

Crosstalk and Loss Analysis Platform (CLAP)

User Manual

Version 5.0, June 2018

OPTICS Lab

The Hong Kong University of Science and Technology

<http://www.ece.ust.hk/~eexu>

I. CROSSTALK AND LOSS ANALYSIS PLATFORM (CLAP)

CLAP - the Crosstalk Noise Analyses Platform - is implemented in C++ and can analyze both coherent and incoherent crosstalk noise, loss, and SNR in optical networks and optical routers based on WDM or a single wavelength. CLAP has a comprehensive library of photonic devices to construct arbitrary optical routers and optical networks, including detailed microresonator model, I/Os, waveguide crossings, waveguide bendings, waveguides, optical terminators, parallel switching elements, crossing switching elements, and filter elements with photo-detectors. Mesh-based, folded-torus-based, and fat-tree-based optical networks are predefined and provided in the network library. CLAP can be easily extended to include more optical network architectures.

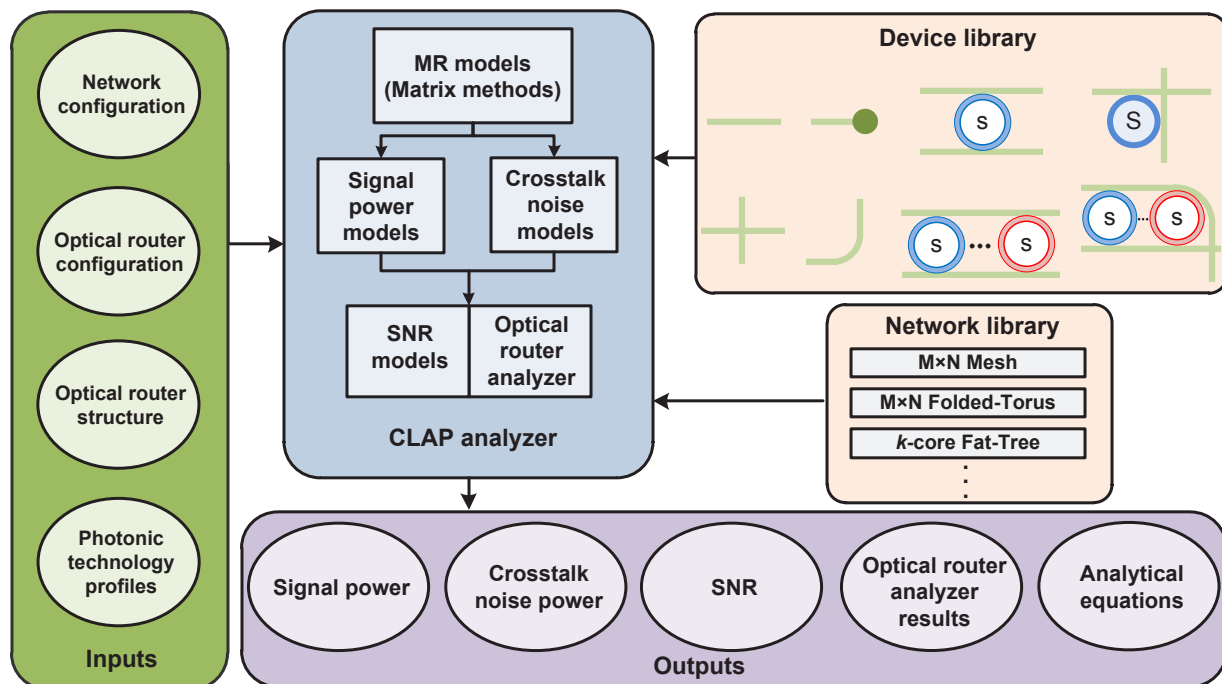


Fig. 1: The internal structure of CLAP.

Fig. 1 illustrates the internal structure of CLAP. The publicly released CLAP is implemented in C++ and Python and is available online with documentation at [1], [2]. CLAP analyzes the power loss, crosstalk noise power, and SNR in free-scale arbitrary optical interconnects and optical routers at the system-level. As can be seen from the figure, CLAP's internal structure includes inputs, a CLAP analyzer, outputs, a device library, and a network library. The analytical models in the *CLAP analyzer* are used for calculations at the network level as well as the optical router level. CLAP has a complete library of basic photonic

devices to construct arbitrary optical routers and interconnects, including I/Os, waveguide crossings, waveguide bendings, waveguides, optical terminators, microresonators, parallel switching elements, and crossing switching elements. Mesh-based, folded-torus-based, and fat-tree-based optical interconnection networks are predefined as the three default network architectures in the *Network Library*. CLAP can be easily extended to include more interconnect architectures. We consider four input files in CLAP: *Network Configuration*, *Router Configuration*, *Router Structure*, and *Technology Profiles*. A careful and simple text-based syntax is considered for different input definitions in CLAP. *Network Configuration* consists of the network size, chip size, and communication pattern among the processor cores, while *Router Structure* includes the definition of the optical router structure. Different configurations of the optical router as well as the input powers at the input ports of the router can be defined in *Router Configuration*. In CLAP, we consider different set of parameters for different optical router. Hence, the folder *Technology Profiles* consists of different photonic device technologies. Each profile will contain the power loss values, crosstalk coefficients, reflectance coefficients, waveguide dimensions, the microresonator's diameter, and the injection laser power. The analytical model for the optical micro-resonators in CLAP is based on the matrix analysis [3]. Based on our proposed analytical models, the signal power, crosstalk noise power, and SNR at the destination of a specific optical signal, defined by the user, can be analyzed in the *CLAP analyzer*. Furthermore, the *Optical router analyzer* helps in analyzing the worst-case as well as the average power loss and crosstalk noise in an optical router under various configurations. The analyzer is optimized to speed up computations in large scale optical interconnection networks. CLAP is capable of analyzing the propagation loss at the device, router, and network levels by using the defined dimensions for the waveguide and microresonator. Also, it calculates the higher orders of crosstalk noise based on the user's preferences. Considering the outputs, besides the numerical values of the signal power, crosstalk noise power, and SNR at the destination of an optical link, CLAP is also capable of generating the analytical equations used to analyze the signal power and crosstalk noise power in optical interconnection networks. It is worth mentioning that the dimension-ordered routing technique, also known as the XY routing algorithm, is used in mesh-based and folded-torus-based networks, while the optical turnaround routing algorithm is considered in fat-tree-based network in CLAP.

This manual reviews the functions and capabilities of CLAP. In particular, we present definitions of optical device parameters, optical elements, router structures, router configurations, and network configurations. Different optical router structure definition and network configuration examples are included in CLAP's package [1].

II. BASIC OPTICAL DEVICES CLASSES DEFINITIONS

Basic optical devices are widely used in constructing optical interconnection networks. We analyze and model the crosstalk noise, power loss, and SNR in such devices in [2]. The following classes describe the basic optical elements and switching elements, shown in Figs. 2 in CLAP. It should be mentioned that each definition should be in a separate line and *"/"* is used for inserting comments. It is worth mentioning that each optical element in CLAP should be assigned with an *ID*, where the *ID* is an integer number that starts from one. We will shortly discuss the *ID* allocation.

A. Class of Microresonator (MR)

Microresonators (MRs) are used as Basic Optical Switching Elements (BOSEs) in the structure of the parallel switching element, Fig. 2(d), and the crossing switching element, Fig. 2(e). An active MR can either be in the ON state or the OFF state. We define a class called MR in CLAP. In this class, *MR_status* is considered to indicate the current status of an MR. When *MR_status=1*, the MR is in the ON state, and *MR_status=0* indicates that the MR is in the OFF state. Also, *MR_num* is defined to assign each MR to a specific PSE or CSE in the optical router structure.

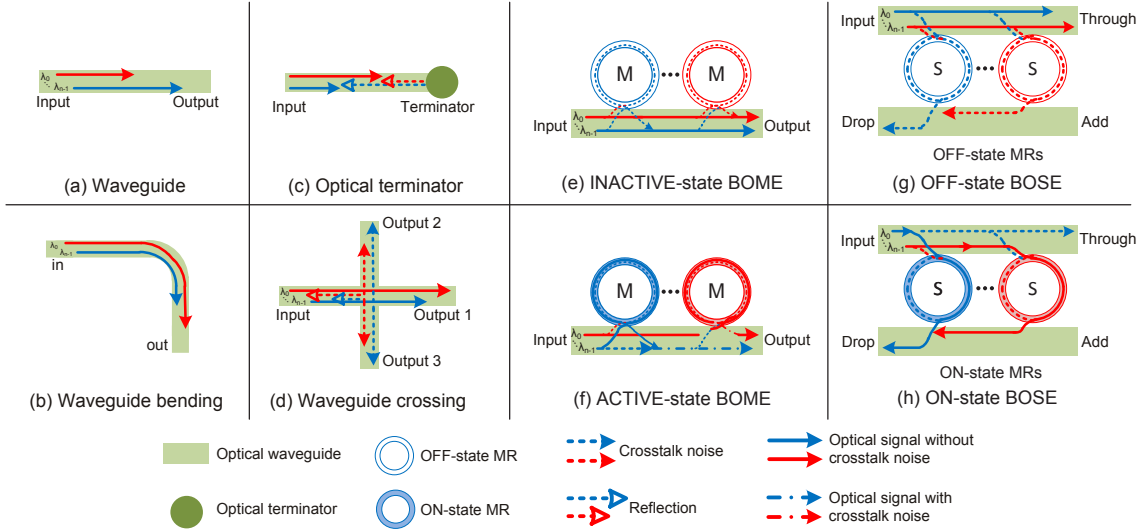


Fig. 2: Different basic optical elements and switching elements defined in CLAP [4]

TABLE I: prt_def values in the general 5×5 optical router

Port	prt_def value	Port	prt_def value
Injection	0	Ejection	1
Input North	2	Output North	3
Input East	4	Output East	5
Input South	6	Output South	7
Input West	8	Output West	9

B. Class of Port

The class of port is considered for defining input and output ports in CLAP. Each optical router structure has a number of input and output ports, where each port has a next and a previous terminal to show the elements the port is connected to. Fig. 3 indicates the port model defined in CLAP. The user can define a port as shown in the following:

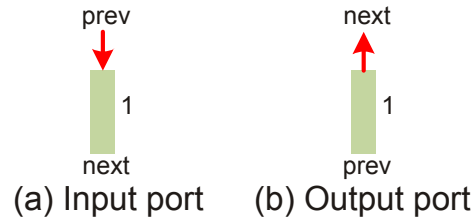


Fig. 3: The port model in CLAP.

```

define PRT id=port id number;
prev=id number of the previous element;
next=id number of the next element;
con=the terminal number of the next/previous element that the port is connected to;
prt_def=the port definition;

```

The port has only one terminal and it is assigned with the number 1. If there is no element at the *next*

TABLE II: prt_def values in the general 4×4 optical router

Port	prt_def value	Port	prt_def value
Input Upper-left	0	Output Upper-left	1
Input Lower-left	2	Output Lower-left	3
Input Lower-right	4	Output Lower-right	5
Input Upper-right	6	Output Upper-right	7

or *prev* terminal of the port, -1 should be assigned to that terminal. Tables I and II indicate the *prt_def* values considered for 5×5 and 4×4 optical routers, respectively. This parameter determines the type of the port in CLAP.

C. Class of Waveguide

The waveguide is also considered in CLAP. The light direction in the waveguide determines the terminals. The input terminal is assigned with the number 1, while the output terminal is assigned with the number 2. The waveguide's length can be determined by *length* which is in μm . Fig. 4 indicates the waveguide model in CLAP. The user can define a waveguide as shown in the following:

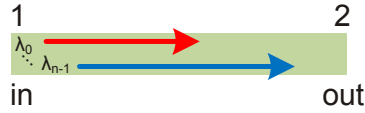


Fig. 4: The waveguide model in CLAP.

```

define WGD id=waveguide id number;
in=the id number of the element connected to the input terminal;
out=the id number of the element connected to the output terminal;
con_in=the terminal number of the element connected to the input terminal;
con_out=the terminal number of the element connected to the output terminal;
length=the waveguide length in μm;

```

D. Class of Waveguide Crossing

The waveguide crossing structure is shown in Fig. 2(a). It has four terminals, which are defined as west, north, east, and south. The numbers assigned to these terminals are 1, 2, 3, and 4, respectively. Fig. 5 indicates the waveguide crossing model in CLAP. The user can define a waveguide crossing in CLAP as shown in the following:

```

define WCR id=waveguide crossing id number;
west=the id number of the element connected to the west terminal;
north=the id number of the element connected to the north terminal;
east=the id number of the element connected to the east terminal;
south=the id number of the element connected to the south terminal;
con_w=the terminal number of the element connected to the west terminal;
con_n=the terminal number of the element connected to the north terminal;
con_e=the terminal number of the element connected to the east terminal;
con_s=the terminal number of the element connected to the south terminal;

```

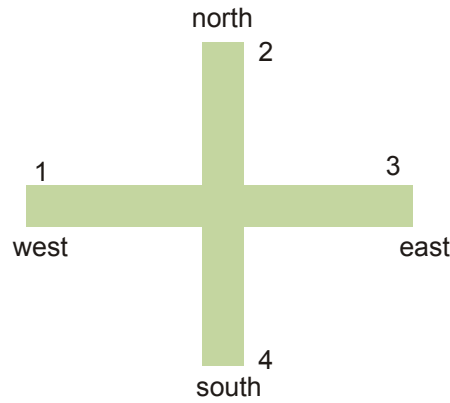


Fig. 5: The waveguide crossing model in CLAP.

E. Class of Waveguide Bending

Fig. 2(c) indicates the waveguide bending structure. The waveguide bending has an input and output terminal, which are assigned with the numbers 1 and 2, respectively. Fig. 6 indicates the waveguide bending model defined in CLAP. The user can define a waveguide bending in CLAP as shown in the

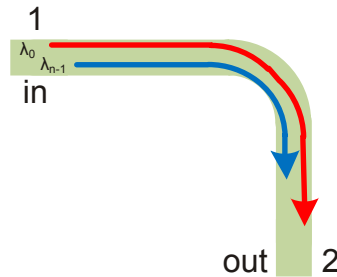


Fig. 6: The waveguide bending model in CLAP.

following:

```

define WBN id=waveguide bending id number;
in=the id number of the element connected to the input terminal;
out=the id number of the element connected to the output terminal;
con_in=the terminal number of the element connected to the input terminal;
con_out=the terminal number of the element connected to the output terminal;

```

F. Class of Optical Terminator

The optical terminator structure is depicted in Fig. 2(b). The optical terminator has an input terminal which is assigned with the number 1. Fig. 7 shows the structure of the optical terminator in CLAP. The

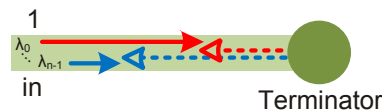


Fig. 7: The optical terminator model in CLAP.

user can define an optical terminator as shown in the following:

```

define OTR id=optical terminator id number;
in=the id number of the element connected to the input terminal;
con_in=the terminal number of the element connected to the input terminal;

```

G. Class of Optical Couplers (i.e. Optical pins)

Optical pin is defined as OPN, which is placed to couple the light from one waveguide (i.e. on-chip waveguide) to another waveguide (i.e. on-board waveguide). It has two terminals: in (input) and out (output). Fig. 7 shows the structure of the optical terminator in CLAP. The user can define an optical

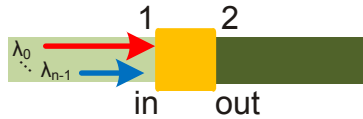


Fig. 8: The optical pin model in CLAP.

terminator as shown in the following:

```

define OPN id=optical pin id number;
in=the id number of the element connected to the input terminal;
out=the id number of the element connected to the output terminal;
con_in=the terminal number of the element connected to the input terminal;
con_out=the terminal number of the element connected to the output terminal;

```

H. Class of Basic Optical Modulating Elements (BOME)

The Basic Optical Modulating Element (BOME) is also considered in CLAP. Fig. 9 indicates the BOME structure in the INACTIVE and ACTIVE state, respectively. BOME is a series of MRs coupled on a single waveguide to modulate the light signal. As depicted, BOME has one input terminal (in) and one output terminal denoted as out. The input terminal is assigned with the number 1, while the output terminal is assigned with the number 2. The MR of the BOME is defined by *PSE_MR* and is an instance from the MR class. The user can define an BOME as shown in the following:

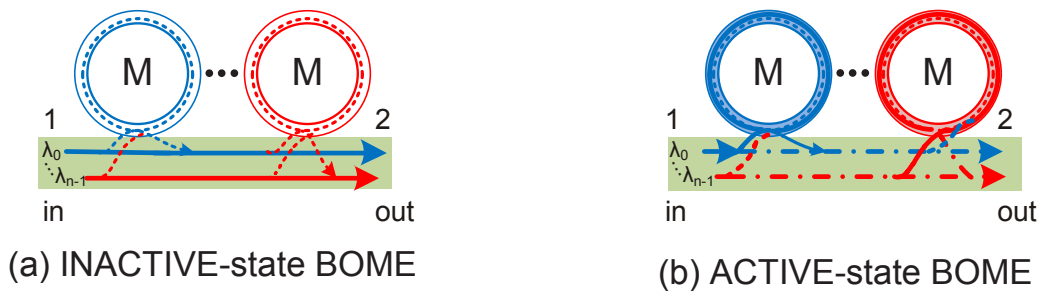


Fig. 9: The waveguide model in CLAP.

```

define OME id=BOME id number;
in=the id number of the element connected to the input terminal;
out=the id number of the element connected to the output terminal;
con_in=the terminal number of the element connected to the input terminal;
con_out=the terminal number of the element connected to the output terminal;

```

I. Class of Parallel Basic Optical Switching Element (Parallel BOSE)

Fig. 2(a) and (b) depict the parallel switching element structure in the OFF state and ON state, respectively. The parallel switching element has four terminals, the input, drop, through, and add ports, which are assigned with the numbers 1, 2, 3, and 4, respectively. It also includes an MR, which is defined by *PSE_MR* and is an instance from the MR class. The parallel switching element model in CLAP is illustrated in Fig. 10. The user can define a PSE in CLAP using the following syntax:

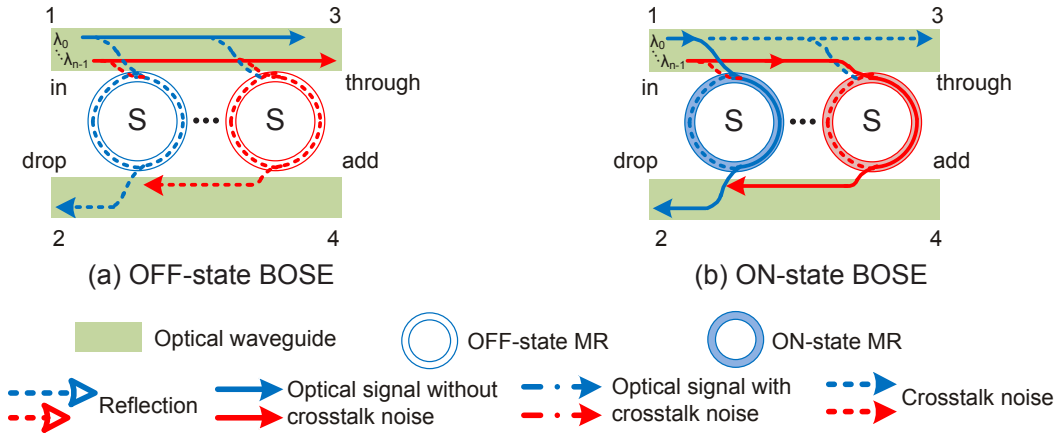


Fig. 10: The parallel switching element model in CLAP.

```

define PSE id=Parallel BOSE id number;
in=the id number of the element connected to the in terminal;
drop=the id number of the element connected to the drop terminal;
through=the id number of the element connected to the through terminal;
add=the id number of the element connected to the add terminal;
con_in=the terminal number of the element connected to the in terminal;
con_d=the terminal number of the element connected to the drop terminal;
con_t=the terminal number of the element connected to the through terminal;
con_a=the terminal number of the element connected to the add terminal;
MR=the MR number of the Parallel BOSE;

```

J. Class of Crossing Basic Optical Switching Element (Crossing BOSE)

Fig. 2(e) indicates the crossing switching element in the OFF and ON state. Similar to the waveguide crossing class, the CSE has four terminals including west, north, east, and south, which are assigned with the numbers 1, 2, 3, and 4, respectively. Moreover, it includes an MR which is defined as *CSE_MR* and is an instance from the MR class. In order to specify the MR location in the CSE, the *CSE_MR_Loc* is defined. There are four possible locations for the MR in the CSE, which are shown in Fig. 11. It is worth mentioning that when WDM is used, there is no need to define different MRs associated with a CSE or PSE separately. In other words, all the MRs can be treated like a single MR. The following shows how the user can define a CSE in CLAP:

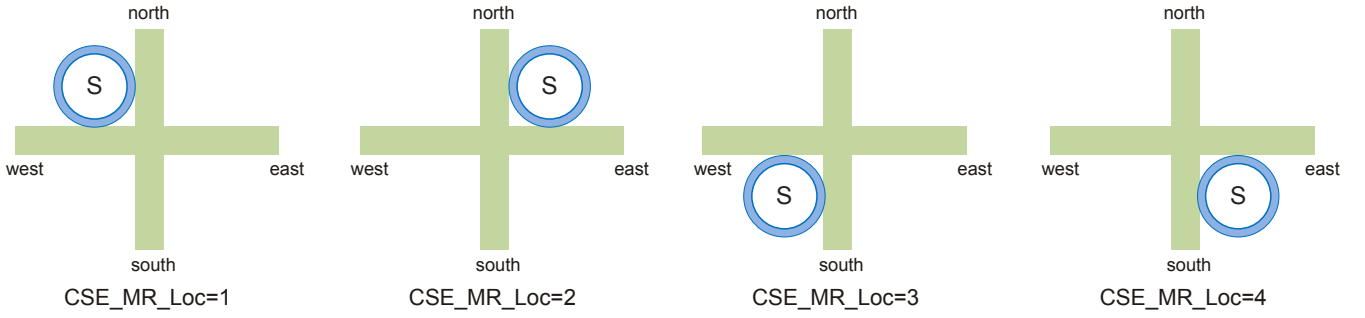


Fig. 11: The crossing switching element model in CLAP.

```

define CSE id=Crossing BOSE id number;
west=the id number of the element connected to the west terminal;
north=the id number of the element connected to the north terminal;
east=the id number of the element connected to the east terminal;
south=the id number of the element connected to the south terminal;
con_w=the terminal number of the element connected to the west terminal;
con_n=the terminal number of the element connected to the north terminal;
con_e=the terminal number of the element connected to the east terminal;
con_s=the terminal number of the element connected to the south terminal;
MR=the MR number of the Crossing BOSE;
MR_L=CSE MR location based on Fig. 11;

```

III. INPUTS IN CLAP

A. Photonic Technology Profiles

In CLAP 5.0, the user can define a set of required parameters for the basic optical elements and switching elements as a photonic technology profile. First, a technology profile number of an optical router should be defined in *Router_Structure_Definition.txt* under the parameter called *tech_profile_number*. A set of photonic device parameters could then be defined as a text file in the folder named *Technology_Profiles*, following the format *Technology_Profile_<tech_profile_number>.txt*. Hence, in CLAP 5.0, multiple sets of device parameters could be defined for different sets of routers. The user can define the power loss values, crosstalk coefficients, reflectance coefficients, waveguide dimensions, the microresonator's diameter, and the injection laser power in this text file. Please note that each parameter should be described in a separate line. Also, for different power loss and crosstalk coefficient values, it is highly suggested to use positive numbers. The different dimensions should be in μm . In CLAP 5.0, we have provided pre-defined parameters which are based on the matrix analysis of the micro-resonator [3].

1) *Insertion loss, crosstalk coefficient, and back-reflection coefficient in the waveguide crossing*: The keywords considered for the insertion loss, crosstalk coefficient, and back-reflection coefficient in the waveguide crossing are *Lc*, *Kc*, and *Kr*, respectively. The following can be used to define these parameters in CLAP:

```

Lc=waveguide crossings's insertion loss value in dB;
Kc=waveguide crossing's crosstalk coefficient value in dB;
Kr=waveguide crossing's back-reflection coefficient value in dB;

```

2) *Different losses and crosstalk coefficients in the parallel switching element*: The keywords considered for the passing loss and drop loss in the parallel switching element are *L_pse_off* and *L_pse_on*,

respectively. Moreover, K_{pse_off} is the keyword to define the crosstalk coefficient of the PSE in the OFF state, and K_{pse_on} is the keyword to define the crosstalk coefficient of the PSE in the ON state. The following can be used to define different loss and crosstalk coefficient values in the PSE in CLAP:

L_{pse_off} =PSE's passing loss value in dB;
 L_{pse_on} =PSE's drop loss value in dB;
 K_{pse_off} =crosstalk coefficient value per PSE in the OFF state in dB;
 K_{pse_on} =crosstalk coefficient value per PSE in the ON state in dB;

In Clap 5.0, users could also define the passing loss, insertion loss and crosstalk noise for a series of photo detectors, or Basic Optical Filtering Elements (BOFEs). Being different from Parallel BOSE or Crossing BOSE, photo detectors, or BOFEs, are MRs coupled into a single waveguide to detect the optical signal at the end of the optical path. Hence, in this version, we have added those parameters which can be defined by the following:

L_{det_off} =BOFE's passing loss value in power percentage;
 L_{det_on} =BOFE's insertion loss value in power percentage;
 K_{det_on} =Crosstalk coefficient for BOFEs;

In case of using WDM, those detectors' parameters need to be in a series. For example, if we use 64 wavelengths, we need to define a series of 64 parameters, each should be written in one line and ended with a semi-colon.

Those losses and crosstalk coefficients could also be achieved by the provided MR model. The description of this program is given in section B of this chapter.

3) *Reflectance coefficient in the optical terminator*: The keyword considered to define the reflectance coefficient in the optical terminator is Kt . The following can be used to define this parameter in CLAP:

Kt =optical terminators's reflectance coefficient value in dB;

4) *Bending loss in the waveguide bending*: The keyword considered to define the bending loss in the waveguide bending is Lb . The following can be used to define this parameter in CLAP:

Lb =Waveguide bending's bending loss in dB;

5) *Propagation loss*: The keyword considered to define the propagation loss per cm is Lp . The following can be used to define the propagation loss in CLAP:

Lp =propagation loss in dB/cm;

6) *Polarization coefficient*: The keyword considered to define the polarization coefficient for the optical waveguide is $Lpol$. The following can be used to define the polarization coefficient in CLAP. It is noted this parameter should be defined in dB (i.e. treated as power loss for the waveguide section).

$Lpol$ =Waveguide's polarization coefficient in dB;

7) *Polarization coefficient*: The following can be used to define the coupler loss (i.e. optical pin loss) in CLAP. It is noted this parameter should be defined in dB.

Lcpl=Optical coupler, or Optical pin loss in dB;

8) *Input optical power (laser power)*: The keyword considered to define the input optical power at the laser source is *Pin*. The following can be used to define the input power in CLAP:

Pin=input optical power in dBm;

9) *WDM-Related Parameters*: The keywords considered to define the Free-Spectral Range (FSR), the MR's quality factor (Q), and the MR's wavelength range are defined in the following:

FSR=Free-Spectral Range in nm;
MR_Q=MR's quality factor;
MR_wvlngth_range=MR's wavelength range in nm;

10) *Miroresonator's diameter and waveguide's width*: The keywords considered to define the diameter of the microresonator and the width of the waveguide in μm are *MR_Dimension* and *WG_width*, respectively. The following can be used to define these parameters in CLAP:

MR_Dimension=microresonator's dimension in μm ;
WG_width=waveguide's width in μm ;

B. XTalk release program

This program provides the loss and crosstalk parameters for a given structure of MR (or series of MRs). In Clap 5.0, we utilize the Matrix method for our analyses of signal power loss and crosstalk noise. The usage detail of this program could be found in README.txt in *MR_Model* folder. Users can define all the MRs' parameters in the file named *Cal_Matrix.py* and run the *call_python.cpp* for the results. It should be noted that users need to install Python 3.x or above together with the python lib (i.e. anaconda 3).

C. Optical Router Structure Definition

The user can define any optical router structure in CLAP. We consider an input text file, called *Router_Structure_Definition.txt*, which allows the user to define the optical router structure. The input and output ports' configurations as well as the routing algorithm decide which MRs need to be turned ON or OFF. After setup of all the MRs, the *find_path* function helps find the path from the input port towards the output port in the router structure. Moreover, it calculates the length of the waveguide over which the optical signal has travelled to reach the output port.

Prior to defining the router structure, each element in the structure should be assigned with an integer and positive *ID*. Firstly, each MR should be assigned with an *ID* starting from zero. Secondly, an *ID*, which starts from one, should be assigned to each basic device element and switching element in the optical router structure. It is highly recommended that the user follows the following elements order when assigning different *IDs* to the basic elements: ports, waveguide crossings, waveguide bendings, optical terminators, CSEs, PSEs, and waveguides. Fig. 12 depicts an example of assigning *ID* numbers to the Crux optical router structure. Also, Fig. 12 indicates the Crux optical router for WDM-based optical interconnection networks when the total number of wavelengths is *W* as an example. When using WDM, you can assign each group of MRs in a CSE or PSE with a single *ID*.

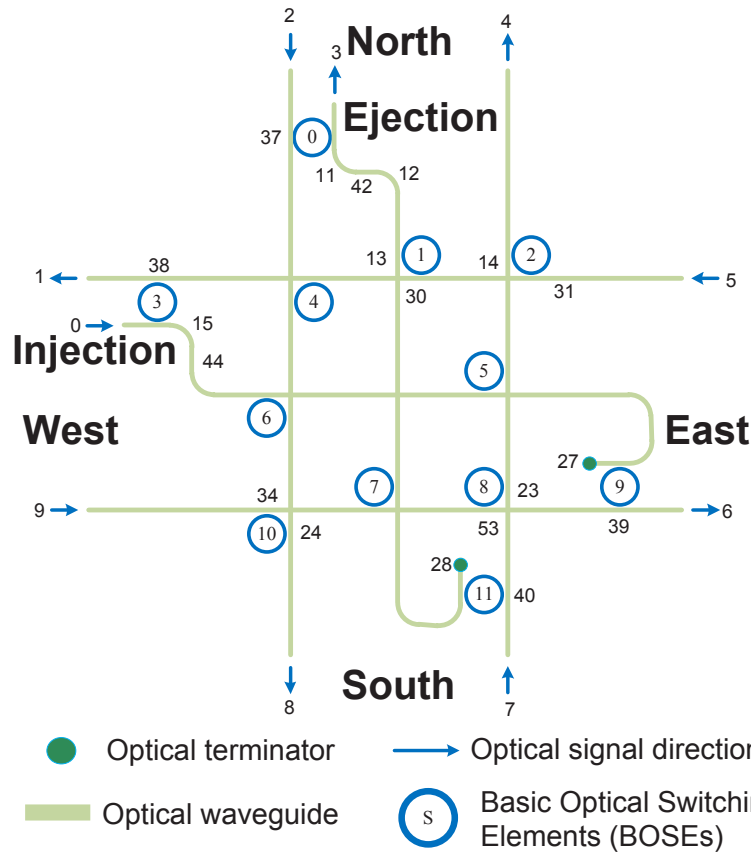


Fig. 12: An example of assigning *ID* numbers to the Crux optical router structure.

As mentioned before, the input text file *Router_Structure_Definition.txt* is considered to define the optical router structure in CLAP. The overall structure of this input file is defined in Algorithm (1). Please note that the definition order should remain unchanged. Also, *MR_config*, *start*, and *end* are keywords.

Algorithm 1 The overall structure of the *Router_Structure_Definition.txt* input file.

```
//Specify the number of each element
MR_config
.
//Configure different MRs
.
start
.
//Basic optical elements should be defined here
.
end
```

1) *Specify the number of each element*: The user needs to specify the number of each element at the very beginning of the *Router_Structure_Definition.txt* file as shown in the following:

TechProfile=*the technology profile number of a particular router*;
 #PRT=*the total number of ports in the router structure*;
 #MR=*the total number of MRs in the router structure*;
 #WCR=*the total number of waveguide crossings in the router structure*;
 #WBN=*the total number of waveguide bendings in the router structure*;
 #OTR=*the total number of optical terminators in the router structure*;
 #OTR=*the total number of optical pins in the structure*;
 #CSE=*the total number of CSEs in the router structure*;
 #PSE=*the total number of PSEs in the router structure*;
 #WGD=*the total number of waveguides in the router structure*;

2) *Configure different MRs*: The user needs to configure different MRs to enable the optical signal to make a turn from an input port towards an output port. MRs need to be configured based on the considered routing algorithm. The user can configure different MRs in the optical router structure using the following method. When the user sets an MR, it will be turned on.

MR_config

if input=*the input port id number*; output=*the output port id number*; set MR=*the id number of the MR*;

3) *Basic optical elements definition*: After identifying the number of each element and configuring different MRs, the user needs to define each optical element in the optical router structure using the syntax detailed in the previous section. The overall elements definition should be as following:

```

start
define PRT ...
define WCR ...
define WBN ...
define OTR ...
define OPN ...
define CSE ...
define PSE ...
define WGD ...
end
  
```

D. Optical Router Configuration

The optical router analyzer function helps the user to analyze the signal power, crosstalk noise power, and SNR in arbitrary optical routers. While the router structure is defined in the *Router_Structure_Definition.txt* file, the user can define the optical router configuration in the *Router_Configuration.txt* input file. In this text file, the user can define the input powers at different input ports and the order of the crosstalk analysis, and configure the optical router by setting different pairs of input/output ports. The order of the crosstalk noise analysis is the degree of the crosstalk noise coefficients in the analyses; if the crosstalk order equals two, the analyzer will compute all the crosstalk noise coefficients to the degree of two in the router. The user should follow the syntax described in Algorithm 2 while describing the router configuration. In this algorithm, *input_port_id* and *output_port_id* are the *id* numbers of the input and output ports. Please note that the definition order should remain unchanged. *xtalk_order*, *config_start*, and *config_end* are keywords. **Please note that when using WDM, you need to define a crossing switching element as a combination of a parallel switching element and a waveguide crossing**

Algorithm 2 The overall structure of the *Router_Configuration.txt* input file.

```
xtalk_order=the preferred highest degree of the crosstalk noise coefficients in the results;
set_wdm Number of Wavelengths;
config_start
//The configuration of different input and output ports
.
from input_port_id to output_port_id;
.
config_end
//set the optical power at different input ports
.
prt_id=the port id of the input port set_pwr=optical power in dB;
.
```

E. Network Configuration

The user can configure the network in the input text file called *Network_Configuration.txt*. The network size, the chip size, and the communication pattern among the processor cores can be defined in this input file. We consider three optical interconnect architectures in CLAP, mesh-based, folded-torus-based, and fat-tree-based networks. CLAP can be easily extended to include more interconnect architectures. In every architecture, each processor core can be addressed using a coordinate, as shown in Figs. 13 and 14. Utilizing the coordinates, the user can define the communication pattern among different processor cores in the network. The overall structure of the *Network_Configuration.txt* input file is shown in Algorithm 3. The same definition order as shown in the algorithm needs to be followed. *com_pattern start* and *com_pattern end* are keywords.

Algorithm 3 The overall structure of the *Network_Configuration.txt* input file.

```
//Network size
//Chip size in cm2
com_pattern start
.
//defining the communication pattern among the processor cores
.
com_pattern end
```

1) *Network size initiation:* The network size in mesh-based and folded-torus-based networks can be defined as $M \times N$, where M and N are, respectively, the number of processor cores in a row and a column of the mesh-based or folded-torus-based networks' floorplans. The following can be used to define M and N .

```
M=the number of processor cores along the x axis;
N=the number of processor cores along the y axis;
```

Also, when fat-tree-based networks are considered, k defines the number of processor cores (network size) in the floorplan of fat-tree-based networks. The following can be used to define the number of processor cores, k , in fat-tree-based networks:

k =the number of processor cores in the fat-tree-based network;

2) *Chip size initiation*: The chip size should be defined in cm^2 . The user can initiate the chip size based on the following in CLAP:

chip_size=the chip size in cm^2 ;

3) *Communication pattern definition*: The communication pattern among different processor cores can introduce crosstalk noise to the considered optical signal. The user needs to make sure that there is no conflict in the communication pattern among the cores. The following shows how to define the communication pattern in CLAP. When using mesh-based or folded-torus-based networks, the *Source* and *Destination* should be the coordinates of the source and the destination processor cores in the form of x, y . However, when using fat-tree-based networks, the *Source* and *Destination* should be the processor number of the source and destination processor cores.

com_pattern start

.

from *Source1* to *Destination1*;

from *Source2* to *Destination2*;

.

Com_pattern end

Table III summarizes different input files and their applications in CLAP.

TABLE III: Different input files in CLAP

Input file name	Application
<i>input.txt</i>	Determines the architecture type and the considered optical link
<i>Device_Parameters.txt</i>	Contains the basic device parameters' initiations
<i>Router_Structure_Definition.txt</i>	Defines the optical router structure
<i>Router_Configuration.txt</i>	Used to configure the optical router
<i>Network_Configuration.txt</i>	Defines network's properties and the communication pattern

F. Top level configuration

When all the parameters for device parameters, router structure and network structure have been set up, the user can start using CLAP by selecting an optical interconnect architecture as well as the coordinates of the source and the destination of the optical link under the analysis in the input file called *input.txt*, as in the following:

arch_type=Optical Interconnect Architecture;

set_wdm Number of Wavelengths;

from *Source* to *Destination*;

Optical Interconnect Architecture can be either *mesh*, if a mesh-based network is considered, *ftorus*, if a folded-torus-based network is considered, or *free*, if a fat-tree-based network is considered. The user can define the number of optical wavelengths, when utilizing WDM, using *set_wdm*. Also, *unset_wdm*

assumes that there is only one wavelength considered in CLAP (equivalent to *set_wdm 1*);). Moreover, the *Source* and *Destination* coordinates should be defined as x, y , for mesh-based and folded-torus-based networks and *Processor number*, for fat-tree-based networks. For example, the following indicates that the user has selected the use of mesh-based networks and is also interested in analyzing the optical signal from the processor core (1, 2) towards the core (3, 5).

```
arch_type=mesh;
from 1,2 to 3,5;
```

Figs. 13 and 14 help in defining the *Source* and *Destination* coordinates in mesh-based, folded-torus-based, and fat-tree-based networks. To facilitate the matching between the processor cores and optical routers in Fig. 13(b), each optical router is assigned with the corresponding processor's coordinates.

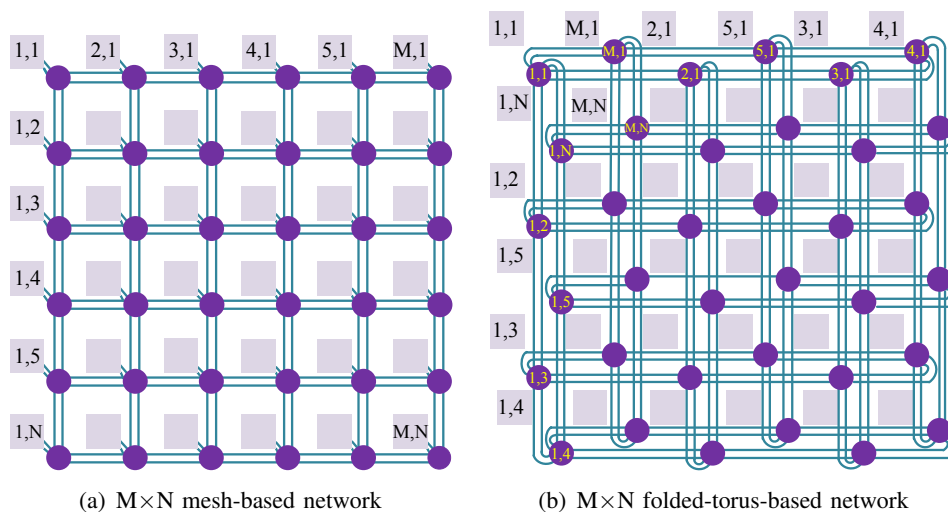


Fig. 13: Processors' coordinates in mesh-based and folded-torus-based optical interconnection networks.

G. Input files and commands

Figure 15 summarized the necessary steps to use our CLAP 5.0. As mentioned in the introduction, CLAP has three main functions: MR models calculator, router model analyzer and optical network analyzer. To start CLAP, the user should select among the three functions as well as identify the corresponding input or output method. The following code should be used to start CLAP:

```
./clap function output_method
```

Considering the *function*, the user can choose *configure* to set up the MR models. The user can also choose *network* to use the network-level analyzer or *router* to use the router-level analyzer. Besides, for network and router analyzer, *s* can be used to show the results on the screen, while *f* can be used to report the results in a text file, called *Network_Analyzer_Results.txt*, when the network-level analyzer is called, or *ORouter_Analyzer_Results.txt*, when the router-level analyzer is called. In the rest of this section, we detail the different inputs of CLAP shown in Fig. 1.

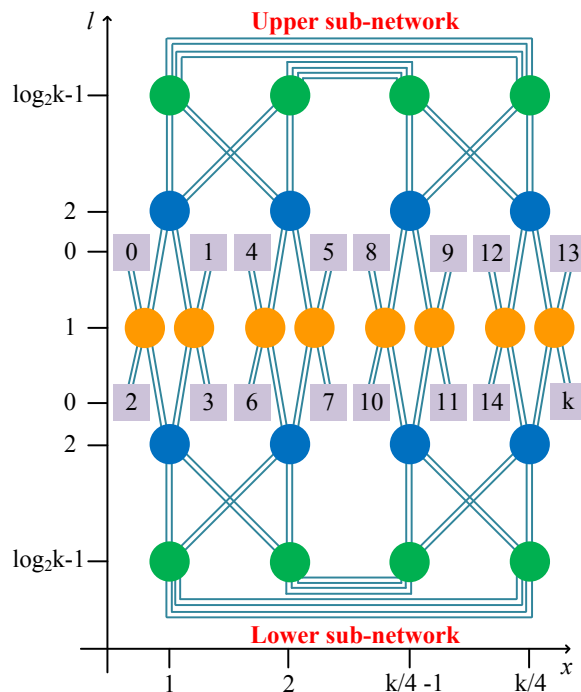


Fig. 14: Processors numbers in fat-tree-based networks.

Usage	Folder/File	Clap command
Setting up Parameters	Technology profiles folder <i>Technology_Profile_<number>.txt</i>	Input by user Use Xtalk_Release program to generate necessary parameters and input to <i>Technology_Profile_<number>.txt</i>
Setting up Optical Router	<i>Router_Structure_Definition.txt</i> and <i>Router_Configuration.txt</i>	Input by user
Setting up Optical Network	<i>Network_Configuration.txt</i> and <i>inputs.txt</i>	Input by user
Analyze worst-case SNR of a specific Optical Router	<i>Router_Structure_Definition.txt</i> and <i>Router_Configuration.txt</i>	>> <i>./clap router <s> or <f>*</i>
Analyze worst-case/average case of an Optical Interconnection Network with a particular traffic pattern	<i>Network_Configuration.txt</i> and <i>inputs.txt</i>	>> <i>./clap network <s> or <f>*</i>

* <s> means showing the result to the screen, <f> means printing the result(s) to the file *output.txt*

Fig. 15: Summarized steps to use CLAP.

IV. CLAP ANALYZER

The proposed analytical models in [2], [5] are integrated into the CLAP analyzer. Based on the analytical models and the user's inputs, the CLAP analyzer calculates the signal power, crosstalk noise power, and SNR at the destination of the optical link specified by the user. As Fig. 1 indicates, the analytical models are used to perform the analyses at the network and router levels in optical interconnection networks. The analyzer is optimized to speed up computations in large scale optical interconnection networks.

V. OUTPUTS IN CLAP

According to the user's preference, CLAP indicates the signal power, crosstalk noise power, and SNR at the destination of the defined optical link either on the screen or in an output text file. Different outputs of CLAP can be seen in Table IV. As shown in the table, CLAP also generates the analytical equations used for analyzing the signal power and crosstalk noise power in optical interconnects.

TABLE IV: Different output files in CLAP

Output file name	Application
<i>Signal_Power_Equation.txt</i>	Contains the signal power equation
<i>Crosstalk_Noise_Power_Equation.txt</i>	Contains the crosstalk noise power equation
<i>ORouter_Analyzer_Results.txt</i>	Contains the router analyzer results
<i>Network_Analyzer_Results.txt</i>	Contains the network analyzer results

REFERENCES

- [1] M. Nikdast and J. Xu. Crosstalk noise and loss analysis platform (CLAP). [Online]. Available: http://www.ece.ust.hk/~eexu/index_files/crosstalk.htm
- [2] M. Nikdast, J. Xu *et al.*, "Systematic analysis of crosstalk noise in folded-torus-based optical networks-on-chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 3, pp. 437–450, March 2014.
- [3] J. K. S. Poon, J. Scheuer, S. Mookherjea, G. T. Paloczi, Y. Huang, and A. Yariv, "Matrix analysis of microring coupled-resonator optical waveguides," *Opt. Express*, vol. 12, no. 1, pp. 90–103, Jan 2004. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-12-1-90>
- [4] L. H. K. Duong, P. Yang, Z. Wang, Y. S. Chang, J. Xu, Z. Wang, and X. Chen, "Crosstalk noise reduction through adaptive power control in inter/intra-chip optical networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2018.
- [5] Y. Xie, M. Nikdast, J. Xu, X. Wu, W. Zhang, Y. Ye, X. Wang, Z. Wang, and W. Liu, "Formal worst-case analysis of crosstalk noise in mesh-based optical networks-on-chip," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 21, no. 10, pp. 1823–1836, 2013.