Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

FlexSensor: Automated measurement software for rapid photonic circuits capturing

Christoph Schmidt [a,*], Jakob Wilhelm Hinum-Wagner [b], Reinhard Klambauer [a], Alexander Bergmann [a]

[a] *Graz University of Technology, Institute of Electrical Measurement and Sensor Systems, Graz, Austria*
[b] *ams-OSRAM AG, Tobelbaderstraße 30, Premstaetten, Austria*

ARTICLE INFO

Keywords:
Silicon photonics
Wafer-level measurements
Measurement automation
Graphical user interface

ABSTRACT

This paper introduces the automation measurement software *FlexSensor* for capturing resonant spectra, an innovative and extensible software program developed explicitly for measuring and evaluating wafer-level Silicon Photonic (SiPh) circuits. Wafer-level Silicon Photonics allows the integration of numerous optical components and structures on a single wafer. However, researchers and engineers need precise and repeatable measurements to characterize them and face significant challenges when dealing with large numbers of complex systems on a single wafer. A toolchain gap hampers the measuring of such highly integrated photonic structures: While the setup necessitates the integration of an optimized hardware and software toolchain, there is neither software nor a standardized way to implement a reproducible measurement routine for a massive set of measurements.

FlexSensor allows integration and control of external hardware (tunable lasers, analog–digital converters) and supports measurement data storage and evaluation. The software enables researchers and engineers to efficiently analyze the spectral response of photonic structures and facilitate rapid measuring.

Code metadata

Current code version	v6.0.0b
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-24-00065
Permanent link to Reproducible Capsule	–
Legal Code License	GNU General Public License (GPL) Version 3
Code versioning system used	git
Software code languages, tools, and services used	Python 3.10
Compilation requirements, operating environments	Python packages: matplotlib, numpy, pyside6, pyyaml, pandas, pyqtgraph, rich, scipy. See ReadMe in GitLab repository.
If available Link to developer documentation/manual	See ReadMe in GitLab repository.
Support email for questions	christoph.schmidt@tugraz.at

1. Motivation and significance

Wafer-level Silicon Photonic (SiPh) allows the integration of many optical components and structures/devices in the form of Photonic Integrated Circuits (PICs) on the same chip thanks to the sub-micrometer cross-section dimension [1] and enables to applications ranging from high-bandwidth digital communication, allowing the transmission terabits of data per second to sensing. Resonant devices such as wafer-level Microring-/Racetrack Resonators (MRRs) and Mach–Zehnder Interferometers (MZIs) are of particular interest. However, the complex nature of these components and the sub-micron scale, combined with the need for precise and repeatable measurements, presents significant challenges for researchers and engineers when measuring.

Traditional wafer-level manual measurements are time-consuming, work-intensive, prone to errors, and limit measurement scalability. In

* Corresponding author.
E-mail addresses: christoph.schmidt@tugraz.at (Christoph Schmidt), jakob.hinumwagner@ams-osram.com (Jakob Wilhelm Hinum-Wagner), reinhard.klambauer@tugraz.at (Reinhard Klambauer), alexander.bergmann@tugraz.at (Alexander Bergmann).

https://doi.org/10.1016/j.softx.2024.101879
Received 8 February 2024; Received in revised form 20 June 2024; Accepted 2 September 2024
Available online 10 September 2024

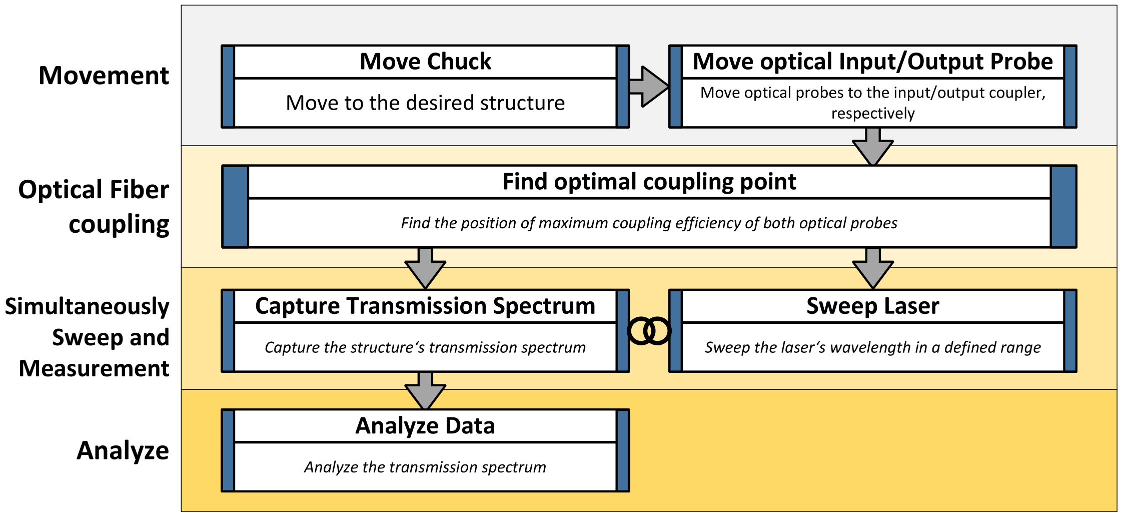

Movement
Move Chuck
Move to the desired structure
Move optical Input/Output Probe
Move optical probes to the input/output coupler, respectively
Optical Fiber coupling
Find optimal coupling point
Find the position of maximum coupling efficiency of both optical probes
Simultaneously Sweep and Measurement
Capture Transmission Spectrum
Capture the structure's transmission spectrum
Sweep Laser
Sweep the laser's wavelength in a defined range
Analyze
Analyze Data
Analyze the transmission spectrum

Fig. 1. Measurement methodology: After moving to the structure, the light is optically.

addition to the preliminary calibrations required prior to measurement (Section 2.4), the duration of the measurement tasks is influenced by multiple factors, including the number of measurements, required mechanical movements, and the desired wavelength sweep.

In theory, the time needed for one structure measurement can vary from a couple of seconds to a few minutes, and for hundreds of structures, these time can accumulate to hours of measurement time. The duration of one measurement depends on the length of the wavelength sweep and the movement of the chuck and probing tips. This time can, however, be prolonged due to various reasons:

- The chuck movement when traveling to the structure (Fig. 1, *Movement*) is usually relatively fast and depends mainly on the size of the photonic structure and the spatial distance to travel from one structure to another. For high distances (e.g., structures on different dies) and a lot of structures, this step can add up to a considerable amount of time.
- The alignment (Fig. 1, *Optical Fiber Coupling*) of the probing fiber tips is based on a threshold algorithm, where both tips execute high-speed XY scan motion, while simultaneously sampling the optical power meter as the probe moves [2]. This process usually takes only a couple of seconds. If no coupling point is found instantaneously, a light search has to be performed, which may vary from a couple of seconds to minutes, respectively.
- The wavelength sweep (Fig. 1, *Simultaneous Sweep And Measurement*) does not vary much in time and is almost entirely dependent on the sweeping speed in $\mathrm{nm/s}$. When using motorized tunable lasers, the tuning speed is defined by the system's mechanical capabilities and the capabilities of the Analog Digital Converter (ADC), used for sampling the analog data during the sweep.

The measurement methodology, however, follows a straightforward approach (Fig. 1): The emitted light from a laser, which is optically coupled to the component (e.g., using grating couplers), is transmitted, collected, and subjected to analysis. Grating couplers are very sensitive to input light polarization, and when not optimized, very low or no optical power is measured. This polarization dependency can be problematic, as it may lead to inaccurate measurements and characterization. Therefore, it is crucial for the user to pay attention to the polarization state of the input light to ensure reliable data. Then by scanning the wavelength range, the mode spacing of the optical devices can be identified as sharp peaks in the spectrum. The transmission spectrum of these structures can be acquired by precisely positioning optical fibers over the input and output of the resonant optical devices and sweeping over the wavelength using a tunable laser. Then, valuable information about the structure or the whole wafer can be derived by determining the spectral spacing within this acquired spectrum. This mode spacing makes it possible to determine the Free Spectral Range (FSR) in those resonant structures and to extract significant properties. Properties such as the group index n_g, the Full-Width-Half-Maximum (FWHM), and the extinction ratio can be calculated from this spectrum, allowing the characterization of waveguides [3]. Measuring hundreds of SiPh-devices leads to a growing demand for automated measurement techniques. In semiconductor manufacturing and research, *probe (stations) systems* ("wafer probers") [4–7] emerged as indispensable tools for measuring and assessing the properties of PICs at the wafer level. Exploiting the sub-micron accuracy in positioning possibilities [8] of those devices, they allow precise alignment of the probing tips with the devices on the wafer. Despite their remarkable versatility, certain probing systems have limitations regarding their Application Programming Interface (API), compatibility with external hardware components, and potential failures when operated manually. In cases where exhaustive probing becomes necessary, a manual approach can significantly impede productivity, hamper data collection efficiency, prolong the time required for comprehensive characterization, and allow the introduction of operating errors.

1.1. Significance and motivation

This paper introduces *FlexSensor*, a software program for automated wafer-level measurement using a *FormFactor Cascade Summit200* [5] probe system to capture resonant PIC device spectra. Accurate and repeatable measurement of photonic structures necessitates integrating a meticulously composed and optimized toolchain encompassing both software and hardware components. Fully integrated measurement systems often require additional expensive hardware, time, knowledge of the measuring system, and programming skills. This work focuses on the capabilities of *FlexSensor*, which allows automation of the measurement task and the integration and control of additional external devices (laser, data logger). The bare probing system does not typically offer such devices and the ability to include them directly in the measurement setup.

Commercially available systems such as [9] promise an automated measurement experience. However, for most measurement routines, additional automation scripts need to be developed nevertheless. Especially interfacing and using such external devices with these probing systems can be challenging without in-depth programming knowledge. Applications that control hardware often must be multi-threaded to manage concurrent hardware control tasks without blocking other

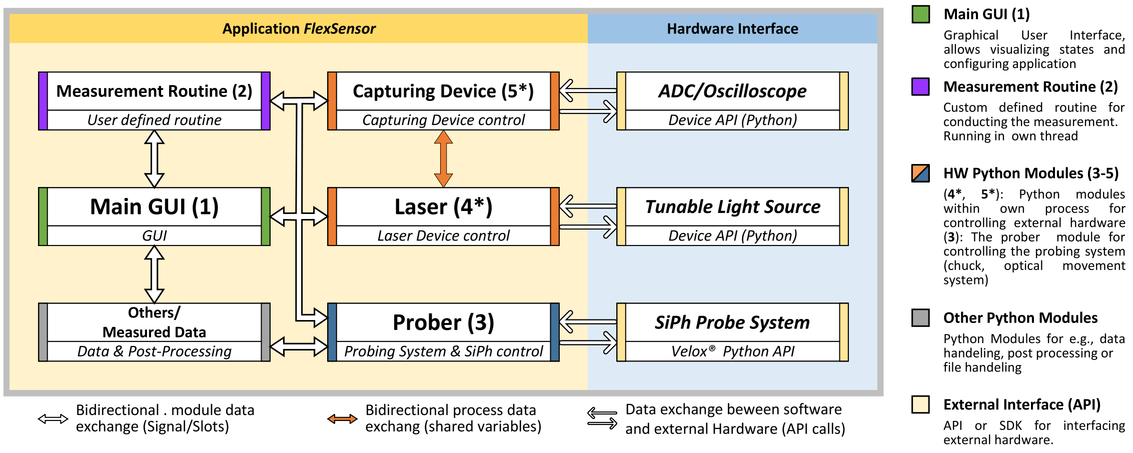

Application FlexSensor
Hardware Interface
Measurement Routine (2)
User defined routine
Capturing Device (5*)
Capturing Device control
ADC/Oscilloscope
Device API (Python)
Main GUI (1)
GUI
Laser (4*)
Laser Device control
Tunable Light Source
Device API (Python)
Others/ Measured Data
Data & Post-Processing
Prober (3)
Probing System & SiPh control
SiPh Probe System
Velox® Python API
Bidirectional . module data exchange (Signal/Slots)
Bidirectional process data exchang (shared variables)
Data exchange beween software and external Hardware (API calls)
Main GUI (1)
Graphical User Interface, allows visualizing states and configuring application
Measurement Routine (2)
Custom defined routine for conducting the measurement. Running in own thread
HW Python Modules (3-5)
(4*, 5*): Python modules within own process for controlling external hardware (3): The prober module for controlling the probing system (chuck, optical movement system)
Other Python Modules
Python Modules for e.g., data handeling, post processing or file handeling
External Interface (API)
API or SDK for interfacing external hardware.

Fig. 2. Core design of *FlexSensor*. The *Main GUI (1)* is the user interface, the *Measurement Routine* (2) interfaces the hardware modules (*probe station* with SiPh interface (3), *Laser (4*)*, *Capturing Device (5*)*) when a measurement is running. Modules (4*), (5*), and the *Measurement Routine (2)* are designed for easy replacement by custom implementations or other hardware interfacing classes, thus allowing a very adaptable use.

hardware. Multi-threading or multiprocessing allows the application to execute multiple threads or processes simultaneously, enabling parallel processing tasks such as input/output operations, sensor data handling, and real-time control, which is crucial for responsive hardware control systems. In cases where extensive data processing is required, multiprocessing can be essential, although its implementation is not trivial. Users often face challenges regarding synchronization between the software or even blocking of software parts. Even interfacing the hardware and correct data processing can sometimes be problematic. This often requires the user to implement complex threading or multiprocessing applications, write custom interfaces for the hardware, and take care of file handling.

Therefore, *FlexSensor* helps to minimize those potential errors when operating the probing system and can be used without much prior knowledge.

The proposed software is written in Python®3 and offers an innovative and extensible Graphical User Interface (GUI), created using the *pyside6* library, which provides access to the complete *Qt 6.0+* framework [10]. The setup comprises different modules and is developed explicitly for measuring and evaluating PICs, leveraging the comprehensive functionalities offered by an integrated probing system. The code implemented in this software has been validated and tested in recent work [11–13].

The existence of similar applications [14,15], targeting a similar area of research, shows the importance of such measurement. Although other applications target a similar area of research, *FlexSensor* is the first open-source application to the author's knowledge that utilizes a highly specialized wafer probing system. Only probing systems of this kind enable leveraging the possibilities offered by wafer-level measurement and performing qualitative and quantitative sub-micron photonic measurements.

2. Software description

FlexSensor is a standalone Python®application. Its versatility enables immediate deployment, without prior programming knowledge. While its post-processing capabilities are particularly noteworthy, the software's core functionality is the use of the automated measurement routine to acquire the resonant spectra of photonic devices, integrated within a specialized measurement system, as detailed in Section 2.3.

2.1. Software architecture

The software purposely uses individual (python) modules (Fig. 2, each module marked by a number), most of them able to run separately, to make it as easy as possible to adapt and reuse the given code tree. Thanks to this modularity, parts of the software can easily be adapted or superseded by including other or new modules, software parts, or new measurement routines. This is particularly interesting when the user needs to include new external hardware not initially supported by this software (Section 2.3). The software has been written following the *Model View Controller (MVC)* paradigm and uses Qt's *signals and slots* mechanism to exchange data. Exceptions are the *Laser (4*)* and *Capturing Device (5*)* modules, which run parts of the code as separate processes and must be mentioned separately. This ensures that a mutual blocking of the *Laser (4*)* and the *Capturing Device (5*)* does not occur during the measuring process. The synchronization between these modules is implemented via *shared variables*, and the data is exchanged with the other modules via *signals and slots*. While *FlexSensor* features more modules, the most important modules (Fig. 2) are described in this section.

Main GUI Module (1). `MainWindow`: The "Main GUI" with its underlying main controller implementation is the user-facing component of *FlexSensor*. It is responsible for creating the GUI that users interact with to control and monitor measurements. This module provides a graphical interface for users to visualize real-time data and access various features of the application's external devices and functionalities (e.g., file and folder settings).

Measurement routine Module (2). `MeasurementRoutines`: The "Measurement Routine" module is the backbone of the *FlexSensor* application, providing the essential commands for collecting and processing data. Fig. 3 shows the standard measuring routine. This module encapsulates the core logic for orchestrating measurements, managing data acquisition, and performing required calculations or data transformations. It enables seamless integration of measurement instruments, such as the *Laser (4*)* and *Capturing Device (5*)*, facilitating efficient and reliable data collection for the specific application. This module allows the user to write or alter a measurement routine without needing prior complex connecting or initialization of the devices.

Prober Module (3). `Prober`: The "Prober" module acts as the interface to the probing system's API via the prober's operating software Velox®. Velox® comes preinstalled with the used probing system. The "Prober" module facilitates communication with the prober hardware (Fig. 4, e.g., chuck, Optical Movement System (OMS), scope), enabling precise positioning and control of the wafer and the probes. It abstracts the intricacies of prober control, providing an interface for the software to interact with the prober seamlessly.

For finding the optimal coupling point, the prober offers two *active alignment tools* for automatic alignment of the fiber tips: The *Area*

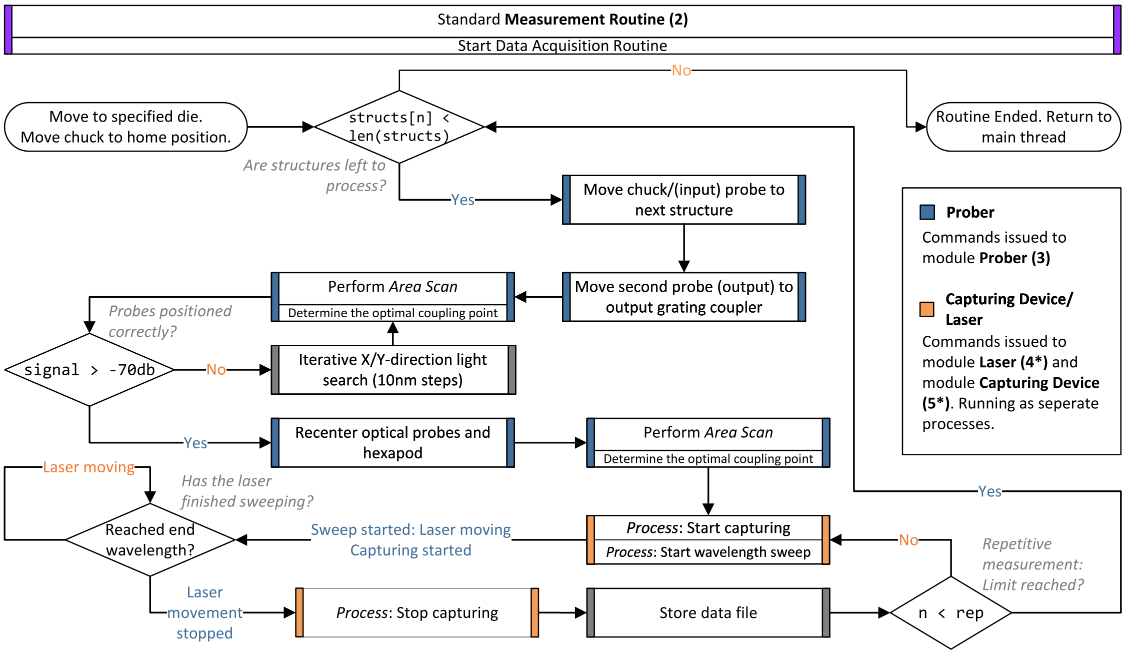

Standard Measurement Routine (2)
Start Data Acquisition Routine
Move to specified die. Move chuck to home position.
structs[n] < len(structs)
Are structures left to process?
No
Yes
Routine Ended. Return to main thread
Move chuck/(input) probe to next structure
Move second probe (output) to output grating coupler
Perform Area Scan
Determine the optimal coupling point
Probes positioned correctly?
signal > -70db
Iterative X/Y-direction light search (10nm steps)
Recenter optical probes and hexapod
Laser moving
Has the laser finished sweeping?
Reached end wavelength?
Sweep started: Laser moving Capturing started
Process: Start capturing
Process: Start wavelength sweep
Repetitive measurement: Limit reached?
n < rep
Laser movement stopped
Process: Stop capturing
Store data file
Prober
Commands issued to module Prober (3)
Capturing Device/ Laser
Commands issued to module Laser (4*) and module Capturing Device (5*). Running as seperate processes.

Fig. 3. Standard Measurement Routine. After processing the input files (at application startup), and running the initial probe calibration (by the user), the (input and output) probes are positioned accordingly, and the measurement is started. Using the standard measurement routine allows the caption of the transmission spectrum of an photonic devices. The measurement routine presented in this figure are evaluated in Section 3.

Scan and the *Gradient Scan*. The software *FlexSensor* interfaces with Velox® and its API (Fig. 4, *Hardware* and *API*) and uses the feature *Area Scan* for optimally aligning the structure. The *Area Scan* is used to identify the optimal coupling point on the device. During the scan, the *Nanocube* (OMS) (Fig. 5(a)) execute high-speed XY scan motion, while simultaneously sampling the optical power meter as the probe moves. This feature uses single-frequency sinusoid and spiral scans, which are much faster than raster or serpentine scans [16]. An algorithm and precise mechanics allow determining the exact peak quickly, even for multiple occurring peaks in the power spectrum. The power coupling data is associated with *Nanocube* position at each XY point within the scan (usually $\mathrm{th}_{\mathrm{pwr}} > -75\,\mathrm{dB}$), and the XY position where the peak power occurred is defined as the coupling point. The power threshold of this scan is used to differentiate between a coupling point above the noise floor $\mathrm{th}_{\mathrm{pwr}}$ and a point that does not contain a coupling signal [2,16]. In cases where no optimal coupling point can be found, *FlexSensor* will omit these structures. Usually, if the Device under Test (DUT) is malfunctioning, meaning under normal operating circumstances, no or insufficient light is transmitted, the *Area Scan* and thus *FlexSensor* will fail to find a coupling point. The software *FlexSensor* only conducts measurement if a coupling point can be found, i.e., light can be transmitted through the structure. If *FlexSensor*'s light search algorithm (utilizing *Area Scan*) fails to find a coupling point, the software will continue by omitting the structure and containing with the next one. If the polarization of the light and the grating coupler does not match or the user has chosen an off-resonance port (drop port) *FlexSensor* will behave as if the device is malfunctioning. At this stage of development, *FlexSensor* is not capable of distinguishing between an erroneous configuration, such as off-resonance or off-polarization, or malfunctioning devices. The author however suggest, that the software can be extended by a polarization controller, which allows to programmatically control of the polarization.

Laser Module (4).* `Laser`: The module provides the laser's control GUI and serves as the controller for the laser (*Sacher TEC diode laser* [17]), integrated into the measurement setup (Fig. 4). This module leverages a C++-to-Python API-interface provided by *Sacher Lasertechnik GmbH*,[1] facilitating seamless integration into the measurement process. It provides the necessary functions and commands to control the laser, allowing the application to sweep the wavelength precisely during measurements.

The driver encapsulates the functionalities inherent in the API, adopting a Pythonic paradigm to execute operations such as wavelength adjustment and connection management. To ensure a non-blocking and correct measurement, the module performs the sweeping process within a separate (multi-)process to prevent sample losses. An important aspect of this driver is its inherent connection and tight coupling (marked by the *) to the code of the associated *Capturing Device* (5*); initiating and stopping a wavelength sweep automatically starts and stops the capturing process of the *Capturing Device* (5*), enabling efficient and precisely controlled data acquisition.

The speed of the setup presented in this paper (Section 3) is limited to 10 nm/s. For systems with higher movement speed, the user should consider that too fast movement may result in vibrations and, thus, error-prone measurements.

Capturing device Module (5).* `AD2CaptDevice`: This module provides the device's control GUI and functions to interface and control the USB-oscilloscope Digilent Analog Discovery 2 [18] used as analog-to-digital data acquisition hardware. The device is connected to an optical power meter and reads and converts the analog power values from the Optical Power Meter (OPM) to digitalized values. Similar to the laser module, the capturing operates within its own (multi-)process to ensure non-blocking measurements. Additionally, the module maintains a continuous data stream of the read power level to the GUI, but only captures data in response to specific requests initiated by either the *Laser* (4*) module or the user. Moreover, it allows users to adjust settings (e.g., sample rate).

[1] The API is not publicly available. Thus, it needs to be requested by the user.

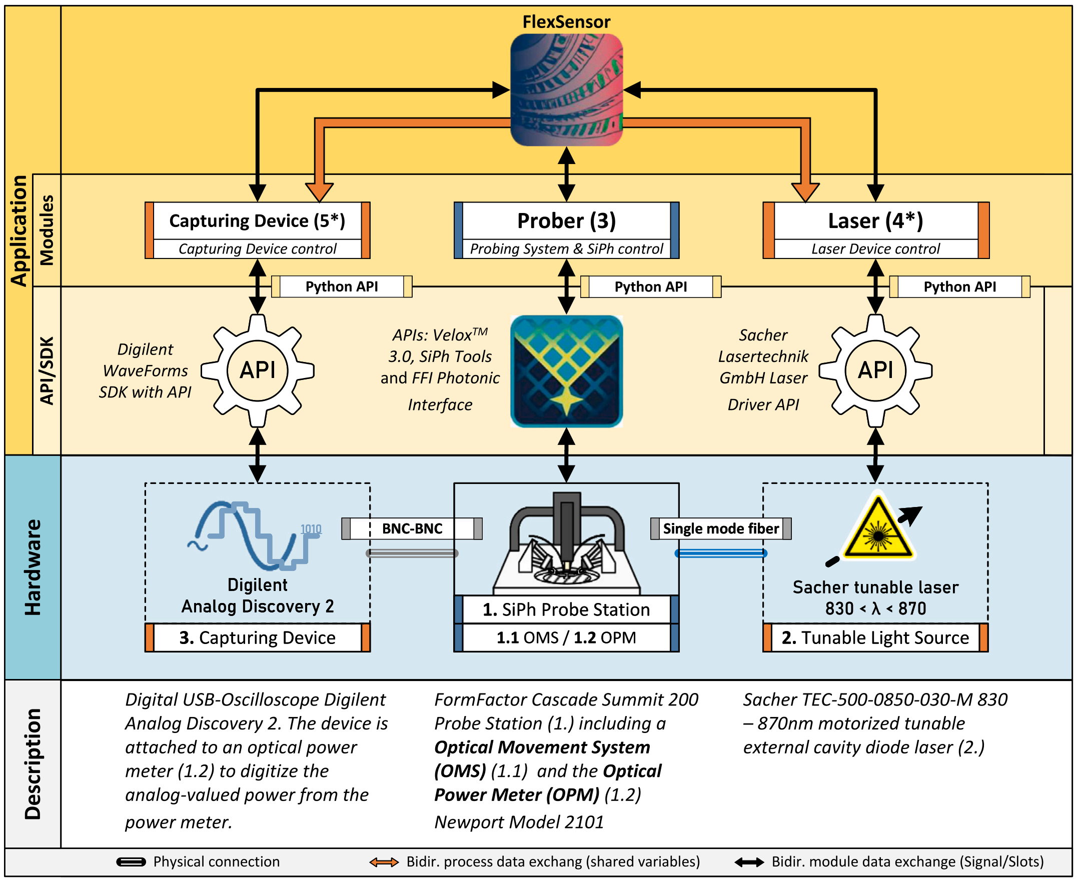
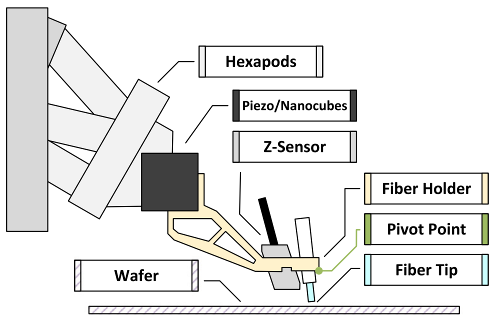
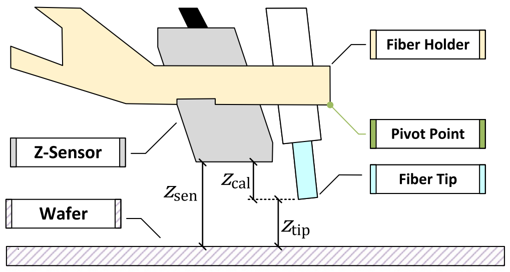

FlexSensor
Application
Modules
API/SDK
Hardware
Description
Capturing Device (5*)
Capturing Device control
Prober (3)
Probing System & SiPh control
Laser (4*)
Laser Device control
Python API
Python API
Python API
Digilent WaveForms SDK with API
API
APIs: Velox™ 3.0, SiPh Tools and FFI Photonic Interface
Sacher Lasertechnik GmbH Laser Driver API
API
Digilent Analog Discovery 2
3. Capturing Device
BNC-BNC
1. SiPh Probe Station
1.1 OMS / 1.2 OPM
Single mode fiber
Sacher tunable laser 830 < λ < 870
2. Tunable Light Source
Digital USB-Oscilloscope Digilent Analog Discovery 2. The device is attached to an optical power meter (1.2) to digitize the analog-valued power from the power meter.
FormFactor Cascade Summit 200 Probe Station (1.) including a Optical Movement System (OMS) (1.1) and the Optical Power Meter (OPM) (1.2) Newport Model 2101
Sacher TEC-500-0850-030-M 830 – 870nm motorized tunable external cavity diode laser (2.)
Physical connection
Bidir. process data exchang (shared variables)
Bidir. module data exchange (Signal/Slots)

Fig. 4. Measurement setup for evaluating the resonant spectra. The (**1.**) SiPh Probe Station is equipped with an Optical Movement System (OMS) (**1.1**) and an Optical Power Meter (**1.2**).
The setup consists of a (**3.**)-Capturing Device attached to the (**1.2**)-Optical Power Meter, via a *BNC-to-BNC* connection.
The (**2.**)-Tunable Light Source connects via a *single mode fiber* to the (**1.**)-SiPh Probe Station [9]. The measurement process is centrally controlled by *FlexSensor*.



(a) The Optical Movement System (OMS) of the probe station and with the *fiber tip holder* (Fig. 5b). The fiber tip can be precisely positioned using the Hexapods and the Piezo/Nanocubes. The pivot point must be calibrated before using the probe station.



(b) Schematic drawing of the *fiber tip holder*. The fiber is mounted using a *fiber tip holder* with an integrated capacitive sensor (Z-Sensor). Using the prober's calibration scheme (Step 1), the distance z_cal between fiber height z_{tip} and sensor height z_{sen} is calibrated.

Fig. 5. Optical Movement System (OMS) of the probe station.

2.2. Software functionalities

The application combines and tackles four significant aspects and offers advanced capabilities beyond plain spectra measurement. This section aims to elaborate on the major features offered by the software.

1. **Reducing potential errors by minimizing manual interaction (i.e., writing predefined measurement routines)**
   FlexSensor allows the definition of new and exhaustive measurement routines by utilizing pythonized API calls to the prober with extended error handling. Using the given possibilities allows for an extended measurement routine in which the most crucial steps, from positioning to hardware device setup and capturing can be handled. For straightforward management and configuration, a human-readable *yaml*-configuration file is used to customize *FlexSensor*. A supportive *vas*-file (see *Supplementary Information, Section 1*) for managing the structures to be measured is utilized to make measurements repeatable. The *vas*-file contains the relative coordinates measured from an (arbitrary) selected origin and is stored in the pythonic format. When using the relative formation, a initial software calibration 2.4 must be performed.
2. **Controlling other devices, such as tunable lasers and digital capturing devices**
   The prober system's software Velox® [19] enables precise positioning of the wafer and the fibers. However, establishing a measurement routine and incorporating additional hardware components, such as tunable laser systems or ADCs into the measurement setup is usually too complex to do manually. The *FlexSensor* measurement software facilitates this and interfaces with optical instruments, such as tunable lasers, optical power meters, and analog–digital converters (Fig. 4). It provides control and automation capabilities, allowing users to configure the measurement parameters, set the wavelength scanning range, and acquire spectral data from the capturing instruments.
3. **Allow directly include uncertainty information in the captured data (e.g., statistical evaluation)** Measurement and calculation results are stored in table-like *pandas* data frames, allowing powerful and flexible data analysis [20]. This also allows easy addition and adaptation of new or already implemented calculations and uncertainties, enabling quick export of the measurement with additional user-defined information, calculation, and uncertainty data in one file. The measurement can be directly exported in formats such as MATLAB `mat` or as `cvs`, allowing easy processing of the captured data.
4. **Facilitate and ease the development of future measurement routines (e.g., by including development tools)**
   The *FlexSensor* software also tackles the problem of an absent "prober simulator". Debugging is an unavoidable step toward a working routine during writing measurement routines. However, conducting these debugging activities on a live system without hardware simulators often requires more time and can pose potential risks. To illustrate, movement errors, wrong inputs, and misconfiguration of the system, for example, can damage the sensitive waveguide fibers or even the whole system. At worst, expensive part replacements and time-consuming re-calibration of the whole system will need to be done. This problem is addressed by including a simulator mode for the *FormFactor* prober system, allowing the development of routines without a real-world system. Therefore, the software allows users to validate their probing strategies before executing them on the hardware, saving time when developing new routines and anticipating potential pitfalls.

2.3. Suitable hardware

FlexSensor requires suitable hardware to run the measurement. Understanding the hardware prerequisites for running a software system is critical for its successful deployment and operation. The suggested measurement system for the software consists of three parts (Fig. 4 Hardware). A brief overview of the materials and devices (given below) used is given to make the experiments comprehensible. The SiPh-probing system (1. Probe Station) is the obligatory basis of the measurement system and includes the (1.1 Optical Movement System (OMS)) and an (1.2 Optical Power Meter (OPM)).

1. **Probe Station** *FormFactor Cascade Summit200* [5], including
   1.1 **Silicon Photonics Alignment Systems**
   OMS *Physik Instrumente H-811.S2 Hexapods* [8] and *Physik Instrumente P-616 NanoCube* [21]
   1.2 **Optical Power Meter**
   High Dynamic Range (HDR) optical power meter *Newport Model 2101* [22]
2. **Tunable Laser**
   e.g.,Tunable external cavity (TEC) diode laser
   Sacher TEC-500-0850-030-M [17]
3. **Data Acquisition Device**,
   e.g., a digital USB oscilloscope
   USB-Oscilloscope *Digilent Analog Discovery 2* [18]

The system is outlined in Fig. 4.

2.4. Prerequisites

The software *FlexSensor* need some prerequisites to be set before use.

Before using the software, a calibration of the system must be made. In this paper, the authors distinguish between the calibration of the underlying hardware and software system (Section 2.4.1, *system calibration*) and the specific initialization needed before using this software (Section 2.4.2, *software initialization*).

2.4.1. System calibration

The *system calibration* is the initial calibration of the measurement system (probe system, laser, ADC, optical power meter) and independent from *FlexSensor*, thus not directly related to the presented software. The *system calibration* involves all steps that are needed to initially calibrate the probe system, the laser, and the power meter. The calibration functions, e.g., compensate for aligning the motion axes of the optical positioners to the motion axes and coordinates of the probe station.

It should be noted that the needed system calibration depends on the measurement setup. Therefore, the actual calibration routine can vary. Thus, a rigorous description of the *system calibration* is outside of the scope of this paper. In this case, the user is referred to the manufacturer's calibration scheme.

The steps involve:

1. Calibrating the fiber distance (Z-Sensor) z_{cal} (Fig. 5(b))to establish the relationship between the Z-sensor signal (capacitive sensor) z_{sen} and the distance between the fiber tip and wafer z_{tip}.
2. Calibration of the chuck movement.
3. Calibration of the Piezo (Nanocubes) and the Hexapods.
4. Optical wafer alignment.

A brief overview of steps when calibrating the probe system can be found in the *Supplementary Information, Section 2 (Prerequisites)*.

Table 1
Waveguide parameters used. The parameters indicated were used for the group index calculation and the verifying simulation.

Parameter	Unit	Distribution	Value	Half bandwidth
Width	nm	Rect.	460	±20
Height	nm	Rect.	240.885	±20
Length 1	um	Rect.	$10 \cdot 10^-6$	$\pm 2.160 \cdot 10^{-3}$
Length 2	nm	Rect.	$10.36361 \cdot 10^-6$	$\pm 2.160 \cdot 10^{-3}$
Ratio (ΔL)	1	Triangular	1.036361	Combined

2.4.2. Software initialization

Prior to using the software, an initial initialization must be done. For this, the wafer is moved to the arbitrarily selected origin (*zero point*) with the chuck. The origin is the point from which all the coordinates in the *vas*-File (*Supplementary Information, Section 1 (Required Files)*) are based. The coordinates in the structure file rely on the relative positions measured from this point. After the Z-height has been calibrated (*System calibration*), the correct fiber-to-wafer height is adjusted using Velox® and stored. The height of the tips above the wafer's top surface (Fig. 5, z_{tip}) is measured using a capacitive distance sensor and is usually set to 50 nm. To ease the initialization task, *FlexSensor* implements a setup wizard, which guides the user through the whole process. The wizard is directly included in the software and makes sure that all mandatory steps are made.

1. Setting the original (zero point): The structure file relies on 'relative' positions to an original (zero point). The user manually selects a point and store the coordinates. From this position, the movement for every coordinate given in the *vas*-file (x, y) is calculated.
2. Define the current distance between the two probes: This is used to establish the spatial relation between the input fiber and the output fiber and a user-selected origin (zero point). Only if both fibers know the correct position to each other can the software function properly. By default, the probe system has no information about the relative position of both fibers to each other.

3. Illustrative example

This chapter aims to present a comprehensive analysis of a specific measurement and the corresponding measurement results using *FlexSensor*, showcasing a carefully designed measurement process (Fig. 3) that yields accurate and reproducible outcomes.

It aims to showcase the software's automation capabilities. For in-depth data analysis and discussion, the reader is referred to [13]. For statistical reasons, repeated measurements should be performed within a few days to encounter external parameters that can influence the measurement. This includes temperature, connector, and physical fiber positions, or polarization changes due to fiber stresses. An essential parameter for understanding light propagation in photonic circuits is the group index, denoted by n_g. It represents the speed of light within the photonic structure and determines the group velocity [3]. Accurate and efficient measurement of the group index is vital for optimizing the design and performance of photonic devices and is important for the channel spacing of wavelength filters and wavelength division multiplexers [23].

A 200 mm LPCVD silicon nitrite (Si_3N_4) photonic wafer was used for the experimental analysis, and a single Mach–Zehnder Interferometer (MZI, Table 1) was selected for investigation among the various test structures available.

All measurements were conducted using the automation software *FlexSensor*.

To ensure statistical significance and robustness in the analysis, the single measurements were repeatedly conducted 100 times fully automatically. Fig. 6 shows three of the 100 captured measurement results (top) and the evaluated *Free Spectral Range* of one measurement (bottom). The measurement encompasses the wavelength range from 830 to 870 nm, resulting in a transmission spectrum characterized by 43 peaks.

3.1. Wafer measurement

An important aspect to investigate is the behavior of a particular parameter across different positions on the wafer. The automated probing software *FlexSensor* also measured the transmission spectrum of selected devices across the entire wafer. Five different *Mach–Zehnder Interferometers* were selected to ensure robustness in the evaluation, each having different ratios in their arm length difference ΔL. Consequently, the final value of n_g is determined by combining the measurement results obtained from all five structures.

FlexSensor's measurement routine resulted in over 6000 independent measurements, each conducted fully automatically in approximately 8 h, demonstrating the automation capabilities of the application (see Fig. 7).

4. Impact

FlexSensor is designed for specific probing system hardware and offers researchers and engineers using this probing system software that significantly enhances the pursuit of existing research queries. By streamlining the measurement and evaluation processes, thereby expediting experimental cycles, it enables deeper insights into the behavior of photonic structures. By automating repetitive and time-consuming tasks, *FlexSensor* makes it possible to capture and analyze massive amounts of measurement data without the need for constant intervention by the user.

Accurately evaluating waveguide properties is pivotal in characterizing structures or the fabrication process. As researchers and engineers strive to develop efficient waveguide-based systems, it becomes imperative to understand the inherent uncertainties associated with the measurement process.

In practice, systematic and random deviations are always present for any measurement. When determining a measurement result, it is crucial to address and correct systematic deviations that may affect the accuracy of the measurement. These quantitative indications of the quality of the measurement are obligatory to assess a measurement's reliability [24]. By addressing the uncertainties associated with waveguide parameter evaluation, *FlexSensor* application helps to enhance the accuracy and reliability of waveguide characterization by providing statistical assessment tools. The insights from this research contribute to the advancement of waveguide-based systems and pave the way for improved performance and efficiency in various applications. Results of such studies can be found in [11–13].

Due to the high degree of automation (movement, capturing, storage, and post-processing), *FlexSensor* reduced possible human errors and enhanced the reliability of the measured data.

However, the fundamental idea is to provide an easy-to-use and easy-to-adopt framework for a broad kind of SiPh measurement. Therefore, *FlexSensor* has been designed module-wise, so that modules can be exchanged with little knowledge of the application's source code.

5. Conclusions

FlexSensor is a novel and extensible silicon photonic wafer-level measurement software program. It combines and tackles important features when measuring with highly integrated optical structures: 1. minimizing manual interaction (and thus potential errors) by defining measurement routines, 2. controlling other devices, 3. enabling the inclusion of uncertainty information in the captured data directly and 4. facilitating the development of future measurement routines.

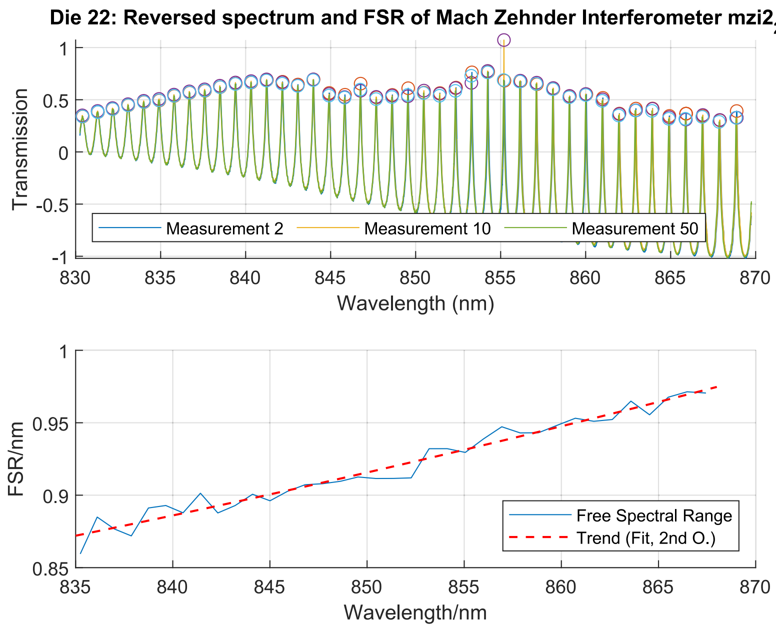


Fig. 6. Exemplary: Captured transmission spectrum and the calculated *FSR* of the *Mach–Zehnder Interferometer*. The measurement was conducted automatically using *FlexSensor*.

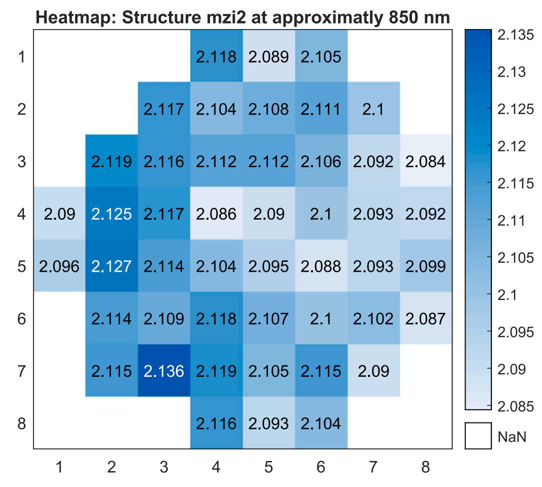


Fig. 7. Group index n_g measured across the wafer.

The software aims to exploit the full capabilities of a semi- or fully automated SiPh wafer probing system. Traditional manual wafer-level measurements are time-consuming, work-intensive, prone to errors, and limit the scalability of measurements. *FlexSensor* was developed with the explicit aim of measuring and evaluating PICs, leveraging the comprehensive functionalities offered by a probing system. It does this by introducing sophisticated algorithms, more fail-safe measurement and calibration routines, extended hardware control, and new development features (hardware simulators). The software offers a full range of functions from construction of a new measurement routine to post-processing the captured data.

FlexSensor has already been used to conduct measurements in, e.g., [11–13]. The exemplary group index n_g measurement was conducted to give an introductory example of the software possibilities. Not only was the software capable of conducting an uncertainty analysis of one structure with 100 repeated measurements, it also allowed measuring 48 different dies across the wafer with five different structures and 25 repetitions each (Fig. 7). This resulted in over 6000 individual measurements.

Future work will include new measurement routines, more sophisticated statistical post-processing, and support for more hardware (such as lasers and capturing devices).

6. Statement

During the preparation of this work, the author used *Grammarly* in order to improve readability and language. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

CRediT authorship contribution statement

Christoph Schmidt: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Data curation, Conceptualization. **Jakob Wilhelm Hinum-Wagner:** Writing – review & editing. **Reinhard Klambauer:** Writing – review & editing. **Alexander Bergmann:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2024.101879.

References

[1] Bogaerts W, Pathak S, Ruocco A, Dwivedi S. Silicon photonics non-resonant wavelength filters: Comparison between AWGs, echelle gratings, and cascaded Mach–Zehnder filters. In: Broquin J-E, Nunzi Conti G, editors. SPIE oPTO. San Francisco, California, United States; 2015, p. 93650H. http://dx.doi.org/10.1117/12.2082785, URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2082785.

[2] FormFactor Inc. User guide Velox 3. 2019.

[3] Chrostowski L, Hochberg ME. Silicon photonics design. Cambridge; New York: Cambridge University Press; 2015.

[4] Keysight Technologies Inc. Keysight parametric test solutions. NX5402A Silicon Photonics Wafer Test Syst 2024. URL https://www.keysight.com/zz/en/product/NX5402A/silicon-photonics-wafer-test-system.html.

[5] FormFactor Inc. Cascade SUMMIT200 20mm semi-/fully-automated probe station: Datasheet. 2023, URL https://www.formfactor.com/download/summit200-data-sheet/?wpdmdl=6375&refresh=646c8d37ef6351684835639.

[6] Physik Instrumente (PI) GmbH & Co KG. Active Alignment Systems and Tools for Photonics. URL https://www.physikinstrumente.com/en/products/fast-multi-channel-photonics-alignment-systems.

[7] MPI Corporation. MPI Fully Automatic Probe Systems. URL https://www.mpi-corporation.com/ast/engineering-probe-systems/mpi-fully-automated-systems/.

[8] Physik Instrumente (PI) GmbH & Co KG. 6-axis motion hexapod (H-811.S2): Datasheet. 2023, URL https://www.physikinstrumente.com/fileadmin/user_upload/physik_instrumente/files/datasheets/H-811.S2-Datasheet.pdf.

[9] FormFactor Inc. Autonomous SiPh measurement assistant brochure. 2021, URL https://www.formfactor.com/download/autonomous-silicon-photonics-measurement-assistant-brochure/?wpdmdl=11959&refresh=650a922257dfa1695191586.

[10] PySide6: Project page. 2023, URL https://pypi.org/project/PySide6/.

[11] Hinum-Wagner J, Buchberger A, Schmidt C, Schörner C, Rist D, Hoermann SM, et al. Design optimization of silicon nitride-based micro-ring resonator systems in a CMOS mass production environment. In: Cheben P, Čtyroký J, Molina-Fernández In, editors. Integrated optics: design, devices, systems and applications VII. Prague, Czech Republic: SPIE; 2023, p. 20. http://dx.doi.org/10.1117/12.2665322, URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12575/2665322/Design-optimization-of-silicon-nitride-based-micro-ring-resonator-systems/10.1117/12.2665322.full.

[12] Hinum-Wagner JW, Hörmann SM, Sattelkow J, Buchberger A, Schörner C, Janka S, et al. Ab-initio modeling of bend-losses in silicon nitride waveguides from visible to near-infrared. In: Kahan MA, editor. Optical modeling and performance predictions XIII. San Diego, United States: SPIE; 2023, p. 6. http://dx.doi.org/10.1117/12.2677328, URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12664/2677328/Ab-initio-modeling-of-bend-losses-in-silicon-nitride-waveguides/10.1117/12.2677328.full.

[13] Schmidt C. Investigations of the group index of silicon nitride waveguides (Master's thesis), Graz University of Technology; 2022.

[14] Soto-Perdomo J, Morales-Guerra J, Arango JD, Montoya Villada S, Torres P, Reyes-Vera E. OptiGUI DataCollector: A graphical user interface for automating the data collecting process in optical and photonics labs. SoftwareX 2023;24:101521. http://dx.doi.org/10.1016/j.softx.2023.101521, URL https://linkinghub.elsevier.com/retrieve/pii/S2352711023002170.

[15] Harun SW, Emami SD, Arof H, Hajireza P, Ahmad H. LabVIEW applications for optical amplifier automated measurements, fiber-optic remote test and fiber sensor systems. In: De Asmundis R, editor. Modeling, programming and simulations using labVIEW&#8482; software. InTech; 2011, http://dx.doi.org/10.5772/13247, URL http://www.intechopen.com/books/modeling-programming-and-simulations-using-labview-software/labview-applications-for-optical-amplifier-automated-measurements-fiber-optic-remote-test-and-fiber-.

[16] Scott J, Vorndran S. How to improve active photonics alignment performance. 2023, Whitepaper.

[17] Sacher Lasertechnik GmbH. Motorized tunable Littman/Metcalf laser system - Lion: Product page. 2023, URL https://www.sacher-laser.com/home/scientific-lasers/tunable_lasers/littman/tec_530_motorized_littman-metcalf_laser_system_lion.html.

[18] Digilent. Analog discovery 2: Product page. 2023, URL https://digilent.com/reference/test-and-measurement/analog-discovery-2/start.

[19] FormFactor Inc. Cascade Velox 3: Brochure. 2021, URL https://www.formfactor.com/download/cascade-probe-systems-brochure/?wpdmdl=6745&refresh=650c17415e0201695291201.

[20] The pandas development team. Pandas-dev/pandas: Pandas. 2020, http://dx.doi.org/10.5281/zenodo.3509134.

[21] Physik Instrumente (PI) GmbH & Co KG. NanoCube® nanopositioner (P-616): Datasheet. 2023, URL https://www.physikinstrumente.com/fileadmin/user_upload/physik_instrumente/files/datasheets/P-616-Datasheet.pdf.

[22] New Focus Inc. High-dynamic-range power sensors. 2003, URL https://www.newport.com/medias/sys_master/images/images/h5e/hb7/8797002432542/2101-2103-User-Manual-RevC.pdf.

[23] Ouyang B, Caro J, Bogaerts W, Xing Y. Silicon ring resonators with a free spectral range robust to fabrication variations. Opt Express 2019;27. http://dx.doi.org/10.1364/OE.381643.

[24] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, et al. Evaluation of measurement data – Guide to the expression of uncertainty in measurement. 2008, URL https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf/cb0ef43f-baa5-11cf-3f85-4dcd86f77bd6.