

The presented work was submitted to the
Institute of Integrated Photonics

Development of a Control System for Photonic Integrated Circuits

Master's Thesis in Electrical Engineering

written at

RWTH Aachen University

by

Arka Dipta Das

submitted on

20.11.2020

Examiner

Univ.-Prof. Jeremy Witzens, PhD

Supervisor

Seyed Alireza Tabatabaei Mashayekh, M.Sc.

Communicated by

Univ.-Prof. Jeremy Witzens, PhD

Authorship

I hereby declare that I wrote my Master's Thesis on my own and that I have followed the regulations relating to good scientific practice of RWTH Aachen University in its latest form. I did not use any unacknowledged sources or means and I marked all references I used literally or by content.

Date

Signature

Abstract

Silicon photonics is at the helm of the data communication technologies used for densely-integrated wavelength-based optical communication [1]. Its CMOS compatibility allows the increasing demand for high-bandwidth and low-power demands to be easily met. Micro-ring resonators have garnered a lot of attention as a versatile optical device that can be used as optical modulators, switches and filters. However, the high thermo-optic coefficient of silicon ($1.8 \times 10^{-4} \text{K}^{-1}$ at 300K [2]) and frequency sensitivity to fabrication tolerances ($\sim 100 \text{GHz/nm}$) are reportedly deleterious to their use in dense wavelength division multiplexing applications where the channel wavelength spacings are tight and temperatures can vary as much as 15°C [3]. Numerous potent solutions for addressing this challenge have been proposed such as athermalization [4, 5] and thermal stabilization by control based methods [6, 7]. However, there is scant literature on the application of control-based methods on higher-order ring resonator based optical devices.

In this work, we show experimentally how wavelength-locking and tracking of a higher order coupled-resonator optical waveguide (CROW) optical add-drop multiplexer (OADM) with integrated heaters can be carried out using low-speed electronics, a feedback signal via photodiodes and a LabVIEW-based control algorithm. We discuss about the standard methods used to actively compensate the local thermal variations and track the resonant frequency of the OADMs, the concept and architecture of the control algorithm that we developed and how we tested it on a silicon photonic based integrated optical system. We also discuss the factors that make it challenging to implement such a control system. The algorithm is fairly simple to integrate for multiple CROW based optical filters and can carry out the wavelength tracking of several CROW filters in parallel. It is able to recover the wavelength from a $\pm 1^\circ\text{C}$ deviation in chip temperature during tracking within 30 s. The modular structure and parameter-based operation makes it versatile for applications with a range of optical devices that modulate the optical intensity. Additionally, for future development, our algorithm is well suited to carry out controlled increase of bandwidth at the cost of some optical intensity.

Acknowledgements

First and foremost I am grateful to Prof. Jeremy Witzens for giving me the opportunity to complete my master's thesis at the Institute of Integrated Photonics (IPH). I express my gratitude to the whole team of IPH for their continuous assistance and advice during the past months. I wish to acknowledge the supervision provided by Mr. Seyed Alireza Tabatabaei Mashayekh, who constantly supported me during my thesis work. I would also like to express my gratitude to Dr. Alvaro Moscoso Mártir for providing his valuable inputs and assistance. My heartiest thanks are extended to Ms. Juliana Müller and Mr. Andrea Zazzi for also providing their insights. A very special thanks to my parents, sister and my friends who encouraged and supported me during this phase and throughout my studies.

Contents

Statutory Declaration in Lieu of an Oath	ii
1 Introduction	1
2 Technical Background	7
2.1 Athermal methods	8
2.2 Control-based methods	9
3 Concept and Architecture	13
3.1 Concept	13
3.2 Architecture	15
3.2.1 Abstraction of the algorithm	15
3.2.2 The structure and behavior of the program	17
4 Challenges and Solutions	23
5 Measurements	35
5.1 The NLFT Project	35
5.2 Experimental Setup	36
5.3 Execution	40
5.4 Results	43
6 Summary and Outlook	47
A The LabVIEW program	49
A.1 Main VI	51
A.2 Interface VI	55

A.3 Handover VI	57
A.4 Algorithm VI	59
A.5 Main Sweep VI	62
List of Figures	65
List of Tables	69
Bibliography	71
References of Figures	75
Glossary	77

Introduction

Optical Networks

Optical networks are the backbone of today's communication networks. They allow unprecedented data transmission capacities and are reliable in terms of both performance and longevity. In a typical optical network, high-speed bit streams are transmitted from one node to another through optical fibers and at each of these nodes, data is either extracted, added or both operations can be carried out simultaneously. Extracted optical signals are passed on to electronics via transducers such as photo-receivers/photo-diodes (PDs).

Technologies such as wavelength division multiplexing (WDM) introduced multi-fold data transmission capabilities via a single optical fiber. Further developments brought Coarse Wavelength Division Multiplexing (CWDM) and Dense Wavelength Division Multiplexing (DWDM) [8]. The International Telecommunication Union (ITU) established a frequency grid for DWDM which starts with a channel spacing recommendation of integer multiples of 100GHz and goes down to 25GHz or even 12.5GHz (ultra DWDM), allowing operation of more than 160 channels. [9] The objective of WDM is to increase transmission bandwidth by utilizing a larger portion of the C-Band (1530 nm-1565 nm) transmission window. [10] This is done by putting different channels (each at a unique wavelength) onto the same optical fiber. The channels themselves may have different data rates. At the receiving end the desired channels are extracted from the transmission spectrum with the help of **optical filters**, without affecting the other channels in any significant manner. Amongst all the different filters used for applications analogous to WDM, the optical add-drop filter is a widely used filter, often employed as a simple device to introduce WDM capabilities in metro/access networks. It enables a specific channel to be "dropped" from the link and another channel to be "added" to it at the same wavelength.

Optical Filters

An optical filter is a passive optical device/component that is employed to alter the spectral distribution of the optical radiation passing through it [10]. In most applications however, they are used to absorb or expel optical radiation within a chosen bandwidth. This range can either be fixed or tunable, depending on the construct of the device. For most DWDM based applications, these filters are required to have a sharp roll-off and a flat passband. They can be broadly classified into the following two classes:-

FIR Filters: Finite Impulse Response (FIR) filters have a finite response to an impulse input. In simpler words, such a filter has no feedback and thus the filter response goes to zero within a finite time period. Filters utilizing the Mach-Zehnder topology and filters based on Arrayed Waveguide Gratings are notable examples.

IIR Filters: Infinite Impulse Response (IIR) filters on the other hand are able to sustain their finite impulse response for an infinite period (theoretically) with the help of a feedback response. Thus, their response is non-zero for an indefinite period of time. Fiber Bragg Gratings, thin-film filters (e.g. Fabry-Perot filters) and **CROW filters** fall under this category.

OADM's and CROW filters

OADM (optical add-drop multiplexer) is a type of optical filter which is used in wavelength-division multiplexing systems for multiplexing and routing different channels of light into or out of a single mode fiber (SMF) or optical waveguide (in the case of system-on-a-chip configuration). It can also be used as a traditional filter wherein only those signals that fall within the passband of the filter are allowed to be transmitted through it. It is one of the fundamental constructional blocks of the modern day telecommunication network.

CROW (coupled-resonator optical waveguide) are OADM's which can be broadly defined as a cascade of ring resonators that transfer light from one end of the assembly to the other by using near-neighbor coupling. They are composed of *micro-ring resonators* and are generally fabricated on silicon or similar refractive index materials. They are typically used as high performance add/drop filters with on-off ratios exceeding 20dBs. They can be composed of one, two or more ring resonators put in a cascaded topology. The bandwidth of the resulting device is therefore a combination of the transfer functions of each ring resonator.

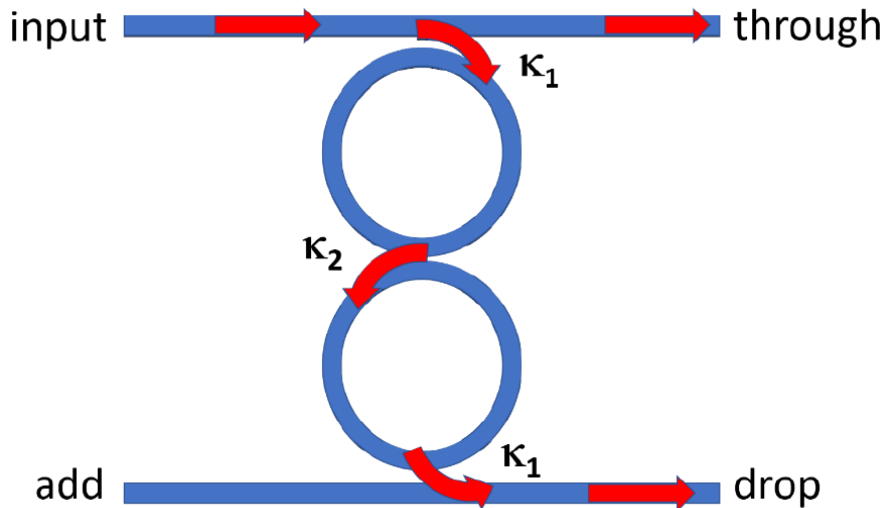


Figure 1.1 A second order CROW filter with coupling coefficients κ_1 between straight waveguide and ring resonator and κ_2 between the two ring resonators. [41]

The characteristics of the filter such as steep roll-off, flat passband, high on-off ratio) are dependent on the energy flow in the rings. To achieve a passband with steeper roll-off and strong rejection in stop-band, several ring resonators need to be cascaded. In such an implementation, the resonant frequency of each ring in the cascade must be matched by design. However, several factors such as fabrication limitations, environmental disturbances, physical effects, etc. are responsible for considerable deviations in the final product of the fabrication. A SRR (Single Ring Resonator) is shown in figure 1.2. In a SRR (and CROW filters with odd numbers of rings), the input and the drop ports lie on the same side (left side of figure 1.2(a)) while the through and inject ports are on the other side (left side of figure 1.2(a)). In case of even order CROW filters, the input and drop ports are diagonally opposite to each other.

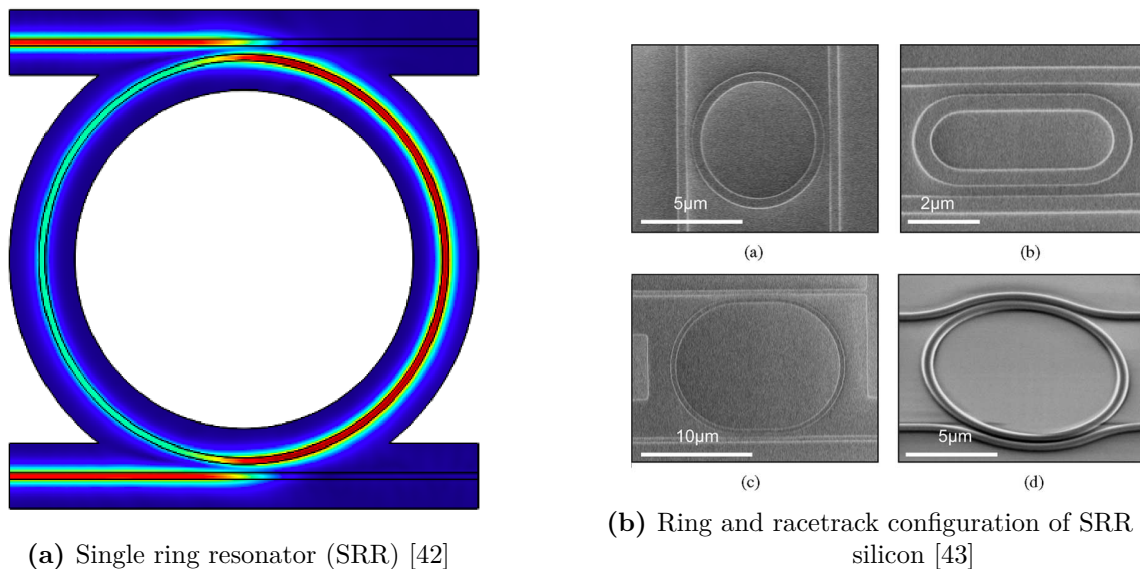


Figure 1.2 A figure

Theoretically, it is possible to cascade a large number of ring resonators to get the desired filter response. More ring resonators improve the roll-off and can provide up to 100dB on-off ratio. Theoretically, approximately 650 rings can be cascaded until further improvement becomes negligible and experiments have proven that 235 rings can be used to design advanced on-chip optical interconnects which are projected to be used in computer systems that operate in the ExaFLOPS regime [11].

Transfer Matrix Formalism of CROW filters

The formalism for CROW filters takes into account input/output coupling, loss, resonator sizes, variations in coupling strengths, etc. which are encountered in practical applications. They have been used to rigorously analyze realistic CROW filters. The following excerpt has been taken from Poon et al. [12].

To begin with, an infinite chain of coupled *ring resonators* is used to obtain its dispersion relation. Both forward and backward propagating waves exist in each individual *ring resonator*.

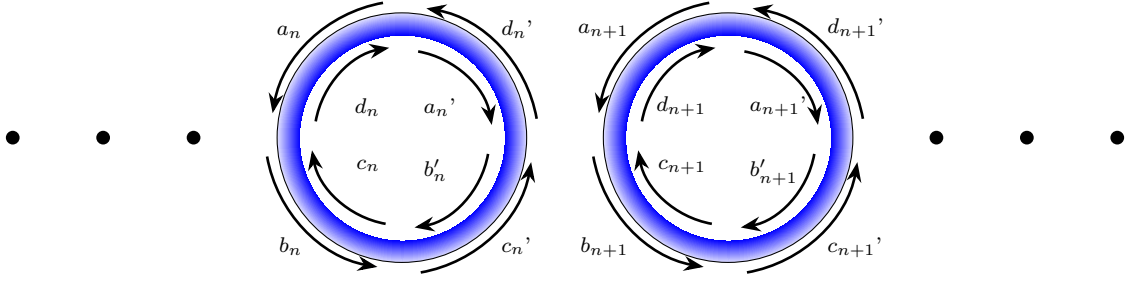


Figure 1.3 An infinitely long chain of coupled *ring resonators*, with forward and backward propagating field components. [12]

It is assumed that the coupling region is sufficiently long compared to λ , so that the light circulating in one direction in a resonator is phase matched to only one of the two other counter-propagating degenerate modes in the neighboring resonators. The coupling between two resonators can then be described by,

$$\begin{bmatrix} b'_n \\ b'_{n+1} \end{bmatrix} = \begin{bmatrix} t & \kappa \\ -\kappa^* & t^* \end{bmatrix} \begin{bmatrix} a'_n \\ a'_{n+1} \end{bmatrix}, \quad \begin{bmatrix} d'_n \\ d'_{n+1} \end{bmatrix} = \begin{bmatrix} t & \kappa \\ -\kappa^* & t^* \end{bmatrix} \begin{bmatrix} c'_n \\ c'_{n+1} \end{bmatrix} \quad (1.1)$$

where t and κ are the dimensionless transmission and coupling coefficients over the coupling length respectively. The matrix is unitary and unimodal and therefore, $|t|^2 + |\kappa|^2 = 1$. If a vector with the different field components is defined,

$$tx_n = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}_n, \quad (1.2)$$

then equation 2.1 can be rewritten as,

$$x_{n+1} = \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} x'_n \equiv \mathbb{P}x'_n \quad (1.3a)$$

$$P = \frac{1}{\kappa} \begin{bmatrix} -t & 1 \\ -1 & t^* \end{bmatrix} \quad (1.3b)$$

The propagation of the field around the ring is accompanied by a phase shift leading to consequent attenuation,

$$x_{n+1} = \begin{bmatrix} 0 & Q \\ Q & 0 \end{bmatrix} x'_n \equiv \mathbb{Q}x'_n \quad (1.4a)$$

$$Q = \begin{bmatrix} 0 & e^{-i\beta R\pi} \\ e^{i\beta R\pi} & 0 \end{bmatrix} \quad (1.4b)$$

where R is the ring radius and $\beta = n(\omega) \omega/c + i\alpha$, where $n(\omega)$ is the frequency dependent effective index and α is the loss (or gain) per unit length of the ring. Eq. 1.3 and 1.4 put together gives,

$$x_{n+1} = \mathbb{P}\mathbb{Q}x_n \quad (1.5)$$

Eq. 1.5 is a generalized equation where the matrices $\mathbb{P}$ and $\mathbb{Q}$ can be specified at each frequency to take into consideration the frequency dependence of the effective index, loss and transmission and coupling coefficients.

Motivation

One major challenge that plagues the functionality of CROW filters is the sensitivity of their resonant frequencies to thermal variations. This hinders reliable operation of the optical system of which it is a part. Much work has been carried out to tackle this weakness of CROW filters. Some approaches employ static measures that passively counteract the sensitivity by using negative thermo-optic coefficient materials or by combining with balanced modulators like the MZMs whereas others take a more active and dynamic approach where the relative thermal variations are minimized by using a control system. Chapter 2 is a literature review on these methods and discusses this in more detail. Our work has been centered on a project (see section 5.1) that aims to overcome the optical signal transmission limitations due to the non-linear Shannon Limit and non-linear effects. It relies on the concept of NFT (non-linear Fourier transform) to deliver on the objectives. In doing so, some of the signal processing load of the system is borne by the optical domain. The photonic integrated circuit used for carrying out the optical domain processing has numerous *2nd order CROW filters* and *4th order CROW filters*. The control algorithm we discuss in this text has been used to stabilize the resonant frequencies of these CROW filters. To realize this, a part of it has been tailored to suit the design of the optical chip, for example, the sequence of operation of the channels meant to control the different on-chip CROW filters. However, in essence it can be used for the control of higher order CROW filters that have integrated heaters and uses optical intensity as a process variable of the feedback-based control system.

Technical Background

Coupled-resonance optical waveguide filters must be designed such that the resonant frequency of each constituent ring resonator is matched to the center frequency of the filter. This is however met with several challenges.

Environmental factors and physical factors are two major contributors to deviations in the spectral transfer function of these filters. The control over localized systematic or random spatial variations in the waveguide dimensions due to fabrication are limited by the concurrent technologies [13]. Coupling induced frequency shift (CIFS) is yet another factor that is accountable for perturbations in the flat-top passband response of the filter. CIFS is a second order self-coupling effect that leads to frequency splitting which then decides the device bandwidth [14]. It has been reported that dynamically tuning a CROW filter can resolve the problem of shifting of the resonant frequency (f_r) [1]. However, doing so requires electrical power in the order of $\mu\text{W}/\text{GHz}^{-1}$.

Modern fabrication technologies have enabled silicon to be the material of choice for low-cost and energy efficient photonic circuits. As discussed earlier, optical ring resonators are an integral component of most such circuits. However, optical ring resonators made in silicon have high wavelength selectivity which is especially susceptible to thermal fluctuations due to the high thermo-optic coefficient of silicon ($1.86 \times 10^{-4}\text{K}^{-1}$). The relation of a ring resonator's resonant wavelength to temperature is given by

$$\frac{d\lambda}{dT} = \left(n_{eff}\alpha_{sub} + \frac{\partial n_{eff}}{\partial T} \right) \frac{\lambda_0}{n_g} \quad (2.1)$$

where λ_0 is the resonant frequency, n_{eff} is the effective refractive index, α_{sub} is the substrate expansion coefficient and n_g is the group refractive index. [4, 15]. Since the optical mode in the device is tightly confined in the waveguides and the thermo-optic coefficient of SiO_2 is lower by an entire order of magnitude ($1.0 \times 10^{-5}\text{K}^{-1}$), it is sufficient to ignore the wavelength dependence of the terms in eq. 2.1. We may also ignore the α_{sub} term since for a silicon substrate it is ($2.6 \times 10^{-6}\text{K}^{-1}$), which is two orders of magnitude lower than the thermo-optic coefficient of silicon [16]. The reduced form of eq. 2.1 can then be rewritten as

$$\frac{d\lambda}{dT} = \left(\frac{\partial n_{eff}}{\partial T} \right) \frac{\lambda_0}{n_g} \quad (2.2)$$

The extent to which the functionality of the ring resonator is affected is dependent on the Q-factor. Nonetheless, temperature deviations of greater than 1 K is enough to take it out of resonance [6]. The stable range of $\pm 1\text{K}$ is too minuscule when compared to the temperature

deviations in operative environments of silicon photonic circuitry. It is important to take note that the absolute temperature is irrelevant here. Instead, it is the relative temperature fluctuations of the ring resonator during active operation that poses the greatest threat to its functionality.

The measures for the mitigation of this dependence can be broadly classified into (1) methods that focus on reducing the susceptibility of the *ring resonator* to temperature variations ("athermal" solutions) and (2) methods that actively compensate the variations and keeps the local temperature of the ring resonator constant ("control-based" solutions) [17]. The advantage of the athermal approach is that it does not require auxiliary photonic structures such as photodetectors or optical couplers. In addition to that, it needs no extra power and thus is a low-energy solution. The downside is tighter fabrication tolerances or compatibility of the materials used for this approach. On the other hand, the control-based solutions are relatively independent of fabrication and material limitations. Some of the solutions may require the integration of additional photonic devices and actuators (heaters) with the *ring resonators*. The main disadvantage of this approach is active energy consumption.

2.1 Athermal methods

Athermal methods employ novel techniques to reduce the effect of temperature changes to the optical devices. Even though the devices do not completely become insensitive to the thermal fluctuations, they become impervious enough to prevent an impact on their performance. There are two ways of *athermalizing* the devices, (1) either by using materials in waveguide claddings that have negative thermo-optic coefficient or (2) by integrating the ring resonator in a thermally balanced interferometer. *Trimming* is another popular method but we shall not be discussing it here.

Athermalization by Negative thermo-optic coefficient materials

In this method, the waveguides which are intended to be athermalized are put within a cladding of a material with a negative thermo-optic coefficient [18]. The result of this technique is a waveguide that has an overall thermo-optic coefficient equivalent to zero. The thermo-optic coefficient is then defined by the equation

$$\frac{\partial n_{eff}}{\partial T} = \Gamma_{core} \frac{\partial n_{core}}{\partial T} + \Gamma_{cladding} \frac{\partial n_{cladding}}{\partial T} + \Gamma_{substrate} \frac{\partial n_{substrate}}{\partial T} \quad (2.3)$$

where Γ is the modal confinement factors for the core, cladding and substrate respectively [19]. The fundamental objective is to engineer the modal confinement factor of the cladding and choose its material such that the thermo-optic coefficient when put together, can negate the corresponding parametric terms of the core and the substrate as shown in eq. 2.3. It is worthwhile to note that the term for the core can be significantly large when silicon is used. As a consequence, completely nullifying it with the cladding term is not possible in most applications, albeit it is not always required to be zero. In addition to these considerations, the propagating mode also requires to be reconfigured so that a good part of it is distributed in the cladding. One popular approach for doing so is reducing the width of the waveguide or by using a slotted structure [4, 16, 20]. The TDWS (temperature dependent wavelength shift) could be reduced

to $< 5\text{pm/K}$ over a range of 50 K and more recently to 0.2pm/K [21]. These methods require stringent fabrication tolerances and are often difficult to implement. Some work has been done to address this problem by utilizing photosensitive materials such as chalcogenide glasses. They allow the dimensions of the waveguides to be trimmed post-fabrication. In recent years, TiO_2 (titanium dioxide) has proven to be a reliable negative *thermo-optic coefficient* material. It is CMOS compatible with a negative *thermo-optic coefficient* of $(-1.8 \times 10^{-4}\text{K}^{-1})$ [22], which is about the same as that of silicon. With TiO_2 , a TDWS of $< 2\text{pm/K}$ over a temperature range of 5 K could be achieved. These devices are also capable of current injection induced resonance shifts which enable them to operate at high frequencies [17]. However, many hindrances still plague this methodology since the polymers used are susceptible to degradation due to large temperature swings, chemical corrosion, UV aging and are mechanically not very robust. It must also be noted that the modifications to the mode to reduce modal confinement will also have an impact on the ring resonator's straight and bent configurations, consequently impacting Q-factor and footprint [19].

Athermalization by interferometric structures

Yet another way of making the *ring resonator* insensitive to thermal variations is by integrating it with interferometric structures such as the MZM (Mach Zehnder Modulator). The balancing of the two arms of an MZM can be carried out in a manner that makes it insensitive to temperature changes as a whole. As a consequence, the guided modes in the two arms experience different effective *thermo-optic coefficients*. It has been shown that this can be attained by changing the width of the waveguides thereby altering the guided mode [5, 23]. This concept can be adopted for athermalizing *ring resonators*, wherein the *ring resonator* is integrated into one of the arms of the MZM [24]. The thermal sensitivity of the MZM and the *ring resonator* are able to compensate each other and produce a net zero sensitivity to temperature variations.

Both the methods for passive athermalization discussed so far come with their fair share of advantages and disadvantages. The first method requires no additional footprint and allows modulation at higher frequencies. On the other hand, it has tighter fabrication tolerances and requires additional layers. The second method's strength lies in its monolithic construct and CMOS compatible fabrication requirements however it loses out on maintaining a smaller footprint.

2.2 Control-based methods

Control-based methods focus on maintaining the local temperature of the *ring resonator* to ensure the stability of the *ring resonator's* resonant frequency. It uses external electronic circuitry to directly or indirectly detect and then compensate the local temperature. This means that this procedure requires additional power and must have actuators (heaters) integrated into the *ring resonators*. It is a dependable option for applications where the footprint of the photonic structures must be kept small and the costs of making them insensitive to temperature must be kept low. It also brings with it the advantage of readjusting the resonant frequency, consequently shifting the passband to a different region of the spectrum.

The temperature is generally controlled by using heaters which increase the temperature of the *ring resonators*. Despite relying on unidirectional actuation (meaning that to keep the *ring*

resonator at its resonant frequency, we can only heat it to move it to higher wavelengths/lower frequencies) it can have holistic control of the resonant frequency by exploiting its periodic nature.

The system required for implementing this method can be de-constructed into three parts, (1) the monitoring system that detects the *process variable*, (2) the *compensator*, which takes the process variable and generates the actuator output signal to compensate the error and (3) the actuator or heater that affects the change in temperature. The monitoring can be carried out directly by using on-chip thermocouples or indirectly by selecting an appropriate indicator of the thermal variations (like drop-port intensity, through-port intensity, BER, etc.) and using photodetectors or similar transducers. These devices are readily available and they reproduce faithful signal which can be reliably used by the control systems. The remaining two components are not standardized and therefore worth taking a closer look at.

Integrated heaters

The ability to manipulate the local temperature of a photonic structure is provided by integrated heaters. Integrated heaters are resistive structures that are usually made of nichrome, titanium or doped silicon. They are placed in close proximity to the ring resonators. Passing a current through these structures heats them up which in turn heats up the local region and allows minute control of the local temperature. In some unique cases, such as for *ring resonators* made out of carrier-injection materials, the same effect can be achieved by controlling the bias current at the diode junction [6, 25, 26]. The figures of merit of integrated heaters are their *tuning efficiency* represented as (mW/nm) or (mW/GHz) and *tuning speed* which is usually in the range of (μ s). It is often more convenient to express the tuning efficiency as the amount of power required to move the resonant frequency by one FSR, i.e., as (mW/FSR) or alternatively as (mW/ 2π). The overall metric of tuning efficiency for different *ring resonator* is not skewed by this representation since an increase in the diameter of the resonator corresponds to a decrease in its FSR [27], where FSR is given by $\frac{c}{n_g L_r}$ where c is speed of light in vacuum, n_g is the group index and L_r is the circumference of the *ring resonator*. In conventional layouts, the heaters are positioned on top of the waveguides but they are not allowed to come directly in contact so that the waveguide modes remains intact. A layer of SiO₂ between the two layers is required to protect the optical mode and reduce scattering losses. Some tuning efficiency is sacrificed since a part of the heat flux is dissipated in the surrounding insulation layer. Such a configuration allows a tuning efficiency of ~ 42 mW/FSR at best and a tuning speed of 14 μ s [28]. A workaround for improving the tuning efficiency is by using disk resonators which are similar in functionality and construct to ring resonators. The heaters can be fabricated into the interior of the disk while the modes propagate in the periphery. The drawback is that the FSR is impacted due to the presence of higher modes. Some work has been done in development of whispering gallery mode resonators which overcome the aforementioned problem and are able to achieve tuning efficiency of 20 mW/FSR tuning speeds of ~ 1 μ s [29]. An alternative to using these strategies is to use air trenches around and under the heater to minimize the heat dissipation into the surrounding insulation and substrate [15, 30, 31]. In spite of large improvements in tuning efficiency and time, these methods increase the thermal impedance of the *ring resonator*.

Solutions for Control-based methods

The control-based method requires an electronic circuitry which acts as a closed-loop feedback controller with the capability to read analog signals, process the input and generate analog signals. The electronic circuitry can be de-constructed into three main parts, (1) the monitoring system, (2) the control algorithm and (3) the actuator drive system. In most cases, the control algorithm is likely to be based on the widely used PID methodology, which is simple, robust and reliable [6]. It is worthwhile to note that there is no single method so far that simultaneously meets all the requirements such as accuracy, response time and speed, stability, costs, energy-efficiency and robustness. Direct monitoring of the temperature-based resonant frequency drifts of *ring resonator* has been demonstrated by using a p-n diode based temperature sensor [32]. Other methods have used an indirect approach by measuring indicators such as infrared imaging, intensity of the signal and its BER (bit-error-rate) as well.

Before the control system enters the state where it is constantly monitoring and maintaining the local temperature of the ring resonator, it must be configured to ensure the actuated ring resonators are aligned to the correct laser wavelengths. This process of initialization of the compensator is known as *wavelength locking*.

It has been shown that monitoring the optical power at the through port and setting the ring resonator to get minimum power transmission is sufficient for “wavelength locking” and stabilization [33]. Monitoring the optical signal at the drop port of the *ring resonator* is an alternative to using the through port, which is especially helpful in cases where losing power at the through port due to a tap has an impact on the power budget. Since this method used the mean modulated power, low-speed electronics can be used for the control and low-speed photodiodes for monitoring the optical power.

The bit-error-rate of the signal is an alternate means of monitoring the TDWS. In this implementation, the BER is directly monitored and optimization of the same is carried out by tuning the *ring resonators* [7]. The downside of this method is the requirement of costlier, high-speed electronics and more complex algorithms.

The main challenge of controlling the resonant frequency of *ring resonators* lies in the fact that their resonant spectrum is highly symmetric. It presents an ambiguity to the compensator which cannot easily determine if it should increase or decrease the actuator signal. Use of interferometric structures and homodynes have proven to be effective ways of obtaining the anti-symmetric error signal [34]. However, these methods are dated and better and simpler alternatives are now available. A simple yet effective approach is to use a dithering signal to break the symmetry and generate an anti-symmetric error signal [35]. In this method, a small modulation is applied to the heater of the *ring resonator* to obtain a small modulation in the optical signal. The generated optical signal is either out-of-phase or in-phase to the actuating signal, depending on which side the laser is with respect to the axis of symmetry of the resonance. By mixing the actuator signal and the modulated signal, information about the side can be obtained as shown below,

$$\cos(f_d t) \otimes \cos(f_d t + \phi) = \frac{1}{2} [\cos(2f_d t + \phi) + \cos(\phi)] \quad (2.4)$$

where f_d is the dithering frequency and ϕ is relative phase of the modulated optical signal. If the higher harmonic is filtered, we are left with the DC term. The polarity of the DC term is then used to determine the position of the resonance with respect to the laser. This method

is simple, uses low-speed circuitry and is energy efficient. Besides the photodiode required for monitoring, it does not require any extra photonic structures [17, 35].

Concept and Architecture

3.1 Concept

The stabilization of the resonant frequencies is a two step process. First, the CROW must be positioned to the desired laser wavelength. This step is known as *wavelength-locking*. The second step is where the algorithm constantly tracks the laser wavelength. This step is therefore called *tracking*.

Our application is based on using optical intensity at the drop-port of the CROW filters. Timudrogan et. al. [33] has experimentally demonstrated the optical wavelength recovery of ring resonators that is capable of compensating both fabrication and thermally induced resonant frequency variations in a silicon photonic chip. They used an algorithm that carried out continuous search of the minima of the through port or the maxima of the drop port intensity by using coarse and fine loops. We have discussed in section 2.2 about the practicality of this method. The control objective is thus maintaining maximum intensity at the drop port and ensuring that it is a consequence of being aligned to the correct laser wavelength. Since the maximum intensity is used, the control system does not require a set-point. Instead, it always finds the maximum value and uses it to readjust the heaters. The problem for *wavelength-locking* lies herein that in an integrated photonic chip, the waveguide assigned to the drop port of one filter could also be sharing bandwidth with other channels or laser wavelengths. Therefore, the intensity at the drop port could be the consequence of another laser wavelength but the control algorithm would have no means to identify if it is optimizing the correct channel. In such a case, it is necessary to keep two factors in mind for designing the algorithm. The first factor being, (1) the algorithm has to start from a roughly pre-aligned state where it has a fair margin before it is affected by optical intensity crosstalk from another channel and (2) the sequence in which it actuates each channel so that only the correct CROW filter is actuated ensuring the variation in intensity is a direct consequence of the actuation. For tackling the first point, we use *look-up tables*, which will be discussed in section 3.2.2. The second point is dealt with programmatically.

The *wavelength-locking* is non-trivial, since in most applications, there are more than one channel sharing the same bus waveguide. In the elementary case of only one wavelength being dropped to the monitored waveguide, wavelength locking can be achieved by simply conducting sequential, large-range current (or voltage¹) sweeps of each heater assigned to one CROW. If sufficient power is being dropped, then even if none of the *ring resonators* of the CROW are

¹Later in the text, we refer to current sweeps as voltage sweeps since the heater resistances are constant and the electronics used for actuation handles the control signal as a voltage.

by design close to the designated resonant wavelength, some variation in optical intensity at the drop port is detected when one of the rings undergoes a large-range sweep. The current at which the variation occurs can then be applied statically to that ring and the sweep of the next ring can ensue. As more rings of the CROW go through this process, more intensity is detected since every step brings the filter closer to the resonant frequency. However, the true maxima of the intensity is not reached since the large sweeps also mean higher thermal crosstalk between the rings. To remedy this, the process can be repeated, albeit with a reduced sweep range. This reduces the thermal instabilities and the rings are able to tune closer to the designated wavelength.

Wavelength Locking

In fig. 3.1, we can see a representation of the optical intensity profile obtained from carrying out a large-range current sweep. A current sweep that ranges from zero to the *first FSR current* has been referred to as *large-range sweep* in this text². The optical intensity profile when mapped with the actuating current values, can provide the information of how much current is required to drive the respective ring to the laser wavelength. In our algorithm, we carry out sweeps more than once while reducing the step size so that the locking is more precise.

Tracking

There are several strategies for wavelength tracking. In section 2.2, we have discussed about the requirement to break the symmetry of the intensity profile to determine the direction in which the ring must be driven. One of the simpler methods achieves this by applying a dithering signal to the heater [35]. We have adopted the idea of dithering and used the optical intensity response to feed the algorithm a process variable. However, how we use the input deviates from the cited method. In order to keep the algorithm simple and to have a wider application base, we planned to fine-tune the sweeps to meet the needs of tracking of wavelengths. The tracking therefore was planned to be based on a dynamic sweep function which can intelligently adapt its range and the step-size. As shown in fig. 3.1, the slope around the top of the curve can be treated as quasi-flat within a very small region around the maximum point of the profile. After *wavelength-locking*, the algorithm would need to carryout small sweeps in this range and constantly readjust the heaters to the new maxima. These sweeps have step sizes in the order of micro-amperes and hence we call them *micro-sweeps*. With the appropriate hardware, this can be implemented with ease. For our experiments, we used step-sizes of 0.5 μA . In the event of a large disturbance, the shift in resonance will cause the micro-sweep to induce larger variations in the optical intensity due to the quasi-Gaussian nature of the intensity profile. This criterion can be used to adapt the step-size of the sweep to speed-up the recovery. The advantage of this method is that it invalidates the control challenge introduced by the symmetry of the optical intensity profile. If the ring's resonant wavelength falls below the designated wavelength, each step of the dither will detect a ramp-like optical intensity profile with a positive slope and correct to the highest intensity point. In the contrasting case, the detected intensity will have

²This range for our experiments was 0 mA to 6.5 mA for the heaters of the 2nd order CROW filter with a track resistance of 1.6 k Ω and 0 mA to 8 mA for the heaters of the 4th order CROW filter which had a track resistance of 1 k Ω .

a negative slope ramp and again the algorithm will correct to the highest point. Recursions of these steps are capable of quickly localizing the resonant frequency back to the laser wavelength.

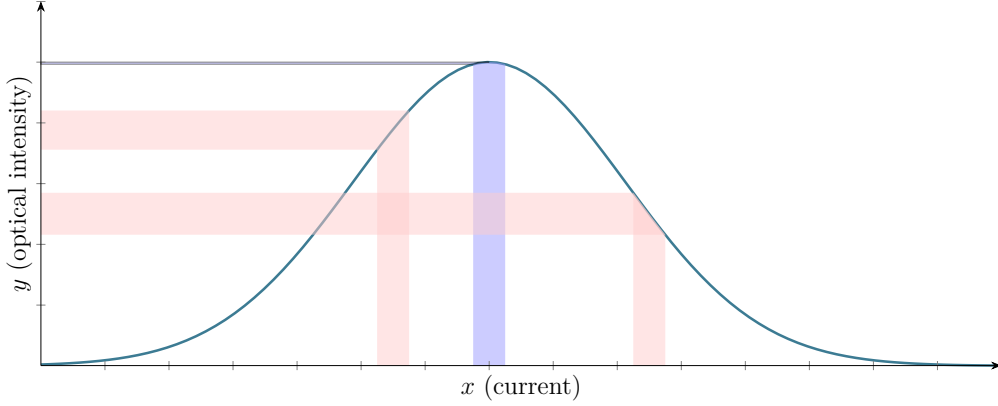


Figure 3.1 The quasi-Gaussian profile of the optical intensity obtained during a current sweep is shown here. The same range of current change induces different shifts in optical intensity depending on how close the CROW is to its resonant frequency. When dithering of current is carried out at the maxima, the disturbance to the optical intensity is minimum.

3.2 Architecture

The architecture of the control program is based on programming principles of LabVIEW. Therefore, before we delve into the main discussion about the structure of the program, we will take a brief look at LabVIEW terminologies and design patterns.

In LabVIEW, a program is known as *Virtual Instrument (VI)* and it is written with the help of objects known as *terminals*, *subVIs*, *functions*, *constants*, *structures*, and *wires*. These objects are placed in an environment known as the *Block Diagram*. The *terminals* are the input and output nodes of the program. The data taken from the input terminals are passed to the *functions* and the functions pass the processed result to the output terminals. It is also possible to pass fixed values to the functions and this is done by using *constants*. *Structures* are environments within the block diagram that allow a section of the code to be put under special conditions such as while, if-else, sequential execution, etc. *SubVIs* can be considered as user-made functions or code-blocks within the main VI. LabVIEW programs are usually made by following some well-known architectures known as design patterns. A program can employ one or more of these design patterns depending on the purpose and use cases of the application.

With this knowledge in mind, we will now discuss the architecture of our control algorithm.

3.2.1 Abstraction of the algorithm

As we already know, the main objective of the algorithm is to ensure the resonant frequencies of the *ring resonators* in a CROW filter remain stable throughout the duration of operation of the filter. In doing so, we also take into consideration (1) the modularity, (2) responsiveness, (3) stability and (4) ease of configuration for wider applications. The design patterns that

seemed most suitable for these requirements were the Producer/Consumer model and the State-machine model. The former is used to optimize the data exchanges and the latter is used by the algorithm to respond according to different conditions that arise during the alignment of the filter.

Producer/Consumer design pattern

The Producer/Consumer model is a program architecture used for multiprocess algorithms. [36, 37] This architecture streamlines the data by decoupling the processes that produce and consume data at different rates. It uses parallel loops which can be classified into (1) those which generate data and (2) those that process the produced data. This allows several sets of data to be acquired and then be processed in sequence. Since our application employs a control board as an external hardware and a computer for running the algorithm, it is inherently asynchronous due to their different operating speeds. It is possible to devise a program that uses a single loop to acquire the data and then process it but this would cap the operation speed to the sum of the execution time of the two processes leading to inefficient utilization of system resources and reduced overall speed (in our case, reaction time of the control algorithm). The producer/consumer design allows the data to be queued up in a loop while the processing can be carried out at its own pace in a separate loop. This is fundamentally more efficient since data acquisition can be carried out much faster than it can be processed. The shortcoming of this method is that it requires more resources, especially memory, since the buffers need to be sufficiently large to accommodate the data queue. However this is remedied by choosing LabVIEW as our programming platform, which has an in-built function that can ensure data-loss does not occur due to buffer overflow. In common practice when the buffer is full, the producer loop is either put to sleep or it discards data. In LabVIEW, if consumption is slower than production, the queue will eventually become full and the producer code will be forced to wait until the consumer has dequeued an element before a new element can be queued up.

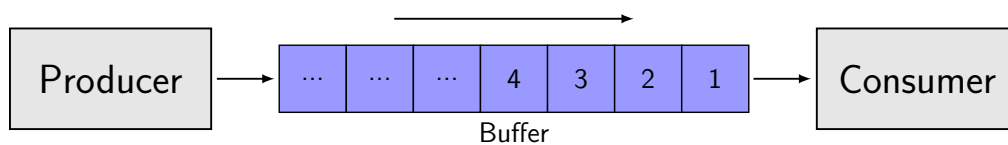


Figure 3.2 The data from the hardware is collected by the producer and is enqueued within it. The consumer takes data from the queue and processes it.

This architecture suited our requirements since the algorithm needs a constant stream of data which it processes to generate the response. The stream of data was obtained from the control board with a section of the program that acted as the producer. The algorithm itself acted as the consumer in our program. In our application, the buffer will never fill up since the producer is synchronized to the hardware and the consumer is synchronized with the producer (discussed in section 4). Even though this reduces the speed of the program to the speed of the slower of the two processes, it is still faster in comparison to the case where the speed of the program is the sum of the two processes.

State-machine design pattern

The state-machine architecture is used in applications where distinguishable states exist. It changes its own state to conform to the application's state based on user-input or internal decision based calculations. Each state does something unique and can call other states. This design pattern is a good fit for our application because it conforms with modularized architectures. The different states can call on the same sub-routines without encountering conflicts since at any point of time, only one state of the program is under operation. Since the alignment of the filters involved different stages, we could classify them as states and introduce conditions based on which the program could change state automatically or through user-interaction.

Modules of the program

The program has been broken down into functional modules to enable flexibility for different applications and more importantly, to keep the program streamlined. It also makes efficient use of the state machine design pattern as the same modules can be called by different states. Each of these modules are in fact subVIs of the main VI. The modules can be categorized as:-

1. **Interface module:** This module is used as the *producer* in the program. It collects the monitored intensity data from the electronic hardware and enqueues them for the algorithm module. It also handles the voltage values generated by the algorithm module and enqueues them to send them over to the electronic hardware for applying the control signals to the actuators. In our case, the intensity data is the photodiode signal at the drop port of the OADMs and the actuators are the heaters of the OADMs. We will refer to it as the *Interface VI*. The nomenclature *interface* is not to be confused with *user-interface*.
2. **Algorithm module:** This is the module that houses the state machine. It has different states or modes that call on the sweep module with the appropriate settings. We will refer to it as the *Algorithm VI*.
3. **Handover module:** This module is required for communicating between the Algorithm module and the main VI. It also conditions the intensity data with a digital filter before passing them to the Algorithm module. This module along with the Algorithm module act as the *consumer*. We will refer to it as *Handover VI*.
4. **Sweep module:** The sweep module is the core of the program. It is used to generate the various voltage sweep values and map the intensity response to the voltage steps. It then processes this data to pass the optimum set point to the Algorithm module. We will refer to it as *Main Sweep VI*.

3.2.2 The structure and behavior of the program

Hierarchy and execution sequence

The program can be broken down into three levels of hierarchy, (1) the top-most level, (2) Level-1 and (3) level-2. The main program is run via the main VI, which is the top-most level of the program. The *Interface VI*, *Handover VI* and the *Algorithm VI* are all in *level-1* and

are parallel to each other. The *Main Sweep VI* is nested in the *Algorithm VI* and therefore at *Level-2*. The global variables are inter-level operators and therefore not classified into any level. Figure 3.3 shows graphically how the different VIs are hierarchically related. All Level-1 VIs run as soon as the main program is launched. However, the *Main Sweep VI* at Level-2 is called only when the *Algorithm VI* calls it. The *Algorithm VI* is configured to call the *Main Sweep VI* as many times as the number of rings in the filter for which it is being used. During a call, the execution of the *Algorithm VI* is paused until execution focus is returned back to it after the conclusion of the sweep, as shown in figure 3.4.

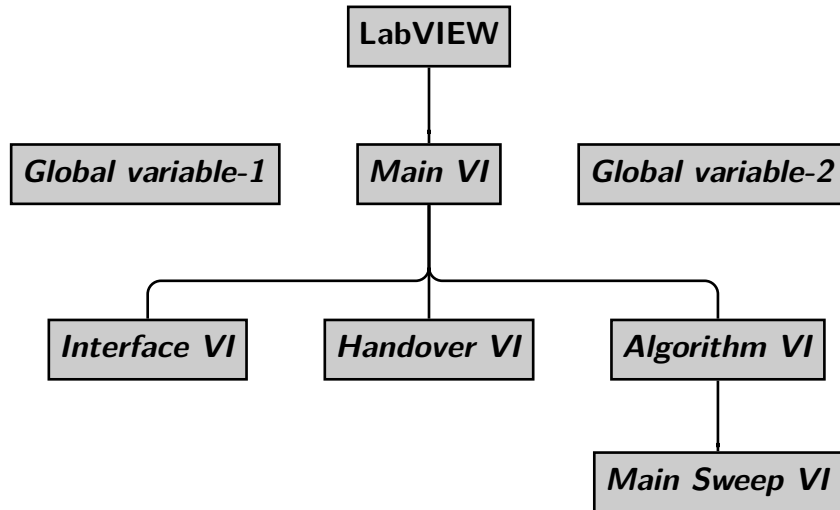


Figure 3.3 The hierarchy of the different modules/VIs. The *Main VI* is what the user interacts with. The subVIs *Interface VI*, *Handover VI* and *Algorithm VI* are nested within it. The *Main Sweep VI* is nested within the *Algorithm VI*. The global variables are accessible from any level of the hierarchy and thus not indicated as a branch in the hierarchy tree.

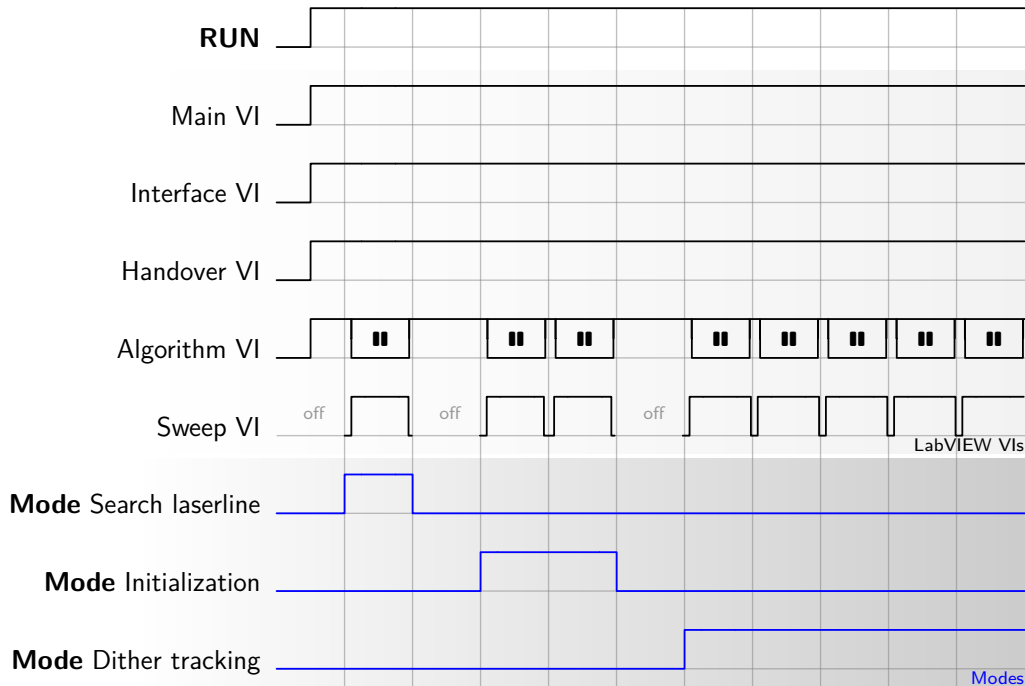


Figure 3.4 This diagram is for understanding how and when the different VIs run. It is based on general timing diagrams. A high state here represents the running VIs. All the VIs (and subVIs) run in parallel except the *Main Sweep VI*, which is called every time the algorithm is called or *launched*. For the sake of simplicity, the diagram is used to illustrate the case when the program is used for a 1st order OADM. If a mode is kept enabled for more than one cycle, the *Algorithm VI* and *Main Sweep VI* will repeat their cycles until the mode is switched off. All the modes use the *Main Sweep VI* but with different settings.

Data flows in the program

In LabVIEW, it is not possible to nest loops³ and expect the child loops and the parent loop to run simultaneously. Execution is transferred to the innermost loop and once it finishes its iterations, the loop that encloses it continues with its execution. It is therefore impossible to run different tasks simultaneously with separate for-loops nested in parallel inside a while-loop. The for-loops will not run simultaneously but rather LabVIEW will automatically assign priorities to the parallel for-loops that are placed inside the parent while-loop and then execute them in that sequence. Due to this reason, the program had to use three non-nested while-loops in parallel to each other, in the main VI. One loop was used to run the main VI and the *Interface VI*, another for the *Handover VI* and the last one for the *Algorithm VI*. These modules or subVIs are dependent on exchange of data between them. However, data cannot be transferred to a parallel while-loop via a *wire*⁴ in the block diagram. This is where we had to introduce *global variables*. Global variables are variables that are used to access and pass data among several VIs that run simultaneously. We used two global variables, to which we will refer to as *Global variable-1* and *Global variable-2*. *Global variable-1* is used for communication between

³while loop or for loop in LabVIEW

⁴a tool to transfer data among block diagram objects.

Handover VI and *Algorithm VI*. *Global variable-2* is used for communication between *Handover VI* and *Main Sweep VI*. The predominant reason for introducing two separate global variables is because they are intended for data transfer between two different layers of the program hierarchy.

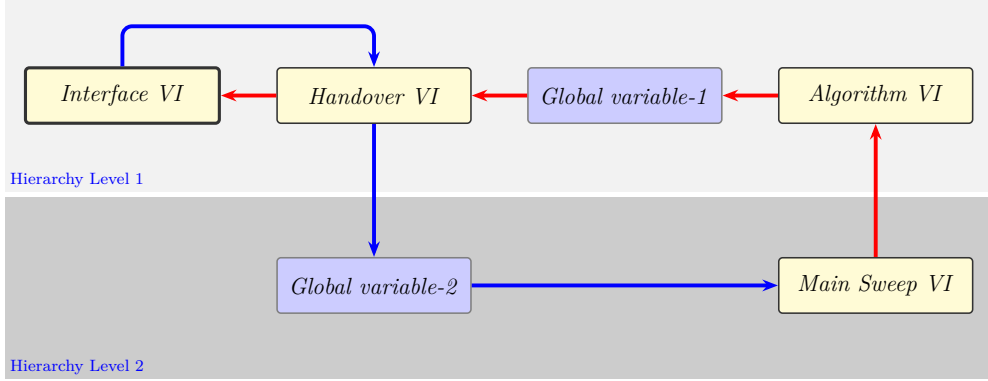


Figure 3.5 Data flow between the various subVIs and global variables

As shown in fig. 3.5, the interface VI is the source of the intensity data and sink of the voltage value data. The data flow path of the intensity values data have been represented with blue arrows and the same for the voltage values have been represented with red arrows. The intensity values are streamed-in from the electronic hardware by the *Interface VI* and then passed to the *Handover VI*. The function of the *Handover VI* is to use a digital filter to reduce noise in the intensity data and to handle the *look-up table*, which we will discuss at a later stage. The intensity data is then transferred from this VI to the *Main Sweep VI* with the help of *Global variable-2*. The *Main Sweep VI* takes this data when it is called and then as soon as the sweep starts, it records each intensity value as a function of the heater voltage values it generated. When the sweep concludes, the maximum intensity in this data is determined along with the corresponding heater voltage. This voltage, henceforth referred to as *optimum voltage*, is passed to the *Algorithm VI*. The *Algorithm VI* stores the optimum voltages for all rings of the target CROW filter in an array which we refer to as the *look-up table*. Once the *Algorithm VI* has conducted sweeps and extracted all the optimum voltages for the target CROW filter, it passes the *look-up table* to the *Handover VI* via *Global variable-1*. It is worth noting here, that as the *Algorithm VI* iterates through the different rings of the CROW filter, it also ensures that the optimum voltage of the last swept ring is sent to the *Handover VI* and consequently the corresponding heater before the sweep transitions to the next ring. This is done with the help of a subsidiary of the *look-up table* that passes the values via *Global variable-1*. In fact, the *look-up table* can be considered an *object* that is passed constantly between the *Handover VI* and the *Algorithm VI* to ensure the appropriate voltages are being applied to the heaters of the concerned filter at all times. The *Handover VI* passes the *look-up table* values to the *Interface VI* which converts the values to a data-stream and sends it to the actuators via the electronic hardware. For applications requiring the control of multiple optical filters, the *Handover VI*, *Algorithm VI*, *Main Sweep VI* and global variables can be replicated to work separately for each channel. The *Interface VI* remains the same except some modifications that allow it to read and write multiple channel data to the electronic hardware.

Decision flows and states of the program

The *Algorithm VI* contains the different states of the algorithm. There are three states that are required for the *wavelength-locking* and tracking of the resonant frequency and there is an additional optional state which stops actuation when a preset intensity is obtained. The first state, called the *Search Laserline* mode, is used to roughly determine the voltages required to tune the filter. In this mode, the sweep is configured to use the full FSR voltage as the upper limit of the range and zero volt as the lower limit. The resolution of the sweep is also maintained low since we do not require precise voltage values at this stage. The goal of this mode is to simply detect even the slightest variation in monitored optical intensity by “shaking” the rings around. After each ring of the CROW filter is put through a sweep, the optimum value is applied to it and the next ring is swept. Once it is determined that approximately the same voltage values are being detected for consecutive cycles, the algorithm moves to the next case which is called the *Initialization*. This mode is used to carry-out the *wavelength-locking*. The sweep is configured to a range which is a certain fraction of the full range and is centered around the voltage obtained in *Search Laserline* mode. This range is made to compress itself over consecutive passes until it reaches a user-defined minimum range called *delta*. The consecutive reduction of sweep range reduces the voltage step size (since resolution of the sweep is kept constant) and thus thermal disturbances are also reduced, allowing a more precise optimum voltage value to be identified. When the sweep range reaches *delta*, the algorithm switches to the *Dither-tracking* mode. In this mode, the range remains *delta* but the resolution of the sweep is reduced to have only enough steps such that the sweep can detect a variation in intensity that is only just greater than the noise levels. In our experiment, a resolution of ten steps sufficed. As depicted in fig. 3.1, this mode is used to dither the ring resonator near the optima so that only a very small variation occurs in the monitored intensity. It is in fact, the micro-sweep that we discussed in section 3.1, under the topic of *Tracking*. During the dithering, three cases are possible. (1) When the optimum voltage is within the dithering range, (2) when the optimum voltage is greater than the upper limit of the dithering range and (3) when the optimum voltage is lesser than the lower limit of the dithering range. In the first case, the dithering is simply a miniature version of *Initialization*. In the second and third case, it is realignment to that extrema of the optical intensity’s ramp-like profile which has the highest intensity. With sufficient recursions of the micro-sweep, it returns to the first case. The Idle mode is used when the continuous intensity variations due to the dithering is not desirable. In such a case, an intensity set-point is entered by the user and the algorithm stops dithering when the intensity is above or equal to it.

To summarize the decision flow, we refer to figure 3.6. As shown in the figure, the Search Laserline mode carries out fast, coarse voltage sweeps to find any variation in intensity and sets the ring to the corresponding voltage value. Then the Initialization mode takes over and conducts voltage sweeps over a smaller range and keeps reducing the range by a given factor until it reaches the minimum allowed range. Next, the Dither Tracking mode carries out micro-sweeps until the intensity sees no further increase. A value slightly less than this intensity can be set manually or automatically as the desired intensity for that filter. If the intensity remains greater than the desired intensity for a certain period of time, the algorithm moves to Idle mode and keeps monitoring the intensity while holding the heater voltages. If the intensity falls below the desired value, it returns to Dither Tracking. In our experiments, we chose not to allow the dithering to go into Idle mode and carried out continuous tracking by dithering.

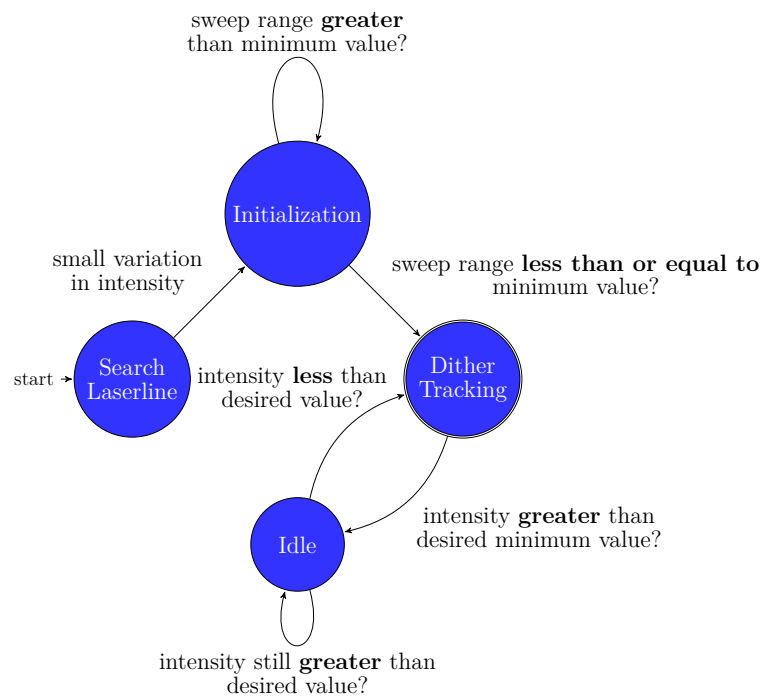


Figure 3.6 State machine diagram of the algorithm.

Challenges and Solutions

The main challenges faced during the implementation of the project can be primarily categorized as those related to engineering. At the very first stage of the implementation, we were faced with a (1) *lag or delay* problem that introduced an offset between the intensity read-back data from the control board and the value of the corresponding current applied to the heaters of the CROWs. The second most significant challenge was the swing in the global chip temperature during wavelength locking and tracking. Once these two variables were eliminated, the LabVIEW program was analyzed and it was found out that there were (2) race-conditions of the data within the program.

Delay

During the 2nd order CROW alignment, we noticed a discrepancy between the recored intensity values and the corresponding voltage values. This occurred when the algorithm was made to run at its highest rate. The bizarre problem of this read-back offset was especially complex to pinpoint since it could have multiple contributors.

The issue had the footprint of asynchronicity between the intensity values getting reported to the system and the voltage values being sent out. In reference to figure 4.1, if we consider the control algorithm to be the starting node of the data flow, (1) the voltage values are sent from the algorithm (LabVIEW) to the microcontroller via USB. (2) The microcontroller receives this data and writes it to the DAC. (3) The DAC then applies the voltages to the CROW filter ring heaters through the respective channels. (4) the CROW filter responds to the actuation and the (5) intensity detected at the drop-port photodiodes changes. The photodiode signal is converted to a voltage signal which is (6) read by the ADCs. The ADCs send the values back to the (7) microcontroller which forwards the values to (8) LabVIEW via USB. This is one cycle of the process. The asynchronicity was speculated to be stemming in between step 1 to 4 or step 4 to 6 or step 6 to 8.

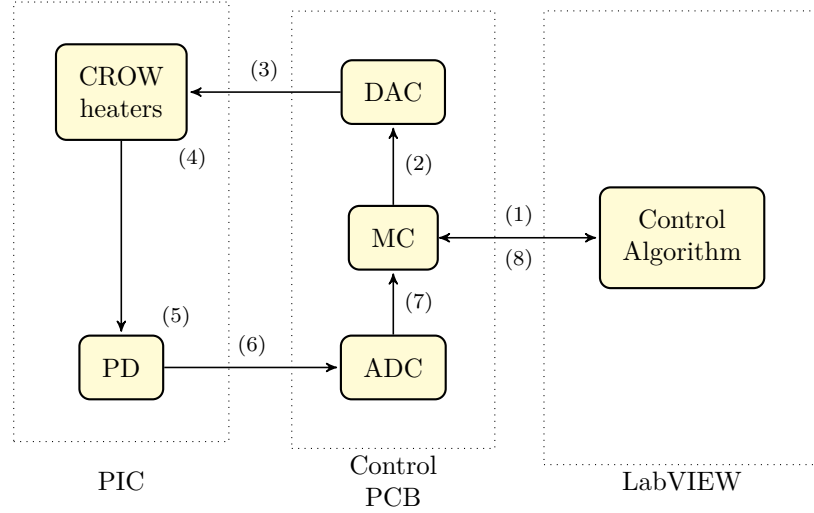


Figure 4.1 Schematic of system

From the software end, the LabVIEW program had independent loops that carried out parallel execution of the respective program, as described in section 3.2.2. The *Interface VI* and the *Main Sweep VI* were found to be running at different speeds. In other words, each the *loop time* of these VIs were different. Therefore, assuming both loops started running at $t=0$, when the interface VI carried out one cycle of read and write with the microcontroller, the *Main Sweep VI* would have run through more than one iteration, leading to the mismatch of intensity versus voltage data. Also, when the algorithm switched over from one ring to another, the new sweep data would often contain a part that was a residue of the previous sweep. Before all the intensity data from the last sweep could be read, the algorithm switched to the next ring and as a consequence, the first few voltage steps were mapped to the intensity variations due to the last few steps of the previous sweep.

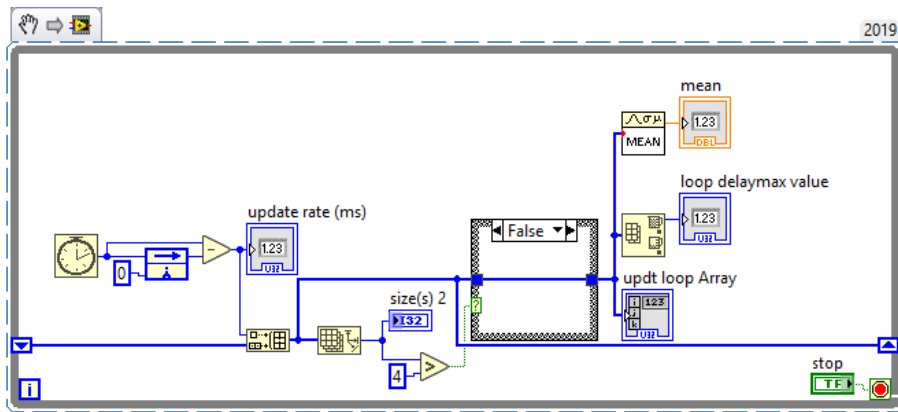


Figure 4.2 Loop timing measurement within each while loop of the VI.

The asynchronicity between these two loops had to be eliminated, ensuring they worked in tandem. We implemented a signaling method to make the *Main Sweep VI* wait for a "ready" acknowledgment flag from the *Interface VI*. We initialized a boolean variable to FALSE and

passed it to the interface VI where it was modified to TRUE and then passed it to the *Main Sweep VI*. This flag was then used to stop an empty while loop inside the *Main Sweep VI*. Before the main sweep begins, execution is transferred to the empty while loop until a flag from the *Interface VI* is received. The flag terminates the while loop and execution is returned to the *Main Sweep VI* which then proceeds with the next increment of the sweep. This ensures the synchronous operation of the two VIs and thus makes the sweep data deterministic. After the increment, the flag is again set to FALSE and returned to the *Interface VI*. This method of stalling the sweep loop's iterations until the data is ready is known as *polling*. It is a commonly used programming technique for handling race conditions and establishing synchrony.

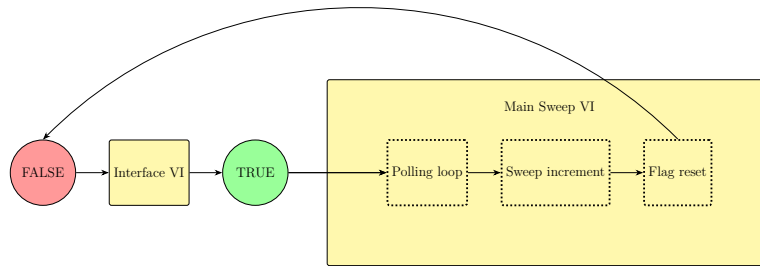


Figure 4.3 The flag moves from the *Interface VI* to the *Main Sweep VI*. Inside the sweep VI, it terminates the polling loop, allows the incrementation of the sweep and then resets itself before being sent back to *Interface VI*.

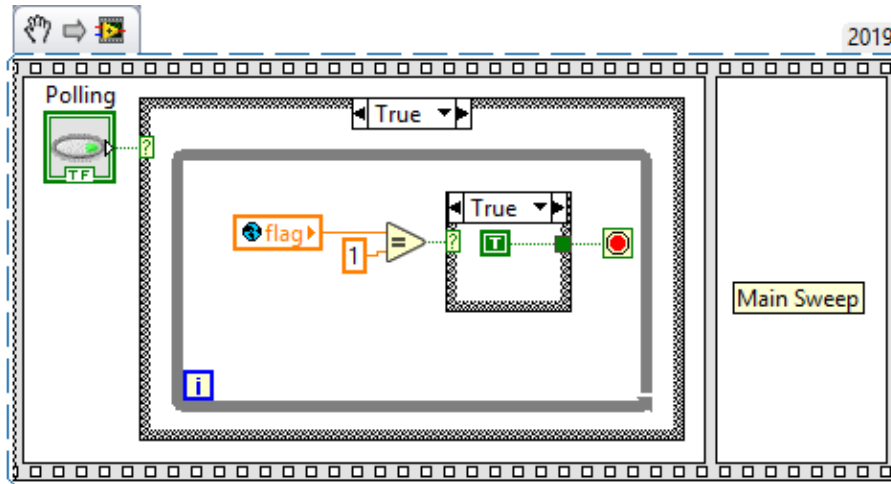


Figure 4.4 The main sweep is executed only when the flag matches the assigned value allowing execution to move to the second frame. The whole frame is contained in a for loop which executes for a number of times that is equal to the resolution of the sweep. The “frame” is known as *sequence* structure in LabVIEW.

Implementation of the polling helped improve the asynchronicity to a certain extent. However, we still observed some residual effects which by no means could be ignored and therefore decided to investigate the hardware. Our suspicion now was that the ADCs were not fast enough at registering the PD signal data to the microcontroller. To corroborate this theory, we

had to measure the execution times of the different sections of the microcontroller program. In addition to that, we applied pulse trains to the CROW ring heaters and compared the intensity response. As shown in figure 4.5, the charts obtained in LabVIEW had a telling lag while the oscilloscope showed a negligible lag between the two signals. Note that the oscilloscope was reading signals directly at the output of the DAC and the input of the TIAs. This was a reliable indicator of some delays being introduced in step 6 to 8 (refer fig.4.1).

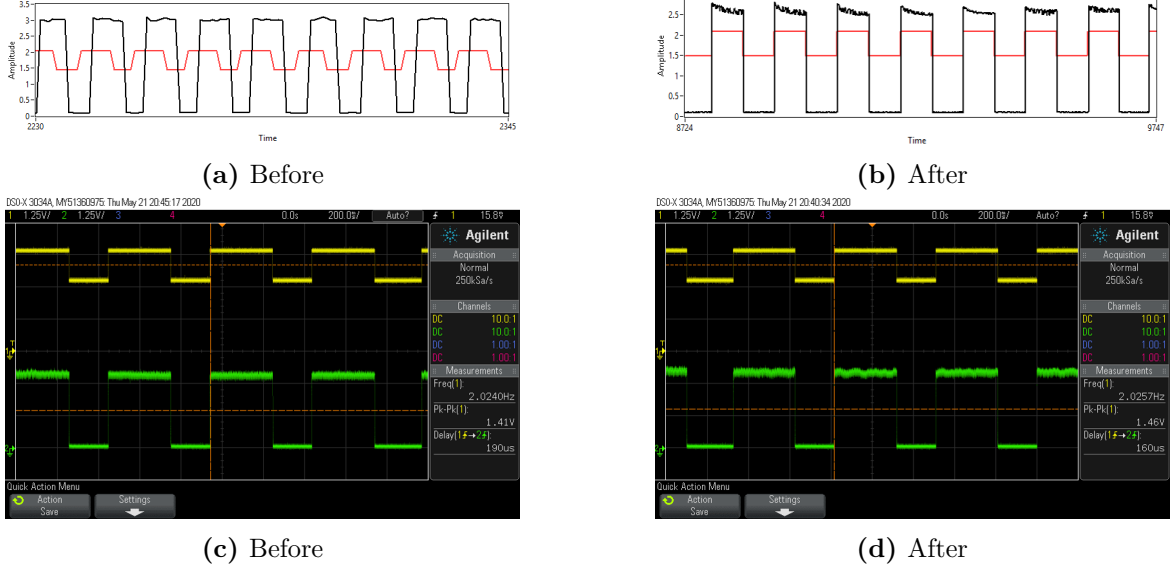


Figure 4.5 The pulse train (red in (a) & (b), yellow in (c) & (d)) applied to the heaters and the intensity response of the chip (black in (a) & (b) and green in (c) & (d)). The shift in the signals seen in (a) was found out to be due to larger execution time of function used to read the ADC.

Thus it was necessary to find out the execution time of different sections of the microcontroller code. There are several ways of running timing tests on a microcontroller, some of which rely on a software based implementation while some on hardware based implementations. The software based methods rely heavily on the specific model of the microcontroller and more often than not, implementation is complicated. On the other hand, a less refined but almost equally reliable way to do the same is using some GPIO pins of the microcontroller to signal the end of a section of the microcontroller program. We simply programmed one such pin to pull-up at the beginning of a section and pull-down at the end of that section. The waveform of the signal captured on a DSO (oscilloscope) can be interpreted to find the relevant execution times. We found that the section of the microcontroller program that read the channel values of the ADC required 17.8 ms to pass all the data to the microcontroller. We optimized the function that was used to read the ADC's channel data. This led to an improvement in the execution time of this section of the program which was then 2 ms. This was sufficient improvement to reduce the lag in the data so that the alignments could be carried out reliably.

Chip Temperature

The initial attempts to control the ring resonators were marred due to large variations introduced in the chip temperature on application of the heater signals. The *look-up tables* used for the algorithm were invalidated due to the changes in chip temperature and this caused the *wavelength-locking* to fail. The spectrum shifted by 0.075 nm for every degree centigrade change in the chip temperature. When all heaters were turned on, they consumed approximately 300 mA and with a voltage of 10 V, this resulted in ~ 1.5 W being pumped into an area of approximately 10×6 mm². This could heat-up the chip beyond 60°C.

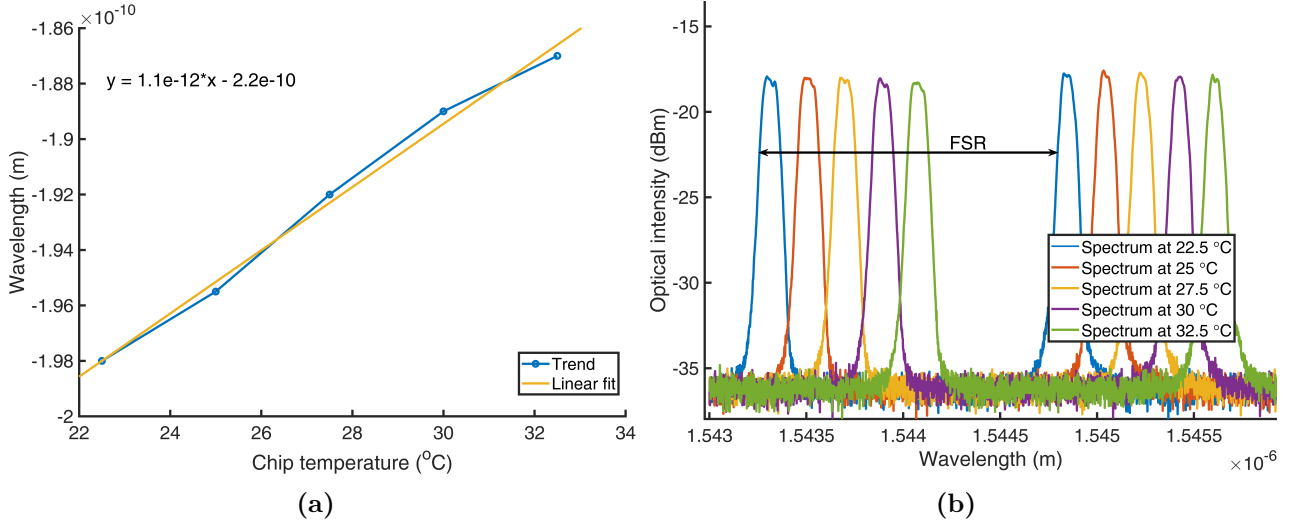


Figure 4.6 The spectrum shifts linearly with changes in the chip temperature. In figure (a), we can see the shift of the resonant frequency of one of the two-ring CROWs with respect to temperature. Figure (b) shows the spectrum of the drop-port of the two-ring CROW over five different chip temperatures.

To remedy this, we installed a thermoelectric cooler (TEC) on the chip carrier. The TEC was able to maintain a stable chip temperature with the help of a temperature controller. It is worth mentioning here that the TEC required a heat sink which is large enough to maintain a fairly stable thermal gradient between the chip and sink. Smaller heat sinks resulted in thermal runaways. Once this was functional, the algorithm was able to perform reliably.

Parallelism

Even though we initially assumed that controlling each CROW filter independently and simultaneously is the best approach, we had to choose an alternative methodology to ensure overall performance. Our first program architecture was based on the concept of *parallelism*. Implicit-parallelism is a strength of LabVIEW and we exploited it for simultaneous real-time control of each filter. It maximizes utilization of the number of cores of a CPU to run different program-snippets simultaneously. LabVIEW does this by identifying sections of the program that can be executed independently and then grouping them as *clumps* in a run queue. Even

though this concept is an in-built feature, in principle it can also be used to design the program. Instead of having a sequence of operations in queue within one while loop, it is more pertinent to have some of the tasks run in parallel. Specific to our requirements, we had two layers of such tasks. (1) The first layer is the set of tasks that can be parallelized for one channel. (2) The second layer consists of all of these channels themselves. This means the programs inside each parallelized while loop are unique to a channel and LabVIEW can switch execution among them such that one channel is not left waiting for the entire duration of another channel under execution. Despite its potency, this was not the best solution for our test application since the thermal gradients caused by the heaters within the *PIC* dictated that we actuate each set of CROW filters sequentially (as opposed to parallel actuation) to ensure the initialization is fast and efficient. As the temperature of the *PIC* increased due to consecutive initializations, the spectrum saw a red shift and consequently the previously tuned filters would move out of alignment. To ensure that the algorithm was not confused by this phenomenon, we opted for a systematic, sequential approach at the cost of instantaneous response of the control system. As shown in the figure 4.7, we used a *stacked sequence structure* to circulate through the execution of different channels. However, we continued to maintain parallelization of the loops necessary for the Producer-Consumer design (see section 3.2.1).

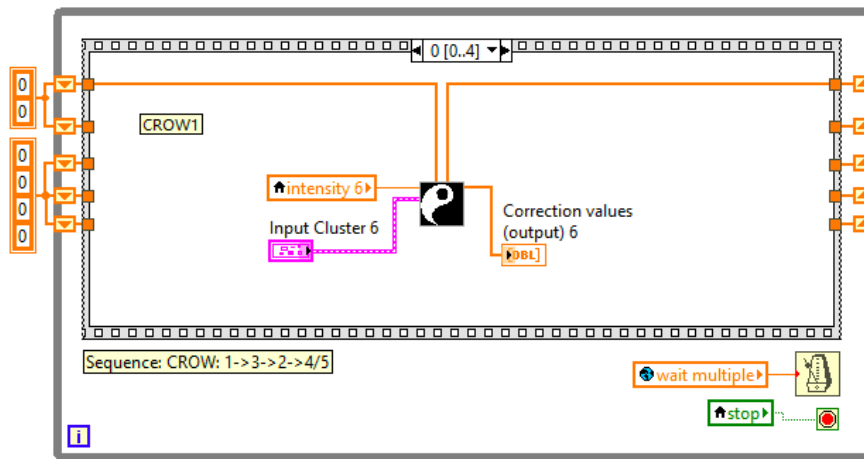
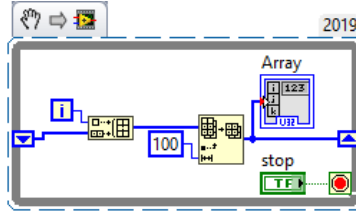


Figure 4.7 The *stacked sequence* structure in LabVIEW was used to sequentially go through the control of each CROW filter.

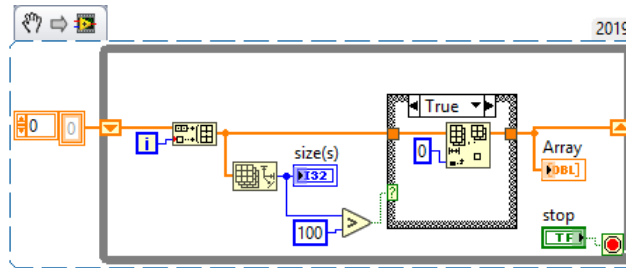
Memory Optimizations

The program utilizes functions that stream data through it and then uses this dataset to find the optimum intensity point. To this end, a structure that we used initially for testing one channel turned out to be inefficient later on when we expanded to multiple channels. This structure was used to stack the most recent data to the bottom of an array. An *Array Subset* block was used to take a slice of this data which was used to calculate the optimum position (refer fig. 4.8). For one channel, LabVIEW's internal memory management was sufficiently capable of handling the cached array that was being used to feed the Array Subset block. However, when we expanded to sixteen channels, the bulk of the cached data tended to slow down the whole VI

after it was run for some time. A small modification allowed us to ensure the size of the cached array was also under our control. This greatly improved the performance of the program and the issue of the slow down. We were able to detect this problem by using the *Performance and Memory...* tool in LabVIEW (found under Tools ► Profile). The modification deleted the oldest value from the array when the designated array size was exceeded.



(a) The structure that allowed recursive growth of the array.



(b) The modified structure that ensured that the array size remained constant.

Figure 4.8 The modifications made to the code to ensure that all arrays that were used to stream data were at a constant size and did not grow in an unbounded manner.

Flushing variable

Due to the nature of *shift registers* in LabVIEW and the way the voltage sweeps (of the heaters of the ring resonators in a single CROW filter) were designed, an unwanted transient was introduced in the intensity profile during a changeover sweep from one ring resonator to another. Shift-registers essentially store the value from the last iteration of a loop and pass it on the next iteration. This is essential for the sweep function. However, during a switchover event, the voltage value of the last sweep was sent out as the first voltage value of the next sweep before it started to send out its own values. In spite of being simple in theory, this problem was convoluted due to the use of *global variables* and real-time transfers of the data between various levels of the program. To fix this, we introduced a “flushing” global variable that communicated between the *Handover VI* and the *Main Sweep VI* of a channel. This variable prevents the first value generated by the *Main Sweep VI* from being passed to the main VI. Instead, for the first iteration, it passes the previously known *look-up table* value. It functions by preventing the consecutive sweep from launching until it is ensured that new intensity data has entered the shift-registers of the loop (Shift-registers are special structures in LabVIEW that store data from previous iterations of a loop).

De-tuning

A requirement for achieving equal intensities at the drop port of the CROW filter required us to introduce a mechanism for de-tuning the filters after they were fully aligned by the *Initialization* step. To do this, we introduced a modification that did the following. Once the optimum intensity was reached by using the *Initialization* mode, the *Dither-tracking* mode is turned on, thereby confining the sweeps to a minuscule range. Following that, the “Detune” mode is turned on and the desired intensity at the drop port is set as a percentage of a manually set intensity value which is approximately the best achieved intensity. From here on, the sweeps are programmed to stop execution as soon as the set intensity is detected. The mechanism of the dithering is able to gradually pull down the intensity since it either pushes the resonators up or down in the spectrum until the desired intensity is attained and then maintains it. Further development of this modification will enable the algorithm to carry out controlled increase of bandwidth at the cost of some optical intensity.

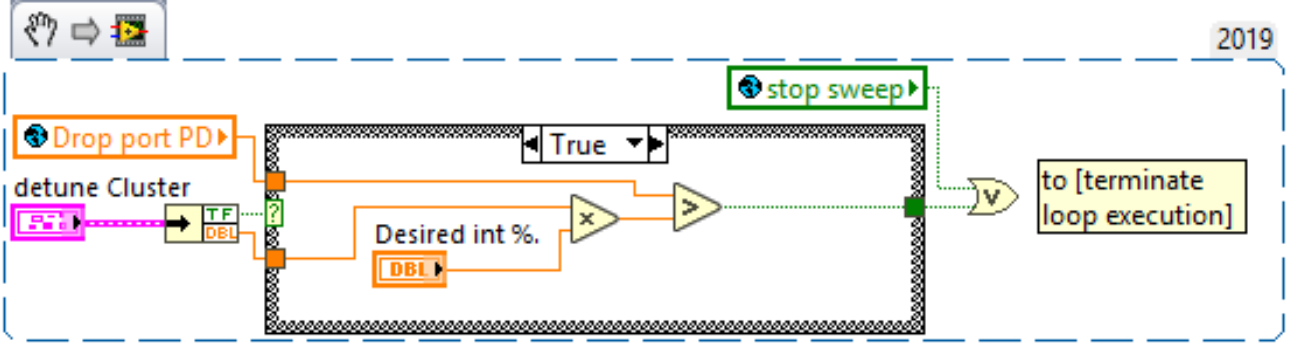


Figure 4.9 Detuning snippet

Non-linearity Compensation

The relationship between the heater voltage and the resonant wavelength is non-linear (refer figure 4.11). As the voltage is increased, the amount by which the resonant wavelength shifts also increases. This resulted in non-uniform performance of the algorithm for different CROW filters. In our experiments, and this is also valid in general application, it was noticed that some of the CROW filters required lower voltages whereas the others required much higher voltages (such as full FSR voltage) for the *wavelength-locking*. As discussed earlier in section 3.1, the dithering of the heaters are carried out over a single peak-to-peak range. When this dither signal is applied to the heaters that are already at a very high voltage, the undulations in the signal output of these CROWs are much higher compared to the ones at lower voltages. The ability of the algorithm to not significantly affect the signal quality is hampered due to this. To mitigate this, the FSR voltage range was divided up into three sections. In the first section, which is from zero volts to 33.33% of FSR voltage, the dithering amplitude was left unchanged, and for the two consecutive sections (33.33%-66.66% and 66.66%-100%), the amplitude was reduced by factors of 0.50 and 0.25 respectively.

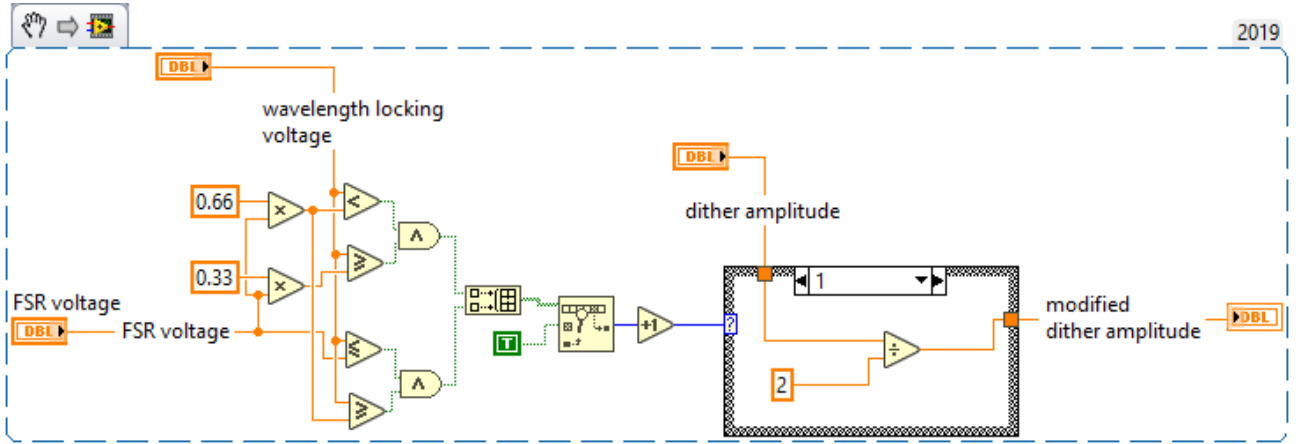


Figure 4.10 The solution used to counter the effects of non-linearity depicted as a LabVIEW code-snippet.

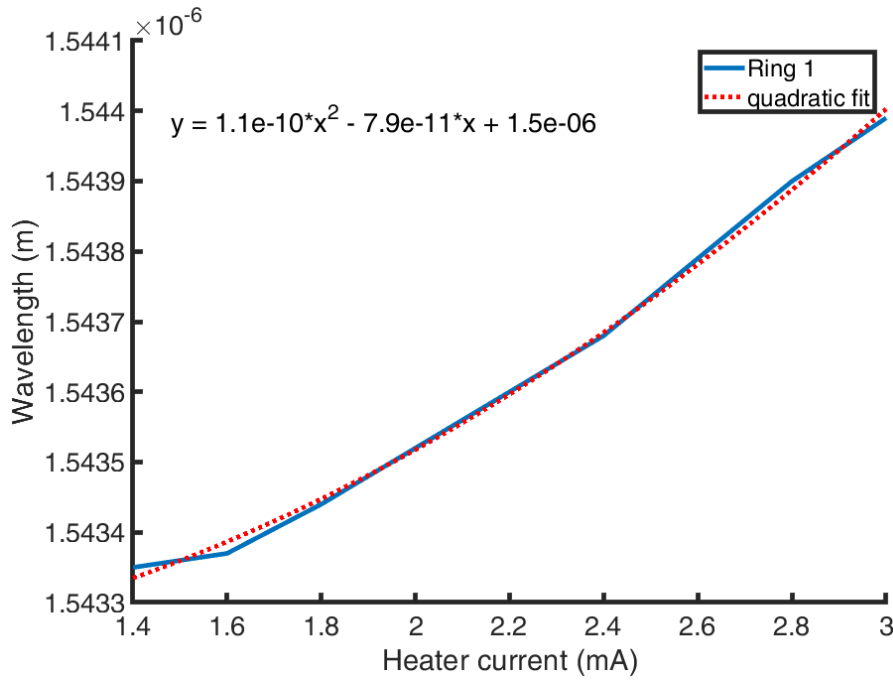


Figure 4.11 The quadratic relationship between the resonant frequencies of the first and second ring of a two ring CROW filter versus the current applied to their heaters.

Synchronization with the hardware and data validation

The NI-VISA function has an option to modify its behavior based on how we want it to respond to calling threads. For this, there are two modes available, namely the *asynchronous* mode and *synchronous* mode under the option *synchronous I/O mode*. The exchange of data between the VI and the hardware occurs through the ADE (Application Development Environment). During a write operation, the ADE takes the data from the raw code (VI) and passes it on

to the physical resources of the computer which in turn transmit it to the hardware. These transfers also pass through several layers of memory.

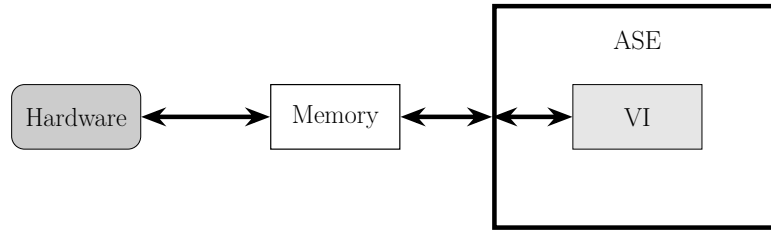


Figure 4.12 Data exchange between hardware and LabVIEW through ADE using NI-VISA.

For read operations, the following sequence occurs,

1. The VISA read function is called
2. The calling thread is locked
3. The transfer completes
4. Polling begins to determine when the data being read has finished transferring into VISA memory from the hardware memory
5. The read data is transferred from VISA memory to ADE
6. The calling thread returns

It is recommended to use the synchronous mode for large data transfers as the calling thread remains locked until step 5. This means, the calling thread is not able to access other parts of the application during the transfer of data from the external hardware. For write operations, the sequence is reversed.

This is one major step to ensure the synchrony of the data transfer between LabVIEW and the hardware. However, to ensure that the microcontroller itself was not sending data before the corresponding heater voltages were applied, we implemented a signaling mechanism that ensured all executions were sequential. A *flag* or *token* was introduced into the data that was modified as it passed through various sections of the LabVIEW and microcontroller program. In the *Main Sweep VI*, this flag was set to *symbol 1* at the end of the sweep and the passed on to the *Interface VI*. The interface VI passed the data along with the flag to the microcontroller where it was modified again to *symbol 2* after the execution of the *DAC section*. Subsequently, it moves to the *ADC section* where it is modified to *symbol 3* and transmitted back to LabVIEW. In LabVIEW, the *VISA read* module picks up the flag, checks that it is *symbol 3* and if it is so, validates the ADC's data.

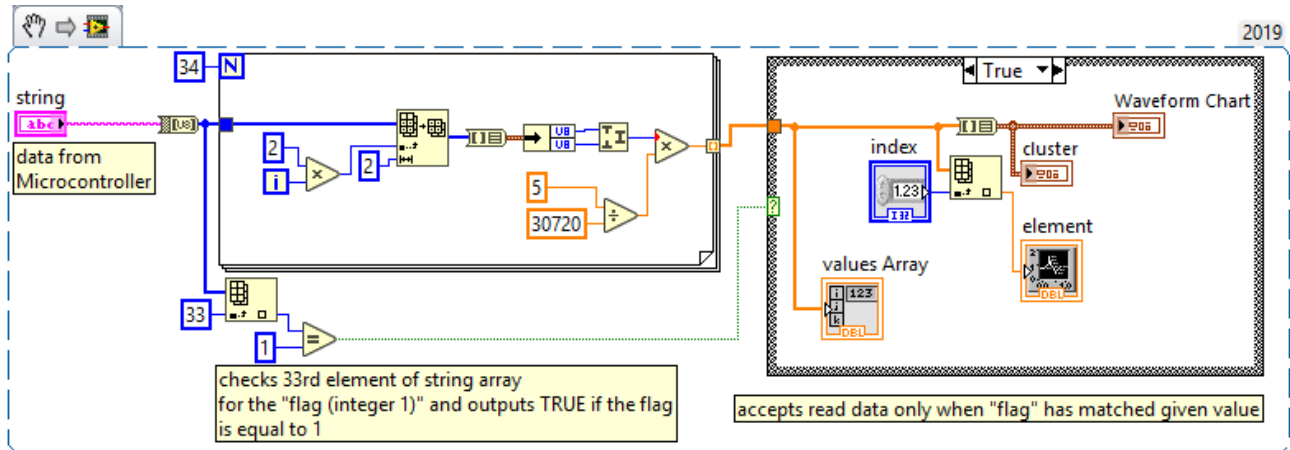


Figure 4.13 The read data is accepted into the algorithm only when the flag matches the assigned value (here, the integer 1)

Noise

Noise in the raw intensity signal that was read by the program was a concern since it reduced the accuracy of the alignments when the voltage sweeps were reduced to smaller ranges. We used the *point-by-point Butterworth Filter function* in LabVIEW to filter the noise digitally. The filter was used in the *Handover VI* where it was processed and then sent to the *Main Sweep VI*. It improved the standard deviation of the noise in the intensity signal from 6 mV to 2 mV.

Measurements

5.1 The NLFT Project

The control algorithm has been developed with the intent to control the CROW filters of used in several different SiP systems. However, during the development, it was tested on and initially configured for a project that aimed at using solitons for nonlinear Fourier transform (NLFT) based transmission systems [38]. The project is about the realization of a silicon photonic (SiP) transmitter (Tx) that modulates four first-order solitons and multiplexes them with adjustable time-delays. The SiP system used for this purpose employs coupled-resonator optical waveguide based OADMs. Four 2^{nd} order CROW OADMs drop four carriers which act as the first-order solitons. The carriers are then directed into two IQ modulators with MZMs as each arm and a phase shifter in one of the two arms. The output of each IQ modulator is then split and directed into two waveguides. The first waveguide has no delay lines and the second waveguide has two delay lines. Consecutively, 4^{th} order CROW OADMs are used to drop the modulated carriers from these waveguides into a common waveguide which is routed to the system output. To maintain a stable performance of the output signal, it was necessary to stabilize the resonant frequencies of the 2^{nd} order CROW OADMs and 4^{th} order CROW OADMs of this system.

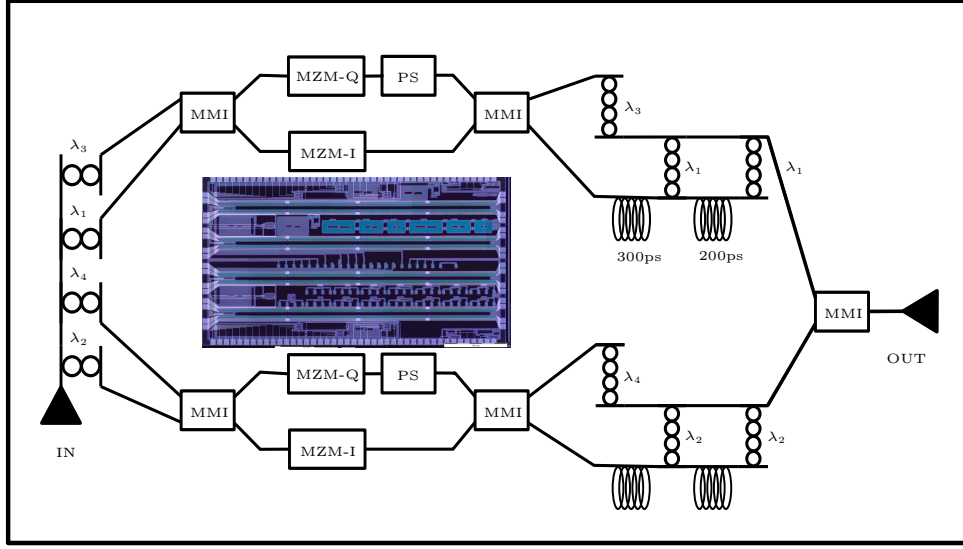


Figure 5.1 A comb source is injected to the input of the chip through a grating coupler. Four consecutive comb lines ($\lambda_1, \lambda_2, \lambda_3$ and λ_4) are selected with the 2^{nd} order CROW OADMs. λ_1 and λ_3 are combined with an MMI and sent into the first IQ modulator. At its output, an MMI is used to split the modulated carrier into two streams. The first stream has a 4^{th} order CROW OADM that drops λ_3 while the second stream allows the introduction of a 300ps or a 500ps time delay before another 4^{th} order CROW OADM drops λ_1 . Both λ_1 and λ_3 are dropped into a common waveguide. Similarly, λ_2 and λ_4 are also dropped into another common waveguide. Both the common waveguides are then combined with an MMI and sent to an output grating coupler. (Inset) The silicon system chip.

5.2 Experimental Setup

For the first phase of tests, the Agilent 8164A/B Lightwave Measurement System along with 81640B tunable laser module was used to set a single laser wavelength to which we aligned the CROWs individually. It was also used to check the optical spectrum after alignment by using its wavelength sweep mode. The laser was a Class 1M InGaAsP EC-Laser with a wavelength range of 1495-1640 nm. Maximum continuous wave output power was < 15 mW. The power meters that accompanied these measurements were the Agilent 81634B and 81636B. It has an InGaAs sensor, wavelength range of 1250 nm to 1640 nm and power range of -80 dBm to $+10$ dBm. The tunable laser and the power meters were connected to a fiber array that coupled to the system chip via grating couplers. The chip was mounted on a chip carrier and all signal and ground pads were wire bonded to the carrier. The carrier had pin headers that allowed access to the terminals of the chip. These pin headers were used to connect the inputs and outputs of the control board. The control board was designed in-house and consisted of three main sections, (1) the TIA (OPA2380) and ADC (ADS1158) section that picked up the signal from the monitoring photodiodes on the chip, (2) the DAC (MAX5735) and buffer (OPA4251) section which generated the actuator signal for driving the heaters and (3) the microcontroller (ATmega256A3B-AU) that was used to control these peripherals and communicate with a computer that ran LabVIEW [39]. We used the control board as an interface between the

system chip and LabVIEW but it is also capable of functioning as a standalone controller and has been used for the control of SRRs.

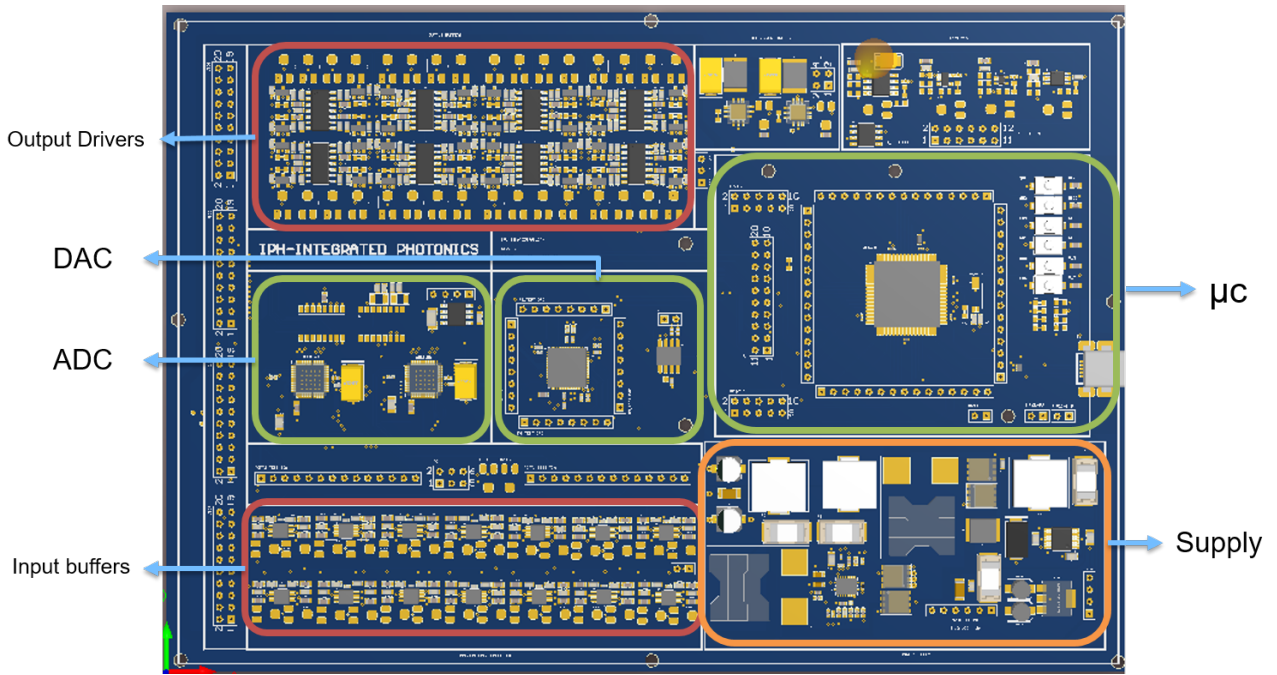


Figure 5.2 The control board with the various sections labeled, namely, the TIA (OPA2380) and ADC (ADS1158) sections that picked up the signal from the monitoring photodiodes, the DAC (MAX5735) and driver (OPA4251) sections which generated the actuator signal for driving the heaters and the microcontroller (ATxmega256A3B-AU) that was used to control the peripherals and communicate with a computer that ran LabVIEW.

LabVIEW from National Instruments has been used to implement the control algorithm. It is a graphical programming software which is widely used for hardware applications, measurements and control.

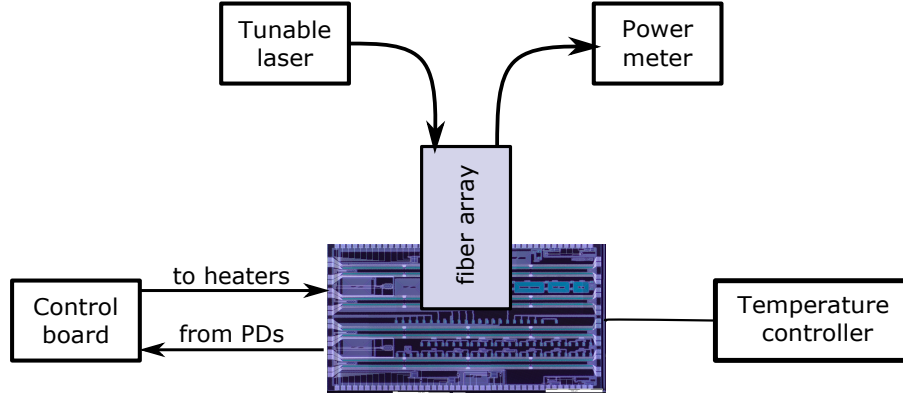
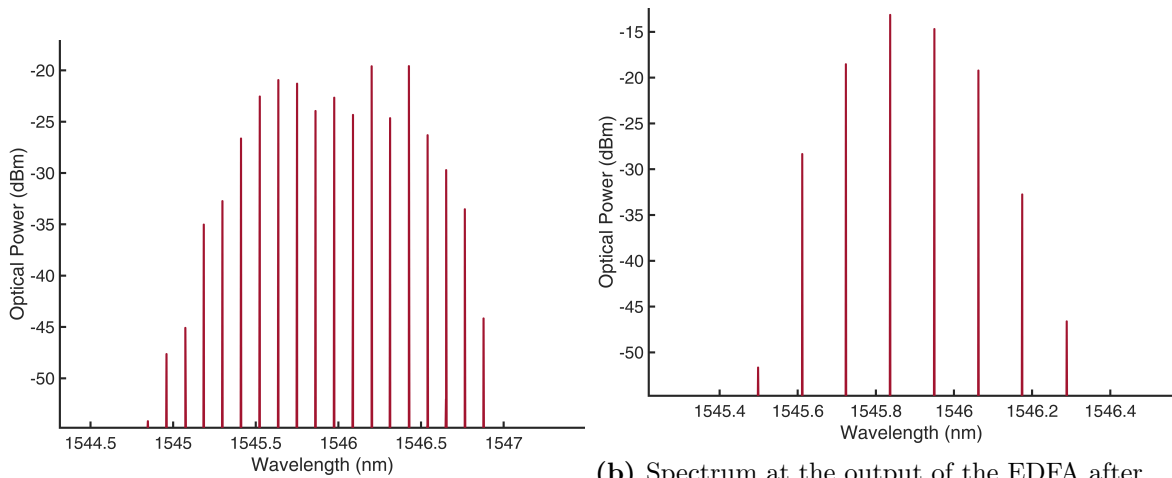


Figure 5.3 A tunable laser was used to inject the carrier into the chip with a fiber array. The through port of the 2nd order CROW OADMs, drop port of 4th order CROW OADM and system out were connected to power meters as required, via other ports of the same fiber array. The inputs and outputs of the control board were connected to the pin headers of the chip carrier. The chip carrier had a TEC mounted on it which was controlled by a temperature controller.

For the second phase of tests, a comb laser was used for aligning the CROW filters to consecutive comb lines. We used the Pilot Photonics LYRA-OCS-1100 which has a flat comb of coherent wavelengths with low optical linewidth, tunable central wavelength and tunable wavelength spacing.

The Amonics C+L-Band PM Erbium Doped Fiber Amplifier AEDFA-PM-CL-30-R-FC was used as the optical amplifier for the comb laser. It has a saturation output power of $> +30$ dBm, small signal gain @ -6 dBm input signal of > 29 dB and operation wavelength in the range 1528 nm to 1562 nm + 1570 nm to 1603 nm.

We also introduced an optical filter between the comb laser and the EDFA to reduce the net optical power being fed to the system chip. The optical filter had a passband of 39GHz.



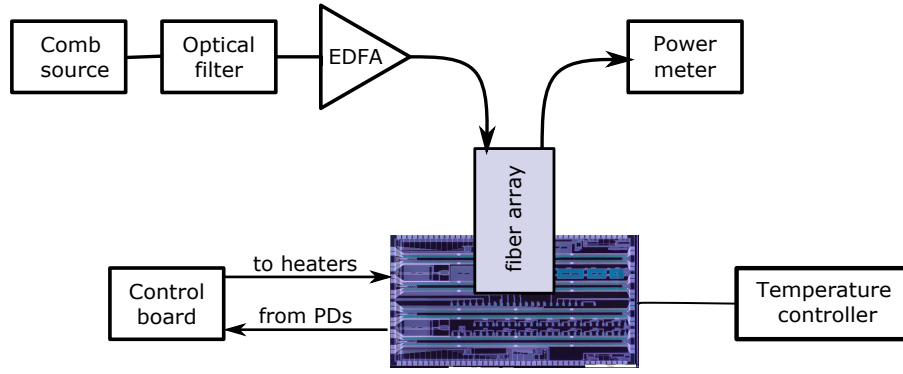
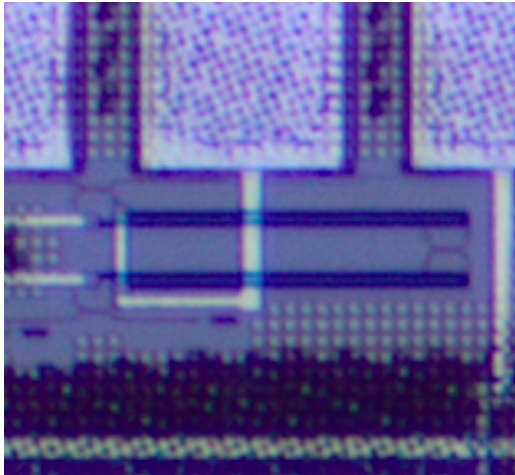
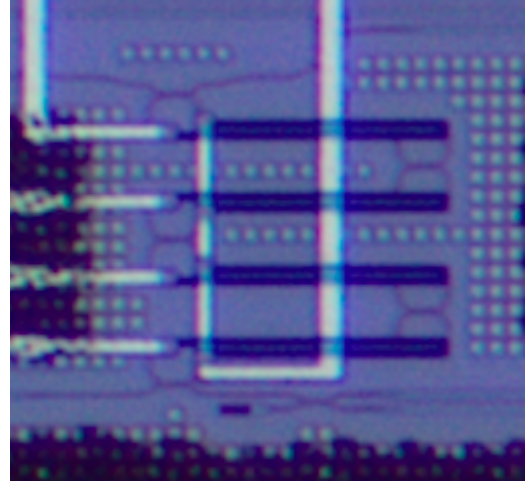


Figure 5.5 For the second phase of the measurements, the tunable laser was replaced with a comb source. An optical filter and an EDFA were added to the output of the comb source.

When fully aligned, the two ring CROW had a bandwidth of 10.8 GHz at the drop port when properly tuned. The FWHM was 0.086 nm and the Q-factor was 17987. They had an insertion loss of 2 dB. The FSR was 190 GHz. The heater resistance was 1.6 k Ω . A minimum current of 6 mA (or 9.6 V) was required to tune the filter up to the first FSR. The four ring CROW had a bandwidth of 18.5 GHz at the drop port when properly tuned. The FWHM was 0.148 nm and the Q-factor was 10450. Insertion loss was measured to be 3 dB. They had an FSR of 280 GHz. The heater resistance in this case was 0.97 k Ω . A current of 5.5 mA or 5.335 V was required for reaching the first FSR. Insertion losses through the grating coupler was 6.5 dBm. The aforementioned data has been obtained from measurements of test structures of the devices.



(a) Two-ring CROW



(b) Four-ring CROW

Figure 5.6 The CROW filters used for testing the algorithm.

It is worthwhile to note here that it is possible to align the 4th order CROW OADMs only when sufficient optical power from the drop port of the 2nd order CROW OADMs reaches the input port of the 4th order CROW OADMs. In addition to that, it was inadvisable to carry out simultaneous alignments of the two since the 4th order CROW OADM drop port intensity would have been affected due to the on-going 2nd order CROW OADM alignment.

5.3 Execution

In the first phase of measurements, our algorithm was able to initialize the filter and obtain the highest possible intensity at the drop port of the 2nd order CROW OADM. However, due to the lack of a tap at the drop port, it was only possible to carryout a wavelength sweep of the through port with the power meter. We observed a split in the passband of the through port spectrum that indicated that the two rings were not exactly at the same wavelength (figure 5.7(a)). This ripple, to the best of our understanding, is a result of the anti-crossing behavior in micro-ring resonators [40]. To ensure how the drop port of the same filter looked, we used the on-chip photodiode at the drop port instead of the power meter. We ran a two-dimensional wavelength sweep with a LabVIEW VI that mapped sweep wavelengths to the on-chip photodiode's response signal. The resulting plot confirmed that the alignment was successful and that the drop port photodiode was indeed receiving the highest intensity.

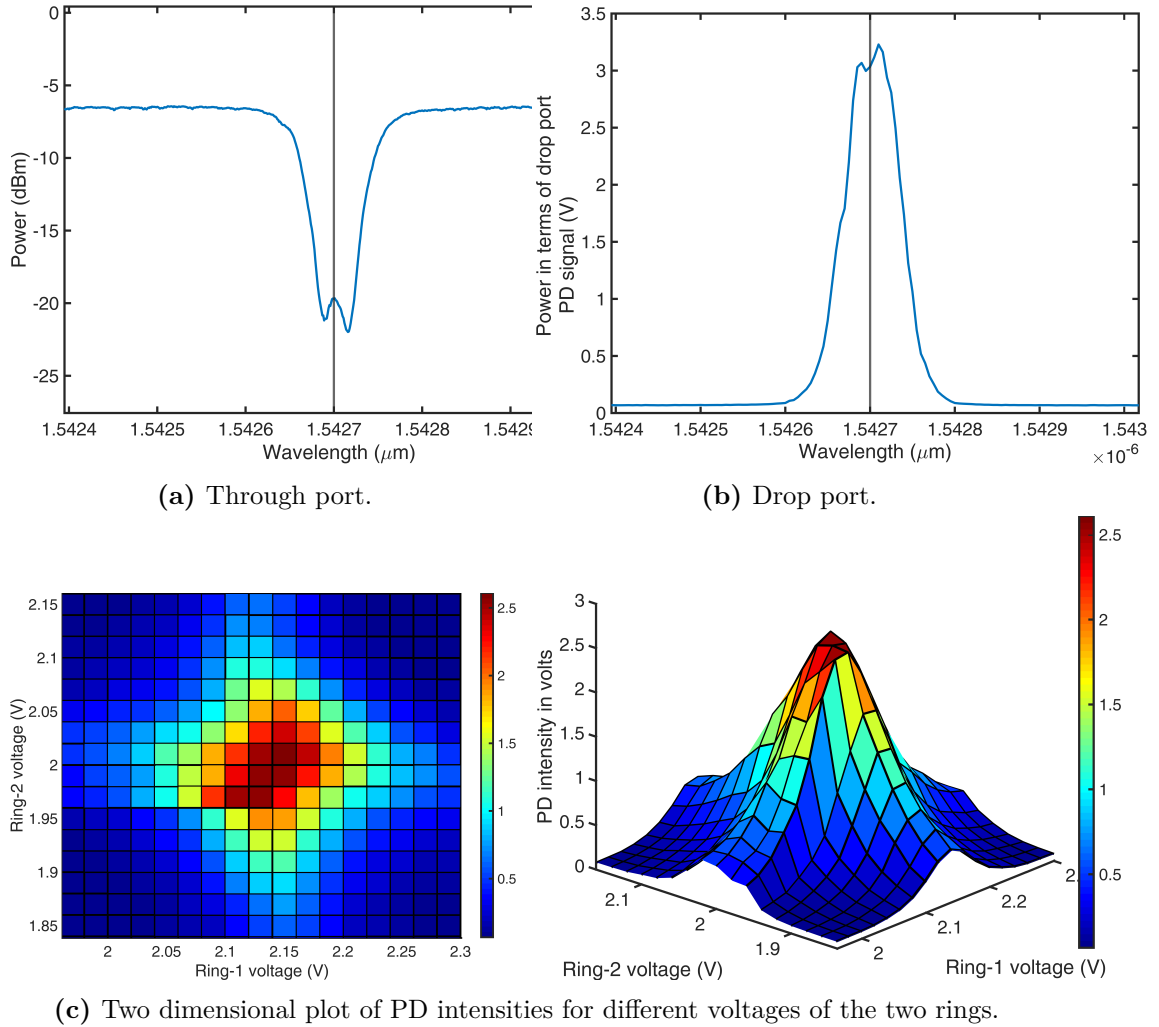


Figure 5.7 (a) The split in the through port spectrum, (b) the corresponding drop port and (c) the measurement carried out to crosscheck the voltages that produced the spectrum in Fig:(a).

For the 4th order CROW OADM, we used a -20 dB tap to inject light at a point that bypassed the 2nd order CROW OADMs and the IQ modulators. The spectra of the two 4th order CROW OADMs with the delay lines have been shown in figure 5.8. In the figure, the -20 dB drop in optical power due to the tap has been compensated for the ease of reading the figure.

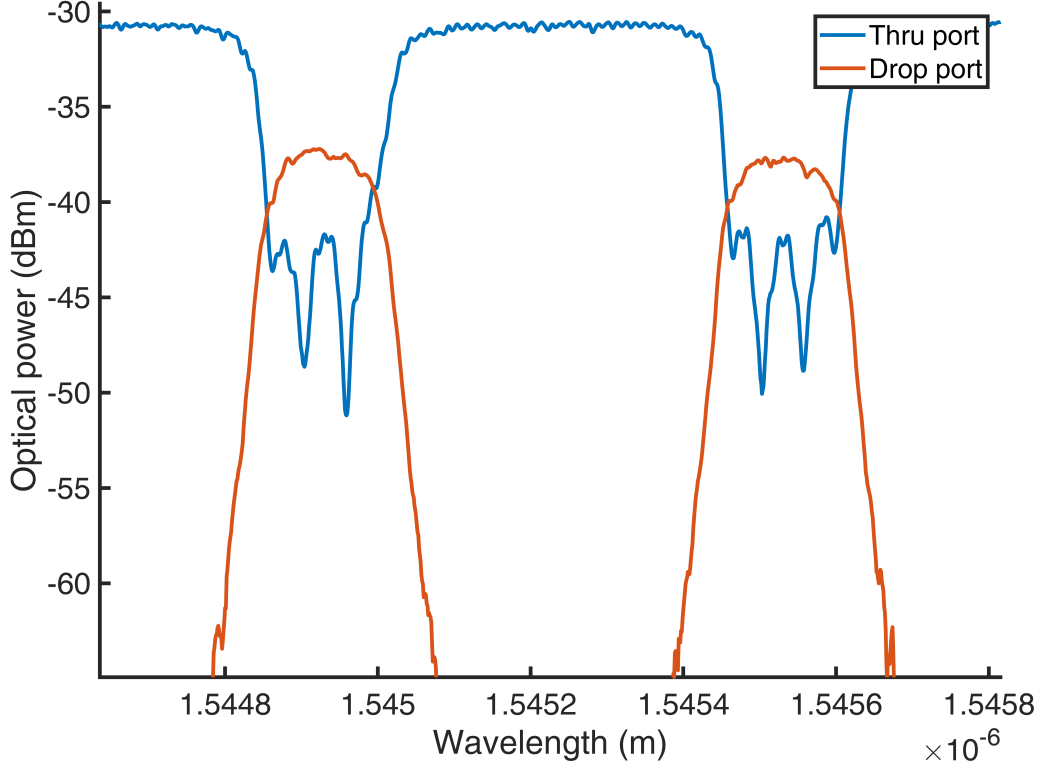


Figure 5.8 The through port and drop port spectra of two of the 4th order CROW filters.

The second phase of measurements were carried out to align all of the four channels of the system chip to four consecutive lines of the comb spectrum. The requirement for carrying this out was that the comb lines had enough power to ensure the monitoring photodiodes generated a sufficiently large signal for the algorithm to detect a variation in intensity. We connected the EDFA to boost the comb source output to 0 dBm per line. The net output power of the EDFA was reduced by filtering away some of the unused comb lines at its input to ensure the integrity of the optical grating couplers and waveguides. Once the 2nd order CROW OADMs were aligned, we needed to detune two of the four 4th order CROW OADMs with the delay lines to clear-up the spectrum for the other 4th order CROW OADMs. This was carried out by using the -20 dB tap again. Thus, in terms of the wavelengths, the 4th order CROW OADM for λ_3 was aligned and for λ_1 and we chose the 4th order CROW OADM with the 300ps delay line for λ_1 and similarly for consecutive measurements, the corresponding OADMs for λ_4 and λ_2 . Once four comb lines were chosen, the flexibility to align them to any clear region of the spectrum was gone. This led to the issue of finding the right chip temperature. As the entire spectrum has a blueshift or a redshift depending on the chip temperature, some chip temperatures were unsuitable as the alignments at those temperatures needed several of the OADMs to be moved by almost one FSR, leading to extreme heating of the entire chip. The TEC often could not

compensate these high temperatures and would go into a thermal runaway. In such cases, we had to raise or lower the chip temperature to find a temperature where those OADM's would already be close to the comb lines and would require a relatively small actuation to align.

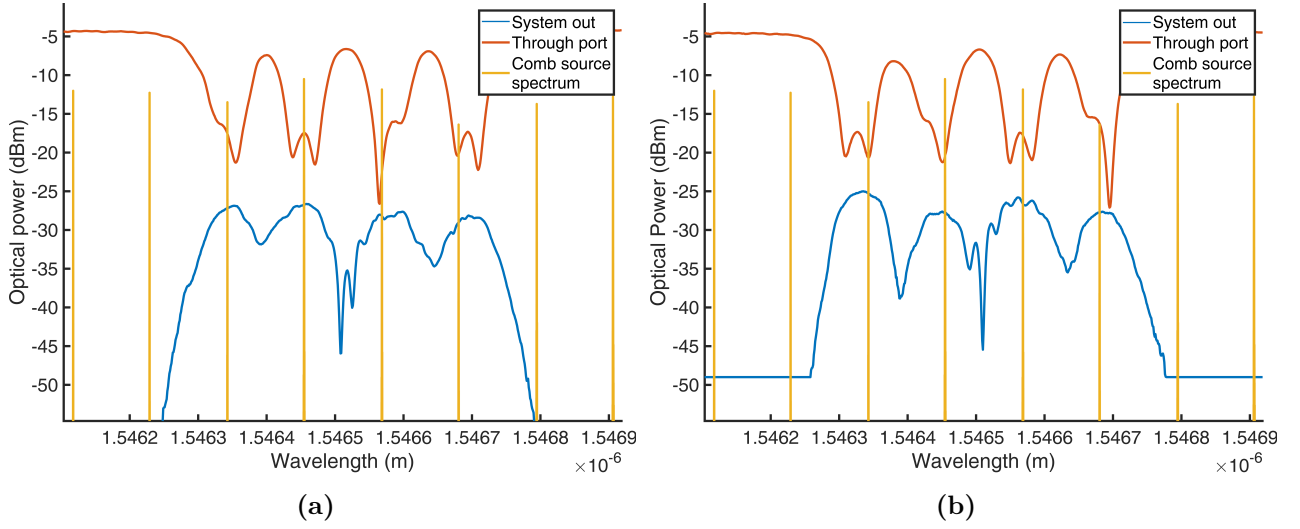


Figure 5.9 (a) The through port and drop port spectra of the 2nd order OADM's after **manual alignment** to the comb lines. The manual alignments were carried out by using the tunable laser to mimic the comb lines. Thus, the comb spectrum shown here is only for reference. (b) The same spectrum after using the comb source and the algorithm for alignment.

The graphs shown in figure 5.9 do not represent the drop ports of the 4th order CROW OADM's properly since they are a superposition of the 2nd order CROW OADM drop port spectrum and 4th order CROW OADM drop port spectrum. Hence, we used the -20 dB tapping to check the drop ports directly by stopping the algorithm momentarily and carrying out a wavelength sweep through it. We ensured that this was carried out fast enough so that the filters did not have enough time to get significantly detuned during the period when the algorithm was halted. Shown below are the first attempts to align the 4th order OADM CROW's with the algorithm. Here we only show the spectra related to the 300 ps delay line.

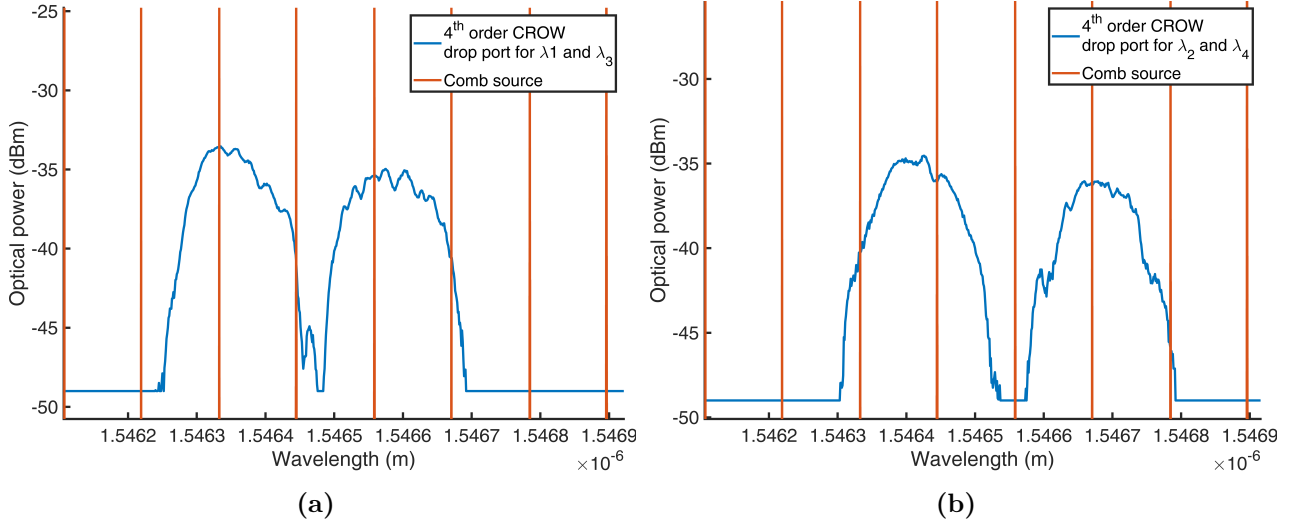
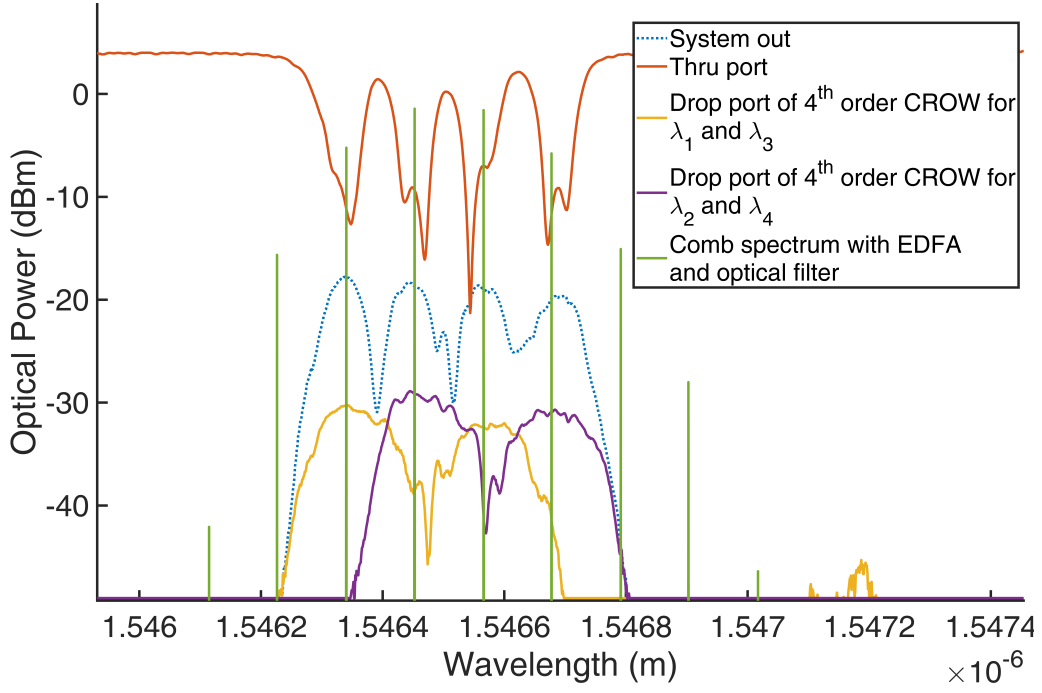


Figure 5.10 (a) The drop ports of the 4th order OADM CROWs for λ_1 and λ_3 , (b) The drop ports of the 4th order OADM CROWs for λ_2 and λ_4 . These correspond to the alignments done with the algorithm.

5.4 Results

We were able to get consistent performance from the control program. It was able to repeatedly align and track the resonant frequencies of the 2nd order CROW OADMs and 4th order CROW OADMs. We carried out the alignments on another system chip and were able to align and track with equal ease. The algorithm was able to carryout a wavelength lock and align the drop ports where the 3 dB bandwidths remained in the range of 14 GHz to 19 GHz for the 4th order CROW OADMs. The passbands were found to be centered around the laser wavelength on all occasions. It was noted that the *look-up tables* are critical for wavelength-locking when more than one OADMs share a bus waveguide. Maintaining the chip temperature is another factor that plays a profound role in enabling a successful wavelength-lock. During *Dither-tracking*, the temperature controller should also have at least a small margin from thermal-runaway temperatures. The algorithm is immune to slow variations and transient changes in optical intensity but it may not be able to deliver the best results in case the variations are persistent and at higher frequencies.

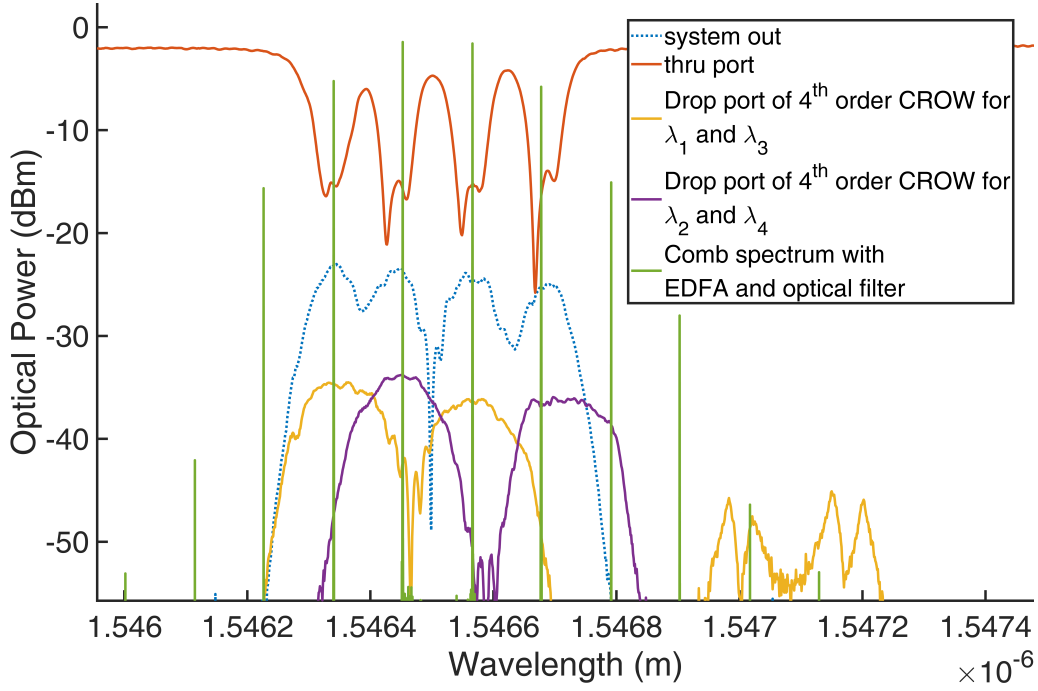


(a) The spectrum after carrying out the alignments with the first delay line in service.

4^{th} order CROW	Bandwidth (GHz)
λ_1	17.18
λ_2	14.92
λ_3 (with 300ps delay line)	14.92
λ_4 (with 300ps delay line)	17.68

(b) Bandwidths of the drop ports of the 4^{th} order CROW OADMs.

Figure 5.11 Results of the alignment of the four channels of the NLFT system chip with the first delay line in service and the second delay line moved away to prevent interference.



(a) The spectrum after carrying out the alignments with the second delay line in service.

4^{th} order CROW	Bandwidth (GHz)
λ_1	16.30
λ_2	14.17
λ_3 (with 500ps delay line)	15.55
λ_4 (with 500ps delay line)	18.56

(b) Bandwidths of the drop ports of the 4^{th} order CROW OADMs.

Figure 5.12 Results of the alignment of the four channels of the NLFT system chip with the second delay line in service and the first delay line moved away to prevent interference.

The algorithm was also tested for disturbance rejection. We varied the temperature of the chip by changing the set-point of the temperature controller. The speed of the temperature change had no impact on the recovery. The magnitude of the variation did influenced its recovery speed. In the tracking mode, it is able to recover from a temperature deviation of $\pm 1^\circ C$ without any difficulty. At $\pm 2^\circ C$ the 4^{th} order CROW OADMs fail to recover. However, this can be solved by changing the dithering magnitude. The following table lists the results:-

	2 nd order CROW	4 th order CROW
Temperature deviation	$\pm 1^{\circ}C$	$\pm 1^{\circ}C$
Recovery time	30 s	60 s
Speed of control algorithm	20 ms	20 ms

a

	2 nd order CROW	4 th order CROW
Temperature deviation	$\pm 2^{\circ}C$	$\pm 2^{\circ}C$
Recovery time	45 s	no recovery
Speed of control algorithm	20 ms	20 ms

b

Table 5.1 (a) The first disturbance rejection test was carried out with a temperature disturbance of $\pm 1^{\circ}C$ and (b) the second test with temperature disturbance of $\pm 2^{\circ}C$. *Speed of control algorithm* is the duration before a loop goes to its next iteration.

Summary and Outlook

We have demonstrated the ability to carryout a wavelength-lock and wavelength tracking of cascaded second order and fourth order CROW filters with the help of a low-speed electronic control board and a LabVIEW based control algorithm. Higher order CROW filters can also be purportedly controlled by the same methodology. The program is capable of handling multiple CROW filters simultaneously. The structure of the algorithm makes it highly adaptable to a large number of intensity based optical devices which need to be controlled actively.

The present algorithm was able to achieve wavelength locks of higher order CROW filters within one minute. It is able to restore the CROW filters back to the desired wavelengths when the system temperature deviates by $\pm 1^\circ C$. However, the algorithm can recover from larger temperature deviations by increasing its sweep range when the CROW filter does not have neighboring channels.

The performance of the algorithm can be potentially improved by implementation of more standardized synchronization mechanisms in LabVIEW such as *notifiers*. The global variables can be replaced with less memory intensive options such as *queues*. The algorithm can be made to recover faster by augmenting the program to dynamically adapt the dither tracking step sizes based on how large a difference the real-time intensity has as opposed to the last achieved optimum intensity.

The LabVIEW program

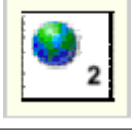
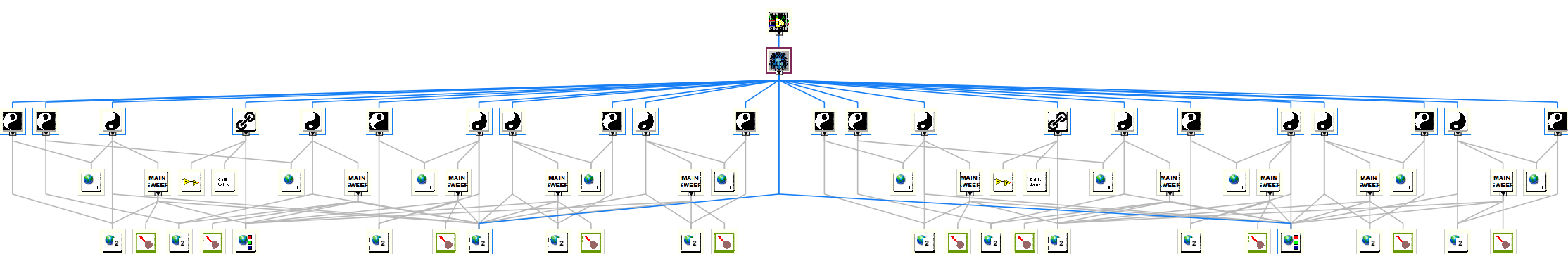
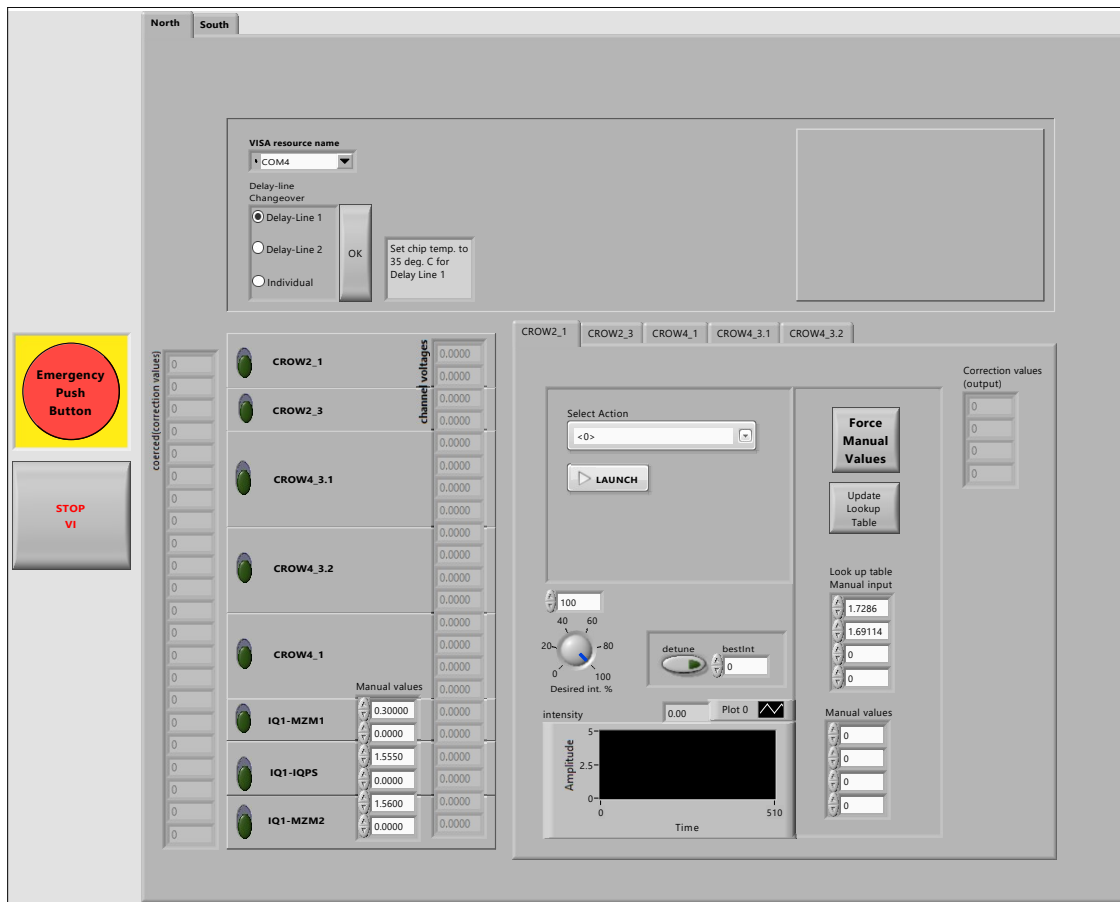
Icon	Name
	Main VI
	Interface VI
	Handover VI
	Algorithm VI
	Main Sweep VI
	Global Variable 1
	Global Variable 2

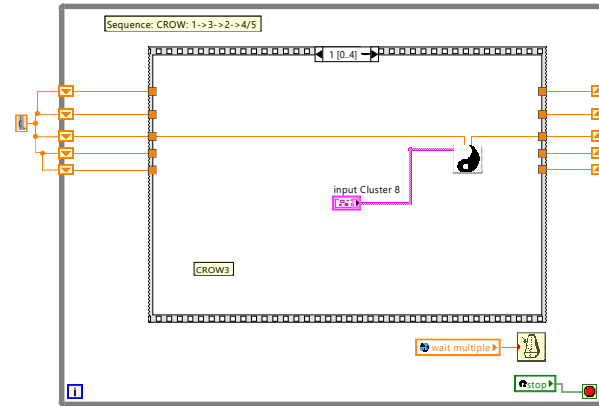
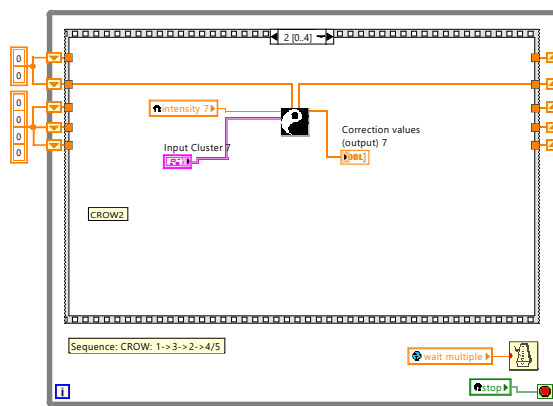
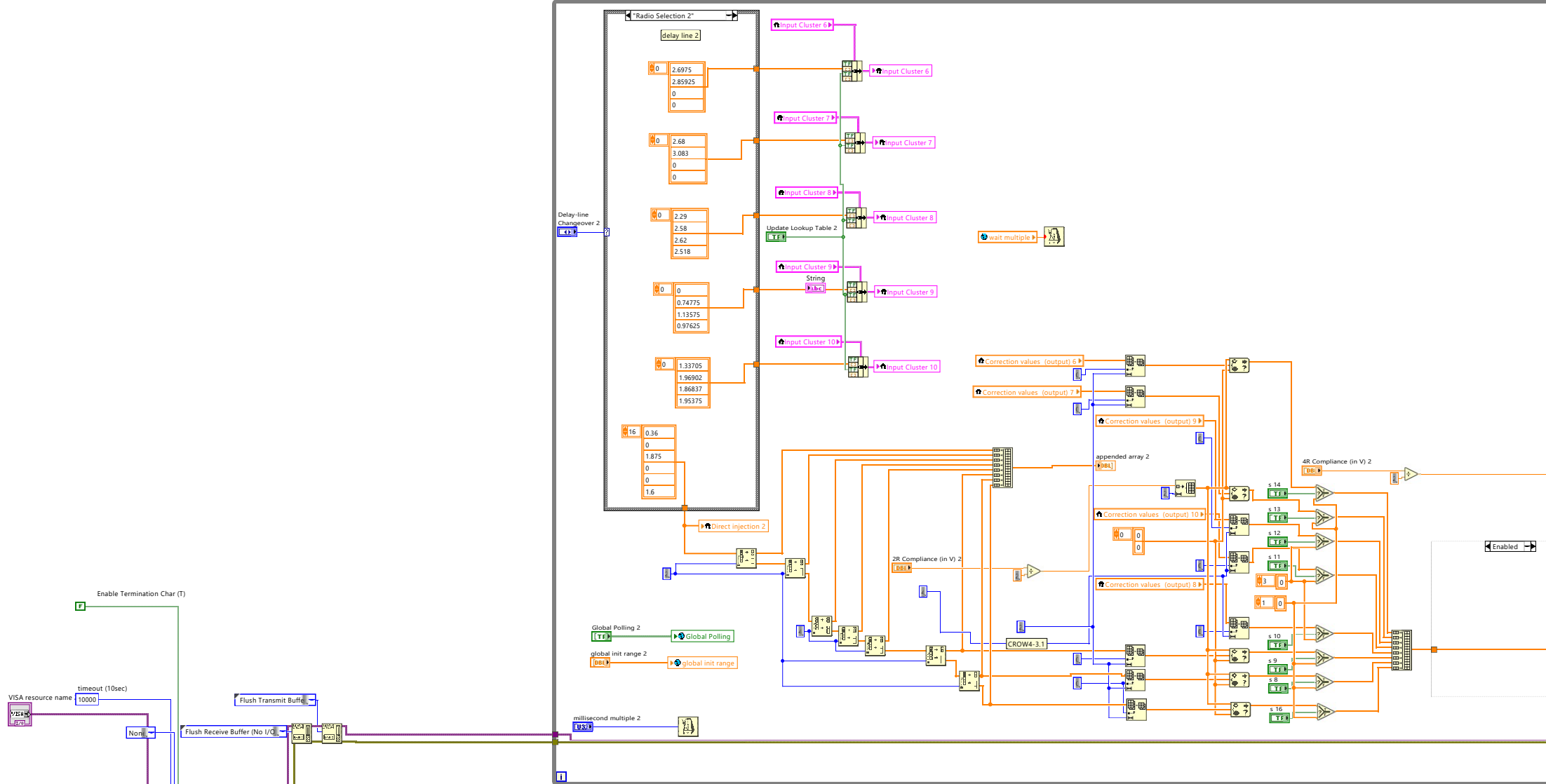
Table A.1 Legend to the LabVIEW Hierarchy diagram.

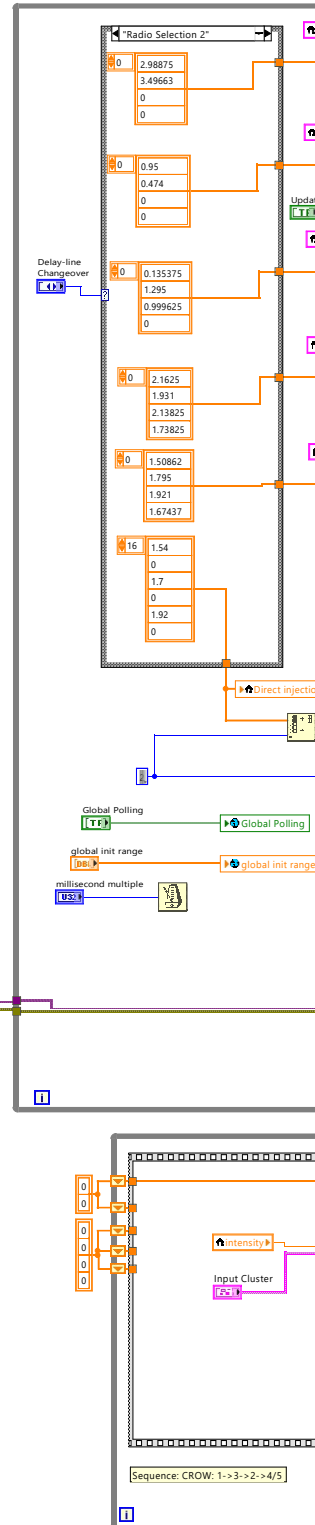
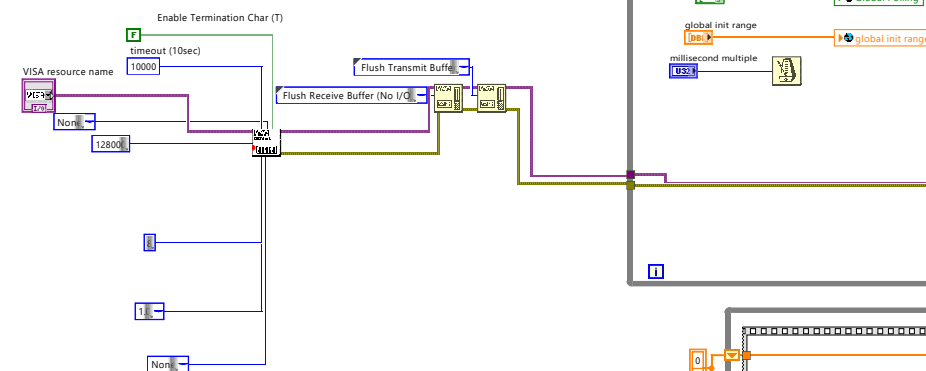
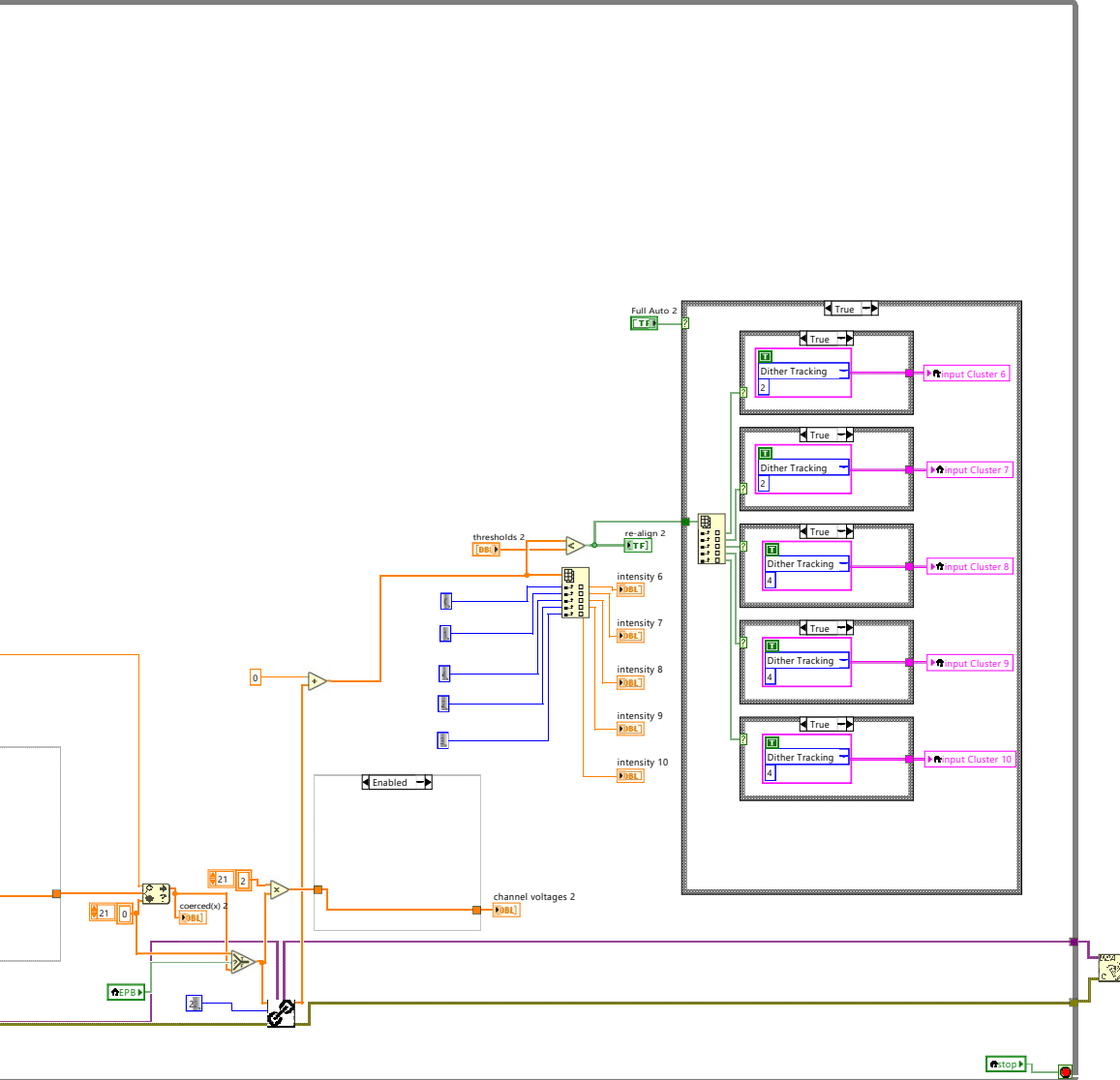


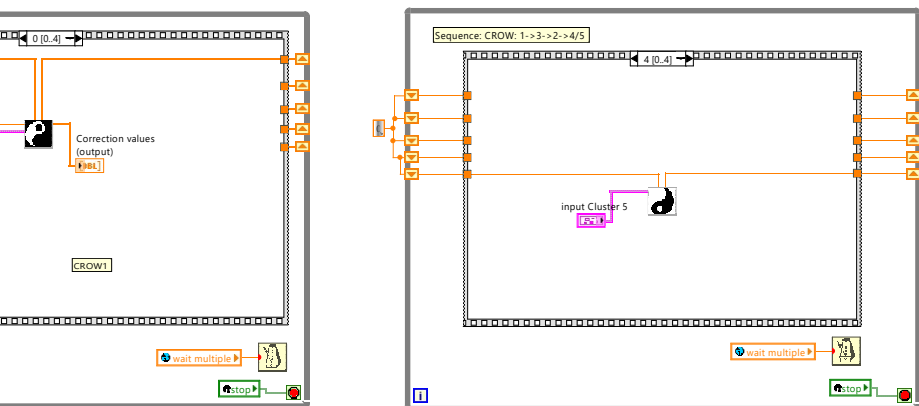
LabVIEW Hierarchy
for control of 10
CROW filters

A.1 Main VI

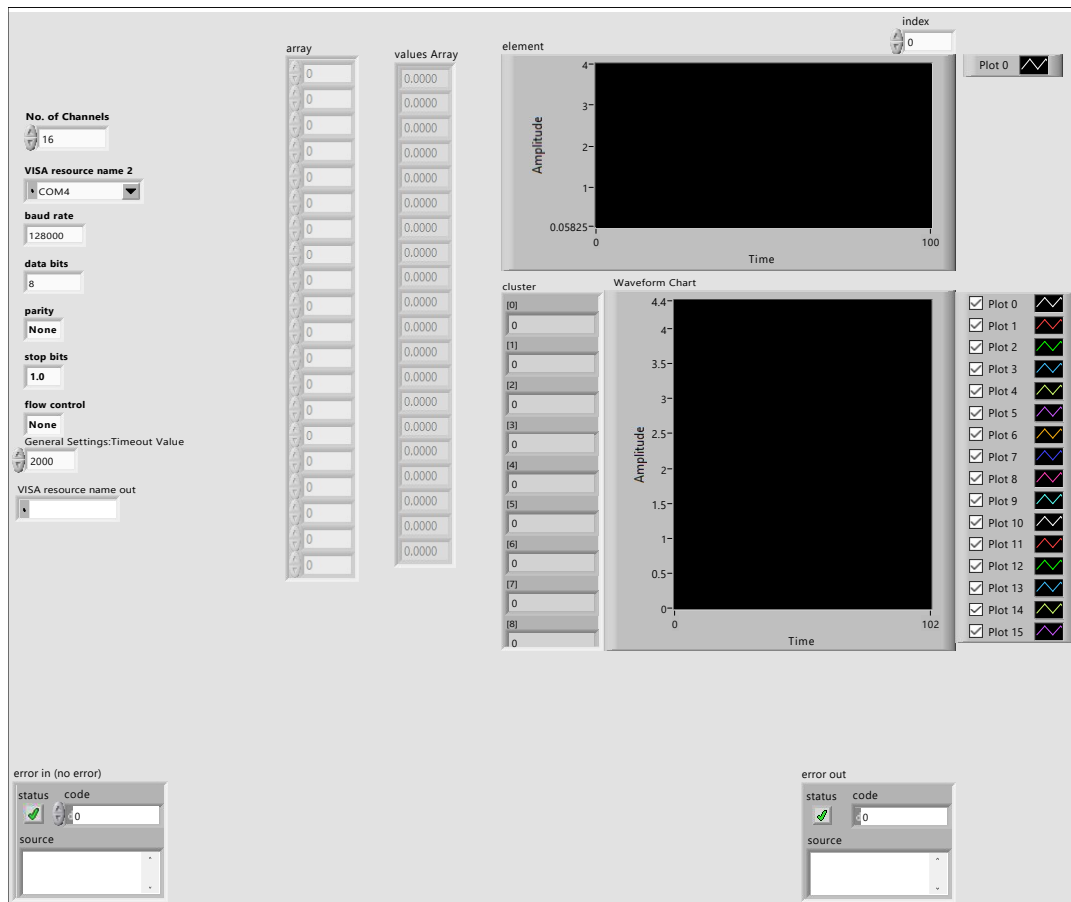


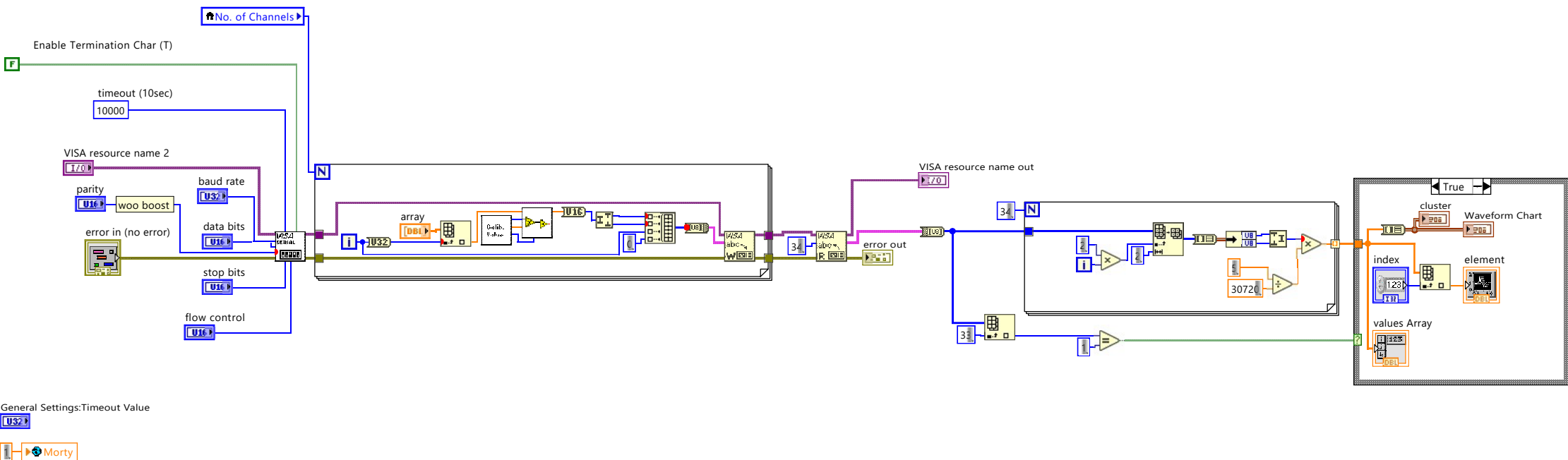




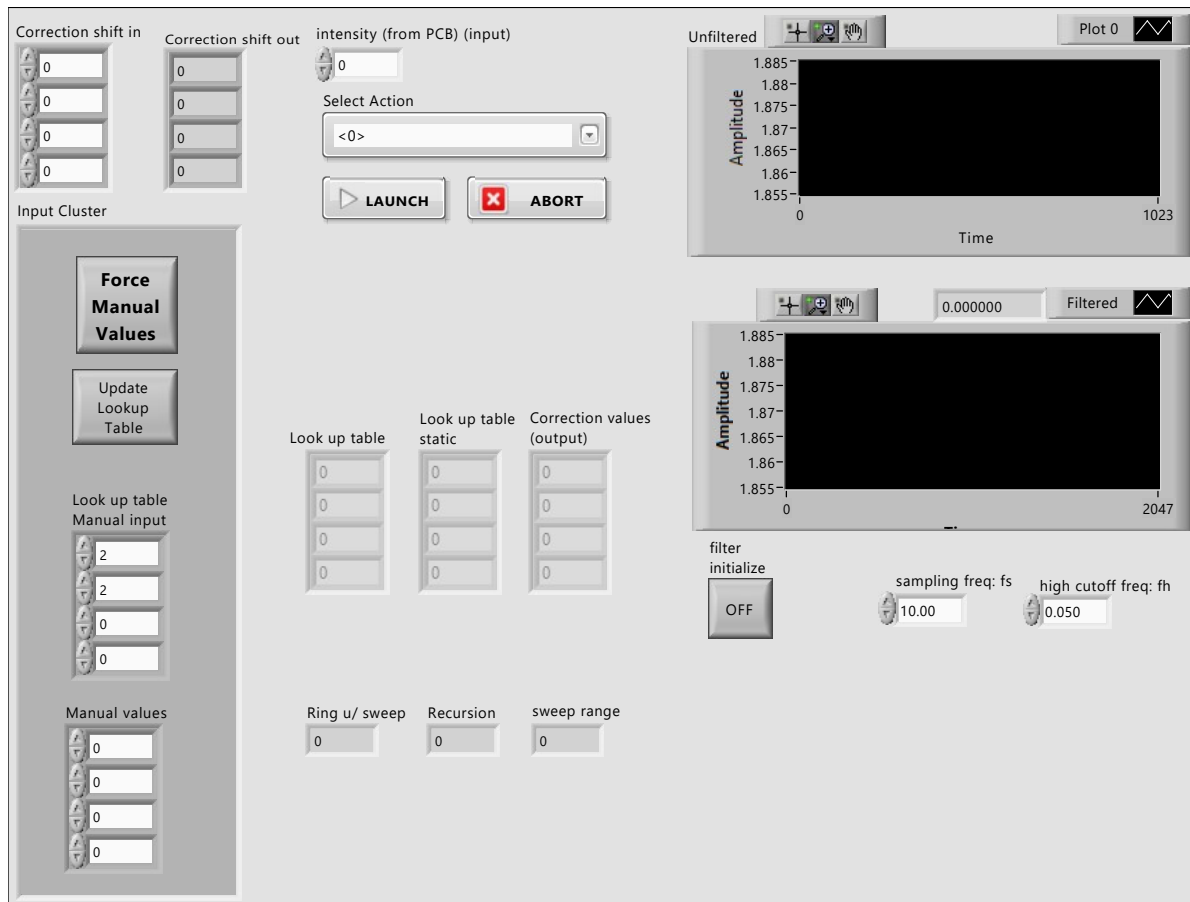


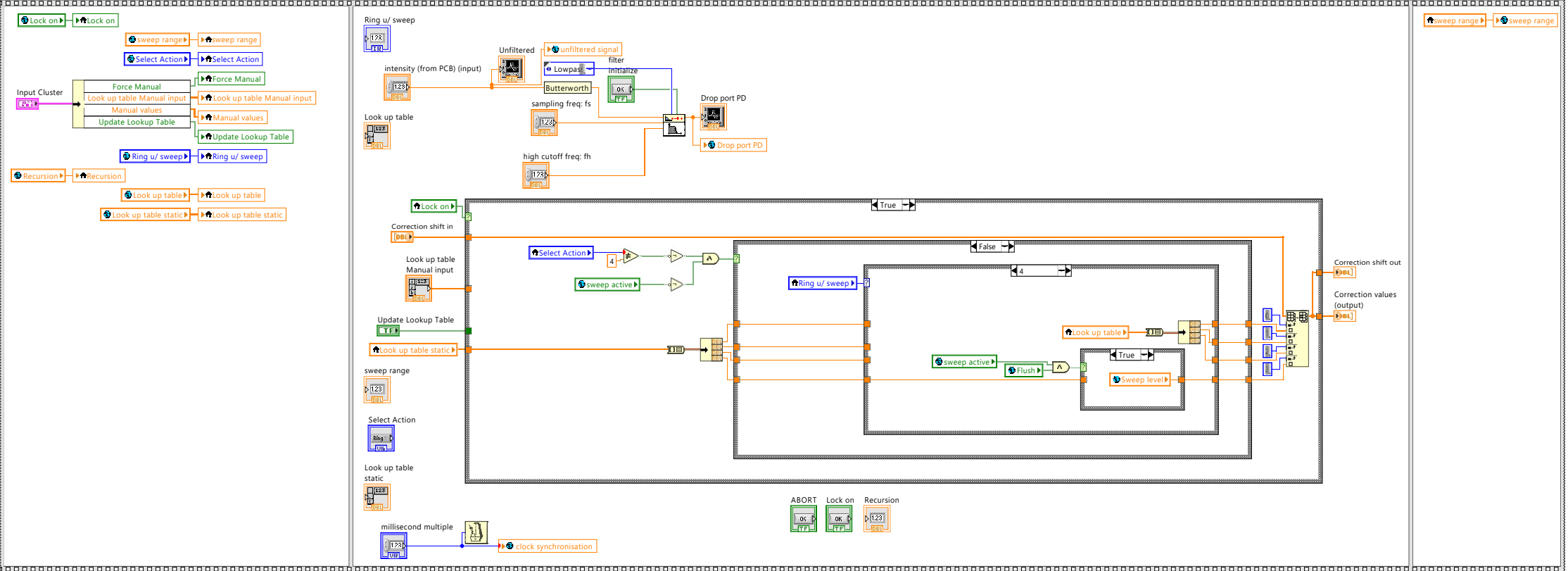
A.2 Interface VI



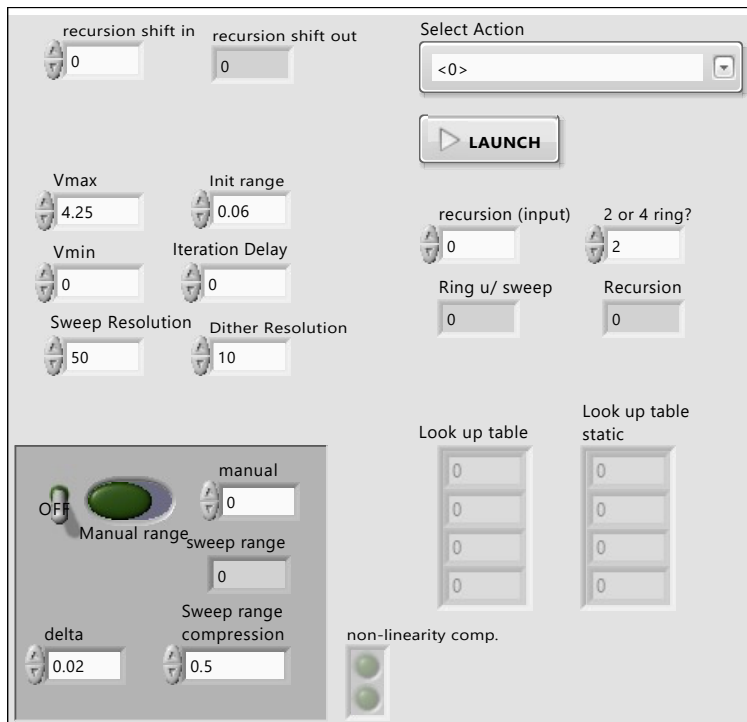


A.3 Handover VI



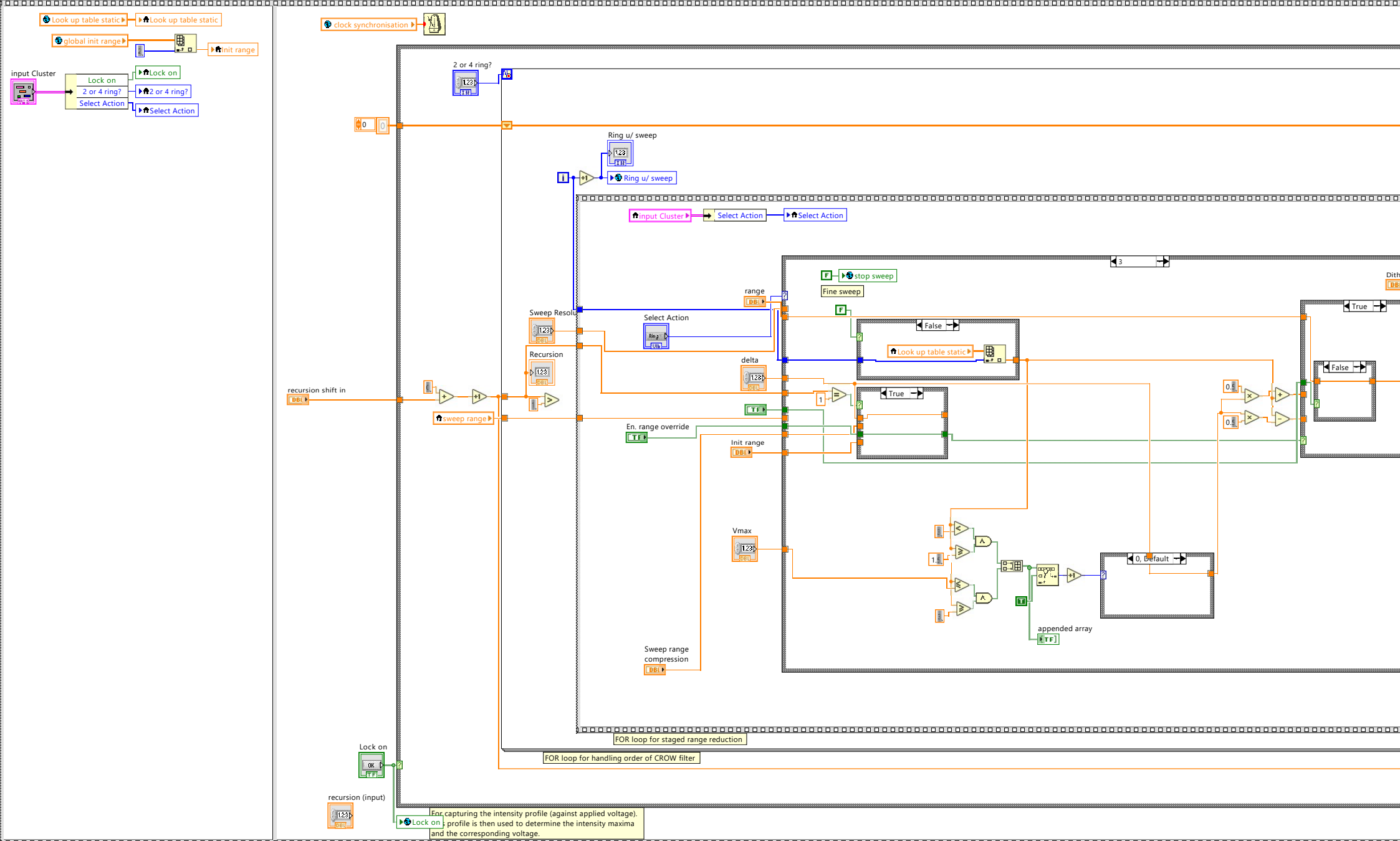


A.4 Algorithm VI

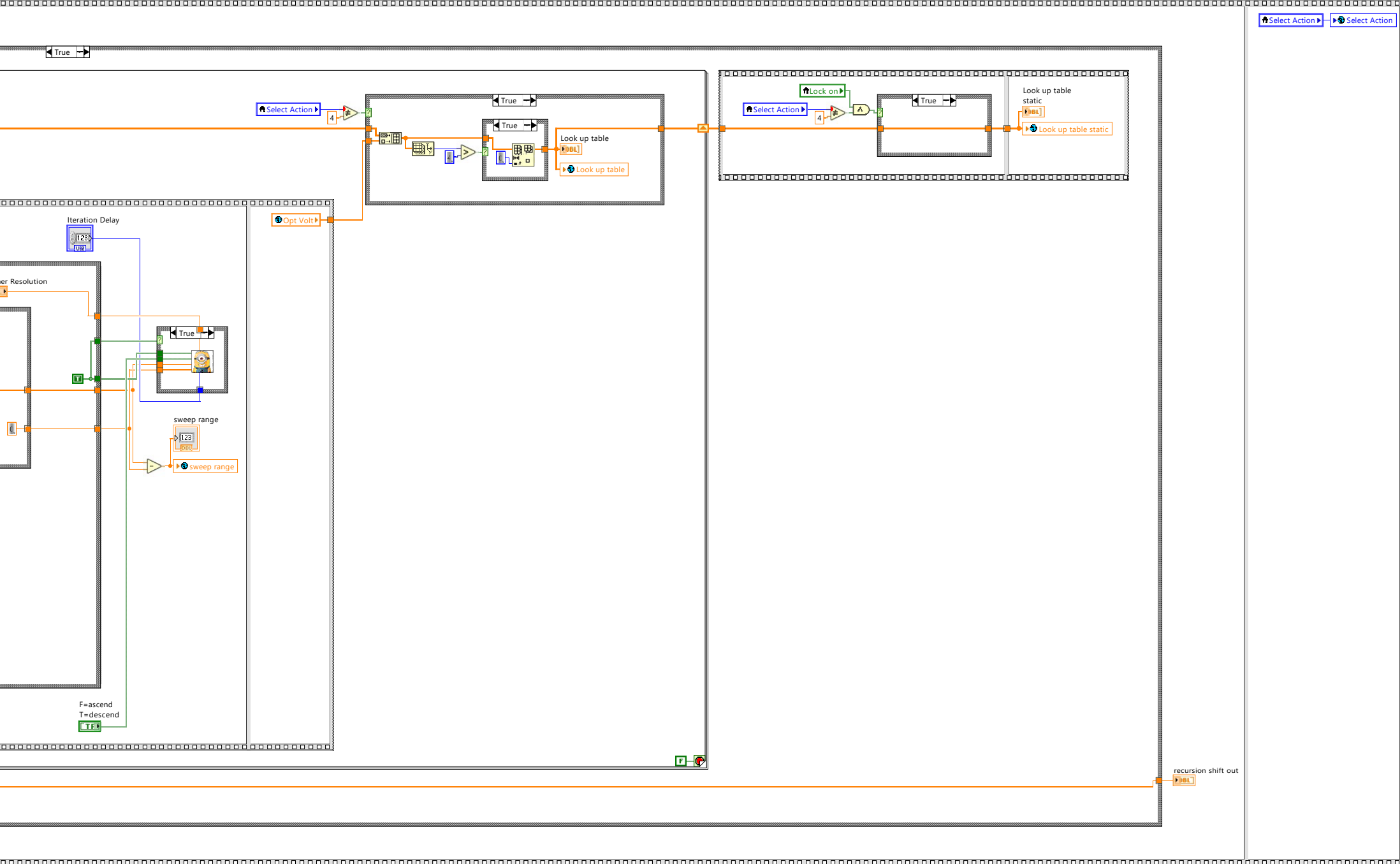


The LabVIEW front panel for Algorithm VI contains the following controls and indicators:

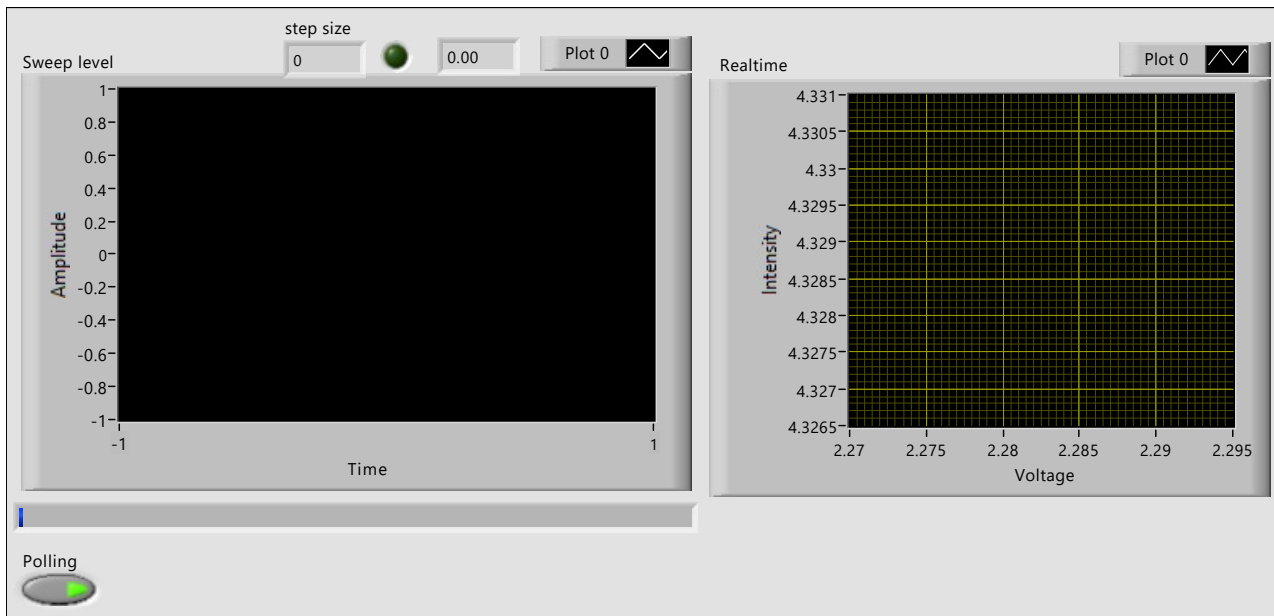
- recursion shift in**: A numeric display set to 0.
- recursion shift out**: A numeric display set to 0.
- Select Action**: A dropdown menu currently showing "<0>".
- LAUNCH**: A button with a play icon.
- Vmax**: A numeric display set to 4.25.
- Init range**: A numeric display set to 0.06.
- Vmin**: A numeric display set to 0.
- Iteration Delay**: A numeric display set to 0.
- Sweep Resolution**: A numeric display set to 50.
- Dither Resolution**: A numeric display set to 10.
- recursion (input)**: A numeric display set to 0.
- 2 or 4 ring?**: A numeric display set to 2.
- Ring u/ sweep**: A numeric display set to 0.
- Recursion**: A numeric display set to 0.
- Look up table**: A vertical stack of four numeric displays, all set to 0.
- Look up table static**: A vertical stack of four numeric displays, all set to 0.
- non-linearity comp.**: A vertical stack of two green indicator lights.
- Manual range**: A sub-panel containing:
 - manual**: A numeric display set to 0.
 - Manual range**: A numeric display set to 0.
 - delta**: A numeric display set to 0.02.
 - Sweep range compression**: A numeric display set to 0.5.



For capturing the intensity profile (against applied voltage).
Lock on profile is then used to determine the intensity maxima
and the corresponding voltage.



A.5 Main Sweep VI





List of Figures

1.1	A second order CROW filter with coupling coefficients κ_1 between straight waveguide and ring resonator and κ_2 between the two ring resonators. [41]	2
1.2	A figure	3
1.3	An infinitely long chain of coupled <i>ring resonators</i> , with forward and backward propagating field components. [12]	4
3.1	The quasi-Gaussian profile of the optical intensity obtained during a current sweep is shown here. The same range of current change induces different shifts in optical intensity depending on how close the CROW is to its resonant frequency. When dithering of current is carried out at the maxima, the disturbance to the optical intensity is minimum.	15
3.2	The data from the hardware is collected by the producer and is enqueued within it. The consumer takes data from the queue and processes it.	16
3.3	The hierarchy of the different modules/VIs. The <i>Main VI</i> is what the user interacts with. The subVIs <i>Interface VI</i> , <i>Handover VI</i> and <i>Algorithm VI</i> are nested within it. The <i>Main Sweep VI</i> is nested within the <i>Algorithm VI</i> . The global variables are accessible from any level of the hierarchy and thus not indicated as a branch in the hierarchy tree.	18
3.4	This diagram is for understanding how and when the different VIs run. It is based on general timing diagrams. A high state here represents the running VIs. All the VIs (and subVIs) run in parallel except the <i>Main Sweep VI</i> , which is called every time the algorithm is called or <i>launched</i> . For the sake of simplicity, the diagram is used to illustrate the case when the program is used for a 1 st order OADM. If a mode is kept enabled for more than one cycle, the <i>Algorithm VI</i> and <i>Main Sweep VI</i> will repeat their cycles until the mode is switched off. All the modes use the <i>Main Sweep VI</i> but with different settings.	19
3.5	Data flow between the various subVIs and global variables	20
3.6	State machine diagram of the algorithm.	22
4.1	Schematic of system	24
4.2	Loop timing measurement within each while loop of the VI.	24
4.3	The flag moves from the <i>Interface VI</i> to the <i>Main Sweep VI</i> . Inside the sweep VI, it terminates the polling loop, allows the incrementation of the sweep and then resets itself before being sent back to <i>Interface VI</i> .	25

4.4	The main sweep is executed only when the flag matches the assigned value allowing execution to move to the second frame. The whole frame is contained in a for loop which executes for a number of times that is equal to the resolution of the sweep. The “frame” is known as <i>sequence</i> structure in LabVIEW.	25
4.5	The pulse train (red in (a) & (b), yellow in (c) & (d)) applied to the heaters and the intensity response of the chip (black in (a) & (b) and green in (c) & (d)). The shift in the signals seen in (a) was found out to be due to larger execution time of function used to read the ADC.	26
4.6	The spectrum shifts linearly with changes in the chip temperature. In figure (a), we can see the shift of the resonant frequency of one of the two-ring CROWs with respect to temperature. Figure (b) shows the spectrum of the drop-port of the two-ring CROW over five different chip temperatures.	27
4.7	The <i>stacked sequence</i> structure in LabVIEW was used to sequentially go through the control of each CROW filter.	28
4.8	The modifications made to the code to ensure that all arrays that were used to stream data were at a constant size and did not grow in an unbounded manner.	29
4.9	Detuning snippet	30
4.10	The solution used to counter the effects of non-linearity depicted as a LabVIEW code-snippet.	31
4.11	The quadratic relationship between the resonant frequencies of the first and second ring of a two ring CROW filter versus the current applied to their heaters.	31
4.12	Data exchange between hardware and LabVIEW through ADE using NI-VISA.	32
4.13	The read data is accepted into the algorithm only when the flag matches the assigned value (here, the integer 1)	33
5.1	A comb source is injected to the input of the chip through a grating coupler. Four consecutive comb lines ($\lambda_1, \lambda_2, \lambda_3$ and λ_4) are selected with the 2 nd order CROW OADMs. λ_1 and λ_3 are combined with an MMI and sent into the first IQ modulator. At its output, an MMI is used to split the modulated carrier into two streams. The first stream has a 4 th order CROW OADM that drops λ_3 while the second stream allows the introduction of a 300ps or a 500ps time delay before another 4 th order CROW OADM drops λ_1 . Both λ_1 and λ_3 are dropped into a common waveguide. Similarly, λ_2 and λ_4 are also dropped into another common waveguide. Both the common waveguides are then combined with an MMI and sent to an output grating coupler. (Inset) The silicon system chip.	36
5.2	The control board with the various sections labeled, namely, the TIA (OPA2380) and ADC (ADS1158) sections that picked up the signal from the monitoring photodiodes, the DAC (MAX5735) and driver (OPA4251) sections which generated the actuator signal for driving the heaters and the microcontroller (ATxmega256A3B-AU) that was used to control the peripherals and communicate with a computer that ran LabVIEW.	37

5.3	A tunable laser was used to inject the carrier into the chip with a fiber array. The through port of the 2 nd order CROW OADMs, drop port of 4 th order CROW OADM and system out were connected to power meters as required, via other ports of the same fiber array. The inputs and outputs of the control board were connected to the pin headers of the chip carrier. The chip carrier had a TEC mounted on it which was controlled by a temperature controller.	38
5.5	For the second phase of the measurements, the tunable laser was replaced with a comb source. An optical filter and an EDFA were added to the output of the comb source.	39
5.6	The CROW filters used for testing the algorithm.	39
5.7	(a) The split in the through port spectrum, (b) the corresponding drop port and (c) the measurement carried out to crosscheck the voltages that produced the spectrum in Fig:(a).	40
5.8	The through port and drop port spectra of two of the 4th order CROW filters. .	41
5.9	(a) The through port and drop port spectra of the 2nd order OADMs after manual alignment to the comb lines. The manual alignments were carried out by using the tunable laser to mimic the comb lines. Thus, the comb spectrum shown here is only for reference. (b) The same spectrum after using the comb source and the algorithm for alignment.	42
5.10	(a) The drop ports of the 4 th order OADM CROWs for λ_1 and λ_3 , (b) The drop ports of the 4 th order OADM CROWs for λ_2 and λ_4 . These correspond to the alignments done with the algorithm.	43
5.11	Results of the alignment of the four channels of the NLFT system chip with the first delay line in service and the second delay line moved away to prevent interference.	44
5.12	Results of the alignment of the four channels of the NLFT system chip with the second delay line in service and the first delay line moved away to prevent interference.	45

List of Tables

- 5.1 (a) The first disturbance rejection test was carried out with a temperature disturbance of $\pm 1^{\circ}\text{C}$ and (b) the second test with temperature disturbance of $\pm 2^{\circ}\text{C}$.
Speed of control algorithm is the duration before a loop goes to its next iteration. 46
- A.1 Legend to the LabVIEW Hierarchy diagram. 49

Bibliography

- [1] Antonio Canciamilla, M Torregiani, Carlo Ferrari, Francesco Morichetti, RM De La Rue, A Samarelli, M Sorel, and A Melloni. Silicon coupled-ring resonator structures for slow light applications: potential, impairments and ultimate limits. *Journal of Optics*, 12(10):104008, 2010.
- [2] J Komma, C Schwarz, G Hofmann, D Heinert, and R Nawrodt. Thermo-optic coefficient of silicon at 1550 nm and cryogenic temperatures. *Applied Physics Letters*, 101(4):041905, 2012.
- [3] Mack W Riley, James D Warnock, and Dieter F Wendel. Cell broadband engine processor: Design and implementation. *IBM Journal of Research and Development*, 51(5):545–557, 2007.
- [4] Jie Teng, Pieter Dumon, Wim Bogaerts, Hongbo Zhang, Xigao Jian, Xiuyou Han, Mingshan Zhao, Geert Morthier, and Roel Baets. Athermal silicon-on-insulator ring resonators by overlaying a polymer cladding on narrowed waveguides. *Optics express*, 17(17):14627–14633, 2009.
- [5] Biswajeet Guha, Alexander Gondarenko, and Michal Lipson. Minimizing temperature sensitivity of silicon mach-zehnder interferometers. *Optics express*, 18(3):1879–1887, 2010.
- [6] Kishore Padmaraju, Johnnie Chan, Long Chen, Michal Lipson, and Keren Bergman. Thermal stabilization of a microring modulator using feedback control. *Optics express*, 20(27):27999–28008, 2012.
- [7] William A Zortman, Anthony L Lentine, Douglas C Trotter, and Michael R Watts. Bit-error-rate monitoring for active wavelength control of resonant modulators. *IEEE Micro*, 33(1):42–52, 2012.
- [8] Avizit Basak, Md Zargis Talukder, and Md Rakibul Islam. Performance analysis and comparison between coarse wdm and dense wdm. *Global Journal of Research In Engineering*, 2013.
- [9] ITU-T. *Spectral grids for WDM applications: DWDM frequency grid*.
- [10] Dominik Gerhard Rabus. *Realization of optical filters using ring resonators with integrated semiconductor optical amplifiers in GaInAsP/InP*. PhD thesis, Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, 2002.

- [11] Michael L. Cooper, Greeshma Gupta, Mark A. Schneider, William M. J. Green, Solomon Assefa, Fengnian Xia, Yurii A. Vlasov, and Shayan Mookherjea. Statistics of light transport in 235-ring silicon coupled-resonator optical waveguides. *Optics Express*, 18(25):26505, dec 2010.
- [12] Joyce KS Poon, Jacob Scheuer, Shayan Mookherjea, George T Palocz, Yanyi Huang, and Amnon Yariv. Matrix analysis of microring coupled-resonator optical waveguides. *Optics express*, 12(1):90–103, 2004.
- [13] Shankar Kumar Selvaraja, Wim Bogaerts, Pieter Dumon, Dries Van Thourhout, and Roel Baets. Subnanometer linewidth uniformity in silicon nanophotonic waveguide devices using cmos fabrication technology. *IEEE Journal of Selected Topics in Quantum Electronics*, 16(1):316–324, 2009.
- [14] Miloš A Popović, Christina Manolatou, and Michael R Watts. Coupling-induced resonance frequency shifts in coupled dielectric multi-cavity filters. *Optics express*, 14(3):1208–1222, 2006.
- [15] Qing Fang, Junfeng Song, Xianshu Luo, Lianxi Jia, Mingbin Yu, Guoqiang Lo, and Yuliang Liu. High efficiency ring-resonator filter with nisi heater. *IEEE Photonics Technology Letters*, 24(5):350–352, 2011.
- [16] Payam Alipour, Amir Hossein Atabaki, Ali Asghar Eftekhari, and Ali Adibi. Athermal performance in titania-clad microresonators on soi. In *Frontiers in Optics*, page FThQ6. Optical Society of America, 2010.
- [17] Kishore Padmaraju and Keren Bergman. Resolving the thermal challenges for silicon microring resonator devices. *Nanophotonics*, 3(4-5):269–281, 2014.
- [18] Yasuo Kokubun, Norihide Funato, and Masanori Takizawa. Athermal waveguides for temperature-independent lightwave devices. *IEEE Photonics Technology Letters*, 5(11):1297–1300, 1993.
- [19] Vivek Raghunathan, N Ye Winnie, Juejun Hu, Tomoyuki Izuhara, Jurgen Michel, and Lionel Kimerling. Athermal operation of silicon waveguides: spectral, second order and footprint dependencies. *Optics express*, 18(17):17631–17639, 2010.
- [20] Jong-Moo Lee, Duk-Jun Kim, Hokyun Ahn, Sang-Ho Park, and Gyungock Kim. Temperature dependence of silicon nanophotonic ring resonator with a polymeric overlayer. *Journal of lightwave technology*, 25(8):2236–2243, 2007.
- [21] Milan M Milošević, Frederic Y Gardes, David J Thomson, and Goran Z Mashanovich. Temperature insensitive racetrack resonators for near infrared applications. In *Optical Fiber Communication Conference*, pages OTh4D–5. Optical Society of America, 2012.
- [22] Shiyang Zhu, GQ Lo, Jieli Xie, and DL Kwong. Towards athermal nanoplasmonic resonators based on cu-tio₂-si hybrid plasmonic waveguide. In *Optical Fiber Communication Conference*, pages OW3F–1. Optical Society of America, 2013.

- [23] Mutsunori Uenuma and Teruaki Motooka. Temperature-independent silicon waveguide optical filter. *Optics letters*, 34(5):599–601, 2009.
- [24] Biswajeet Guha, Bernardo BC Kyotoku, and Michal Lipson. Cmos-compatible athermal silicon microring resonators. *Optics express*, 18(4):3487–3493, 2010.
- [25] Sasikanth Manipatruni, Rajeev K Dokania, Bradley Schmidt, Nicolás Sherwood-Droz, Carl B Poitras, Alyssa B Apsel, and Michal Lipson. Wide temperature range operation of micrometer-scale silicon electro-optic modulators. *Optics letters*, 33(19):2185–2187, 2008.
- [26] Cheng Li, Rui Bai, Ayman Shafik, Ehsan Zhian Tabasy, Geng Tang, Chao Ma, Chin-Hui Chen, Zhen Peng, Marco Fiorentino, Patrick Chiang, et al. A ring-resonator-based silicon photonics transceiver with bias-based wavelength stabilization and adaptive-power-sensitivity receiver. In *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pages 124–125. IEEE, 2013.
- [27] John E Cunningham, Ivan Shubin, Xuezhe Zheng, Thierry Pinguet, Attila Mekis, Ying Luo, Hiren Thacker, Guoliang Li, Jin Yao, Kannan Raj, et al. Highly-efficient thermally-tuned resonant optical filters. *Optics Express*, 18(18):19055–19063, 2010.
- [28] F Gan, T Barwicz, MA Popovic, MS Dahlem, CW Holzwarth, PT Rakich, HI Smith, EP Ippen, and FX Kartner. Maximizing the thermo-optic tuning range of silicon photonic structures. In *2007 Photonics in Switching*, pages 67–68. IEEE, 2007.
- [29] Michael R Watts, William A Zortman, Douglas C Trotter, Gregory N Nielson, David L Luck, and Ralph W Young. Adiabatic resonant microrings (arms) with directly integrated thermal microphotronics. In *Conference on Lasers and Electro-Optics*, page CPDB10. Optical Society of America, 2009.
- [30] Po Dong, Wei Qian, Hong Liang, Roshanak Shafiiha, Ning-Ning Feng, Dazeng Feng, Xuezhe Zheng, Ashok V Krishnamoorthy, and Mehdi Asghari. Low power and compact reconfigurable multiplexing devices based on silicon microring resonators. *Optics express*, 18(10):9852–9858, 2010.
- [31] Po Dong, Wei Qian, Hong Liang, Roshanak Shafiiha, Dazeng Feng, Guoliang Li, John E Cunningham, Ashok V Krishnamoorthy, and Mehdi Asghari. Thermally tunable silicon racetrack resonators with ultralow tuning power. *Optics express*, 18(19):20298–20304, 2010.
- [32] Christopher T DeRose, Michael R Watts, Douglas C Trotter, David L Luck, Gregory N Nielson, and Ralph W Young. Silicon microring modulator with integrated heater and temperature sensor for thermal control. In *Conference on Lasers and Electro-optics*, page CThJ3. Optical Society of America, 2010.
- [33] Erman Timurdogan, Aleksandr Biberman, Douglas C Trotter, Chen Sun, Michele Moresco, Vladimir Stojanović, and Michael R Watts. Automated wavelength recovery for microring resonators. In *CLEO: Science and Innovations*, pages CM2M–1. Optical Society of America, 2012.

- [34] TW Hansch and B Couillaud. Laser frequency stabilization by polarization spectroscopy of a reflecting reference cavity. *Optics communications*, 35(3):441–444, 1980.
- [35] Kishore Padmaraju, Dylan F Logan, Jason J Ackert, Andrew P Knights, and Keren Bergman. Microring resonance stabilization using thermal dithering. In *2013 Optical Interconnects Conference*, pages 58–59. IEEE, 2013.
- [36] Edsger W. Dijkstra and DIJKSTRA EW. Information streams sharing a finite buffer. 1972.
- [37] Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE transactions on software engineering*, (2):125–143, 1977.
- [38] Jonas Koch, Alvaro Moscoso-Mártir, Juliana Müller, Florian Merget, Stephan Pachnicke, and Jeremy Witzens. Silicon photonics dwdm nlft soliton transmitter. *arXiv preprint arXiv:2004.03183*, 2020.
- [39] SEYED ALIREZA TABATABAEI MASHAYEKH. Tuning and controlling optical telecommunication channels based ring resonator modulators. Master’s thesis, ING - Scuola di Ingegneria Industriale e dell’Informazione, 2016/2017.
- [40] VN Mitryakhin, P Yu Shapochkin, MS Lozhkin, Z Hatzopoulos, A Tzimis, P Savvidis, and Yu V Kapitonov. Anticrossing of optical modes in coupled microcavities. In *Journal of Physics: Conference Series*, volume 1400, page 066032. IOP Publishing, 2019.

References of Figures

- [41] Lumerical, <https://support.lumerical.com/hc/en-us/articles/360051107554-Tunable-CROW-Filter>. *Tunable CROW Filter*.
- [42] Georgios N. Tsigaridas. A study on refractive index sensors based on optical micro-ring resonators. *Photonic Sensors*, 7(3):217–225, jun 2017.
- [43] Wim Bogaerts, Pieter Dumon, Dries Van Thourhout, Dirk Taillaert, Patrick Jaenen, Johan Wouters, Stephan Beckx, Vincent Wiaux, and Roel G Baets. Compact wavelength-selective functions in silicon-on-insulator photonic wires. *IEEE Journal of Selected Topics in Quantum Electronics*, 12(6):1394–1401, 2006.

Glossary

Symbols (glossaries)

2nd order CROW filter an OADM based filter that has two closed loop waveguide or ring resonators.. pages 5, 14

4th order CROW filter an OADM based filter that has four closed loop waveguide or ring resonators.. pages 5, 14

A

ADC section the section of the microcontroller that calls the read-from-ADC function and sends the data to LabVIEW. page 31

ADE Application Development Environment. pages 30, 31

Algorithm VI The Algorithm VI is used to switch between the different states of the algorithm.. pages 17–21, 61

B

BER bit-error-rate. page 11

C

CMOS complementary metal oxide semiconductor. page 9

CROW coupled-resonator optical waveguide. pages iii, 2, 5, 13–15, 20, 21, 23, 26, 28, 29, 35, 36, 38–45, 61, 62

D

DAC section the section of the microcontroller that receives the data from LabVIEW and calls the write-to-DAC function. page 31

delta the minimum setting for the range of the voltage sweep carried out by the Main Sweep VI.. page 21

Dither-tracking The Dither Tracking mode carries out micro-sweeps until the intensity sees no further increase.. pages 21, 28, 43

F

FSR free spectral range. pages 10, 14, 21, 29

G

Global variable-1 A variable used in the program to transfer data from one parallel loop to another.. pages 18–20

Global variable-2 A variable used in the program to transfer data from one parallel loop to another. In contrast to Global Variable-1, this is used between two levels of hierarchy.. pages 18–20

H

Handover VI The Handover VI transfer data between the Algorithm VI and Interface VI. It also filters the intensity signal and transfers it to the Main Sweep VI.. pages 17–20, 23, 28, 61

I

Initialization *Initialization* is a mode that takes over after the Search Laserline mode and conducts voltage sweeps over a smaller range and keeps reducing the range by a given factor until it reaches the minimum allowed range.. pages 21, 28

Interface VI The interface VI is the part of the program that communicates with the electronic hardware (the control board) used in our application.. pages 17–20, 31, 32, 61

L

look-up table The look-up table stores the voltage or current values that need to be applied to the optical filters to align them to the laser wavelength.. pages 13, 20, 28, 32, 43

M

Main Sweep VI The Main Sweep VI is the reconfigurable VI that is configured by each state of the algorithm. It carries out the voltage sweeps and maps the intensity response to the sweep data.. pages 17–20, 23–25, 28, 31, 32, 61

Main VI The VI which is used by the operator. It houses all the level 1 hierarchy loops.. page 18

MZM Mach Zehnder Modulator. pages 5, 9

N

NFT non-linear Fourier transform. page 5

nonlinear Fourier transform NLFT. page 35

O

OADM optical add-drop multiplexer. pages iii, 2, 17, 35, 36, 38–44, 62

P

PIC Photonic Integrated Circuit. page 26

R

ring resonator An optical ring resonator is a set of waveguides in which at least one is a closed loop coupled to some sort of light input and output.. pages 3, 4, 8–11, 13, 15, 61

S

Search Laserline Search Laserline is a mode of the algorithm that carries out fast, coarse voltage sweeps to find any variation in intensity and sets the ring to the corresponding voltage value.. page 21

SRR Single Ring Resonator. pages 3, 37

T

TDWS temperature dependent wavelength shift. pages 8, 9, 11

thermo-optic coefficient . page 9

TiO₂ titanium dioxide. page 9

V

VI Virtual Instrument, is the basic building block of programs written in the G Language. It is similar to a function or subroutine in other programming languages. It includes the Front Panel (containing Controls and Indicators), the Block Diagram (containing Control Terminals, Wires, and Structures and various other GObject nodes), the VI's Icon and its Connector Pane (as well as compiled executable code which is hidden behind the scenes).. page 15

W

wavelength-locking the process of aligning the passband of an optical filter like the ring resonator to a signal wavelength.. pages 13, 14, 21, 29, 32