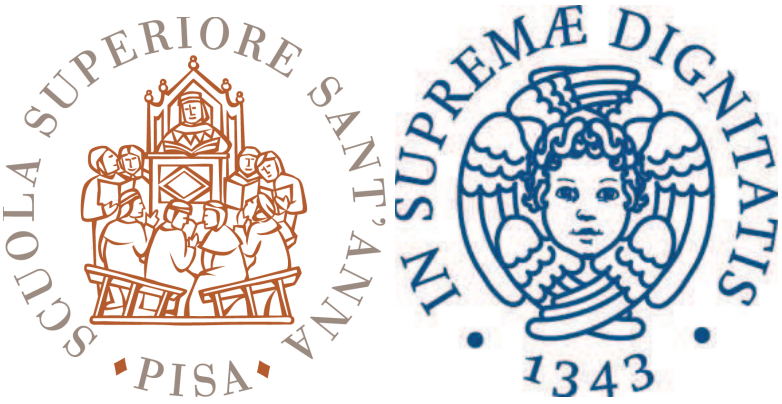

Sant'Anna Superior School of Pisa
University of Pisa

International Master Degree in
Computer Science and Networking

On-Chip Optical Interconnection Networks for Multi/Manycore Architectures

Author:
Gianmarco Saba

Supervisors:
Marco Vanneschi
Piero Castoldi

October 2, 2012

Alla mia famiglia,
per il loro amore e imperturbabile sostegno
nel'aiutarmi a completare questo importante percorso.

Abstract

The rapid development of multi/manycore technologies offers the opportunity for highly parallel architectures implemented on a single chip. While the first, low parallelism multicore products have been based on simple interconnection structures (single bus, very simple crossbar), the emerging highly parallel architectures will require complex, limited degree interconnection networks. This thesis studies this trend according to the general theory of interconnection structures for parallel machines, and investigates some solutions in terms of performance, cost, fault-tolerance, and run-time support to shared memory and/or message passing programming mechanisms.

Contents

Chapter 1

Introduction

1.1 Summary

In this thesis we investigate the current state of the art of the research in the field of on-chip optical interconnection networks targeting multi and many-core architectures and we propose some new solutions discussing advantages and disadvantages of each proposal. The added contribute of this thesis consists first in presenting in an organized way the various research currently going on in the field of optical on-chip interconnects: from the CMOS integration strategies to the design of various photonic integrated components. The thesis then considers some of the parallelization and high level architectural issues inherent to the high performance computing largely studied in the computer science literature. Such background built in the first part of the thesis is then exploited later to analyze the various interconnection networks already proposed in the literature and to propose new solutions and analyze advantages and disadvantages of them. This work does not claim to be exhaustive or final: its purpose is to present in an organic and consistent manner all the techniques and technologies currently involved in this wide area of studies which are on-chip optical interconnection networks targeting manycore architectures. We attempt to do so adopting a new methodology. While in the past the design of on-chip interconnects was subject to a pure integrated circuit engineering approach (bottom-up), and a design strategy starting from the process level is realistically unfeasible, we now try to consider the driving requirements coming from the world of the design and implementation of high performance parallel applications with the physical and engineering constraints implementing a meet-in-the-middle approach. A lot of theory has been developed and has still to be developed in the field of design of parallel applications but some of the most remarkable process level paradigms have already made their way in the commercial and scientific environments. The quality of the mapping of process level computational patterns on real parallel hardware architectures is of key im-

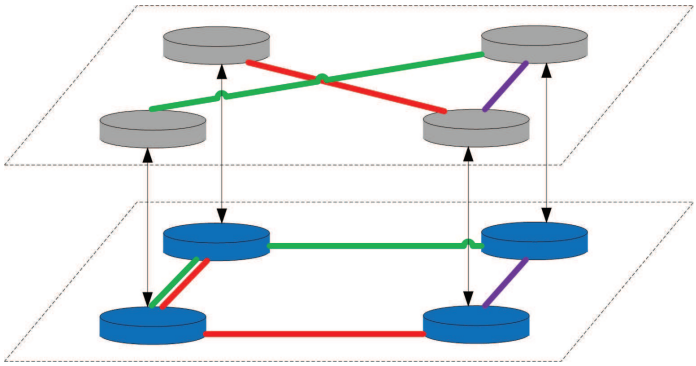

Figure 1.1: *An Example of the Optical Network Design Problem.*

portance in order to achieve the highest performance possible. The goal of the on-chip optical interconnection networks in this mapping is to provide an improved quality of service to the communication patterns implicitely defined by process-level parallel modules. The problem is very similar to the optical network design problem (Figure 1.1) already extensively studied in the field of optical networks where a logical topology defined at an upper layer needs to be mapped to the physical topology at a lower level respecting constraints derived from the required QoS (latency, bandwidth) and resource utilization.

1.1.1 Structure of the Thesis

The thesis is divided in four parts. The first two constitute the basis on which the last two are founded.

In the first part, *Design of On-Chip Optical Networks* we explore the technologies and solutions proposed in the most modern scientific literature regarding integrated photonic components. These are the building blocks of every integrated optical interconnection network and therefore need to be studied.

In the second part, *Architecture Paradigms and Cost Models*, we discuss the most common parallel architectural paradigms with their associated cost models and some of the most common process level parallel paradigms widely exploited nowadays.

In the third part of the thesis, *Indirect Networks*, building on the concepts developed in the two previous parts, we study several cases of on-chip optical indirect networks.

Finally, in the fourth part of the thesis, *Direct Networks* building again on the concepts developed in the first two parts, we study several cases of direct networks.

In the third and fourth parts, we analyze the different aspects of an interconnection network: from the global clock distribution, to the data com-

munication and then the structure of the switches. Every one of this aspects can contain new proposals marked by dedicated sections called *"Proposal"* or presentation of solutions from the literature in sections called *"Study Cases"*.

In the remainder of this chapter we first provide an historical perspective (Section 1.2) of the fields of high performance computing architectures and optical networks which are the ancestors of the new field of integrated photonic networks. In section 1.3 we list the evaluation parameters which will lead our analysis. In Section 1.4.1 we analyse the drawbacks of the electrical on-chip interconnects and in Section 1.4.2 the proposed electrical solutions to these issues. In Section 1.4.3 we introduce the advantages of using optical interconnects with respect to the electrical interconnects. In Section 1.4.4 we discuss the current limits of the integrated photonic technologies. Finally, in Section 1.4.5 we present the results of the most recent comparative analyses between electrical and optical on-chip interconnects proposed in the literature.

1.2 An Historical Perspective

The field of *High Performance Computing* (HPC) is earning central importance in science and business research. Increasingly complex simulation scenarios need to be studied with enhanced precision and accuracy. Examples of today common scientific applications include simulations for *weather forecast* where we have to deal with huge data sets typically composed by signals originated by a grid of probes geographically distributed that must be analyzed within strict deadlines (it is useless to get today the weather forecast of yesterday!). The *analysis of ocean currents* is another example of computation that can gain many advantages from a parallel implementation. Ocean currents are one among the factors that influence the way marine mammals and other marine animals schedule their migrations during cold seasons; superficial ocean currents also influence the routes used by intercontinental cargo ships and, if properly exploited, can result in faster trips and then less petrol consumption. In extreme cases, a deep knowledge of local currents can make the difference in ship racing competitions: maybe within few years each ship of the America's Cup will be connected to a proprietary server farm in order to compute the fastest routes! *Fluid dynamics*, a phisics discipline that studies the flow of fluids (gases and liquids) and which includes aerodynamics and hydrodynamics, is one of the most promising fields that can improve the efficiency of land and water vehicles. The computational structure of this problem is similar to that of weather forecast and ocean currents analysis. In the recent times, the field of computer vision earned a lot of attention from the industry world. With the term *computer vision*, we attempt to group all the analysis, modifying and high-

level understanding of images or video frames [1]. This kind of computations can be described either by real time and offline alghoritms. As an example, these are the same algorithms which can be found in modern cameras to automaticaly focus on the smile of a person (face detection) or stabilize the image and adjust the brightness; or in surveillance systems (face recognition), biometrics (fingerprint and retina autentication), medicine, safety systems (car pedestrian detection systems or road sign detection). *Drugs discovery* is another example of scientific computation that can be higly parallelized.

After all these examples, it is clear that parallel computing platforms are nowadays establishing the boundaries of which scientific challenges can be solved and they are the leading tools for scientific discoveries. In the picture below we can observe chemicals (gray spheres) that can dock onto a designated target in the body, such as a protein (red ribbons).

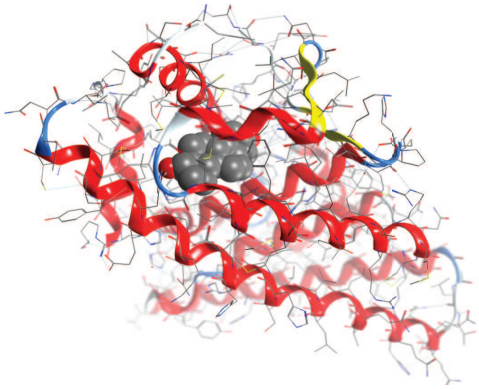

Figure 1.2: *From "Supercomputing research opens doors for drug discovery", Oak Ridge National Laboratory - Print Press Release.*

In order to keep the pace of the computational requirements, processor producers started scaling the size of transistors (and correspondly scaling quadratically the transistor density) of about 35% per year [14] and increasing the clock frequency so that after some time they reached the limit imposed by factors among which there are cooling and signal interference. At that point they realized that this strategy, with the goal of keeping the Moore's law rate, was infeasible for the long term evolution of the next generation architectures; so they started integrating many processing cores inside the same chip operating at lower clock voltages and clock rates [14] in order to optimize performance per Watt (FLOPS/Watt) and alleviate the clock frequency issue: above certain operating frequencies, the electromagnetic interference between closely placed electrical channels detrimentally affects the overall performance of such systems. This new approach named *"More than Moore's Law"* focuses on the integration of the old-fashion discrete

components inside the same chip. The approach is the same that allowed in few years to integrate within the same PDA device basic components such as a processing unit, a photocamera, various sensors and so on. The integration of many processing nodes (cores) inside the same chip is the most suitable and most accepted solution to increase the performance of next generation multiprocessors. Whatever the speed of a single processing node, a group of them suitably integrated and used with the proper criteria will always be faster and more performant. As mentioned before, this solution also allows to reduce the relative power consumption which is not a negligible issue: it is enough to consider that data intensive applications running for Internet search engines and social networks require more than 100MW per single datacenter in order to be executed. The multiplicity of processing nodes, i.e. cores at the on-chip level recalls the paradign of distributed systems on chip. Some chip producers are nowadays speaking of *On-Chip Cloud Computing*. The innovations of the high performance computing are also pushing the development of mobile devices with a relevant potential computational bandwidth. In order to be feasible, some of these systems must be integrable on-chip and with a very low power consumption such that not to affect the battery lifetime of these devices. If this objectives will be achieved, we will have do deal with massively distributed grids able to completely change the constraints in the approach to parallel and distributed computing.

If this could be considered an HPC hystorical perspective summary, ten years before parallel computing and parallel architectures were envisioned, the world of telecommunications experienced a breakthrough with the advent of optical fibers and optical transmission systems. Coaxial cables used at that time for long haul communications started being replaced with optical fibers in front of a lower attenuation coefficient (2dB/Km initially). Continuous wave lasers operating at room temperature, multimode fibers and suitable photodetectors operating in the first and second windows were the costitutive elements of this revolution. Optical transmission systems offered impressive advantages with respect to the previous generation of electrical systems. Electromagnetic interference was no more an issue due to the nature of light; longer unamplified spans could be reached without the need of intermediate regeneration. The huge spectral bandwidth of optical fiber media suggested later the development of WDM (*wavelength division multiplexing*) systems in which several channels (up to 160 in current Dense-WDM systems) could travel at the same time in a single fiber cable. Superior robustness with 25 years of expected lifetime of a fiber cable and lighter weight made the production cheaper and more convenient. During the years, the development of better components such as modulators and photodetectors increased the allowed transmission speeds per channel. The bandwidth of a single optical channel increased from few Mbit/s at the beginning to 1 Gbit/s, 10 Gbit/s and 40 Gibt/s nowadays. Multiplying this capacity of a single channel by the number of wavelengths which is now

possible to multiplex and considering the very low attenuation coefficient of modern fibers (0.2 dB/Km) and the invention in the 1990 of optical amplifiers, we are today able to deploy submarine optical communication links on single mode fibers of 7000 Km with capacities greater than 1 Tbit/s.

The meeting point of the development of either high performance computing architectures and the optical communication systems has been, more recently, the attempt to include the latest in the design of the former, also called *Systems on Chip* (SoC). The classical optical components are now cheap and integrable with CMOS technology. For this reason it is now possible to integrate a complete optical transmission system on a VLSI CMOS die [[4]]. It is expected that as VLSI processes *feature size* (the minimum size of a transistor or a wire in either the x or y dimension [14]) will shrink in the future, the interconnection between the various on-chip computational nodes will be a limiting factor for these systems [2].

1.3 Evaluation Parameters

During the study that will follow, we will consider various possible design strategies for on-chip optical interconnection networks and we will analyze some of them already proposed in the scientific literature of the last years for multi-many core architectures. We will compare all these approaches with respect to the way they influence the three most important parameters for an high performance computation:

- the *bandwidth* which is defined as the total amount of work done in a given time [14];
- the *latency* or *response time* that has been defined as the time between the start and the completion of an event [14];
- the *scalability*, i.e. the property of increasing the size of the system keeping a proportional cost and performance.

Of course, in order to achieve a cross disciplinary evaluation of the on-chip optical interconnection networks for multi-many core architectures, we will correlate these parameters with engineering parameters, such as:

- *design cost* expressed in terms of number of transceivers (couples of transmitters and receivers), modultors, filters, pin count, area constraints, waveguides number.
- *clock distribution*;
- *bandwidth* associated to the transceivers and waveguides.

1.4 The Need for Optical On-Chip Interconnects

A lot of effort has been given to reduce the latency of the gates while physical detrimental effects like resistance and electromagnetic interference still mainly impair the global performances of NoC (*Networks On-Chip*). For this reason, the delay of electrical wires is, in general, of the order of several clock cycles. The first assessment for the need of an on-chip optical interconnect traces back to 1984 when in [12] Goodman et al. anticipated that the speed of MOS circuits would have been limited by interconnection delays rather than gate delays.

1.4.1 Drawbacks of Electrical Interconnects

In [15], the main issues related to the electrical wiring are presented and an evaluation of the effects of miniaturization of gates and wires is developed. The main drawbacks of electrical wiring are mainly three:

- *load added to driving gates*, due to wire capacitance;
- *signal delay* caused by wire resistance, capacitance and inductance and proportional to the product of the first two of them.

$$delay \;=\; resistance \;\times\; capacitance \qquad [14] \tag{1.1}$$

  It is not sufficient to reduce the size of the wire since resistance and capacitance would increase depending also on geometry, materials and position with respect to other electromagnetic entities. This is probably the detrimental effect that mostly influences also the higher levels of every computing architecture. Larger and larger portions of a clock cycle are consumed by the propagation delay of electrical signals on wires [14].

- *signal noise* originated by inductive and capacitive coupling between wires.

These physical effects influence non-functional properties of Systems On Chip such as the clock distribution. Common electrical clock distribution networks for SoC can have a total length of up to kilometers and consume about $30 - 40\%$ of the total power consumed by a chip [43].

1.4.2 Proposed Electrical Solutions

In order to solve these issues, several strategies have been examined. *Wire-pipelining* [38] is one of these techniques used to reduce the latency of the signal propagation. It consists in storing the transmitted data in intermediate flip-flops distributed along long wires. The transmission becomes in this

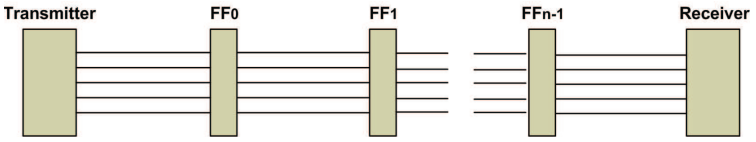

Figure 1.3: *Wire Pipelining implementation with Flip-Flops.*

way pipelined and a shorter clock duration becomes feasible [30] implying an higher bandwidth for the link.

However, there are several issues related to wire-pipelining [38]:

- the insertion of extra flip-flops in a circuit causes increased transfer delays due to the large number of registers required;
- an increase in the number of clock cycles and therefore of the operating frequency increases the power consumption.
- we experience a reduced throughput.

The problem of clock distribution at the upper layers of the integrated circuits has been addressed lowering the frequency of the global clock distribution networks and using local frequency multiplication circuits placed between the global and local clock networks [43]. The drawback of this solution, of course, is an increased power consumption.

1.4.3 Advantages of Optical Interconnects

We can summarize the advantages of optical interconnects with respect to electrical interconnects as:

- *Higher signal propagation speed* and therefore *lower signal propagation latency.*
- *Lower power consumption.*
- *Higher bandwidth* thanks to the wavelength multiplexing obtained with DWDM (*Dense Wavelength Division Multiplexing*).
- *Increased immunity to electromagnetic noise.*

These three basic physical advantages imply other technical advantages such as:

- *Improved clock injection and signal quality.* The use of ultra short optical pulses (picosenconds) to drive the interconnects and distribute the clock signal as advocated by [35], can remove the skew and jitter from the clock signals resulting in optimal timing and rising edges faster then those generated electrically.

- *Distance independent bit rates* between two communicating components.

- *Larger bandwidth density* (i.e. the ratio between the bandwidth of a link and the area occupied by the waveguides).

1.4.4 Drawbacks and Limitations of Optical Interconnects

Among the limitations on the benefits of exploiting optical interconnects, we have the memory latency problem. This problem, of course, does not directly regard the on-chip dimension since the DRAM memory is placed outside the chip of the processor but severely impacts the overall performance of many-core parallel processors diminishing the advantageous gain brought by the introduction of on-chip interconnects. In order to understand the problem we have to recall the structure of modern DRAM (*Dynamic Random Access Memory*) memories. This kind of memories stores bits in matrixes of elementary memory cells. Each memory cell is composed by a capacitor and a transistor which together form a sample and hold circuit. The capacitor stores a charge which represents the state of the bit (0 when the charge is under a threshold). The transistor, on the other hand, is used to allow read and write accesses to the capacitor. The bit stored in the cell, when read, is compared with the value of a timed reference cell and is then forwarded along a bit line wich is usually shared with several other memory cells.

The maximum number of bits per line can be obtained computing the ratio between the capacitance of the storage node and the capacitance of the bit line. When the charge is taken from the capacitor (the read operation is destructive with respected to the hosted charge and the latest needs therefore to be periodically refreshed) and is gated along the bit line, it spreads by the RC (*resistance-capacitance*) effect impairing the access time that turns out to depend quadratically on the number of bits per line Hence the latency of the memory access cannot be reduced directly by optical interconnect despite the fact that in complex chip to chip and board to board distributed memory organizations, the introduction of optical memory interconnects can alleviate the problem [17]. Cache hierarchies have been elected as the most performant way to address this problem. Another technological issue more strictly related to optical on-chip interconnects is the fact that optical components require a thigh temperature stability, specially when using wavelength division multiplexing (WDM).

1.4.5 Electrical vs Optical Interconnects

A comparison between the scaling properties of electrical and optical point-to-point interconnects is presented in [8] where the results stated that while the delay of electrical interconnects remains approximately steady at im-

proved scaling technologies, its optical counterpart is capable of directly proportional delays (Table 1.1):

Year	2004	2007	2010	2013	2016
Technology	90nm	65nm	45nm	32nm	22nm
Electrical delay $[ps/cm]$	311.9	313.2	291.3	312.0	317.8
Optical delay $[ps/cm]$	238.9	173.3	145.4	127.7	114.9

Table 1.1: *Delay (ps/cm) of Electrical and Optical P-t-P Interconnects.*

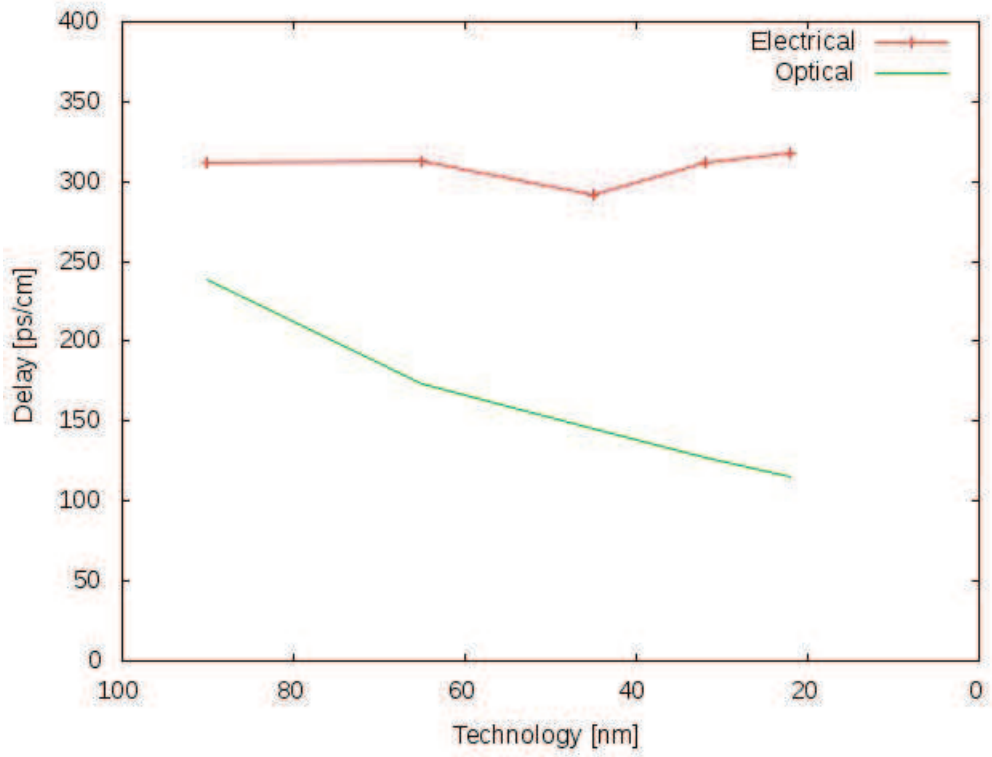

Figure 1.4: *Delay (ps/cm) of Electrical and Optical p-t-p interconnects.*

From Table 1.2 we can recognize that the power consumption of electrical interconnects increases with the scaling technology but is always greater than its equivalent optical implementations.

The ratio between the power required in the optical interconnect and the power required in the electrical interconnect:

$$R = \frac{P_{optical}}{P_{electrical}} \tag{1.2}$$

Year	2004	2007	2010	2013	2016
Technology	90nm	65nm	45nm	32nm	22nm
Electrical int. power [mW]	9.8	16.9	21.7	33.4	45.3
Optical int. power [mW]	1.4	2.4	3.7	6.2	11.5

Table 1.2: *Power (mW) of Electrical and Optical p-t-p Interconnects.*

for different scaling technologies is shown in Table 1.3 and in the graph depicted in Figure 1.6

Year	2004	2007	2010	2013	2016
Technology	90nm	65nm	45nm	32nm	22nm
R	7	7.04	5.86	5.39	3.94

Table 1.3: *Optical vs. Electrical Power Requirements for p-t-p Interconnects.*

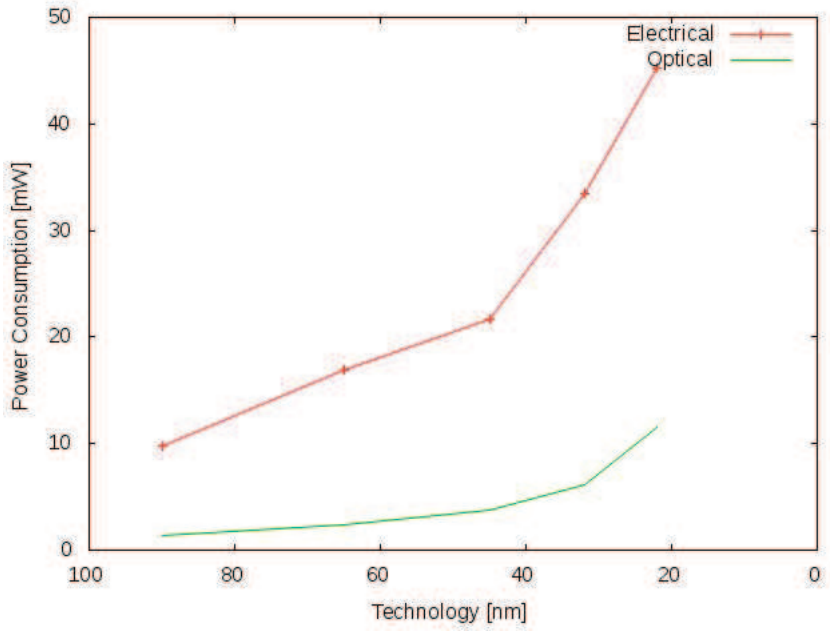

Figure 1.5: *Power (mW) of Electrical and Optical P-t-P Interconnects.*

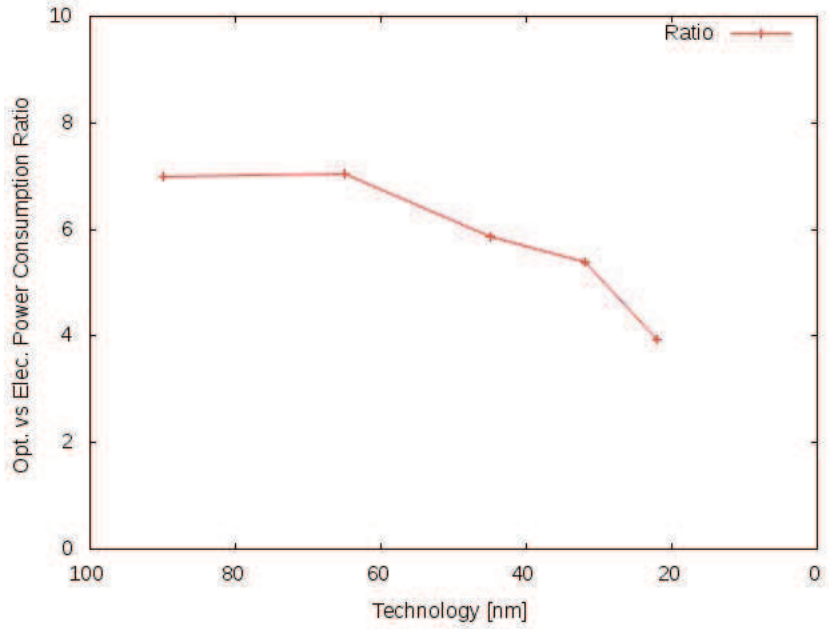

Figure 1.6: *Optical vs. Electrical Power Requirements for p-t-p Interconnects.*

Part I

Design of On-Chip Optical Networks

Chapter 2

Optical NoC Design

In this chapter we discuss in further detail the issues that optical networks on-chip promise to solve and consider some of the proposals that have been made in the literature. All the solutions that follow are the result of the research in the field of the last years. In Section 2.1 we analyze the solutions proposed to solve the global clock distribution problem discussed in Chapter 1. In Section 2.2 we consider some of the most promising solutions for integrating optical components on CMOS chips.

2.1 Optical Clock Distribution

In [43] and [39], an optical clock distribution network (CDN) is proposed, analyzed and compared with equivalent electrical solutions. The optical H-tree is shown in Figure 2.1:

The clock signal is generated by an off-chip vertical cavity surface emitting laser (VCSEL) and, after being coupled to a passive waveguide, it is distributed to all the targets where it is finally converted to the electrical domain and then distrubuted to the local electrical networks. The planar waveguide is realized with a core in silicon (Si) and a cladding of silicon dioxide (SiO2) in order to achieve high guiding with bend radius of up to few μm for the waveband 1.3-1.55 μm. The waveguide operates in single mode regime in such a way to avoid modal dispersion and has a thickness of 0.2 μm and is 0.5 μm wide. The transmission loss has been estimated to be 1.3 dB/cm and the loss at each Y-junction is 0.2 dB. The prototype has been realized with the 70 nm technology.

2.2 CMOS Integration

There are currenty many technologies that has been proposed for the integration of optical communication components on CMOS chips. In this section we consider the most promising.

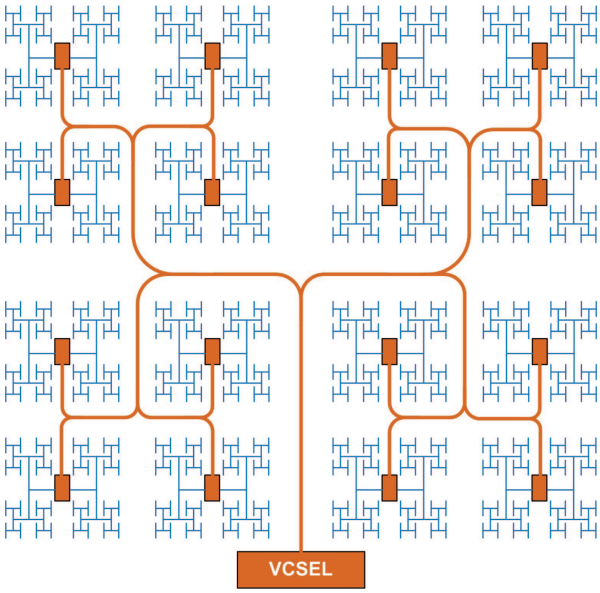

Figure 2.1: *Optical Global Clock Distribution with an H-tree [43].*

2.2.1 3D Integration

In the *3D integration* proposal [39] the basic idea is to grow an optical interconnection layer on top of a pre-fabricated electrical layer (Figure 2.2).

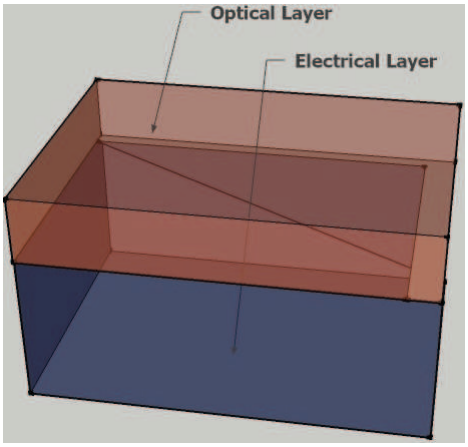

Figure 2.2: *Schematic View of a CMOS 3D Integration of an Optical Layer.*

With this strategy, the common CMOS procedure remains independent of the fabrication of the upper optical interconnection network. In Figure 2.3, for example, a III-V semiconductor VCSEL laser source is placed on the top of the chip and connected to an adjacent waveguide which brings the signal to the correspondents III-V semiconductor photodetectors. The

waveguide is realized using Si for the core and SiO_2 for the cladding. The electrical driver circuit for the laser is placed on the bottom of the wafer and is connected to the uppermost optical layer with a stacked electrical connection. On the other hand, the receiver circuit, placed again at the bottom of the wafer, is connected through another electrical and vertically stacked link to the photodetector in order to collect the photodetected data.

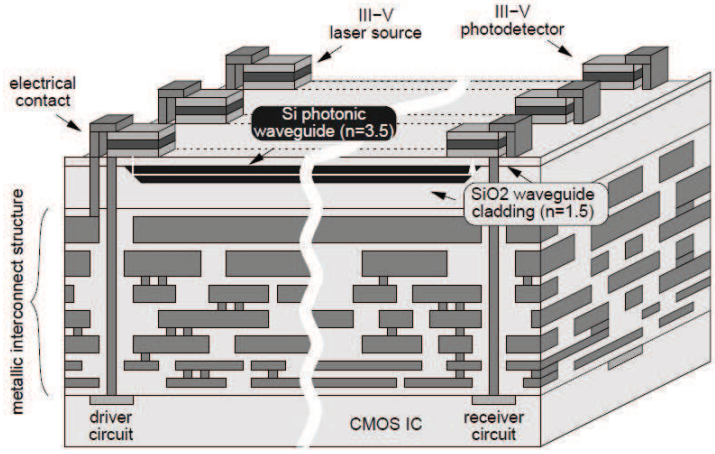

Figure 2.3: *Cross-Section of Hibridised Interconnection Structure [39].*

2.2.2 Monolithic Integration

The monolithic integration strategy corresponds to develop the various optical components using the standard process for logic on-chip components. Monolithic integration usually requires some further processing but its overall cost can be less than the one required by 3D integration because specialized processes are not required and, at the operational phase, they require less power and area to interface electrical and optical components [18].

2.2.3 Free-Space Interconnect

Another solution envisioned by the designer is the integration of the communication layer through on-chip free-space optical interconnects. As stated in [2], the strength of this solution consists in the possibility of providing a very high density of interconnections.

Chapter 3

Optical Components

In this chapter we analyze which are the main architectural components which constitute an on-chip optical interconnection network. The general structure of optical components proposed in literature and their integration with each other are studied in order to understand which are the main ingredients of an on-chip optical network with CMOS integration. While 10 years ago the cost of the integration of optical devices with the CMOS technology and its incompatibility was considered a huge hurdle in the development of photonic integrated chips, the chip technology evolution completely changed the expectations on that [21]. In order to consider the state-of-the-art technologies, we sometimes refer to the last tables provided by the ITRS (*International Technology Roadmap for Semiconductors*) in 2010. We start from the analysis of a simple point to point optical on-chip communication link. In an on-chip optical communication link, like in their counterparts of geographical size, a point to point communication link is basically composed by three main components: a *transmitter*, a *waveguide* and a *receiver*: Figure 3.1.

Figure 3.1: *Logical View of a Point to Point Optical Link.*

The purpose of the transmitter is, provided a driving electrical and digital signal, to carve the information on an optical carrier which can be generated within the transmitter or provided from outside. A transmitter can, in principle, transmit symbols (bits) in parallel exploiting a limited set of different wavelength. The *receiver* is a composite component which photodetects the incoming photons generating a proportional electrical current; converts the current in voltage changes and finally digitalizes its values. A receiver must also, if used in conjunction with a multiwavelength transmitter, filter the different wavelength components of light prior to photodetection. The

waveguide is nothing else but the structure through which the light travels. It is usually characterized by different implementation structures, each one with different guiding properties.

3.1 Transmitters

The *transmitter* always includes a laser source which is a solid state LASER (*Light Amplification by Stimulated Emission of Radiation*). A laser is an optical oscillator that, provided a feedback, is able to emit a coherent beam of light [41]. In typical designs, this component is placed outside the chip [20] due to the relatively large chip area that it occupies and to the critical temperature stability that it requires and that must be provided by dedicated controllers. The light beam generated in this way is characterized by an amplitude and therefore an energy constant over time. In order to imprint the digital data onto the optical carrier, two strategies are exploited: the first consists in directly modulating the optical carrier using a digital LASER driving current provided by the LASER *driver* and is therefore called *direct modulation*. The second strategy consists in designing an external device called *optical modulator* which, opportunely placed after the laser can carve the amplitude of the optical carrier according to its digital driving current provided by the *driver* module following the OOK (*On-Off Keying*) encoding rule.

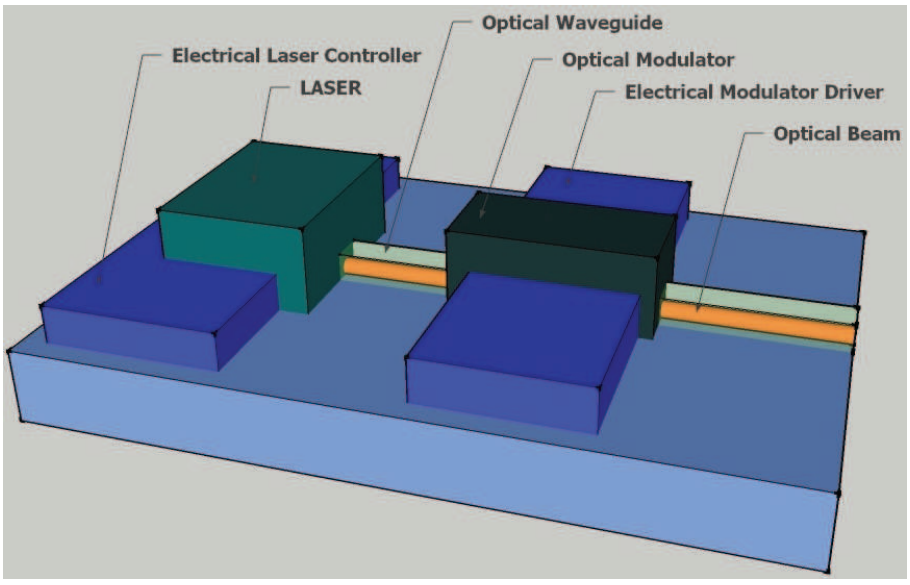

Figure 3.2: *Internal Structure for an Optical Transmitter.*

The second strategy presents many advantages in terms of quality of the eye diagram of the modulated optical signal and, therefore, in our analy-

sis and in the modern literature we consider *external modulation*. At the output of the optical transmitter, an *optical coupler* leads the light into the *waveguide*.

3.1.1 On-chip Ge-on-Si lasers

In recent articles such as [33], [27], [28], [34], [29], [5], [6] Ge-on-Si lasers are investigated. As discussed in the previous chapter, monolithic integration of lasers could enable a wide integration of photonic components on-chip. Germanium (Ge) is almost a direct gap material and is highly compatible with silicon CMOS. Photoluminescence effects have been demonstrated at room temperature using edge emitting waveguide devices. The emission band was in the range $1590 - 1610\ nm$. In order to obtain these results, germanium has been band-engineered with n-type doping.

3.1.2 Off-chip VCSELs

The most efficient and common off-chip microresonator lasers are the *Vertical Cavity Surface-Emitting Lasers* (VCSELs). Their structure is such that light emerges from the top face of a 1D planar microresonator [41].

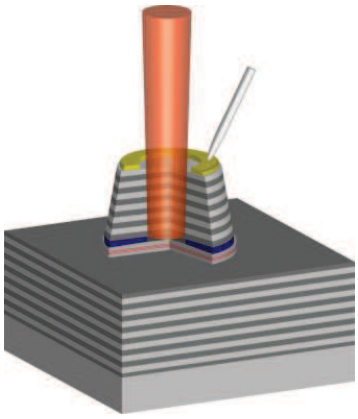

Figure 3.3: *VCSEL Working Principle.*

The VCSELs can be realized with a diameter as small as about 1 μm and can incorporate other features such as direct modulation at speeds up to 40 $Gbit/s$ [41]. The choice of this kind of lasers for the realization of on-chip optical interconnects is motivated by the fact that they have high packaging densities compared to the other types: $InGaAs$ quantum-well VCSELs with a diameter of 2 μm, an height of 5.5 μm and a operating wavelength of 970 nm have already been fabricated [41].

3.1.3 On-chip Raman Laser

In [40], the first on-chip continuous wave silicon Raman laser has been demonstrated. The laser is composed by a p-i-n diode embedded in a silicon waveguide and its cavity includes multilayer dielectric film-based mirrors. The lasing wavelength can be adjusted by changing the wavelength of the pump laser. The waveguide on which the laser is based is a low loss silicon-on-insulator (SOI) rib waveguide. The front mirror presents a double reflectivity of 71% for the lasing wavelength and of 24% for the pump at the 1550 nm wavelength. The other mirror presents a wideband reflectivity of about 90% for both the lasing wavelength and the pump.

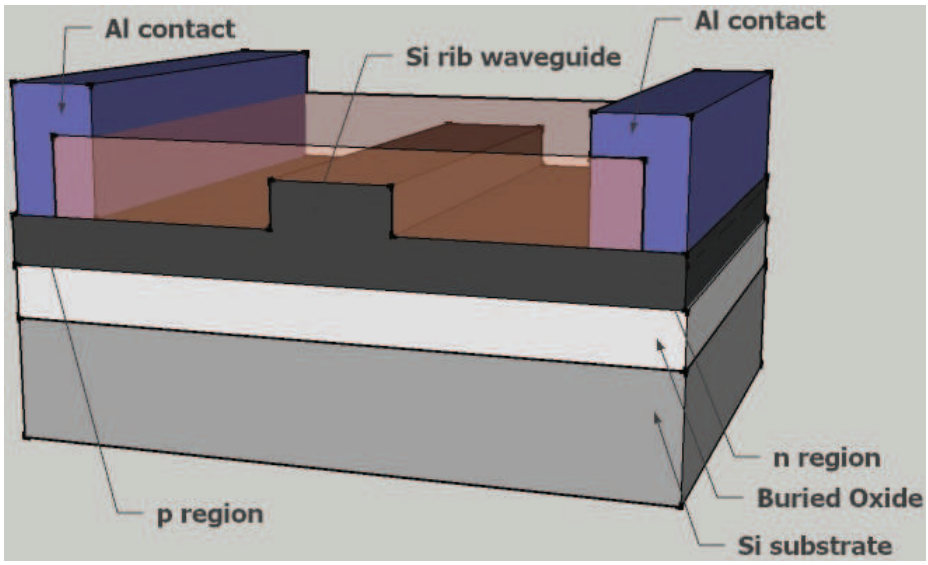

Figure 3.4: *Structure of the Raman laser [40].*

The lasing cavity is obtained with an s-band configuration of the waveguide and has a total length of 4.8 cm. The lasing is obtained using a reversed bias of 25V and a pump power of about 700mW obtaining a single-pass net gain greater than 3dB.

Despite the fact that this was the first demonstration of on-chip Raman laser (2005), its characteristics in terms of wavelength tuning and linewidth (80 MHz) could be good but for the fact that the setup required the chip to be mounted on a thermo electric cooler keeping the temperature of it fixed at 25 degrees Celsius: a temperature much lower than the one expected for an operating manycore chip.

3.2 Modulators

Optical modulators can write the electrical digital data onto the phase or the amplitude of an input optical signal. Exploiting the *Mach-Zender Interferometer* (MZI) scheme, phase modulation can be transformed in amplitude modulation.

One of the biggest challenges [26] that has been faced in the realization of silicon modulators has been the lower speed compared with those realized using III-V semiconductor compounds.

In [26], an high speed silicon optical modulator based on the MOS technology and operating at frequencies higher than 1GHz has been demonstrated. The modulator is a phase shifter but it has been tested in a MZI scheme. The modulator has been realized doping in an n-type way the silicon layer of the SOI wafer and realizing a p-type doped polysilicon rib with a 12 nm thick gate oxide between the two doped layers. The modulator is a single mode device operating at wavelengths of about 1550 nm and has an higher efficiency for TE polarized light (it is highly polarization sensitive). The length of the phase shifting waveguide is 10 mm making impossible its integration in a manycore architecture.

3.2.1 Microring Modulators

One of the most promising wavelength sensitive optical components which can be realized using *silicon-on-insulator* technology are the *optical microring resonators*. *Microresonators* are optical resonant structures where at least one of the dimensions approaches the size of few wavelengths or less [41]. Their basic structure is composed by two parallel waveguides intersected by a microring.

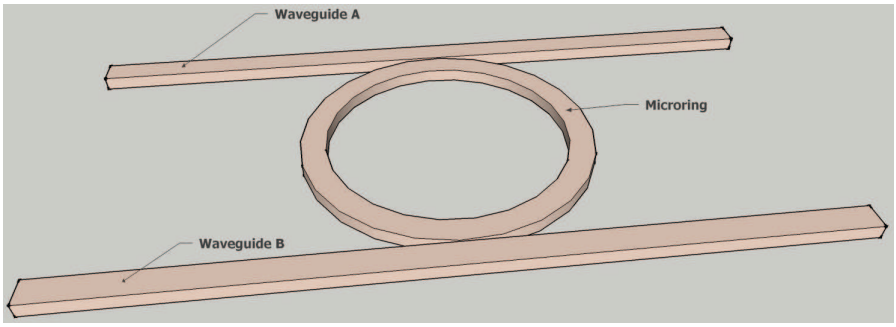

Figure 3.5: *Microring Resonator in Parallel Waveguide Configuration.*

Micro-ring resonators can be used as external modulators in indirect modulation configurations. In [11] an optical microring resonator with a radius of 1.5 μm is presented. The footprint of such devices can be as small as 10 μm and they can have a bandwidth of up to 12.5 Gb/s, a power

dissipation as low as 0.1 mW and an extinction ratio higher than 9 dB [47]. One disadvantage of microring modulators is the variation of the resonant wavelength due to variation in the operating temperature. It is clear that this issue requires on-chip temperature control which can increase the area required by such components.

3.3 Waveguides

In this chapter we analyze the technologies applied to the realization of the light conduits: the waveguides. The term *waveguide* is a general term but in the common practice is used to refer to chip-scale optical guides.

The *optical waveguide* is a passive optical component composed by a light pipe based on a slab, strip or cylinder of dielectric material embedded in another dielectric material of lower refractive index. The refractive index of the core

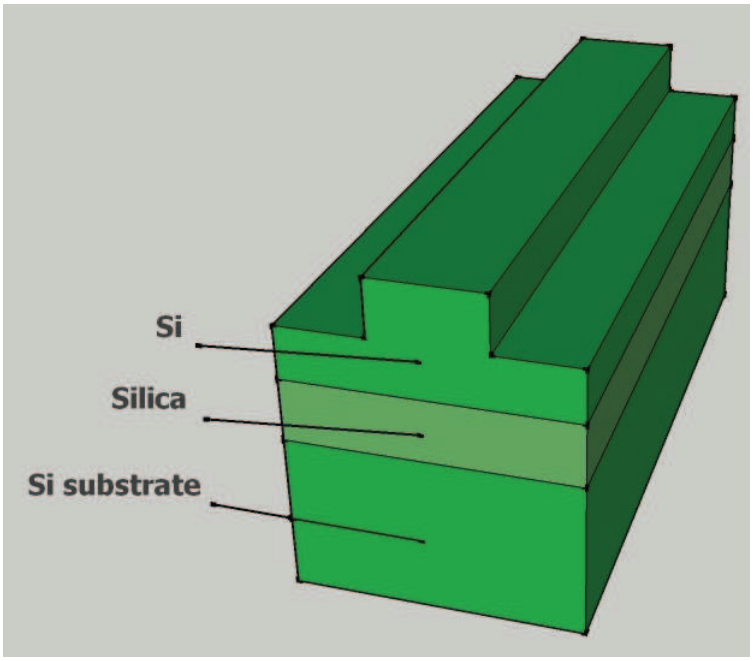

Figure 3.6: *A Waveguide Fabricated with the Silicon-On-Insulator (SOI) Technology.*

material influences the speed at which light travels inside the waveguide: the lower the refractive index, the higher is the speed. A critical characteristic of this passive component is the maximum ray of curvature that it can support: steep curves can overcome the guiding property of the medium producing high losses in the power of the signal. Also waveguide crossings can produce high losses. The guiding property of the waveguide depends

on its thickness and on the difference of the refractive indexes respectively of the core and of the cladding materials: the higher the difference, the higher the light confinement. The main characteristics of waveguides are the *propagation loss*, i.e. the amount of optical power lost per unit of distance (centimeters in chip facturing); the *numerical aperture*, the *effective index* and the corresponding *latency per unit of length*.

3.3.1 Rib Waveguides

In [7], the possibility for sub-micrometric (partially etched) waveguides on silicon-on-insulator (SOI) are explored. The studied prototype has been realized with *reactive ion-etching* (RIE). The rib width was 1 μm and the height was 0.38 μm.

3.3.2 Hollow Metal Waveguides (HMWG)

This kind of waveguides is characterized by a propagation loss smaller than 0.05 dB/cm and they are easy to fabricate. The HMWGs have a numerical aperture smaller than 0.01 which allows for the insertion of beamsplitters with low losses; they also have an effective index of about 1 which makes them one of the fastest types of waveguides able to guarantee extremely low latencies: 0.033 ns/cm.

Propagation Loss	$< 0.05 dB/cm$
Numerical Aperture	< 0.01
Effective Index	~ 1
Latency per unit Length	$0.033 ns/cm$

Table 3.1: *Summary of the HMWG characteristics.*

This type of waveguides has a rectangual cross section with the size of a human hair [42]. The core material is a metal with high refractive index. This kind of waveguide is also particulary resistent to temperature variations which make them a good candidate for chip to chip and on-chip interconnections.

3.4 Receivers

The *optical receiver* receives the transmitted optical stream and performs an optoelectrical conversion of it. Its main building blocks are a

photodetector and a *trans-impedance amplifier* (TIA) [20]. The photodetector generates an electrical current according to the received optical power. The most important parameters for a photodetector are the *quantum efficiency* which is the ability to convert the incoming optical power

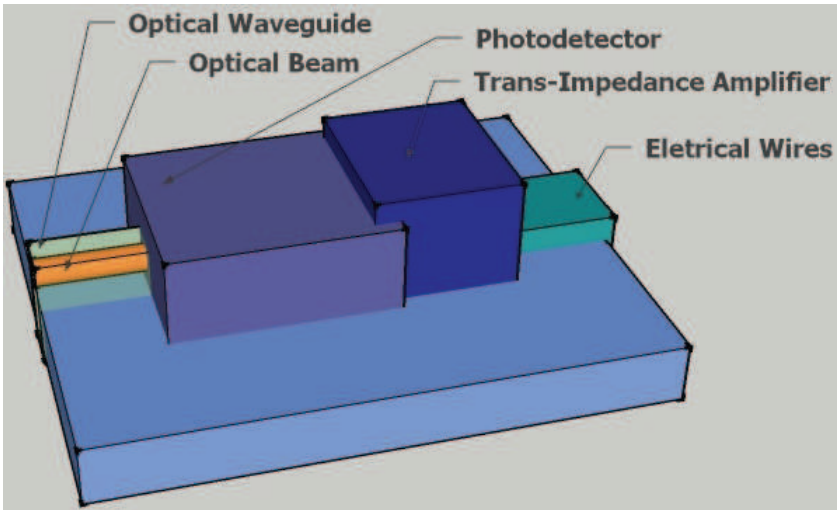

Figure 3.7: *Internal Structure of an Optical Receiver.*

to a correspondent electrical current, and the size. The trans-impedance amplifier then converts the current to a voltage thresholded by other stages to digital levels [20].

3.4.1 Germanium on SOI Photodetectors

In [22], receivers based on Ge-on-SOI (Ge-on-silicon-on-insulator) photodiodes with associated high-gain CMOS amplifiers are investigated. These receivers, using Ge-on-SOI lateral p-i-n photodiodes, can operate at 15 Gb/s with a sensitivity of -7.4 dBm (with a BER of 10^{-12}) and a supply voltage of 2.4 V. They offer also a good temperature stability: the same receivers can operate with a 5 Gb/s sensitivity constant up to 93^{o}C and at 10 Gb/s up to 85^{o}C. A detection bandwitdth of 19 Gb/s using a supply voltage of 1.8 V and a single-ended high speed receiver front-end has been demonstrated. CMOS IC can operate at 10 Gb/s using a 1.1 V supply while consuming only 11 mW.

3.5 Filters

3.5.1 Microring Resonator Filters

As for the case of microring resonators used as modulators, microring-based optical filters require a strict temperature control in order not to modify the resonant wavelength. Their working principle is simple: they extract from an adjacent waveguide a single wavelength signal capturing it in the resonant ring. Their characteristics are similar to ring modulators.

Part II

Architectural Paradigms and Cost Models

Chapter 4

Introduction

After having analyzed some of the most important issues for the design and realization of the most notable on-chip optical components and before going through the high level structures of the on-chip optical interconnection networks, we first need to characterize the two main architectural paradigms which can be found in current multi-many core architectures. This classification will allow us to figure out for each interconnection network which is (are) its target architectural paradigms.

4.1 On-Chip Shared Memory Architectures

In shared memory on-chip MIMD (*Multiple Instruction Stream - Multiple Data Stream*) architectures, several cores are connected to each other through an *interconnection network* [45]. The connection provided by the network can be direct or indirect. Since the cores are usually the result of a standard design, they require an *interface unit*, also called *wrapping unit*, in order to communicate with the other cores using the firmaware protocol imposed by the interconnection network the firmware messages pass through. In shared memory systems (SMPs) the messages are load/store memory requests that have a scope which can be local or external to the memory of the core. The *interface unit W* is capable of intercepting these requests and, in case of reference to external memory blocks, it operates the proper transformation of the request in order for it to be forwarded along the interconnection network [45].

Among the shared memory multiprocessors, depending on the organization of the shared memory, we can distinguish *Uniform Memory Access* (UMA) architectures and *Non Uniform Memory Access* architectures.

Chapter 5

UMA: Uniform Memory Access Architectures

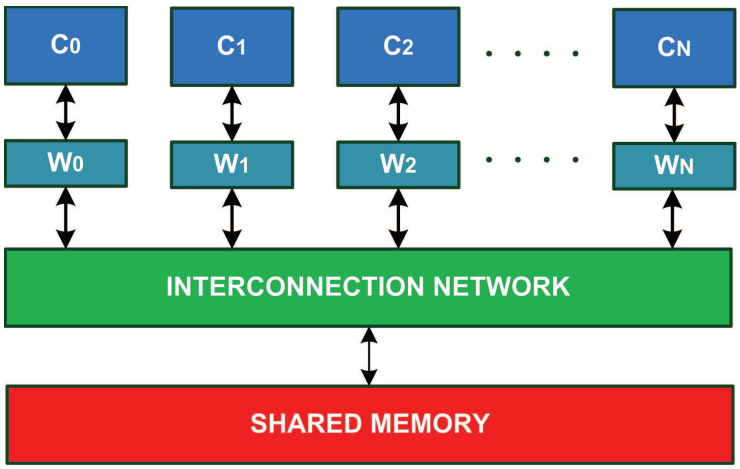

Figure 5.1: *Logical Structure of a UMA Architecture.*

In this chapter we consider the Uniform Memory Access (UMA) architecture. In UMAs the shared memory is a separate block phisically connected to the various cores through the interconnection network. The communication between two or more cores takes place with a `store` and `load` in and from some shared memory areas in the main memory. This model of communication eases the task of the communication run time support designers but, at the same time, represents a bottleneck for the scalability of the architecture as the number of cores increases.

To the *base latency* of the memory, i.e. the latency without memory block contention L_{mem} (which is dependent on the implementation) we have to add the latency of the interconnection network ($L_{network}$) that has po-

tentially to deal with congestion and other limiting factors [45]:

$$L_{comm} = 2\ L_{network} + L_{mem} \tag{5.1}$$

The access latency of the memory can be decreased implementing it with independent modules and/or partially hidden integrating cache hierarchies inside the core itself: typically one or two levels. With this considerations, the logical view of our SMP model becomes:

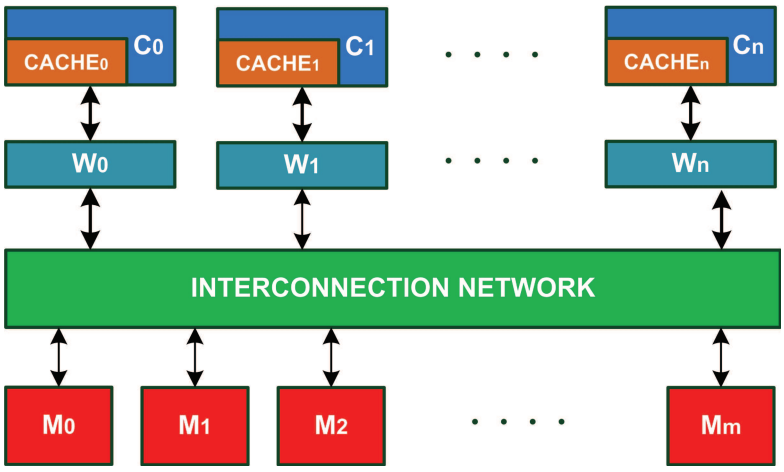

Figure 5.2: *Extended Logical Structure for a UMA Architecture.*

The cost model for an interprocess communication remains the same but the access to the memory reserved to each process can be reduced reading the data directly in the cache built-in inside the core area. The cost model for an access to the memory of each process can then be formulated as:

$$L_{memAccess} = f L_{cache} + (1 - f)(L_{network} + L_{mem}) \tag{5.2}$$

where $f \in [0, 1]$ is the fraction of accesses to data items already in cache and L_{cache} is the latency of the cache. If the cache is single and built within the core, its latency is practically 0 resulting in:

$$L_{memAccess} = (1 - f)(L_{network} + L_{mem}) \tag{5.3}$$

On the other hand, if the cache is structured in a hierarchy of progressively faster and smaller units, the cost model varies becoming more complex and reflecting the different latencies of each level. Since the memory access latency is equal for all the cores, we have that the most natural mapping of processes on cores is the anonimous one: the execution of a process can be scheduled on different cores every time and every time it is awaken from the wait state it can again be assigned to different cores. This solution also allows for a good load balancing among the cores.

5.1 Study Cases

In the embedded applications field, *Tilera* realized several interesting many-core architectures.

5.1.1 Tilera TILE64 Processor

The Tilera TILE64 Processor family is equipped with 64 general purpose cores organized in an 8×8 grid interconnected by a proprietary mesh interconnect with a 37 Tbps aggregate bandwidth. Each core incorporates a CPU which exploits ILP with a three-way 32-bit VLIW pipeline with a 64-bit instruction bundle; L1 and L2 caches and a non-blocking switch. The cores operate at a frequency between 700 and 866 MHz for a correspondent power consumption of 19-23 W at 700 MHz running a full application. Furthermore idle tiles can be set to a low power sleep-mode state.

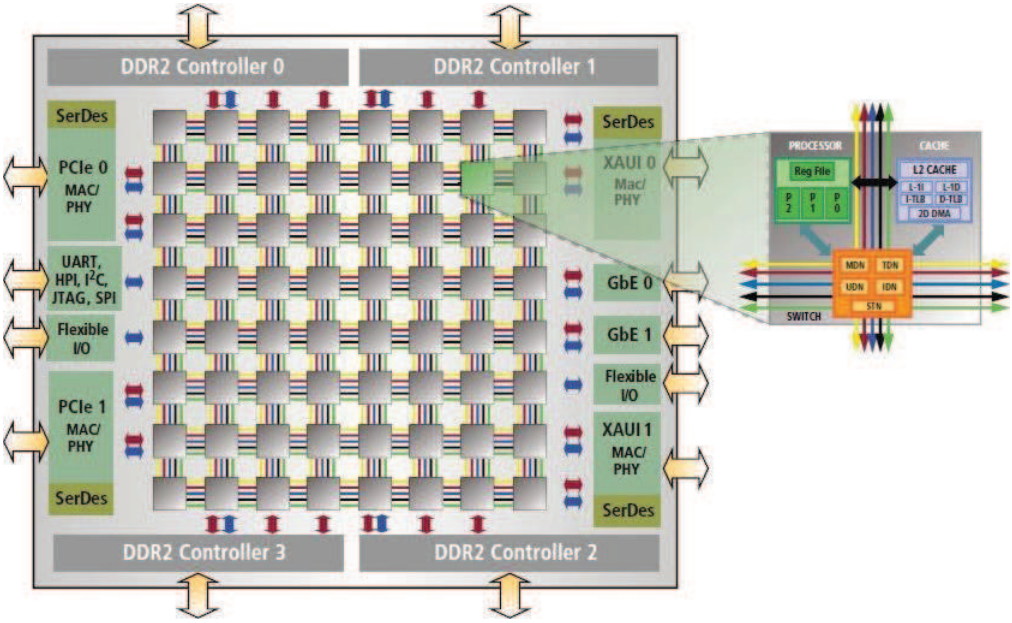

Figure 5.3: *Tilera TILE64 Processor.*

On the same chip, 2 10 *GbE* network interfaces, 200 *Gbs* of memory bandwidth obtained with 4 64 bit DDR2 memory controllers and high performance I/O interfaces are present. Each core is C/C++ programmable and can execute a different operating system or a group of them can run a multiprocessor operating system such as SMP Linux. While the chip is overall a general purpose one, some networking functions can be very efficiently executed on such a platform enabling 20 *Gbs* nProbe; more than 15 *Gbs* Snort; H.264 HD encode for 10 streams of 1080×720 pixels and x.264 HD encode for 4 streams of 720×480 pixels.

5.1.2 Tilera-Gx8036 Processor

The Tilera-Gx8036 Processor is optimized for networking and multimedia applications. It features 36 cores organized in a 6×6 grid and operating in the range of $[1.0, 1.2]$ GHz. It integrates Two 72-bit DDR3 controllers with ECC support. Every core includes a 32 KB L1 instruction cache, a 32 KB L1 data cache and a 256 KB L2 cache. The whole chip has a shared 9 MB L3 coherent cache.

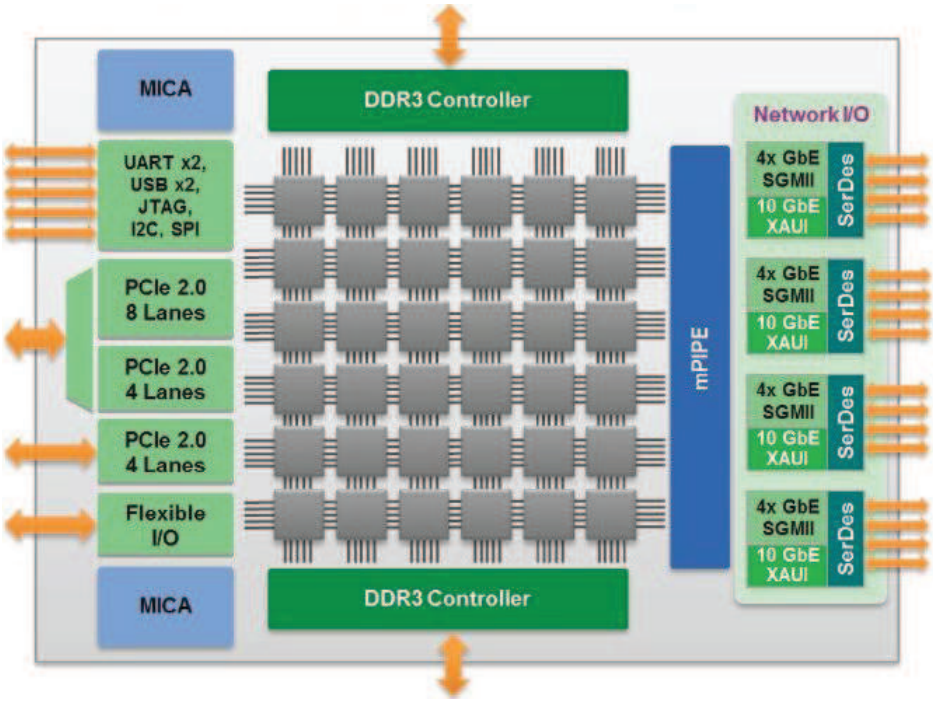

Figure 5.4: *Tilera Gx8036 Processor.*

The proprietary interconnect is composed by 5 overlapped and independent low latency mesh networks. The non-blocking network has a 60 Tbps aggregate bandwidth and is characterized by cuthrough switching with 1 clock cycle per hop.

5.1.3 Tilera 100 Core

The highest core count processor of the series produced by Tilera has been the 100 core prototype chip. Its structure, very similar to the other models, is shown in the figure below:

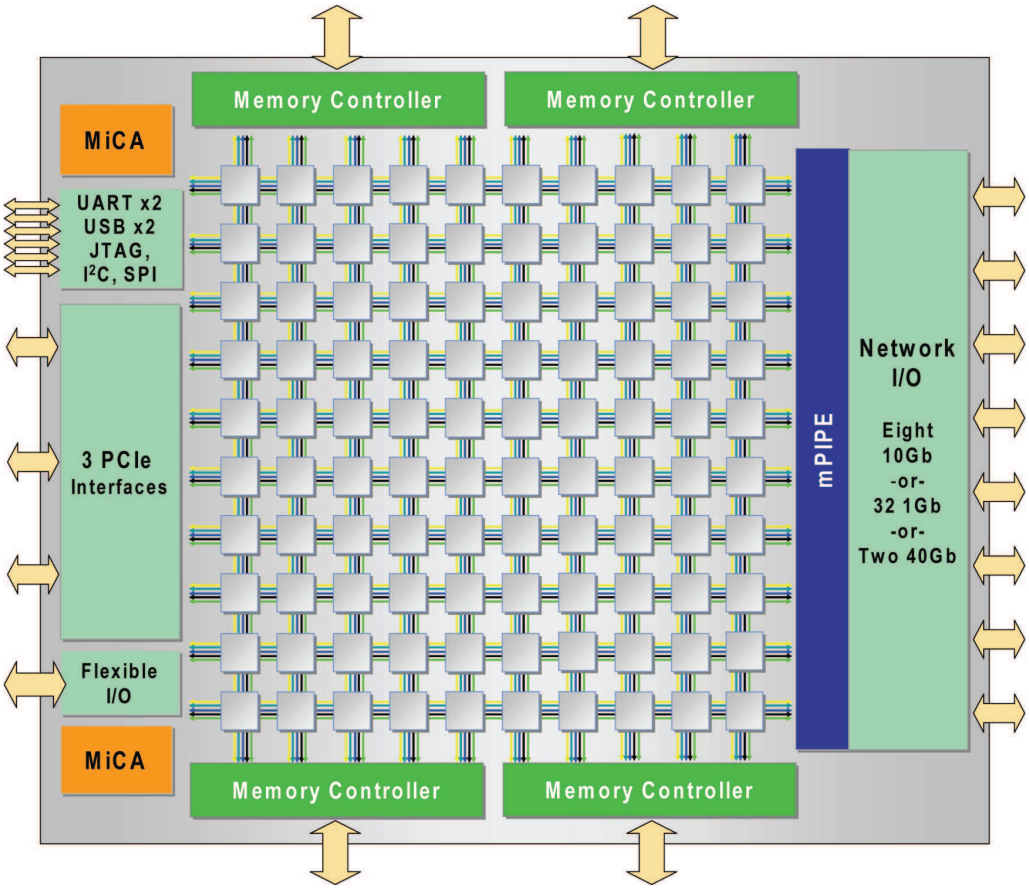

Figure 5.5: *Tilera Gx 100 Core Processor.*

5.1.4 AMD Opteron 6200

AMD recently realized a 16 core processor for servers. The processor shown in Figure 5.6 is composed by 8 tiles containing 2 cores. Every core has an operating clock frequency ranging from 2.7 to 3.4 GHz. The L3 cache is 16 MB large and is shared among all the tiles. The whole chip consumes about 140 W.

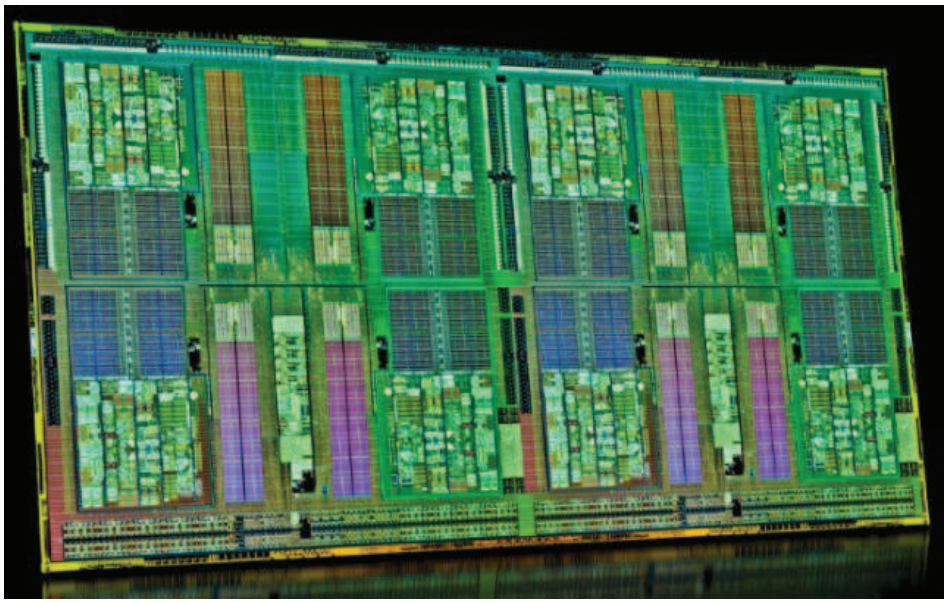

Figure 5.6: *AMD Opteron 6200, a 16 Core Processor.*

5.1.5 AMD FX Family of Processors

The multicore revolution is currently involving also the desktop world. AMD has realized the family of processors FX (Figure 5.7) with up to 8 cores realized with a 32nm SOI (*Silicon On Insulator*) technology and operating at frequencies in the range between 3.60 and 4.20 GHz.

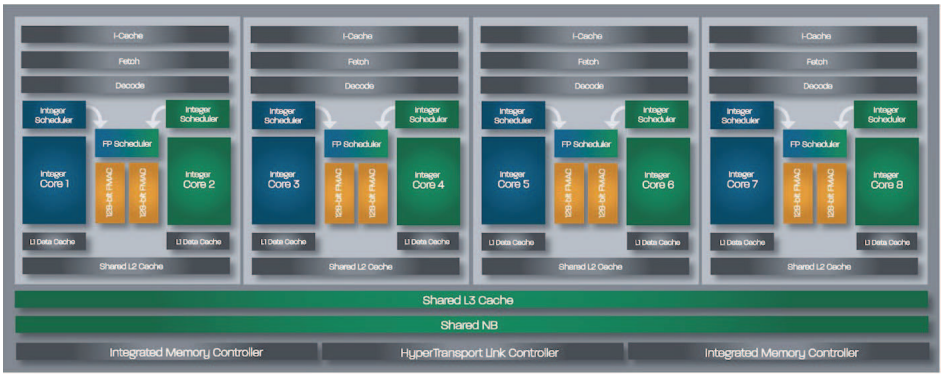

Figure 5.7: *AMD FX 8 Core Processor.*

The cores are grouped in 4 tiles containing 2 cores each one. All the tiles share an 8MB L3 cache. Each tile has a 2MB L2 cache for data and instructions while each core has an associated 64KB L1 cache for the data and one 64KB L1 cache for the instructions. Also for the case of this family of processors, the memory controller (for up to DDR3 1866) and the I/O

controller are integrated on chip and shared among all the tiles. The power consumption for this family of processors is 125W.

5.1.6 Intel Xeon Processor E7-8870

The Intel Xeon Processor E7-8870 64-bit processor (Figure 5.8) designed for servers and HPC is equipped with 10 general purpose cores and is capable of executing up to 20 threads at the same time. The frequency at which operates every core ranges from 2.4 to 2.8 GHz.

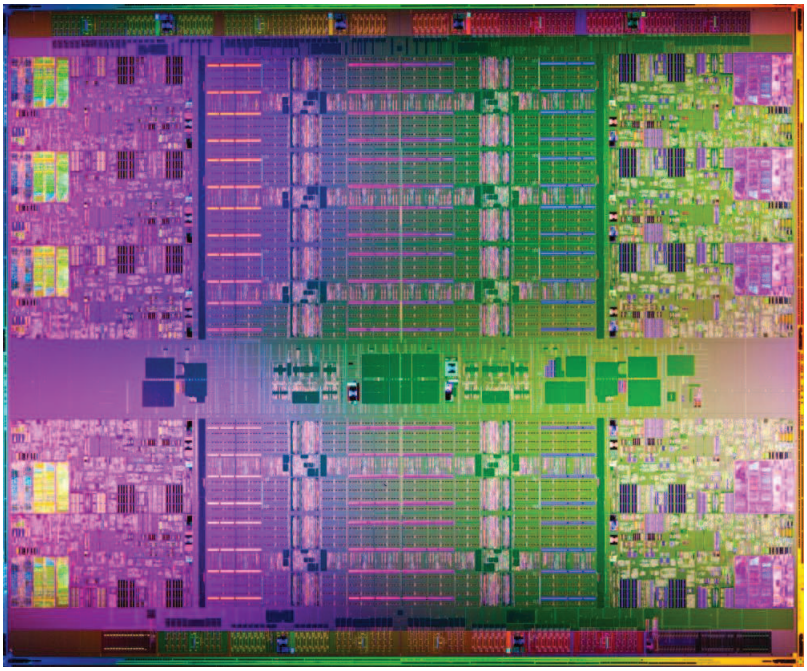

Figure 5.8: *Intel Xeon E7 10 Core Processor.*

The L3 cache memory is 30 MB large and is shared among all the cores. Every core has its own 256 MB dedicated L2 data cache. Differently from all the other architectures presented before, the Intel Xeon Processor E7-8870 presents a L1 cache within each core that is characterized by a size of 32 KB for the instructions and a size of 16 KB for the data. This asymmetry is unique in the market. The processor chip is realized with the 32 nm lithography technology.

Chapter 6

NUMA: Non Uniform Memory Access Architectures

In *Non Uniform Memory Access* (NUMA) architectures the shared memory is composed by a set of N modules equally distributed among the cores.

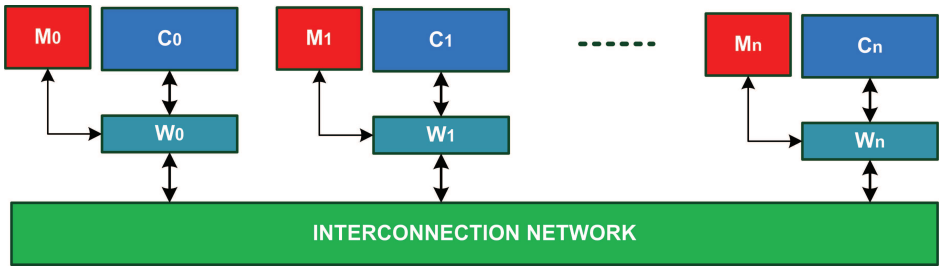

Figure 6.1: *Logical Structure for a NUMA Architecture.*

The memory access strategy can be enhanced with respect to the one exploited by SMP architectures exploiting the principle of locality of data and instructions. Each core can access its memory module with a much lower latency with respect to the other blocks. Following the caching considerations of the previous chapter, we can state the latency of the memory access of one core as:

$$L_{memAccess} = gL_{localMem} + (1-g)(L_{network} + L_{remoteMem}) \tag{6.1}$$

where $g \in [0,1]$ is the fraction of accesses to the memory module local to the core, $L_{localMem}$ is the access latency of the local memory module and $L_{remoteMem}$ is the access latency of a remote memory module.

Due to the different latencies of accessing a local or a remote memory module, a natural consequence is the dedicated mapping of processes to

cores. Keeping the execution of a process on the same core allows it to execute the majority of the memory accesses on its local block.

The latency of the local memory module can be still partially hidden integrating cache hierarchies inside each core. With this considerations, the logical view of our NUMA model becomes:

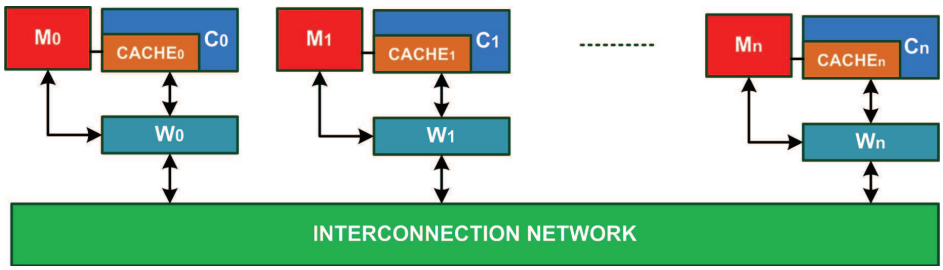

Figure 6.2: *Extended Logical Structure for a NUMA Architecture.*

This improvement in the architecture is confirmed by then new cost model:

$$L_{memAccess} = g\Big(fL_{cache} + (1-f)L_{localMem}\Big) + (1-g)(L_{network} + L_{remoteMem}) \tag{6.2}$$

where $f \in [0,1]$ is the fraction of accesses to the local cache within the core and L_{cache} is the latency of the cache.

This solution is, of course, less prone to load balancing but can outperform the SMP one in terms of scalability and average latency performance.

The role of the interconnection network in the overall performance of the system is, in this case, reduced even if the memory coherence and synchronization issues impose a non negligible overhead.

6.1 Study Cases

Several manycore architectures have already been demonstarted at research level and some of them are also already commercialized for special-purpose applications such as packet inspection, intrusion detection and prevention, video transcoding/transrating. In this section we analyze the most recent prototypes that have been realized.

6.1.1 Intel 80 Core

In [44] (2008), an 80 tile manycore architecture is presented. The architecture is organized in a 2D 10x8 mesh network and each tile is operating at 4 GHz. Each core is equipped with a PE (processing element) and a message

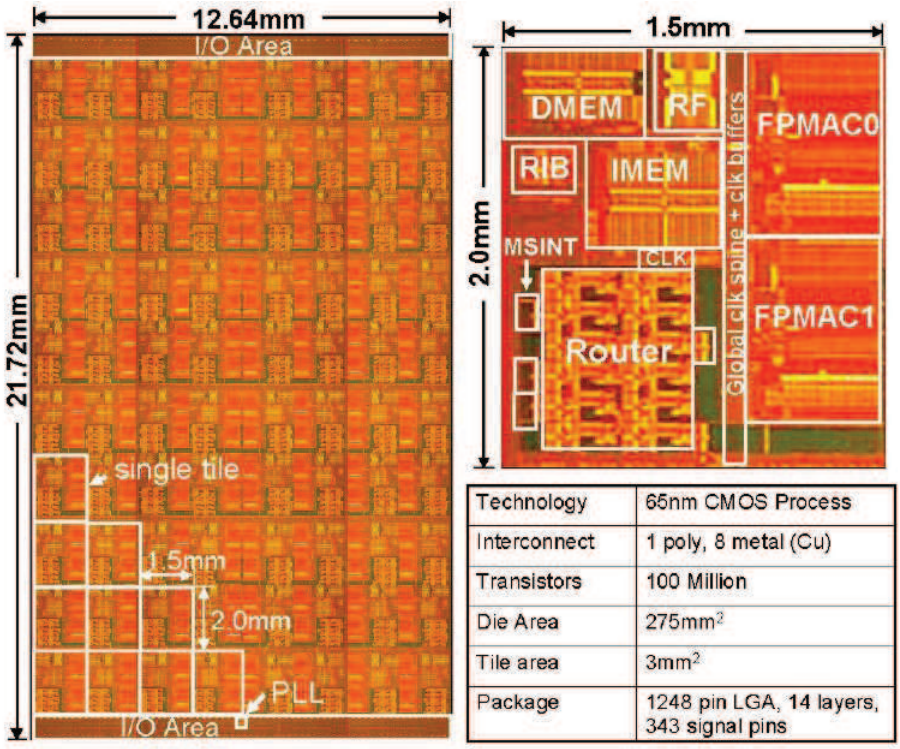

Figure 6.3: *From [44], micrograph view of the whole chip (on the left) and of a single tile (on the right) of the Intel 80 tile.*

passing router characterized by 5 ports with mesochronous (same frequency but unknown phase) interfaces.

Each tile is composed by a *router interface block* (RIB), a *data memory* (DMEM), an *instruction memory* (IMEM), a *router* and two *floating point multiply-acumulator* units (FPMACs).

The interconnection network is completely electrical and characterized by a 256 GB/s bisection bandwidth. A router interface block provides packet encapsulation/decapsulation from/to the processing elements (PE). The 4 GHz 5-port router performs wormhole switching exploiting 2 logical lanes for dead-lock free routing and a fully non-blocking crossbar switch with a total bandwidth of 80 GB/s. The interconnection network operates at the same frequency of the processing elements. Each lane is composed by a 16 FLIT (*FLow control unIT*) queue, arbiter and flow control logic. The routing task is executed by the 5 stages pipelined router where 2 of them are used for a round-robin arbitration scheme: the first binds an input port to an output port and the second selects a pending FLIT from one of the two leanes. The clock is distributed with a global mesochronous clocking that allows tiles to communicate whatever the phase of the received signal. This

architecture takes the power consumption issue in serious account: each tile is composed by 21 sleep mode enabled parts and each core router can put independently each one of the 5 interfaces in sleep mode. Programming such architecture is not easy as explained in [32]. This architecture has proven the need for optical global interconnects: the power required to make the global interconnect work is about 28% of the total power budget.

6.1.2 Intel Hybrid 48 Cores

A first step towards the integration of optical communications in a NUMA architecture is through an hybrid electro-optical approach: the processing cores are divided in *local* and usually small groups called *tiles*, each one internally interconnected using metallic and therefore electronic interconnects. These *tiles* of closely placed cores are then interconnected among them with *global* optical interconnects [2].

The term hybrid electro-optical is also often used to indicate the presence of an electronic network topologically parallel to the optical one used to resolve links and interface contentions, setup end-to-end transparent paths or performing other utility tasks.

In [16] and [31] (2010), a 48 core architecture evolution of the previous 80 core is analyzed. Contrarily to the 80 core architecture characterized by a minimal instruction set and no operating system, the SSC processor this time featured an x86 instruction set and a linux operating system. The purpose of this proposal was to explore the programmability of a possible manycore architecture. The architecture is composed of 24 dual core tiles organized in a 6×4 2D mesh.

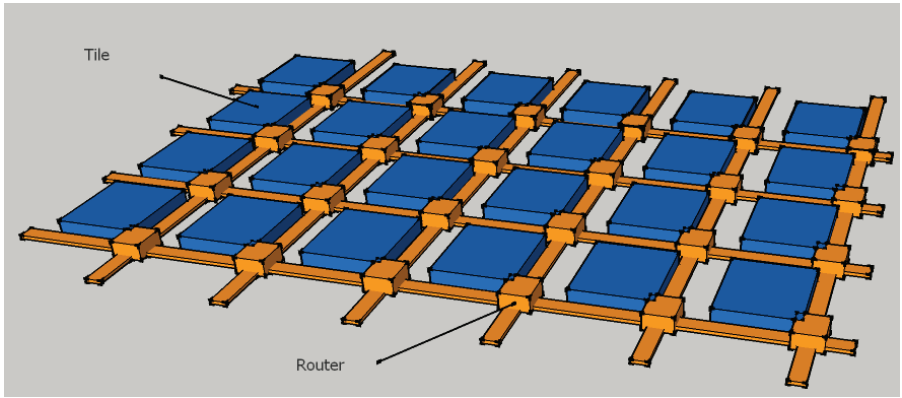

Figure 6.4: *From [31], 3D view of the 48 core processor.*

Each tile is composed by two cores, independent L1 instruction and data caches, a unified L2 cache, a *mesh interface unit* (MIU) which allows the interconnection network to work at lower frequencies than the tiles, and a 16

KB message passing buffer. Each tile is an off-the-shelf processor and is connected to the interconnection network through a router that directly communicates with the MIU for packetizing and encapsulating/decapsulating the messages to be transferred.

Chapter 7

Cost Model

7.1 Cost Model for On-Chip Optical Interconnection Networks

In this section we develop a cost model for the evaluation of the performance of the different interconnection structures analyzed later. Our novel cost model targets the on-chip optical interconnection networks not from a general purpose viewpoint but, rather, from a parallel processing viewpoint. The outcomes of such analysis will hold for all the optical interconnects dedicated to the communication between independent processing nodes. This goal will be achieved because we will start from the general theory of how to structure parallel computations through *structured parallel paradigms*. These paradigms have been proved optimal for different computational grains: from the case of distributed memory systems such as clusters, grids, but also shared memory multiprocessor and multi/manycore systems. The theory developed can theoretically be applied also to the extreme cases of embedded systems, transactional systems or whatever, given that they aim to run parallel applications. Furthermore, the wide acceptance of these paradigms is well established since the 1990 when M. Cole proposed them first [9]. The structured parallel paradigms operate on streams of input data elements also called tokens.

The approach we are going to follow is the one based on the *data flow* computational model. This model is, differently from the others, supported by a solid formal definition where a general application can be represented as a *dependency graph* [Figure 7.1] where the different processes are nodes and the dependences between them are directed arcs connecting two nodes.

We now extend the original model: each process in the composition (a node in the graph) is labeled with its correspondent *service time* (t_S) defined as the time required to completely process an input data element. The directed arcs are labeled with the *communication latency* (L_{comm}). The applications start from the execution of a single or multiple initial processes

and terminates, whatever path during the execution, with the execution of one ore more final processes. The time required to traverse the graph from just before the execution of the initial process until the completion of the last process is called *computation time* (t_{COMP}) and can be computed as the sum of the service times and the latencies respectively of the processes and of the communications along the longest path from the first to the last process.

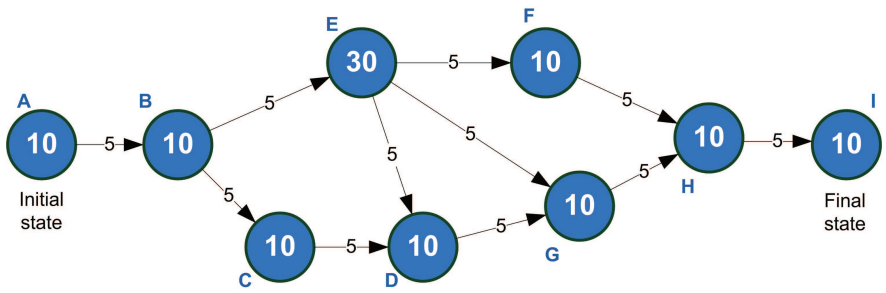

Figure 7.1: *A Dependency Graph.*

In the *first stage* of analysis of the graph, it is possible to operate optimizations in order to reduce bottlenecks and increase the parallelism of the execution of different tasks. This analysis and the following optimizations are independent with respect to the possible underlaying architectures. The outcome of this phase of transformation of the dependency graph is another semantically equivalent dependency graph [Figure 7.2] in which some of the previous processes have been splitted in several subprocesses following the structured parallel paradigms and eventually reducing the grain of their computations.

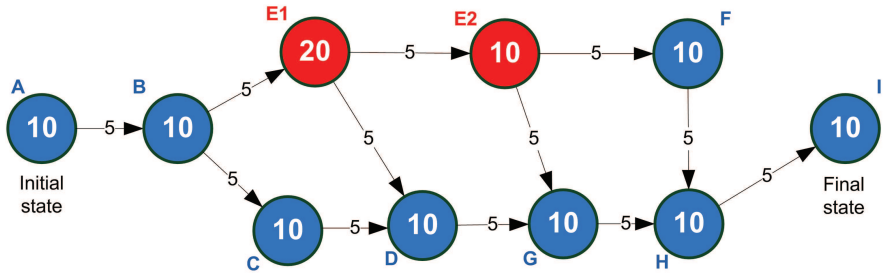

Figure 7.2: *The Optimized Dependency graph.*

In the *second stage*, once we find out an optimal solution to the parallelization problem, we have to find the best mapping of the processes on real processors. In principle, this task could be achieved assigning to each process a dedicated processor and implementing in this way a 1:1 mapping. Unfortunately, in general, the number of available processing nodes P is less

than the number of processes. We have also to consider that, while in the previous stage of analysis each process could communicate with each other through a dedicated logical channel whose latency was simply given by the size of the data elements exchanged, moving to a real physical dimension such as the on-chip one, we have to take into account the physical topology of the interconnection network, its routing policies, asynchrony degree, congestion and other relevant parameters. The mapping or scheduling of the processes on real cores on a chip becomes now a complex problem. This stage is obviously architecture dependent and is usually done at compilation time following a simplified and parametric cost model.

More formally, the *computation time* T_{comp} can be written as the sum of the *service times* of the processing nodes met in the path of a stream data element from the first node until the last one (i.e. until the end of the computation), plus the communication latencies between consecutives processing nodes in such path. We are interested to minimize the computation time, i.e. to:

$$\min\{T_{comp}\} = \min\Big\{\sum_{i=1}^{N} T_{s_i} + \sum_{i=1}^{N-1} L_{comm_i}\Big\} \tag{7.1}$$

where N is the number of processing nodes that a stream input data element has to traverse before being produced in output by the application. When cores of a multi/manycore processor are equipped with communication coprocessors, i.e. special purpose integrated circuits devoted to the execution of the interprocess communicaion procedure, then there is the possibility to partially overlap the computation with the communication executed by a processing node. Since we are interested on how the design of different optical interconnection networks can affect the overall computation time, we can neglect the first summation term in 7.1 since it is, from a communication viewpoint, just a positive quantity summed to a quantity that must be overall minimized. We are then left with:

$$\min\Big\{\sum_{i=1}^{N-1} L_{comm_i}\Big\} \tag{7.2}$$

At this point, we can expand the term L_{comm} with several considerations. Regardless of the type of communcation implemented (circuit switched or packet switched), we consider the latency of a one step communication as the sum of a *setup time* (T_{setup}), a *transmission time* (T_{transm}) and a *release time* ($T_{release}$):

$$L_{comm} = T_{setup} + T_{transm} + T_{release} \tag{7.3}$$

The *setup time* T_{setup} is the sum of the time required to build the envelope for the message to be sent (T_{env}); the time to take the first routing

decision choosing the correct output interface in case of multiple outgoing interfaces (T_{route}); the time to configure the hardware local to the transmitting core (T_{conf}); the time to cross connect the intermediate switching nodes in a circuit switched operational mode (T_{xcon}) and the time to setup the I/O buffer of the receiving core (T_{rcv}):

$$T_{setup} = T_{env} + T_{route} + T_{conf} + T_{xcon} + T_{rcv} \tag{7.4}$$

We start our analysis assuming that these operations are executed sequentially and cannot be overlapped in time.

The *transmission time* T_{transm} is the time required to transmit a single stream data element. Each stream data element is generally a complex data structure consituted by a long string of bits. The transmission time can therefore be rewritten as the product of the size in bits of the stream data element by the time taken to transmit a single bit (T_{bit}):

$$T_{transm} = M\ T_{bit} \tag{7.5}$$

where M is the size in bits of a stream data element. Notice that T_{bit} is the inverse of the bandwidth of the communication link: $1/B$.

The *release time* is the time required, in case of dynamic establishment of communication links, to release the resources previously allocated for the communication in the transmitting core ($T_{src_{rls}}$), the intermediate switching nodes ($T_{switch_{rls}}$) and the receiving core ($T_{dest_{rls}}$):

$$T_{release} = T_{src_{rls}} + T_{switch_{rls}} + T_{dest_{rls}} \tag{7.6}$$

We can now rewrite 7.2 as:

$$\min \Big\{ \sum_{i=1}^{N-1} T_{setup} + T_{transm} + T_{release} \Big\} = \tag{7.7}$$

$$\min \Big\{ \sum_{i=1}^{N-1} (T_{env}+T_{route}+T_{conf}+T_{xcon}+T_{rcv})+(M\ T_{bit})+(T_{src_rls}+T_{switch_rls}+T_{dest_rls}) \Big\} \tag{7.8}$$

7.2 Stream Parallel Paradigms

Given this initial firmware level cost model, we now consider the structured parallel paradigm coming from the parallel computing theory. We analyze those that target stream based computations: *pipeline* and *farm*. This choice is not restrictive since single data element computations can be reduced to equivalent stream computations [46] and, however stream computations are the most interesting for large scale applications.

7.2.1 Pipeline

The *pipeline* paradigm [46] has a topology which is formed by a set of processing nodes

$$P = \left\{ p_i \right\}_{i=1}^{p} \tag{7.9}$$

each able to compute a different function $f_i(\ )$ on each input data item d_j and organanized in a chain fashion linked by a set of unidirectional links

$$E = \left\{ e_i = \{p_i, p_{i+1}\} \right\}_{i=1}^{p-1} \tag{7.10}$$

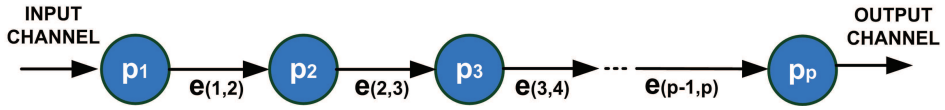

Figure 7.3: *The Topology of a Pipeline Pattern.*

In such a topology, a stream of input data items

$$D = \left\{ d_i \right\}_{j=1}^{d} \tag{7.11}$$

is provided in input to the pipeline from the input channel.

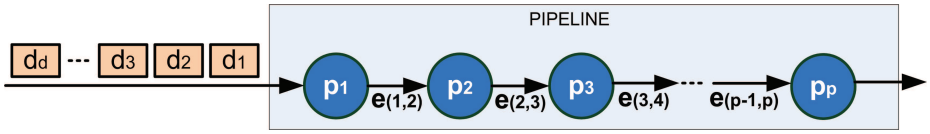

Figure 7.4: *Step 1 of the Pipeline Computation.*

The first element of the chain p_1 receives at a certain time instant a data item d_1 from the input stream. Then it performs the computation $f_1(d_1)$ using its memory and computational resources and sends to the output channel e_1 the data item d'_1. d'_1 becomes now the input for the processing

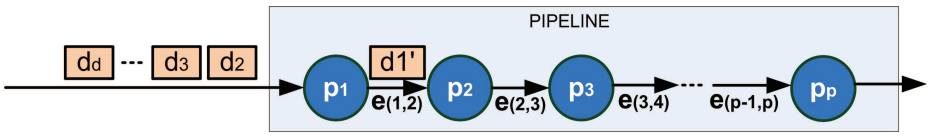

Figure 7.5: *Step 2 of the Pipeline Computation.*

node p_2 which performs its computation $f_2(d'_1)$ and sends again the result d''_1 to the output channel $e_{(1,2)}$ and so on until the last stage of the pipeline is reached and the output data items at its output are the final results of the

whole computation. When p_1 terminates the computation on d_1 and sends d'_1 to p_2, it becomes free and is ready to receive the next item d_2 in input and process it. This behavior is replicated along every stage of the pipeline.

This parallel pattern is based on the possibility of dividing a sequential computation into a sequence of temporally disjoint tasks which can be performed in parallel on different input data items.

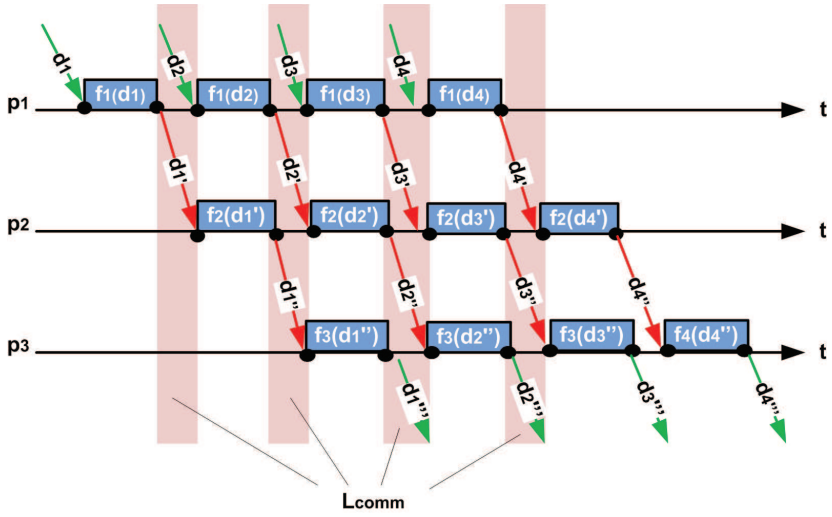

Figure 7.6: *Time Analysis of the Pipeline Parallel Pattern*

Exploiting this topology, it is possible to reduce the service time at the cost of increasing the latency.

7.2.2 Farm

The *farm* is the second stream parallel paradigm that we consider. It is based on the replication of a pure function [46] in a set of independent processes.

The topology of this structured paradigm is composed by a process called *emitter* (e) which is connected with w processes called *workers* (w_i) through w dedicated channels. Each worker is then connected through another dedicated channel to a special process called *collector* (c):

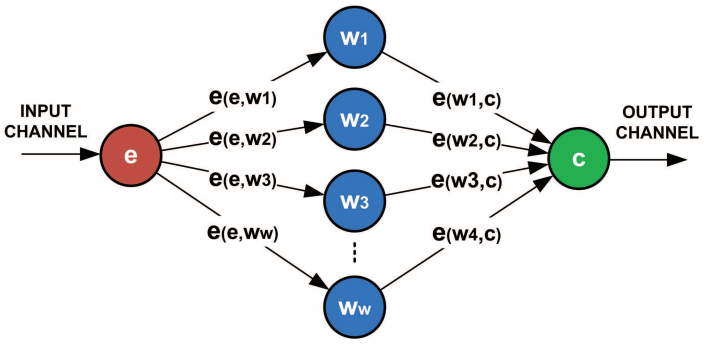

Figure 7.7: *The Topology of a Farm Pattern.*

The scheme however, in order to reduce the bandwidth requirement for the tasks computed by the *emitter* and the *collector*, can be changed implementing *emitter* and *collector* as trees:

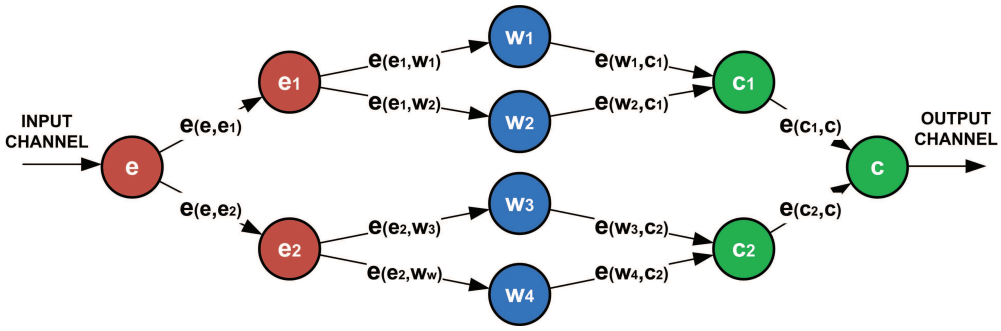

Figure 7.8: *The Topology of a Farm Pattern with Emitter and Collector Trees.*

or rings:

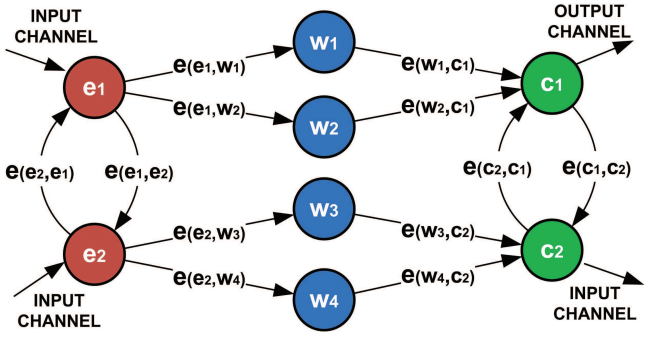

Figure 7.9: *The Topology of a Farm Pattern with Emitter and Collector Rings.*

Similarly to a pipeline, a stream of input data items

$$D = \left\{ d_i \right\}_{j=1}^{d} \tag{7.12}$$

is provided in input to the farm from the input channel.

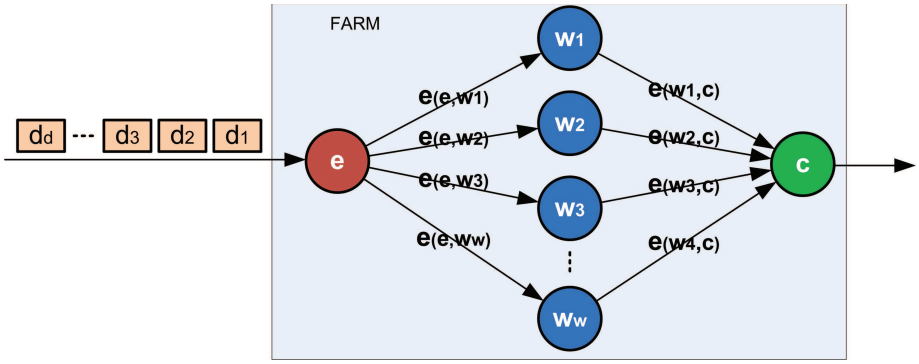

Figure 7.10: *Step 1 of the Farm Computation.*

The *emitter* process e receives a stream on input data elements $d_1, d_2, d_3, ...d_d$ and assigns each of them to a *worker* w_i according to a certain scheduling algorithm like, for instance, *round-robin* or, better, *on-demand*:

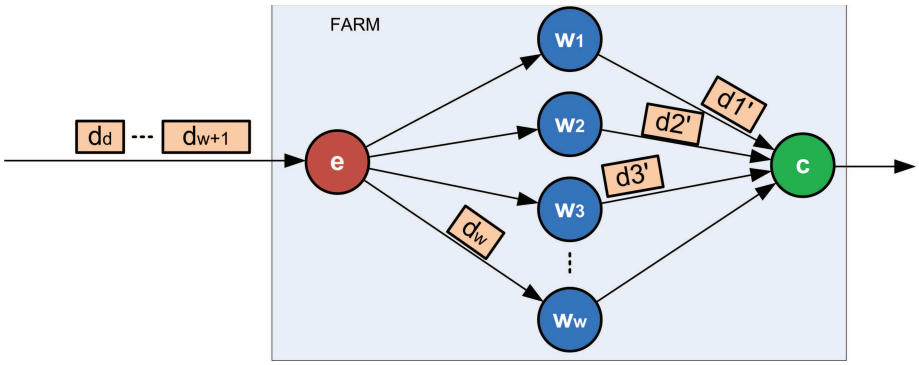

Figure 7.11: *Step 2 of the Farm Computation.*

Each *worker* process computes the function $f(\ )$ on each stream data item it receives and sends the result of its computation d'_j to the *collector* process c.

The main advantage of this structured parallel paradigm is the decrease of the service time distributing the input data elements to the various workers in order to achieve load balancing. We notice that the 3 main blocks *emitter*, *workers* and *collector* resemble a pipeline structure [46].

Part III

Indirect Networks

Chapter 8

Star

The *star* interconnection network topology is the first network topology we consider. This family of interconnection networks is characterized by heavy limitations and issues that we will address later but can be interesting under an optical perspective. In Section 8.1 we analyze the topological properties of the network. In Section 8.2 we propose a solution for clock distribution. In Section 8.3 we propose a solution for data communication between processing elements. In Section 8.4 we propose a design strategy for the switching node and discuss advantages and disadvantages of every design choice. In Sections 8.5 and 8.6 we discuss the effects of our design choices respectively on the pipeline and on the farm computational patterns.

8.1 Topology

The topology for such interconnection network is very simple: it is composed of a single switching node s_1 which is directly connected to each one of the n processing nodes through a bidirectional link. The only switching node has a degree equal to n and each processing node a degree equal to 1. There are a total of n links and therefore $2n$ interfaces. The switching node s_1 is the crosspoint of all the communications among the PEs. From this description it is clear the reason behind the choice of the name star. The number of processing elements that can be connected to the switching node can be incremented with step 1 except in particular cases in which we have extra constraints in the internal architecture of the switching node (we will not address these cases). On the other hand, the number of switching nodes always remains 1. The network diameter is constant and very low: 2. This is all what concerns the topology.

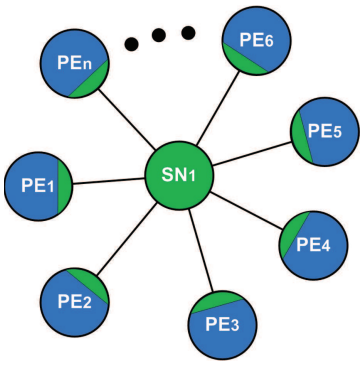

Figure 8.1: *Logical Structure of the Star Topology.*

Processing Nodes	n	$\bigcirc(n)$
Switching Nodes	1	$\bigcirc(1)$
SNs Scalability Coefficient	0	$\bigcirc(0)$
PEs Scalability Coefficient	1	$\bigcirc(1)$
Links	n	$\bigcirc(n)$
Processing Nodes Degree	1	1
Switching Node Degree	n	$\bigcirc(n)$
Interfaces	$2n$	$\bigcirc(n)$
Network Diameter	2	$\bigcirc(2)$

Table 8.1: *Summary of the Star Topology Properties.*

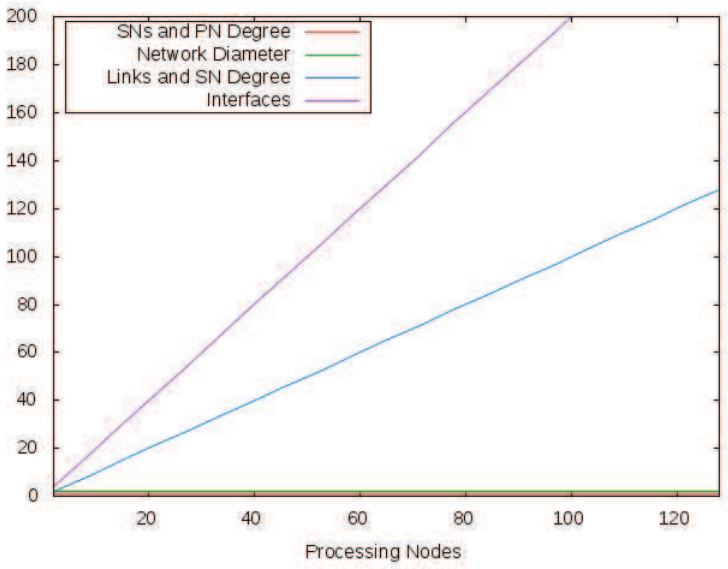

Figure 8.2: *Topology Statistics for the Optical Star Network.*

8.2 Clock Distribution

8.2.1 Proposal

The distribution of the optical clock signal could be implemented completely inside the single switching node s_1. In order to do not interfere with the optical data stream from/to each PE, we could think to distribute the clock with a dedicated wavelength λ_c different from all the others used for data transmission.

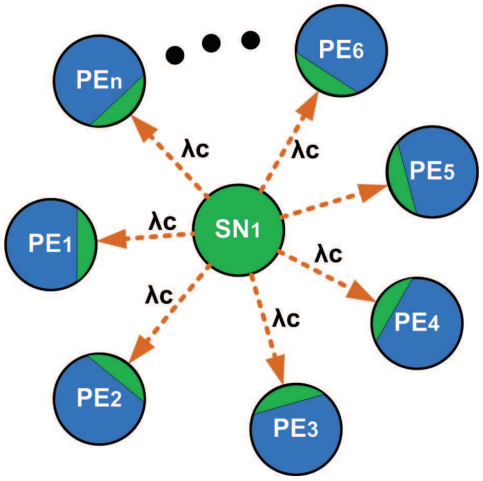

Figure 8.3: *A Possible Clock Distribution Strategy for a Star Topology.*

We remark that, with high probability, the greatest advantage for a star on-chip optical interconnection network is the fact that, generating the optical clock signal inside the switching node and transmitting it out of all the active transceivers (we could also think to turn off some or all the PEs if they are not busy with a computation), the clock skew could be greately reduced and the quality of the signal would be high and almost the same for all the PEs. No intermediate nodes must be traversed and the attenuation of the clock signal is influenced only by the waveguide (about 0.05dB/cm). The same fact could be counted as an advantage for the transmission of data signals between a couple of processing nodes if we could neglect the attenuation internal to the switching node as we will see in the next section.

8.3 Data Communication

As stated earlier, each PE_i is connected to the switching node SN_1 through a bidirectional link.

8.3.1 Proposal

Such a link can be easilly implemented as a single waveguide w_i and realizing the upstream and downstream from and to the PE_i as two counter propagating beams. In order to avoid interference with the clock signal, we could use a different wavelength: λ_d.

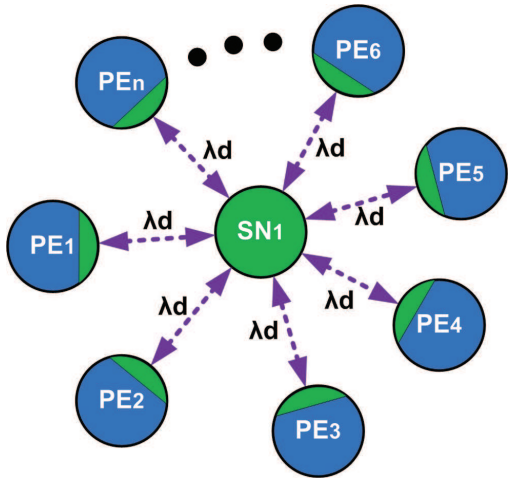

Figure 8.4: *Solution 1 to WA for the star network.*

This startegy requires the use of only 2 wavelengths (λ_c and λ_d) as a whole but implies a huge effort from the switching node SN_1 in order to avoid collision of messages, handling buffering and eventually message reordering.

Number of Wavelengths	2

The majority of these operation cannot be currently implemented in the optical domain and therefore opto-electric and electro-opto conversions would have to take place. This solution is really not scalable since as soon as the number of PEs increases, congestion becomes a bottleneck. Furthermore, if using always only two wavelengths can appear a scalable solution, it is somehow wasteful of the intrinsic bandwidth of the current generation waveguides. The structure of the PE transceiver would be composed of an optical circulator to separate upstream and downstream beams; a filter to separate the clock signal at λ_c and the data at λ_d; and a receiver and transmitter couple:

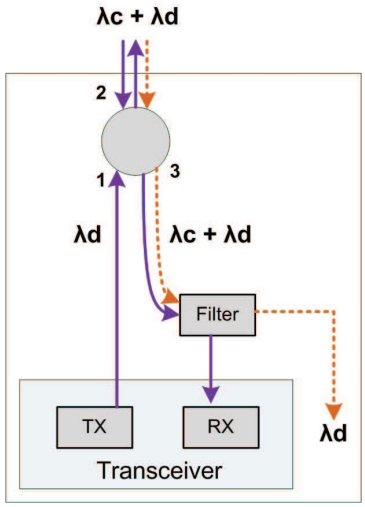

Figure 8.5: *Architecture of the PE Optical Interface.*

8.4 Structure of the Switching Nodes

In the case of the star network, the design of the switching node is really complex and is severely impacted by the number of attached processing nodes. The switching node is equipped with n interfaces.

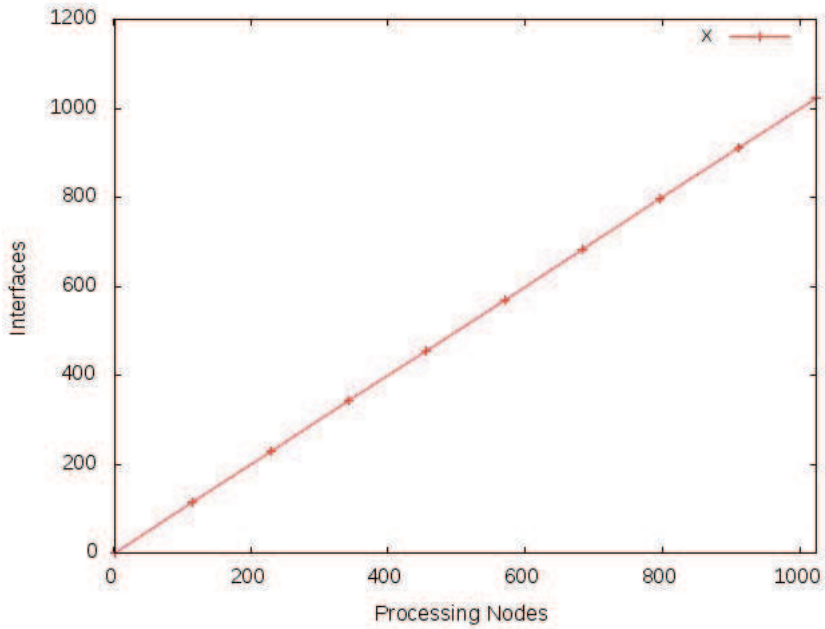

Figure 8.6: *Number of Interfaces needed per number of PE.*

It is clear that, starting from the issue of the pin count for the switching node, the star network has probably the worst scalability we could think of. Even the perfect clock distribution cannot compensate for the complexity of the design of the switching node and the latency of the data streams due to congestion and opto-electric conversions of the signal.

8.4.1 Proposal

The switching node could generate the optical clock signal within a clock generator module whose output could be splitted in power among all the interfaces and provided in input to a multiplexer internal to the every interface. In order to deal with interface transmission contention, every interface can receive electrical packets and store them in an electrical priority queue. From here, the packets can be serialized and forwarded to the optical transmitter that transmits the generated optical data stream to a multiplexer to join the data transmission at wavelength λ_d with the clock signal at wavelength λ_c. The joint signal can then reach the optical circulator that inserts it in the integrated waveguide. On the receive side, the optical input signal coming from a PE is passed directly to the receiver by the optical circulator. The implementation of the electrical data bus is not of our concern but is is clear that its implementation complexity represents a bottleneck for the switching node and for the whole network.

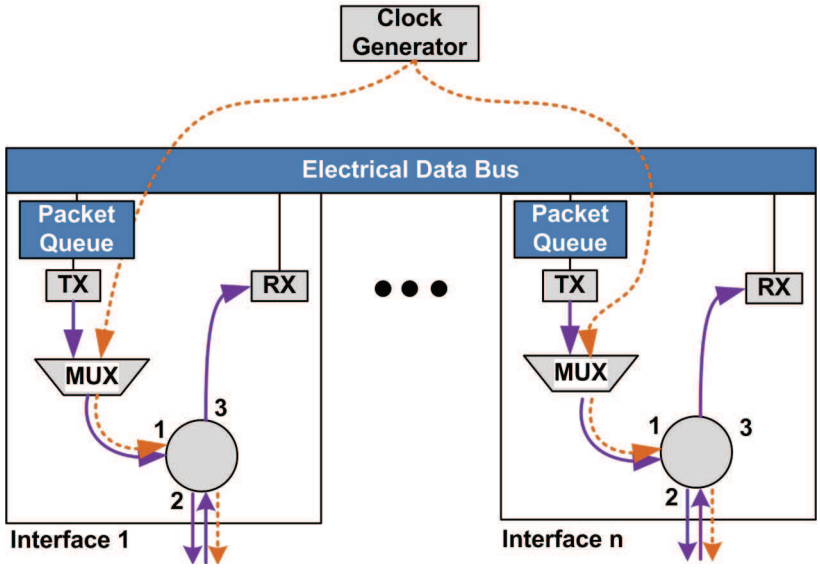

Figure 8.7: *A Possible Design Strategy for the Switching Nodes.*

8.5 Design Effects on the Pipeline Pattern

In this section we analyze the consequences of the design choices discussed until now on the communication latency experienced by a stream parallel application structured as an ideal pipeline. The ideal pipeline computation is composed by $m < N$ processes (4 in the example below) which are mapped to m consecutive processing nodes, i.e. cores, in our star network.

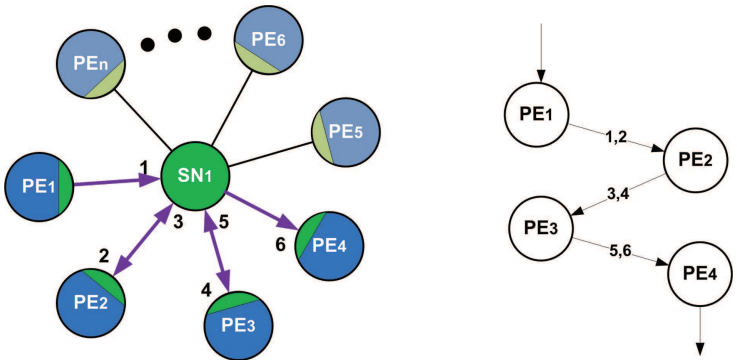

Figure 8.8: *Ideal mapping of the Pipeline Pattern on the Star Topology.*

The pipeline communication latency L_{comm} experienced by the parallel application would be equal to:

$$L_{comm} = \sum_{i=1}^{m-1} \Big(L_{comm}(PE_i, SN_1) + L_{SN_1} + L_{comm}(SN_1, PE_{i+1}) \Big) \quad (8.1)$$

where $L_{comm}(PE_i, SN_1)$ is the time required to transmit the packet from the PE to the SN; L_{SN_1} is the latency of the switching node and $L_{comm}(SN_1, PE_{i+1})$ is the time required to transmit the packet from the SN to the destination PE. The data packet is transmitted from the interface of every PE at a rate of B bps taking a time equal to:

$$T_{transm} = B \ \textbf{sizeof}(packet) \quad (8.2)$$

where B is the transmission bandwidth and **sizeof**$(packet)$ is the size of the packet in bits.

The optical packet arriving to the i-th interface of the SN_1 connected to the PE_i is converted to the electrical domain by the receiver and is forwarded to the electrical data bus which must deliver it to the packet queue correspondent to the destination interface. Once the packet exits from the packet queue, it is serialized and transmitted at the bandwidth of the optical subsystem. The latency of the switching node can be stated as:

$$L_{SN_1} = T_{OE} + T_{bus(i,i+1)} + T_{queue(i+1)} + T_{EO} \quad (8.3)$$

where T_{OE} is the time required to execute the opto-electric conversion of the packet; $T_{bus(i,i+1)}$ is the time spent by the packet moving from the i-th interface to the $i+1$-th interface traversing the electrical bus; $T_{queue(i+1)}$ is the time spent by the packet waiting in the $i+1$-th transmission queue and T_{EO} is the time required to perform the electro-optic conversion. T_{OE} and T_{EO} are negligible with respect to $T_{bus(i,i+1)}$ and $T_{queue(i+1)}$ so we can rewrite L_{SN_1} as:

$$L_{SN_1} \simeq T_{bus(i,i+1)} + T_{queue(i+1)} \tag{8.4}$$

The time taken to transmit the optical packet from the SN to the PE is equal to 8.2.

After this analysis we are ready to restate the communication latency of a 1:1 mapped ideal pipeline as a function of its interconnection network:

$$L_{comm} = \sum_{i=1}^{m-1} \Big(B\,\mathbf{sizeof}(packet) + (T_{bus(i,i+1)} + T_{queue(i+1)}) + B\,\mathbf{sizeof}(packet) \Big) \tag{8.5}$$

which can be rewritten as:

$$L_{comm} = \sum_{i=1}^{m-1} \Big(2\big(B\ \mathbf{sizeof}(packet)\big) + (T_{bus(i,i+1)} + T_{queue(i+1)}) \Big) \tag{8.6}$$

8.6 Design Effects on the Farm Pattern

After having considered the design effects on the performance of the pipeline pattern, we focus on the effects of the design on a parallel application with the structure of an indeal farm. As usual, we assume that each process composing the application is mapped 1:1 to a distinct processing node or core. Let k be the ariety of the farm, with one emitter process and one collector process. The application requires a total of $m = k + 2$ processes: Figure 8.9.

The farm communication latency L_{comm} is equal to:

$$\begin{aligned} L_{comm} = \quad & L_{PE_e,SN} + \\ & \mathbf{max}\Big\{ L_{SN,PE_i} + L_{PE_i,SN} \mid i \in [e+1, e+k] \Big\} + \\ & L_{SN,PE_c} \end{aligned} \tag{8.7}$$

where $L_{PE_e,SN}$ is the latency of the communication from the emitter core (PE_e) to the switching node; L_{SN,PE_i} is the latency of communcation from the switching node to the i-th core executing the i-th worker; $L_{PE_i,SN}$ the latency from the i-th core executing the i-th worker and the switching

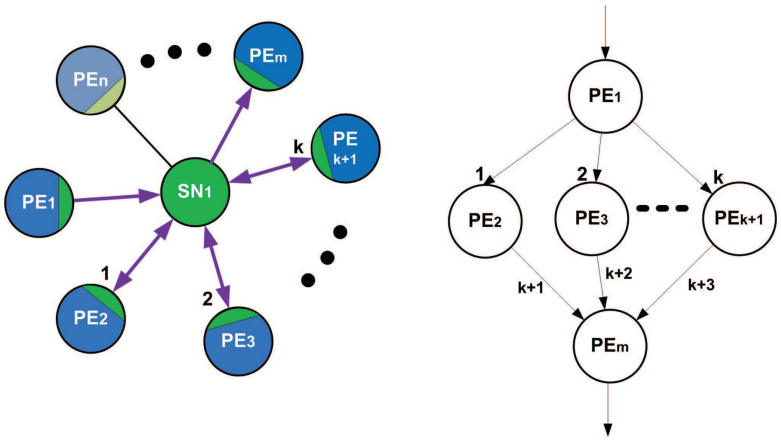

Figure 8.9: *Ideal mapping of the Farm Pattern on the Star Topology.*

node and L_{SN,PE_c} the latency from the switching node to the core executing the collector process. This cost model assumes that the time required by each worker process is identical since we want to isolate possible latencies caused by task unbalancing among the workers due to the nature of the data stream items that are received. We recall that the farm paradigm considers the replication among the workers of a pure function so no unbalance is directly presumable for the set of instructions that must be executed.

Keeping some of the considerations made in the previous chapter, we can restate the cost model at the firmware level as:

$$\begin{aligned} L_{comm} = \quad & B\ \textbf{sizeof}(packet) + \\ & \textbf{max}\Big\{T_{bus(e,i)} + T_{queue(i)+T_{bus(i,c)}+T_{queue(c)}} \mid i \in [e+1, e+k]\Big\} + \\ & B\ \textbf{sizeof}(packet) \end{aligned} \tag{8.8}$$

or better as:

$$\begin{aligned} L_{comm} = \quad & 2\big(B\ \textbf{sizeof}(packet)\big) + \\ & \textbf{max}\Big\{T_{bus(e,i)} + T_{queue(i)+T_{bus(i,c)}+T_{queue(c)}} \mid i \in [e+1, e+k]\Big\} \end{aligned} \tag{8.9}$$

In the formula above, $T_{bus(e,i)}$ is the time required by the electric bus of the switching node to move the electrical packet from the interface e to the interface i and, similarly, $T_{bus(i,c)}$ is the time required to move the packet from interface i to interface c. $T_{quque(i)}$ and $T_{queue(c)}$ are the queuing time respectively in the i-th interface queue and in the collector interface queue.

For the final cost model we have obtained we can make some considerations. The first consideration is that, depending on the implementation of the electrical bus, the value of $T_{bus(e,i)}$ changes as i changes. Further nodes

are usually reached later than closer ones. The same holds for $T_{bus(i,e)}$. Another aspect that must be considered is that $T_{queue(c)}$ increases certainly as the ariety k of the farm increases.

Chapter 9

Crossbar

The crossbar is an indirect network that allows for the communication of many computing nodes at the same time without contention [10]. The crossbar is commonly used to interconnect a set of cores with a set of memory modules, implementing the SMP UMA architectural paradigm. In Section 9.1 we study the topology of the crossbar network. In Section 9.2 we propose a scheme for the distribution of the clock signal to the processing nodes. In Section 9.3, an optimized and integrated strategy for global optical clock distribution in proposed. In Section 9.4 we propose a possible implementation strategy of the switching nodes. In Section 9.5 we discuss the possible design effects on the performance of pipeline and farm computational patterns. Finally, in Section 9.6 we consider several study cases reported in literature.

9.1 Topology

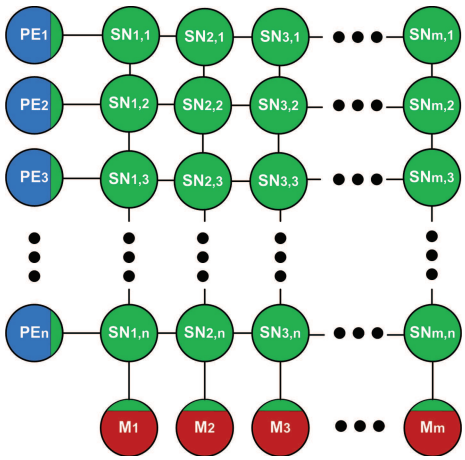

Figure 9.1: *Topology of a Crossbar Network.*

The network is composed by two disjoint sets of nodes: n processing nodes and m shared memory modules which are interconnected by an $n \times m$ network that is structured as a grid of switching nodes. Each computing node or memory module has a degree of 1 while $n + m - 2$ switching nodes have degree 3; 1 switching node has degree 2 and the remaining $(n-1)(m-1)$ switching nodes have degree equal to 4. The network diameter can be easilly calculated since, in the worst case, PE_1 will communicate with M_m, traversing first m switching nodes horizontally and then n switching nodes vertically.

Processing Nodes	n	$\bigcirc(n)$
Memory Nodes	m	$\bigcirc(m)$
Switching Nodes	nm	$\bigcirc(nm)$
SNs Scalability Coefficient	n or m	$\bigcirc(n)$ or $\bigcirc(m)$
PNs Scalability Coefficient	1	$\bigcirc(1)$
MNs Scalability Coefficient	1	$\bigcirc(1)$
Links	$2nm$	$\bigcirc(nm)$
Processing Nodes Degree	1	$\bigcirc(1)$
Memory Nodes Degree	1	$\bigcirc(1)$
Switching Node Degree	$\frac{3(n+m-2)+2-4(nm-n-m+1)}{nm}$	$\bigcirc(4)$
Interfaces	$4nm$	$\bigcirc(nm)$
Network Diameter	$n + m$	$\bigcirc(n + m)$

Table 9.1: *Summary of the Crossbar Topology Properties.*

The architectural scalability is very flexible: we can add from a minimum of one processing node with m new switching nodes or from a minimum of 1 memory module with n more switching nodes. There a total of $2nm$ links and $4nm$ interfaces. This interconnection structure is particularly suitable for interconnecting the set of processing nodes with the set of memory modules.

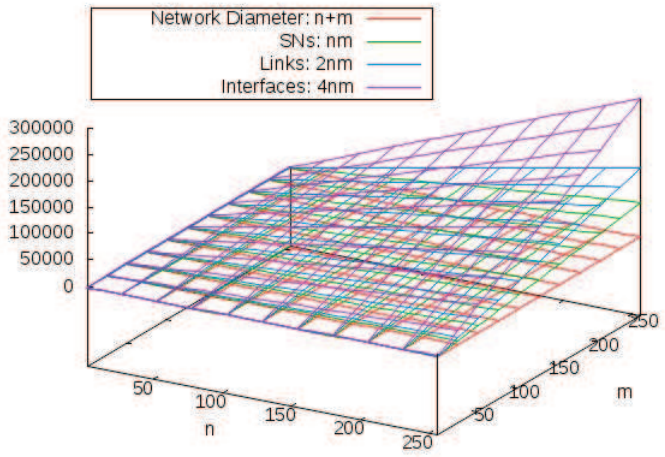

Figure 9.2: *Topology Statistics for the Optical Crossbar Network.*

9.2 Clock Distribution

In this section we consider all the possible strategies for the distribution of the clock signal to the processing elements. The clocking of the memory modules will not be addressed. We recall that, in order to reduce the clock skew, we must find out a way to make the signal reaching each PE traverse a path with an ideally identical length and attenuation. The considerations of the previous chapter regarding the use of a clock signal modulated on a wavelength different from the ones used for the data still holds. We decide then to exploit the wavelength λ_c for the distribution of the clock signal.

9.2.1 Proposal

In the first possible strategy we add a clock distribution firmware module to the network. The clock signal is generated (eventually converted to the electrical domain if no full optical generation is available) and tuned on the λ_c wavelength. At this point, a cascaded set of power splitters organized as a tree distributes the signal to the n switching nodes $SN_{m,i}$, $i \in [1, n]$. The ariety of this tree must be designed taking into account the precision with which the splitters have been realized and their cost so the ariety of the tree can vary depending on these parameters. The key idea is that, after the clock signal enters in each row of the interconnection network, it is forwarded along each row directly to the i-th processing node. The intermediate switching nodes should act as a transparent channel without detecting the clock and trying to limit the power loss of the clock signal as much as possible. The

optical interfaces of each processing node should finally filter the λ_c clock wavelength from the aggregate input wavelengths; convert the signal from the optical to the electrical domain and use it. This strategy is very powerful because after the clock signal leaves the clock generator firmware module, it passes through a light conduit, being splitted in terms of optical power and is received by each PE with an almost perfect synchro. As stated before, we must pay attention to the quality of the power splitters in order to guarantee a level of received optical power above a certain thresholds to all the processing nodes. The latency of the clock signal distribution is equal to the time required by the λ_c wavelength light signal to travel through the waveguide core material. A possible optimization could be obtained integrating the power splitters within the switching nodes. In this way the physical length of the lightpath experienced by the clock signal could be reduced at the cost of an higher design complexity of the switches and a possible difference in their implementation from switch to switch since only some of them would implement the function of power splitter.

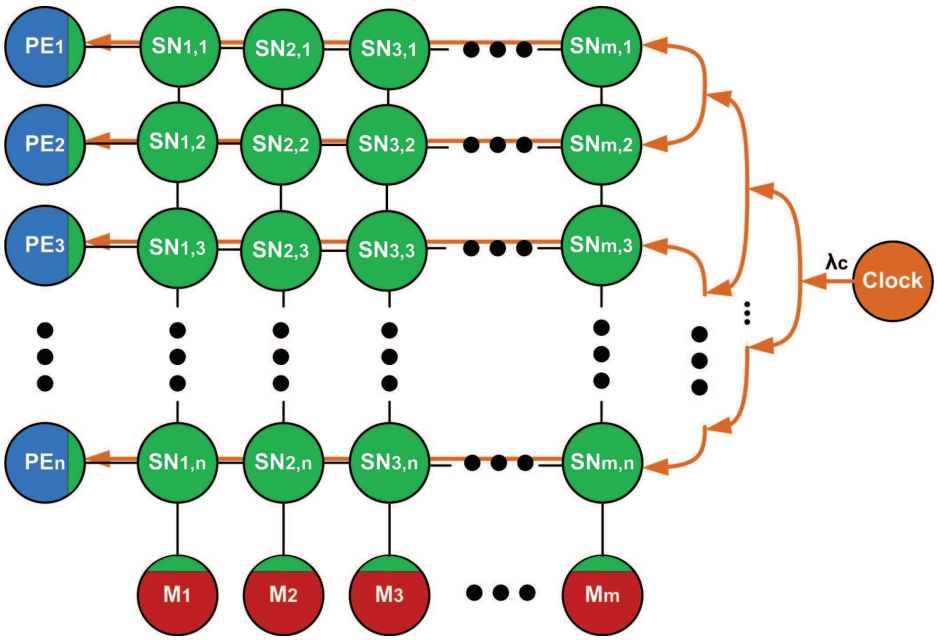

Figure 9.3: *A Possible Strategy for Clock Distribution in an On-Chip Optical Crossbar.*

9.3 Data Communication

The crossbar interconnection network is probably the network that reflects the SMP uniform memory access architectural paradigm the most. We have

a set of n processing nodes and a disjoint set of m memory modules. The two sets are linked by the intermediate interconnection network.

9.3.1 Proposal

The data communication among cores takes place writing and reading shared memory areas within memory modules. A first strategy could be the association of a different wavelength to each memory module: λ_1 to m_1, λ_2 to m_2 and so on up to λ_m to m_m. Of course, we need the condition $\lambda_c \neq \lambda_i$, $i \in [1, m]$ to be satisfied in order to avoid interference. Each PE should be able to do source routing shifting the wavelength of the transmission data signal to the one targeting the desired memory module. This could be achieved, for example, with a tunable laser. For each row of the crossbar, the i-th switching node should be able to deflect the i-th wavelength to the vertical downward output waveguide and let the signal at the other wavelengths pass through. The other way round communications, i.e. from a

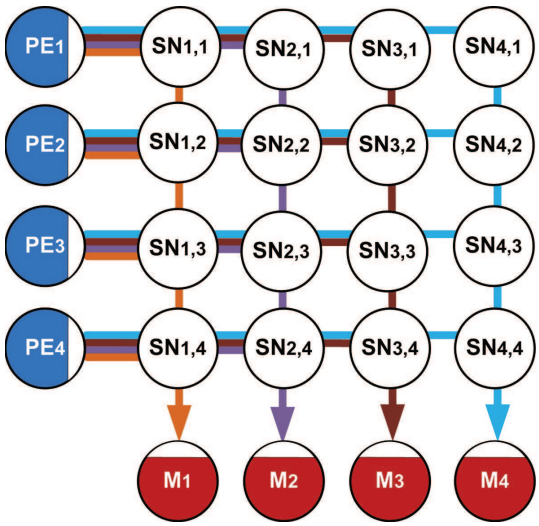

Figure 9.4: *PE-to-M Wavelength Assignment for Data Communication in an On-Chip Optical Crossbar.*

given memory module to a processing node, could be implemented similarly utilizing n wavelengths different than λ_c:

Notice that since two counterpropagating light beams at the same wavelength do not interfere, we can reuse the set of wavelengths used for the processing node to memory communication for the memory to processing node communication. As a whole we would require a total of $1 + \mathbf{max}(n, m)$ different wavelengths.

Number of Wavelengths	$1 + \mathbf{max}(n, m)$

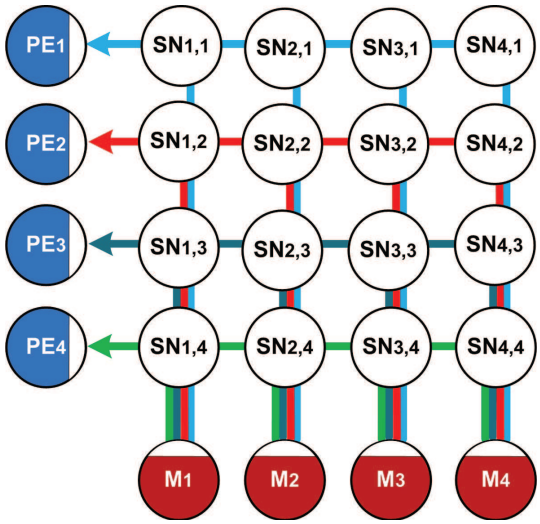

Figure 9.5: *M-to-PE Wavelength Assignment for Data Communication in an On-Chip Optical Crossbar.*

This fact can pose significat issues for the scalability of our solution: while the filters technology and the integrated waveguide's guided waveband are improving year by year as stated by the International Roadmap for Semiconductors, up to now we could think to use only about 64 wavelengths so we cannot scale the network at our will. This wavelength utilization can be seen as another extreem with respect to the solution proposed for the star topology in the previous chapter: the number of wavelengths required is now linear with respect to the number of nodes rather than constant as we can observe in Graph 9.6.

9.4 Structure of the Switching Nodes

We can design now the structure of the switching node. We start designing the structure of the SNs with degree 4 represented in Figure 9.7 since the others will be special cases with less functionalities. We consider the SN at the position (i, j) in the interconnection grid with $i \neq m$ and $j \neq 1$.

The beam provided in input to the west interface is composed by a maximum of m streams encoded in m different wavelengths. After entering in the port 1 of the optical circulator OC_1, the aggregate beam exits from port 2 and is provided in input to the demultiplexer DMX_1 which separates the i-th wavelength from all the others. This wavelength carries the data signal targeted to the $i-th$ memory module. The extracted wavelength beam is provided in input to the optical circulator OC_3 and forwarded first out of the port 2 of the optical circulator and then out of the south interface of the switching node. The other wavelengths different from λ_i provided as

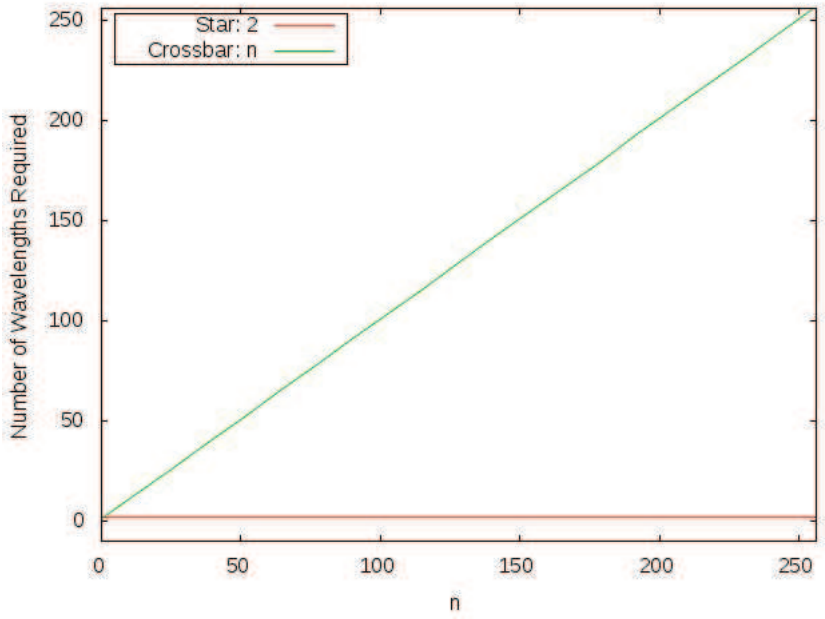

Figure 9.6: *Comparison of Wavelengths Utilization Between Star and Crossbar Networks.*

second output by DMX_1 traverse OC_2 from port 1 to port 2 and exit from the switching node through the east interface.

The optical beam provided in input to the south interface of the switching node is composed by a maximum of n streams encoded on n different wavelengths. The aggregate beam is provided in input to OC_3 through port 2 and exits from port 3 reaching the demultiplexer DMX_2. The filter separates the stream at λ_j and forwards it to OC_1 which directs it to the west interface of the switching node. The remaining aggregate of wavelengths which is provided in output by DMX_2 traverses OC_4 from port 3 to port 1 and is forwarded out of the north interface.

The input beam to the north interface is composed by the only wavelength λ_i and, after traversing OC_4 from port 1 to port 2 and OC_3 from port 1 to port 2, it is inserted into the waveguide which brings it out of the south interface. The signal in input to the north interface is summed to the one filtered by DMX_1 and therefore it is clear that they must be interleaved in time in order to avoid interference.

Finally, the beam provided in input to the east interface is composed by the clock signal imprinted on the wavelength λ_c and the signal eventually targeting PE_j already routed by a switching node connected at the east side of the switching node. The clock signal traverses first OC_2 from port 2 to port 3 and then OC_1 from port 3 to port 1 and is forwarded out of the west interface. The stream at λ_j behaves similarly traversing OC_2 from

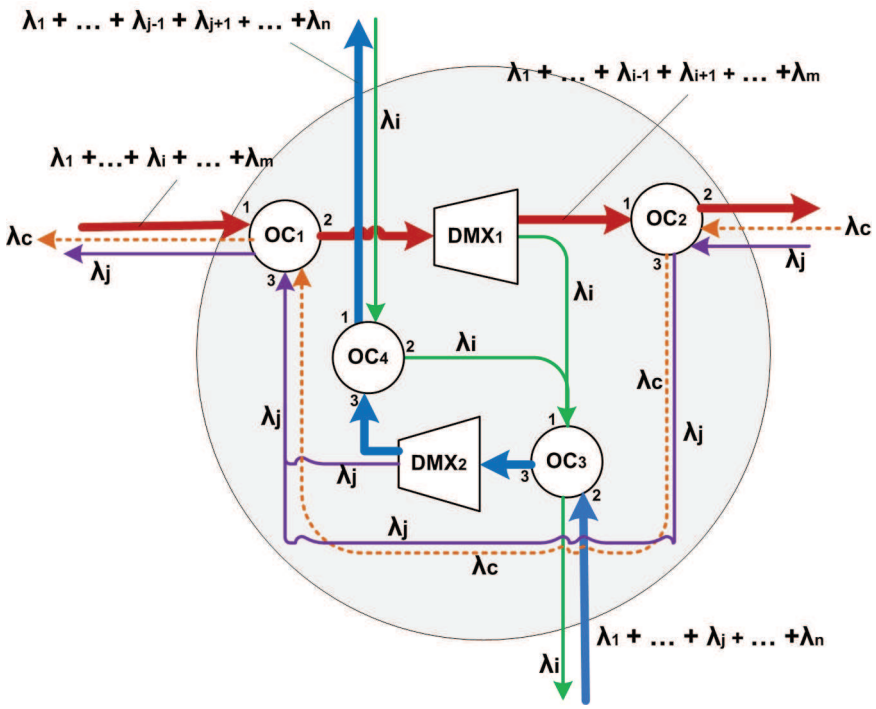

Figure 9.7: *Structure of a Switching Node $SN_{i,j}$ with $i \neq m$ and $j \neq 1$.*

port 2 to port 3, joining the λ_j signal filtered by DMX_2 and traversing OC_1 from port 3 to port 1 before being directed out of the west interface of the switching node. Again, we have that the stream at λ_j filtered by DMX_2 must be interleaved in time with respect to the one entering from the east interface in order to not interfere with each other. Notice that the number of waveguide crossings are present in the number of 2.

In Figure 9.8, we see the proposed structure for the switching nodes $SN_{i,1}$ with $i \neq m$. In the architecture of this second kind of switching nodes, the north interface is absent. The management of the aggregate beams in input to the west interface and to the east interface is exactly the same as for the fist kind of switching nodes discussed before. The beam coming from the south interface is, in this case, composed by only the λ_1 wavelength and, after traversing OC_3 from port 2 to port 3, it is summed to the signal coming from the east interface. After traversing OC_1 from port 3 to port 1, the signal is forwarded out of the west interface. In this implementation only one demultiplexer and 3 optical circulators are required.

In Figure 9.9, the design architecture of the third kind of switching node $SN_{m,j}$ with $j \neq 1$ is illustrated. In this case we have that the east interface is not present. While the lightpath and management of the beams coming from the north and south interfaces are exactly the same as in the first kind of switches, the beam present at the input of the west interface

is composed of only the λ_m wavelength and hence does not require to be demultiplexed. After traversing OC_1 from port 1 to port 2, the signal is coupled to a waveguide that is later joint with the waveguide guiding the same wavelength signal coming from the north interface.

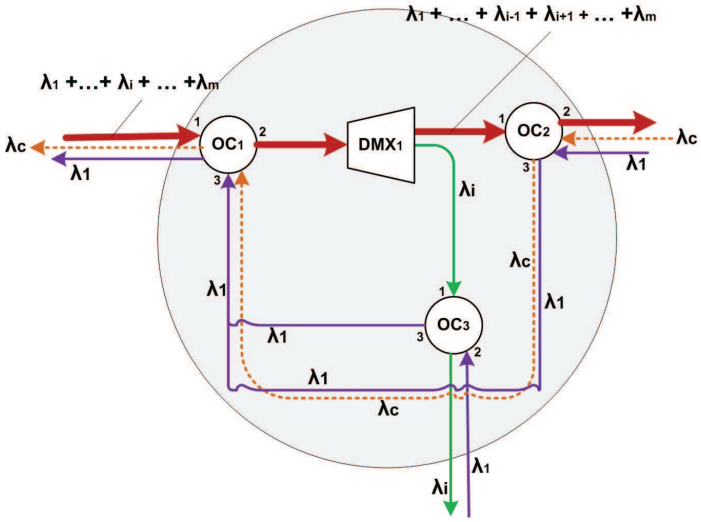

Figure 9.8: *Structure of a Switching Node $SN_{i,1}$ with $i \neq m$.*

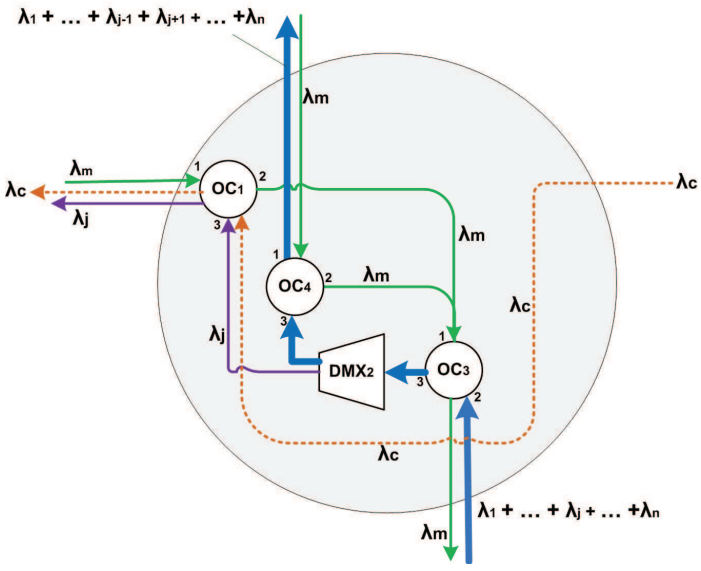

Figure 9.9: *Structure of a Switching Node $SN_{m,j}$ with $j \neq 1$.*

In Figure 9.10, we finally present the architecture of $SN_{1,m}$:

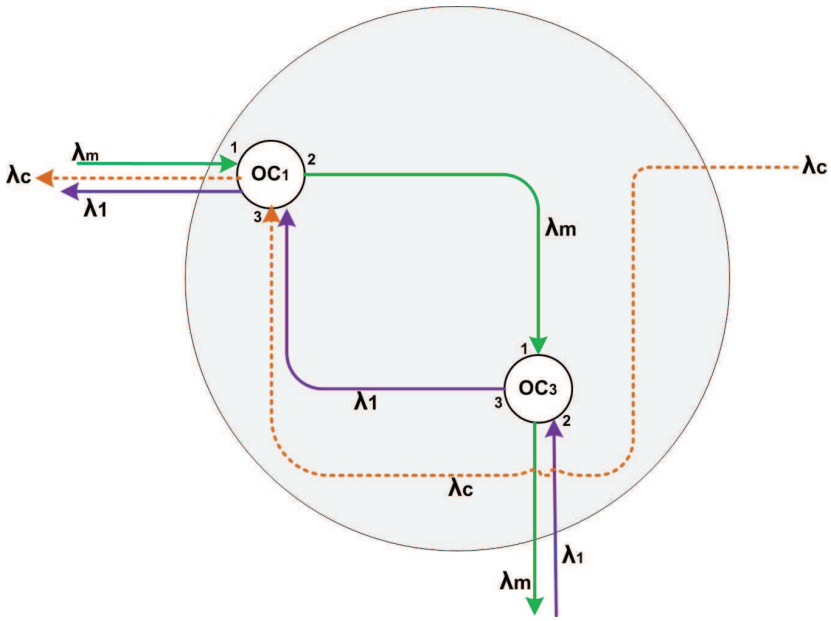

Figure 9.10: *Structure of a Switching Node* $SN_{i,1}$ *with* $i \neq m$.

No north nor east interfaces are present. The clock signal is directly guided to OC_1 through port 3 and exits form port 1 before being guided out of the west interface. The input beam of the west interface is composed by the only λ_m wavelength signal which, after traversing OC_1 from port 1 to port 2 and OC_3 from port 1 to port 2, leaves the switching node from the south interface. The symmetrical behavior is reserved to the signal carved on the wavelength λ_1 and arriving in input at the south interface before being redirected to the west interface.

9.5 Design Effects on Pipeline and Farm Patterns

Studying the effects of the various design strategies on the pipeline and farm patterns is very complex because we cannot make assumptions on the locations of the shared memory areas used for the communication. We can however notice, from the previous considerations on the path contention, that two processing elements cannot communicate with the same memory module at the same time without causing a disrupting interference on the signal. Symmetrically, two memory modules cannot communicate at the same time with the same processing element without provoking path contention and therefore signal interference. Since no valid optical buffering is currently available, such contention should be addressed at higher levels than hardware and could exploit the presence of a parallel electrical grid

interconnecting the various processing nodes, memory modules and switching nodes in order to synchronize the various components of the network. Beside this issue, we have obtained a crossbar network where, after a data stream has been encoded at a given wavelength λ_i, the optical signal propagates at the speed of light in the cludding material of the waveguide until it is detected by the destination target. Considering the average size of a chip and an average clock duration, we can result in a one clock cycle latency interconnection network. Of course, the effective length of the lightpath (and hence the duration of the transmission latency) connecting PE_n with M_1 could be considerably less than the one connecting PE_1 with M_m but they can be both traversed within the same deadline (the clock cycle duration). The bandwidth of the network would depend on too many parameters among which there is the signal to noise ratio of the received signal which depends on the power budget, the attenuation introduced in every traversed switching node and the sensitivity of the photodetectors used.

9.6 Study cases

The first proposal for an optical crossbar can be considered the one presented in [39] in 2004.

9.6.1 Case 1

In [18], two simple and effective implementation solutions for an on-chip optical crossbar are proposed.

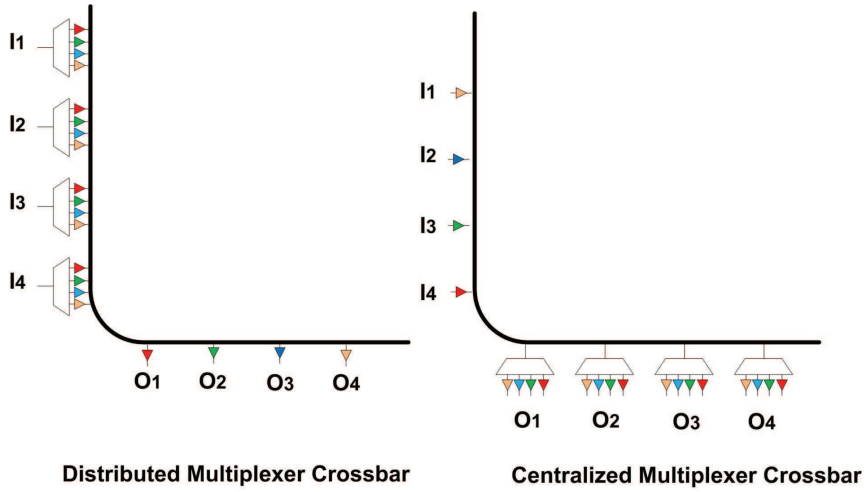

Figure 9.11: *Distributed and Centralized Crossbars [18].*

In the *distributed* version, each output port is associated with a dedicated wavelength and a receiver tuned on this wavelength translates the

received signal from the optical to the electrical domain. Each input port is equipped with an array of transmitters where each one is tuned on a wavelength among those associated to the output ports. Input port to output port communications are then arbitration free. If more input ports wish to communicate with the same output port then arbitration is required.

In the centralized version, each input port is characterized by a transmitter tuned on a different wavelength. On the other hand, every output port is composed by an array of receivers where every receiver is tuned on the wavelength of a different input port. In this way, all the input ports can simultaneously communicate with a single output port.

9.6.2 Case 2

In [19], a 4×4 fully passive wavelength-switched optical crossbar based on microring resonators add-and-drop filters is presented. The proposed interconnection structure allows for bidirectional communication exploiting add-drop filters between master (M) units and target (T) units. Each switching node is implemented as a waveguide intersection with two side by side coupled microring resonators:

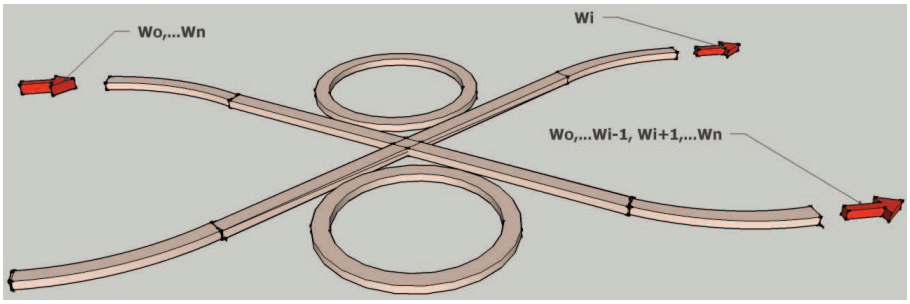

Figure 9.12: *Switching Node Implementation [19].*

and behaves in two different ways: the microring resonant wavelength W_i is left passing straight in the forward direction, while the other non-resonant frequencies $W_0, ...W_{i-1}, W_{i+1}, ...W_n$ are addressed to the diagonal output port:

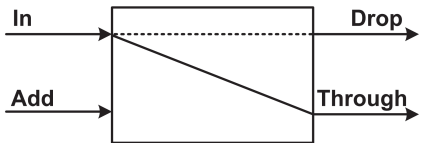

Figure 9.13: *External Behaviour of a Switching Node [19].*

The sample architecture (Figure 9.15) which is presented has cardinality 4×4 but can be expanded to 16×16 simply adding further switching

node rows. The switching nodes are characterized by a fixed resonant wavelength decided, once for all, at the design stage that is obtained enlarging or shrinking the radius of the microring resonators during the next fabrication process. Each master unit and target unit is equipped with a number of transceivers

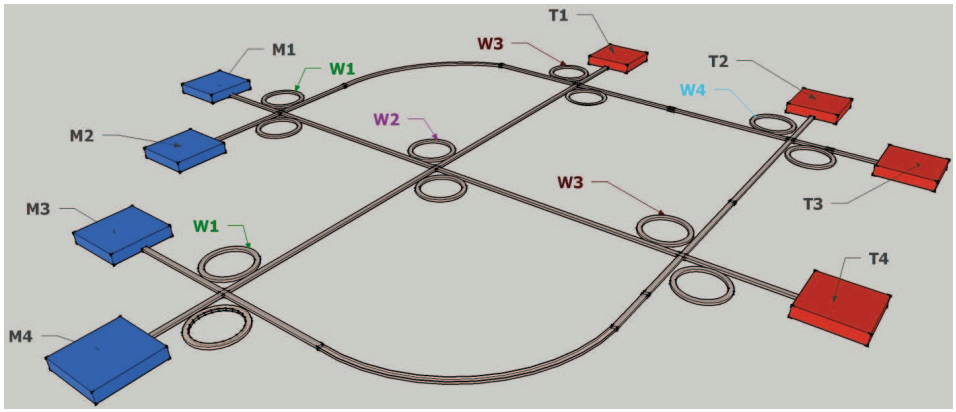

Figure 9.14: *Proposed Crossbar Architecture [19].*

This architecture proposed in 2009 has been fabricated and demonstrated on a SOI substrate with the deep UV lithography and has an area of $50\mu m \times 50\mu m$ which is the smallest reported until now.

9.6.3 Case 3

In [3], a novel on-chip optical crossbar network interconnecting a 256 manycore architecture with 16 DRAM memory modules is presented. Such a monolithically integrated optical network exploits DWDM (*Dense Wavelength Division Multiplexing*) to provide area and power efficient communication between the manycore architecture and the memory modules.

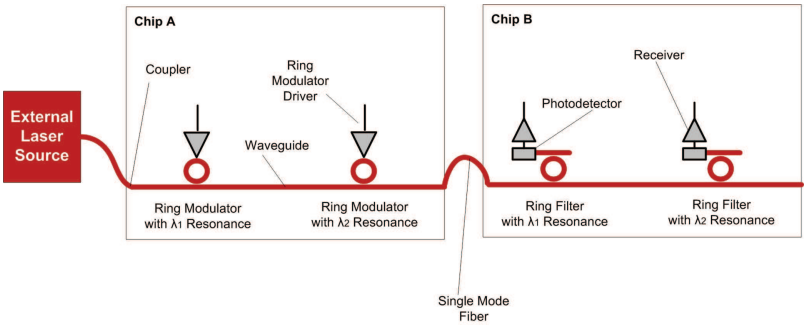

Figure 9.15: *Photonic Link with two Point-to-point Channels Implemented with Wavelength Division Multiplexing [3].*

Experimental results are illustrated for optical integrated components built with the $65nm$ technology. The proposed optical crossbar allows to reach performance improvements of about $8 - 10$ times with respect to optimized fully electrical networks.

Chapter 10

Tree

The tree topology can be implemented as an indirect network in a NUMA architecture where the leaves of the tree are processing elements (cores) and the other nodes of the topology are switching nodes. In this chapter we study this setting where the ariety of the switching nodes is k in order to be as general as possible in our analysis. In Section 10.1 we provide a summary of the topological properties of the tree network. In Section 10.2 we propose the integration of an optical H-tree for clock distribution within the network. Finally, in Section 10.3 we propose a simple solution for data communication and explain its main drawbacks.

10.1 Topology

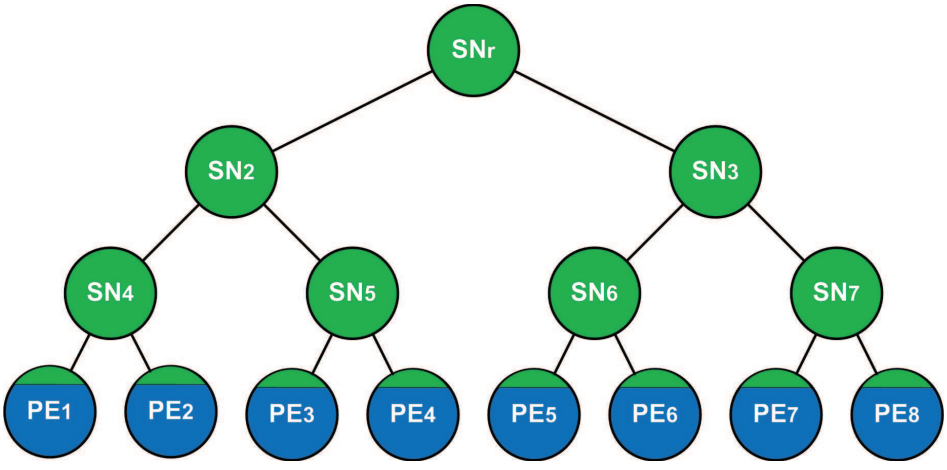

Figure 10.1: *Structure of a Tree Network with* $k = 2$, $l = 4$ *and* $n = 8$.

The topology of the network is composed by $n = k^l$ processing nodes and $n - 1$ switching nodes. From a special switching node called *root* which

is the only that occupies level $l = 1$, k other switching nodes are connected by dedicated bidirectional links. Then each of the other switching nodes ($l > 1$) is connected to its k *children* nodes by dedicated bidirectional links. The leaves of the tree which are the processing nodes are connected by a single bidirectional link to their *fathers*. The graph constructed in this way is acyclic. In order to realize a balanced network, the number of processing nodes n must be a power of the ariety.

Processing Nodes	k^{l-1}	$\bigcirc(k^l)$
Switching Nodes	$k^{l-1} - 1$	$\bigcirc(k^l)$
SNs Scalability Coefficient	$k^l - k^{l-1} - 1$	$\bigcirc(k^l)$
PNs Scalability Coefficient	$k^l - k^{l-1}$	$\bigcirc(k^l)$
Links	k^{l-1}	$\bigcirc(k^l)$
Processing Nodes Degree	1	$\bigcirc(1)$
Switching Node Degree	$k + 1$	$\bigcirc(k)$
Interfaces	$2k^{l-1}$	$\bigcirc(k^l)$
Network Diameter	$2l - 2$	$\bigcirc(2l)$

Table 10.1: *Summary of the Tree Properties.*

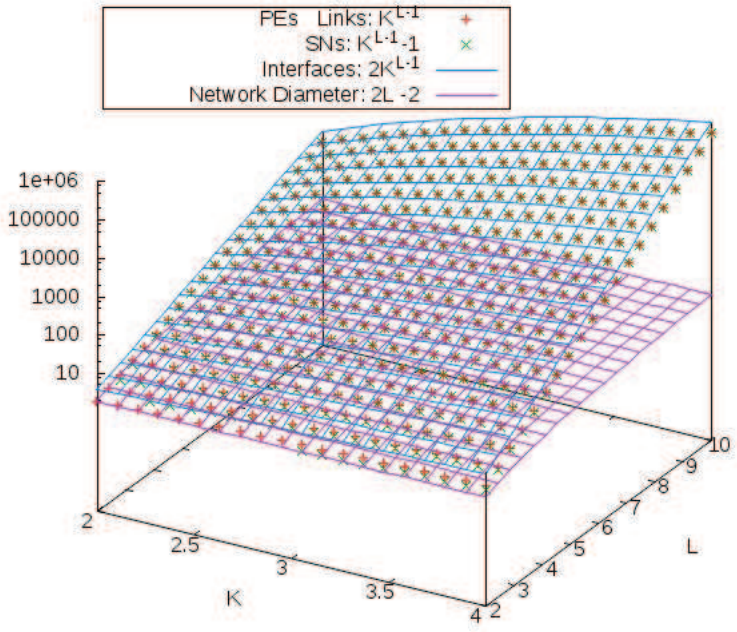

Figure 10.2: *Topology Statistics for the Optical Tree Network.*

10.2 Clock Distribution

10.2.1 Proposal 1

A straightforward approach to clock distribution, keeping in mind the constraints on clock skew, power budget and synchronization, could be the one illustrated in Figure 10.4 which exploits an optical H-tree [43]. An external firmware module could generate the clock signal and forward it to the root switch; the latest could then split the power of the signal and forward it out of the k interfaces. The same could be done by the remaining switching nodes. The processing elements would just have to detect the signal. The clock signal, in order to not interfere could be carved on a λ_c wavelength different from the ones used for data communication.

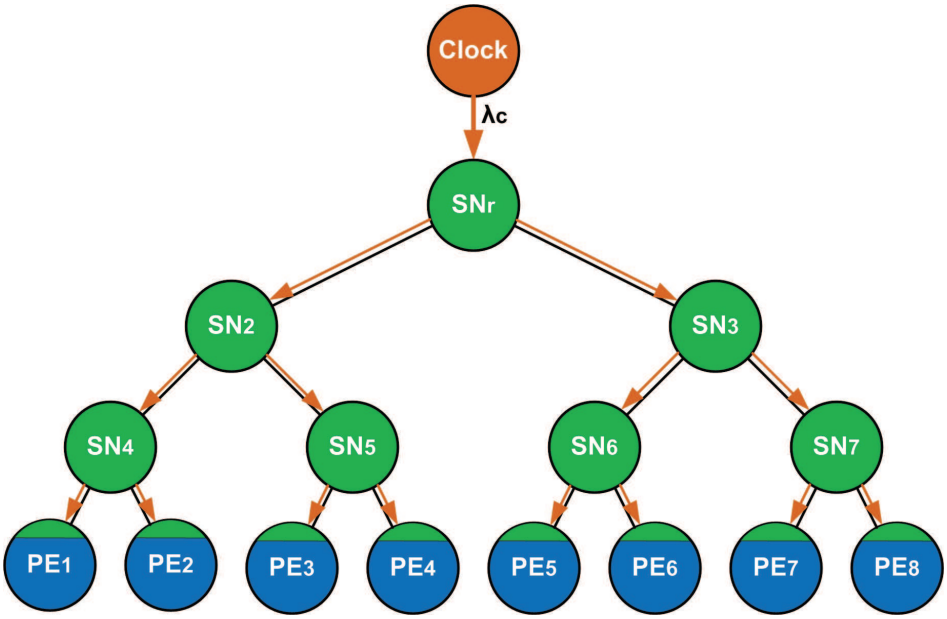

Figure 10.3: *A First Startegy for Clock Distribution in a Tree Network with* $k = 2$, $l = 4$ *and* $n = 8$.

10.2.2 Proposal 2

The splitting of the power of the clock signal could be implemented directly within the switching nodes or, whether this represents an unnecessary burden for the design of the architecture of the switches, in an external network as in the Figure below. Advantages and disadvantages of this second solution should be tested properly and also depend on the chip layout of the tree.

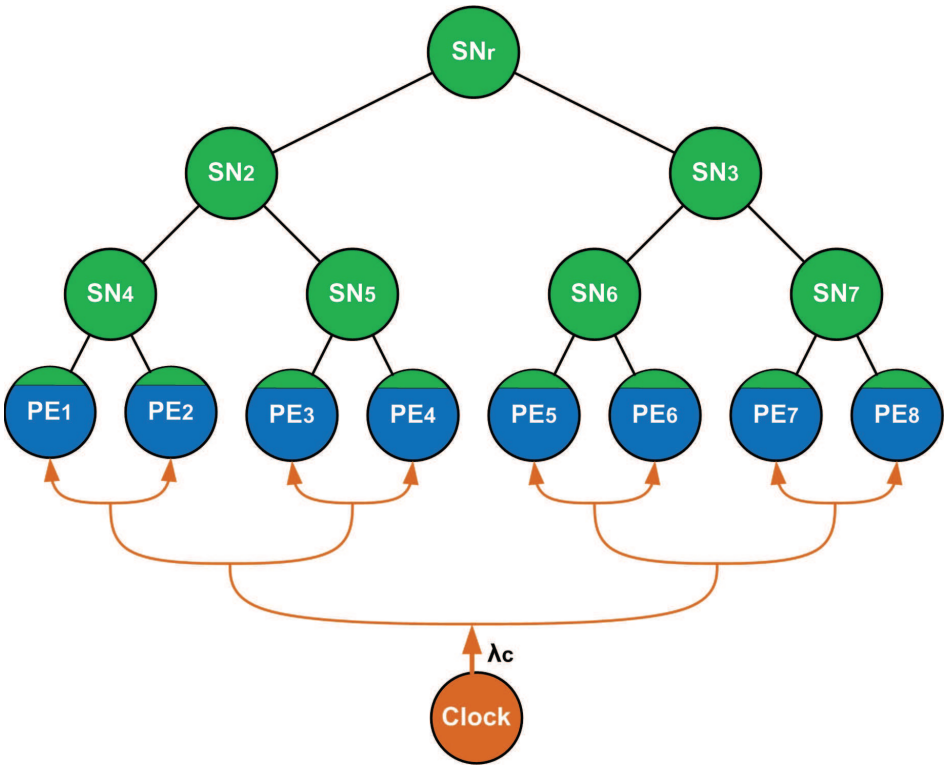

Figure 10.4: *A Second Strategy for Clock Distribution in a Tree Network with $K = 2$, $L = 4$ and $N = 8$.*

10.3 Data Communication

10.3.1 Routing Strategy

The routing algorithm is simple because there is only one path from a source s to a destination d. A direct consequence of this fact is that circuit switching is universlly adopted rather than packet switching in absence of path contention. A message that leaves the source processing element travels upward until a least common ancestor switching node with the destination processing node is encountered and then it is routed back down. In the worst case, a message must make $2 \log_k n$, i.e. $2(l-1)$ steps in order to reach its destination.

10.3.2 Proposal

A first simple strategy for data communication would be transmitting all the data streams on a wavelength λ_d different than λ_c used for the clock signal. However this solution would result in path and interface contention as in Figure 10.5. In order to solve the problem, since there is no currently

available optical buffering, we would be obliged to do first opto-electric conversion of the switching node input signal, then buffering for the time needed to wait for the availability of the output interface and finally do electro-optic conversion before forwarding the data stream out of the switching node. In this case, the most suitable routing strategy would be packet switching. This strategy would result in a very high latency and a consistent power consumption and therefore its benefits with respect to a fully electrical solution could be minimum if not totally absent. Due to these reasons, a simple tree network is not suitable for an optical implementation.

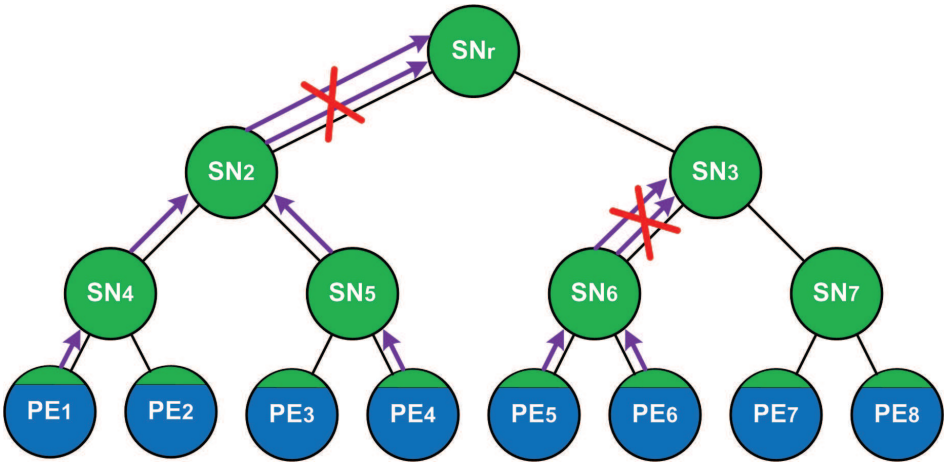

Figure 10.5: *Data Communication Interference in a Tree Network with* $k = 2$, $l = 4$ *and* $n = 8$.

In the next chapter we analyze the most known variant to the tree network which solves the problem of path contention.

Chapter 11

Fat Tree

A fat tree (FT) is a routing network based on a complete binary tree which has been introduced by C. E. Leiserson in 1985 [25]. In Section 11.1 we discuss the topology of the fat-tree network. In Section 11.2 we analyze the problem of clock distribution. In Section 11.3 we propose a simple strategy for data communication. In Section 11.4 we discuss a possible implementation of the switches for our proposal. In Section 11.5 and 11.6 we discuss respectively the design effects on the performance of the pipeline and the farm computational patterns. Finally, in Section 11.7 we consider some study cases extracted from the literature.

11.1 Topology

As in a common tree, a set of N processing elements is placed at the leaves of the tree and they are interconnected by a set of $N-1$ switching nodes.

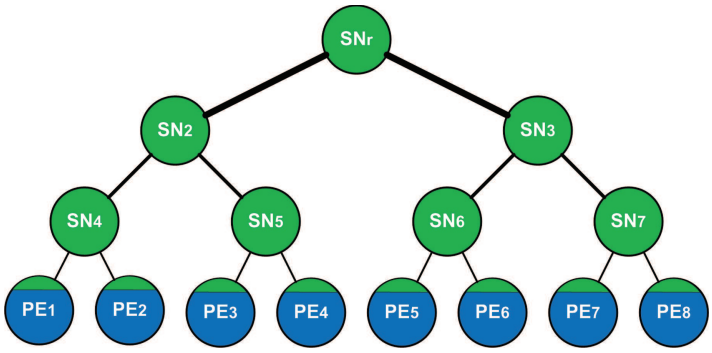

Figure 11.1: *The Fat Tree topology.*

Each link is characterized by a capacity c which, moving from the leaves up to the root, proportionally increases. This is the main difference with a common binary tree. In this newtork topology, the time required to deliver

a message from a source to a destination processor is $O(\log_2 n)$.

11.2 Clock Distribution

The clock distribution strategies that could be exploit do not differ at all form those that can be used for the tree network. An optical H-tree [43] can be adopted again as the best solution.

11.3 Data Communication

11.3.1 Routing Strategy

As for the tree network, a message travels upward until a least common ancestor switching node is encountered and then it is routed back down according to the least significant bits in the address of node j. In the original electrical solution, a progressive address is assigned to each processing node from the left to the right (or viceversa). At each switching stage, the message can be routed on two output channels so that $\log_2 n$ bits are sufficient to address any destination even if this holds only for unicast communications which, in parallel computing, are not always the case. Due to the path uniqueness and an increased bandwith of the higher links that avoids interference, there is no reason to use packet (message) switching and therefore circuit switching is usually employed. In the first proposal by Leiserson, the network was studied considering hardware volume in an electrical implementation in which bits were transmitted in a sequence. The first bit was used to understand if a message was transmitted at a certain time. Then the following bits were representing first the destination address and then the payload bits.

11.3.2 Proposal

A full optical solution to the data communication problem in the fat tree network could be to assign to each processing node a dedicated wavelength λ_i with $\lambda_i \neq \lambda_c$ in order to avoid interference if the H-tree is implemented within the switching network (Figure 11.2).

This static circuit switched solution is characterized by the lowest latency that can be experienced by a stream: once the stream is carved on the correspondent destination wavelength, it is forwarded in a cut-through manner to its destination processing element. Notice that, in this way, we are implementing a kind of reverse fat tree since the bandwidth of links increases as the leaves nodes are approached. This proposed solution can be implemented completely with only optical components. However two or more processing nodes whishing to communicate at the same time with the same destination processing node, can still cause a signal interference

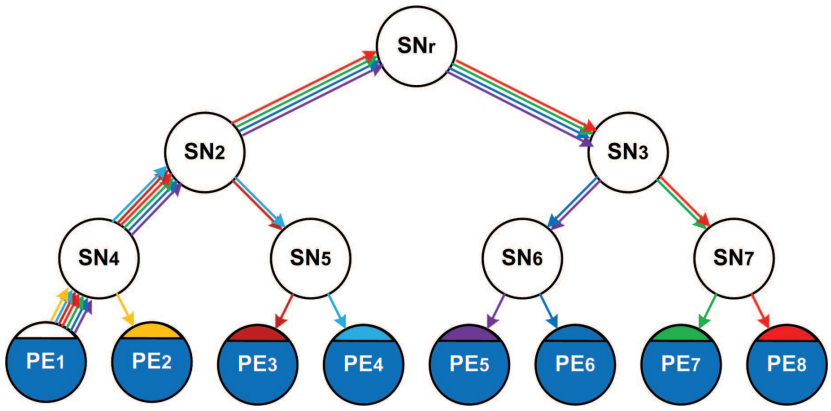

Figure 11.2: *Solution 1 to Data Communication Fat Tree topology.*

carving different data trains on the same wavelength and therefore rising the problem of path contention as happened in the previous chapter in the case of proposal 1 to data communication in a general tree network. This solution also poses serious scalability issues since the maximum number of distinct wavelengths that can be used is currently 64.

11.4 Structure of the Swiching Nodes

11.4.1 Proposal

In Figure 11.3 we present the proposed structure for an intermediate switching node. The idea is to assign adjacent wavelengths to processing elements connected to the same father switching node. Every couple of switching nodes having in common the same father SN can then be assigned adjacent wavelengths. Extending this wavelength allocation to higher level SNs, we will endup in a hierarchy of recursively nested wavebands. This property is the key idea to implement an optimized version of switching nodes where each switching node separates the traffic directed to the left children from the traffic directed to the right children simply performing a waveband binary splitting. The cost for the realization of the switching node can be then greately reduced. Furthermore, we could in this way relax the tight wavelength tuning for some of the filters. On the other hand, waveband filters have still to be investigated and their implementation advantages have not been demonstrated yet. In Figure 11.3, the waveband arriving in input from the north interface (WB_{ND}) is composed by a waveband that addresses children (i.e. south) nodes. This waveband aggregate signal traverses optical circulator OC_1 from port 1 to port 2 and is provided in input to the demultiplexer DMX_1 that separates the left children subwaveband WB_{LD} from the right children subwaveband WB_{RD} which are respectively forwarded out of

the left interface traversing OC_2 from port 1 to port 2 and out of the right interface after traversing OC_3 from port 1 to port 2.

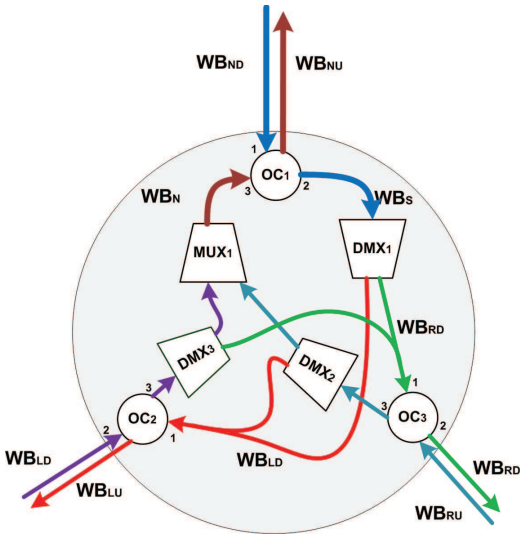

Figure 11.3: *Proposed Structure for an Intermediate Switching Node.*

The waveband provided in input to the right interface (WB_{RU}) first traverses OC_3 from port 2 to port 3 and then is provided in input to DMX_2. Here, the waveband component destined to the left branch (WB_{LD}) is separated to the one directed to higher ancestor switches and, after traversing OC_2 from port 1 to port 2, it exits from the left interface. The remaining band is multiplexed with the signal coming from the symmetrical left branch by MUX_1 and is then forwarded out of the north interface. The left interface input signal is handled symmetrically to the right interface's one. The root switching node shoud have a structure similar to the one depicted in Figure 11.4 which coincides to a simple waveguide connecting the two children switching nodes.

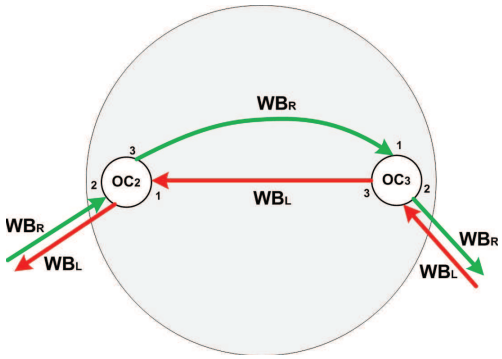

Figure 11.4: *Proposed Structure for the Root Switching Node.*

This solution suffers from an high number of waveguide crossings that could potentially impair the power budget of the switched signals.

11.5 Design Effects on the Pipeline Pattern

Given an ideal pipeline structured application with stages characterized by constant identical computation time, the overall latency of the application depends on the amount of latency in the communication between adjacent stages. In the fat tree network, the communication latency between any two processing elements (once the right to communicate with the destination PE has been acquired) is constant. This fact makes easier the scheduling of processes to processing elements for developers and run-time systems. In absence of other all-to-one collective communications, the performance should be very good.

11.6 Design Effects on the Farm Pattern

As in other proposals presented earlier in the thesis, assuming a 1:1 mapping of processes on processing elements, all-to-one communications like the one between the worker processes and the collector process represent an issue to to the need for arbitration of the contention of the destination interface and associated wavelength. In absence of valid optical buffering and processing, arbitration must be dealt with at higher levels or by a parallel electrical network.

11.7 Study Cases

11.7.1 Case 1

A proposal for an on-chip optical implementation has been done in [13]. This NoC does not require a separate electronic NoC control network since both payload data and network control data are moved through the same optical network.

The data are transmitted exploiting circuit switching while the control signals are propagated using packet switching. The network is based on OTAR (*Optical Turnaround Router*) router which performs the turnaround routing algorithm and is composed by crossing and parallel switching elements. The switching elements make use of optical microresonators fabricated on CMOS compatible SOI (*silicon-on-insulator*) substrates. Changing the control voltage of a microresonator it is possible to change the resonance frequency. If a microresonator is turned off, the resonance frequency is λ_{off} while when it is on the resonance frequency becomes λ_{on}. If λ_{on} matches the

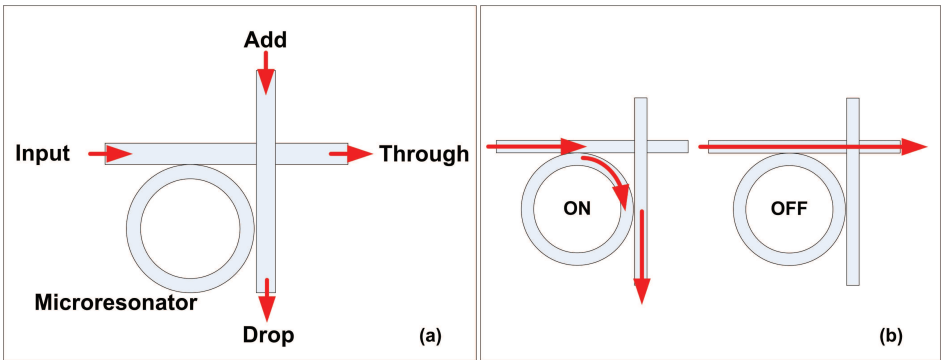

Figure 11.5: *(a) Microresonator strucure and (b) On/Off operation.*

wavelength of the incoming data signal, the light is trapped in the circular lane of the microresonator and is then transferred to the *drop* output port.

If, otherwise, the device is turned off and λ_{off} is different than the data signal wavelength, the signal passes through the straight waveguide and is forwarded out of the *through* port.

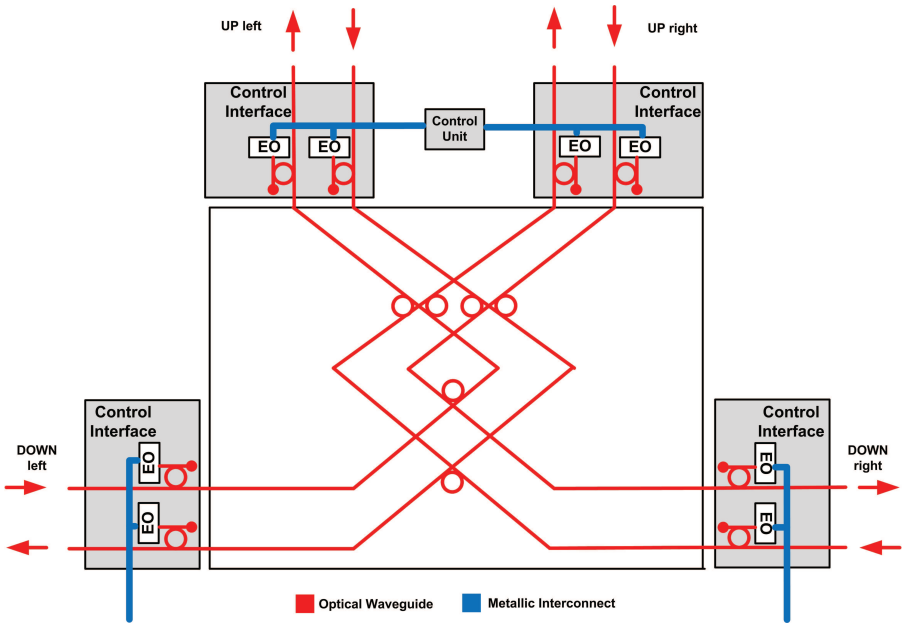

Figure 11.6: *Architecture proposed for the OTAR [13].*

Chapter 12

Clos Network

Clos networks are a kind of indirect and 3-stage networks proposed first by Charles Clos In 1953 for the first telephone switching systems. In Section 12.1 we study the Clos network topology. In Section 12.2 we study consider some issues related to the tradeoffs between non-blocking data communication and SNs pin count. Finally, in Section 12.3 we present an interesting study case extracted from the literature.

12.1 Topology

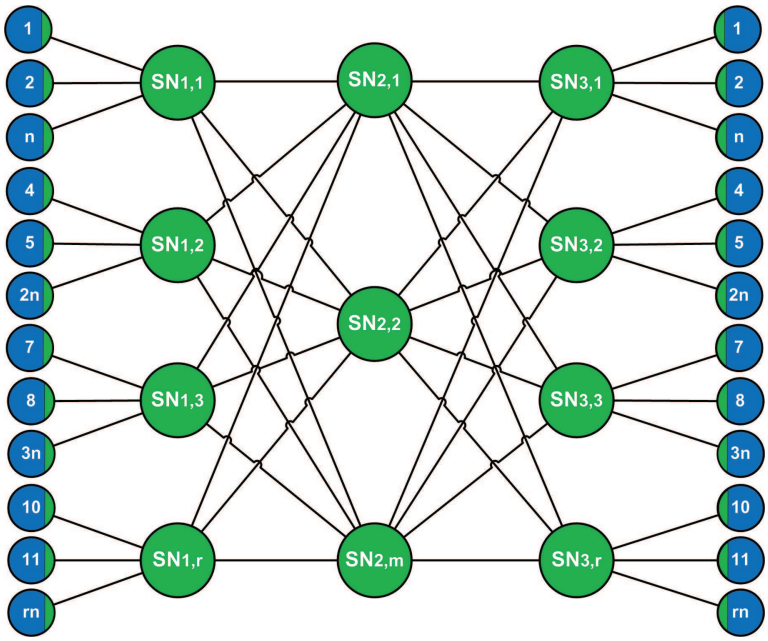

Figure 12.1: *A Sample* $(3, 3, 4)$ *Clos Network.*

The topology of a symmetric Clos network is characterized by a triple of parameters: (m, n, r). m is the number of switching nodes in the second stage (the middle one); n is the number of input (and output) ports of each input switch (first stage switch) and output switch (third stage switch). Finally, r is the number of input and output switches. The topology is composed by $2r+m$ switching nodes and can interconnect rn processing elements with each other (as shown in Figure 12.1) or rn processing elements with rn memory modules in the case of a uniform memory access architecture. Every switching node in the first stage is a $n \times m$ switch. Simmetrically, every third stage switching node is a $m \times n$ switch while the middle stage switching nodes are $n \times n$ switches.

Processing Nodes	rn	$\bigcirc(rn)$
Switching Nodes	$2r + m$	$\bigcirc(r + \frac{m}{2})$
PNs Scalability Coefficient	n	$\bigcirc(n)$
Links	$2(nr + mr)$	$\bigcirc(nr + mr)$
Processing Nodes Degree	1	$\bigcirc(1)$
Switching Node Degree	$\frac{2rn+4rm}{2r+m}$	
Interfaces	$4(nr + mr)$	$\bigcirc(nr + mr)$
Network Diameter	4	$\bigcirc(4)$

Table 12.1: *Summary of the Clos Network Properties.*

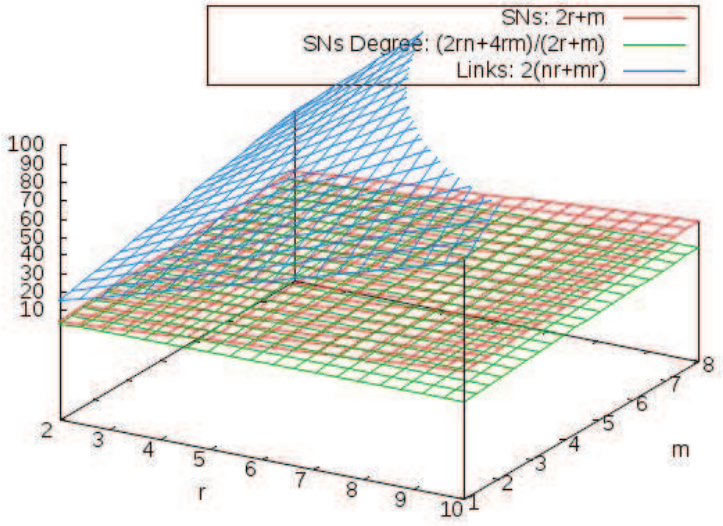

Figure 12.2: *Topology Statistics for an Optical Clos Network with* $n = 3$.

12.2 Data Communication

In the Clos network, every communication experiences a 4 hop process. In order to have strictly non-blocking unicast communications, the condition that needs to be satisfied is $m \geq 2n - 1$. Considering the on-chip scale in which we wish to implement this network, we have to take into account the number of interfaces that each switching node needs to implement. High values of n or m can result in a very high number of interfaces and then should be avoided. For implementing a 64 core network with $n = 2$ we would require only 3 intermediate switching nodes but they would require at least 64 unidirectional interfaces to be non-blocking for unicast traffic. Increasing the number of the interfaces, the electrical circuitry would start consuming oo much power. For this reason, a suitable way to scale this network would consist in implementing recursively the switching nodes as symmetric and asymmeric Clos networks.

12.3 Study Cases

12.3.1 Case 1

In [18], an on-chip optical Clos network is proposed and compared with other on-chip optical networks to demonstrate its advantages in terms of optical power, area and uniformity of latency and throughput. In Figure 12.3, a 2-ary 3-stage sample of the proposed Clos network is illustrated.

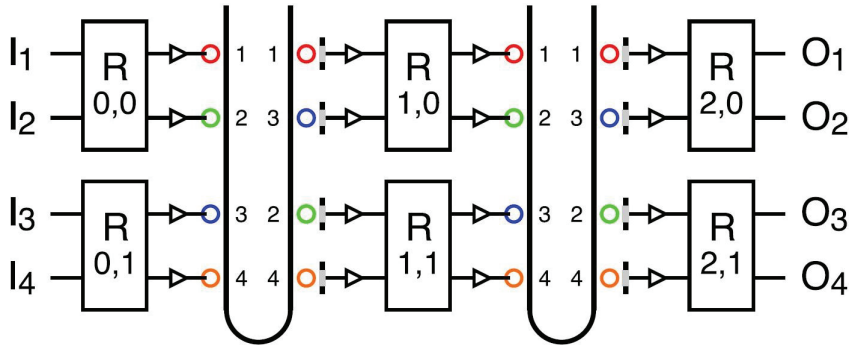

Figure 12.3: *2-ary 3-stages Optical Clos Network [18].*

While the routers are implemented electrically, the intermediate links are implemented with photonics. Each router is provided with two input ports tuned on two different wavelengths and two output ports tuned on different wavelengths as well. One pair of input and output ports are characterized by the same wavelength. The purpose of the electrical routers is to do buffering and arbitration of the output ports in order to simulate non-blocking

communications between input and output ports. An improvement to this solution is illustrated in Figure 12.4:

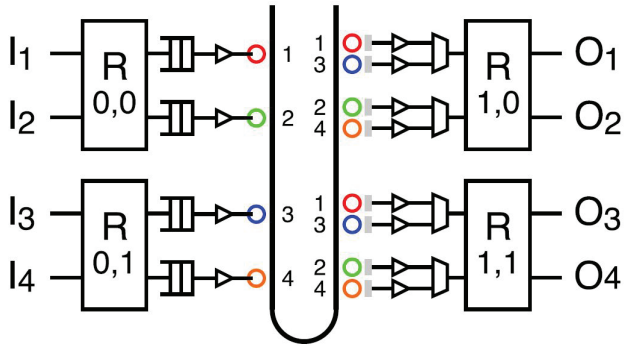

Figure 12.4: *2-ary 3-stages Optical Clos Network Optimization [18].*

In this optimized solution, the intermediate electrical routers are substituted by optical wavelength multiplexing, avoiding one stage of EOE conversion.

The analytical analysis has been done on an hypothetical 22 nm chip with 64 tiles operating at 5 GHz on a 400 mm^2 area. The results show that the considered optical Clos network has remarkable advantages in terms of lower area and thermal tuning cost, higher tolerance to photonic losses with respect to the optical crossbars proposed until now.

Part IV

Direct Networks

Chapter 13

Bus

The simplest architecture historically used for interconnecting CPU chip with main memory and with I/O units is the *bus*. This interconnection network is based on a shared medium which represents a bottleneck as the number of connected elements increases [10]. The strictly directional nature of light [41] does not allow the implementation of a bus based on a classical shared medium:

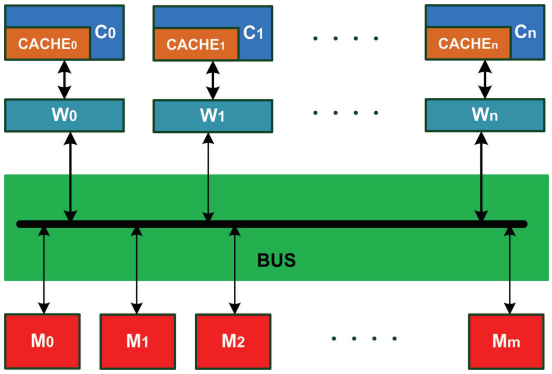

Figure 13.1: *Logical View of the Bus Network Topology*

In the literature, therefore, the proposals regarded bus interconnects based on ring structures or two separated on-way links. In Section 13.1 we analyze the topology of a bus network. In Section 13.2 we discusss how to integrate optical clock distribution in the network and in Section 13.3 we propose a solution for the data communication. In Section 13.4 we present the design for a node of our proposed optical bus ring while in Sections 13.5 and 13.6 we discuss the design effect respectively on the performance of the pipeline pattern and the farm pattern. Finally, in Section 13.7 we consider some study cases drawn from the literature.

13.1 Topology

The network is composed by n nodes connected with bidirectional links to a ring waveguide where optical packets circulate until they are received by some units. In Figure 13.7 a single waveguide optical ring bus is illustrated. Each node has a connectivity degree of 1. The network diameter, independently by the fact that the propagation direction of light is clockwise or counterclockwise, is always 1. The topological difference between an optical ring bus and a ring network is the fact that, as explained in later chapters, a ring is composed by a collection of bidirectional links which implement point to point communication with only the adjacent nodes. On the other hand, in an optical ring bus, each node is connected by a bidirectional link to a undirectional optical ring which is shared with all the other nodes. The intersections of the bidirectional link waveguides in the unidirectional shared optical ring(s) are implemented with Y-junctions. An optical ring bus implements explicitely the one-to-all communication paradigm.

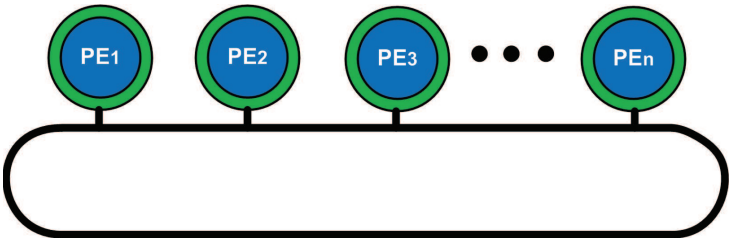

Figure 13.2: *Topology of a Single Waveguide Optical Bus.*

Processing Nodes	n	$\bigcirc(n)$
PEs Scalability Coefficient	1	$\bigcirc(1)$
Links	n	$\bigcirc(n)$
Processing Nodes Degree	1	1
Interfaces	n	$\bigcirc(n)$
Network Diameter	1	$\bigcirc(1)$

Table 13.1: *Summary of the Bus Topology Properties.*

In table 13.2, we have a summary of the topological properties for the optical ring bus just proposed. Notice that we can populate the topology with the granularity of a single processing element without influencing the design: the network diameter remains 1.

13.2 Clock Distribution

The optical clock distribution can be realized with an optical H-tree [43] embedded within the chip:

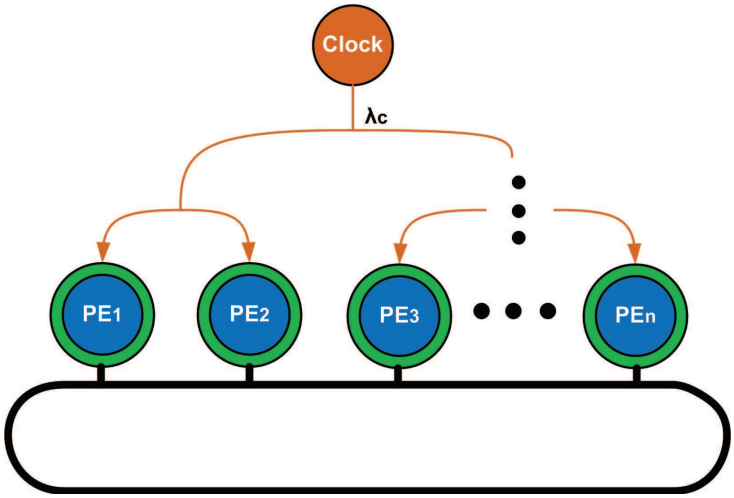

Figure 13.3: *H-tree [43] Clock Distribution for an Optical Bus.*

13.3 Data Communication

The bus interconnection network is one the cheapest network which can be realized but its performance definitely depend on the contention degree of its shared medium: the optical ring in this case.

13.3.1 Proposal

A first solution would be, as usual, assigning a different wavelength to each of the n processing elements. This solution would require a single ring waveguide and could scale up to a number of processing elements equal to the number of available wavelengths (64 with the technology at the time of this writing). Simultaneous all-to-one communications are still an issue and require arbitration. Arbitration can be obtained at higher levels; for example with autonomous dedicated firmware modules or better with another optical ring waveguide.

The proposed solution can exploit both packet-switching and circuit switching but, in the second case, we must pay attention to break long communications in order to avoid starvation of the other processing elements wishing to communicate with the same destination node. Neglecting arbitration, this design solution allows for the point to point communication between two processing elements with a constant latency independently on

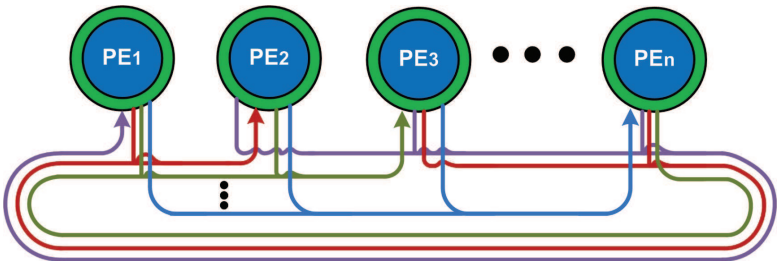

Figure 13.4: *Example of Wavelength Assignment and Data Streams Circulation.*

their location in the topology. Once a PE has acquired the right to communicate with the desired destination, a lightpath is established and therefore the total communication latency is equal to the time required to acquire the right to transmit, $T_{booking}$, plus the time to transmit the message: T_{transm}.

$$L_{comm} = T_{booking} + T_{comm} \tag{13.1}$$

This feature reveals very advantageous if we consider that some PEs could be executing a different computation: the latency of the interconnect as seen by a parallel application is not influenced by the presence of other parallel tasks executing on a disjoint set of PEs.

13.4 Structure of the nodes

The main advantage of the bus network reveals when we analyze the structure of its nodes: Figure 13.5. Only 3 components are required: a filter (demultiplexer); a receiver and a wavelength tunable transmitter (which can be exchanged with an array of transmitters or a tunable wavelength converter).

The demultiplexer is required in order to separate the i-th wavelength assigned to the node from the others and forward it to the receiver which converts the incoming optical signal to the electrical domain. Finally, the wavelength-tunable transmitter encodes the transmitted output optical beam on a wavelength that depends on the address of the destination node j.

Component	**Quantity**
Demultiplexers	n
Receivers	n
λ-Tunable Transmitters	n
Waveguides	1

Table 13.2: *List of Components Required in the Bus Ring Network.*

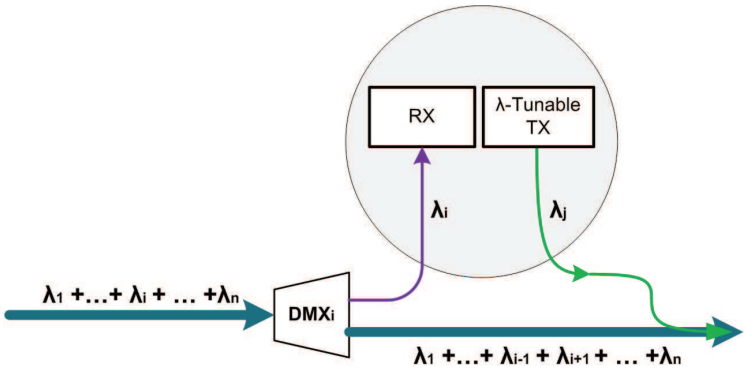

Figure 13.5: *Structure of the Node in the Proposed Optical Ring Bus Network.*

13.5 Design Effects on the Pipeline Pattern

In this section we analyze the effects of the previously discussed optical ring bus design starategies on the pipeline parallel pattern. Due to the process location independence of the interconnect latency, pipeline processes could be allocated everywhere in the network without any difference in the communication latency experienced by a data item. This property can ease the scheduling of applications reflecting this parallel paradigm on processing elements.

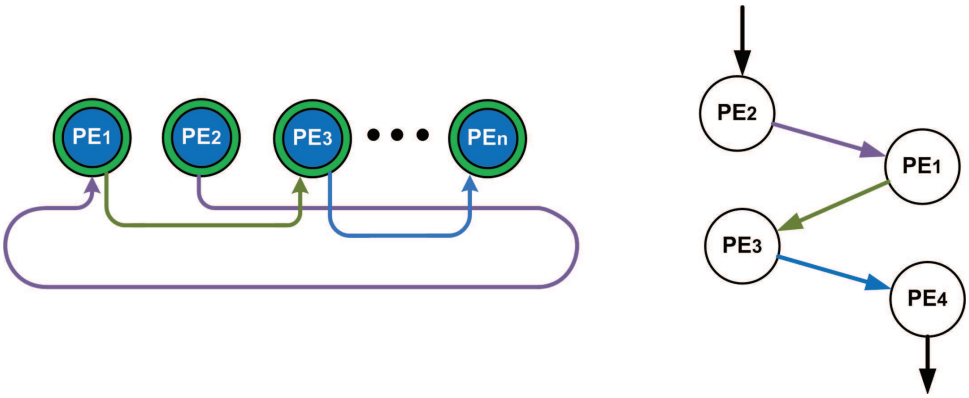

Figure 13.6: *Example of Scheduling of Pipeline Paradigm Processes on the Proposed Optical Bus.*

13.6 Design Effects on the Farm Pattern

The design effect on the farm pattern are discussed in this section. In absence of other parallel computations being executed, the emitter to workers

communication typical of the farm paradigm, could be executed with a latency equal to that of a simple point to point communication. No contention arises. On the other hand, workers to collector communications will result in the contention of the use of the wavelength associated to the collector. Higher latencies, that can depend on the arbitration strategy and the level in which they are implemented, can therefore be expected.

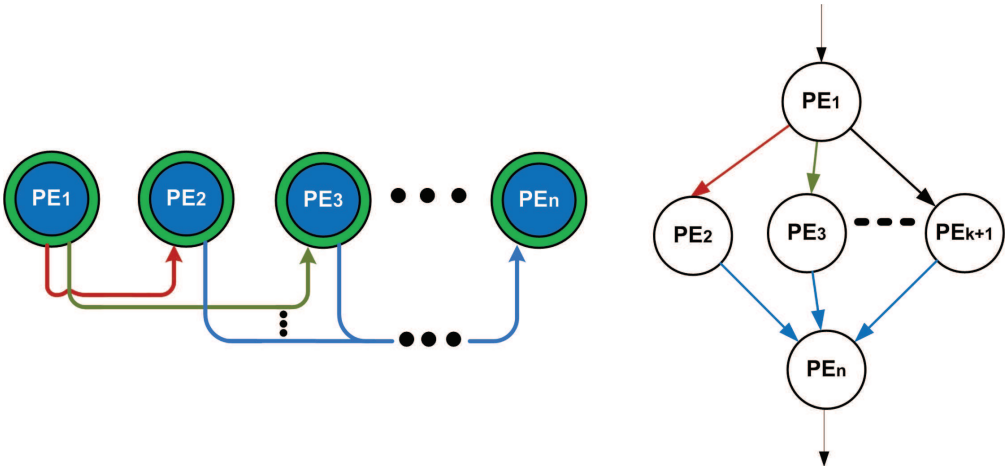

Figure 13.7: *Example of Scheduling of Farm Paradigm Processes on the Proposed Optical Bus.*

13.7 Study cases

13.7.1 Case 1

In [42], a proposal for the scheme of a multi-drop optical bus for chip to chip communication is presented together with the characteristics of its constituting optical components. We choose to analyze the scheme since, in principle it could also be deployed for on-chip communication. The bus is characterized by a master unit m, a set of slave units S and two sets of waveguides D and U:

$$bus = \{m, S, W\} \qquad S = \{s_i\}_{i=1}^{n} \qquad W = \{D, U\}$$

The master unit broadcasts signals to the slaves using the set D of waveguides which, in turn, reply to the master unit using a separate counter directional set U of waveguides with $D \cap U = \emptyset$. Both the bus for the *downward* (D, from the master to the slaves) and the bus for the *upward* (U, from the slaves to the master) are composed of 10 waveguides.

$$D = \{d_i\}_{i=1}^{10} \qquad U = \{u_i\}_{i=1}^{10}$$

In the downward links, a mirror couples the signals transmitted by the master unit to the downward waveguides with a 100% coupling. Each slave unit along the path, then, taps some optical power from the waveguides using an optical beamsplitter. The beamsplitters configuration is such that each slave unit can extract the same amount of power due to the fact that each beamsplitter has different reflectivity R_i and transmissivity T_i. These two parameters must also take into account the propagation loss:

$$k = e^{-\alpha L}$$

along each span of length L between two consecutive beamsplitters and, of course, the loss introduced by the beamsplitters themselves.

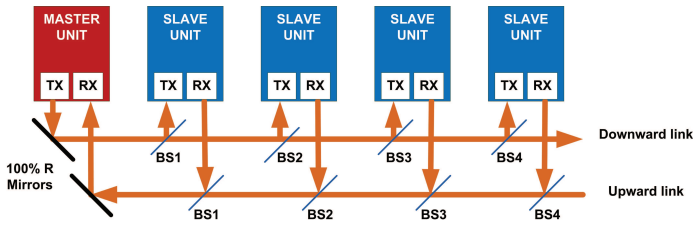

Figure 13.8: *Structure of the multidrop bus [42]*

Each transmitter is equipped with an array of 10 VCSELs (*Vertical Cavity Surface Emitting Laser*) and exploits direct modulation for a total bandwidth of 10 Gb/s. The optical beams produced in this way are then coupled to the waveguides with 90^o turned mirrors and converted back to the electrical domain by an array of 10 GaAs (Gallium Arsenide) PIN receivers: Only one slave unit at time can communicate with the master unit. The

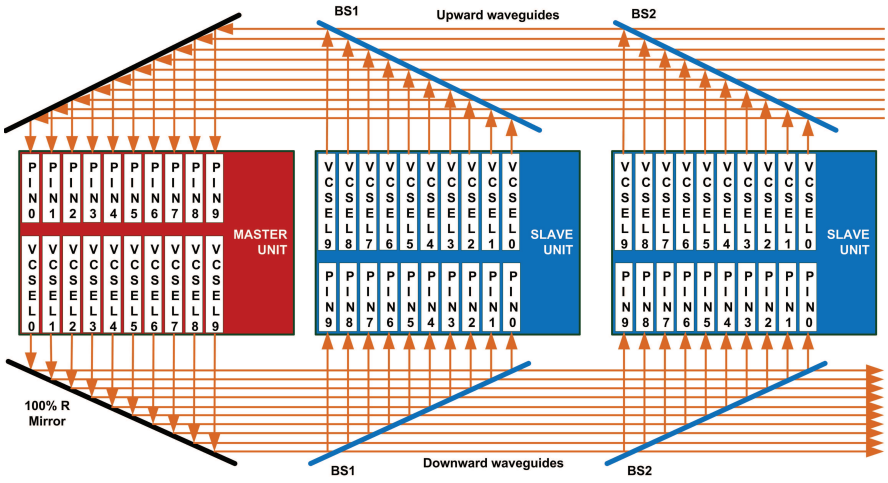

Figure 13.9: *Detailed Structure of master and slave units [42]*

scheduling of the *upward* communications could take place on demand or through an arbitration mechanism.

13.7.2 Case 2: Reliable Optical Bus (ROBUS)

In [36], the *Scalable Processor-Independent Design for Electromagnetic Resilience (SPIDER)* is proposed together with its *Reliable Optical Bus (ROBUS)*. The goal of this architecture is to provide an architecture characterized by an enhanced fault tolerance to electromagnetic interferences. Such an architecture is intended for highly reliable embedded control systems such as safety-critical aircraft functions.

Topology

The topology allows for the interconnection of n processing elements and is composed by n *Bus Interface Units (BIU)* and m *Redundancy Management Units (RMU)*. Each BIU is connected through a bidirectional link to a single PE. Furthermore, each RMU is bidirectionally connected to all the BIUs like in a star network.

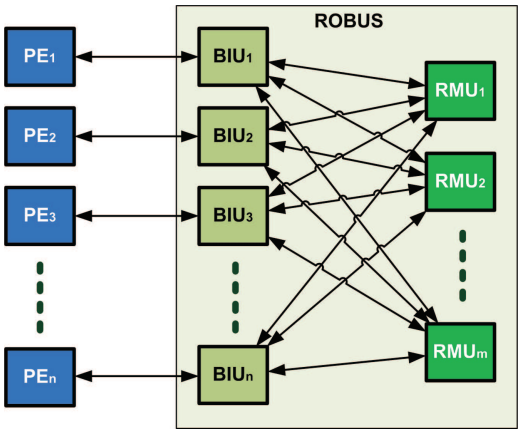

Figure 13.10: *Communication Topology of the ROBUS network [36].*

Chapter 14

Ring

A *one dimensional toroidal mesh*, also called *linear array with wraparound* or simply *ring*, is the lowest dimension toroidal topology which can be thought. It can be conceived as an extension of a linear array in which an additional link (the *wraparound*) connects the two peripheral nodes. The ring interconnection topology has a long story in the field of optical telecommunications where it has been deployed in various fashions in metro and regional optical communication networks. We analyze now its on-chip implementation and its corresponding performance. In Section 14.1 we analyze its topology.

14.1 Topology

The ring topology is a direct network constituted by a set of n processing nodes disposed in a chain fashion and connected with their predecessor and successor by dedicated bidirectional links for a total of n links:

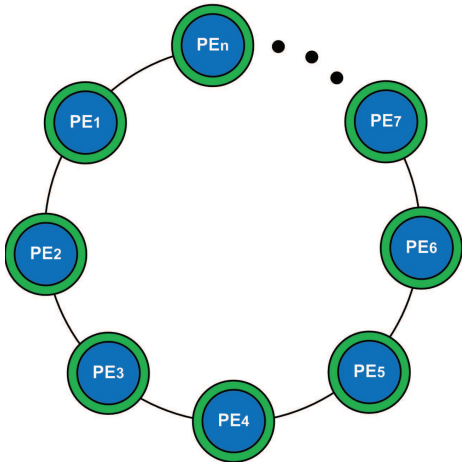

Figure 14.1: *Topology of a Ring Network.*

The diameter d of the network is equal to one half the number of nodes ($d = \frac{n}{2}$). Since there are n nodes each one with a degree of 2 and considering that each node can be reached from 2 counter propagating directions, there are a total of $2n$ interfaces.

Number of Nodes	n	$\bigcirc(n)$
Number of Links	n	$\bigcirc(n)$
Node Degree	2	2
Number of Interfaces	$2n$	$\bigcirc(n)$
Network Diameter	$\frac{n}{2}$	$\bigcirc(n)$

Table 14.1: *Summary of the Ring Topology Properties.*

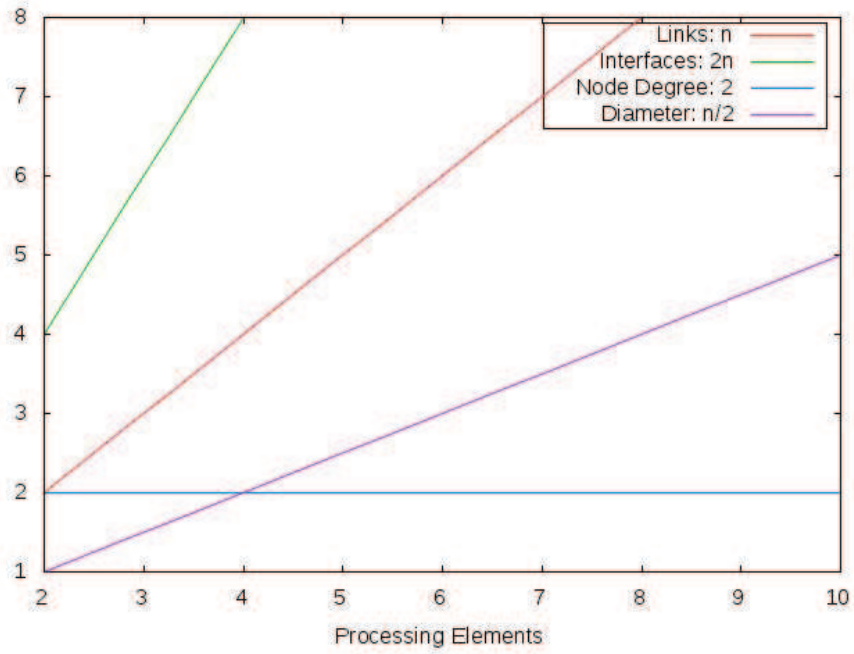

Figure 14.2: *Topology Statistics for an Optical Ring Network.*

14.2 Study Cases

14.2.1 Case 1: Optoelectrical Hierarchical Bus

In [20], an optoelectrical hierarchical bus interconnection network based on a ring topology is considered in order to analyze how to exploit optical technologies to replace global electrical on-chip interconnects. A 64 SMP manycore architecture based on the 32 nm technology is assumed as the

scenario. The architecture comprises also 16 L2 caches where each one of them is shared by 4 cores. L3 cache and main memory are outside the chip. An optical ring loop interconnects 4 optoelectrical switches. Each switch electrically interconnects 4 L2 caches. The bus is internally composed by an address bus of 64 bits, a data bus of 72 and a snoop response bus of 8. Each bit of the bus is carried by a dedicated waveguide.

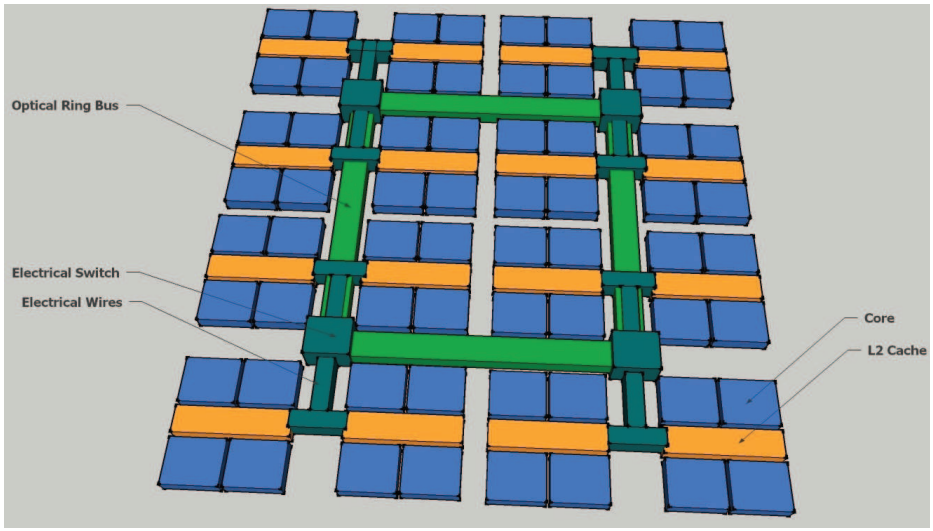

Figure 14.3: *Floorplan of the Bus Network Topology [20]*

14.2.2 Case 2: ORNoC - Optical Ring Network-on-Chip

In 2011, Le Beux et al. [24] proposed a contention-free new architecture called *Optical Ring Network-on-Chip* (ORNoC) and a methodology for wavelength / waveguide assignment. The network architecture is characterized by the fact that given a wavelength, this can be used for several communications within the same waveguide differently than the other architectures. The network is an electro-optical hybrid and distinguishes between the electrical portion and the optical portion. The electrical portion is composed of a set of clusters of computing nodes. Each of the processing nodes sharing the same cluster are interconnected among them with an electrical local NoC. On the other hand, the optical portion contains the optical components required to interconnect globally the different clusters.

Following this technical distinction is the differentiation between two type of communications:

- *intra-cluster/intra-layer* communications taking place among processing nodes belonging to the same electrical cluster and consisting of data flows;

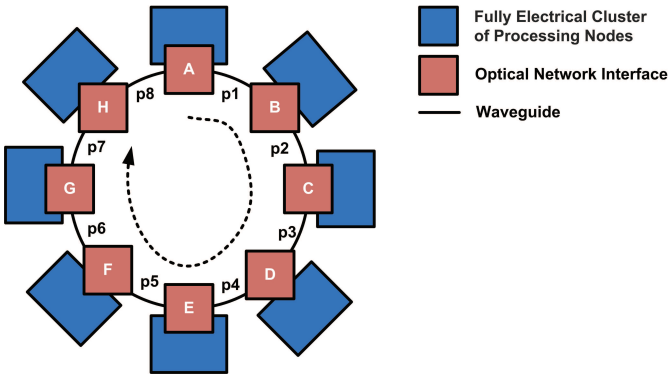

Figure 14.4: *Sample physical architecture of ORNoC [24].*

- *inter-cluster/inter-layer* communications used for the transmission of data flows between processing nodes belonging to different electrical clusters.

As can be easily infered from the above description, the OPNoC is used for the inter-cluster/inter-layer communications.

Routing Strategy

The routing of a data flow from a source processing node to a destination processing node follows three basic steps: in the first, the data flow is routed from within the source electrical cluster to the (or to one of the) *Optical Network Iterface* (ONI) together with the ID of the destination. At this point, at the second step, the ONI serializes the electrical data and drives the intensity-modulator of the corresponding transmitter obtaining electro-optic conversion according to the values of the various bits. After that, the optical data flow enters in the ORNoC, traverses some intermediate nodes and is then received by the destination ONI. Here the optical signal is photodetected producing the corresponding photocurrent and a CMOS circuit converts the analog signal back to its digital equivalent deserializing also the transmitted bits. Finally, the electrical data flow is routed through the destination electrical cluster until it reaches the correct destination node. In the following Figure, we have a set of 6 available wavelength. We notice that we can exploit each one of them for different communications; for example the red wavelength could be used for the communication of A with B as well as for the communication of B with C and so on. We could also use the green wavelength for the communication from B to H. The remaining wavelengths could be used as well for other end to end communications.

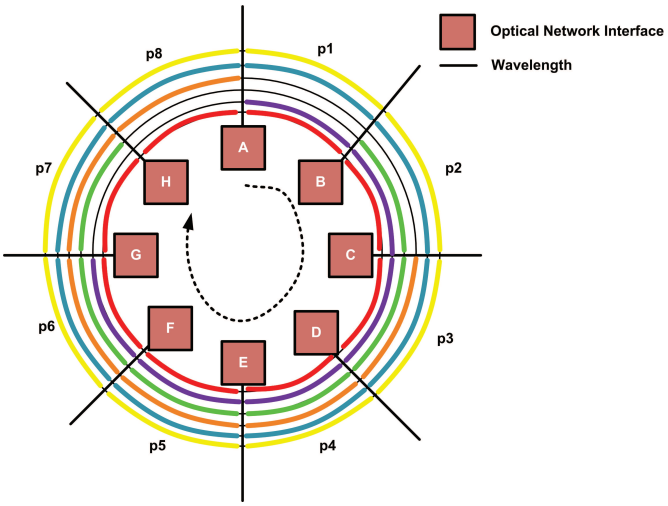

Figure 14.5: *Logical View of the ORNoC [24].*

Router Architecture

The router (ONI) architecture for the ORNoC interconnection network is composed by a data serialization electronic circuit; a CMOS microresonator driver circuit and a receiver and transmitter endpoints. The receiver part is equipped with a set of microresonators $\{m_1, ..., m_w\}$ where w is the number of wavelengths available. The microresonators are statically designed in order to extract from the bunch of incoming wavelengths a specific one in order to implement a drop port. The single channel optical signal extracted in this way is then photodetected by a photodetector and the correspondingly generated analogic photocurrent is then digitalized and deserialized by a proper CMOS electrical circuit. On the other side, the transmitter part is equipped with a set of lasers $\{l_1, ..., l_w\}$ each one specifically designed for emission of an optical beam at a fixed wavelength.

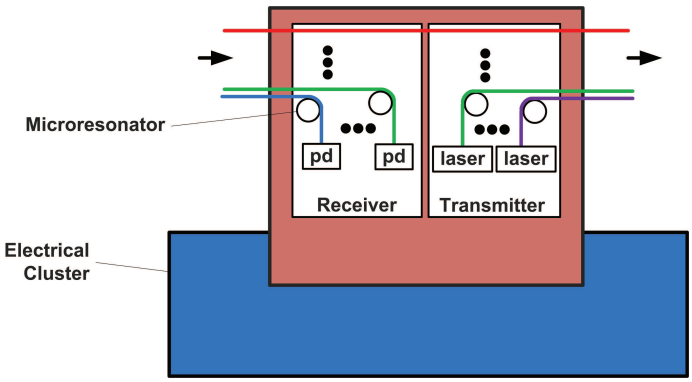

Figure 14.6: *Structure of an ORNoC ONI [24].*

Chapter 15

2D HERT

2D Hierarchical Expansion of Ring Topology

This novel hierarchical topology has been proposed by Koochi, Abdollahi and Hessabi at the 2011 Iternational Symposium on Networks On-Chip [23]. All the previously considered topologies are nothing else but an attempt to adapt traditional electrical interconnects using optical components. As stated in the research paper, this kind of approach does not fully exploit the physical properties of light and of the correspondent photonic components. Furthermore, while other architectures proposed in the literature are based on a separate electrical network in order to crossconnect the switches and resolve optical contentions for channels and ports, implying an high initial setup latency, 2D-HERT (*2D Hierarchical Expansion of Ring Topology*) places the phisical properties of light at a premium in the design of its architecture. In order to resolve contentions, 2D-HERT exploits Wavelength Division Multiplexing (WDM) and passive routing.

15.1 Topology

The topology (Figure 15.2) is composed by clusters of processing cores interconnected by local optical 1D rings. The clusters are then hierarchically interconnected by global optical 2D rings. There are a total of k clusters disposed in diagonals. Each diagonal contains m (even) supernodes and each supernode is composed of 4 routers and their correspondent 4 processing cores. The total number of processing cores N which can be interconnected is then equal to:

$$N = 4 \times m \times k \tag{15.1}$$

The topology keeps a constant node degree of 4 for any number of processing cores which keeps easy the realization of the routers.

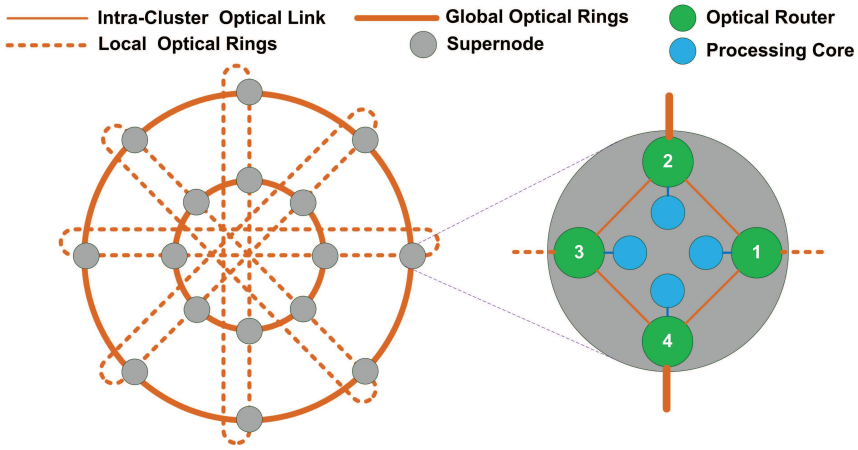

Figure 15.1: *The Topology of a 2D-HERT Network with 64 Cores.*

From the following figure we could notice that it seems easy to reach the integration of a very large number of processing cores while keeping a reasonably low number of diagonals and/or supernodes (clusters) per diagonal. However, we will see next that due to the WDM nature of the contention resolution scheme, some issues will arise due to the current technological limitation in the integrated WDM technology.

15.2 Layout

The layout proposed for 2D-HERT (Figure 15.3) has a small number of waveguide intersections in order to reduce the power loss and, at the same time, the complexity and cost of its realization.

Number of Nodes	$4mk$	$\bigcirc(mk)$
Number of Links	$4mm$	$\bigcirc(n)$
Node Degree	4	4
Number of Interfaces	$2nm$	$\bigcirc(nm)$
Network Diameter	$n+m-1$	$\bigcirc(n+m)$

Table 15.1: *Summary of the 2D HERT Topology Properties.*

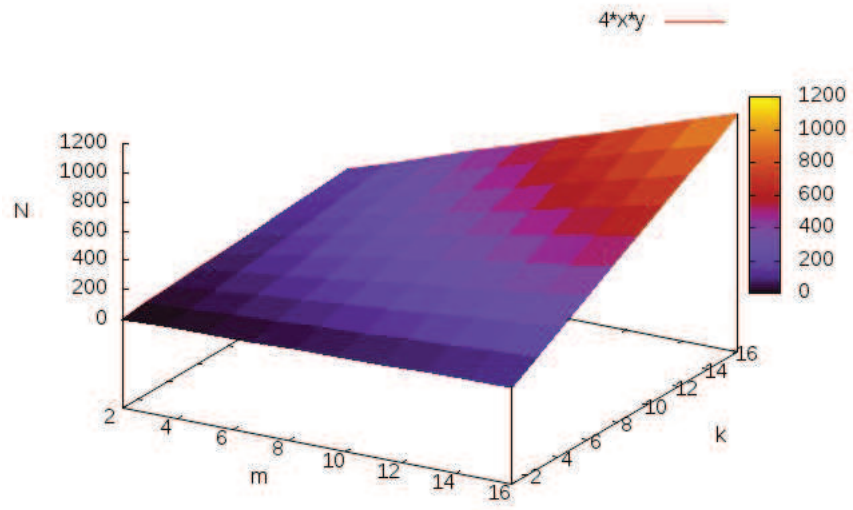

Figure 15.2: *Number of Nodes N for Varying m and k.*

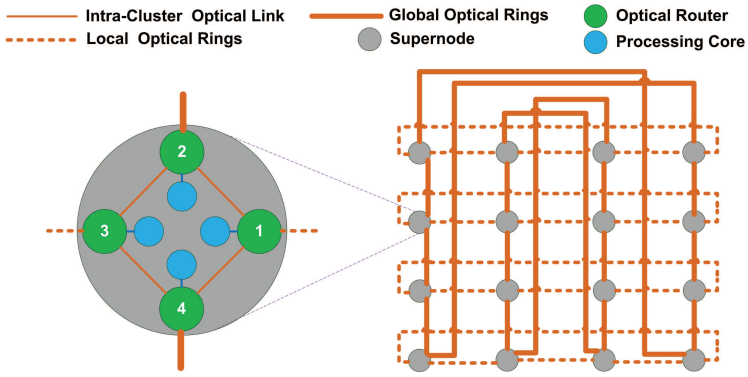

Figure 15.3: *The Layout of a 2D-HERT Network with 64 Cores.*

15.3 Routing Algorithm

Each optical router and its corresponding processing core can be uniquely identified by the triplet (d, s, r) where $d \in [0, k)$ is the index for the diagonal, $s \in [0, m)$ the index for the supernode within a diagonal and $r \in [0, 4]$ the index of a router within a given supernode. In [23], *Circular-first Routing* is proposed as routing algorithm designed ad-hoc for the 2D-HERT topology. The algorithm consists in traversing first the circular links and, once reached the destination diagonal, exploit the 1D rings to reach the destination supernode where intra-cluster links are used to reach one of the 4

optical routers and its corresponding processing core.

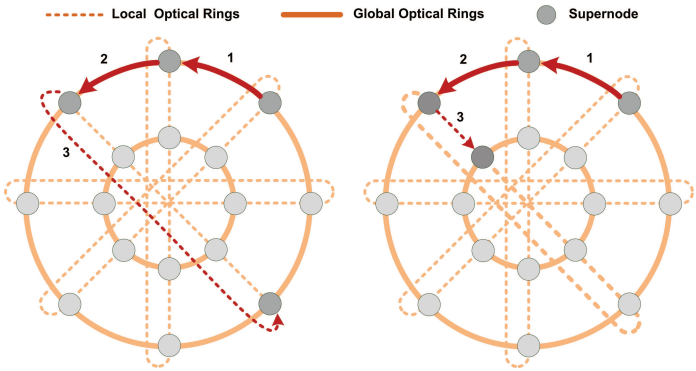

Figure 15.4: *Two Routing Examples Employing Circular-First Routing Scheme.*

The algorithm finds the minimal path from the source node to the destination node. The distance can be calculated as the sum of the steps taken to reach the correct destination diagonal moving along the global optical rings and the steps taken within the local optical ring belonging to the destintion diagonal. The wraparound link can be exploited depending on the position of the destination supernode (cluster) withing the diagonal. Starting from the source supernode, the optical global link can be traversed moving out from two interfaces. The interface that is chosen is the one that allows to minimize the number of steps along the optical global ring. Therefore the number of steps taken along the global link are:

$$\min\left(abs(d_{src} - d_{dst}), k - abs(d_{src} - d_{dst})\right) \tag{15.2}$$

Similarly, once the destination diagonal has been reached and depending on the position of the destination supernode, the local optical link can be traversed moving out from two different interfaces. Again, the interface chosen is the one that minimizes the number of moves. The number of steps along the diagonal is given by:

$$\min\left(abs(s_{src} - s_{dst}), m - abs(s_{src} - s_{dst})\right) \tag{15.3}$$

In summary, we have that the distance between two nodes src and dst is:

$$D = \begin{array}{l} \min\left(abs(d_{src} - d_{dst}), k - abs(d_{src} - d_{dst})\right) + \\ \min\left(abs(s_{src} - s_{dst}), m - abs(s_{src} - s_{dst})\right) \end{array} \tag{15.4}$$

Using this routing strategy, the network diameter (15.5) is equal to the sum of the longest path along a global optical link ($\lfloor k/2 \rfloor$) and the longest path that can be taken within a diagonal ($m/2$)

$$D_{max} = \lfloor k/2 \rfloor + m/2 \tag{15.5}$$

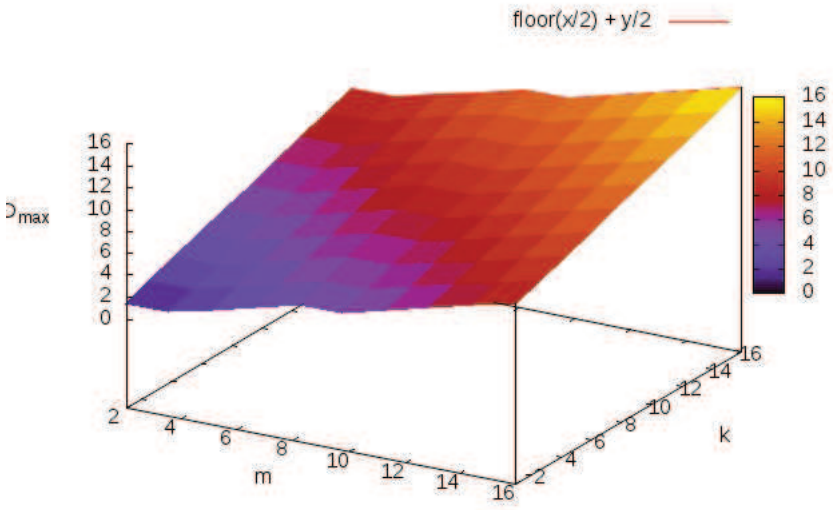

Figure 15.5: *Network Diameter D_{max} for varying m and k.*

15.4 Routing Architecture

The routing architecture for 2D-HERT is based on the *Wavelength-Switched Optical Router* (WSOR). WSOR is a passive optical component based on wavelength selective switches which eliminates the need for an initial electrical processing in order to cross-connect the most used SOI-based microring resonators. WSOR routes incoming optical packets on a wavelength base. At each WSOR connected to a processing core is associated one wavelength. The optical data streams are then modulated on different wavelengths depending on their destination WSOR. Note how the feasibility of this strategy is heavily impacted by the number of wavelengths technologically available in the realization of the optical infrastructure. The maximum number of WDM channels required by the architecture also called *maximum degree of multiplexing* (MDM) can be computed as the maximum number of data streams that can flow on a physical waveguide at the same time. The analysis presented requires the assumption that only one data stream is targeted to a given node at a time and considers the circular-first routing scheme. If we focus on a diagonal d, an optical data stream has to pass through at most $m/2$ waveguides and supernodes since also the wraparound links can be exploited to reach the destination supernode. Since each supernode is equipped with 4 optical routers each one with a dedicated wavelength, the

MDM for the local optical rings is then:

$$MDM_{lr} = 4\ \frac{m}{2} = 2m$$

The MDM for the global optical rings can be similarly computed. Since the topology is symmetric, we can move along the optical global links out of two possible interfaces. For this reason, the number of steps will be at most $k/2$. If we now also consider that each diagonal contains a total of $4m$ nodes, we have that the MDM for the global optical rings is equal to:

$$MDM_{gr} = 4\ m\ \frac{k}{2} = 2mk = \frac{N}{2}$$

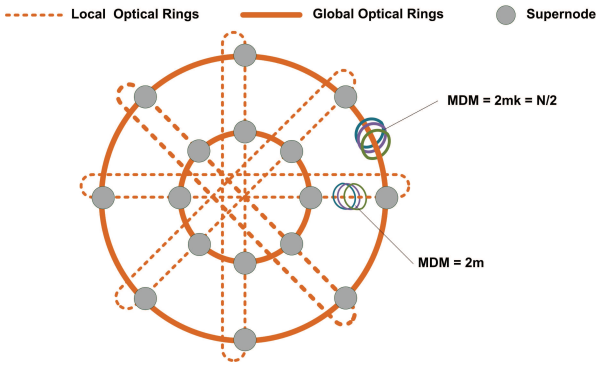

Figure 15.6: *Maximum Degree of Multiplexing for 2D-HERT.*

It is clear that the maximum number of channels that can traverse a link of 2D-HERT at a given time is given by the maximum between the MDM of the local optical rings and the MDM for the global optical rings and is therefore equal to $N/2$:

$$MDM = max\{MDM_{lr}, MDM_{gr}\} = max\{2m, 2mk\} = 2mk = \frac{N}{2} \quad (15.6)$$

Since the MDM is $N/2$, i.e. half the number of optical routers in the architecture and in order to utilize the minimum number of wavelegths required, each wavelength channel is assigned to two routers. In order to do so, the k diagonals are divided in two groups of adjacent diagonals and each diagonal is assigned $N/2$ different wavelength. Being an optical router univocally identified by the triplet (d_1, s_1, r_1), the other optical router which is assigned the same wavelength has address (d_2, s_2, r_2) where $d_1 = d_2\ mod\ (k/2)$, $s_1 = s_2$ and $r_1 = r_2$:

Despite this distribution of wavelengths to the nodes and the assumption that only one data stream can target a specific node at a time, it is still possible to have interference if two nodes n_1 and n_2 belong to different groups.

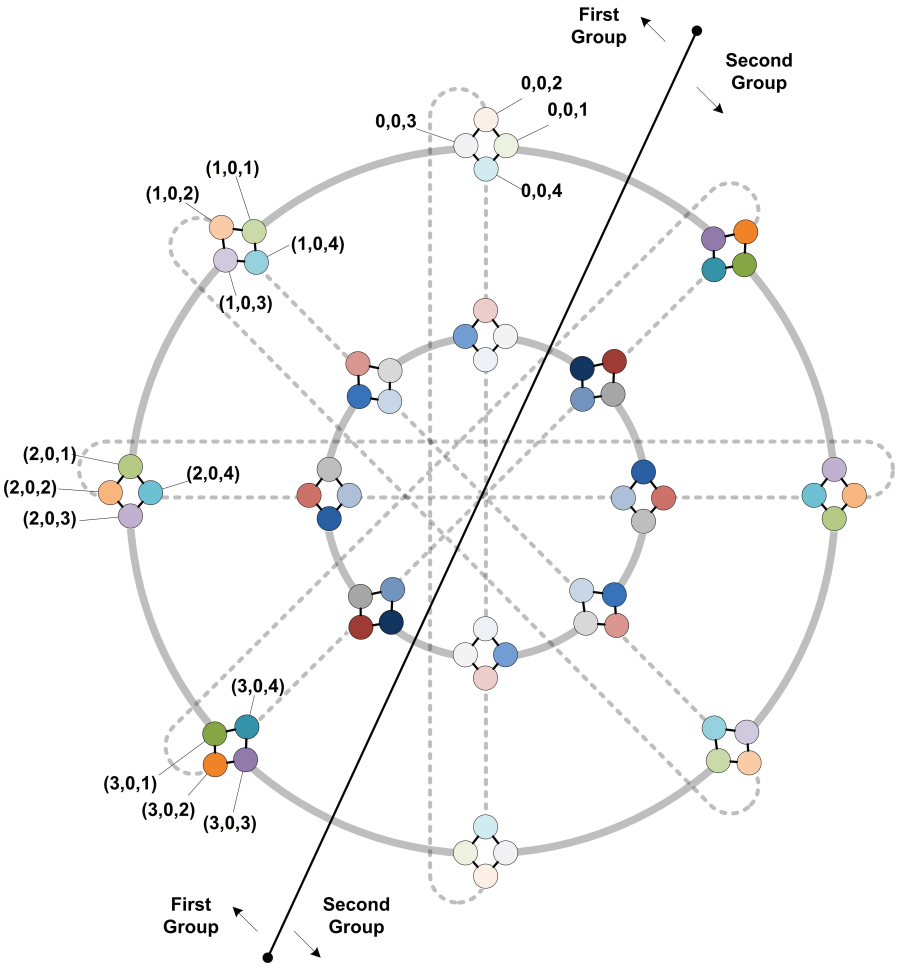

Figure 15.7: *An example of wavelength assignment in 2D-HERT.*

In order to solve this problem, the architecture is designed in such a way to discriminate the direction of the optical streams: within a supernode the clockwise direction is exploited to route optical streams to another node in the same wavelength group and counter clockwise direction is used for optical streams targeting nodes of the other wavelength group.

As we can notice in Figure 15.1, we can distinguish two types of WSOR:

- one type connecting two intra-supernode routers and another supernode on the same diagonal link which is called *radial router* and
- another type connecting two intra-supernode routers and another supernode on the same global link which is called *circular router*.

The two types of WSOR routers have different architectures.

Each WSOR routher could possibly inject n optical stream packets at a time where $1 \leq n \leq N-1$. Since each wavelength is assigned to two different routers, each WSOR router is equipped with two injection ports. The injection port is assumed to be selected by the processing core.

In order to extract an optical data packet characterized by its wavelength and direction of propagation, an *Optical Add and Drop* component (OAD) realized with *Ejection Microring Resonators* (EMRs) is used to extract (demultiplex) packets targeted to a given node.

Chapter 16

Hybrid Networks

16.1 Study Cases

16.1.1 Case 1: ET-PROPEL

In [37](2010), Morris and Kodi propose ET-PROPEL (*Extended-Token based Photonic Reconfigurable On-Chip Power and Area-Efficient Links*): a nanophotonic architecture which combines wavelength division multiplexing (WDM) and space division multiplexing (SDM), optical tokens and nanophotonic crossbars to develop a two-hop network for 256 cores. The interconnect is developed as a multilevel network.

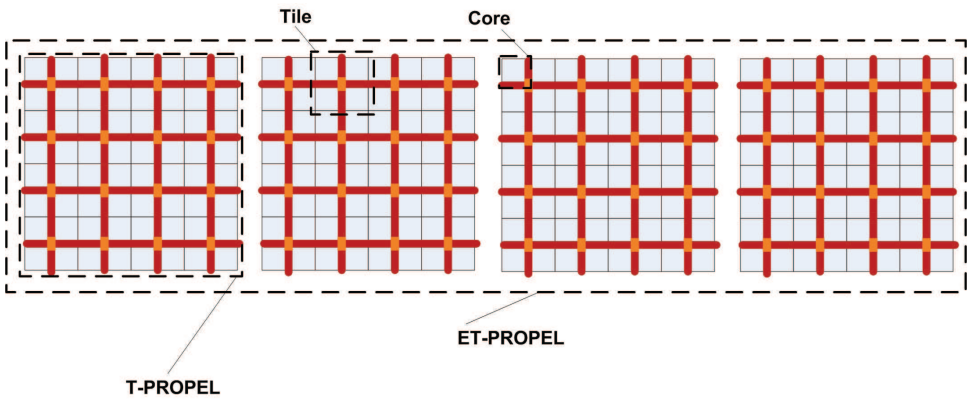

Figure 16.1: *Hierarchical Architecture of ET-PROPEL [37].*

In the first level, an electrical all-to-all net connects 4 cores wihin each tile in order to exploit locality. Each core has a dedicated L1 cache.

At the second level, an optical crossbar network with shared optical tokens works as an intermediary with the third level arbitration-free global optical crossbar [Figure 16.1]. The crossbar is composed by 16 tiles: 4 per dimension.

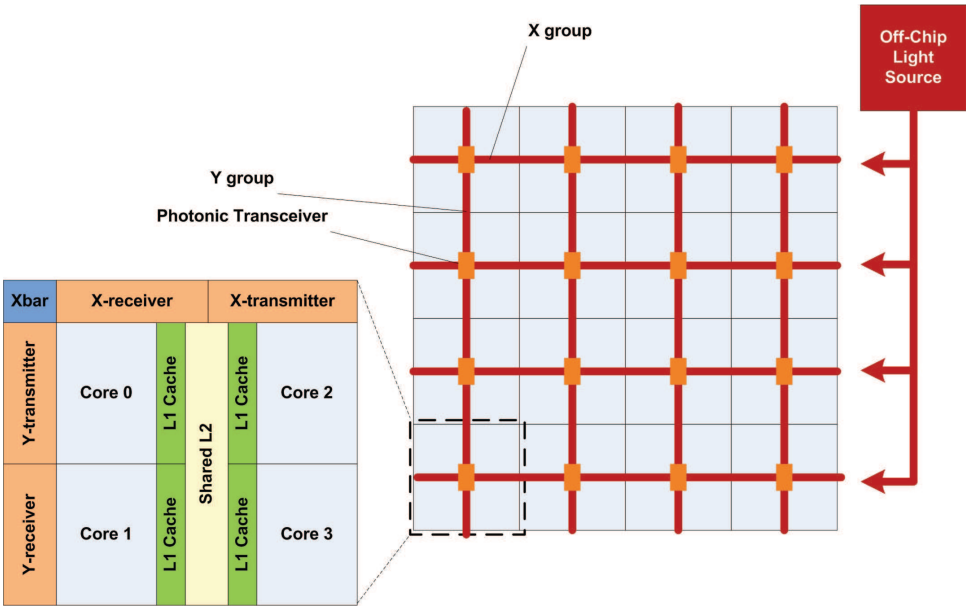

Figure 16.2: *Proposed Layout of a T-PROPEL Architecture for 64 Cores [37].*

If the source and the destination tiles are in the same row (x group), an electrical, arbitration free and fully connected communication is adopted. If, on the other hand, source and destination tiles are located on different rows (y groups), then optical tokens shared among tiles of the same x group are used. On the x direction, each waveguide is associated to a couple of source and destination tiles and all the wavelengths that traverse it are used for such point-to-point communication as in Figure 16.3. Each tile has 3 (i.e. $\sqrt{n}-1$) waveguides and then there are 12 (i.e. $\sqrt{n}(\sqrt{n}-1)$) waveguides per row. An additional ring waveguide is added for each row. This is the ring where an optical token arbitrates the access to the y direction waveguides. Considering also the token waveguide, each row of the net has 13 waveguides (i.e. $\sqrt{n}(\sqrt{n}-1)+1$).

In the optical token ring, 12 wavelengths circulate: one for each tile not in the same row, i.e. $n(n-1)$. When a tile wants to communicate with another tile in a different row, it acquires the token at the wavelength corresponding to the destination tile, communicates directly for a certain amount of time and then reinserts the token in arbitration ring.

At the third (optical) level, each cluster of tiles is interconnected by a multi-root optical fat tree. Tiles on different clusters but with the same coordinates (x,y) are connected with 4×4 optical crossbars with 64 wavelengths [Figure 16.4].

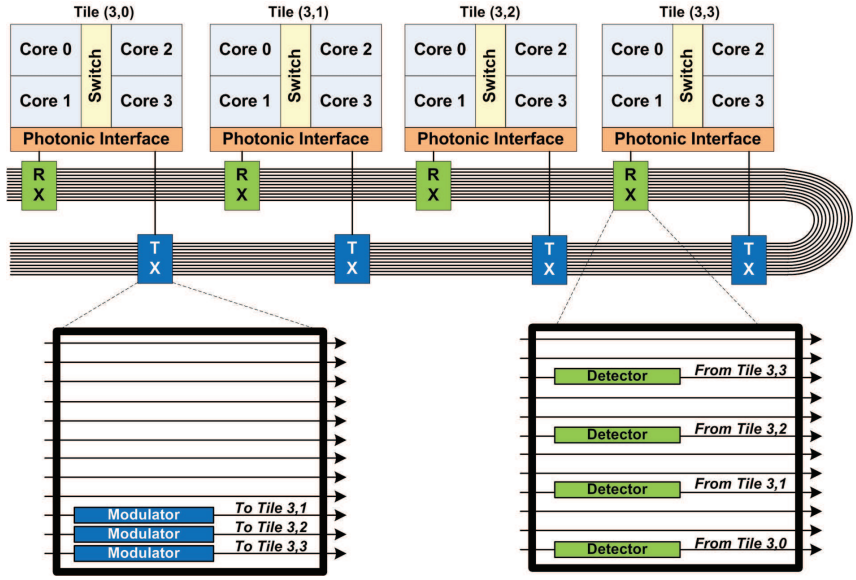

Figure 16.3: *The Routing and Waveguide Assignment Proposed for x-direction Communication. [37].*

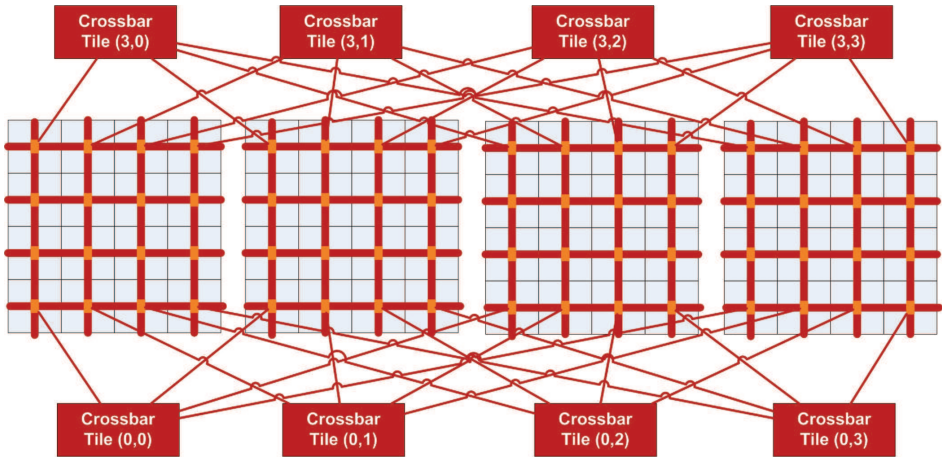

Figure 16.4: *Hierarchical Architecture of ET-PROPEL [37].*

16.1.2 Case 2

In [3], an hybrid opto-electrical manycore processor-to-memory network is presented using *local meshes to global switches* (LMGS), i.e. a topology which connects small groups (typically meshes) of on-chip cores to global off-chip switches located near the memory modules. The proposal includes the implementation of the on-chip network among cores and the global off-chip interconnect with the global switches.

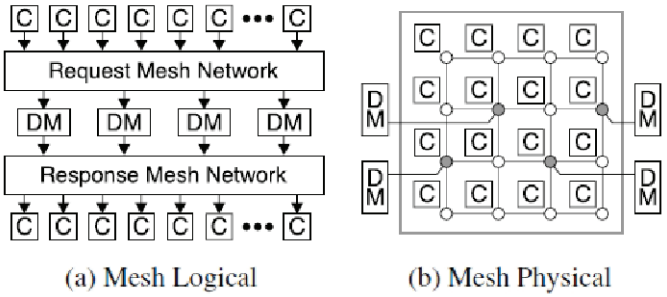

Figure 16.5: *Processor-to-Memory logical and physical interconnect topology*

The topology considered is logically divided into a topology for the memory requests and a topology for the memory responses. These two are also physically separated in order to avoid deadlock. Some of the on-chip mesh routers include also an *access point* (AP) which works as an interface between the on-chip network and the channel which connects to a DRAM memory module. In this way all the requests of a mesh of cores are collected by the AP and then forwarded to the memory module. The responses travel on the reverse path. The memory address space is interleaved across the APs in order to balance the load and provide better performance.

With the consideration that the most promising strategy to integrate photonics in on-chip interconnects is to use an external laser source and ring resonator-based modulators and filters, a CMOS bulk-compatible implementation is derived.

A *waveguide* is realized in a poly-Si layer over a shallow trench oxide isolation. Since the shallow trench is too thin to properly confine the modes propagating in the poly-Si core avoiding leakage losses into the Si substrate, using a new technique to etch the Si substrate underneath the poly-Si core it is possible to substitute it with air providing efficient confinement.

A *resonant ring filter* is obtained cascading a micro ring (two for higher precision) which, being designed with different radius, can select different wavelengths. Up to 64 wavelengths can be theoretically simultaneously transmitted and, considering bidirectional communications and an interleaved assignment of the counter-propagating wavelengths, we can reach 128 wavelengths for a *Dense Wavelength Division Multiplexing* (DWDM) final transmission.

The photonic *modulator* is still realized as a resonating microring which uses minority charge-injection to change the resonant frequency of the ring.

Photonic *receivers* can be realized with epitaxial Ge photodetectors which are quite common today.

Chapter 17

Conclusions

17.1 Summary

In this master thesis we have investigated the current state of the art of on-chip optical interconnection networks for multi-manycore architectures. In Chapter 1 we have first given an hystorical perspective of the fields of high performance computing and optical communication networks. In Section 1.3 we have stated which are the parameters of interest during the analysis and evaluation of the various optical interconnection networks that have been considered from the literature and that have been proposed. These parameters can be grouped in application parameters such as *bandwidth*, *latency*, *scalability* and engineering parameters such as *design cost*, *clock distribution*, *row bandwidth*. In Section 1.4, the need for optical on-chip interconnects is motivated with the drawbacks of their electrical counterparts such as *load added to driving gates*, *signal delay*, *signal noise* and their physical advantages like high signal propagation speed and hence lower signal propagation latency, lower power consumption and increased immunity to electromagnetic noise. On the other hand, one of the biggest problems with optical interconnects could come from the opto-electronic and electro-optic conversions which were considered responsible for an increase of the power consumption and latencies as happened for the long-haul optical communication systems. This issue could be suppressed if the optoelectronic devices are properly integrated in the electronic circuits [35]. The first chapter ends presenting the results of the most recent comparative research showing the advantages of integrated photonics over electrical IC at different integration technologies.

In Chapter 2, *"Optical NoC Design"* we have discussed the design issue of global chip clock distribution and remarked the quality of the solution proposed in the literature consisting in the optical H-tree. In section 2.2 we have investigated which are the mainstream solutions for integrating photonic components with CMOS circuitry. Monolithic integration, free-

space interconnects and 3D integration have received lot of attention but monolithic integration has emerged as the most promising strategy.

In Chapter 3, *"Optical Components"*, we have investigated some of the most recent examples of photonic integrated components that have been proposed in literature. The structure of transmitters have been analyzed in Section 3.1 and the various possibilities for light generation such as off-chip VCSELs (Vertical Cavity Surface Emitting Lasers), on-chip Raman lasers and Ge-on-Si on-chip lasers are briefly described. Ge-on-Si lasers represent a more promising alternative until now. In the rest of the chapter we discussed about modulator (microring resonators), waveguide structures (Section 3.3), Receivers (Section 3.4) and Filters (Section 3.5).

The second part of the thesis dealt with the architectural paradigms and the associated cost models emerged in the field of high performance parallel architectures. In Chapter 4 we introduced a taxonomy for the classification of the various architectural paradigms. In chapter 5 we summarized the characteristics, advantages and drawbacks of shared memory uniform memory access architectures (UMA) while in Chapter 6 we considered the non-uniform memory access architectures (NUMA). In both chapters, the role of the on-chip interconnection network is extracted and analyzed in order to understand how different designs could affect the overall performance of the architectural paradigms. Furthermore several research and commercial manycore architectures are illustrated as study cases. In Chapter 7 we built a firmware cost model starting from the most modern methodology for designing parallel applications at the process level. The two most important stream parallel computational patterns: *pipeline* and *farm* are then summarized since the analysis of the networks in the remaining of the thesis was against the latencies experienced by the communications in the context of them.

In the third part of the thesis, building on the background developed in the first two parts, we analyzed some of the most common interconnection networks categorized in two main subgroups: indirect networks and direct networks. In these chapters we analyzed different interconnection networks proposing some possible implementations and presenting the most significative solutions coming from the literature of the last few years.

17.2 Considerations

While the thesis' goal was to investigate the current state of the art of the on-chip optical interconnection networks, we proposed some optical architectures describing the structure of related optical switches and discussing possible routing strategies. In Chapter 8 we proposed first a solution to integrate a clock distribution network within a star network and then a possible strategy for data communication. While this proposal has no significative

advantages, the integrated global optical clock distribution network represents a possibly very performant solution that needs to be investigated and will probably be adopted in the near future. We also investigated, for the first time, the design effects of the proposed solution on the pipeline and farm parallel computational patterns developing a simple cost model for the network communication latency.

In Chapter 9 we proposed an optical crossbar architecture for the interconnection of chip cores or tiles with external main memory blocks. The solution based on wavelength division multiplexing allows for high communication bandwidth and, thanks to a uniform latency can potencially make easier the job of developers of parallel applications or of run time supports in finding the best mapping of processes on processors. The high level structuring of the 4 types of switching nodes is described without making assumptions on the internal structure or characteristics of the optical components. Again, possible effects of the mapping of pipeline or farm structured parallel computations on our interconnection network are analyzed. Finally 3 interesting study cases from the literature are summarized with their most remarkable features.

In Chapter 10 we propose to integrate an optical H-tree for clock distribution within the switching nodes of a tree indirect network. The issues related to the implementation of a simple optical tree network introduce the analysis of the fat tree network in Chapter 11. An optical WDM fat tree is proposed together with the structure of the intermediate and root switching nodes. After discussing again the effect on pipeline and farm patterns, a study case from the literature has been analyzed.

The study of the Clos network of Chapter 12 revealed several issues regarding a possible current implementation of an on-chip Clos network. The problem of pin count detrimentally affects this network topology and the current absence of valid solutions for optical buffering forces the designers to adopt the electrical domain to design the intermediate routers in order to implment queueing, routing and arbitration.

While indirect networks have been extensively studied, direct networks will surely emerge as the number of cores that could be implemented in a single chip will drastically increase: in the future SoC, communication will earn central importance in the overall computation.

We started the analysis of the direct networks from the classical bus network. A proposal for clock distribution integration, architecture, routing and data communication is made in Chapter 13. The performance effects on pipeline and farm patterns is investigated. As for the other proposed networks, all-to-one collective communications represent a source of contention and need for arbitration that, since cannot be done in the optical domain, must be dealt with at higher levels of the architecture (firmware) or with a parallel electrical network sensing the optical one. The last case is the one of hybrid networks that exploit electrical all-to-all interconnects at local

level and optics at global level or, as an alternative, they have an extension identical to the parallel optical network.

In Chapter 14 we present one study case from the literature regarding an optical on-chip ring network. This topology has undoubtely received the greatest attention at the beginning of the optical NoC era.

17.3 Future Work

In Chapter 15 we report a novel photonic interconnection network presented in 2011 at the International Symposium on Networks on chip by Koochi et al. The approach followed represents the very new concept: until now, all the design proposals for optical on-chip networks relayed on the adaptation of classical electrical networks. On the other hand, taking into account the physical properties of light at the beginning of the design of an optical network can, as demonstrated, exploit better its peculiarities allowing for a much higher degree of scalability unmatched by the classical approach. This is the approach we wish to suggest to those who plan to work in this field with, as an extra hint, to try to integrate in the design phase the considerations on the parallel essence of the overall architecture that must be developed.

A less attractive alternative to the approach presented in Chapter 15 is the realiztion og hybrid hierarchical networks where different topologies are used at different levels of interconnection. In Chapter 16 we present some study cases regarding hybrid networks which exploit differnt network topologies at different interconnection levels always within the chip.

Future work will surely regard the investigation of possible solution for improving the quality of on-chip lasers which are, at the current state, the optical components which requires the higher attention since has been introduced only in the last two years (Ge-on-Si) lasers.

All the optical networks that we proposed in this thesis need to be further studied and experimentally vaildated.

The work of this thesis also inspired a design strategy for a novel interconnection network that needs to be investigated both theoretically and experimentally and that will be addressed in further documents.

List of Tables

List of Figures

Bibliography

[1] *Commun. ACM*, 55(6), 2012.

[2] N.K. Bambha, S.S. Battacharyya, and G. Euliss. Design considerations for optically connected systems on chip. In *System-on-Chip for Real-Time Applications, 2003. Proceedings. The 3rd IEEE International Workshop on*, pages 299 – 303, june-2 july 2003.

[3] C Batten, A Joshi, J Orcutt, C Holzwarth, M Popovic, J Hoyt, F Kartner, R Ram, V Stojanovic, and K Asanovic. Building manycore processor-to-dram networks with monolithic cmos silicon photonics. *Micro, IEEE*, 2009.

[4] K. Bergman. Silicon photonic on-chip optical interconnection networks. In *Lasers and Electro-Optics Society, 2007. LEOS 2007. The 20th Annual Meeting of the IEEE*, pages 470 –471, oct. 2007.

[5] J.T. Bessette, R. Camacho-Aguilera, Yan Cai, L.C. Kimerling, and J. Michel. Optical characterization of ge-on-si laser gain media. In *Group IV Photonics (GFP), 2011 8th IEEE International Conference on*, pages 130 –132, sept. 2011.

[6] R. Camacho-Aguilera, J. Bessette, Yan Cai, L.C. Kimerling, and J. Michel. Electroluminescence of highly doped ge pnn diodes for si integrated lasers. In *Group IV Photonics (GFP), 2011 8th IEEE International Conference on*, pages 190 –192, sept. 2011.

[7] E. Cassan, S. Lardenois, D. Pascal, L. Vivien, M. Heitzmann, N. Bouzaida, L. Mollard, R. Orobtchouk, and S. Laval. Intra-chip optical interconnects with compact and low-loss light distribution in silicon-on-insulator rib waveguides. In *Interconnect Technology Conference, 2003. Proceedings of the IEEE 2003 International*, pages 39 – 41, june 2003.

[8] Guoqing Chen, Hui Chen, M. Haurylau, N.A. Nelson, D.H. Albonesi, P.M. Fauchet, and E.G. Friedman. On-chip copper-based vs. optical interconnects: Delay uncertainty, latency, power, and bandwidth density

comparative predictions. In *Interconnect Technology Conference, 2006 International*, pages 39 –41, 2006.

[9] Murray Cole. *Algorithmic skeletons: structured management of parallel computation*. MIT Press, Cambridge, MA, USA, 1991.

[10] Jose Duato, Sudhakar Yalamanchili, and Lionel Ni. *Interconnection Networks - An Engineering Approach*. Morgan Kaufmann.

[11] P. Dumon, W. Bogaerts, V. Wiaux, J. Wouters, S. Beckx, J. Van Campenhout, D. Taillaert, B. Luyssaert, P. Bienstman, D. Van Thourhout, and R. Baets. Low-loss soi photonic wires and ring resonators fabricated with deep uv lithography. *Photonics Technology Letters, IEEE*, 16(5):1328 –1330, may 2004.

[12] J.W. Goodman, F.J. Leonberger, Sun-Yuan Kung, and R.A. Athale. Optical interconnections for vlsi systems. *Proceedings of the IEEE*, 72(7):850 – 866, 1984.

[13] Huaxi Gu, Jiang Xu, and Wei Zhang. A low-power fat tree-based optical network-on-chip for multiprocessor system-on-chip. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, pages 3 –8, april 2009.

[14] John L. Hennessy and David A. Patterson. *Computer Architecture: a Quantitative Approach*. Morgan Kaufmann.

[15] R. Ho, K.W. Mai, and M.A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490 –504, April 2001.

[16] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Pailet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, R. Van Der Wijngaart, and T. Mattson. A 48-core ia-32 message-passing processor with dvfs in 45nm cmos. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pages 108 –109, feb. 2010.

[17] Dawei Huang, T. Sze, A. Landin, R. Lytel, and H.L. Davidson. Optical interconnects: out of the box forever? *Selected Topics in Quantum Electronics, IEEE Journal of*, 9(2):614 – 623, march-april 2003.

[18] A. Joshi, C. Batten, Yong-Jin Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic. Silicon-photonic clos networks for global on-chip communication. In *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, pages 124 –133, may 2009.

[19] A. Kazmierczak, W. Bogaerts, E. Drouard, F. Dortu, P. Rojo-Romeo, F. Gaffiot, D. Van Thourhout, and D. Giannone. Highly integrated optical 4 × 4 crossbar in silicon-on-insulator technology. *Lightwave Technology, Journal of*, 27(16):3317 –3323, aug.15, 2009.

[20] N. Kirman, M. Kirman, R.K. Dokania, J.F. Martinez, A.B. Apsel, M.A. Watkins, and D.H. Albonesi. Leveraging optical technology in future bus-based chip multiprocessors. In *Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on*, pages 492 –503, dec. 2006.

[21] N. Kirman, M. Kirman, R.K. Dokania, J.F. Martinez, A.B. Apsel, M.A. Watkins, and D.H. Albonesi. On-chip optical technology in future bus-based multicore designs. *Micro, IEEE*, 27(1):56 –66, jan.-feb. 2007.

[22] S.J. Koester, C.L. Schow, L. Schares, G. Dehlinger, J.D. Schaub, F.E. Doany, and R.A. John. Ge-on-soi-detector/si-cmos-amplifier receivers for high-performance optical-communication applications. *Lightwave Technology, Journal of*, 25(1):46 –57, jan. 2007.

[23] S. Koohi, M. Abdollahi, and S. Hessabi. All-optical wavelength-routed noc based on a novel hierarchical topology. In *Networks on Chip (NoCS), 2011 Fifth IEEE/ACM International Symposium on*, pages 97 –104, may 2011.

[24] S. Le Beux, J. Trajkovic, I. O'Connor, G. Nicolescu, G. Bois, and P. Paulin. Optical ring network-on-chip (ornoc): Architecture and design methodology. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pages 1 –6, march 2011.

[25] Charles E. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Trans. Comput.*, 34:892–901, October 1985.

[26] Ansheng Liu, Richard Jones, Ling Liao, Dean Samara-Rubio, Doron Rubin, Oded Cohen, Remus Nicolaescu, and Mario Paniccia. A high-speed silicon optical modulator based on a metal-oxide-semiconductor capacitor. *Nature, Letters to Nature*.

[27] Jifeng Liu, Rodolfo E. Camacho-Aguilera, Yan Cai, Jonathan T. Bessette, Xiaoxin Wang, Lionel C. Kimerling, and Jurgen Michel. Ge laser and on-chip electronic-photonic integration. In *Opto-Electronics and Communications Conference (OECC), 2012 17th*, pages 277 –278, july 2012.

[28] Jifeng Liu, Xiaochen Sun, R. Camacho-Aguilera, Yan Cai, L.C. Kimerling, and J. Michel. Optical gain and lasing from band-engineered ge-

on-si at room temperature. In *OptoeElectronics and Communications Conference (OECC), 2010 15th*, pages 520 –521, july 2010.

[29] Jifeng Liu, Xiaochen Sun, R. Camacho-Aguilera, Yan Cai, J. Michel, and L.C. Kimerling. Band-engineered ge-on-si lasers. In *Electron Devices Meeting (IEDM), 2010 IEEE International*, pages 6.6.1 –6.6.4, dec. 2010.

[30] Ruibing Lu, Guoan Zhong, Cheng-Kok Koh, and Kai-Yuan Chao. Flip-flop and repeater insertion for early interconnect planning. In *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings*, pages 690 –695, 2002.

[31] T.G. Mattson, R.F. Van der Wijngaart, M. Riepen, T. Lehnig, P. Brett, W. Haas, P. Kennedy, J. Howard, S. Vangal, N. Borkar, G. Ruhl, and S. Dighe. The 48-core scc processor: the programmer's view. In *High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for*, pages 1 –11, nov. 2010.

[32] Timothy G. Mattson, Rob Van der Wijngaart, and Michael Frumkin. Programming the intel 80-core network-on-a-chip terascale processor. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC '08, pages 38:1–38:11, Piscataway, NJ, USA, 2008. IEEE Press.

[33] J. Michel, R.E. Camacho-Aguilera, Yan Cai, N. Patel, J.T. Bessette, M. Romagnoli, B. Dutt, and L.C. Kimerling. An electrically pumped ge-on-si laser. In *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 and the National Fiber Optic Engineers Conference*, pages 1 –3, march 2012.

[34] J. Michel, Jifeng Liu, L.C. Kimerling, R. Camacho-Aguilera, J.T. Bessette, and Yan Cai. A germanium-on-silicon laser for on-chip applications. In *Lasers and Electro-Optics (CLEO), 2011 Conference on*, pages 1 –2, may 2011.

[35] D.A.B. Miller. Optical interconnects to silicon cmos. In *Device Research Conference, 2001*, pages 35 –38, 2001.

[36] P.S. Miner, M. Malekpour, and W. Torres. A conceptual design for a reliable optical bus (robus). In *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, volume 2, pages 13D3–1 – 13D3–11 vol.2, 2002.

[37] R.W. Morris and A.K. Kodi. Power-efficient and high-performance multi-level hybrid nanophotonic interconnect for multicores. In *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pages 207 –214, may 2010.

[38] V. Nookala and S.S. Sapatnekar. Designing optimized pipelined global interconnects: algorithms and methodology impact. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 608 – 611 Vol. 1, may 2005.

[39] Ian O'Connor and Frederic Gaffiot. On-chip optical interconnect for low power. *Ultra Low-Power Electronics and Design*, pages 1 – 20, 2004.

[40] Halsheng Rong, Richard Jones, Ansheng Liu, Oded Cohen, Dani Hak, Alexander Fang, and Mario Paniccia. A continuous-wave raman silicon laser. *Nature, Letters to Nature.*

[41] Bahaa E.A. Saleh and Malvin Carl Teich. *Fundamentals of Photonics.* Wiley Interscience.

[42] M. Tan, P. Rosenberg, Jong Souk Yeo, M. McLaren, S. Mathai, T. Morris, J. Straznicky, N.P. Jouppi, Huei Pei Kuo, Shih-Yuan Wang, S. Lerner, P. Kornilovich, N. Meyer, R. Bicknell, C. Otis, and L. Seals. A high-speed optical multi-drop bus for computer interconnections. In *High Performance Interconnects, 2008. HOTI '08. 16th IEEE Symposium on*, pages 3 –10, aug. 2008.

[43] G. Tasik, F. Gaffiot, Z. Lisik, and I. O'Connor. Optical versus electrical clock system in future vlsi technologies. In *SOC Conference, 2003. Proceedings. IEEE International [Systems-on-Chip]*, pages 261 – 262, sept. 2003.

[44] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, and N. Borkar. An 80-tile 1.28tflops network-on-chip in 65nm cmos. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pages 98 –589, 2007.

[45] Marco Vanneschi. *Parallel Architectures.* University of Pisa.

[46] Marco Vanneschi. *Structuring and Design Methodology for Parallel Applications.* University of Pisa.

[47] Qianfan Xu, S. Manipatruni, B. Schmidt, J. Shaky, and M. Lipson. 12.5 gbit/s silicon micro-ring silicon modulators. In *Lasers and Electro-Optics, 2007. CLEO 2007. Conference on*, pages 1 –2, may 2007.

Chapter 18

Acknowledgements

May the reader forgive me if I switch to Italian, my native language, to thank all the people who helped me in this project and in the path along the whole master course.

Vorrei ringraziare, prima di tutti, la mia famiglia per il supporto che mi ha dato durante tutto il percorso del master, incoraggiandomi a intraprendere questo percorso di studi e sostenendomi nei momenti di difficoltà. Ringrazio i miei relatori, il Professor Marco Vanneschi e il Professor Piero Castoldi per avermi dato, al termine di un master veramente innovativo, la possibilità di affrontare nella tesi un tema altrettanto innovativo, interessante e pieno di possibilità future. Non posso non ringraziare tutte le persone che con con utilissime discussioni mi hanno aiutato a comprendere alcuni aspetti del corso di studi e della tesi, tra cui Davide La Rosa, Filippo Venuti, Venkatraman Gopalakrishnan, Emanuele Vespa, Angela Italiano, Nicola Maggi, Francesco Piccinno, Daniele De Sensi, Francesca Pacini, Sina Fazel, Tudor Serban, Yonas Seifu, Md Sabbir Ahmed, Stefania Vittori, e i dottori Nicola Andriolli e Isabella Cerutti. Ringrazio anche tutti gli amici che anche a distanza mi hanno sempre incoraggiato e aiutato: Luca Sanna, Jonida Lilay, Daniele Etzi, Stefano Cotza, Andrea Cabras. Come dimenticare poi gli altri compagni di questa avventura: Marilena Stano, Simone Giuliani, Francesca Martinelli, Andrea Bozzi, Roberto (Bob), Mayank Sinha, Giuseppe De Vivo, Emnet Tsadiku Abdo, Yelkal Mulualem, Dean De Leo, Celeste Concari, Valentina Macchione, Dolcey Torres, Ion Popescu, Maria Teresa Galiulo, Nancy Wu, Melat Tesfaye, Leta Melkamu, e the Pakistani guys.