## **Stony Brook University**



# OFFICIAL COPY

The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.

© All Rights Reserved by Author.

## **Optoelectronic Fiber Based**

## **Fluorescence Detection Sensor Systems**

A Dissertation Presented

by

Vladislav A. Kuzminskiy

to

The Graduate School

in Partial fulfillment of the

Requirements

for the Degree of

**Doctor of Philosophy** 

in

**Electrical Engineering** 

Stony Brook University

**May 2007** 

#### STONY BROOK UNIVERSITY

#### THE GRADUATE SCHOOL

#### Vladislav A. Kuzminskiy

We the dissertation committee for the above candidate for the PhD. Degree, hereby recommend acceptance of this dissertation.

Serge Luryi, Advisor of Dissertation

Distinguished Professor, Department of Electrical & Computer Engineering

Mikhail Gouzman, Co-Advisor of Dissertation

Adjunct Professor, Department of Electrical & Computer Engineering

Wendy Tang, Chairperson of Defense

Associate Professor, Department of Electrical & Computer Engineering

Michael Gurvitch, Professor

Department of Physics & Astronomy

This dissertation is accepted by the Graduate School

Lawrence Martin,

Dean of the Graduate School

#### Abstract of the Dissertation

### **Optoelectronic Fiber Based**

## **Fluorescence Detection Sensor Systems**

by

Vladislav A. Kuzminskiy

**Doctor of Philosophy** 

in

**Electrical Engineering** 

Stony Brook University

2007

The goal of this Dissertation is study, design, and implementation of fiber based fluorescent sensors and sensing techniques. Novel techniques for Emission-Excitation fluorescent analysis and the time resolve method are offered, studied and presented. A novel type of Emission-Excitation Fluorimeter (EEF) with Linear Interference Filters (LIF) as the wavelength selective elements is proposed. The method of the selective detection for mixture of different fluorescent samples was developed. The limits in resolution LIF based EEF were found analytically and experimentally. Proposed the use of a single photon sensitive Time Resolve Spectroscopy (TRS) as an additional identification parameter to avoid a high dependence of EEF on the temperature.

A new technique of the direct decay time measurement was offered, studied and tested to make TRS free of any optical filters, based on proprietary patented Single Photon Time Interval (SPTI) method, which was tested as the main part of novel long-life oxygen and temperature sensors.

In the frame of this research a concept of FPGA core units (coreboards) was offered, allowing the use of the same set of hardware in lots of different applications. A few generations of FPGA core units were designed, prototyped, manufactured and tested. The concept of FPGA coreboards made possible to design a whole family of FPGA based SPTI control devices and perform series of experiments and tests.

A new flexible and low-cost rapid prototyping technology was developed and used. For fast and low cost optoelectronic prototyping a family of novel C-mountable devices was proposed, including fast modulated LED optical sources, 2-D adjustable holders, posts and mounts, dielectric and diamagnetic optical and electro-optical breadboards and many more.

Presented research was supported by several industrial and state grants. In the frame of the research 3 articles were submitted (1 already published), 3 new technology disclosures, 1 invited presentation (annual 2006 AVS meeting) and presentation on LIFT 2006 and annual 2007 "New LI Technologies" exhibition were made.

## LIST OF FIGURES

<b>Fig.1</b> . Experimental optical EEF setup with two LIFs6
<b>Fig.2.</b> Calibration curves for two LIFs used in the experiments
<b>Fig.3.</b> Spectral distribution of the excitation LIF8
<b>Fig.4.</b> Excitation and emission spectra of 10 <sup>-8</sup> M solution of Rhodamine 6G
obtained in the EE fluorimeter10
<b>Fig.5</b> . The EE spectrum of 10 <sup>-9</sup> M Rhodamine 6G
<b>Fig.6.</b> Cy3 dye emission spectrum measured by conventional and novel techniques12
<b>Fig.7.</b> EE spectra of CY2:CY3 mixtures a) 10:1; b) 1:6
Fig.8. Two-point time constant measurement technique
Fig.9. Schematic illustration of the integration technique used by Sholes
and Small (1980)22
<b>Fig.10.</b> Illustration of the balanced integration method
<b>Fig.11.</b> Illustration of linearized digital curve fit technique25
<b>Fig.12.</b> Time-gating diagram of operation
<b>Fig.13.</b> General layout of the sensor
Fig.14. Raw fluorescence output vs air atmospheric pressure and its
Stern-Volmer linearization (right) with the values of $\frac{N_0}{N} - 1$ scaled
vs O <sub>2</sub> partial pressure35
Fig.15. Mean-root-square error $\boldsymbol{\eta}$ of the measurements vs number of
counts $N_c$ and the function $1/N_c^{-1/2}$
<b>Fig.16.</b> Temperature – decay time characterization of popular Cr <sup>3+</sup> doped crystals40

<b>Fig.17.</b> Laboratory setup for temperature control of crystals with optical access	
to the testing ruby ball	42
Fig.18. Dielectric breadboard	45
Fig.19. Fast modulated LED light source and its PCB implementation	47
Fig.20. Fast modulated LED light source and its pin description	47
Fig.21. Small power supply for modulated LED light source	47
Fig.22. C-mount 2" holder	48
Fig.23. Two coordinate adjustable holder	49
Fig.24. Integrated 2-D position unit	50
<b>Fig.25.</b> Construction of a Fresnel lens from its corresponding asphere	51
Fig.26. Illustration of longitudinal spherical aberration	52
Fig.27. Illustration of the superior focus of an aspheric lens	54
Fig.28. Angle mounts	55
Fig.29. Electrical breadboard holder	55
Fig.30. Temperature sensor open space setup.	58
<b>Fig.31.</b> Plastic fiber 1:15 AMP/AMP differential splitter	60
Fig.32. Photon tight 1:15 FFF AMP/SMA combined compact differential	
splitter based on plastic fibers	60
Fig.33. Construction of photon tight 1:15 FFF AMP/SMA combined compact	
differential splitter	61
Fig.34. Glass 1:48 SMA-SMA compact photon-tight differential splitter	61
Fig.35. Design and implementation of the Ultimate Sensor's Fiber Bundle	
Splitter/Combiner	62

Fig.36. Actual detection system I	63
<b>Fig.37.</b> The schematics of FSG integrated with the light source	64
<b>Fig.38.</b> Implementation of FSG integrated with the light source	64
Fig.39. Mobile test bench.	65
Fig.40. Actual detection system II	66
Fig.41. Schematic element for FPGA coreboard.	69
Fig.42. Time Control Unit schematic. First prototype.	71
Fig.43. Time Control Unit. Print board layout.	73
Fig.44. Time Control Unit (TCU)	73
Fig 45. Connection diagram for Master Slave Serial Configuration	77
Fig.46. Footprint and pinout for VQ100 package of Xilinx FPGA	
and actual view of the chip XC3S200	79
<b>Fig.47.</b> FPGA based control unit schematic. Generations 1-3	80
<b>Fig.48.</b> Schematic of the generation 4 and 5 coreboards	81
<b>Fig.49.</b> FPGA Core unit layout. Generation 3	82
<b>Fig.50.</b> LPKF ProtoMat C30S. Making top layer of FPGA core unit	82
Fig.51. Three first generations of FPGA coreboards	84
<b>Fig.52.</b> Fourth generation of the FPGA coreboard	85
Fig.53. Fifth generation of FPGA coreboard	85
<b>Fig.54.</b> Designing 5 <sup>th</sup> generation FPGA coreboard's printboard	86
Fig.55. Top layer with the soldermask, bottom layer with the soldermask.	
5 <sup>th</sup> generation coreboard manufacturing file set	87

Fig.56. Fully adjustable control signal sequence generator with indication
of control signal parameters and counting results87
Fig.57. Fully adjustable control signal sequence generator with indication
of control signal parameters and counting results88
<b>Fig.58.</b> Motherboard and front panel boards layouts
<b>Fig.59.</b> Main screen of ISE software complex90
<b>Fig.60.</b> Pros and contras of MP and FPGA, and the benefits of their co-use91
<b>Fig.61.</b> Schematic of the use of Basic Stamp 2 microcontroller92
<b>Fig.62.</b> FPGA + BasicX system mounted on an evaluation circuit board93
<b>Fig.63.</b> Microcontroller Basic Stamp 294
<b>Fig.64.</b> The use of FPGA and MP combination in the project94
<b>Fig.65.</b> Portable Single Photon Counter96
<b>Fig.66.</b> Decaying fluorescence and single photon detector
Fig.67. Pressure dependence of fluorescent intensity of the ruthenium-complex
probe in air100
<b>Fig.68.</b> Graphical interpretation of τ-meter concept
<b>Fig.69.</b> Graphical interpretation of τ-meter error
<b>Fig.70.</b> Graphical presentation of the equal region simulation
<b>Fig.71.</b> Graphical presentation of the simulation
<b>Fig.72.</b> Compact integrated measurement system
Fig 73 Temperature – decay time characterization of ruby crystal

## TABLE OF CONTENTS

LIST OF FIGURES	V
INTRODUCTION	1
1. FLUORESCENT DETECTION OVER FIBER SENSOR NETV	VORK 3
1.1. PRINCIPLES OF FLUORESCENT DETECTION	6
1.1.1 Scattered light consideration	9
1.1.2 Data collection and processing	9
1.2 EXPERIMENTAL RESULTS. EXPERIMENTAL EE SPECTRA	12
1.3 EVALUATION OF THE EFFICIENCY OF THE OPTICAL SETUP	14
1.3.1 Throughput of the excitation arm	14
1.3.2 Throughput of the detection arm	14
2. DECAY TIME CHARACTERIZATION	19
2.1 Two-point time constant measurement	19
2.2 Integration method	22
2.3 DIGITAL CURVE FIT METHOD	25
2. OXYGEN SENSOR	27
3.1 Principle of operation	29
3.2 System overview	33
3.3 Tests and results	34
4. DESIGN AND IMPLEMENTATION OF THE TEMPERATURE	SENSOR 39
A 1 SELECTION OF THE SENSITIVE SUBSTANCE	40

4.3. N	YEW COMPONENTS FOR RAPID PROTOTYPING. NOVEL PROTOTYPING TECHNIQUE.	42
4.3	.1. Fast modulated LED light source	46
4.3	.2 Adjustable holders and optical components.	48
6. DESI	IGN, IMPLEMENTATION AND TESTING OF THE DECAY TIME	
MEASU	URING SYSTEM	67
6.1 A	NALOG TIME INTERVAL CONTROLLER.	71
6.2 FI	PGA IMPLEMENTATION.	74
6.3 FI	PGA BOARD GENERATIONS	83
6.4. N	IICROPROCESSOR AND FPGA	90
6.5. P	ORTABLE SINGLE PHOTON COUNTER.	95
7. TII	ME INTERVAL MEASUREMENTS	96
7.1.	EXPLANATIONS OF TIME INTERVAL MEASURING TECHNIQUE.	100
7.2.	MATHEMATICAL MODEL FOR A 3 PHOTON TECHNIQUE	102
7.2.	MATHEMATICAL MODEL FOR A 3 PHOTON TECHNIQUE	103
7.2.	GENERATOR – SINGLE PHOTON SOURCE IMITATOR	105
7.4.	FULL INTEGRATED IMPLEMENTATION – "THREE PHOTON TIME INTERVAL	
METE	R/INTEGRATOR"	108
CONCI	LUSION	111
BIBLIC	OGRAPHY	113
APPEN	DICES	125
Prog	RAM CA7. CONTROL UNIT FOR FLUORESCENT THERMOMETER.	125
Prog	RAM TIM1. BASIC TIME INTERVAL METER	135

PROGRAM CC09. IMPLEMENTATION OF "INTERVAL+COUNTER" CONCEPT, GENERATO	R
SIMULATOR FOR DACAY FLUORESCENT PROCESS.	144
PROGRAM PCC03. PORTABLE SINGLE PHOTON COUNTER.	158
PROGRAM BS2_FPU. BASICSTAMP2 WITH FLOATING POINT UNIT. MICROPROCESSOR	
PART OF THE TEMPERATURE MEASUREMENT SYSTEM	173
PROGRAM BSX_02. BASICX+FPGA TAU MEASUREMENT SYSTEM	185
BASIC PATENT	192

#### Introduction

In the last two decades the field of fluorescent spectroscopy enjoyed remarkable growth. No longer a mere research tool, it has given rise to a variety of convenient instruments used in many biological applications. This growth has relied on the explosion in data processing techniques and the extensive development of fluorescent probes.

The conventional spectra of fluorescence are either a plot of the intensity of fluorescence of an analyte excited by the light of a fixed wavelength as a function of the emission wavelength, or a plot of fluorescence intensity received within a fixed spectral band as a function of the excitation wavelength. These emission and excitation spectra are special cases of a more general three-dimensional representation of fluorescence intensity as a function of both the excitation and the emission wavelengths. The resultant dual excitation-emission dependency enables spectral characterization of an object simultaneously with respect to both its excitation and emission. The method offers a fundamental improvement in the sensitivity and reliability of detection, especially when multiple fluorophores are present in the same location.

The three-dimensional representation of fluorescence even from a single-pixel object requires copious amount of data. If either of the two conventional fluorescence descriptions takes n data points, the EE needs  $n^2$  points. This necessitates a significant increase in the data acquisition rate, the processing time and the computer memory. This is probably why all existing devices for the excitation-emission (EE) analysis are limited to a single-pixel mode (the analyzed object may be macroscopic but uniform over its volume, e.g., a well-mixed sample in a cuvette).

The applications of EE method can be roughly divided in two groups – environmental monitoring tools and bio-medical equipment. In the environmental studies, the technique has been widely used for detection of endogenous fluorescence of various chemicals. The bio-medical EE equipment can, in turn, be divided into *in vitro* and *in vivo* types. *In vitro* devices are similar to the environmental equipment, whereas *in vivo* devices employ fiber-optic endoscopes for excitation and emission signal delivery to and out of the tissue under study.

#### 1. Fluorescent detection over Fiber sensor network

Typically, the EE systems employ two monochromators as the dispersion elements for both the excitation and the emission light. The first monocromator steps up the excitation wavelength. The emission spectrum is scanned with the second monochromator while keeping the excitation wavelength constant, [1, 2, 3]. Most commercial EE systems are based on such two-monochromator design and suffer from relatively slow scanning and data acquisition. Significantly shorter EE spectra acquisition times (minutes) can be achieved with a device comprising a monochromator for dispersion of the excitation light and a spectrograph to capture the emission spectra, such as an in-vivo system for interrogation of oral cavity mucosa [4]. Very short acquisition times (seconds) have been demonstrated in experimental spectro-fluorimeters with the single-measurement EE fluorimetry [5, 6] where two spectrographs were used both for excitation and emission detection from a uniform sample. In this system, a broadband light source irradiated the entrance slit of the excitation spectrograph to spatially disperse the emerged light beam across the sample. The emitted fluorescence was gathered on the entrance slit of the second spectrograph rotated by 90° with respect to the exit slit of the excitation spectrograph. The 2D fluorescence data were detected by a CCD or CMOS arrays. A drawback of the technique is that it sets very stringent requirements to both excitation and detection optics. To the best of our knowledge, there was an attempt to market one such spectro-fluorimeter [7]. However, when we inquired about the availability of this device, we were told that it had been removed from the market due to technical problems.

Beside spectral decomposition of broadband light, other methods of tunable excitation have been reported. Often lasers are used as the sources for multiwavelength excitation. For example, a nitrogen-dye lasers combination as a tunable excitation source has been used in an *in-vivo* endoscope device for gastrointestinal studies [8]. Another sophisticated *in-vivo* endoscope system employs a N<sub>2</sub> laser that pumps a sequence of dyes placed on a rotating wheel [9]. Yet another laser-activated tunable wavelength excitation source for pollution analysis [10] employs the effect of frequency doubling and stimulated Raman scattering of a Q-switched Nd:YAG laser to generate many beams of various wavelengths that are separated with a system of prisms and launched into separate optical fibers for conveyance to the sample. In another EE setup, a group of marine scientists used a tunable optical parametric oscillator (OPO) as a tunable excitation source for macroalgae research [11]. An original solution was found by NRL scientists [12] who used for tunable excitation in their EE assembly a set of seven lightemitting diodes (LED) within the range between 370 and 636 nm.

These latter EE systems are experimental and custom-made for specific applications. All known commercial EE fluorimeters are based on two monochromators and require two-dimensional scanning. Implementation of the EE method in practical fluoroscopy has been rather narrow so far. In most publications, the technique has been limited to the identification and differentiation of auto-fluorescence. In spite of its high fluorescence-identification ability, it has not yet found its golden application in biological fluorescent-detection instruments that differentiate objects labeled with a variety of fluorescent markers. We believe the fluorescence-detection industry shies away from EE fluorescence for three reasons – price, complexity, and weight/size.

An inexpensive, robust and compact emission wavelength disperser combined with a broadband or tunable excitation source would stimulate a broader utilization of the EE technique for mainstream fluorescent detection instrumentation. It would be very attractive to employ electronically tunable filters. There are several options available today: acousto-optic tunable filters (AOTF) [13, 14], liquid crystal (LCTF) [15, 16] and surface plasmon [17] tunable optical filters. All these experimental instruments can perform in the EE setup (e.g., the experiments with the EE dye identification were carried out with two AOTF, one for excitation and another for emission [13]). However, in addition to their specific performance flaws (e.g., high power consumption for AOTF and very low transmission coefficient for LCTF) these devices depend on a precise modulation of resonant cavities by piezoelectric spacers and hence require temperature stabilized environment.

#### 1.1. Principles of fluorescent detection

The schematics of the bench-top EEF is shown in Figure 1. The key feature of the setup is that its light-dispersive elements are Linear Interference Filters whose transmittance wavelength varies linearly over the length of the filters. Linear interference filters are manufactured by Schott and are available from Edmund Optics, Barrington NJ. The LIF spectral range covers both the visible and the IR regions (400-700 and 400-1000 nm ranges are available) with the light transmission coefficient of 30 to 45% and the extinction factor for the wavelengths other than in the transmitted range ~10<sup>4</sup>. The LIFs come in two sizes, 2.5×6 and 2.5×20 cm², and weigh, respectively, 22 and 75 g. In present experiments, two filters with 2.5×20 cm² dimensions were chosen to scan the 300-700 nm spectral range.

The excitation arm of the fluorimeter (Figure. 1) consists of a broadband light

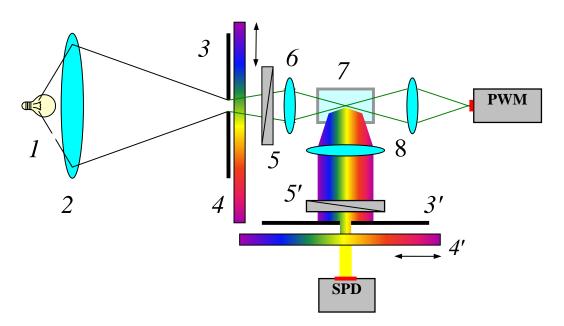


Fig. 1. Experimental optical EEF setup with two LIFs

source 1 (halogen lamp) imaged on a diaphragm 3 with a wide-angle objective 2 to illuminate a quasi-monochromatic region on the excitation LIF 4. The exiting excitation beam of  $\sim 13$  nm bandwidth is focused into the sample cuvette by an additional lens 6 placed in front of the sample 7. A spectrophotometer and/or an optical power meter PWM are used to monitor intensity and spectrum of the excitation light. The detection arm includes a wide-angle light condenser 8 that directs the quasi-parallel beam on the detection diaphragm 3'. The beam transmitted through the detection LIF 4' is directed to the SPD optical entrance. In addition, two crossed polarizers 5 and 5' are installed in the emission excitation and arms for attenuation of the scattered light.

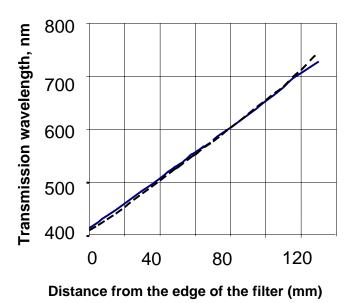


Fig. 2. Calibration curves for two LIFs used in the experiments

The spectral tuning of both the excitation and detection LIFs is achieved by a mechanical shift of the excitation filter relative to the excitation diaphragm and subsequent mechanical scan of the detection filter relative to the emission diaphragm.

The motion and positioning of both filters was regulated by two computer-controlled linear step motors.

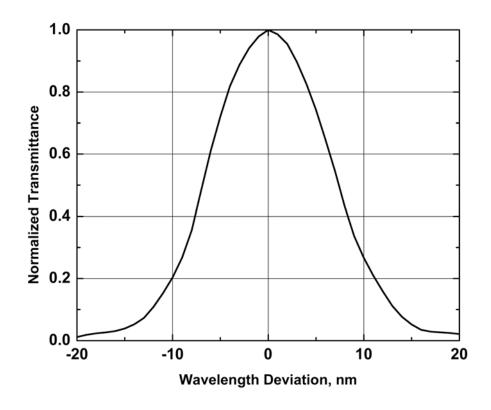


Fig. 3. Spectral distribution of the excitation LIF

Spectral calibration of both LIFs was performed with a spectrophotometer (SP-2000, Ocean Optics, Dunedin, FL). Figure 2 shows the calibration curves for the excitation and detection LIFs. An example of the spectral distribution of the excitation beam of the first LIF tuned to a wavelength  $\lambda = 525$  nm is shown in Figure. 3. Spectral width of the transmitted band varies from 12.2 nm at the blue edge of the filter (410 nm) to 16 nm at the red edge (720 nm) with an approximately constant halfwidth of 13 nm in the region between 430 and 580 nm. We found that the spectral band of the transmitted light was virtually independent on the diaphragm width below 2 mm so that 2-mm-wide diaphragm

was chosen to provide the maximum transmitted intensity while keeping the spectral width at minimum.

#### 1.1.1 Scattered light consideration.

In the present setup, the relatively wide spectral width of the excitation beam creates a well-known problem of scattered excitation light that interferes with the fluorescent signal in the detection arm. The scattered light is mostly generated by the Raleigh scattering of the excitation light on the solvent molecules in the sample cuvette, by impurities dispersed in the solvent or by high-concentrations dye. To minimize the intensity of scattered light in the detection channel, a system of two crossed polarizers with the contrast ratio of  $\sim 10^2$  each was installed as shown in Figure 1. In practice, the scattered signal does not influence the measurements of the fluorescent spectrum with the exception of a narrow region where excitation and emission wavelength are very close. The effect becomes more significant at very low concentration of dyes when the fluorescent light intensity becomes comparable with the intensity of scattered light.

The scattered light was accounted for by our software as described in the next section.

#### 1.1.2 Data collection and processing.

The data collection was controlled by a PC based control module. The module initiates a series of measurements of the fluorescence intensity by selecting the emission and excitation wavelength bands for each measurement in accordance with a specified data collection scheme that determines how the samples of the spectra will be distributed across the excitation-emission plane. The data collection schemes fall into two categories:

with uniform sampling and with non-uniform sampling. While uniform sampling is useful when no *a priori* knowledge about EE spectrum is available, several optimized schemes with reduced number of sample points and data collection times can be applied

information exists allowing for an appropriate choice of non-uniformly sampled data. For example, sampling density can be increased in those areas of excitation-emission plane where the magnitude of the

some

priori

when

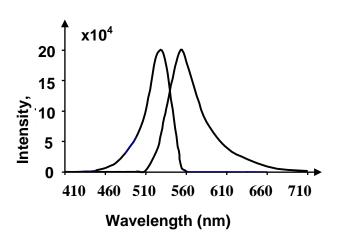


Fig. 4. Excitation and emission spectra of 10<sup>-8</sup> M solution of Rhodamine 6G obtained in the EE fluorimeter

data gradient exceeds a certain level.

A set of measurements resulting from the experiment is processed by the dataprocessing software. To improve data quality and simplify data presentation, the data
undergoes re-sampling and deconvolution. The re-sampling positions the data at the
nodes of a regular grid on the excitation-emission plane while choosing the grid size to
incorporate the finest features of the EE spectrum. The new data are obtained by
approximating the region of the experimental EE spectrum with spline functions and
evaluating EE values at the new nodes based on the approximation. The deconvolution
procedure (2D modification of the Jansson method [18]) is applied to the re-sampled data
to reduce "blurring" due to the relatively wide band-pass of the LIFs. On the basis of the
experimentally measured "blurred" data and known characteristics of the excitation and

emission filters the estimates of the reconstructed "deblurred" data are generated. The iterative procedure optimizes the estimates by minimizing the mean square deviation of the convolution with the filter characteristics from the original 'blurred' data.

One of the functions of the software was to account for the scattered light. Its intensity is measured at the edges of the ( $\lambda_{excit}$ ,  $\lambda_{emiss}$ ) two-dimensional space where the excitation wavelength is sufficiently far from the emission wavelength and the fluorescent signal is substantially weaker then the scattered signal. We found that the amplitude of detected scattered signal increases with the wavelength in sync with the spectral characteristic of the lamp (linear in the 400-700 nm range) and has a halfwidth corresponding to the halfwidth of the excitation beam. Armed with these facts, we could interpolate the shape and magnitude of the scatter into the area of overlapped emission and scattered light signals. For each data point, the magnitude of scattered intensity is calculated and subtracted from the measured total light intensity at a given spectral point.

#### 1.2 Experimental results. Experimental EE Spectra.

The performance of the setup in detecting fluorescence markers was studied with a number of fluorescent dyes dissolved in water and placed in a  $1\times1\times1$  cm<sup>3</sup> sample cuvette. Figure 5 exhibits conventional excitation and emission spectra of  $10^{-8}$  M solution of Rhodamine 6G (Exiton, Dayton OH) measured in our setup.

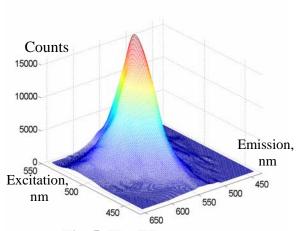


Fig. 5. The EE spectrum of 10<sup>-9</sup>M Rhodamine 6G

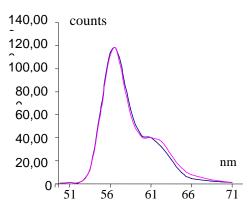
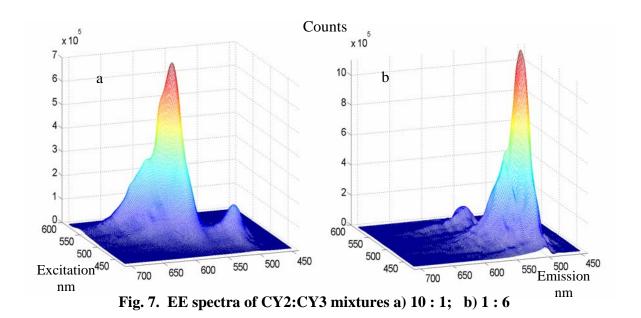


Fig. 6. Cy3 dye emission spectrum measured by conventional (----) and novel (----) techniques.

Comparison with the standard curves obtained with a monochromator showed no significant differences in the spectra. The 3D-image of EE spectrum obtained using a 10<sup>-9</sup> M solution of the same dye is shown in Figure 5. A scan of the Cy3 dye (Amersham Biosciences) emission spectrum (Figure 6) reveals all the spectral details, only slightly smoothed in comparison with the monochromator scan of the same dye excited by laser. Thus, the transition from the narrow-band excitation/detection scheme (laser + monochromator) to the 13 nm bandwidth is not detrimental to the spectral resolution. To demonstrate the ability of our method to distinguish between different dyes of strongly

varied abundance, we experimented with mixtures of two dyes. Figures 7 (a, b) display the two dimensional EE spectra obtained with mixtures of Cy2 and Cy3 dyes with concentration ratios of 10 to 1 (a) and 1 to 6 (b), respectively. In spite of the significant differences in concentration both dyes are distinctly separated in the plots and can be easily identified and quantified. This experiment demonstrates the power of the EE technique. A conventional fluorescence scan could not detect the presence of the lower concentration dye in Figure 7a.



#### 1.3 Evaluation of the efficiency of the optical setup

The main goal of the experiments described in previous sections is to demonstrate the feasibility of the LIFs as the light dispersing elements in the fluorimeter assembly rather than the optimization of its optical throughput. However, it is important to estimate the total losses in the system and determine where the heaviest losses occur. The optical throughput of the system is the product of losses at various elements in the optical path. The fluorescent response of the system also factors in the efficiency of the fluorescent excitation. In this section we evaluate the losses and throughput and compare them with the experimental results.

#### 1.3.1 Throughput of the excitation arm.

Elements of the excitation optics (Figure 1) include the halogen lamp, wide-angle optical condenser, diaphragm to fit the excitation beam into a rectangular shape, excitation LIF, polarizer and the focusing lens. We calculated parameters necessary to estimate the optical loss such as stereographic angle of light collection by the condenser and the ratio of the diaphragm area to the area of the source image on the diaphragm. With known values of the LIF transmission coefficient, polarizer transmission factor, and reflections from the air-glass optical boundaries, we estimated the throughput of the excitation arm  $K_{exc}$  to be equal to  $4.2\times10^{-4}$ . This is similar to the throughput of a single monochromator [19].

#### 1.3.2 Throughput of the detection arm.

Elements of the detection optics include condenser, detection diaphragm, detection LIF, detection polarizer, and the SPD. We calculated parameters necessary to

estimate the optical loss such as stereographic angle of the fluorescent collection, the ratio of the diaphragm area to the area of the fluorescent image on the diaphragm, ratio of the SPD input area to the fluorescent beam cross-section. Knowing the LIF transmission factor, the transmission of the detection polarizer, the SPD efficiency, and reflections from the air-glass boundaries, we estimated the throughput of the detection arm  $K_{det}$  to be equal to  $4\times10^{-6}$ . The losses in the detection arm exceed those in the excitation arm by two orders of magnitude; one order of magnitude was added by the additional lossy geometry factors (smaller stereometric collection angle, ratio of the SPD input area to the fluorescent beam cross-section), another factor of ten is due to the 10% SPD efficiency.

To reconcile the calculated efficiency with the experimental results we have to factor in the efficiency of the fluorescent excitation. Considering, for example, the experiment with the  $10^{-9}$  M Rhodamine sample (Figure 5) with the peak of the measured fluorescent signal  $N_F \sim 20,000 c/s$ , we can compare it with the value calculated from known parameters of the dye and optical throughput of the setup. The efficiency of the fluorescent excitation, that is, the ratio of the number of fluorescent photons per second  $N_F$  emitted from an observed volume to the number of excitation photons per second  $N_I$  incident to the volume can be calculated from the following expression [20]:

$$N_{F} / N_{I} = \Phi \cdot \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{d} \tag{1}$$

Here  $\Phi$  is the quantum yield (close to unity for Rhodamine 6G [21]); n is the dye density  $(6\times10^{11}~\text{cm}^{-3}~\text{for}~10^{-9}~\text{molar concentration of Rhodamine 6G)}$ ;  $\sigma$  is the absorption cross-section in cm<sup>2</sup> ( $\sigma = 3.8\times10^{-21}\times\epsilon$ , where  $\epsilon$  is the dye molar extinction coefficient ( $\epsilon = 10^5$  for Rhodamine 6G at the maximum of excitation spectrum [22]); d is the length of the observed volume (volume of fluorescent radiation gathered by the lens and limited by the

diaphragm in the detection arm) along the direction of excitation beam. Substituting all parameters, we find  $N_F$  /  $N_I = 5.2 \times 10^{-5}$  and the expected fluorescent response of the system  $R_F$  is the product  $K_{exc} \times K_{det} \times N_F$  /  $N_I = 4.2 \times 10^{-4} \times 4 \times 10^{-6} \times 5.2 \times 10^{-5} = 8.3 \times 10^{-14}$ .

The number of the incident excitation photons was estimated as following. We calculated that the lamp radiated  $\sim 0.18W$  within the 13 nm linewidth band, or, translating into the number photons (at 550 nm),  $N_{550}$ ,  $\sim 5\times10^{17}$  ph/s. Multiplying it by the fluorescent response  $R_F$  we obtain the expected signal as  $5\times10^{17}\times8.3\times10^{-14}=4\times10^4$  counts/s, a remarkably good agreement with the 20,000 c/s measured experimentally. This result ascertains that the estimation of the EEF efficiency accounts for all major losses.

Results obtained on our simple experimental setup indicate that it is feasible to implement an inexpensive, lightweight and robust LIF-based fluorimeter. The lion share of losses comes from the poor collection of the excitation light and the fluorescence by the inefficient optics. Nevertheless, the assembly demonstrated its ability to confidently detect and quantify the EE spectra of dyes with the molar concentration down to 10<sup>-9</sup> and to quantitatively distinguish dyes with vastly different concentrations present at the same location. We believe that several easily achievable improvements in the system such as using light sources with higher luminosity, maximizing the diaphragm area, increasing the collection angle of excitation and detection condenser, and fitting the shapes of the light source and the fluorescing volume to the geometry of excitation and detection diaphragms, etc., can increase its sensitivity by at least two orders of magnitude

Our system employs two LIFs, one for creating tunable excitation and the other for analysis of the fluorescence. The operation of the EEF thus requires coordinated mechanical scanning of both filters. In the present design both LIFs are moved on rails by two linear step motors In future designs, to insure the longevity of moving parts it will be prudent to use frictionless magnetic floating platforms with programmable linear motion systems.

To reduce or completely eliminate mechanical scanning, the fluorimeter should employ an alternative source of tunable excitation while still relying on the detection LIF for analysis of fluorescence. In our view, one of the promising sources of tunable excitation is an optical parametric oscillator (OPO) whose advantages include wide tuning range, ease of tuning, narrow line width and directionality of the light beam. A single-scan device will contain only one LIF in the fluorescent arms subjected to mechanical scanning. The detection can be realized with an SPD located behind the LIF. A no-scan device will contain no moving parts. In this device, to accommodate simultaneous detection at various wavelengths, an extended fluorescent source will be formed by the OPO beam piercing the cuvette along its length. The resulting fluorescent trace will be projected by the detection optics on the detection LIF. For the detection of the decomposed fluorescence, a linear light-sensitive array (SPD, CCD or CMOS) can be placed in close proximity to the LIF.

A bench-top prototype of an excitation-emission fluorimeter based on the commercially available linear interference filters as light-dispersion elements was developed and the feasibility and performance of the linear interference filters was tested. The performance of the fluorimeter was verified by obtaining two-dimensional EE

spectra of several fluorescent dyes of different molar concentrations and their mixtures. In the course of this work we spectrally calibrated the filters and found the algorithm to account for the scattered light. We optimized the spectral characteristics of the pseudo-monochromatic excitation and developed a data processing for re-sampling and deconvolution techniques that minimizes the data blurring due to the finite wavelength range of the excitation.

#### 2. Decay time characterization

The mathematical methods we are using in the data processing are clearly described in different articles[23].

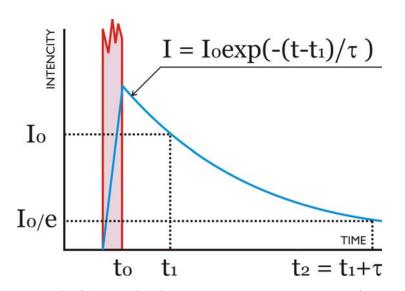
The common feature of the schemes in this category is that the excitation light applied to the fluorescent material is a high intensity 'delta' function pulse (e.g. a laser pulse or that from a flash lamp) or a rectangular pulse, and the measurement is derived from the observation of the fluorescence decay after the removal of the excitation light. The following are given as outlines of some typical schemes of this type which have been used in thermometry applications.

#### 2.1 Two-point time constant measurement

This is a very straightforward method which was used by several groups of workers in the early stage of the development of a fluorescence-based sensor system. The fundamental principle of this technique can be illustrated as is shown in Figure 8. The approach is to compare the intensity level at two points along the exponential decay curve after the excitation pulse has terminated. The circuitry employed to do this is designed to measure first the value of decaying signal, Io, that occurs at a fixed time, t1, after the termination of the excitation pulse. A second intensity level, of value Io/e, is then calculated and established as a reference. When the decaying signal falls to that level, the time,  $t_2$ , at which 'crossover' occurs is noted. The interval between  $t_2$  and  $t_1$  is the time constant, I, of the exponentially decaying signal. In most cases, the fluorescence decay process observes a single or 'quasi-single' exponential law, such as that shown in Figure 8. Thus the time constant,  $\tau$ , may then be used as the measure of fluorescence lifetime.

Timing circuits are provided for measuring precisely the time between  $t_2$  and  $t_1$ . That time difference can be correlated directly with the temperature of the fluorescent sensor by reference to an empirically deter-mined 'look up' table stored digitally within the instrument.

This type of method is simple and inexpensive in relation to the elec-tronic components used. Since the fluorescence signal is measured after the excitation pulse is



 $\label{prop:constant} \textbf{Fig. 8.Two-point time constant measurement technique} \; .$ 

over, the detector optics do not have to be designed to discriminate strongly against stray signals from the excitation source. However, a significant disadvantage of this type

of system is that the signal is only measured at two specific times and, as a result, precision is greatly limited.

This two-point measurement technique was used in the system reported by Wickersheim and Sun where a lamp phosphor, tetravalent manganese activated magnesium fluorogermanate, was incorporated as the fluorescent sensor itself. With a xenon flash lamp used as the excitation source, the observed fluorescence lifetime ranges from approximately 0.5 ms at  $450^{\circ}$ C to more than 5 ms near the liquid nitrogen boiling temperature (-196°C). It is reported that an accuracy of  $\pm 2^{\circ}$ C over the whole range can be

achieved without calibration of the instrument. With a single point calibration,  $\pm 0.2^{\circ}\text{C}$  accuracy is claimed.

#### 2.2 Integration method

To achieve higher precision from the pulse measurement approach, several techniques have been developed which are based on the integration of the decaying fluorescence signal over different periods of time. One example of this is the signal processing scheme used by Sholes and Small in their early non-fiber study of ruby fluorescence decay. As illustrated schematically in Figure 9, when the decaying fluorescence falls below a preset level, the start of the measurement process is triggered. The signal is integrated at two fixed delay times,  $T_1$  and  $T_2$ , and then the integration values over these periods of time, A and B are sampled. After the signal has decayed to zero, the integrator is reset and restarted. Integrated noise and dc offset levels are then sampled for the same two fixed delays, and are given by C and D, which are equivalent to the noise and dc offset components in A and B respectively. Therefore,

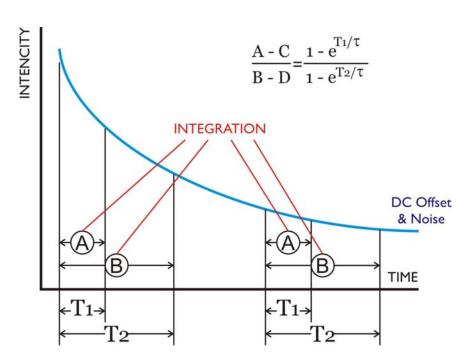


Fig. 9: Schematic illustration of the integration technique used by Sholes and Small (1980)

the fluorescence lifetime,  $\tau$ , can be obtained by solving the following implicit relation.

Another example is the balanced integration method described by Sun. This technique was designed to achieve  $0.01^{\circ}$ C resolution using tetravalent manganese activated magnesium fluorogermanate, the same sensor material which had been used with the two-point measurement by Wickersheim and Sun as mentioned above. The system utilizes a dual-slope integrator to balance the integrals of two sequential areas under the decay curve of the phosphor, as shown in Figure 10. The first integration is carried out over a selected fixed time interval. Integration starts at a predetermined time,  $t_1$ , after the excitation pulse has terminated. Upon completion of this initial integration, i.e. at time  $t_2$ , the polarity is reversed and negative integration (deintegration) begins. This continues until the second integrated area exactly equals the first and the combined integral equals zero. The time of the zero crossing,  $t_3 - t_4$ , is measured with high resolution (to <20ns). The observed value of  $t_3$  can then be related to the decay time of the phosphor and hence to its temperature.

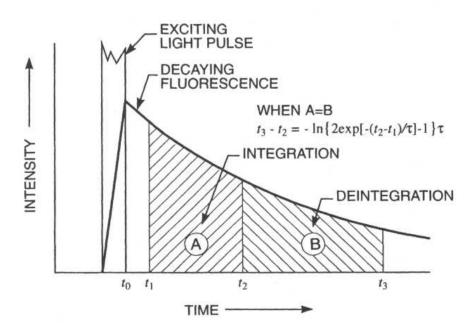


Fig 10. Illustration of the balanced integration method

The result of this balanced integration system is very sensitive to any background signal, whether optical or electronic. Hence ac noise must be minimized and any dc offsets must be measured and corrected, in a way similar to that used by Sholes and Small and illustrated in Figure 9. It should be also be pointed out that the dynamic range of this system is fairly narrow (~100°C for a given choice of the initial integration period), relative to that of the less precise two-point system.

# 2.3 Digital curve fit method

Recently Luxtron has introduced a modular system, WTS-11, designed to monitor winding temperatures in power transformers, utilizing a tech-nique which takes advantage of the newest, commercially available, high-speed digital signal processors (DSPs), and it has been described in detail by Sun who termed it the 'digital curve fit method.'

In this system, an unnamed fluorescent material is used, the excitation spectrum of which decaying luminescent signal results. A selected portion of each decay curve is digitized, after the detected signal has passed through a low noise, wide bandwidth amplifier. The digital samples, after correction for any offset, shown in Figure 11, are then processed by the DSP to provide the best exponential curve by means of a least squares curve fitting technique. The exponential is first converted to a straight line by

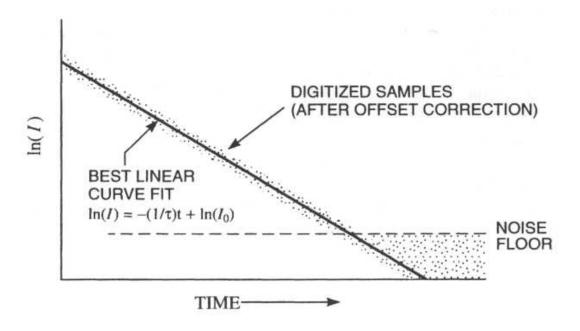


Fig. 11. Illustration of linearized digital curve fit technique

taking the natural logarithm of the digitized signal values. The slope of the best fit straight line resulting is proportional to the decay time of the luminescence.

The result of a number of curve fits are averaged further to reduce the effects of noise. The averaged decay time may then be compared with values stored in a digital look-up table to determine the temperature of the sensor.

The use of this method with a DSP has resulted in a relatively compact instrument. It was reported that the development of this technique was aimed at eliminating the dependence on many of the drift-prone components normally included in detection systems and should thereby improve system stability. However, no assessment of it in respect of such aspects was given in the literature.

In fact, an investigation into the errors caused by baseline offset and noise in the least-squares estimation of exponential lifetime has already been carried out by Dowell and Gillies[23]. It shows that the two-parameter estimation approach, which only estimates Io and  $\tau$  and assumes zero baseline (or dc) offset such as that illustrated in Figure 11, is rather sensitive to small, residual baseline offsets. The normalized lifetime-estimation error is proportional to the normalized offset. The ratio is greater than 3 when the normalized observation time,  $\beta$ >3, where  $\beta = T/\tau$ , T is the actual time during which the signal is observed and x is the lifetime. Therefore, it would appear that the performance of the digital-curve-fit technique would greatly rely on the effectiveness of the measure utilized to correct the dc offset in the observed signal. A three-parameter least-squares estimation, that is, including the estimation of the dc offset, might be a better solution. However, it requires a reiterative numerical algorithm, and this will not only increase the complexity of programming required, but also greatly prolong the signal processing time.

# 2. Oxygen sensor

Our first experiments with the time-resolve spectroscopy (decay time measurement) were provided as a design of an oxygen sensor with extended life time.

Fluorescent sensors or artificial noses with optical detection of luminescence from the materials sensitive to certain analyses have been developed by a number of manufacturers, e.g. [25-31], to mention only the most recent references. Unlike electrochemical, the fluorescent sensors consume no detected analyte and can be used both in gas and viscous liquid media. They have no membrane to change or solutions to fill, no chemically changing sensing elements and, which is the most important for application in nosing the potentially flammable or explosive environment, no electrodes or conducting wires connected to the sensor. They are immune to EMI, to interference from various other chemicals, and to changes in pH and salinity if the last parameters are not aimed to measure by the fluorescing materials sensitive to them [27].

One of widely practiced applications of the fluorescence sensing is the detection of oxygen by emission quenching of fluorophores with relatively long luminescence lifetimes. Among many other fluorescing materials, the most popular are the metal-ligand complexes dissolved in oxygen-permeable silicon polymers, which display luminescence decay times of microsecond timescale [32-39]. Commonly, the metal-ligand complexes in polymer matrices are used for measuring oxygen concentration in gaseous and liquid media at temperatures below 200 °C, although there is a report about a high-temperature (up to 650 °C) fluorescence oxygen sensor based on a metal halide compound [40]. Both the intensity and the decay time of fluorescence depend on oxygen concentration according to the Stern-Volmer equation.

Correspondingly, intensity-based and lifetime-based sensing methods are developed for measuring oxygen concentrations [41-44]. While the intensity-based technique is time-integrated, most lifetime-based methods utilize the phase-shift fluorimetry between the sinusoidal modulated excitation and the resulting fluorescence emissions [44]. This relatively elegant technology is rapidly gaining considerable industry attention.

Commercial oxygen sensors are currently available from various manufacturers. Quite popular are the FOXY sensors from Ocean Optics based on the ruthenium complex in the sol-gel matrix deposited on a tip of an optical fiber. A like sensor is also manufactured by AVANTES Inc. The sensors based on silica aerogels are also under development at Microstructured Materials Group of LBL. The lifetime-based sensors for measuring dissolved oxygen are produced by In-Situ, Inc. The SMSI<sup>TM</sup> oxygen sensor from Sensors for Medicine and Science, Inc. has been developed for many medical, biological, and food applications. A versatile OLED-based fluorosensor has been also developed at the USDOE's Ames Laboratory.

Conventionally, a means for spectral filtering is needed to separate fluorescence from excitation light reflected by the optical surfaces or scattered in the optical system both for intensity-based and lifetime-based methods such as the phase-shift fluorimetry, e.g., dispersive elements or band-pass optical filters placed in the detection channel. The relatively long lifetime of fluorescence from the metal-ligand complexes makes viable the use of time-gating technique provided that the excitation sources are capable of emitting short pulses with sufficiently sharp edges. The method of time-gating is widely used in imaging technology for tracking the fast changing objects, in particular, in the fluorescence time-resolved microscopy [45].

## 3.1 Principle of operation

Remarkably, the time-gating method opens a possibility of excluding any spectral filtering means from the detection systems related to the sensing technology. The main challenge here is that the fast decay of fluorescence after turning-off the excitation pulse limits the photon flux on a detector and requires sufficiently sensitive optical detectors for measurements. A lifetime-based system employing the single-photon detection of fluorescence in combination with the time-gating technique for measuring oxygen concentrations is outlined below.

A number of fluorescence photons  $N_f(t)$  per time unit emitted after an abrupt turning-off the excitation pulse in a fluorophore can be written as:

$$N_{\rm f}(t)=N_{\rm f}^{0}e^{-\frac{t}{\tau}},$$

where  $N_f^0 = A_f n^0$  is the number of fluorescence photons per unit time emitted by a fluorescing molecular species in the mixture at the moment of excitation turn-off,  $A_f$  is the average fluorescence transition rate,  $n^0$  is the number of excited molecules in the irradiated volume, and  $\tau$  is the corresponding decay time of fluorescence determined both by the fluorescence transition rate  $A_f$  and by fluorescence quenching. When the characteristic excitation pulse length exceeds the time of internal conversion between the excited singlet levels (saturation time), which is usually faster then the competing processes of de-excitation of the exited levels, a steady-state excitation is reached in a fluorescing sample for a given intensity of excitation, thus the number of excited molecules can be estimated as  $n^0 = k_a N_0 \tau S d$ , where  $k_a$  is the absorption coefficient,  $N_0$  is

the number of excitation photons per unit time incident onto the unit surface of a sample with a thickness  $d << 1/k_a$ , and S is the cross-section of an excitation beam. The total number of fluorescence photons emitted after the excitation turn-off ( $t_{off} << \tau$ ) is:

$$N_{\rm f} = \int_{0}^{\infty} A_f N_0 k_a \tau S d \exp(-\frac{t}{\tau}) dt = \Phi k_a \tau N_0 V,$$

where  $\Phi=A_f\tau$  is the quantum yield of fluorescence and V=Sd is the volume of the irradiated sample. The presence of quenching chemicals influences the total number of emitted photons mostly through its dependence on  $\tau$  ( $N_f \propto \tau^2$ ) provided that other essential factors such as  $A_f$  and  $k_a$  suffer relatively lesser changes. In practice, when the photon counting technique is used, a number of counts N=k  $\epsilon$   $N_f$  after the excitation pulse is additionally reduced with respect to the number of luminescent photons by the optical efficiency of a fiber-optic transmission line k and by the count-factor  $\epsilon$  of a photodetector.

The relatively low fluorescence conversion of the probes together with the overall efficiency of light capture by the optical transmission system significantly reduce

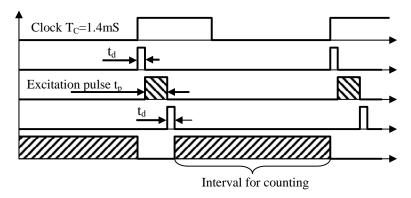


Fig. 12. Time-gating diagram of operation

the fluorescence intensity at the detector input, therefore it will be much more realistic to arrange the excitation as a continuous series of light pulses on the condition that the detector is opened for photon counting within the intervals between the pulses. Thus, the counts are integrated during a relatively large time interval as shown in Fig. 12, to ensure the sufficiently low stochastic error of photon counting proportional to  $1/N_c^{\ 1/2}$ , where  $N_c$ = N F  $T_c$ , F is the pulse frequency, and  $T_c$  is the duration of the detection cycle. Some reduction of the total number of the detected fluorescent photons after each excitation pulse due to cutting-off the tail of the decaying fluorescence can be made negligible if  $\tau$  <  $F^{1} - t_{p} - 2t_{d}$ , where  $t_{p}$  is the excitation pulse length and  $t_{d}$  is a time delay between the end of the excitation pulse and the front edge of the counting interval (detector is opened for counts) as well as between the end of counting interval and the front edge of the following excitation pulse. Advantageously, the low-power excitation sources (e.g., lowintensity LEDs) can be employed in the system to produce, nevertheless, a sufficiently large counts for each detection cycle of reasonable duration  $T_c \sim 1$  sec. To exclude any chance of detecting the excitation light, a time delay  $t_{\text{d}} <\!< \tau$  between the end of the excitation pulse and the detector opening as well as between the end of counting interval and the leading edge of the each following excitation pulse is necessary to assure the complete extinction of the reflected and scattered excitation light in the optical system: t<sub>d</sub>  $> 2CL_n$ , where C is the speed of light in vacuum and  $L_n \sim L/n$  is the optical length of the transmission line between the source and the fluorescing probe. Thus, no filtering is necessary for sensor operation and for measuring the luminescence. Other requirements are:

- It is be desirable to select the time gap between the back edge of a previous excitation pulse and the leading edge of a following pulse ( $\frac{1}{F}$   $t_p$ ) not less then the fluorescence decay time  $\tau$ , therefore the maximum frequency of excitation pulses should not exceed  $F_{max} = (\tau + t_p)^{-1}$ .
- For a chosen light source  $(N_0)$  and a fluorophore  $(\Phi, \tau)$  the minimum time interval  $T_c$  of a single cycle of measurements is determined by a tolerable stochastic error of counts  $\eta \sim 1/N_c^{-1/2}$  and the maximum pulse frequency  $F_{max}$  as:

$$T_c \geq 1/\,\eta^2 N F_{max} = (\tau + t_p)/\,(\eta^2\;k\;\epsilon\;\Phi k_a \tau N_0 V). \label{eq:tc}$$

## 3.2 System overview

The experimental oxygen sensor is a single channel module consisting of a LED, a bifurcated fiber-optic cable, a commercial oxygen

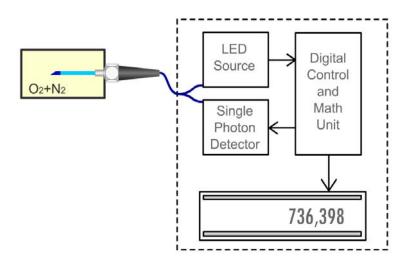


Fig. 13. General layout of the sensor

sensor FOXY-OR125

from Ocean Optics. Inc., a single-photon photodetector Hamamatsu 7400 (180 – 800 nm) in an assembly with a fast counter (< 2 ns time resolution, our design) based on broadband electronics, and a control unit for the necessary time-gating of the LED and the photodetector (Fig. 13).

The pulsed radiation of a blue LED (LEDtronics, Inc. #BP280CWPB3K-3.6Vf-050T) with the maximum spectral power (470 nm) near the maximum of excitation spectrum of the probe ( $\lambda = 450$  nm) is inserted in one of prongs of a bifurcated fiber cable. The probe is connected to the stem of the bifurcated cable and inserted into a vessel containing a gas mixture. It is excited by a pulsed LED that emits a series of short ( $t_p = 1$  to 10 microseconds, controllable) pulses with sharp (100 nanoseconds) edges. The photodetector is turned on at  $t_d = 500$  ns after the end of each excitation light pulse and turned off 500 ns before the leading edge of each following pulse. Overall length of the fiber-optic cable is about 1 m, so that the extinction time of excitation light in the system

is ~120 ns. Total integration time  $T_c$  of a single measurement cycle can vary from 0.1 s to virtually infinity. The following settings have been chosen for the test:  $t_p = 1,5$  µs,  $T_c = 1$  s, F = 1 kHz. The 'dark' count of the counter (approximately 400 counts per second) was measured before each series of measurements. The reading range of the counter measured to be linear up to  $10^6$  counts per second.

#### 3.3 Tests and results

The sensor was tested at a testing stand that contained a chamber equipped with a vacuum pump, the valves to controllably deliver air, nitrogen or oxygen, and a manometer. The measurements were performed with the system configured as shown in Fig. 13. Excitation was effected with a blue LED emitting a series of light pulses as described in the previous section with the average power of  $10 \,\mu W$ . Before each test the chamber was pumped out to the residual pressure of  $10^{-3}$  Torr and kept at this low pressure for 10 minutes to provide degassing of the sol-gel matrix and an overcoat on the probe's tip. The fluorescence output  $N_{c0}$  in vacuum or in nitrogen atmosphere served as the reference point for each following series of measurements. The response time of the sensor was approximately 1 min due to slow diffusion of gas through the overcoat above the sol-gel film therefore each series of readings was taken 2-3 min after the pressure or mixture change in the chamber.

An example of sensor response dependence on air pressure obtained for atmospheric air under constant room temperature is shown in Fig. 14. Each experimental point is an average of series of 20 consecutive readings. As expected, the standard deviation of readings for each particular pressure follows the  $N_c^{-1/2}$  law (Fig. 15). The graph in Fig. 14 displays some deviation from the Stern-Volmer law at higher pressure

close to normal atmospheric pressure, which is, apparently, the characteristic of the FOXY probe. The best fit corresponds to the dependence of the  $(N_0/N-1)$  parameter on partial pressure of oxygen  $P_p$  expressed by the formula  $(N_0/N-1)=aP_p-bP_p^2$  with  $a=0.0057~Torr^{-1}$  and  $b=5.2\times10^{-6}~Torr^{-2}$  or by the commonly used expression  $(N_0/N-1)=AP_p+BP_p/(1+bP_p)$  [14] with  $A=7.5\times10^{-3}~Torr^{-1}$ ,  $B=-1.75\times10^{-3}~Torr^{-1}$ , and  $b=-2.3\times10^{-3}~Torr^{-1}$ .

A series of tests were also performed with different mixtures of pure nitrogen and oxygen for various initial pressures of nitrogen in the chamber. The data for the sensor's response recalculated for the partial oxygen pressure were analogous to those obtained for atmospheric air however stronger deviation from the Stern-Volmer law was observed. Further detailed study is planned to find a possible cause of such stronger deviation; among others, the influence of the overcoat on the probe's tip and the erroneous readings of a digital manometer calibrated for atmospheric air will be examined.

The sensor based on single-photon counting and appropriate time-gating of

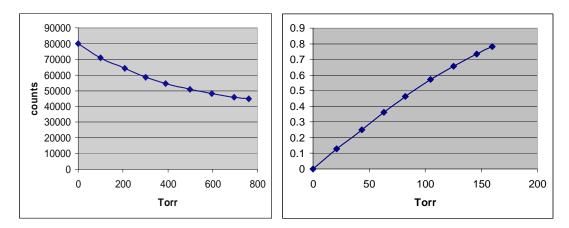


Fig. 14. Raw fluorescence output vs air atmospheric pressure (left) and its Stern-Volmer linearization (right) with the values of  $\frac{N_0}{N}-1$  scaled vs  $O_2$  partial pressure.

excitation and detector operation has demonstrated good performance in application for measuring oxygen concentrations. It is potential to operate with a wide variety of fluorescence probes available on the market and with other sensors currently under development. The reproducibility of the system was evaluated during the test in air and no deviation was found from the standard parameters of the sensing probe in a series of test runs.

As a matter of fact, the commercial LEDs utilized as the excitation sources for fluorescent measurements emit a peak radiation between 400 and 500 nm and, as we observed with a spectral analyzer, there is a longwave tail up to 700 nm of much lower relative intensity  $\sim 10^{-3}~I_m$ , where  $I_m$  is the spectral intensity at the wavelength of maximum brightness of LED. This weak at first sight radiation overlapping the spectral band of fluorescence (600 – 700 nm) can, nevertheless, interfere with the fluorescence because the efficiencies of most fluorescence sensing devices (conversion coefficients for

the fluorescing materials combined with the optical efficiency of light capture and losses in the transmission lines) are also of the order of  $10^{-3}$  –  $10^{-5}$  and no filter or dispersing element is able to completely suppress this interference. The conventional oxygen detection systems would firstly require

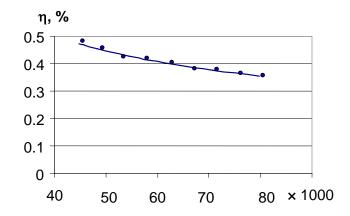


Fig. 15. Mean-root-square error  $\eta$  of the measurements vs number of counts  $N_c$  (dots) and the function  $1/N_c^{1/2}$  (line).

strong filtering of excitation light beyond the spectral band of fluorescence and secondly the sufficiently efficient fluorescence dyes should be incorporated to overcome the residual LED light from the longwave tail within the transparency band of the filter. Apparently, the only radical solution of this problem is the use of the system with timegating as it is outlined above because no radiation from a LED can reach a detector *in principle*.

It should be also emphasized, that the response of the system is digital by the very nature of single-photon counting, therefore it contains no analog amplifiers, transducers, or other noisy electronics. The error of measurements is determined exclusively by the stochastic nature of photon counting and corresponds to the standard deviation for a stochastic process. A relative error due to dark count did not exceed 1% and could be actually reduced by the factor of 10 by the corresponding increase of either pulse repetition rate, or LED brightness, or both. Besides, the stability of the dark count ~ 400 counts for each series of measurements within the stochastic deviation of ~ 20 counts allows subtracting it from the raw counts as a systematic error thus further reducing its impact on the accuracy of measurements. Another essential advantage of the system with the digital response is that, when the range of counts is shifted to the higher portion of linear range of the counter, the relative error of measurements can be reduced below 0.1 %. The sensor is very simple and robust. It contains no optical elements but a fiber-optic cable. No optical adjustment is required. No optical element can be detuned or degrade during operation therefore the system can reliably operate in harsh environment with severe mechanical vibrations. It consumes very low electric power for feeding a relatively low-intensity LED, a digital counter, and a display unit, all based on the

microelectronic technique. The control module can be fed by a conventional battery being minituarized to the size of a pack of cigarettes which facilitates its use in field conditions.

Potential applications of the time-gated sensor with single-photon counting for measuring the fluorescence response of dyes include: oxygen sensing in avionics, environmental monitoring, medicine and biology, sensing of other analytes in industry, chemistry, and medicine, as well as detection of toxic agents.

# 4. Design and implementation of the temperature sensor

In the upper part we show the possibility of implementation Single Photon Detection technique in fluorescent decay measurement (oxygen sensor). To extend a measure technique we'll implement the same technology for another measurement – measure a temperature value. The temperature measurement is mach more broad and popular compare with oxygen. Addition reason for us is simplicity of prototyping. In the frame of such sub-project we are planning to make an additional research in the area of fast and flexible opto-electronic prototyping.

#### 4.1 Selection of the sensitive substance.

Based on literature source [60-99] we selected the primary temperature sensitive substance –  $Cr^{3+}$  doped insulating crystals. Such crystals are popular objects for solid state lasers and their fluorescence nature is well researched. Presentation of general areas of the temperature's sensitivity for different  $Cr^{3+}$  doped crystals is presented on Fig.16.

Easiest modeling object is a ruby crystal – it's demonstrate very good sensitivity for both environmental (zone B) and high (zone C) temperature region. Such temperatures can be provided by digitally controlled laboratory heater. The highest decay time and a lower cost in comparison to other crystals is an additional reason for the selection of a ruby crystal as a primary sensitive object. We design a laboratory setup for

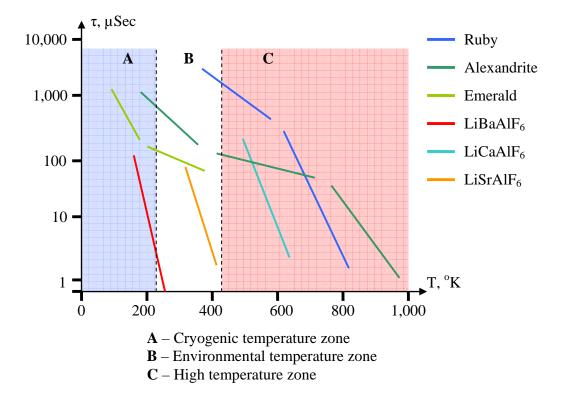


Fig.16. Temperature – decay time characterization of popular  $Cr^{3+}$  doped crystals.

temperature control of the ruby ball. Such setup presented on Fig. 4.1.2. It is consists from two parts: heater and temperature controller.

We designed a special holder for ruby ball. Such holder combines in its construction a good thermal conductivity with the heater and also a good optical access to the ball from an outside optical system. Almost 90° angle of fluorescent light collection is available in our design.

Heating part of the setup connected with low thermal resistance to the ruby holder and has a heating element and a thermistor. Using standard 5-pin connector and an extension cable our heater connected to the thermo controller.

As a thermo controller we are using an adapted soldering station ZD-929C. We increased resolution in the temperature measurement up to  $1^{\circ}$ C and make more precise temperature setting. As a result of our modification we can set any temperature of ruby ball from the room level (300  $^{\circ}$ K) up to 700  $^{\circ}$ K.

# 4.3. New components for rapid prototyping. Novel prototyping technique.

The experimental part of almost all research projects requires a creation of certain experimental setups and building of models and prototypes. Sometimes it takes a considerable amount of time to build a good setup or prototype, and this time could be way longer then the experiment itself.

The things come even worse when you need to perform several experiments or a series of experiments to, say, compare or choose the optimal parameters of the system.

In this case a certain universal solution is necessary. And such things are commonly used in optics for instance. There are no optical labs not equipped with special

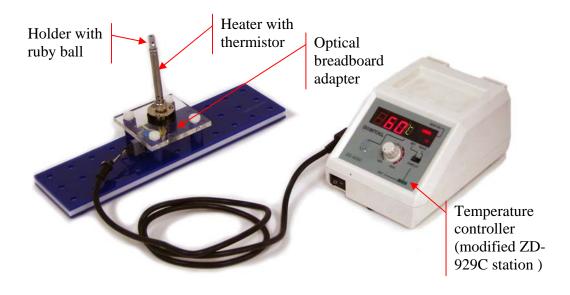


Fig.17. Laboratory setup for temperature control of crystals with optical access to the testing ruby ball.

reusable optical components, mounts, holders, sliders, rotary fixtures, optical benches etc, which being installed on special optical tables help a researcher to handle almost any task and build almost any experimental setup. All the components are standardized and compatible with each other, except some really exotic parts with a very limited use for some special purposes. The standard used in the US is so called 1 inch standard. In some European companies metric standards for optical components are used, but those are not that widely spread.

As I noted, the sets for optical prototyping include standardized posts, mounts, holders, fixtures mounted on optical tables or breadboards helping to make certain constructions for optical prototyping purposes. Those components are made of high quality metal, those are very precise, heavy and very costly. One component could cost several hundred or even thousand dollars.

In my research I build lots of different electro-optical systems with number of components, with a big deal of very unique ones. In our first experiments we used to use shop bought optical components with some "home made" ones, but that definitely wasn't convenient mostly because any even simple experiment used to require an optical table and a component set of several thousand dollars value.

So one day we came to an idea that it would be nice to have something standardized, universal, light and cheap. And one more quality we needed – the setups should be portable for tests and demonstrations. Honestly, I put considerable efforts to find something like that which would already offered on the market, but what I found were the products similar to the main brands like Thorlabs or Newport, metal made, heavy and expensive.

Having spent time trying to find without any good results some inexpensive and light substitute to optical breadboards and tables offered on the market, I finally came out with an idea of designing our own acrylic breadboard for numbers of experiments we were performing at that time.

In a possession of our lab there is an amazing laser robotic system used for cutting and engraving extruded and custom acrylic sheets. So my first idea was just to cut a 6mm thick board with holes in positions corresponding to ones on a standard optical breadboard (1 inch standard). And having made standard threads in the holes on the board we would get a cheap and light substitute for heavy and expensive optical breadboard. That definitely would work but not without some negative side effects. The first drawback was the fact that the threads made in acrylic sheet wouldn't be durable. And the second one was the necessity of making 121 threads in 11''x 11'' breadboard, which would be very time consuming if make those manually. 11''x 11'' size just came from a standard size of acrylic sheets offered on the market.

Actually there was a possibility of making a breadboard just without threads in the holes but it would not be a 100% substitute for the standard breadboard and of coarse wouldn't be as convenient and elegant as we expected. The threads in the holes are a must because optical posts and mounts require them. It is just impossible to fix the most of components with a nut from the other side of the breadboard.

Having faced the above drawbacks of a simple acrylic breadboard, we agreed to use the idea of acrylic 6mm breadboard without threads in the holes just as a construction part for some special design cases. But to solve our main problem, we were supposed to

invent something else, and that was how the idea of a layered breadboard structure appeared.

We invented and submit a patent application for the idea of using a layered structure in the manufacturing of light and cheap 1" standard compatible electro-optomechanical breadboards. The outstanding unique qualities of our breadboards are that those are appeared to be dielectric, diamagnetic and totally transparent for  $\gamma$ -rays. Those

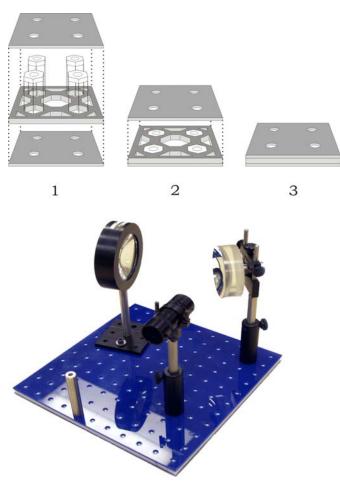


Fig. 18. Dielectric breadboard.

qualities are very useful in the research setups and prototypes built in our laboratory.

Our breadboard design is shown on the Fig.18. You can see the elements and the way the board is assembled.

The main idea of our breadboard construction is a special layered structure. The core 6mm layer contains special sockets designed that so to

accommodate standard plastic nuts which solve the problem with making threads; top and bottom 3mm layers of the structure keep the nuts in the sockets preventing them from

falling out and make the surfaces of the breadboard precise. All the construction is glued by acrylic glue which makes the construction uniform, stable and concrete. To assemble the breadboard I designed a special tool which makes the positioning of the layers easy and precise. The breadboards we produce are of different sizes and shapes. The patent on this idea and technology is filed.

## 4.3.1. Fast modulated LED light source

As it was presented into the upper part of the given research we need a modulated light source for all types of decay measurements. To avoid a limitation with selection of an optical wavelength we selected a LED based modulated light source as a basic element of the system.

In frame of our research we have designed a family of LED based light sources. The main reason for such a design was an absence of the fast modulated light sources for different wavelengths compatible to standard holders. Because of the fact that in our experiments with fluorescence of different  $Cr^{3+}$  doped insulating crystals we are inspecting the measurements of the wide range of a decay time – the quality of the light source becomes a very important parameter.

The shortest decay time for expected material (ruby) and temperature (700  $^{\circ}$ K) is 500  $\mu$ Sec. It means that we have to use neither a light source with turnoff time nor more then 1  $\mu$ Sec. So a special circuit was designed to match these constraints. Its electrical schema and PCB implementation is presented on Fig.19.

The device is powered by 9 volt (battery or AC/DC plug-in power supply). Voltage regulator IC1 supports digital circuits by 3.3 Volts. Schmidt-trigger IC2 helps to

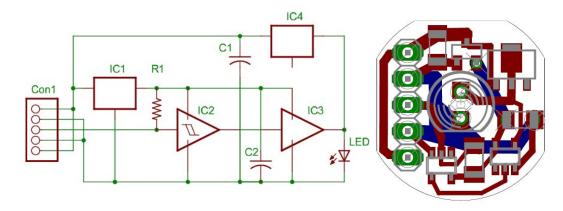


Fig.19. Fast modulated LED light source and its PCB implementation.



Fig. 20. Fast modulated LED light source and its pin description.



Fig. 21. Small power supply for modulated LED light source.

provide a sharp rise and fall edges of the optical pulse. Input terminal of IC2 forced to +3.3Volts. It's protects the circuit from "floating" input. IC3 is an open drain gate. It provides bypass for the current direct to the ground. As a source of the regulated current we are using 20 mA current source IC4. Total current consuming is smaller then 21 mA.

Physical implementation of the modulated LED source was done in the frame of

C-mount optical standard. We designed a special holder which allowed us to mount the modulated LED source into any optical C-mount standard device. All components were designed and manufactured out of colored acrylic utilizing the laboratory's facilities. Fig.20 demonstrates the final version of the modulated LED source. We designed and implemented 15 of such devices, covering wavelengths from 385 up to 980 nM. As a part of a universal C-mountable modulated light source we designed a small power supply. It can be

installed on any standard breadboard or optical table.

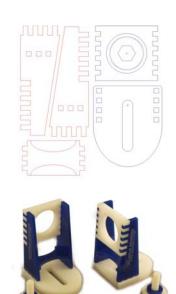


Fig. 22. C-mount 2" holder.

Power supply is presented on Fig.21 It works with a standard 9 Volt battery or with external power source and also converts signals of standard TTL level (coming through BNC) to LVTTL level of our modulated LED light source.

#### 4.3.2 Adjustable holders and optical components.

As I noticed before, building an experimental setup could be a challenge. Having VersaLaser<sup>TM</sup> robotic system I designed a set of posts, fixtures, mounts and holders, and I'm still working on the improvement of their construction and on inventing new products and solutions. Here I would like to present some of the most commonly used designs.

*C-mount 2'' high holder*. This totally dielectric mount is very easy to use in all sorts of applications requiring the use of C-mount standard parts, such as cameras, light sources,

lenses etc. This mount has the C-mount thread, the optical axis of the mounted device is 2" from the surface of the breadboard. The mount is adjustable so that to find the right focal distance, for instance, when working with optical setups. This was the first device we started from. So you could see that some of the construction ideas of this holder are inherited in our other designs.

We had used this holder in a number of experiments and it inspired us to a creation of another device which would provide more flexibility to the setups giving the possibility of 2-d adjustments. This is a 2-coordinate slider with an interchangeable central plug, with allows us to use this device for a variety of different setups. For example we use this slider to hold and adjust Fresnel lenses, SMA connectors for fiber lines (for coupling for instance), C-mount light sources. There is lots of use of this particular mount in the projects our laboratory participates in. For instance, this mount

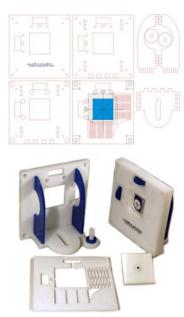


Fig. 23. Two coordinate adjustable holder.

was used to hold, adjust and focus a  $\gamma$ -ray source in one of our projects not directly connected with this research. Fig.24 presents the main part of the universal holder – Integrated 2-D positioning unit. The unique design of the device allowed us to hold 1.5"×1.5" plate with any device installed on it and provide with  $\pm$  0.1" adjustment displacement in XY coordinates. On Fig. 43A four typical central components are presented:

• SMA fiber tip mount;

- 25 mm lens holder;
- Frensel lens holder;
- C-Mount holder (with installed LED driver).

The most interesting solution in this device is the use of so called flat springs in the core part. The whole core layer with springs is made as a single part in one mechanical operation, even the springs are made of acrylic utilizing its elastic properties. The step, period and the walk of the springs are calculated to meet the requirements of the stress and torsion so that to make the device durable and reliable.

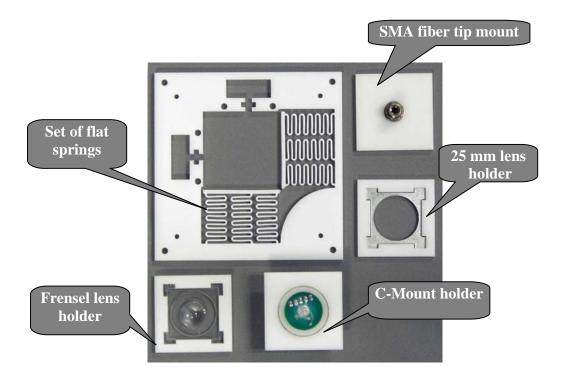


Fig. 24. Integrated 2-D position unit.

Here I would like to say a couple of words about the use of Fresnel lenses in our novel prototyping technique. Have you ever seen an optical setup without lenses? I did, but it is more an exception then a rule. Lenses are used to focus, collimate light, build

images and so on. In the scope of our research, we created a lot of different optical setups.

It was recognized that the contour of the refracting surface of a conventional lens defines its focusing properties. The bulk of material between the refracting surfaces has no effect.

In a Fresnel lens the bulk of material has been reduced by the extraction of a set of coaxial annular cylinders of material, as shown in Figure 25.

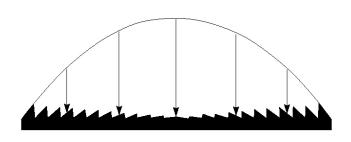


Fig. 25. Construction of a Fresnel lens from its corresponding asphere.

Each groove of the Fresnel lens is a small piece of the aspheric surface, translated toward the plano side of the lens. The tilt of each surface must be modified slightly from that of the original portion of

aspheric surface, in order to compensate for the translation.

The contour of the curved surface is thus approximated by right circular cylindrical portions, which do not contribute to the lens' optical properties, intersected by conical portions called "grooves." Near the center of the lens, these inclined surfaces or "grooves" are nearly parallel to the plane face; toward the outer edge, the inclined surfaces become extremely steep, especially for lenses of low f—number. The inclined surface of each groove is the corresponding portion of the original aspheric surface, translated toward the plano surface of the lens; the angle of each groove is modified slightly from that of the original aspheric profile to compensate for this translation.

The main innovation embodied in Fresnel's design was that the center of curvature of each ring receded along the axis according to its distance from the center, so as practically to eliminate spherical aberration.

Modern computer-controlled machining methods can be used to cut the surface of each cone precisely so as to bring all paraxial rays into focus at exactly the same point, avoiding spherical aberration. Better still, newer methods can be used to cut each refracting surface in the correct aspheric contour (rather than as a conical approximation to this contour), thus avoiding even the width of the groove (typically 0.1 to 1 mm) as a limit to the sharpness of the focus. Even though each groove or facet brings light precisely to a focus, the breaking up of the wavefront by the discontinuous surface of a Fresnel lens degrades the visible image quality. Except in certain situations discussed later, Fresnel lenses are usually not recommended for imaging applications in the visible light region of the spectrum.

The grinding and polishing techniques used in the manufacture of conventional optics lead to spherical surfaces. Spherical surfaces produce optics with longitudinal spherical aberration, which occurs when different annular sections of the optic bring light

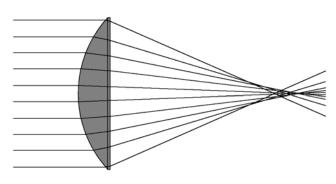


Fig. 26. Illustration of longitudinal spherical aberration.

rays to a focus at different points along the optical axis. This phenomenon is illustrated for a positive focal length, plano-convex conventional lens in Figure 26.

The rays shown were traced through an f/1.3 spherical-surface

lens; the focus is evidently spread out over a considerable distance along the optical axis.

As is evident from the figure, the longitudinal spherical aberration is very strong.

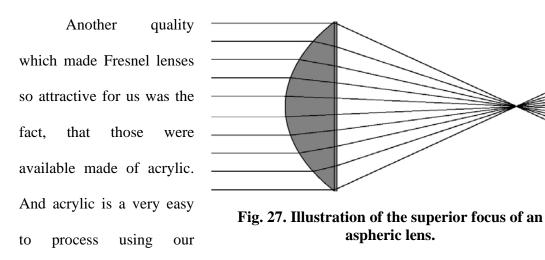
Figure 27 shows an aspheric lens of the same focal length and f–number; note that the surface contour is modified from the spherical profile in such a way as to bring rays passing through all points on the lens to a focus at the same position on the optical axis.

The combination of the aspheric surface (which eliminates longitudinal spherical aberration) and the thinness of the lens (which substantially reduces both absorption losses in the material and the change of those losses across the lens profile) allows Fresnel lenses with acceptable performance to be made with very large apertures.

The lens shown has the same focal length and aperture as the spherical lens shown in Figure 4, yet longitudinal spherical aberration is entirely absent. It is possible even to make aspheric lenses with f-numbers smaller than one.

Many of the spherical-surface lenses in common use are biconvex, often with the same curvature on the two sides. Such lenses exhibit substantial spherical aberration and other aberrations as well, but are symmetric in their properties. In almost all instances, Fresnel lenses are plano-convex. This fact, along with their aspheric profiles and their low f – numbers, leads to strongly asymmetric behavior.

The Fresnel lenses are very easy to operate with because they are flat. It is way easier to build a holder for a flat part. The Fresnel lenses are not used in imaging systems because of the boundaries of the grooves, but in the case of parametric optical setups those are very common.



VersaLaser robotic system. So having designed some positioning tooling we were able to cut lenses of a necessary sizes and shapes.

What's about optical qualities of acrylic – those are totally acceptable. Optical quality acrylic is the most widely applicable material, and is a good general-purpose material in the visible. Its transmittance is nearly flat and almost 92% from the ultraviolet to the near infrared; acrylic may additionally be specified to be UV transmitting (UVT acrylic) or UV filtering (UVF acrylic). The transmittance of our standard acrylic materials between 0.2  $\mu$ m and 2.2  $\mu$ m is shown in Figure 10 for a thickness of 1/8" (3.2 mm). Standard acrylic thicknesses are 0.060" (1.5 mm), 0.090" (2.3 mm), and 0.125" (3.2 mm).

Angle mounts. Sometimes it is very important to build special 3-D constructions to save the space and make the setups more effective and easy to work with. So I designed two types of 90-45-90 and 90-90 degree angle mounts. These devices are ones of the most complicated in manufacturing. They have a lot of parts and are very critical to the quality and accuracy of the assembling. To make sure that the angles are precise, we designed some special instruments to fixate the parts while gluing them together. The



Fig. 28. Angle mounts.

angle mounts are very easy to use. They bring a lot of flexibility to the prototype design.

Electrical breadboard holder. We can't even imagine now how it would be possible to make electro-optical designs without using of this holder. The use of this simple part gives the great flexibility and a true mobility to the prototype. It is always good to have everything installed in one place with minimum loose connections. So once the electronics is installed on the prototype

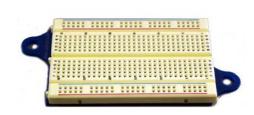


Fig. 29. Electrical breadboard holder.

base, it comes very easy and convenient to transport and store it. Having designed this

holder we started positioning all the set of the discussed devices as opto-electronic but not just optical.

To recapitulate this part, I would like to mention that the real beauty of our technology is in the ability to manufacture unique customized prototypes really fast. Using our methods and technology we can manufacture breadboards of any shape. In my experiments with a ruby crystal as a sensing element, I needed a breadboard with a 2 inch diameter hole in the middle to install the regulated heater. Had I tried to find something like that on the optics market I wouldn't have found anything similar for sure, because simply speaking such things just didn't exist. Or as another example, one of our potential partners does small precise mechanical and hydraulic robotic systems, and sometimes they need to set a mounting breadboard into a mechanism itself. It means that sometimes they need a breadboard of a particular size and shape. So we designed for them some special customized breadboards with smaller threads and shorter step. So that came a breadboard for small setups with 2.56 screw size and 0.25" step. The construction allows us to install this board on 1 inch standard breadboard or an optical table using standard screws.

The top view of the setup is presented on Fig. 30. The general design is based on "classic" 90° angle configuration. The lightweight modular prototype design helps us to modify the setup for different types of measurements. We are currently using the setup as a test bench for all new and known techniques of decay time measurement. The setup is mounted on the top of modular dielectric breadboard (we used 4 such breadboards for our setup). I have designed a special Time Interval Generator, which controls the time

sequence of the experiment. Time Interval Generator runs an LED driver and Counting / Processing Unit.

Optical system consists of 4 Frensel lenses (2 collimation and 2 focusing ones). Each lens is installed into a 2-D position unit for the adjustments.

As a main photosensitive unit I used Single Photon Sensitive Module (Hamamatsu).

In the optical center of my installation I have installed a ruby ball on the top of the heating element. The thermo controller is installed near the Counting / Processing Unit and is used to control the actual temperature of the ruby ball.

The setup on Fig. 30 is presented without of light protective cover which is used to prevent external photons to affect the measurements of fluorescence.

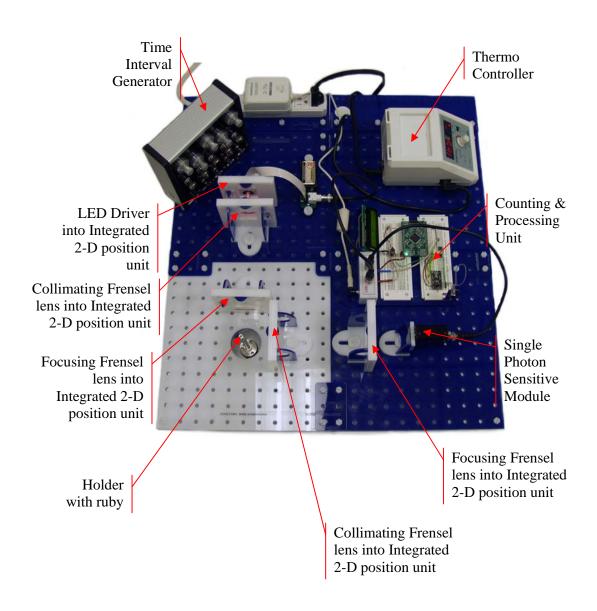


Fig. 30. Temperature sensor open space setup.

# 5. Design, implementation, and characterization of the photon-tight differential fiber optical splitter/combiner

Working with single photon counting one should always take into consideration the fact that optical losses that could not be counted in other situations, in this case can ruin all the experiment, just because some part of the photons would not be registered.

We started our experiments with industrial 50:50 fiber optic splitters offered on the market. The problem we faced was 50% loss of fluorescent photons on the transmission line towards SPD (single photon detector). Actually, in Y-shape optical splitter, based on a fiber bundle, one half of fibers go right (in our case towards the light source) and the others deliver the rest of fluorescent photon flux towards the single photon detector. All the photons delivered to the light source branch are lost.

In our first experiments we demonstrated that our system worked even in 50:50 splitter configuration.

While losses in 50:50 Y-shape don't stop the system, those can severely decrease the sensitivity of the devise though. This is why we decided to use so called differential splitters. The key behind this is very natural. The less fibers of the bundle we use to deliver the light to the sensor the lower losses of fluorescent photons we would get on the way back. Naturally, the intensity of the light coming from the light source is far higher then the intensity of the fluorescence. This is why it is logical to dedicate smaller amount of the fibers in the bundle to the light source and larger amount of fibers to the photon detector.



Fig. 31. Plastic fiber 1:15 AMP/AMP differential splitter.

We invested certain amount of efforts to find fiber differential splitters on the marked, but surprisingly we didn't find any offers on devices of that sort. Some companies proposed us customized solutions. But those were not we were looking for. So as a result of this search it was decided to try design and manufacturing of differential splitters using our own facilities.

I designed a few generations of the splitters starting from plastic 1:15 AMP splitter ending with glass 1:48 SMA-AMP compact photon tight splitter. We decided to stop using classic Y-shapes and switch to special designed boxed units which are definitely less sensitive to vibration and other mechanical distur-bances, considerably smaller in size, photon tight, and faster to implement and copy.

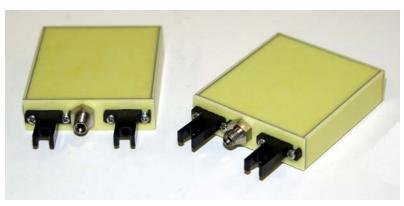


Fig. 32. Photon tight 1:15 FFF AMP/SMA combined compact differential splitter based on plastic fibers

The first designs of photon-tight optical splitters used so called "PSB case technology" which is probably very familiar to any electrical engineer.

One of my first compact designs is presented on Fig. 31 and 32.

It's easy to recognize several zones of soldering on Fig 33. Such photon-tight

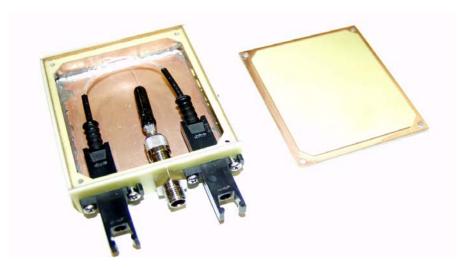


Fig. 33. Construction of photon tight 1:15 FFF AMP/SMA combined compact differential splitter

technology provides a guarantee of the quality of splitters.

To avoid a problem with the photon leakages of AMP plastic connectors we implemented an advanced SMA configuration of the splitter. Such splitter is presented on Fig. 34. In the same splitter we tested a glass based fiber bundle. Configuration of such device is presented on Fig. 35.



Fig. 34. Glass 1:48 SMA-SMA compact photontight differential splitter.

Fiber splitters are manufactured commercially mainly for the telecommunication industry.

A typical commercial device has a **Y** shape such that the stem of the fiber represent the flux to be split, and the two prongs of the **Y** 

represent the two fluxes after the splitting. In commercial fiber-optics splitters, the stem of the **Y** consists of one large-diameter fiber. Several small-diameter fibers are butted to its cross section and their number is split in two to form two prongs of the **Y**. This commercial design is optimized for splitting the total flux into two equal fluxes. However, it would incur a considerable power loss if the flux is unequally split or if the device is used as a combiner.

The developed by our lab Fiber Bundle Splitter/Combiner that can function as both a splitter and a combiner and allows splitting and combining the fluxes in any desired proportion, with a value of cross talk less then 1%. This design is ideally suited

for the delivery of the excitation to a fluorescing object and fluorescence to a detector. The stem of the Y consists, instead of one thick fiber, of a bundle of fibers of smaller diameter (say, ten 250 µm fibers) that can be divided in any proportion into two prongs. Most of them go to the detector, and some (one) goes to the excitation source. In this way, there is no loss at the joint of the Y stem and the prongs. We intend similar technology implementation the integrated FSC/Pulsing Light Source.

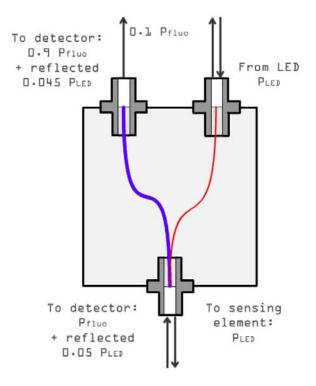


Fig. 35. Design and implementation of the Ultimate Sensor's Fiber Bundle Splitter/Combiner.

In our first single photon sensitive integrated fluorescence detection unit we

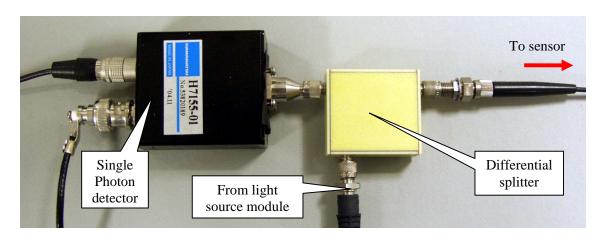


Fig. 36. Actual detection system I.

utilized a fiber optic based photon tight SMA splitter design. On Fig. 36 such detection unit is presented in a demo configuration.

To recapitulate this part, these devices are used in the optical scheme for splitting or merging two unequal light fluxes: the UV excitation from the LED to be delivered to the sensing probe and the fluorescence from the probe to be delivered to the Fluorescence Detection System. We integrate the FBSC with the Pulsing Light Source, to miniaturize the device and reduce the length of the fibers.

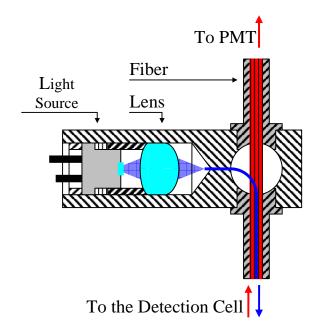


Fig. 37. The schematics of FSG integrated with the light source.

arrives to the detector and - in another scheme - it would have drowned the fluorescent signal. Fortunately, in the post-excitation scheme the fluorescence is delivered to the detector in its entirety (90%) after the illumination source is turned off. During the excitation interval the detector inactive. The is implementation of the integrated FSG is presented on Fig. 38. The presented Fig. 38. Implementation of FSG integrated device was our first try in the use of acrylic components as elements of optical

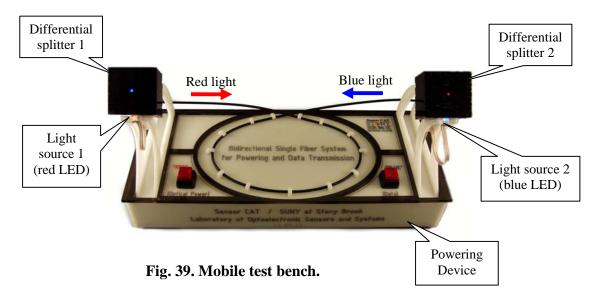
The integrated FBSC/Pulsing Light Source is schematically shown in Fig. 37. Owing to the ingenious design of the FBSC, the excitation light is delivered in its entirety to the sensing element. Some of the excitation light is reflected from interfaces glass-air reflection) and a fraction (0.1) of reflected excitation (0.04%)



with the light source.

designs. The design shows the possibility of fast and repeatable prototyping and manufacturing. First combined differential fiber optic splitter was tested in Sensor CAT laboratory of optoelectronic sensors and demonstrated very good and repeatable results (90% of excitation light delivered to the optical target and 87% of emission light delivered from the fluorescent target to the output optical terminal).

For the testing of new designed optoelectronic elements and also for a



demonstration purposes I designed a mobile portable test bench. Such bench is presented on Fig. 39. Test bench was mostly designed to compare different internal structures of the optical fiber bundles. The combined optical splitters/light sources are connected through a single optical fiber. It demonstrates very clearly the amazing capabilities of differential splitter/combiner used for the bidirectional data/power transmission through a single fiber. The demonstration device shows that the light from both lightsources is delivered to the targets without any distorsion or mixture through a single optical fiber at the same time. The different colors of the lightsources is choosen so that to make the demonstration more clear.

Based on the successful design of the combined differential fiber optic splitter with LED based light source we extended our initial design the integrated single photon sensitive fluorescence detection unit. Such unit is presented on Fig. 40. It has only electrical connectors and **SMA** fiber optic connector. Practically it is first free of adjustment integrated fluorescent detector.

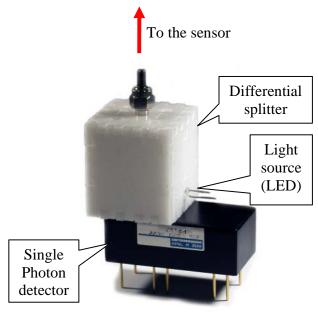


Fig.40. Actual detection system II.

#### 6. Design, implementation and testing of the decay time measuring system

Everybody knows, that the electrical designs, especially advanced ones, are very time consuming. To make a working device prototype a designer is supposed to take series of steps. The truth is that to create a schematic of the devise is not quite enough unfortunately – the challenge just begins after that. Firstly, the device should be tested on the real components. In my practice I saw lots of cases when real devices worked very differently in real components then being modeled in some software simulator. Now there exist a large variety of really sophisticated software electronic simulators which take into consideration lots of parameters of real devices so that to make the simulations closer to the real life, but nothing is better then to test a real prototype.

Sometimes it is possible to perform a rapid prototyping of the device, using breador wire- boards for instance. In many cases it could be a perfect solution. If your circuit is
not large and is not sensitive to long connections, if the circuit doesn't work on high
frequencies, if it doesn't use surface mount devices (SMD) – that would work. Otherwise
a rather long manufacturing process should be run, and usually it takes several weeks and
even months if you don't have your own manufacturing facility. So, once the schematic is
finished the designer needs to design printboard layouts for the device. Making
printboards is more art, then science. The most of electronic CAD programs offer
autoroute functions to trace the board automatically. Actually, the key behind the process
of tracing is the following. The schematic is being represented as a graph with nodes
corresponding to the pins of the components and edges representing the connections
between them. Now, the components should be placed so that to allow the maximum

number of the connections (traces) to be routed without intersections. It is not a simple task. A lot of mathematics and algorithms stand behind this. Sometimes it is possible, sometimes not. If the schematic graph is not planar, the breadboard should have several layers or use jumpers. In my designs I used two layered printboards and I really proud that I managed to design a circuit with 100 pin microchip with the supporting electronics only in two layers.

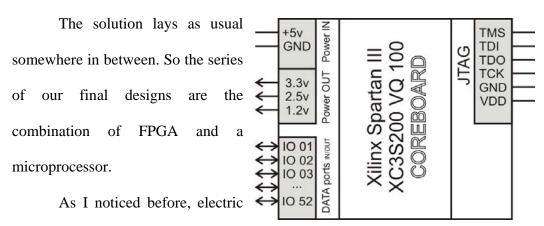
So, the previous discussion was to acknowledge that the design of real electronic devices is a considerably time consuming process, which involves a few stages and if a mistake is made somewhere, the only way to fix it is to start from the beginning.

When you design a particular device, you know at least what you expect it to be. What is about the research projects – this is only an experiment who would give the answer which parameters, timing etc are right. And this is what happened after I designed my first analog control sequence generator. It played its role, helped us to prove that the concept worked and that was it - no further use of it. To change the parameters of the device there was the only way – to design a new one. We understood that this would happen over and over again. The setups will change, the sensing materials will differ, the conditions of experiments will vary. The redesigning the electronics every time some of the above change could make the research go forever. So the only solution for this problem was to build a programmable device with the same hardware for all the cases.

There exist two main types of programmable devices – microprocessor systems, and PLDs (programmable logic devices). From the point of view of programming, the microprocessor systems are definitely more desirable, because of the flexibility of programming languages and function libraries, advanced mathematical functions. The

drawbacks of the microprocessors are that they are driven by so called "stops" or "interrupts", they don't have asynchronous inputs and they are rather slow. The first two make the microprocessor systems not really real-time. The last one doesn't allow use them as the systems with a fast response.

On the other hand, PLDs are nothing else but advanced digital circuits, whose schematics are described by special descriptive programming languages, such as Verilog, VHDL, aHDL. No doubt, that these reprogrammable devices are faster and more useful in the real-time systems because of their ability to handle asynchronous inputs and outputs. But in comparison with the microprocessor systems, PLDs' languages are low level and as a result, those are harder to program and configure. For example, FPGAs, which we took a decision to use, have only basic math functions – which is logical. But that fact brings us a lot of problems, because we register a photon flux which is stochastic, but to work with statistics, we need to calculate many different parameters using rather sophisticated formulas.



designs are very time consuming and costly. So I came with the idea

Fig. 41. Schematic element for FPGA coreboard.

to make the FPGA coreboard true universal. As all advanced systems, FPGAs require

supporting circuitry. First, it uses several Vcc voltages, in my case 3.3v, 2.5v and 1.2v. Second, a flash memory chip is necessary to store the configuration. And last, there are lots of components like capacitors and resistors are necessary for the first two. So my idea was to put all those on the same coreboard. I mean everything used for FPGA support – memory, voltage regulators, oscillators, reset circuit, JTAG programming port. So the coreboard should have +5v Vcc pin, Ground pin and a set of FPGA inputs/outputs. Simply speaking, my whole FPGA coreboard should look just like a large microchip, powered by standard TTL +5 volts. Everything else is on board. You just plug it in, give it power, program it using JTAG com interface and enjoy the FPGA system.

# 6.1 Analog time interval controller.

Our first control unit was purely analog and was designed as a single unit.

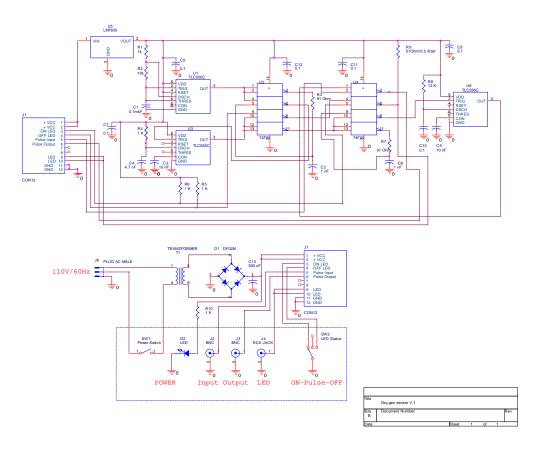


Fig. 42. Time Control Unit schematic. First prototype.

The time control unit consists of two parts: control part with power supply and digital time sequence unit. Control board with power supply implements the AC-DC conversion of 110V/60Hz AC into 12V DC. Bridge rectifier  $D_1$  and capacitor  $C_{13}$  support the whole device with necessary DC current. The power switch with light indication (SW<sub>1</sub> and  $D_2$ ), BNC connectors  $J_2$ ,  $J_3$ , and one RCA connector  $J_4$  a. are installed on the front panel. The LED control switch SW<sub>2</sub> is also installed on the device's front panel. The digital time sequence board connected with control board via 12-pin single row connector  $J_1$ . BNC connector  $J_2$  works as the input terminal for a stream of short TTL pulses from

the Single Photon Detector. After gating the selected pulses left the unit through the connector J<sub>3</sub>, and are counted by an external counter / frequency meter.

The control switch  $SW_2$  has three positions: LED OFF, LED ON and Pulse. The last one is used for actual measurement and the first two for the system calibration. Integrated timer  $U_1$  generates the clock signal for the whole time sequence unit generating rectangular TTL pulses with  $1400\mu S$  period. A combination of logical elements  $U_2$  is used for the generation of protective intervals between excitation pulse and so called "time window" for the fluorescent response collection. Timer  $U_3$  is a monovibrator defining the width of an excitation pulse (about  $140~\mu S$ ). Fast TTL chip  $U_4$  is used as a current driver for LED, and also as a fast switch for gating the pulses coming from the Single Photon Detector. Timer  $U_3$  is used for control of the gate (time window). A width of the time window is determined by the length of the excitation pulse and protection intervals.

To design the prototype print board layout OrCAD Layout Plus was used. We decided to implement the prototype using the bottom sided print board method with top layer jumpers using automated routing 1Sd\_VH\_JMPR.st strategy. The print board outline is presented on Fig.43.

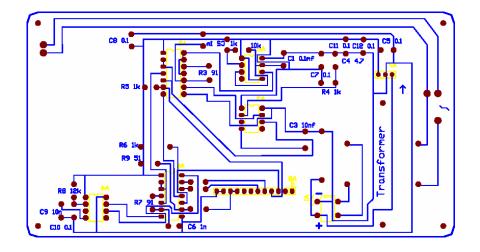


Fig. 43. Time Control Unit. Print board layout.

The print board was manufactured using LPKF S30 robotic prototyping system.

The appearance of the unit and implemented board is presented in Fig.44.

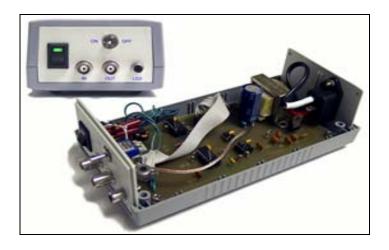


Fig. 44. Time Control Unit (TCU).

We used presented TCU for initial decay time measurements. In such measurements we integrated the intensity of fluorescent light inside time windows. In such a way time decay calculated as a function of an intensity of fluorescent light – it's oldest and simplest way for time decay measurements.

The drawback of such technique is the dependence on the optical losses in a line of light delivery to and from the sample. Because of this we started new designs with a primary aid to exclude all analog signals from both parts of the setup: from the controlling unit as well as from the processed signal itself. To make our system as fast as possible we selected an FPGA technique.

#### **6.2 FPGA implementation.**

Having got very good results in the experiment we decided to upgrade the existing control unit. Using analog unit we show that the technology worked and realized that we faced necessity to make the system more intelligent and more tolerant to the input signal disturbances, electromagnetic fields influences etc. Naturally, the new generation of the core unit was decided to be purely digital.

Working on research projects one should understand that especially in the first series of experiments the system should be very flexible to fit changing circumstances. This is why we came with the idea to avoid necessity of numbers of control unit redesigns by using programmable core. I made some investigation and came to the decision that the only technology which would fit us is FPGA.

Using microcontrollers and microprocessors was very inviting from the point of view of the user interface and visualization capabilities. I run through some systems offered on the market and found out that all of the microprocessor systems I was considering had two common disadvantages. The first one is that a microprocessor work is based on so named interruptions and this is why the reaction on an incoming event could not be immediate. The second problem is that the system resolution is limited by the processor's speed of operation and, say, to increase the resolution all the system would be totally redesigned using a faster processor. So all the flexibility of a microprocessor based system lays in the boundaries imposed by certain type and model of microprocessor.

FPGAs (Field Programmed Gate Arrays) are the microchips with reprogrammable logic. In fact, FPGA is an instrument which allows a designer to build a digital circuit

using its behavioral parameters. In most cases FPGA design is a behavioral programming of the system. Using special programming languages such as VHDL, Verilog, AHDL, SystemC, system designer operates with certain logical notations of system behavior elements instead of using gates as a building blocks of a digital circuit. System behavioral program can be converted into actual gate layout compatible with the FPGA microchip used. So the main advantages of FPGA technology in the case of our project are the following. System reaction time is minimal because FPGA is simply speaking just a digital circuit. And secondary, in the case of FPGA, to achieve a better resolution, one needs just to find faster FPGA chip leaving the most of software absolutely intact.

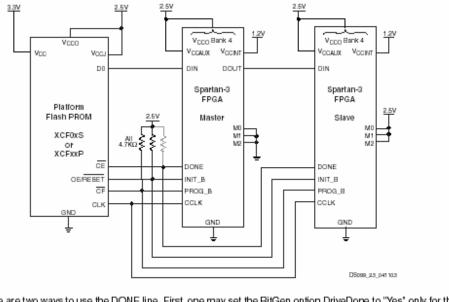
FPGA technology has disadvantages also. Programming FPGA is not as simple and fast as microprocessor one. The designer is supposed to care about arithmetical core of the system because mathematical functions of language are very limited. For instance, a division operation should be programmed using bit logic. Or a loop with variable length could not be synthesized either.

It was obvious that the new control unit was supposed to indicate the results. As a display I decided to use a 2x16 sequential LCD unit BPI-216 offered by Scott Edwards Electronics Inc. BPI-216 LCD modules combine a serial interface with a 2-line by 16-character LCD. The combination receives serial data at 2400 or 9600 baud and displays it on the LCD. The Unit has two options – text and instruction.

To make the system work with the chosen LCD display I was supposed to design a special unit inside FPGA which would implement a serial RS-232 interface with 9600 baud rate to show the results. Actually using this part of FPGA core opens some new interesting abilities of our system. For instance, because our device has a serial interface

implemented, it can be easily connected to a computer to transmit results and maybe that is more important, to receive commands from computer. In fact, this interface will definitely be used in our future experiments when statistical data will be collected and processed on a computer.

Having implemented certain amount of device generations I came to the decision that the best technology for the end used device is in fact a combination of FPGA and



- Notes:
- Notes:

  1. There are two ways to use the DONE line. First, one may set the BitGen option DriveDone to "Yes" only for the last FPGA to be configured in the chain shown above (or for the single FPGA as may be the case). This enables the DONE pin to drive High; thus, no pull-up resistor is necessary. DriveDone is set to "No" for the remaining FPGAs in the chain. Second, DriveDone can be set to "No" for all FPGAs. Then all DONE lines are open-drain and require the pull-up resistor shown in grey. In most cases, a value between 3.3 KΩ to 4.7 KΩ is sufficient. However, when using DONE synchronously with a long chain of FPGAs, cumulative capacitance may necessitate lower resistor values (e.g. down to 330Ω) in order to ensure a rise time within one clock cycle.
- For information on how to program the FPGA using 3.3V signals and power, see 3.3V-Tolerant Configuration Interface

Fig 45. Connection diagram for Master Slave Serial Configuration

microprocessor technology. The next generation will contain both FPGA and microprocessor. FPGA will be used for system response and control signal generation purposes. Microprocessor will be used for calculations and visualization.

All electronic design and modeling was made using OrCAD 10.0. I've chosen Xilinx FPGA and flash memory chips. As a PROM to FPGA Device interface configuration the Master-Serial Mode was used.

In Master Serial mode, the FPGA drives CCLK pin, which behaves as a bidirectional I/O. The FPGA in the center of Figure 45 is set for Master Serial mode and connects to the serial configuration PROM and to the CCLK inputs of any slave FPGAs in a configuration daisy-chain. The master FPGA drives the configuration clock on the CCLK pin to the Xilinx Serial PROM, which, in response, provides bit-serial data to the FPGA's DIN input. The FPGA accepts this data on each rising CCLK edge. After the master FPGA finishes configuring, it passes data on its DOUT pin to the next FPGA device in a daisy-chain. The DOUT data appears after the falling CCLK clock edge.

The Master Serial mode interface is identical to Slave Serial except that an internal oscillator generates the configuration clock (CCLK). A wide range of frequencies can be selected for CCLK, which always starts at a default frequency of 6 MHz. Configuration bits then switch CCLK to a higher frequency for the remainder of the configuration.

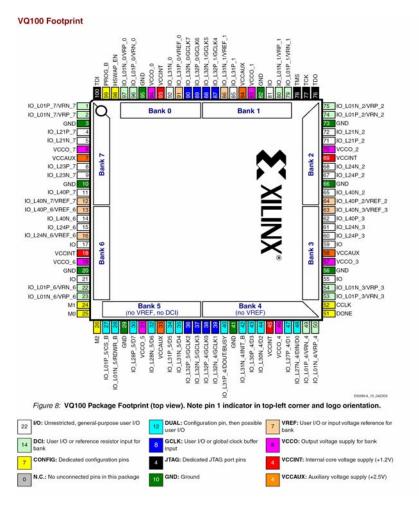


Fig. 46. Footprint and pinout for VQ100 package of Xilinx FPGA and actual view of the chip XC3S200.

The XC3S50 and the XC3S200 devices are available in different packages. The most convenient for us is 100-lead very-thin quad flat package, VQ100. Both devices share a common footprint for this package as shown in figure 46.

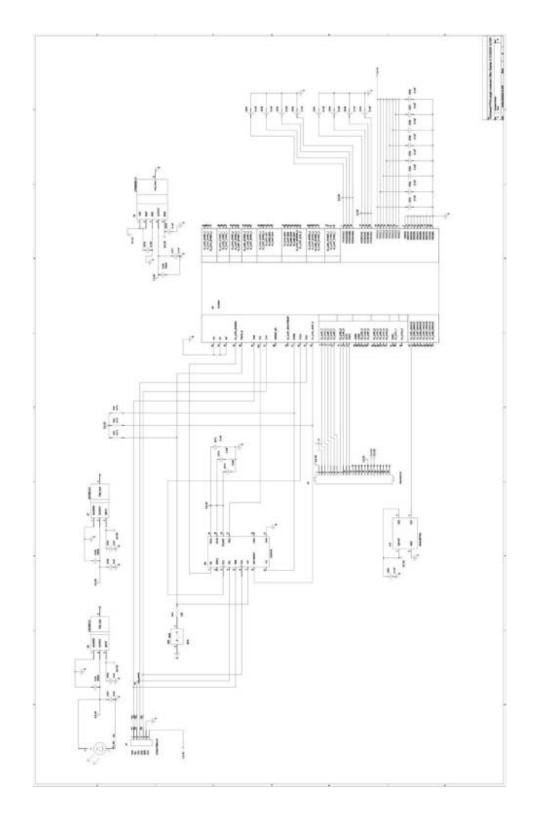


Fig. 47. FPGA based control unit schematic. Generations 1-3.

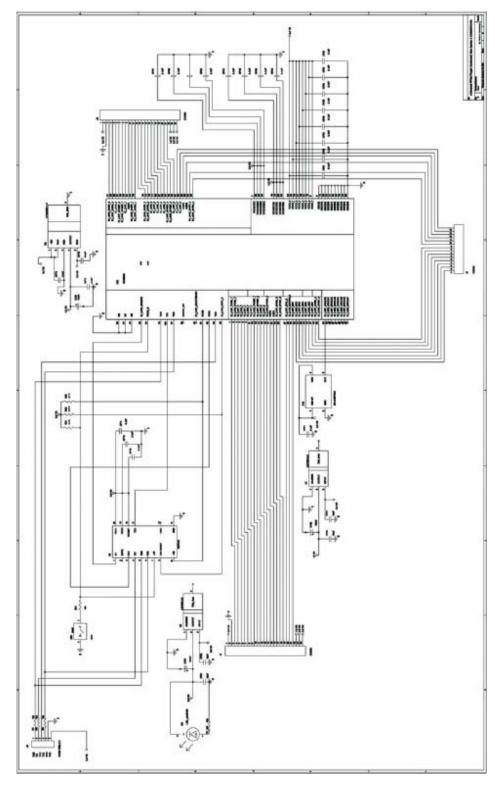


Fig. 48. Schematic of the generation 4 and 5 coreboards.

For the first generation of FPGA central unit Xilinx XC50S VQ-100 was chosen as the core and Xilinx XCF01S as a PROM. I designed a two-sided printboard and manufactured it using LPKF ProtoMAT C30S prototyping machine.

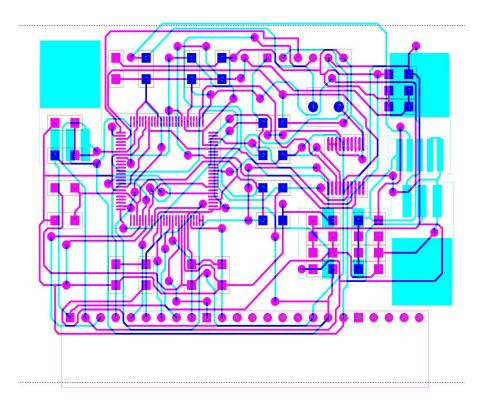


Fig. 49. FPGA Core unit layout. Generation 3.



Fig. 50. LPKF ProtoMat C30S. Making top layer of FPGA core unit.

LPKF ProtoMat

C30S is a mechanical engraving prototyping system used for rapid customized printboard fabrication.

FPGA technology gives amazing flexibility of electronic designs. Being developed once electronic circuit can be easily reprogrammed to meet absolutely different needs and solve absolutely different tasks. Operating with precise resonators FPGA systems work as extremely high quality digital generators. In my design I used 50MHz resonator to get 20 ns resolution in control signal generation.

# 6.3 FPGA board generations.

The idea of making an FPGA coreboard as a unit gave us perfect results. The main idea of this construction is that the coreboard is designed only once. After that you just plug it into your printboards or electrical breadboards. If you open a device I made for out research purposes you could see the coreboard plugged in and some minor circuitry – and this is the beauty of the idea. The coreboard can be represented just as a single element.

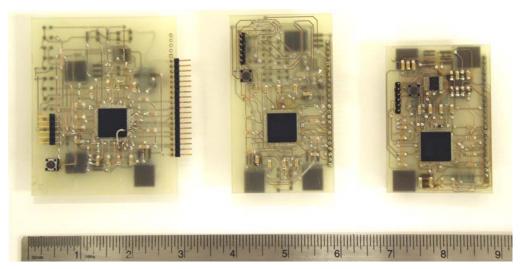


Fig. 51. Three first generations of FPGA coreboards.

While working on this project I designed five generations of the coreboards. The first three was using only nine data ports of the FPGA microchip. That was pretty much enough for the particular designs of that system, but I took a decision not to stop on that, but to make a real product out of the coreboard, utilizing all the data ports, offered by the microchip. So the forth and fifth generations have 52 input/output ports.



Fig. 52. Fourth generation of the FPGA coreboard.

I would like to notice here, that the printboards for the first three coreboard generations were made using OrCAD 10 automatic routing algorithms. When I started working on the forth and fifth generations with way more complexity and number of nodes and edges of corresponding schematic graph, to my surprise, it appeared that the

algorithms were unable to solve the routing task in two, even in four layers. I didn't have any choice but trace the whole board manually. And I'm really proud of the results because not only did I succeed, but also managed to route whole the device in only two



Fig. 53. Fifth generation of FPGA coreboard.

layers. That definitely proves that the artificial intelligence is still a bit behind the natural one.

The schematic of the fifth board generation is similar to the fourth, but the placement of components is changed very much. The idea was to place all small components in the top layer of the board leaving only large voltage regulators, oscillator

and large capacitors on the bottom layer. This makes the automatic manufacturing of the boards easier and faster, and helps to make the heat dissipation of the voltage regulators more effective.

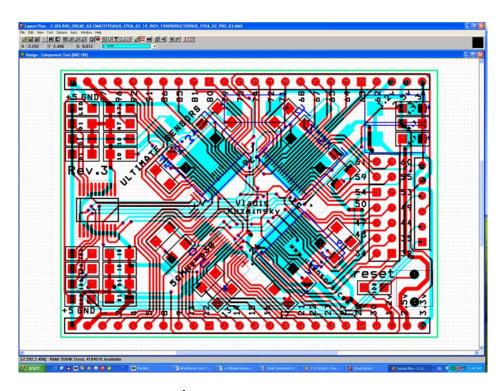


Fig. 54. Designing 5<sup>th</sup> generation FPGA coreboard's printboard.

The result of the printboard design is a set of GERBER files which represent the layers of the board. On the following figure I demonstrate the most important layers – two signal layers (top and bottom) and two solder mask layers (top and bottom). Another layer without which a manufacturing is impossible is so called throughole layer, where all the holes and the board in- and outline are defined.

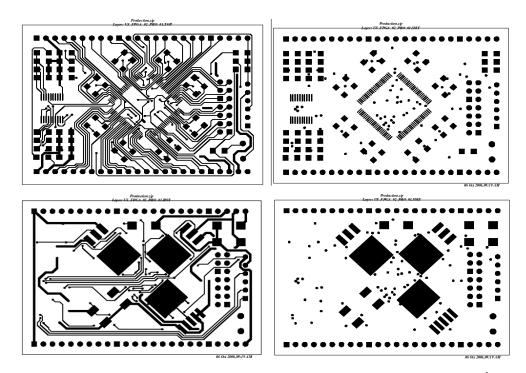


Fig. 55. Top layer with the soldermask, bottom layer with the soldermask.  $5^{\rm th}$  generation coreboard manufacturing file set.

In our designs I implemented a lot of different devices, starting with fully adjustable control signal sequence generator and ending with all-in-one two window photon integrator, photon's interval meter, and as a final generation of the devices –



Fig. 56. Fully adjustable control signal sequence generator with indication of control signal parameters and counting results.

FPGA and Microprocessor combinations utilizing speed and real-time abilities of FPGAs and flexibility and math apparatus of microprocessors.

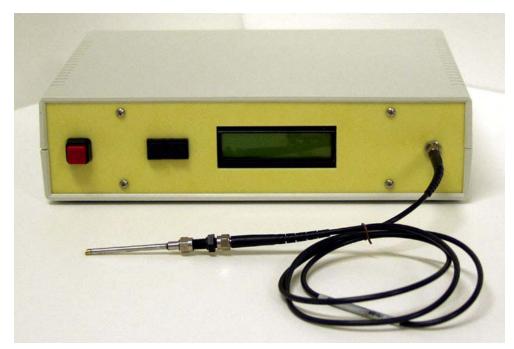


Fig. 57. Fully adjustable control signal sequence generator with indication of control signal parameters and counting results.

In first generations of setup the core unit was separated from the LED driver and photon detector. It generated control signal sequences for the LED driver and received impulse packets from the Single Photon detector to count and process them. In later generations of the device we implemented all-in one technology assembled LED, LED driver, regulation circuits, Single Photon Detector, optical splitter in one unit, having only one optical output for the sensor.

It was very important for us to switch from several devices to all-in-one construction because it considerably saves the design and fabrication expenses and decreases the size and reliability of the system on the whole. Note the blue coils which occupies almost whole the space of the case. It is an optical splitter offered on the market. It is expensive, large and hard to use in the real devices. To make our devices compact, reliable and cheap we invented our own line of differential optical splitters/combiners. Chapter 5 is mostly dedicated to the construction and principals of operation of those.

The device was build using motherboard-daughterboard technology. Layouts of the motherboard and LCD front panel board are shown in figure 58.

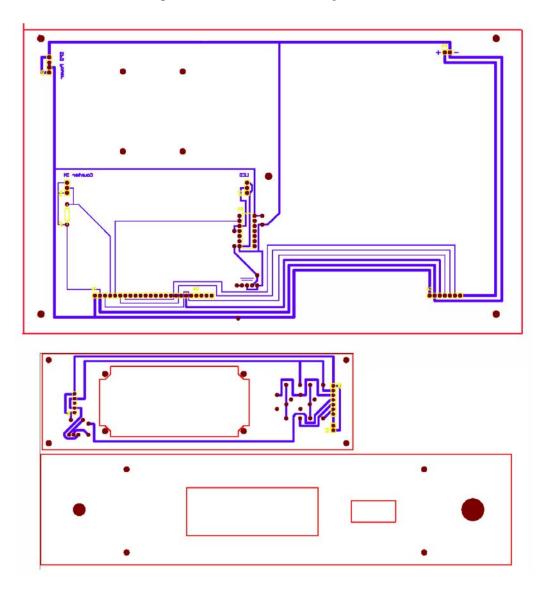


Fig. 58. Motherboard and front panel boards layouts.

The most important component of FPGA system is the software uploaded to the chip. I developed lots of versions of programming code for our devices. (See appendices). Xilinx supports their customers with necessary programs to develop code for FPGA microchips. So in our project we use ISE 6.2 software. As a programming language I took VHDL.

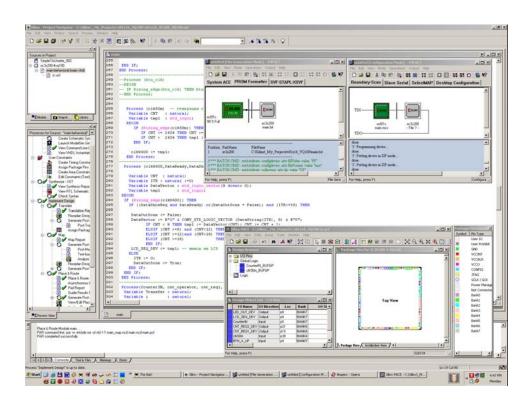


Fig. 59. Main screen of ISE software complex.

# 6.4. Microprocessor and FPGA.

The systems we designed were not complete from the point of view of the functionality. Those were not direct meters, but the meters of certain parameters used to find the measured value. In other words, our devices showed the number of photons in

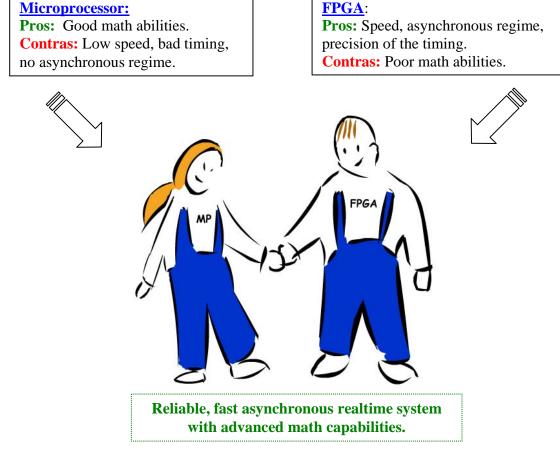


Fig. 60. Pros and contras of MP and FPGA, and the benefits of their co-use.

the intervals, time intervals between successive photons in a photon flux, but not the real value of the temperature or oxygen concentration. So having checked all the results of our experiments we started thinking about building a real meter, which would be able to calculate and show the real measured parameters. As I mentioned already, FPGAs are just a programmable circuits and this is the reason why they are very limited in math

abilities. Some of FPGAs have an implemented on the same chip microprocessor to widen the arithmetical function of the system. I made some research on this to find the right chip and redesign the system based on it. But what I found was that embedded microprocessors in FPGA chip are not as powerful and fast as we would like it to be. Another bad thing about it was the necessity to redesign the coreboard to match a new chip. It would be total redesign. So to make it simpler so far we decided to use a standalone microprocessor or a microcontroller to be exact.

The calculations were expected to be rather sophisticated, because of a probabilistic nature of photon emission. In fact, to get a considerably reasonable result one should perform a number of measurements and approximate those using known

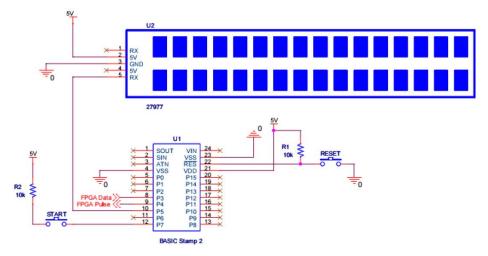


Fig. 61. Schematic of the use of Basic Stamp 2 microcontroller. Pins 8 and 9 are used for the data exchange with FPGA coreboard.

mathematical methods. In other words, we needed to make say ten thousand cycles of fluorescent parameter measurements, approximate them, and based on the so called statistical result calculate the decay parameter  $\tau$  and using a special experimentally made lookup table find the temperature or the oxygen concentration corresponding to it. So that came obvious the necessity of a rather advanced floating point calculation unit to work with such a large statistic.

Having considered all the pros and contras we agreed to use a tandem of FPGA and microprocessor. This would be a first time solution helping us to demonstrate the system functionality.

The concept of the new system I would describe in the following way. For each of 10000 iterations the microprocessor sends one short start pulse and waits for the values sent back by FPGA using RS-232 protocol.

Each time the microprocessor receives the data from FPGA unit, it approximates it, stores it and once it's done with that it generates a new start signal. Finished with all 10000 iterations, the microprocessor calculates decay parameter of the fluorescence process and according to that determines the temperature or the oxygen concentration

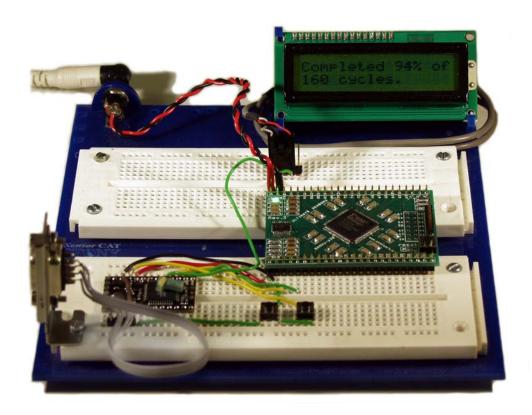


Fig. 62. FPGA + BasicX system mounted on an evaluation circuit board.

using the lookup table. After that it shows the results on the LCD screen.

On the FPGA's side, the FPGA waits for the start signal from the microprocessor and once it gets it, performs one cycle of the measurement. After that FPGA sends the measurement results to the microprocessor by

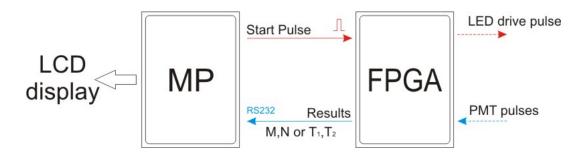
RS-232 serial communication protocol.

So I started with a very well known microcontroller Basic Stamp-2<sup>TM</sup>. It was very easy to program and test. But the problem was that even this microcontroller had limited math



Fig 63. Microcontroller Basic Stamp 2

abilities. To make it operate with floating point numbers an additional co-processor chip was supposed to be installed. The FPU (Floating Point Unit) is an eight pin microchip with minimum of supporting circuitry.



Generation of the start signals for each experiment iteration, accumulation of experimental data, calculation of the statistical results, calculation and demonstration of the final results. Generation of the LED drive signals, determination of the experiment parameters, transmission of the results to MP unit.

Fig. 64. The use of FPGA and MP combination in the project.

But even having all these we couldn't finalize the system. Because of the memory shortage, it was capable only to collect and approximate the data from the FPGA unit.

Another issue here was the speed. To perform all 10000 iterations it took about 30 minutes. So I took a decision to switch to another microcontroller which would be faster and better from the point of view of calculations. So I decided to try Basic-X<sup>TM</sup> microcontroller. It was faster, more flexible to program and didn't require a co-processor chip.

The next step to improve the system was PIC microprocessor. To use it we were supposed to design a special board in the style of FPGA coreboard with whole supporting circuitry installed on the board.

All program listings can be found in the appendices.

# 6.5. Portable Single Photon Counter.

One of the side products of the current research was a Portable Single Photon Counter. Our single photon counter (SPC) is a unique device which combines a top level photon registration system (based on HAMAMATSU<sub>(R)</sub> photodetectors) and the advanced digital circuitry and control system based on the FPGA coreboard. SPC is a portable hand-held unit with an LCD screen showing the measurement results.



Fig. 65. Portable Single Photon Counter

Single Photon

Counter allows you to register and count single photons in 1 sec, 10 sec or external (user defined) time window. The device has standard fiber input, BNC input for external counting window. The measurement

for 1 sec and 10 sec regimes starts by the "Start" button. Or in the case of "Extermal window" regime choosen, the photons are counted inside the first full external impulse. The results for external window regime are stored separately.

#### 7. <u>Time interval measurements.</u>

Method TIM is based on replacement of exponentially decaying process with one with exponentially growing time between times with equal intensity. The source of the signals of that sort is SPD and LIF converters (Light to Intensity converters)/

TIM method comes from the technique offered in patent US Pat. #16,944,407 (Luriy, Gorfinkel, Gouzman). The main idea is in determination of the time intervals in between electrical impulses. The results could be integrated (as in well known methods) or be calculated in one experiment. Because T=k/I for SPD and  $T=k/\rho_i$  for LIF, different data processing patterns could be applied.

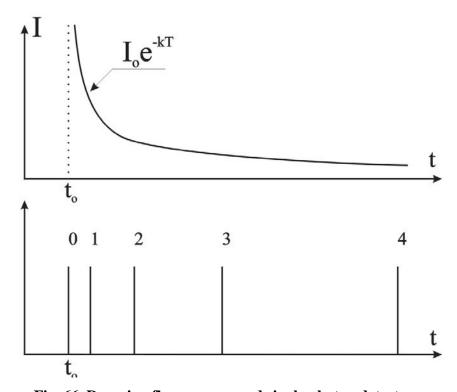


Fig. 66. Decaying fluorescence and single photon detector

Must be noticed, that in contrast to well known integration methods TIM method allows to:

- use extremely small illumination powers
- improve statistical stability of experiment results

Measuring a decay time in time resolve spectroscopy is a common technique. Early researchers [...] utilized a variety of methods for selection of fluorescent respond from excited light and purify the process of decay time measurement from interference with measurement distance, excitation and emission spectrums overlapping, and rising/falling characteristics of the setup.

One of the first responses on listed problems was our open space universal setup for decay time measurement. Simplified design of such setup aloud us to test common and new techniques in the same experimental environment (same light sources, optical systems and detection principle). As it was presented in previous parts we started from "classic" time-window technique. Large number of experiments was provided on an original setup designed by Dr. Oleg Semyonov and supported by Parker-Hannifin Corp. In frame of intensive experimental research Dr. Semyonov has demonstrated one of most fundamental limitations of time-window technique — low repeatability of results in experiments with low optical power. The weakness of such experiments is in small number of useful photons. In Dr. Semyonov's experiments such weakness was compensated by large number of repeatable measurements with hardware base integration. To avoid any further mistakes with LED based decay time measurements system we have provided an accurate calculation of all losses in tract of generation, delivery and registration of photons in decay-time measurement experiments. Such

calculation was provided under supervision of Dr. Nadia Lifshitz. In calculation was used a popular ruthenium complex based sensor (Ocean Optics, Fl).

Absorption and emission maxima of the ruthenium complex lie at about 450 and 650 nm, respectively. Let us first estimate the *efficiency of fluorescent excitation* for the Ru complex, that is, the ratio between the fluorescent and the excitation fluxes. It can be expressed as [23]:

$$K_f = I_f / I_{exc} = C \times k \times d \times Q$$
;

Here C is the fluorophore concentration,  $k = \ln 10 \times \varepsilon$ , ( $\varepsilon$  being the fluorophore extinction coefficient), Q the quantum yield, and d the thickness of the matrix. Taking the typical values  $C = 3 \times 10^{-5}$  mol/L [24],  $\varepsilon = 3 \times 10^4$  L·mol<sup>-1</sup>·cm<sup>-1</sup> [251], Q = 0.044 [26] and  $d=1\mu m$ , we find  $K_f$  on the order of  $10^{-5}$ .

The fluorescent decay can be calculated with sufficient accuracy (1%) when at least  $10^4$  single-photon counts are registered in a single measurement. We estimate that the efficiency of the fluorescent delivery is ~ 0.01 (with PMT efficiency ~ 10%, and other optical losses combined). Thus, we need at least  $10^6$  fluorescent photons to be emitted and hence the number of excitation photons necessary to produce this fluorescence is  $10^{11}$  photons and the excitation energy for one measurement cycle is  $6 \times 10^{-8}$  J. If the measurements were taken every second, the *average power required of the pulsed excitation source* is about 60 nW.

Unfortunate, most of generated inside fluorescent sample photons are lost because of time gate process (we cannot count generated in excitation time photons). As result, only 8-10% of all fluorescent photons can be used for measurements (decreasing the length of an excitation pulse we can increase this number).

To compare this calculation with real experiments we provided a simple intensity

measurement for similar fluorescent sensor. Such test was provided by Dr. Semyonov on his experimental oxygen sensor setup. As it presented on Fig. 67. in a single fluorescent intensity measurement we can detect only 4 to 8 events (registered photons). Because in Dr. Semyonov's experiments each measurement was repeated approximately 1,000 times per second, the total count will be close to the predicted by analytical calculations 10,000 (8,000 in our experiments).

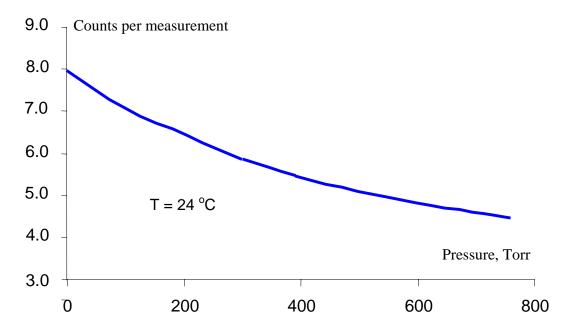


Fig. 67. Pressure dependence of fluorescent intensity of the ruthenium-complex probe in air.

#### 7.1. Explanations of time interval measuring technique.

Several years ago was proposed alternative technique of operation with photon flux. Such technique was patented by group of SUNY's leading researchers [US Pat. #16,944,407 "Method and apparatus for detecting radiation", Luryi Serge, Gorfinkel Vera, Gouzman Mikhail, Filed: Nov., 1998, Issue Date: September 13, 2005]. Dr. Semyonov proposed an advanced technique for implementation the main idea of the patent to the decay time measurement. He proposed to use information about all registered photons in form of three numbers: two time intervals between three selected photons and total number of registered after this event photons. Below we'll explain new technique.

The measured values are  $t_1$ ,  $t_2$  and/or  $N_{app}$ :

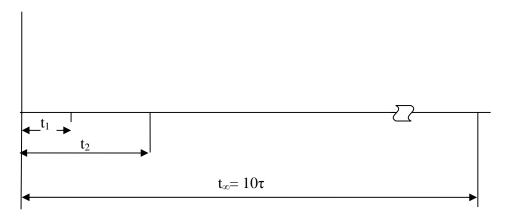


Fig. 68. Graphical interpretation of  $\tau$ -meter concept.

where  $t_1$  is the time interval between arrival of first photon (appearance of first count) and second photon (count),  $t_2$  is the time interval between arrival of first photon (count) and third photon (count), and  $t_{\infty}$  is the time interval for approximation  $N_{app}$  of counting of all photons  $N_{\infty} = I_0 \tau$  incident on a detector after the moment of detection of

first photon (appearance of first count) during a period of  $K \times \tau_0$  where  $I_0$  and  $\tau_0$  are the flux of luminescence incident of the detector and luminescence decay time at a given temperature and concentration taken as a constant . Accuracy of this approximation is:

1) If the measured values are  $t_1$  and  $N_\infty \approx N_{app}$  for K>5, from the formula for luminescence decaying exponentially

$$N = N_{\infty}(1 - e^{-t/\tau}),$$

where N is the number of photons (counts) detected during interval t, the decay time is easily calculated as:

$$\tau = t_1/\ln(N_{\infty}^{-1} - 1),$$

with the accuracy:

$$\frac{\Delta \tau}{\tau} = \frac{\Delta t_1}{t_1} + \frac{1}{N_{\infty} (1 - N_{\infty}^{-1}) \ln(1 - N_{\infty}^{-1})} \frac{\Delta N_{\infty}}{N_{\infty}} ,$$

which solves the problem of measuring any physical value depending on  $\tau$ .

The dependence of relative error  $\Delta\tau/\tau$  on  $N_{\infty}$  if  $\Delta t_1/t_1=1\%$  and  $\Delta N_{\infty}/N_{\infty}=1\%$  :

2) If the measured values are  $t_1$  and  $t_2$ , and corresponding numbers of detected photons (number of counts) are  $N_1=1$  and  $N_2=2$ , from two equations for number of photons:

$$N_1 = N_{\infty} (1 - e^{-\frac{t_1}{\tau}})$$
 and  $N_2 = N_{\infty} (1 - e^{-\frac{t_2}{\tau}})$ :

$$\frac{N_2}{N_1} = 2 = \frac{1 - e^{-\frac{t_2}{\tau}}}{1 - e^{-\frac{t_1}{\tau}}}$$
 or:

$$2e^{-\frac{t_1}{\tau}} - e^{-\frac{t_2}{\tau}} - 1 = 0 \tag{1}$$

In its general form this transcendent equation can be solved only numerically for two given (measured)  $t_1$  and  $t_2$ . The problem is that every time it will have a number of roots (solutions for  $\tau$ ) and it can create a difficulty of choice for a processing electronics. Obviously, the desired solution must be firstly real and secondly positive, so a processor can be programmed to choose only from those solutions. If such a solution for given  $t_1$  and  $t_2$  is only one, the problem of measuring of a physical parameter depending on  $\tau$  is solved. We have to investigate the possible solutions for some realistic  $\tau$  to be sure that the selected root is inside of area of physically available value of  $\tau$ .

Another way to calculate value of  $\tau$  is a 3 photon technique. Such model is presented in following part of the thesis.

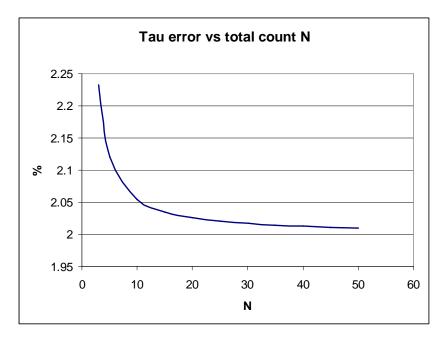


Fig. 69. Graphical interpretation of  $\tau$ -meter error.

# 7.2. Mathematical model for a 3 photon technique.

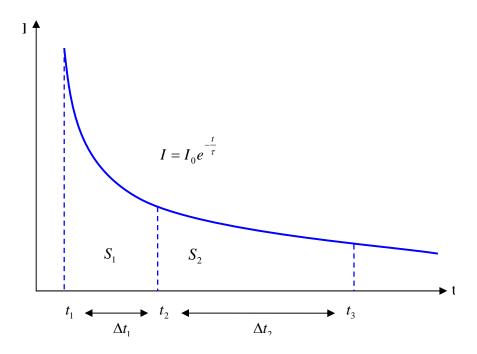


Fig. 70. Graphical presentation of the equal region simulation

$$S_{1} = \int_{t_{1}}^{t_{2}} e^{-\frac{t}{\tau}} dt = \tau \left( e^{-\frac{t_{1}}{\tau}} - e^{-\frac{t_{2}}{\tau}} \right)$$

$$S_{2} = \int_{t_{2}}^{t_{3}} e^{-\frac{t}{\tau}} dt = \tau \left( e^{-\frac{t_{2}}{\tau}} - e^{-\frac{t_{3}}{\tau}} \right)$$

$$S_{1} = S_{2}$$

$$e^{-\frac{t_{1}}{\tau}} - e^{-\frac{t_{2}}{\tau}} = e^{-\frac{t_{2}}{\tau}} - e^{-\frac{t_{3}}{\tau}}$$

$$e^{-\frac{t_{1}}{\tau}} - 2e^{-\frac{t_{2}}{\tau}} + e^{-\frac{t_{3}}{\tau}} = 0$$
Multiply by  $e^{\frac{t_{2}}{\tau}}$ :

$$e^{\frac{t_2 - t_1}{\tau}} - 2 + e^{\frac{t_2 - t_3}{\tau}} = 0$$

$$\Delta t_1 = t_2 - t_1$$

$$\Delta t_2 = t_3 - t_2$$

$$e^{\frac{\Delta t_1}{\tau}} + e^{\frac{\Delta t_2}{\tau}} = 2$$

$$\Delta t_2 = -\tau \ln \left( 2 - e^{\frac{\Delta t_1}{\tau}} \right)$$

where

$$\Delta t_1 > 0$$

$$2 - e^{\frac{\Delta t_1}{\tau}} > 0 \Longrightarrow \frac{\Delta t_1}{\tau} < \ln 2$$

#### 7.2. Generator – single photon source imitator.

Before going into the real experiments I decided to build a special system which would allow check the mathematical base of the concept. For that purpose I needed a special simulator for a decaying process with known decay constant registered by a single photon counter. Simply speaking, I needed a special generator producing a series of short pulses with an increasing distance between them so that to match a decaying process registered by a SPD. So to check the counting system I would just substitute the output of the SPD by the output of the generator with known  $\tau$  parameter, and just make sure that the calculated  $\tau$  matches the real one.

The generator was implemented on the FPGA coreboard using VHDL programming language. The program is attached to the report in the Appendix III.

As an example we provide (in form of a table) the pre-calculated sequence of "photon" pulses for simulated process of fluorescence decay. To simplify the calculations we selected  $\tau = 5 \text{mS}$ . The simulation results are presented in Table 1.

Serial	Time interval between	Absolute time of
number of the	current and previous "phonons".	"photon" arriving.
"photon".	μS	μS
1	0	0
2	100.00	100.00
3	102.04	202.04
4	104.17	306.21
5	106.38	412.59
6	108.70	521.29

7	111.11	632.40
8	113.64	746.04
9	116.28	862.32
10	119.05	981.36
11	121.95	1103.32
12	125.00	1228.32
13	128.21	1356.52
14	131.58	1488.10
15	135.14	1623.24
16	138.89	1762.13
17	142.86	1904.99
18	147.06	2052.06
19	151.52	2203.57
20	156.25	2359.83

Also the simulation result in a graphical form is presented in Fig. 71.in a linear time scale.

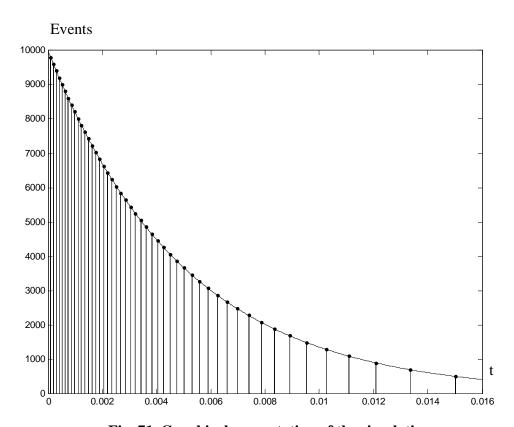


Fig. 71. Graphical presentation of the simulation

# 7.4. Full integrated implementation – "Three photon time interval meter/integrator".

In my first implementation of three photon interval meter I used a simple integration technique: in each act of illumination I calculated a value of  $\tau$  using the first three detected photons. After that I calculated the average value of  $\tau$  and its dispersion. Our first experiments were performed on an open space setup (see part 4.3.). Having got



Fig. 72. Compact integrated measurement system.

good results in the open space setup I built an integrated setup. It's presented on Fig. XC.

This device is designed using all our novel components and techniques.

In comparison with the open space one the compact system is 28 times less effective. The main reason is low optical efficiency in LED and the sensor (ruby ball) coupled to the fiber.

Because of the flexibility of FPGA structure, I had an opportunity to test three main configurations of the  $\tau$ -meter. These are the three main configurations which were implemented:

- Intensity analyzer;
- Two time-gaited window analyzer;

• Three photon time interval meter.

The results of the practical tests are presented in Fig. 73. On presented graph we

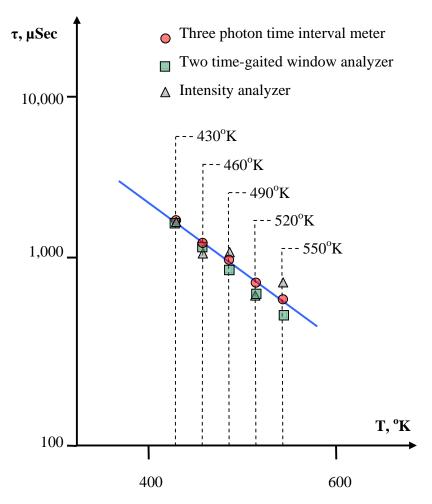


Fig.73. Temperature – decay time characterization of ruby crystal.

can see almost perfect matching to the theoretical line by three photon time interval meter. Note that the three photon time interval meter has some additional advantages:

- The method needs just tree photons per emission act for calculation of  $\tau$ ;
- If number of photons more then 3 we can analyze the sequence as a series of photon's "triad".

The drawback of the proposed technique is that the minimum number of photons is three and the following constraint for the time intervals -  $\Delta t_{2-3} > \Delta t_{1-2}$ . In contrast, the

older techniques allowed us to use smaller numbers of photons combined with the large number of measurements to collect enough statistical data.

#### **Conclusion**

This multidisciplinary work started with an idea of an improvement of existing methods of fluorescent detection and proceeded to completely novel technologies and instruments. In this long way we implemented a lot of new devices and instruments, performed series of experiments, developed and practically proved new methods and concepts, created several novel technologies and products.

In the scope of this research we finished several industrial projects, published and prepared several papers and filed several patents applications. The following goals were achieved:

- Two classic and one complete new fluorescent detection methods were implemented, tested and compared;
- The "Three Photon Technique" of the fluorescent decay time measurement was offered, modeled, implemented and tested for the first time;
- The concept of the universal FPGA coreboard was put into practice;
- Novel rapid prototyping technology was developed. This concept put the basis for a set of products ready for a mass production.

Among our industrial partners were the companies such as Parker Hannifin, D'Adario, North American String Association, Brookhaven Lab, Cooper Electric, Symbol Technologies, Keyspan, Emistar LLC.

The instruments and solutions which I developed while working on this project are widely used in other works of the Sensor CAT. I was the designer of the FPGA coreboards, author of all the configuration and modeling programs, constructor of the acrylic rapid prototyping devices, a supervisor of the robotic systems.

I'm glad that my designs opened new horizons for Optoelectronic Sensors and Systems Laboratory of SUNY Sensor CAT.

And with a special proud I would like to say that based on the work I've done we received an industrial grant and we are starting working on it in two weeks after my dissertation defense.

## **Bibliography**

- 1. I. M. Warner, G. D. Christian, and E. R. Davidson, "Analysis of multicomponent Fluorescence Data", Anal. Chem., 49, 564-573 (1977).
- 2. P. R. Rava, R. Richards-Kortum, M. S. Feld, J. J. Baraga "Contour mapping of spectral diagnostics" US patent 5.345,941, granted April 13, 1993
- 3. K. J. Albert and D. R. Walt, "High-Speed Fluorescence Detection of Explosives-like Vapors," Anal. Chem. 72, 1947-1955 (2000);
- 4. A. Zuluaga, U. Utzinger, A. Durkin, H. Fuchs, A. Gillenwater, R. Jacob, B. Kemp, J. Fan and R. R. Richards-Kortum, "Flurescence Excitation-Emission Matrices of human tissues: a system for in vivo measurements and method of data analysis", Appl. Spectroscopy, *53*(3), 302-311 (1999).
- 5. A. R. Muroski, K. S. Booksh, and M. L. Myrick, "Single measurement Excitation Emission matrix spectrofluometer for determination of hydrocarbons in ocean water", Anal Chem., 68, 3534-38 (1996).
- 6. R. D. Jiji, G. A. Cooper, K. S. Booksh "Excitation-emission matrix fluorescence based determination of carbamate pesticides and polycyclic aromatic hydrocarbons", Analytica Chimica Acta, *397*, 61-72 (1999).
- 7. Spex 3D from Jobin Yvon, \$40K, information available at the Jobin Yvon site http://www.jobinyvon.co.uk/jy/fluorescence/fluorescence.htm#ThreeD
- 8. R. Richard-Kortum, L. Tong, M. S. Feld, "Spectral diagnosis of diseased tissue", US patent 5.421,337, granted June 6, 1995.

- 9. R. A. Zangaro, L. Silveira, R. Manochoran, G. Zonios, I. Itzkan, R. R. dasari, J. Van Dam, M. S. Feld "Rapid multiexcitation fluorescent spectroscopy for *in vivo* tissue diagnostics", App. Optics, *35*(25) 5211-5219 (1996).
- 10. EPA publication EPA/600/R-99/041: J. E. Kenny, "Laser fluorescence EEM probe for Cone Penetrometer pollution analysis" United States Environmental Protection Agency, Office of Research and Development, Washington DC, June 1999.
- 11. C. Kieleck, B. Bousquet, G. Le Brun, J. Cariou, J. Lotrian, "Laser induced fluorescent imaging: application to groups of macroalgae identification", J. of Physics B, *34* 2561-2571 (2001).
- 12. S. J. Hart, R. D. Jiji, "Light emitting diode excitation emission matrix fluorescence spectroscopy", Analyst, *127*(12), 1693-1699 (2002).
- 13. C. D. Tran and R. J. Furlan, "Spectrofluorometer based on Acousto-Optics tunable Filter for rapid scanning of multicomponent sample analysis", Anal. Chem., 65, 1675-1681 (1993).
- J. Romier, J. Selves, J. Gastellu-Etchegorry, "Imaging spectrometer based on Acousto-Optics Tunable Filter", Rev. Scientific Instruments, 69(8), 2859-2867 (1998).
- 15. K. J. Zuzak; M. D. Schaeberle, E. N. Lewis I. W. Levin "Visible reflectance hyperspectral imaging: Characterization of a noninvasive, in vivo system for determining tissue perfusion", Anal. Chem., 74, 2021-2028 (2002).
- 16. H. R. Morris, C. C. Hoyt, P. J. Treado, "Imaging spectrometers for fluorescence and Raman microscopy acoustooptic and liquid crystal tunable filters", Appl. Spectroscopy, 48 (7), 857-866 (1994).

- 17. S. D. Russell R. L. Shimabukuro, A. D. Ramirez and M. G. Lovern, "Surface Plasmon tunable filter for multiband spectral imaging, SPAWAR Systems Center Biennial review 2001, pp. 117-121.
- Deconvolution with applications in spectroscopy. Edited by P. A. Jansson.
   Academic Press, New York, 1984, 342 p.
- 19. See calculations of the throughput of a monochromator, ORIEL manual, p. 4-13.
- A. P. Larson, H. Ahlberg, and S. Folestad, "Semiconductor laser induced fluorescence detection in picoliter volume flow cell", Applied Optics, 32(6), 794-805 (1993).
- 21. C. V. Bindhu, S. S. Harilal, V. P. N. Nampoori, and C. P. G. Vallbhan, "Solvent effect on absolute fluorescence quantum yield of Rhodamine 6G determined used transient thermal lens technique", Modern Physics Letters B, *13* (16), 563-574 (1999).
- 22. DNA Technology Technical Bulletin. Fluorescence excitation and emission.
  <a href="http://www.idtdna.com/program/techbulletins/Fluorescent">http://www.idtdna.com/program/techbulletins/Fluorescent</a> Dye Labeled Oligonu cleotides.asp
- K.T.V. Grattan, Z.Y.Zhang "Fiber optic fluotescence thermometry" Chapman & Hall, 19-24, 1995
- N. Stromberg and S. Hulth, An ammonium selective fluorosensor based on the principles of coextraction, *Annal. Chem. Acta*, 443(2) (2001) 215.
- N. Stromberg and S. Hulth, A fluorescence ratiometric detection scheme for ammonium ions based on the solvent sensitive dye MC 540, *Sensors and Actuators B*, 90(1-3) (2003) 308.
- 26 S. Huls, R. C. Aller, P. Engstrom, et al., *Limnol. Oceanogr.*, 47(1) (2002) 212.

- 27 N. Opitz, P. J. Rothwell, B. Oeke, and P. S. Schwille, Single molecule FCS-based oxygen sensor (O<sub>2</sub>-FCSensor): a new intrinsically calibrated oxygen sensor utilizing fluorescence correlation spectroscopy (FCS) with single fluorescent molecule detection sensitivity, *Sensors and Actuators B*, 96(1-2) (2003) 460.
- U. E. Spitchiger-Keller, Chemical Sensors and Biosensors for Medical and Biological Applications, Wiley-VCH, New York, 1998.
- J.R. Lakowicz (ed), *Topics in Fluorescence Spectroscopy*, Vol.4, Probe Design and Chemical Sensing, Plenum Press, New York ,1994.
- 30 A. Sharma and S. G. Schulman, *Introduction to Fluorescence Spectroscopy*, Wiley-Interscience, New York, 1999.
- J. R. Bacon and J. N. Demas, Determination of oxygen concentrations by luminescence quenching of a polymer-immobilized transition-metal complex, *Anal. Chem.*, 59 (1987) 2780.
- 32 G. Vishnoi, M. Morisawa, S. Muto, *Optical Review*, 5(1) (1998) 13.
- H. Chuang and M. A. Arnold, Linear calibration function for optical oxygen sensors based on quenching of ruthenium fluorescence, *Anal. Chem. Acta*, 368 (1998) 83.
- J. N. Demas and B. A. DeGraff, in J. R. Lakowitcz (ed.), *Topics in Fluorescent Spectroscopy*, Vol.4, Plenum Press, New York, 1994, pp. 71-107.
- W. Xu, K.A. Kneas, J.M. Detmas, and B.A. DeGraff, Oxygen Sensors Based on Luminescence Quenching of Metal Complexes: Osmium Complexes Suitable for Laser Diode Excitation, *Anal. Chem.*, 68 (1996) 2605.

- Y. Amao, K. Miyakawa, I. Okura, Novel optical oxygen sensing device: a thin film of a palladium porphyrin with a long alkyl chain on an alumina plate, *J. Mater. Chem.*, 10 (2000) 305.
- A. Mills and F.C. Williams, Chemical influences on the luminescence of ruthenium diimine complexes and its response to oxygen, *Thin Solid Films*, 306 (1997) 163.
- 38 I. Gryczynski, Z. Gryczynski, G. Rao, and J.R. Lakowicz, *Analyst*, 124 (1999) 1041.
- 39 R.N. Ghosh, D.G. Osborn, and G.L. Baker, *Proceedings of IEEE Sensors*, 22-2, 807, 2003.
- 40 M. E. Lippitsch, S. Draxler, and D. Kieslinger, Luminescence lifetime-based sensing: New materials, new devices, *Sensors and Actuators B*, 38-39 (1997) 96.
- S. B. Bambot, R. Holavanahali, J. R. Lakowiwicz, G. M. Carter, and G. Rao, Phase fluorometric sterilizable optical oxygen sensor, *Biotechnol. Bioeng.*, 43 (1994) 1139.
- W. Trettnak, C. Kolle, F. Reininger, C. Dolezal, and P.O. Leary, Miniaturized luminescence lifetime-based oxygen sensor instrumentation utilizing a phase modulation technique, *Sensors and Actuators B*, 35-36 (1996) 506.
- C. McDonagh, C. Kolle, A. K. McEvoy, D. L. Dowling, A. A. Cafolla, S. J. Cullen, and B. D. MacCraith, Phase fluorometric dissolved oxygen sensor, *Sensors and Actuators B*, 74(1-3) (2001) 124.
- 44 R. M. Clegg, P. C. Schneider, and T. M. Jovin, Fluorescence Lifetime-Resolved Imaging Microscopy, in A.M. Verga Schleggi, S. Martellucci, A.N. Chester, and R. Pratesi (ed.), *Biomedical Optical Instrumentation and Laser-Assisted*

- *Biotechnology*, NATO ASI Series E: Applied Sciences Series, Vol. 325, Kluwer Academic Publishers, Boston, 1996, p. 143.
- S. J. Pember, C. M. France and B. E. Jones, A multiplexed network of optically powered, addressed and interrogated hybrid resonant sensors, Sensors and Actuators A: Physical, 1995, v. 47, pp. 474-477
- E. Nieuwkoop, Th. Kapsenberg, G.K. Steenvoorden, and A.J.A. Bruinsma, Optically Powered Sensor System Using Conventional Electrical Sensors, SPIE Proceedings, v. 1511, Fiber Optic Sensors: Engineering and Applications, (1991).
- 47 R. Peňa, C. Algora, J.R. Matias, and M. Lopes-Amo, Applied Optics, 1999, 38, #12, 2463-2466.
- 48 S. Weiss, J. Werthen, and A. Anderson, Optically Powered Sensor Technology, ISA Conf. Proceedings, 1997, Orlando, Florida.
- 49 T.C. Banwell, R.C. Estes, L.A. Reith, P.W. Shumate, Jr., E.M. Vogel, Journal of Lightwave Technology, 1993, 11, #3, 481-494.
- T. Yasui, J. Ohwaki, and M. Mino, A Stable 2-W Supply Optical Powering System, Transactions of the Institute of Electrical Engineers of Japan, 2002, 122, #7, 1089-1093.
- 51 Photonic Power Systems, Inc. website, www.photonicpower.com
- Pilling, N.A. Holmes, R. Jones, G.R., Optically powered hybrid current measurement system, Electronics Letters, 1993, 29, #12, 1049-1051
- New optically powered ultravoltage current transformer

  Wang, Yutian, Wang, Litian, Shi, Jinshan, Zheng, Longjiang, Proc. SPIE 1998,

  Vol. 3555, p. 13-17, Optical and Fiber Optic Sensor Systems

- Feng, Maria Q., Optically Powered Electrical Accelerometer and Its Field

  Testing, Journal of Engineering Mechanics, 1998, 124, No. 5, 513-519
- 55 D.P. Hilliard, US Patent No. 5,389,762, (1995)
- Data Protocol for Optically Powered Sensors, NASA Tech Briefs, Vol. 21, No. 6, p. 6a, (1997).
- S. P. Jackson, H. Kirkham, and Col. McLyman, Transmitting Power to Sensor

  Circuitry via Modulated Light, NASA Briefs:

  <a href="http://www.nasatech.com/Briefs/Feb98/NPO19603.html">http://www.nasatech.com/Briefs/Feb98/NPO19603.html</a>
- S. P. Jackson and H. Kirkham, Transmitting Data Signals via Fiber-Optic Power Lines, NASA Briefs: <a href="http://www.nasatech.com/Briefs/Feb98/NPO19605.html">http://www.nasatech.com/Briefs/Feb98/NPO19605.html</a>
- Uzi Efron (Editor), Spatial Light Modulator Technology: Materials, Devices, and Applications (Optical Engineering, Vol 47), Marcel Dekker; (September 29, 1994)
- Zhang, Z. Y., Grattan, K. T. V. and Palmer, A. W. (1992). Sensitive fiber optic thermometer using Cr: LiSAF fluorescence for bio-medical sensing applications.

  [Proceedings of 8th Optical Fiber Sensors Conference, Monterey, California, January, pp. 93-96.
- Zhang, Z. Y., Grattan, K. T. Y. and Palmer, A. W. (1992). Fiber-optic high-temperature sensor based on the fluorescence lifetime of alexandrite. *Review of Scientific Instruments* 63 (8): 3869-3873.
- Zhang, Z. Y., Grattan, K. T. Y. and Palmer, A. W. (1993). Thermal characteristics of alexandrite fluorescence decay at high temperatures, induced by a visible laser diode emission. *Journal of Applied Physics* 73 (7): 3493-3498.
- Kang, Z. Y. (1993). Fiber optic fluorescence thermometry. PhD thesis (London:

- City University).
- Zhang, Z. Y., Grattan, K. T. Y. and Palmer, A. W. (1991). A novel signal processing scheme for a fluorescence based fiber-optic temperature sensor.

  \*Review of Scientific Instruments 62 (7): 1735-1742.
- Wickersheim, K. A. (1992). Fiberoptic thermometry: aπ overview. <u>In:</u>
  Temperature Its Measurement and Control in Science and Industry, Vol. 6, Part
  2 (New York: American Institute of Physics), pp. 711-714.
- Wickersheim, K. A. and Sun, M. H. (1987). Fiberoptic thermometry and its applications. *Journal of Microwave Power* 22 (2): 85-94.
- Wickersheim, K. A. and Hyatt, W. D. (1990). Commercial applications of fiberoptic temperature measurement. *SPIE Proceed in Rs* 1267: 84-96.
- Wickersheim, K. A. and Alves, R. V. (1979). Recent advances in optical temperature measurement. *Research & Develoyment* 21: 82.
- 69 Wickersheim, K. A. (1986). A new fiber optic thermometer system for use in medical hyperthermia. *SPIE Proceedin!!.s* 713: 150-157.
- Wickersheim, K. A. (1991). Application of fiber optic thermometry to the monitoring of winding temperatures in medium and large power transformers. SPJE Proceedings 1584: 3-14.
- Payne, S. A., Chase, L. L. and Wilke, G. D. (1989). Optical spectroscopy of the new laser materials, LiSrAIF<sub>6</sub>:Cr<sup>3+</sup> and LiCaAIF<sub>6</sub>:Cr<sup>3+</sup>. *Journal of Luminescence* 44 (3): 167-176.
- McClure, D. S. (1962). Optical spectra of transition-metal ions in corundum. *The Journal of Chemical Physics* 36 (10): 2757-2779.
- Harmer, A. L. (1986). Fiber optic sensors for offshore process control

- instrumentation. <u>In:</u> Proceedings of Optical Fiber Sensors Conference (OFS '86), Informal Workshop at Tusukuba Science City, 1986, YH (Pub: Institute of Electronics and Communication Engineers of Japan, *Tokyo*).
- Grattan, K. T. Y., Selli, R. K. and Palmer, A. W. (1988). Ruby decay-time fluorescence thermometer in a fiber-optic configuration. *Review of Scientific Instrument* 59 (8): 1328-1335.
- Grattan, K. T. Y. and Palmer, A. W. (1985). Infrared fluorescence 'decay-time' temperature sensor. *Review o! Scientific Instruments* 56 (9): /784-1787.
- Grattan, К. Т. У., Palmer, A. W. and Willson, С. А. (1987b). A miniaturized microcomputer-based neodymium 'decay-time' temperature sensor. *Journal of Physics E: Scientific Instruments* 20: 12011205.
- 77 Sclli, R. K. (1987). Fiber optic temperature sensors using fluorescent phenomena.

  \*PhD Thesis\*\* (London: City University).
- Bosseltann, T., Reule, A. and Schroder, J. (1984). Fiber-optic temperature sensor using fluorescence decay time. <u>In:</u> Proceedings of 2nd Conference oπ Optical Fiber Sensors (OFS '84). *SPJE Proceedings* 514: 151-154.
- Augousti, A., Grattan, K. T.Y. and Palmer, A. W. (1987). A laser-pumped temperature sensor using fluorescent decay tire of alexandrite. *Journal of Lightwave Technology LT-5* (6): 759-762.
- Augousti, A. T., Grattan, K. T. Y. and Palmer, A. W. (1988). Visible LED pumped fiber-optic temperature sensor. *JEEE Transactions on Instrumentation and Measurement* 37 (3): 470-472.
- Noel, B. W., Turley, W. O., Lewis, W., Tobin, K. W. and Beshears, O. L. (1992).

  Phosphor thermometry on turbine-engine blades and vans. [In: Temperature Its

- measurement and control in science and industry, ed. By J. F. Schooley, vol. 6, part I and 2, pp. 1249-1254.
- Walling, J., Peterson, O. G., Jenssen, H. P., Morris, R. C. and O'Dell, E. W. (1980). Tunable alexandrite lasers. *IEEE Journal of Ouantum Electronics* OE-16 (12): 1302-1315.
- Fonger, W. H. and Struck, C. W. (1975). Temperature dependences of Cr+3 radiative and nonradiative transitions in ruby and emerald. *Physical Review B* 11 9: 3251-3260.
- Armagan, G. and Bartolo, B. O. (1988). Mechanism for the thermal dependence of the Cr (o Nd energy transfer in garnets. /EEE Journal ofOuantum Electronics 24 (6): 974-978.
- Healy, S. M., Donnelly, C. J., Glynn, T. J., Imbusch, G. F. and Morgan, G. P. (1990). Temperature dependence of the luminescence of GSGG:Cr<sup>3+</sup>. *Journal of Luminescence* 46 (1): 1-7.
- Andrews, L. J., Beall, G. H., and Lempicki, A. (1986). *Journal of Luminescence* 36: 65-75.
- Rasheed, F., O'Donnell, K. P., Henderson, B. and Hollis, O. B. (1991). Disorder and the optical spectroscopy of Cr<sup>3+</sup> -doped glasses: 1. Silicate glasses. *Journal of Physics Condensed Matter* 3 (12): 1915-1930.
- 88 Henderson, B., Yamaga, M., Gao, У. and O'Ooппell, K. P. (1992). Disorder and nonradiative decay of Cr<sup>3+</sup> -doped glasses. *Physical Review B* 46 (2): 652-661.
- Draxler, S. and Lippitsch, M. E. (1993). Optical sensors based oπ fluorescence decay time. *Proceedings of the Symphosium* on *Chemical Sensors* II, *Proceedings* 93-7, ed. by M. Butler *et al.* (Pennington: The Electrochemical Society): 69 -73.

- 20 Lippitsch, M. E., Pusterhofer, J., Leiner, M. J. P. and Wolfbeis, O. S. (1988).
  Fiber-optic oxygen sensor with the fluorescence decay time as the information carrier. *Analytica Chimica Acta* 205 (1-2): 1-6.
- 91 Menoni C. S. and Spain, I. L. (1983). Ultra-high pressure measurement. *in:* High Pressure Measurement Techniques, ed. by G. N. Peggs (London and New York: Applied Science Publishers): 125-175.
- Holzapfel, W B. (1991). High pressure (above 2GPa). <u>In:</u> Units and Fundamental Constants in Physics and Chemistry, ed. "y J. Bortfeldt and B. Kramer, pp. 2-177 2-184 (Berlin: Landoldt-Boernstein).
- 93 Sato-Sorensen, V. (1986). Measurements of the lifetime of the ruby R I line under high pressure. *Journal of Applied Physics* 60 (8): 2985-2987.
- Sato-Sorensen, Y. (1987). Measurements of the lifetime of the ruby RI line and its application to high-temperature and high-pressure calibration in the diamondanvil cell. In: High-Pressure Research in Mineral Physics, edited by M. H. Manghnani and Y. Syono (Washington OC Tena Scientific, Tokyo / American Geophysical Union, pp. 53-59.
- Eggert, J. H, Goettel, K. A. and Silvera, 1. F. (1989). Ruby at high pressure II. Fluorescence lifetime of the R line to 130 GPa. *Physical Review B* 40 (8): 5733-5738.
- 6 Kottke, T. and WiHiams, F. (1983). Pressure dependence of the alexandrite emission spectrum. *Physical Review B* 28 (4): 1923-1927.
- Jia, W., Shang, Y. S., Tang, R. M. and Yao, Z. Y. (1984). Pressure effects of fluorescence R-lines of chrysoberyl BeAI204:Cr<sup>3+</sup>. *Journal of luminescence* 31 & 32: 272-274.

- Jahren, A. H., Kruger, M. B. and Jeanloz, R. (1992). Alexandrite as a high-temperature pressure calibrant, and implications for the ruby fluorescence scale. *Journal oz ADDlied Physics* 71 (4): /579-1582.
- 99 Shen, У. R., Gregorian, Т. and Holzapfel, W. B. (1991). Progress in pressure measurements with Ішпіпессепсе sensors. *High Pressure Research* 7: 73-75.

#### **Appendices**

## Program CA7. Control unit for fluorescent thermometer.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD LOGIC ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--library UNISIM;
--use UNISIM.VComponents.all;
-- CA07 <- CA03 <- CJ15
-- August 7. Number of iterrations is 3000
-- Vladis Kuzminsky
-- This version differs of CJ15 by increased delay between
LED pulse
-- and windows.
-- Counter for fluorescent thermometer.
-- Implemented parametrical timing definition
-- Cycle starts by button, performs once and stops until
-- the button is pressed again.
-- The screen is refreshed ONLY in the end of the
measurement cycle.
-- So to implement a real-time parameter change - the
timing system
-- should be redesigned to scan the parameters' buttons and
refresh
-- the screen.
-- APPROVED
entity main is
    Port ( clk50m : in std_logic;
           LCD_SEQ_DEV : out std_logic;
           CounterIN : in std_logic;
           LED_OUT_DEV : inout std_logic;
           BTN_1 : in std_logic;
           CNT_REQ1_DEV : out std_logic;
           CNT_REQ2_DEV : out std_logic;
           CONTROL1 : out std_logic;
           CONTROL2 : out std_logic;
           CONTROL3 : out std_logic;
           CONTROL4 : out std_logic;
           CONTROL5 : out std_logic);
```

```
end main;
architecture Behavioral of main is
  Type LCDSymbolArray is array(0 to 39) of natural;
  Type LCDNumberArray is array(0 to 6) of natural;
Signal RST : std_logic;
Signal clk_gen_counter : std_logic_vector(5 downto 1);
Signal strobe : std_logic; -- generator with largest
frequency
Signal strobe2x : std_logic;
Signal strobe4x : std_logic;
Signal strobe8x : std_logic;
Signal strobe16x: std_logic;
Signal strobe32x: std_logic;
Signal clk9600 : std_logic;
Signal btn_lock : std_logic;
Signal LED_LOCK : std_logic;
Signal start_sig : std_logic := '0';
Signal start_sig1: std_logic := '0';
Signal start_seq : std_logic; -- min strobe and BTN_1
Signal s1,s2,s3 : std_logic;
Signal W1,W2 : std_logic;
  Signal cnt_done : std_logic;
  Signal cnt_rst
                    : std_logic;
                    : std_logic;
  Signal cnt_req1
  Signal cnt_req2
                     : std logic;
  Signal DataReady : boolean := False;
  Signal DataShowReq : Boolean := False;
  Signal DataOutDone : boolean := True;
  Signal DataString : LCDSymbolArray;
  Signal LCDNumber1 : LCDNumberArray;
  Signal LCDNumber2 : LCDNumberArray;
Signal show_req : std_logic;
Signal lcd_busy : std_logic;
Signal WRK : boolean;
  Signal High
                    : std_logic;
  Signal Low
                    : std_logic;
  Signal FullCycle : integer;
```

```
begin
High <= '1';
Low <= '0';
  CONTROL1 <= start_sig;</pre>
  CONTROL2 <= start_seq;</pre>
  CONTROL3 <= cnt req1;
  CONTROL4 <= cnt_req2;</pre>
  CONTROL5 <= clk9600;
-- CONTROL1 <= High;
-- CONTROL2 <= High;
-- CONTROL3 <= High;
-- CONTROL5 <= High;
-- CONTROL4 <= High;
  strobe2x <= clk_gen_counter(1);</pre>
  strobe <= clk50m;</pre>
  start_seq <= strobe2x and BTN_1 and btn_lock;</pre>
Process (clk50m) is
BEGIN
 IF rising_edge(clk50m) THEN
    clk_gen_counter <= clk_gen_counter + 1;</pre>
 END IF;
END Process;
Process (clk50m, start_sig, btn_lock)is
 Variable Lock : boolean;
 Variable cnt : integer;
 Variable BTNLOCK : integer;
BEGIN
  IF RST = '1' THEN btn_lock <= '1'; cnt := 0;</pre>
  ELSIF rising_edge(clk50m) THEN
    IF FullCycle < 25000000 THEN BTNLOCK := 25000000;</pre>
    ELSE BTNLOCK := FullCycle;
    END IF;
     IF start_sig = '1' THEN Lock := true; END IF;
     IF Lock THEN
         cnt := cnt + 1;
         IF cnt >=2 and cnt < BTNLOCK THEN btn_lock <= '0';</pre>
END IF;
        IF cnt >= BTNLOCK THEN btn_lock <= '1'; Lock :=</pre>
false; cnt := 0; END IF;
     END IF;
  END IF;
END Process;
```

```
Process (clk50m, start_seq)is
BEGIN
  IF RST = '1' THEN
     s1 <= '0';
     s2 <= '0';
     s3 <= '0';
     start_sig <= '0';
  ELSIF rising_edge(clk50m) THEN
     s1 <= s2;
     s2 <= s3;
     s3 <= start_seq;
    start_sig <= (not(s1))and(not(s2))and s3 ;</pre>
  END IF;
END Process;
Process (clk50m) -- генерация clk9600 для RS232
 Variable CNT : natural;
 Variable tmp1 : std_logic;
BEGIN
  IF (Rising_edge(clk50m)) THEN
    IF CNT <= 2604 THEN CNT := CNT + 1; END IF;
    IF CNT > 2604 THEN tmp1 := not tmp1; CNT := 0; END IF;
  END IF;
  clk9600 <= tmp1;
END Process;
----- END generators -----
Process (clk50m) --- RESET
Variable i : integer;
Variable CNT : integer;
Variable DONE : integer;
BEGIN
 If rising_edge(clk50m) THEN
  If DONE = 2 THEN RST <= '0';</pre>
  ELSE
    CNT := CNT + 1;
    IF CNT = 2 THEN DONE := 2; END IF;
  RST <= '1';
  END IF;
 END IF;
END Process;
Process(cnt_done,RST)
BEGIN
  IF RST = '1' THEN
             DataString(13) <= 48;</pre>
             DataString(14) <= 48;</pre>
```

```
DataString(15) <= 48;</pre>
            DataString(16) <= 48;</pre>
            DataString(17) <= 48;
            DataString(18) <= 48;</pre>
            DataString(19) <= 48;</pre>
            DataString(33) <= 48;</pre>
            DataString(34) <= 48;
            DataString(35) <= 48;</pre>
            DataString(36) <= 48;</pre>
            DataString(37) <= 48;</pre>
            DataString(38) <= 48;</pre>
            DataString(39) <= 48;</pre>
ELSIF Rising_edge(cnt_done) THEN
            DataString(0) <= 254;</pre>
                                           -- CLS
                                           -- 1
            DataString(1) <= 128;</pre>
                              <= 254i
            DataString(2)
                              <= 128;
            DataString(3)
            DataString(4)
                              <= 32;
                                           ___
                              <= 32;
            DataString(5)
                              <= 32;
            DataString(6)
                              <= 32;
            DataString(7)
            DataString(8)
                              <= 32;
            DataString(9) <= 32;</pre>
            DataString(10) <= 32;</pre>
            DataString(11) <= 77;</pre>
            DataString(12) <= 61;</pre>
         --IF Work_BTN = '1' THEN
            DataString(13) <= LCDNumber1(0)+48;</pre>
            DataString(14) <= LCDNumber1(1)+48;</pre>
            DataString(15) <= LCDNumber1(2)+48;</pre>
            DataString(16) <= LCDNumber1(3)+48;</pre>
            DataString(17) <= LCDNumber1(4)+48;</pre>
            DataString(18) <= LCDNumber1(5)+48;</pre>
            DataString(19) <= LCDNumber1(6)+48;</pre>
         --END IF;
                              <= 254;
            DataString(20)
                              <= 192i
            DataString(21)
            DataString(22) <= 254;</pre>
            DataString(23) <= 192;</pre>
            DataString(24)
                              <= 67;
                               <= 65;
            DataString(25)
                               <= 48+0;
            DataString(26)
```

```
DataString(27) \leftarrow 48+7;
              DataString(28) <= 32;</pre>
              DataString(29) <= 32;</pre>
              DataString(30) <= 32;</pre>
              DataString(31) <= 78;</pre>
              DataString(32) <= 61;</pre>
          --IF Work_BTN = '1' THEN
              DataString(33) <= LCDNumber2(0)+48;</pre>
              DataString(34) <= LCDNumber2(1)+48;</pre>
              DataString(35) <= LCDNumber2(2)+48;</pre>
              DataString(36) <= LCDNumber2(3)+48;</pre>
              DataString(37) <= LCDNumber2(4)+48;</pre>
              DataString(38) <= LCDNumber2(5)+48;</pre>
              DataString(39) <= LCDNumber2(6)+48;</pre>
          --END IF;
            DataReady <= true;</pre>
  END IF;
END Process;
Process (clk9600, DataReady, DataShowReq, DataString) --
    Variable CNT : natural;
    Variable ITR : natural :=0;
    Variable DataVector : std_logic_vector(9 downto 0);
    Variable tmp1
                     : std_logic;
 BEGIN
  IF RST = '1' THEN
    DataOutDone <= False;</pre>
    ITR := 0;
  ELSIF (Rising_edge(clk9600)) THEN
    IF ((DataShowReq and DataReady) or(DataOutDone =
False)) and (ITR<=39) THEN
     DataOutDone <= False;</pre>
     DataVector := B"1" & CONV_STD_LOGIC_VECTOR
(DataString(ITR), 8) & B"0";
            IF CNT < 9 THEN tmp1 := DataVector(CNT); CNT :=</pre>
CNT + 1;
            ELSIF (CNT >=9) and (CNT<12) THEN tmp1 := '1';
CNT := CNT + 1;
            ELSIF (CNT >=12) and (CNT<18) THEN LCD_SEQ_DEV
<= High; tmp1 := '1'; CNT := CNT + 1;
            ELSIF (CNT >=18)
                                            THEN LCD SEQ DEV
<= High; tmp1 := '1'; CNT := 0; ITR := ITR + 1;
            END IF;
     LCD_SEQ_DEV <= tmp1; -- вывод на LCD
    ELSE
      ITR := 0;
      DataOutDone <= True;</pre>
```

```
END IF;
  END IF;
 END Process;
Process (clk50m, RST, CounterIN)
   Variable CNT : natural;
   Variable CNT1
                    : natural;
   Variable CNT LOC : natural;
   Variable CycleCNT : natural;
   Variable NImpulse : natural;
   Variable tmp1 : std_logic :='1';
   Variable tmp2 : std_logic :='0';
   Variable tmp3 : std_logic :='0';
   Variable tmp4 : std_logic :='0';
   Variable tmp6 : boolean;
   Variable tmp5 : std_logic;
   Variable tmp7 : std_logic;
   Variable tmp8 : std_logic;
   Variable LED_T : natural;
   Variable CNT_T : natural;
   Variable WRK : std_logic;
   Variable Del : natural;
   Variable Tled: natural;
   Variable Tdel: natural;
   Variable TW1 : natural;
   Variable TW2 : natural;
   Variable Tr : natural;
   Variable MC : natural;
   Variable NM : natural;
   Variable CC : integer;
BEGIN
IF RST = '1' THEN CycleCNT := 0; CNT := 0; CNT_LOC := 0;
ELSIF (Rising_edge(clk50m)) THEN CNT := CNT + 1; CNT_LOC
:= CNT_LOC + 1;
   Tled:= 2000 / 20; -- 2mks
   Tdel:= 10000 / 20; -- 10mks
   TW1 := 30000 / 20; -- 30mks
   TW2 := 300000 / 20; -- 300mks
   Tr := 2696000 / 20; -- 3000mks-300mks-2mks-2mks
   MC := Tled+Tdel+TW2+Tr; -- measurement cycle
   NM := 3000; -- iteration number
```

```
CC := MC*NM; -- experiment duration
  -- CC := 1500000;
   FullCycle <= MC*NM+2000+5005000; -- for button lock
 IF (start sig = '1') THEN
    WRK := '1';
    CNT := 0; CNT_LOC := 0; CycleCNT :=0;
 END IF;
 IF WRK = '1' THEN
 IF (CycleCNT < NM) THEN
-- LED_OUT_DEV
        (CNT_LOC >= 0) and (CNT_LOC < Tled) THEN tmp1 :=
   IF
low; END IF;
   IF
        (CNT_LOC >= Tled)
                                               THEN tmp1 :=
high; END IF;
 -- CNT_REQ1
   If (CNT_LOC >= 0) and (Cnt_LOC < Tled+Tdel)</pre>
THEN tmp2 := Low; END IF;
   If (CNT_LOC >= Tled+Tdel)and (Cnt_LOC < Tled+Tdel+TW1)</pre>
THEN tmp2 := high; END IF;
   If (CNT_LOC >= Tled+Tdel+TW1)
THEN tmp2 := low; END IF;
 -- CNT REQ2
   If (CNT_LOC >= 0) and (Cnt_LOC < Tled+Tdel)</pre>
THEN tmp3 := Low; END IF;
   If (CNT_LOC >= Tled+Tdel)and (Cnt_LOC < Tled+Tdel+TW2)</pre>
THEN tmp3 := high; END IF;
   If (CNT_LOC >= Tled+Tdel+TW2)
THEN tmp3 := low; END IF;
   If CNT_LOC >= (Tled+Tdel+TW2+Tr) THEN CycleCNT :=
CycleCNT + 1; CNT_LOC := 0; END IF;
  ELSE -- > NM
     tmp1 := high;
     tmp2 := low;
     tmp2 := low;
  END IF; -- CycleCNT
                    1 500 000 000
               ___
-- counter reset
   If (CNT >= 0) and (Cnt < CC + 2000)
                                           THEN tmp4 := Low;
END IF;
   If (CNT >= CC+2000) and (Cnt<CC+2000+10) THEN tmp4 :=
                END IF;
high;
```

```
If CNT >= CC + 2000 + 10
                                         THEN tmp4 := low;
END IF;
-- count done
   If (CNT >= 0) and (Cnt < CC + 1000)
                                         THEN
                                                tmp5 :=
              END IF;
   If (CNT \ge CC+1000) and (Cnt < CC+1000+5) THEN
                                                 tmp5 :=
High;
               END IF;
  If CNT >= CC + 1000 + 5
                                         THEN tmp5 :=
Low;
               END IF;
-- DataShowReq 5 000 000 = 100 milisec - time for LCD
data transmission
   If (CNT >= 0)and(Cnt < CC + 2000)
                                            THEN tmp6 :=
               END IF;
   If (CNT >= CC+2000) and (CNT < CC+2000+5000000) THEN tmp6 :=
true;
              END IF;
   If CNT >= CC+2000+5000000
                                            THEN
                                                   tmp6 :=
false;
               END IF;
END IF; -- WRK
  If CNT >= MC*NM+2000+5005000 THEN CNT := 0; CycleCNT
:= 0; CNT LOC :=0; WRK := '0'; END IF;
-- Impulses
  LED_OUT_DEV <= tmp1;</pre>
   CNT_REQ1_DEV <= tmp2;</pre>
   CNT_REQ2_DEV <= tmp3;</pre>
   cnt_req1 <= tmp2;</pre>
   cnt_req2
              <= tmp3;
   cnt_rst
               \leq tmp4;
   cnt_done <= tmp5;</pre>
   DataShowReq <= tmp6;</pre>
 END IF;
END Process;
_____
----- Process(CounterIN, cnt_req1, cnt_rst,
cnt done) is -- Counter1
  Variable Transfer : natural;
                  : natural;
 Variable i
 Variable LCDCNT : natural;
 BEGIN
    IF cnt_rst = '1' THEN
       LCDCNT := 0;
        LCDNumber1(0) <= 0;
       LCDNumber1(1) <= 0;
       LCDNumber1(2) <= 0;
       LCDNumber1(3) <= 0;
       LCDNumber1(4) <= 0;</pre>
       LCDNumber1(5) <= 0;
       LCDNumber1(6) <= 0;
```

```
ELSIF Rising_Edge(CounterIN) THEN
      IF cnt_req1 = '1' THEN
        LCDCNT := LCDCNT + 1;
        Transfer := 1;
        For i in 6 downto 0 LOOP
         IF LCDNumber1(i) < 9 and Transfer = 1 THEN</pre>
LCDNumber1(i) <= LCDNumber1(i) + Transfer; Transfer := 0;</pre>
         ELSIF Transfer = 1 THEN LCDNumber1(i) <= 0;</pre>
Transfer := 1;
        END IF;
       END LOOP;
      END IF;
   END IF;
 END Process;
_____
----- Process(CounterIN, cnt_req2, cnt_rst,
cnt_done) is -- Counter2
  Variable Transfer : natural;
  Variable i
                   : natural;
  Variable LCDCNT : natural;
 BEGIN
      IF cnt rst = '1' THEN
       LCDCNT := 0;
        LCDNumber2(0) <= 0;
        LCDNumber2(1) <= 0;</pre>
       LCDNumber2(2) <= 0;
        LCDNumber2(3) <= 0;
        LCDNumber2(4) <= 0;
        LCDNumber2(5) <= 0;
       LCDNumber2(6) <= 0;
   ELSIF Rising_Edge(CounterIN) THEN
      IF cnt_req2 = '1' THEN
        LCDCNT := LCDCNT + 1;
        Transfer := 1;
        For i in 6 downto 0 LOOP
         IF LCDNumber2(i) < 9 and Transfer = 1 THEN</pre>
LCDNumber2(i) <= LCDNumber2(i) + Transfer; Transfer := 0;</pre>
        ELSIF Transfer = 1 THEN LCDNumber2(i) <= 0;</pre>
Transfer := 1;
        END IF;
        END LOOP;
      END IF;
   END IF;
 END Process;
end Behavioral;
```

## **Program TIM1. Basic time interval meter.**

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--use UNISIM.VComponents.all;
-- Basic time interval meter.
entity main is
    Port ( clk50m : in std_logic;
           LCD_SEQ_DEV : out std_logic;
           Counter_IN : in std_logic;
           LED_OUT_DEV : inout std_logic;
           BTN_1 : in std_logic;
           BTN_2 : in std_logic;
           CONTROL1 : out std_logic;
           CONTROL2 : out std_logic;
           CONTROL3 : out std logic;
           CONTROL4 : out std_logic;
           CONTROL5 : out std_logic);
end main;
architecture Behavioral of main is
Signal RST : std_logic;
Signal clk_gen_counter : std_logic_vector(5 downto 1);
Signal strobe : std_logic; -- generator with laggest
frequency
Signal strobe2x : std_logic;
Signal strobe4x : std_logic;
Signal strobe8x : std_logic;
Signal strobe16x: std_logic;
Signal strobe32x: std_logic;
Signal clk9600 : std_logic;
Signal btn_lock : std_logic;
Signal LED_LOCK : std_logic;
Signal start_sig : std_logic := '0';
Signal start_sig1: std_logic := '0';
Signal start_seq : std_logic; -- min strobe and BTN_1
Signal s1,s2,s3 : std_logic;
Signal W1,W2 : std_logic;
```

```
Type LCDSymbolArray is array(0 to 39) of natural;
  Type LCDNumberArray is array(0 to 8) of natural;
  Signal DataString : LCDSymbolArray;
  Signal LCDNumber1 : LCDNumberArray;
  Signal LCDNumber2 : LCDNumberArray;
Signal show_req : std_logic;
Signal lcd_busy : std_logic;
begin
strobe2x <= clk_gen_counter(1);</pre>
strobe32x<= clk_gen_counter(5);</pre>
strobe <= clk50m;</pre>
start_seq <= strobe2x and BTN_1 and btn_lock; -- and
(not(LED_LOCK));
  CONTROL1 <= start_sig;
  CONTROL2 <= W1; --LED_OUT_DEV;</pre>
  CONTROL3 <= W2; --LED_LOCK;
-- CONTROL4 <= W1;
----- generators
Process (clk50m) is
BEGIN
 IF rising_edge(clk50m) THEN
    clk_gen_counter <= clk_gen_counter + 1;</pre>
 END IF;
END Process;
Process (clk50m)
 Variable CNT : natural;
 Variable tmp1 : std_logic;
BEGIN
  IF (Rising_edge(clk50m)) THEN
    IF CNT <= 2604 THEN CNT := CNT + 1; END IF;
    IF CNT > 2604 THEN tmp1 := not tmp1; CNT := 0; END IF;
  END IF;
  clk9600 <= tmp1;
END Process;
----- END generators -----
Process (clk50m)
Variable i : integer;
Variable CNT : integer;
Variable DONE : integer;
BEGIN
 If rising_edge(clk50m) THEN
  If DONE = 2 THEN RST <= '0';</pre>
  ELSE
```

```
CNT := CNT + 1;
    IF CNT = 2 THEN DONE := 2; END IF;
 RST <= '1';
  END IF;
 END IF;
END Process;
Process (strobe, start_sig, btn_lock)is -- Variable Lock :
boolean;
Variable cnt : integer;
  IF RST = '1' THEN btn_lock <= '1'; cnt := 0;</pre>
  ELSIF rising_edge(strobe) THEN
     IF start_sig = '1' THEN Lock := true; END IF;
     IF Lock THEN
        cnt := cnt + 1;
        IF cnt >=2 and cnt < 50000000 THEN btn_lock <=
'0'; END IF;
        IF cnt >= 50000000 THEN btn_lock <= '1'; Lock :=</pre>
false; cnt := 0; END IF;
     END IF;
  END IF;
END Process;
--Process (strobe, start_sig, btn_lock, lcd_busy)is
-- Variable Lock : boolean;
-- Variable cnt : integer;
-- Variable s1,s2: std_logic;
--BEGIN
-- IF RST = '1' THEN btn_lock <= '1'; cnt := 0; s1 := '0';
s2 :='0';
-- ELSIF rising_edge(strobe) THEN
       IF start_sig = '1' THEN Lock := true; END IF;
     IF Lock THEN
        s1 := s2; s2 := lcd_busy;
          IF s1 = '1' and s2 = '0' THEN btn_lock <= '1';
Lock := false; END IF;
   END IF;
-- END IF;
--END Process;
Process (strobe, start_seq)is
BEGIN
  IF RST = '1' THEN
     s1 <= '0';
     s2 <= '0';
     s3 <= '0';
     start_sig <= '0';
```

```
ELSIF rising_edge(strobe) THEN
     s1 <= s2;
     s2 <= s3;
     s3 <= start_seq;
    start_sig <= (not(s1))and(not(s2))and s3;
  END IF;
END Process;
Process (Counter_IN) is
  Variable WRK : boolean;
  Variable i : integer;
BEGIN
   IF start_sig1 = '1' THEN WRK := true; i := 0;
   ELSIF rising_edge(Counter_IN) and WRK THEN
        Case i is
             when 0 \Rightarrow W1 <= '0'; W2 <= '0'; i := i + 1;
             when 1 => W1 <= '1'; W2 <= '0'; i := i + 1;
             when 2 \Rightarrow W1 <= '0'; W2 <= '1'; i := i + 1;
             when others => W1 <= '0'; W2 <= '0'; WRK :=
false;
        END Case;
   END IF;
END Process;
----- LED signal -----1 mks 0.5mks-----
Process (start_sig, clk50m) is
  Variable CNT : integer;
  Variable WRK : boolean;
BEGIN
  IF start_sig = '1' THEN WRK := true; CNT := 0;
  ELSIF rising_edge(clk50m) and WRK THEN
     CNT := CNT + 1;
     IF CNT >= 1 and CNT < 50 THEN LED_OUT_DEV <= '1';</pre>
LED_LOCK <= '1'; END IF;
     IF CNT >= 50 and CNT < 75 THEN LED_OUT_DEV <= '0';</pre>
LED LOCK <= '1'; END IF;
     IF CNT >= 75 THEN LED_OUT_DEV <= '0'; LED_LOCK <= '0';</pre>
WRK := false; CNT := 0; END IF;
 END IF;
END Process;
Process (LED_OUT_DEV,clk50m) is --
  Variable s1,s2 : std_logic;
  Variable i : integer;
 Variable WRK : boolean;
BEGIN
    IF RST = '1' or start_sig = '1' THEN
       WRK := false;
       i := 0;
```

```
start_sig1 <= '0';
    ELSIF rising_edge(clk50m) THEN
            s1 := s2; s2 := LED_OUT_DEV;
            IF s1 = '1' and s2 = '0' THEN WRK := true; i :=
0; END IF;
            IF WRK THEN
                i := i + 1;
                IF i = 25 THEN start_sig1 <= '1'; END IF;</pre>
                IF i > 26 THEN start_sig1 <= '0'; i := 0;</pre>
WRK := false; END IF;
            END IF;
    END IF;
END Process;
  Process(strobe, start_sig, W1) is -- Counter1
  Variable Transfer : natural;
  Variable i
                     : natural;
  Variable LCDCNT : natural;
 BEGIN
   IF RST = '1' or start_sig = '1' THEN
        LCDCNT := 0;
        LCDNumber1(0) <= 0;</pre>
        LCDNumber1(1) <= 0;</pre>
        LCDNumber1(2) <= 0;</pre>
        LCDNumber1(3) <= 0;</pre>
        LCDNumber1(4) <= 0;</pre>
        LCDNumber1(5) <= 0;
        LCDNumber1(6) <= 0;
        LCDNumber1(7) <= 0;</pre>
        LCDNumber1(8) <= 0;
   ELSIF Rising_Edge(strobe) THEN
      IF (W1 = '1') THEN
        Transfer := 2;
        For i in 8 downto 0 LOOP
         IF LCDNumber1(i) < (10 - Transfer) THEN</pre>
LCDNumber1(i) <= LCDNumber1(i) + Transfer; Transfer := 0;</pre>
         ELSE LCDNumber1(i) <= LCDNumber1(i) + Transfer -</pre>
10; Transfer := 1;
         END IF;
--
        END LOOP;
        Case LCDNumber1(8) is
              when 0 \Rightarrow LCDNumber1(8) \iff 2; Transfer \iff 0;
              when 2 => LCDNumber1(8) <= 4; Transfer := 0;
```

```
when 4 \Rightarrow LCDNumber1(8) <= 6; Transfer := 0;
             when 6 => LCDNumber1(8) <= 8; Transfer := 0;
             when 8 \Rightarrow LCDNumber1(8) <= 0; Transfer := 1;
             when others=>LCDNumber1(8) <= 0; Transfer:=0;
        END Case;
        For i in 7 downto 0 LOOP
         IF LCDNumber1(i) < 9 and Transfer = 1 THEN
LCDNumber1(i) <= LCDNumber1(i) + Transfer; Transfer := 0;</pre>
         ELSIF Transfer = 1 THEN LCDNumber1(i) <= 0;</pre>
Transfer := 1;
         END IF;
        END LOOP;
      END IF;
           IF (LCDCNT > 99999999) THEN -- overflow
     LCDNumber1(0) <= 160-48;
     LCDNumber1(1) <= 79-48;
     LCDNumber1(2) <= 94-48;
     LCDNumber1(3) <= 69-48;
     LCDNumber1(4) <= 82-48;
     LCDNumber1(5) <= 70-48;
     LCDNumber1(6) <= 76-48;
     LCDNumber1(7) <= 79-48;
     LCDNumber1(8) <= 87-48;
           END IF;
   END IF;
 END Process;
Process(strobe, start_sig, W2) is -- Counter2
  Variable Transfer : natural;
                    : natural;
  Variable i
  Variable LCDCNT : natural;
 BEGIN
   IF RST = '1' or start_sig = '1' THEN
        LCDCNT := 0;
        LCDNumber2(0) <= 0;
        LCDNumber2(1) <= 0;
        LCDNumber2(2) <= 0;
        LCDNumber2(3) <= 0;
        LCDNumber2(4) <= 0;</pre>
        LCDNumber2(5) <= 0;
        LCDNumber2(6) <= 0;
        LCDNumber2(7) <= 0;</pre>
        LCDNumber2(8) <= 0;
```

```
ELSIF Rising_Edge(strobe) THEN
       IF (W2= '1') THEN
         Case LCDNumber2(8) is
              when 0 \Rightarrow LCDNumber2(8) \iff 2; Transfer \iff 0;
              when 2 \Rightarrow LCDNumber2(8) \ll 4; Transfer \approx 0;
              when 4 \Rightarrow LCDNumber2(8) <= 6; Transfer := 0;
              when 6 => LCDNumber2(8) <= 8; Transfer := 0;
              when 8 \Rightarrow LCDNumber2(8) <= 0; Transfer := 1;
              when others => LCDNumber2(8)<=0; Transfer := 0;
         END Case;
         For i in 7 downto 0 LOOP
          IF LCDNumber2(i) < 9 and Transfer = 1 THEN</pre>
LCDNumber2(i) <= LCDNumber2(i) + Transfer; Transfer := 0;</pre>
          ELSIF Transfer = 1 THEN LCDNumber2(i) <= 0;</pre>
Transfer := 1;
          END IF;
         END LOOP;
      END IF;
            IF (LCDCNT > 99999999) THEN -- overflow
     LCDNumber2(0) <= 160-48;
     LCDNumber2(1) <= 79-48;
     LCDNumber2(2) <= 94-48;
     LCDNumber2(3) <= 69-48;
     LCDNumber2(4) <= 82-48;
     LCDNumber2(5) \ll 70-48;
     LCDNumber2(6) <= 76-48;
     LCDNumber2(7) <= 79-48;
     LCDNumber2(8) <= 87-48;
            END IF;
   END IF;
 END Process;
Process(RST, start_sig)
BEGIN
  IF RST = '1' or start_sig = '1' THEN
              DataString(13) <= 48;</pre>
              DataString(14) <= 48;</pre>
              DataString(15) <= 48;</pre>
              DataString(16) <= 48;</pre>
              DataString(17) <= 48;</pre>
              DataString(18) <= 48;</pre>
              DataString(19) <= 48;</pre>
```

```
DataString(33) <= 48;</pre>
            DataString(34) <= 48;
            DataString(35) <= 48;
            DataString(36) <= 48;</pre>
            DataString(37) <= 48;</pre>
            DataString(38) <= 48;</pre>
            DataString(39) <= 48;</pre>
ELSIF W2'event and W2 = '0' THEN
            DataString(0) <= 254;</pre>
                                         -- CLS
                                         -- 1
                             <= 128;
            DataString(1)
            DataString(2)
                             <= 254i
            DataString(3)
                             <= 128;
                             <= 32;
            DataString(4)
            DataString(5)
                             <= 84i
                             <= 49;
                                       -- 1
            DataString(6)
            DataString(7)
                             <= 61;
                                       -- =
            DataString(8)
                             <= LCDNumber1(1)+48;
            DataString(9) <= LCDNumber1(2)+48;</pre>
            DataString(10) <= LCDNumber1(3)+48;</pre>
            DataString(11) <= LCDNumber1(4)+48;</pre>
            DataString(12) <= LCDNumber1(5)+48;</pre>
            DataString(13) <= LCDNumber1(6)+48;</pre>
            DataString(14) <= LCDNumber1(7)+48;</pre>
            DataString(15) <= LCDNumber1(8)+48;</pre>
            DataString(16) <= 48; --
            DataString(17) <= 110; -- n
            DataString(18) <= 115; -- s
            DataString(19) <= 32;</pre>
            DataString(20)
                             <= 254;
            DataString(21)
                              <= 192;
                              <= 254;
            DataString(22)
                             <= 192;
            DataString(23)
                              <= 32;
            DataString(24)
            DataString(25)
                              <= 84;
                              \leq 50;
            DataString(26)
            DataString(27)
                              <= 61;
            DataString(28)
                              <= LCDNumber2(1)+48;</pre>
            DataString(29)
                              <= LCDNumber2(2)+48;</pre>
                              <= LCDNumber2(3)+48;
            DataString(30)
            DataString(31)
                              \leq LCDNumber2(4)+48;
            DataString(32)
                              <= LCDNumber2(5)+48;
            DataString(33)
                              <= LCDNumber2(6)+48;</pre>
```

```
DataString(34) <= LCDNumber2(7)+48;</pre>
             DataString(35) <= LCDNumber2(8)+48;</pre>
             DataString(36) <= 48;</pre>
             DataString(37) <= 110;</pre>
             DataString(38) <= 115;</pre>
             DataString(39) <= 32;</pre>
  END IF;
END Process;
Process (clk9600,show_req,DataString,RST) -
 Variable CNT : natural;
 Variable ITR : natural :=0;
 Variable DataVector : std_logic_vector(9 downto 0);
 Variable tmp1
                  : std_logic;
BEGIN
  IF show_req = '1' or RST = '1' THEN lcd_busy <= '1';</pre>
  ELSIF (Rising_edge(clk9600)) THEN
    IF (lcd_busy = '1') and (ITR<=39) THEN
     DataVector := B"1" & CONV_STD_LOGIC_VECTOR
(DataString(ITR), 8) & B"0";
IF CNT < 9 THEN tmp1 := DataVector(CNT); CNT := CNT + 1;</pre>
           ELSIF (CNT >=9) and (CNT<12) THEN tmp1 := '1';
CNT := CNT + 1;
           ELSIF (CNT >=12) and (CNT<18) THEN LCD_SEQ_DEV
<= '1'; tmp1 := '1'; CNT := CNT + 1;
           ELSIF (CNT >=18)
                                          THEN LCD_SEQ_DEV
<= '1'; tmp1 := '1'; CNT := 0; ITR := ITR + 1;
           END IF;
     LCD_SEQ_DEV <= tmp1; --
    ELSE
      ITR := 0;
      lcd busy <= '0';</pre>
    END IF;
  END IF;
END Process;
```

143

## Program CC09. Implementation of "interval+counter" concept, generator simulator for dacay fluorescent process.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--library UNISIM;
--use UNISIM.VComponents.all;
-- Experiment 1. Implementation of the formula
"interval+counter"
       1<->2<->3
-- Starts by external pulse coming from the microprocessor.
Measures the interval from first phonon to the second, and
the number of photons in the fixed interval. Transmits the
results back to microprocessor by RS232
    Implemented a generator-simulator of the decay process.
entity main is
    Port ( clk50m : in std_logic;
           BS_StartSig : in std_logic;
           LCD_SEQ_DEV : out std_logic;
           Counter_IN_DEV : inout std_logic; -- inverted
           LED_OUT_DEV : inout std_logic;
               BS_SEQ_DEV : inout std_logic;
           BTN_1 : in std_logic;
           BTN_2 : in std_logic;
                 TESTGEN : out std_logic;
           CONTROL1 : out std_logic;
           CONTROL2 : out std logic;
           CONTROL3 : out std_logic;
           CONTROL4 : out std_logic;
           CONTROL5 : out std_logic;
                 CONTROL6 : out std logic;
                 CONTROL7 : out std_logic;
                 CONTROL8 : out std_logic;
                 CONTROL9 : out std_logic;
                 CONTROL10 : out std_logic
                 );
end main;
architecture Behavioral of main is
```

```
Signal RST : std_logic;
Signal clk_gen_counter : std_logic_vector(5 downto 1);
Signal strobe : std_logic; -- generator with largest
frequency
Signal strobe2x : std_logic;
Signal strobe4x : std_logic;
Signal strobe8x : std logic;
Signal strobe16x: std_logic;
Signal strobe32x: std_logic;
Signal clk9600 : std_logic;
Signal clk19200 : std_logic;
Signal mpTRRate : std_logic;
Signal btn_lock : std_logic;
Signal LED_LOCK : std_logic;
Signal start_sig : std_logic := '0';
Signal start_sig1: std_logic := '0';
Signal start_seq : std_logic; -- min strobe and BTN_1
Signal s1,s2,s3 : std_logic;
Signal W1,W2,nW2, WF, nWF : std_logic;
  Type LCDSymbolArray is array(0 to 39) of natural;
  Type BSSymbolArray is array(0 to 27) of natural;
  Type LCDNumberArray is array(0 to 8) of natural;
  Signal DataString : LCDSymbolArray;
  Signal DataStringBS : BSSymbolArray;
  Signal LCDNumber1 : LCDNumberArray;
  Signal LCDNumber2 : LCDNumberArray;
  Signal LCDNumber3 : LCDNumberArray;
Signal show_req : std_logic;
Signal lcd_busy : std_logic;
Signal BS_busy : std_logic;
Signal Counter_IN : std_logic;
Signal Probe1 : std_logic;
begin
mpTRRate <= clk19200;</pre>
nW2 \ll not W2;
nWF <= not WF;
```

```
Counter_IN <= Counter_IN_DEV;</pre>
start_sig <= not BS_StartSig; -- BS</pre>
strobe2x <= clk gen counter(1);</pre>
strobe32x<= clk_gen_counter(5);</pre>
--TESTGEN <= clk_gen_counter(5);
--counter IN <= strobe32x;
strobe <= clk50m;
start_seq <= strobe2x and BTN_1 and btn_lock; -- and
(not(LED_LOCK));
  CONTROL1 <= start_sig;
  CONTROL2 <= W1;
  CONTROL3 <= W2;
  CONTROL4 <= WF;
  CONTROL5 <= Counter_IN;
  CONTROL6 <= LED_OUT_DEV;</pre>
  CONTROL7 <= show_req;
  CONTROL8 <= start_sig1;</pre>
  CONTROL9 <= Probe1;
  CONTROL10 <= LED_OUT_DEV;</pre>
---- generators
Process (clk50m) is
BEGIN
 IF rising edge(clk50m) THEN
    clk_gen_counter <= clk_gen_counter + 1;</pre>
 END IF;
END Process;
Process (clk50m)
 Variable CNT : natural;
 Variable tmp1 : std_logic;
BEGIN
  IF (Rising edge(clk50m)) THEN
    IF CNT <= 2604 THEN CNT := CNT + 1; END IF;
    IF CNT > 2604 THEN tmp1 := not tmp1; CNT := 0; END IF;
  END IF;
  clk9600 <= tmp1;
END Process;
                   -- generation clk19200 for RS232
Process (clk50m)
 Variable CNT : natural;
 Variable tmp1 : std_logic;
BEGIN
  IF (Rising_edge(clk50m)) THEN
    IF CNT <= 1302 THEN CNT := CNT + 1; END IF;
    IF CNT > 1302 THEN tmp1 := not tmp1; CNT := 0; END IF;
```

```
END IF;
  clk19200 <= tmp1;
END Process;
Process (clk50m) -- decay modeling generator
  Variable CNT, i, iter : integer;
  Variable WRK : boolean;
BEGIN
  IF start_sig1 = '1' THEN WRK := true; CNT := 0; i := 2;
iter := 0; TESTGEN <= '0';
  ELSIF rising_edge(clk50m) and WRK THEN
    CNT := CNT + 1;
IF CNT < i
                               THEN TESTGEN <= '0'; END IF;
IF CNT >= i
                               THEN TESTGEN <= '1'; END IF;
            and CNT < i+1
IF CNT >= i+1 and CNT < i*2 THEN TESTGEN <= '0'; END IF;
IF CNT = i*2
                             THEN TESTGEN <= '1'; i := i*2;
iter := iter + 1; END IF;
IF iter >= 100 THEN TESTGEN <= '0'; WRK := false; CNT :=
0; END IF;
  END IF;
END Process;
----- END generators -----
Process (clk50m)
Variable i : integer;
Variable CNT : integer;
Variable DONE : integer;
BEGIN
 If rising_edge(clk50m) THEN
  If DONE = 2 THEN RST <= '0';</pre>
  ELSE
    CNT := CNT + 1;
    IF CNT = 2 THEN DONE := 2; END IF;
  RST <= '1';
  END IF;
 END IF;
END Process;
Process (strobe, start_sig, btn_lock)is
 Variable Lock : boolean;
Variable cnt : integer;
BEGIN
  IF RST = '1' THEN btn_lock <= '1'; cnt := 0;</pre>
  ELSIF rising_edge(strobe) THEN
     IF start_sig = '1' THEN Lock := true; END IF;
     IF Lock THEN
```

```
cnt := cnt + 1;
        IF cnt >=2 and cnt < 50000000 THEN btn_lock <=
'0'; END IF;
        IF cnt >= 50000000 THEN btn_lock <= '1'; Lock :=</pre>
false; cnt := 0; END IF;
     END IF;
  END IF;
END Process;
Process (Counter_IN) is
  Variable WRK : boolean;
  Variable i : integer;
BEGIN
   IF start_sig1 = '1' THEN WRK := true; Probe1 <= '1'; i</pre>
:= 1;
   ELSIF rising_edge(Counter_IN) THEN
       IF WRK = true THEN
             Case i is
         when 0 \Rightarrow W1 \Leftarrow '0'; W2 \Leftarrow '0'; i \Rightarrow i + 1;
             when 1 => W1 <= '1'; W2 <= '0'; i := i + 1;
             when 2 => W1 <= '0'; W2 <= '1'; i := i + 1;
             when others => W1 <= '0'; W2 <= '0'; WRK :=
false; Probe1 <= '0';</pre>
          END Case;
           ELSE
              W1 <= '0'; W2 <= '0';
       END IF;
   END IF;
END Process;
- LED signal and fixed window WF -----1 mks 0.5mks-----
Process (start_sig1, clk50m) is
  Variable CNT : integer;
  Variable WRK : boolean;
BEGIN
  IF start sig1 = '1' THEN WRK := true; CNT := 0;
  ELSIF rising_edge(clk50m) and WRK THEN
     CNT := CNT + 1;
     IF CNT >= 1 and CNT < 50 THEN LED_OUT_DEV <= '1';</pre>
LED_LOCK <= '1'; WF<= '0'; END IF;
     IF CNT >= 50 and CNT < 75 THEN LED_OUT_DEV <= '0';</pre>
LED LOCK <= '1'; END IF;
     IF CNT = 75 THEN LED OUT DEV <= '0'; LED LOCK <= '0';
WF<='1'; END IF;
     IF CNT >= 2000 THEN WF <= '0'; WRK := false; CNT :=
0; END IF;
  END IF;
END Process;
```

```
Process (LED_OUT_DEV,clk50m) is --
  Variable s1,s2 : std_logic;
  Variable i : integer;
  Variable WRK : boolean;
BEGIN
    IF RST = '1' or start_sig = '1' THEN
       WRK := false;
       i := 0;
       start_sig1 <= '0';
    ELSIF rising_edge(clk50m) THEN
           s1 := s2; s2 := LED_OUT_DEV;
           IF s1 = '1' and s2 = '0' THEN WRK := true; i :=
0; END IF;
           IF WRK THEN
               i := i + 1;
               IF i = 25 THEN start_sig1 <= '1'; END IF;</pre>
               IF i > 26 THEN start_sig1 <= '0'; i := 0;</pre>
WRK := false; END IF;
           END IF;
    END IF;
END Process;
---- Counters -----
  Process(strobe, start_sig, W1) is -- Counter1
  Variable Transfer : natural;
  Variable i
                     : natural;
  Variable LCDCNT : natural;
 BEGIN
   IF RST = '1' or start_sig = '1' THEN
             LCDCNT := 0;
        LCDNumber1(0) <= 0;</pre>
        LCDNumber1(1) <= 0;
        LCDNumber1(2) <= 0;</pre>
        LCDNumber1(3) <= 0;</pre>
        LCDNumber1(4) <= 0;
        LCDNumber1(5) <= 0;
        LCDNumber1(6) <= 0;
        LCDNumber1(7) <= 0;
        LCDNumber1(8) <= 0;</pre>
   ELSIF Rising_Edge(strobe) THEN
      IF (W1 = '1') THEN
        Transfer := 2;
        For i in 8 downto 0 LOOP
         IF LCDNumber1(i) < (10 - Transfer) THEN</pre>
LCDNumber1(i) <= LCDNumber1(i) + Transfer; Transfer := 0;</pre>
```

```
ELSE LCDNumber1(i) <= LCDNumber1(i) + Transfer -</pre>
10; Transfer := 1;
         END IF;
        END LOOP;
        Case LCDNumber1(8) is
              when 0 \Rightarrow LCDNumber1(8) \iff 2; Transfer \iff 0;
              when 2 \Rightarrow LCDNumber1(8) <= 4; Transfer := 0;
              when 4 \Rightarrow LCDNumber1(8) <= 6; Transfer := 0;
             when 6 => LCDNumber1(8) <= 8; Transfer := 0;
             when 8 \Rightarrow LCDNumber1(8) <= 0; Transfer := 1;
              when others => LCDNumber1(8)<= 0; Transfer:= 0;
        END Case;
        For i in 7 downto 0 LOOP
         IF LCDNumber1(i) < 9 and Transfer = 1 THEN
LCDNumber1(i) <= LCDNumber1(i) + Transfer; Transfer := 0;</pre>
         ELSIF Transfer = 1 THEN LCDNumber1(i) <= 0;</pre>
Transfer := 1;
         END IF;
        END LOOP;
      END IF;
           IF (LCDCNT > 99999999) THEN -- overflow
     LCDNumber1(0) <= 160-48;
     LCDNumber1(1) <= 79-48;
     LCDNumber1(2) <= 94-48;
     LCDNumber1(3) <= 69-48;
     LCDNumber1(4) \le 82-48;
     LCDNumber1(5) <= 70-48;
     LCDNumber1(6) <= 76-48;
     LCDNumber1(7) <= 79-48;
     LCDNumber1(8) <= 87-48;
           END IF;
   END IF;
 END Process;
-- Second window
Process(strobe, start_sig, W2) is -- Counter2
  Variable Transfer : natural;
  Variable i
                     : natural;
  Variable LCDCNT : natural;
 BEGIN
   IF RST = '1' or start_sig = '1' THEN
        LCDCNT := 0;
```

```
LCDNumber2(0) <= 0;
        LCDNumber2(1) <= 0;</pre>
        LCDNumber2(2) <= 0;
        LCDNumber2(3) <= 0;
        LCDNumber2(4) <= 0;
        LCDNumber2(5) <= 0;
        LCDNumber2(6) <= 0;
        LCDNumber2(7) <= 0;
        LCDNumber2(8) <= 0;
   ELSIF Rising_Edge(strobe) THEN
      IF (W2= '1') THEN
        Case LCDNumber2(8) is
              when 0 \Rightarrow LCDNumber2(8) \iff 2; Transfer \implies 0;
              when 2 \Rightarrow LCDNumber2(8) \ll 4; Transfer \approx 0;
              when 4 \Rightarrow LCDNumber2(8) \iff 6; Transfer \implies 0;
              when 6 => LCDNumber2(8) <= 8; Transfer := 0;
              when 8 \Rightarrow LCDNumber2(8) <= 0; Transfer := 1;
              when others=>LCDNumber2(8)<= 0; Transfer := 0;
        END Case;
        For i in 7 downto 0 LOOP
         IF LCDNumber2(i) < 9 and Transfer = 1 THEN</pre>
LCDNumber2(i) <= LCDNumber2(i) + Transfer; Transfer := 0;</pre>
         ELSIF Transfer = 1 THEN LCDNumber2(i) <= 0;</pre>
Transfer := 1;
         END IF;
        END LOOP;
      END IF;
            IF (LCDCNT > 99999999) THEN -- overflow
     LCDNumber2(0) <= 160-48;
     LCDNumber2(1) <= 79-48;
     LCDNumber2(2) <= 94-48;
     LCDNumber2(3) <= 69-48;
     LCDNumber2(4) <= 82-48;
     LCDNumber2(5) <= 70-48;
     LCDNumber2(6) <= 76-48;
     LCDNumber2(7) <= 79-48;
     LCDNumber2(8) <= 87-48;
            END IF;
   END IF;
 END Process;
-- Fixed window
Process(Counter_IN, start_sig, WF) is -- Counter3
```

```
Variable Transfer : natural;
  Variable i : natural;
  Variable LCDCNT : natural;
 BEGIN
   IF RST = '1' or start_sig = '1' THEN
             LCDCNT := 0;
        LCDNumber3(0) <= 0;
        LCDNumber3(1) <= 0;
        LCDNumber3(2) <= 0;
        LCDNumber3(3) <= 0;
        LCDNumber3(4) <= 0;
        LCDNumber3(5) <= 0;
        LCDNumber3(6) <= 0;
        LCDNumber3(7) <= 0;</pre>
        LCDNumber3(8) <= 0;
   ELSIF Rising_Edge(Counter_IN) THEN
      IF (WF= '1') THEN
          Transfer := 1;
        For i in 8 downto 0 LOOP
         IF LCDNumber3(i) < 9 and Transfer = 1 THEN</pre>
LCDNumber3(i) <= LCDNumber3(i) + Transfer; Transfer := 0;</pre>
         ELSIF Transfer = 1 THEN LCDNumber3(i) <= 0;</pre>
Transfer := 1;
         END IF;
        END LOOP;
      END IF;
           IF (LCDCNT > 99999999) THEN -- overflow
     LCDNumber3(0) <= 160-48;
     LCDNumber3(1) <= 79-48;
     LCDNumber3(2) <= 94-48;
     LCDNumber3(3) <= 69-48;
     LCDNumber3(4) <= 82-48;
     LCDNumber3(5) <= 70-48;
     LCDNumber3(6) <= 76-48;
     LCDNumber3(7) <= 79-48;
     LCDNumber3(8) <= 87-48;
           END IF;
   END IF;
 END Process;
-- transmission to BS - starts on falling edge of WF
Process(RST, start_sig,nWF)
BEGIN
```

```
IF RST = '1' or start_sig = '1' THEN
             DataStringBS(0) <= 0;</pre>
             DataStringBS(1) <= 0;</pre>
             DataStringBS(2) <= 0;</pre>
             DataStringBS(3) <= 0;</pre>
             DataStringBS(4) <= 0;</pre>
             DataStringBS(5) <= 0;</pre>
             DataStringBS(6) <= 0;</pre>
             DataStringBS(7) <= 0;</pre>
             DataStringBS(8) <= 0;</pre>
             DataStringBS(9) <= 0;</pre>
             DataStringBS(10) <= 0;</pre>
             DataStringBS(11) <= 0;</pre>
             DataStringBS(12) <= 0;</pre>
             DataStringBS(13) <= 0;</pre>
             DataStringBS(14) <= 0;</pre>
             DataStringBS(15) <= 0;</pre>
             DataStringBS(16) <= 0;</pre>
             DataStringBS(17) <= 0;</pre>
             DataStringBS(18) <= 0;</pre>
             DataStringBS(19) <= 0;</pre>
             DataStringBS(20) <= 0;</pre>
             DataStringBS(21) <= 0;</pre>
             DataStringBS(22) <= 0;</pre>
             DataStringBS(23) <= 0;</pre>
             DataStringBS(24) <= 0;</pre>
             DataStringBS(25) <= 0;</pre>
             DataStringBS(26) <= 0;</pre>
ELSIF rising_edge(nWF) THEN
             DataStringBS(0) <= LCDNumber1(1);</pre>
             DataStringBS(1) <= LCDNumber1(2);</pre>
             DataStringBS(2) <= LCDNumber1(3);</pre>
             DataStringBS(3) <= LCDNumber1(4);</pre>
             DataStringBS(4) <= LCDNumber1(5);</pre>
             DataStringBS(5) <= LCDNumber1(6);</pre>
             DataStringBS(6) <= LCDNumber1(7);</pre>
             DataStringBS(7) <= LCDNumber1(8);</pre>
             DataStringBS(8) <= 0;</pre>
             DataStringBS(9) <= LCDNumber2(1);</pre>
             DataStringBS(10) <= LCDNumber2(2);</pre>
             DataStringBS(11) <= LCDNumber2(3);</pre>
             DataStringBS(12) <= LCDNumber2(4);</pre>
             DataStringBS(13) <= LCDNumber2(5);</pre>
             DataStringBS(14) <= LCDNumber2(6);</pre>
```

```
DataStringBS(15) <= LCDNumber2(7);</pre>
               DataStringBS(16) <= LCDNumber2(8);</pre>
               DataStringBS(17) <= 0;</pre>
               DataStringBS(18) <= LCDNumber3(0);</pre>
               DataStringBS(19) <= LCDNumber3(1);</pre>
               DataStringBS(20) <= LCDNumber3(2);</pre>
               DataStringBS(21) <= LCDNumber3(3);</pre>
               DataStringBS(22) <= LCDNumber3(4);</pre>
               DataStringBS(23) <= LCDNumber3(5);</pre>
               DataStringBS(24) <= LCDNumber3(6);</pre>
               DataStringBS(25) <= LCDNumber3(7);</pre>
               DataStringBS(26) <= LCDNumber3(8);</pre>
  END IF;
END Process;
Process(RST, start_sig,nWF)
BEGIN
  IF RST = '1' THEN
               DataString(0)
                                 <= 254;
                                               -- CLS
               DataString(1) <= 128;</pre>
                                               -- 1
               DataString(2) <= 254;</pre>
                                 <= 128;
               DataString(3)
                                  <= 32;
               DataString(4)
               DataString(5) <= 84;</pre>
                                            -- T
               DataString(6) <= 49;</pre>
                                            -- 1
               DataString(7) <= 61;</pre>
                                             -- =
               DataString(8) <= 48;</pre>
               DataString(9)
                                 <= 48;
               DataString(10) <= 48;</pre>
               DataString(11) <= 48;</pre>
               DataString(12) <= 48;</pre>
               DataString(13) <= 48;</pre>
               DataString(14) <= 48;</pre>
               DataString(15) <= 48;</pre>
               DataString(16) <= 48;</pre>
               DataString(17) <= 110;</pre>
               DataString(18) <= 115;</pre>
               DataString(19) <= 32;</pre>
               DataString(28) <= 48;</pre>
               DataString(29) <= 48;</pre>
               DataString(30) <= 48;</pre>
               DataString(31) <= 48;</pre>
               DataString(32) <= 48;</pre>
               DataString(33) <= 48;</pre>
```

```
DataString(34) <= 48;</pre>
            DataString(35) <= 48;</pre>
            DataString(36) <= 48;
            DataString(37) <= 110;</pre>
            DataString(38) <= 115;</pre>
            DataString(39) <= 32;</pre>
ELSIF rising_edge(nWF) THEN
                                         -- CLS
            DataString(0)
                             <= 254;
            DataString(1)
                             <= 128;
                                         -- 1
            DataString(2)
                             <= 254;
            DataString(3)
                             <= 128;
            DataString(4)
                             <= 32;
            DataString(5)
                             <= 84;
                                       -- T
                                       -- 1
            DataString(6) <= 49;</pre>
            DataString(7) <= 61;</pre>
            DataString(8)
                             <= LCDNumber1(1)+48;</pre>
            DataString(9)
                             <= LCDNumber1(2)+48;
            DataString(10) <= LCDNumber1(3)+48;</pre>
            DataString(11) <= LCDNumber1(4)+48;</pre>
            DataString(12) <= LCDNumber1(5)+48;</pre>
            DataString(13) <= LCDNumber1(6)+48;</pre>
            DataString(14) <= LCDNumber1(7)+48;</pre>
            DataString(15) <= LCDNumber1(8)+48;</pre>
            DataString(16) <= 48; -- space
            DataString(17) <= 110; -- n
            DataString(18) <= 115; -- s
            DataString(19) <= 32;</pre>
            DataString(20)
                             <= 254;
                              <= 192;
            DataString(21)
            DataString(22)
                             <= 254;
            DataString(23)
                             <= 192;
                              <= 32;
            DataString(24)
                              <= 78; -- N
            DataString(25)
                              <= 32;
            DataString(26)
            DataString(27)
                             <= 61;
            DataString(28)
                              <= LCDNumber3(1)+48;</pre>
            DataString(29)
                              <= LCDNumber3(2)+48;</pre>
            DataString(30)
                              <= LCDNumber3(3)+48;</pre>
            DataString(31)
                              <= LCDNumber3(4)+48;</pre>
                              <= LCDNumber3(5)+48;
            DataString(32)
            DataString(33)
                              <= LCDNumber3(6)+48;</pre>
            DataString(34)
                              <= LCDNumber3(7)+48;</pre>
            DataString(35)
                              <= LCDNumber3(8)+48;</pre>
```

```
DataString(36) <= 32;</pre>
             DataString(37) <= 32;</pre>
             DataString(38) <= 32;</pre>
             DataString(39) <= 32;</pre>
  END IF;
END Process;
Process (W2, WF, strobe) is --
  Variable s1,s2 : std_logic;
  Variable i
                 : integer;
  Variable WRK : boolean;
BEGIN
    IF RST = '1' or start_sig = '1' THEN
       WRK := false;
           s1 := '0';
           s2 := '0';
       i := 0;
       show_req <= '0';
    ELSIF rising_edge(strobe) THEN
           s1 := s2; s2 := WF; -- s2 := W2;
      IF s1 = '1' and s2 = '0' THEN WRK := true; i := 0;
END IF;
           IF WRK THEN
                i := i + 1;
                IF i = 100 THEN show_req <= '1'; END IF;</pre>
                IF i > 102 THEN show_req <= '0'; i := 0;</pre>
WRK:=false; END IF;
           END IF;
    END IF;
END Process;
Process (clk9600, show_req, DataString, RST)
 Variable CNT : natural;
 Variable ITR : natural :=0;
 Variable DataVector : std_logic_vector(9 downto 0);
                      : std_logic;
 Variable tmp1
BEGIN
  IF show_req = '1' or RST = '1' THEN lcd_busy <= '1';</pre>
  ELSIF (Rising_edge(clk9600)) THEN
    IF (lcd_busy = '1') and (ITR<=39) THEN
     DataVector := B"1" & CONV_STD_LOGIC_VECTOR
(DataString(ITR), 8) & B"0";
           IF CNT < 9 THEN tmp1 := DataVector(CNT); CNT :=</pre>
CNT + 1;
           ELSIF (CNT >=9) and (CNT<12) THEN tmp1 := '1';
CNT := CNT + 1;
           ELSIF (CNT >=12) and (CNT<18) THEN LCD_SEQ_DEV
<= '1'; tmp1 := '1'; CNT := CNT + 1;
```

```
ELSIF (CNT >=18)
                                          THEN LCD_SEQ_DEV
<= '1'; tmp1 := '1'; CNT := 0; ITR := ITR + 1;
           END IF;
     LCD_SEQ_DEV <= tmp1; --</pre>
    ELSE
      ITR := 0;
      lcd_busy <= '0';</pre>
    END IF;
  END IF;
END Process;
Process (mpTRRate, show_req, DataStringBS, RST) --
Variable CNT : natural;
Variable ITR : natural :=0;
Variable DataVector : std_logic_vector(9 downto 0);
Variable tmp1
                     : std_logic;
BEGIN
  IF show_req = '1' or RST = '1' THEN BS_busy <= '1';</pre>
  ELSIF (Rising_edge(mpTRRate)) THEN
    IF (BS_busy = '1') and (ITR<=27) THEN
     DataVector := B"0" & not CONV_STD_LOGIC_VECTOR
(DataStringBS(ITR), 8) & B"1";
           IF CNT < 10 THEN tmp1 := DataVector(CNT); CNT :=</pre>
CNT + 1;
           ELSIF (CNT >=10) and (CNT<20)THEN tmp1 := '0';
CNT := CNT + 1;
           ELSIF (CNT >=20)
                                          THEN tmp1 := '0';
CNT := 0; ITR := ITR + 1;
           END IF;
     BS_SEQ_DEV <= not tmp1; --
    ELSE
      ITR := 0;
      BS_busy <= '0';
    END IF;
  END IF;
END Process;
end Behavioral;
```

## Program PCC03. Portable single photon counter.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--library UNISIM;
--use UNISIM.VComponents.all;
-- Portable single photon counter
-- Vladis Kuzminsky
entity main is
    Port ( clk50m : in std_logic;
          counterIN : in std_logic;
          cntReqEXT : in std_logic;
          btnMODE : in std_logic;
          btnSequence : in std logic;
          btnStart : in std_logic;
          LCD_SEQ_DEV : out std_logic;
          cntREQ : out std_logic;
          test1
                     : out std_logic;
          test2
                     : out std_logic);
end main;
architecture Behavioral of main is
  Signal signal_dff : std_logic;
  Signal signal_dff1 : std_logic;
  Signal signal_dff2 : std_logic;
  Signal signal_dff3 : std_logic;
  Signal b_mode : std_logic;
  Signal b_regime : std_logic;
  Signal b_Sequence : std_logic;
  Type LCDSymbolArray is array(0 to 39) of natural;
  Type LCDNumberArray is array(0 to 8) of natural;
  Signal DataString : LCDSymbolArray;
  Signal LCDNumber : LCDNumberArray;
  Signal LCDNumberW : LCDNumberArray;
  Signal clk1m : std_logic;
  Signal clk100hz: std_logic;
  Signal clk50hz : std_logic;
  Signal clk5hz : std_logic;
```

```
Signal clk0_5hz: std_logic;
  Signal clk9600 : std_logic;
  Signal High : std_logic;
Signal Low : std_logic;
                : bit := '1';
  Signal RST
  Signal Go Go : std logic;
  Signal Go_Go_EXT : std_logic;
  Signal cnt_done : std_logic;
  Signal cnt_req : std_logic;
  Signal DataShowReq : boolean := False;
  Signal DataOutDone : boolean := True;
  Signal Busy : boolean := True;
Signal cnt_rst : std_logic;
  Signal DataReady : boolean := False;
  Signal Regime : natural;
  Signal SequenceMODE: natural;
  Signal CntReqExtINV: std_logic;
  Signal EXT_WRK_Signal
                          : std_logic;
  Signal EXT_WRK_Signal_Operator : std_logic;
  Signal EXT_WIN : std_logic;
  Signal EXT_WIN_INV : std_logic;
  Signal test_sig: std_logic;
  Signal Step1,Step2 : std_logic;
begin
 High <= '1';
 Low <= '0';
 CntReqExtINV <= not CntReqExt;</pre>
 EXT_WIN_INV <= not EXT_WIN;</pre>
 cntREQ <= cnt_req;</pre>
 test1 <= EXT_WIN;</pre>
 test2 <= Go_Go_EXT;</pre>
-- 1 MHz ------
Process(clk50m,clk1m)
Variable CNT : natural := 0;
BEGIN
  IF Rising_Edge(clk50m) THEN
    CNT := CNT + 1;
IF (CNT > 0) and (CNT <= 25) THEN clk1m <= '0'; END IF;
```

```
IF (CNT > 25) and (CNT < 50) THEN clk1m <= '1'; END IF;
                         THEN clk1m <= '1'; CNT := 0;
IF CNT >= 50
END IF;
 END IF;
END Process;
-- 100 Hz ------
Process(clk1m)
Variable CNT : natural := 0;
BEGIN
 IF Rising_Edge(clk1m) THEN
   CNT := CNT + 1;
IF (CNT > 0) and (CNT <= 5000) THEN clk100hz <= '0';
END IF;
IF (CNT > 5000) and (CNT < 10000) THEN clk100hz <= '1';
END IF;
IF CNT >= 10000
                              THEN clk100hz <= '1';
CNT := 0; END IF;
 END IF;
END Process;
-- 50 Hz -----
Process(clk100hz)
Variable CNT : natural := 0;
BEGIN
 IF Rising_Edge(clk100hz) THEN
   CNT := CNT + 1;
IF (CNT > 0) and (CNT <= 1) THEN clk50hz <= '0'; END IF;
IF (CNT > 1) and (CNT < 2) THEN clk50hz <= '1'; END IF;
                        THEN clk50hz <= '1'; CNT := 0;
IF CNT >= 2
END IF;
 END IF;
END Process;
-- 5 Hz ------
Process(clk100hz)
Variable CNT : natural := 0;
BEGIN
 IF Rising Edge(clk100hz) THEN
   CNT := CNT + 1;
    IF (CNT > 0) and (CNT <= 10) THEN clk5hz <= '0';
END IF;
    IF (CNT > 10) and (CNT < 20)
                                THEN clk5hz <= '1';
END IF;
    IF CNT >= 20
                                THEN clk5hz <= '1';
CNT := 0; END IF;
 END IF;
END Process;
-- 0.5 Hz -----
Process(clk5hz)
Variable CNT : natural := 0;
```

```
BEGIN
  IF Rising_Edge(clk5hz) THEN
   CNT := CNT + 1;
                             THEN clk0_5hz <= '0';
 IF (CNT > 0) and (CNT <= 10)
END IF;
                                 THEN clk0_5hz <= '1';
IF (CNT > 10) and (CNT < 20)
END IF;
IF CNT >= 20
                                  THEN clk0 5hz <= '1';
CNT := 0; END IF;
 END IF;
END Process;
--RESET------
Process (clk50m)
Variable i : integer;
Variable CNT : integer;
Variable DONE : integer;
BEGIN
 If rising_edge(clk50m) THEN
  If DONE = 10 THEN RST <= '0';</pre>
 ELSE
   CNT := CNT + 1;
    IF CNT = 10 THEN DONE := 10; END IF;
 RST <= '1';
 END IF;
 END IF;
END Process;
   Process (clk50m) --
   Variable CNT : natural;
   Variable tmp1 : std_logic;
   BEGIN
      IF (Rising_edge(clk50m)) THEN
        IF CNT <= 2604 THEN CNT := CNT + 1; END IF;
        IF CNT > 2604 THEN tmp1 := not tmp1; CNT := 0;
END IF;
     END IF;
    clk9600 <= tmp1;
   END Process;
 Process (clk9600, DataReady, DataShowReq, DataString, RST) --
   Variable CNT : natural;
   Variable ITR : natural :=0;
   Variable DataVector : std_logic_vector(9 downto 0);
   Variable tmp1
                      : std_logic;
 BEGIN
```

```
IF (Rising_edge(clk9600)) THEN
    IF ((DataShowReq and DataReady) or(DataOutDone =
False)) and (ITR<=39) THEN
     DataOutDone <= False;</pre>
     DataVector := B"1" & CONV_STD_LOGIC_VECTOR
(DataString(ITR), 8) & B"0";
            IF CNT < 9 THEN tmp1 := DataVector(CNT); CNT :=</pre>
CNT + 1;
            ELSIF (CNT >=9) and (CNT<12) THEN tmp1 := '1';
CNT := CNT + 1;
            ELSIF (CNT >=12) and (CNT<18) THEN LCD_SEQ_DEV
<= High; tmp1 := '1'; CNT := CNT + 1;
            ELSIF (CNT >=18)
                                            THEN LCD_SEQ_DEV
<= High; tmp1 := '1'; CNT := 0; ITR := ITR + 1;
            END IF;
     LCD_SEQ_DEV <= tmp1; -- вывод на LCD
    ELSE
      ITR := 0;
      DataOutDone <= True;</pre>
    END IF;
  END IF;
 END Process;
_____
Process(cnt_done,RST, Regime,Busy)
BEGIN
  IF RST = '1' THEN
              DataString(13) <= 48;</pre>
              DataString(14) <= 48;</pre>
              DataString(15) <= 48;</pre>
              DataString(16) <= 48;</pre>
              DataString(17) <= 48;</pre>
              DataString(18) <= 48;</pre>
              DataString(19) <= 48;</pre>
              DataString(33) <= 48;</pre>
              DataString(34) <= 48;</pre>
              DataString(35) <= 48;</pre>
              DataString(36) <= 48;</pre>
              DataString(37) <= 48;</pre>
              DataString(38) <= 48;
              DataString(39) <= 48;</pre>
  ELSIF Rising_edge(cnt_done) THEN
              DataString(0) <= 254;</pre>
                                           -- CLS
              DataString(1) <= 128;</pre>
                                           -- 1
              DataString(2) <= 254;</pre>
              DataString(3) <= 128;</pre>
```

```
IF (Busy = True)THEN
        DataString(4)
                         <= 87;
                          <= 82;
        DataString(5)
                          <= 75;
         DataString(6)
END IF;
IF not Busy THEN
         DataString(4)
                          <= 32;
                          <= 32;
         DataString(5)
                          <= 32;
        DataString(6)
END IF;
         DataString(7)
                         <= 32;
         DataString(8)
                         <= 32;
         DataString(9) <= 77;</pre>
        DataString(10) <= 61;</pre>
      IF Regime = 0 THEN
         DataString(11) <= LCDNumberW(1)+48;</pre>
         DataString(12) <= LCDNumberW(2)+48;</pre>
         DataString(13) <= LCDNumberW(3)+48;</pre>
         DataString(14) <= LCDNumberW(4)+48;</pre>
         DataString(15) <= LCDNumberW(5)+48;</pre>
        DataString(16) <= LCDNumberW(6)+48;</pre>
         DataString(17) <= LCDNumberW(7)+48;</pre>
         DataString(18) <= LCDNumberW(8)+48;</pre>
      ELSE
        DataString(11) <= LCDNumber(1)+48;</pre>
         DataString(12) <= LCDNumber(2)+48;</pre>
         DataString(13) <= LCDNumber(3)+48;</pre>
         DataString(14) <= LCDNumber(4)+48;</pre>
         DataString(15) <= LCDNumber(5)+48;</pre>
         DataString(16) <= LCDNumber(6)+48;</pre>
         DataString(17) <= LCDNumber(7)+48;</pre>
        DataString(18) <= LCDNumber(8)+48;</pre>
      END IF;
         DataString(19) <= 32;</pre>
                         <= 254;
         DataString(20)
                           <= 192;
         DataString(21)
                         <= 254;
         DataString(22)
        DataString(23)
                           <= 192;
        If Regime = 0 THEN -- exter
         DataString(24)
                           <= 101;
         DataString(25)
                           <= 120;
```

```
DataString(26) <= 116;</pre>
 DataString(27) <= 101;</pre>
 DataString(28) <= 114;</pre>
END IF;
If Regime = 100 THEN -- 1 s
 DataString(24) <= 32;</pre>
 DataString(25) <= 1+48;</pre>
 DataString(26) <= 32;</pre>
 DataString(27) <= 115;</pre>
 DataString(28) <= 32;</pre>
END IF;
If Regime = 1000 THEN -- 10 s
 DataString(24) <= 1+48;</pre>
 DataString(25) \leftarrow 0+48;
 DataString(26) <= 32;</pre>
 DataString(27) <= 115;</pre>
 DataString(28) <= 32;</pre>
END IF;
DataString(29) <= 32;</pre>
IF SequenceMode = 0 THEN -- sngl
  DataString(30) <= 115;</pre>
  DataString(31) <= 110;</pre>
  DataString(32) <= 103;</pre>
  DataString(33) <= 108;</pre>
END IF;
IF SequenceMode = 1 THEN -- cycl
  DataString(30) <= 99;</pre>
  DataString(31) <= 121;</pre>
  DataString(32) <= 99;</pre>
  DataString(33) <= 108;
END IF;
IF Regime = 0 THEN
  DataString(30) <= 176;</pre>
  DataString(31) <= 176;</pre>
  DataString(32) <= 176;</pre>
  DataString(33) <= 176;</pre>
END IF;
  DataString(30) <= 32;</pre>
  DataString(31) <= 32;</pre>
  DataString(32) <= 32;</pre>
  DataString(33) <= 32;</pre>
 DataString(34) <= 32;</pre>
 DataString(35) <= 32;</pre>
```

```
DataString(36) <= 32;</pre>
             DataString(37) <= 32;</pre>
             DataString(38) <= 32;</pre>
             DataString(39) <= 32;</pre>
           DataReady <= true;</pre>
 END IF;
END Process;
______
Process (clk100Hz,RST,Go_Go,Go_Go_EXT,Regime, CntReqExtINV,
Busy, test_sig)
Variable CNT1 : natural := 0;
Variable CNT2 : natural := 0;
Variable WRK : natural := 0;
Variable WRKEXT : natural := 0;
Variable CNT_EXT : natural := 0;
Variable EXT_WRK : natural;
Variable IterCount : natural;
BEGIN
IF RST = '1' THEN Busy <= False; WRK := 0; WRKEXT := 0;
CNT1 := 0; CNT2 := 0; CNT_EXT := 0; EXT_WRK := 0;
                   EXT_WRK_Signal_Operator <= High;</pre>
IterCount := 0;
ELSIF Rising_Edge(clk100Hz) THEN
       IF (Go\_Go = '1') and (WRK = 0) THEN WRK := 1;
IterCount := 1;
                  CNT1 := 0; END IF;
     IF (Go_Go_EXT = High) and (WRKEXT = 0)and (Regime = 0)
THEN
     WRKEXT := 1;
     CNT\_EXT := 0;
     --test2 <= High;
END IF;
       IF (WRK = 1) and (Regime > 0) THEN
          Busy <= True;
               IF CNT2 < IterCount THEN
                  CNT1 := CNT1 + 1;
                   IF (CNT1 \geq 0) and (CNT1 < 2) THEN
test sig
             <= High;
                        END IF;
                   IF (CNT1 >= 2)
                                                   THEN
test_sig
            <= Low;
                       END IF;
```

```
IF (CNT1 >= 0) and (CNT1 < 2) THEN
cnt_done
             <= Low;
                        END IF;
                   IF (CNT1 >= 2) and (CNT1 < 3) THEN
cnt_done
             <= High;
                        END IF;
                  IF (CNT1 >= 3)
                                                 THEN
cnt_done
            <= Low;
                        END IF;
                  IF (CNT1 >= 0) and (CNT1 < 4) THEN
DataShowReq <= False;</pre>
                        END IF;
                  IF (CNT1 >= 4) and (CNT1 < 6) THEN
IF (CNT1 >= 6) and (CNT1 < Regime+6 + 3)THEN
DataShowReq <= False;</pre>
                       END IF;
                   IF (CNT1 \geq 0) and (CNT1 < 5) THEN
Cnt_RST
            <= Low;
                        END IF;
                  IF (CNT1 >= 5) and (CNT1 < 6) THEN
Cnt RST
            <= High;
                        END IF;
                  IF (CNT1 >= 6)
                                                THEN
Cnt_RST <= Low;
                        END IF;
                   IF (CNT1 >= 0) and (CNT1 < 6)
THEN CNT_REQ <= Low;
                     END IF;
                   IF (CNT1 >= 6) and (CNT1 < Regime+6)
THEN CNT_REQ <= High; END IF;
                   IF (CNT1 >= Regime+6)
THEN CNT_REQ <= Low;
                     END IF;
              --IF (CNT1 >= 0) and (CNT1 < Regime+6 + 1)
                        END IF;
THEN cnt_done <= Low;
                   IF (CNT1 \geq Regime+6 + 1) and (CNT1 \leq
Regime+6 + 3) THEN cnt_done <= High; END IF;
                   IF (CNT1 \geq Regime+6 + 3)
THEN cnt_done <= Low;
                        END IF;
               IF (CNT1 >= 0) and (CNT1 < Regime+6 + 3)
THEN DataShowReq <= False; END IF;
                  IF (CNT1 \geq Regime+6 + 3) and (CNT1 \leq
Regime+6 + 4) THEN DataShowReq <= True; END IF;</pre>
                  IF (CNT1 \geq Regime+6 + 4)
THEN DataShowReq <= False; END IF;
                   IF (CNT1 \geq Regime + 11)
THEN WRK := 0; Busy <= False; CNT1 := 0; END IF;
      ELSE
            IF Regime > 0 THEN
             CNT1 := CNT1 + 1;
```

```
IF (CNT1 \geq 0) and (CNT1 < 10 + 1)
THEN cnt_done
               <= Low;
                          END IF;
                   IF (CNT1 >= 10 + 1) and (CNT1 < 10 + 2)
THEN cnt done
              <= Hiqh;
                          END IF;
                   IF (CNT1 >= 10 + 2)
THEN cnt done <= Low;
                          END IF;
                   IF (CNT1 \geq 0) and (CNT1 < 10 + 3)
THEN DataShowReq <= False; END IF;
                   IF (CNT1 >= 10 + 3) and (CNT1 < 10 + 4)
THEN DataShowReq <= True;</pre>
                            END IF;
                   IF (CNT1 >= 10 + 4)
THEN DataShowReq <= False; END IF;
                   IF (CNT1 >= 10 + 7)
     CNT1 := 0; WRK := 0; END IF;
THEN
       END IF; -- Working
   END IF; -- clk1m
END Process;
Process (CntReqExtINV, EXT_WRK_Signal,
EXT_WRK_Signal_Operator)is
BEGIN
 IF RST = '1' THEN EXT WRK Signal <= Low;
 ELSE
    IF rising_edge(CntReqExtINV)
                                         THEN EXT_WRK_Signal
<= High;
    ELSIF EXT_WRK_Signal_Operator = Low THEN EXT_WRK_Signal
<= Low;
    END IF;
 END IF;
END Process;
 Process(CounterIN, CNT_REQ, EXT_WIN, Cnt_RST,
Regime,cnt_done) is -- Counter1 internal windows
  Variable Transfer : natural;
  Variable i
                    : natural;
  Variable LCDCNT : natural;
 BEGIN
   IF cnt_RST = '1' THEN
        LCDCNT := 0;
        LCDNumber(0) <= 0;
        LCDNumber(1) <= 0;
        LCDNumber(2) <= 0;
```

```
LCDNumber(3) <= 0;
       LCDNumber(4) <= 0;</pre>
       LCDNumber(5) <= 0;
       LCDNumber(6) <= 0;
       LCDNumber(7) <= 0;</pre>
       LCDNumber(8) <= 0;</pre>
  ELSIF Rising Edge(CounterIN) THEN
      IF (cnt_req = High) THEN
       -- LCDCNT := LCDCNT + 1;
       Transfer := 1;
       For i in 8 downto 0 LOOP
        IF LCDNumber(i) < 9 and Transfer = 1 THEN
LCDNumber(i) <= LCDNumber(i) + Transfer; Transfer := 0;</pre>
        ELSIF Transfer = 1 THEN LCDNumber(i) <= 0;</pre>
Transfer := 1;
        END IF;
       END LOOP;
     END IF;
          IF (LCDCNT > 99999999) THEN -- overflow
    LCDNumber(0) <= 160-48;
    LCDNumber(1) <= 79-48;
    LCDNumber(2) <= 94-48;
    LCDNumber(3) <= 69-48;
    LCDNumber(4) <= 82-48;
    LCDNumber(5) <= 70-48;
    LCDNumber(6) <= 76-48;
    LCDNumber(7) <= 79-48;
    LCDNumber(8) <= 87-48;
          END IF;
  END IF;
END Process;
 _____
Process (clk50m, Step1, Step2, EXT_WIN) is
BEGIN
  IF RST = '1' THEN Step1 <= Low; Step2 <= Low;</pre>
  ELSIF rising_edge(clk50m) THEN
     Step1 <= Step2;
     Step2 <= EXT_WIN;</pre>
  END IF;
END Process;
-----
```

168

```
Process(CounterIN, CNT_REQ, EXT_WIN, Regime, Step1,Step2)
is -- Counter2 external window
  Variable Transfer1 : natural;
  Variable i1
                      : natural;
  Variable LCDCNT : natural;
 BEGIN
   IF (Step1 = Low) and (Step2 = High) THEN
        LCDNumberW(0) <= 0;</pre>
        LCDNumberW(1) <= 0;</pre>
        LCDNumberW(2) <= 0;
        LCDNumberW(3) <= 0;</pre>
        LCDNumberW(4) <= 0;</pre>
        LCDNumberW(5) <= 0;</pre>
        LCDNumberW(6) <= 0;</pre>
        LCDNumberW(7) <= 0;</pre>
        LCDNumberW(8) <= 0;
        LCDCNT := 0;
   ELSIF Rising_Edge(CounterIN) THEN
      IF ((EXT_WIN = High)and(Regime = 0)) THEN
        LCDCNT := LCDCNT + 1;
        Transfer1 := 1;
        For i1 in 8 downto 0 LOOP
         IF LCDNumberW(i1) < 9 and Transfer1 = 1 THEN</pre>
LCDNumberW(i1) <= LCDNumberW(i1) + Transfer1; Transfer1 :=</pre>
0;
         ELSIF Transfer1 = 1 THEN LCDNumberW(i1) <= 0;</pre>
Transfer1 := 1;
         END IF;
        END LOOP;
      END IF;
           IF (LCDCNT > 99999999) THEN -- overflow
     LCDNumberW(0) <= 160-48;
     LCDNumberW(1) <= 79-48;
     LCDNumberW(2) <= 94-48;
     LCDNumberW(3) <= 69-48;
     LCDNumberW(4) \le 82-48;
     LCDNumberW(5) <= 70-48;
     LCDNumberW(6) <= 76-48;
     LCDNumberW(7) <= 79-48;
     LCDNumberW(8) <= 87-48;
           END IF;
   END IF;
```

```
END Process;
______
_____
Process (b_mode,Busy)-- Regime
BEGIN
 IF RST = '1' THEN Regime <= 100;</pre>
ELSIF Rising_edge(b_mode) and not Busy THEN
        IF Regime = 0 THEN Regime <= 100; END IF;
        IF Regime = 100 THEN Regime <= 1000; END IF;
        IF Regime = 1000 THEN Regime <= 0;
                                               END IF;
 END IF;
END Process;
Process(clk1m,btnSequence,Busy)
BEGIN
 If rising_edge(clk1m) THEN
      IF not Busy THEN
        signal_dff2 <= btnSequence;</pre>
        b_Sequence <= btnSequence and (not signal_dff2);</pre>
      END IF;
 END IF;
END Process;
Process (clk1m, b_Sequence, Busy) -- Regime
BEGIN
 IF RST = '1' THEN SequenceMODE <= 0;</pre>
ELSIF (Rising_Edge(clk1m) and (b_Sequence= '1')) and not
Busy THEN
        IF SequenceMODE = 0
                                THEN SequenceMODE <= 1;
END IF;
        IF SequenceMODE = 1
                                THEN SequenceMODE <= 0;
END IF;
END IF;
END Process;
Process(clk100Hz,btnStart,Busy)
BEGIN
 If rising_edge(clk100Hz) THEN
      IF not Busy THEN
        signal_dff <= btnStart;</pre>
        Go_Go <= btnStart and (not signal_dff);</pre>
      END IF;
 END IF;
END Process;
--Process(EXT_WIN_INV,Busy)
--BEGIN
```

```
-- If rising_edge(EXT_WIN_INV) THEN
        IF not Busy THEN
           signal_dff3 <= High;</pre>
           Go_Go_EXT <= High and (not signal_dff3);</pre>
      END IF;
-- END IF;
--END Process;
Process(clk1m,btnMode,Busy)
Variable WT : boolean;
Variable CNT : natural;
BEGIN
 IF RST = '1' THEN WT := false; CNT := 0;
 ELSIF rising_edge(clk1m) THEN
      IF (not Busy) and (not WT)THEN
         signal_dff1 <= btnMode;</pre>
         b_mode <= btnMode and (not signal_dff1);</pre>
         WT := True;
      END IF;
      IF b_mode = '1' THEN WT := true; END IF;
      IF WT THEN
        CNT := CNT + 1;
          IF CNT = 10000 THEN WT := false; CNT := 0; END
IF;
      END IF; -- Wait
 END IF;
END Process;
Process (Go_Go,CntReqEXT,Regime)
Variable Hunt : boolean;
BEGIN
 IF Go_Go = High THEN Hunt := True;
 ELSE
      IF (rising_edge(CntReqEXT) and Hunt)
                                                    THEN
EXT_WIN <= High; Hunt := False; END IF;</pre>
      IF ((CntRegEXT = Low ) and (Hunt = False)) THEN
EXT_WIN <= Low;</pre>
                                 END IF;
 END IF;
END Process;
Process(clk50Hz,EXT_WIN,Busy,Go_Go_EXT) --
Variable Hunt : boolean;
BEGIN
```

# Program BS2\_FPU. BasicStamp2 with floating point unit. Microprocessor part of the temperature measurement system.

```
'{$STAMP BS2}
'{$PBASIC 2.5}
'-----
_____
·-----
'----- uM-FPU definitions (V2.1) ------
'-----
PIN 10
FpuClk
                          ' SPI clock (connects to
SCLK/SCL)
FpuOut
             PIN 9
                       ' SPI data out (connects to
SIN/SDA)
FpuIn
             PIN 9
                       ' SPI data in
                                   (connects to
SOUT)
#IF ($STAMP = BS2SX) OR ($STAMP = BS2P) OR ($STAMP = BS2PX)
#THEN
 ResetTime
           CON 625
                     ' 500 usec reset pulse
#ELSE
 ResetTime
           CON 250
                     ' 500 usec reset pulse
#ENDIF
'-----
SELECTA
           CON $00
                     ' select A register
                     ' select B register
           CON $10
SELECTB
            CON $20
                     ' select A and write float to
FWRITEA
register
FWRITEB
           CON $30
                      ' select B and write float to
register
FREAD
           CON $40
                     ' read float from register
FSET
           CON $50
                     'A = REG
LSET
           CON $50
                     'A = REG
                     'A = A + REG (float)
FADD
           CON $60
           CON $70
                     'A = A - REG (float)
FSUB
FMUL
           CON $80
                     'A = A * REG (float)
FDIV
           CON $90
                     'A = A / REG (float)
           CON $A0
                     'A = A + REG (long)
LADD
                     173
```

```
' A = A - REG (long)
LSUB
               CON $B0
                           'A = A * REG (long)
LMUL
               CON $C0
LDIV
               CON $D0
                            'A = A / REG (long)
SORT
               CON $E0
                           ' A = sqrt(A)
                           'A = ln(A)
               CON $E1
LOG
LOG10
               CON $E2
                           'A = log(A)
                           ' A = e ** A
EXP
               CON $E3
               CON $E4
                           ' A = 10 ** A
EXP10
                           ' A = sin(A) radians
               CON $E5
FSIN
FCOS
               CON $E6
                           'A = cos(A) radians
                           ' A = tan(A) radians
               CON $E7
FTAN
               CON $E8
                           ' A = nearest integer <= A
FLOOR
                           ' A = nearest integer >= A
CEIL
               CON $E9
                          ' A = nearest integer to A
ROUND
               CON $EA
                           'A = -A
NEGATE
              CON $EB
                           ' A = |A|
               CON $EC
FABS
                           ' A = 1 / A
INVERSE
              CON $ED
               CON $EE
                             ' A = A / (PI / 180) radians
DEGREES
to degrees
               CON $EF
                            ' A = A * (PI / 180) degreees
RADIANS
to radians
               CON $F0
                          ' synchronization
SYNC
FLOAT
                 CON $F1
                               ' copy A to register 0 and
float
FIX
               CON $F2
                            ' copy A to register 0 and fix
               CON $F3 ' compare A and B
FCOMPARE
LOADBYTE
                  CON $F4
                                ' write signed byte to
register 0, convert to float
                 CON $F5
                                ' write unsigned byte to
LOADUBYTE
register 0, convert to float
                                ' write signed word to
LOADWORD
                  CON $F6
register 0, convert to float
                                ' write unsigned word to
                 CON $F7
LOADUWORD
register 0, convert to float
                       ' read zero terminated string
READSTR
               CON $F8
                  CON $F9
ATOF
                               ' convert ASCII to float,
store in A
               CON $FA
                            ' convert float to ASCII
FTOA
               CON $FB
                           ' convert ASCII to long, store
ATOL
in A
LTOA
               CON $FC
                            ' convert long to ASCII
                            ' get the status of A register
FSTATUS
               CON $FD
               CON $FE
                            ' extended opcode
XOP
NOP
               CON $FF
                           ' nop
FUNCTION
               CON $00
                           ' (XOP) user functions 0-15
               CON $90
                          ' (XOP) read low 8 bits of A
READBYTE
(long)
```

```
READWORD
                 CON $91
                               ' (XOP) read low 16 bits of A
(long)
READLONG
                 CON $92
                               ' (XOP) read 32-bit long value
from A
                                 ' (XOP) read floating point
READFLOAT
                  CON $93
value from A
LINCA
                CON $94
                              ' (XOP) A = A + 1 (long)
LINCB
                CON $95
                              ' (XOP) A = A + 1 (long)
                              ' (XOP) A = A - 1 (long)
                CON $96
LDECA
                CON $97
                              ' (XOP) A = A - 1 (long)
LDECB
LAND
                CON $98
                                (XOP) A = A AND B (long)
                                (XOP) A = A OR B (long)
LOR
                CON $99
LXOR
                CON $9A
                              ' (XOP) A = A XOR B (long)
LNOT
                CON $9B
                              ' (XOP) A = NOT A (long)
                                   ' (XOP) status of A AND B
LTST
                   CON $9C
(long)
                   CON $9D
                                   ' (XOP) shift A by B bits
LSHIFT
(long)
                  CON $A0
                                  ' (XOP) select A and write
LWRITEA
long to register
                                  ' (XOP) select B and write
LWRITEB
                  CON $B0
long to register
                                ' (XOP) read 32-bit long from
                 CON $C0
LREAD
register
                 CON $D0
                                ' (XOP) A = A / REG (unsigned
LUDIV
long)
                              ' (XOP) A = A ** B
                CON $E0
POWER
ROOT
                CON $E1
                              ' (XOP) A = the Bth root of A
                              ' (XOP) A = minimum of A and B
FMIN
                CON $E2
                CON $E3
                              ' (XOP) A = maximum of A and B
FMAX
                 CON $E4
FRACTION
                                ' (XOP) load register 0 with
the fractional part of A
ASIN
                CON $E5
                              ' (XOP) A = asin(A) radians
ACOS
                              ' (XOP) A = acos(A) radians
                CON $E6
ATAN
                CON $E7
                              ' (XOP) A = atan(A) radians
ATAN2
                CON $E8
                              ' (XOP) A = atan(A/B)
                              ' (XOP) long compare A and B
LCOMPARE
                CON $E9
LUCOMPARE
                 CON $EA
                                ' (XOP) unsigned long compare
A and B
                              ' (XOP) long status
LSTATUS
                CON $EB
LNEGATE
                CON $EC
                              ' (XOP) A = -A (long)
                              '(XOP) A = |A| (long)
LABS
                CON $ED
LEFT
                CON $EE
                              ' (XOP) right parenthesis
                CON $EF
                              ' (XOP) left parenthesis
RIGHT
                 CON $F0
                                ' (XOP) load register 0 with
LOADZERO
zero
                                ' (XOP) load register 0 with
LOADONE
                 CON $F1
1.0
LOADE
                CON $F2
                              ' (XOP) load register 0 with e
```

```
CON $F3
                              ' (XOP) load register 0 with
рi
LONGBYTE
                CON $F4
                              ' (XOP) write signed byte to
register 0, convert to long
                            ' (XOP) write unsigned byte to
LONGUBYTE
               CON $F5
register 0, convert to long
LONGWORD
                CON $F6
                              ' (XOP) write signed word to
register 0, convert to long
LONGUWORD
               CON $F7
                            ' (XOP) write unsigned word to
register 0, convert to long
IEEEMODE
                   CON $F8
                                   ' (XOP) set IEEE mode
(default)
PICMODE
               CON $F9
                            ' (XOP) set PIC mode
                                   ' (XOP) calculate code
CHECKSUM
                   CON $FA
checksum
BREAK
               CON $FB
                            ' (XOP) debug breakpoint
TRACEOFF
               CON $FC
                            ' (XOP) turn debug trace off
TRACEON
               CON $FD
                            ' (XOP) turn debug trace on
                CON $FE
                             ' (XOP) send debug string to
TRACESTR
trace buffer
                            ' (XOP) get version string
VERSION
               CON $FF
               CON $5C
                           ' sync character
SyncChar
_____
dataWord
               VAR
                       Word
                                     ' data word
dataHiqh
                 VAR
                          dataWord.HIGHBYTE ' high byte of
dataWord
dataLow
                 VAR
                          dataword.LOWBYTE ' low byte of
dataLow
dataByte
               VAR
                       dataLow
                                     ' (alternate name)
                                        ' opcode (same as
opcode
                VAR
                         dataHigh
dataHigh)
                VAR
                        dataHiqh
                                       ' register (same as
req
dataHigh)
format
                         dataLow
                                         ' format (same as
                VAR
dataLow)
                                         ' status (same as
status
                VAR
                         dataLow
dataLow)
                                        ' Zero status bit
status Zero
                VAR
                         status.BIT0
(0-not zero, 1-zero)
status_Sign
                         status.BIT1
                                        ' Sign status bit
                VAR
(0-positive, 1-negative)
                           status.BIT2 ' Not a Number
status_NaN
                 VAR
status bit (0-valid number, 1-NaN)
```

LOADPI

	VAR infinite, 1			3 ' I	infinity	status
	=======================================			uM-FPU	defir	nitions
'======= =============================	========	======	:====:	======		=====
	=======================================		main		defir	nitions
	=======					=====
=======						
	=======	======	:=====	======	======	=====
Reset:						
establish 'proper example di: 'message the uM-FPU	if the rese	ons with	the	uM-FPU.	The fol	lowing
	u_Reset s <> SyncCha "uM-FPU not		п	,	reset t	he FPU
version nu	Print_Versio mber " - Detected		ialize		play the	
' (Your	initializati	on code w	ould be	e inserte	ed here.	1
'======= =========	=======================================	======	=====	======	======	=====
·		main rou	tine -			
'========	=======================================	=======	=====	======	======	=====

## Main:

```
' Coded for ESE 440
' Version 1.0 7-14-2006
' Changes:
   Version 1.0 - Final version, everything works
   Version 0.42 - Sum works
                      Lots of DEBUG statements removed to
conserve memory
       Version 0.40 - Program is pin-compatible with
prototyping board
  Version 0.35 - Mostly content complete
                 Added LCD display - flawless
                 Am yet to interface with FPGA
                 Added progress display
    Version 0.31 - Array 's' returned. Memory now managed
effectively
                       Sum returned is satisfactory - 100
iterations take 30 seconds
                          Formatting, removed unnecessary
variables, neatification
   Version 0.30 - Removed array s entirely to save memory;
using array f twice
                 Added pulse
                 Added formula for sum
                 Added iterations
    Version 0.29 - Made the program loop to start after
finish
   Version 0.28 - Reduced number of LONG digits to 7
   Version 0.26 - Added LCD display - flawed
   Version 0.25 - Added Start button on Pin 0
   Version 0.2 - Added FPU functionality
'----- uM-FPU V2 Register Definitions -----
______
firstNum
               CON
                      1
                                      ' uM-FPU register 1
                                      ' uM-FPU register 2
secondNum
               CON
                      2
               CON
                                      ' uM-FPU register 3
sum
'----- Variable Definitions ------
_____
                                       ' float variable -
readNum
               VAR
                     Byte
read from FPU
                  VAR
                        Byte(9)
                                           ' signed byte
variable - First number
                        Byte(9)
                                           ' signed byte
                  VAR
variable - Second number
```

```
iter
               VAR
                    Word
                                      ' Iterations count
(max of 16000)
inCount
               VAR
                    Byte
                                     ' This counter will
be used in loops for serial input
                                       ' This determines
RunTo
               VAR
                     Word
the number of iterations to run to
'----- Constant Definitions ------
baudRate
                     16468
                                 ' Define the Baud Rate
               CON
coming from FPGA here
                                  '16468 = 9600 Baud, 8
data bits, No parity, INVERTED
                                    ' What pin is the LCD
LCDpin
               CON
                     5
connected to?
BaudLCD
               CON 32
                                  ' What is the Baud rate
of the LCD?
PulsePin
               CON 4
                                  ' This pin pulses once
every iteration
DatPin
                CON 3
                                       ' What pin is FPGA
inputting on?
RunTo = 100
                                         ' Set number of
iterations HERE!
HIGH LCDpin
                                ' LCD initialization
PAUSE 100
SEROUT LCDpin, BaudLCD, [18, 12] 'Clear LCD
PAUSE 50
INPUT 7
                         ' Setting pin7 as an input pin to
start operation
DEBUG "Push start to begin.", CR
SEROUT LCDpin, BaudLCD, ["Push start.", CR]
                          ' Program will wait here until
wait0:
start button is pushed
IF IN7 <> 0 THEN wait0
IF IN7 = 0 THEN IN7 = 1
SEROUT LCDpin, BaudLCD, [18, 12]
                                                  ' Clear
LCD
PAUSE 50
DEBUG "Beginning ", DEC RunTo, " iterations. ", CR
  '--- sum = 0
                                 ' Initialize sum to 0
  SHIFTOUT FpuOut, FpuClk, MSBFIRST, [sum, XOP, LOADZERO,
FSET]
```

```
SEROUT LCDpin, BaudLCD, [128, "Working..."] 'Display
progress
FOR iter = 1 TO RunTo
                         ' Iterations to run the program
until-----
                           ' Send out a minimum duration
'PULSOUT PulsePin,0
pulse every iteration
HIGH PulsePin
PAUSE 1
LOW PulsePin
  SERIN DatPin, baudRate, [f(7)] ' Capturing the
first number
  SERIN DatPin, baudRate, [f(6)]
  SERIN DatPin, baudRate, [f(5)]
  SERIN DatPin, baudRate, [f(4)]
  SERIN DatPin, baudRate, [f(3)]
  SERIN DatPin, baudRate, [f(2)]
  SERIN DatPin, baudRate, [f(1)]
  SERIN DatPin, baudRate, [f(0)]
 SERIN DatPin, baudRate, [readNum(0)] ' Discard the last
digit
  SERIN DatPin, baudRate, [s(7)]
                                          ' Capturing the
second number
  SERIN DatPin, baudRate, [s(6)]
  SERIN DatPin, baudRate, [s(5)]
  SERIN DatPin, baudRate, [s(4)]
  SERIN DatPin, baudRate, [s(3)]
  SERIN DatPin, baudRate, [s(2)]
  SERIN DatPin, baudRate, [s(1)]
  SERIN DatPin, baudRate, [s(0)]
  SERIN DatPin, baudRate, [readNum(0)] 'Discard the last
digit
DEBUG "First number received = ",DEC f(7),DEC f(6),DEC
f(5), DEC f(4), DEC f(3), DEC f(2), DEC f(1), DEC f(0), CR
DEBUG "Second number received = ",DEC s(7),DEC s(6),DEC
s(5), DEC s(4), DEC s(3), DEC s(2), DEC s(1), DEC s(0), CR
       firstNum = (((f(4) + (f(5)*10) + (f(6)*100) +
(f(7)*1000))*100)*100) +
  '((f(0)) + (f(1)*10) + (f(2)*100) + (f(3)*1000))
  SHIFTOUT FpuOut, FpuClk, MSBFIRST, [firstNum, XOP, LEFT,
XOP, LEFT, XOP, LEFT,
```

LOADBYTE, f(4), FSET, XOP, LEFT, LOADBYTE, f(5),

FSET,

FWRITEB, \$41, \$20, \$00, \$00, FMUL, XOP, RIGHT, FADD, XOP, LEFT,

LOADBYTE, f(6), FSET]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [FWRITEB, \$42, \$C8, \$00, \$00, FMUL,

 $\label{eq:condition} \texttt{XOP, RIGHT, FADD, XOP, LEFT, LOADBYTE, f(7),} \\ \texttt{FSET,}$ 

FWRITEB, \$44, \$7A, \$00, \$00, FMUL, XOP, RIGHT, FADD, XOP, RIGHT, FSET,

FWRITEB, \$42, \$C8, \$00, \$00, FMUL]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [XOP, RIGHT, FSET,

FWRITEB, \$42, \$C8, \$00, \$00, FMUL, XOP, RIGHT, FSET, XOP, LEFT,

LOADBYTE, f(1), FSET, FWRITEB, \$41, \$20, \$00, \$00]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [FMUL, XOP, RIGHT, FADD, XOP, LEFT,

LOADBYTE, f(2), FSET, FWRITEB, \$42, \$C8, \$00, \$00, FMUL, XOP, RIGHT,

FADD, XOP, LEFT, LOADBYTE, f(3), FSET, FWRITEB, \$44, \$7A, \$00, \$00,

FMUL, XOP, RIGHT, FADD]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [XOP, RIGHT, FADD]

'--- secondNum = (((s(4) + (s(5)\*10) + (s(6)\*100) + (s(7)\*1000))\*100)\*100) +

'((s(0)) + (s(1)\*10) + (s(2)\*100) + (s(3)\*1000))

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [secondNum, XOP, LEFT, XOP, LEFT, XOP, LEFT,

LOADBYTE, s(4), FSET, XOP, LEFT, LOADBYTE, s(5), FSET,

FWRITEB, \$41, \$20, \$00, \$00, FMUL, XOP, RIGHT, FADD, XOP, LEFT,

LOADBYTE, s(6), FSET]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [FWRITEB, \$42, \$C8, \$00, \$00, FMUL,

FWRITEB, \$44, \$7A, \$00, \$00, FMUL, XOP, RIGHT, FADD, XOP, RIGHT, FSET,

FWRITEB, \$42, \$C8, \$00, \$00, FMUL]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [XOP, RIGHT, FSET, FWRITEB, \$42, \$C8, \$00, \$00, FMUL, XOP, RIGHT,

FSET, XOP, LEFT,

XOP, LEFT, LOADBYTE, s(0), FSET, XOP, RIGHT,

FSET, XOP, LEFT,

LOADBYTE, s(1), FSET, FWRITEB, \$41, \$20, \$00, \$00]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [FMUL, XOP, RIGHT, FADD, XOP, LEFT,

LOADBYTE, s(2), FSET, FWRITEB, \$42, \$C8, \$00, \$00, FMUL, XOP, RIGHT,

FADD, XOP, LEFT, LOADBYTE, s(3), FSET, FWRITEB, \$44, \$7A, \$00, \$00,

FMUL, XOP, RIGHT, FADD]

GOSUB Fpu\_Wait

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [XOP, RIGHT, FADD]

'--- sum = (sum + (firstNum / secondNum))

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [sum, XOP, LEFT, XOP, LEFT, FSET+sum,

XOP, LEFT, FSET+firstNum, FDIV+secondNum, XOP, RIGHT, FADD, XOP, RIGHT,

FSET, XOP, RIGHT, FSET]

SEROUT LCDpin, BaudLCD, [148,DEC (RunTo-iter),"/",DEC RunTo," "]

NEXT ' Next Iteration, loop ends here----

'--- sum = (sum/RunTo)

SHIFTOUT FpuOut, FpuClk, MSBFIRST, [sum, XOP, LEFT, XOP, LEFT, FSET+sum,

LOADWORD, RunTo.HIGHBYTE, RunTo.LOWBYTE, FDIV, XOP, RIGHT, FSET,

XOP, RIGHT, FSET]

SEROUT LCDpin, BaudLCD, [12," Done!"] 'Display progress DEBUG "Value of sum : "

xxxxxxxxxxxx

GOSUB PRINT\_FLOAT'FORMAT

DEBUG CR

```
DEBUG "Try to display float : "
SEROUT LCDpin, BaudLCD, [13, "Sum = "]
 opcode = FTOA
 SHIFTOUT FpuOut, FpuClk, MSBFIRST, [opcode, format]
 GOSUB Fpu_Wait
                                ' wait until uM-FPU
is ready
 SHIFTOUT FpuOut, FpuClk, MSBFIRST, [READSTR]
 DO
                                    ' display zero
terminated string
   SHIFTIN FpuIn, FpuClk, MSBPRE, [dataByte]
   IF (dataByte = 0 OR dataByte > 127) THEN EXIT
   DEBUG dataByte
   SEROUT LCDpin, BaudLCD, [dataByte]
 LOOP
SEROUT LCDpin, BaudLCD, [CR]
DEBUG CR
DEBUG "Program stopped.", CR, CR
 END
'-----
'----- uM-FPU SPI support routines (V2.1) --
______
·-----
Fpu_Reset:
                               ' set clock and data
 LOW FpuClk
lines Low
 LOW FpuOut
 PULSOUT FpuClk, ResetTime
                                ' send reset pulse
to uM-FPU
 PAUSE 8
                                    check for
synchronization
 SHIFTOUT FpuOut, FpuClk, MSBFIRST, [SYNC]
 SHIFTIN FpuIn, FpuClk, MSBPRE, [status]
 RETURN
Fpu_Wait:
 INPUT FpuIn
                                 ' (required for 2-
wire interface)
Fpu_Wait2:
 is ready
```

## RETURN

```
Print_Version:
                                           ' get the uM-FPU
version string
  SHIFTOUT FpuOut, FpuClk, MSBFIRST, [XOP, VERSION]
  GOTO Print_String2
                                      ' print it
Print Float:
  format = 0
                                       ' set format to zero
(free format)
                                      ' (fall through to
Print_FloatFormat)
Print_FloatFormat:
  opcode = FTOA
                                         ' convert floating
point to formatted ASCII
                                      ' print the value
 GOTO Print_String
Print_Long:
                                          ' set format to 0
  format = 0
(free format)
                                      ' (fall through to
Print_LongFormat)
Print_LongFormat:
 opcode = LTOA
                                            ' convert long
integer to formatted ASCII
                                      ' (fall through to
Print_String)
                                          ' send conversion
Print_String:
command
  SHIFTOUT FpuOut, FpuClk, MSBFIRST, [opcode, format]
Print_String2:
                                        ' wait until uM-FPU
 GOSUB Fpu_Wait
is ready
 SHIFTOUT FpuOut, FpuClk, MSBFIRST, [READSTR]
                                            ' display zero
 DO
terminated string
    SHIFTIN FpuIn, FpuClk, MSBPRE, [dataByte]
    IF (dataByte = 0 OR dataByte > 127) THEN EXIT
   DEBUG dataByte
 LOOP
 RETURN
'======== end of uM-FPU SPI support routines
```

## Program BSX\_02. BasicX+FPGA tau measurement system.

'Version history: Version 1.0 Final version, everything works Start & Reset button added Summation now displays correctly Version 0.22 Attempting to iron out bugs in FPGA communication Version 0.1 LCD works Public Sub Main() 'Allocate global data buffers for serial transfer Dim InBuffer(1 To 30) As Byte 'LCD data buffer Dim OutBuffer(1 To 30) As Byte 'LCD data buffer Dim InBufferFPGA(1 To 30) As Byte 'FPGA data buffer Dim OutBufferFPGA(1 To 30) As Byte 'FPGA data buffer Dim C1(9) As Byte 'We only need 9 \* 1 spaces to catch 9 values of first number Dim C2(9) As Byte 'We only need 9 \* 1 spaces to catch 9 values of second number Dim CN(9) As Byte 'This is to capture the 9 possible digits of N Dim FirstNum As Single 'The first number could be huge after recombination; Single is the largest data type Dim SecondNum As Single 'The second number could be huge after recombination Dim NumPhot As Single 'The number  $\circ f$ photons in the time interval Dim Sum As Single 'The sum needs be as accurate as possible; Single is the largest floating point data type Dim Tau As Single 'The decay parameter that is calculated every iteration and must be averaged at the end Dim CurrentIteration As Long 'How many iterations have been completed? Dim Temp As Byte 'Just variable а for temporary calculations Dim Temp2 As Byte 'Just another variable for temporary calculations Const TotalIterations As Long = 160 'The number of iterations Const ProgressUpdateRate As Long = 5 'Increase this

to speed up process (by reducing update frequency)

```
Const FPGA_baud As Long = 9600
                                   'The incoming baud
rate to the FPGA
Const FPGA_pin As Byte = 13
                                         'The
                                                pin
                                                       that's
connected to the FPGA (pulse)
Const FPGA_data_pin As Byte = 16
                                    ' The
                                             pin
                                                      that's
receiving data from FPGA (serial)
Const LCD_baud As Long = 19200
                                         'The outgoing baud
rate to the LCD
Const LCD_pin As Byte = 17
                                         'Pin
                                                       that's
transferring data TO to LCD
Const debug_disable As Byte = 0
                                         ' 1
                                                  No
                                                       status
messages on console, 0 = otherwise
                                                  button
Const start_button As Byte = 5
                                        'Start
                                                           is
connected to pin 5
                                        'Variable to detect
Dim start_run As Byte
start button
'The following code initializes Catcher1
C1(0) = 0
C1(1) = 0
C1(2) = 0
C1(3) = 0
C1(4) = 0
C1(5) = 0
C1(6) = 0
C1(7) = 0
C1(8) = 0
'The following code initializes Catcher2
C2(0) = 0
C2(1) = 0
C2(2) = 0
C2(3) = 0
C2(4) = 0
C2(5) = 0
C2(6) = 0
C2(7) = 0
C2(8) = 0
'The following code initializes Catcher for NumPhot (CN)
CN(0) = 0
CN(1) = 0
CN(2) = 0
CN(3) = 0
CN(4) = 0
```

```
CN(5) = 0
CN(6) = 0
CN(7) = 0
CN(8) = 0
Sum = 0!
                                        'Initializing
                                                         Sum
to 0
FirstNum = 0!
                                        'Initializing
FirstNum to 0
SecondNum = 0!
                                   'Initializing
                                                   SecondNum
NumPhot = 0!
                                        'Initializing
NumPhot to 0
'LCD communication
'Define port 3 to use pin 0 (dummy pin) as input, pin
<LCD_pin> as output.
'Use non-inverted logic, no parity, 8 data bits.
Call OpenQueue(InBuffer, 30)
Call OpenQueue(OutBuffer, 30)
Call DefineCom3(0, LCD_pin, bx0000_1000)
Call OpenCom(3, LCD_baud, InBuffer, OutBuffer)
     'Open the port to LCD
Call PutQueueStr(OutBuffer, Chr(22) & Chr(12) & Chr(17))
     'Clear display, turn on backlight
Call Sleep(0.01)
     'Recover from clear display
'Wait for start button to be pressed
Call PutQueueStr(OutBuffer, "Press start.")
If debug_disable = 0 Then
     Debug.Print "Press start to begin."
End If
Do
     start run = GetPin(5)
     'Read status of start button
Loop Until (start_run = 0)
If debug_disable = 0 Then
     Debug.Print "Beginning ";CStr(TotalIterations);"
cycles."
End If
Call PutQueueStr(OutBuffer, Chr(12))
     'Clear LCD
Call Sleep(0.01)
     'Recover from LCD clearing
Call PutQueueStr(OutBuffer, "Completed 0% of")
```

```
Call PutQueueStr(OutBuffer, Chr(148))
Call PutQueueStr(OutBuffer, CStr(TotalIterations))
Call PutQueueStr(OutBuffer, " cycles.")
Call Sleep(0.01)
Call ClearQueue(InBufferFPGA) 'Clear the input buffer
Call ClearQueue(OutBuffer)
                                        'Clear the output
buffer
     For CurrentIteration = 1 To TotalIterations
          'Open port 3 to use pin <FPGA_data_pin> as input
and pin 0 (dummy) as output
          'Use INVERTED logic, no parity, 8 data bits
          Call OpenQueue(InBufferFPGA, 20)
          Call OpenQueue(OutBufferFPGA, 20)
          Call DefineCom3(FPGA_data_pin, 0, bx10001000)
          Call
                  OpenCom(3, FPGA baud,
                                               InBufferFPGA,
OutBufferFPGA)
          'The code below generates a pulse on FPGA_pin
(which is needed to trigger the FPGA's output)
          Call PutPin (FPGA_pin, bxOutputHigh)
                                                       'Set
FPGA pulse pin HIGH
          Call PutPin (FPGA_pin, bxOutputLow)
                                                       'Set
FPGA pulse pin LOW
          'The code below captures 9 digits of each number
          For Temp = 0 To 8
first number (9 digits)
               Call GetQueue(InBufferFPGA, Temp2, 1)
               C1(Temp) = Cbyte(Temp2)
                                                  'Store
digit in C1
          Next
          For Temp = 0 To 8
                                                       'Get
second number (9 digits)
               Call GetQueue(InBufferFPGA, Temp2, 1)
               C2(Temp) = CByte(Temp2)
                                                  'Store
digit in C2
          Next
          For Temp = 0 To 8
                                                       'Get
number of photons (9 digits)
               Call GetQueue(InBufferFPGA, Temp2, 1)
               CN(Temp) = CByte(Temp2)
digit in CN
          Next
```

```
following debug code
         'The
                                      shows
                                            the
                                                  digits
gathered from the FPGA (in order), for debugging purposes
         If debug_disable = 0 Then
              'Debug.Print
                                "FirstNum=";CStr(C1(0));"
";CStr(C1(1));"
                    ";CStr(C1(2));" ";CStr(C1(3));"
                    ";CStr(C1(5));"
";CStr(C1(4));"
                                         ";CStr(C1(6));"
";CStr(C1(7));" ";CStr(C1(8))
              ";CStr(C2(1));"
                    ";CStr(C2(2));"
                                          ";CStr(C2(3));"
";CStr(C2(6));"
";CStr(C2(7));" ";CStr(C2(8))
         End If
         'Recombination begins here
         FirstNum = 0!
    'Initialize values to 0 every iteration
         SecondNum = 0!
    'Initialize values to 0 every iteration
         NumPhot = 0!
              For Temp = 0 To 8
                  FirstNum = FirstNum + (Csng(C1(8-Temp))
* (10.0 ^ Temp))
                  SecondNum = SecondNum + (CSnq(C2(8-
Temp)) * (10.0 ^ Temp))
                  NumPhot = NumPhot + (CSng(CN(8-Temp)) *
(10.0 ^ Temp))
             Next
         'The following debug code shows the two numbers
after recombination
         If debug_disable = 0 Then
              'Debug.Print
"Firstnum=";CStr(CLng(FirstNum))
              'Debug.Print
"Secondnu="; CStr(CLng(SecondNum))
         End If
         FirstNum = FirstNum / (10.0 ^ 9)
    'Compensate for the fact that FirstNum is always in nS
         'Sum = (Sum + (FirstNum / SecondNum))
    'First part of Averaging equation
         Tau = Tau + (FirstNum/Log(1!-(1!/NumPhot)))
    'First part of decay equation
         'The following code updates the LCD with current
progress
         If (CurrentIteration Mod ProgressUpdateRate) = 0
Then
```

```
'LCD communication
               'Define port 3 to use pin 0 (dummy pin) as
input, pin 17 as output.
               'Use non-inverted logic, no parity, 8 data
bits.
               Call OpenQueue(InBuffer, 30)
               Call OpenQueue(OutBuffer, 30)
               Call DefineCom3(0, LCD_pin, bx0000_1000)
     'Define COM to LCD on previously chosen pin (LCD_pin)
               Call
                       OpenCom(3, LCD_baud,
                                                    InBuffer,
OutBuffer)
               Call Sleep(0.01)
               If debug_disable = 0 Then
                    Debug.Print
                                                   "Progress:
"; CStr(CByte((CSng(CurrentIteration)/Csng(TotalIterations))
*100!));" %"
               End If
               Call PutQueueStr(OutBuffer, Chr(138))
     'Position cursor at position 10
               Call
                                      PutOueueStr(OutBuffer,
CStr(CInt((CSng(CurrentIteration)/Csng(TotalIterations))*10
0!)))
          'Show progress %
               Call PutQueueStr(OutBuffer, "%")
               Call Sleep(0.005)
          End If
     Next
Tau = Abs(Tau)
     'Tau can never be negative
'Sum = Sum/CSng(TotalIterations)
     'Second part of averaging equation
Tau = Tau/CSnq(TotalIterations)
     'Second part of decay equation
Call PutQueueStr(OutBuffer, Chr(12) & Chr(17))
     'Clear display, turn on backlight
Call Sleep(0.01)
     'Recover from clear display
Call PutQueueStr(OutBuffer, "Job completed.")
     'Show completion
If debug_disable = 0 Then
     Debug.Print "Job complete!";" Tau ="
     Debug.Print CStr(Tau)
```

```
End If
Call PutQueueStr(OutBuffer, Chr(148))
     'Position cursor @ Line 2, Position 0
Call PutQueueStr(OutBuffer, "Tau=")
Call PutQueueStr(OutBuffer, CStr(CSng(Tau)))
     'Show the final result
Call PutQueueStr(OutBuffer, Chr(128))
     'Infinite loop, waiting for Reset button to be pushed
Do
Call Sleep(2.0)
Call PutQueueStr(OutBuffer, Chr(128))
Call PutQueueStr(OutBuffer, "Job complete! ")
Call Sleep(2.0)
Call PutQueueStr(OutBuffer, Chr(128))
Call PutQueueStr(OutBuffer, "Press Reset. ")
Loop
End Sub
```

## **Basic Patent**



## (12) United States Patent Luryi et al.

US 6,944,407 B2 (10) Patent No.: (45) Date of Patent: Sep. 13, 2005

## METHOD AND APPARATUS FOR DETECTING RADIATION

Inventors: Serge Luryi, Old Field, NY (US); Vera Gorfinkel, Stony Brook, NY (US); Mikhail Gouzman, Lake Grove, NY

Assignee: The Research Foundation of State University of New York, Stony Brook,

Subject to any disclaimer, the term of this (\*) Notice:

patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: 10/322,293 (22) Filed: Dec. 18, 2002

(65) **Prior Publication Data** 

US 2003/0123883 A1 Jul. 3, 2003

## Related U.S. Application Data

Division of application No. 09/186,248, filed on Nov. 4, 1998, now Pat. No. 6,528,801.

Int. Cl.<sup>7</sup> ...... H04B 10/06; H04B 10/158 .... **398/202**; 398/151; 398/186; 380/229; 380/256; 380/263 U.S. Cl. .....

398/186, 98, 101, 398/102, 103, 151, 156, 202, 203; 380/263, 203, 210, 229, 256

#### (56)References Cited

## U.S. PATENT DOCUMENTS

5,171,534	A	12/1992	Smith et al.
5,757,912	A *	5/1998	Blow
5,784,157	A	7/1998	Gorfinkel et al.
5,793,049	A	8/1998	Ballard
5,818,057	A	10/1998	Buck
5,940,545	A	8/1999	Kash et al.
6,043,506	A	3/2000	Heffelfinger et al.
6.071.748	Α	6/2000	Modlin et al

#### 6,137,584 A 10/2000 Seidel et al.

#### FOREIGN PATENT DOCUMENTS

9823941 4/1998 WO

#### OTHER PUBLICATIONS

"Coded multipulse modulation in optical communication "Code multipuise modulation in optical communication systems"; Atkin et al; Communications, IEEE Transactions on, vol.: 42, Issue: 234, Feb.—Apr. 1994, pp.: 574–582.\* D.Y. Chen et al., "Single Molecule Detection in Capillary Electrophoresis: Molecular Shot Noise as a Fundamental Limit to Chemical Analysis," Analytical Chemistry vol. 68, pp. 600-606 (1906). pp. 690-696 (1996). W.R. McCluney, "Introduction to Radiometry and Photom-

w.K. McCliney, Inflouence to Radiometry and Photometry," Artech House, Jun. 30, 1994, pp. 114–122.

Alan Smith, "Selected Papers on Photon Counting Detectors," SPIE (Milestone Series), vol. MS413, Feb. 4, 1998 (ISBN: 0–8194–2788–8), pp. 194–202, published by the Society of Photo–optical Instrumentation Engineers (SPIE).

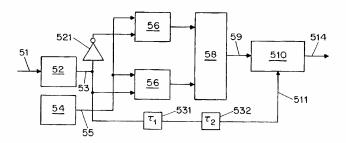
\* cited by examiner

Primary Examiner-M. R. Sedighian Assistant Examiner—Nathan Curs
(74) Attorney, Agent, or Firm—Baker Botts LLP

## ABSTRACT

In analyzing radiation from a communication link, single-photon counting can be used to advantage especially at low levels of radiation energy, e.g. in the detection of optical radiation. Preferred detection techniques include methods in which (i) received optical radiation is intensity-modulated in accordance with a preselected code, (ii) wherein it is the optical radiation which is intensity-modulated with the pre-selected code, and (iii) wherein the radiation modulated with a preselected code is received. For registration of the signals received by a sensing element of a single-photon detector, time of arrival is recorded, optionally in conjunction with registration of time intervals. Advantageously, in the interest of minimizing the number of pulses missed due to close temporal spacing of pulses, D-triggers can be included in counting circuitry.

## 3 Claims, 2 Drawing Sheets



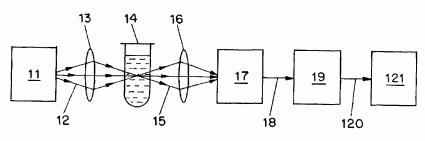


FIG. 1

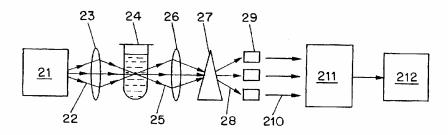


FIG. 2

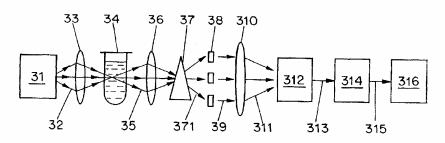
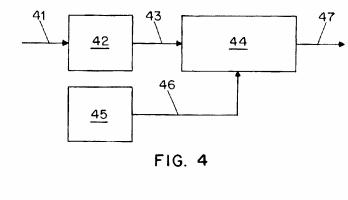
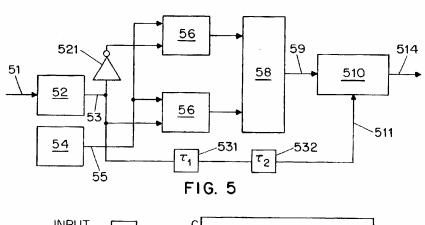
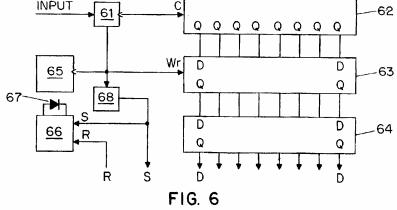


FIG. 3







#### 2

## METHOD AND APPARATUS FOR DETECTING RADIATION

#### TECHNICAL FIELD

The invention is concerned with analytical technology and, more specifically, with the detection of a fluorescent species or fluorophore in a sample. This application is a division of application Ser. No. 09/186,248, filed 4 Nov. 1998, now U.S. Pat. No. 6,528,801.

#### BACKGROUND OF THE INVENTION

Fluorescent species or fluorophores emit fluorescent radiation when suitably stimulated by stimulating radiation. The emitted radiation can be used for chemical/biological analytic purposes, e.g. in determining whether a fluorophore of interest is present in a sample and in quantifying its concentration. One analytic technique of this type is disclosed in U.S. Pat. No. 5,171,534 to Smith et al. wherein DNA fragments produced in DNA sequencing are characterized on the basis of fluorescence of chromophores tagged to the fragments. Stimulating electromagnetic radiation may be monochromatic, or may include significant energy in a plurality of energy bands, e.g. as disclosed in U.S. Pat. No. 5,784,157 to Gorfinkel et al.

The stimulating radiation usually varies in time, either stochastically or regularly. Regular variation of the radiation intensity can be introduced artificially by modulating the intensity of the radiation source or the transmittance or reflectance of a filter element in the optical path. Regularly modulated radiation may be termed as encoded radiation if the temporal variation of the radiation is used as a carrier of information. Associated with such encoded radiation is a temporal code, i.e. a time-domain function which corresponds to the temporal evolution of the intensity of modulated radiation. A time-domain function can be formed as a linear combination of several suitable functions whose respective contributions to the linear combination can be quantified reliably. Suitable in this respect are sinusoidal functions of time, for example, oscillating at distinct frequencies.

In prior-art techniques, the encoded radiation is considered as continuous, with the time dependence of detected radiation intensity regarded as a continuous time-domain 45 function.

Further background includes several known singlephoton detection techniques for which W. R. McCluney, Introduction to Radiometry and Photometry, Artech House. 1996, pp. 114-122 provides a general introduction. Such techniques are designed for measuring modulated radiation, and they can be classified into two groups: (a) asynchronous photon counting and (b) synchronous detection. As described in Alan Smith, Selected Papers on Photon Counting Detectors, SPIE, Vol. MS 413, 1998, methods (a) of 55 asynchronous photon counting involve the detection of a number of photons during a fixed time interval, e.g. one second, called the registration interval. These methods allow the determination of an average frequency of photon arrival. This frequency varies in time, either stochastically or regularly, and synchronous counting can be employed to measure the time variation. An essential limitation of this method is associated with the impossibility of measuring frequencies of modulation that are higher than the repetition rate of registration intervals. This difficulty is inherent in the principle of asynchronous counting, which is to keep track of the total number of photons received during the registra-

tion interval rather than register their times of arrival. A difficulty arises when the highest frequency  $\mathbf{f}_{mod}$  in the modulation spectrum of modulation radiation is comparable to or higher than the average frequency  $\mathbf{f}_{phot}$  of single-photon detection. In this case, if the frequency limit is increased by reducing the time interval chosen for counting, the technique becomes increasingly inefficient because the counter will count nothing during most registration intervals

Methods (b) of synchronous detection involve measurement of the time of arrival of incident single photons. This time may be referenced to an "absolute" clock, or may be measured relative to or "synchronously with" a triggering excitation signal. The triggering signal may be associated with the arrival of the first of detected photons, for example, Such methods are particularly valuable for application to fast processes, e.g. the fluorescent decay of a single excited dye molecule as described, e.g., by D. Y. Chen et al., "Single Molecule Detection in Capillary Electrophoresis: Molecular Shot Noise as a Fundamental Limit to Chemical Analysis", Analytical Chemistry, Vol. 68 (1996), pp. 690–696, typically requiring special electronics for handling fast temporal variations. An essential limitation of these methods is associated with the difficulty of maintaining records of high temporal resolution for a relatively long time. Thus, detecting photon arrivals at the temporal resolution corresponding to nanosecond time intervals over a one-second period requires acquisition of a billion data records. This makes methods of synchronous detection difficult to apply to the photometry of relatively slowly varying modulated singlephoton fluxes

#### SUMMARY OF THE INVENTION

We have recognized that, in detecting a fluorescent species in a sample, single-photon counting can be used to advantage, especially at low levels of fluorescent signal energy. Preferred detection techniques include methods in which (i) fluorescence-stimulating radiation is intensity-modulated in accordance with a preselected code, (ii) wherein it is the fluorescent radiation which is intensity-modulated with the preselected code, and (iii) wherein modulation with a preselected code is applied to a sample to influence a property, e.g. temperature, pressure, or an electric or magnetic field strength or frequency which functionally affects emitted fluorescent radiation.

Preferably, for registration of the signals from a sensing element of a single-photon detector, time of arrival is recorded, optionally in conjunction with registration of time intervals. Advantageously, in the interest of minimizing the number of pulses missed due to close temporal spacing of pulses, D-triggers can be included in counting circuitry.

The preferred techniques are generally applicable to photometry of time-encoded single-photon or particle fluxes. They involve measurement of time intervals between single-photon/particle arrivals combined with data analysis that permits decoding of the encoded radiation, i.e., discrimination between alternative possible codes and quantification of different combinations of mixtures of the codes. The techniques provide for the time intervals between successive pulses to be measured asynchronously, without requiring an external clock reference or special triggering signal. They provide for efficient measurement and decoding of time-encoded single-photon or particle fluxes.

## BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 is a schematic of a preferred first technique in accordance with the invention, using a modulated light source.

FIG. 2 is a schematic of a preferred second technique in accordance with the invention, using a dispersive element.

FIG. 3 is a schematic of a preferred third technique in ccordance with the invention, involving temporal encoding of different spectral components of a fluorescent signal.

FIG. 4 is a schematic of a preferred fourth technique in accordance with the invention, for registration of temporal parameters of a stochastic sequence of pulses of constant or similar shape.

FIG. 5 is a schematic of a preferred fifth technique in accordance with the invention, wherein the fourth technique is integrated with the measurement of time intervals.

FIG. 6 is a schematic of a preferred sixth technique in accordance with the invention, wherein the fourth technique is augmented for further minimization of pulses lost to registration

#### DETAILED DESCRIPTION

For purposes of the present description, no distinction 20 need be made between "photon" and "quantum", as each can result in a detector signal, typically an electrical signal or pulse for electronic processing in accordance with techniques of the invention. Use of other types of signal processing is not precluded, e.g. by opto-electronic or purely optical means. It is understood that, in alternative processing means, a detector signal or a pulse being processed can be other than an electric signal or pulse

A. Single-Photon Detection in Methods for Fluorophore Identification

A special illumination technique is used, with a plurality of modulated narrow-band sources, each modulated according to its own distinguishable time-domain function. The narrow-band sources excite different fluorophores differently, so that the emitted fluorescent radiation is encoded with information about the nature and composition of illuminated fluorescent species. Photons are detected individually.

In a preferred first embodiment as illustrated by FIG. 1, a modulated multi-band light source producing encoded radiation of excitation of fluorescence is combined with singlephoton detection of encoded fluorescence signal.

FIG. 1 shows the light source 11 producing a radiation flux 12 which, via an optical illumination system 13, is incident on the container 14 holding a fluorescent sample. The radiation flux 12 comprises a plurality of spectral bands, each modulated according to its own distinguishable timedomain function. Fluorescent radiation 15 emitted by the fluorescent sample is received by an optical receiver system, e.g. an objective 16, and is directed to the optical input of a single-photon detector 17. The output of the detector 17 is a stochastic stream 18 of electric pulses of similar shape, and information about the intensity of the received fluorescent radiation in a set time interval is contained in the average frequency of the pulses arriving in the interval. The temporal characteristics of the stream 18 of electric pulses are registered in a suitable form by the recorder 19 which is described below in further detail, in connection with FIGS, 4 and 5. In a preferred embodiment, the stochastic stream of pulses is characterized in terms of the spacing in time between arrivals of successive pulses. The detection system may be complemented by communication means 120 for transferring the recorded information at an appropriate rate from the recorder 19 to a signal processor unit 121.

A preferred second embodiment as illustrated by FIG. 2 can be viewed as an improvement over a known method for multicolor fluorescent detection, e.g. as disclosed in the

above-referenced patent to Smith et al. In this technique, the fluorescent radiation emitted by an excited molecule is optically analyzed into distinct wavelength channels, e.g. by a prism or a diffraction grating. The intensity of fluorescent radiation in each of the wavelength channels is then determined by photometric means. In the preferred second embodiment, sensitivity is enhanced by the use of single-

photon detection. FIG. 2 shows radiation 22 from a modulated optical source 21 being focused by a lens 23 onto a fluorescent sample 24. The modulated optical source 21 may produce one or several spectral bands that are modulated either together or independently with distinct time domain functions. Fluorescence 25 emitted by the sample 24 in response to the incident radiation 22 is directed by an objective 26 to an optical processor which comprises a dispersive element 27, e.g. a prism or a diffraction grating, and a set 29 of single photon detectors (SPD). The dispersive element 27 effects spectral analysis of the fluorescent signal.

Each of the SPD's produces at its output a stochastic stream of electrical pulses of similar shape, and information about the intensity of the received fluorescent radiation is contained in the temporal characteristics of the stochastic stream. With reference to FIG. 2, the temporal characteristics 210 from each SPD are registered by a recorder 211 whose structure is described below in further detail in connection with FIGS. 4 and 5. In a preferred embodiment, also described below in further detail in connection with FIGS. 4 and 5, the description of the stochastic stream of pulses is specified in terms of the time separations between arrivals of successive pulses. The detection system further comprises a signal processor unit 212 and means for transferring the recorded information at an appropriate rate from the recorder 211 to the signal processor unit 212

FIG. 2 illustrates combination of a modulated light source for excitation of fluorescence with a dispersive element for analyzing the fluorescent response into distinct spectral bands, and single-photon detection of modulated fluorescence in each of the spectral bands. Additionally, as in FIG. 1, the modulated light source can be multi-band also, so that the radiation flux 22 comprises a plurality of spectral bands, each modulated according to its own distinct time domain function. In this case, a preferred technique is advantageous further in that different fluorescent species are distinguished both by their fluorescence emission spectrum and their fluorescence excitation spectrum. This enhances the fidelity of fluorophore identification.

A preferred third embodiment of the invention, illustrated by FIG. 3, can be viewed as an improvement over a known technique for multicolor fluorescent detection, e.g. as applied according to the above-referenced patent to Smith et al. The known technique is combined with single-photon detection, using a modulation technique disclosed in U.S. patent application Ser. No. 08/946,414, filed Oct. 7, 1997 now abandoned, by Gorfinkel et al. In accordance with the latter technique, radiation reflected, transmitted, or fluorescently emitted by an object is encoded in such a way that the encoded radiation carries information about properties of the object, e.g. its color as characterized by reflected wavelengths, or the identity and quantitative content of fluorescent species present in the object. In the present embodiment of the invention, temporal encoding of different spectral components of a fluorescent signal is combined with single-photon detection of the encoded spectral components, for enhanced sensitivity.

FIG. 3 shows radiation 32 from optical source 31 being

focused by an objective 33 onto a fluorescent sample 34. In

5

contrast to the embodiments illustrated by FIGS. 1 and 2, the optical source 31 need not be modulated, and the radiation 32 may or may not be encoded. Fluorescence 35 emitted by by an objective 36 onto an optical processor which comprises a dispersive element 37, e.g. a prism or a diffraction grating, and a set of optical modulators 38. The dispersive element 37 effects spectral analysis of the fluorescence 35. The spectral components are directed onto a set of optical modulators 38 which modulate in time the resolved spectral components in such a way that each different resolved spectral component is coded by a distinct function of time. The modulated components 39 of the fluorescent spectrum are combined by an optical element 310 into an optical flux 311 focused onto the optical input of the single-photon detector 312. The output of the detector 312 represents a stochastic stream 313 of electrical pulses of similar shape, whose temporal characteristics are registered by the recorder 314 which is described below in further in connection with FIGS. 4 and 5. In a preferred embodiment, also described below in further detail, the description of the stochastic 2 stream of pulses is specified in terms of the temporal separation between arrivals of successive pulses. The detection system further comprises means 315 for transferring the recorded information at an appropriate rate to a signal processor unit 316

B. Single Photon Detection of Modulated Photon Fluxes
A preferred fourth embodiment of the invention is illustrated by FIG. 4, of a method for registration of temporal parameters of a stochastic sequence of pulses of constant or similar shape.

The recorder of FIG. 4 operates with a controlled time resolution, controlled by a clock 45 which provides a regular sequence 46 of electrical pulses of constant shape which define the recording time intervals. A stochastic stream 41 of electric input pulses may originate from a sensing element of a single-photon detector which is typically a photomultiplying tube (PMT) or an avalanche photo diode (APD).

The input pulses are not required to be of the same shape. With an APD, a special avalanche quenching circuit is used, either passive or active. Typically, the APD is pre-biased into its avalanche regime, for the first photon to initiate the avalanche. To prepare for the next photon arrival, the avalanche has to be quenched. It may be advantageous to use a so-called forced-quenching circuit which regularly quenches the avalanche condition, irrespective of whether an avalanche had actually been initiated, so that the arrival of photons and the time of quenching are not correlated. As a result, the avalanche-pulse duration will be stochastic also, depending on the time of photon arrival relative to subsequent quenching.

The stream of pulses 41 is directed to an n-state cyclic state-shift device or register 42. Such a device has n successive stable states which may be numbered 0, 1, 2, ..., n-1, with a change from a state k to its successor state k+1 being triggered by an input pulse, and with state n-1 having state 0 as its successor state. Between input pulses, the n-state cyclic state-shift device 42 retains its state. For example, for a 2-state cyclic state-shift device a flip-flop can be used, having a sequence of stable states 0, 1, 0, 1, ..., with each input pulse causing a transition from 0 to 1 or from 1 to 0. It is not necessary that the cyclic state-shift device return to its initial state when its state is read. This is in contrast to conventional photon counters where each reading of the counter data is accompanied by resetting the state of the counter back to the initial state.

For the sake of specificity, without limiting the invention, a flip-flop will be assumed in the following further descrip-

6

tion of FIG. 4. The output from the flip-flop represents a stochastic sequence 43 of rectangular pulses of variable length. The sequence 43 is directed to a recording device 44, which can be realized as an analog or digital signal recorder. The output signal 47 is transferred from the recording device 44 to a signal processor (not shown).

44 to a signal processor (not shown).

The recorder of FIG. 4 operates essentially in an asynchronous mode. But, in contrast to asynchronous photon counters which record the total number of photons arriving in a particular time interval, the preferred recorder records their times of arrival. Accuracy of recording of the arrival time is controlled by the clock 45.

Time intervals are recorded without measuring the duration of the intervals. This function can be performed by one of a number of devices known to those skilled in the art, placed in an electrical circuit serially with the recorder and using its output signal 47. For example, a general-purpose computer can be used to process the array of data acquired by the recording device 44.

In some applications it may be advantageous to integrate in a single device the functions of registering the time intervals between successive single photon detections and measurement of the time intervals. Such an integrated preferred fifth embodiment of the invention is illustrated by FIG. 5, for a stochastic stream of electric pulses 51 to which the shape and APD-quenching considerations concerning pulses 41 of FIG. 4 are applicable also.

As shown in FIG. 5, a stochastic stream of electric pulses 51 is directed onto a flip-flop 52. Its output represents a stochastic sequence 53 of rectangular input pulses of variable length. The sequence 53 is split three ways between counters 56 and 56' and the controlled delay line 531. The counter 56 receives the signal from the flip-flop directly, and the counter 56' receives its signal through an inverter 521. Thus, the counters 56 and 56' are controlled by opposite-phase signals. Instead of a flip-flop, 52, an n-state cyclic state-shift device can be used, as described with reference to FIG. 4. Advantageously in this case, instead of two counters, 56 and 56', up to n counters can be used.

56 and 56', up to n counters can be used.

A clock 54 provides a regular sequence 55 of electric pulses of constant shape which are counted by the counter 56. Exemplarily, counter 56 is that counter whose input signal equals 1 at the time of clock pulse arrival. Advantageously, if the pulses 51 originate from and APD, the external quenching circuit which periodically forces the APD out of its avalanche regime can be synchronized by the clock 54. There is no advantage in increasing the quenching frequency beyond the clock frequency which provides the basic discretization of time in the technique.

When a photon is detected and an electric pulse 51 enters the flip-flop 52, one of the counters 56 and 56' stops counting and the other begins counting. The one counter that has just stopped counting then contains the record 57 of how long the interval between two successive pulses has lasted, measured in terms of the number of clock cycles counted. The record 57 is transferred to the recording device 510 through a commutator 58 which serves to provide successive recording at intervals of time so that, while one time interval is being recorded, the next one is being measured. The commutator 58 is controlled by a switch signal which is derived by input signals 53 delayed by a characteristic time  $\tau_1$  corresponding to the response time of the counter 56. The output of the commutator 58 represents a sequence of codes 59 describing the measured time intervals between detected photons. The codes 59 appear at the output of the commutator 58 in stochastic fashion corresponding to the detection of incoming photons and delayed by the time interval which

is the sum of  $\tau_1$  and the response time  $\tau_2$  of the commutator itself. It is advantageous, therefore, to control the recording device 510 by switch signals which are derived from the input signals 53, delayed from the moment of flip-flop switching by the time  $\tau_1\text{+}\tau_2.$  The output 514 of the recording device 510 represents the same sequence 59 of codes describing the measured time intervals between detected photons. In contrast to the sequence 59, which is accumulated in time stochastically, the sequence 514 can be transmitted in a regular fashion, e.g. at a constant rate, for further processing

Further to the technique illustrated by FIG. 4, FIG. 6 illustrates inclusion of D-triggers for minimizing the number of pulses uncounted due their close spacing in time. Electric 15 pulses from a single-photon detector output are directed through a fast switch 61 to the input C of a synchronous 8-bit binary counter 62. The result of the count is passed to the storage register 63 as an 8-bit word or byte. To avoid changing the state of the counter 62 during storage, the synchronous pulse generator 65 shuts off the switch 61 simultaneously with sending a short record pulse to the input Wr of the storage register 63. The output from the storage register 63 goes through the buffer 64 directly to the parallel port of a computer. Operational control error indicator is facilitate by a logic comparator 66 equipped with an LED (light emitting diode) 67. The parallel computer port is synchronized by a synchronous pulse through a delay line 68 with a suitable delay τ. The same delayed pulse synchronizes the logic comparator 66.

For an exemplary embodiment of the the technique illustrated by FIG. 6, the following may be specified and realized: a discretization frequency of 125 KHz, a maximum number of pulses per discretization interval of 256, a minimum time between registered pulses of 20 ns, a maximum 35 average frequency of registered pulses of 32 MHz, and a maximum fraction of missed photons of 0.25%

Techniques of the invention can be used to advantage in a variety of applications involving encoded electromagnetic radiation, including multicolor luminescent detection based 40 on fluorescence spectroscopy and fluorescence excitation spectroscopy. They can be used in general sensor applications with other modulated luminescence signals, e.g., those based on various spectroscopic techniques such as transmission, absorption, reflection, or Raman spectra, as well as electro-luminescence, chemiluminescence and the like. The techniques are especially useful for detecting weak signals, e.g. those prevalent in optical communication links where signals are transmitted over long optical fibers.

What is claimed is:

1. A method for detecting a radiated signal from an optical communication link, comprising the steps of:

8

- (a) detecting successive photons of an intensitymodulated signal from said optical communication link with the modulation being over time in accordance with a preselected code;
- (b) determining time intervals, each time interval being a period of time between instances of detection of two single successive photons;
- (c) recording a sequence of said time intervals; and
- (d) comparing the recorded sequence with said code. 2. Apparatus for detecting a radiated signal from an
- optical communication link, comprising:
  - (a) a detector moiety for detecting successive photons from an intensity-modulated signal from said optical communication link, with the modulation being over time in accordance with a preselected code;
- (b) a time-interval determination moiety operationally coupled to said detector moiety for determining time intervals, each time interval being a period of time between instances of detection of two single successive photons:
- (c) a recorder moiety operationally coupled to said timeinterval determination moiety for recording a sequence of said time intervals; and
- (d) a comparator moiety operationally coupled to said recorder moiety for comparing the recorded sequence with said code.
- 3. Apparatus for detecting a radiated signal from an optical communication link, comprising:
  - (a) detector means for detecting successive photons of an intensity-modulated signal from said optical communication link, with the modulation being over time in accordance with a preselected code;
  - (b) time-interval determination means operationally coupled to said detector means for determining time intervals, each time interval being a period of time between instances of detection of two single successive
  - (c) recorder means operationally coupled to said timeinterval determination means for recording a sequence of said time intervals; and
  - (d) comparator means operationally coupled to said recorder means for comparing the recorded sequence with said code.