THE PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY FOR HIGHER EDUCATION AND SCIENTIFIC RESEARCH
IBN KHALDOUN UNIVERSITY - TIARET
FACULTY OF APPLIED SCIENCES
ELECTRICAL ENGINEERING DEPARTMENT

GRADUATION MEMOIR

To obtain The Master's Degree

Field: *Science and Technology*

Sector: *Electrical Engineering*

Specialty: *EMBEDDED SYSTEMS*

Theme

IA sur FPGA : Vers une surveillance automatisée de la biodiversité végétale des plantes médicinales

Prepared by:

- TERLEBAS LOUBNA
- ABDERRAHMANE ABDELWAHED MOKHTAR

Presented in front of the jury composed of:

Dr. BELARBI Mustapha	MCA	Supervisor
Dr. BELHADJI YOUCEF	MCA	President
Dr. NASRI Djilali	PROF	Examiner
Dr. MAAMAR NOUREDDINE	MAA	Examiner

2023/2024

Theme

AI on FPGA: Towards Automated Monitoring of the Plant Biodiversity of Medicinal Plants

Acknowledgements

In the first place :

We would like to thank **ALLAH** who has given us the health and ability to complete and present this thesis.

Our sincere and heartfelt thanks
Are specifically addressed to our supervisor **MUSTAPHA BELARBI**
Whom we were fortunate to have as our supervisor for his trust, continuous encouragement, and close guidance throughout the completion of our thesis

We extend our sincerest thanks to the members of the jury who honor us by evaluating our work.

Our deep gratitude and sincere thanks go to all the teachers who have supported us throughout our academic journey.
We would like to express our heartfelt thanks to our parents for their support, patience, and encouragement throughout our academic journey.

Not forgetting to express our gratitude to our siblings and close friends for their encouragement as well.

Thank you all.

Dedications

♡ I dedicate this work ♡

To my family, friends and those who supported and encouraged me throughout the hard times in this journey.

I am especially grateful to my beloved parents who have always shown endless support ,love and believed in me and provided me the motivation and all sorts of support to achieve my goals.

To my professors thank you for your guidance,wisdom and patience.

Finally to my colleagues with whom I have shared countless hours of hard work and perseverance ,thank you for being there trough every step of this journey .

Terlebas Loubna

dedications

♡ I dedicate this work ♡

In the first place :

To my parents, for all their sacrifices, support, encouragement, and love, which have been the reason for my success. May God grant them good health and a long life

To my wife WAFA, for her immense patience, I sincerely thank her, especially for her unwavering moral support and countless pieces of advice throughout my thesis.

Not forgetting my elder brother Kamal,For his unwavering encouragement throughout my academic journey, I dedicate this work to him with deep gratitude.

To my sisters Hadjer and Khaoula, and my brothers Zakaria and Younes, for their constant encouragement, I dedicate this work with deep gratitude.

To my dear friends, I thank you for your sincere friendship and encouragement. And to all the people who, through their love

and encouragement, have paved the way for me to reach the heights of knowledge.

ABDERRAHMANE ABDELWAHED MOKHTAR

ملخص

يقدم هذا المشروع الابتكاري نموذج بحث مبتكر حول نظام ذكي للتعرف الآلي ومراقبة النباتات الطبية على مستوى بيئة بيولوجية، متضمنًا دمج خوارزميات الذكاء الاصطناعي والتعلم العميق موازاة مع استغلال القدرات الهائلة لمصفوفة البوابات المنطقية القابلة للبرمجة، التي تتجلى في سرعة الأداء العالية والتعدد في استخدامها.

يعرض هذا المشروع آلية تعرف ومراقبة سريعة في الوقت الحقيقي مع أداء عال ومحافظة على كفاءة الطاقة.

هذا المشروع من شأنه المساهمة في تسهيل وتسريع البحث الصيدلاني، بالإضافة إلى استغلال توفر النباتات الطبية بكثرة، خاصة في الهضاب العليا والصحراء، من أجل توفيرها كمادة أولية لإنتاج الأدوية والمستحضرات التجميلية وخلق الثروة في المجال الصيدلاني.

الكلمات المفتاحية : الذكاء الاصطناعي ، التعلم العميق ، مصفوفة البوابات المنطقية القابلة للبرمجة

Abstract

This innovative project presents a pioneering research model on an intelligent system for recognition and monitoring of medicinal plants at a biological environment level, encompassing the integration of artificial intelligence (AI) and deep learning (DL) algorithms in parallel with harnessing the immense capabilities of programmable logic gate arrays (FPGA), evidenced by their high performance and versatility of use.
This project showcases a fast real-time recognition and monitoring mechanism with high performance, while maintaining energy efficiency.
This project is poised to facilitate and expedite pharmaceutical research, in addition to exploiting the abundance of medicinal plants, particularly in highlands and deserts, to provide them as raw material for drug and cosmetic production, thereby creating wealth in the pharmaceutical field.

Keywords: Artificial Intelligence (AI), Deep Learning (DL), Programmable Logic Gate Array (FPGA).

Résumé

Ce projet innovant propose un modèle de recherche novateur sur un système intelligent de reconnaissance et de surveillance des plantes médicinales au niveau d'un environnement biologique, en intégrant des algorithmes d'intelligence artificielle (IA) et d'apprentissage profond (DL) en parallèle avec l'exploitation des capacités massives de la matrice des portes logiques programmables (FPGA), se manifestant par une haute performance et une polyvalence d'utilisation.

Ce projet présente un mécanisme de reconnaissance et de surveillance rapide en temps réel avec une performance élevée, tout en préservant l'efficacité énergétique.

Ce projet contribuera à faciliter et accélérer la recherche pharmaceutique, ainsi qu'à exploiter l'abondance des plantes médicinales, en particulier dans les plateaux et les déserts, afin de les utiliser comme matière première pour la production de médicaments et de produits cosmétiques, créant ainsi de la richesse dans le domaine pharmaceutique.

Mots clés: Intelligence artificielle (IA), Apprentissage profond (DL), matrice des portes logiques programmables (FPGA) .

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
ANN	Artificial neural networks
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
FPGA	Field Programmable Gate Array
PLDs	Programmable Logic Devices
CLB	Logic Blocks
VHDL	VHSIC Hardware Description Language
VGA	Video Graphic Arry
MAC	Multiplier and Accumulator
PWL	Piece Linear Approximation
$\sum$	Addition
ϵ	Epsilon
$f'(x)$	Derivative
$\frac{\partial f}{\partial x}$	Partial Derivative
∇	Nabla

General Introduction

Within the framework of the agreement signed between the Ministries of Higher Education and Scientific Research on one hand, and the Ministry of Pharmaceutical Industry on the other hand, in late December 2022, under this agreement, it is envisaged to commence joint research, development, and innovation activities aimed at reducing imports and creating wealth in the pharmaceutical field, in addition to the abundant availability of medicinal plants, especially in the highlands and the desert, in order to provide them as raw materials for drug production. It has become imperative for Algerian products to seek new ideas that facilitate the production process. Therefore, there is a need to focus on scientific research and development in this field. Investment in research and development is crucial, particularly through the establishment of supportive and research institutions in this field, especially with projects to establish university incubators to advance sustainable development.

In our research, we used the VGA protocol and the convolutional neural network (CNN) algorithm in the field of computer vision and deep learning.

This integrated approach allowed us to achieve promising results in the field of computer vision by leveraging the complementary strengths of the VGA protocol and convolutional neural networks.

Research objectives

The objective of this dissertation is to create a CNN model using the Mimas V2 FPGA board. The model will be capable of recognizing images and classifying them into a single category, indicating whether these images of medicinal plants are diseased or not.

Structure of the Thesis:

After this general introduction, the remainder of our work is structured as follows :

The first chapter :

In this first chapter, we will present an overview of the fundamental concepts necessary to understand our project. We will begin by defining artificial intelligence, explaining its basic principles, subdomains, and common applications. Next, we will discuss FPGAs, highlighting their distinctive features, such as reprogrammability and parallel processing, which make them particularly well-suited for computationally demanding tasks like image processing. Finally, we will explore image processing techniques, which are essential for extracting relevant information from images of medicinal plants.

The second chapter :

In this chapter, we will delve into the details of FPGA (Field-Programmable Gate Arrays) and their significance in our project. We will begin with a general overview of FPGAs, explaining their architecture and functionality. Following that, we will examine the Mimas V2 board. We will then continue with a discussion on the VGA protocol, utilized for graphical display, and its integration on FPGAs. Lastly, we will introduce Convolutional Neural Networks (CNNs) and their architecture.

The third chapter:

In the third and final chapter, we will move on to practical application by implementing an example on the subject: simulating the CNN algorithm on Xilinx and the results obtained.

CHAPTER 1

General Overview of Artificial Intelligence

1.1 Introduction

Artificial Intelligence (AI) is a field of computer science and computer systems that emphasizes frameworks to perform tasks that conventionally are perceived as requiring human cognition and intelligence. It is an industry that has been progressing and integrating into our daily lives through the technologies we use. With recent scientific developments, Machine Learning (ML) and Deep Learning (DL) have been prominent names associated with AI. While there is an overlap among all of them, they are not the same [18].

This chapter will give a short overview on how Artificial Intelligence can be defined, the condensed history of Artificial Intelligence, and different techniques to develop AI. It will also explain the relationship between the three main concepts within AI: machine learning (ML), and deep learning (DL). Additionally, the chapter will explore the advantages of using FPGAs in AI, particularly in the context of image processing, and provide a classification for their application in AI.

1.2 Definition of artificial intelligence

Human intelligence "can be so clearly described that a machine can be constructed to imitate it," according to the field's founders. This sparked philosophical debates about the mind and the ethical implications of building artificial intelligence that is human-like; these concerns have long been explored by myth, fiction, and philosophy. If AI's logical capacities are not supervised, science fiction writers and futurologists have claimed that it could constitute an existential threat to humanity. [14]

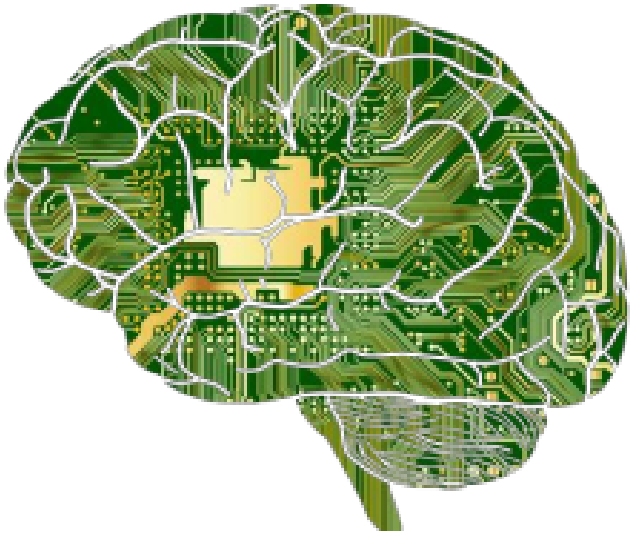

Figure 1.1: Artificial intelligence simulation on the human brain [14].

The term AI is often used to describe machines that mimic human cognitive functions such as learning, understanding, reasoning or problem-solving. AI has two main dimensions, as shown in Table 1 below. The AI definitions on top of the table relate to processes and reasoning, whereas the ones at the bottom side address behaviour. The definitions on the left side of the table measure success in terms of fidelity to human performance, whereas the ones on the right-side measure against an ideal concept of intelligence and rationality. [25]

Table 1.1: Some definitions of artificial intelligence

Thinking Humanly	**Thinking Rationally**
"The exciting new effort to make computers think . . . machines with minds, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . ." (Bellman, 1978)	"The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly	**Acting Rationally**
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"Computational Intelligence is the study of the design of intelligent agents." (Poole et al., 1998) "AI . . . is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

1.3 The brief history of artificial intelligence

In this figure, we will provide an overview of the history of artificial intelligence:

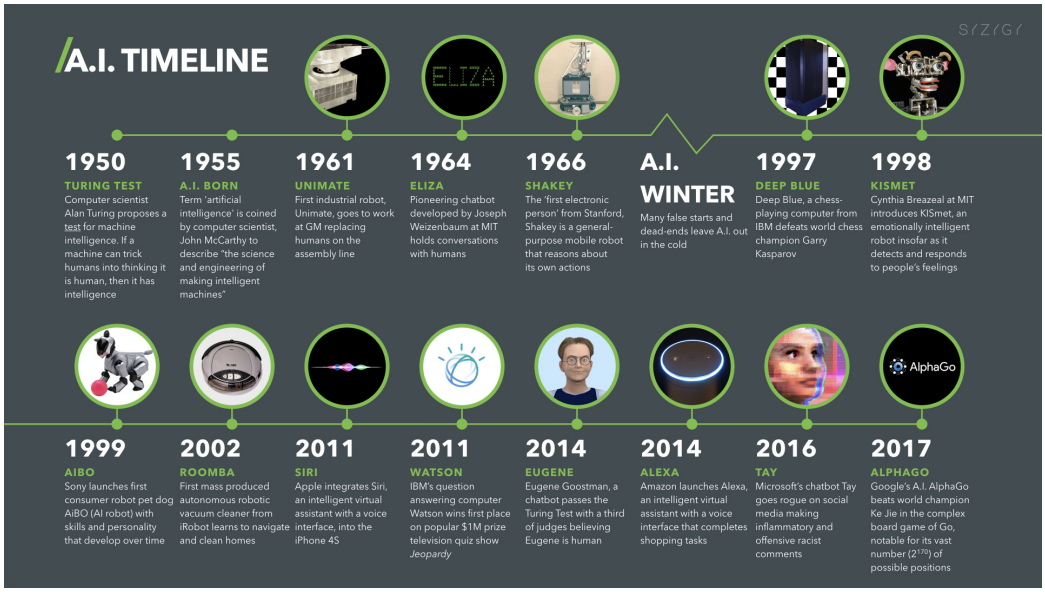

Figure 1.2: Artificial-Intelligence-AI-Timeline-Infographic [20].

1.4 the Main Fields of AI

- Machine Learning
- Deep Learning
- Computer Vision
- Robotics
- Natural Language Processing
- Speech Recognition

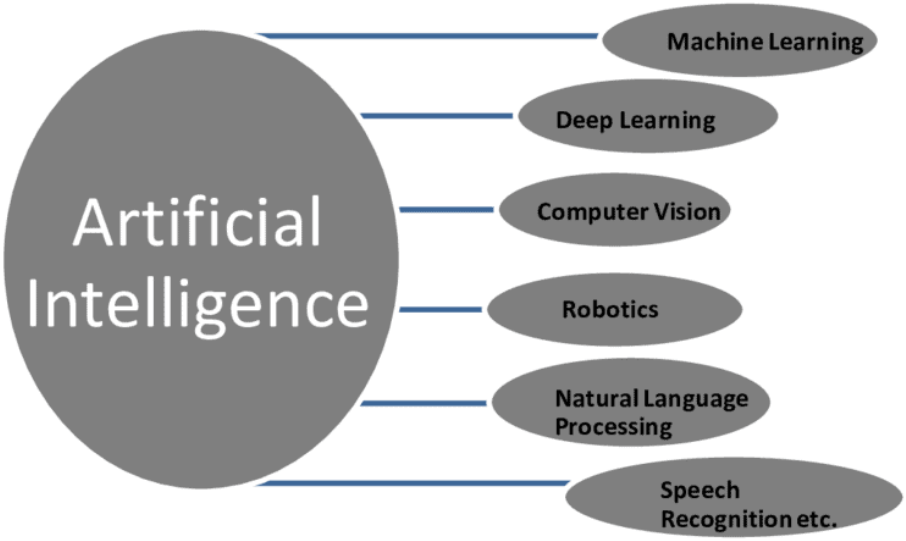

Figure 1.3: Sub-fields-of-Artificial-Intelligence.[google source]

1.5 Machine learning

Machine learning (ML) is a branch of artificial intelligence that systematically applies algorithms to synthesize the underlying relationships among data and information. For example, ML systems can be trained on automatic speech recognition systems (such as iPhone's Siri) to convert acoustic information in a sequence of speech data into semantic structure expressed in the form of a string of words. ML is already finding widespread uses in web search, ad placement, credit scoring, stock market prediction, gene sequence analysis, behavior analysis, smart coupons, drug development, weather forecasting, big data analytics, and many more applications. ML will play a decisive role in the development of a host of user-centric innovations. [1]

1.5.1 Types of Machine Learning

Many different types of Machine Learning techniques have been developed to solve problems in various fields. These Machine Learning techniques can be classified into three types depending on the training method .

- **Supervised learning:**
  Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. [17]

- **Unsupervised learning:**
  These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data. The unsupervised learning algorithms learn few features from the data. When new data is introduced, it uses the previously learned features to recognize the class of the data. It is mainly used for clustering and feature reduction. [17]

- **Reinforcement learning:**
  Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize some notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning. [17]

1.6 Deep learning

In general, Artificial Intelligence, Machine Learning, and Deep Learning are related as follows: “Deep Learning is a kind of Machine Learning, and Machine Learning is a kind of Artificial Intelligence.”as shown in Figure [1.4].

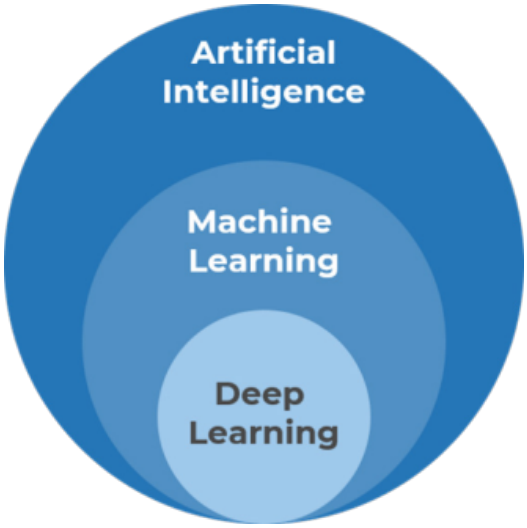

Figure 1.4: Artificial intelligence, machine learning, and deep learning.[synoptek source]

Deep learning is a subset of machine learning that focuses on neural networks with many layers (hence "deep") to model and understand complex patterns and representations in data. It allows computers to build complex concepts out of simpler concepts by stacking multiple layers of neurons, each layer transforming the input data into a more abstract and composite representation. This hierarchical structure enables deep learning models to excel at tasks such as image and speech recognition, natural language processing, and more. [16]

1.6.1 Types of Deep Learning Networks

This paragraph discusses the most famous types of deep learning networks, including Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). CNN will be explained in detail in Chapter 2 due to the importance of this type, as it is the most widely used in many applications among other networks.
Recurrent neural networks: Unlike conventional networks, RNN uses sequential data in the network. Since the embedded structure in the sequence of the data delivers valuable information, this feature is fundamental to a range of different applications.

1.7 AI on FPGA

With the development of science and technology, artificial intelligence is becoming more perfect. FPGA,as an artificial intelligence chip, has also received more attention. It has become an essential part of this field and is widely used in industry and life.

1.8 The unique advantages and applications of FPGA in artificial intelligence

1.8.1 Flexible and configurable

In the practical application of deep learning neural network, in a scene with very high real-time requirements, the critical factor affecting the delay is to process the image quickly. FPGA can flexibly change the circuit structure by using its flexibility. Make hardware features take full advantage of their performance. For example, weight data is a necessary factor in the calculation process of the neural network. The data of each layer may be the same, so weight sharing and reducing the amount of data storage are effective means to speed up the running speed of FPGA. The FPGA can also increase the batch processing capability, reduce the bandwidth without affecting the accuracy, and use the parallelism of the FPGA to process multiple data simultaneously. Similar methods include light weights, compact networks ..etc. [33]

1.8.2 Special Optimizations for Convolutional Neural Networks

Computational operations and floating-point calculations are the basic operating principles of convolutional neural networks. The storage characteristics of neural networks can be enhanced by designing FPGA computing units, bandwidth, and local memory, thereby improving the performance of FPGAs at various levels and dimensions. Give full play to the efficiency and performance of the AI computing stage [33].

1.8.3 Deterministic low latency

Most application scenarios of artificial intelligence deep learning are in mobile devices or edge industrial devices, especially real-time monitoring. The highest requirements in this field are drones or autonomous driving. The time delay of reasoning will affect the response time and distance of car braking. Since FPGA has flexible and customizable I/O, it can ensure the provision of deterministic low-latency I/O. And FPGA can provide deterministic system delay, make full use of the parallelism of the chip, and reduce computing delay. Therefore, it can meet the needs of artificial intelligence. Deep learning has high requirements for real-time performance, and its performance can be better than that of humans. This is the most basic requirement for real-time performance in drones or autonomous driving [33]

1.9 The development of FPGA in the future

Looking back at the development history of FPGA for decades, it has always followed Moore's Law in the chip industry. Whether it is from its structure, application scenarios, or development tools, there will be new changes almost every once in a while. Looking forward to the future, the application of artificial intelligence and high-level synthesis will be the development direction of FPGA

1.10 Image Processing in Artificial Intelligence

Before we discuss into using artificial intelligence for image processing, we need to know what an image and a digital image are, as well as the characteristics of digital images.

1.10.1 Digital image

An image may be defined as a two-dimensional function, f (x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of f are all finite, discrete quantities, we call the image a digital image. [6]

1.11 The characteristics of a digital image

The image is a structured set of information characterized by the following parameters:

1.11.1 The Pixel

The pixel is the size of the smallest element of the image. It also refers to a point in the image matrix. The pixel can have a dimension associated with the spatial resolution of the image; indeed, it can be seen that the image is composed of a multitude of small squares.

1.11.2 The Resolution

Resolution is the number of points (pixels) that make up the image, that is, its "digital dimension" (the number of columns of the image multiplied by its number of rows).

1.12 Image Processing

image processing is the manipulation of an image in order to enhance it or extract information from it.

The major focus of digital image processing is on two things:

- Enhancement of image data for human evaluation;
- Image data processing for communication, caching and representation for uncontrolled machine perception . [31]

1.13 Fundamental Steps in Digital Image Processing.

- **Image Acquisition**:This step involves getting the image that needs to be processed. The image can be acquired using sensor strips, sensor arrays, etc
- **Image Enhancement**: This phase enhances the quality of the captured image to extract hidden information from it for further processing.
- **image Restoration**: It is an objective process that improves the image appearance by making use of probabilistic and mathematical models of image degeneration.

The fundamental steps involved in digital image processing are shown in Figure 1.5.

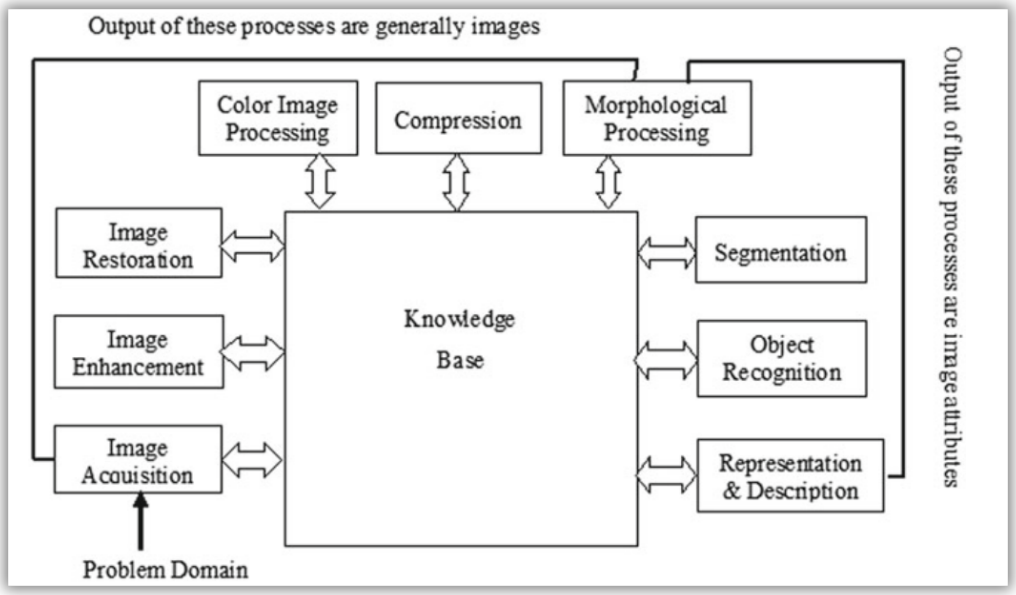

Figure 1.5: Fundamental steps in digital image processing. [6]

1.14 Image Classification

Image classification is the task of assigning a label or class to an entire image. Images are expected to have only one class for each image. Image

classification models take an image as input and return a prediction about which class the image belongs to.

1.15 Image classification motivations

The goal of image classification is to develop a system that can assign a class automatically to an image. Thus, this system makes it possible to carry out an expertise task that can be costly to acquire for a human being, particularly because of physical constraints such as concentration, fatigue or the time required by a large volume of image data.
The applications of automatic image classification are numerous and range from document analysis to medicine to the military field. Thus, I find applications in the medical field such as the recognition of cells and tumors handwriting recognition for checks postal codes. In the field of biometrics such as face recognition, fingerprints, irises. The common point for all these applications is that they require the establishment of a processing chain from the available images composed of several steps to attend to a decision. Each step of the implementation of such a classification system requires the search for appropriate methods for optimal overall performance, namely the feature extraction phase and the learning phase. Typically, I have image data from which I need to extract relevant information translated in the form of digital vectors. This extraction phase allows us to work in a digital space. It is then necessary to elaborate in the learning phase, from these initial data, a decision function for deciding the membership of a new datum to one of the classes in the presence. [4]

1.16 Conclusion

In this chapter, we defined the concept of artificial intelligence (AI) and explained the relationship between the three concepts of AI, machine learning (ML), and deep learning (DL), which are all part of AI. We also discussed the advantages of FPGAs in AI, particularly in image processing, and their classification for use in AI.

CHAPTER 2

THE DESIGN OF CNNS FOR IMAGE CLASSIFICATION.

2.1 Introduction

Deep learning belongs to the broader family of machine learning methods and currently provides state-ofthe-art performance in a variety of fields, including medical applications. Deep learning architectures can be categorized into different groups depending on their components. However, most of them share similar modules and mathematical formulations.

The basic concepts of deep learning will be presented to provide a better understanding of these powerful and broadly used algorithms. The analysis is structured around the main components of deep learning architectures, focusing on convolutional neural networks and autoencoders. [19]

In this chapter, we are going to give a presentation on FPGAs, in particular the MIMAS v2 board also we will focus on the architecture of convolutional neural networks as will as their algorithms and methods which make it possible to classify images.

2.2 Programmable logic devices

Programmable Logic Devices (PLDs) have a great hardware complexity and constitute one of the most dynamic areas of Microelectronics. These PLDs have a large number of different non-excluding concepts and require CAD tools for digital systems design [21]. They include three types of devices FPGA, SPLD and the CPLD. The SPLD are the simplest structures in the PLDs family, their simplicity resides in circuits architecture comparing to FPGA and CPLD who have more complexity.

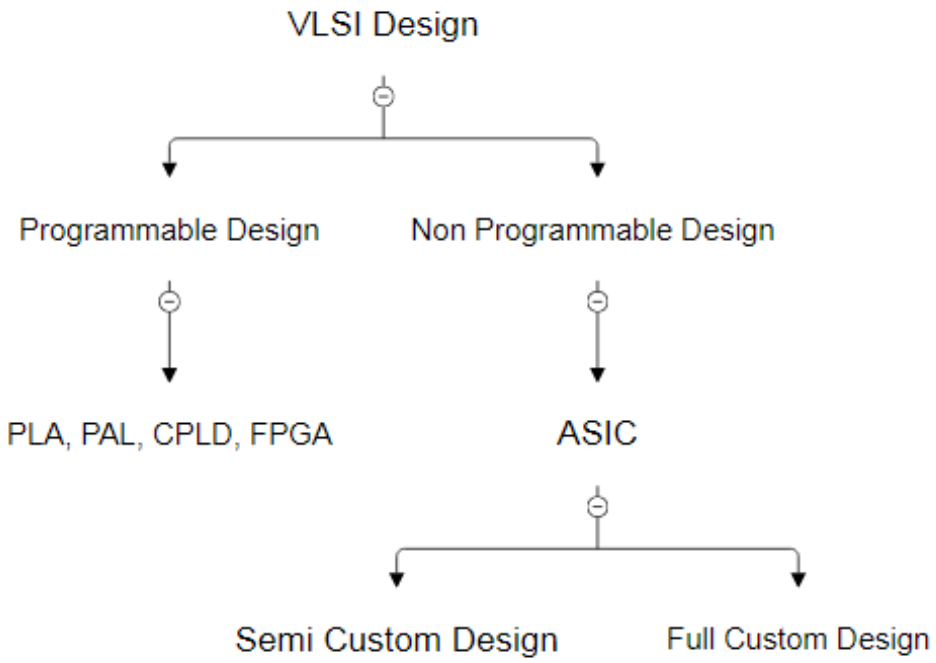

Figure 2.1: Classification of VLSI circuits(google source).

2.2.1 FPGAs

Field-programmable gate arrays (FPGAs) is an integrated circuit that can be programmed or reprogrammed, designed for the implement of logic and digital circuits. These FPGAs have been developed from programmable logic devices (PLDs), programmable array logic (PAL), generic array logic (GAL) and another programmable logic device [26]. FPGAs can be considered as a programmable printed circuits board that interconnect tens or hundreds standard logic circuits (called blocks, cells or modules) to fulfill a custom design - all on a single chip [8]. FPGAs have more flexibility than application specific integrated circuit (ASIC), and effectively solve the limitation of finite number of programmable logic gates. Their emergence has given users great convenience [26]. Users no longer need to use expensive ASICs to implement circuit functions, and the reprogrammable nature of FPGA allows users to change their logic design at any time to meet different application scenarios. FPGAs have greatly promoted the development of integrated circuit technology and pointed out the direction for the future development of integrated circuit industry [9].

2.2.2 Internal architecture of FPGAs

FPGAs have different internal architectures by reasons of the variety of the manufactures, generally they are consisted of the following elements:

- **Logic Blocks (CLB):** These are the basic blocks in FPGA, containing look-up tables (LUTs), flip-flops and multiplexers. LUTs allow im-

plementing logical operations while flip-flops store and transmit the results of these operations.

- **Programmable Interconnect Point (PIP):** The interconnects are the network of wire and programmable switches that connect CLBs.

- **Input/output Blocks:** These blocks facilitate communications between FPGA and external devices.

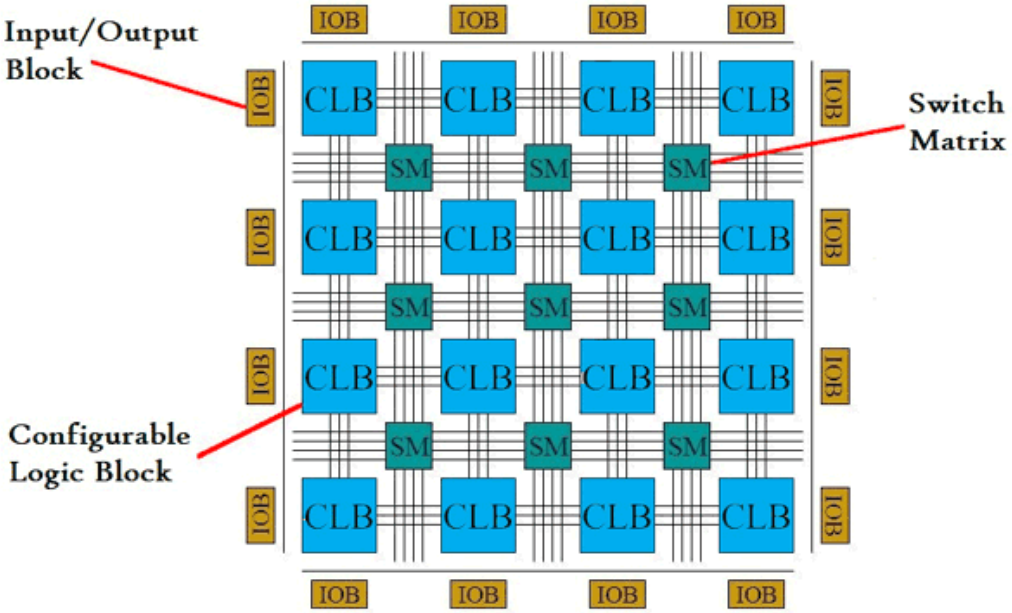

Figure 2.2: FPGA-Structure(google source).

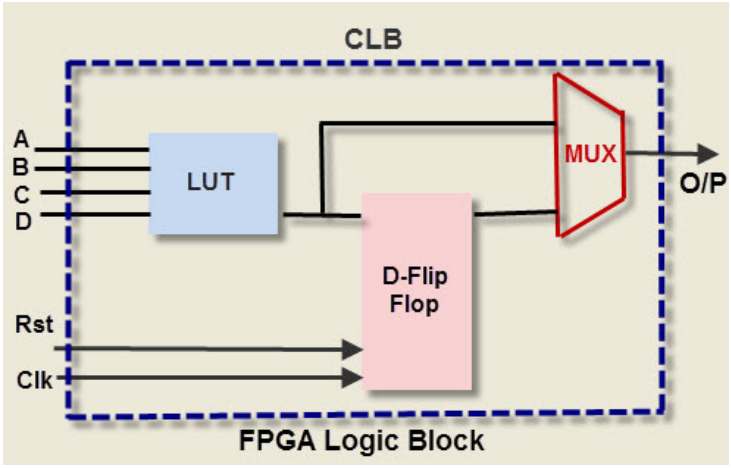

Figure 2.3: CLB Architecture(google source).

Modern FPGAs also contain hundreds components for commonly used functions such as full processor cores, communication cores, arithmetic cores, and block RAM (BRAM). In addition, current FPGA trends are tending toward a system-on-chip (SoC) design approach, where ARM coprocessors and FPGAs are commonly found on the same fabric [7].

2.2.3 FPGAs Technologies

FPLs are found in virtually all memory technologies: SRAM, EPROM, E2PROM, and the most antifuse [11].The specific technology defines whether the device is reprogrammable or one-time programmable. Most SRAM devices can be programmed by a single-bit stream that reduces the wiring requirements, but also increases programming time (typically in the ms range). SRAM devices, the dominate technology for FPGAs, are based on static CMOS memory technology, and are re- and in-system programmable. They require, however, an external "boot" device for configuration. Electrically programmable read-only memory (EPROM) devices are usually used in a one-time CMOS programmable mode because of the need to use ultraviolet light for erasure. CMOS electrically erasable programmable read-only memory (E2PROM) can be used as re- and in-system programmable. EPROM and E2PROM have the advantage of a short setup time. Because the programming information is not "downloaded" to the device, it is better protected against unauthorized use. A recent innovation, based on an EPROM technology, is called "flash" memory. These devices are usually viewed as "pagewise" in-system reprogrammable systems with physically smaller cells, equivalent to an E2PROM device [3].The characteristics of these devices are summarized in the Table.1.

Table 2.1: Comparaison de différentes technologies [3]

Technologie	SRAM	EPROM	EEPROM	Antifuse	Flash
Repro-grammable	yes	yes	yes	-	yes
In-system Programmable	yes	-	yes	-	yes
Volatile	yes	-	-	-	-
Copy protected	-	yes	yes	yes	yes
Exemples	Xilinx Spartan Altera Cyclone	Altera MAX5K Xilinx XC7K	AMD MACH Altera MAX 7K	Actel ACT	Xilinx XC9500 Cypress Ultra 37K

2.3 FPGA Development tools

In our project, we have used the last version of Xilinx ISE which supports SPARTAN 6.

2.3.1 ISE Design Suite 14.7

ISE Design Suite 14.7 is a version of the software tool developed by Xilinx, specially designed to synthesize and analyze HDL designs in FPGA and CPLD. It consists various options,including the Embedded Development Kit(EDK), a Software Development Kit(SDK) and ChipScope Pro that can fabricate complex electronic circuits and develop system.

2.3.2 VHDL language.

VHDL is a hardware description language. It describes the behavior of an electronic circuit or system, from which the physical circuit or system can then be attained (implemented) [24].
VHDL stands for VHSIC Hardware Description Language. VHSIC is itself an abbreviation for Very High Speed Integrated Circuits, an initiative funded by the United States Department of Defense in the 1980s that led to the creation of VHDL. Once the VHDL code is declared, it can be used either to implement the circuit in a programmable device (from Altera, Xilinx, Atmel, etc.) or can be submitted to a foundry for fabrication of an ASIC chip. Currently, many complex commercial chips (microcontrollers, for example) are designed using such an approach [24].

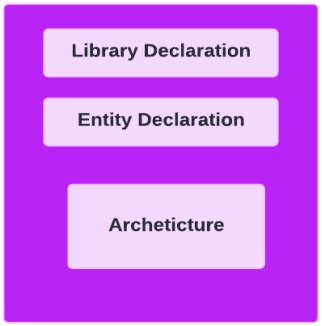

Figure 2.4: Fundamental VHDL Units.

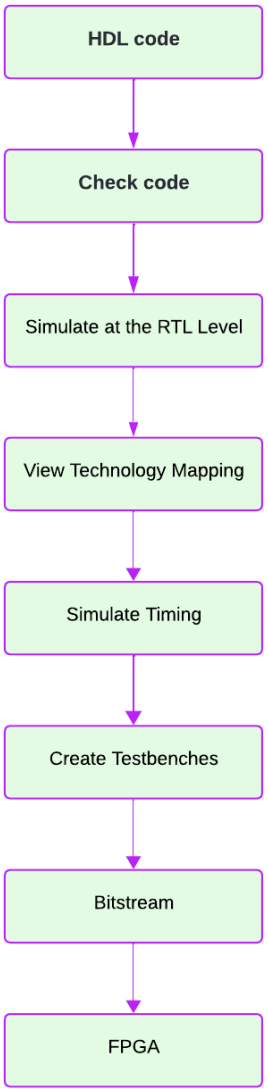

Figure 2.5: Xilinx ISE Flow.

2.4 Study of the Mimas v2 board

In our project, we are using the Mimas v2 board which is presented in Figure below. The choice of this card depends on the specific needs of our project in term of the flexibility and the parallelism. This board is also a great choice for the development of product prototypes.

2.4.1 Mimas v2

MIMAS V2 is a feature-packed yet low-cost FPGA Development board featuring AMD Spartan-6 FPGA. it's specially designed for experimenting and learning system design with FPGAs. This development board features AMD SPARTAN XC6SLX9 CSG324 FPGA with onboard 512Mb DDR SDRAM. The USB 2.0 interface provides fast and easy configuration down-

load to the onboard SPI flash. No need to buy an expensive programmer or special downloader cable to download the bitstream to the board [22].

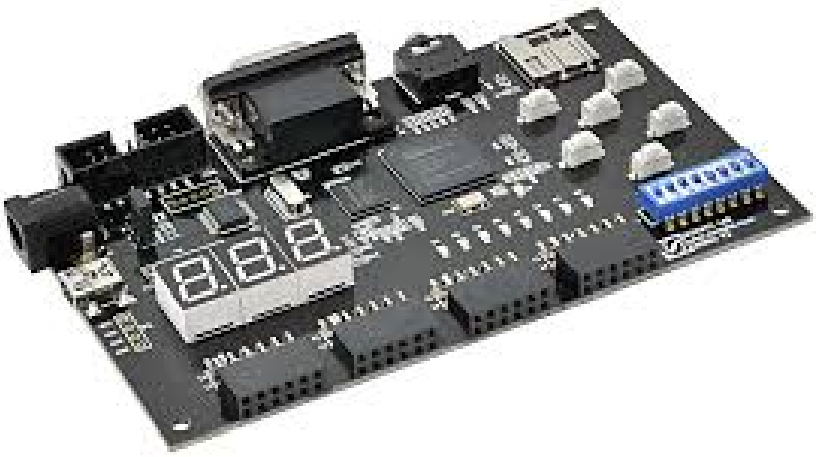

Figure 2.6: Mimas v2 board [22].

2.4.2 Features board

- FPGA: AMD Spartan XC6SLX9 in CSG324 package.
- DDR Memory: 166MHz 512Mb LPDDR (MT46H32M16LF/W949D6CBHX6E).
- Flash memory: 16 Mb SPI flash memory (M25P16).
- USB 2.0 interface for On-board flash programming.
- FPGA configuration via JTAG and USB.
- 8 LEDs, Six Push Buttons, and 8 way DIP switch for user-defined purposes.
- VGA Connector.
- Stereo Jack.
- Micro SD Card Adapter.
- Three-Digit Seven Segment Displays.
- 32 IOs for user-defined purposes.
- Four 6×2 Expansion Connectors.
- Onboard voltage regulators for single power rail operation.

2.4.3 Components of the Mimas v2 board

The following figure summarizes all parts of Mimas v2 board:

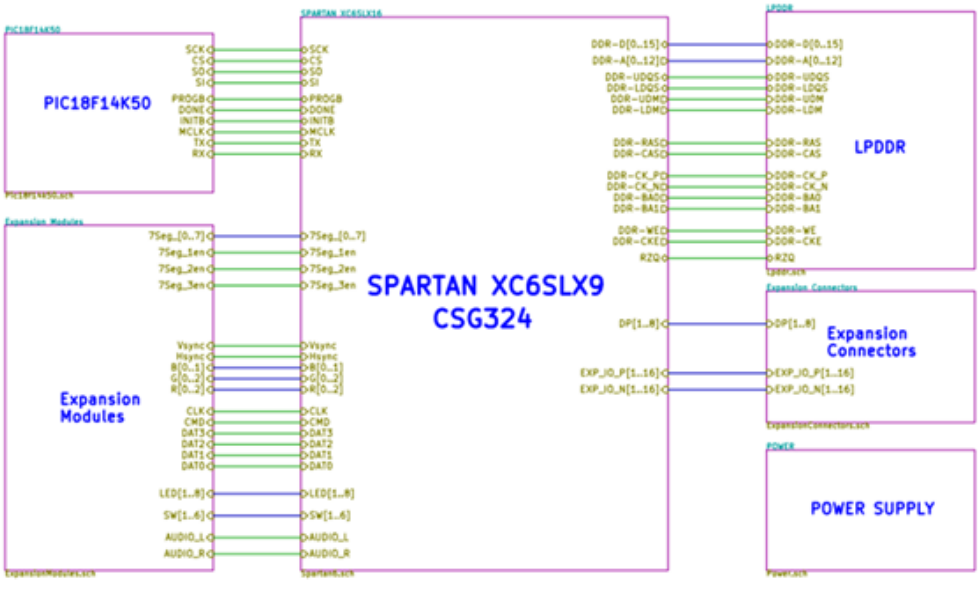

Figure 2.7: Schema of Mimas v2 spartan 6 FPGA components [22].

- **PIC18F14K50**
  It's an 8-bit microcontroller from the PIC18 family manufactured by Microchip technology. It gives us the permission to control USB-UART communication between the code in FPGA and all applications running on PC.

- **SPARTAN XC6LX9 CSG324**
  SPARTAN 6 is the heart of the Mimas v2 board, manufactured by Xilinx, designed for embedded systems, signal processing and other applications.

Table 2.2: Spartan-6 XC6SLX6 Features [24].

Feature	Description
Slices	1,430
Logic Cells	9,152
CLB Flip-Flops	11,440
Maximum Distributed RAM (Kb)	90
Block RAM (18 Kb each)	32
Clock Management Tiles (CMT)	2
Maximum Single-Ended Pins	200
Maximum Differential Pairs	100
DSP48A1 Slices	16
Memory Controller Blocks	2

- **Expansion modules**
  Expansion modules represent the components we need in our project such as the sensors, actuators, etc.
- **LPDDR**
  LPDDR, also known as LPDRAM, stands for "low-power double-data rate memory" and commonly used in mobile devices. Micron's LPDDR memory solutions support speedy, high-band-width data rates without compromising power efficiency [23].
- **Expansion connectors**
  There are several types of connectors in Mimas v2 which make possible to manage the transmission of signals, power supply, data, etc. Among these connectors:
  - **USB Connector**
    The USB connector allows you to upload the code from Pc to the FPGA.
  - **GPIO Connector**
    This board contains 32 users, IO pins that can be used for various custom applications [**?**]. Through this connector, we can interface mimas v2 board with external devices.
  - **TAG Connector**
    The JTAG connector provides access to FPGA's JTAG pins [22]. It is used for downloading the bitstream into FPGA to configure it.
  - **Display Connector**
    The Mimas v2 board contains a VGA connector that gives you the access to connect external displays or monitors to this board.
- **Power supply**
  MIMAS V2 can be powered directly from USB port and we have to use a USB port that can power the board properly. It is recommended to connect the board directly to the PC instead using a hub. It is practically very difficult to estimate the power consumption of the board, as it depends heavily on our design and the clock used. XILINX provides tools to estimate the power consumption, an external supply can be applied to the board in case we cannot offer enough power from USB. For instance, external battery can be used as an external power source. MIMAS V2 requires two different voltages, a 3.3V and a 1.2V supply. On-board regulators divide these voltages from the USB/Ext power supply [23].

2.5 VGA protocol

The video graphic array (VGA) is a group of graphics display system, fonded by IBM campany in 1987. It is mostly used for computer monitors,

with a high-definition resolution video standard. It has the ability to transmit a sharp detailed image. VGA uses separate wires to transmit the three color component signals, vertical and horizontal synchronisation signals. Red, green and blue are the main three signals which send color information to VGA monitor [11].

Table 2.3: 3-Bit Display Color Codes [11].

VGA_R	VGA_G	VGA_B	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Pink
1	1	0	Yellow
1	1	1	White

2.5.1 VGA Port

The Mimas v2 board included a VGA connector. This connector provides us a connection of VGA monitor to the FPGA. It is composed of 15 pins, as illustrated in the Figure 2.7.

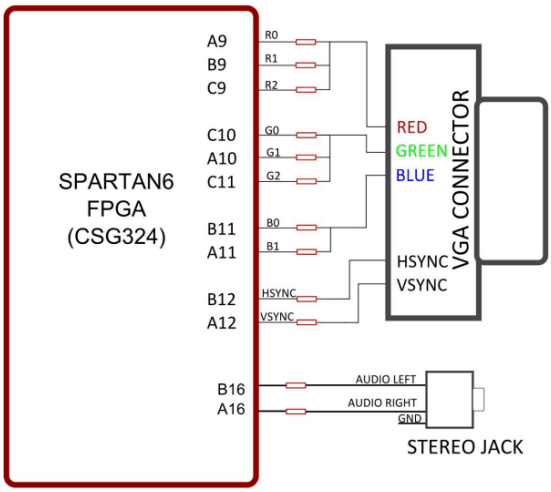

Figure 2.8: VGA connector with FPGA [22].

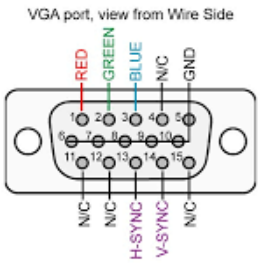

Figure 2.9: Vga port(google source).

2.5.2 VGA timing specification

The VGA supports several resolutions each resolution has a set of signals that defines and displays an image. Any display consists of VGA signals which is RGB data, and the other one is Synchronisation signals. The RGB data are the basic colors signals who carry the color information for each pixel in the image. The HSync and VSync are the ones who limit

the display area,they synchronize the image display when there is a horizontal pulse and a vertical pulse is a new line and a new frame should be started. The quality of this image depends on the refresh rate and the pixels frequency. Timing specification of some resolutions are summarised in the tables below.

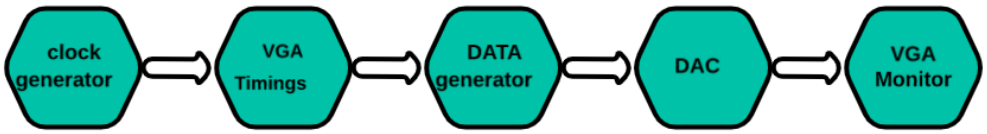

Figure 2.10: Bloc diagram VGA protocol.

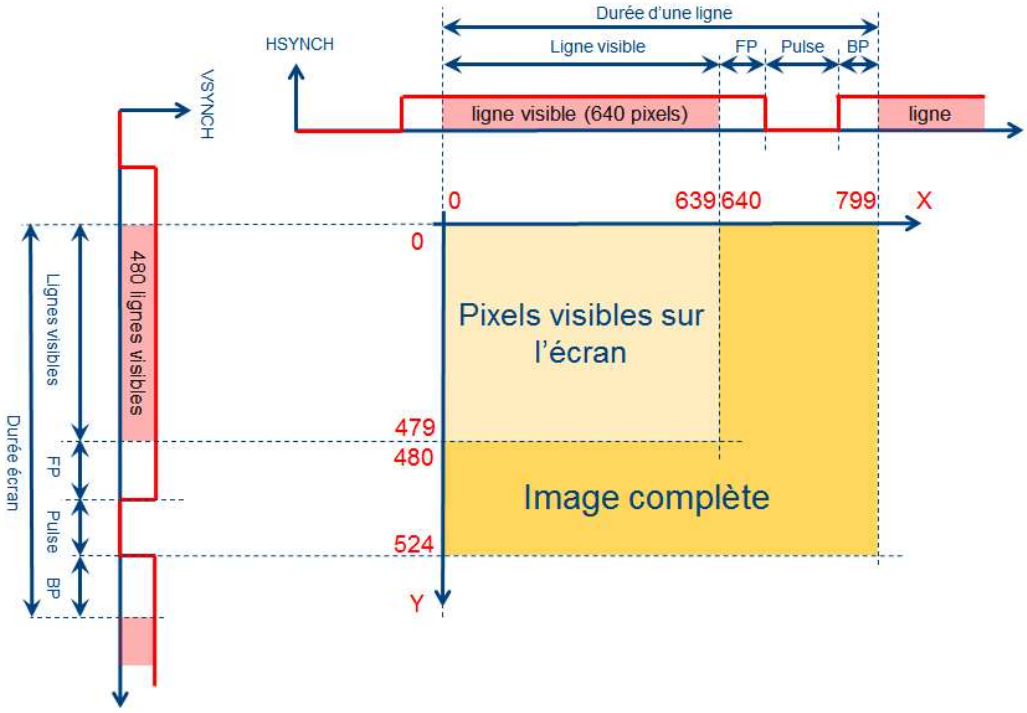

Figure 2.11: VGA synchronisation diagram(google source).

Table 2.4: Examples of VGA modes and corresponding horizontal time parameters[18].

Résolution (HxV)	**Refresh rate (Hz)**	**Clock (MHz)**	**V. LOW (pixels)**	**VBP (pixels)**	**V. HIGH (pixels)**	**VFP (pixels)**
640x480	60	25	96	48	640	16
640x480	75	36	96	48	640	16
800x600	56	40	128	88	800	40
800x600	75	50	160	120	800	40
1024x768	60	65	136	160	1024	24
1280x1024	60	108	112	248	1280	48

Table 2.5: Examples of VGA modes and corresponding vertical time parameters[18].

Résolution (HxV)	**Refresh rate (Hz)**	**Clock (MHz)**	**V. LOW (lines)**	**VBP (lines)**	**V. HIGH (lines)**	**VFP (lines)**
640x480	60	25	2	33	480	10
640x480	75	36	2	33	480	10
800x600	56	40	4	23	600	1
800x600	75	50	3	21	600	1
1024x768	60	6	136	29	768	3
1280x1024	60	3	112	38	1024	1

In the Mimas v2, there are two reference's frequencies 100MHz and 12MHz to generate all VGA resolutions using one of these two frequencies, we need to:

Manage the VGA clock: VGA clock management depends on the reference frequency f_r of the FPGA board.

- If **VGA clock <f_r:** we implement a frequency divider in the FPGA which divides the reference frequency to obtain the pixels clock frequency.

- If **VGA clock >f_r:** the PLL circuit is used to generate the reference frequency, the implementation of a PLL in the FPGA allows you to multiply this frequency to obtain the pixels clock frequency.

Manage the horizontal and the vertical signals: the HSync and the VSync are generated from the pixel clock and are adjusted according to the horizontal settings: Horizontal display, Right border, Sync pulse, Left border and the vertical parameters: Vertical display, Lower border, Sync pulse, Upper border.

2.6 Artificial Neural Networks

An ANN is a ML algorithm based on the concept of a human neuron. It is a biologically inspired computational model, consisting of processing elements (neurons) and connections between them with coefficients (weights) attached to the connections. ANNs are inspired by the brain

structure and for this reason it is important to define the main components under which a neuron, dendrites, cell body, and axon works. Dendrites are a network that carries electrical signals to the cell body. The cell body adds and collects the signals. The axon carries the signal from the cell body to other neurons using a long fiber [2]. In case of artificial neuron the information comes into the body of an artificial neuron via inputs that are weighted (each input can be individually multiplied with a weight). The body of an artificial neuron then sums the weighted inputs, bias and "processes" the sum with a transfer function. At the end an artificial neuron passes the processed information via output(s) [30].

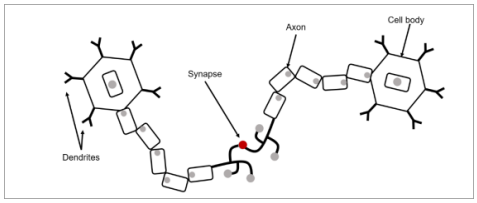

Figure 2.12: Structure of a biological neuron [30].

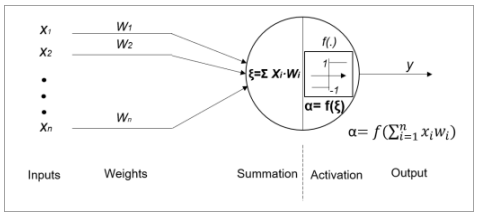

Figure 2.13: The basic scheme of a neuron [2].

2.7 Convolutional Neural Networks

Convolutional Neural Networks (CNNs or ConvNets) are machine learning algorithms and a type of artificial neural networks, commonly used in image related tasks such as image recognition, image classification, and object detection. They are consisted of multiple layers, the convolution layers provide the ability to extract features maps of the input image while fully connected layers classify this image according to these detected features maps.

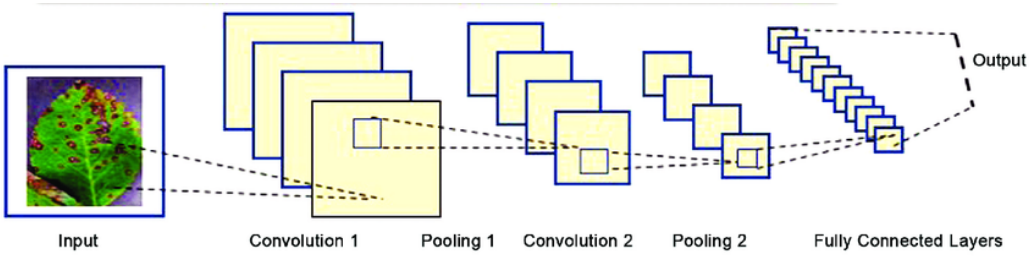

Figure 2.14: Convolution neural network architecture (google source).

2.7.1 Architecture of ConvNet

Convolutional neural networks are composed of the following layers:

- **Convolution layer**

  In image processing, convolution layer is based on a set of filters applied to the input image for extract feature maps, these kernels generally are 3x3 or 5x5 matrices that are sliding over the image with a stride of 1. At each position, we multiply the elements of kernel with the element of image located at the same filter position then we add the obtained values.

$$y = f(\sum_{i=0}^{n} x[n] * k[n]) \tag{1}$$

  y: convolution layer output.
  f: activation function.
  x: input image.
  k: input kernel.

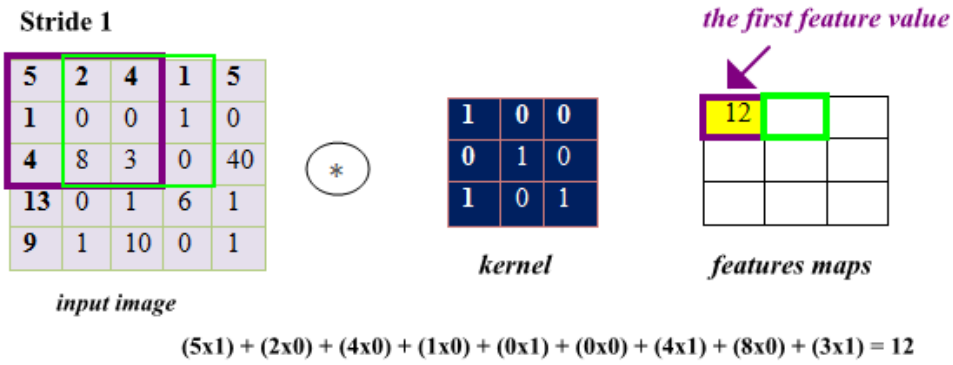

Figure 2.15: The convolution operation.

- **Pooling layer**

  After the convolution layer and the extraction of the feature maps, we apply the pooling layer(subsampling operation). Due to this process, we can reduce the size of feature maps and conserve the important characteristics.

  There are two types of operation pooling:

  - **Max pooling:** The pooling process divides the feature maps into a set of 2x2 regions for each region we choose the maximum value.

  - **Average pooling:** The pooling process divides the feature maps into a set of 2x2 regions for each region we calculate the average.

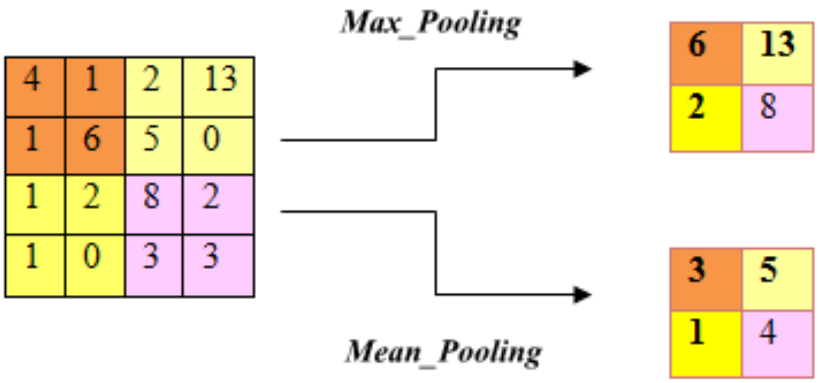

Figure 2.16: The pooling process.

- **Fully connected layer**

  The fully connected layer is the lastest layer in the convolutional neural network,is also known by densely connected layer, each neuron in this layer is connected the neuron in the previous layer, and every neuron in first hidden layer does a multiplication of the weight and the receive input.

  The output of fully connected layer is defined as:

$$z = \sum_{i=0}^{n} x[n] * w[n] + b \tag{2}$$

  x: input of prevision layer.
  w: weight.
  b: bias.

  An activation function is applied to the output z, this activation function depends on the task which are doing. In case where we are doing a binary classification, we can apply a sigmoid function at the output layer.

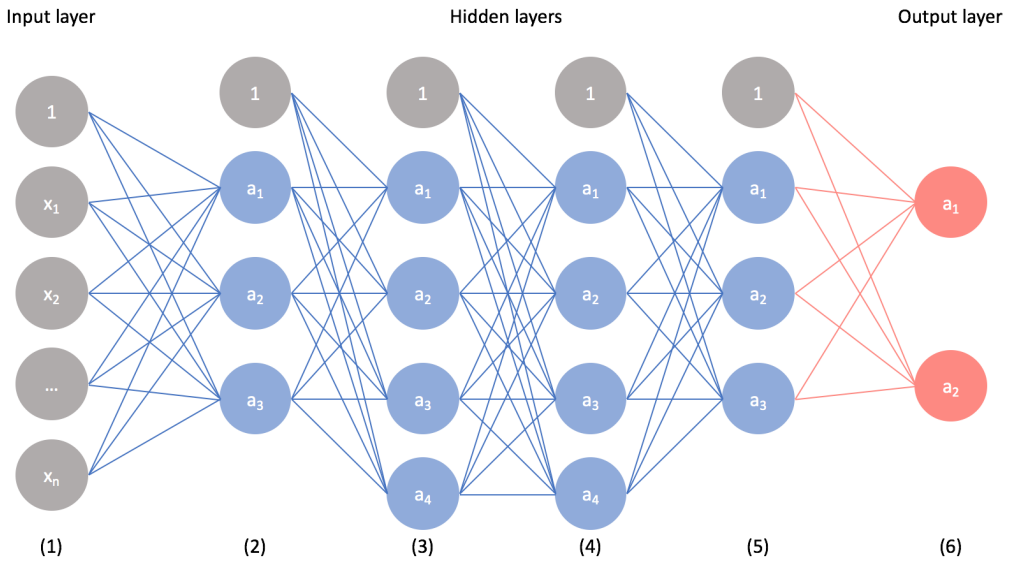

Figure 2.17: Fully connected layer(google source).

All operations performed in the layers of the neural networks depend on basic mathematical operations such as addition and multiplication. For this, we are going to implement parallel multiplication on FPGA using array multiplier and full adders(details in chapter3).

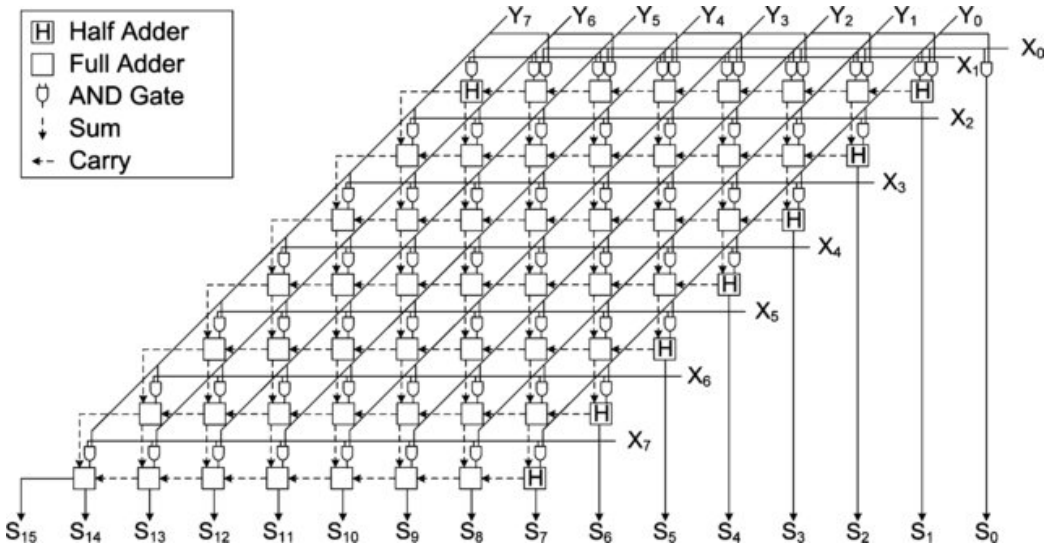

Figure 2.18: 8x8 array multiplier [29].

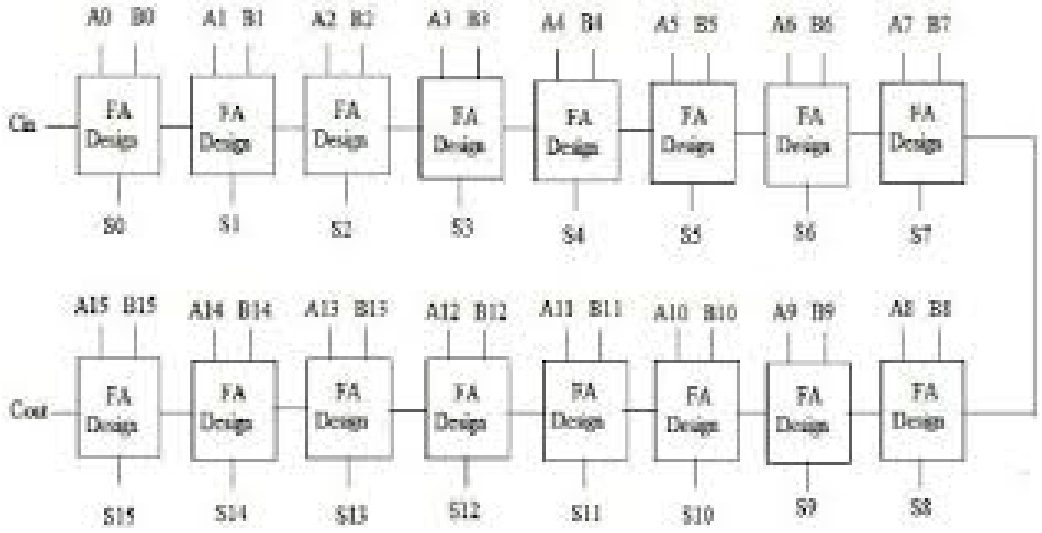

Figure 2.19: 16-bits adder [27].

2.7.2 Gradient Descent algorithm

Most deep learning algorithms involve some form of optimization, which refers to the task of either minimizing or maximizing a function $f(x)$ by changing x. Typically,we frame optimization problems in terms of minimizing $f(x)$, known as the objective function or criterion. When minimizing it, we may also refer to it as the cost function, loss function, or error function [15].

Suppose we have a function $y = f(x)$, where both x and y are real numbers. The derivative $f'(x)$ gives the slope of $f(x)$ at the point x, indicating how a small change in the input affects the output:

$$f(x + \epsilon) \approx f(x) + \epsilon f'(x) \tag{3}$$

The derivative is useful for minimizing a function because it shows how to change x to make a small improvement in y. For example:

$$f(x - \epsilon \operatorname{sign}(f'(x))) < f(x) \tag{4}$$

for a small enough ϵ. Thus, we can reduce $f(x)$ by moving x in small steps opposite to the derivative, a technique called gradient descent [12].

For functions with multiple inputs, we use the concept of partial derivatives. The partial derivative $\frac{\partial f}{\partial x_i}$ measures how f changes as only x_i increases at point x. The gradient generalizes the derivative notion to vectors: the gradient of f is the vector containing all partial derivatives, denoted $\nabla_x f(x)$, each a new point defined [12]:

$$x' = x - \epsilon \nabla_x f(x) \tag{5}$$

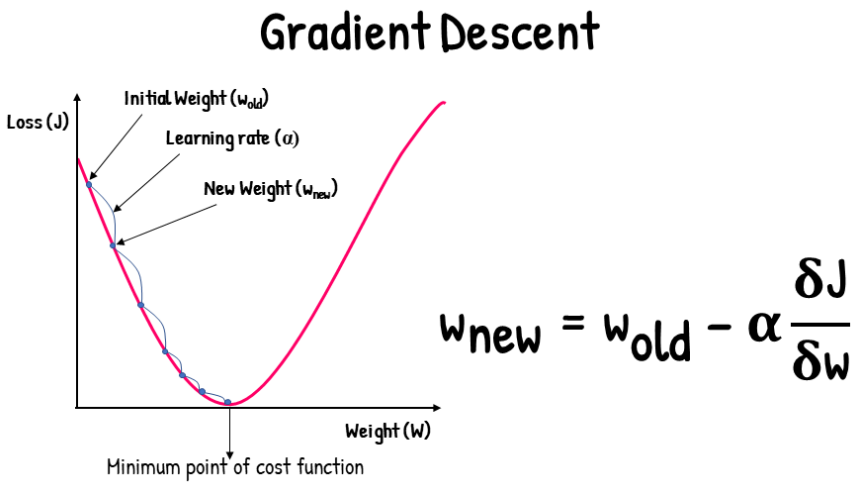

Figure 2.20: Gradient descent(google source).

2.7.3 Backpropagation algorithm

Our goal during training is to find values for the weights and biases in the neural network that will allow it to make effective predictions. For convenience, we group these parameters into two vectors: one for the weights w and one for the biases b [5]. We optimize these vectors using a backpropagation, this technique is based on two phases:

- **Forward Propagation:**
  In this part, the input image is passed through convolutional neural networks layers that we have explained, at each layer we are applied activation function. From the final output, we calculate the loss function:

$$L(W) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{6}$$

  y_i: the observed class.
  $\hat{y}_i$: is the prediction.
  w: the model parameters.

  This function is a Cross entropy loss function, used for binary classification[1] that calculate the difference between the prediction($\hat{y}$) and the actual label.

- **Backward Propagation:**
  To update the weights and biases, we will use the gradient descent algorithm:

$$w' = w - \alpha \frac{\partial L}{\partial weights} \tag{7}$$

[1]Classify an image through two categories.

$$b' = b - \alpha \frac{\partial L}{\partial biases} \tag{8}$$

w': new weight.
w: weight.
b': new bias.
b: bias.
α: learning rate.

In order to apply gradient descent we must calculate the derivative (gradient) of the loss function w.r.t. the model's parameters: [13]

$$L(W) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Predictions are given by:

$$\hat{y}_i = \sigma(z_i) \tag{9}$$

where:

$$z_i = \mathbf{w}^T \mathbf{x}_i + b \tag{10}$$

To calculate the gradient of $L(W)$ with respect to W, we will use the chain rule:

$$\frac{\partial L(W)}{\partial W} = \frac{\partial L(W)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial W} \tag{11}$$

The first term derivative:

$$\frac{\partial L(W)}{\partial \hat{y}_i} = -\left(\frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right) \tag{12}$$

The second term derivative:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{13}$$

$$\frac{d}{dz} \left(\frac{1}{1 + e^{-z}} \right) = \frac{e^{-z}}{(1 + e^{-z})^2} \tag{14}$$

Since $\hat{y} = \frac{1}{1+e^{-z}}$, we can rewrite it as:

$$\frac{d\hat{y}}{dz} = \hat{y}(1 - \hat{y}) \tag{15}$$

The third term derivative:

$$\frac{\partial z}{\partial W} = \mathbf{x} \tag{16}$$

Now, we combine these derivatives:

$$\frac{\partial L(W)}{\partial W} = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})}\hat{y}(1-\hat{y})\mathbf{x} = (\hat{y} - y)\mathbf{x} \tag{17}$$

$$\frac{\partial L(b)}{\partial b} = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})}\hat{y}(1-\hat{y}) = (\hat{y} - y) \tag{18}$$

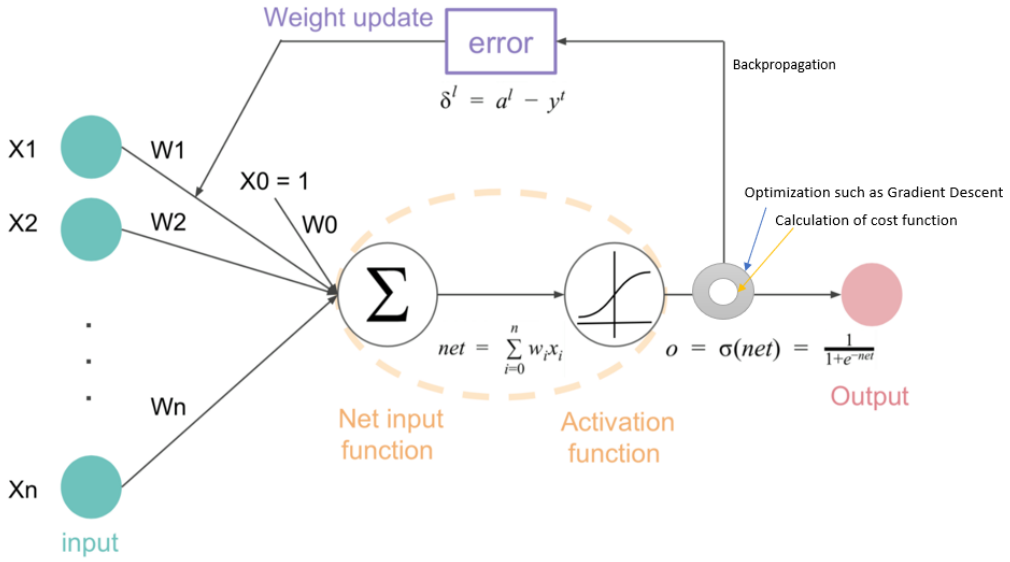

Figure 2.21: The process of backpropagation algorithm(google source).

2.7.4 Activation function

The most important component in the structure of neural networks is their net inputs, which are processed using a scalar-to-scalar function known as the activation function, threshold function, or transfer function. This function outputs a value called the unit's activation. Activation functions limit the amplitude of a neuron's output, often referred to as squashing functions, as they compress the output signal's permissible amplitude range to a finite value. These functions are crucial for solving non-linear problems in neural networks [10].

2.7.5 Types of activation functions

There are different non linear networks activation functions:

– **Sigmoid / Logistic Activation Function**:
Sigmoid function is commonly used in binary classifications in order to make decisions whereas the outputs are between 0 and 1, in instance if the threshold level is greater than 0.5 then the input is classified as mint, it is represented as :

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{19}$$

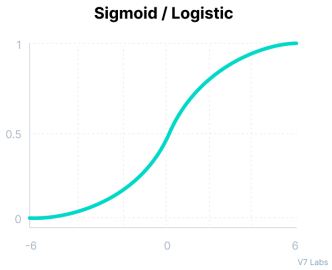

Figure 2.22: Sigmoid function [32].

– **Tanh Function (Hyperbolic Tangent)**:
Tanh function is very similar to the sigmoid/logistic activation function, and even has the same S-shape with the difference in output range of -1 to 1. In Tanh, the larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to -1.0. Mathematically it can be represented as [32]:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{20}$$

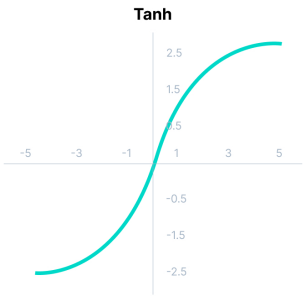

Figure 2.23: Tanh Function (Hyperbolic Tangent) [32].

– **ReLU Function**:
ReLU stands for rectified liner unit and is a non-linear activation function which is widely used in neural network. The upper hand of using ReLU function is that all the neurons are not activated at the same time. This implies that a neuron will be deactivated only when the output of linear transformation is zero. It can be defuned mathematically as [28]:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{21}$$

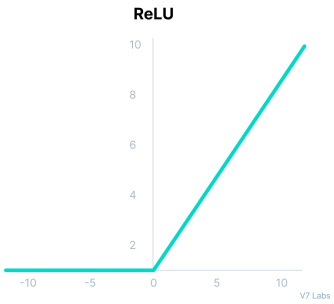

Figure 2.24: ReLu function [32].

– **Softmax Function**:
The Softmax function can be viewed as an extension of multiple sigmoid functions. While a sigmoid function outputs values between 0 and 1, representing probabilities for binary classification, the Softmax function is designed for multiclass classification problems. It returns the probability distribution across multiple classes for each data point. When constructing a neural network for multiclass classification, the output layer will have a neuron corresponding to each target class, enabling the network to predict the probabilities for each class accurately. The Softmax function can be mathematically represented as follows [28]:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{22}$$

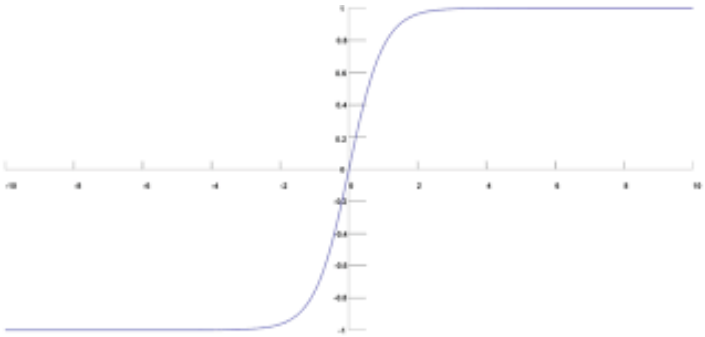

Figure 2.25: Softmax function5(google source).

Table 2.6: The use of activation functions [32].

Problem Type	**Last-layer Output Nodes**	**Hidden-layer activation**	**Last-layer activation**	**Loss function**
Binary classification	1	Relu	Sigmoid	Binary Cross Entropy
Multi-class single-label classification	Number of classes	Relu	Softmax	Categorical Cross Entropy
Multi-class single-label classification	Number of classes	Relu	Sigmoid(one for each class	Binary Cross Entropy
Regression to arbitrary values	1	Relu	None	MSE
Regression to arbitrary values 0 and 1	1	Relu	Sigmoid	MSE/Binary Cross Entropy

2.8 Conclusion

Through this chapter, we studied the MIMAS v2 SPARTAN 6 board, we saw their development environment and their programming language also we focused on the convolutional neural networks and VGA protocol which allow us to classify and display an image.

CHAPITRE 3

System implementation

3.1 Introduction

From what we saw in the previous chapter on the design of CNN in FPGA, architectures and algorithms.

In this chapter, we will present the implementation of the blocks necessary for our proposed system: monitoring of medicinal plants in real time.

The objective of this chapter is to exploit the parallel processing capabilities of FPGA for the purpose of implementing CNN functions and algorithms.

3.2 Number representation

The input to the CNN are generally positive while the weights and biases can be positive or negative, also they have a fractional part. We have two choices to represented these numbers, floating point representation and fixed point representation. the first representation makes it possible to represent large numbers with great precision but this implementation is difficult also consume so mach resources, the other representation is faster and consumes fewer hardware resources but their precision is limited. In our project, we are going to implement the numbers with a fixed point representation because we don't need a large dynamic range where the input value used in cnn are normalized between 0 to 1 or -1 to 1.

Example:
The representation of 15.84 in Q5.10 which 5 bits representing integer part and 10 bits representing fractional part.
$(15)_{10} = (01111)_2$
$(0.84_{10} = (1101011100)_2$
$(15.84)_{10} = (011111101011100)_2$

3.3 CNN Blocks

Our system is based on the following blocks:

3.3.1 Parallel-Parallel multiplier Block

Multiplier is the most important component in CNN, composed by full adder units, half adder units and logic gates AND.
AND logic gates are employed to perform binary multiplication between the input bits of a and b, resulting in partial products. These partial products are added together by half adders and full adders. This structure allows simultaneous and rapid multiplication using these simple and efficient devices.

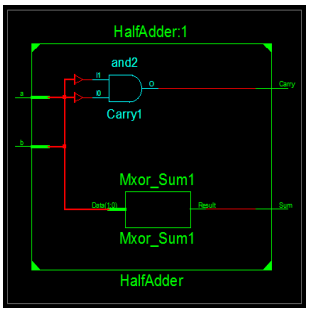

Figure 3.1: Half Adder circuit.

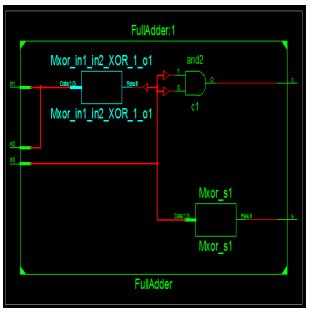

Figure 3.2: Full Adder circuit.

From these circuits, we will implement the 8-bits multiplier:

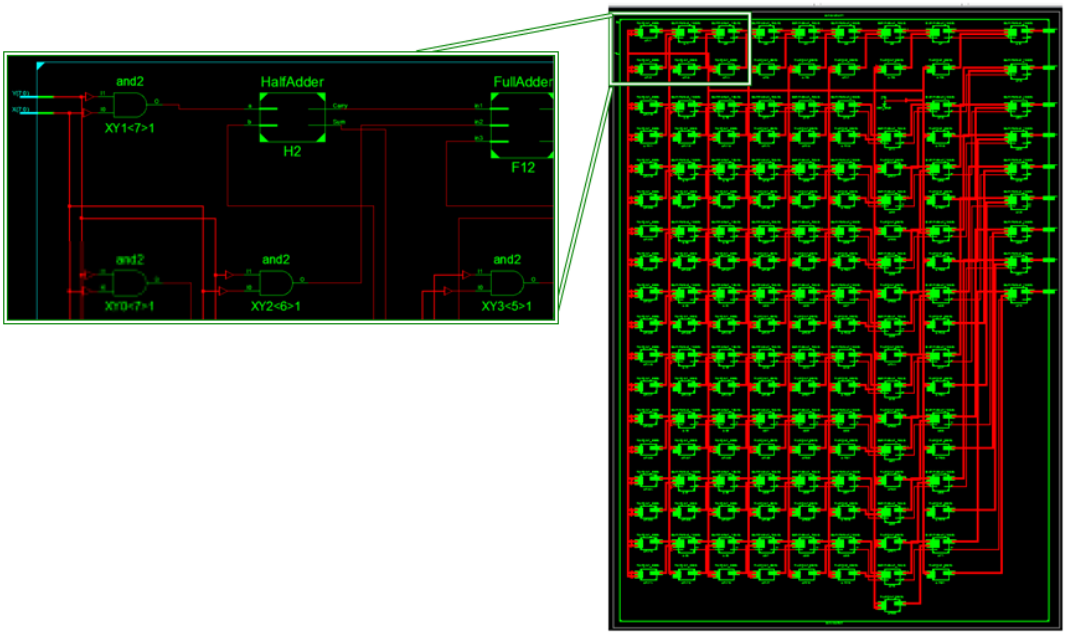

Figure 3.3: Internal architecture of multiplier.

This overall figure describes the multiplier component and the zoomed-in portion of it illustrates a part of its internal architecture where the AND logic gates are used for multiplying bit by bit ($XY1$ and $XY2$ are the inputs of each logic gates) the outputs of these two logic gates are the inputs of the half adder H2 which is responsible for adding the two inputs.
The full adder F12 has three inputs which are the two outputs of the half adder H2 and the output of the logic gate AND $XY3$, the output of this component is the addition of the logic gate AND(xy1 ,xy2,xy3) plus the carry.

3.3.2 Convolution Block

The basic component of convolution are the MACs which compose by multipliers and accumulators.

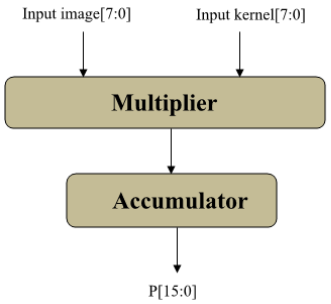

Figure 3.4: Multiplier-Accumulate Units (MAC).

We used the 8 bits array multiplier to perform the multiplication between the pixels of the input image and the kernel used, then we added all the results obtained through this process by accumulators.

The first figure contains several components which are the multiplier and accumulator that are illustrated an the Figure 3.6.

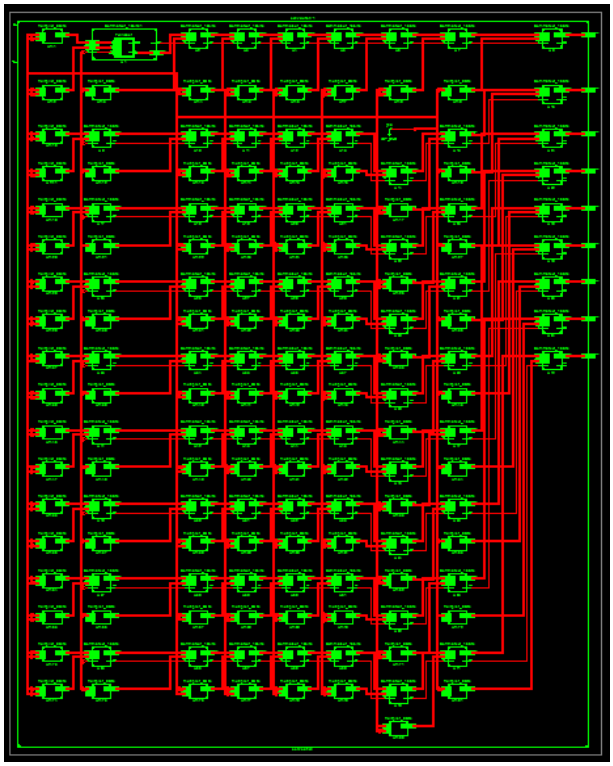

Figure 3.5: Internal architecture of convolution block.

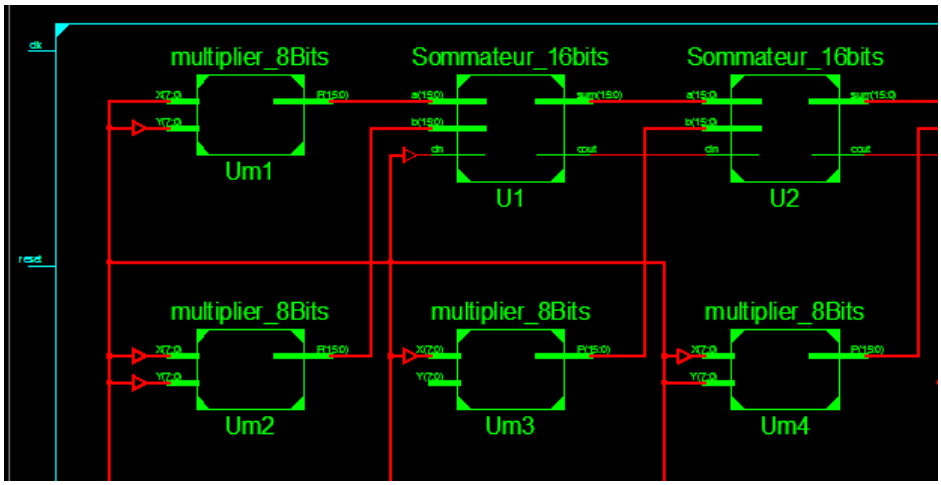

Figure 3.6: Part of internal architecture of convolution block.

3.3.3 ReLu Function Block

AS we explained in chapter two, this function introduces non linearity into the neural network, it is the most efficient due to its simplicity of calculation. This is what allows us to optimize the performances of complex neural network models.

The flowchart below explains how this function works.

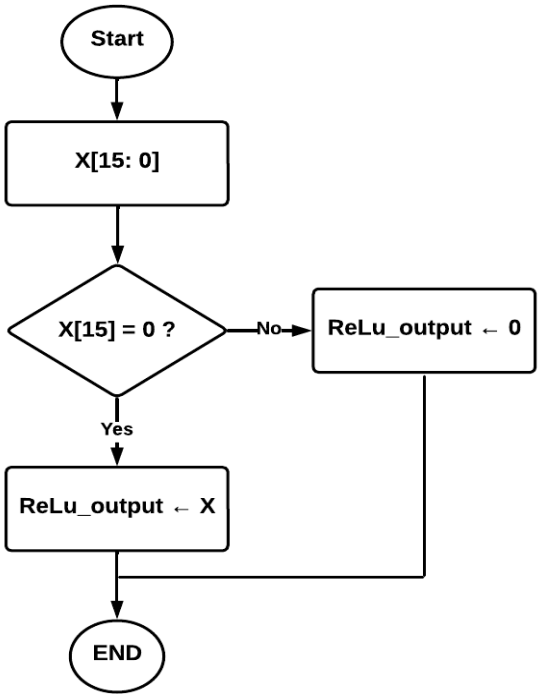

Figure 3.7: ReLu Function Flowchart.

The circuit that actually allows this process to be done is shown in the following Figures:

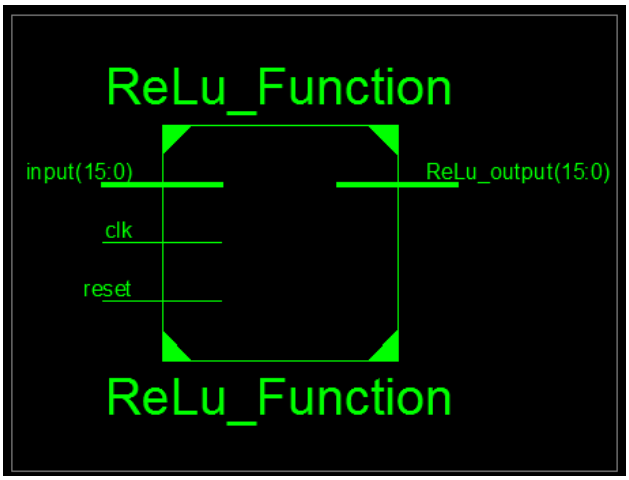

Figure 3.8: ReLu Function Block.

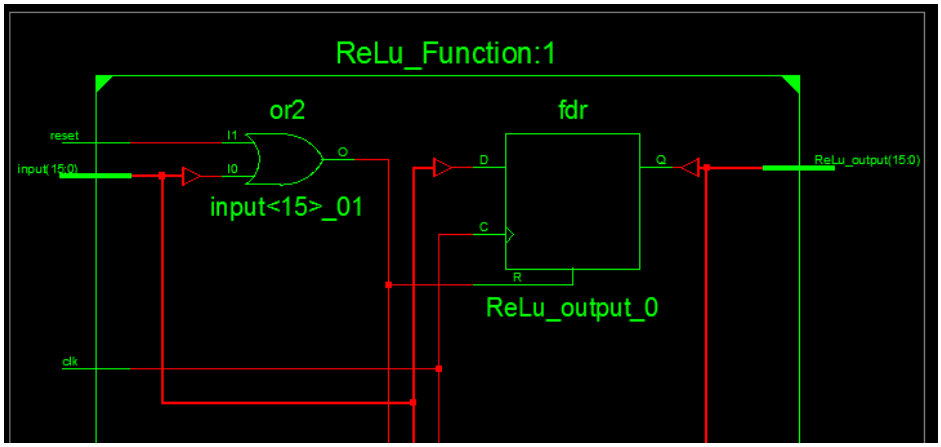

Figure 3.9: Part of internal architecture of ReLu Function Block.

This figure illustrates a logic gate OR2 connected with D flip flop fdr, if the *input*[15] is positive(the most significant bit is equal to zero), the *ReLu_output* is going to be set as it is in the *input*, but if the *inpu*[15] was negative(the most significant bit is different to zero) the*ReLu_output* will automatically be declared as a zero.

3.3.4 Max Pooling Block

This function compares the four inputs values to find the maximum among these values. As shown in the following flowchart.

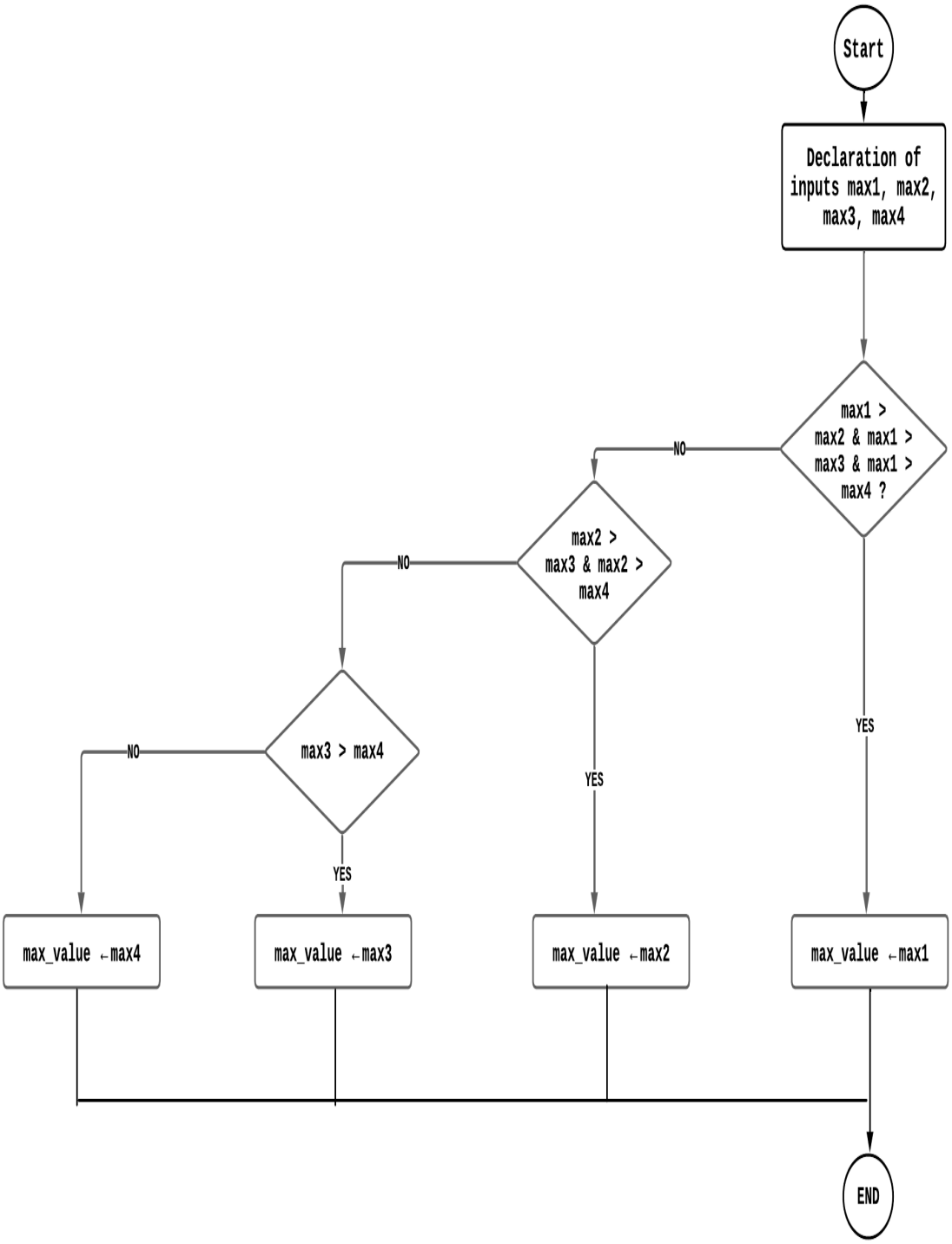

Figure 3.10: Max Pooling Flowchart.

Implementing this process in FPGA:

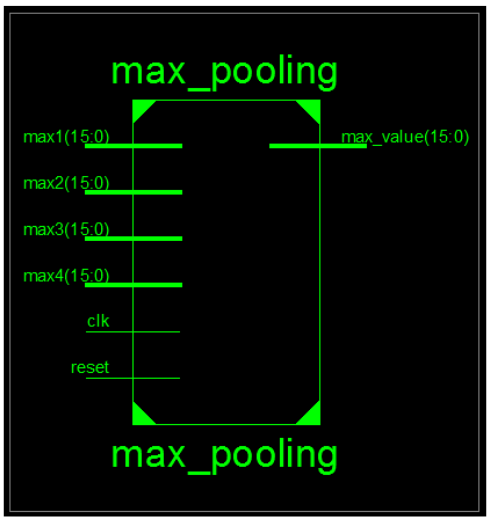

Figure 3.11: Max Pooling Block.

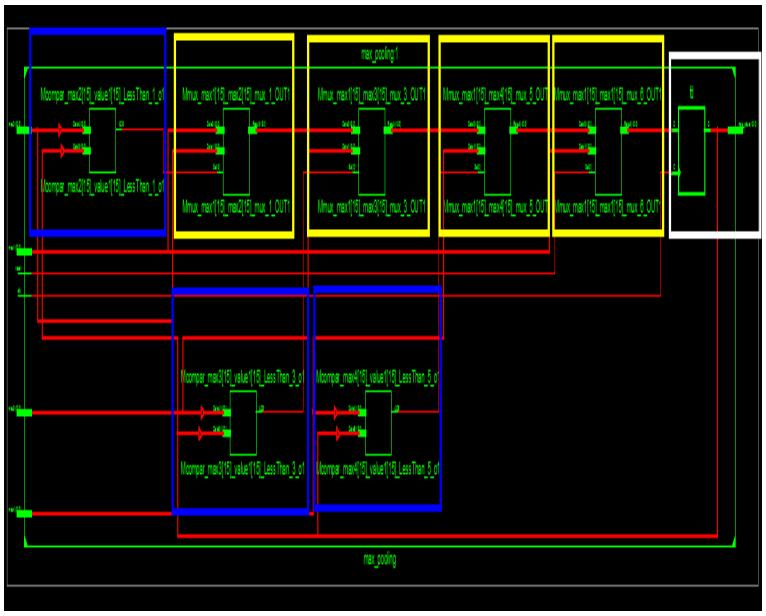

Figure 3.12: Internal architecture of Max Pooling Block.

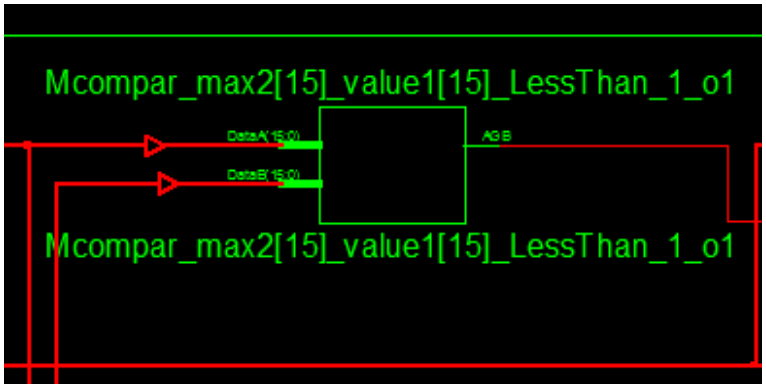

Figure 3.13: Mcompar.

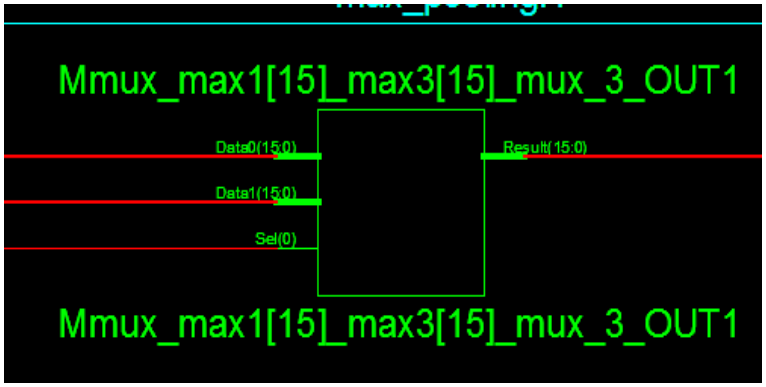

Figure 3.14: Mmax.

The figure 3.11 represents the general block of Max Pooling process while the figure 3.12 represents the internal architecture of this block which is composed by three Mcompar, four Mmux and a flip flop D. The blue framed portion in the second figure is illustrated in the figure 3.13 that represents a comparator, this component compares between two values Data1(15:0) and Data2(15:0) while the yellow framed portion is a multiplexer that selects the maximum value, the selected maximum value is sent to the input of Flip Flop D, on each rising edge of the clock signal C the input D appears at the output Q, as shown in the following figure.

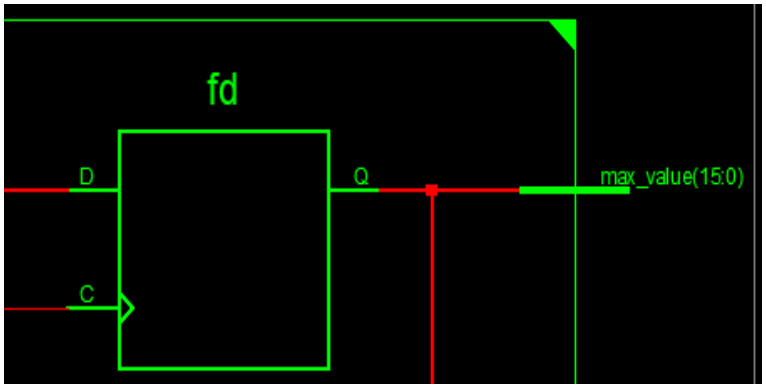

Figure 3.15: D Flip Flop.

3.3.5 Fully Connected Block

In this step, we will implement the circuits that perform the process shown in Figure 3.15.

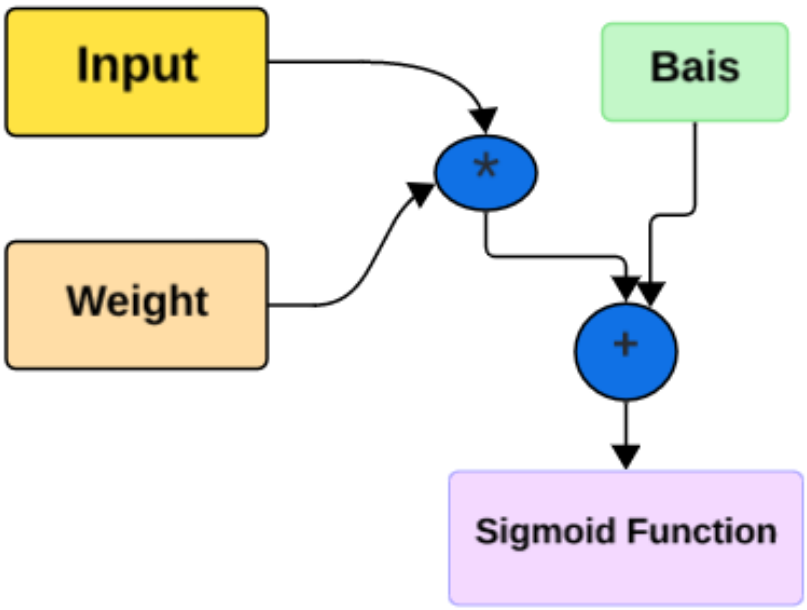

Figure 3.16: Fully connencted layers process.

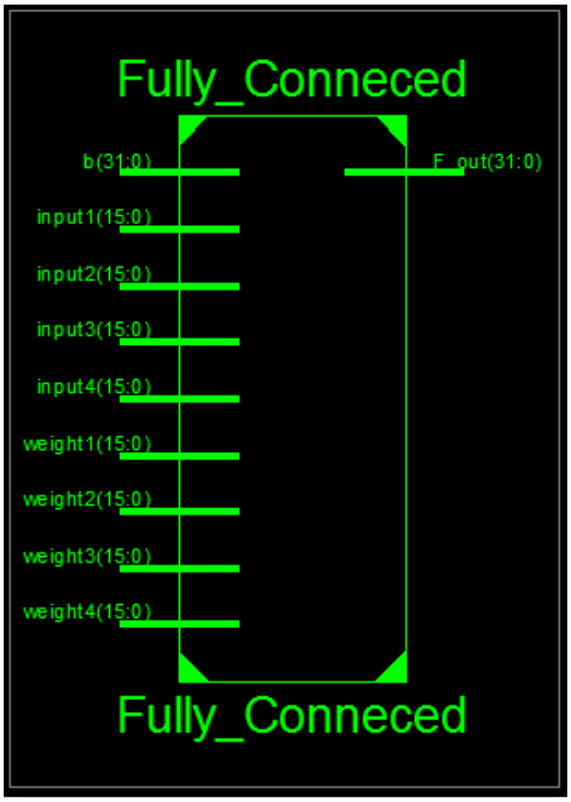

Figure 3.17: Fully connected layer block.

This figure illustrates the fully connected block, We have four inputs, each inputs has a weight, and a bias while the figure below represents the internal architecture of this block where the multipliers(mul) are used for multiplying these inputs with their corresponding weights and the accumulators(Som) for adding the partial products whith a bias.

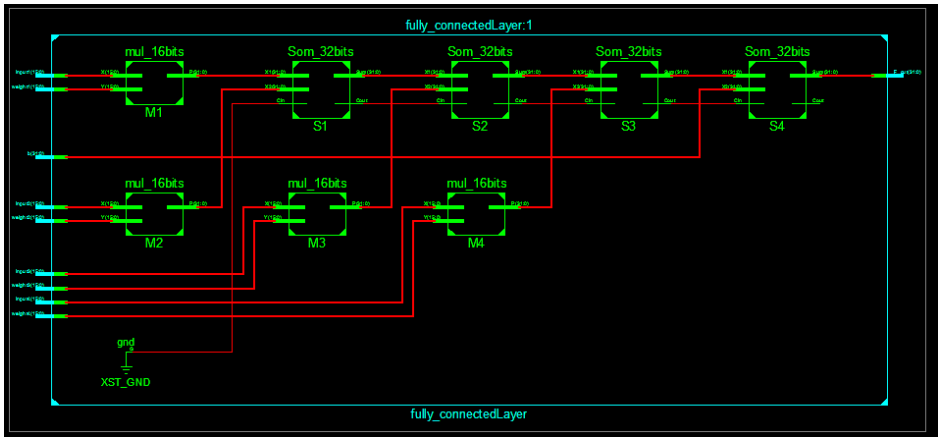

Figure 3.18: Internal architecture of Fully Connected layer Block.

For building a circuit that applied a sigmoid function at the output of fully connected layers, firstly we need to approximate it. We will use

the Piece Wise Linear Approximations PWL method which replaces this function by line segments, it is defined by:

$$l_i(x) = a_i x + b_i \tag{23}$$

l_i : The approximation of the sigmoid function for each interval of $[x_i, x_{i+1}]$.

a_i and b_i are the two coefficients that can obtain lines as close as possible to the function.

$$a_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \tag{24}$$

$$b_i = f(x_i) - a_i x_i \tag{25}$$

We notice that the sigmoid function limited by $y = 1$ and $y = 0$, also it has a symmetric point y = 0.5. For implementing this function with these caricaturists in FPGA, we will choose these intervals: $x < -6$, $-6 \leq x < -2$, and $-2 \leq x \leq 2$, $x > 2$:

$$f(x) = \frac{1}{1 + e^{-x}}$$

$f(-6) = 0.002$
$f(-2) = 0.119$
$f(2) = 0.881$
$f(6) = 0.998$

- Segment 1: $x < -6$
  $l = 0$

- Segment 2: $-6 \leq x < -2$

  $a_2 = 0,0293$, $b_2 = 0,177$
  $l = 0.0293x + 0.177$

- Segment 3: $-2 \leq x \leq 2$

  $a_3 = 0,191$, $b_3 = 0,5$
  $l = 0.191x + 0.5$

- Segment 4: $x > 2$
  $l = 1$

the following figure represents the comparison between the sigmoide function and its approximation in matlab.

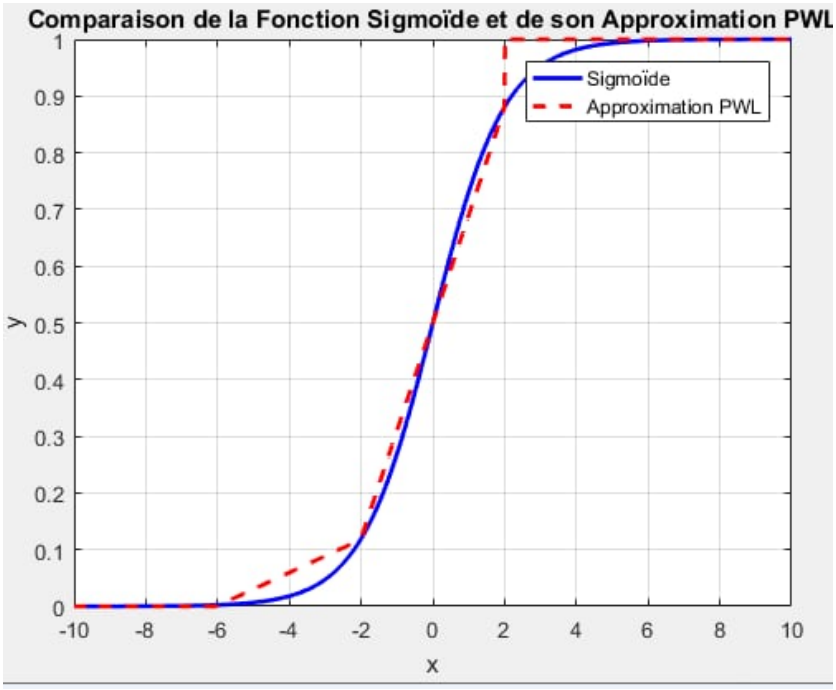

Figure 3.19: The sigmoid function with its approximation in Matlab.

Implementing this function on an FPGA:

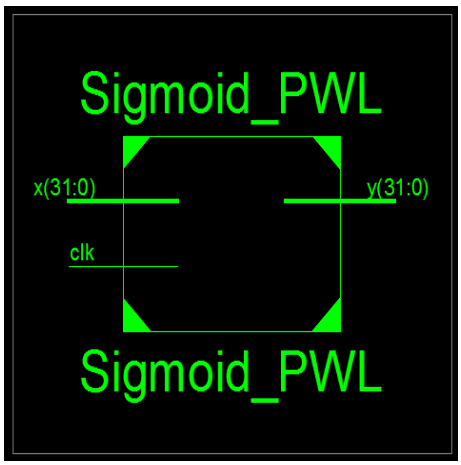

Figure 3.20: Sigmoid Function.

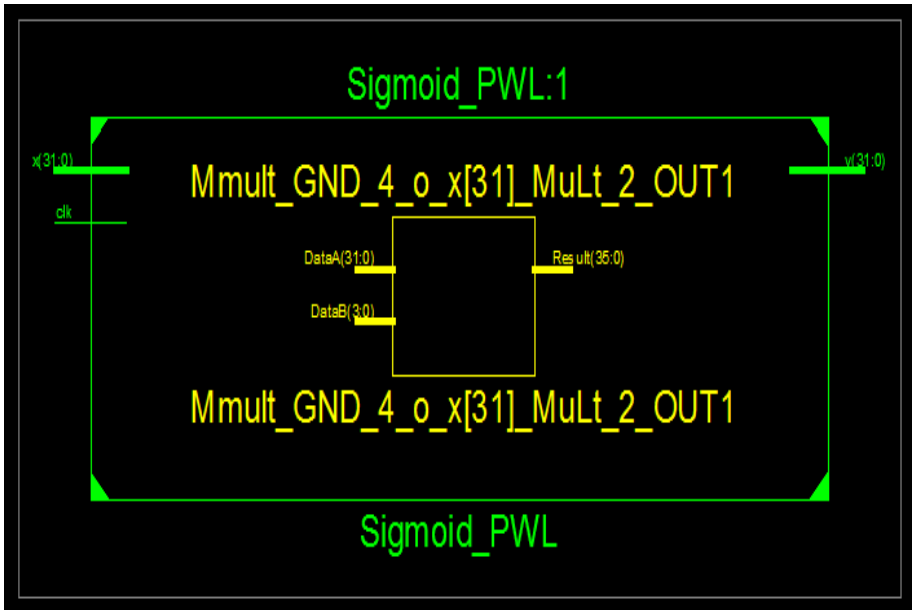

Figure 3.21: Sigmoid Function.

3.3.6 Backpropagation Block

Firstly, we are going to implement a binary cross entropy function in vhdl. The formula for this function is as follows:

$$BCE = -[p \cdot log(q) + (1-p) \cdot log(1-q)] \tag{26}$$

where:
p: The true label (0 or 1)
q: The predicted probability

We use the Piece Wise Linear Approximations PWL method which replaces this function by line segments, like we used it in the implementation of sigmoide function.
We choose these intervals: $x < 0.25$, $0.25 \leq x < 0.5$, and $0.5 \leq x \leq 0.75$, $x > 1$:

$$f(x) = \frac{1}{1+e^{-x}}$$

$f(0.25) = 0.5629$
$f(0.5) = 0.6225$
$f(0.75) = 0.6796$
$f(1) = 0.7311$

- Segment 1: $q < 0.25$
  $l = 0$

– Segment 2: $0.25 \leq q < 0.5$

  $a_2 = 0,02384$, $b_2 = 0,5033$
  $l = 0.02384q + 0.5033$

– Segment 3: $0.5 \leq q \leq 0.75$

  $a_3 = 0,2384$, $b_3 = 0,5033$
  $l = 0.2384q + 0.5083$

– Segment 4: $q > 1$
  $l = 1$

  Implementing this function on an FPGA:

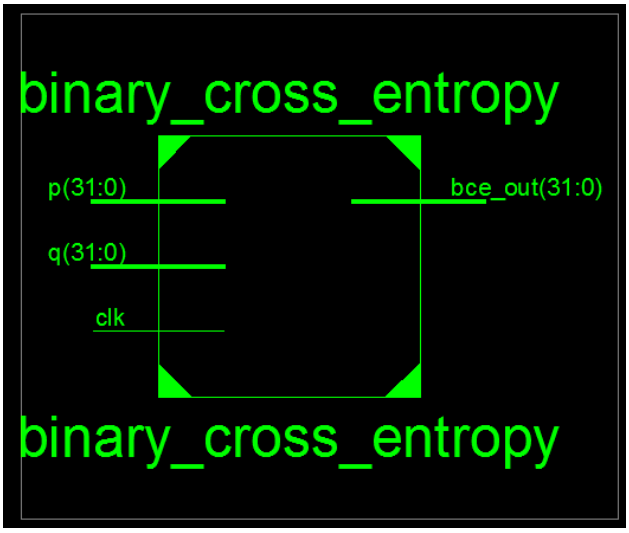

Figure 3.22: Binary Cross Entropy Block.

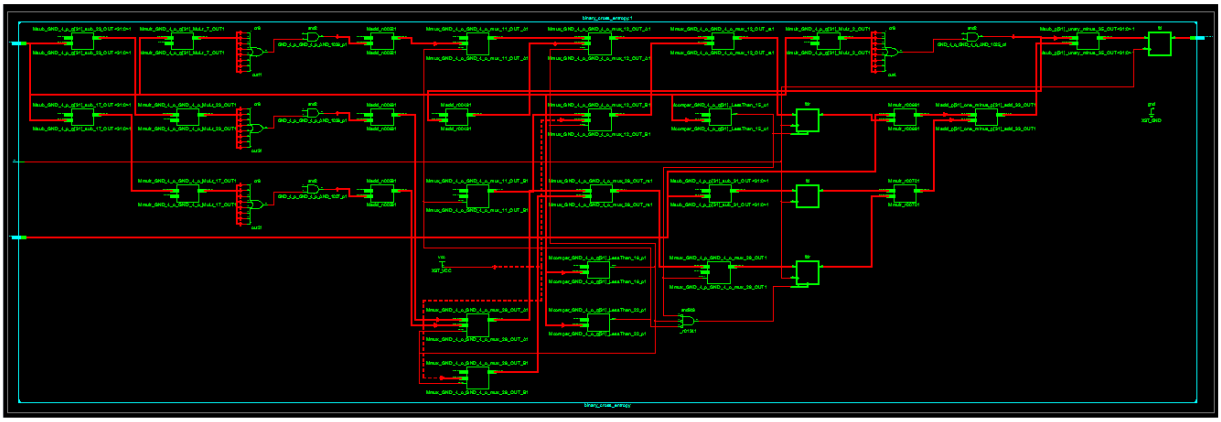

Figure 3.23: Internal architecture of cross entropy block.

The following figures represent a backpropagation module using the gradient descent algorithm, this module receives input value, output value of fully connected layers, target value and the previous and bais. we use this information to update the network weights and biases based on the calculated error.

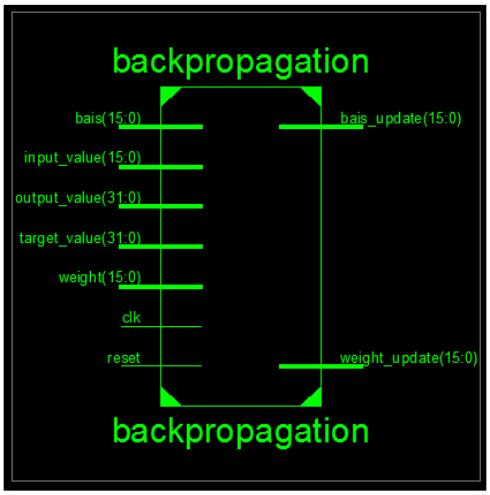

Figure 3.24: Backpropagation Block.

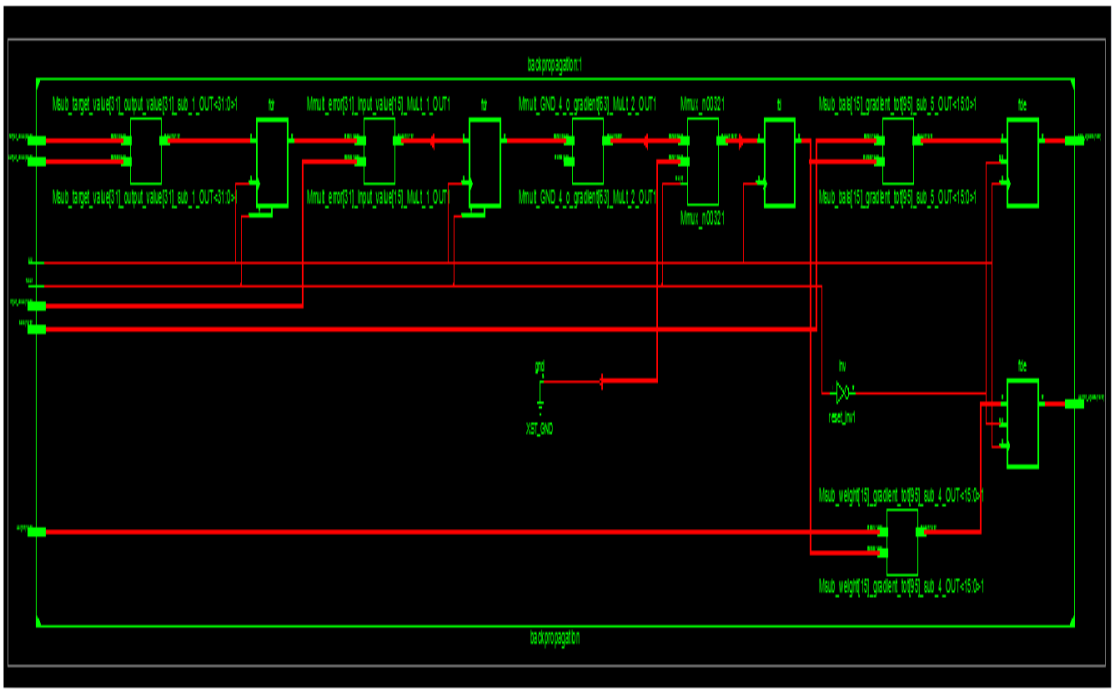

Figure 3.25: Internal architecture of backpropagation block.

3.3.7 Comparator

The comparator is a device that helps us to determine which category the input image belongs to. If the input is greater than the threshold, then the input is classified as a positive class when the input is less than this threshold that is classified as a negative class. This process is presented in the following flowchart while its circuit is illustrated in the figure 2.23.

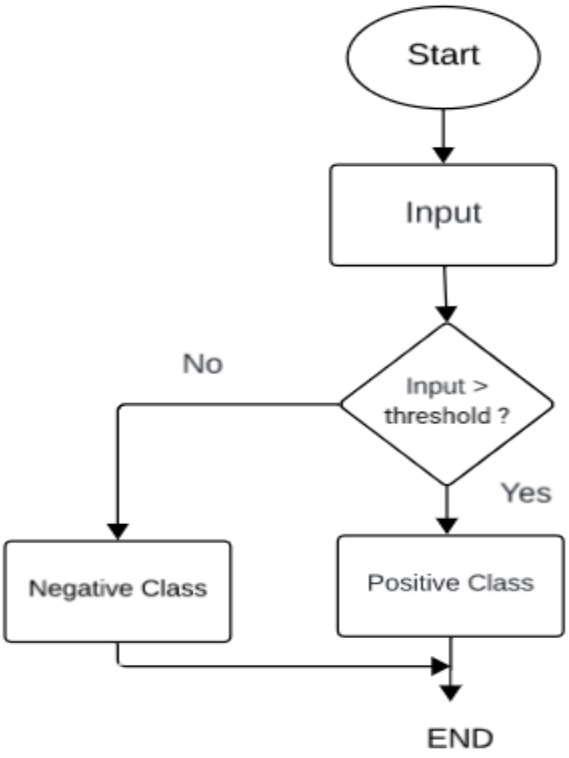

Figure 3.26: Comparator flowchart.

From this flowchart we will create comparator circuit:

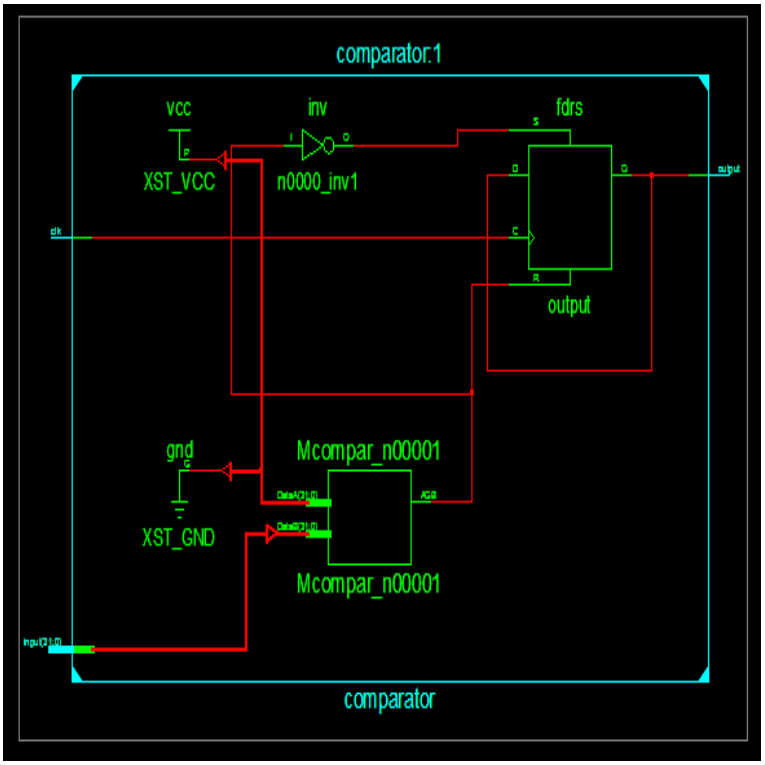

Figure 3.27: Comparator circuit.

After implementing all the necessary blocs on convolutional neural networks, we combine them into one code to achieve the following diagram:

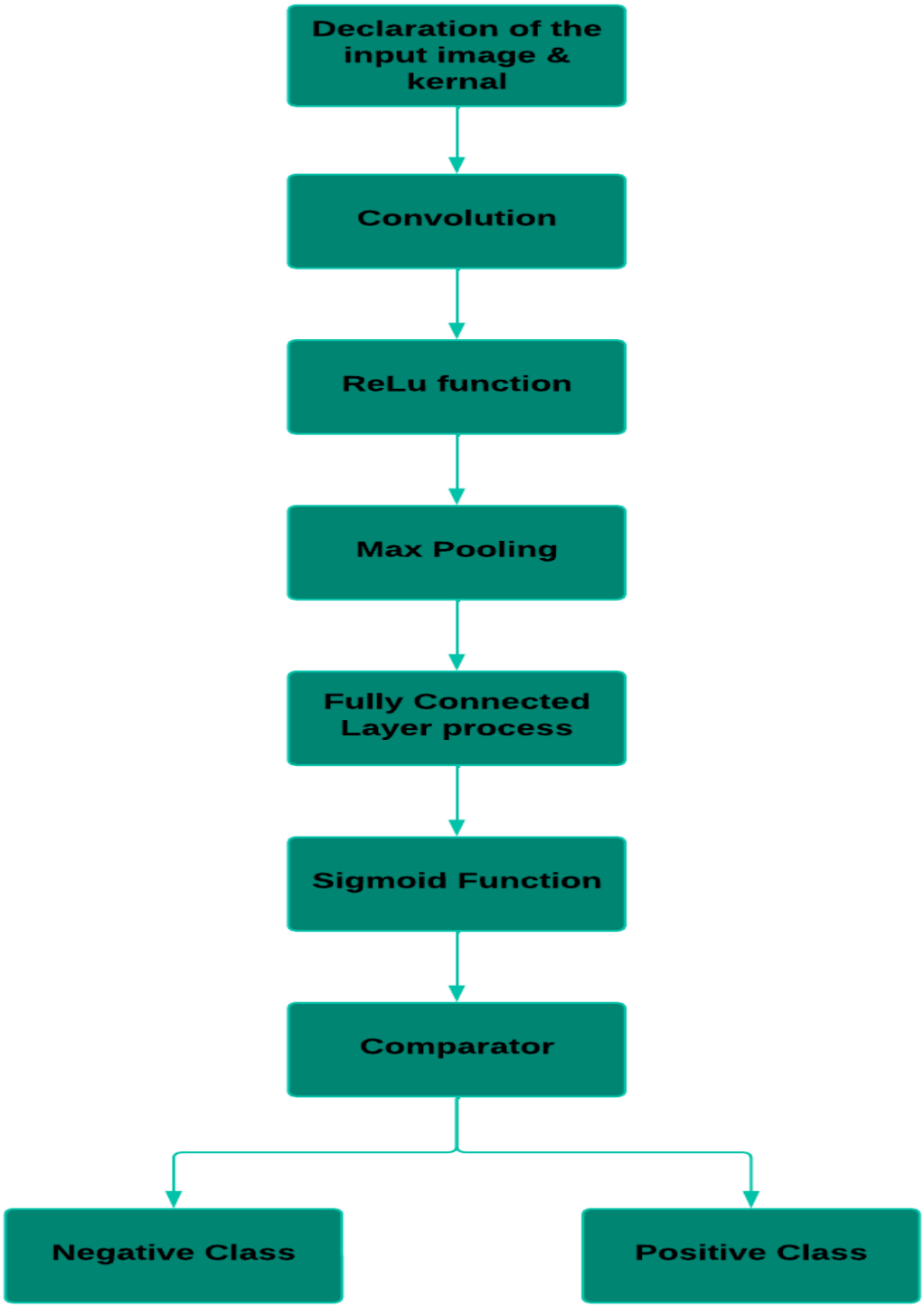

Figure 3.28: CNN Diagram

The RTL schematic of our project that is combined all blocks of CNN:

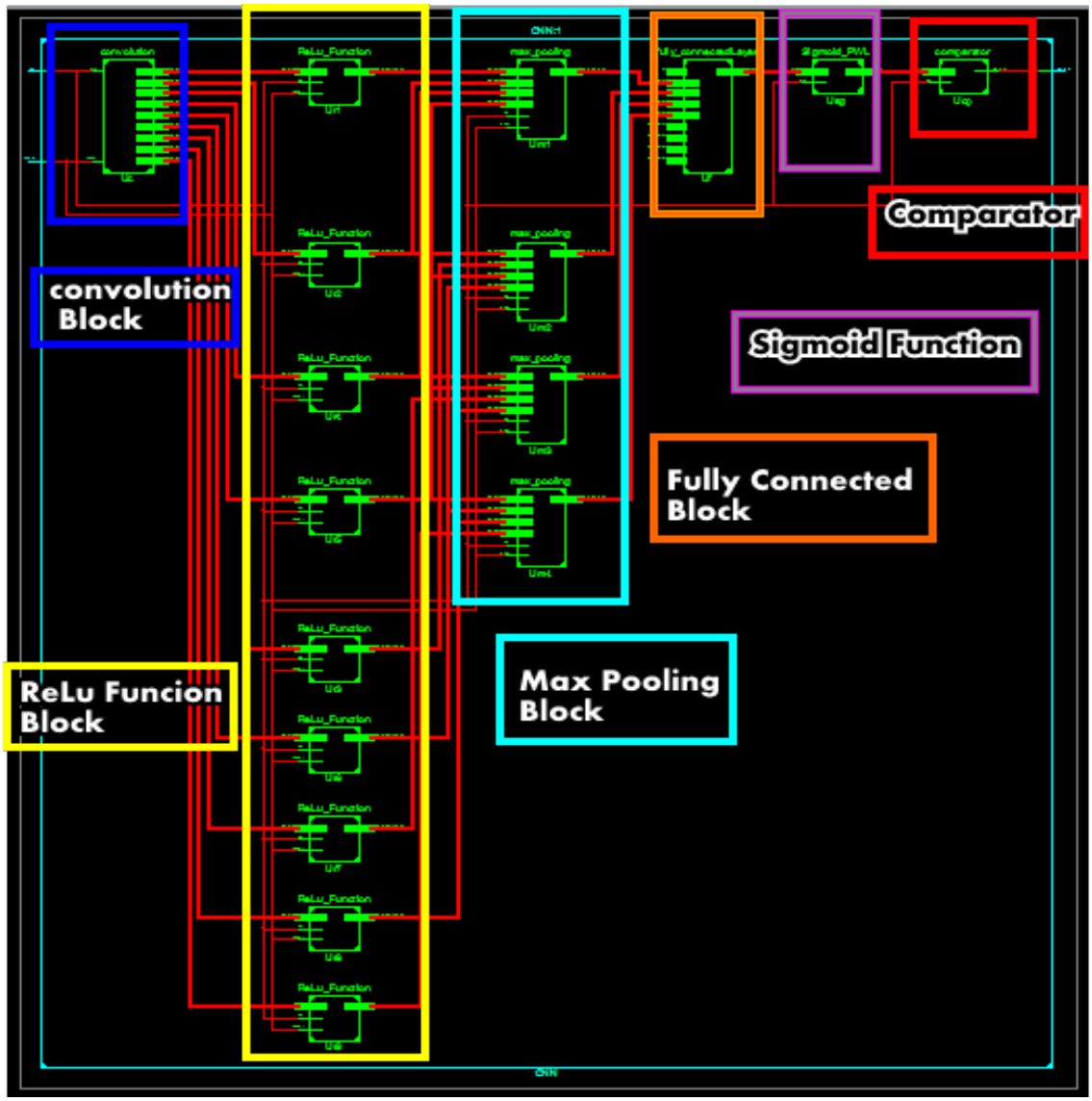

Figure 3.29: Enter Caption

3.4 Displaying Image

In this section, we will explain the steps which allow a displaying an image.To illustrate that, we are going to display a flag on a monitor with a resolution 1280*1024 and a pixel clock 108MHz:

* Firstly, we need to generate the pixel clock, we are going to use the clock of Mimas v2 100MHz in this case we don't need to implement a clock devider or a PLL circuit in our board because this frequency is acceptable.

* Declare the horizontal and vertical parameters of this resolution.

* We need two counters, a vertical counter to count the number of line in an image and a horizontal counter follows pixel position in an each line.

* When the counter rectifies the active pixel we are going to send the RGB data.

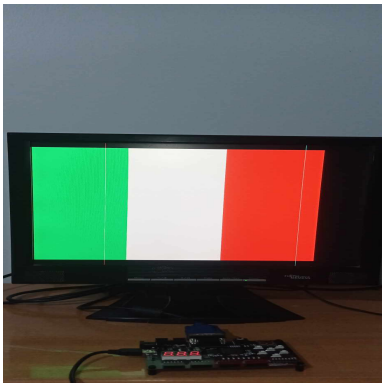

Figure 3.30: Displaying an image with VGA Protocol.

3.5 Conclusion

In conclusion, this chapter presents the hardware implementation of all operations and techniques used in convolutional neural networks.

We started the chapter with the implementation of the most important component, Parallel-Parallel multiplier which is used in the different operations to accelerate calculations. Then we presented the CNN blocks and the result of VGA protocol.

General Conclusion

In this work, we have developed our libraries that allows us to apply CNN on FPGA.
Initially, we defined artificial intelligence and provided a brief overview of its history and types. We also highlighted the advantages of using FPGAs in the field of artificial intelligence and the role of image processing and classification.

Next, we conducted a comprehensive review of the layers and their roles in a Convolutional Neural Network (CNN) model, and we modeled it using VHDL. We also covered the general principles of neural networks in the field of deep learning.

In conclusion, this work is considered a flexible basis in various fields, based on the libraries we created it we can process several images for instance medicinal image analysis, crop image analysis and analysing experimental data in real time.

Bibliography

[1] Mariette Awad and Rahul Khanna. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Springer nature, 2015.

[2] * Manuel Toledano-Ayala 2Edgar Rivas-Araiza 2 Axel Escamilla-García 1, Genaro M. Soto-Zarazúa 1 and Abraham Gastélum-Barrios. Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development. May 2020.

[3] U.Meher. Baese. *Digital Signal Processing Field Programmable Gate Arrys*. Springer, 2001.

[4] Chesner Desir. *Classification automatique d'images, application à l'imagerie du poumon profond*. PhD thesis, Université de Rouen, 2013.

[5] Deep Learning Foundations and Concepts. *Christopher M.Bishop, Hugh Bishop"*. 2024.

[6] Rafael C Gonzalez and Richard E Woods. Segmentation. *Digital image processing[U+2016],(Pearson Prentice Hall, 3rd edition, 2008)*, 2006.

[7] Shawki Areibi Griffin Lacey, Graham Taylor. Deep learning on fpgas: Past, present, and future. February 2016.

[8] Chenming Hu. *Interconnect devices for field programmable gate arrayn*. PhD thesis, University of California,Berkeley, 1993.

[9] Hongrui Huang. *Analysis of FPGA theory and its development potential*. PhD thesis, University of Wuhan, china, 2023.

[10] Bekir Karlik and A. Vehbi Olgac. Performance analysis of various activation functions in generalized mlp architectures of neural networks. 2003.

[11] Monauwer Alam2 Khan Huma Aftab1. *Design of VGA Controller using VHDL for LCD Display using FPGA*. PhD thesis, University, India, june 2014.

[12] Phil Kim. *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*. 2017.

[13] V7 Labs. Neural networks activation functions: The complete guide.

[14] Ahmed Laiche. *Deep Learning on FPGA (Simulation and Implementation)*. PhD thesis, UNIVERSITY OF OUARGLA.

[15] Deep Learning. *IChristopher M.Bishop, Hugh Bishop*. 2017.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[17] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9(1):381–386, 2020.

[18] Sai Mannam. Applications of deep learning in healthcare. *J Young Invest*, 1:15–17, 2020.

[19] Ninon Burgos Olivier Colliot Maria Vakalopoulou, Stergios Christodoulidis. Deep learning: basics and convolutional neural networks (cnn). *Springer*, Oct 2023.

[20] Paul Marsden. *Digital quality management in construction*. Routledge, 2019.

[21] L. Pardo J. Álvarez E. Mandado M.D. Valdés, M.J. Moure. *Using Hypermedia for Programmable Logic Devices Education*. PhD thesis, University of Vigo, Spain, August 1997.

[22] Numato Lab. Mimas v2: Spartan-6 fpga development board with ddr sdram.

[23] Numato Lab. Mimas v2 user manual.

[24] Volnei A. Pedroni. *Circuit Design with VHDL*. Springer, 2004.

[25] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.

[26] R. Maerani Deswandri D. Fitri Atmoko S. Santoso, T. Jojok Suryono and A. Manurung. Development process of fpga based technology for control rod drive systems of experimental power reactor. *Nuclear Engineering and Design*, Nov 2022.

[27] Anjan Kumar Balwinder Raj Shashikant Sharma, Manisha Pattanaik. Forward body biased multimode multithreshold cmos technique for ground bounce noise reduction in static cmos adders. 2013.

[28] siddharth sharma and simone sharma. Activation functions in neural networks. 2020.

[29] Magnus Själander. Multiplication acceleration through twin precision. 2009.

[30] Kenji Suzuki. Dartificial neural networks - methodological advances and biomedical applications. April, 2021.

[31] Scott E Umbaugh. *Digital image processing and analysis: applications with MATLAB and CVIPtools*. CRC press, 2017.

[32] V7 Labs. Neural networks activation functions: The complete guide.

[33] Heyang Xu. Fpga: The super chip in the age of artificial intelligence. In *Journal of Physics: Conference Series*, volume 2649, page 012018. IOP Publishing, 2023.