A Digital Frequency Source for Movement of Ultracold Atoms by Acousto-Optic Deflection

Thomas McKellar

A thesis submitted for the degree of Master of Science at the University of Otago, Dunedin, New Zealand.

October 3, 2014
Supervisor: Dr. Niels Kjærgaard

Jack Dodd Centre for Quantum Technology, Department of Physics
Abstract

This thesis documents the development of a frequency source based on a field programmable gate array (FPGA) and a direct digital synthesiser (DDS). The source was used to drive the two inputs of a dual-axis acousto-optic deflector (AOD), an integral component in a steerable optical tweezer unit designed for use with ultracold $^{87}$Rb and $^{40}$K. By synchronously changing the frequencies of the paired inputs to the AOD, an optical beam was able to be moved in two dimensions. The frequency source was used to trap and smoothly move samples of ultracold $^{87}$Rb over several millimetres, and by quickly toggling the inputs between multiple pairs of frequencies, multiple clouds were confined and moved in time-averaged potentials. The utility of the FPGA/DDS unit is shown, most notably by simultaneously evaporating four independent clouds through the Bose-Einstein Condensate (BEC) phase transition.
Acknowledgements

I’ve been very fortunate to have had such profound role models during my time in the Ultracold Atoms group. Thank you, Niels, for taking me on for that first summer project and having me stay. It has been a great privilege. I have a lot of respect for your dedication and ability. Amita, your insights and attention to detail, and patience, are legendary. A special thank you for all your guidance when my device was used in the main lab, and for fitting all the lifetime data that appears in Section 4.2 of this thesis.

To Bianca and Kris, Ana and Julia, Milena and Ryan, I’ve felt a strong sense of teamwork throughout this project and you’ve all been part of it. I hope I contributed to it. I wish you all the best of luck in your further studies. Thanks especially to Kris, Ana, and Julia, for doing all the mechanical work on the optical tweezer and for trialling the first versions of frequency control code I produced.

I’ve had a lot of help from outside the group, too. Thanks to all the department staff, you’ve been really great. Same to my office mates, thanks for all the technical advice and non-technical discussions.

The list doesn’t end within the department either. I’ve been very lucky to have such great friends, inside and outside the tramping club. Thanks guys! And Mum, Dad, Lochiel, and Sophia, and Lewie, you’ve all been very supportive, and fun, and quirky some times, and you all mean a lot to me.

And finally, thank you Laura D., for being so flippin’ awesome.
Contents

Abstract ii

Acknowledgements iii

Some Technical Abbreviations vi

1 Introduction 1

2 Background 3
   2.1 Cold Atom Trapping and Transport 3
   2.2 Cooling and Trapping in the Ultracold Atoms Lab 5
   2.3 Optical Chopsticks 8

3 Hardware and Software 9
   3.1 Inside a Complete Direct Digital Synthesiser 9
   3.2 Field Programmable Gate Arrays 10
   3.3 Hardware Used in This Thesis 11
   3.4 Core Algorithms 12
   3.5 FTW Preparation 15
   3.6 Amplitude Control 20
   3.7 Summary of FTW Behaviours 22
      3.7.1 Two Cloud Split and Hold 23
      3.7.2 Four Cloud Evaporation 23
      3.7.3 Unidirectional Cloud Movement 24

4 Deployment 25
   4.1 DDS Outputs 25
   4.2 Multiplier Calibration 28
   4.3 Induced Breathing 31
5 Application

5.1 Four Cloud Condensation ......................... 33
5.2 Cooling by Trap Deformation .................... 38
5.3 Comparison of Different Motion Profiles ........ 42
  5.3.1 Analytic Expressions for the Six Profiles .... 42
  5.3.2 Expected Position of a Cloud After Time of Flight ... 45
  5.3.3 Observation of Sloshing ..................... 46

6 Summary ........................................ 51

A Dipole Trapping in Gaussian Beams ............... 52

B Additional FPGA programs ........................ 54
  B.1 Collision Programs ............................ 54
  B.2 An FPGA-based Timing Control Unit ............ 56

Bibliography ..................................... 58
Some Technical Abbreviations

**AOD**: acousto-optic deflector. A device that deflects and modulates the amplitude and frequency of a laser beam. An AOD differs from an acousto-optic modulator, an AOM, in that its primary function is to deflect the beam.

**ASF**: amplitude scale factor. A binary value which defines the amplitude of the output of a DDS. The AD9959 DDS used in this thesis work accepted a 10-bit ASF.

**BEC**: Bose-Einstein condensate. An exotic form of matter in which many atoms in an atomic cloud populate the ground state of the harmonic trap it is confined in.

**DDS**: direct digital synthesiser. Electronic hardware designed to output a sinusoidally oscillating voltage.

**FPGA**: field programmable gate array. An extremely useful and versatile piece of electronic hardware which is able to be programmed by the user to perform a specific application.

**FTW**: frequency tuning word. A binary value which defines the frequency of the sinusoidal output of a DDS. The AD9959 DDS used in this thesis work accepted a 32-bit FTW.

**I/O**: input/output.

**IP trap**: Ioffe-Pritchard trap. A type of magnetic trap with electromagnetic coils arranged in a particular geometry, used in our lab to trap and cool the atoms before passing them to the optical dipole trap.

**LUT**: look-up table. A commonly used component of digital circuitry which maps an input into an output by a memory reading operation rather than a logical process or computation.

**PROM**: programmable read-only memory. A component on the FPGA designed to store configuration data, but also used to store and communicate to the FPGA numerical parameters which defined the output of the DDS.

**RF**: radio frequency. Either a frequency less than 300 GHz, or a voltage which oscillates at a frequency less than 300 GHz.
Chapter 1

Introduction

Over the last century, atomic physics has undergone an extraordinary evolution, from beginnings in spectroscopy to modern measurements of highly controlled atom-atom interactions [1]. This broad field underpins a number of very active areas of research, such as quantum engineering, quantum information processing, and quantum metrology. Such fields call for highly engineered methods to trap and cool atoms, and for precise control of the internal and position states of matter cooled to its most fragile form.

The Ultracold Atoms Group at the University of Otago have the capability to cool both bosonic \(^{87}\)Rb and fermionic \(^{40}\)K to the degenerate regime. Recent areas of research have included spin waves [2], non-destructive probing [3], and same state collisions between clouds of ultracold \(^{87}\)Rb [4]. Bose-Einstein condensation was first achieved in 2011 [5], and more recently \(^{40}\)K has been cooled to degeneracy. The ability to work with atoms of different species in arbitrary quantum states is a defining characteristic of the apparatus, with collisions between atoms of different states and species under highly tunable conditions being a long term goal of the group.

Optical trapping is often used in the final stages of an experiment, allowing ultracold atomic samples in any quantum state to be confined. Independent of this thesis work, a versatile optical tweezer unit was designed and installed in the experiment [6]. With its dual-axis acousto-optic deflector (AOD), the device is able to deflect a single laser beam in two dimensions, with deflection angles defined by two radio frequency (RF) inputs. With well chosen RF signals, the beam intersects a perpendicular, stationary beam to form a crossed-beam dipole trap. By deflection of the movable tweezer beam, translation of an optical trap, and thereby its cargo, is enabled.

Static confinement of an ultracold cloud is straightforward to achieve, by driving the AOD with two signals of constant frequency. Cloud movement, especially smooth movement, is more challenging and requires careful synchronised control of the two RF inputs. The goal of this thesis work was to develop a frequency source to steer the tweezer beam. For this purpose, a field programmable gate array (FPGA) was programmed to control a direct digital synthesiser (DDS) which could then drive the paired inputs to the optical tweezer.

Chapter 2 gives some context for the work presented in this thesis. It begins with a short review of trapping techniques, particularly in the context of an ultracold collider. It follows with an overview of the ultracold \(^{87}\)Rb machine at Otago, which existed before this thesis began. Last is the reasoning that led to the use of a frequency toggling method used
to trap multiple clouds simultaneously. The technical demands of the method required the frequency source described here to be developed.

Chapter 3, “Hardware and Software”, describes the hardware of the two electronic devices used in the device, a DDS and an FPGA. Technical details of the programming process follow, including part numbers of the evaluation boards used and the computational processes that take place in the FPGA. A summary of the three types of program rounds out this chapter.

Chapter 4 and Chapter 5 contain experimental results. The former presents characterisation data that confirms the FPGA/DDS unit behaves as it is expected to. The latter contains results that fall into the “Applications” category. The work undertaken during the project is summarised in Chapter 6, and some outlooks for future work are discussed.
Chapter 2

Background

2.1 Cold Atom Trapping and Transport

Motivations for moving atomic clouds vary, from improving optical access to transferring a sample into a more isolated environment [7], spatially merging clouds of different atomic species [8], and probing the structure of a surface [9]. The Otago apparatus has been designed to function as an atomic collider, and in this application, where the energy involved in a collision plays a pivotal role, a high degree of spatial control is crucial.

Transport of ultracold clouds has been realised in several types of trap. Highly prominent are magnetic traps, in which atoms with magnetic dipole moments are confined by strong, static magnetic fields. A simple way to achieve a movement in this type of trap is to move the sources of the magnetic fields, whether permanent magnets or electromagnetic coils, and this is commonly done in our laboratory (see Section 2.2). Clouds have also been passed between sets of wound coils and over electromagnetic chips by carefully ramping currents that generate magnetic fields [10, 11], passing the atoms from trap site to trap site as in Figure 2.1(a). Superimposed magnetic traps have been used to facilitate high energy-collisions [12, 13]. However, their utility extends only to atoms in magnetically trappable Zeeman substates.

An alternative type of trap is the optical lattice, the interference pattern of two counter-propagating laser beams red-detuned from an atomic resonance (Figure 2.1(b)). Under the influence of the optical dipole force, neutral atoms will be attracted to regions of high intensity of the resultant periodic potential (see [14] or [15], or Appendix A for a description of the dipole force in the context of a crossed-beam dipole trap). If the frequencies of the beams are equal the interference pattern is a standing wave, and the atoms will be held stationary. But if they differ slightly, the antinodes will translate with constant velocity through space, giving a way to realise movement of particles trapped at these sites. Due to the high trap stiffness along the axis of the lattice, large accelerations and thus high velocities can be achieved without atom loss. It has been shown that by using a non-diverging Bessel mode for one of the beams, the atoms can be supported against gravity over distances of tens of centimetres [16], overcoming a typical limitation in optical trapping. Unintuitively, their application even extends to collisions. Very low energy collisions have been observed between atoms in a single cloud excited to symmetric momentum states with optical lattice pulses [17]. More relevant to the apparatus at Otago, macroscopic splitting of atoms in different spin states has been performed by careful tuning.
of the relative polarisations of lattice beams to create a superposition of state-selective trapping potentials [18, 19]. However, neither scheme achieves complete flexibility of high energy collisions between spatially separated clouds in arbitrary states.

Perhaps the most versatile tool for trapping is an optical tweezer. This type of trap, like an optical lattice, works by the optical dipole force. However, a tweezer does not rely on a standing wave to cause axial confinement, and the particle may range from a neutral atom to a bacterium several micrometres in size [20]. 3-D trapping potentials may be created by tightly focusing a single tweezer beam to a narrow waist (Figure 2.1(c)), where the high beam divergence results in an appreciable trapping force in the axial direction as well as radial directions [21], or by intersecting two collimated beams at a point (Figure 2.1(d)). The latter type of trap has been chosen for use in our experiment, for a number of reasons. First, optical trapping allows experiments to be made on atoms in states that are not magnetically trappable. Just as important, the position of the atom cloud and the external magnetic field can be controlled independently, a profound capability for any cold atom experiment as it allows the magnetic field to be used to tune atom-atom interactions. The ability to turn the trap on and off in microseconds, rather than milliseconds in a magnetic trap, is a major perk. But, most importantly, simple deflection of laser light is straightforward, allowing easy manipulation of the atoms it traps.

Figure 2.1: Movement of ultracold atoms with (a) magnetic traps, (b) an optical lattice, and (c & d) optical tweezers.
When moving an ultracold cloud in any type of trap, a movement must be carefully engineered to cause minimal heating and leave the sample with minimal in-trap movement. This can be achieved with two types of movement: adiabatic, and non-adiabatic. When an atom cloud is moved over distances much greater than its natural width in a time period comparable to or shorter than the trapping period along the axis of movement, the transport is non-adiabatic [22, 23]. In this regime, it is possible to move atoms without causing heating or in-trap movement. By carefully choosing transport parameters, a cloud can be moved in such a way that it will slosh within its trap as it is accelerated, then be brought perfectly to rest as it is decelerated. The amplitude of the final centre of mass motion of the cloud is given by the Fourier transform of the velocity profile. Such motion is advantageous in that it can be very fast, but requires low trapping frequencies and careful control of the trap accelerations. It is therefore better suited to deep, low-frequency magnetic traps, and not to optical traps with frequencies of hundreds of Hertz. If a cloud is moved over macroscopic distances in a time much longer than the trapping period, the motion is known as adiabatic. It is generally slower than non-adiabatic movement, but does not require the same careful engineering. Such movement is well suited to the optical traps used in this thesis work. But in spite of its name, adiabatic movement usually does impart energy to its cargo, and some such movements can be smoother than others. Some attention has been given to moving atoms with especially smooth motion profiles. In [4], trapezoidal acceleration profiles are implemented in splitting and colliding clouds. In [16], a piecewise cubic acceleration profile is used to move atoms trapped in an optical lattice, giving a very smooth motion during which the jerk, the time derivative of acceleration, changes continuously. In [24], focused retroreflected dipole traps forming an optical lattice, too, are shifted on coupled translation stages with a sinusoidal acceleration to and deceleration from a maximum velocity. However, most movements of ultracold clouds within the adiabatic regime tend to use a constant or linear velocity profile. For most work in this thesis, atomic clouds were moved with a profile that minimised the integral of the square of the jerk through the motion [25]. A few other motion profiles within the regime were compared experimentally at the end of the experimental work.

2.2 Cooling and Trapping in the Ultracold Atoms Lab

The cooling apparatus described here, sketched in Figure 2.2 and documented in Ana Rakonjac’s Ph.D. thesis [5], existed before this project began. It is similar to the system described in [7]. Both $^{87}$Rb and $^{40}$K are initially cooled from background vapour in a magneto-optic trap (MOT). Further cooling is achieved by running an experimental sequence which is preprogrammed into the computer control program, RebeKa, and has duty cycle of about 100 s. From the MOT, rubidium atoms undergo compressed MOT and optical molasses phases to cool them further, and are optically pumped into the $|F = 2, m_F = 2\rangle$ state. Then they are physically transported 53 cm from the MOT chamber to a Science Cell, a chamber evacuated to below $10^{-11}$ torr and devoid of background atomic vapour. The transport is carried out in a movable magnetic quadrupole trap labelled as “Transfer Coils” in Figure 2.2. Here they are passed to an Ioffe-Pritchard (IP) trap [26] for forced radio-frequency (RF) evaporative cooling, which cools the atoms to the microKelvin regime and beyond. It was this evaporative cooling process that led to
the first observations of condensation [5].

After cooling, ultracold samples are able to be loaded into a crossed-beam optical dipole trap. The two trapping beams are generated by a 1064 nm Nd:YAG fibre laser and divided by a system of polarising beam splitters. Each beam is then passed through an acousto-optic modulator (AOM). By regulating the amplitudes of the 110 MHz inputs to the AOMs, via voltage variable attenuators controlled from RebeKa (ATTs in Figure 2.2), the optical powers in the beams are able to be controlled independently. Following their amplitude modulations, the beams are coupled into fibres and shuttled to the main experiment table. One beam is passed horizontally through the Science Cell in the direction of the MOT chamber, and is hereon called the H-beam. The other propagates vertically from below - the V-beam.

It is this crossed-beam dipole trap that forms the movable optical tweezers. Before it is reflected vertically the V-beam passes through a dual-axis acousto-optic deflector (AOD)\(^1\). The AOD is able to deflect the transmitted beam in two orthogonal directions, illustrated

\(^1\)a DTSXY-250-1064 made by AA Opto-Electronic

![Figure 2.2: Schematic of the experiment, with particular focus on the Science cell and optical tweezers. The FPGA and DDS, in green at the bottom right, accept inputs from the main experimental control PC and output to the AOD RF inputs. This figure was adapted from one already published [27], which was drawn by Niels Kjærgaard.](image-url)
in Figure 2.3 (a). Much like an AOM, it deflects and modulates a beam by the acousto-optic effect (see, for example, [28]). The AOD, however, has been designed to effect a wide range of deflections rather than amplitude or frequency modulation. The deflection angles \( \theta_x, \theta_z \) are fixed by the frequency of sinusoidal signals driving the \( f_x \) and \( f_z \) inputs to the AOD.

After being transmitted through this device, the V-beam is reflected and propagates upwards through the Science Cell. It passes through a lens, whose front focal point coincides with the centre of the AOD. As a result, the light always propagates parallel to the axis of the lens, regardless of the deflection angles assigned by the AOD. After a careful mechanical alignment of the AOD and lens, the deflected and refracted beam propagate almost exactly vertically, as Figure 2.3 (b) shows. Further, the lens is an F-Theta Lens\(^2\), which ensures the V-beam focuses to a waist at a constant height, in spite of the varying path lengths. Together, the AOD and F-Theta Lens allow the V-beam to be translated in a plane over distances up to 6 mm in either direction. The development and characterisation of this steerable optical tweezer unit was carried out independent of the frequency-control work described in this thesis, and is recorded in Kris Roberts’ Honour’s dissertation [6].

While the V-beam could be moved, the H-beam remained fixed, and in this way it acted as a stationary waveguide. The horizontal plane at the height of the H-beam was defined to be the \( xz \)-plane; by varying \( f_x \) and \( f_z \) within a 60 - 90 MHz range, the beam could be made to intersect this plane at any point over a 6 mm \( \times \) 6 mm area. Raising \( f_x \) caused this point of intersection to move in a definite direction - this direction defined the \( x \)-axis. Similarly, the direction of movement under an increase in \( f_z \) defined the \( z \)-axis. The H-beam ran as parallel to the \( z \)-axis as mechanical alignment allowed, so the V-beam could be moved significant distances in this direction without losing its intersection with the H-beam. Changes in \( f_x \) would quickly cause misalignment, and this control was usually changed only to correct for misalignment of the beams when \( f_z \) varied by a large amount from its central value.

\(^2\)custom-made by Eskma Optics with a 130 mm focal length

Figure 2.3: (a) Oblique view of deflection of the incident beam as it passes through the dual-axis AOD. The frequency of input sinusoid \( f_x \) maps to a horizontal deflection angle \( \theta_x \), and similarly for \( f_z \) and \( \theta_z \). (b) Side-on view of the 1st order deflected beams being reflected upwards and propagating in parallel after refraction through the F-Theta Lens. The focal length of the lens is \( d_1 + d_2 \).
CHAPTER 2. BACKGROUND

2.3 Optical Chopsticks

To bring about any sort of multi-cloud manipulation with optical tweezers, multiple points of intersection are needed - and hence, multiple tweezer beams. One solution would have been to drive the dual-axis AOD with two pairs of frequencies simultaneously. Of the four resulting output beams, only two would have intersected the waveguide correctly to trap a sample, and half the optical power would have been lost. Generalising to \( n \) pairs of frequencies, only \( 1/n \) of the light is retained for trapping, while \( 1 - 1/n \) bypasses the waveguide and is lost (see Figure 2.4). As an alternative, we toggle quickly between pairs of frequencies, creating \( n \) time-averaged potential minima along the static waveguide - optical chopsticks [29]. While this method circumvents the large scale loss of optical power, it introduces sidebands to the spectrum of both AOD drive signals, and a small amount of optical power is still lost. Most detrimentally, some of the sideband power coincides with the waveguide, creating local minima which could trap escaped atoms.

In our experiment, the frequency toggling is phase continuous [30]. The toggle rate is defined by the hop time, \( \tau_h \), the length of time for which each trap site is addressed. Thus, the period of a whole cycle, in which \( n \) sites are addressed sequentially, will be

\[
\tau_{cycle} = n \tau_h.
\] (2.1)

The toggle frequency should be fast compared to the trapping frequency of the toggling beam [31], with the trapping frequency itself being the absolute lower limit. The resulting optical potential is reduced by the time averaging [32]. In our experiment, with the optical beam addressing a certain number of trap sites, the time-averaged optical power was simply the beam power divided by the number of trap sites.

Using the AOD approach, manipulation of cold clouds of atoms via optical tweezers boils down to fast, exact control of the two signals driving the dual-axis AOD. To achieve smooth movement and, in the context of a collider, precise collision energies, the signals must vary on the microsecond time scale. A number of options existed to control the signals. The original collision experiments in our lab were carried out with voltage controlled oscillators (VCOs) controlled by an arbitrary waveform generator [4]. Expensive DDS packages are available, which read back series of data from memory. One such device, a WieserLabs FlexDDS, was used during the development of the tweezer unit. The bulk of this thesis work was in developing a low cost alternative: a commercial DDS under the control of an FPGA.

---

Figure 2.4: Top-down view of optical power in the \( xz \)-plane for various AOD drive schemes.

a) Simultaneously driving the AODs with two pairs of frequencies causes loss of half the optical power from the trapping points. b) Loss of \( 1 - 1/n \) of the optical power, for \( n = 4 \). c) Retention of most optical power by toggling between pairs of frequencies, with slight loss into sidebands.
Chapter 3

Hardware and Software

3.1 Inside a Complete Direct Digital Synthesiser

A direct digital synthesiser (DDS) is a device whose analogue output voltage is a sinusoidal wave produced in a digital manner [33]. In this way it is fundamentally different to a voltage controlled oscillator (VCO), another device commonly used as a frequency source. A DDS works essentially by dividing down the frequency of a very fast external clock signal. During every clock cycle, the DDS calculates a digital value which is then mapped to an analogue output voltage. There are two components of the calculation, performed by a phase accumulator and a phase-to-amplitude converter. Last is a digital-to-analogue converter (DAC), which completes the circuitry known as a complete-DDS [34]. These three components are sketched in Figure 3.1 and discussed in greater detail below.

The core of the phase accumulator is a register storing an $N$-bit data string. This value, the phase word, represents a positive integer, and due to the finite length of the register it must be of modulus $2^N$. Under normal operation, the phase word is ramped linearly in time by repeatedly adding to it an $N$-bit positive constant value and rewriting the phase word register with the sum. When the stored phase word is large enough for the sum to be greater than $2^N - 1$, the maximum value representable with $N$ bits, the most significant bit (MSB) of the sum is discarded and the phase word “rolls over” to a value of or near zero. The frequency at which the phase word cycles is defined by its ramp rate, determined by the system clock frequency and by the value of the constant which is repeatedly added to it. For this reason, the constant is named the frequency tuning word (FTW).

Clearly, if the phase word evolves periodically then the output of the DAC will be periodic too. To achieve a sinusoidal output, the linearly evolving phase word must be mapped to a sinusoidally evolving value. The product of this mapping is determined with a Look Up Table (LUT) approach - by using the FTW to address a Read Only Memory (ROM) - and it remains a digital value. It is this value that is passed to the DAC for conversion to an analogue output. The amount of memory required can be reduced with tricks that exploit the symmetry and the small angle approximation of a sinusoid [35]. As the instantaneous frequency is the time derivative of phase, it is linearly related to the FTW, the rate of phase accumulation. The frequency output by a DDS with an $N$-bit FTW and running
Control of the amplitude and phase offset is enabled by extra circuitry between these three chief blocks. A user defined phase offset is added to the accumulated phase word before the first conversion, allowing direct phase control. This sum is not the one stored in the phase register, so the phase offset has no effect on the accumulation rate. Finally, the output of the LUT is multiplied by another digital value, the amplitude scale factor (ASF). The product is used as the input to the DAC.

Hence, an input clock signal and three digital values - the frequency tuning word, the phase offset, and the amplitude scale factor - determine the time evolution of a DDS output. By precisely programming and reprogramming these digital parameters the sinusoidal output can be made to jump, scan, and toggle in all three parameters in near arbitrary patterns. This flexibility has been exploited for many applications, including moving optical traps with AODs [36] using a DDS under microcontroller control. In the current application a field programmable gate array was used as a controller, and the resulting system could act on the microsecond timescale.

3.2 Field Programmable Gate Arrays

Field programmable gate arrays (FPGAs) are digital logic devices ideally suited to an extraordinary range of applications. These devices smooth the boundary between application specific integrated circuits (ASICs), non-configurable devices with very high speed performance, and processors, which are versatile but reconfigure the data path every time an instruction is executed [37]. They were introduced to physically fill the gap between the predesigned, rigidly inflexible integrated circuit components (such as microprocessors, timers, memories) used in logic systems [38]. An FPGA is a dense unconnected array
of digital logic components such as look up tables (LUTs), multiplexers, flip-flops, and random access memories (RAMs). These units are able to be connected by a user with a specific application in mind, and the device thereby has the potential to perform a nigh-uncountable number of distinct functions.

A typical FPGA is largely comprised of a two-dimensional array of configurable logic blocks (CLBs). Each of these contains one or two small LUTs and D flip-flops, buffer logic, and input/output (I/O) ports. CLBs are interspersed among General Routing Matrices (GRMs), which connect transmission paths running between the blocks. Remarkably, the network of connections within and between the CLBs is reconfigurable. All connections between components are passed through CMOS transmission latches \[39\] whose “connection” inputs are managed by small local static RAM (SRAM) units. The SRAMs are programmed from an external programmable read only memory (PROM) or CPU. While the space and power requirements of the SRAM network are considerable, the product - a device that can run at the nanosecond timescale of an ASIC but with flexibility akin to a processor - is extraordinarily powerful.

The CLB/GRM array is bordered by input/output blocks (IOBs) which interface between the logic array and external devices. These, too, are able to be connected almost arbitrarily to internal signal paths. Many FPGAs are available on development boards which come with peripherals such as USB connections, analogue-digital or digital-analogue converters, crystal oscillators for clocking, differential I/O headers, external PROMs and flash memories. These peripherals come connected directly (and non-reconfigurably) to the IOBs, resulting in a versatile package.

The final strength of an FPGA lies in its easy configurability. While the desired behaviour can be managed at the gate level, it is more common to use a hardware definition language (HDL) such as Verilog or VHDL (in which V stands for VHSIC, Very High Speed Integrated Circuit). The chip developer’s software readily converts these higher level code instructions into hardware arrays and configuration files.

In the end, the high performance, ease of use, and reconfigurability of an FPGA make it the ideal choice for many digital applications. Indeed, FPGAs are commonly used to program direct digital synthesisers \[40\].

### 3.3 Hardware Used in This Thesis

The FPGA used in the work for this thesis was a Xilinx Spartan-3AN (XC3S700 AN) Starter Kit board \[41, 42\]. All designs were clocked by the on-board 50 MHz crystal oscillator (input pin E12 on this board). The designs were written in VHDL with the Xilinx Project Navigator software, and this software would also synthesise and implement the design (map it to a hardware array appropriate for this chip), and generate the file needed to program the device. The designs were written with Xilinx’s iMPACT software through a USB connection.

The FPGA controlled an Analog Devices AD9959/PCBZ DDS \[43\] running in Manual Control mode. This DDS took a 32-bit FTW, a 10-bit ASF, and a 14-bit phase offset. The phase computation and conversion to analogue could run with clock frequencies of up to 500 MHz. However, digital communication was limited to a maximum clock frequency
of 50 MHz. Although the DDS had four channels, able to be controlled independently, only two channels were used simultaneously throughout this work.

The devices were connected by a length of ribbon cable. The FPGA’s differential I/O “Transmit” header (port J15) was used for all data channels, with the mapping of pins given in Figure 3.2. The dedicated SMA clock I/O connector (SMA output U12) was used as a source for both the DDS’s reference clock (SMA input J9), the input of which is used for the system clock, and the communication clock (SCLK). The reference clock was multiplied up to 500 MHz on the DDS by an on-board phase-lock loop (PLL), so both communication and computation were driven at their maximum rates. Finally, the DDS was run in 4-bit serial mode, where data was transferred synchronously on four ports.

With the DDS in Manual Control mode, a write operation was indicated by lowering the “Chip Select” (CS) input and holding it low for the duration of a data transfer. Register data was preceded by an address byte, to indicate which register was to be rewritten (whether FTW, ASF, phase offset, or one of various control registers), and a pause of one clock edge in the data transfer. Then the bytes of data would follow, followed by an Update signal lasting a single clock cycle. Additionally, if specific output parameters were to be written to a particular channel it had to be defined with a Channel Select instruction of two bytes, followed by a one-edge pause in the data transfer. Under these conditions, a single FTW could be rewritten and updated in twelve clock cycles (240 ns). Independent FTWs and ASFs for two channels could be written and updated in 49 clock cycles (980 ns).

Throughout the project, heavy use was made of the Xilinx XCF04S serial Platform Flash PROM, which could be written directly from Xilinx’s iMPACT software. Designed primarily for storing configuration data, the memory would automatically be read and the FPGA configured as soon as it was switched on. However, the device was able to access the memory once it was running, which allowed user data to be appended to the configuration data and read back for use in the FPGA’s application. In this way, numerical parameters defining its behaviour - a list of FTWs to be uploaded to the DDS, for example, or the number of clock edges to wait between predefined events - could be communicated to the FPGA.

### 3.4 Core Algorithms

With the AD9959 DDS, there were two ways to carry out a controlled sweep in frequency. One was to reprogram a series of FTWs in quick succession, realising a continually changing frequency trajectory at timescales much longer than the reprogramming time. The other was to use the DDS’s in-built linear sweep function, which caused the inner FTW to ramp linearly. In this mode, FTW steps were able to be made as frequently as every 8 ns, a much finer time resolution than would be possible if the FTWs were programmed externally. Programs implementing the latter type of movement were trialled, and did have some success. But they were soon rejected in favour of the former type, which was found to be far more flexible. This section gives a technical description of the algorithms involved in the former type of program, used for the bulk of the thesis work. As a reference, a brief description of the latter type is found in Appendix B, as well as some images of atoms split and collided with these programs.
In programs that did not use the linear sweep function, the core behaviour of the FPGA was a loop which periodically reprogrammed the FTWs and ASFs for two channels of the DDS. Typically, this “upload cycle” would take 50 clock edges (1 $\mu$s). Each cycle, the core loop would load the values available and trigger the preparation of those to be uploaded in the next. When the system was in an idle state the preparation process amounted to nothing, and the FTWs would remain unchanged. But when the system was performing a frequency sweep, the FTWs would change in accordance with the desired frequency trajectory. The method of preparing the FTWs is the subject of Section 3.5.

To achieve simultaneous movement of two clouds, two pairs of FTW registers existed. The core loop would load from one register pair for a fixed number of upload cycles. Then it would swap to the other and load that the same number of times. The result, depicted in Figure 3.3(a), was an output wave which toggled between two sinusoidal modes of different frequency, with a hop time

$$\tau_h = n\tau_{\text{load}},$$

where each register was loaded $n$ times in a row and the time of one upload sequence was $\tau_{\text{load}}$. Similarly, cycling through four different FTW registers (Figure 3.3(b)) and loading each $m$ times resulted in an output which toggled among four frequencies, with hop time

$$\tau_h = m\tau_{\text{load}}.$$

Usually, $m$ was chosen to be about half of $n$, which meant the rate at which sites were addressed remained constant. Hop times would typically be on the order of 6 $\mu$s for two clouds, or 3 $\mu$s for four clouds, such that each site was addressed at a rate of $\sim$83 kHz. At these toggle rates, orders of magnitude greater than the trapping frequency, the clouds truly would experience simultaneous trapping forces; even though the optical trap toggled between the sites, the atoms would experience a potential averaged over the toggle period.

At all times, the DDS would output two frequencies of non-zero amplitude. It would have been straightforward to turn the outputs on and off, especially using the “Power down” pin on the DDS, but it was desirable to drive the AOD constantly, at least for...
Figure 3.3: Demonstration of how a toggling output was implemented. Each horizontal bar lasts for one load period (usually chosen to be $1 \mu$s). (a) A two cloud splitting process with $n = 4$. (b) A four cloud splitting process, during both the first and second splitting phases. Notice how each trap site is addressed once per $4\tau_h$, but the time it is addressed for halves when two clouds are split into four.

several seconds before the dipole trap was switched on. In this way, the AOD would be kept warm, eliminating thermal effects. With the AOD inputs constantly on, the vertical beam of the trap was switched on and off with the V-beam fibre coupled AOM.

A cloud movement would be triggered by an external signal, a sequence trigger, read on pin A11 on the J2 I/O differential header. When the external pin was asserted, the FPGA would run through a series of phases of the movement, outputting FTWs to the DDS throughout. The sequence would run until a low signal was registered on the same pin, at which time it would reset. If the trigger input was held high, the FPGA would run to the end of its sequence of movements, then wait in a “Done” state - still outputting the final, static FTWs - until the trigger signal was lowered. In this way, trap sites could be moved and held in place indefinitely.

When moving a cloud along a desired trajectory, the series of FTWs that would follow that trajectory was well defined and finite. The FPGA would count the number of upload cycles from the beginning of the movement, and when it reached the predesignated count total (loaded from the memory) it would progress to its next behavioural state: sometimes a motion phase continuing the movement (in the case of a piecewise defined trajectory), sometimes a phase in which the clouds would be held at rest. The times of all phases were communicated via memory as 24-bit numbers.

Supporting the central DDS programming loop were smaller submodules of the design. One module would detect a memory write from the computer and automatically read the data after its completion, to be stored in logic on the FPGA. As data in the Xilinx PROM was unable to be addressed, a memory read operation entailed a readback of the entire memory contents. Data was prefaced by a characteristic 32-bit “synchronisation word” [44]. Using a shift register, the memory read module would compare the memory output to the preprogrammed synchronisation word. Before the pattern was located, the data would be discarded after this comparison. Once it had been found, a predetermined number of
bytes following it would be stored in logic on the FPGA to be used as parameters. By using different synchronisation words, multiple sets of data could coexist in the memory, allowing the system to run a sequence of experiments. After an experiment, the synchronisation word would be interpreted as a number and increased by one, resulting in a different synchronisation word which marked the following data set in the memory.

Finally, there was a simple module which would write information to the LCD screen: the version of code the FPGA was currently programmed with (essential during the design process), the time and date at which the memory file was created (to be used as a label for a particular set of parameters used in an experimental run), and the number of the current experimental sequence.

### 3.5 FTW Preparation

This section describes in detail the process of preparing a pair of frequency tuning words for the style of program that did not utilise the DDS’s linear ramps. This procedure would be triggered at the beginning of an upload cycle, during which the results of the previous preparation sequence were uploaded. A preparation usually took 21 clock cycles (420 ns), and the FTWs would be ready well before they were called to be uploaded to the DDS.

The FTW pairs could have been read by the FPGA from an on-board memory. Instead, a calculation approach was used to compute the FTWs in real time, avoiding a large-scale memory reading. Before an experimental sequence, a few hundred bytes of data defining the FTW trajectory were read from the single bit serial output of the Platform Flash PROM. This data was interpreted in a preprogrammed way and defined a complex FTW behaviour which could last over timescales from microseconds to several seconds. The behaviour was limited to that allowed by the FPGA design - a simple rewrite of the user defined memory data could not bring about an alternate behaviour. All that could be changed were the parameters shaping the behaviour: the distance of a movement, the time of each phase, and the wave amplitude, for example. However, any behaviour could be brought about by reprogramming the device to interpret the user data and act in the desired way.

As an example, the most straightforward movement is one at constant velocity. For a computation based method, this trajectory can be defined with just a few parameters: the initial FTW, an accumulation rate, and a final FTW. The accumulation rate can be defined either as an amount to increase the current FTW by on each upload cycle, or a number of cycles to wait before the next accumulation event, or both. An alternative to defining the final FTW is to define the number of upload cycles the movement phase must last. In any case, this simple trajectory can be fixed by a few bytes of memory-read data, rather than a lengthly string of FTW pairs.

To achieve a smoother movement than one at constant velocity, the FTWs were programmed to move along higher order polynomial trajectories. Initially, it appears that this process would require multiplications. But for large $t$ ($\sim$100,000 upload cycles over 100 ms) and precise coefficients taken to many decimal places, the resources required for a single array multiplier [38] were huge - prohibitively so. It would have been possible to construct a series of sequential binary multipliers [39], but the time resolution of the FTW upload loop would have suffered greatly. In addition, the final result would have been a
sum of large positive and negative products which would cancel to give the right result - meaning rounding errors in the cumbersome calculation could lead to large deviations in the sum from its desired value.

As an alternative, an accumulation approach was developed, which exploited the fact that the series of FTWs defining the motion were to be calculated sequentially. We chose to implement fifth-order polynomials, largely so we could move clouds with a smooth trajectory that minimised the integral of the square of the jerk in a one-dimensional motion between two stationary states [25]. This method could have been extended to higher order polynomials, with the cost in hardware increasing linearly with order. A description of the method follows; in this discussion, typewriter font indicates the name of a register on the FPGA, whereas italicised font denotes a number. The name of a register written in italics indicates a number, the binary value stored in that register. A double arrow (⇐) assigns a numeric value to a register.

Consider the general fifth order polynomial function

\[ FTW(t) = FTW_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \]  

(3.2)

where \( FTW(t) \) is the value of the frequency tuning word at time \( t \) and the \( a_i \) are constants.

By defining new constants \( b_1, b_2, b_3, b_4, b_5 \) such that

\[
\begin{pmatrix}
    a_1 \\
    a_2 \\
    a_3 \\
    a_4 \\
    a_5
\end{pmatrix}
=
\begin{pmatrix}
    1 & 1/2! & 2/3! & 6/4! & 24/5! \\
    0 & 1/2! & 3/3! & 11/4! & 40/5! \\
    0 & 0 & 1/3! & 6/4! & 35/5! \\
    0 & 0 & 0 & 1/4! & 10/5! \\
    0 & 0 & 0 & 0 & 1/5!
\end{pmatrix}
\begin{pmatrix}
    b_1 \\
    b_2 \\
    b_3 \\
    b_4 \\
    b_5
\end{pmatrix}
\]  

(3.3)

\( FTW(t) \) can be manipulated to the form

\[
FTW(t) = FTW_0 + b_1 t + b_2 \frac{t(t+1)}{2!} + b_3 \frac{t(t+1)(t+2)}{3!} + b_4 \frac{t(t+1)(t+2)(t+3)}{4!} + b_5 \frac{t(t+1)(t+2)(t+3)(t+4)}{5!}.
\]  

(3.4)

It will be shown that if the function is to be evaluated only at sequential, evenly spaced times, the \( t \)-dependent fraction terms in Equation 3.4 can be calculated in a far more hardware-efficient way than the obvious multi-operand multiplication required to evaluate the powers of \( t \) in Equation 3.2 directly.

Consider the sequence \( S_1 \equiv 1, 2, 3, 4 \ldots \). The \( K \)th term in the sequence will be

\[ S_1(K) = \sum_{k=1}^{K} 1 = K. \]  

(3.5)

Likewise, consider the sequence of numbers,

\[ S_2 \equiv 1, 3, 6, 10 \ldots, \]  

(3.6)

defined as

\[ S_2(K) = \sum_{k=1}^{K} S_1(k). \]  

(3.7)

But also notice the strict quadratic form of the terms of \( S_2 \):

\[ S_2(K) = \frac{K(K + 1)}{2!}. \]  

(3.8)
Similarly, the list of such sequences can be extended indefinitely as

\[ S_N(K) = \sum_{k=1}^{K} S_{N-1}(k), \quad (3.9) \]

the \( K \)th term of each being the sum of the first \( K \) terms of the next sequence below it. By manipulation of Equation 3.9, it is seen

\[ S_N(K) = S_{N-1}(K) + \sum_{k=1}^{K-1} S_{N-1}(k) \quad (3.10) \]

\[ = S_{N-1}(K) + S_N(K-1). \quad (3.11) \]

The \( K \)th term of the \( N \)th sequence can be obtained by taking the most recent term in that sequence and adding the \( K \)th term of the \((N-1)\)th sequence. The terms of sequence \( N \) obtained in this way will have the polynomial form

\[ S_N(K) = \prod_{n=1}^{N} \frac{(K + n - 1)}{N!} \quad (3.12) \]

\[ = \frac{1}{N!} (K)(K + 1)(K + 2) \ldots (K + N - 2)(K + N - 1). \quad (3.13) \]

Finally, consider the sequence \( R_1 \equiv \gamma S_1 = \gamma, 2\gamma, 3\gamma, \ldots \). Accumulating to find a sequence \( R_2 \) gives

\[ R_2(K) = \sum_{k=1}^{K} R_1 \quad (3.14) \]

\[ = \gamma S_2(K) \quad (3.15) \]

and likewise for higher order sequences. A multiplication factor has no effect on the relative values of the terms in a set of sequences.

Hence, values of a polynomial function of the form \( S_N \) at natural number arguments can be constructed by sequential accumulation without any multiplication, by first calculating all sequences of lower order than \( N \). Further, any polynomial function can be written as a linear combination of the \( S_N \); Equation 3.3 is an example of the necessary invertible transformation between coefficients.

The advantage of the accumulation method becomes apparent when the FTWs must be calculated in order. The natural number \( K \), the number of steps in the series of computations, is reached at time \( t_K \), which is \( K \) times the period of the upload sequence. Six registers store values which, at \( t_K \), hold values which are linear combinations of the terms \( S_1(K) \) through \( S_5(K) \). Let \( d5FTW_z \), \( d4FTW_z \), \( d3FTW_z \), \( d2FTW_z \), \( dFTW_z \), and \( FTW_z \) be the six registers, with values \( b_5 \), \( b_4 \), \( b_3 \), \( b_2 \), \( b_1 \), and \( FTW_z \) at \( t = t_0 \). When a FTW preparation process is triggered, on five consecutive clock edges the following five computations are executed:

\[ d4FTW_z \leftarrow d4FTW_z + d5FTW_z \]
\[ d3FTW_z \leftarrow d3FTW_z + d4FTW_z \]
\[ d2FTW_z \leftarrow d2FTW_z + d3FTW_z \]
\[ dFTW_z \leftarrow dFTW_z + d2FTW_z \]
\[ FTW_z \leftarrow FTW_z + dFTW_z. \]
As a result, their values of the numbers stored by the registers are
\[
\begin{align*}
d5FTW_z(t_K) &= b_5 \\
d4FTW_z(t_K) &= b_4 + b_5 S_1(K) \\
d3FTW_z(t_K) &= b_3 + b_4 S_1(K) + b_3 S_2(K) \\
d2FTW_z(t_K) &= b_2 + b_3 S_1(K) + b_4 S_2(K) + b_5 S_3(K) \\
dFTW_z(t_K) &= b_1 + b_2 S_1(K) + b_3 S_2(K) + b_4 S_3(K) + b_5 S_4(K) \\
FTW_z(t_K) &= FTW_{z,0} + b_1 S_1 + b_2 S_2(K) + b_3 S_3(K) + b_4 S_4(K) + b_5 S_5(K).
\end{align*}
\]

By making the substitution of Equation 3.13 for each term, we see \(FTW_z(t_K)\) has exactly the form of Equation 3.4. Thus the \(z\)-channel output value is computed in five summations.

In the current application, movement of ultracold clouds along a stationary guide beam, the \(x\)-frequency was a linear function of the \(z\)-frequency,
\[
f_x = \alpha \times \Delta f_z + f_{x,0},
\]
where \(\Delta f_z\) is the deviation of the \(z\)-frequency from its initial value \(f_{z,0}\). The initial \(x\)-channel frequency, \(f_{x,0}\), held no relationship to \(f_{z,0}\), it was purely determined by the positioning of the two-axis AOD. But the change in \(f_x\) from this value, \(\Delta f_x\), was proportional to the change in \(f_z\) from its initial value, with \(\alpha\) the proportionality constant. It was more hardware efficient to take the computed value for \(dFTW_z\) and multiply it by \(\alpha\) to find \(dFTW_x\) than to accumulate this value independently. A sequential binary multiplier was constructed, taking a 15-bit \(\alpha\) and multiplying \(dFTW_z\) by it over 15 clock edges.

Finally, in some programs multiple pairs of FTW registers existed. The FTW upload cycle, which ran continually, would select which FTW to load from without affecting the contents of the registers themselves. During a trap movement sequence all FTW registers would accumulate simultaneously, regardless of which would be loaded to the DDS during the upload cycle. In all applications discussed here, clouds were moved symmetrically or in parallel, and thus only one pair of \((dFTW_x, dFTW_z)\) values were needed. The values would be added to the rising FTW and subtracted from the falling one.

Some programs were designed to toggle among four trap sites, so four pairs of FTW registers coexisted. As Figure 3.4 illustrates, clouds were divided with two phases of binary splitting: the initial cloud into two, then each of those into two more. In these programs, the first two FTW register pairs, \((FTW_{0_x}, FTW_{0_z})\) and \((FTW_{1_x}, FTW_{1_z})\) were accumulated identically during the first movement phase, and so too were \((FTW_{2_x}, FTW_{2_z})\) and \((FTW_{3_x}, FTW_{3_z})\):
\[
\begin{align*}
(FTW_{0_x}, FTW_{0_z}) \leftrightarrow (FTW_{0_x}, FTW_{0_z}) &- (\alpha dFTW_z, dFTW_z) \\
(FTW_{1_x}, FTW_{1_z}) \leftrightarrow (FTW_{1_x}, FTW_{1_z}) &- (\alpha dFTW_z, dFTW_z) \\
(FTW_{2_x}, FTW_{2_z}) \leftrightarrow (FTW_{2_x}, FTW_{2_z}) + (\alpha dFTW_z, dFTW_z) \\
(FTW_{3_x}, FTW_{3_z}) \leftrightarrow (FTW_{3_x}, FTW_{3_z}) + (\alpha dFTW_z, dFTW_z).
\end{align*}
\]

During the second phase, registers \((FTW_{0_x}, FTW_{0_z})\) and \((FTW_{2_x}, FTW_{2_z})\) accumulated identically, exactly symmetric to \((FTW_{1_x}, FTW_{1_z})\) and \((FTW_{3_x}, FTW_{3_z})\). The same accumulation process ran with the same initial values \((b_1, b_2, b_3, b_4, b_5)\), but at the final
summation the bits of the registers $dFTW_x$ and $dFTW_z$ were shifted down by one, halving
the numeric value of the number:

\[
\begin{align*}
(FTW_{0x}, FTW_{0z}) &\leftarrow (FTW_{0x}, FTW_{0z}) - (\alpha dFTW_z, dFTW_z)/2 \\
(FTW_{1x}, FTW_{1z}) &\leftarrow (FTW_{1x}, FTW_{1z}) + (\alpha dFTW_z, dFTW_z)/2 \\
(FTW_{2x}, FTW_{2z}) &\leftarrow (FTW_{2x}, FTW_{2z}) - (\alpha dFTW_z, dFTW_z)/2 \\
(FTW_{3x}, FTW_{3z}) &\leftarrow (FTW_{3x}, FTW_{3z}) + (\alpha dFTW_z, dFTW_z)/2.
\end{align*}
\]

Thus, at any time only one pair of $(dFTW_x, dFTW_z)$ values was needed.

Last, piecewise-defined FTW trajectories were easy to bring about with this method, as long as each piece of the curve was described by a quintic polynomial function. Each phase of the movement required a different set of $b_i$ coefficients. At the time of changeover, the last accumulation computation with the old coefficients would run, then the new coefficients would immediately replace the contents of the $dFTW$, $d2FTW$, $d3FTW$, $d4FTW$, and $d5FTW$ registers. On the following upload cycle, the accumulation would be carried out with the new coefficients. In this way, functions like the one in Figure 3.5 could be realised. In calculating the coefficients, before they were passed to the FPGA, each piece of the curve had to be expressed in terms of $(t - \tau_{\text{change}})$, where $\tau_{\text{change}}$ was the changeover time, to get the right values. When an FPGA program was to implement a motion of this sort, it would load all coefficient values from memory before it was triggered, and store them in logic until they were called.

---

Figure 3.4: The $z$-frequency trajectories of the four trap sites during a split sequence. Each FTW register was accumulated through the sequence. Notice the orange and green curves are mirror images, the yellow curves are identical (apart from initial conditions), and the purple curves, too, are identical. The dark blue segments represent short wait phases of equal length, for all FTW registers.
Figure 3.5: A simple piecewise-defined trajectory consisting of a quadratic acceleration phase, a period of movement at constant velocity, and a quadratic deceleration phase. The FTW accumulation during each third of the movement is fixed by different sets of coefficients. After the movement, the FTW is held static by setting all values of \( d_{\text{FTW}} \) through \( d_{5\text{FTW}} \) to zero, while allowing the computations to continue.

### 3.6 Amplitude Control

As there were two output channels from the DDS, there were two binary amplitude scale factors to define. Both were ten bits, so could take on integer values from 0 to 1023. Because the outputs were used to deflect the same beam, the amplitudes of both signals were associated with the final optical power of the vertical tweezer beam, and so the diffraction efficiency of the AOD unit could be controlled with either.

In the end, amplitude control of both channels was employed, for two different purposes. The \( x \)-channel ASF was able to be controlled by the user: its initial value could be set, and a series of linear ramps could be executed. The \( z \)-channel was intended to correct for the dependence of the diffraction efficiency on the driving frequencies. Figure 3.6 gives the diffraction efficiency of the dual-axis AOD over a range of \( z \)-frequencies from 63.97 MHz to 86.91 MHz. The variation in the diffraction efficiency, between 54\% and 73\% over 63.97 MHz to 86.91 MHz, is significant. At frequencies lower than 63.97 MHz the beam was highly attenuated, probably because it became physically blocked. To counter this, the frequencies over this range were split into 48 bins\(^1\) of size 0.4883 MHz. For each bin, the \( z \)-axis AOD was driven at the central FTW value of that bin and the \( z \)-channel ASF changed until the diffraction efficiency reached a target value, 53.9\%. The ASFs are plotted in Figure 3.7. Through this process, the \( x \)-channel frequency was held constant at 72.0 MHz at an ASF of 1023. As a rule, the \( z \)-channel frequency typically varied over tens of MHz during the movement along the waveguide, the \( x \)-channel would remain within 0.35 MHz of its initial frequency, so no attempt to correct for variations

---

\(^1\)each of size \( 2^{22} \) in units of the FTW
Figure 3.6: Diffraction efficiency vs $z$-channel frequency, for frequencies over the range 62.99 MHz to 86.91 MHz. Both $x$-channel and $z$-channel ASFs were set to their maximum values, 1023, and the $x$-channel frequency was $f_x = 72.0$ MHz.

Figure 3.7: The ASF that gave a diffraction efficiency of 53.9%, for 48 bins with $z$-frequencies between 63.97 MHz to 86.91 MHz.

due to deviation in this frequency was made. Once these ASFs had been found, they were programmed into an LUT in the VHDL code. Each upload cycle, after the FTWs were computed, the $z$-channel ASF was looked up based on bits 27 to 22 of the $z$-channel
FTW - examples are given in Table 3.1. This process took place in time for the ASF to be ready for upload.

<table>
<thead>
<tr>
<th>Bin no.</th>
<th>$z$-Frequency Tuning Word, bits 27 to 22</th>
<th>Frequencies (MHz)</th>
<th>Resulting $z$-Amplitude Scale Factor (from Figure 3.7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>000011</td>
<td>63.96 - 64.45</td>
<td>1023</td>
</tr>
<tr>
<td>4</td>
<td>000100</td>
<td>64.45 - 64.94</td>
<td>1014</td>
</tr>
<tr>
<td>5</td>
<td>000101</td>
<td>64.94 - 65.43</td>
<td>1006</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>001010</td>
<td>67.38 - 67.87</td>
<td>931</td>
</tr>
<tr>
<td>11</td>
<td>001011</td>
<td>67.87 - 68.35</td>
<td>910</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 3.1: Example entries of the $z$-channel ASF look up table. Note that for the FTW, bit 31 is the most significant bit and bit 0 is the least significant bit.

As for the $x$-channel ASF, it was held static or linearly ramped. An initial $x$-ASF was fixed, and after that any changes to the ASF were defined with three values: the time at which an ASF ramp would begin after the sequence trigger, the gradient at which it ran at, and the final ASF. During a ramp, the ASF would increment in steps of 1, with a minimum step time of 1 upload cycle (usually chosen to be 1 $\mu$s). The gradient was determined by the number of upload cycles the FPGA would wait between incrementation events - if it waited for 1000 cycles between steps, each of of 1 $\mu$s, the ASF gradient would be $1 \text{ ms}^{-1}$. While this method restricted an ASF ramp to a minimum time, the ASF could be set to ramp its entire range in 1.023 ms, which was deemed to be sufficiently fast.

Up to four amplitude ramps could be programmed. (This number could easily have been increased with extra programming, but it proved to be more than enough for the uses explored in this thesis.) The ramps could only be triggered sequentially: if the second ramp was set up to begin at an earlier time than the first, it would not register at all. If this was the case then the DDS would not perform the third or fourth ramps either, as they had to come after the second. However, if the second ramp was programmed to begin after the first had started but before it had ended, the ASF register value would jump to the final value of the first ramp. From there, the second ramp would progress at its set gradient to its set final value. It was possible to make the amplitude jump discontinuously using this process. A slow ramp could be programmed to some desired final value, but it could be interrupted immediately by a ramp of one step. The ASF would jump to the final value of the first ramp when the second began, then change by only one ASF point, effectively jumping in amplitude.

### 3.7 Summary of FTW Behaviours

Using the FTW loading method of movement, three main classes of program were prepared for use on the atoms, although a number of versions of each did exist: programs that would split into two and hold, programs that would split into four and evaporatively cool, and programs that moved a cloud of atoms in one direction along the waveguide. This section gives an overview of each.
3.7.1 Two Cloud Split and Hold

Some of the experimental work contained in this thesis involved this type of program. Atoms would be loaded into a time averaged double-well trap whose minima were separated by 40 $\mu$m (0.2 MHz), sketched in Figure 3.8. When triggered, the DDS would split the cloud by symmetrically raising and lowering the pairs of FTWs that mapped to a trap site. While the tweezer movements could take the form of any quintic function, usually a smooth motion that minimised the net-jerk-squared through the movement was used. (The functional form of this motion is given in Section 5.3.1 and derived in [25].) After the movement, the FTWs would remain constant until reset.

This class of program was used fairly early on, and did not use the LUT-based frequency-dependent $z$-amplitude described in Section 3.6. Also, data was loaded to the DDS in a slightly different way to the heavily used four-cloud programs: instead of loading two FTWs and two ASFs on every upload cycle, each upload cycle lasted long enough to upload two FTWs only. As a result, the upload cycle would take only 33 clock edges to run, giving 660 ns time resolution. The $z$-FTW was always uploaded, but the data uploaded during the other time alternated between the $x$-ASF, the $z$-ASF, and the $x$-FTW. The two ASFs would be uploaded when the frequency toggled to address another trap site, then the $x$-FTW was uploaded until the end of the current hop period. It was thought this scheme would be sufficient because the ASFs and $x$-FTW changed slowly relative to the $z$-FTW. Later, it was discarded because the FPGA program became somewhat opaque, and because a 1 $\mu$s load time had been found to be sufficiently fast.

3.7.2 Four Cloud Evaporation

Four cloud programs were also created as natural extensions of the two-cloud split projects, except four pairs of FTW registers existed and were loaded from sequentially. For the atoms, the initial split phase was identical. It was followed by a “wait” phase of variable length, usually chosen to be about 1 ms. Then the two clouds would be split into four, moving each sub-cloud by half the distance in the same time as the first split movement as described in Section 3.5.

Most four cloud programs were intended to evaporatively cool the clouds after splitting. Additional movements were included with this in mind: a “shuffle” in $x$-frequency with the $z$-frequency held constant, shown in Figure 3.9. This would displace the V-beam by a small amount, distorting the trapping potential from that of an optimal crossed-beam trap. In particular, with the clouds displaced from the axis of the H-beam, the trapping

Figure 3.8: The general process carried out by a two cloud program: beginning from a double-well trap, the tweezer trap sites would move symmetrically apart and come to rest after a fixed time, then be held there until the program was reset at the end of the imaging sequence.
CHAPTER 3. HARDWARE AND SOFTWARE

Figure 3.9: A four cloud project would halve a cloud, then divide again to get four clouds. These projects had the capability to shift the \( x \)-FTWs after the split sequence, and would shift the tweezer beams back after a programmed wait time.

Figure 3.10: One cloud projects would only shift a single tweezer in one direction. Potential in the vertical direction would be weakened, allowing hot atoms to escape under gravity along the axis of the V-beam. The shuffle phase was followed by a wait phase, then a symmetric shuffle in \( x \)-frequency to bring the atoms back. All clouds were shuffled in the same direction. Finally, these motions could also take the form of a general quintic polynomial, implemented with the same accumulation method described above. This time, however, the computed \( FTW_z \) values were added directly to the \( x \)-FTWs without passing though the multiplier, which allowed a movement to be carried out when the multiplier was set to zero.

3.7.3 Unidirectional Cloud Movement

The third class of program was one that would simply shift a single cloud along the waveguide, as Figure 3.10 depicts. In these programs, only one FTW register existed, giving movement without splitting or any sort of FTW toggling. They programs were used to compare different motion profiles: for example comparing a constant velocity motion to one with trapezoidal acceleration. The profiles are defined in Section 5.3.

Several programs of this sort existed because some of the motion profiles were piecewise defined. This required the FPGA to shift between several consecutive movement phases, using new coefficients at the beginning of each one before continuing with the accumulation process. Programs existed that had two, three, and five consecutive motion phases. In all cases, all sets of coefficients defining the motion were loaded from memory immediately after a memory write. The values were stored in logic until they were called at the beginning of a phase of motion. These projects did not attempt to correct for the variation in diffraction efficiency with \( z \)-frequency.
Chapter 4

Deployment

This chapter documents early use of the FPGA/DDS system. First, control of the amplitude and frequency of the DDS output is demonstrated, and found to be in accordance with the expected behaviour. Following are some results from early use of the system with the optical tweezer unit. Data collected during the calibration of the $x$-channel frequency is presented. Finally, an ultracold cloud is transferred to the tweezer trap and split into two, and the sub-clouds are excited to a visible breathing motion by tightening and relaxing both traps simultaneously.

4.1 DDS Outputs

When under control of the FPGA, the outputs of the DDS were simultaneously observed on an oscilloscope. This section presents some demonstrations the system was behaving as it was intended to.

Figure 4.1 shows the Channel 0 (upper) and Channel 1 (lower) outputs of the DDS as it toggles between two pairs of frequency tuning words. For this image, the frequency pairs $(f_0, f_1)$ were chosen to be $(6.3\,\text{MHz}, 3.3\,\text{MHz})$ and $(2.1\,\text{MHz}, 1.2\,\text{MHz})$, giving periods on the order of the hop time of $1\,\mu\text{s}$. The DDS outputs are clearly seen to be phase continuous, not phase coherent. Note that these frequencies were much lower than the frequencies used to drive the AODs, which accepted inputs from 60 MHz to 90 MHz.

A frequency split, used to separate one cloud into two daughter clouds, is demonstrated in Figure 4.2. Starting from frequencies 75.23 MHz and 75.43 MHz, which when driving the optical tweezer beam would create a double-well potential with minima $40\,\mu\text{m}$ apart, the lower and upper $z$-channel frequencies were lowered and raised by 7.5 MHz in 150 ms in a smooth quintic polynomial trajectory (a Minimum-Jerk movement, described in Section 5.3.1 and [25]). This would correspond to a movement of 1.5 mm for each optical trap. Data was collected by only allowing the computation to step by 5000 calculation cycles at a time; with the upload time set to $1.2\,\mu\text{s}$, this meant a time evolution of 6 ms. After this short period, the accumulation process would halt and the FPGA would continue to upload unchanging FTWs. A button press would trigger the continuation of the computation sequence, which would halt after another 5000 cycles. In this way, the frequency would remain fixed for as long as it took to make a measurement of it. Measurements
Figure 4.1: Oscilloscope traces of the toggled outputs of Channel 0 (upper, blue) and Channel 1 (lower, green). The frequency pairs \((f_0, f_1)\) toggle between \((6.3\,\text{MHz}, 3.3\,\text{MHz})\) and \((2.1\,\text{MHz}, 1.2\,\text{MHz})\), giving oscillation periods comparable to the hop time. Notice the phase continuity.

were made with an Agilent spectrum analyser\(^1\), and the upper and lower frequencies were measured on separate runs. In parallel, the FPGA would write the contents of its two FTW registers to the peripheral LCD screen, allowing them to be read back as binary numbers during the same periods when the computations had halted.

As described in Section 3.6, two types of amplitude control were used: one user-defined, the other preprogrammed. A demonstration of user-defined control of the \(x\)-channel amplitude is presented in Figure 4.3. In this somewhat artificial example, the \(x\)-channel amplitude scale factor was ramped from an initial value of 90 to the maximum value of 1023 in 1.86 ms, then down to 300 in 2.17 ms, then jumped to 900 immediately afterwards. When this ASF control was used on the atoms, it was either to increase the optical power of the tweezer beam as it split into two (which would halve the trapping potential of each beam unless the RF power was increased), to weaken the confinement along the \(z\)-axis, to excite breathing (Section 4.3), or to evaporatively cool the clouds (Section 5.1).

It was expected there would be some latency between a reprogramming of a set of FTWs and ASFs and the resultant change in the DDS outputs. The time delay was measured by asserting an I/O pin on the FPGA simultaneously with the “Update” signal input to the DDS, and recording the DDS outputs over the subsequent time. The delay between the assertion of the update signal and the change in frequency was measured to be very small: 72 ns, as Figure 4.4 shows. This is considerably faster than the response time of the AOD, measured to be 2 \(\mu\)s [6]. It should be noted a much greater latency, greater than the hop time of the toggled signal, would not necessarily have been detrimental: as long as the computations were pipelined, such that the the output still changed after every hop period, the changing FTW sequence could have been triggered in advance to mitigate any time delay, and the toggled output would have been identical. But prompt

\(^1\)Agilent E4405B
Figure 4.2: Tracking of the lower (blue) and upper (green) $z$-channel frequency and FTW through a 1.5 mm, 150 ms split sequence. Points are measured frequencies, from a spectrum analyser, or FTWs read from the FPGA’s LCD display. Note the $x$-axis is counted in units of the upload time, $1.2 \mu s$, and points were measured in jumps of 6 ms, $5000 \times$ this time interval.

Figure 4.3: A demonstration of amplitude control of the $x$-channel ASF. The frequency was set to 15 kHz, a very low value compared to the normal operating range of the output.
Figure 4.4: A screenshot of the readings on three channels of an oscilloscope. Channel 1 (navy blue) is the $x$-frequency output, Channel 2 (cyan) is the $z$-frequency output, and Channel 3 (purple) is read from I/O pin C10 of the FPGA. Pin C10 rose and fell at the same times as the “Update” signal input to the DDS, and thus the time between the rising edge of pin C10 and the change in frequency outputs is the latency of the process. The $(f_x, f_z)$ frequency pairs used here, $(45\text{ MHz}, 40\text{ MHz})$ and $(1.5\text{ MHz}, 5\text{ MHz})$, were chosen to contrast in making this measurement and were outside the range of the AOD.

Changes in frequency eliminated any need for such a measure.

### 4.2 Multiplier Calibration

As described in Section 2.2, the atoms were restricted to movement along a fixed waveguide. This meant cloud movements could have been performed with a single AOD. Such a system has been implemented by our group [5], but the difficulty in achieving adequate mechanical alignment was found to be a limiting factor: during a long range movement from the point of loading, the vertical beam would become misaligned and drag atoms from the horizontal guide beam.

The dual-axis AOD was a key component in extending the range of device, enabling active correction for any misalignment of the two beams throughout a movement. Due to the linearity of the waveguide, the optimal $x$-frequency trajectory could be described as a linear function of the $z$-frequency (Equation 3.16), a fact which was exploited in the computation of the FTW pairs.
Figure 4.5: Above: clouds 3 mm apart and well aligned with the waveguide. Below: severe spilling into the waveguide from clouds 64.7 µm from their optimal positions. The two pairs of clouds differed only in the values of their $x$-FTW multipliers. Both images were taken after a 6 ms time of flight.

In Figure 4.5 clouds well aligned after a 1.5 mm movement in each direction are compared to clouds that were well aligned at the central loading point but became misaligned through the same movement. This misalignment arose from a non-optimal $\alpha$ multiplier in Equation 3.16. At their final positions, they are displaced by 64.7 µm from the positions of the upper pair of clouds. The trajectories differed by 2.47° in the $xz$-plane.

A good alignment with the waveguide at all times, exemplified in the upper image of Figure 4.5, was maintained by optimising the central $x$-frequency $f_{x0}$ and multiplication factor $\alpha$ in Equation 3.16. In both alignment procedures the atom number was optimised, at the centre of the science cell and after movements to the edges of the region visible to the imaging apparatus. In all experimental runs, the values for $f_{x0}$ and $\alpha$ were communicated to the FPGA in the usual way via memory. The data taken during the optimisation has been published in [27].

First, the optimal initial $x$-frequency was found. Atoms were transferred from the IP trap into a double well optical potential, created by toggling the $z$-frequency between 75.1 MHz and 75.3 MHz with a hop time of 6.6 µs, with the $x$-frequency fixed. The minima were spaced by 40 µm, while the $1/e^2$ V-beam radius was 60 µm. After a 40 ms load time, the magnetic trap was turned off in 10 ms. The atoms, confined only in the crossed-beam dipole trap, were held for 100 ms. During this time, a better aligned potential would confine the atoms more strongly. The atoms were then released and imaged after a 12.5 ms time of flight, and the cloud population was inferred. The experiment was repeated for a 72.25 - 72.55 MHz range of $x$-frequencies, mapping to positions extending over 60 µm, half the $1/e^2$ diameter of the H-beam. The measured atom numbers are plotted in the central section of Figure 4.6.

The frequency that gave the highest atom number was found to be 72.39 MHz. Using this value, atoms were loaded into the double-well trap, then the cloud split and each sub-cloud moved by 1.8 mm along the horizontal beam. Among shots the $\alpha$ multiplier value ranged from -0.03517 to 0.01703. All movements were made in 150 ms with a trajectory...
that minimised the integral of the jerk squared over the time of the motion (see [25] or Section 5.3.1). The atom numbers were measured after a further 100 ms hold at their final positions. The advantage of being able to split the cloud to make the measurements is immediately clear: by probing two sites simultaneously, the number of experimental shots required is halved. Figure 4.6 plots the number measurements at ±1.8 mm.

After the alignment process, the optimised values were confirmed by measuring the lifetime of the clouds at several points along the waveguide. Lifetimes were measured after movements with the optimal multiplier value and after movements during which the x-frequency was held fixed at 72.39 MHz. Here, too, two sites were simultaneously probed by dividing the initial cloud, saving experiment runtime. At each site, atom numbers were measured after five hold times that increased in 200 ms intervals, beginning with 100 ms. All time series were fitted with exponential decay curves and the lifetimes from the fitting are plotted in Figure 4.7.

With a non-zero multiplier value, the lifetime of the cloud was seen to increase at all sites along the waveguide. The lifetime increase after movements greater than 0.3 mm was expected, and significant. Finally, it is noted that this data was collected three months before that in Sections 4.3 and Chapter 5 and over this time it was natural a slow drift would occur. The initial x- and z-frequencies and the α multiplier were reoptimised before these data sets were taken, and as such the values used through the rest of the thesis are slightly different.
4.3 Induced Breathing

As a demonstration of the ability to control the amplitude of the DDS outputs, a cloud was split into two sub-clouds. The samples were spatially separated and the amplitude of the vertical trapping beam was sharply raised in amplitude, then returned to its initial value shortly afterwards. This tightening and relaxing of the trap excited the thermal cloud into oscillations at double the trapping frequency - “breathing” [45]. Measurements of the width of the clouds along the axis of the horizontal beam allowed the trapping frequency in this direction to be determined experimentally.

Clouds were moved by a four cloud split program, which maintained a constant diffraction efficiency through the motion with the LUT based $z$-ASF amplitude control described in Section 3.6. The wait time after the first split phase was chosen to be long and all cloud images were able to be gathered before the second split phase began. From the point of loading, the clouds were each moved by 1.5 mm in 150 ms with a minimum-jerk-squared trajectory, similar to the one used for the multiplier calibration. During this movement, the amplitude of the FPGA-controlled vertical beam was held constant with an $x$-channel ASF of 490. Immediately afterwards, the ASF was linearly ramped to 1023 over 0.5 ms. After a wait of 4.5 ms, the ASF was dropped back to its initial value in another 0.5 ms linear ramp. The sequence was run 13 times, with an image taken 0.5 ms later on each run. All images were taken at 5 ms time of flight.

For each image, cloud widths were obtained by fitting a Gaussian function to the absorption profile. The $z$-widths are plotted in Figure 4.8, along with the first five images of the
series. Clear oscillations of the width are observed, easily visible in the images themselves. The oscillation frequencies were measured to be \( w_{\text{left}} = 2\pi \times 298 \text{ Hz} \) and \( w_{\text{right}} = 2\pi \times 302 \text{ Hz} \), implying trapping frequencies of \( w_{\text{left}} = 2\pi \times 149 \text{ Hz} \) and \( w_{\text{right}} = 2\pi \times 151 \text{ Hz} \). A calculation of the optical power, taking into account the sharing of power between the trap sites, suggests the trapping frequencies should both be somewhat higher: \( 2\pi \times 233 \text{ Hz} \). This calculation naively halved the total optical power in the beam. In the AOD, the access time, the time an acoustic wave took to traverse the aperture of the crystal, was \( 2 \mu s \) [6]. As the two-cloud hop time used in this experiment was \( 4.8 \mu s \), on the order of the access time, there may have been a substantial loss of optical power into non-trapping beams as depicted in Figure 2.4(a).

The breathing measurements were taken to be a calibration of the optical power. Knowing the beam waist to be \( 45 \mu m \), the time-averaged optical powers at both effective trap site were calculated using Equations A.4 and A.7 in Appendix A. It was found to be 116 mW for the left well and 119 mW for the right. The similarity of the numbers confirms the LUT based z-channel amplitude control was correcting for variation in diffraction efficiency, at least at these two sites.

![Figure 4.8: Observation of breathing in two clouds separated by 3 mm. a) First five images of a sequence of 13, showing the first half cycle of an oscillation. b) Measured cloud widths of left (blue circles) and right (green squares) clouds, after being excited to a breathing motion by raising and lowering the V-beam optical power. The lines are exponentially decaying sinusoidal fits to the data.](image)
Chapter 5

Application

This chapter contains experimental results that demonstrate application of the FPGA/DDS frequency source. After the early calibration and testing of the device, it was used to passively evaporate independent thermal clouds by slowly reducing the optical trapping potentials. Next, an alternate method of cooling the clouds was investigated: slowly deforming the crossed-beam trap by moving the vertical beam in the $x$ direction. Last, in an exploration of the smoothness of different "adiabatic" movements, a single cloud was translated with different motion profiles. The amplitudes of the in-trap sloshing after the motions are compared.

5.1 Four Cloud Condensation

In quantum engineering applications, arrays of independent Bose-Einstein condensates may become an important tool. Arrays of independent BECs have been prepared by several groups, usually by RF-induced evaporative cooling [46, 47] in wide magnetic traps with imposed repulsive optical barriers. In [46] and [47], the lack of independently controllable traps limited use of the condensates to merging and studies of interference. Optical traps, in contrast, offer supreme versatility, and ultracold atoms have been confined in highly non-trivial potentials [48, 32, 31, 49] which hold great promise for engineering. Further, all-optical evaporation has become a prominent method used to prepare Bose-Einstein condensates. Experiments in 2001 [50] saw the first samples brought through the BEC transition without magnetic influence, although magnetic trapping played a prior role in the experiment. There, a sample of ultracold $^{87}\text{Rb}$ was confined in a crossed-beam dipole trap and the potential depth lowered non-linearly over about 2 s. It was essential the depth was lowered slowly, such that the rethermalisation rate remained faster than that of atom loss. Since, optical evaporation has been extended with a “tilted trap” method that preserves the local trapping potential parameters while lowering the depth, maintaining a high collision rate and giving a faster evaporation [51, 52].

We set out to condense four distinct clouds by the method used in [50]. The hunt for the BEC transition involved an extensive search through a large space of parameters. To begin with, a cloud was evaporatively cooled in the IP trap then transferred to a double-well$^1$ crossed-beam dipole trap, with a horizontal beam power of 1.45 W and a

$^140 \mu m$ separation
total vertical beam power\(^2\) of 0.235 W, in 45 ms plus a 10 ms IP trap turn-off time. The sample was split into two by moving the two wells in a Minimum-Jerk profile over 0.5 mm in 150 ms, a very slow and smooth ramp on an atomic timescale. Beginning 10 ms before the end of this movement, the optical power in each well was raised to 0.315 W by a 30 ms linear ramp in \(x\)-channel ASF to its maximum value (1023) in preparation for the split into four wells.

After a 1 ms wait time, each well was simultaneously split further into two wells which moved by 0.25 mm in 150 ms. The result was a linear array of four wells, each 0.5 mm from its neighbours. Note that throughout this process, FTWs were loaded sequentially from four registers which addressed the far left, second left, second right, and far right wells, as described in Section 3.5. As a result, the effective hop time halved during the second split, while the total toggle period remained constant.

After a further 1 ms hold time, the evaporation process began. The steps described here were determined experimentally, by optimising each one individually. The clouds were shifted by 5 µm in the \(x\) direction, with a Minimum-Jerk profile over 50 µs. This movement was intended to deform the trapping potential slightly to encourage loss of the hot atoms vertically along the V-beam axis, where they would fall under gravity. However, the effectiveness of this procedure was hardly explored. Beginning at the same time, the H-beam optical power was lowered from RebeKa by reducing the input to the voltage variable attenuator controlling the amplitude of the signal driving the H-beam AOM\(^3\). The optical power in the H-beam over time is plotted in Figure 5.1.

\(^2\)\(x\)-channel ASF of 490, for a single well
\(^3\)"Collider attenuation 1" lowering sequence: 7.8 V \(\rightarrow\) 6.5 V in 160 ms, 6.5 V \(\rightarrow\) 6.2 V in 170 ms, 6.2 V \(\rightarrow\) 6.1 V in 100 ms, 6.1 V \(\rightarrow\) 6.0 V in 300 ms, 6.0 V \(\rightarrow\) 5.6 V in 2s

Figure 5.1: The optical power in the H-beam during the optimised evaporation sequence. The attenuation voltage was ramped linearly for the first 160 ms, which gives a kink as the relationship between voltage and diffraction efficiency is non-linear at high voltages.
An example of the optimisation of one parameter is given here: the weakening of the V-beam trapping potential. Under the conditions described above, the V-beam amplitude was lowered by a linear ramp in $x$-channel ASF. This ramp, which always began at its maximum value of 1023, was triggered 330 ms into the evaporation sequence, about where the H-beam lowering started to shallow off. Every ramp took 1400 ms and ran at a different gradient, to arrive at a different final ASF between 1022 (virtually no decrease) and 200 (6.9% of the initial optical power). The atom numbers and phase space densities at the end of the ramps are plotted in Figure 5.3(a) & (b). At low ASFs the spilling from the trap sites into the horizontal waveguide became significant. To give some comparison of the extent of the spilling, the optical density was summed in five regions peripheral to the trap sites that were still contained within the horizontal beam. Figure 5.2 is an image showing the regions. The fraction of the total OD found in these five regions of Figure is plotted in Figure 5.3(c), and indicates that at final ASF values lower than about 350 the clouds suffer a detrimental amount of spilling.

Using the $x$-channel ramp timings described above, and a final ASF value of 380, condensation of all four clouds was observed. Figure 5.4 shows a series of images taken at later times on each experimental run. In the last image, the condensate fractions of the four clouds are 70%, 56%, 63%, and 62% from left to right, with condensate populations of $\sim$34000, 22000, 33000, and 25000 atoms. Of interest is early emergence of a significant condensate fraction in the second right well. The conditions at this point must have been particularly favourable for BEC formation.

Figure 5.2: Image of two experimental runs used in the $x$-channel ASF optimisation data, showing four clouds spaced over a total distance of 1.5 mm and the five regions used to measure atom spilling into the waveguide. Notice the spilling at an ASF of 200 (optical power of 0.14 W) compared to an ASF of 400 (optical power of 0.53 W). These images were taken at a 10 ms time of flight.
Figure 5.3: Plots of the (a) atom number and (b) peak phase space density of the four clouds for several different cooling sequences with different ramps in V-beam amplitude. The fraction of the optical density found in the five regions of Figure 5.2 beyond the trap sites is plotted in (c), and is seen to increase sharply when the final ASF value is below 400.
Figure 5.4: Time series of condensation formation in four clouds. Plots on the right show the integrated OD in the vertical direction, all with the same scale. All images are taken after a 21 ms time of flight.
5.2 Cooling by Trap Deformation

To extend on the work on optical evaporation, measurements were made on four clouds as the vertical beam was slowly displaced in the horizontal direction, distorting the crossed-beam trap. Such a method might be able to be used in a cooling technique similar to the tilted trap of [51] or [52], in which the trapping frequency was maintained while trap depth was lowered with external magnetic or optical fields. With an intentional misalignment, the trapping potentials are able to be weakened without reducing the optical powers, which are left as free variables. Notably, with a sufficiently strong V-beam the hottest atoms should be able to be lost from the system completely, by falling under gravity along the axis of this beam rather than remaining in the horizontal waveguide. Here, misalignments of tens of microns were seen to cool the clouds significantly.

In this series of experiments, an atomic cloud was split into four in the same movements described in Section 5.1: by two 150 ms movements with a 1 ms wait in-between. This time, the H-beam had an optical power of 0.843 W. The V-beam power began at a value of 0.235 W and was raised in the same way as in the passive evaporation experiments, to 0.315 W in each of two wells with a 30 ms linear ramp in ASF beginning 10 ms before the end of the first motion. This time, the optical powers were held constant in both tweezer beams while the clouds were slowly displaced in the horizontal direction from their initial positions by raising the \( x \)-frequency. This “shuffle” movement had a minimum jerk profile, like the splitting sequence. It began 1 ms after the end of the second split phase.

All shuffle movements were chosen to take 2 s, with several different distances investigated. Because the \( 1/e^2 \) radius of the stationary horizontal beam was 60 \( \mu m \), distances were on the order of 100 \( \mu m \) or less. Images were taken at 250 ms intervals, beginning 750 ms into the movement. For each interval, two images were recorded on different experimental runs, at 6 ms and 12 ms times of flight. The temperatures of the clouds were determined by their expansion over the intervening time.

The clouds at the ends of the array, at \( \pm 0.75 \text{ mm} \), are analysed here. It was suspected the clouds exchanged hot particles along the waveguide, which would have interfered with the temperature of the samples particularly at the inner two trap sites. While the two inner clouds were seen to lose atoms, their temperatures did not seem to follow a definite trend until the latter stages of the process, at which point they would cool. For the two outer clouds, this particle exchange was reduced. The atom number and optical density are plotted in Figure 5.5, and their temperatures in Figure 5.6. Temperatures were measured from the 6 ms time of flight expansion in two spatial dimensions. Temperatures along the \( y \) and \( z \) axes are recorded here, as it was uncertain whether the distortion would affect the atom momenta in one dimension to a greater extent than the other. It is noted that a plot of the phase space densities would have been more meaningful, but the calculation was not straightforward. The horizontal trapping frequency was no longer known after the trap had been distorted. These values could be determined by careful analytic or numerical consideration of the combined optical potential.

Also plotted in Figure 5.5 is the fraction of optical density measured in the waveguide on either side of the array, called the “Measure of loss into waveguide”. From the images

---

4“Collider Attenuation 1” set to 6.3V
5\( x \)-channel ASF of 490
6rising to the maximum ASF, 1023
at 6 ms time of flight, the total optical densities existing in a small region beyond each end of the array were computed. The regions were much like those seen at the extremes of Figure 5.2, but translated upwards as the clouds had fallen for a shorter time. They were divided by the optical density integrated over the whole array to obtain the fraction existing in each region. These fractions were taken to be measures of the loss of atoms into the waveguide.

Clear trends are observed: for larger horizontal displacements, more atoms are lost, and the cloud is measured to be at a lower temperature in both the horizontal (z-axis, along the waveguide) and vertical (y-axis) directions. Intuitively, it was unsurprising that an

Figure 5.5: For a range of displacements, the atom number and optical density over 1 s during the latter part of the shuffle, and a measure of the spilling into the waveguide beyond the ends of the array.
increasing displacement caused cooling. Just as when the optical powers were lowered, a gradual pushing of the atoms from the point of greatest trap depth would have an evaporative effect. It was also unsurprising that larger shuffle distances caused greater spilling into the waveguide. As the horizontal displacement grew, the sum of the potentials would become closer and closer to a saddle point. At this point, atom movement into the waveguide and to the V-beam axis would be unimpeded, and at larger displacements the beams would not combine to give a local minimum at all.

For both clouds, the loss rates of atoms and of temperature are graphed in Figure 5.7. The rates were determined by fitting the plots of Figures 5.5 and 5.6 with straight lines. Strictly, the values decrease exponentially [53], but here they were well modelled by straight lines. For both wells, the atom number at $t = 1750$ ms with a 90 $\mu$m displacement was omitted from the fitting process, as it deviated markedly from the otherwise linear progression of those points. As noted above, at this point non-evaporative atom loss was very prominent, causing the massive drop in optical density and atom number. At this time, the well was displaced 88.5 $\mu$m from the optimal aligned position.

On the whole, both loss rates increased with shuffle distance. The ratio of the rates, interpreted as the temperature lost per atom, also increased with larger shuffle distances.
Figure 5.7: For a range of displacements, the atom loss rate (top left), temperature loss rate (top right), and the temperature lost per atom (bottom). In all plots, the far left cloud (-0.75 mm) is plotted as blue diamonds and the far right (+0.75 m) as green circles.

If cooling by deformation is taken to the BEC phase transition, greater deformations should be able to produce larger BECs, as fewer atoms need to be lost to reach the critical temperature. The evaporation method was not pursued to this point, due to time constraints and as this demonstration is sufficient to show the use of the frequency source in such applications. If it ever is, a thorough analysis of the combined trapping potentials should be undertaken. In particular, the movement profile will have a major effect on the efficiency of the process. Here a minimum-jerk profile was used, because it was easy to implement and because the clouds would decelerate over a large distance during the second half of the motion, weakening the trap at a slower and slower rate. However, there will certainly be more efficient trajectories that bring about the cooling effect. As the FPGA/DDS can implement general quintic polynomial movements, there is scope for an extensive investigation.
5.3 Comparison of Different Motion Profiles

As a final demonstration of the capabilities of the tweezer control unit, one cloud of cold $^{87}\text{Rb}$ was moved with several different motion profiles, and over several different times with one particular profile. It was expected that during a motion, a cloud would be excited to slosh inside the tweezer trap. The sloshing - fundamental mode oscillation at the trapping frequency - would then be able to be observed after the trap motion had ceased. The amplitude of the sloshing after a motion was taken to be a measure of the smoothness of the movement.

Measurements of the time evolution of the position of the cloud required the experimental sequence to be run multiple times under the same conditions, with the imaging process triggered at a later time on each run. Necessarily, this meant the data was not a series of measurements of the position of one cloud, but of twenty-one different clouds which had evolved under identical experimental conditions. A data sequence was susceptible to drifts in conditions over the time it was taken, usually about 35 minutes. Efforts were made to remove possible sources of drift: the dipole trap laser was always turned on and experiment run a number of times before a series of images were taken, to warm up the trapping laser and magnetic coils.

All movements were computed by the method described in Section 3.5. Each was either a simple polynomial (Constant velocity, Cubic, or Minimum-Jerk) or a piecewise defined polynomial function (Triangular Velocity, Quintic Stitch, or Trapezoidal Acceleration). The realisation of piecewise-defined polynomial trajectories is described in Section 3.5.

5.3.1 Analytic Expressions for the Six Profiles

Six motion profiles were used in this work: three simple polynomials and three piecewise-defined stitched polynomials. They are briefly described here. All movements are carried out over distance $D$ in time $\tau$, and were from a completely stationary state at $t = 0$. The displacement, velocity, acceleration, and jerk of the profiles are plotted in Figure 5.8.

**Constant Velocity**

A movement at constant velocity is described by

$$d(t) = \frac{D}{\tau} t.$$  \hspace{1cm} (5.1)

When moved at constant velocity, the cloud was subjected to very large instantaneous accelerations at the beginning and end of the movement.

**Triangular Velocity**

This motion profile is piecewise defined: the instantaneous acceleration is

$$a(t) = \begin{cases} 
\frac{4D}{\tau^2} & 0 < t \leq \frac{\tau}{2} \\
-\frac{4D}{\tau^2} & \tau/2 < t \leq \tau
\end{cases}.$$  \hspace{1cm} (5.2)
which gives a displacement of
\[
d(t) = \begin{cases} 
0 < t \leq \frac{\tau}{2} & \left(\frac{2D}{\tau^2} t^2 + \frac{2D}{\tau^2} (t - \frac{\tau}{2})^2 + \frac{D}{2}\right) \\
\frac{\tau}{2} < t \leq \tau & \left(-\frac{2D}{\tau^2} (t - \frac{\tau}{2})^2 + \frac{2D}{\tau^2} (t - \frac{\tau}{2}) + \frac{D}{2}\right)
\end{cases}
\] (5.3)

This profile avoids the large accelerations of a movement at constant velocity, but instead has large instantaneous jerks, at \( t = 0, t = \tau/2, \) and \( t = \tau. \)

**Cubic Displacement**

It was possible to define a displacement profile that was cubic in time. The acceleration would jump to an initial value and change smoothly from there, until another discontinuity at the end:
\[
a(t) = -\frac{12D}{\tau^3} t + \frac{6D}{\tau^2}. \] (5.4)

The displacement \( d \) at time \( t \) is given by
\[
d(t) = -\frac{2D}{\tau^3} t^3 + \frac{3D}{\tau^2} t^2. \] (5.5)

This movement subjected the cloud to large jerk at the beginning and end points, comparable to the Triangular Velocity profile.

**Minimum-Jerk**

This profile was used for movements of the cloud in all other applications of the DDS/FPGA unit. Strictly a minimum net-jerk-squared trajectory, the derivation of this simple quintic polynomial is given in [25]. The trajectory that minimises the action
\[
H(x(t)) = \frac{1}{2} \int_{t=0}^{\tau} \dddot{x}^2 dt \] (5.6)
over a distance \( D \) is found to be given by
\[
d(t) = \frac{10D}{\tau^3} t^3 - \frac{15D}{\tau^4} t^4 + \frac{6D}{\tau^5} t^5. \] (5.7)

At all times the acceleration changes smoothly. At \( t = 0 \) and \( t = \tau \) there are discontinuities in the jerk, and at other times the jerk is continuous. In their analysis, Shadmehr and Wise question the merit of a minimum-jerk trajectory over one that minimises snap (fourth time derivative of position) or other higher derivatives. They note that a natural reaching motion has a ratio of \( v_{peak}/v_{avg} \) most similar to this fifth-order polynomial movement.

**Quintic Stitch**

The motion profile used by Schmid *et al.* in [16] was recreated. They defined the piecewise function with
\[
a(t) = \begin{cases} 
\frac{D}{\tau^2} \left(-\frac{7040}{9} \left(\frac{t}{\tau}\right)^3 + 320 \left(\frac{t}{\tau}\right)^2\right) & 0 < t \leq \frac{\tau}{4} \\
\frac{D}{\tau^2} \left(\frac{3200}{9} \left(\frac{t}{\tau}\right)^3 - \frac{1600}{3} \left(\frac{t}{\tau}\right)^2 + \frac{5120}{3} \frac{t}{\tau} - \frac{160}{9}\right) & \frac{\tau}{4} < t \leq \frac{3\tau}{4} \\
\frac{D}{\tau^2} \left(-\frac{7040}{9} \left(\frac{t}{\tau}\right)^3 + \frac{6960}{3} \left(\frac{t}{\tau}\right)^2 - \frac{5120}{3} \frac{t}{\tau} + \frac{4160}{9}\right) & \frac{3\tau}{4} < t \leq \tau
\end{cases} \] (5.8)
which gives a displacement of

\[
d(t) = \begin{cases} 
D \left( -\frac{352}{9} \left( \frac{t}{\tau} \right)^5 + \frac{980}{3} \left( \frac{t}{\tau} \right)^4 \right) & 0 < t \leq \frac{\tau}{4} \\
\frac{D}{\tau^2} \left( \frac{160}{\tau^3} (t - \frac{\tau}{4})^5 - \frac{200}{\tau^4} (t - \frac{\tau}{4})^4 + \frac{20}{\tau^5} (t - \frac{\tau}{4})^3 + \frac{35}{\tau^6} (t - \frac{\tau}{4})^2 + \frac{65}{8\tau^7} (t - \frac{\tau}{4})^3 + \frac{38}{3\tau^8} \right) & \frac{\tau}{4} < t \leq \frac{3\tau}{4} \\
\frac{D}{\tau^2} \left( -\frac{352}{\tau^3} (t - \frac{3\tau}{4})^5 + \frac{200}{\tau^4} (t - \frac{3\tau}{4})^4 + \frac{20}{\tau^5} (t - \frac{3\tau}{4})^3 - \frac{35}{\tau^6} (t - \frac{3\tau}{4})^2 + \frac{65}{8\tau^7} (t - \frac{3\tau}{4})^3 + \frac{209}{3\tau^8} \right) & \frac{3\tau}{4} < t \leq \tau
\end{cases}
\]

This profile was engineered to give a very smooth transport: the acceleration and jerk both change smoothly throughout the motion, with initial and final values of zero for both quantities.

**Trapezoidal Acceleration**

The sixth mode of motion is defined by an acceleration profile that is trapezoidal in time,

\[
a(t) = \begin{cases} 
\frac{36D}{\tau^3} & 0 < t \leq \frac{\tau}{6} \\
\frac{6D}{\tau^2} & \frac{\tau}{6} < t \leq \frac{\tau}{4} \\
-\frac{36D}{\tau^2} (t - \frac{\tau}{3}) + \frac{6D}{\tau^2} & \frac{\tau}{3} < t \leq \frac{2\tau}{3} \\
-\frac{6D}{\tau^2} & \frac{2\tau}{3} < t \leq \frac{5\tau}{6} \\
\frac{36D}{\tau^3} (t - \frac{5\tau}{6}) - \frac{6D}{\tau^2} & \frac{5\tau}{6} < t \leq \tau
\end{cases}
\]

(5.10)

When doubly integrated, using \(v(0) = 0\) and \(d(0) = 0\), the displacement is found to be

\[
d(t) = \begin{cases} 
\frac{6D}{\tau^3} t^3 & 0 < t \leq \frac{\tau}{6} \\
\frac{3D}{\tau^2} (t - \frac{\tau}{6})^2 + \frac{6D}{\tau^2} (t - \frac{\tau}{6}) + \frac{D}{36} & \frac{\tau}{6} < t \leq \frac{\tau}{3} \\
-\frac{6D}{\tau^2} (t - \frac{\tau}{3})^2 + \frac{3D}{\tau} (t - \frac{\tau}{3})^2 + \frac{3D}{\tau^2} (t - \frac{\tau}{3}) + \frac{7D}{36} & \frac{\tau}{3} < t \leq \frac{2\tau}{3} \\
-\frac{3D}{\tau^2} (t - \frac{2\tau}{3})^2 + \frac{3D}{\tau^2} (t - \frac{2\tau}{3}) + \frac{29D}{36} & \frac{2\tau}{3} < t \leq \frac{5\tau}{6} \\
\frac{6D}{\tau^3} (t - \frac{5\tau}{6})^3 - \frac{3D}{\tau^2} (t - \frac{5\tau}{6})^2 + \frac{D}{\tau^2} (t - \frac{5\tau}{6}) + \frac{35D}{36} & \frac{5\tau}{6} < t \leq \tau
\end{cases}
\]

(5.11)

Here, the choice was made for the constant acceleration phases (phases 2 and 4) to each take a sixth of the total time of the movement, the same as the first and last phases. This was in part motivated by the implementation of the profile: it meant only one “time” value had to be communicated to the FPGA, as each phase lasted the same time (again from the negative jerk phase, during which the timer counted twice from zero to the time limit). But this choice was arbitrary; any ratio of phase 1/phase 2 times could have been chosen and would have been straightforward to implement. For a given ratio of these times, the total displacement \(D\) and time \(\tau\) uniquely define the analytic expression for the movement, which will be different to that of Equation 5.11.

Plots of the jerk, acceleration, velocity, and displacement are shown in Figure 5.8. Apart from the movement at constant velocity, the displacement trajectories are remarkably similar. However, the differences become very obvious as derivatives are taken. In particular, notice the smoothly changing jerk of the Quintic Stitch. This profile turned out to be the smoothest.
5.3.2 Expected Position of a Cloud After Time of Flight

When a cloud undergoing sloshing is released from its trap and allowed to fall for some time, its position will be determined by its position and velocity at the time of release. Oscillation will primarily be in the direction of movement, which in our system is more or less parallel to the \( z \) axis. Assuming the oscillations are damped, its in-trap motion can be described by

\[
z(t) = Ae^{-\beta t} \sin(\omega t).
\] (5.12)
If the cloud is then released at \( t_0 \) and allowed to fall a time \( t_1 \), its final horizontal displacement will be

\[
\begin{align*}
\frac{dz}{dt} \bigg|_{t=t_0} &= A e^{-\beta t_0} \sin(\omega t_0) + A t_1 e^{-\beta t_0} (\omega \cos(\omega t_0) - \beta \sin(\omega t_0)) \\
&= A' e^{-\beta t_0} \sin(\omega t_0 + \delta),
\end{align*}
\]

where

\[
A' \equiv \sqrt{(1 - \beta t_1)^2 + \omega^2 t_1^2 A}, \quad \delta \equiv \omega t_1 / (1 - \beta t_1).
\]

Hence, by incrementing \( t_0 \) and holding \( t_1 \) constant we expect to observe damped oscillations, of greater amplitude than \( A \). Note that \( A' \) depends almost linearly on \( t_1 \), which allows the sloshing to be amplified with longer time of flight.

In our experiment, sloshing was observed, but not about a constant centrepoint. Instead, the centre of oscillation seemed to drift slowly, back towards the initial point of the motion. Such behaviour most likely arose from an anharmonic trapping potential. As only the sloshing amplitude was of interest, the drift was accounted for by adding an exponential decay term to Equation 5.12, such that the in-trap motion was

\[
z(t) = A \sin(\omega t) + Be^{-\gamma t},
\]

and the horizontal position at the time of imaging, Equation 5.15, is replaced with

\[
z(t_0 + t_1) = A' e^{-\beta t_0} \sin(\omega t_0 + \delta) + (1 - \gamma t_1) Be^{-\gamma t_0},
\]

in which the definitions of \( A' \) and \( \delta \) are unchanged. This term was only included to improve the fit to the data, and no quantitative conclusions were drawn from it.

### 5.3.3 Observation of Sloshing

All movements recorded here were over 1.5 mm. In previous applications of the tweezer control unit, movements over this distance had usually taken about 150 ms, a relatively long time. In this investigation, faster movements were used, to increase excitation. All movements took place with a horizontal beam power\(^7\) of 1.45 W and a vertical beam power\(^8\) of 0.066 W. The \( z \)-frequency was decreased for all movements, which meant the cloud moved to the left. The time of flight was 12 ms for each image, and no frequency toggling of any sort was employed.

First, clouds were moved under different displacement profiles. With profiles having constant and triangular velocity, cubic displacement, and minimum-jerk quintic displacements, the clouds were moved in 90 ms. For the motion at constant velocity, this time was just slow enough to avoid leaving many atoms behind at the beginning of the movement. For each profile, 21 images were taken, at 1 ms intervals starting at the moment the trap came to rest. The time evolution of the position in the \( z \) direction is plotted in Figure 5.9.

---

\(^7\)“Collider Attenuation 1” set to 7.8 V

\(^8\)z channel ASF set to 1023, x channel ASF set to 300
inset in the same figure. Offsets between the final positions of the clouds are on the order of a few microns, and are attributed to slow drift of the trap centre - all data sets were taken on different days. Oscillation amplitudes extracted from the data and the sloshing amplitudes they imply are recorded in Table 5.1.

When it came to the profiles with smoothly changing acceleration, the Quintic Stitch and Trapezoidal Acceleration profiles, a movement in 90 ms was too slow to excite appreciable sloshing. Movement time was decreased to 30 ms. In order to achieve some comparison with the 90 ms data set, a Minimum-Jerk motion was run over the same timescale. The cloud positions after a 12 ms time of flight are plotted in Figure 5.10 and the inferred sloshing amplitudes are written in Table 5.2. Surprisingly, the movement that minimised the net-jerk-squared caused oscillations of considerably greater amplitude than both of these movements.

It was very apparent a movement at constant velocity brought about the coarsest motion of the profiles used here. The amplitude of the in-trap sloshing was an order of magnitude greater than after any other type of profile. It was no surprise that this type of motion was rough: the velocity profile has two discontinuities, while the trap velocity with all other profiles is continuous.

Of the other motion profiles, the movement with triangular velocity and cubic displace-
CHAPTER 5. APPLICATION

Figure 5.10: Position at 12 ms time of flight in the 20 ms following 1.5 mm, 30 ms movements. Points are measured values and continuous lines are fits to the data. The Minimum-Jerk movement is included to compare, qualitatively, the smoothness of the Quintic Stitch and Trapezoidal Acceleration movements to those in Figure 5.9.

<table>
<thead>
<tr>
<th>Profile over 30 ms movement</th>
<th>Measured $A'$ (µm)</th>
<th>Inferred sloshing amplitude (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum-Jerk</td>
<td>30.9</td>
<td>2.12</td>
</tr>
<tr>
<td>Quintic Stitch</td>
<td>3.74</td>
<td>0.257</td>
</tr>
<tr>
<td>Trapezoidal Acceleration</td>
<td>11.6</td>
<td>0.806</td>
</tr>
</tbody>
</table>

Table 5.2: Sloshing amplitudes after smoother motions over 30 ms. In-trap sloshing amplitudes were calculated with Equation 5.16.

...
nuity in jerk was more detrimental than the larger integrated jerk-squared of the other two smooth profiles.

From this brief investigation, a preliminary conclusion can be drawn: that the higher the order of time derivative in which discontinuities appear, the smoother the motion. For trajectories that are continuous in the same order derivative, it appears the trajectory with the greatest single discontinuity will cause the greatest sloshing when the trap is stationary again. Perhaps investigations into trajectories that minimise the net square of the snap (fourth time derivative of position), or higher derivatives crackle and pop (fifth and sixth time derivatives of position, respectively), and restrict all lower order derivatives to begin and end at values of zero, will give even smoother movements.

Finally, Minimum-Jerk movements were used to transport a cloud by 1.5 mm in different times, ranging from 30 ms to 120 ms. This was, in part, to justify the use of a 150 ms movement when the cloud was split and evaporated. The $z$ positions of the clouds over the same 20 ms timescale are plotted in Figure 5.11. Unsurprisingly, the amplitudes decrease as $\tau$, the time of the movement, increases. Without the amplification effected by a time of flight, the sloshing amplitude of the cloud after a 120 ms movement was 0.12 $\mu$m, orders of magnitude smaller than the in-trap cloud width of $\sim 25 \mu$m. Figure 5.12 plots the measured oscillation amplitudes and those inferred for the in-trap motion. In Sections 5.1 and 5.2, clouds were first split in a 150 ms movement over 0.5 mm before evaporation, with subsequent 150 ms, 0.25 mm movements. The minuscule in-trap motions measured here, for movements over three times the distance in shorter times, justify the choice of these values.
Figure 5.11: Position at 12 ms time of flight in the 20 ms following 1.5 mm Minimum-Jerk movements over different times.

Figure 5.12: Measured oscillations in position after a 12 ms time of flight (blue circles) and inferred in-trap sloshing amplitude (red triangles) versus time for 1.5 mm Minimum-Jerk movements over.
Chapter 6

Summary

In the work undertaken for this thesis, a DDS was programmed, via an FPGA, to produce two voltage outputs suited to a specific application: smooth movement of ultracold atoms in optical crossed-beam traps, by acousto-optic deflection of one of the trapping beams. The FPGA was programmed, in the VHDL hardware definition language, to communicate at $\sim 1 \mu s$ intervals a series of frequency tuning words and amplitude scale factors to the DDS, defining independent but coherent outputs on two channels of this device. The digital frequency source was able to simply plug into an optical tweezer unit, and it was shown it was able to move clouds of ultracold $^{87}$Rb smoothly, tracking a stationary horizontal waveguide over distances of several millimetres. Control of the optical trap stiffness was also demonstrated, by pinching and relaxing a cloud to excite a clear breathing motion.

To move atomic samples in smooth movements, the paired frequency outputs of the DDS were required to follow non-linear trajectories. An accumulation approach was developed that carried out the computations of all parameters defining the outputs on the FPGA. It was found to be effective, and in some sense it was advantageous, as preparing and writing a large memory file for an extended experimental sequence can be expensive in machine time. The method replaced multi-operand multiplications with five additions. To our knowledge, it has not been used before.

The frequency source was successfully used in three applications, hinting at its potential for future use. A number of different motion profiles were implemented and used to move clouds over a 1.5 mm distance, with movements with continuous acceleration performing considerably better than those discontinuous in acceleration. A cooling method, in which a crossed-beam dipole trap was deformed rather than the trapping potentials weakened, was briefly trialled, and promising results were obtained. Most notably, four macroscopically spaced clouds were evaporatively cooled beyond the BEC phase transition. Such an array of condensates, in independently controllable traps, could be used in a number of quantum engineering applications beyond the capabilities of past magnetic BEC array experiments. We note the recent observation of macroscopic spin squeezing of an extended array of condensates in an optical lattice [54], a convincing demonstration of an application of such a system.
Appendix A

Dipole Trapping in Gaussian Beams

Here, a brief account of dipole trapping is given, based largely on [14] and [55]. The dipole force arises when an electromagnetic field interacts with a neutral but polarisable particle. The net time-averaged force which acts on the induced electric dipole can be associated with a conservative potential, which depends on the particle’s polarisability. For red-detuned light, the particles are found to be attracted to regions of high intensity, the basis of an optical trap.

The cross sectional intensity distribution of a Gaussian laser beam of power $P$ and a $1/e^2$ radius of $\sigma$ is

$$I(r) = \frac{2P}{\pi\sigma^2} \exp\left(-\frac{r^2}{\sigma^2}\right) \quad (A.1)$$

For radiation of angular frequency $\omega$ which is far detuned from the resonance $\omega_0$, the optical potential is found to be

$$U_\text{dip}(r) = \frac{3\pi c^2}{2\omega_0^3} \frac{\Gamma}{\Delta} I(r), \quad (A.2)$$

where $\Delta$ is the detuning $\omega - \omega_0$ of the optical frequency from resonance and $\Gamma$ is the natural line width of the transition. For the $D_2$ $5S_{1/2} \rightarrow 5P_{3/2}$ transition in $^{87}\text{Rb}$, this value is $2\pi \times 6.065$ MHz [56].

For a collimated Gaussian laser beam, particles will be trapped only in the radial directions, and will diffuse readily along the beam axis. But the intersection of two such beams will give a local potential that is attractive in all three spatial dimensions, with the strongest trapping in the direction orthogonal to both beam axes. In our crossed-beam trap, the total potential was the sum of the horizontal and vertical potentials,

$$U_\text{tot} = U_H + U_V. \quad (A.3)$$

We can define constants

$$U_{H0} \equiv U_H(0) = \frac{3\pi c^2}{2\omega_0^3} \frac{\Gamma}{\Delta_H} \frac{2P_H}{\pi\sigma_H^4}, \quad U_{V0} \equiv U_V(0) = \frac{3\pi c^2}{2\omega_0^3} \frac{\Gamma}{\Delta_V} \frac{2P_V}{\pi\sigma_V^4}, \quad (A.4)$$

which are both negative due to the negative detunings. Then, to quadratic order the total potential can be approximated as

$$U_\text{tot} \approx U_{H0}(1 - \frac{2x^2}{\sigma_H^2}) + U_{V0}(1 - \frac{2y^2}{\sigma_V^2}) \quad (A.5)$$

which becomes

$$U_\text{tot} = U_{H0} + U_{V0} - 2\left(\frac{U_{H0}}{\sigma_H^2} + \frac{U_{V0}}{\sigma_V^2}\right)x^2 - 2\frac{U_{H0}}{\sigma_H^2}y^2 - 2\frac{U_{V0}}{\sigma_V^2}z^2. \quad (A.6)$$
Hence we have a 3 dimensional harmonic oscillator, with trapping frequencies

\begin{align*}
\omega_x &= 2 \sqrt{\frac{U_{1H}}{m\sigma_H^2}} + \frac{U_{V0}}{m\sigma_V^2}, \\
\omega_y &= 2 \sqrt{\frac{U_{1H}}{m\sigma_H^2}}, \\
\omega_z &= 2 \sqrt{\frac{U_{V0}}{m\sigma_V^2}},
\end{align*}

(A.7)
Appendix B

Additional FPGA programs

This appendix contains descriptions of other programs written during this thesis work. First, an alternate way to program the AD9959 DDS is described, which was used when the FPGA/DDS first drove the optical tweezer unit. These projects successfully collided thermal clouds of $^{87}\text{Rb}$, and examples of the scattering patterns observed are given here. Second is an overview of a pulse generator program used to control the imaging sequence in the experiment. This program ran on a different FPGA and has become a permanent part of the experiment.

B.1 Collision Programs

The work discussed in the body of this thesis used a certain style of FPGA program, one in which the DDS was continually programmed with pairs of frequency tuning words. Initially, a different style of project was designed: one that used the Linear Sweep Mode of the DDS. By setting an initial FTW, two values that defined a ramp rate, and a final FTW, the DDS board could be programmed to perform a linear sweep from one frequency to another. While this function provided a straightforward way to near-continuously change the frequency, it was not very appropriate when trying to move clouds smoothly, with dynamic and sometimes non-linearly changing velocities. The matter was complicated further by the need for a toggled output, which meant each linear sweep needed to last for $\sim8\mu$s or less. However, an attempt was made to use it, and some success was had. This section sketches the inner algorithms and the application of the programs.

In programs of this style, the long-term behaviour of the frequency output was defined by programming the FPGA with a few integer values. It would then program the DDS with a series of linear sweeps, stitched front to back, which crudely approximated motion with a trapezoidal acceleration profile. The slope of the first phase was set with two integer values, the Rising Delta Word (RDW) and the Linear Sweep Ramp Rate (LSRR). In all programs discussed here the system clock ran at 500 MHz, so the initial frequency ramp rate was

$$\frac{\Delta f}{\Delta t} = \frac{500 \text{ MHz}}{2^{32}} \frac{\text{RDW}}{8 \text{ ns} \times \text{RSRR}} = \frac{\text{RDW}}{\text{RSRR}} \times \frac{500}{2^{35}} \text{ MHz/ns}. \quad (B.1)$$

During each phase of the movement, a trap site would move linearly. For the first few phases of the motion, the slope of the linear ramps would increase at an increasing rate,
approximating parabolic curves. This was achieved by setting the gradient of the \( n \)th phase to be the initial gradient multiplied by the \( n \)th number in the series of triangle numbers, 1, 3, 6, 10, \ldots, a little like the accumulation method described in Section 3.5. The number of phases over which this happened was programmed before the movement. Next, there were a series of phases during which the slope increased linearly, as if with a fixed positive acceleration. Finally, there would be a few phases during which the slope increased at a decreasing rate, mirroring the initial parabolic set of phases and taking the slope smoothly to its maximum value. From then on, the slope decreased symmetrically to bring the cloud to rest. An example of a short motion of this type is shown in Figure B.1(a) & (b).

Each of these “linear” sections was actually broken into a number of 6.9 \( \mu \)s ramps, which alternated between high-and-rising and low-and-falling frequencies. As a rising ramp reached its maximum value the DDS would toggle to a lowering ramp, exactly symmetric to the rising one. A rising ramp on output channel 0 was triggered by asserting the DDS’s Profile Pin 0 input via the FPGA. While the pin was held high, the “sweep accumulator” register linearly incremented inside the DDS, and the internal FTW along with it. When the pin was dropped, the accumulator was programmed to decrement at the same rate. However, at this time the DDS would jump to new values for the initial and final FTWs. In this way the output frequency, now lowering from a lower frequency, would toggle to

![Figure B.1: Sketches of a series of linear sweeps stitched front to back. (a) The normalised gradients of a short string of linear ramps, showing a parabolic increase in slope over three phases, a single phase maintaining a constant ramp gradient, and a three phases during which the slope increases at a decreasing rate to reach a maximum slope. Following is a symmetric decrease in ramp rate to zero. (b) The normalised displacement due to the movement with slopes following (a). (c) Close up of the actual DDS output frequency at the beginning of a split sequence. The blue lines are the instantaneous output, which toggles between two sites. The red dotted lines extrapolate each section to show the frequency the DDS must begin from when it toggles back.](image-url)
address the other trap site. The initial frequency for a linear sweep addressing a certain cloud was always extrapolated from the last linear sweep addressing that same cloud, as the red dotted lines in Figure B.1(c) indicate.

This style of project was used to collide the clouds. After the splitting movement there was a wait phase, then one during which the clouds were moved with constant acceleration to a collision. This second movement phase was the same as the first in all matters described above. It was effected by taking the settings for the initial acceleration phase of the split and multiplying every slope by a constant value, to give a fast acceleration and a high maximum velocity. The hop time was the same in both phases. Figure B.2 gives two images of scattering patterns recorded using these projects.

In some final notes, the DDS would output frequencies on two channels, but the output of channel 1 would remain constant throughout the movement. The alignment of the z-axis and the H-beam was good enough to use a static x-frequency drive for sufficiently small separation distances. No attempt to control the amplitude was made, beyond setting the ASF to its maximum value of 1023 for both channels. This type of project did not use any type of memory reading: all integer numbers were programmed into the logic by hand, with the buttons, rotary knob, LEDs, and LCD screen on the FPGA board.

Figure B.2: Collisions run by the FPGA/DDS unit showing s-wave (left) and d-wave (right) scattering. These images were acquired by Amita Deb and Julia Fekete during their use of the program.

B.2 An FPGA-based Timing Control Unit

In a completely different application of FPGA logic, a program was created that integrated with the main experimental control program, Rebeka, to function as a pulse delay generator to run the dipole trapping and imaging phases of the experiment. With the FPGA’s internal clock running at 50 MHz, the time resolution of the program was a significant improvement on RebeKa’s 14 $\mu$s clock period. This program, fondly named Imogen, has become a permanent component of the control circuitry.

RebeKa would pass timing control over to Imogen at the beginning of the dipole trapping phase of an experimental run. From this point on, Imogen controlled the TTL inputs to the H- and V-beam AOMs, the trigger to the FPGA/DDS unit, and the various components involved in the imaging sequence, all by digital outputs on the J2 and J15 headers. The “Trap laser out” output would be asserted as soon as the timing control was passed over. After a “load time”, during which the dipole trap was held stationary, the
FPGA/DDS unit was triggered to run its pre-programmed FTW sequence. Throughout, Imogen counted to the time at which the imaging phase of the experiment began. At this point, the dipole trap was turned off by lowering “Trap laser out” and after a designated time of flight the cloud would be pulsed with resonant light, and the shadow imaged. All three times were able to be communicated to the FPGA via the memory write/read operation described in Section 3.4. And as that section describes, many sets of times could coexist in the memory, and could be read consecutively to automatically run a series of images at different times and/or different times of flight.

An imaging sequence itself requires coordinated control of three digital outputs: the probe light resonant with $^{87}$Rb (passed through an AOM to allow a quick turn-on and turn-off), the camera shutter, and the camera trigger. Before imaging, the AOM is driven to keep it warm while the camera shutter remains closed. 5.5 ms before the moment of imaging, the probe light is turned off, and 0.5 ms later the shutter opened. After a further 5 ms, the probe laser is turned on for 15 $\mu$s and at this time the image is recorded. In reality, there was a 6 ms latency associated with the camera trigger, so this signal was always asserted 6 ms earlier than the probe pulse, 0.5 ms before the probe light was turned off for the first time. The camera trigger was held high for 8 ms and the camera shutter was always closed 0.5 ms after the end of the probe pulse. In each experimental cycle, this process was repeated 300 ms later, to collect a reference image with no atoms, and then again with no probe light after another 300 ms. All three images were necessary for the data processing. All timings described here remained fixed for all shots, and were not read from memory.

The probe light was not used only for this purpose, it was also used as an optical pumping beam, to transfer the atoms to the $|2,2\rangle$ state. It was necessary to be able to activate the AOM deflecting this probe light independently of the imaging sequence. To this end, the digital probe and shutter output were the products of two OR gates, between inputs from RebeKa and the inner probe light signal and shutter signals. The probe light could then be turned on and the shutter opened by either source. It was also important the imaging sequence was able to be run without always turning on the dipole trapping laser, as it was often necessary to image the atoms at an earlier time. To achieve this the H- and V-beam attenuators were set to zero from RebeKa, so although the “Trap laser on” signal was always asserted there would be no light to interact with the atoms.

Imogen was not restricted to imaging rubidium: parallel to the counters controlling the rubidium probe light and shutter were an identical set controlling probe light resonant with potassium. Sympathetically cooled $^{40}$K has been successfully imaged with the device. For this species, the shutter opens and closes at the same time as the rubidium shutter, although an independent signal exists in case the times need to be changed for one species. The probe pulse begins at the same time but lasts three times as long. As both species are imaged by the same camera, only one camera trigger signal exists.

To extend this program’s utility to recent areas of research in the lab, extra outputs have been written that can be turned on and off at times read from memory. One set of these outputs is intended to be used to trigger microwave sweeps that will drive transfers between atomic ground states. Another is intended to trigger a train of pulses that will dispersively probe the sample.
Bibliography


