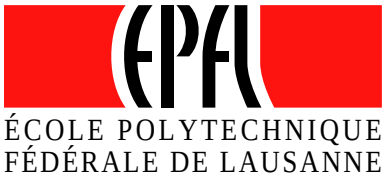


Faculté des Sciences et Techniques de l'Ingénieur (STI)
Section Microtechnique (MT)
Institut d'Imagerie et d'Optique Appliquée (IOA)

TOPOLOGY AND RECONFIGURATION OF OPTICALLY INTERCONNECTED SYSTEMS

School of Engineering and Physical Sciences
Department of Physics
Optically Interconnected Computing Systems Group (OIC)

Diploma Work 2003

Yann Tissot

Professor (EPFL): R-P. Salathé
Professor (HW): A.C. Walker
Supervisor (HW): J.F. Snowdon

Topology and Reconfiguration of Optically Interconnected Systems

Yann Tissot, Microengineering

Supervisor: J.F Snowdon
Professor: R-P. Salathé

Free-space optical interconnections involve many issues as many redundancies affect the efficiency of the entire interconnection system. In order to keep the efficiency high and redundancy low, OIC wants to combine a reconfigurable structure to the optoelectronic unit (Node) and assess the benefits. The Node would be a "sandwich" of a processing unit (CPU) combined with an FPGA and an optical interface composed of VCSELs flip-chipped on the FPGA and Photodiodes.
The optical interconnection system is called the Optical Highway (OH) which uses polarisation configuration to direct the signal to the right destination.

The project aimed to find and valid a new design of the optical interface (mapping of the emitters and receivers).

In order to achieve this new design, we first set a computational model of how a group of links (beams) behaves from one node to another simultaneously. This is the crucial part since it figures out the main issue due to the imaging system the optical highway represents. This step enables us to set the design of the optical interface.

In order to validate the design, we simulated the implementation of a CC Topology without redundancy on the optical interface i.e with the best efficiency. The best results obtained are 87.5 % of efficiency for an array of 8 nodes of 4x4 VCSELs/PDs. We then assess the redundancies on the

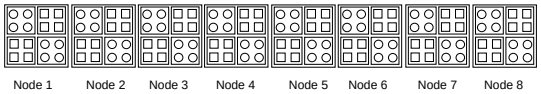


Figure 1: Design of the optical interface (array tested with the best efficiency)

polarising structure and see that they are minimised but unavoidable by the use a static polarising structure (HWP) when they can be avoidable by the use of a dynamic polarising structure (LCD).

Finally the design was validated experimentally by first implementing a physical pattern of 3 laser spots and second switching their polarisation state throughout the OH.
The results perfectly fits the computational

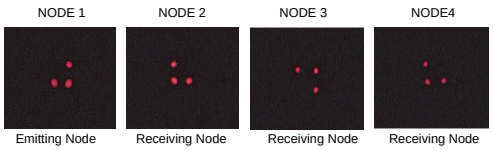


Figure 2: Experimental results

model and so the design is validated.

Topology and Reconfiguration of Optically Interconnected Systems

Yann Tissot, Microtechnique

Superviseur: J.F Snowdon
Professeur: R-P. Salathé

L'utilisation de l'optique classique dans le cadre d'interconnections optique implique plusieurs problèmes de redondances ce qui affecte l'efficacité du système dans son integralité. De manière à garder une efficacité élevée et peu de redondances, OIC voudrait combiner une structure reconfigurable a l'unité optoélectronique (appelée "Node") et en évaluer les bénéfices. Le "Node" serait un "sandwich" composé d'un processeur, d'un FPGA et d'une interface optique composée de VCSELs et PDs "flip-chipped" sur le FPGA. Le système d'interconnection optique est appelé "Optical Highway" (OH) et utilise les états de polarisation pour diriger le signal à la bonne destination.
Le projet a pour but de trouver et valider une conception de l'interface optique (placement des émetteurs et récepteurs).
La première étape en vue de la réalisation de l'interface optique est l'élaboration d'un modèle computationnel du comportement d'un groupe de liens émanant d'un node simultanémant. Cette partie est cruciale puisqu' elle met en évidence les problèmes du système d'imagerie que représente l'OH. Cette étape nous permet de concevoir l'interface optique. De manière a valider la conception on simule l'implémentation d'une topologie CC sans redondance sur l'interface optique, c-à-d avec la meilleure efficacité. Le meilleur résultat obtenu est une efficacité de 87,5% pour une rangée de 8 nodes de 4x4 VCSELs/PDs. On évalue par la suite

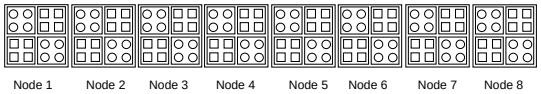


Figure 3: Conception de l'interface optique (rangée testée avec la meilleure efficacité)

les redondances sur la structure polarisante de l'OH. Il en résulte qu'elles sont minimes mais inévitables par l'utilisation d'une structure passive (HWP) alors qu'elles peuvent être évitées par l'utilisation d'une structure active (LCD).
Finalement on valide expérimentalement la conception de l'interface optique tout d'abord en implémentant 3 taches lasers puis en tournant leur états de polarisation le long de l'OH. Les résultats expérimentaux correspondent au modèle computationnel et donc la conception de l'interface optique est validée.

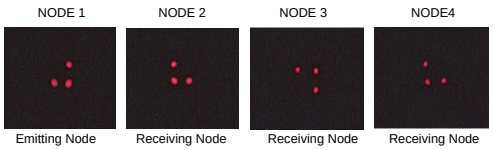


Figure 4: Résultats expérimentaux

Contents

" If you perceive that there are four possible ways in which something can go wrong, and circumvent these, then a fifth way, unprepared for, will promptly developp."

Murphy's Law

1 Introduction

A parallel computing architecture allows a collection of processors to communicate and cooperate to solve problems with a large efficiency.
However, the massive amount of data transport (i.e large amount of links) this architecture implies makes the electronic interconnections very difficult to set. On the other hand, it doesn't provide a large bandwidth due mainly to the line capacitances [1]. That is the reason why people focus more and more on optical interconnections to provide both a high number of channels (several hundreds [2]) and a high bandwidth.

1.1 Optical Interconnection at OIC

Amongst the different optical technologies, OIC (*Optically Interconnected Computing Group*[1]) focusses free-space optic interconnections since fibers bring about many coupling and multiplexing problems for a large amount of links.
The particular system OIC wants to use to make the interconnections is a combination of a "hard wired" optical system and an electronic reconfigurable system.

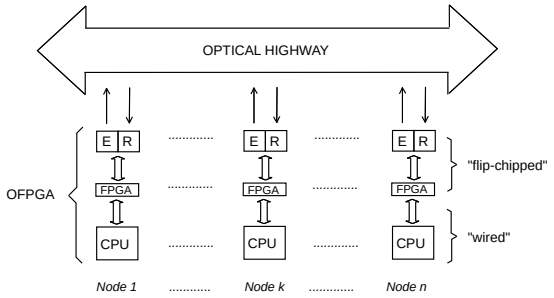


Figure 1.1: Sketch of all the Optoelectronic Interfaces. *E=Emitters, R=Receivers, FPGA=Reconfiguration structure, CPU=Processing system.* All these elements form a "sandwich" structure called a Node.

[1]School of Engineering and Physical Sciences at Heriot-Watt University, Edinburgh

The optical interconnections are realised by an optical system called "Optical Highway". This optical system, which has been invented at Heriot-Watt University, uses free-space optic and polarisation configuration to direct the optical signal throughout the array of processing systems (see section 2.1).

The reconfiguration system is an FPGA (*Field Programable Gate Array*) which underlies the optical interface (emitters and receivers) thanks to the use of flip-chip process. This reconfigurable logic structure (see section 2.2.2) allows to set different connection patterns on identical chips (and so emitters and receivers which communicate).

The optical interface that communicates between the different processing units is set by arrays of VCSELs (*Vertical Cavity Emitting Lasers*) and PDs (*Photodiodes*) which board is directly "flip-chipped"[2] on the FPGA to form altogether a Smart-Pixel Node also called OFPGA(*Optical Field Programable Gate Array*).
This chip design is part of the "POCA[3]" project currently on going at Heriot-Watt.

1.2 Aim of the project

The concept of adding a reconfigurable structure to the processing unit and "flip-chipping" the optical interface on it is a solution which OIC still haven't assessed all the benefits. One of the biggest issue regarding the Optical Highway is that, depending on the design of the optical interface[4], it both involves a low efficiency (amount of active structures which are working at the same time on the optical interface) and a high amount redundancies (shared channels)[5].

The first aim of this project is to know if we can increase the efficiency on this specific node by using the reconfiguration possibilities, and on the other hand if we can reduce redundancies.
Depending on the results on the redundancies assessment, we should choose to use different polarising structure through the Optical Highway:

- A static one (*HWP*[6]) if we can avoid any redundancies by the use of the FPGA
- A dynamic one (*LCD*[7]) if we can only reduce the redundancies by the use of the FPGA

In order to make conclusions on these demands, we chose to implement the Optical Highway on and throughout the OFPGA nodes for a *Completely Connected Network*

[2]Actually only the VCSELs board is flip-chipped on the FPGA since the PDs can be made on the same substrate as the FPGA.
[3]Programable Optoelectronic Computer Systems
[4]The optical interface is defined in section 2.2.1
[5]Both of this concept are defined in details in section 3.4 and 3.5
[6]Half-Wave Plate
[7]Liquid Crystal Display and more particuliary Twisted Nematic Structure

Topology (CCT) regarding efficiency and redundancies issues.

The main steps of this project are:

- *The design and mapping of the emitters and the receivers on the nodes (to set the optical interface) and through the Optical Highway for any kind of Topology.*

  This part should give us a computational model of how the different interconnections behave through the entire system.
  It first implies an assessment of the current solutions OIC already has[8] and should provide a very versatile design for any kind of network topology.

- *The Implementation of a Completely Connected Network Topology (CCT or Fully Connected Topology[9]) regarding the efficiency aim and underlying issues[10] as well as the best use of the polarising structure[11].*

  This step includes the simulation of the connection pattern on the chip and on the polarising structure. It then enables us to assess the redundancies which occur.

- *The Experimental assessment of the computational model through the Optical Highway.*

  We implement a pattern of connection through the Optical Highway for a few number of nodes to confirm the validity of the design.

[8] See chapter 4
[9] See Appendix 1
[10] See chapter 3 "Analysis of the mapping problem"
[11] See Chapter 2

2 The Optoelectronic System

Optical and electronic technologies that enable optoelectronic interconnections are many. Within the frame work of this project, we introduce the different technologies that OIC wants to use and which exist right now (Optical Highway) and which should take place in the sequel of this project (Flip-chipped nodes) to enable us to set a further mapping on the entire optoelectronic system. A large survey of the different technologies can be seen in details in[2],[3],[4] .

2.1 Optical Highway

The main interconnection structure which takes place in this project is an optical system called the Optical Highway.
This optical system enables optoelectronic interconnection by using free-space optic to drive the optical signal and polarisation configuration to direct it to the right destination. It envisages the use of a large bandwidth low latency combined with a large amount of channels [2],[4].

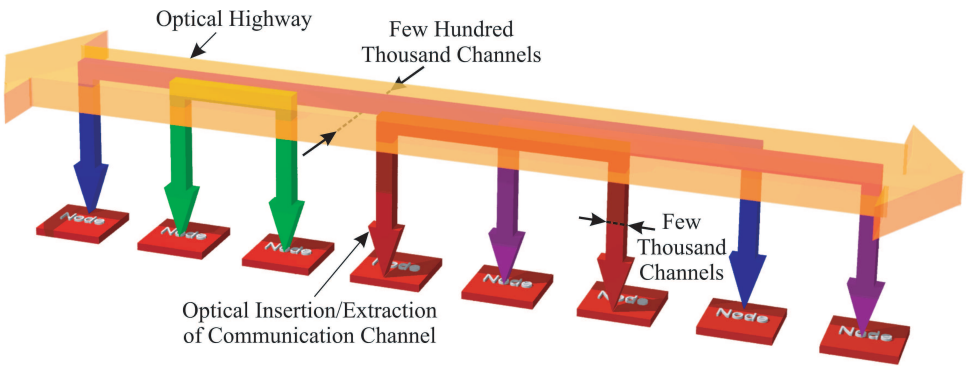


Figure 2.1: General Schematic of the Optical Highway with random links (different colours).

The schematic 2.1 shows us random possibilities of interconnections between different

processing systems depending on a choosen network topology.
The signal is only routed by the polarisation state of the optical signal which can be changed within the optical highway.
The optical highway uses different optical systems to direct the optical signal:

- *A 4-f lens system* combined to a couple of *polarising beam splitters* to "transport" and focus the optical signal.
- *Polarisors* to set a linear polarisation state to the optical signal at the input of the optical highway.
- *Polarising structure* to change selectivitely the linear polarisation state by 90 degrees within the optical highway.

As we will see the signal "transport" is realised by an imaging system (lenses) and that will bring the main issues for the design of the optical interface (see chapter 3).

2.2 The Nodes

We call the optoelectronic part of the system the Smart-Pixel "Node" which gathers 2 different structures in a "sandwich" we choosed to call "Node" regarding the computing literature. These three structures are:

- *The optical interface* which is the optical part of the nodes and includes emitters (VCSELs) and receivers (PDs) arrays.
- *The Chip* which is the entire electric part of the node and includes the processing system as well as the reconfigurable logical structure (FPGA).

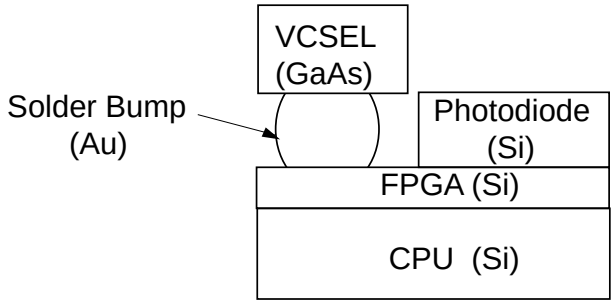


Figure 2.2: Schematic of the Node. CPU, FPGA and PDs are grown on the same substrate when VCSELs are "flip-chipped" on the FPGA

2.2.1 The optical interface

Emitters

The *emitters* (fig. 2.3) which are planned to be used are VCSELs (*Vertical Cavity Surface Emitting Laser*) which enable an easy integration on the chip and through the Optical Highway .
They emit perpendicular to the surface of the chip and do not need dicing compared to common laser diode. They enable smaller structures and consume less power[2]. On the other hand, a high density can be reached and now it becomes common to find 8x8 or 16x16 arrays. We can expect these sizes to be much larger depending on the increasing reliability of the process technologies. Concerning the wavelength, OIC planned to use 850nm VCSELs since the other 2 windows of the telecommunication wavelengths (1.3 and 1.55 μm) are still tricky to do technologically[1]

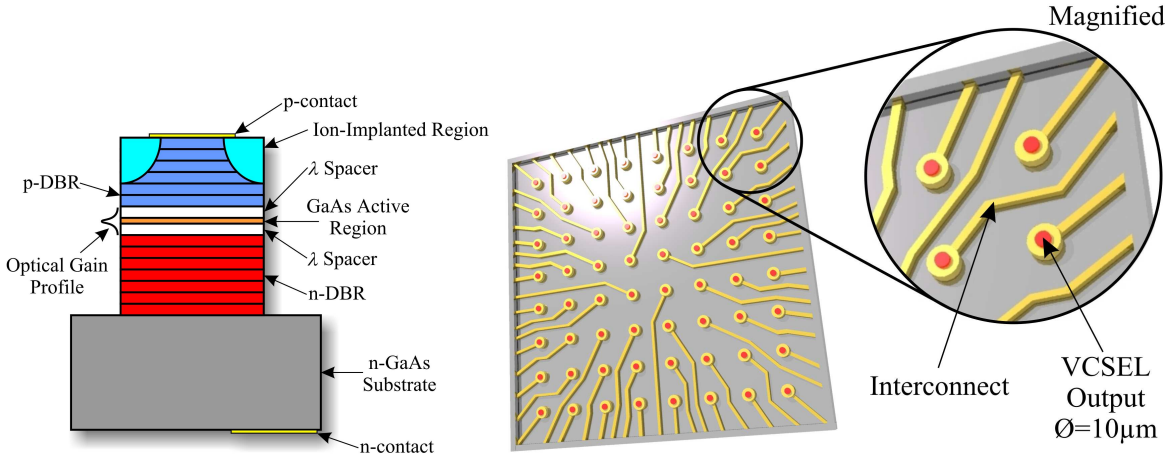


Figure 2.3: Detailled Structure and Array of the VCSELs [2]

Receivers

The *receivers* (fig. 2.4) which are planned to be used are PDs (*Photodiodes*) which again enable an easy integration on the chip. Indeed, they can easily be fabricated in large arrays using existing technologies and can be made on the same substrate as the FPGA [2].

[1] Many research groups are currently focussing on VCSELs of 1.3μm with good results, however these VCSELs still remains difficult to fabricate but should be widely available in the further months.

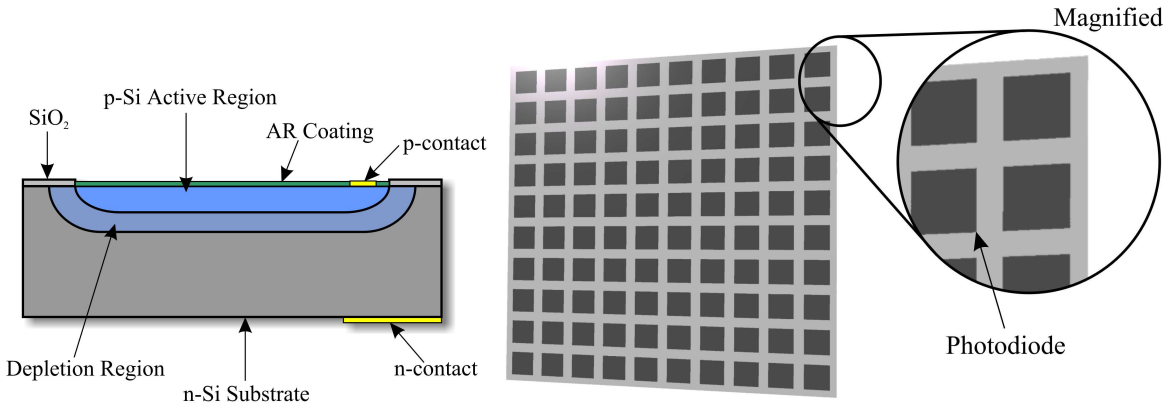


Figure 2.4: Detailled Structure and Array of the PDs [2]

2.2.2 The reconfigurable structure

The reconfigurable structure is an FPGA (*Field Programable Gate Array*).
This device is an integrated programable logic structure. It enables to program any kind of connections between the processing unit and the optical interface.

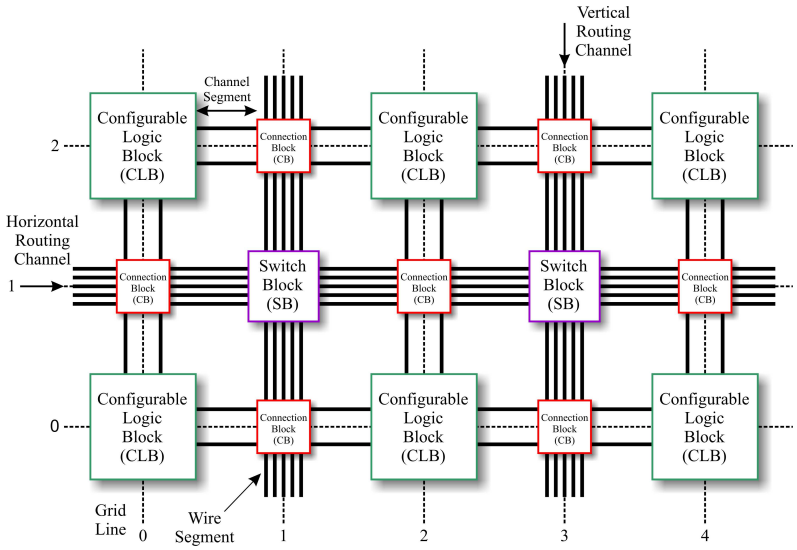


Figure 2.5: The FPGA. A combination of a Configurable Logic Block add to a Switch Block is set above each pins of the processing unit[3].

Let us consider the previous system OIC theoretically assessed, i.e a processing unit with an optical interface.
Each pin of the processing chip is dedicated to an emitter or a receiver.

However, by adding a reconfigurable logic structure, we enable to dedicated any pin to any emitter of receiver without changing the design of the processing unit.
By using the FPGA, we can set any connection pattern on the processing unit and so specify from which other node the output signal of the optical highway comes.
This degree of freedom improves the flexibility of the entire system. This is the first time the OIC introduces the reconfigurable structure combined with the Optical Highway and they want to see (as we will see further) how does the system behaves and what better properties we can get when implementing a choosen network topology on the entire system.

2.2.3 Working Model

A optical signal (controlled by the processing unit and electrically directed by the FPGA) is emitted from a VCSEL on the optical interface which takes place at the focal length of the first input lens of the Optical Highway. The optical signal is then linearly polarised by a polarisor and collimated by the first input lens. It enters the Optical Highway thanks to a 45 degrees polarising beam splitter.
A patterned polarising structure offers the possibility to direct the signal to the next node by leaving the linear polarisation state unchanged or to direct it to the following nodes by turning the linear polarisation state of 90 degrees. By this way, switching the polarisation state of 90 degres in the optical field means to set a electrical state of "0" or "1" on the next node.
After going through the polarising structure, the optical signal is reflected by the polarising beamsplitter and focalised on the next node depending on its polarisation state. The scheme of operating is repeated as many times as many nodes we have and depending on the polarising pattern on the polarising structures, we can decide or to keep the signal in the Optical Highway by "switching" its polarisation state by 90 degrees or to direct it to the neighbour node by leaving its polarisation state the same[2].

[2]As the fig. 2.6 illustrates it, once the polarisation state turned to keep the signal into the Optical Highway, we have to turned it again to go out of the Optical Highway.

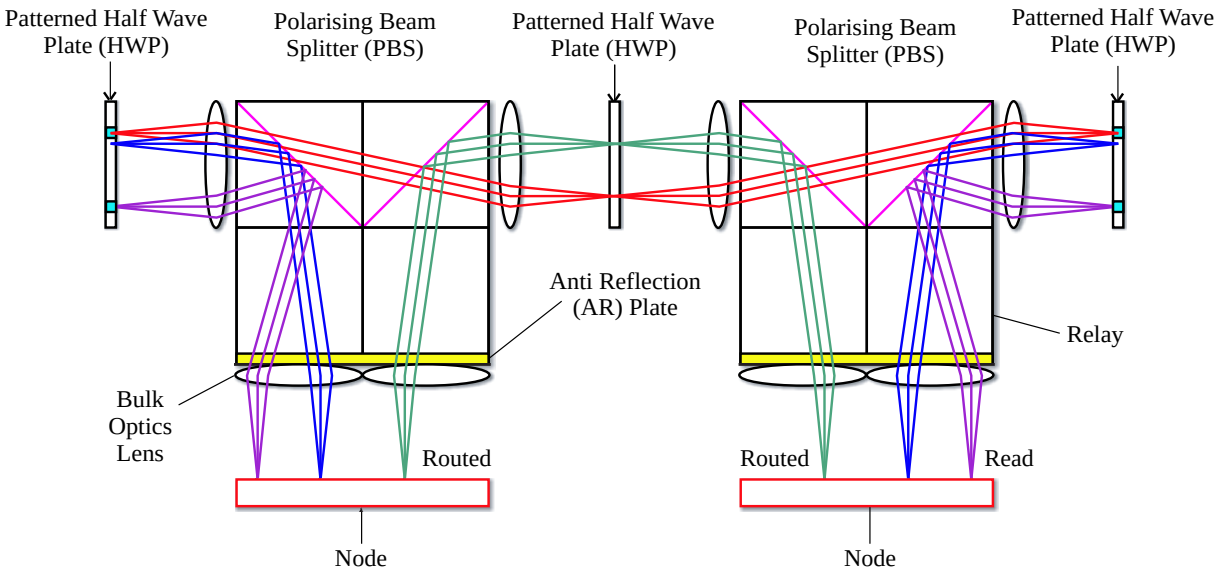


Figure 2.6: *Red Ray* has a linear polarisation state turned by 90 degrees by the polarising structure at the first place and so is not reflected by the first beam splitter and is directed to the following nodes. At the second stage, its polarisation state is turned again to be able to be reflected by the PBS of the next stage and so focalised on the corresponding node. *Pink Ray* has gone through many stages (remained in the Optical Highway) and has its polarisation state turned by 90 degrees to be able to be reflected again by the PBS and focalised on the right node. *Blue and Green Rays* illustrate links between 2 neighbour nodes and so the polarisation state remains the same through the polarising structure.

3 Analysis of the mapping problem

One of the main issue with the implementation of the Optical Highway is the mapping of the emitters and receivers on the nodes and of the nodes throughout the optical highway. Indeed, both must fill different requirements from the basic optic to the computing implementation.
This part formulates the mapping problem by identifying in details these different requirements and issues involved in any implementation (of any network topology) of the Optical Highway[1].These requirements also aimed to enable us to set a computationnal model of what happened through the entire system as well as to compare and assess the current solution OIC has at the moment[2].

3.1 Optical requirement

The optical system uses optical relay system (several pair of lenses), mirrors, polarisors[3], emitters (VCSELs) and receivers (PDs). Each of these devices implies different issues that put constraints in the design and mapping of each nodes.

3.1.1 Image Reversing Optical System

The optical relays uses throughout the optical system are couple of 4-f lens system. The input and output lenses are simple converging lenses when the effective relay ones are barrels of several lenses. Theses optical systems packaged in a barrel can be seen from the computationnal point of view as simple converging lenses[4].
In that configuration (4-f lenses system) a single pair of lenses provides an image which has a reversed orientation compared to the object when a couple of pair of lenses provides

[1] Of course, the requirements detailled here do not represent the totality of the requirements the entire system should fill (for instance, we deliberatly omit all time, latency requirement regarding data transport and synchronisation). However, they are the key ones and give us the basis for any further design or assess of any mapping solution.

[2] See chapter 4

[3] To set the computationnal model we set the optical elements as their function in the entire system. Of course, mirrors are actually polarising beam splitters and polarisors includes HWP or LCD as well.

[4] However, these barrels have been designed in the field of the Ph.D thesis of C Barrett [5] to limit aberration for a wavelength of 850nm.

an image with the same orientation. This implies that each image of a node through a

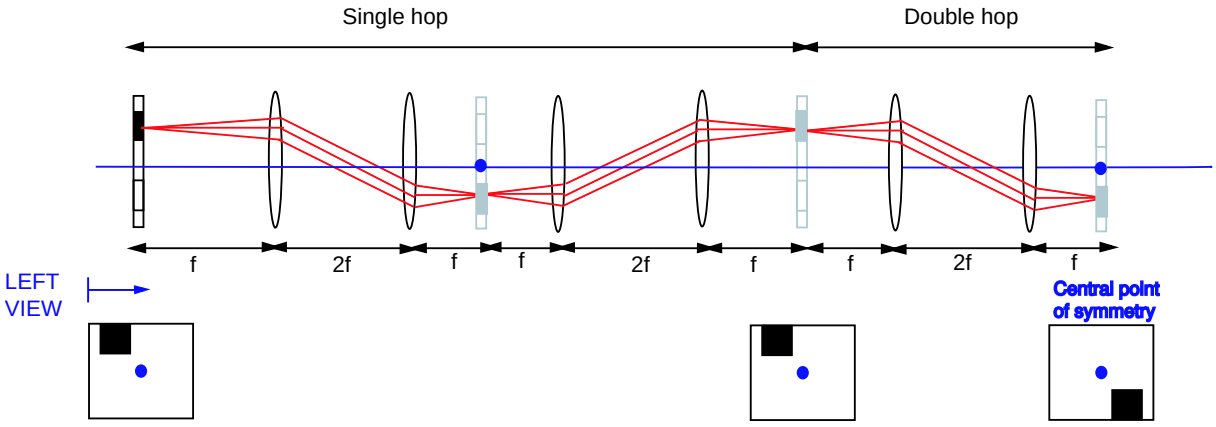


Figure 3.1: Reverse Imaging system. *For a single hop the image on the next node has a reversed orientation. For a dobble hop the image orientation remains the same as the orientation of the object.*

single hop (4 lenses) will have the same orientation on the next node. And of course each image of a node through a double hop (6 lenses) will be reversed, i.e it will be *central symmetric*[5] to the "object-node".

For the computationnal model, we can set the following statements[6]:

- The image of a node through $4(n+1)$ {n=0,1,2,3, ...} number of lenses i.e an odd number of hops has the same orientation as the "object-node".
- The image of a node through $2(2n+1)$ {n=0,1,2,3, ...} number of lenses i.e an even number of hops has an *central symmetric* orientation.

3.1.2 Mirror Symmetry

The effect of the mirrors on the image is an *axial symmetry* operation with the axis perpendicular to the optical highway and half way to each connected nodes.

[5]The central point of symmetry is the center of the node and of the Optical Highway

[6]We set the effect of the 4-f system and of the mirror separately since it doesn't occur on the same dimension. This conclusion are only true for a system made of 4-f systems without mirrors.

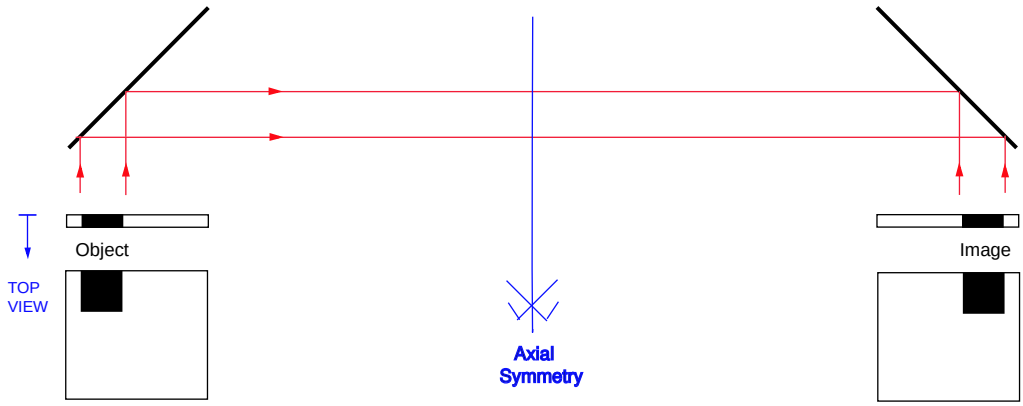


Figure 3.2: Flip imaging system. *Whatever is the number of hops, the image is flipped vertically (axial symmetry) compared to the object*

3.1.3 Power, Aberrations

Of course the main issue once the computationnal model set will be both power consumption and aberrations through the optical highway and the entire system. A large survey of these two very important facts are detailled in [2].
We won't detail them in the field of this study since, as we will see later, we test the computational method on a small array.
However, these points will be the main ones for a sequel to this project and we have to keep in mind they exist and will be the final factor of quality of a further physical design at a large scale.

3.2 Computing Versatility requirement

The main "computing" requirement for the mapping of the nodes is that it must be as versatile as possible. Indeed, we want to implement network topologies which could be totally opposite. For instance the Completely Connected Topology (CCT, see fig 3.3) uses the largest number of links among all the existing network topologies when the Hypercube Topology (HT) uses the fewer number.

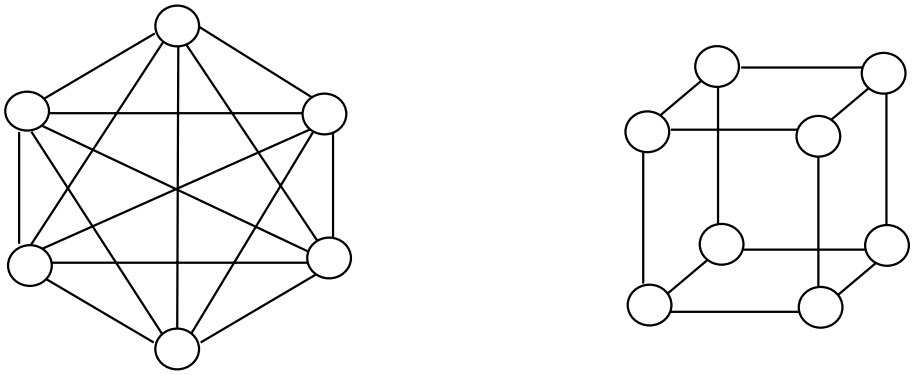


Figure 3.3: Graphical representaion of the CC and Hypercube Topologies. The O are processing units when the – are bidirectionnal links regarding the computing litterature[6][7]

3.3 Packaging requirement

The packaging requirement is closely linked to the technology requirements. CMOS technology cannot provide all the types of design. And of course some things are impossible to fabricate in CMOS for the moment. So we have identified two main requirements regarding the technology:

- *All the nodes must be the same*: This is of course linked to the cost several masks and design would imply. The more replicable the node, the cheaper the system will be .
- *All the nodes must be square*: This fact has to do with both processing and FPGA system which are square and underly the optical interface. It would be tricky to imagine an optical interface which doesn't fit the architecture underlying.

3.4 Efficiency of Nodes

We call efficiency of a node the amount of possible "working" structures (i.e emitters and receivers) on at the same time.
This is the key requirement this project focusses on.
The design (and mapping) of the nodes must bear the limit case where all the emitters (resp. receivers) on a chip are emitting (resp. receiving) at the same time. This will be the final factor of quality of the implementation.
The larger amount of "active structures" which are able to work at the same time works the better will be the efficiency of the entire array.

3.5 Redundancy

Depending on the design and of the mapping of the nodes, many redundancies occur.
The redundancies affect two devices on the entire system.
The first one is the polarising structure where several optical signals can be focussed on the same "pixel[7]" and so direct to the same destination when each signal should be directed to another destination. The receivers where the different emitting signals arrive and so the processor cannot then identify the origin of the signal.
Physically, they represent shared channel and we want to see if it is possible to avoid them by using the dynamic systems (FPGAs or LCD) or a sensible design. However, previous designs have shown that they seem to be unavoidable for a satisfactory efficiency (see next chapter) and so they have to be analysed, assessed and controlled.

3.6 Formulation of the computationnal model

Regarding the different requirements we set previously the formulation of the computationnal model would be:

> "To evaluate the connection between two nodes, we have to combine both *central and axial symmetry operations* depending of the number of hops and then to set that the image of the emitter is equivalent to a real physical receiver for a versatile, identical and square node design. Everything aimed ideally for maximum efficiency and minimum redundancy".

[7]We consider the spot size of the optical signal = the pixel size for pure theoretical simplification. Of course the spot, due mainly to the aberration can take several pixels size in pratic on the polarising structure.

4 Assessment of the current solutions

OIC already have two solutions to the mapping problem, which are still not satisfactory from the point of view of the efficiency and do not include the reconfiguration possibilities (FPGA underlying). The different requirements we have set previously will help us to find the weaknesses and strength of each one to be able to design a third one (see chapter 5) regarding packaging, reconfiguration possibilities and efficiency requirement.

4.1 Solution 1: Mapping by shifting

This mapping solution uses the physical shift of each (identical) chips along an entire Optical Highway and implement a CC Topology. All the chips are the same, the emitters are on one line in the middle of the chips when the receivers are arranged on two diagonals. Each chip has 2 times more receivers that emitters.

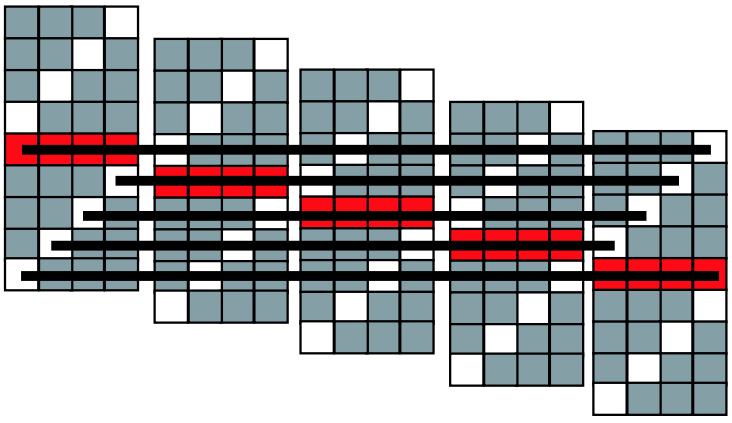


Figure 4.1: Mappping solution by shifting *Red square = Emitters, white square = receivers, grey square = unused space, black lines = optical highway.*

The advantages of this mapping is that it uses only one Optical Highway, and above

all that it provides a design that avoid any shared channels for a CC Topology.
Indeed, this is the physical shift of each chip vertically to the linear Optical Highway which enable to avoid any redundancies.

The main disadvantages are first the unused space on each nodes, the low efficiency of each node and of course the physical shift add to the rectuangular shape of each nodes which implies a tricky implementation respectively of the nodes through the Optical Highway and of the FPGA.
We can assess the efficiency on the nodes for a large number of nodes.
Let us admit we have rectangular nodes of n "square[1]".
The total space of each node would be : $n^2 + n^2 + n = (2n+1)n$ when the total working structure area would be: $3n$.
Let us call the possible nodes "connectable" by this design *N=(n-1)* we obtain an efficiency of :

$$E = \frac{3n}{(2n+1)n} = \frac{3}{2N-1} \quad N = \{3, 4, 5, ...\} \tag{4.1}$$

Which graphically provides:

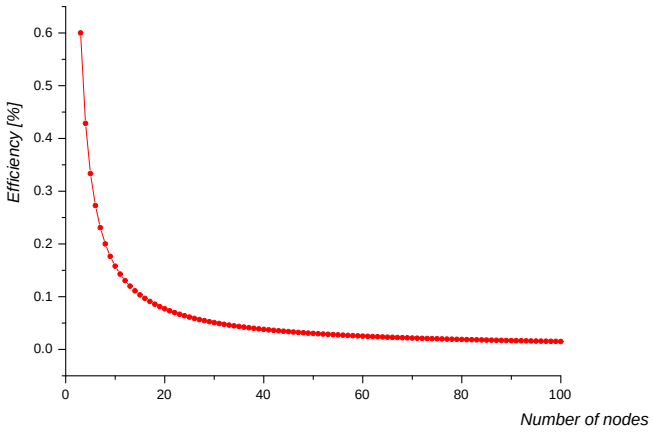


Figure 4.2: Efficiency regarding the number of nodes

We can see for instance that up to 30 nodes the efficiency is under 5%. That means that 95% of the possible working structures (seats) are unsued. This means this mapping solution only fills our requirement for a very small array of chips and would be too expensive to fabricate for a large scale array[2].

[1]Possible working structure seats, see fig. 4.1

[2]Another version of this mapping has been assessed to avoid the shifting down of the chip and to allow a 3-D design. However, as it uses an optical shifting system, it was to difficult to implement optically. It can be seen in details in [8]

4.2 Solution 2: Mapping 2-D

This mapping solution uses identical square nodes on a 2-D mapping. It directs the signal in 2 perpendicular directions and implements a Hypercube Topology.

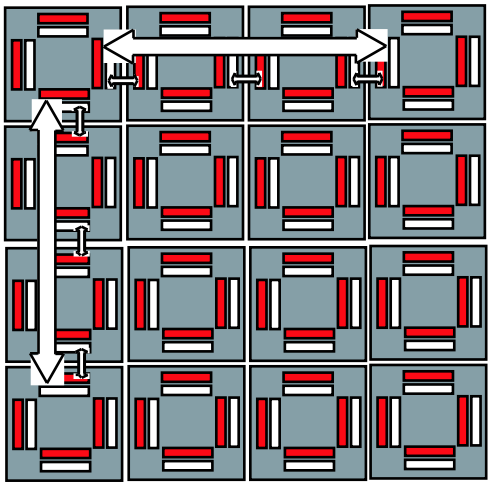


Figure 4.3: Mappping solution 2-D *Red array = Emitters, white array = receivers, grey space = unused space, white arrows = optical highways.*

The main advantage of this solution is first to be a 2-D mapping and so easy to expand without loosing the efficiency of each nodes, on the other hand it provides an efficiency per node of 100 % except for the ones on the edge of the array with only 50%. Which give us a total efficiency for the implementation of the Hypercube Topology of[3] :

$$E = \frac{4 \cdot 0.5 + 12 \cdot 1}{16} = 87.5\% \tag{4.2}$$

The disadvantages are that it doesn't fill the packaging requirement because of the mapping of each emitters and receivers on the edge of the chips (difficult to set the FPGA undelying the structure) and on the other hand it would be tricky to implement a CCT topology on it and keeping the same efficiency.

However, this solution remains a satisfactory one for the implementation of a Hypercube Topology[4].

[3] One had to notice that the design the fig. 4.3 is the largest one possible regarding the Hypercube Topology. Expanding of this design is done by connecting several shape of fig. 4.3 which in opposition to the previous solution provides us a constant efficiency. On the other hand this results is the very best theoretical result we can set. However, regarding the computationnal model, we should expect a lower efficiency due to the even connections links.

[4] A more detailled description can be seen in [9] with a possibility to implement a Hypercube Topology+ which provides more links.

4.3 Summary of the assessing

This table shows us the weaknesses and strength of each solution, it enables us to design another solution on which we will implement different topologies and on which we will simulate on.

Requirements	Shift	2-D
Optical	++	++
Computing	CCT	Hypercube
Packaging	– –	– –
Efficiency	–	++
Redundancies	++(None)	– –

Table 4.1: Summary of the assessing

Note: - No assessment on the above mapping solution have been assessed for another topology that the one written which do them few versatile. We cannot expect the same characteristic for another topology implemented.

By this general assessment we are now able to design a new solution based on the general ideas of these both design by using the best the reconfiguration possibilities (FPGA) to increase the efficiency.

5 Highly symmetric optical interface design

The design of the optical interface must fill all the requirements we have discussed before (chapter 3). This part explains in details how it has been design and why it fullfills these requirements.

5.1 Design of the optical interface

Regarding the computational model we set (sect. 3.6) we define 2 connections cases:
For an odd number of hops the image will be only reversed axially (due to the mirrors) and so both nodes must be the same to make meet imaged emitters with physical receivers.
For an even number of hops the image will be both reversed axially and centrally and so the nodes must be reversed to meet imaged emitters with physical receivers (see fig. 5.1).

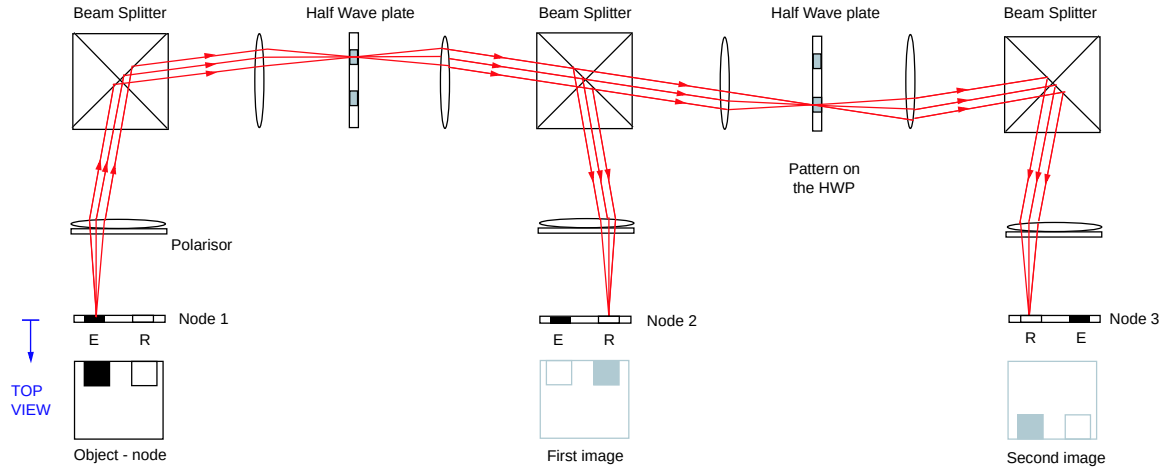


Figure 5.1: Reverse Imaging system for an unidirectional link. *For a single hop the image on the next node is just axially symmetric (mirrors) to the source node. For a double hop the image orientation is both centrally (lenses) and axially (mirrors) reversed compared to the source node.*

Now, regarding the packaging requirements (sect. 3.3), let us take a square chip with 16 pins on which have implemented the reconfigurable blocks which enable to connect any pin to any VCSELs or PDs[1]. Let us place one emitter on the upper left corner (5.2 (a)). Regarding the optical requirement, for each odd hop (1,3,5, ...) the optical system will provide an image on the node $2(n+1)\{n = 0, 1, 2, 3...\}$ that is axially symmetric (vertical axis) to the node n.
If the emitter on position 1 of the node n is working, it will be image on the node number $2(n+1)\{n = 0, 1, 2, 3...\}$ on the position 4 (fig 5.2 (a)). Let us place a second emitter, next to the first one, it will be imaged on the position 3 and so on.
But as each link must be bidirectional we have to do the same operation with the emitters of the "receiving" node.
In this way, the emitter on position 1 and 2 on the "receiving" node (which is now the emitting one) will be imaged on the position 4 and 3 (resp.) of the first "emitting" node (which now receives). This position must be filled with a "physical" receiver (fig 5.2 (b)).
We finally obtain this chip design for each odd hop (fig. 5.2 (c)):

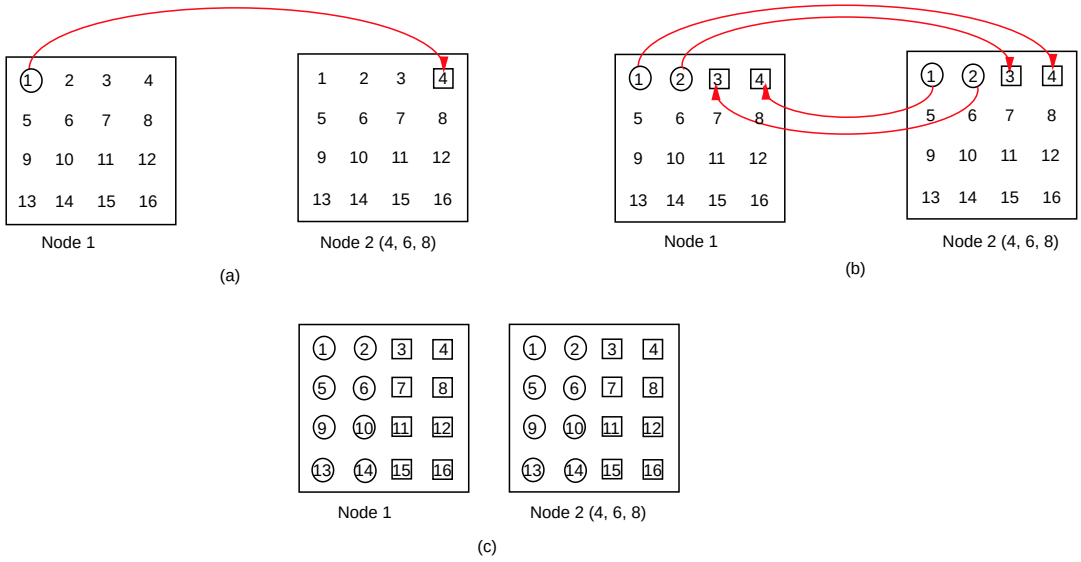


Figure 5.2: Building of the connection pattern for an odd number of hops.
O= VCSEL, Boxes = PDs.Top view.

On the other hand, for each even hop (2,4,6, ...) the optical system will provide an image on the node number $2(n+3)\{n = 0, 1, 2, 3...\}$ that is the composition of a central and an axial symmetry (vertical axis) which gives in fact a "horizontal" axis symmetric image to the node n (fig. 5.3 (a)).
If the emitter on position 1 of the node n is working, it will be image on the node $2(n+3)\{n = 0, 1, 2, 3...\}$ on the position 13. And the emitter on position 2 will be imaged

[1]Every pins-reconfigurable block is dedicated to only one VCSEL or PDs

on the 14 and so on(fig 5.3 (b)).
We finally obtain this optical interface design for each even hop (fig 5.3 (c)):

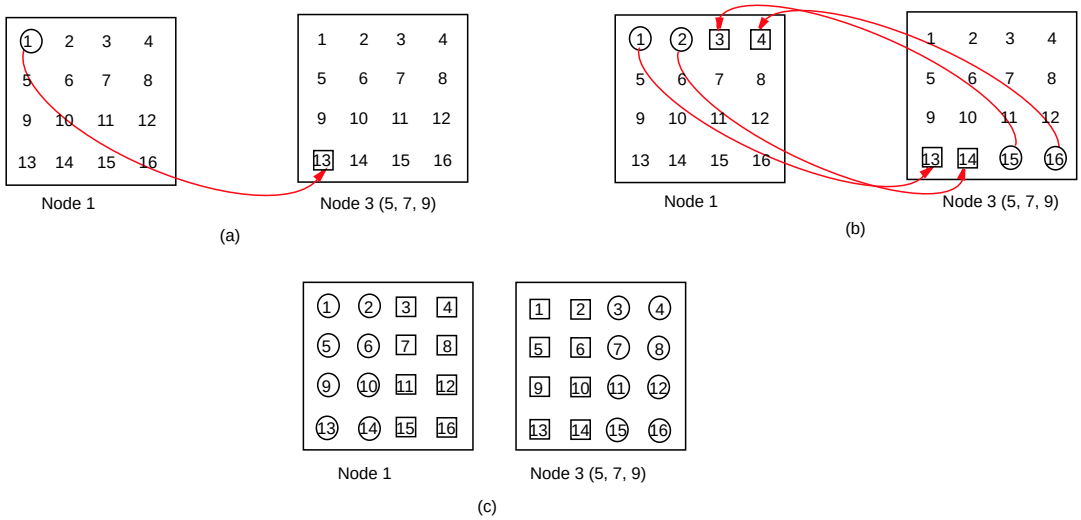


Figure 5.3: Building of the connection pattern for an even number of hops. O= VCSEL, Boxes = PDs. Top view.

Now a big issue still remains. We have found a fair placement of all the emitters and receivers for a single (odd number of) hop or a double (even number of) hop. But only for a bidirectionnal direction from node 1 to node 2 and 3. And of course if we place the 3 nodes on a same array, the communication between node 2 and node 3 will be impossible regarding the computational model(fig 5.4).

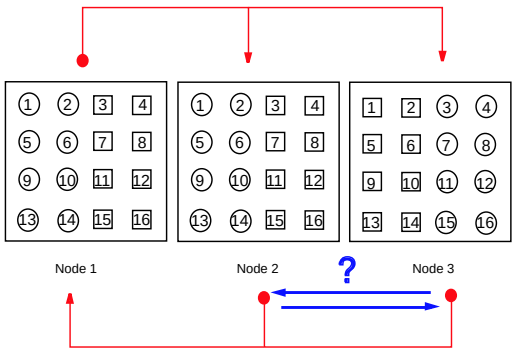


Figure 5.4: Issue in building of the connection pattern. O= VCSEL, Boxes = PDs. Top view.

So, to fullfill the last packaging requirement which impose all the nodes to be identical and to allow the setting of the last communication links between the intermediate nodes, the only solution which occur from the previous analysis is an optical interface with the upper half part reversed compared to the half bottom one (fig 5.5). Thanks to this

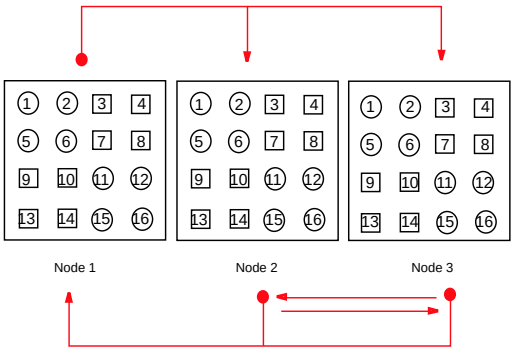


Figure 5.5: Building solution. O= VCSEL, Boxes = PDs. Top view

design, we can use this optical interface for all nodes throughout the optical array(fig 5.6).

A node involving n emitters on it can theoretically be connected to n other nodes to form an array of $n+1$ nodes.

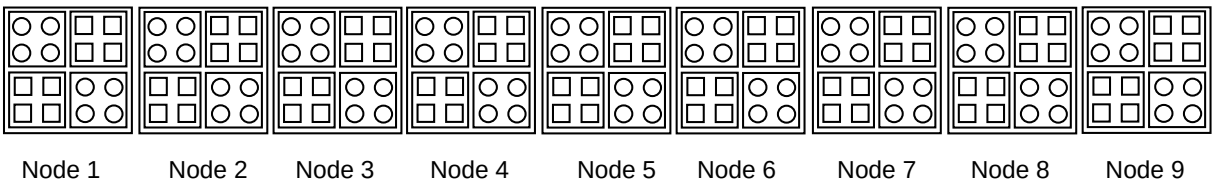


Figure 5.6: Final design of an array of 4x4 optical interface

5.2 Array scaling

The array of fig 5.7 is the maximum nodes you can theorectically connect regarding the choosen square nodes (4x4 active structures).
However, we can expand the array[2] with the respect of the packaging requirements by using bigger square nodes (6x6, 8x8, ...).

[2]This scaling represents only the scaling of the array of nodes, however, a 2-D expanding can also be set regarding the connection pattern we are implementing on the array (chap. 6).

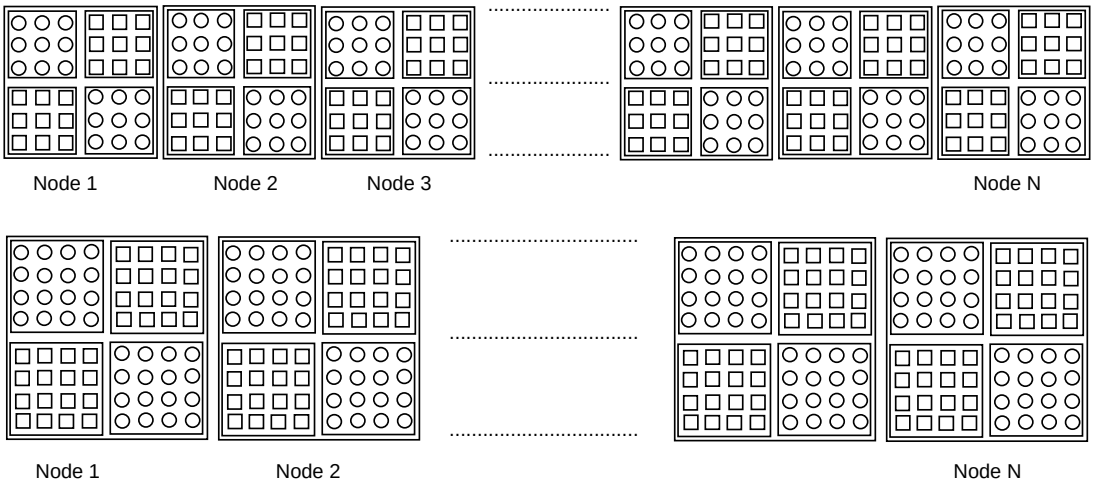


Figure 5.7: Scaling of the array. *A square node of 6x6 has 18 emitters on it and can be connected to 18 other nodes to form an array of N=19 nodes. A square node of 8x8 has 32 emitters on it and can be connected to 32 other nodes to form an array of N=33 nodes*

5.3 Expanding limits

Theoretically the expanding limits directly depends on the choice of the topology to implement.
As we focus on the efficiency, we will see (chap. 6) that it is not always possible to implement a CCT topology with only one emitter connected to one receiver i.e connection links without any redundancies on the optical interface.
So the theoretical expanding limit will be given by the simulation part of this project.

Experimentally this is of course both power loss through the optical highway and aberrations which will set the expanding limits for the implementation.
Both assessment don't take place in the current project but will be the next step of the sequel of the current experimental setup[3] we set.

[3]see chapter 8

6 Implementation of a CCT Topology

The Completely Connected Topology implies each node is connected to all the other ones. Because of the different imaging issues throughout the system, we have introduced before, it is very tricky to set a connection pattern with a high efficiency on each node i.e. without any redundancy on the optical interface .
The redundancies on the optical interface physically represent several signals arriving at the same destination receiver. As a receiver only takes into account the power of the signal, it would be impossible for it to know the origin of each signal it receives.

In order to set a right connection pattern on each nodes *without redundancies on the optical interface* (*with the best efficiency*) we develop a program to completely-connect each nodes together.

6.1 Program implementing a CCT on the optical interface

This program is written in *Matlab* and can be seen in details in appendix when its electronic version is $connection\_pattern\_final\_top.m$.

6.1.1 Parameters

There are only 2 parameters to enter the program:

- `v` : *numbers of active structure*[1] *on a length of a unit square*
- `nodes`: *numbers of nodes to connect together*

That means this program is versatile and we can use it for any expanding regarding the architecture we have defined before.
However, for the following explanations we take an array of nodes with 4x4 active structures, expanding from 3 to 9 nodes.

[1] i.e both VCSELs and PDs

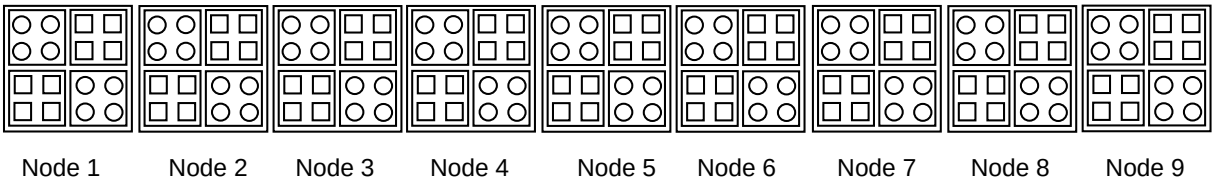


Figure 6.1: Array tested. `v=4, nodes= 1-9`

6.1.2 Method

The program first defines a large `Matrix` which represents the array of nodes connected together.
It then creates a matrix of permutation `I` of the possible connections to the neighbour nodes. For instance, let us take the first node emitting to all the other nodes, the possible connections are the links to the following nodes: `2,3,4,5,6,7,8,9`. And so the matrix of permutation will be :

$$\texttt{I=}\begin{pmatrix} \texttt{9,8,7,6,5,4,3,2} \\ \texttt{9,8,7,6,5,4,2,3} \\ \cdots \\ \texttt{2,3,4,5,6,7,8,9} \end{pmatrix} \tag{6.1}$$

Then it picks up the first node and fills it with the first permutation of the matrix `I`. It builds the connections to all the other nodes regarding the reversing issues we have identified earlier. A zero means a free emitter/receiver (see fig. 6.2).

```
0 8 8 0   0 0 0 0   1 0 0 1   0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0   0 1 1 0
7 6 6 7   0 0 0 0   0 0 0 0   0 0 0 0   1 0 0 1   0 1 1 0   0 0 0 0   0 0 0 0
5 4 4 5   0 0 0 0   0 0 0 0   0 1 1 0   0 0 0 0   0 0 0 0   1 0 0 1   0 0 0 0
3 2 2 3   0 1 1 0   0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0
```

Figure 6.2: Step 1: Pattern on the first node (line of the matrix of permutation) implemented and connections to the other ones set.

After having finished to connect the first nodes to the other ones, it creates another permutation matrix `I` with the possible connections to the following nodes `3,4,5,6,7,8,9`[2] and tries to connect the second node to all the other following with implementing the first line of the matrix `I`. If there are redundancies, it erases all the connections with the following nodes as well as the connection pattern on the second node.
Once done it takes the second line of the permutation matrix `I` and try to implement it. If there are redundancies it chooses the third line of `I` and so on, if there are no redundancies, it keeps on the implementation with the following node.
And so on until, the entire array is connected i.e all the nodes are connected to each other.

[2]For a lower number of node, we keep a constant length for each line of the permutation matrix by filling it with "free" connection links ("0" state) to be able to test absolutely all the connection possibilities

That means, the program creates permutations matrix, regarding the emitting nodes, of length[3]:

`node number`	Length of the Matrix of Permutation
1	$8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 40320$
2	$7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5040$
3	$6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$
4	$5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$
5	$4 \cdot 3 \cdot 2 \cdot 1 = 24$
6	$3 \cdot 2 \cdot 1 = 6$
7	$2 \cdot 1 = 2$
8	1

Table 6.1: Connections possibilities

That means the program is able, regarding the number of `nodes` (here = 9), to test these possibilities of connection:

$$4320^8 = 6,98 \cdot 10^{36} \quad \text{THEORETICAL CONNECTION POSSIBILITIES} \tag{6.2}$$

6.2 Results: Connection pattern on the optical interface

After running the simulation with *connection_pattern_final_top.m*, we pick up the following connection patterns for a number of nodes from 5 to 8, when among all the connection patterns possible for connecting 9 nodes together without redundancy, the program gives no result !

[3]We consider 8 emitting nodes, the 9th one is entirely defined by all the connection links to the other ones.

Connection pattern on the optical interface **Number of nodes: 5**

Node 1	Node 2	Node 3	Node 4	Node 5
0 0 0 0	0 0 0 0	1 0 0 1	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 2 2 0	1 3 3 1
5 4 4 5	5 4 4 5	0 5 5 0	0 1 1 0	2 0 0 2
3 2 2 3	3 1 1 3	2 4 4 2	5 3 3 5	4 0 0 4

Connection on the optical interface **Number of nodes: 6**

Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
0 0 0 0	0 0 0 0	1 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0
0 6 6 0	0 6 6 0	0 0 0 0	0 2 2 0	1 3 3 1	4 1 1 4
5 4 4 5	5 4 4 5	6 5 5 6	6 1 1 6	2 0 0 2	3 2 2 3
3 2 2 3	3 1 1 3	2 4 4 2	5 3 3 5	4 6 6 4	0 5 5 0

Connection pattern on the optical interface **Number of nodes: 7**

Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7
0 0 0 0	0 0 0 0	1 0 0 1	0 7 7 0	0 0 0 0	0 0 0 0	0 4 4 0
7 6 6 7	7 6 6 7	0 7 7 0	0 2 2 0	1 3 3 1	4 1 1 4	2 5 5 2
5 4 4 5	5 4 4 5	6 5 5 6	6 1 1 6	2 7 7 2	3 2 2 3	1 3 3 1
3 2 2 3	3 1 1 3	2 4 4 2	5 3 3 5	4 6 6 4	7 5 5 7	6 0 0 6

Connection pattern on the optical interface **Number of nodes: 8**

Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
0 8 8 0	0 8 8 0	1 0 0 1	0 7 7 0	0 7 7 0	8 0 0 8	0 4 4 0	0 1 1 0
7 6 6 7	7 6 6 7	8 7 7 8	8 2 2 8	1 3 3 1	4 1 1 4	2 8 8 2	3 7 7 3
5 4 4 5	5 4 4 5	6 5 5 6	6 1 1 6	2 8 8 2	3 2 2 3	1 3 3 1	4 5 5 4
3 2 2 3	3 1 1 3	2 4 4 2	5 3 3 5	4 6 6 4	7 5 5 7	6 5 5 6	6 2 2 6

The following table shows the efficiency obtained on the optical interface with the program of implementation depending on the number of nodes connected together. All the connection patterns have been found after testing a certain number of connections possibilities which is included in the factor "Iterations"

number of `nodes`	Iterations	Efficiency on the optical interface
5	4	50%
6	5	62.5%
7	4	75%
8	4	87.5%
9	$5.05\ 10^{15}$	$\oslash$

Table 6.2: Efficiency on the connection pattern tested.

Note: The following efficiency could be theoretically slighly increase. By connecting 2 arrays tested, we could be able to increase the efficiency up to 100%. However, it remains purely theoretical and many issues would occur experimentally. However, this is a solution to assess in a further sequel of this project.

All the connection possibilities have been assessed for a maximum number of node of 9 and the program cannot find any pattern without any redundancy on the optical interface. Regarding this results we can say with confidence that **it is impossible to fully connect 9 nodes without any redundancies on the optical interface.**.

The best result we have is the implementation of a CCT topology with 87.5% of efficiency on the optical interface. This is the implementation we will use for the further simulations (connection pattern and redudancies assessment on the polarising structure).

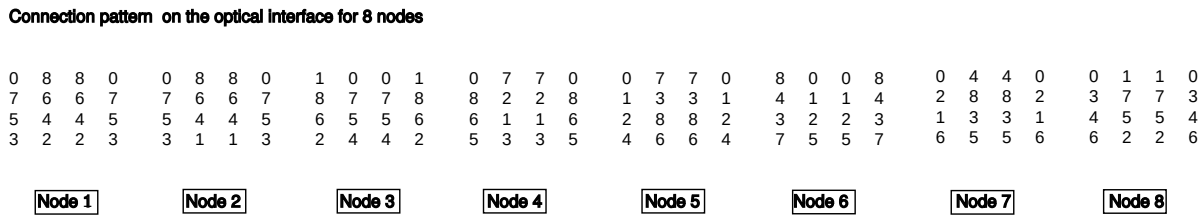


Figure 6.3: Array with the best efficiency for connecting nodes of 4x4 active structures

6.3 Program implementing a CCT on the polarising structure

We developped another program which can be seen in details in appendix, with its electronic version *LCD_connection_pattern_top.m* in order to find the connection pattern on the polarising structure. Of course it depends on the choosen pattern from the previous section and the program *connection_pattern_final_top.m* has to be run before *LCD_connection_pattern_top.m*.

6.3.1 Parameters

The program uses the same parameters as the previous one as well its result.

- `v` : *numbers of active structure on a length of a unit square*
- `nodes`: *numbers of nodes to connect together*
- `Matrix`*: the output result of the program connection_pattern_final_top.m*

6.3.2 Method

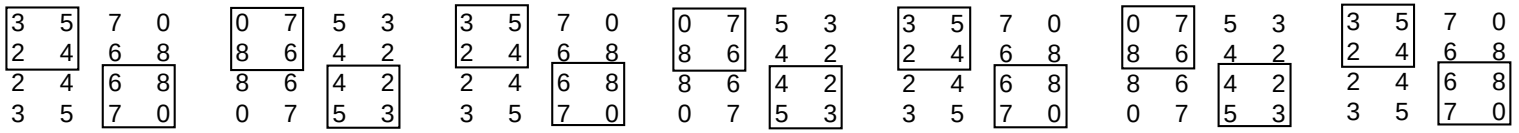
The program starts from the filled output `Matrix` of the previous program.
The idea is to isolate each node of the matrix and its corresponding pattern of connection. Then depending on the placement of the node through the array, we set the first image on the right and on the left polarising pattern and fill a new matrix `LCD_nn` (n=1,...8) by reversing the image on the right (resp. on the left) to fill all the connection pattern on the 7 polarising structures.
We obtains 8 matrix `LCD_nn` corresponding to the image of each emitting node `n` on the 7 polarising structures `LCD_nn`[4]. However, for simplification in programing, the program creates the image on the first polarising structure from the node pattern and reverse it several times to fill the other pattern on the other polarising structure.

6.4 Results: Connection pattern on the polarising structure

After running both programs we obtain the following connection patterns on the polarising structure depending on the emitting node.
Each square means a square of emitter.

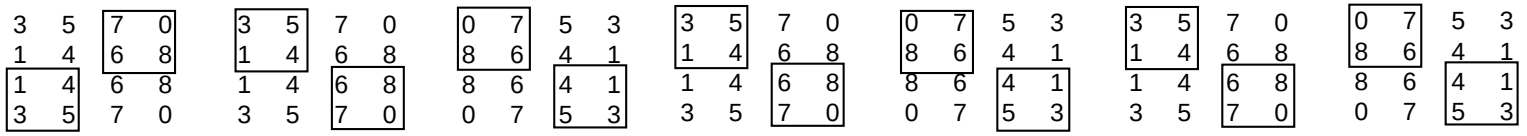
[4]We have to be aware that the program does not distinguish the placement of the image of an emitter or a receiver and so we have to set it afterwards.

Connection pattern on the LCD — Emitting Node: Node 1



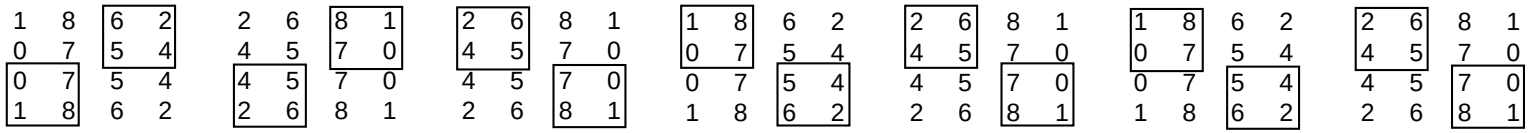
LCD 1 LCD 2 LCD 3 LCD 4 LCD 5 LCD 6 LCD 7

Connection pattern on the LCD — Emitting Node: Node 2



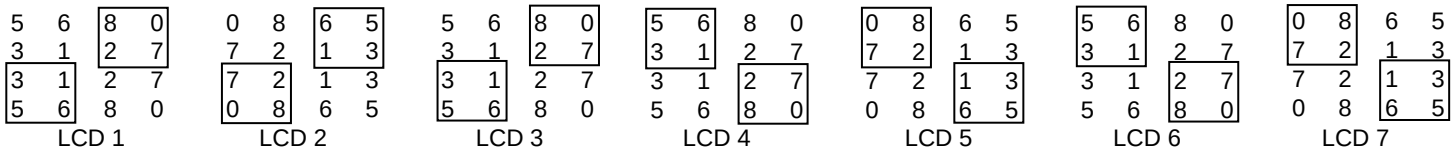
LCD 1 LCD 2 LCD 3 LCD 4 LCD 5 LCD 6 LCD 7

Connection pattern on the LCD — Emitting Node: Node 3

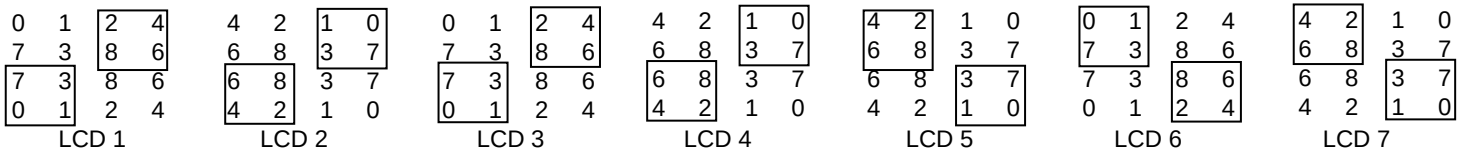


LCD 1 LCD 2 LCD 3 LCD 4 LCD 5 LCD 6 LCD 7

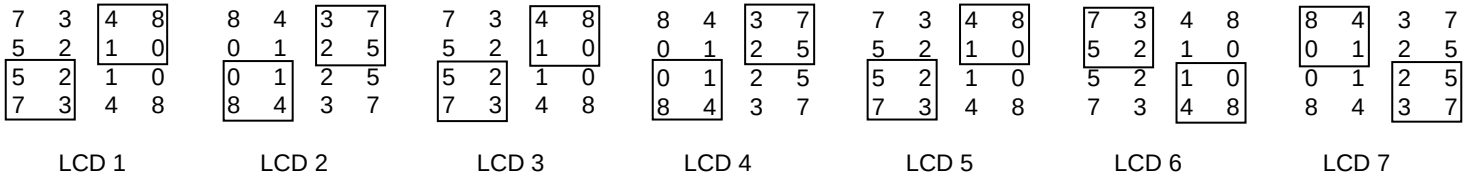
Connection pattern on the LCD — Emitting Node: Node 4



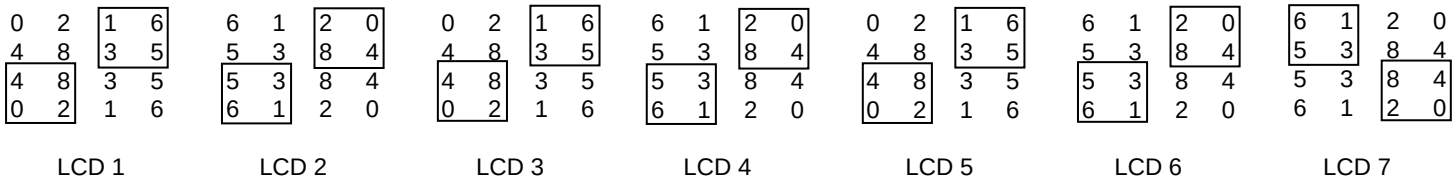
Connection pattern on the LCD — Emitting Node: Node 5



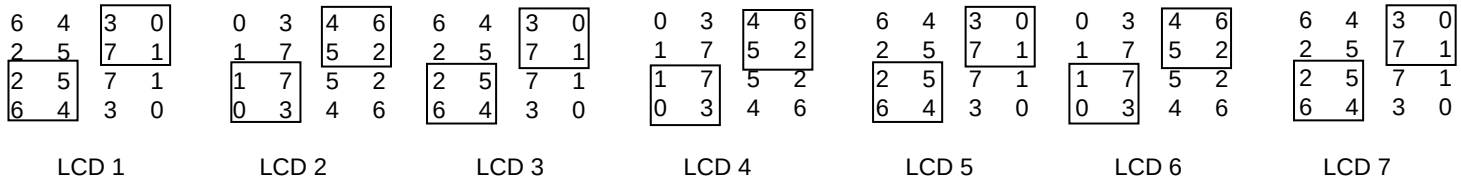
Connection pattern on the LCD — Emitting Node: Node 6



Connection pattern on the LCD — Emitting Node: Node 7



Connection pattern on the LCD — Emitting Node: Node 8



7 Redundancy assessment

At the first place we aimed to have the best efficiency on the optical interface and so to have no redundancies on it.
However, many redundancies can occurs on the polarising structure forming shared channels. These redundancies can reduce the total efficiency of the entire system.
Intuitively we can already say that they occur but we assess them in order to know where exactly they occur for a further possible use of a dynamic polarising structure (LCD).

7.1 Method of Assessment

In order to know where the redundancies occur on the polarising structure, we define a "method" of assessment we will follow here.
The assessment follows a increasing logic from the left to the right, from node 1 to node 8 and uses the results of the second simulation *LCD_connection_pattern_top.m*.

- A polarisation structure is called LCD_n (n=1,2,...7), the node on its left is called n and the node on its right n+1 (see fig. 7.1).
- Each image of the receivers of node n must correspond to a TN structure (linear polarisation state turned) on the LCD set as: "T" state.
- As well, each image of emitters setting the link with the node n+1 of all the nodes on the left (n-1,n-2, ...) must be "turned" and set as: "T" state.
- All the image of emitters and receivers setting the link with node n $>$ n+1 of all the nodes on the left (n-1,n-2, ...) of LCD_n must remain unchanged and set as : "0" state.
- All the other images are set to undefined state (or "T" or "0") : "?". Since this part of the link is free.

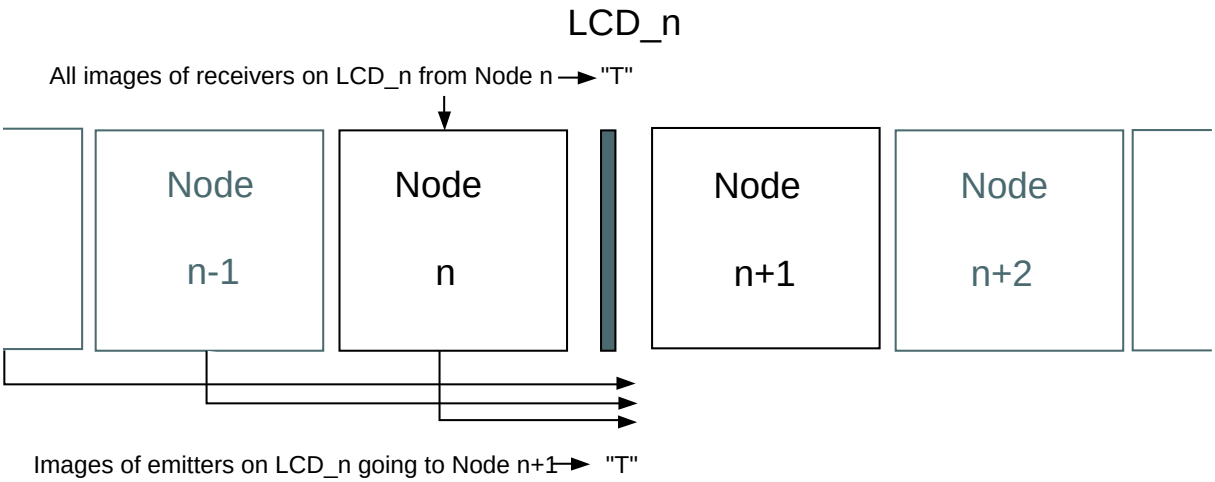


Figure 7.1: Schematic of the method of assessment

The following assessment result shows the different "state" (T, 0, ?) on the image of the corresponding node on the LCD_n.
These images are directly taken from the output results of *LCD_connection_pattern_top.m*.

7.2 Assessment on LCD_1

Figure 7.2: Analysis of the redundancies on LCD_1

7.3 Assessment on LCD_2

Figure 7.3: Analysis of the redundancies on LCD_2

7.4 Assessment on LCD_3

Figure 7.4: Analysis of the redundancies on LCD_3

7.5 Assessment on LCD_4

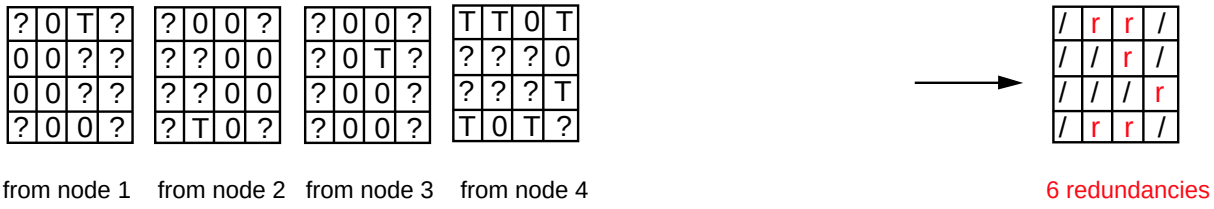


Figure 7.5: Analysis of the redundancies on LCD_4

7.6 Assessment on LCD_5

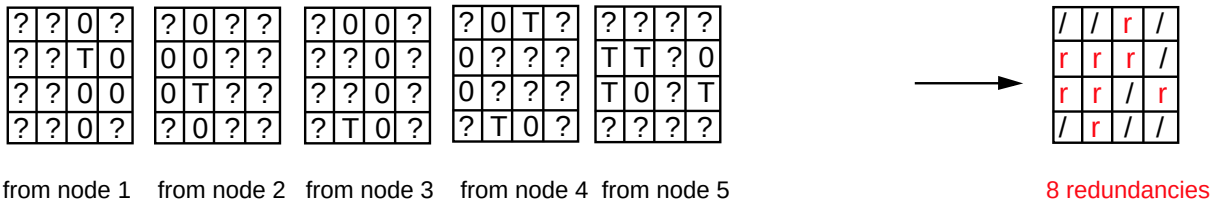


Figure 7.6: Analysis of the redundancies on LCD_5

7.7 Assessment on LCD_6

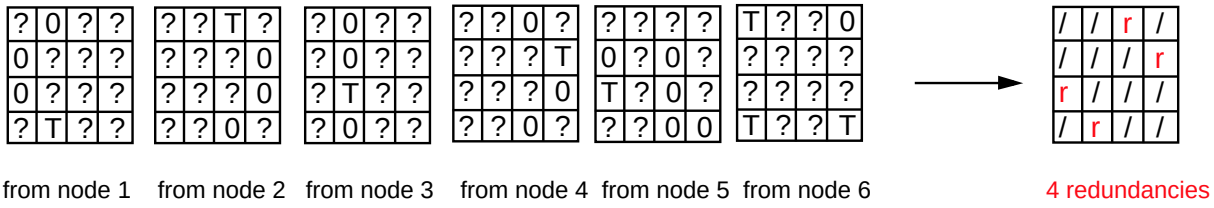


Figure 7.7: Analysis of the redundancies on LCD_6

7.8 Assessment on LCD_7

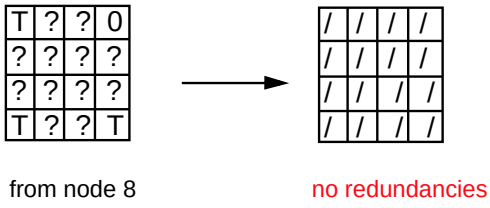


Figure 7.8: Analysis of the redundancies on LCD_7

7.9 Polarising structure static or dynamic ?

The previous assessment doesn't mean we are obliged to use a dynamic polarising structure. It just shows redundancies are occuring along the array and now we are aware of it we have to choose between a good efficiency with broken links but with a static polarising structure or an efficiency of 100% with a dynamic polarising structure.
Indeed by using a pattern HWP we have, regarding the implementation of a CCT:

$$0{+}4{+}8{+}6{+}8{+}4{+}0{=}\ 30 \text{ THEORETICAL BROKEN LINKS}^{1}$$

which for a CCT on the array of 8 nodes it means an efficiency of :

$$E = 1 - \frac{30}{7^7} = 1 - \frac{30}{823.5 \cdot 10^3} = 99.996\% \tag{7.1}$$

When using a dynamic structure offer the possibility to have an efficiency of 100% but on the other hand need a dedicated unit to control it. Further assessment for larger scale combined with a electronic design of a dedicated structure could give us an answer since at this stage of the project we cannot really say which is the best.

7.10 Graphical assessment

The amount of redundancies occuring strongly depends on the position of the polarising structure in the array.
The following figure shows us a symmetry of the assessment of the redundancies.We have no redundancies on the edge of the array and the same amount on the left and on the right side of the polarising structure[2].

This result remains important since at the first place we implemented random pattern of connections ! It might help us in designing a future dedicated unit of control for a dynamic downloadable LCDs patterns.

[1] Broken links means the connections are not set.
[2] But one have to remember they don't occur at the same place.

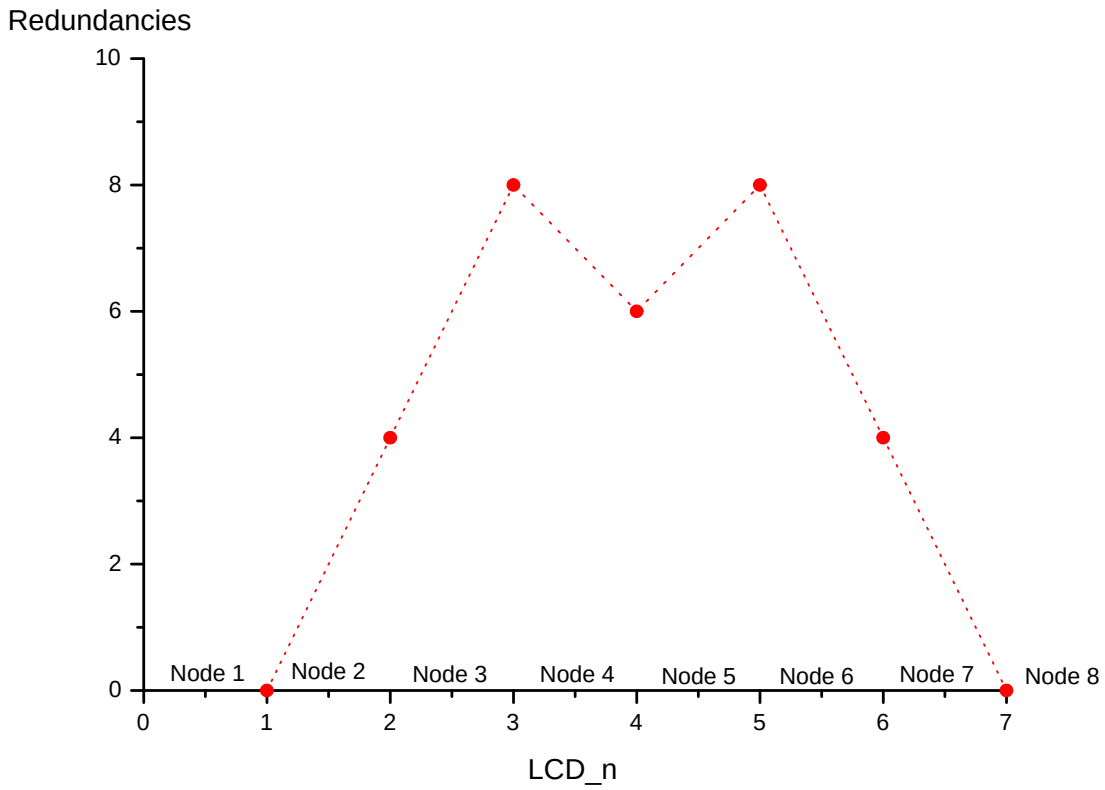


Figure 7.9: Redundancies throughout the array

8 Experimental Assessment

After setting the computationnal model and simulating the implementation of a CCT Topology, the final assessment is of course the experimental one.
If the computational model fits an experimental implementation of a CCT Topology, we can say the design is working and the simulation is correct. This part explains the setup used to prove the model and show the results of the experiment.

8.1 Experimental setup

The optical highway already exists and its design can be seen in details in [5]. The barrel of lenses have a focal length of 4 cm and the system minimise the aberration and power consumption for a wavelength of 790 [nm]. We want to implement a pattern through the first input lens and to probe the pattern at the output with or without switch of polarisation.
This setup is split into 3 systems.

- Laser Source
- Magnification unit
- Optical Highway

8.1.1 Source

In order to check the different image reversing of the computationnal model, we choose to implement a pattern by the use of 3 laser diodes from a package of *Access Pacific (APL 670-10S with driver SK 511)* with the following characteristics:

- Power output : 10 [mW]
- Operating Voltage: 2 [V]
- Wavelength: 670 [nm]

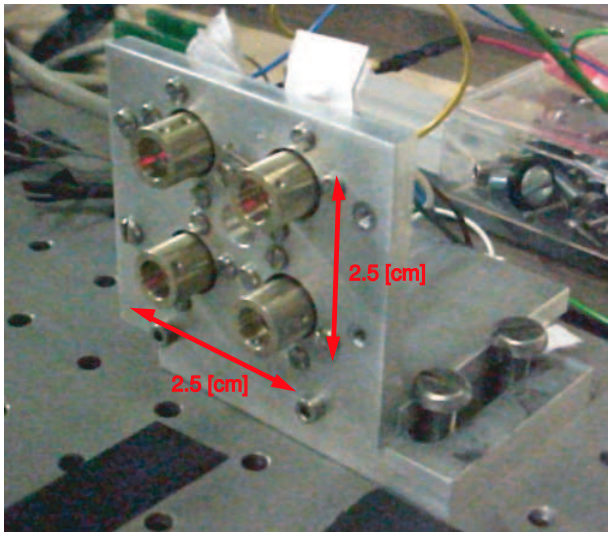


Figure 8.1: Laser diodes forming the original pattern of 3 spots.

Each of them are coupled to a collimator and we have mounted it on a plate of aluminium in a square grid of 2.5 [cm].

That means at the first place we have a pattern of 3 collimated spot seperated by a distance of 2.5 [cm].

8.1.2 Magnification unit

In order to enter the Optical Highway we use two lenses to create the demagnified image of the 3 spots pattern at the focal length of the first input lens of the Optical Highway.

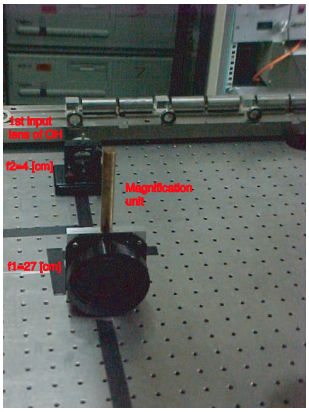


Figure 8.2: Two lenses to enter the optical highway

The two lenses have been choosen regarding the facilities of the laboratory.
They respectively have a focal length of :

$$f_1 = 27[cm] \qquad f_2 = 4[cm] \tag{8.1}$$

Imaging

In order to make a rough approximation before performing all the alignement and adjustments we calculate all the respective object and image distance using the paraxial approximation and thin lens laws[1].

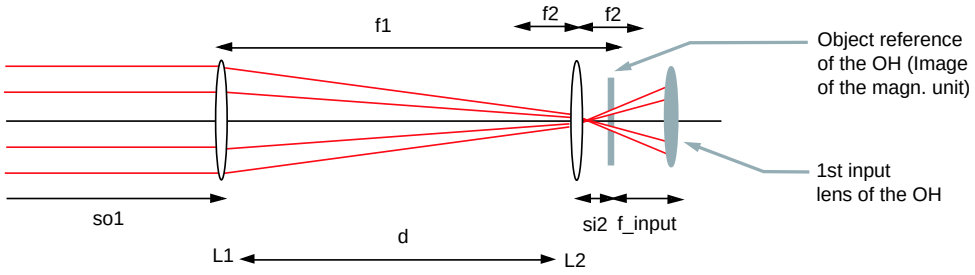


Figure 8.3: Magnification unit.

$$\frac{1}{f_1} = \frac{1}{s_{o1}} + \frac{1}{s_{i1}} \tag{8.2}$$

$$\frac{1}{f_2} = \frac{1}{s_{o2}} + \frac{1}{s_{i2}} \tag{8.3}$$

Regarding the space on the desk we set the distance between both lenses (L1 and L2) at :

$$d = 26[cm] \tag{8.4}$$

Let us take the effect of the first lens (L1) alone.
The image of the 3 collimated beams will be focalised at the focal length of L1, which means:

$$s_{i1} = f_1 = 27[cm] \tag{8.5}$$

Let us take the effect of the second lens (L2), the object distance for this lens will be[2] :

$$s_{o2} = d - s_{i1} = -1[cm] \tag{8.6}$$

[1] Of course, the aim here is not to have a perfect calculation of all the distance for what we should use dioptre laws and radius calculations. However, as we don't know all the characteristics of the lenses, these two simplifications help us to design roughly the magnification unit. All notations corresponds to the notations used in [10]

[2] The negative sign means the image through the first lens is imaged beyond the second one.

Using (8.3), we obtain the final image distance:

$$s_{i2} = \frac{s_{o2} \cdot f_2}{s_{o2} - f_2} = 0.8[cm] \tag{8.7}$$

As the first input lens has a focal lenght of 8 [cm], we have to place the input lense at 8.8 [cm] from L2. The experimental results after adjustements can be seen on fig. 8.5.

Magnification

In order to calculate the magnification of the entire system we assume the object pattern to be directly image through the two lenses. That means we don't take in account 3 single collimated beams but a single object of 3 spots.
The magnification of the entire system (M_T) is the product of the single magnification of each lenses (M_{L1})(M_{L2}):

$$M_{L1} = \frac{s_{i1}}{s_{o1}} \tag{8.8}$$

$$M_{L2} = \frac{s_{i2}}{s_{o2}} \tag{8.9}$$

$$M_T = M_{L1} \cdot M_{L2} \tag{8.10}$$

The object is placed at a distance from L_1 of:

$$s_{o1} = 182[cm]$$

Which means the first magnification[3] will be:

$$M_{L1} = \frac{s_{i1}}{s_{o1}} = \frac{27[cm]}{182[cm]} = 0.148 \tag{8.11}$$

When the second magnification is:

$$M_{L2} = \frac{s_{i2}}{s_{o2}} = \frac{0.8[cm]}{-1[cm]} = -0.8 \tag{8.12}$$

Which gives a total magnification of:

$$M_T = M_{L1} \cdot M_{L2} = 0.148 \cdot (-0.8) = 0.118 \tag{8.13}$$

So, as we start with a object of 3 spots in a square of 2.5 [cm], we should obtain, at the focal length of the first input lens, a pattern of:

$$2.5[cm] \cdot 0.118 = 0.296[cm] \approx 3[mm] \tag{8.14}$$

The experimental result provides a pattern of 3.5 [mm].

[3]We keep here the image distance calculated before.

8.1.3 Optical highway

As said above the optical highway is already set.
All the barrel of lenses can be moved on a single rail when the beamsplitters can be adjusted is all directions.

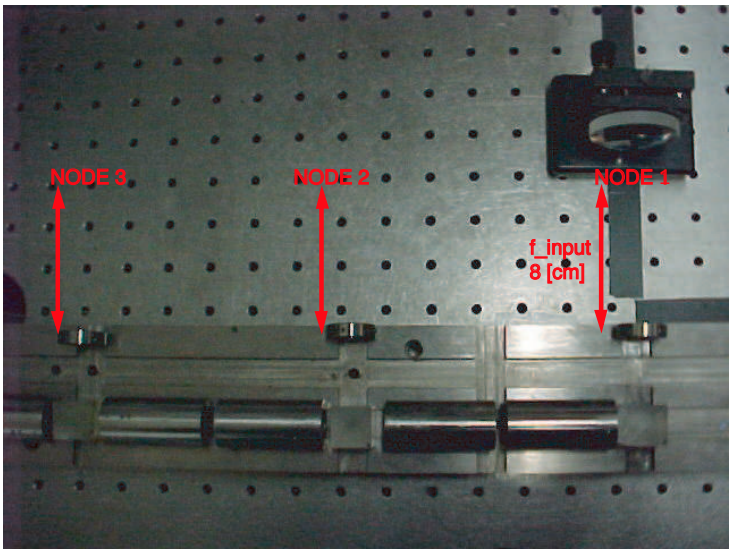


Figure 8.4: The Optical Highway with the last lens L2 of the magnification unit

To switch the polarisation we use 2 plate of LCDs stuck together to make turn the polarisation between each stage.
The quality of both LCD is pretty low and each of them turn a linear polarisation state of 40 degrees after measurement. This is why we stuck them together to obtain a total "switch" of roughly 80 degrees. This is not the ideal "switch" but the aim here is first to demonstrate the principle of the switching through the Optical Highway and this angle seems to be sufficient. However we have to be aware that it is not ideal.

After adjusting and alignement we obtain this final setup:

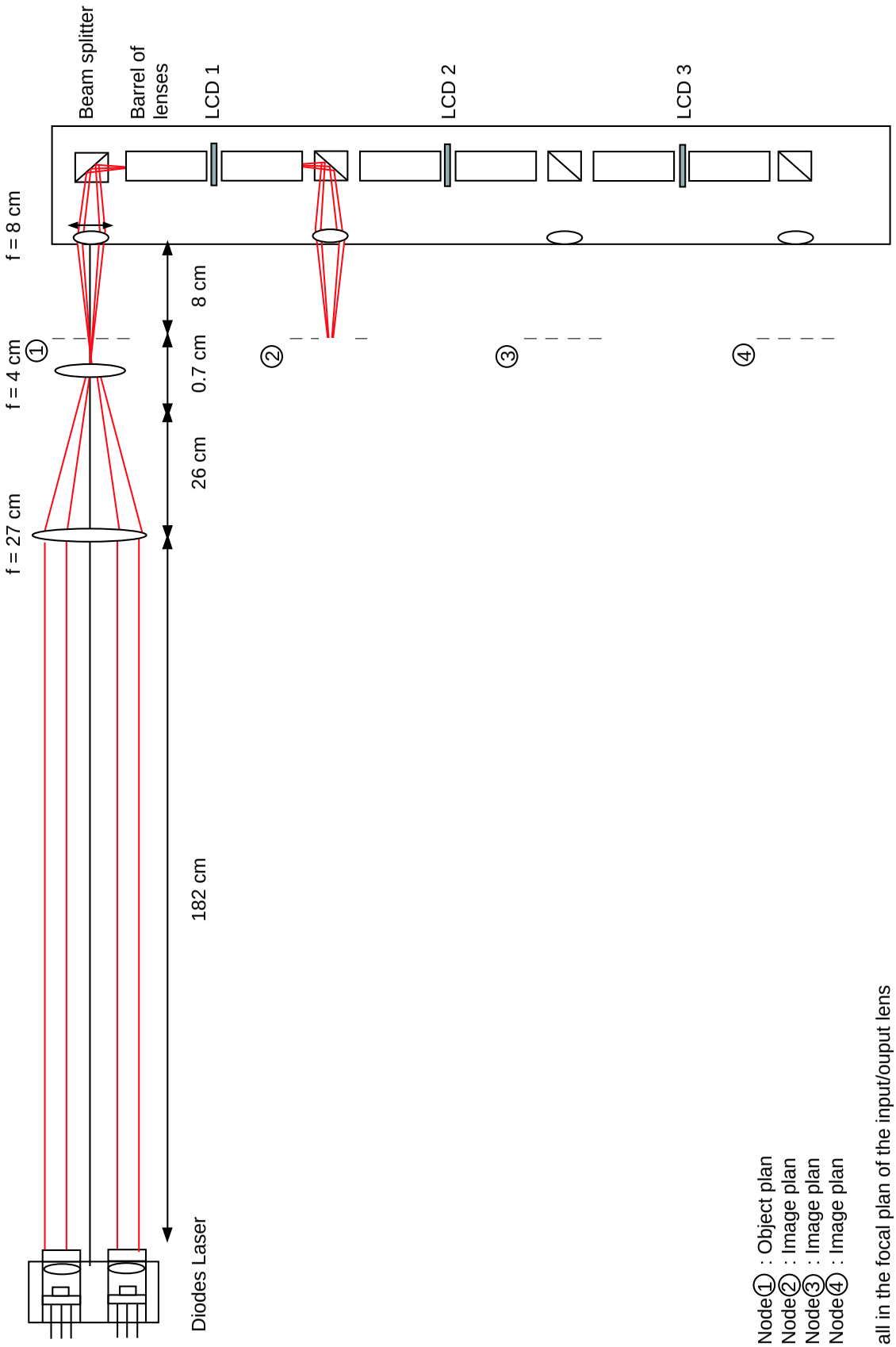


Figure 8.5: Optical highway with pattern input on the right

8.2 Results

8.2.1 Unidirectionnal links without polarisation switch

The first experiment we performed doesn't use any polarising structure.
A input pattern of 3 spots enters the optical highway, and we direct the signal by placing the right polarising beam splitter. That means in order to connect the emitting node 1 to the receiving node 2 we only place the first PBS in the Optical Highway, to connect node 1 to node 3 we only place the second one and so on.
That means the linear polarisation state is only set by the first reflexion on the input PBS and the signal is then directed to the corresponding nodes by using the other PBS as mirrors[4].
The connection performed is an unidirectionnal link with 1 emitting node (Node 1) and 3 receiving ones (Node 2,3,4).

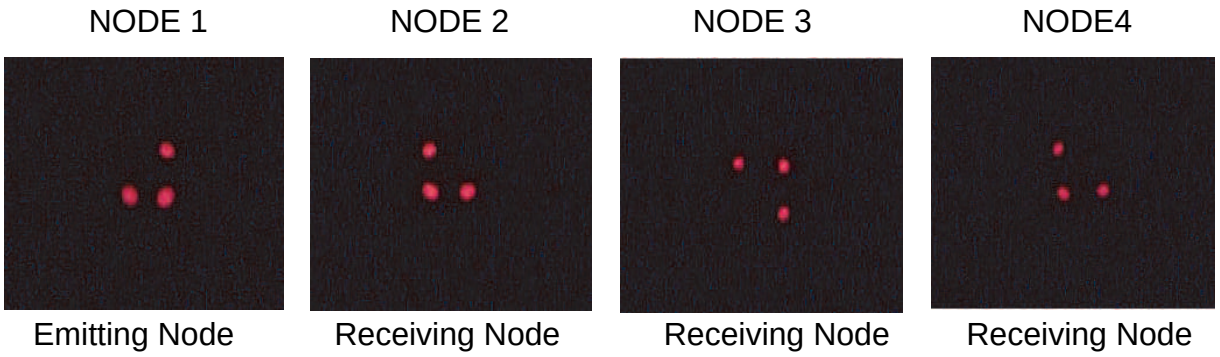


Figure 8.6: Node 1 (Emitting) and its image on Node 2, 3, 4 (Receiving)

The results shows the perfect fit with the computationnal model.
The patterns after odd hops are only flipped by the mirrors when the patterns after even hops are both flipped and central symmetric.

[4]In this first experiment we don't place the first polarisor at the first input lens since the polarisation due the reflexion on the first PBS is enough.

8.2.2 Unidirectional link with polarisation switch

This experiment tries to validate the polarisation control on the image plane between two barrels of lenses on the optical highway. We chose to switch[5] one spot between the second and the third node.
We want to image the bottom left spot of the image pattern of the second node on the third one. We have to be aware of the right spot to switch between the two barrels which create the link since the pattern on this image plans will not be the same. Here are the results:

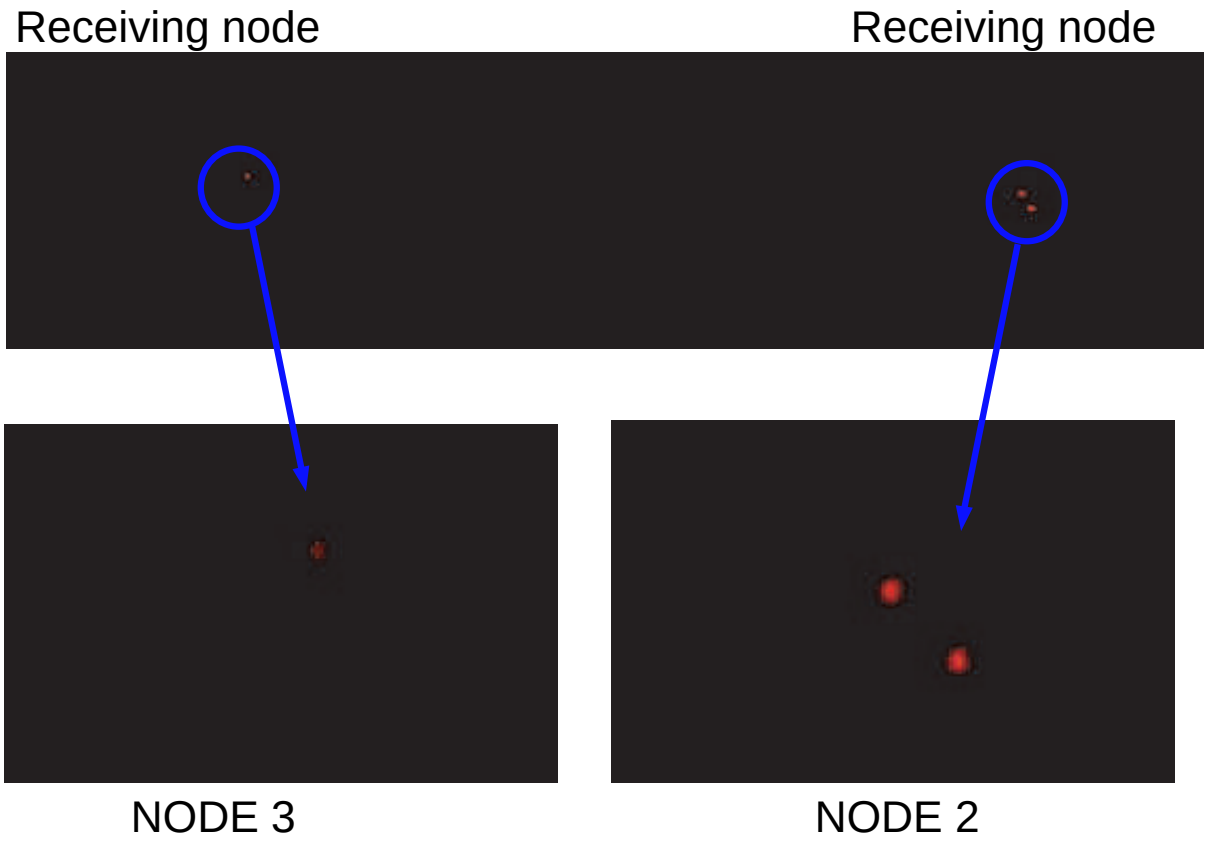


Figure 8.7: Experimental result with swtich of polarisation.

[5]We have not to misunderstand the "switch" of polarisation. Here we use static LCD which, without any voltage applied, "switch" the polarisation of 90 degres when if we electrically "switch" the LCDs, the polarisation state remains the same.

9 Conclusions

The project is split into three main parts in order to set a new design of the optical interface.

The *Analysis part* enabled to set a computational model of how behave the connection signal through the entire system. This is the first achievement of this project and the basis for any design of the optical interface.
Thanks to this model we set a certain design (mapping) of the optical interface we chose to validate by two assessments.

The first one is the *assessment on a simulation of the implementation of a Completely-Connected Topology with the use of the FPGA*. It simulates the setting of the connections on the optical interface for the best efficiency.

The first output result showed us that an efficiency of 100% (without redundancy) on the optical interface is impossible to set.
The best result is obtained for an array of 8 nodes of 4x4 VCSELs/PDs achieving an efficiency of 87.5%.

The second output result showed that unavoidable redundancies occur on the polarising structure. But they are not as many as expected and at this stage of the project either static (HWP) or dynamic (LCD) polarising structures could fit the system. Depending of course on the fact we accept broken links for the HWP or demand 100% of efficiency with the LCD (with a dedicated control unit).

Finally the last and decisive assessment of the design is the *experimental assessment*. Two experiments are performed.
The first one aimed to validate the design by demonstrating the computational model experimentally when the second one demonstrated the validity of the use of patterned polarising structure through the optical highway. The theoretical expectations are fullfilled with the experimental results obtained since what happened experimentally coincide with the computational model.

Analysis, Simulation and Experimental assessments enabled us to demonstrate the advantages of the implementation of a reconfigurable structure in a free-space interconnection system since it provides a design of the optical interface with high efficiency and low redundancy.

10 Future Work

The future work should focuss on the *viability* of the design and in order to assess it several tasks should be achieved:

- Expanding of the array in 2-D array (by combinaison of 1-D arrays) to try to improve the efficiency
- Implementation of other patterns of connections for a Completely Connected Topology
- Assessment of the implementation of other Topologies as the Hypercube Topology
- Characterisation of the experimental setup (power loss, aberrations, ...)
- Fabrication and test of patterned HWPs in the OH
- Fabrication and test of patterned LCDs in the OH with switch possibilities.
- Survey of an electronic design of a control unit structure for downloadable connection patterns on the LCDs

Aknowledgements

I want to thank at the first place Prof. R.-P. Salathé for having encouraged and supported me in an experience abroad and Dr. Snowdon for having enthousiastically accepted my application as an exchange student at Heriot-Watt University.
I also want to thank all the members of the Optically Interconnected Computing Systems Group (OIC) at Heriot-Watt University and in particular Keith Symington for his helpful and very useful advices concerning the project and life in Scotland.

11 Glossary & Definitions

AMOS:	Analysis & Modelling of Optoelectronic Systems, background project,[2]
Bandwidth:	Amount of data which can be moved per unit of time [Bytes/s]
CCT of FCT:	Completely Connected Topology in which each nodes is connected to each other ones
FPGA:	Field Programable Gate Array, the reconfiguration logic structure, 2.2
Hop:	Link between to neighbour nodes.
HT:	Hypercube Topology in which each nodes is connected to 2^n other ones (n=dimension)
HWP:	Half-Wave Plate, polarising structure to switch a linear polarisation state of 90deg, 2.1 & 2.2.3
Latency:	Time-of-deliver between two nodes [s]
Link:	Way between two nodes.
Node:	Communication Unit at a crossroad, here a specific Optoelectronic Smart-Pixel.
OIC :	Optically Interconnected Computing Systems Group
OH :	Optical Highway, the "hard wired" optical system of interconnections, 2.1
PBS:	Polarising Beam Splitter
PD:	Photodiodes, the receivers, 2.2.1
Topology:	How nodes are connected
TN:	Twisted Nematic Structure
VCSEL:	Vertical Cavity Emitting Lasers, the lasers emitters, 2.2.1

12 Appendix

12.1 Program giving the connection pattern on the optical interface

p. 58-64

12.2 Program giving the connection pattern on the polarising structure

p. 65-67

```
clear all;

%-------------------------------------------------------------------------------------------------------------------------
% PROGRAMME QUI PLACENT LES CONNECTIONS POUR UNE CC TOPOLOGY EN FONCTION DU
% NOMBRE DE NODES ET DU NOMBRE D'ELEMENTS ACTIFS
%-------------------------------------------------------------------------------------------------------------------------


%-------------------------------------------------------------------------------------------------------------------------
% INTERFACE USER FRIENDLY
%-------------------------------------------------------------------------------------------------------------------------


choice = menu('Choose the Connection Pattern for :', '3 Nodes','4 Nodes','5 Nodes','6 Nodes','7 Nodes',
          '8 Nodes','9 Nodes' );

if choice == 1
   nodes=3;                  % Nombres de nodes
   stop=0;                   % Variable de sortie de la grande boucle (while)
end

if choice == 2
   nodes=4;
   stop=0;
end

if choice == 3
   nodes=5;
   stop=0;
end

if choice == 4
   nodes=6;
   stop=0;
end

if choice == 5
   nodes=7;
   stop=0;
end

if choice == 6
   nodes=8;
   stop=0;
end

if choice == 7
   nodes=9;
   stop=1;
end
```

```
% DEFINITIONS DES VARIABLES
%-----------------------------------------

v=2;                        % nombre de VCSEL dans carre unitaire1
c=2*(v^2)+1;                % nombre de nodes connections theoriquement possible a d'autres nodes
n_connections=2*v^2;

%nodes=9;                   % Nombre de Nodes

Matrix = zeros(2*v,2*v*nodes)% Definition de la grande matrice representant toute la rangee de nodes (0)

source=1;                   % Numero du node emetteur
numero=source+1;            % Numero des nodes receveurs

max=(nodes-1)/2;            % Definition du nombre d'iteration pour le placement des connections receptrices
                            % depend du node emetteur
total=((2*v)^2)*nodes;      % Nombre total d'elements actifs



vect_perm=[source+1:1:nodes];   % Definition des connections a faire
vect_remp=[zeros(1,(c-nodes))]; % Remplissage du vecteur de connection avec des 0
                                % (si nodes<9)

I=perms([vect_perm,vect_remp]); % Vecteur de toutes les permutation possibles
f=1                             % Numero de la permutation parmi celles du vecteur I
f_prev=1;
p=1                             % Numero de l'element du vecteur i (dans une permutation)

h=0;                            % Nombre de hop


check=0;                        % Variables de controle nombres de zeros
%stop=0;                        % Variables de sortie de la grande boucle


c_iteration=0;                  % Nombre d'iteration
auxiliaire=0;
fin=2*nodes*(n_connections-(nodes-1)); % Fin du programme, lorsque le nombre de connections libres est
                                      %  coherent

it=0;
st=1;
i_prev=[,0,0,0,0,0,0,0,0];

%-----------------------------------------------------------------------------------------------------------------------------------
% DEBUT DU PROGRAMME
%-----------------------------------------------------------------------------------------------------------------------------------

% Ajustement du parametre max en fonction de la parite du nombre de nodes

if ((nodes/2)-round(nodes/2))==0
        max=(nodes/2)-1;
    else
        max=(nodes-1)/2;
end
```

```
% Debut de la grande boucle
% ---------------------------------

while (stop ~= fin)

% ----------------------------------------
% Remplissage du node emetteur
%- ----------------------------------------

vect_perm=[source+1:1:nodes];                % Definition des connections a faire
vect_remp=[zeros(1,(c-nodes))];               % Remplissage du vecteur de connection avec des 0

I=perms([vect_perm,vect_remp]);

i=I(f,:);                                    % Choix d'une permutation

if source==1
   i_prev=i;
end

   for m=1:2*v                               % Prend la matrice du node correspond
      for n=1:v                              % dans la grande matrice
         A(m,n)=Matrix(m,n+(source-1)*2*v);
      end
   end
A

   for m=1:2*v                               % On remplit la matrice avec le vecteur de connection
         for n=1:v
            if A(m,n)==0
               A(m,n)=i(p);
               p=p+1;
            end
         end
      end
A
B= fliplr(A)
N=[A,B]                                      % Motif du node emetteur


   for m = 1:2*v                             % On place le Node plein dans la grand Matrice
         for n = 1:2*v
               if  Matrix(m,n+(source-1)*2*v) == 0
                  Matrix(m,n+(source-1)*2*v)=N(m,n);
               end
         end
   end

Matrix                                       % CHECK !!!

p=1;                                          % Reinitialisation des parametres
numero=source+1;                             % pour la prochaine iteration
```

```
% ------------------------------------------------------------------------------------------
% Remplit les hop pairs et impairs d'un node emetteur sur les nodes suivants
% ------------------------------------------------------------------------------------------


for g=0:max                                % Boucle principale


% Numero des nodes pairs suivantS (EVEN)
%----------------------------------------

if g ~= 0

   if numero<nodes

      h=2*g
      numero=source+h

   Even_hop = flipud(N);                   % Pour hop pair image par symmetrie axiale

   for m=1:2*v                             % Remplace le numero de connection sur le premier node par
         for n=1:2*v                       % le numero du premier node
          if Even_hop(m,n)== numero
             Even_hop(m,n)=source;
          else
             Even_hop(m,n)=0;
          end
       end
   end


   for m=1:2*v
         for n=1:2*v
            if  Matrix(m,n+(numero-1)*2*v) == 0          % Scan la grande matrice et place le motif
               Matrix(m,n+(numero-1)*2*v)=Even_hop(m,n); % de connection lorsque la place est libre
            end
         end
   end

   for m=1:2*v                                           % Boucle de controle, scan la grande matrice
         for n=1:2*v                                     % pour les nodes connectes et compte le nombre
          if Matrix(m,n+(numero-1)*2*v) == source        % de connections (2 par nodes) du node emetteur
               check=check+1;                            % CHECK = (NODES-SOURCE)*2
            end
         end
   end
end
end

Matrix
```

```
% Numero des nodes impairs suivants (odd)
%-----------------------------------------------------------

   if (numero<nodes) | (source==(nodes-1))

   h=2*g+1
   numero=source+h


   Odd_hop = fliplr(N);                  % Pour hop impair image par symmetrie centrale

   for m=1:2*v                           % Remplace le numero de connection sur le premier node par
      for n=1:2*v                        % le numero du premier node
         if  Odd_hop(m,n)== numero
            Odd_hop(m,n)= source;
         else
         Odd_hop(m,n)=0;
         end
      end
   end


   for m=1:2*v
      for n=1:2*v
         if  Matrix(m,n+(numero-1)*2*v) == 0          % Scan la grande matrice et place le motif
            Matrix(m,n+(numero-1)*2*v)=Odd_hop(m,n);  % de connection lorsque la place est libre
         end
      end
   end

   for m=1:2*v                                  % Boucle de controle, scan la grande matrice
         for n=1:2*v                            % pour les nodes connectes et compte le nombre
          if Matrix(m,n+(numero-1)*2*v) == source % de connections (2 par nodes) du node emetteur
               check=check+1;                   % CHECK = (NODES-SOURCE)*2
            end
         end
   end

end

if numero == nodes
   break
end

end
numero=0;
h=0;

Matrix                                          % CHECK !!!
check
```

```
%-------------------------------------------------------------------------------
% Construit le motif du node suivant et l'inclus dans la grande matrice
% -------------------------------------------------------------------------------

if choice==7

stop=1;
else
   stop=0;
end

   for m=1:2*v                          % Scanne la matrice et compte le nombre de zeros
      for n=1:nodes*2*v
         if Matrix(m,n)==0
            stop=stop+1;
         end
      end
   end


if (check~=(nodes-source)*2)            % Si les connections ne sont pas placees sur tous les
                                        % Nodes, erreur et on essaie une autre permutation
            Erreur='ZUT'
            f=f+1
            p=1;

      if f > length(I)

      source=source-1;
      f=f_prev;
      f=f+1;

      end

   for m=1:2*v                          % On reinitialise le node en supprimant
      for n=1:2*v                       % les connections du node "tare" et ses suivants
         if Matrix(m,n+(source-1)*2*v)>source-1
            Matrix(m,n+(source-1)*2*v)=0;
         end
      end
   end

   for m=1:2*v                          % On reinitialise les nodes connectes
         for n=1:(length(Matrix)-2*v*source)
               if  Matrix(m,n+source*2*v)>=source
                  Matrix(m,n+source*2*v)=0;
               end
         end
   end
   check=0;                             % On reinitialise la variable de controle
   h=0;
```

```
else                                % Sinon on passe au nodes emettant suivants
   source=source+1;
   f=f_prev;                        %sauvegarde la position precedente
   f=1;
   check=0;
   h=0;
end

c_iteration=c_iteration+1 ;         % On compte le nombre d'iteration
source

end                                 % end de l'iteration

Matrix
save('Matrix');
c_iteration
stop
fin
```

```
%-------------------------------------------------------------------------------------------------------------------------
% PROGRAMME QUI DETERMINE L'EMPLACEMENT DES CONNECTIONS SUR LA STRUCTURE
% POLARISANTE (LCD) POUR UNE CC TOPOLOGY EN FONCTION DU NOMBRE DE NODES ET DU
% NOMBRE D'ELEMENTS ACTIFS
%
% NOTE: L'element principal "Matrix" est fourni par le programme de connection precedent
%-------------------------------------------------------------------------------------------------------------------------

clear all

load('Matrix');
source=1;
receiver=1;
max=3
nodes=8;
n_lcd=nodes-1;
v=2;

LCD=[zeros(2*v,2*v*(nodes-1))]

%-------------------------------------------------------------------------------------------------------------------------
% DEBUT DU PROGRAMME
%-------------------------------------------------------------------------------------------------------------------------

for source=1:nodes


for m=1:2*v                          % On prend le nodes correspondant dans
      for n=1:2*v                    % dans la grande matrice de connection.
         N(m,n)=Matrix(m,n+(source-1)*2*v);
      end
   end

N
K=rot90(N,-1)                        % On construit le motif de connection sur le LCD a
                                     % droite du node source (et a gauche)


% Remplissage des LCDs suivants le motif choisi
% ---------------------------------------------------------------

   if source < 8

      for j=0:nodes-(source+1)       % On construit les motif de connection sur les
                                     % LCD suivants par symetrie du premier LCD

         for m=1:2*v
            for n=1:2*v
               LCD(m,n+(source-1)*2*v)=K(m,n);
               L(m,n)=LCD(m,n+(source+j-1)*2*v);
            end
         end
   M=fliplr(L);                      % On fait une symmetrie de L
                                     % idem rotation de -90 du N
```

```
        if source+j<n_lcd

            for m=1:2*v                                  % On place le LCD dans la grande
                for n=1:2*v                              % matrice de connection LCD
                    LCD(m,n+(source+j)*2*v)=M(m,n);
                end
            end
        end

        end

    elseif source == 8

        for j=0:-1:-5

            for m=1:2*v
                for n=1:2*v
                    LCD(m,n+(source-2)*2*v)=K(m,n);      % On prend le LCD precedent
                    L(m,n)=LCD(m,n+(source+j-2)*2*v);    % On le place dans la matrice L
                end
            end
        M=fliplr(L);

            for m=1:2*v                                  % On place le LCD dans la grande
                for n=1:2*v                              % matrice de connection LCD
                    LCD(m,n+(source+j-3)*2*v)=M(m,n);
                end
            end
        end
    end

% Remplissage des LCDs precedents le motif choisi
% ------------------------------------------------------------------

if source==2

    for g=0:source-2

    for m=1:2*v
            for n=1:2*v
                LCD(m,n+(source-2)*2*v)=LCD(m,n+(source-1)*2*v);    % On prend le LCD precedent
                L(m,n)=LCD(m,n+(source-g-2)*2*v);                  % On le place dans la matrice L

            end
    end
    end

elseif  source<8

    for g=0:source-3

    for m=1:2*v
            for n=1:2*v
                LCD(m,n+(source-2)*2*v)=LCD(m,n+(source-1)*2*v);    % On prend le LCD precedent
                L(m,n)=LCD(m,n+(source-g-2)*2*v);                  % On le place dans la matrice L
            end
    end
```

```
      L;
M=fliplr(L);                                    % On fait une symmetrie de L

   for m=1:2*v                                  % On place le LCD dans la grande
         for n=1:2*v                            % matrice de connection LCD
            LCD(m,n+(source-2-(g+1))*2*v)=M(m,n);
         end
   end

   end
end

source
L;
M=fliplr(L);
LCD;

magic_str = ['LC_',int2str(source),'=LCD'];
eval(magic_str)

LCD=[zeros(2*v,2*v*(nodes-1))];

end
```

Bibliography

[1] D. Miller, *Rationale and Challenges for Optical Interconnects to Electronic Chips.* Proceedings of the IEEE, Vol. 88, No. 6, June 2000.

[2] K. Symington, G. Russell, and J. Snowdon, *Analysis and Modelling of Optoelectronic Systems - AMOS.* Technical Report, Internal, HW, 2002.

[3] K. Symington, *Optically Interconnected Dynamically Reconfigurable Computing.* Technical Report, Internal, HW, 1999.

[4] K. Symington, *Optically Interconnected Computing Systems.* PhD Thesis, HW, 2001.

[5] C. Barrett and A. Walker, *Reconfigurable Free-Space Interconnect Systems.* PhD Thesis, HW, 1995.

[6] V. Kumar, A. Grama, and A. Gupta, *Introduction to Parallel Computing, Design and Analysis of Algorithms.* The Benjamin/Cummings Publishing Compagny, 1994.

[7] D. Sima, T. Fountain, and P. Kacsuk, *Advanced Computer Architecture, A design Approach.* Addison-Wesley, 1997.

[8] S. J. Russell, G., *Two physical Topologies for an Optical Highway Interconnect.* Technical Report, Internal, HW, 2002.

[9] J. Casswell and A. Walker, *Optical Interconnects for Optoelectronic Systems.* PhD Thesis, HW, 2003.

[10] E. Hecht, *Optics, 3rd.* Addison-Wesley, 1998.

[11] C. Moir, *Interconnections.* Technical Report, Internal, HW, 2002.

[12] C. Gocke, *Optical Implementations of Multinode Networks.* Technical Report, Internal, HW, 1998.

[13] S. J. Layet, B., *Comparison of Two Approaches for Implementing Free-Space Optical Interconnected Networks.* Optics Communications, 189, 1-3, p.39-46, March 2001.