Direct measurement of fluorescence lifetime using high speed data acquisition

MATTIAS OLSSON
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Direct measurement of fluorescence lifetime using high speed data acquisition

Mattias Olsson
ael09mol@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Dr. Fredrik Edman, Dr. Liang Liu and Dr. Khadga Jung Karki

Examiner: Erik Larsson

May 20, 2018

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Time-correlated single photon counting (TCSPC) is commonly used to measure the lifetime of fluorescence from dye molecules. In TCSPC, the sample is excited by a pulsed laser and the arrival time of each of the emitted photons at the detector relative to a trigger is recorded. An important criterion of the technique is that the excitation intensity of the laser has to be low enough such that the detector detects at most one photon per ten laser pulse excitations. As one needs to detect about 10^4 to 10^6 photons in order to generate a time trace of the fluorescence, the measurement has to be repeated about 10^7 times, which makes the technique rather slow. A typical measurement takes about 1s which is very long when the fluorescence lifetimes of most of the dye molecules range from few nanoseconds to microseconds. In this master's thesis, a novel method based on interleaved sampling and waveform averaging to directly measure the fluorescence decay from the dye molecule Rhodamine 6G is presented. A new algorithm using boot-strapped waveform averaging to improve the effective sampling rate as well as the signal to noise ratio of the digitized signal have been implemented and tested. The new algorithm has enabled us to reduce the measurement time to a few tens of a microseconds, which is approximately a million times faster than the TCSPC method.

Table of Contents

List of Figures

Nomenclature

Table of variables used in equations		
Name	Description	SI-unit
F_{ADC}	Frequency of the analog to digital converter	Hz
F_{Laser}	Frequency of the excitation laser	Hz
$NOSPP$	Number of samples per pulse	
$NOUW$	Number of used waves	
$\Delta\tau$	Amount each wave is shifted	s
T	Time between pulse	s
Δt	Time represented by sample	s

Chapter 1

Background

1.1 Fluorescence lifetime measurements

Florescence occurs when molecules absorbs energy from an external source. In spectroscopy, a light source with a short wavelength is used to excite a molecule of interest. The molecule absorbs the the energy coming from the light and, due to stokes shifts, emits light at a longer wavelength. This emission lasts pico to nano seconds with an often exponential decay time. The time constant of the decay is also known as fluorescence lifetime.

Many organic and inorganic molecules fluoresce when they are excited. [1] Fluorescence is a specific and unique property of a molecule. The spectrum of the fluorescence is determined by the energy of the excited state of the molecule.

Fluorescent molecules in a biological environment are surrounded by different kinds of solvents and proteins with which the molecules interact. Such interactions shift the energy levels, which lead to the shifts in the position and the line width of the fluorescence spectrum. [1]

In most of the cases the fluorescence spectra of molecules are broad and overlapping with each other, which limits the capability to identify the different molecules by the spectrum alone. Moreover, the intensity of the fluorescence depends on the concentration of molecules. In biological applications, such as fluorescence imaging of cells, the local concentration of molecules can vary widely. A quantitative analysis of the influence of the environment on the molecules based on the steady state spectra is often ambiguous. The fluorescence lifetime, on the other hand does not depend on the concentration. In addition, it is also highly sensitive to the local environment around the molecules [1], which makes it attractive for many medical applications, such as tissue characterization, diagnosis of pathological conditions, cancer detection, etc. [2, 3, 4]

1.2 TCSPC measuring technique

Different methods such as time correlated single photon counting (TCSPC) [5], phase modulation [6, 7], gated fluorescence detection [8, 9] and direct recording of full transients using fast digitizers [10, 11, 12] have been used to measure the lifetime of fluorescence. TCSPC is the most commonly used technique for the measurement of the fluorescence lifetime.

Compared to other methods, TCSPC has advantageous features such as low systematic errors, the best signal-to-noise ratio (SNR), single photon sensitivity, wide dynamic range and clearly defined Poissonian statistics. [5] Although, TCSPC has been used in point spectroscopy, as well as in imaging applications, [13, 14, 4] the measurement time of a single recording is rather long due to the dead-time of the detection system after each photon is detected.

In order to avoid missing detection of photons during the dead-time one usually uses weak fluorescence such that no photons arrive at the detector during the dead-time. However, this means that single photons from only about 5% of the excitation pulses are detected. The slow data acquisition is a disadvantage of TCSPC method, especially in applications that require fast measurements [2].

1.3 Direct recording measuring technique

More recently direct recording of fluorescence transients, using fast detectors and data acquisition systems have been used as an alternative lifetime measurement technique in clinical applications [10, 11, 12, 2].

Direct recording of the transients is particularly advantageous in samples that are highly fluorescent. In this case, it is even possible to record a transient in real-time. However, in most of the applications a number of recordings are averaged to gain better signal-to-noise ratio. Previously, oscilloscopes have been used to record the data at a sampling rate of about 5 GSa/s, which gives a temporal resolution in data acquisition of about 200 ps. The bandwidth of the oscilloscopes have been about 1 GHz. [10, 11, 12, 2] One of the ways to improve the effective temporal resolution without increasing the cost of instrumentation is to use interleaved sampling. As most of the measurements are done using repetitive excitation, it is straight forward to implement interleaved sampling. [15] However, the traditional methods of interleaved sampling can be time consuming, [15] which offsets the benefits of fast measurement time using direct transient recording.

1.4 Novel measuring technique

In this master's thesis we present a novel technique of fast waveform averaging, which we call boot-strapped waveform averaging (BSWA), to improve the temporal resolution in the direct measurement of fluorescence transients without compromising the speed of the measurement. Our technique provides an effective digitization resolution of about 30 ps. We have implemented and verified the technique by measuring the lifetime of fluorescence from Rhodamine 6G.

Chapter 2

The BSWA technique

The boot-strapped waveform averaging technique (BSWA) is a digital modification of the time interleaved sampling (TIS) [16] technique. In TIS one uses more than one analog to digital converter (ADC) to increase the sampling rate. The repetitive signal to be processed is simultaneously channeled to different ADCs. The ADCs are clocked with a slight phase difference such that they sample different sections of a repetitive signal. Figure 2.1 taken from [16], shows an illustration of the time interleaved sampling with two ADCs. The top wave is the desired signal, the next two signals are the clock signals to ADC 1 and 2, the next two signals after that are the signals sampled by the two ADCs, and the bottom graph shows the final signal obtained by concatenating the sample signals from ADCs 1 and 2.

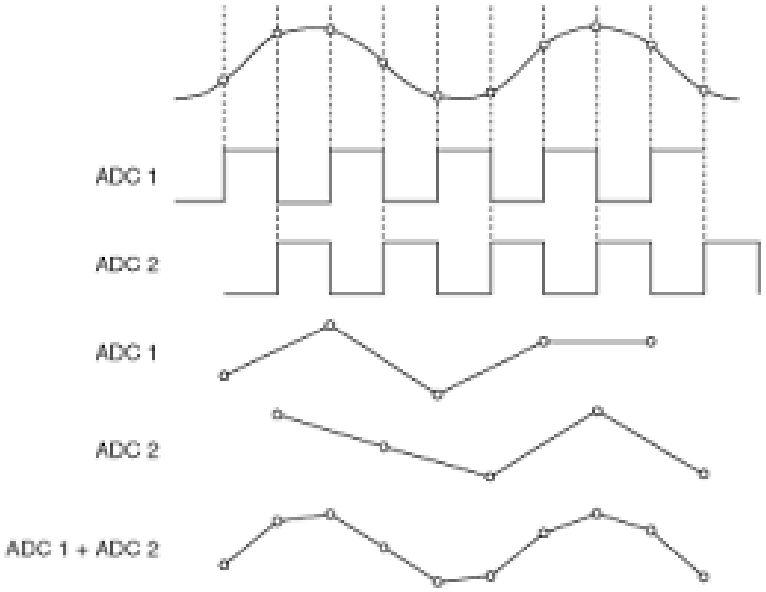

Figure 2.1: Time interleaved sampling.

2.1 The theory behind BSWA

The idea behind the boot-strapped waveform averaging, BSWA, is to interleave unique periods of the signal and by doing so increase the effective sampling rate with less cost and better SNR than achieved with TIS. To do so we need to know the following variables as shown in Figure 2.2. T represents the period of the waveform, which in our case is the time between the laser pulses. The Δt denotes the time interval of the digitization of the signal, and $\Delta\tau$ is the time difference between the end of the periodic signal and the end of the sampling of that period.

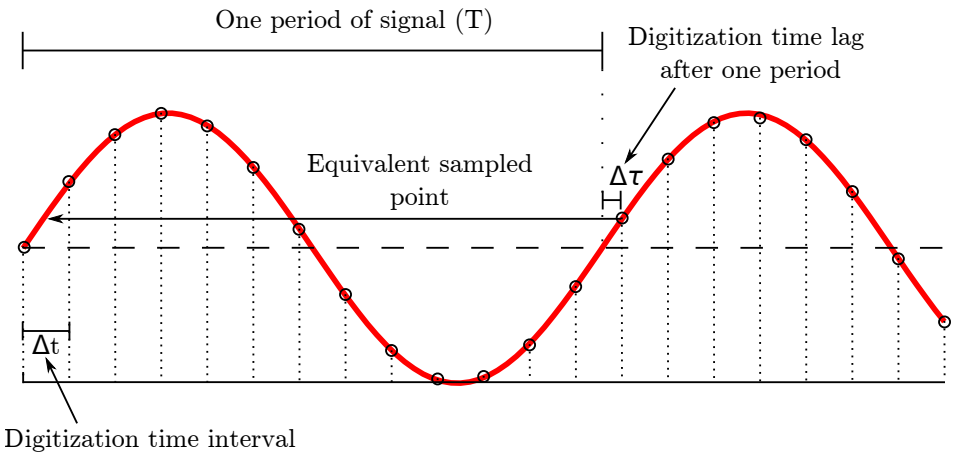

Figure 2.2: Concept of boot-strapped waveform averaging (BSWA).

In general, T is not commensurate with Δt, i.e. $\Delta\tau$ is non zero. Consequently, the following period of the signal is sampled with a phase shift. This effect is utilized to achieve the enhancement over the TIS technique.

The algorithm uses the repetition rate of the oscillator and the sampling frequency of the analog to digital converter (ADC) to determine the condition for the concatenation of the different periods of the signal. Usually, the sampling frequency of the ADC is known or can be set in the ADC but the rate at which the oscillator generates the laser pulses should be measured prior to the implementation of the algorithm. The rate at which the laser pulses are generated can be measured simply by detecting the direct beam of the laser in the same setup for the measurement of the fluorescence. Fast Fourier Transform (FFT) can then be used to obtain the frequency of the generation of the laser pulses from the digitized signal.

The number of digitized samples (NOSPP) acquired between two consecutive laser pulses is given by equation 2.1,

$$NOSPP = \frac{F_{ADC}}{F_L} \tag{2.1}$$

where F_{ADC} is the sampling frequency of the ADC and F_L is the frequency at which the oscillator generates the laser pulses.

We calculate $\Delta\tau$ (shown in Figure 2.2) by using equation 2.2,

$$\Delta\tau = \left(\frac{1}{F_{ADC}} * \lceil NOSPP \rceil\right) - \frac{1}{F_L}. \tag{2.2}$$

From $\Delta\tau$ we can calculate the number of cycles ($NOUW$) after which the digitized waveform repeats. $NOUW$ is given by equation 2.3,

$$NOUW = \frac{\frac{1}{F_L}}{\Delta\tau}. \tag{2.3}$$

As the repeated waveforms are digitized at identical time positions, they can be averaged to improve the SNR to a desired level. The averaged waveform contains $NOUW$ number of time traces of the fluorescence decay signal, which are digitized at slightly different time instances relative to the laser pulses. These different time traces need to be sorted and concatenated in order to generate a single trace with higher effective sampling rate. The sorting of the different traces is done by using the fact that the consecutive traces are shifted in time by $\Delta\tau$. We call this method of sorting "boot-strapping" as it interleaves the samples from the different traces.

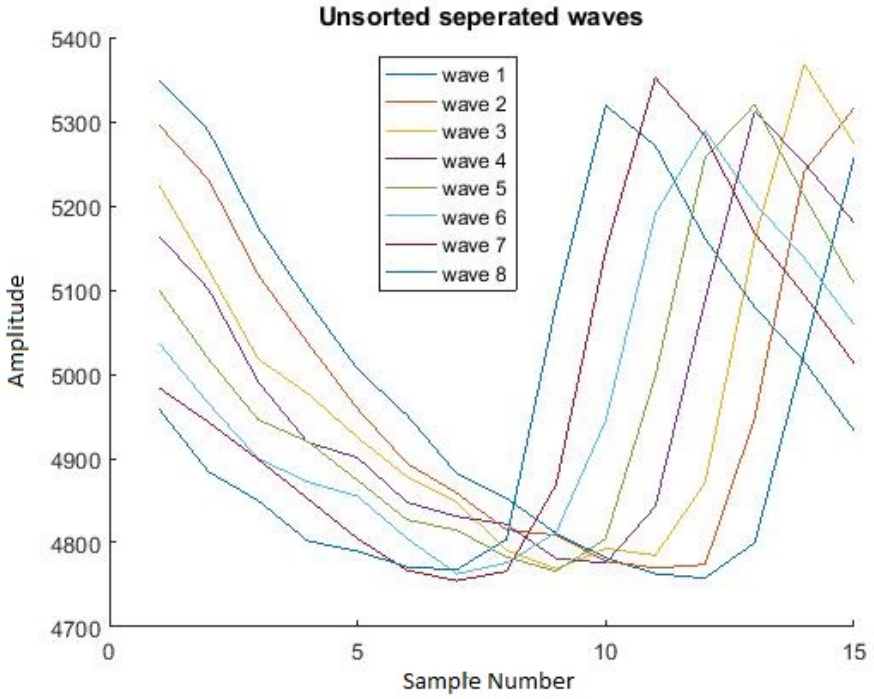

Figure 2.3: Unsorted time traces of the fluorescence signal.

2.2 Example of the BSWA technique

Figure 2.3 shows an example where eight time traces have been obtained after adding 40 waveforms. The digitization is done with a single ADC clocked at 1.25GHz, which reduces the cost and the risk of clock skew and provides us with a better SNR. We have used an avalanche photo detector (APD) with a bandwidth of 1600MHz, in order to get get more exact information with each sample. The high speed APD has also a favorable impact on the SNR since it holds a noise-equivalent power (NEP) of $0.4\frac{pW}{\sqrt{Hz}}$. In this example the signal from the APD is digitized at the rate of 1 GSa/s. The laser pulses are generated at the rate of 70.16 MHz. Thus, $NOSPP \approx 14.2531$, $\Delta\tau \approx 0.2531$ ns and $NOUW \approx 56$ (note only eight of the traces are shown in Figure 2.3). As can be seen in the figure, the relative position of each of the trace with respect to the laser pulse (at the maximum of the signal) is shifted. However, a final florecence decay signal can be obtained by interleaving the traces using the boot-strapping technique described above. Figure 2.4 shows the final fluorescence decay signal after all the time traces in figure 2.3 have been interleaved using boot-strapping. The effective sampling rate after interleaving is 56 GSa/s.

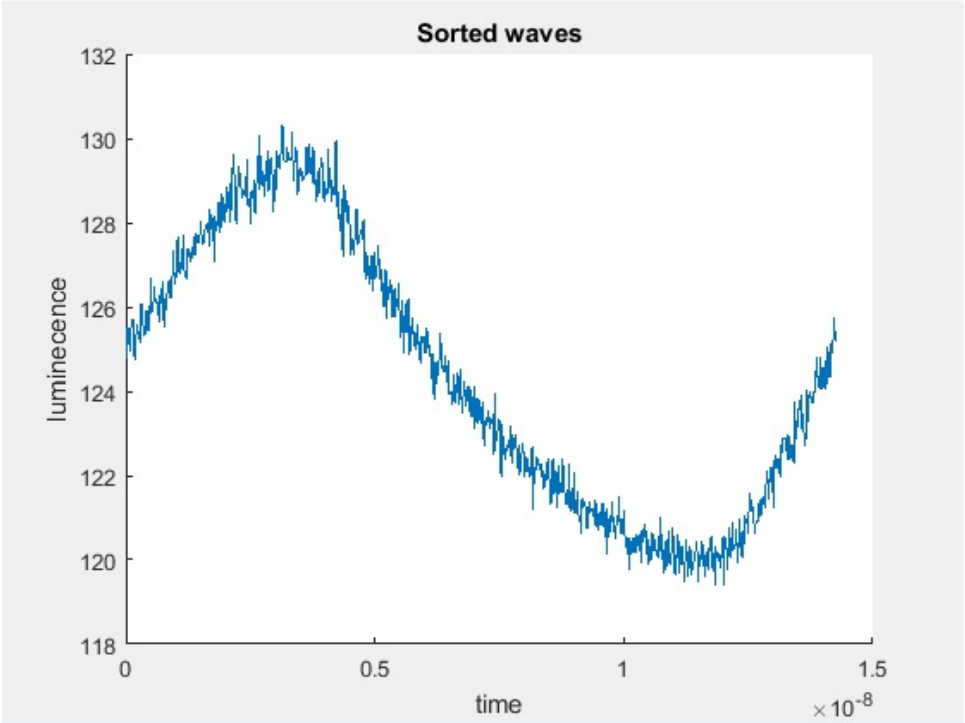

Figure 2.4: Interleaved signal using the BSWA technique.

Chapter 3

Implementation of the Measuring System

The measuring system incorporating the BSWA technique, as shown in figure 3.1, is based on a combination of software running on a personal computer (PC) and dedicated hardware. The hardware is primarily implemented using a field-programmable gate arrays (FPGA) for accelerating some of the computationaly heavy signal processing tasks in the measuring system.

3.1 Measuring system 1.0

The first version ot the measuring system was a software based implemented where the FPGA only was used to transfer data to the PC. Hoowever, the software based implementation had a severe computational bottleneck due to the 5 GB/s capture rate was far greater then the 115,2 KB/s transfer rate of the FPGA, resulting in that the memory of the FPGA became full of data and the continuous sampling would have to stop. This was a major drawback of the measuring system and had to be dealt with.

3.2 Measuring system 2.0

To solve the issue with the software based measuring system we move some of the processing into the FPGA. The dotted section in figure 3.1 indicates the functions that was implemented in hardware in the FPGA. To further optimize the measuring system and preserve the accuracy of the data the averaging operation in figure 3.1 is a mixed software and hardware implementation. When the collected data is averaged we receive both integer and decimal numbers. Handling decimal numbers in hardware usually requires more memory and some additional processing resources and is much more cumbersome to implement compared to using the built-in functionality in the software on the PC. However, the FPGA that was used had limited resources available so we needed to carefully consider what functionality to implement in hardware and what functionality to implement in software in the PC. Therefore, the choice of perform the averaging in software on the PC instead of in hardware in the FPGA was made resulting in shorter implementation time and better control over the numerical accuracy of the system. Yet another reason for splitting up the average operation is the specification of the ADC in relation to the memory size. In our case the ADC FMC125 uses burst sampling

of 16k so we can calculate how many samples we get with each sample burst and thereby know the needed size of the memory. The add-operation was implemented in hardware resulting in that less data points needed to be sent to the PC, and thus removing the bottleneck, which allow us to use continuous sampling.

The resulting hardware implmentation of the system nearly consumed all resources in the FPGA. Consequently, if further improvements such as filtering of the data should be implemented other larger FPGA (an FPGA with more resources) would be needed which would increase the cost of the system

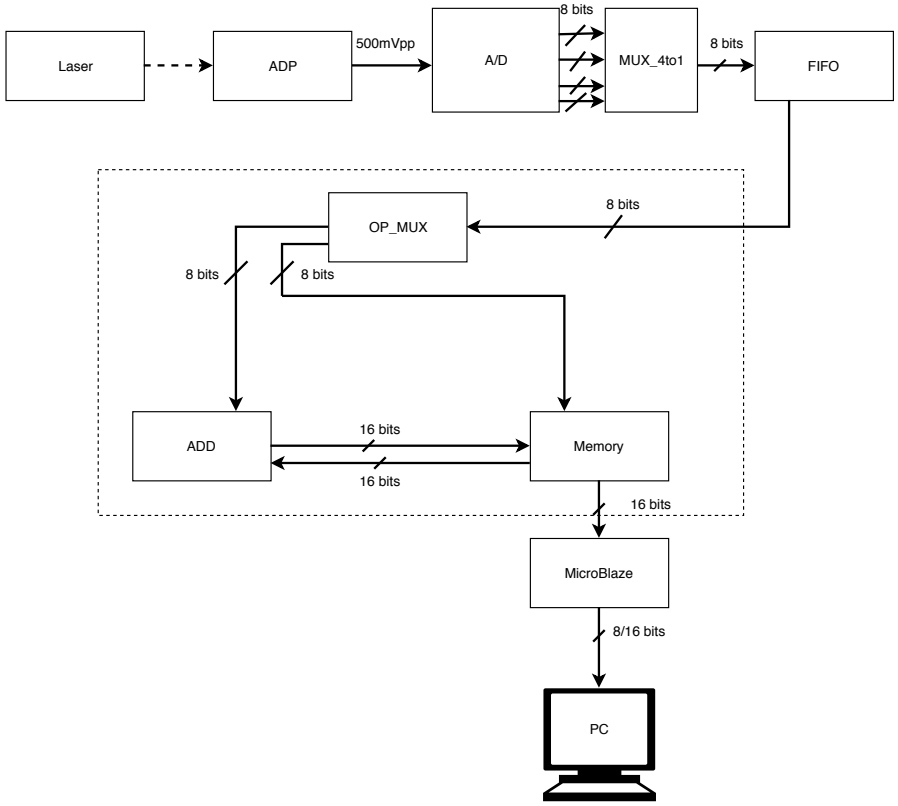

Figure 3.1: A block diagram of the implementation of the measuring system.

3.3 Hardware details of the measuring system

In the hardware implementation of the BSWA based measuring system a high sensitivity avalanche photodetector (APD210) from MenloSystems [17] is detecting the laser pulses coming from the laser. The APD is ideally suited for applications requiring highest sensitivity for low level input signals and is also fast enough for the characterization of pulsed lasers on the nanosecond time scale which was a requirement in this case. The ADP detects the laser pulses and generates an analog voltage response within the range of 500mV peak to peak.

The analog voltage is sent to an analog to digital converter (FMC125) from Abaco Systems [18], former 4DSP. The quad-channel multi-mode ADC offers four 8-bit ADC channels that enable simultaneous sampling of 4, 2, or 1 channel with a maximum sample rate of 1.25 GSPS (4-channel mode), 2.5 GSPS (2-channel mode), or 5.0 GSPS (1-channel mode) [19].

The sample output from the ADC, where each binary value represents a small voltage span of about 2mV, is shown in figure 3.2. As can be seen in the figure, four samples will be sent simultaneously with padding zeros between the samples, so in order for the sampled data to make sense it is important check that the phase offset of the four clocks, generated by the clock distribution circuit AD9517-1 on the ADC card, are correctly set depending on which mode we use to 1-, 2-, 4-channel mode [20].

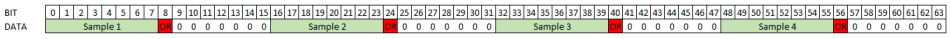

Figure 3.2: Description on how the samples are outputted from the FMC125.

The ADC allows for flexible control of several parameters such as the clock source, sampling frequency, and calibration through the I2C serial communication bus. Determining the desired settings for the clock distribution circuit can be done by using a program from Analog Devices [21]. The setup interface of this program is shown in figure 3.3 where the interface in the figure is set for a sampling rate of 2.5GHz.

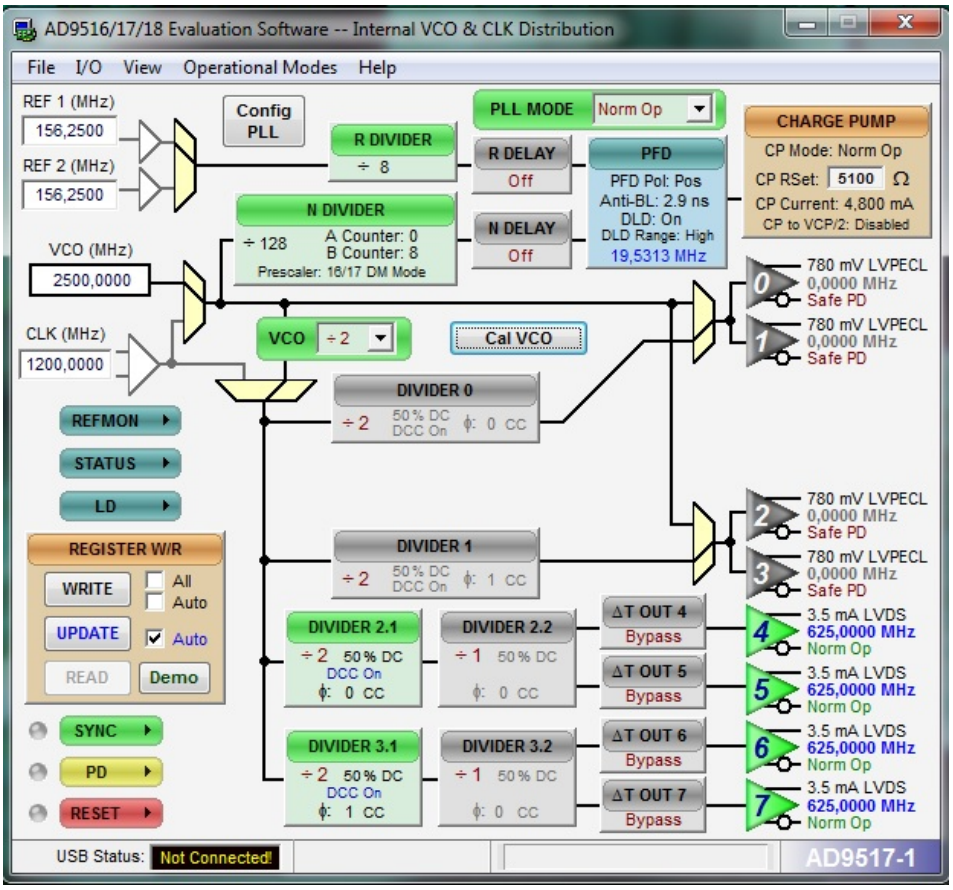

Figure 3.3: GUI for setting up AD9517-1.

The digitized values from the ADC are sent to a 4-to-1 multiplexer (denoted MUX_4to1 in figure 3.1) which divides the ADC output into 4 samples and send them in order to a first-in-first-out (FIFO) memory, implemented in an Xilinx Kintex-7 FPGA card (KC705), that stores up to 16383 Bytes, which is the size of one sampling burst from the ADC for temporary storage.

The samples are sent from the FIFO via a multiplexer (denoted OP_MUX in figure 3.1) to the on-board 2Gb RAM memory on the FPGA card which inour case is configured to store 392 Bytes due to how many unique samples we capture and how many times we encounter an identical sample later on in the sampling process since these samples will be summed up to 196 data points consisting of 16 bits each.

When a set of samples have been stored in the memory we read one sample from the FIFO and one sample from the memory and add them together via an addition function in the FPGA. The sum is then sent to the memory for storage. The ADC uses a burst of 16k samples, and each memory cell in the FIFO is set to hold 16 bits which then would allow for at least 257 samples to be added together

before the configured on-board memory cells overflows. In our case 83 samples are added into a sum creating one data point.

The storage and summation process between the FIFO, the memory and the adder that creates and stores a series of data points in the memory is controlled by a finit state machine (denoted FSM in figure 3.1). When the process of creating and storing of the series of data points is done a soft microprocessor (denoted MicroBlaze in figure 3.1) in the FPGA initiates a data transfer of the series of data points in the memory to a personal computer (denoted PC in figure 3.1) via a standard UART interface. Software in the PC then divides the data points by the number of samples summed up for one data point. After the division a sorting algorithm concatenates the data points. From the created data, installed third party software like Matlab can be used to further analyze and process the collected data.

The MicroBlaze on the FPGA is also used to initialize the ADC and sync the sampling process. The max sampling burst in the system is currently 16383 samples which corresponds to approximately a sample time of 13μs with a sampling rate of 1,25GSa/s. If the sampling needs to be modified this can easily be done using provided tools such as StellarIP or Vivado.

Chapter 4

Fluorescence Lifetime Measurements

In order to validate the implemented hardware system described in chapter 4, the fluorescence decay from the dye molecule Rhodamine 6G was measured. The lab setup used in the measurements is shown in figure 4.1. A mode locked laser oscillator was used as the light source. The laser produces laser pulses with a duration of 10fs at a repetition rate of 70.17MHz. The center wavelength of the laser is 790nm.

The short pulses of the laser induce two-photon absorption in Rhodamine 6G in a solution of methanol. The fluorescence from the sample at 550nm is detected by a fast avalanche photo diode (APD) and the electrical response from the APD is digitized at the rate of 1.25 GSa/s by a fast digitizer. A FPGA board and a PC then process the digitized data into a graph giving us easy access to the time trace of the fluorescence decay.

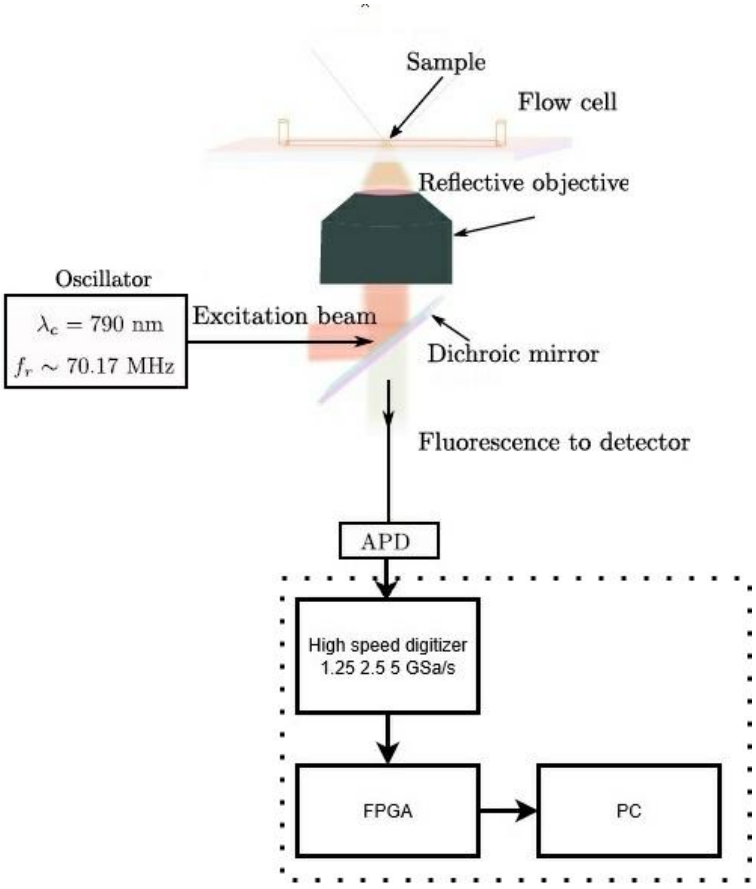

Figure 4.1: Illustration of the lab setup.

Figure 4.2 shows the averaged waveform, which contains 11 time traces of the fluorescence decay consisting of 196 data points. Observe that the first and last data point connect to each other if the data points are rearranged. Figure 4.3 shows the signal concatenated from the different traces in figure 4.2.

The y-axis of the figures are called amplitude since it is actually binary value representing a voltage span which in turn is related to the florescence by a transfer equation related to the APD's material and voltage settings. The y-axis values start at around 125. This is due to the value 127.5 is our DC voltage level. It is done this way since the ADC could be used to sample signal that have data both below and above this value.

The way negative numbers is represented in binary coding often involve setting one bit up as a flag telling us if the number is positive or negative. This reduces the available binary values, which could result in a bigger voltage span for each binary value or that we simply could not measure signal with as big peak to peak values. None of these options are good. So by setting the 0 value to the middle value of the range we avoid this.

The x-axis in figure 4.2 is the sample number presented according the digitization time, the axis in figure 4.3 is the amount of time of a period.

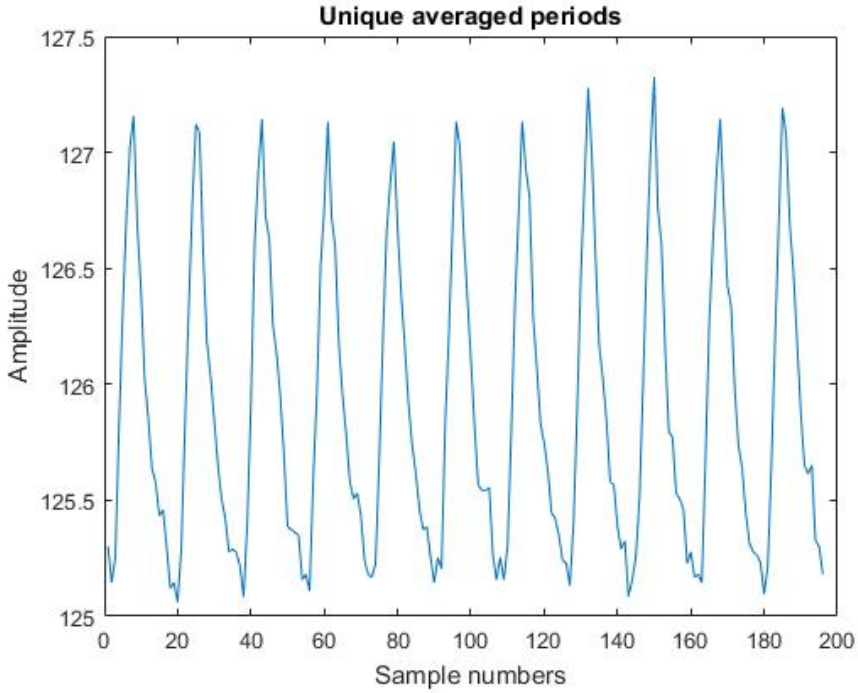

Figure 4.2: Waveform averaged samples from FMC125

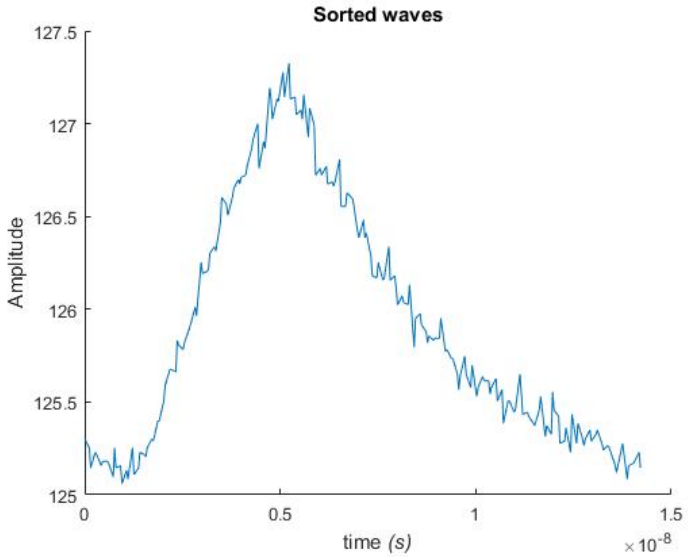

Figure 4.3: Sorted samples from FMC125.

As can be seen in the figure 4.3, the rise time of the signal is a bit slower than the instantaneous rise one expects from the excitation using a 10 fs pulse. The slow rise is mainly due to the limited bandwidth of the detector.

The signal also appears a bit noisy, which is mainly because of the small dynamic range of the ADC. In our system, we have used an 8 bit ADC with the binary encoding seen in the table in figure 4.4 taken from Analog Devices data sheet [20]. Other sources of noise include the error introduced by the various components like the ADP210 NEP, and transmission losses in the signal cable.

Despite this we see an exponential decay which compares well with the expected fluorescent lifetime of Rhodamine 6G. The result is achieved after sampling for 13.1064 μs taking 16k samples, and processed with the technique achieving an effective sampling speed of 13.75 GSa/s.

Differential Analog Input	Voltage Level	Digital Output			
		Binary MSB.....LSB Out-of-range		Gray MSB.....LSB Out-of-range	
+250 mV	>Top end of full scale + ½ LSB	11111111	1	10000000	1
+250 mV +248.05 mV	Top end of full scale + ½ LSB Top end of full scale – ½ LSB	11111111 11111110	0 0	10000000 10000001	0 0
+125 mV +123.05 mV	3/4 full scale + ½ LSB 3/4 full scale – ½ LSB	11000000 10111111	0 0	10100000 11100000	0 0
+0.976 mV –0.976 mV	Mid scale + ½ LSB Mid scale – ½ LSB	10000000 01111111	0 0	11000000 01000000	0 0
–123.05 mV –125 mV	1/4 full scale + ½ LSB 1/4 full scale – ½ LSB	01000000 00111111	0 0	01100000 00100000	0 0
–248.05 mV –250 mV	Bottom end of full scale + ½ LSB Bottom end of full scale – ½ LSB	00000001 00000000	0 0	00000001 00000000	0 0
< –250 mV	< Bottom end of full scale –½ LSB	00000000	1	00000000	1

Figure 4.4: Binary table.

Chapter 5
Future work

There are a number of ways in which the implementation presented in this thesis further can be improved.

One of the main things would be to find a better way of determining the repetition rate of the laser, which for instance could be done by implementing the frequency tracking method developed in the bachelor thesis Frequency Tracking Using Digital Cavities[22]. This would significantly reduce the error in $\Delta\tau$.

Another improvement that could help error reduction is the use of an ADC that uses more than 8 bits when digitizing the signal. This would reduce the voltage span corresponding to each binary value seen in figure 4.4, and thereby increasing the precision.

Implementing the above mentioned improvements would likely require a more powerful FPGA than KC705 board (see the utilization report appendix A) used in this thesis. As the current implementation utilizes about 76% of the Block Random Access Memories (BRAMs) resources, extra memory may be required for integrating the additional features.

One of the limitations of the current implementation is that the clock speed is limited to 200MHz or 800MHz, which slows down the processing of the data captured at the rate of 1.25GSa/s. A faster clock could enable continuous sampling and real time processing of the data. When considering the continuous sampling, replacement of the ADC might be needed since the FMC125 can only use burst sampling limiting the sampling time to a fixed period.

There is also a possibility to improve the IP provided by Xilinx. As they offer us a number of optimization schemes they are currently optimized for area (using as few resources as possible). Other optimization options could be investigated to determine an overal optimized implementation. There is also a possibility that the custom components can be optimized. This could be done, for example, by using Xilinx HDL where the algorithm is written in C/C++ and then converted to an IP consisting of VHDL or Verilog code.

Furthermore, the software implementation could be further improved. For example, additional filtering could be implemented to reduce the noise in the graphs. Figure 5.1 shows an example of this where the envelop functions that trace the maximal is displayed as a red line and the mean values using 22 waves are desplyed as a black line.

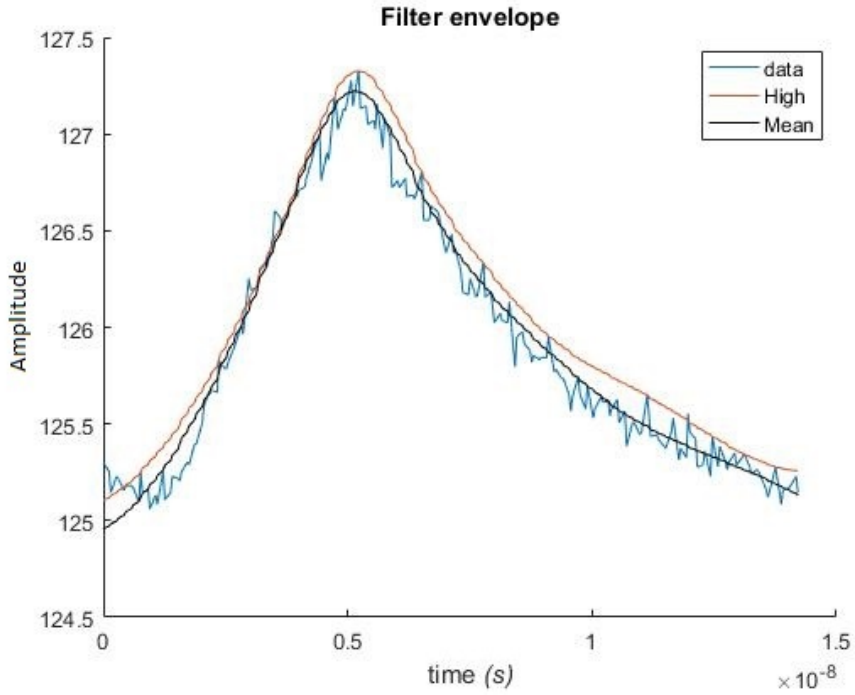

Figure 5.1: Filter example

Chapter 6

Acknowledgement

I would like to thank my supervisors Dr. Fredrik Edman, Dr. Liang Liu and Dr. Khadga Jung Karki, and all the people how have helped me at some point along the way Rakesh Gangarajaiah, Fredrik Zetterblom, Albert Llims, Arun Jeevaraj and all my family and friends.

References

[1] J. R. Lakowicz, "*Principles of fluorescence spectroscopy*," 2006. Springer.

[2] L. Marcu, "*Fluorescence lifetime techniques in medical applications*," vol. 40, pp. 304–331, 2012. Ann. Biomed. Eng.

[3] K. Suhling, L. M. Hirvonen, J. A. Levitt, P.-H. Chung, C. Tregidgo, A. L. Marois, D. A. Rusakov, K. Zheng, S. Ameer-Beg, S. Poland, S. Coelho, R. Henderson, and N. Krstajic, "*Fluorescence lifetime imaging (FLIM): Basic concepts and some recent developments*," vol. 27, pp. 3–40, 2015. Med. Photon.

[4] S. Coda, A. J. Thompson, G. T. Kennedy, K. L. Roche, L. Ayaru, D. S. Bansi, G. W. Stamp, A. V. Thillainayagam, P. M. W. French, and C. Dunsby, "*Fluorescence lifetime spectroscopy of tissue autofluorescence in normal and diseased colon measured ex vivo using a fiber-optic probe*," vol. 5, pp. 515–538, 2014. Biomed. Opt. Express.

[5] D. V. OConnor and D. Phillips, "*Time-correlated single photon counting*," 1984.

[6] I. Salmeen and L. Rimai, "*A phase-shift fluorometer using a laser and a transverse electrooptic modulator for subnanosecond lifetime measurements*," vol. 20, pp. 335–342, 1977. Biophys. J.

[7] E. Gratton and M. Limkeman, "*A continuously variable frequency cross-correlation phase fluorometer with picosecond resolution*," vol. 44, pp. 315–324, 1983. Biophys. J.

[8] S. S. Brody, "*Instrument to measure fluorescence lifetimes in the millimicrosecond region*," vol. 28, pp. 1021–1026, 1957. Rev. Sci. Instrum.

[9] J. G. C. Brown, "*Simplified system for the measurement of fluorescence lifetimes using the stroboscopic method*," vol. 34, pp. 414–415, 1963. Rev. Sci. Instrum.

[10] J. D. Pitts and M.-A. Mycek, "*Design and development of a rapid acquisition laser-based fluorometer with simultaneous spectral and temporal resolution*," vol. 72, pp. 3061–3072, 2001. Rev. Sci. Instrum.

[11] T. J. Pfefer, D. Y. Paithankar, J. M. Poneros, K. T. Schomacker, and N. S. Nishioka, "*Temporally and spectrally resolved fluorescence spectroscopy for the detection of high grade dysplasia in Barrett's esophagus*," vol. 32, pp. 10–16, 2003. Lasers Surg. Med.

[12] Q. Fang, T. Papaioannou, J. A. Jo, R. Vaitha, K. Shastry, and L. Marcu, "*Time-domain laser-induced fluorescence spectroscopy apparatus for clinical diagnostics*," vol. 75, pp. 151–162, 2004. Rev. Sci. Instrum.

[13] P. A. A. D. Beule, C. Dunsby, N. P. Galletly, G. W. Stamp, A. C. Chu, U. Anand, P. Anand, C. D. Benham, A. Naylor, and P. M. W. French, "*A hyperspectral fluorescence lifetime probe for skin cancer diagnosis*," vol. 78, no. 1231001, pp. 1–7, 2007. Rev. Sci. Instrum.

[14] A. J. Thompson, S. Coda, M. B. S. rensen, G. Kennedy, R. Patalay, U. Waitong-Brämming, P. A. A. D. Beule, M. A. A. Neil, S. Andersson-Engels, N. B. e, P. M. W. French, K. Svanberg, and C. Dunsby, "*In vivo measurements of diffuse reflectance and time-resolved autofluorescence emission spectra of basal cell carcinomas*," vol. 5, pp. 1–15, 2012. J. Biophotonics.

[15] V. G. Ivchenko, A. N. Kalashnikov, R. E. Challis, and B. R. Hayes-Gill, "*High-speed digitizing of repetitive waveforms using accurate interleaved sampling*," vol. 56, pp. 1322–1328, 2007. IEEE Trans. Instrum. Meas.

[16] National instruments, "*Time Interleaved Sampling.*" `http://zone.ni.com/reference/en-XX/help/370592N-01/digitizers/timeinterleavedsampling/`, September 2017.

[17] SenloSystems, "*APD210 High Sensitivit Avalanche Photodetector.*" `http://www.menlosystems.com/en/products/photodetectors/apd210/`, November 2017.

[18] Abaco systems, "*FMC125 FPGA Mezzanine Card.*" `https://www.abaco.com/products/fmc125-fpga-mezzanine-card`, September 2017.

[19] Abaco systems, "*FMC125 Datasheet.*" `https://www.abaco.com/download/fmc125-datasheet`, November 2017.

[20] e2v technologies(uk), "*EV8AQ160.*" `https://www.e2v.com/resources/account/download-datasheet/2291`, August 2017.

[21] A. DEVICES, "*EVAL-AD9517-1.*" `http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad9517-1.html`, October 2017.

[22] F. Zetterblom, "*Frequency Tracking Using Digital Cavities.*" `http://www.eit.lth.se/sprapport.php?uid=967`, October 2017.

Appendix A

Utilization Report

```
Copyright 1986-2015 Xilinx, Inc. All Rights Reserved.
----------------------------------------------------------------------------------------------------------------------
| Tool Version : Vivado v.2015.2 (win64) Build 1266856 Fri Jun 26 16:35:25 MDT 2015
| Date         : Wed May 31 15:18:03 2017
| Host         : OBAN running 64-bit Service Pack 1  (build 7601)
| Command      : report_utilization -file design_1_wrapper_utilization_placed.rpt -pb design_1_wra
| Design       : design_1_wrapper
| Device       : xc7k325t
| Design State : Fully Placed
----------------------------------------------------------------------------------------------------------------------

Utilization Design Information

Table of Contents
-----------------
1. Slice Logic
1.1 Summary of Registers by Type
2. Slice Logic Distribution
3. Memory
4. DSP
5. IO and GT Specific
6. Clocking
7. Specific Feature
8. Primitives
9. Black Boxes
10. Instantiated Netlists

1. Slice Logic
--------------

+----------------------------+-------+-------+-----------+-------+
|          Site Type         |  Used | Fixed | Available | Util% |
+----------------------------+-------+-------+-----------+-------+
| Slice LUTs                 | 12186 |     0 |    203800 |  5.98 |
|   LUT as Logic             | 11617 |     0 |    203800 |  5.70 |
```

LUT as Memory	569	0	64000	0.89
LUT as Distributed RAM	120	0		
LUT as Shift Register	449	0		
Slice Registers	12361	0	407600	3.03
Register as Flip Flop	12213	0	407600	3.00
Register as Latch	144	0	407600	0.04
Register as AND/OR	4	0	407600	<0.01
F7 Muxes	462	0	101900	0.45
F8 Muxes	15	0	50950	0.03

1.1 Summary of Registers by Type

Total	Clock Enable	Synchronous	Asynchronous
4	_	-	-
0	_	-	Set
0	_	-	Reset
0	_	Set	-
0	_	Reset	-
0	Yes	-	-
238	Yes	-	Set
5836	Yes	-	Reset
203	Yes	Set	-
6080	Yes	Reset	-

2. Slice Logic Distribution

Site Type	Used	Fixed
Slice	5640	0
SLICEL	3532	0
SLICEM	2108	0
LUT as Logic	11617	0
using O5 output only	2	
using O6 output only	9859	
using O5 and O6	1756	
LUT as Memory	569	0
LUT as Distributed RAM	120	0
using O5 output only	0	

using O6 output only	0			
using O5 and O6	120			
LUT as Shift Register	449	0		
using O5 output only	8			
using O6 output only	165			
using O5 and O6	276			
LUT Flip Flop Pairs	15727	0	203800	
fully used LUT-FF pairs	6812			
LUT-FF pairs with unused LUT	3551			
LUT-FF pairs with unused Flip Flop	5364			
Unique Control Sets	731			
Minimum number of registers lost to control set restriction	2235(Lost)			

3. Memory

Site Type	Used	Fixed	Available	Util%
Block RAM Tile	340	0	445	76.40
RAMB36/FIFO*	338	0	445	75.96
FIFO36E1 only	64			
RAMB36E1 only	274			
RAMB18	4	0	890	0.45
RAMB18E1 only	4			

* Note: Each Block RAM Tile only has one FIFO logic available and therefore can accommodate only o

4. DSP

Site Type	Used	Fixed	Available	Util%
DSPs	5	0	840	0.60
DSP48E1 only	5			

5. IO and GT Specific

Site Type	Used	Fixed	Available	Util%

Bonded IOB	167	167	500	33.40
IOB Master Pads	83			
IOB Slave Pads	82			
Bonded IPADs	0	0	50	0.00
Bonded OPADs	0	0	32	0.00
PHY_CONTROL	0	0	10	0.00
PHASER_REF	0	0	10	0.00
OUT_FIFO	0	0	40	0.00
IN_FIFO	0	0	40	0.00
IDELAYCTRL	5	0	10	50.00
IBUFGDS	0	0	480	0.00
GTXE2_COMMON	0	0	4	0.00
GTXE2_CHANNEL	0	0	16	0.00
PHASER_OUT/PHASER_OUT_PHY	0	0	40	0.00
PHASER_IN/PHASER_IN_PHY	0	0	40	0.00
IDELAYE2/IDELAYE2_FINEDELAY	72	72	500	14.40
IDELAYE2 only	72	72		
ODELAYE2/ODELAYE2_FINEDELAY	0	0	150	0.00
IBUFDS_GTE2	0	0	8	0.00
ILOGIC	72	72	500	14.40
ISERDES	72	72		
OLOGIC	0	0	500	0.00

6. Clocking

Site Type	Used	Fixed	Available	Util%
BUFGCTRL	14	0	32	43.75
BUFIO	3	0	40	7.50
BUFIO only	3	0		
MMCME2_ADV	1	0	10	10.00
PLLE2_ADV	0	0	10	0.00
BUFMRCE	4	0	20	20.00
BUFHCE	0	0	168	0.00
BUFR	6	0	40	15.00

7. Specific Feature

Site Type	Used	Fixed	Available	Util%
BSCANE2	2	0	4	50.00
CAPTUREE2	0	0	1	0.00
DNA_PORT	0	0	1	0.00
EFUSE_USR	0	0	1	0.00
FRAME_ECCE2	0	0	1	0.00
ICAPE2	0	0	2	0.00
PCIE_2_1	0	0	1	0.00
STARTUPE2	0	0	1	0.00
XADC	0	0	1	0.00

8. Primitives

Ref Name	Used	Functional Category
FDRE	6080	Flop & Latch
FDCE	5692	Flop & Latch
LUT6	4308	LUT
LUT5	2523	LUT
LUT4	2434	LUT
LUT3	2173	LUT
LUT2	1682	LUT
CARRY4	498	CarryLogic
SRL16E	473	Distributed Memory
MUXF7	462	MuxFx
RAMB36E1	274	Block Memory
LUT1	253	LUT
SRLC32E	241	Distributed Memory
FDPE	238	Flop & Latch
FDSE	203	Flop & Latch
RAMD32	196	Distributed Memory
LDCE	144	Flop & Latch
IBUFDS	79	IO
ISERDESE2	72	IO
IDELAYE2	72	IO
FIFO36E1	64	Block Memory
RAMS32	44	Distributed Memory
MUXF8	15	MuxFx
BUFG	14	Clock
SRLC16E	11	Distributed Memory
IBUF	6	IO
BUFR	6	Clock

IDELAYCTRL	5	IO
DSP48E1	5	Block Arithmetic
RAMB18E1	4	Block Memory
BUFMR	4	Clock
AND2B1L	4	Others
BUFIO	3	Clock
OBUFT	2	IO
BSCANE2	2	Others
OBUFDS	1	IO
OBUF	1	IO
MMCME2_ADV	1	Clock

9. Black Boxes

Ref Name	Used
fifo_generator_v12_0__parameterized1__3	1
fifo_generator_v12_0__parameterized1	1
fifo_generator_v12_0__3	1
fifo_generator_v12_0	1

10. Instantiated Netlists

| Ref Name | Used |
| dbg_hub_CV | 1 |

Printed by Tryckeriet i E-huset, Lund 2018

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2018-627

http://www.eit.lth.se