MATLAB. Most times, the code is commented to clarify the purpose. The double percentage symbol with texts in between (i.e., %--xx--%) is used to emphasize the main sections of the program flow.

```

%----- Parameters-----%
n = 1.33;           %refractive index of water
wl = 0.64/n;        %wavelength in um (in water)
w0 = 0.53;          %Beam waist in water in um
zr = ((pi*w0^2)/wl); %Rayleigh range in water

x1=0.38;            %x offset in um

```

```

y1=0.38; %y offset in um
z1 = 1.6; %Z offset (z=0 to z=z1) in water
DC=12.1; %Diffusion coefficient:
          %12.1 um^2/s for 40 nm FluoSpheres
          %24.1 um^2/s for 20 nm FluoSpheres

gapx=0.020; %grid spacing in X
gapy=0.020; %grid spacing in Y
gapz=0.020; %grid spacing in Z

x = -0.50:gapx:0.50; %x range in um
y = -0.50:gapy:0.50; %y range in um
z = -5.00:gapz:5.00; %z range in um

l_x = length(x); %length of x range
l_y = length(y); %length of y range
l_z = length(z); %length of z range
R5 = zeros(l_y,l_x,l_z); %R5,R6,R7,R8 3D matrices record intensities
R6 = zeros(l_y,l_x,l_z);
R7 = zeros(l_y,l_x,l_z);
R8 = zeros(l_y,l_x,l_z);
P = zeros(l_y,l_x,l_z);

p1=0.777; p2=0.173; p3=0.039; p4=0.011; % Binning probabilities
TS=13.5e-6; %Time step in seconds
Counts_at_center=5700000; %Background-corrected counts at center/second-
                          %5700000 for 40 nm FluoSpheres
                          %8300000 for 20 nm FluoSpheres

Jumpavg_num=74; %Photons collected for 1 ms
BinTime=13.5e-6*Jumpavg_num;
%CO: Maximum photon count per beam per binning time at the presence of a particle
CO=Counts_at_center*BinTime/ (4*0.18);

Background=100000*BinTime/4; %Background count per bin per binning time

%-----End of Parameters-----%

%-----Generation of 3D intensity matrices-----%
for Z=1:1:l_z
    K1 = 1/((w0^2)*(1+((z(Z)-z1)/zr)^2));
    K2 = 1/((w0^2)*(1+((z(Z)+z1)/zr)^2));
    for Y=1:1:l_y
        for X=1:1:l_x
            % Generate expected relative counts matrices R5, R6, R7, R8
            R5(Y,X,Z)=K1*exp((-2*K1*((x(X)+ x1)^2 +(y(Y) -y1)^2)))*w0^2;%Focus1
            R6(Y,X,Z)=K1*exp((-2*K1*((x(X) - x1)^2 +(y(Y)+y1)^2)))*w0^2;%Focus2
            R7(Y,X,Z)=K2*exp((-2*K2*((y(Y)+ y1)^2 +(x(X)+x1)^2)))*w0^2;%Focus3
            R8(Y,X,Z)=K2*exp((-2*K2*((y(Y)- y1)^2 + (x(X) -x1)^2)))*w0^2;%Focus4

            % Generate expected counts matrices R1, R2, R3, R4
            R1(Y,X,Z)=((p1*R5(Y,X,Z))+(p2*R6(Y,X,Z))+(p3*R7(Y,X,Z))+(p4*R8(Y,X,Z)))*C0+Background;
            R2(Y,X,Z)=((p4*R5(Y,X,Z))+(p1*R6(Y,X,Z))+(p2*R7(Y,X,Z))+(p3*R8(Y,X,Z)))*C0+Background;
            R3(Y,X,Z)=((p3*R5(Y,X,Z))+(p4*R6(Y,X,Z))+(p1*R7(Y,X,Z))+(p2*R8(Y,X,Z)))*C0+Background;
            R4(Y,X,Z)=((p2*R5(Y,X,Z))+(p3*R6(Y,X,Z))+(p4*R7(Y,X,Z))+(p1*R8(Y,X,Z)))*C0+Background;
        end
    end
end
%-----End of generation of 3D intensity matrices-----%

%-----Start Maximum likelihood estimation-----%
%-----%
for K=1 % Do MLE for data sets saved in Excel sheets
    sheet=K;

    % I1: photon counts when trapped

```

```

l1=xlsread('MLE-Data_40nm_sheet1 - poisson.xlsx',sheet,'A1:D1000000');
%-----%
i=1;
sz = size(l1);
l = zeros(sz);
for h=1:1:length(l1(:,1))
    i=1+floor((h-1)/Jumpavg_num);
    l(i,:)=l(i,:)+l1(h,:);
end
%-----%
N1=l(1:i,1);          %Photon counts from beam A1
N2=l(1:i,2);          %Photon counts from beam A2
N3=l(1:i,3);          %Photon counts from beam B1
N4=l(1:i,4);          %Photon counts from beam B2
N=N1+N2+N3+N4;
time = length(N1);     % time = Total time/13.5 us
D1=zeros(length(y),length(x));
max=-1e150;
maxindex=[1 1 1];
old_maxindex=[1 1 1];
mm=0;
cuml_prob=0;

for h=1:1:time        %Start calculation of MLE for all the time periods

    %-----Calculation of MLE per time period-----%
    for i=1:1:(length(z))
        D1=l(h,1)*log(R1(:,i)) + l(h,2)*log(R2(:,i)) + l(h,3)*log(R3(:,i)) + l(h,4)*log(R4(:,i))-R(:,i);
        A=(D1(:));
        for j=1:1:length(A)
            if (A(1,j)>max)
                max=A(1,j);
                [l_row, l_col] = ind2sub(size(D1),j);
                maxindex=[l_row,l_col,i];
            end
        end
    end
    %-----End of calculation of MLE per time period-----%

    if (h==1)
        old_maxindex(1,:) = maxindex(1,:);
    end
    %-----%

    gap_maxindex(1,:)=maxindex(1,:)-old_maxindex(1,:);
    %g: approximated root mean square diffusional distance per axis (unit less)
    g=round(sqrt(2*DC*Jumpavg_num*TS)/gapx);

    %-----Kalman-like filter-----%
    % The filter allows a compatible estimation of the positions with the actual particle diffusion

    if (gap_maxindex(1,1)>g)                %along Y axis
        maxindex(1,1)=old_maxindex(1,1)+g;
    elseif (gap_maxindex(1,1)<-g)
        maxindex(1,1)=old_maxindex(1,1)-g;
    end

    if (gap_maxindex(1,2)>g)                %along X axis
        maxindex(1,2)=old_maxindex(1,2)+g;
    elseif (gap_maxindex(1,2)<-g)
        maxindex(1,2)=old_maxindex(1,2)-g;
    end

    if (gap_maxindex(1,3)>g)                %along Z-axis

```

```

        maxindex(1,3)=old_maxindex(1,3)+g;

elseif (gap_maxindex(1,3)<-g)
    maxindex(1,3)=old_maxindex(1,3)-g;
end
%-----End of Kalman-like filter-----%

old_maxindex(1,:)=maxindex(1,:);

%-----%
s2(h)=((x(maxindex(1,2)))));    %position x
s1(h)=((y(maxindex(1,1)))));    %position y
s3(h)=((z(maxindex(1,3)))));    %position z
%-----%

max=-1e150;
end
%-----End of Maximum likelihood estimations%-----%
end

```

Appendix E. MATLAB Code for 3D Analysis of Flow Directions

This MATLAB program plots the fluid flow directions within the region of interest (ROI) around the midpoint of the four laser foci. The program uses the relative intensities of the laser beams for the analysis, with the approximation that the photon counts N_1 – N_4 are proportional to the intensities I_1 – I_4 , and it applies the trapping algorithm shown in Table 5.1 to determine the fluid flow direction at positions on a 3D grid. Graphs were generated for selected X-Y, X-Z, and Y-Z planes and the graphs shows the expected direction of fluid flow with a blue arrow for a grid point marked in red.

```

%-----Parameters-----%
n = 1.33;           %refractive index of water
wl = 0.64/n;        %wavelength in um (in water)
w0 = 0.53;          %Beam waist in water in um
zr=((pi*w0^2)/wl);  %Rayleigh range in water

x1=0.38;            %x offset in um
y1=0.38;            %y offset in um
z1=1.6;             %Z offset (z=0 to z=z1) in water

gapx=0.03;
gapy=0.03;
gapz=0.3;
x = -0.18:gapx:0.27; %x range in um

```

```

y = -0.18:gapy:0.27;          %y range in um
z = -1.8:gapz:2.7;           %z range in um

l_x = length(x);              %length of x range
l_y = length(y);              %length of y range
l_z = length(z);              %length of z range

R1 = zeros(l_y,l_x,l_z);
R2 = zeros(l_y,l_x,l_z);
R3 = zeros(l_y,l_x,l_z);
R4 = zeros(l_y,l_x,l_z);

%-----End of Parameters-----%

%-----Generation of matrices-----%
for Z=1:1:l_z
    K1 = 1/((w0^2)*(1+((z(Z)-z1)/zr)^2));
    K2 = 1/((w0^2)*(1+((z(Z)+z1)/zr)^2));
    for Y=1:1:l_y
        for X=1:1:l_x
            % Generate expected relative counts matrices R1,R2,R3,R4
            R1(Y,X,Z)=(K1*exp((-2*K1*((x(X)- x1)^2 + (y(Y)-y1)^2)))*w0^2);%A1
            R2(Y,X,Z)=(K1*exp((-2*K1*((x(X)- x1)^2 + (y(Y)+y1)^2)))*w0^2);%A2
            R3(Y,X,Z)=(K2*exp((-2*K2*((y(Y)+ y1)^2 + (x(X)+x1)^2)))*w0^2);%B1
            R4(Y,X,Z)=(K2*exp((-2*K2*((y(Y)- y1)^2 + (x(X)-x1)^2)))*w0^2);%B2
            R(Y,X,Z)=R1(Y,X,Z)+R2(Y,X,Z)+R3(Y,X,Z)+R4(Y,X,Z);

            CA12(Y,X,Z) = R1(Y,X,Z)+R2(Y,X,Z);
            CA13(Y,X,Z) = R1(Y,X,Z)+R3(Y,X,Z);
            CA14(Y,X,Z) = R1(Y,X,Z)+R4(Y,X,Z);
            CA23(Y,X,Z) = R2(Y,X,Z)+R3(Y,X,Z);
            CA24(Y,X,Z) = R2(Y,X,Z)+R4(Y,X,Z);
            CA34(Y,X,Z) = R3(Y,X,Z)+R4(Y,X,Z);

            T13_24(Y,X,Z)=CA13(Y,X,Z) - CA24(Y,X,Z);
            T23_14(Y,X,Z)=CA23(Y,X,Z) - CA14(Y,X,Z);
            T34_12(Y,X,Z)=CA34(Y,X,Z) - CA12(Y,X,Z);

        end
    end
end
%-----End of matrices-----%

%-----%
i=0;
for Z=1:1:l_z
    for Y=1:1:l_y
        for X=1:1:l_x
            i=i+1;
            sx(i)=x(X); sy(i)=y(Y); sz(i)=z(Z);

            if (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) - CA14(Y,X,Z)>0)),(CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
                %01%X-Y-Z-%Vx+,Vy+,Vz+
                Vx(i)=1; Vy(i)=1; Vz(i)=1;

            elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) - CA14(Y,X,Z)>0)),(CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
                %02%X-Y-Z+%Vx+,Vy+,Vz-
                Vx(i)=1; Vy(i)=1; Vz(i)=-1;

            elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) - CA14(Y,X,Z)>0)),(CA34(Y,X,Z) == CA12(Y,X,Z))))
                %03%X-Y-Z0%Vx+,Vy+,Vz0
                Vx(i)=1; Vy(i)=1; Vz(i)=0;

            elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)),(CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
                %04%X-Y+Z-%Vx+,Vy-,Vz+

```

```

Vx(i)=1; Vy(i)=-1; Vz(i)=1;
elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)),(CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
%05%X-Y+Z+%Vx+,Vy-,Vz-
Vx(i)=1; Vy(i)=-1; Vz(i)=-1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)),(CA34(Y,X,Z) == CA12(Y,X,Z))))
%06%X-Y+Z0%Vx+,Vy-,Vz0
Vx(i)=1; Vy(i)=-1; Vz(i)=0;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
%07%X-Y0Z-%Vx+,Vy0,Vz+
Vx(i)=1; Vy(i)=0; Vz(i)=1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
%08%X-Y0Z+%Vx+,Vy0,Vz-
Vx(i)=1; Vy(i)=0; Vz(i)=-1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)>0),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) == CA12(Y,X,Z))))
%09%X-Y0Z0%Vx+,Vy0,Vz0
Vx(i)=1; Vy(i)=0; Vz(i)=0;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) - CA14(Y,X,Z)>0)), (CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
%10%X+Y-Z-%Vx-,Vy+,Vz+
Vx(i)=-1; Vy(i)=1; Vz(i)=1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) - CA14(Y,X,Z)>0)), (CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
%11%X+Y-Z+%Vx-,Vy+,Vz-
Vx(i)=-1; Vy(i)=1; Vz(i)=-1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) - CA14(Y,X,Z)>0)), (CA34(Y,X,Z) == CA12(Y,X,Z))))
%12%X+Y-Z0%Vx-,Vy+,Vz0
Vx(i)=-1; Vy(i)=1; Vz(i)=0;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)), (CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
%13%X+Y+Z-%Vx-,Vy-,Vz+
Vx(i)=-1; Vy(i)=-1; Vz(i)=1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)), (CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
%14%X+Y+Z+%Vx-,Vy-,Vz-
Vx(i)=-1; Vy(i)=-1; Vz(i)=-1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)), (CA34(Y,X,Z) == CA12(Y,X,Z))))
%15%X+Y+Z0%Vx-,Vy-,Vz0
Vx(i)=-1; Vy(i)=-1; Vz(i)=0;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
%16%X+Y0Z-%Vx-,Vy0,Vz+
Vx(i)=-1; Vy(i)=0; Vz(i)=1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
%17%X+Y0Z+%Vx-,Vy0,Vz-
Vx(i)=-1; Vy(i)=0; Vz(i)=-1;

elseif (and(and((CA13(Y,X,Z) - CA24(Y,X,Z)<0),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) == CA12(Y,X,Z))))
%18%X+Y0Z0%Vx-,Vy0,Vz0
Vx(i)=-1; Vy(i)=0; Vz(i)=0;

elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)), (CA23(Y,X,Z) - CA14(Y,X,Z)>0)), (CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
%19%X0Y-Z-%Vx0,Vy+,Vz+
Vx(i)=0; Vy(i)=1; Vz(i)=1;

elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)), (CA23(Y,X,Z) - CA14(Y,X,Z)>0)), (CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
%20%X0Y-Z+%Vx0,Vy+,Vz-
Vx(i)=0; Vy(i)=1; Vz(i)=-1;

elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)), (CA23(Y,X,Z) - CA14(Y,X,Z)>0)), (CA34(Y,X,Z) == CA12(Y,X,Z))))

```

```

    %21%X0Y-Z0%Vx0,Vy+,Vz0
    Vx(i)=0; Vy(i)=1; Vz(i)=0;

    elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)),(CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
        %22%X0Y+Z-%Vx0,Vy-,Vz+
        Vx(i)=0; Vy(i)=-1; Vz(i)=1;

    elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)),(CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
        %23%X0Y+Z+%Vx0,Vy-,Vz-
        Vx(i)=0; Vy(i)=-1; Vz(i)=-1;

    elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)),(CA23(Y,X,Z) - CA14(Y,X,Z)<0)),(CA34(Y,X,Z) == CA12(Y,X,Z))))
        %24%X0Y+Z0%Vx0,Vy-,Vz0
        Vx(i)=0; Vy(i)=-1; Vz(i)=0;

    elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) - CA12(Y,X,Z)>0)))
        %25%X0Y0Z-%Vx0,Vy0,Vz+
        Vx(i)=0; Vy(i)=0; Vz(i)=1;

    elseif (and(and((CA13(Y,X,Z) == CA24(Y,X,Z)),(CA23(Y,X,Z) == CA14(Y,X,Z))), (CA34(Y,X,Z) - CA12(Y,X,Z)<0)))
        %26%X0Y0Z+%Vx0,Vy0,Vz-
        Vx(i)=0; Vy(i)=0; Vz(i)=-1;

    else
        %27%X0Y0Z0%Vx0,Vy0,Vz0
        Vx(i)=0; Vy(i)=0; Vz(i)=0;
    end

    %-----%
    %-----%

end
end
end

%-----XY Planes-----%
m=256;
range=[1:m];

figure(1)
subplot(2,3,1)
quiver3(sx(range)*10,sy(range)*10,sz(range),Vx(range),Vy(range),Vz(range),0.6)
xlabel('x (lmum) \times 0.1 (at z=-1.8 um)')
ylabel('y (lmum) \times 0.1 (at z=-1.8 um)')
zlabel('z (lmum)')
view(0,90)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range)*10,sy(range)*10,sz(range),'r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,2)
quiver3(sx(range+m*3)*10,sy(range+m*3)*10,sz(range+m*3),Vx(range+m*3),Vy(range+m*3),Vz(range+m*3),0.6)%x,y are
in 1e-10 um
xlabel('x (lmum) \times 0.1 (at z=-0.9 um)')
ylabel('y (lmum) \times 0.1 (at z=-0.9 um)')
zlabel('z (lmum)')

```

```

view(0,90)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+m*3)*10,sy(range+m*3)*10,sz(range+m*3),'.r','MarkerSize',6)
%title('Flow Directions')
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,3)
quiver3(sx(range+m*6)*10,sy(range+m*6)*10,sz(range+m*6),Vx(range+m*6),Vy(range+m*6),Vz(range+m*6),0.6)
xlabel('x (\mum) \times 0.1 (at z=0 um)')
ylabel('y (\mum) \times 0.1 (at z=0 um)')
zlabel('z (\mum)')
view(0,90)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+m*6)*10,sy(range+m*6)*10,sz(range+m*6),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,4)
quiver3(sx(range+m*8)*10,sy(range+m*8)*10,sz(range+m*8),Vx(range+m*8),Vy(range+m*8),Vz(range+m*8),0.6)
xlabel('x (\mum) \times 0.1 (at z=0.9 um)')
ylabel('y (\mum) \times 0.1 (at z=0.9 um)')
zlabel('z (\mum)')
view(0,90)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+m*8)*10,sy(range+m*8)*10,sz(range+m*8),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,5)
quiver3(sx(range+m*12)*10,sy(range+m*12)*10,sz(range+m*12),Vx(range+m*12),Vy(range+m*12),Vz(range+m*12),0.6)
xlabel('x (\mum) \times 0.1 (at z=1.8 um)')
ylabel('y (\mum) \times 0.1 (at z=1.8 um)')
zlabel('z (\mum)')
view(0,90)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+m*12)*10,sy(range+m*12)*10,sz(range+m*12),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')

```

```

hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,6)
quiver3(sx(range+m*15)*10,sy(range+m*15)*10,sz(range+m*15),Vx(range+m*15),Vy(range+m*15),Vz(range+m*15),0.6)
xlabel('x (\text{um}) \times 0.1 (at z=2.7 \text{um})')
ylabel('y (\text{um}) \times 0.1 (at z=2.7 \text{um})')
zlabel('z (\text{um})')
view(0,90)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+m*15)*10,sy(range+m*15)*10,sz(range+m*15),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

%-----End of XY Planes-----%

%-----XZ Planes-----%
j=0;
n=16;
range=[1];
for i=1:m
    j=j+1;
    if (mod(j,n)==0)
        j=0;
        k=n;
    else
        k=j;
    end
    range(i)=k+m*floor((i-1)/n);
end

figure(2)
subplot(2,3,1)
quiver3(sx(range)*10,sy(range)*10,sz(range),Vx(range),Vy(range),Vz(range),0.6)
xlabel('x (\text{um}) \times 0.1 (at y=-0.18 \text{um})')
zlabel('z (\text{um}) (at y=-0.18 \text{um})')
view(0,0)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range)*10,sy(range)*10,sz(range),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,2)
quiver3(sx(range+n*4)*10,sy(range+n*4)*10,sz(range+n*4),Vx(range+n*4),Vy(range+n*4),Vz(range+n*4),0.6)
xlabel('x (\text{um}) \times 0.1 (at y=-0.06 \text{um})')

zlabel('z (\text{um}) (at y=-0.06 \text{um})')

```

```

view(0,0)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*4)*10,sy(range+n*4)*10,sz(range+n*4),'.r','MarkerSize',6)
%title('Flow Directions')
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,3)
quiver3(sx(range+n*6)*10,sy(range+n*6)*10,sz(range+n*6),Vx(range+n*6),Vy(range+n*6),Vz(range+n*6),0.6)
xlabel('x (\text{um}) \times 0.1 (at y=0 \text{um})')
ylabel('z (\text{um}) (at y=0 \text{um})')
view(0,0)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*6)*10,sy(range+n*6)*10,sz(range+n*6),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,4)
quiver3(sx(range+n*8)*10,sy(range+n*8)*10,sz(range+n*8),Vx(range+n*8),Vy(range+n*8),Vz(range+n*8),0.6)
xlabel('x (\text{um}) \times 0.1 (at y=0.06 \text{um})')
ylabel('z (\text{um}) (at y=0.06 \text{um})')
view(0,0)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*8)*10,sy(range+n*8)*10,sz(range+n*8),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,5)
quiver3(sx(range+n*12)*10,sy(range+n*12)*10,sz(range+n*12),Vx(range+n*12),Vy(range+n*12),Vz(range+n*12),0.6)
xlabel('x (\text{um}) \times 0.1 (at y=0.18 \text{um})')
ylabel('z (\text{um}) (at y=0.18 \text{um})')
view(0,0)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*12)*10,sy(range+n*12)*10,sz(range+n*12),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

```

```

plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,6)
quiver3(sx(range+n*15)*10,sy(range+n*15)*10,sz(range+n*15),Vx(range+n*15),Vy(range+n*15),Vz(range+n*15),0.6)
xlabel('x (\text{um}) \times 0.1 (at y=0.27 \text{um})')
ylabel('y (\text{um}) (at y=0.27 \text{um})')
view(0,0)
xlim([-2 3])
ylim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*15)*10,sy(range+n*15)*10,sz(range+n*15),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

%-----End of XZ Planes-----%

%-----YZ Planes-----%

range=[1];
for i=2:1:m
    range(i)=range(i-1)+n;
end
n=1;

figure(3)
subplot(2,3,1)
quiver3(sx(range)*10,sy(range)*10,sz(range),Vx(range),Vy(range),Vz(range),0.6)
xlabel('x (\text{um}) \times 0.1 (at x=-0.18 \text{um})')
ylabel('y (\text{um}) (at x=-0.18 \text{um})')
view(90,0)
ylim([-2 3])
zlim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range)*10,sy(range)*10,sz(range),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,2)
quiver3(sx(range+n*4)*10,sy(range+n*4)*10,sz(range+n*4),Vx(range+n*4),Vy(range+n*4),Vz(range+n*4),0.6)
xlabel('x (\text{um}) \times 0.1 (at x=-0.06 \text{um})')
ylabel('y (\text{um}) (at x=-0.06 \text{um})')
view(90,0)
ylim([-2 3])
zlim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*4)*10,sy(range+n*4)*10,sz(range+n*4),'.r','MarkerSize',6)
%title('Flow Directions')
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')

```

```

hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,3)
quiver3(sx(range+n*6)*10,sy(range+n*6)*10,sz(range+n*6),Vx(range+n*6),Vy(range+n*6),Vz(range+n*6),0.6)
ylabel('y (\mu m) \times 0.1 (at x=0 \mu m)')
zlabel('z (\mu m) (at x=0 \mu m)')
view(90,0)
ylim([-2 3])
zlim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*6)*10,sy(range+n*6)*10,sz(range+n*6),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,4)
quiver3(sx(range+n*8)*10,sy(range+n*8)*10,sz(range+n*8),Vx(range+n*8),Vy(range+n*8),Vz(range+n*8),0.6)
ylabel('y (\mu m) \times 0.1 (at x=0.06 \mu m)')
zlabel('z (\mu m) (at x=0.06 \mu m)')
view(90,0)
ylim([-2 3])
zlim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*8)*10,sy(range+n*8)*10,sz(range+n*8),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,5)
quiver3(sx(range+n*12)*10,sy(range+n*12)*10,sz(range+n*12),Vx(range+n*12),Vy(range+n*12),Vz(range+n*12),0.6)
ylabel('y (\mu m) \times 0.1 (at x=0.18 \mu m)')
zlabel('z (\mu m) (at x=0.18 \mu m)')
view(90,0)
ylim([-2 3])
zlim([-2 3])
pbaspect([1 1 1])
hold on
plot3(sx(range+n*12)*10,sy(range+n*12)*10,sz(range+n*12),'.r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

subplot(2,3,6)
quiver3(sx(range+n*15)*10,sy(range+n*15)*10,sz(range+n*15),Vx(range+n*15),Vy(range+n*15),Vz(range+n*15),0.6)
ylabel('y (\mu m) \times 0.1 (at x=0.27 \mu m)')
zlabel('z (\mu m) (at x=0.27 \mu m)')
%set(gca,'fontname','arial','fontweight','normal','fontsize',12)
view(90,0)
ylim([-2 3])
zlim([-2 3])

```

```

pbaspect([1 1 1])
hold on
plot3(sx(range+n*15)*10,sy(range+n*15)*10,sz(range+n*15),'r','MarkerSize',6)
hold on
plot3([-3.3 4.4],[0 0],[0 0],'-k')
hold on
plot3([0 0],[-3.3 4.4],[0 0],'-k')
hold on
plot3([0 0],[0 0],[-4.0 6.0],'-k')
hold off

%-----End of YZ Planes-----%

```

Appendix F. MATLAB Code to Test Trapping Modes

This MATLAB program extends the results of the above 3D flow direction analysis to plot the track or position on the 3D Cartesian grid that a hypothetical fluorescent particle with zero diffusivity would follow if it began at a particular initial position and followed the flow directions produced by the trap. The particle is driven by the fluid flow when the trapping algorithm in Table 5.1 is applied for the laser intensity fields of the system. The program assumes the proportionality of photon counts N_1 – N_4 with the intensities I_1 – I_4 . It is expected that the particle will end at the lowest point of a potential well in the effective potential of the trap, and hence different ending points for different starting points would indicate that the effective potential has more than one well.

```

%-----Parameters-----%

n = 1.33;           %refractive index of water
wl = 0.64/n;        %wavelength in um (in water)
w0 = 0.53;          %Beam waist in water in um
zr = ((pi*w0^2)/wl); %Rayleigh range in water

x1=0.38;            %x offset in um
y1=0.38;            %y offset in um
z1 = 1.6;           %Z offset (z=0 to z=z1) in water

gapx=0.01;
gapy=0.01;
gapz=0.01;

```

```

x = -0.50:gapx:0.50;           %x range in um
y = -0.50:gapy:0.50;           %y range in um
z = -2.50:gapz:2.50;           %z range in um

l_x = length(x);               %length of x range
l_y = length(y);               %length of y range
l_z = length(z);               %length of z range

R1 = zeros(l_y,l_x,l_z);
R2 = zeros(l_y,l_x,l_z);
R3 = zeros(l_y,l_x,l_z);
R4 = zeros(l_y,l_x,l_z);
%-----End of Parameters-----%

%-----Generation of matrices-----%
tic
for Z=1:l_z
    K1 = 1/((w0^2)*(1+((z(Z)-z1)/zr)^2));
    K2 = 1/((w0^2)*(1+((z(Z)+z1)/zr)^2));
    for Y=1:l_y
        for X=1:l_x
            % Generate expected relative intensity matrices R5,R6,R7,R8
            R1(Y,X,Z)=(1000000*K1*exp((-2*K1*((x(X)+ x1)^2 + (y(Y)-y1)^2)))*w0^2);%A1
            R2(Y,X,Z)=(1000000*K1*exp((-2*K1*((x(X)- x1)^2 + (y(Y)+y1)^2)))*w0^2);%A2
            R3(Y,X,Z)=(1000000*K2*exp((-2*K2*((y(Y)+ y1)^2 + (x(X)+x1)^2)))*w0^2);%B1
            R4(Y,X,Z)=(1000000*K2*exp((-2*K2*((y(Y)- y1)^2 + (x(X)-x1)^2)))*w0^2);%B2

            CA12(Y,X,Z) = R1(Y,X,Z)+R2(Y,X,Z);
            CA13(Y,X,Z) = R1(Y,X,Z)+R3(Y,X,Z);
            CA14(Y,X,Z) = R1(Y,X,Z)+R4(Y,X,Z);
            CA23(Y,X,Z) = R2(Y,X,Z)+R3(Y,X,Z);
            CA24(Y,X,Z) = R2(Y,X,Z)+R4(Y,X,Z);
            CA34(Y,X,Z) = R3(Y,X,Z)+R4(Y,X,Z);

            T13_24(Y,X,Z)=CA13(Y,X,Z) - CA24(Y,X,Z);
            T23_14(Y,X,Z)=CA23(Y,X,Z) - CA14(Y,X,Z);
            T34_12(Y,X,Z)=CA34(Y,X,Z) - CA12(Y,X,Z);

        end
    end
end

%-----End of matrices-----%
l=1;
iter=7;
for j=1:iter
    switch(j)
        case 1
            Xcurrent=10;Ycurrent=10;Zcurrent=30;
        case 2
            Xcurrent=80;Ycurrent=10;Zcurrent=30;
        case 3
            Xcurrent=80;Ycurrent=80;Zcurrent=30;
        case 4
            Xcurrent=10;Ycurrent=80;Zcurrent=30;
        case 5
            Xcurrent=30;Ycurrent=40;Zcurrent=200;
        case 6
            Xcurrent=10;Ycurrent=10;Zcurrent=300;
        case 7
            Xcurrent=80;Ycurrent=10;Zcurrent=300;
        otherwise
            Xcurrent=51;Ycurrent=51;Zcurrent=251;
    end
end

```

```

for i=1:1:300
    XX(i)=(Xcurrent*gapx)-0.51;
    YY(i)=(Ycurrent*gapy)-0.51;
    ZZ(i)=(Zcurrent*gapz)-2.51;

    if T13_24(Ycurrent,Xcurrent,Zcurrent)>0
        Xnext=Xcurrent+1;
    elseif T13_24(Ycurrent,Xcurrent,Zcurrent)<0
        Xnext=Xcurrent-1;
    else
        Xnext=Xcurrent;
    end
    if T23_14(Ycurrent,Xcurrent,Zcurrent)>0
        Ynext=Ycurrent+1;
    elseif T23_14(Ycurrent,Xcurrent,Zcurrent)<0
        Ynext=Ycurrent-1;
    else
        Ynext=Ycurrent;
    end

    if T34_12(Ycurrent,Xcurrent,Zcurrent)>0
        Znext=Zcurrent+1;
    elseif T34_12(Ycurrent,Xcurrent,Zcurrent)<0
        Znext=Zcurrent-1;
    else
        Znext=Zcurrent;
    end

    Xcurrent=Xnext;
    Ycurrent=Ynext;
    Zcurrent=Znext;
end

figure(1)
subplot(iter,3,l)
plot(ZZ,'r.','MarkerSize',4)
if l==10
    ylabel('z (\mum)')
end
l=l+1;
%ylim([-2.5 2.5])
grid on
hold on
subplot(iter,3,l)
plot(XX,'r.','MarkerSize',4)
if l==11
    ylabel('x (\mum)')
end
if and((iter==7),(l==20))
    xlabel('Iteration number')
end
l=l+1;
%ylim([-0.5 0.5])
grid on
hold on
subplot(iter,3,l)
plot(YY,'r.','MarkerSize',4)
if l==12
    ylabel('y (\mum)')
end
l=l+1;
%ylim([-0.5 0.5])
grid on
hold off
end
hold off

```

VITA

Kapila Deshappriya Dissanayaka Dissanayaka Mudiyanse is from Colombo, Sri Lanka and received his Bachelor of Science degree in September 2009 from the University of Colombo with a Major in Engineering Physics and Minors in Mathematics and Computer Science. He entered Middle Tennessee State University in Murfreesboro, TN in the Fall Semester of 2012 to read for his Master of Science degree in Engineering Technology and completed it in May 2014. In the Fall Semester of 2014, Kapila joined the department of Physics and Astronomy at the University of Tennessee-Knoxville (UT-Knoxville) to read for his PhD in physics. After completing the core curriculum and qualifying exam at UT-Knoxville, he moved to the UT Space Institute in Tullahoma, in January 2016 to carry on research and advanced coursework in Optical Physics towards his PhD.