REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Mouloud MAMMERI de Tizi-Ouzou

Faculté de Génie électrique et d'informatique

Département d'Automatique

Mémoire

Présenté en vue de l'obtention du diplôme de

Magister en Automatique

Option : **Optimisation et Commande des Systèmes**

Présenté par : M^{elle} SNAOUI Djouher

Thème

Commande Numérique en Force à Base de la Carte FPGA d'une Architecture de Télé-opération à un Seul Degré de Liberté

Membres du Jury :

M^{r} : LAGHROUCHE Mourad	Professeur, Université de TO	**Président**
M^{r} : MELLAH Rabah	Maitre de conférences classe A, Université de TO	**Rapporteur**
M^{me} : DJAMAH Tounsia	Maître de conférences classe A Université de TO	**Examinatrice**
M^{r} : DAOUI Ahmed	Maitre de conférences classe A, Université de TO	**Examinateur**

Avant-propos

Le travail présenté dans ce mémoire a été préparé au sein du laboratoire L2 CSP du Département Automatique de la Faculté de Génie Electrique et Informatique, l'université Mouloud MAMMERI de TIZI OUZOU, sous la direction de M. Rabah MELLAH, Maître de Conférence Classe A.

A cet effet, je suis très honorée que Monsieur MELLAH Rabah du laboratoire L2CSP ait accepté d'être mon promoteur. Je tiens à le remercier très vivement et à lui exprimer ma profonde reconnaissance pour m'avoir proposé ce sujet de recherche pluridisciplinaire et m'avoir accueillie au sein de son laboratoire. Je tiens aussi à le remercier pour sa rigueur scientifique et ses qualités humaines ainsi que pour ses conseils didactiques, ses nombreux encouragements et la très grande liberté qu'il m'a accordée dans l'orientation de mes travaux de recherche.

Je tiens à remercier très chaleureusement les membres du jury, qui ont accepté la charge de juger ce mémoire.

A titre de reconnaissance, je dédie ce travail à toutes les personnes qui se dévouent à la recherche scientifique.

Je dis également un grand merci à mes amis, mes collègues est mes enseignants qui m'ont apporté leur soutien moral et technique quand j'en ai eu besoin.

Finalement, je réserve une place particulière à toutes les personnes qui ont contribué, d'une façon directe ou indirecte à l'achèvement de ce travail et à tous ceux et celles qui m'ont aidée et soutenue durant tout mon parcours. Ils trouvent ici l'expression de ma gratitude et de ma reconnaissance les plus sincères. Car le chemin parcouru depuis le début de mes études est le fruit de plusieurs relations avec de nombreuses personnes, sans les nommer.

M^{elle} : SNAOUI Djouher

Liste des acronymes et nomenclature

Les principaux acronymes et nomenclatures utilisées dans ce mémoire sont rapportés ci-dessus, et d'autre signification ce trouvent explicitées dans le texte

ANL: Argonne National Laboratory
ROV: Remotely Operated Vehicles
TAO : Télé-opération Assistée par Ordinateur
FPGA: Field Programmable Gate Array
CLB : Blocs Logiques Configurables
IOB: Blocs d'Entrées -Sorties
ROM: Read Only Memory
CAO : Conception Assisté par l'Ordinateur
HDL: Hardware Description language.
VHDL: Verry High speed integrated Hardware Description language
ISE: Integrated Software Environment
PLD: Programmable Logic Device
CAN : Convertisseur Analogique Numérique
MLI : modulation de largeur d'impulsion
RTL : Register Transfert Level
JTAG: Joint Test Action Group
LUT : Look Up table

Table des figures

Figure(I.01): PhantoM de Sensable et le CyberGrasp de Virtual Technology .

Figure (I.02) : Description générale d'un système de télé-opération

Figure (I.03): Schéma représentant les différents blocs du système de télé-opération bilatéral

Figure(I.04): Représentation d'un système de télé-opération par modèle 2-ports

Figure (I.05) : Contrôle à 4 C anaux (4C)

Figure (I.06) : Contrôle Position-Position (PP)

Figure (I.07) : Contrôle Force-Position (FP)

Figure (I.08) : Quadripôle schématisant le bloc de communication

Figure (I.09) : Représentation sous forme de réseau d'un système de télé opération.

Figure (II.01) : Architecture interne d'un PFGA

Figure (II.02). Architectures interne d'un CLBs dans les FPGAs Virtex5

Figure (II.03): Structure générale du routage

Figure (II.04): La matrice de commutation

Figure (II.05). La carte Virtex5-ML501 de Xilinx

Figure (II.06). Programmation d'un FPGA avec câble JTAG en utilisant l'application IMPACT

Figure (II.07) : Etapes de conception sur FPGA

Figure (II.08) : Description du code VHDL d'un compteur modulo-8

Figure (II.09) : Schéma RTL d'un compteur modulo-8

Figure (II.10) : Résulta de simulation du compteur modulo-8

Figure (III.01) : Architecture Télé-Opération

Figure (III.02) : les parties du système de télé-opération

Figure (III.03) : Le couplage de la partie commande à la partie opérative

Figure (III.04) : Architecture synoptique du système de télé-opération bilatiral

Figure (III.05) : Dispositif Expérimental de notre Architecture

Figure (III.06) : Schéma de puissance d'alimentation de l'actionneur du vérin

Figure (III.07) : Carte FPGA Vertex XC5VLX50

Figure (III.08) : Dispositif de commande de puissance

Figure (III. 09): Schéma-bloc de la commande en force d'une architecture de télé-opération

Figure (III.10) : La technique de Modulation à Largeur d'Impulsion MLI

Figure (IV.01) : Schéma de conception du circuit de commande

Figure (IV.02) : Signal de dent de scie

Figure (IV.03) : Schéma RTL détaillé de la commande
Figure (IV.04.a) : Schéma de simulation pour la tension d'entrée V=0v
Figure (IV.04.b) : Schéma de simulation pour la tension d'entrée V=1v
Figure(IV.04.c) : Schéma de simulation pour la tension d'entrée V=2v
Figure (IV.04.d) : Schéma de simulation pour la tension d'entrée V=3v
Figure (IV.04.e) : Schéma de simulation pour la tension d'entée V=4v
Figure (IV.04.f) : Schéma de simulation pour la tension d'entrée V=5v
Figure (IV.05) : La configuration des entrés /sorties en utilisant plan Ahead
Figure (IV.06) : Les étapes l'implémentation du circuit de commande
Figure (IV.07) : Choix du mode de programmation
Figure (IV.08.a) : Signal de commande pour la tension d'entée V=0 V.
Figure (IV.08.b) : Signal de commande pour la tension d'entée V=0.5 V
Figure (IV.08.c) : Signal de commande pour la tension d'entée V=1V.
Figure (IV.08.d) : Signal de commande pour la tension d'entée V=1.4 V
Figure (IV.08.e) : Signal de commande pour la tension d'entée V=2 V.
Figure (IV.08.f) : Signal de commande pour la tension d'entée V=2.4 V.
Figure (IV.08.g) : Signal de commande pour la tension d'entée V=3 V.

RESUME

Ce travail de mémoire s'inscrit dans le cadre du développement technologique dans le domaine de la microélectronique. Des solutions numériques telles que les FPGA (Field Programmable Gate Array) peuvent être utilisées comme support pour l'implantation des algorithmes de commande. Ce sont des solutions matérielles ayant pour objectif de contrôler les systèmes de télé-opération en temps réel. C'est une nouvelle approche qui consiste en une amélioration des performances des schémas de contrôle des architectures de télé-opérations. La technologie des circuits logiques programmables FPGA est une alternative innovante par rapport aux processeurs de contrôle conventionnels comme les DSP et les micro-processeurs. C'est pour cette raison que les FPGA font l'objet de recherches actives. A cet effet, nous nous sommes tracés comme objectif de concevoir un circuit de contrôle, c'est-à-dire l'implantation sur cible FPGA une technique de contrôle en force. Cette technique est la plus utilisée pour la commande en force des architectures de télé opération. Il s'agit de la technique de contrôle par Modulation de Largeur d'Impulsion MLI.

La première partie de ce document est dédiée à la présentation, d'une manière générale, des systèmes de télé-opérations ainsi que de leurs critères de performances. La deuxième partie est consacrée à la présentation des circuits FPGA et leurs architectures internes. Nous avons décrit les FPGA de la famille XILINX, car nous nous sommes intéressés précisément à la famille Vertex. Nous avons décrit et détaillé dans cette partie l'architecture externe de la carte FPGA de la famille vertex du cœur ML 501. Et à la fin de cette partie nous avons présenté le langage de description matériel VHDL utilisé pour la programmation de cette carte ainsi que son environnement de conception ISE. La troisième partie est consacrée à la présentation de système de télé-opération à retour de force à un seul degré de liberté, objet de notre étude. On a décrit tous les éléments qui constituent ce système, puis nous avons parlé de l'architecture de commande, il s'agit de l'architecture force - position. Dans cette architecture seulement, **la position** du dispositif maître (**manette)** est transmise à l'esclave. A l'inverse, le dispositif esclave (**vérin)** transmet uniquement **l'effort** qu'il applique sur la charge à soulever au poste maître **(bloc de sensation**), pour que l'opérateur puisse ressentir l'interaction d'effort entre le vérin et son environnement. Enfin, dans cette partie nous avons présenté la technique de contrôle en force la plus couramment utilisée pour la commande des architectures de télé-opération. Il s'agit de la technique de contrôle MLI. La

quatrième partie est consacrée à la conception, simulation et analyse d'un circuit de commande dans l'environnement de programmation **ISE** (**Integrated Software Environment**). Nous avons décrit la stratégie de l'implémentation de la technique de contrôle MLI dans un support FPGA en utilisant le langage de description matériel VHDL. Puis, nous avons présenté les étapes de programmation et d'implantation de cette commande (circuit de commande) sur un support physique FPGA.

Enfin, ce mémoire est achevé par une conclusion générale et quelques perspectives de recherches.

Mots-clés : commande numérique, télé opération, retard, transmission, commande en force, logiciel VHDL, carte FPGA.

Table des matières

Introduction Générale 1
CHAPITRE I Description de l'Architecture de Télé-Opération à un degré de liberté 4
I.1) Introduction 4
I.2) Interfaces haptiques 6
I.3) Présentation et structure générale des systèmes de télé-opération 7
I.3.1) Définition 7
Les principaux éléments d'un système de Télé-opération sont [16] : 7
I.3.2) Représentation mathématique des systèmes de télé-opération 8
I.4) Architectures de contrôle des systèmes de télé-opération bilatérale 10
I.4.1) Architecture de télé-opération 4 canaux 11
I.4.2) Architecture de télé-opération position-position 11
I.4.3) Schéma de télé-opération Force-Position (FP) 12
I.5) Analyse des systèmes de télé-opération 14
I.5.1) Stabilité 14
I.5.2) Transparence 16
I.6) Conclusion 18
CHAPITRE II Description de la carte FPGA et le logiciel d'utilisation 19
II.1) Introduction 19
II.2) Définition des FPGA 21
II.3) Architecture interne d'un FPGA [02] 22
II.3.1) Les blocs logiques configurables (CLB) 22
II.3.2) Blocs d'entrée-sortie (IOB) 23
II.3.3) Ressources d'interconnexions 23
II.4) FPGA de la famille XLINX 25
II.5) Présentation et description des éléments de FPGA ML 501 26
II.7) Langage de description VHDL 31
II.8) L'environnement de conception ISE 33
II.9) Conclusion 37
Chapitre III Présentation de l'architecture de commande de notre système de télé-opération bilatéral à un seul degré de liberté 38
III.1) Introduction 38

III.2) Présentation du projet de télé-opération avec retour d'effort à 1ddl 38
III.3) Description expérimentale du système bilatéral 41
III.4) Schéma de commande en force d'une architecture de télé-opération 46
III.5) Technique de Modulation à Largeur d'Impulsion MLI 48
III.6) Conclusion 50
Chapitre IV Implantation sur la carte FPGA de la commande en force de l'architecture de télé-opération 51
IV.1) Introduction 51
IV.2) Stratégie et implémentation de la commande dans un support FPGA en utilisant VHDL .. 51
IV.2.1) Conception du circuit de commande 52
IV.2.2) Synthèse du circuit de commande 54
IV.2.3) Simulation fonctionnelle avec Isim 55
IV.3) Implantation d'un circuit de commande sur le support physique FPGA 60
IV.3.1) Implement Design 60
IV.3.2) Generate Programming File : 62
IV.4) Programmation du FPGA et vérification 63
IV.4.1) Résultats pratiques 66
IV.4.2) Interprétation des résultats pratiques 71
IV.5) Conclusion 71
Conclusion Générale 73

Introduction Générale

Ce travail de mémoire s'inscrit dans le cadre des développements technologiques dans le domaine de la microélectronique. Des solutions numériques telles que les FPGA (Field Programmable Gate Array) peuvent être utilisées comme support pour l'implantation des algorithmes de commande. Cette solution a pour objectif de contrôler un système de télé-opération en temps réel. C'est une nouvelle approche qui consiste à améliorer les performances de contrôle des architectures de télé-opérations.

La télé-opération est considérée par plusieurs travaux récemment publiés comme un des principaux domaines de la robotique [01]. Les systèmes télé opérants ont une application très vaste, car ils permettent à l'homme d'opérer dans des milieux qui étaient inaccessibles auparavant, pour des raisons d'hostilités notamment: les espace radioactifs, aérospatiaux et fonds marins. Ou bien pour des raisons de maniabilité à échelle réduite, comme la chirurgie mini invasive, la manipulation biologique de micro systèmes …etc.

La réalisation s'inscrit dans le sciage d'opérations en milieux hostile à un degré de liberté où l'utilisateur commande à distance un vérin en translation bidirectionnelle. Si ce système de télé-opération est transparent et le retard produit par le canal de communications est négligeable alors, le vérin reproduit exactement la tâche désirée provenant du site maître. En contre partie, si le vérin rencontre une force dans l'environnement opéré, il se bloc. Un signal de retour de force sera envoyé au site maître, en mettant en marche le bloc de sensation (constitué d'un moteur à courant continue et d'un l'électroaimant) afin que, l'opérateur ressent l'importance de la force mise en jeu entre le vérin et son environnement.

En pratique, l'utilisation des actionneurs, des calculateurs ainsi que le temps de retard entre le dispositif maître et le dispositif esclave, se répercute négativement sur les performances du système. Ces inconvénients nous ont menés à choisir une interface très puissante qui est la carte FPGA.

La flexibilité de la carte FPGA ainsi que sa grande capacité de calcul, nous permettra au gré de changer les performances du système de télé-opération, en agissant sur le programme que nous allons développer nous même sous le langage de programmation matériel VHDL.[02]

Les circuits programmables FPGA sont utilisés comme supports physique pour l'implantation des algorithmes de commande en raison de leur parallélisme des solutions matérielles qui permet d'intégrer sur une seule et unique cible plusieurs algorithmes qui assure plusieurs fonctionnalités et qui peuvent travailler indépendamment les uns des autres. Leurs grandes capacités de calcul font que les délais de temps de calcul sont négligeables en dépit de la complexité des algorithmes de commande à implanter.

Les FPGA permettent de pousser encore plus loin le progrès technologique en électronique. En effet, le degré de performance des FPGA fait d'eux aujourd'hui une solution pour remplacer les ASIC et les processus personnalisés dans des applications de contrôle.

Les FPGA permettent aussi de [02, 03,04] :

1. Réduire les coûts,
2. Augmenter les performances et la fiabilité d'un circuit,
3. Diminuer les délais de mise en place d'un prototype opérationnel
4. Réduire l'encombrement des composants sur circuit,

Ces points sont des atouts conjugués dans un FPGA qui justifient le succès de ces dérnière dans tous les domaines d'applications de l'électronique : L'aéronautique, l'aérospatial, l'industrie, les télécommunications, les télés opérations, les appareils médicaux, …etc. Par ailleurs tous ces domaines trouvent le bon compromis en intégrant les FPGA **[05].**

En effet, outre l'amélioration des performances de contrôle à travers la réduction de temps de calcul et le parallélisme des solutions, cette approche consiste en une méthodologie de développement. Il permet de résoudre l'adéquation entre l'algorithme de commande à implanter et son architecture en vue d'effectuer l'implantation optimisée en temps de ressources consommées et le temps de calcul, toute en réduisant le temps de développement. **[05].**

Pour les architectures de télé opération, plusieurs algorithmes de contrôles peuvent être utilisés. Ces algorithmes comportent souvent plusieurs boucles de régulations imbriquées. Il s'agit de boucles de force ou de couple, de vitesse, de position….etc. la boucle de régulation de force est souvent la plus difficile à implanter car elle constitue généralement la partie la plus sensible de l'algorithme de commande.

Ce travail de a pour objectif d'élaborer une étude complète avec conception, simulation et réalisation d'un circuit numérique de commande. Il s'agit d'un circuit numérique d'une technique de contrôle de force la plus couramment utilisée pour la commande des architectures de télé opération. C'est la technique de contrôle MLI (modulation à largeur d'impulsion) (en anglais

PWM). Cela en vue d'améliorer les performances des systèmes de télé-opération et de réduire le retard dans le canal de communication.

La démarche scientifique adoptée dans ce mémoire suit cette structuration :

Le premier chapitre est dédié à la présentation, d'une manière générale, des systèmes de télé-opération et les interfaces haptiques utilisées à travers des générations. Nous donnons par la suite la représentation mathématique des systèmes bilatéraux à deux canaux. Puis nous allons ensuite présenter les architecture de contrôle des systèmes de télé-opération les plus couramment utilisés plus particulièrement on s'intéressera à l'architecture de contrôle force-position. Et enfin, nous allons étudier la stabilité et la transparence de ces systèmes.

Le second chapitre sera consacré à la présentation des circuits programmables FPGA et leurs architectures internes. Nous allons présenter les FPGA de la famille XILINX, en se limitant seulement aux FPGA de la famille Vertex. Nous allons décrire dans ce chapitre l'architecture externe de la carte FPGA de la famille vertex du cœur ML 501 ainsi que les éléments les plus utilisés et embarqués dans cette carte. Et enfin nous présentons le langage de description matériel VHDL utilisé pour la programmation de cette carte ainsi que son environnement de conception ISE.

Le troisième chapitre s'intéresse à la présentation de notre système de télé-opération à retour de force à un seul degré de liberté. Nous allons décrire tous les éléments qui le constituent. Puis nous allons décrire l'architecture de commande utilisée dans ce projet, qui est l'architecture force - position. Enfin nous allons présenter la technique de contrôle en force la plus couramment utilisée pour la commande des architectures de télé-opération. Il s'agit de la technique de contrôle MLI (Modulation par Largeur d'Impulsion).

Le quatrième chapitre est consacré à la conception, réalisation, simulation et l'analyse des résultats expérimentaux d'un circuit de commande dans l'environnement de programmation **ISE** (**Integrated Software Environment**). Nous allons décrire la stratégie de l'implémentation de la technique de contrôle MLI dans un support FPGA en utilisant le langage de description matériel VHDL. Dans ce chapitre nous présentons les étapes de programmation et de l'implantation de cette commande (circuit de commande) sur un support physique FPGA.

Enfin nous clôturons ce mémoire par une conclusion générale

CHAPITRE I Description de l'Architecture de Télé-Opération à un degré de liberté

Objectif du chapitre

Une représentation ainsi qu'une description mathématique d'un système de télé-opération bilatérale feront l'objet de ce chapitre. Par conséquent, sur la base de cette représentation, des critères de performances seront définis. Ensuite, nous allons présenter les schémas de commande des architectures de télé-opérations les plus couramment utilisés. Pour finir, nous allons analyser ces systèmes pour pouvoir étudier la stabilité et transparence.

I.1) Introduction

La télé-opération commence avec le manipulateur maître-esclave développé par « Ray Goertz » à l'Argonne National Laboratory (ANL) en 1948. Ce système a été conçu pour la manipulation de substances toxiques (manipuler le carburant radioactif). Dans ce premier système, le manipulateur maître possédait la même structure mécanique et les mêmes propriétés cinématiques que celle du manipulateur esclave, donc les limitations physiques de l'esclave étaient perçues naturellement par l'utilisateur. En 1954, l'ANL développe la deuxième génération des télémanipulateurs électriques à retour d'effort. Dans un système à retour d'effort, toute force externe expérimentée par le manipulateur esclave est produite sur le manipulateur maître. Ceci est utilisé pour implanter le contrôle bilatéral : une force appliquée sur l'esclave (respectivement maître) produira un mouvement du maître (respectivement esclave). Les systèmes hydrauliques ont aussi été présents depuis le début de la télé opération. Le premier système de ce type est le (Handyman) de Ralph Mosher développé aux laboratoires de General Electric en 1958. Le Handyman, manipulateur à retour d'effort consistait en deux bras hydrauliques à 10 degrés de liberté, dont deux degrés de liberté pour chaque doigt.

Au début des années 60, le champ d'application de la télé-opération s'étend à l'exploration spatiale. Surtout à la télémanipulation et au contrôle de véhicules mais, avec des retards de transmission. En 1970, l'exploration sous-marine devient l'un des principaux champs d'applications

des systèmes télé opérés. Dans les années 80, les véhicules opérés à distance ROV (Remotely operated vehicles) commencent à être amplement dans les industries des gaz et du pétrole. Les ROV sont utilisés pour la surveillance des oléoducs et l'inspection des structures artificielles sous-marines. L'apparition des ordinateurs et le progrès en matière de technologie ont permis le développement des systèmes de télé-opération de plus en plus performants dans diverses fonctions : le dépannage de satellites, l'exploration des volcans ,le déminage, l'aide aux personnes handicapées (avec, par exemple les fauteuils roulants intelligents…etc.) on trouve plus de détails sur les systèmes de télé-opération dans les références [06,07, 08].

En résumé, Les systèmes de télé-opération ont été largement développés des domaines et des applications divers :

- Atteindre des zones à risque ou dangereuses pour l'homme par exemple pour la détection de fuites dans des conduites de substances nocives [09],
- Explorer des environnements sous-marins pour la surveillance de l'Antarctique et le suivi des régions polaires [10],
- Explorer des environnements terrestres, dans des zones à risque ou suite à une catastrophe naturelle [11]
- Réaliser des actes chirurgicaux. [13].

Les systèmes de télé-opération sont des systèmes commandés à distance. Ils permettent aux opérateurs humains de contrôler un robot dans un environnement lointain, inconnu ou dangereux. L'opérateur humain est en contact direct avec le dispositif maître, et le dispositif esclave est en contacte directe avec l'environnement. L'opérateur humain applique une force sur le maître (un joystick par exemple), générant ainsi une position désirée. Cette position est envoyée par le canal de communication au dispositif esclave. L'esclave exécute la commande reçue. Les forces exercées sur l'esclave sont renvoyées par le canal de communication vers l'interface haptire (un joystick avec retour de force par exemple ou un retour visuel) donc vers l'opérateur humain. Ce retour d'information permet à l'opérateur humain d'évaluer la situation du dispositif esclave. et donc de modifier sa force appliquée. Dans le cas où il n'y a pas de retour d'information, le système de télé-opération est appelé: **système de télé-opération unilatéral**. Avec retour d'information, il est appelé **système de télé-opération bilatéral [14]**

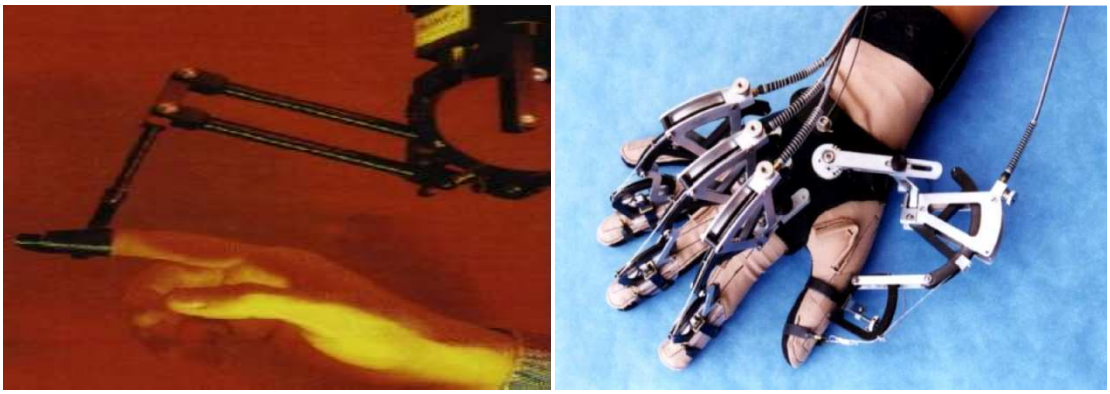
I.2) Interfaces haptiques

Les systèmes de Télémanipulation ont été développés dans le secteur nucléaire, pour permettre la commande gestuelle de robot-manipulateur avec de bonnes propriétés de dextérité et de transparence. **La première génération** de télémanipulateurs est basée sur le principe maitre-esclave, où les bras manipulateurs sont géométriquement semblables et caractérisés par une réversibilité de fonctionnement.

Durant **la décennie 1980-1990, u**ne nouvelle génération des systèmes de Télé-opération apparaît sous forme de Télé-opération Assistée par Ordinateur (TAO). Elle se caractérise par l'introduction d'un calculateur dans la commande. Par la suite, plusieurs périphériques à retour d'effort sont commercialisés : comme « **Kraft » et « Schilling »**.

La dernière **décennie 1990-2000**, marque l'extraordinaire diversification de nouvelles applications (simulateurs, médical, cao, multimédia…). L'intérêt des laboratoires de recherches pour les périphériques haptiques s'est traduit par l'exploration de différents concepts comme « bras manipulateurs à retour d'effort », « joystick à retour d'effort », « exosquelette à retour d'effort », …etc.

Ces **dernières années**, plusieurs périphériques haptiques sont apparus let que : « le Phantom de Sensable » et « CyberGrasp de Virtual Technology » du MIT. Ils sont dérivés des travaux, Impulse de Immersion Corp. [15]



Figure(I.01): Phantom de Sensable et le CyberGrasp de Virtual Technology .

I.3) Présentation et structure générale des systèmes de télé-opération

La figure (I.02) présente une description générale d'un système de télé-opération. Dans cette figure, l'opérateur commande avec un joystick à retour d'effort (maître) l'esclave qui se situe dans un environnement distant via un canal de communication. Dans ce qui suit nous allons définir les principaux éléments qui constituent ce système ainsi que sa représentation mathématique.

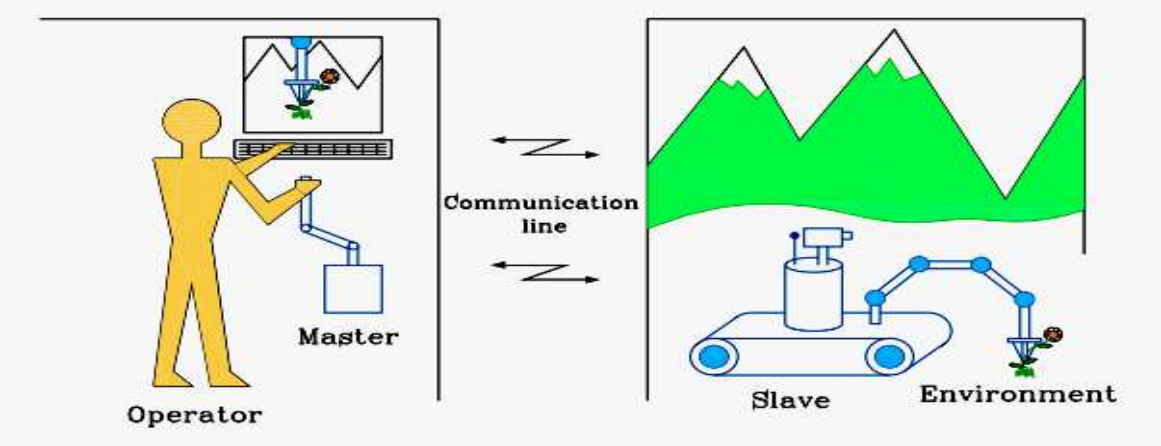


Figure (I.02) : description générale d'un système de télé-opération [13]

I.3.1) Définition

Les principaux éléments d'un système de Télé-opération sont [16] **:**

- **L'opérateur (Operator)** : c'est l'utilisateur qui réalise une tâche donnée sur un environnement distant. Ses actions sont véhiculées à travers tout le schéma de télé-opération pour atteindre le site en question. Cet opérateur reçoit en échange un retour en effort représentant les interactions avec l'environnement distant. Il peut aussi recevoir, suivant l'application, un retour visuel diffusé sur un moniteur vidéo.
- **Environnement distant (Environemet)**: il est composé de divers objets avec lesquels il est possible d'interagir. Physiquement, c'est le manipulateur esclave qui manipule ces objets suivant les demandes de l'opérateur. Ces objets peuvent être apriori connus (structurés), avec un certain degré de connaissance ou bien encore totalement inconnus.

- **Interface maître (Master)** : c'est une interface haptique capable de restituer une perception des interactions avec l'environnement distant. L'opérateur la manipule comme s'il réalisait directement une tâche sur l'environnement distant. Cette interface est interconnectée avec le bloc de sensation.
- **Manipulateur esclave (Slave)** : c'est le robot qui se charge de reproduire et d'imiter le comportement et les gestes de l'opérateur au niveau du site distant. Il doit en plus être capable, en utilisant toute l'instrumentation qu'il possède, de fournir et transmettre les informations nécessaires au site maître et donc à l'opérateur.
- **Canal de transmission** (**Communication line**): c'est le support qui permet la transmission de données d'un site à un autre. Suivant la position physique d'un site par rapport à l'autre, la transmission peut se réaliser à travers des câbles, des ondes radio etc . . .. Connu ou estimé, le retard de transmission qui caractérise ce support est utilisé pour régler et optimiser les performances du schéma de télé opération.

La **Figure (I.03)** représente un schéma de télé-opération bilatéral dans lequel un site local communique et interagit avec un site distant .

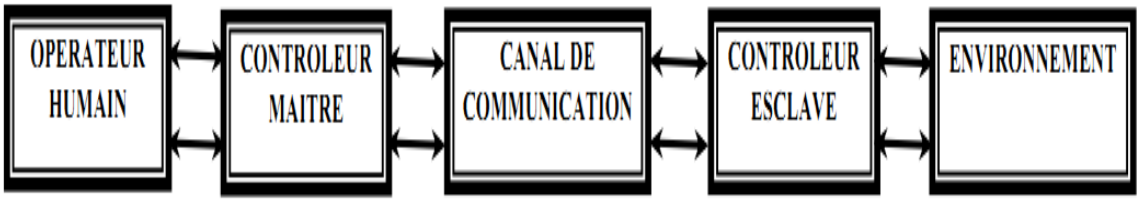


Figure (I.03): Schéma représentant les différents blocs du système de télé-opération bilatéral

I.3.2) Représentation mathématique des systèmes de télé-opération

Pour analyser les systèmes de télé-opération et étudier leur transparence, « **Hannaf ord »** a proposé de les représenter sur base d'un modèle 2-ports (c'est-à-dire : deux canaux de communication : le premier canal transmis la position du maître vers l'esclave et le deuxième canal transmis la force de contacte entre l'esclave et son environnement vers le maître) comme s'est montré dans la **figure (I.04),** dérivé des réseaux électriques. Un système de télé-opération peut être

considéré comme une boite noire qui définit une relation entre les positions ($\boldsymbol{\theta_m, \theta_s}$) du maître et d'esclave et les forces d'interaction $\mathbf{(T_h, T_e)}$ entre l'opérateur et son l'environnement. Cette relation est exprimée par la forme matricielle donnée en (I.01). [17]

$$\begin{bmatrix} \mathrm{T_h} \\ \theta_s \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} * \begin{bmatrix} \theta_m \\ T_e \end{bmatrix} \qquad \text{(I.01)}$$

Cette matrice est la matrice d'immittance hybride H de dimension 2x2. Elle caractérise complètement le système de télé-opération (dynamique du maitre et de l'esclave, contrôleurs, communications). Dans ce formalisme hybride, les paramètres $h_{ij}(s)$ définissent les relations entre les positions et forces du maître et l'esclave. Ils seront utilisés pour étudier les performances et la stabilité du système. Ces paramètres peuvent être interprétés physiquement :

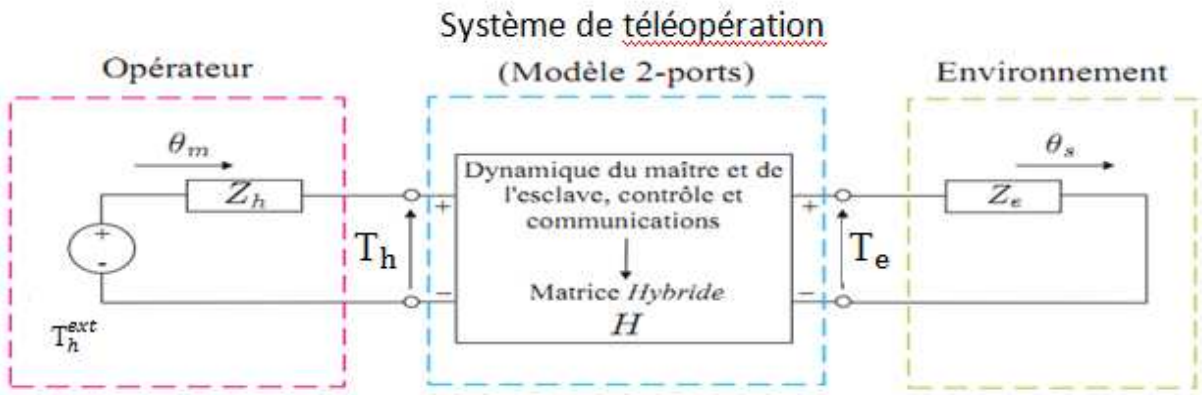
- Pour $\boldsymbol{T_e = 0}$ **(**c'est-à-dire la force de l'interaction entre l'esclave et son environnement est nulle) :

- $\boldsymbol{h_{11} = \frac{T_h}{\theta_m}}$ c'est **l'impédance ressentie par l'opérateur** lorsque l'esclave est en mouvement libre (aucune interaction avec l'environnement). Ce terme doit donc être minimisé et idéalement égal à zéro.
- $\boldsymbol{h_{21} = \frac{\theta_s}{\theta_m}}$; c'est la fonction de transfert du **Tracking en Position**, en mouvement libre, qui doit tendre vers 1 sur la plage de fréquence la plus large possible. (idéalement $\rightarrow = 1$)

- Pour $\boldsymbol{\theta_m = 0}$ **(**c'est-à-dire il n'a pas de transmission de position du maître vers l'esclave) :

- $\boldsymbol{h_{12} = \frac{T_h}{T_e}}$; c'est le **Tracking en force**, lors d'un contact, lorsque la position du maître est maintenue constante.
- $\boldsymbol{h_{22} = \frac{\theta_s}{T_e}}$; c'est **l'admittance de contact à l'esclave**. Pour une position du maître fixe, elle représente **le mouvement de l'esclave en présence d'une force de contact**. Elle aura pour effet de diminuer la raideur présentée à l'utilisateur par rapport à celle de l'environnement.

Les deux derniers termes sont exprimés dans la situation où **la position du maître est maintenue constante**. Cela n'est cependant pas facilement réalisable en pratique. Alors (Aliaga et al.) ont donc proposé de définir un set de quatre critères de performance facilement identifiable par l'expérience. Plus de détaille dans la référence [08]



Figure(I.04): Représentation d'un système de télé-opération par modèle 2-ports

En théorie, pour une analogie parfaite avec le formalisme des réseaux électriques, il faudrait exprimer la matrice H en fonction des vitesses du maître et de l'esclave. Cependant, nous avons choisi ici d'utiliser les positions étant donné que dans la grande majorité des cas, c'est la position et non la vitesse qui est mesurée.

Suivant le nombre d'actionneurs utilisés, d'autres représentations ou bien architecture peuvent être considérées. Citons par exemple la représentation en impédance (ou admittance) qui relie les deux forces aux deux positions du maitre et de l'esclave (ou inversement) [17].

I.4) Architectures de contrôle des systèmes de télé-opération bilatérale

Les principales architectures de contrôle mises en œuvre dans les systèmes de télé-opération haptique sont habituellement classées par rapport au nombre et au type d'informations échangées entre le maître et l'esclave. Les stratégies les plus simples implémentent un échange de deux signaux (un dans chaque sens), basés sur les mesures de position et de force. Les stratégies les plus avancées utilisent un nombre supérieur de canaux de communication pour permettre d'atteindre de meilleures performances. Le schéma de télé-opération générique à **4 canaux** proposé par Lawrence est certainement le plus connu et le plus utilisé par la communauté scientifique. Ce schéma constitue une base à partir de la quelle il est possible de retrouver toutes les Autres architectures. Parmi elles, nous en extrairons les deux plus utilisées à savoir le schéma **position –position** et **position - force** [16].

I.4.1) Architecture de télé-opération 4 canaux

Le schéma de télé-opération proposé par Lawrence et donné sur la Figure **(I.05**), c'est un schéma à 4 canaux (deux canaux pour la transmission de la position et deux pour la transmission de la force) à travers lesquels les sites local et distant s'échangent mutuellement leurs positions et les efforts qui leurs sont appliqués. Ce schéma à 4 canaux a été proposé afin de garantir une transparence parfaite et des marges de stabilité satisfaisantes surtout pour les systèmes de télé-opération bilatérale. Il consiste à échanger simultanément les quatre informations de position et de force entre le maître et l'esclave. Cette implémentation nécessite donc un capteur de position et un capteur de force de chaque côté.

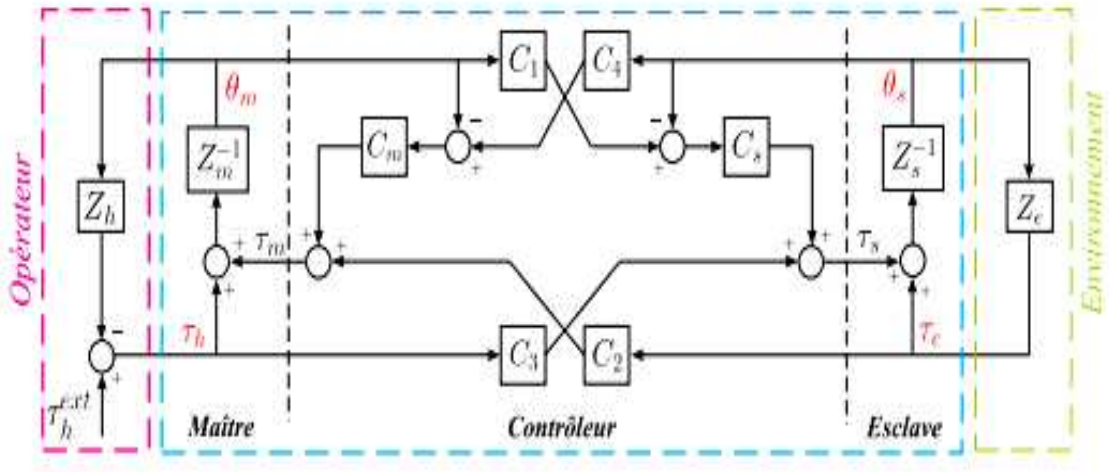


Figure (I.05) : Contrôle à 4 Canaux (4C) [17]

Avec : $C_1, C_2, C_3, C_4,$ sont des actionneurs.

I.4.2) Architecture de télé-opération position-position

Le schéma de télé-opération position-position s'obtient en fixant $C_2 = C_3 = 0$ dans l'architecture à 4 canaux donnée en figue **(I.05**). Dans ce schéma, seules les positions des manipulateurs sont utilisées. Des effets haptiques peuvent être obtenus sans recourir à l'utilisation d'un capteur de force. La méthode de contrôle Position-Position se base exclusivement sur l'échange des mesures de positions entre le maître et l'esclave. Le retour d'effort est alors créé sur base de l'erreur de position entre les deux entités. Historiquement, c'est la première méthode qui a été implémentée sur les systèmes de télé-opération haptique. Le schéma de contrôle de la méthode Position -Position est illustré à la **Figure (I.06).** Par rapport à la stratégie Force-Position, le renvoi du signal de force de l'esclave vers le maître a été remplacé par le signal de position, à travers le

canal de communication C_4. Chaque côté est contrôlé en position sur base des consignes échangées.

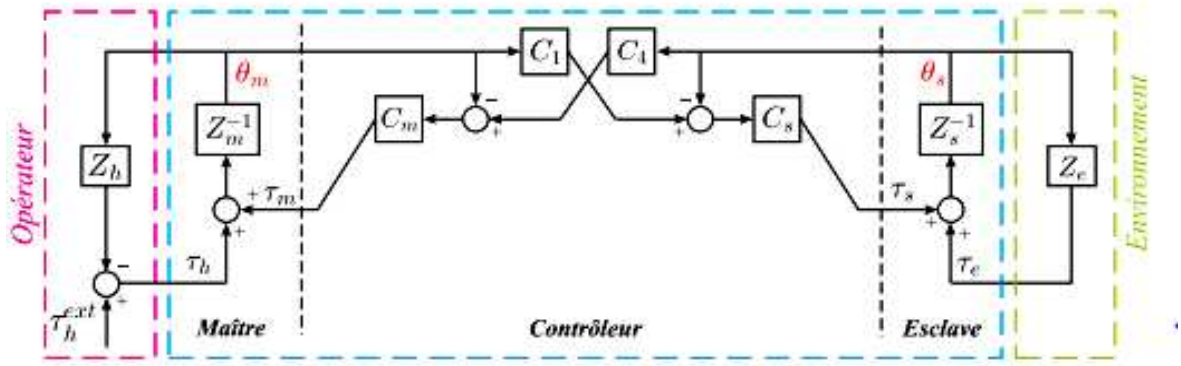


Figure (I.06) : Contrôle position-Position (PP) [17]

I.4.3) Schéma de télé-opération Force-Position (FP)

En choisissant $C_3 = C_4 = 0$ dans le schéma à 4 canaux de Lawrence, le schéma de télé-opération obtenu se réduit à un schéma **Position-Force** donné par la Figure **(I.07**). Dans ce cas, seule la position de la station maître est transmise au manipulateur distant. À l'inverse, le manipulateur esclave transmet uniquement les efforts qu'il applique sur l'environnement à l'opérateur. Par rapport au schéma 4 canaux, un seul capteur d'effort est utilisé et uniquement deux données sont transmises d'un coté à l'autre du système. Ce schéma de télé-opération permet à l'opérateur de percevoir les vrais efforts appliqués par le manipulateur esclave sur son environnement [17].

La méthode de contrôle Force-Position c'est la méthode utilisé dans notre projet. Elle est également appelée contrôle par impédance ou en anglais Direct Force Feedback. Cette méthode est la plus naturelle à imaginer pour un retour de force. En effet, elle est basée sur la mesure de l'interaction entre l'esclave et l'environnement par un capteur de force, ainsi la force d'interaction est calculée par la simulation et renvoyée comme consigne vers le maître.

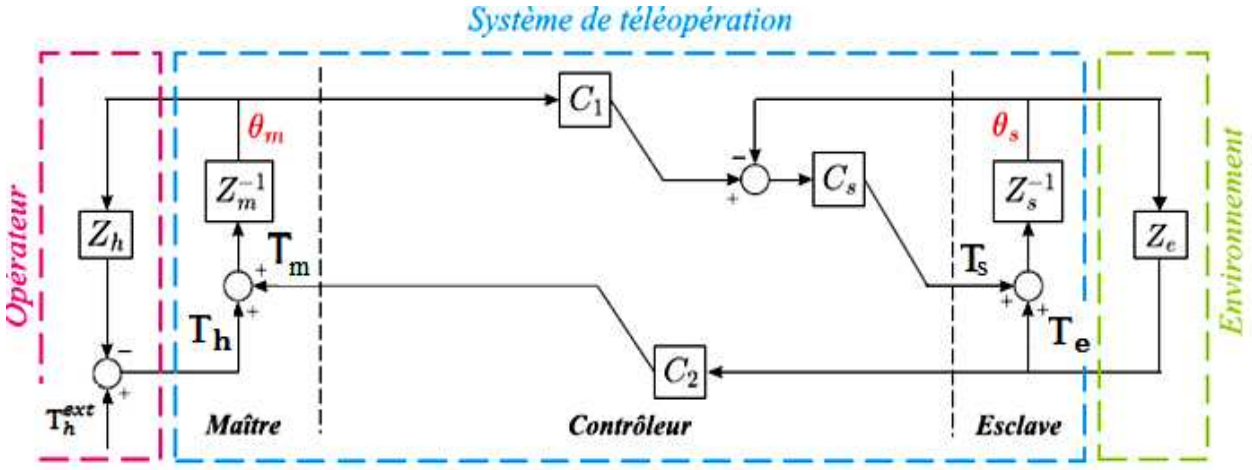


Figure (I.07) : Contrôle Force-Position (FP)[17]

La position de l'utilisateur $\boldsymbol{\theta_m}$ est mesurée par l'encodeur du maître. Et elle sera envoyée comme consigne vers l'esclave, à travers le actionneur (canal de communication) $\boldsymbol{C_1}$. La différence avec la mesure de l'encodeur du robot $\boldsymbol{\theta_s}$ est alors injectée dans le contrôleur de position $\boldsymbol{C_s}$ pour calculer la commande motrice $\mathbf{T_s}$. En même temps, l'interaction entre l'esclave et son environnement $\boldsymbol{T_e}$ est mesurée par le capteur de force, et sera renvoyée comme consigne vers le maître, à travers le canal de communication $\boldsymbol{C_2}$, pour générer le retour d'effort $\mathbf{T}^1_{\boldsymbol{m}}$ Les commandes des moteurs d'après la figure (I.07) sont donc définies par :

$$\begin{cases} \mathrm{T}_m = C_2\, T_e \\ \mathrm{T}_s = C_s(C_1\theta_m - \theta_s) \end{cases} \qquad \text{(I.02)}$$

Avec : [17]

$\boldsymbol{C_s = (k_d^s s + k_p^s)}$: La fonction de transfert du contrôleur de position de type Proportionnel / Dérivé (contrôleur PD)

- $\boldsymbol{C_1}$ **:** le bloc qui permet de transférer la position du maître vers l'esclave.
- $\boldsymbol{C_2}$: le bloc qui permet de transférer la force entrainée par l'esclave vers le maître.

Les relations de $\boldsymbol{C_1}$ $\boldsymbol{et}$ $\boldsymbol{C_2}$ sont données par les équations (I.03) et (I.04) respectivement

$$C1 = \mu e^{-s\Delta 1} \qquad \text{(I.03)}$$

$$C_2 = \beta e^{-s\Delta 2} \qquad \text{(I.04)}$$

[1] $\boldsymbol{T_m}$ ***Dans notre projet est égale au signale de modulation de largeur d'impulsion (PWM)***

- $\boldsymbol{\mu}$: Gains d'amplification de position
- $\boldsymbol{\beta}$: Gains d'amplification de force
- $\boldsymbol{\Delta 1}$ **et** $\boldsymbol{\Delta 2}$: les délais dans les communications

Remarque : La transformé de Laplace d'un retard d'une durée Δ est donnée par $e^{-s\Delta}$. Cela conduit aux relations données par les équations (I.03) et (I.04)

I.5) Analyse des systèmes de télé-opération

L'analyse des systèmes de télé-opération se fait à travers l'évaluation de la stabilité et de la transparence. La stabilité est une nécessité de fonctionnement indispensable en tant que système dynamique. La transparence, quant à elle, se définit plutôt comme un critère de qualité relatif à la perception de l'opérateur.

I.5.1) Stabilité

Pour les systèmes à retour d'effort, il n'ya pas de problème tant que le robot esclave est libre (la force retournée à l'opérateur est nulle), par contre les contacts poseront de sérieux problèmes de stabilité. Lorsque l'on considère un robot en contact avec l'environnement, la stabilité de la commande en boucle fermée dépend non seulement de la dynamique du robot, mais aussi de celle de l'environnement.

Pour notre application, la stabilité de la commande peut être obtenue en se basant sur un modèle connu de l'environnement. L'outil privilégié pour garantir la stabilité de tel système est l'analyse de la passivité. Plus précisément, on cherche à ce que le port d'interaction (correspondant au point d'échange de puissance entre le système et la source d'énergie extérieur) entre le robot et l'environnement soit passif (ne crée pas d'énergie). Une passivité de ce port vérifiée, le système interagira via ce port, de façon stable, avec n'importe quel environnement passif. Donc la stabilité est déduite de la passivité.

Pour étudier la stabilité et la transparence on utilise les vitesses du maître et de l'esclave$\left[\dot{\theta}_m \; et \; \dot{\theta}_s\right]$ et pour les obtenir, il suffit de dériver leurs positions$[\theta_m \; et \; \theta_s \;]$.

- **Stabilité par la passivité**

Le système de communication peut être modélisé par un quadripôle à 2 ports caractérisé par sa matrice hybride H(s) **Figure (I.08).**

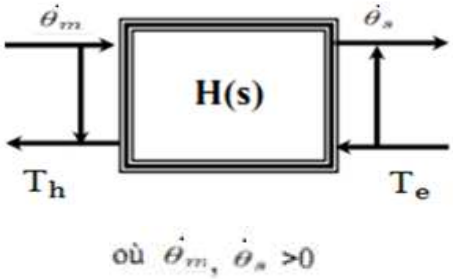


Figure (I.08) : Quadripôle schématisant le bloc de communication

Les entrées de ce quadripôle sont la vitesse de la manette $\dot{\theta}_m$ et la force de contacte entre l'esclave avec son environnement T_e, les sorties sont la vitesse de l'esclave $\dot{\theta}_s$ et la force de l'opérateur T_h

On a choisi le sens des flèches indiquée par la **figure (I.08)** pour respecter l'emplacement des dispositifs maître et esclaves.

La référence est choisie de façon à ce que les vitesses dans le quadripôle soient positives. La matrice hybride de ce système est définie par la relation (I.05) [21]:

$$\begin{bmatrix} T_h(s) \\ \dot{\theta}_s(s) \end{bmatrix} = \begin{bmatrix} h_{11}(s) & h_{12}(s) \\ h_{21}(s) & h_{22}(s) \end{bmatrix} * \begin{bmatrix} \dot{\theta}_m(s) \\ T_e(s) \end{bmatrix} \quad \text{(I.05)}$$

S : est la variable de Laplace.

La condition nécessaire et suffisante pour la stabilité est extrêmement difficile à obtenir puisque le système est multi-variables (plusieurs entrées, plusieurs sorties). Cependant, une condition suffisante pour la stabilité basée sur la passivité est bien adaptée à cette application.

Cette méthode permet en plus de s'affranchir des contraintes imposées par les impédances de l'opérateur et de l'environnement éloigné, et ainsi de garder le système stable quels que soient ces deux derniers facteurs. On définit les vecteurs suivants [21]:

$$T(t) = [\, T_h(t) \;\; T_e(t)]^t \;\; et \;\; \dot{\theta} = \left[\, \dot{\theta}_m \; et \; \dot{\theta}_s \right]$$

Qui sont des fonctions de temps de carré sommable. (t dénote la transposition). Alors le système est passif si on a :

$$\int_0^t T^t\ (t)\ \dot{\theta}(t)\ dt = -S_0 \quad t > 0 \tag{I.06}$$

S_0 : Est un nombre fixé souvent associé avec une énergie initiale stockée à t = 0.

Si le système est initialement relaxé, c'est-à-dire si les sources du système sont nulles à t = 0, et aucune des inerties n'a une vitesse initiale, alors $S_0 = 0$. Et lorsque l'inégalité devient une égalité, le système est dite non dissipatif où sans pertes. Plus concrètement, on voit que $\boldsymbol{T^t\ (t)\ \dot{\theta}(t)}$ représente une puissance et que son intégration représente une énergie. Le système est passif si l'énergie qu'il voit entrer est plus importante que celle qu'il fournit à l'extérieur. On comprend ainsi qu'un système passif est nécessairement stable. A partir de la matrice hybride du système on peut déterminer si celui-ci est passif ou non, on définit matrice de dispersion S(s) du système définie par l'expression suivante [19,20, 22]

$$T^t\ (s) - \dot{\theta}(s) = S(s)(T^t\ (s) + \dot{\theta}(s)) \tag{I.07}$$

Explication : d'après l'équation donnée en relation (I.05) on peut exprimer les forces en fonction des vitesses : $\mathrm{T_h(t)} = h_{11}\dot{\theta}_m + h_{12}T_e$ et $T_e = (\dot{\theta}_s - h_{21}\dot{\theta}_m)/h_{22}$.

On trouve d'après les relations (I.05) et (I.07) que

$$S(s) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} (H(s) - 1)(H(s) + 1)^{-1} \tag{I.08}$$

Théorème [19,20] :

Un système est passif quelles que soient ses entrées $\boldsymbol{T\ et\ \dot{\theta}}$ si et seulement si $\boldsymbol{|S| = 1}$

I.5.2) Transparence

La notion de transparence est associée aux performances d'un système de télé opération. La transparence parfaite (idéale) est atteinte lorsque l'opérateur a l'impression de manipuler directement les objets de l'environnement, sans ressentir ni les effets dynamiques du manipulateur maître, ni ceux du manipulateur esclave. C'est à dire le robot esclave doit dupliquer exactement les déplacements du robot maître. Et le robot maître doit restituer le plus fidèlement possible à l'opérateur les efforts exercés par l'esclave sur son environnement. Dans la littérature, plusieurs

définitions de la transparence ont été proposées dont **la correspondance d'impédance entre celle ressentie par opérateur et celle de l'environnement,** qui permet d'exprimer le degré de transparence en termes d'impédance ressentie par l'utilisateur [08,16].

- **Définition de la Transparence**:

Par analogie avec l'électricité, on peut modéliser le système de télé-opération par un ensemble de blocs en cascade, représentant le site maître, le site esclave et entre les deux le réseau de communication identifié par un quadripôle à deux ports de matrice hybride H(s) **figure (I.09)** [22]**.**

Figure (I.09) : Représentation sous forme de réseau d'un système de télé opération.

Les forces sont analogues à des tensions et les vitesses à des courants Cette analogie est possible si on considère que le système de télé-opération est linéaire. Lorsque le robot esclave est en contact avec son environnement, la vitesse de l'effecteur de l'esclave $\dot{\theta}_s(\boldsymbol{t})$ et la force $\mathbf{T_e}$ qu'il exerce sur son environnement ne sont plus indépendantes. Elles sont reliées par l'impédance $\boldsymbol{Z_e}$ transmise par l'environnement de l'esclave [23].

$$\mathrm{T_e} = Z_e(\dot{\theta}_s) \tag{I.09}$$

Si l'opérateur perçoit l'environnement distant comme s'il était sur place, alors la force exercée par l'opérateur sur le bras maître $\boldsymbol{T_h}$ et la vitesse du bras maître $\dot{\theta}_m$ devraient avoir la même relation : c'est-à-dire pour les mêmes forces $\mathbf{T_e} = \boldsymbol{T_h}$ on désire obtenir les mêmes vitesses, soit $\dot{\theta}_m = \dot{\theta}_s$. Ceci nécessite que l'impédance transmise $\boldsymbol{Z_{th}}$ou ressentie par l'opérateur, soit définie telle que

$$T_h = Z_{th}\,(\dot{\theta}_m) \tag{I.10}$$

Donc, pour que la transparence soit parfaite, il faut satisfaire les conditions de transparence données par (I.11) :

$$\begin{cases} \boldsymbol{Z_{th}} = Z_e \\ \dot{\theta}_s = \dot{\theta}_m \end{cases} \quad \text{(I.11)}$$

Remarque :

Le rapport $Z_{th}(\boldsymbol{s})/\boldsymbol{Z_e}(\boldsymbol{s})$ est utilisé par certains auteurs pour définir la transparence du système de télé opération. D'après Lawrence, pour que la transparence d'un système de télé-opération, soit idéale, il faut que ce rapport tende vers 1 c'est-à-dire $(\boldsymbol{Z_{th}}(\boldsymbol{s})/\boldsymbol{Z_e}(\boldsymbol{s}) \rightarrow \mathbf{1})$. Relation (I.11)

I.6) Conclusion

Dans ce chapitre, nous avons introduit la notion des systèmes de télé opération. Nous avons présenté le formalisme mathématique approprié à leur description. Et comme l'analyse des systèmes de télé-opération se fait à travers l'évaluation de la stabilité et de la transparence, ainsi nous avons présenté la notion de la stabilité et les critères de la transparence. Les méthodes de contrôles des architectures de télé-opération ont été présentées. La méthode de contrôle **force-position** est la plus déraillé dans ce chapitre. Cette méthode est la plus simple à mettre en œuvre, en termes de capteurs et de contrôle, pour réaliser une télé-opération haptique. Elle n'échange que deux signaux entre le maître et l'esclave. C'est la méthode que nous allons utiliser dans notre travail.

L'utilisation un circuit programmable intégré FPGA rend la mise en œuvre du contrôle plus compliquée. Toutefois, cela permet d'atteindre une transparence idéale et d'obtenir un système plus performant. En pratique, l'avantage principal d'utiliser un circuit intégré FPGA, est la qualité de transmission de la raideur en raison de son parallélisme de solution et sa grande capacité de calcule. La stratégie de contrôle de ces systèmes en utilisant les FPGA permet d'utiliser des interfaces haptiques avec une très forte dynamique sans connaissance de son modèle et sans contrôle local en force. Cette stratégie sera détaillée dans le chapitre 3 sur un système maître/esclave à 1 ddl afin d'illustrer les aspects présentés. Cependant l'utilisation des cartes FPGA exige une très bonne connaissance de ses caractéristiques pour assurer une adéquation avec les besoin du travail à réalisé ainsi qu'une maitrise parfaite des outils d'implémentation. Ce qui sera détaillé dans le chapitre suivant.

CHAPITRE II Description de la carte FPGA et le logiciel d'utilisation

Objectif du chapitre

L'objectif du présent chapitre est de présenter, d'une façon générale, les circuits programmables FPGA (Field programmable Gate Array), afin de se familiariser avec ce type de circuits. Nous allons présenter les FPGA de la famille XLINX en s'intéressant particulièrement à la famille Vertex, et plus précisément à la famille Vertx 5 de Cœur ML501. Par la suite, on va aborder l'outil de conception assisté par ordinateur en utilisant le langage de description VHDL.Enfin, on décrit d'une manière globale, la suite de développement ISE 14.2.

II.1) Introduction

Les FPGA se situent entre les réseaux logiques programmables et les circuits logiques près diffusés.

1- **Les réseaux logiques programmables** : sont des composants qui ne nécessitent aucune étape technologique supplémentaire pour être personnalisés, ce sont des circuits standards, programmables par l'utilisateur grâce aux différents outils de développement.

2- **Les prés diffusés :** sont des circuits intégrés basés sur l'utilisation des réseaux de cellules dont les blocs sont préalablement diffusés. Il faut créer les connexions entre ces blocs.

Les FPGA combinent donc à la fois la **souplesse** de la programmation des réseaux logiques programmables et **les performances** des circuits prés diffuses. En d'autres termes le FPGA permet d'avoir une architecture conçue sur mesure à haute densité dans un circuit électronique, et avec la possibilité de modifier cette architecture quand de nouvelles applications apparaissent.

Le langage HDL comme le VHDL ou le Verilog sont utilisés pour décrire les fonctionnalités qui seront implémentées sur le composant. La description matérielle est traduite dans un fichier de configuration pour le FPGA cible. [11]

La technologie FPGA actuellement disponible permet de construire des systèmes qui sont configurables ou reconfigurables. L'idée commune à tous les circuits configurables est de proposer, non pas une fonction figée dont le champ d'application est le plus large possible, mais une structure adaptable. La reconfiguration consiste à modifier, en cours de fonctionnement, l'architecture matérielle. Cela permet de concevoir des circuits intelligents qui peuvent s'auto-adapter à leur environnement.

Les circuits FPGA sont programmables ou reprogrammables sur les cartes implantées par l'utilisateur. Cette reconfiguration est une propriété nécessaire face aux systèmes à charges et contraintes variables. Les dernières technologies qui sont utilisées ont permis la satisfaction des besoins forts en densité. Selon l'architecture des FPGA, chaque point d'interconnexion peut être réalisé selon deux technologies qui définissent deux classes différentes des FPGA à savoir **La technologie de programmation SRAM e**t la technologie de programmation **à base d'inti-fusibl.** La technologie de programmation SRAM permet d'avoir une reconfiguration rapide des FPGA. Les points de connexions sont des ensembles des transistors commandés. Quant la technologie de programmation **à base d'inti-fusible**, les points d'interconnexions sont de type ROM. c'est-à-dire la modification du point est irréversible [03, 05,24].

L'avantage des circuits programmables FPGA est de pouvoir être reprogrammable, Contrairement aux circuits intégrés de type ASIC. Ce qui rend cette solution modulable et donne la possibilité de modifier le programme générique de base afin de le rendre spécifique au circuit utilisé. Ces circuits programmables permettent aussi d'obtenir les meilleurs performances car la technologie FPGA utilise du parallélisme matériel, qui offre une puissance de calcul supérieure à celle des processeurs de signaux numériques (DSP). BDTI, une importante société d'analyse et de « *benchmarking* », a publié des études montrant que les FPGA peuvent offrir une puissance de traitement par dollar plusieurs fois supérieure à celle d'une solution DSP dans certaines applications. Contrôler les entrées et sorties (E/S) au niveau matériel permet d'obtenir des temps de réponse plus courts ainsi que des fonctionnalités spécifiques, qui répondent mieux aux besoins de l'application. [25,26]

Actuellement, on trouve sur le marché des circuits FPGA de faible, moyenne et haute densité produits grâce aux deux principaux producteurs de circuits logiques programmables :

Xilinx et Altera. Sur le même marché, on trouve plusieurs autres producteurs de circuits FPGA, on peut dont : Actel, Abound Logic, Achronix, Atmel, Cypress, Lattice Semiconductor, etc. L'apparition nombreuses des FPGA sur le matché mondial ont révolutionné certains domaines de contrôle numérique et de plus en plus utilisés pour intégrer des architectures numériques complexe. Ils sont devenus les plus populaires en matière d'implantation et de prototypage des circuits numérique après leur apparition sur le marché en 1984. La clé maitresse de leurs réussites est l'espace de programmation de ces derniers. Leurs utilisations actuelles couvrent les deux domaines : civil et militaire. Parmi ces applications nous citons [03]:

Informatique : périphérique spécialisés

Machinerie industrielle : contrôle des machines

Télécommunication : traitement d'images et filtrage

Instrumentation : équipement médical, prototypage

Transport : contrôle des avions et métrons

Aérospatiale : les satellites

Militaire : radar, communication protégée, la détection de la surveillance

Et autre **….**

II.2) Définition des FPGA

Un FPGA (Field-Programmable Gate Array) est un dispositif semi-conducteur contenant des composants à logique programmable (Blocs logiques) et des interconnexions programmables (routage programmable). Les blocs logiques peuvent exécuter n'importe quelle fonctionnalité depuis les fonctions logiques de base (comme : AND, OR, XOR, NOT) jusqu'aux fonctions complexes combinatoires. Les FPGAs comprennent aussi des éléments de mémoire (bascules, blocs complets de mémoire). Quelque soit la fonction logique, elle peut être réalisée par l'interconnexion des blocs logiques. La principale caractéristique des FPGAs est la reconfiguration qui peut être même effectuée même en cours de fonctionnement. On parlera alors de reconfiguration dynamique. **[04]**

II.3) Architecture interne d'un FPGA [02]

Les circuits FPGA sont structurés sous forme de matrices d'éléments logiques de base, constitués de portes logiques qui présentent physiquement sur le circuit. Les circuits FPGA du fabricant Xilinx utilisent deux types de cellules de base, les cellules d'entrées/sorties appelées IOB et les cellules logiques appelées CLB. Ces différentes cellules sont reliées entre elles par un réseau d'interconnexions configurable. Donc les trois blocs principaux sont les blocs logiques configurables, les blocs d'entrées/sorties et les ressources de communications. Il existe aussi des circuits FPGA qui, contiennent des microprocesseurs. La **figure (II.01)** mous montre l'architecture interne des cartes FPGA.

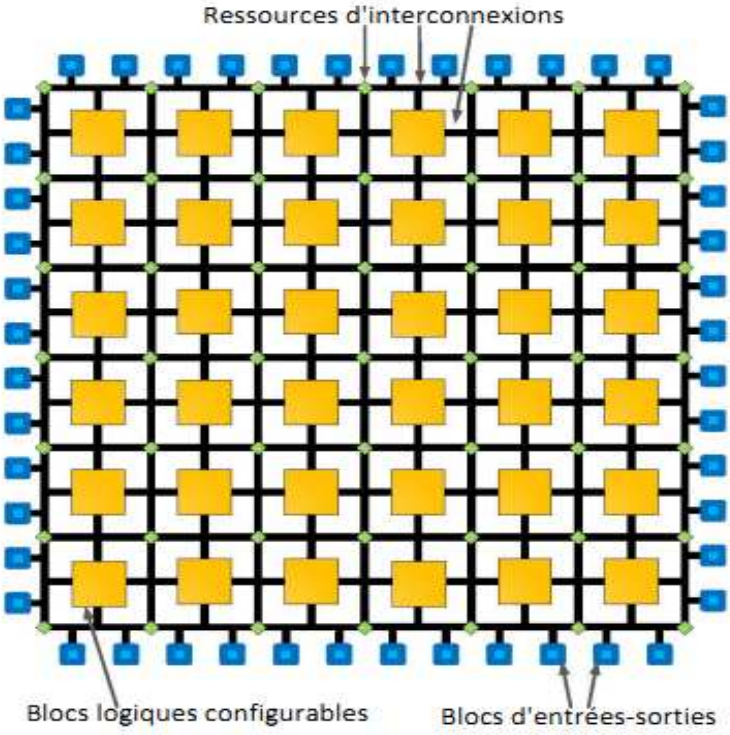


Figure (II.01) : Architecture interne d'un PFGA [02]

II.3.1) Les blocs logiques configurables (CLB)

Les blocs logiques configurables sont les principaux éléments d'un FPGA. Ils peuvent avoir un ou plusieurs générateurs de fonctions réalisées avec des tables de correspondance (LUT -look-up tables) qui peuvent mettre en œuvre une logique arbitraire en fonction de leur configuration. Autour d'une LUT, il y a une logique d'interconnexion qui permet les liaisons à destination et vers la LUT. Elle est mise en œuvre à

l'aide de portes logiques et de multiplexeurs. Pendant le processus de configuration d'un FPGA, les mémoires des LUT sont écrites pour y implémenter une fonction, et la logique qui l'entoure est configurée pour router correctement les signaux afin de construire un système complexe. Il existe différents types de CLB en fonction du FPGA utilisé. Pour Xilinx le bloc logique configurable est appelé slice. L'architecture du slice (avec des modifications mineures) est utilisée dans toutes les familles FPGA Virtex. La **figure (II.02)** nous Présente l'architecture interne d'un CLB de vertex 5 qui contient quatre Slices :

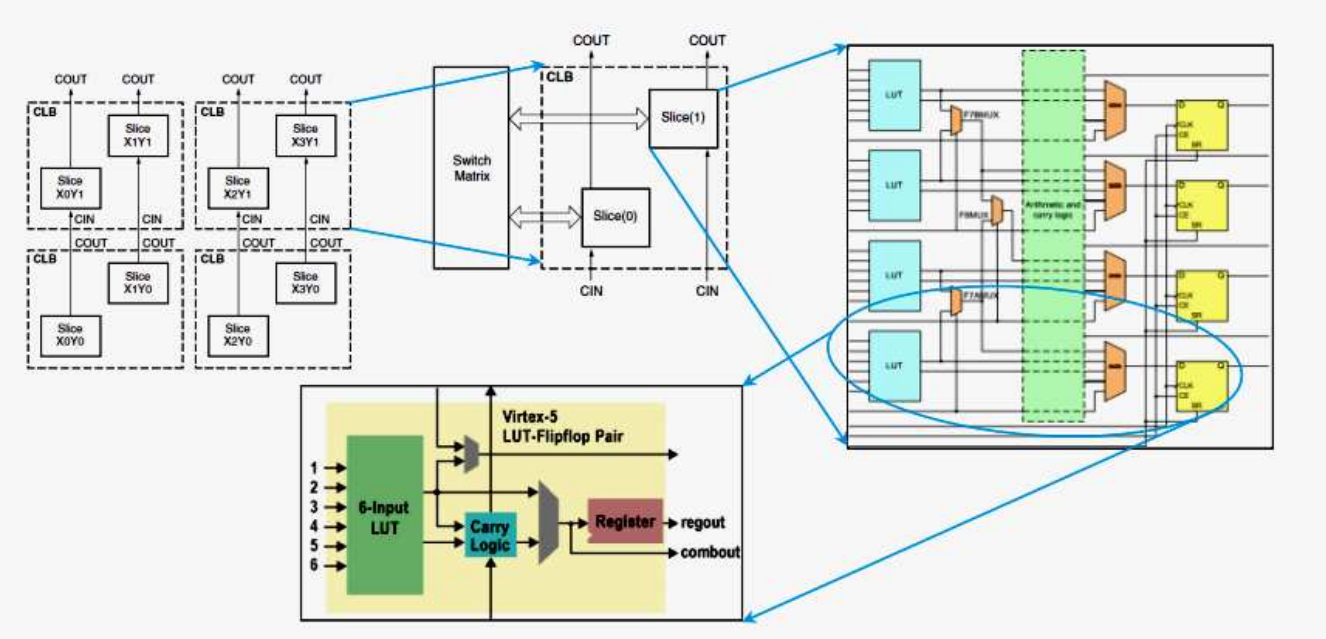


Figure (II.02). Architectures interne d'un CLBs dans les FPGAs Virtex5

II.3.2) Blocs d'entrée-sortie (IOB)

Les blocs d'entrée-sortie permettent l'interconnexion de la logique interne aux ports d'entrées et de sorties du FPGA. Les IOB ont leur propre mémoire de configuration, elle stocke les standards de tension et la direction des ports. Ces blocs sont présents sur toute la périphérie du circuit FPGA. Chaque bloc IOB contrôle une broche du composant et il peut être défini en entrée, en sortie, en signaux bidirectionnels ou être inutilisé (haute impédance).

II.3.3) Ressources d'interconnexions

Les ressources d'interconnexion au sein d'un FPGA, permettent la connexion arbitraire des CLB et des IOB. Les connexions internes dans les circuits FPGA sont composées de segments

métallisés. Parallèlement à ces lignes, nous trouvons des matrices programmables réparties sur la totalité du circuit, horizontalement et verticalement entre les divers CLB. Elles permettent les connexions entre les diverses lignes, à l'aide des transistors MOS dont l'état est contrôlé par des cellules de mémoire vive ou RAM. Le rôle de ces interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible. Pour parvenir à cet objectif, Xilinx propose trois sortes d'interconnexions selon la longueur et la destination des liaisons comme le montre la **Figure (II.03).**

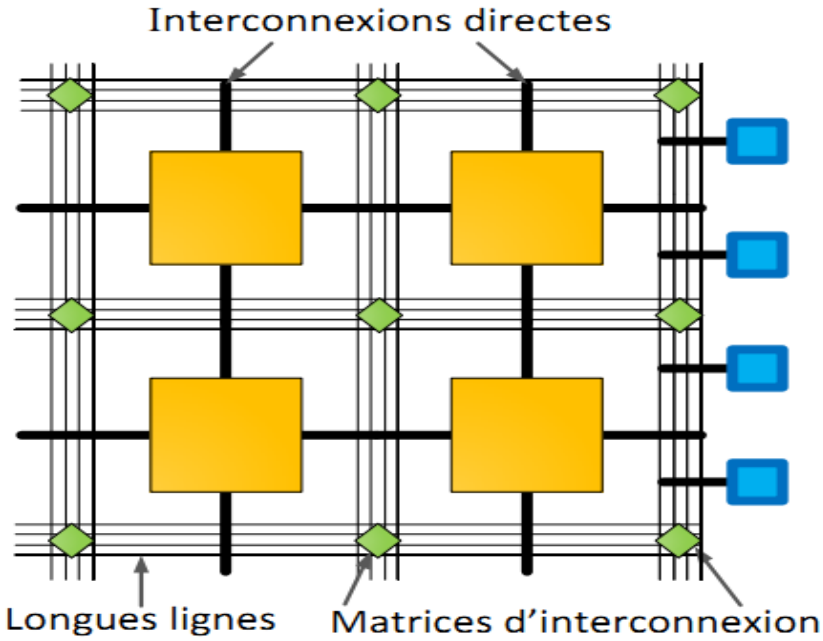


Figure (II.03): Structure générale du routage [02]

- **Interconnexions directes**

Ces interconnexions permettent l'établissement des liaisons entre les CLB et les IOB. Il est possible aussi de connecter directement certaines entrées d'un CLB aux sorties d'un autre.

- **Longues lignes**

Ce sont de longs segments métallisés parcourant toute la longueur et la largeur du FPGA, elles permettent éventuellement de transmettre avec un minimum de retard les signaux entre les différents éléments dans le but d'assurer un synchronisme aussi parfait que possible. De plus, ces longues lignes permettent d'éviter la multiplicité des points d'interconnexion

- **Les matrices d'interconnexions**

Ce sont des aiguilleurs situés à chaque intersection. Leur rôle est de raccorder les longues lignes entre elles selon diverses configurations. Ces interconnexions sont utilisées pour relier un CLB à n'importe quel autre CLB sur le FPGA, selon plusieurs possibilités comme le montre la figure suivante

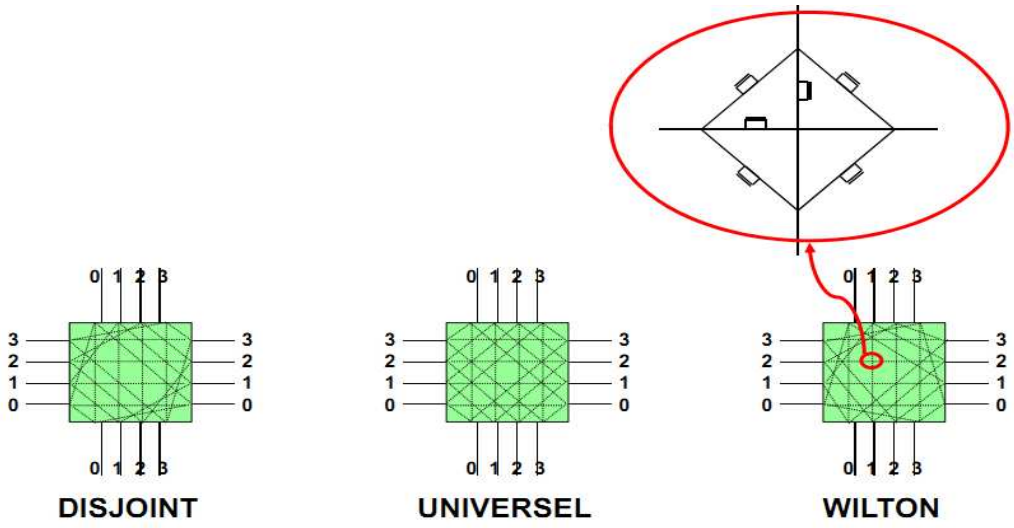


Figure (II.04): La matrice de commutation [27]

Remarque : Certains de ses canaux sont spécifiques aux signaux d'horloges.

II.4) FPGA de la famille XLINX

Xilinx est une entreprise américaine de semi-conducteurs créé en 1984. Inventeur du FPGA avec un premier produit en 1985, Xilinx fait partie des plus grandes entreprises spécialisées dans le développement et la commercialisation de composants logiques programmables, et des services associés tels que les logiciels de CAO électroniques.

Xilinx fabrique une large gamme de FPGA pour diverses applications. En effet, l'offre commerciale de Xilinx est découpée en plusieurs gammes : (FPGA hautes performances : gamme Virtex, FPGA pour la fabrication en grande série : gamme Spartan. Les plus onéreux sont les FPGA Virtex (Virtex II/pro, Virtex4, Virtex5, Virtex6, Virtex7). En 2010, Xilinx a lancé les FPGA-Virtex6, en technologie 40nm, avec 3 catégories différentes (LX, LXT et SXT). Même si les Virtex6 ne disposent pas de processeur PowerPC405, la dernière catégorie est optimisée, spécialement, pour les applications lourdes en traitement numérique du signal. Ensuite,

en 2012, Xilinx a lancé les FPGA-Virtex7 (avec une seule catégorieXC7VxxxT) fabriqués en technologie 28nm. Afin de satisfaire à terme les applications très exigeantes en performances de calcul et en bande passante tout en limitant la puissance consommée, Xilinx va proposer dans le courant du second semestre 2012 des circuits qui couplent plusieurs FPGA Virtex7 au sein d'un même boîtier via une technologie d'interconnexion 3D. Selon Xilinx, le boîtier Virtex7 LX2000T en technologie 28 nm sera le premier FPGA multi-puce au monde et affichera une densité de portes logiques 3,5 fois supérieure au FPGA 40 nm le plus puissant actuellement disponible.

Le tableau (II.02) organise cette famille (virtex) selon l'année de fabrication ainsi que la technologie utilisée et nombre de cellules de chaque une d'elles.

Famille	Année	Technologie	Max cellules
Virtex 4	2004	90 nm	200 000 LUT 4
Virtex 5	2007	65 nm	330 000 LUT 5
Virtex 6	2010	40 nm	760 000 LUT 6
Virtex 7	2012	28 nm	2 000 000 LUT 7

Tableau (II.01): la famille vertex de Xilinx [27]

Les composants Virtex5 sont disponibles, à bon prix, avec plusieurs catégories (LX, LXT, SXT, TXT et FXT). Dans notre travail nous utilisons la carte FPGA de la famille Vitex5 et de catégorie LX avec la plateforme ML501.

II.5) Présentation et description des éléments de FPGA ML 501

Une plateforme d'évaluation FPGA est un circuit imprimé contenant un circuit FPGA, un processeur (éventuellement), des périphériques d'E/S permettant l'interaction avec l'environnement externe. Ces plateformes permettent non seulement de tester de façon rapide et efficace des prototypes de test mais peuvent aussi être utilisées comme solution finale après validation. Pour cela, elles sont équipées de mémoire ROM leurs permettant de garder de façon définitive la solution finale.

Nous utilisons la plateforme d'évaluation XC5VLX50-1FFG676. C'est à dire le cœur de la plateforme ML501 est un circuit FPGA de type Virtex5 XC5VLX50-1FFG676, dont les

caractéristiques sont données dans **tableau (II.03).** La carte ML501 est une plateforme riche en composants pour l'évaluation et le développement des multiples applications. Ces composants permettent un accès facile et pratique aux ressources disponibles dans le circuit FPGA [80]. Les ports de la vidéo, de l'audio et les ports de communication ainsi que les ressources mémoires généreuses donnent la fonctionnalité et la flexibilité à cette plateforme et la rendent une plateforme de développement FPGA typique. La plateforme ML501 dispose donc de : [04, 30,31]

- **Mémoires** : Xilinx Platform Flash XCF32P ; 9Mb ZBT synchronous SRAM ; 32MB Intel P30 Strata Flash ; 2MB SPI Flash ; System ACE™ Compact Flash configuration controller avec un connecteur Compact Flash ; IIC-EEPROM de 8Kbits.

- **Connectivité** : Port JTAG utilisé avec un câble parallèle (III) ; câble parallèle (IV) ou bien plateforme de téléchargement USB ; Ethernet PHY 10/100/1000 tri-speed ; Codec Stéréo AC97 avec connecteurs line-in/microphone, line-out/headphone, et audio difital SPDIF ; Piezo Speaker ; Port PS/2 pour souri/clavier ; des E/S d'extension ; Port série type RS-232 ; Connecteur vidéo (DVI/VGA).

- **Composants divers** : Générateur de système d'horloge programmable ; CPLD type XC95144XL de Xilinx ; support d'oscillateur d'horloge (3.3V) avec un oscillateur à 100MHz ; contrôleur de température et de tension ; boutons-poussoirs à usage général ; afficheur LCD de 16 caractères sur 2 lignes ; entrée d'alimentation ; LED d'indication (d'alimentation, d'initialisation et d'activation) ainsi que plusieurs switchers et ports d'E/S.

La figure (II.05) nous montre le placement des ces composants et autres dans cette carte.

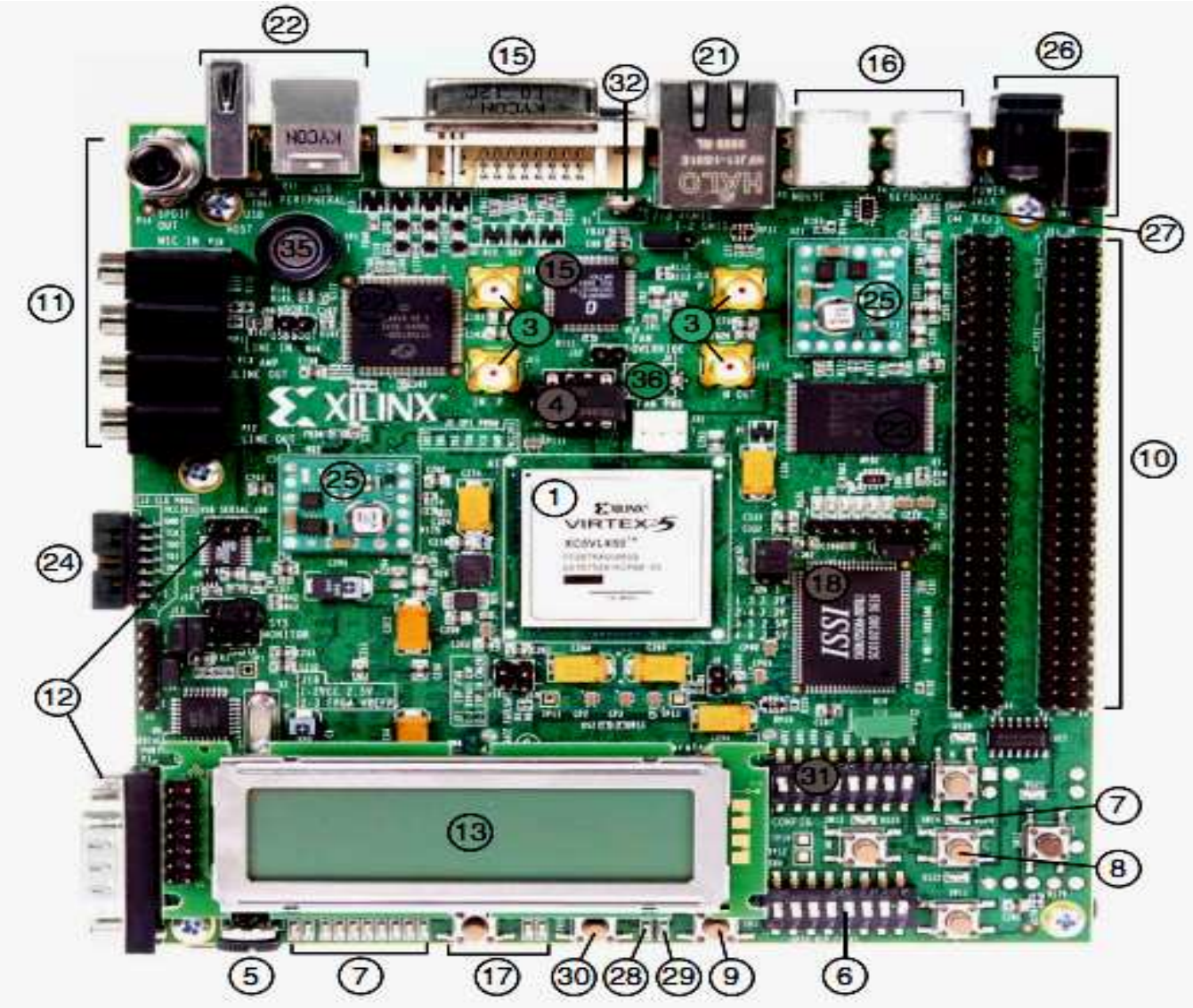


Figure (II.05). La carte Virtex5-ML501 de Xilinx [30].

Le circuit FPGA XC5VLX50-1FFG676 est un circuit de la deuxième génération ASMBL (Advanced Silicon Modular Block) de Xilinx. La puce LX50 intègre également un bloc de gestion des horloges qui permet une répartition d'horloges parfaite, il dispose d'un maximum de 560 E/S **Tableau (II.3)**. Elle dispose aussi de 14 Banks avec des impédances contrôlées, la tension d'alimentation de chaque Bank peut varier de 1,8 à 3,3 Volts. Nous pouvons configurer ce circuit à l'aide de plusieurs dispositifs : câble JTAG, System ACE controller, Platform Flash PROM, Linear Flash memory et SPI Flash memory [28,30].

Circuit FPGA	CLB (Configurable Logic blocks)			DSP 48E	Blocs BRAM			Total I/O Banks	Max User I/O
	(Row×col)	Slices	Max RAM distr.		18 Kb	36 Kb	Max(Kb)		
XC5VLX50	120 × 30	7.200	480 Kb	48	96	48	1.728	17	560

Tableau (II.3) : Les différentes caractéristiques du circuit Virtex5 (LX50)

Dans notre cas nous programmons notre carte FPGA à partir d'un ordinateur en utilisant un câble JTAG avec l'application appropriée IMPACT montré en **figure (II.06**).

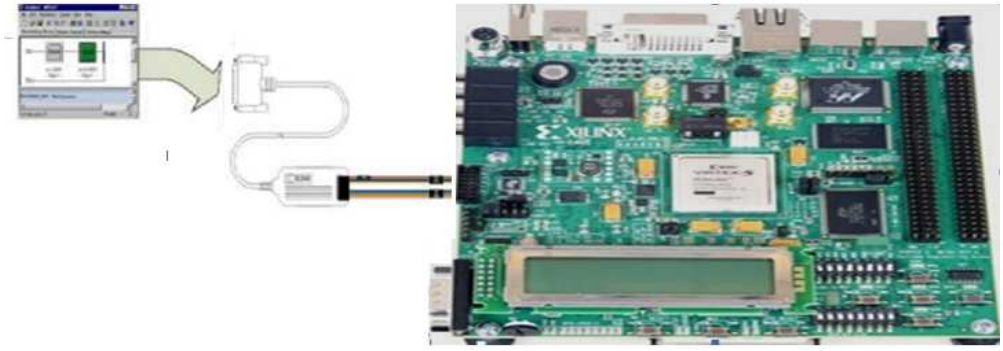


Figure (II.06). Programmation d'un FPGA avec un câble JTAG en utilisant l'application IMPACT**.**

II.6) Méthodologie et étapes de programmation d'un FPGA

La conception des architectures de commande s'effectue en utilisant les outils de CAO. La saisie s'effectuée graphiquement ou via un langage de description matériel de haut niveau, nommé également langage HDL (Hardware Description language). Deux langages HDL sont les plus couramment utilisés, à savoir le VHDL (Verry High speed integrated Hardware Description language) et le Verelog. Ces deux langage sont standardisés et offrent au concepteur différentes niveaux de description, et surtout l'avantage d'être portable et compatible avec toutes les technologies FPGA [11].

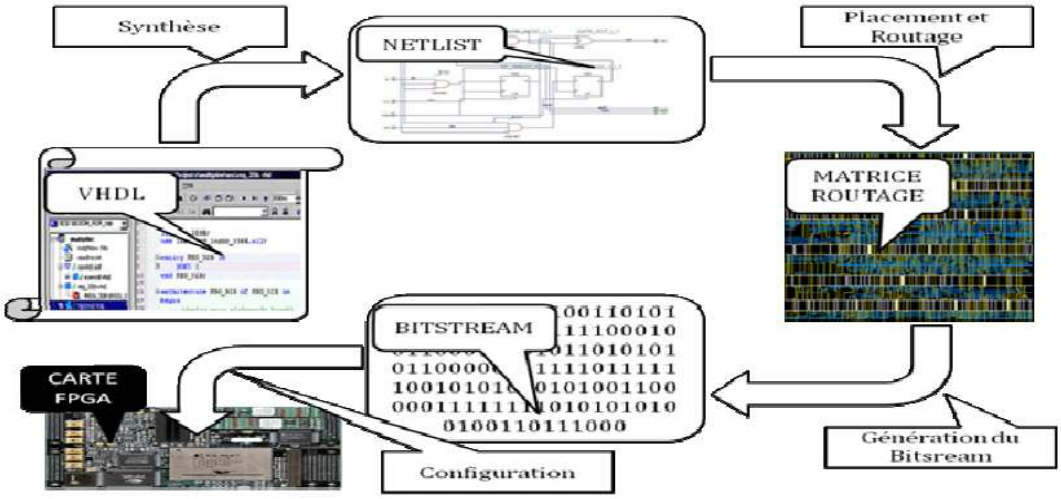


Figure (II.07) : Etapes de conception sur FPGA [24]

La figure **(II.07)** résume les différentes étapes de programmation d'un FPGA. Le synthétiseur des outils CAO génère dans un premier temps, un Netlist qui décrit la connectivité de l'architecture. Puis l'outil de placement-routage place de façon optimale tous les composants et effectue le routage entre les différentes cellules logiques. Ces deux étapes permettent de générer un fichier de configuration à télécharger dans la mémoire de configuration du FPGA. Ce fichier est appelé bitstream. Il peut être directement chargé sur FPGA à partir d'un ordinateur.

L'outil de description HDL de Xilinx permet de développer des solutions sur FPGA en faisant du Co-Design. Le flot de conception Hardware permet d'avoir en finalité des modules qui seront implémentés sur FPGA. la conception et le développement de la partie Hardware suit une méthodologie bien définie sur plusieurs phases jusqu'à son implémentation effective sur FPGA en passant par des étapes de vérification très rigoureuses assurant ainsi son bon fonctionnement. L'outil de simulation Modelsim est aussi très utilisé durant l'étape de vérification. Il permet non seulement la simulation du fonctionnement d'un module mais aussi la simulation du fonctionnement de toute une plateforme FPGA. Les étapes de synthèse et de placement/routage sont accomplies par le langage VHDL de façon transparente en utilisant les outils fournis avec l'environnement ISE (**Integrated Software Environment**) de Xilinx, qui nous allons présenter après la description du Langage VHDL [03.11].

II.7) Langage de description VHDL

L'abriviation VHDL signifie VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit). Ce langage a été écrit dans les années 80 aux États-Unis pour réaliser la simulation de circuits électroniques. Par la suite a été étendu en lui rajoutant des extensions pour permettre la conception (synthèse) de circuits logiques programmables (P.L.D. Pro-grammable Logic Device) [32].

Le VHDL est un langage de description matériel destiné à représenter le comportement et l'architecture d'un système électronique indépendamment d'un fournisseur d'outils. Et techniquement il est incontrôlable car c'est un langage puissant, moderne qui permet une excellente lisibilité, une haute modularité et une meilleure productivité des descriptions. Il permet de mettre en œuvre les nouvelles méthodes de conception. En outre, le développement du l'ensemble synthétisable du langage VHDL est de mieux en mieux définit. Cependant, la majorité des outils de synthèse sont compatibles avec cette norme, ce qui reflète un véritable standard pour la synthèse automatique avec le langage VHDL [33].

Pagailleurs un langage de description hardware tel que VHDL permet de décrire et **de synthétiser rapidement** des circuits contenants l'équivalent de 5 à 20 milles portes logiques, tandis qu'un même un travail en utilisant les équitations booléennes prendrait quelques mois de travail.

C'est un langage de haut niveau, moderne, puissant qui permet l'écriture du code décrivant des circuits logique complexes. Il offre en plus la possibilité de créer des bibliothèques de composants réutilisables d'un projet à l'autre. Il permet de créer une fonction sans se préoccuper de composant dans laquelle elle sera implémentée. En outre, il ne faut pas donc devenir intimement familier avec l'architecture d'un circuit particulier pour optimiser l'utilisation des ressources et les performances

- **Structure d'une description VHDL**

Une description VHDL est composée de deux parties indissociables à savoir :

1- L'entité (**ENTITY**), elle permet de définir le NOM de la description VHDL ainsi que les entrées et sorties utilisées [32].

Syntaxe :

```
entityNOM_DE_L_ENTITE is
        Port (Description des signaux d'entrées /sorties …);
end NOM_DE_L_ENTITE;
```

Exemple : concevoir l'entité d'un compteur modulo-8 sur l'environnement de développement ISE

```
entity compteur is
    Port ( clk : in  STD_LOGIC;
           Q : out  STD_LOGIC_VECTOR (3 downto 0));
end compteur;
```

2- L'architecture (**ARCHITECTURE**), elle contient les instructions VHDL décrivant le fonctionnement. Elle établit à travers les instructions les relations entre les entrées et les sorties. [32]

Syntaxe:

```
Architecture DESCRIPTION of NOM_DE_LA DESCRIPTION is
        Zone de la declaration ;
begin
        Déclaration de la structure logique ;
end DESCRIPTION;
```

Exemple : concevoir l'architecture de même compteur

```
architecture Behavioral of compteur is
signal Q_BUS_INTERNE : std_logic_vector(3 downto 0);

begin
   process(clk)
     begin
      if(clk'event and clk='1') then

                  Q_BUS_INTERNE <= Q_BUS_INTERNE + 1;
      end if;
      end process;
      Q <= Q_BUS_INTERNE;
end Behavioral;
```

II.8) L'environnement de conception ISE

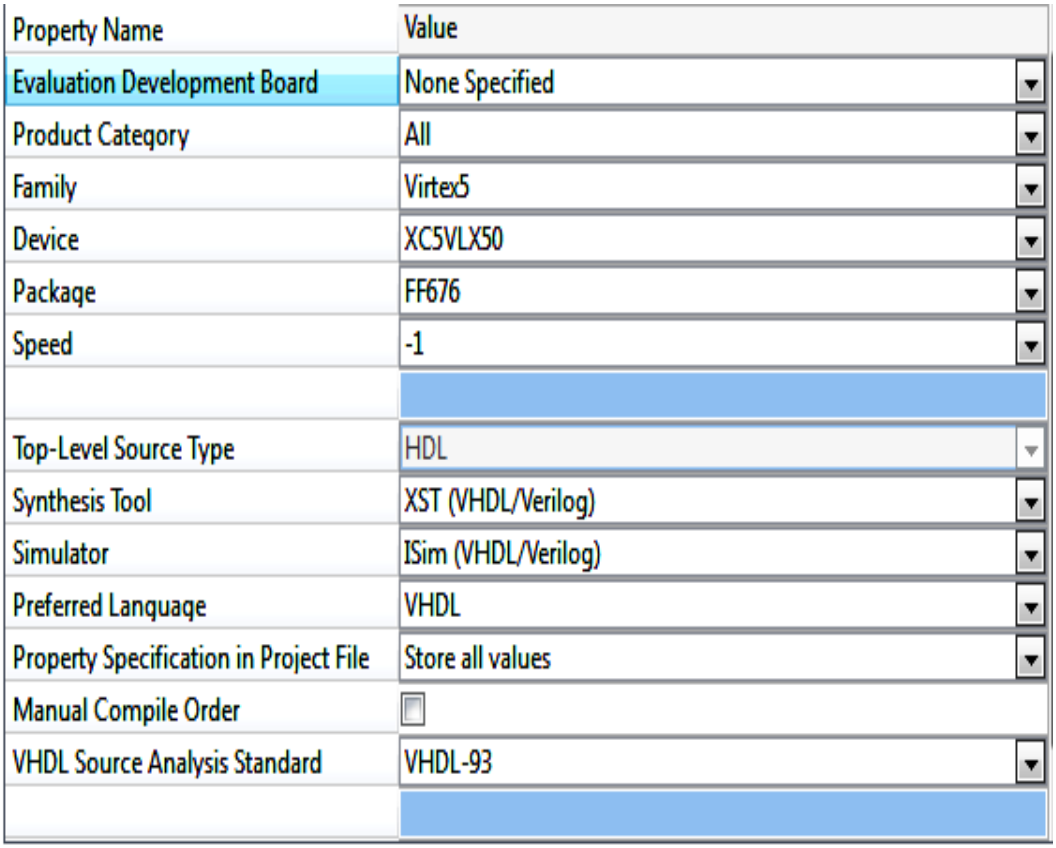
Le Navigateur de projet ISE sera utilisé comme outil de conception dans notre travail. Cet outil de Xilinx permet de créer des projets comportant plusieurs types de fichiers (HDL, schématique, UCF, EDIF, etc.), de compiler, d'effectuer des design rule check (DRC), de créer des contraintes d'implémentation dont des contrainte de timings sur les horloges, de déterminer l'emplacement des broches, et de créer des bancs d'essai de simulation (testbench). Le Navigateur de projet ISE offre un environnement de conception centralisé extrêmement efficace qui regroupe tous les outils nécessaires à la conception, la simulation et à l'implémentation d'un projet d'envergure ainsi qu'à la configuration de la carte, ML501 en ce qui nous concerne**[34]**

.

Pour créer un nouveau projet ISE ayant pour cible le FPGA virtex5 du cœur ML501. Nous allons lancer la suite ISE version 14.2, en faisant double-cliquer sur l'icône du Navigateur de projet ISE, puis sélectionner Start > Programs > Xilinx ISE design suite 14.2 > ISE Design Tolls > 32-bit Project Navigator.

En effet pour créer notre projet, on doit choisir d'abord le nom du projet puis on sélectionne schematic ou VHDL Module pour Top-Level. Par la suite nous allons entrer les valeurs caractérisant la carte ML501 qui sont les suites :

- Product Category: **All**
- Family: **virtex5**
- Device: **XC5VLX50**
- Package: FF676
- Speed: -1
- Synthesis Tool: XST (VHDL/Verilog)
- Simulator: ISE Simulator (VHDL/Verilog)
- Preferred Language: VHDL.



Une fois le fichier VHDL est édité, nous allons synthétiser et vérifier la syntaxe du design afin de trouver des erreurs de syntaxe. Enfin nous vérifions le fonctionnement du design en utilisant le simulateur Isim pour créer un banc d'essai contenant les stimuli d'entrée (testbensh). Après la **synthèse** (transformation en éléments logiques de base), on passe à la phase de **l'implémentation du design** qui consiste à suivre les étapes qui sont décrites précédemment.

Exemple : création d'un code VHDL d'un compteur modulo-8 puis l'implanté sur un caret FPGA.

Pour l'outil de développement, il faut prendre en considération la plateforme cible qui est disponible (**la carte ML501** qui a comme cœur le circuit **FPGA Virtex5 XC5VLX50-1FFG676**). Nous utiliserons la **suite ISE** (**Integrated Software Environment**) version 14.2.

- **1erÉtape** : **Description du code VHDL**

Nous présentons la description du compteur avec le langage de description hardware **VHDL** sur la figure (II.08)

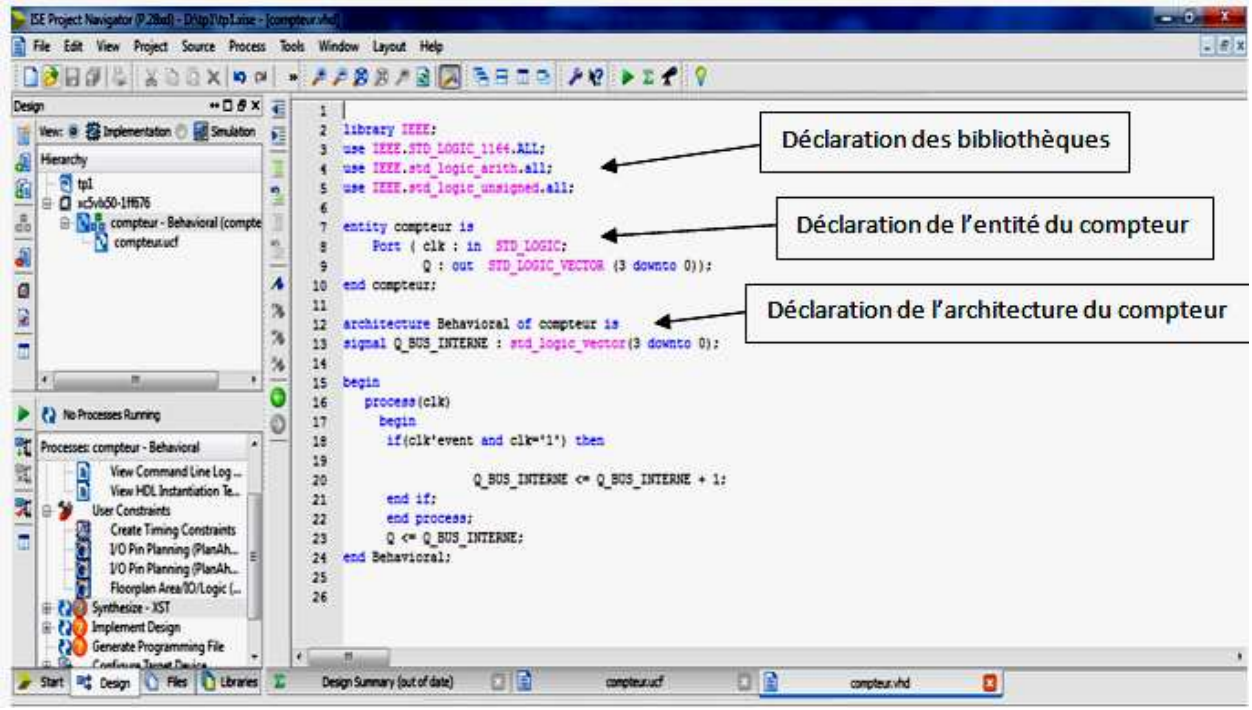


Figure (II.08) : description du code VHDL d'un compteur modulo-8

- **2eme Étape: Synthèse**

La description HDL du circuit est transformé en un assemblage (Netlist) de primitives de base (portes logiques, bascules …).après la synthèse on peut visualiser le niveau RTL comme le montre la figure (II.09):

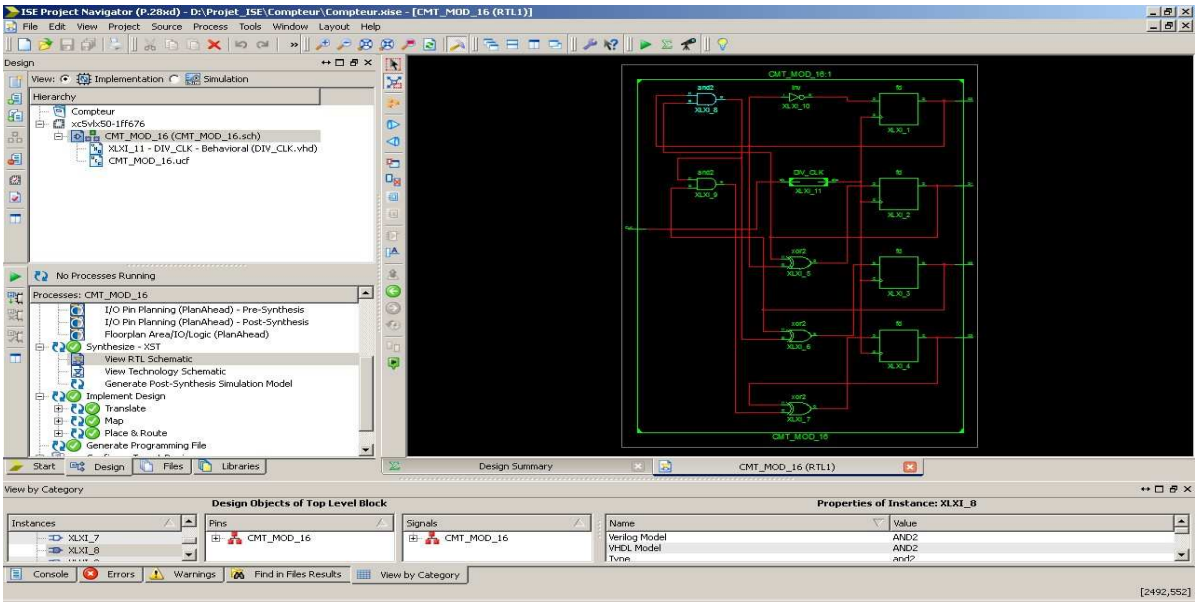


Figure (II.09) : schéma RTL d'un compteur modulo-8

- **3eme Étape**: **Simulation fonctionnelle avec Isim**

Une fois l'étape de conception terminée, nous allons effectuer l'étape de simulation fonctionnelle du projet à l'aide d'un testbench et du simulateur Isim. Pour cela on a généré:

- Un fichier testbench VHDL à l'aide du navigateur ISE.
- Une fois appelé par ISE, Isim prendra en charge la simulation et la présentation des résultats.

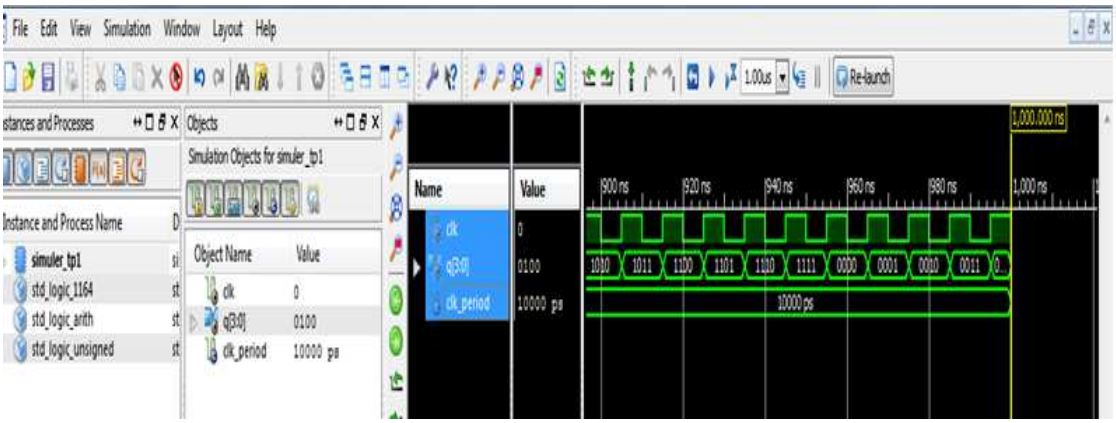


Figure (II.10) : Résulta de simulation du compteur modulo-8

- **4eme Etape: Implémentation du projet:**

L'étape de l'implémentation du projet regroupe trois phases

- **La translation** la Netlist issue de la phase de synthèse en des primitives FPGA.
- **Mappage**: chaque élément de la **Netlist** est associé à un type de cellule logique FPGA.
- **Placement:** qui place les éléments de la Netlist physiquement sur le circuit FPGA.

En utilisant le «Plan Ahead » ainsi que datasheet de la carte ML501 pour associer ces broches

- **5eme Étape:**

Après la phase de l'implémentation du projet, nous allons générer le fichier de programmation « .BIT » pour configurer **la carte ML501**

- **Phase finale**: **Programmation du FPGA en mode JTAG:**

En mode de programmation **JTAG**, il est nécessaire de reprogrammer la carte à chaque remise sous tension car le programme du FPGA est recopié dans la mémoire volatile.

Nous allons lancer le « iMPACT » puis nous téléchargeons le fichier « .bit » dans l'espace de travail et enfin nous cliquons sur « program » pour que notre programme sera transféré dans le support FPGA. C'est-à-dire la carte est configurée.

II.9) Conclusion

Ce chapitre est dédié à la présentation des circuits programmable FPGA (Field programmable Gate Array). Nous avons présenté les d'FPGA de la famille XILINX, et plus particulièrement la famille Vertex 5. Nous avons présenté l'architecture externe de la carte FPGA Vertex5 du cœur ML501. En plus dans ce chapitre noua avons présenté l'architecture interne des circuits programmables FPGA pour pouvoir les configurer. Afin de programmer cette carte nous avons choisi le langage de programmation matériel VHDL pour le présenter ainsi que son environnement de développement ISE

Chapitre III Présentation de l'architecture de commande de notre système de télé-opération bilatéral à un seul degré de liberté

III.1) Introduction

D'une manière générale, un système de télé-opération se décompose en deux parties qui coopèrent : l'une est dite **opérative (esclave)** l'autre est dite **commande (maître).**

Notre architecture de télé-opération qui est disponible au niveau du Laboratoire L2CSP, consiste à commander en force et à distance un vérin en translation bidirectionnelle avec une manette décrivant un mouvement de rotation. Le vérin constitue la partie opérative et l'ensemble (manette et bloc de sensation haptique) constitue la partie commande.

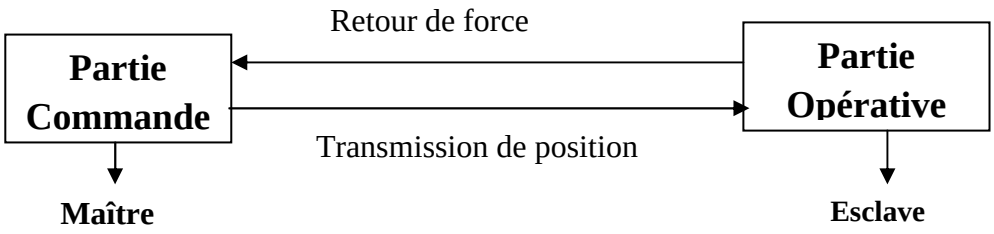


Figure (III.01) : Architecture Télé-Opération

Ce chapitre traite la présentation de cette architecture, ainsi que des éléments qui la constituent. Ensuite nous allons parler du type de commande de cette architecture.

III.2) Présentation du projet de télé-opération avec retour d'effort à 1ddl

Nous avons dit qu'un système de télé-opération consiste à exécuter des tâches à distance en utilisant un système électromécanique (Maitre-esclave). Il se décompose en deux parties : la partie commande qui est présentée à gauche sur la **figure (III.a)** et la partie opérante à droite illustrée sur la **figure (III.b).** La manette est utilisée comme interface haptique pour transmettre l'information de l'interaction de la force mise en jeu entre le vérin, et son environnement à l'utilisateur. Le vérin

à son tour, localisé à distance tente de reproduire les tâches imposées par le poste Maître (l'utilisateur). Il faut également permettre les mouvements libres de la manette (c'est-à-dire que le vérin reproduit librement le mouvement de cette dernière). Un niveau de puissance élevé améliore les performances du système. Mais cela n'est pas indispensable pour retransmettre la plupart des informations haptiques. Par conséquent, pour un meilleur rendu haptique le système doit pouvoir calculer la force musculaire de l'operateur afin qu'il puisse ressentir l'importance de la charge manœuvrée. Ainsi, l'utilisateur aura une idée sur la charge qu'il est en train de manipuler.

Figure (III.a): partie commande

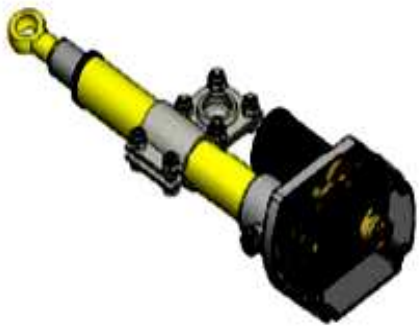


Figure (III.b): partie opérative

Figure (III.02) : Parties du système de télé-opération

Le problème qui se pose c'est que le couplage direct (c'est-à-dire couplage sans interface) ne permet pas à l'utilisateur d'actionner la manette. Parce que pour tourner le bloc de démultiplication il faudrait une force importante, qui ne relève pas des moyen de l'operateur c'est-à-dire que l'opérateur ne peut pas bouger la manette, pour cette raison, on a opté pour un découplage mécanique commandé électriquement afin de libérer la manette au moment de l'actionnement (transmission de la position) et de faire subir le rendu haptique au moment où la charge se présente sur le vérin opérant dans l'environnement (retour de force). Ceci est illustré sur la **figure (III.03)**

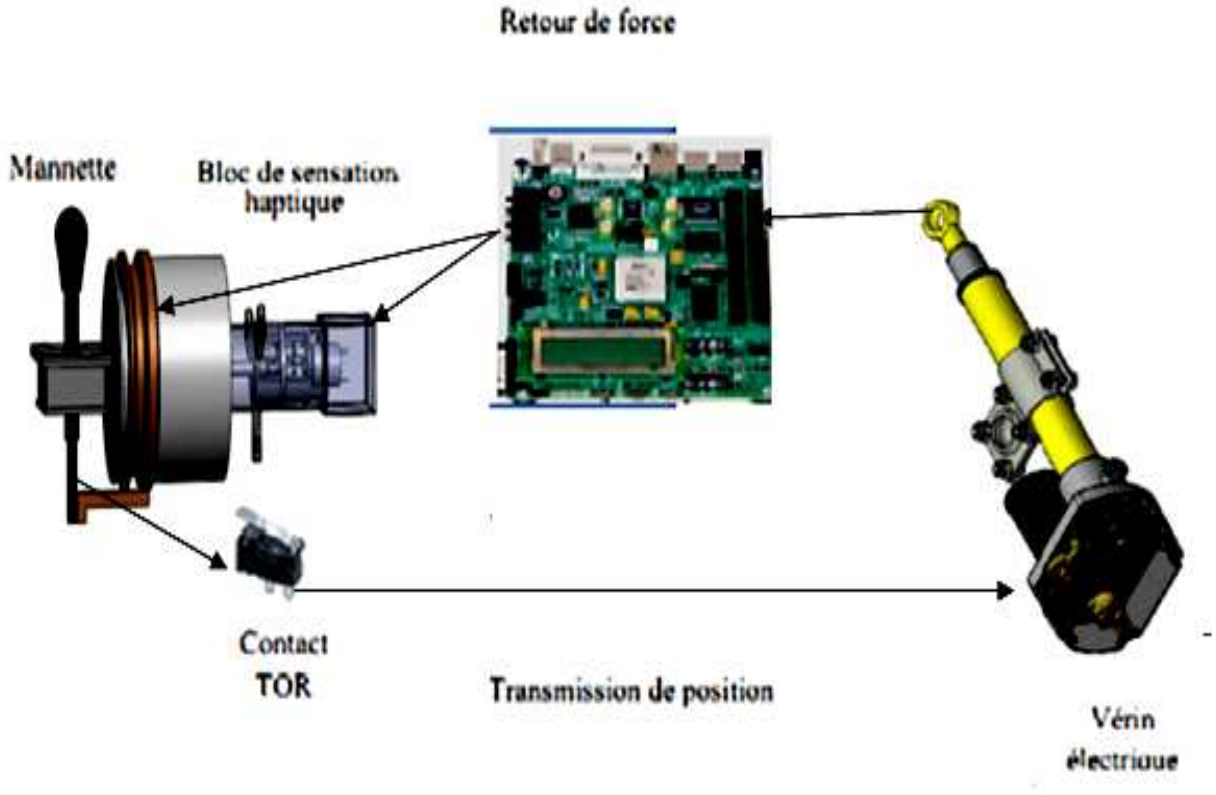


Figure (III.03) : Couplage de la partie commande à la partie opérative

Mais dans la réalisation pratique d'autres éléments sont indispensables pour une mise en marche de cette architecture, tel que un convertisseur analogique numérique, un amplificateur de puissance, un étage de puissance et d'autres composants que nous allons présenter au cours de ce chapitre.

L'objectif principal des systèmes de télé opération, est de chercher les critères permettant d'améliorer la qualité et les performances des manipulations. Le critère principal recherché dans notre cas c'est la **capacité de l'interface** utilisée pour :

1- Retransmettre le plus fidèlement possible l'information de la force présentée au niveau du bras manipulateur.
2- Réduire le retard de transmission des données entre les deux dispositifs maître et esclave.

Pour cela plusieurs interfaces permettent de répondre aux critères demandés, comme les arduino, les microcontrôleurs, la carte FPGA ets …..

Mais les carte FPGA répondent mieux aux critères exigés pas les systèmes de télé-opérations, car les FPGA sont caractérisées par ses parallélises de solutions qui permettent de

rendre ces systèmes transparents ainsi que ses grandes capacités de calcules qui permet de stabiliser ces systèmes en éliminant le temps de retard dans les canaux de communications.

III.3) Description expérimentale du système bilatéral

Comme nous l'avons mentionné dans la partie introduction, le système de télé-opération est décomposé de deux parties reliées par un canal de communication. Chaque partie est constituée des éléments nécessaires pour faire fonctionner plus fidèlement le système. Il est possible de définir au mieux une structure générale d'une architecture de commande d'un système de télé-opération comme le montre la **figure (III.04). La figure (III.05)** présente le dispositif expérimental à base de la carte FPGA. La présence de retard dans un système de télé-opération entraine son instabilité. Nous avons utilisé cette carte, dans le but d'améliorer la stabilité ainsi que les performances des systèmes de télé opération et de réduire le temps de retard entre les dispositifs maître et esclave.

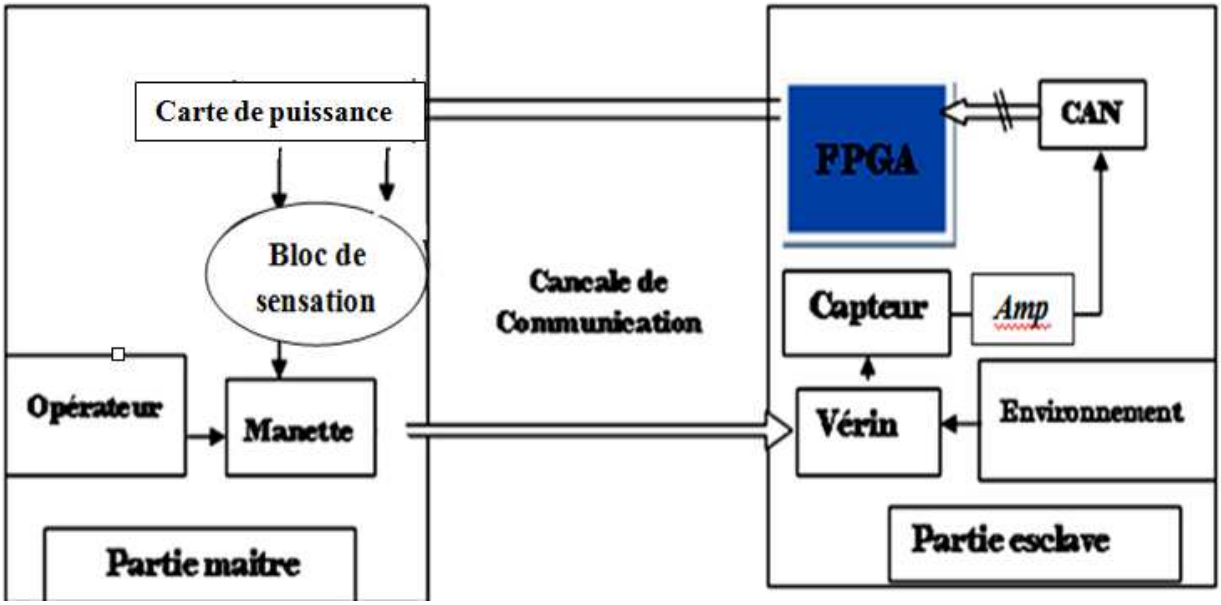


Figure (III.04) : Architecture synoptique du système de télé-opération bilatéral

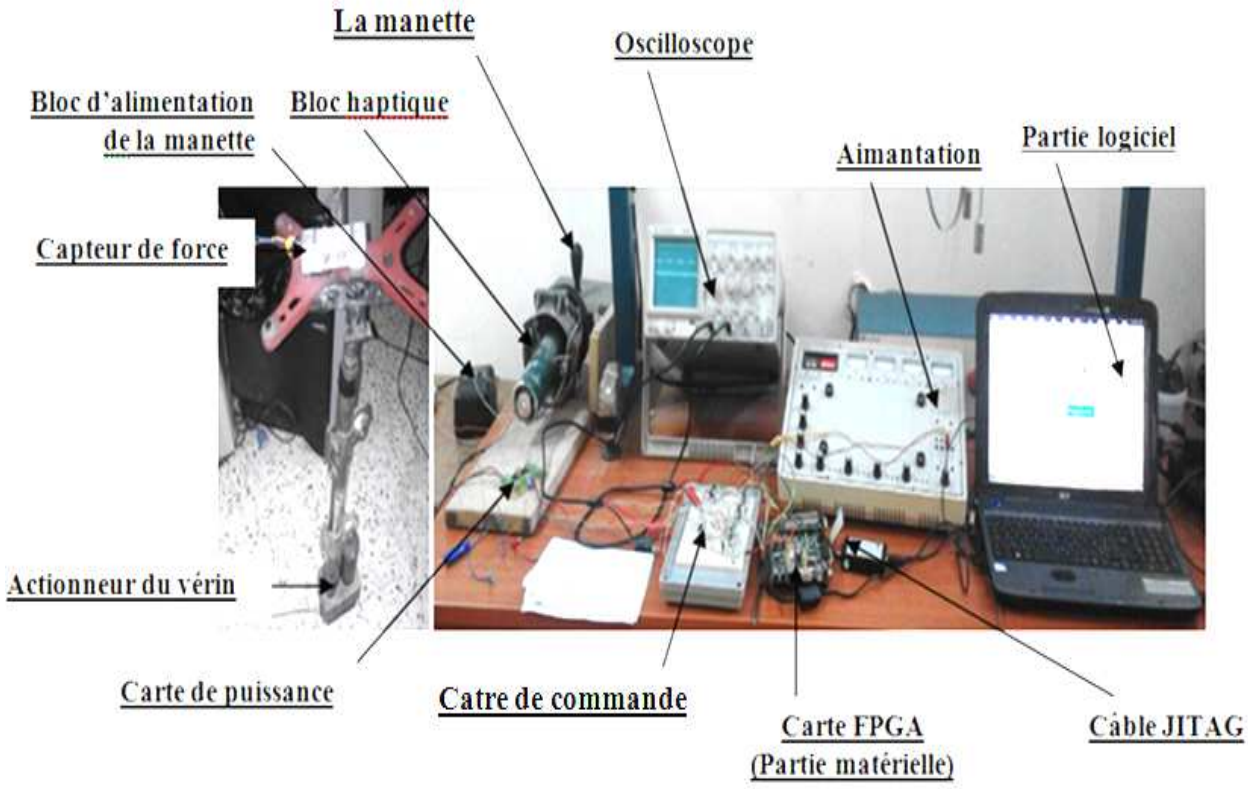
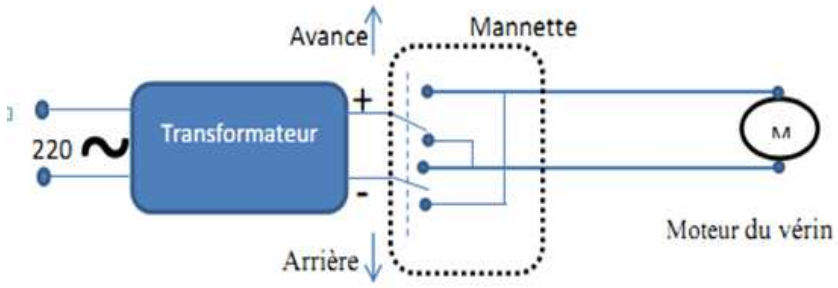


Figure (III.05) : Dispositif expérimental de l'architecture utilisée.

La structure générale d'une architecture de commande d'un système de télé-opération réalisée dans le laboratoire L2CSP est essentiellement constituée de deux parties qui sont reliées par un câble de communication. La partie maitre (partie commande) et Partie esclave (partie opérative). Chaque partie contient des éléments primordiaux au fonctionnement du système**. L'opérateur** qui représente le cerveau du système, il s'agit de l'être humain qui donne des ordres relatifs au cahier des charges à partir du dispositif maitre (manette). Son objectif dans ce projet est de faire bouger la manette en avant ou en arrière et de ressentir le rendu haptique (le retour de la force). **La manette** à son tour joue le rôle de l'interface entre l'opérateur et le bloc de sensation. Elle est équipée de deux commutateurs Tout-où-Rien qui sont reliés directement au vérin. La commande en position de celui-ci (vérin) dépendra de la précision et du doigté de l'utilisateur. Pour asservir un vérin en position, l'opérateur devra estimer la position de sa tige. Ainsi, pour pouvoir commander ce vérin, on a couplé l'actionneur équipant le vérin à un réducteur de vitesse. Ce dernier fait tourner une vis sans fin, afin d'avancer la tige de façon très lente, et de manière à ce que l'operateur puisse manipuler la sortie et l'entrée de la tige avec précision. Le

schéma de puissance d'alimentation qui permet de commander le vérin dans les deux sens est illustré dans la **Figure(III.06**).



Figure(III.06) : Schéma de puissance d'alimentation de l'actionneur du vérin

D'après cette figure, si la manette est en position avancée, le vérin est commandé en sortie. Le deuxième sens indique que la manette est en position Arrière, l'actionneur du vérin tourne dans un autre sens et le vérin est commandé en entrée.

Pour un retour d'effort permettant de ressentir des forces mises en jeu au cours de l'opération, on a équipé la tige du vérin d'un **capteur de force** (dont le principe de fonctionnement est donné en annexe A). Cet élément est de type jauge de contrainte. Il est utilisé en vue de renvoyer la déformation que le matériau subit par le biais de l'élongation des résistances montées dans un pont de Wheatstone détaillé en Annexe A . Cette déformation est renvoyée sous forme de tension analogique donnée en relation (III.01).

$$V_{out} = \frac{V_{in}}{2} k\varepsilon \qquad \text{(III. 01)}$$

- V_{out} : La tension an Analogique
- V_{in} : tension d'aimantation du pont
- k : facteur de jauge de contrainte
- ε : la déformation que le matériau subi

D'après la relation (III.01) on constate que la tension délivré par le pont est proportionnelle à la déformation. Par conséquent plus les efforts subis par la tige sont importants, plus la tension mesurée au pont est grande.

L'amplification de cette tension est nécessaire pour qu'elle soit ressentie par la carte que nous avons utilisée pour le traitement des informations. Pour cela nous avons utilisé l'INA126

(Annexe B). Cet amplificateur est sensible, précis et fidèle aux faibles bruits. Il est constitué de deux amplificateurs opérationnels et offre une excellente performance avec un très faible courant de repos (175 A/canal). La tension amplifiée sera envoyée au Convertisseur Analogique Numérique (CAN0804) caractérisé par une résolution de 8 bits (le principe de fonctionnement de ce convertisseur est présenté en **Annexe** D**)** avant qu'elle soit envoyée à la **carte FPGA**, car celle-ci ne dispose pas d'entrées analogique**s.** Cette dernière représente l'élément essentiel de notre travail. Elle s'occupe du traitement électronique des opérations en utilisant plusieurs éléments que nous avons détaillés dans le chapitre II. Par ailleurs, cette plateforme est dotée de plusieurs types de périphériques qui peuvent être utilisés selon les besoins du concepteur et la nature de l'application à développer. **La figure (III.07)** présente les différents constituants de la carte ML501 utilisé dans notre travail.

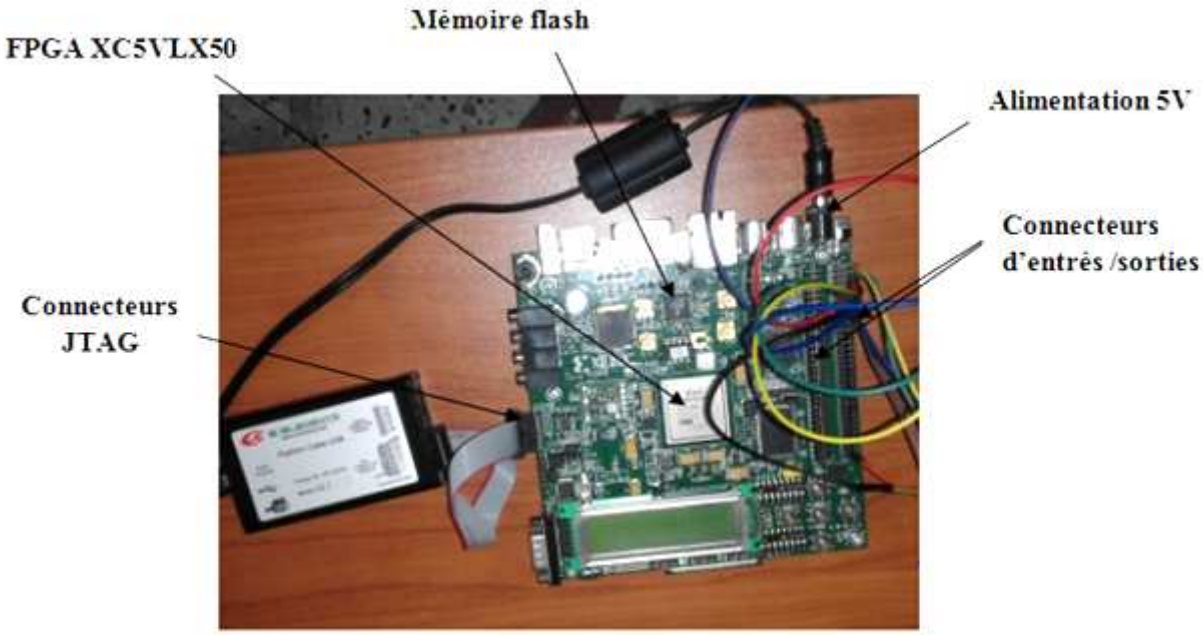


Figure (III.07) : Carte FPGA Vertex XC5VLX50.

- **Alimentation 5V** : Ce connecteur assure la mise sous tension de la carte FPGA

- **Connecteurs d'entrés/sortis** : le rôle de ces connecteurs c'est de relier l'extérieur vers intérieure de la carte.

- **FPGA XC5VLX50** : Ce circuit représente le cœur de la carte FPGA. Il permet l'analyse et le traitement des données

- **Mémoire flash** : ce composant intégré assure la sauvegarde momentanée du programme implanté. en mode de programmation il est nécessaire de reprogrammer la carte à chaque remise sous tension car le programme du FPGA est recopié dans la mémoire volatile**.**

- **Connecteur JTAG :** Assure la transformation du programme qui transféré en **bitsream** du partie logiciel vers la partie matériel (le support physique FPGA)

Le signal sortant du capteur est reçu par la carte FPGA. Si celui-ci correspond à une valeur d'effort supérieur à la force générée par les frottements et l'inertie du système, la carte envoie deux signaux simultanément pour commander le **bloc de sensation**. Le premier signal est utilisé pour actionner **l'électro-aimant** afin d'avoir le couplage entre le bloc du rendu haptique et la manette, l'autre signal est envoyé vers le **moteur à courant continu** sous forme de signal MLI (modulation de largeur d'impulsion). Ce signal produit le retour d'effort en vue de pouvoir commander l'amplitude de la force ressentie. Pour contrôler ce type de moteur, il existe essentiellement deux méthodes. La première consiste à utiliser un amplificateur de puissance avec une tension proportionnelle à la vitesse recherchée. Cette technique montre, à l'évidence, une perte considérable de puissance dans l'amplificateur, ce qui dégrade les performances d'une stratégie de commande appliquée au contrôle d'un système, car l'énergie est limitée. La meilleure solution consiste à contrôler le moteur avec un rapport cyclique adapté à la vitesse. Cette technique s'adapte mieux au contrôle des moteurs de qualité moyenne que l'on trouve plus communément.

Par conséquent, pour commander le bloc de sensation haptique on utilise systématiquement un hacheur à transistor. La commande du transistor est un signal carré à rapport cyclique variable et à période fixe. En effet, la période est adaptée à la constante de temps du moteur et le rapport cyclique permet de faire varier la vitesse du moteur donc du bloc de sensation.

Dans notre projet, cette commande sera générée par la carte FPGA en fonction de la tension produite par le pont de Wheatstone (amplificateur).

En revanche on ne peut pas faire tourner le moteur avec le signal de modulation de largeur d'impulsion MIL (PWM en anglais) provenant de la carte FPGA, car ce dernier est un signal logique qui varie entre 0 et 5 Volt est faible en puissance. Or ce moteur nécessite une tension de

12V pour juste l'exciter. Pour pallier à problème, nous avons réalisé un dispositif de commande de puissance équipé des transistors IRF640 telle que représentée sur la **Figure (III.08)**

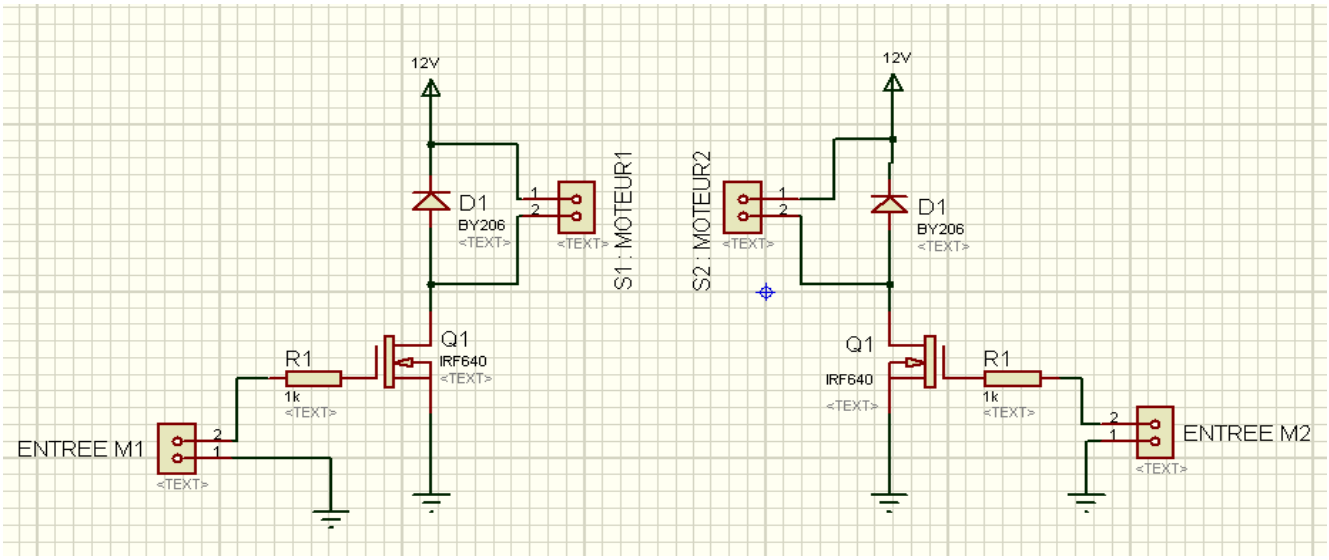


Figure (III.08) : Dispositif de commande de puissance.

Dans notre réalisation, nous avons utilisé d'autres composants tels qu'un **régulateur 7805 (annexe D). C**'est un régulateur de tension positive, qui constitue un idéal pour stabiliser une tension à une valeur fixe. Ceci convient pour tout montage électronique nécessitant une tension fixe.

Par ailleurs, on a utilisé un photo-coupleur 4N35 dans le but d'isoler la partie puissance et la partie commande.

III.4) Schéma de commande en force d'une architecture de télé-opération

Le but de ce projet est d'asservir une architecture de télé-opération qui simulera un poids, c'est à dire à retour de force. Le circuit de commande de puissance du bloc de sentation est conçu et interconnecté avec le circuit FPGA. Ce dernier envoie deux signaux. Le premier signal est le signal de modulation de largeur d'impulsion MLI (PWM) qui sera envoyé au premier transistor implanté sur la carte de puissance, celui-ci fera varier la vitesse du moteur à courant continu et produit le retour de force. Le deuxième signal est continu et sera envoyé au deuxième transistor. Ce transistor est implanté sur la même carte (un dispositif de commande de puissance) pour alimenter

l'électro-aimant. Cela produit le couplage entre le bloc du rendu haptique et la manette pour générer le blocage de la manette qui permet à l'opérateur de ressentir la force entrainée par le vérin (esclave) dans un l'environnement.

Le signal analogique provenant du capteur doit être converti en un signal numérique après avoir l'amplifié pour qu'il soit lu et ressenti par la carte FPGA. Pour ce qui est des jauges de contrainte, un amplificateur différentiel est implémenté de façon à lire la déformation présentée sur les plaques du capteur. Cette déformation est facilement convertie en tension en utilisant le pont de Wheatstone. **La figure (III.09)** Présente le schéma bloc de la commande en force de l'architecture de télé-opération en question.

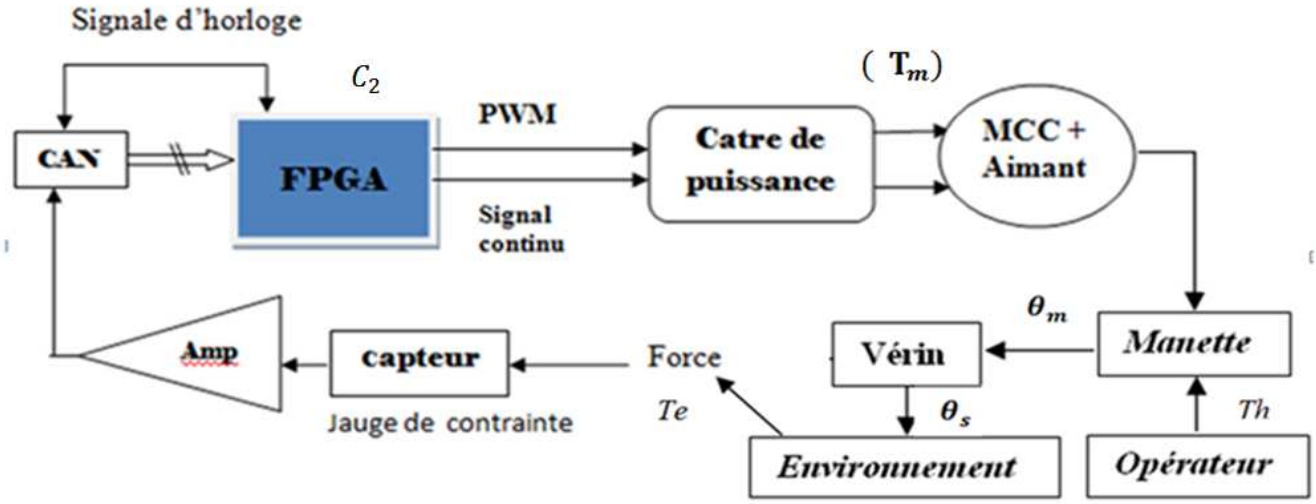


Figure (III. 09): Schéma-bloc de la commande en force d'une architecture de télé-opération.

D'après cette figure et en référence au **chapitre I**, en utilisant l'architecture de contrôle Force-position, La position du maître $\boldsymbol{\theta_m}$ est envoyée comme consigne vers l'esclave et l'interaction entre l'esclave et son environnement $\boldsymbol{T_e}$ est mesurée par le capteur de force, et est renvoyée comme consigne vers le maître, à travers le canal de communication $\boldsymbol{C_2}$ **(FPGA)**, pour générer le retour d'effort $\mathbf{T_m}$ La commande du moteur est donc définie par : $T_m = C_2\, T_e$ avec $C_2 = \beta e^{-s\Delta 2}$.

$\boldsymbol{C_2}$: C'est le bloc qui permet de transférer la force entrainée par l'esclave vers le maître.

Comme la capacité de notre interface utilisée nous offre des meilleurs performances grâce à sa la grande capacité de calcule qui permet d'annuler le temps de retard dans les canaux de communications, ce qui donne $e^{-s\Delta 2} = e^{-s*0} = 1$, ainsi le parallélisme de ses solutions qui permet de retransmettre le plus fidèlement possible l'information de la force, ce qui

donne l'estimation approxime de $\beta \; vers \; 1$. Ce ci permet de produire la transparence du système. Alors on déduit que les commandes des deux sites sont comme suit : $T_e = \mathbf{T_m} = -T_h$ et $\theta_s = \theta_m$

- $\boldsymbol{\theta_m}$ **:** Représente la position du maître **(**Signal sortant de la manette et transmis directement à l'actionneur du vérin.
- $\boldsymbol{\theta_s}$ **:** Représente la position de l'esclave (la position du vérin).
- $\mathbf{T_e}$ **:** Représente le signale de la force de l'interaction entre l'esclave et son environnement
- $\mathbf{T_m}$ **:** Représente le signal **de modulation de largeur d'impulsion MLI** que l'on décrira dans les sections qui vont suivre en utilisant le langage VHDL.

La technique de modulation de largeur d'impulsion est la plus utilisée pour la commande des architecteurs de télé-opération. Le principe de cette technique est donné dans la section suivante.

III.5) Technique de Modulation à Largeur d'Impulsion MLI

C'est la commande qui permet d'alimenter le moteur avec une tension en créneaux c'est-à-dire en largeur d'impulsion. Il existe plusieurs techniques pour obtenir un signal de modulation en largeur d'impulsion.

Dans notre travail, la technique utilisée, consiste à comparer deux signaux :

1- le premier c'est le **signal de mesure** (signal provient du capteur)

2- le deuxième c'est le **signal triangulaire** (signal en dents en scie)

On a utilisé le signale en dents de scie pour fixer la fréquence d'exécution de l'opération.

Le résultat est un signal de type Modulation de Largeur d'Impulsion MLI dont la période est égale à la période du signal triangulaire et un rapport cyclique. Autrement dit c'est la durée pendant laquelle le signal de commande est supérieur au signal triangulaire.

Donc, le principe utilisé pour réaliser un rapport cyclique variable est de comparer un signal triangulaire avec un autre signal de tension continu variable. La figure suivante montre cette technique.

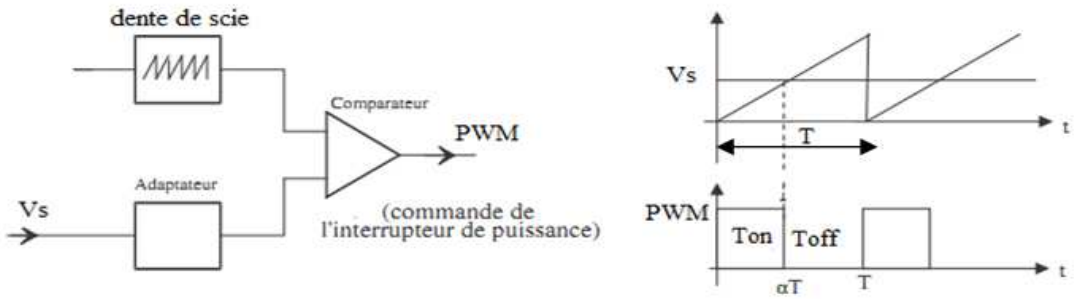


Figure (III.10) : Technique de Modulation à Largeur d'Impulsion MLI.

Avec :

- Vs : est la tension de référence continue variable ou bien tension de mesure
- PWM : La tension carrée désirée pour la commande.
- T : la periode du signal en dents de scie
- T_{on} :le temps où le signal Vs supérieur au signal en dents de scie
- Toff : le temps où le signal Vs inférieur au signal en dents de scie
- α : c'est le rapport cyclique $\alpha = \frac{ton}{T}$

Le calculateur utilise la technique de modulation à largeur d'impulsion (MLI) citée précédemment pour générer un signal carré de fréquence de 1KHZ, et d'amplitude de 5V et d'un rapport cyclique $\alpha = \frac{U(z)}{5} \quad \Longrightarrow \quad donc: \quad U(z) = 5 * \alpha$

L'avantage principal de la commande de modulation à largeur d'impulsion (PWM en anglais) est de réduire les pertes de puissance dans les dispositifs de commutation. Beaucoup de circuits numériques peuvent produire des signaux de PWM, tels que les microcontrôleurs, les Arduino, les PICs et les FPGA.

Dans notre cas, le circuit numérique utilisé pour générer cette technique est la carte FPGA. Pour cela, un signal numérique en dents de scie et un autre signal numérique de conduite de données sont exigés. Un comparateur examine les deux valeurs (signaux numériques) afin de produire un signal de modulation à largeur d'impulsion.

Le langage de description de matériel (VHDL) est employé pour produire ces signaux exigés par cette carte FPGA.

III.6) Conclusion

Dans ce chapitre, nous avons présenté l'architecture d'un système de télé-opération à un degré de liberté réalisé au laboratoire L2SCP qui est utilisé dans le projet ainsi que les éléments qui le constituent. Nous avons ensuite choisi une loi de commande qui correspond le mieux à l'architecture force-position adoptée. Il s'agit de la commande MLI. Nous avons choisi cette technique de commande, car elle est considérée comme une des techniques les plus utilisée pour la commande en force des architectures de télé-opération, et pour sa simplicité de programmation et d'implantation sur des supports numériques tels que les FPGA. Le chapitre suivant nous présentera l'implantation de cette technique de commande sur un support physique programmable FPGA.

Chapitre IV Résultas de simulation et expérimentaux de la commande en force de l'architecture de télé-opération à base de la carte FPGA

IV.1) Introduction

Ce chapitre présente les deux parties simulation et expérimentale du projet, qui consiste en la réalisation d'un circuit MLI adapté à la commande d'une architecture de télé-opération. Cette technique permet la variation de la vitesse d'un moteur à courant continu constituant l'élément principal du bloc de sensation haptique de notre architecture de télé-opération.

Pour faire varier la vitesse d'un moteur à courant continu en utilisant cette technique, il suffit de faire varier le rapport cyclique. Ce rapport est fonction de la tension d'alimentation. Dans ce projet, cette technique de commande est produite en utilisant la carte FPGA. La variation du rapport cyclique est due à la variation de l'entrée de cette carte qui dépend du retour de force comme le montre la **figure (III.09)**, dans le chapitre III. Alors que le chapitre présent fixe comme objectif la conception du circuit de commande, qui admet pour entrée le retour de force provenant du site esclave et pour sotie les signaux qui permettent d'activer le bloc de sensation haptique (site maître). Pour ce faire, nous avons utilisé le langage de programmation VHDL et la suite de développement ISE 14.2, en suivant le flot de conception d'un circuit FPGA donné en **Figure (II.07)**

IV.2) Stratégie et implémentation de la commande dans un support FPGA en utilisant VHDL

Dans ce qui suit nous allons présenter les étapes de conception d'un circuit de commande. Nous allons faire la conception du circuit de commande à l'aide de l'approche de conception FPGA par schéma. Une foie les toutes étapes de conception sont vérifées nous allons prosséder à l'implantation de ce circuit de commande dans un support physique FPGA **[37]**

IV.2.1) Conception du circuit de commande

Pour concevoir le circuit de commande MIL (PWM), il faut démarrer par le principe donné en section V dans le chapitre III. D'après ce principe, pour concevoir un circuit MLI il faut un comparateur, un bloc générateur du signal en dents de scie et un autre signal constant et qui varie en amplitude. Pour cela nous décrivons le code VHDL texte qui remplis le cahier des charges du comparateur et le bloc générateur du signale en dents de scie ainsi que le diviseur d'horloge. Puis nous allons créer leurs blocs schématiques en utilisant « Create Schematic Symbol » dans l'environnement ISE. Une fois tout est fini nous allons interconnecter ces bloc entre eux pour générer les signaux de commande demandés. La **Figure (IV.01)** nous montre la conception du système (schéma de commande) qui produit les signaux de commande.

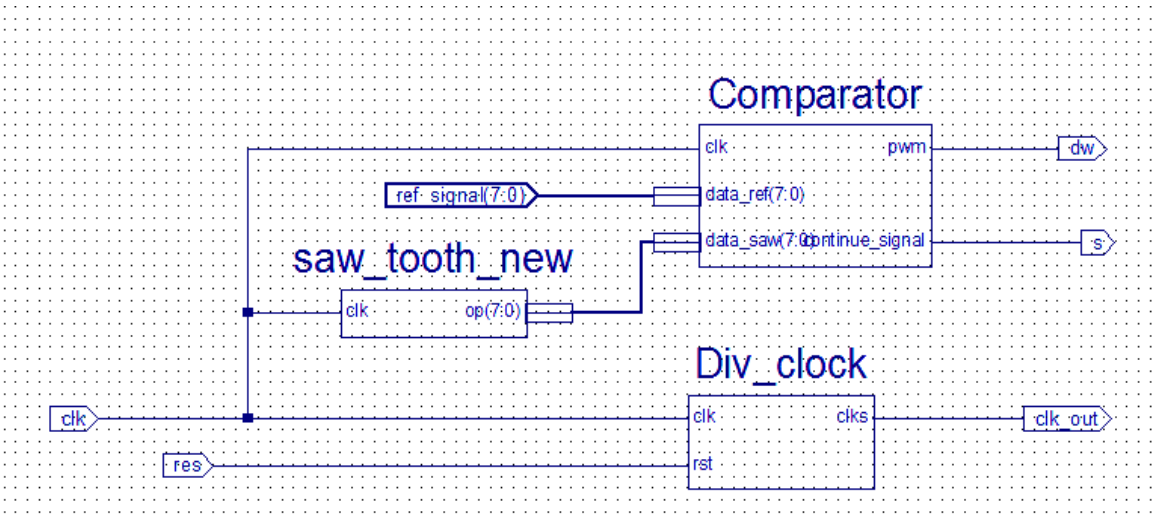


Figure (IV.01) : Schéma de conception du circuit de commande.

D'après cette figure on constate que le comparateur reçoit en entrée le signal d'horloge (clk), le signale en dents de scie (saw_too+th_new) ainsi que le signale (ref_signal). Le dernier signale représente le signal de retour en force de notre architecture qui varie en fonction de la force de l'environnement. Cette force convertie en une tension en utilisant le pont de Wheatstone implanté dans le capteur, puis convertie en un autre signal numérique pour qu'il soit lu par la carte FPGA.
Pour cela, il nous faut un dispositif qui puisse comparer deux valeurs. La première valeur représente le signal en dents de scie et la deuxième représente les données saisies par le capteur de force sur la Carte FPGA. Ce même comparateur délivre en sortie les signaux de commande qui permettent d'actionner le bloc la sensation haptique. Ces signaux sont : le signal continu (S) qui excite

l’électroaimant et le signal de modulation à largeur d’impulsion PWM (dw), qui mit en marche le moteur CC. Dans la section suivant nous allons détailler le contenu des blocs illustrés dans le schéma de conception donné en **figure (IV.01)**

IV.2.1.1) Bloc diviseur d’horloge (Div_clock)

Les propriétés de l'horloge dépendent de la fréquence du FPGA. Les diviseurs d’horloge fournissent des possibilités de synchronisation avancées à l'application de FPGA.

Dans cette conception, le signal clk est utilisé pour produire le clk_out. Cette valeur est employée comme signal de clock pour activer le convertisseur analogique numérique (CAN) de 8 bits.

IV.2.1.2) Bloc de la dent de scie (Saw_tooth_new)

Ce compteur est employé pour produire un signal en dents de scie en utilisant 8bits d’entrée (bus d’entrée de capacité 8). Le code de ce compteur est écrit en utilisant le langage VHDL. Il compte de $(00000000)_2$ à $(11111111)_2$. La valeur maximum est $(255)_{10}$, lorsque ce compteur arrive à cette valeur, il revient à zéro. Cela avec une fréquence peut aller jusqu’à 400 Khz (la valeur maximal) qui représente la fréquence du signal de largeur d’impulsion PWM,

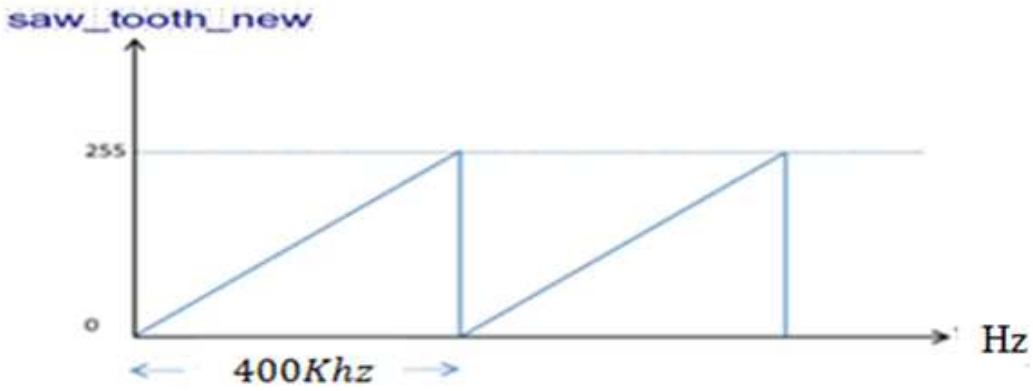


Figure (IV.02) : Le signal de dent de scie

IV.2.1.3) Comparateur (comparator)

Ce comparateur a trois entrées et deux sorties :

- **Première entrée**, contient les données A venant du bloc qui génère le signal en dents de scie.
- **Deuxième entrée**, les données B venant du signal de référence (mesure), C'est le signal délivré par le capteur ce qu'on appelle le retour de force.
- **Troisième entrée**, c'est le signal d'horloge (clk).

Le comparateur analyse les données A et B.

1- Si les données A sont inferieures aux des données B, la sortie sera 1.
2- Si les données A sont supérieures aux données B, la sortie sera 0.

Les sorties du comparateur sont : Le signal PWM et S (continue signal).
Ces signaux sont employés pour commander le moteur à courant continu, et l'éctro-aimant.

IV.2.2) Synthèse du circuit de commande

Avant de générer le schéma du circuit global de la commande de largeur de modulation MLI, on a créé tous les blocs associés séparément en utilisant un code VHDL, à l'aide du synthétiseur du logiciel ISE design suit 14.2de XILINX présenté dans le chapitre II. Cependant, on a pu simuler, synthétiser et créer le symbole schématique du circuit de commande.

- **Résultat de la synthèse**

Pour obtenir les résultats de la synthèse on clic sur **« Synthesize –XST »** qui permet de faire la synthèse du circuit en blocs configurables se trouvant sur le FPGA. La description HDL ce circuit est transformée en un assemblage (**Netlist**) de primitives de base (portes logiques, bascules …). Après la synthèse, on peut visualiser le **niveau RTL** comme suit

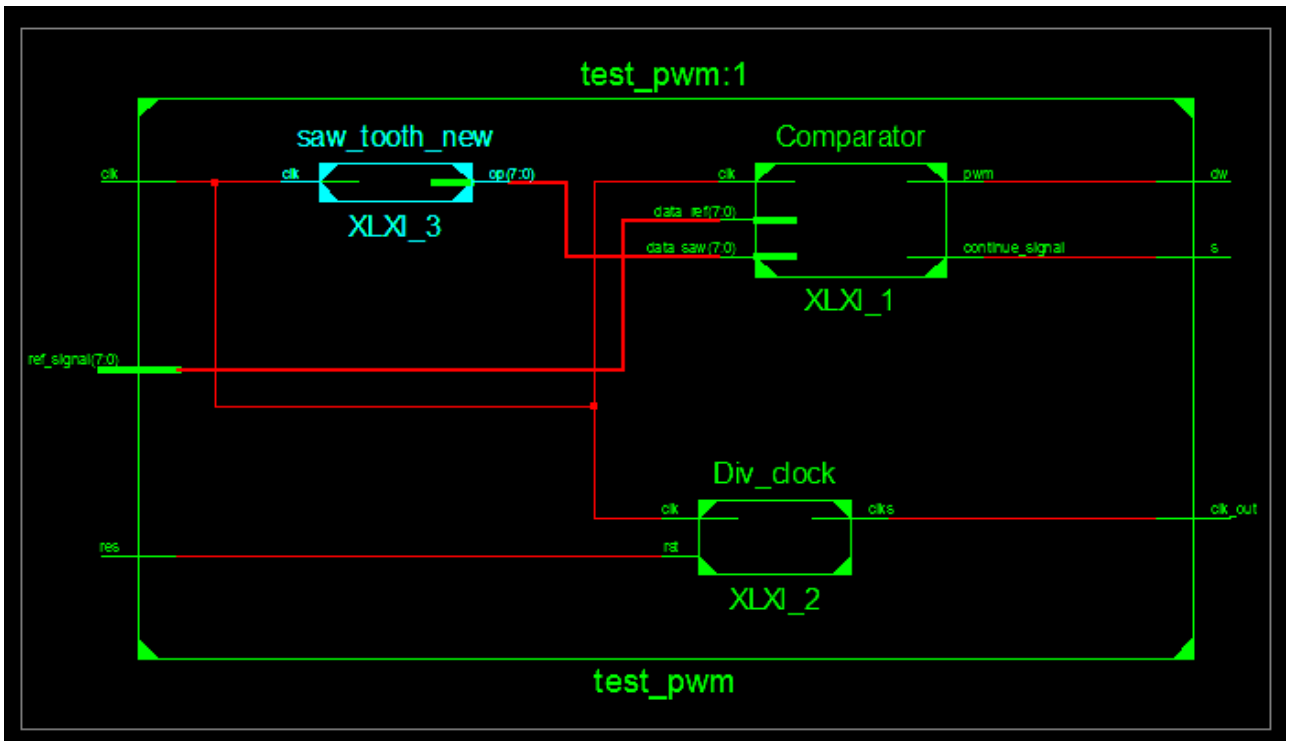


Figure (IV.03) : Schéma RTL détaillé de la commande.

IV.2.3) Simulation fonctionnelle avec Isim

Une fois l'étape de conception est terminée, nous allons effectuer l'étape de simulation fonctionnelle du projet à l'aide d'un testbench et du simulateur Isim. Pour cela, on a généré un fichier testbench VHDL à l'aide du navigateur ISE. Une fois appelé par ISE, Isim prendra en charge la simulation et la présentation des résultats.

IV.2.3.1) Résultats de la simulation

La valeur délivrée par le capteur est comprise entre zéro et cinq (5) volts, et comme nous allons utiliser un convertisseur de 8 bits, alors la valeur numérique convertie en décimal est comprise entre 0 et $2^8 - 1 = 255$ et en binaire soit entre (0000000) et (11111111). Donc pour obtenir l'équivalence en numérique de cette valeur analogique (tension), il faut faire cette conversion :

Pour une valeur maximum en analogique, correspond une valeur maximum en digital (numérique).

$$V_D = \frac{V_A * 255}{5}$$

Avec :

V_D : est la valeur digitale (la valeur numérique)

V_A : est la valeur analogique (la mesure du capteur)

Une fois cette valeur est convertie, elle sera envoyée à la carte FPGA pour le traitement. Pour trouver les résultats de simulation, on utilise le simulateur Isim à l'aide d'un tesbench, en fixant le vecteur de mesure pour chaque valeur allant de 00000000 à 11111111 (0 à 5 volts), puis on donne les résultats de simulations qui sont illustrés dans les figures suivantes.

- **<u>Premier cas</u>** : La valeur délivrée par le capteur est égale à zéro volt c'est à dire que le vecteur d'entrée V_D mesure [7 :0] =0000000

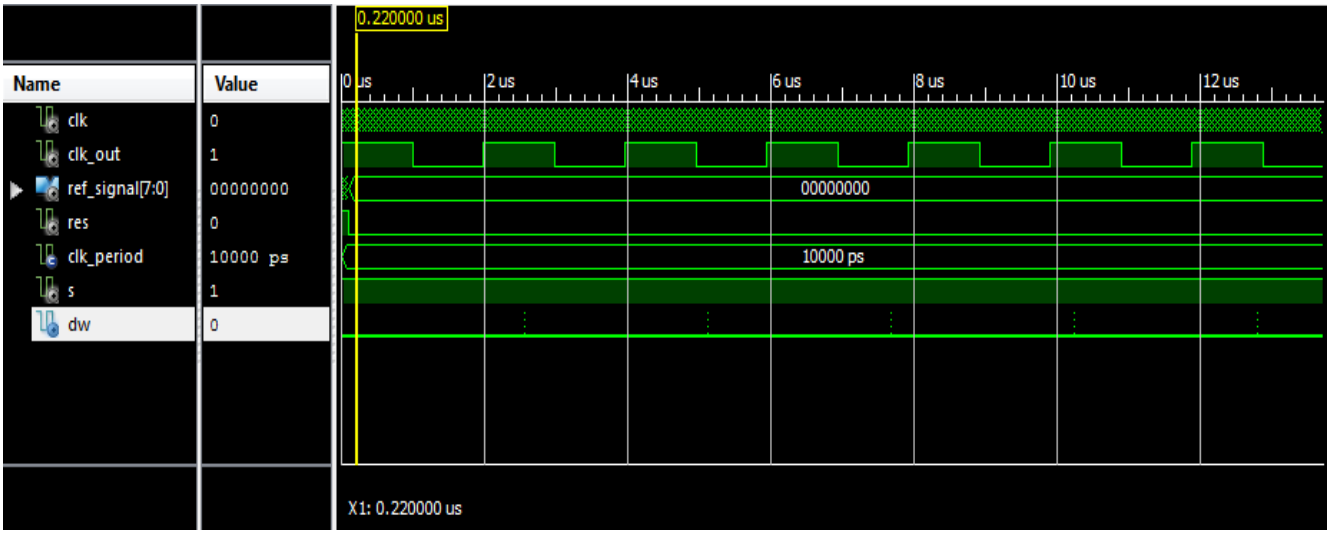


Figure (IV.04.a) : Schéma de simulation pour la tension d'entrée V=0 V.

$$\alpha = \frac{ton}{T} = \frac{0}{2.56} = 0 \rightarrow \alpha = 0\%$$

- **<u>Deuxième cas</u> :** La valeur délivrée par le capteur est égale à un volt, c'est à dire que le vecteur d'entrée V_D mesure [7 :0] =00110011

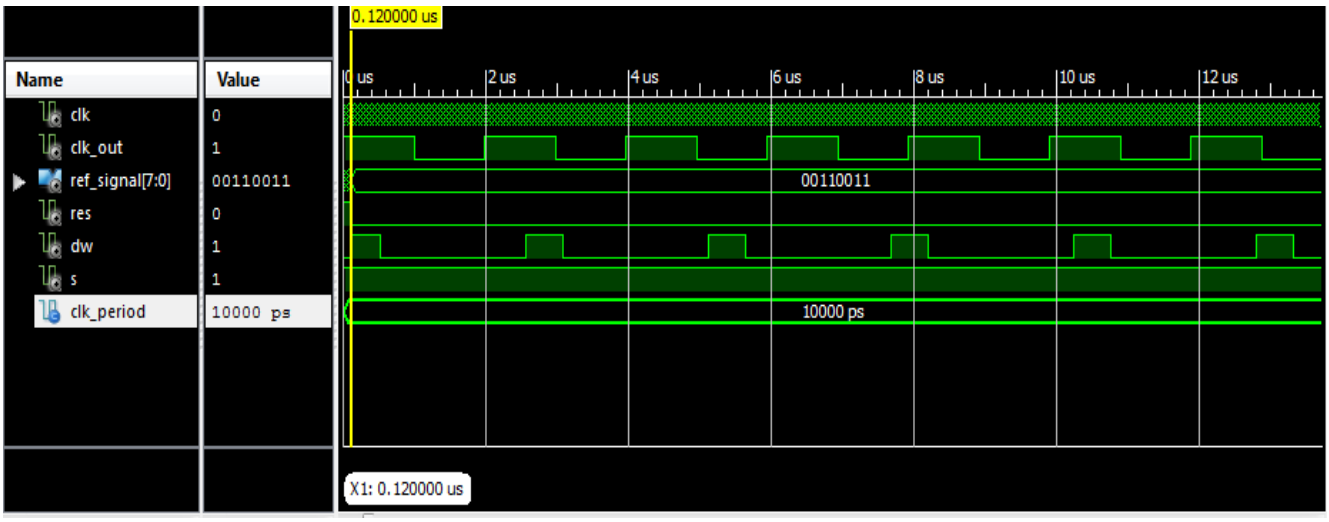


Figure (IV.04.b) : schéma de simulation pour la tension d'entrée V=1v

$$\alpha = \frac{ton}{T} = \frac{0.51}{2.56} = 0.2 \rightarrow \alpha = 20\%$$

- **<u>Troisième cas :</u>** La valeur délivrée par le capteur est égale à deux volts, c'est à dire le vecteur d'entrée V_D mesure [7 :0] =01100110.

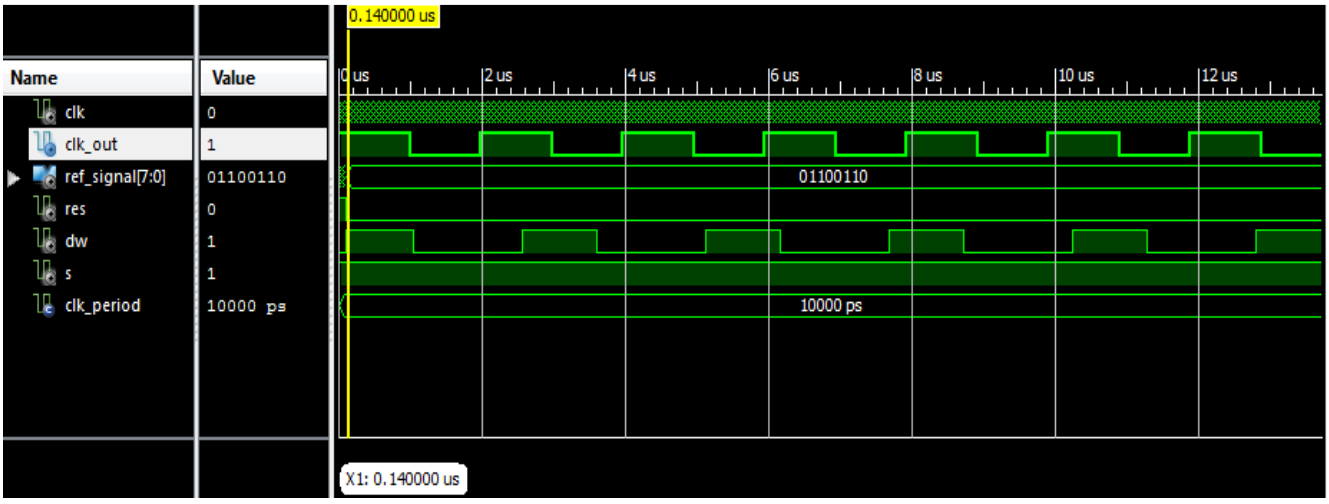


Figure (IV.04.c) : Schéma de simulation pour la tension d'entrée V=2 V.

$$\alpha = \frac{ton}{T} = \frac{1.03}{2.56} = 0.4 \rightarrow \alpha = 40\%$$

- **<u>Quatrième cas :</u>** La valeur délivrée par le capteur est égale à trois volts c'est-à-dire que le vecteur d'entrée V_D mesure [7 :0] =10011001

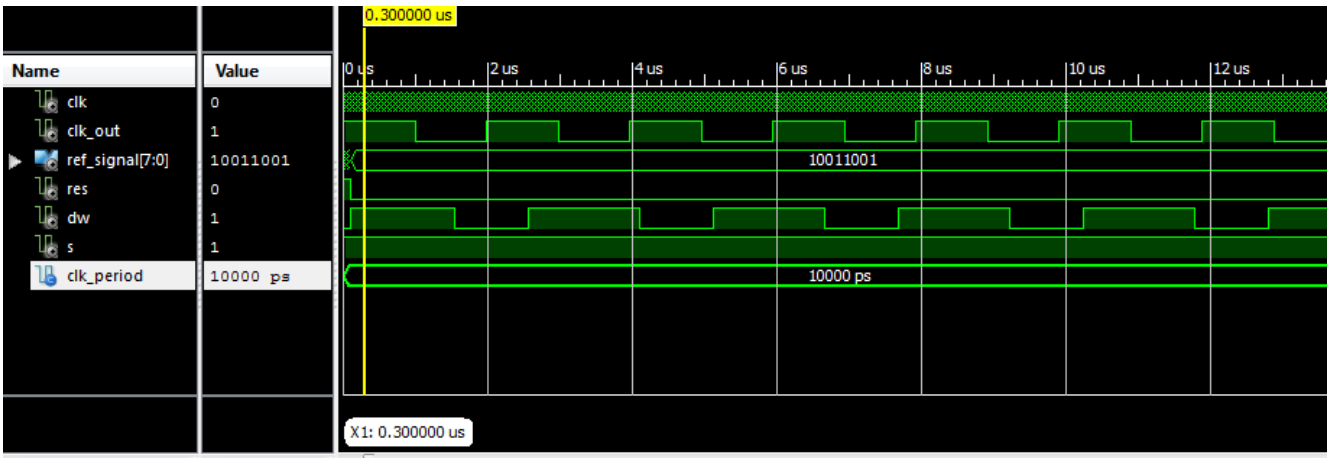


Figure (IV.04.d) : Schéma de simulation pour la tension d'entrée V=3 V.

$$\alpha = \frac{ton}{T} = \frac{1.54}{2.56} = 0.6 \rightarrow \alpha = 60\%$$

- **<u>Cinquième cas :</u>** La valeur délivrée par le capteur est égale à quatre volts c'est à dire que le vecteur d'entrée V_D mesure [7 :0] =11001100

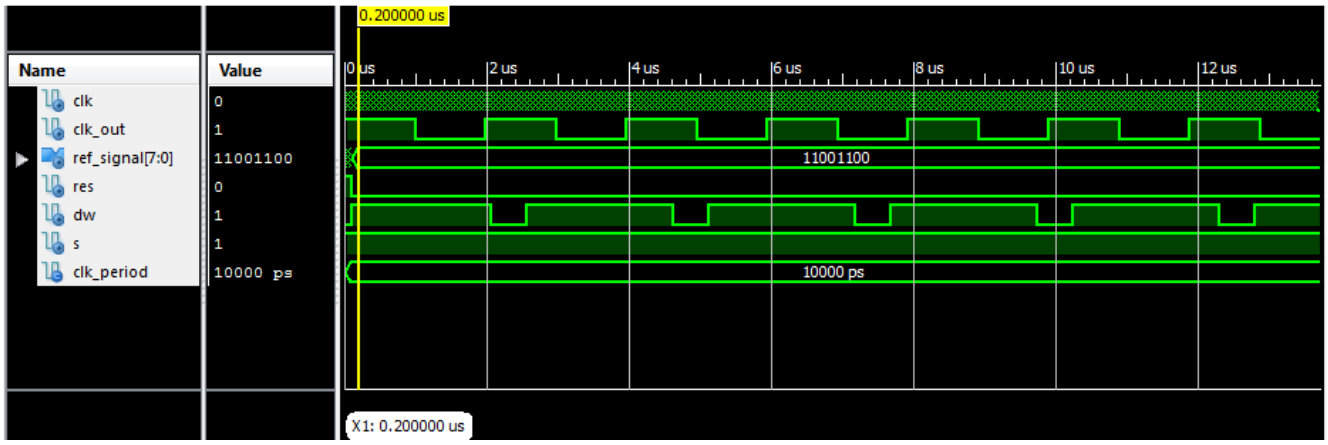


Figure (IV.04.e) : Schéma de simulation pour la tension d'entée V=4 V.

$$\alpha = \frac{ton}{T} = \frac{1.54}{2.56} = 0.8 \rightarrow \alpha = 80\%$$

- **<u>Sixième et dernier cas :</u>** La valeur délivrée par le capteur est égale à cinq volts c'est à dire que le vecteur d'entrée V_D mesure [7 :0] =1111111

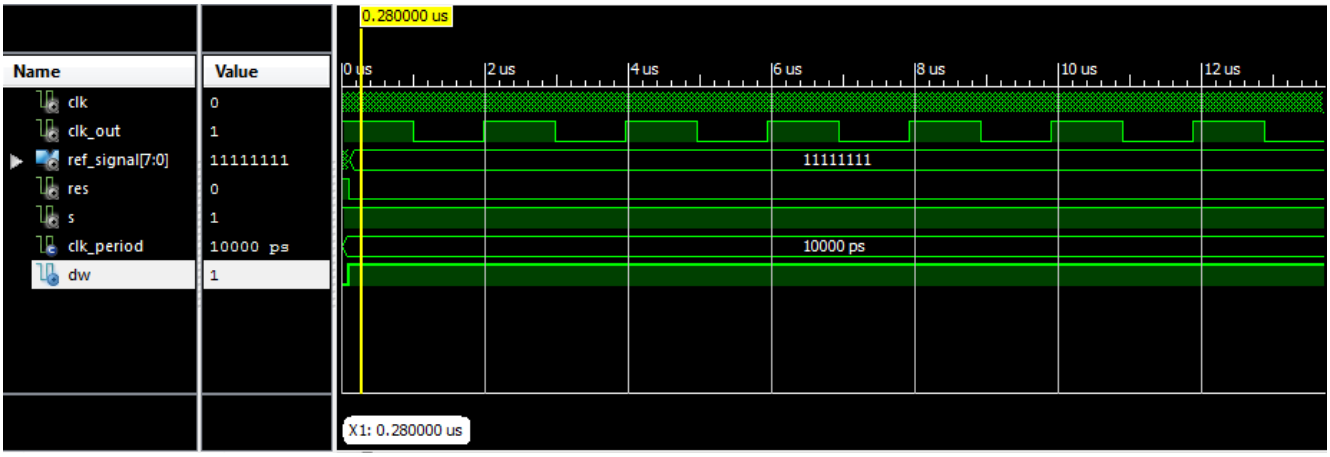


Figure (IV.04.f) : Schéma de simulation pour la tension d'entrée V=5 V.

$$\alpha = \frac{ton}{T} = \frac{2.56}{2.56} = 1 \rightarrow \alpha = 100\%$$

IV.2.3.2) Interprétation des résultats de simulation

La manipulation réalisée au cours de notre travail consiste en la réalisation et la mise en œuvre d'un circuit de commande MLI adapté à la commande d'une architecture de télé-opération et son implantation sur un circuit programmable FPGA

D'après ces résultats de simulation, on remarque que, dans la **Figure (IV.04.a),** pour la valeur tension d'entrée qui est nulle le rapport cyclique est égal à zéro (0%), ce qui implique que la manette se déplace librement, donc le vérin ne trouve pas d'obstacle (force) et peut reproduire librement les tâche imposées par la manette. Dans les **figures (IV.04.b), (IV.04.c), (IV.04.d), (IV.04.e),** on remarque que le rapport cyclique augmente à chaque fois que la tension d'entrée augmente (20%,40%,60%,80% respectivement). Ceci produit l'augmentation de la vitesse du moteur constituant le bloc de sensation. **Dans la Figure (IV.04.f)** le rapport cyclique est égal à un (100%), la vitesse du bloc de sensation haptique est maximale. alors on résume que si la tension mesurée (la tension délivrée par le pont) est importante, le rapport cyclique est aussi importante, Alors la vitesse du bloc de sensation augmente, ce qui engendre le blocage de la manette ainsi que le ressenti de la force par l'utilisateur. En contre partie, si la manette se bloque l'opérateur ne peut pas commander librement le site esclave. Par conséquent le vérin s'arrête.

D'après ces résultats de simulation, on constate que le rapport cyclique est proportionnel à l'entrée. Donc le circuit de commande MLI que nous avons conçu est adapté au contrôle de notre

architecture de télé-opération. Avec ces résultats qui satisfont notre cahier des charges, l'étape de l'analyse et de simulation s'achève. Nous entamons la partie réalisation en implantant notre circuit de commande dans le support physique FPGA.

IV.3) Implantation d'un circuit de commande sur le support physique FPGA

L'application de flot de conception donné en **Figure (II.07)** sur le schéma de commande peut être décrite comme suit : après une description comportementale en VHDL de chaque unité et module (schéma bloc), une synthèse est exécutée sur l'architecture globale. Cette étape n'est autre qu'une transformation du scchéma de commande pour chaque cycle d'horloge en un ensemble d'équations booléennes. On parle ainsi du niveau **RTL** ***« Register Transfert Level »***. Pour compléter ce flot de conception, le fonctionnement correct de cette commande a été validé par des simulations pour différents entrées. Cette opération a été effectuée à partir d'un fichier de test que l'on nommera « ***test-bench »*** et a été réalisée par l'utilisation de l'outil de simulation Modelsim. Les résultats de simulation sont montrés dans les **Figures (IV.04).**

Prés avoir réalisé le options « **Synthesize –XST** » pour la vérification du fonctionnement du système nous allons réaliser la suite du flot pour générer le fichier de programmation (**bitsream**). Pour ce la nous allons réaliser les étapes suivante : **« Implement Design** » et « **Generate Programming File** ».

IV.3.1) Implement Design : Permet de faire les **mapping et routage** nécessaires pour placer la Netlist sur le FPGA, en suivant ces étapes :

- **translation :** La translation transforme la Netlist issue de la phase de synthèse en des primitives FPGA. En plus des primitives standards, la Netlist issue de cette phase peut contenir des primitives **plus élaborées**. La translation est généralement optimisée de façon à tirer profit au mieux des ressources FPGA.

- **Mappage:** chaque élément de la **Netlist** est associé à un type de cellule logique FPGA

- **<u>Placement et routage :</u>** qui place les éléments de la Netlist **physiquement** sur le circuit FPGA, en tenant compte d'un certain nombre de contraintes : **délais de propagation**, **placement par rapport aux E/S,** etc. Dans cette étape nous allons sélectionner **« Analyse Timing/ floorplan Design (plan Ahead) »** pour configurer les entées - sorties **I/O** de notre carte FPGA. La **figure (IV.05**) nous montre la technique de cette configuration. Par ailleurs, vu l'emplacement précis de quelques signaux d'entrée/sortie, en l'occurrence le reset, l'horloge et les signaux qui présent de retour de force, nous avons défini un fichier de contrainte « **.ucf »** qui porte leurs emplacements conformément au datasheet de la carte ML501 [30].

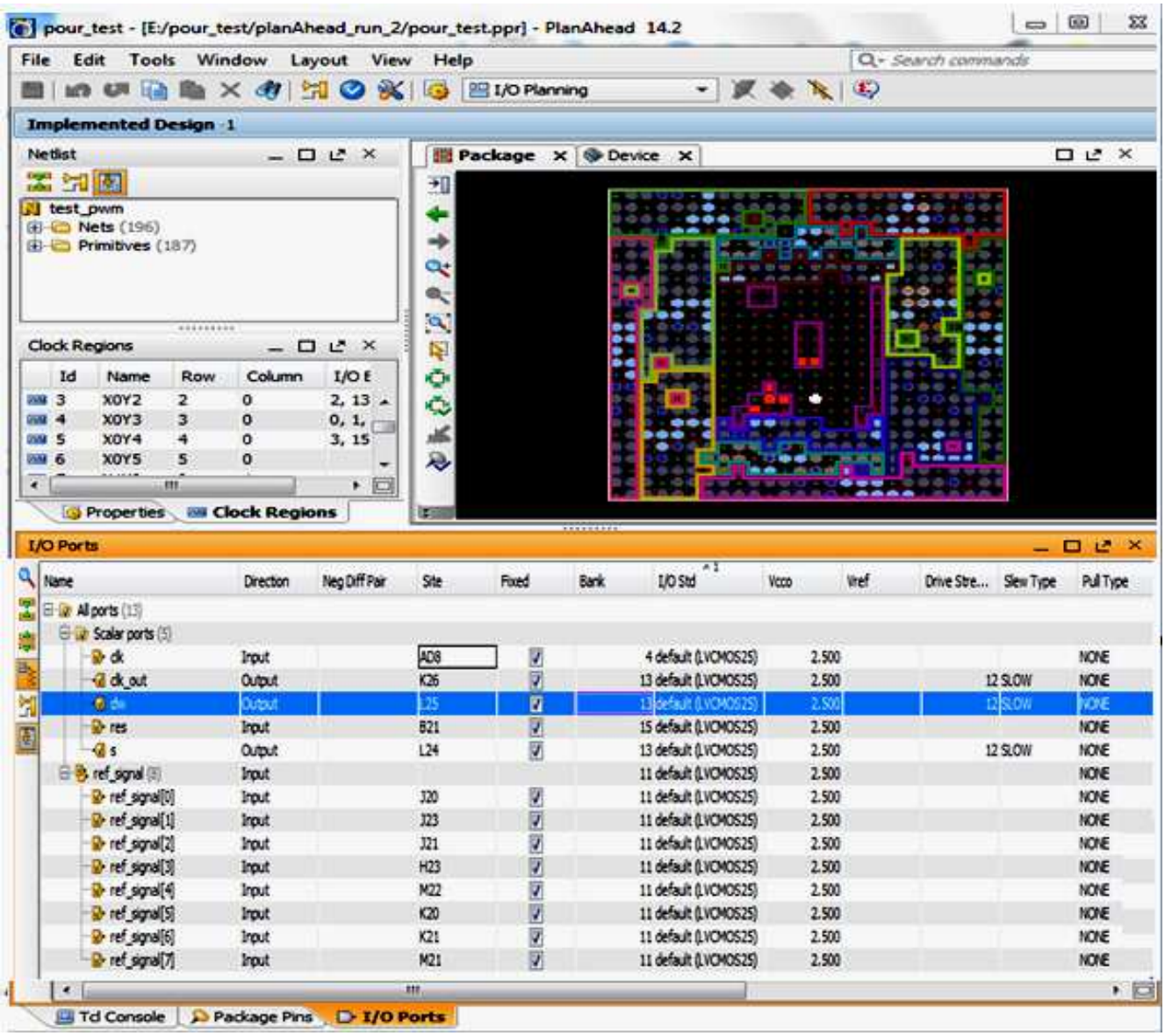


Figure (IV.05) : Configuration des entrés /sorties en utilisant plan Ahead.

Le tableau (IV.01) représente un extrait de ce datasheet. Contenant les entrés –sorties que nous allons utiliser pour la configuration de notre carte ML501.

J6 Pin	Schematic Net Name	FPGA Pin
2	HDR1_2	J20
4	HDR1_4	J23
6	HDR1_6	J21
8	HDR1_8	H23
10	HDR1_10	M22
12	HDR1_12	K20
14	HDR1_14	K21
16	HDR1_16	M21
18	HDR1_18	L25
20	HDR1_20	L24
22	HDR1_22	K26

Tableau (IV.01) : Extrait du guide de Xilinx pour la programmation des entrée/sortie de la carte ML501 [30].

Ce tableau montre les entrées – sorties choisissaient pour notre configuration. La colonne « **J6 pin** » représente les pins physiques implantés dans la carte que nous avons utilisée. Les pins (2, 4, 6, 8, 10, 12, 14, 16) sont réservés pour les entrées de notre système c'est-à-dire que c'est pins repesent le vecteur d'entrée de 8 bits. Les pins (18, 19) sers à la récupération les sorties du système qui représentent : le signal de modulation de largeur d'impulsion (PWM) et le signal continu respectivement. Ces signaux ont pour rôle d'activer le bloc de sensation haptique. La pins 22 repesent le signal d'horloge qui permet d'activer le convertisseur Analogique Numérique **CAN0804.** Et la colonne **« FPGA pin »** repesent le code de programmation de chaque pin utilisé dans le fichier de l'extension **« .ucf »**.

IV.3.2) Generate Programming File : Cette étape permet de Générer le fichier « **.bit** » utilisé pour programmer le FPGA c'est-à-dire convertir toutes les informations en un fichier pouvant être téléchargées sur la carte FPGA pour la programmer.

Remarque : Après chaque étape, un message de succès s'affiche dans la console. Dans le cas contraire, il faut corriger les erreurs ou inspecter les avertissements (warnings) pour assurer qu'ils ne proviennent pas d'erreurs dans notre code. Idéalement, on devrait voir s'afficher les trois crochets verts illustrés à **la figure (IV.06).** Ainsi lors de l'exécution de ces différentes étapes, plusieurs données deviendront disponibles dans le **« Design Summary** ».

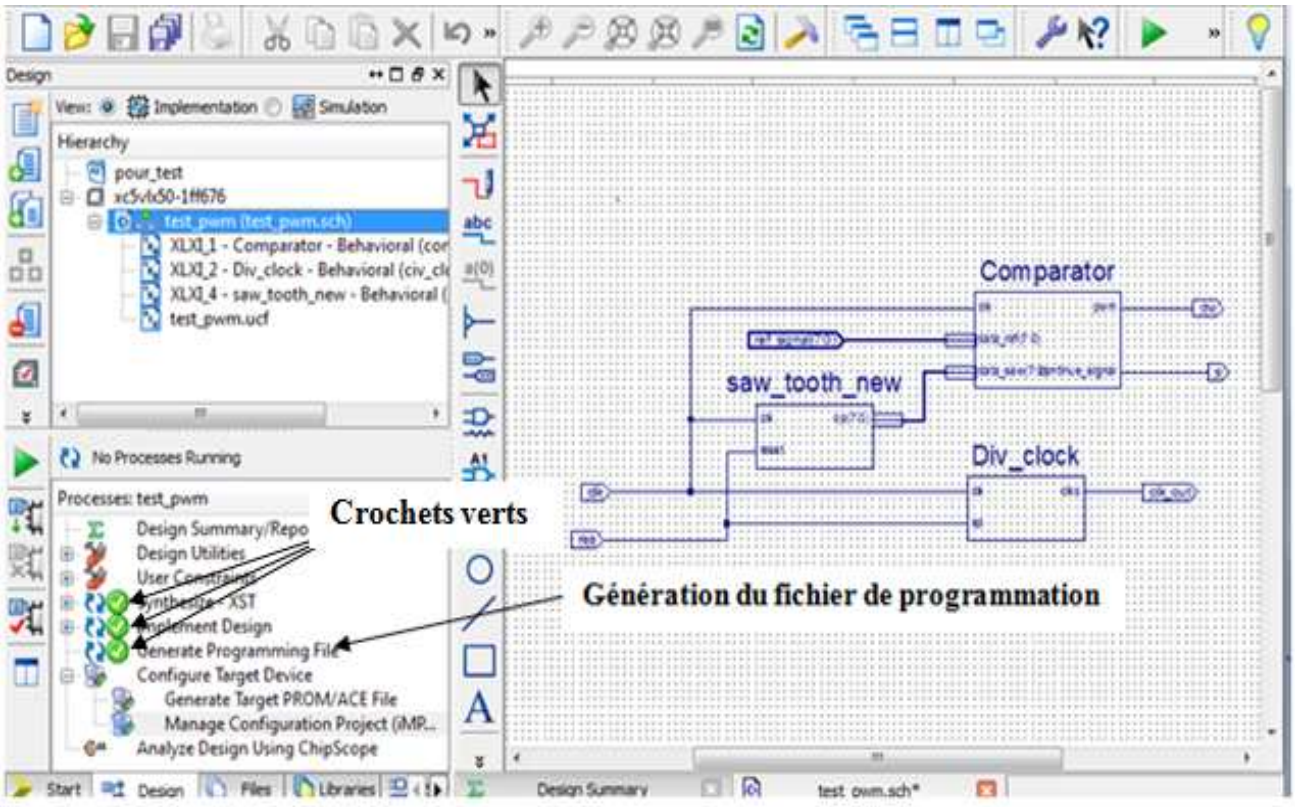


Figure (IV.06) : Etapes l'implémentation du circuit de commande.

IV.4) Programmation du FPGA et vérification

Si tout se passe bien lors des étapes de flot de conception, un fichier de programmation « test pwm.bit » sera placé dans le répertoire projet et sera prêt à être chargé sur le support FPGA. La carte FPGA devrait déjà être allumée et reliée au poste de travail (PC) par le câble USB. Cette étape est la dernière dans le processus (flot) de conception, elle va nous permettre de charger le fichier VHDL (schématique) sur le support FPGA choisi (Vertex XC5VLX50). Dans cette étape nous avons utilisé l'interface fournie par XLINX **Manage Configuration project**

« **iMPACT »** pour la configuration de notre carte. Le transfert de la solution finale est assuré par un **câble JTAG** (Joint Test Action Group).

Dans Xilinx ISE, on clic deux fois sur « **Configure Target Device** » comme le montre le schéma de **la figure (IV.08).**

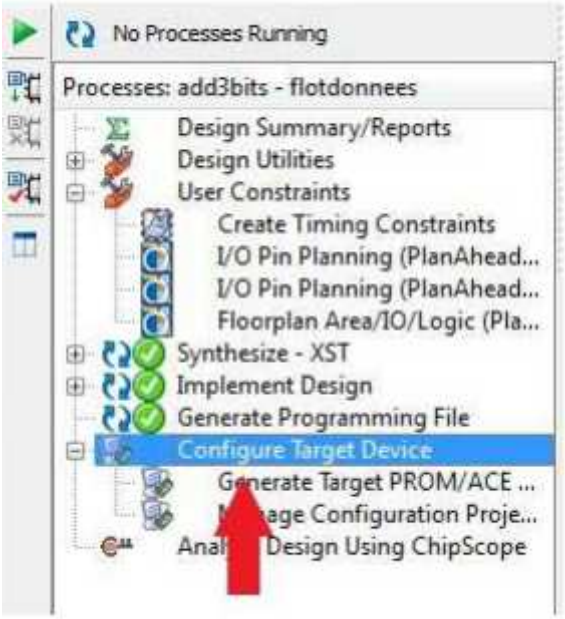
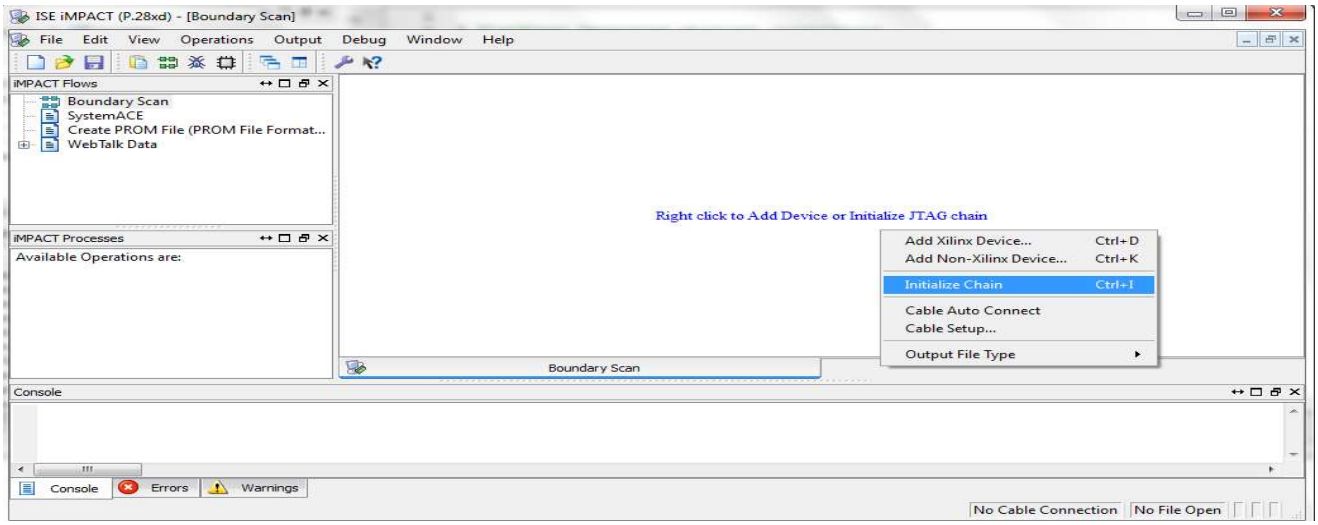
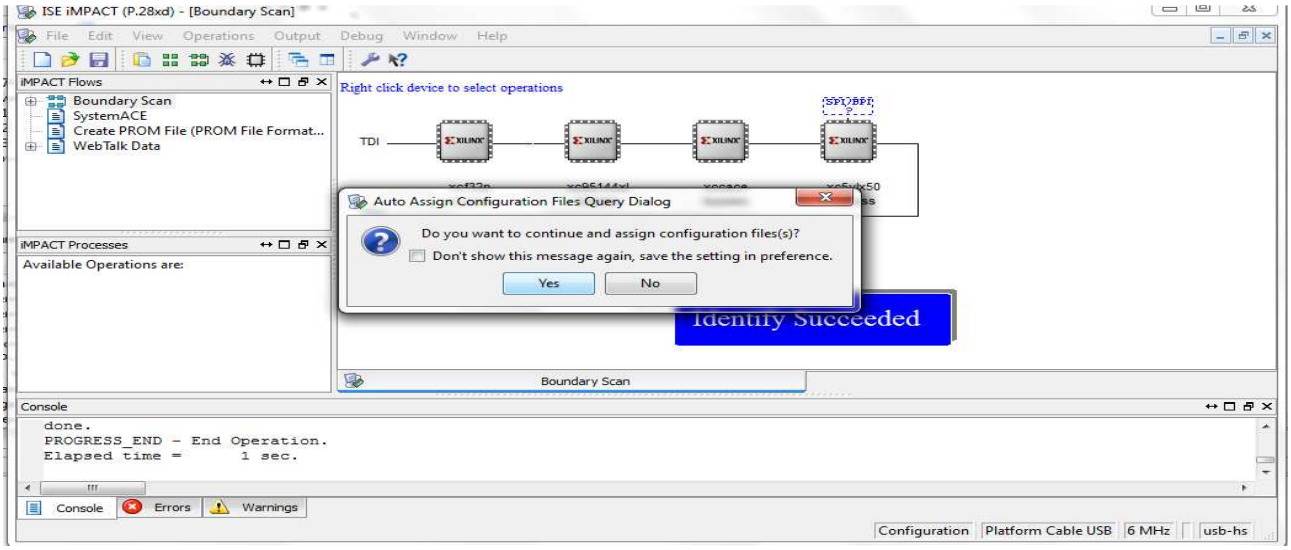


Figure (IV.07) : Choix du mode de programmation.

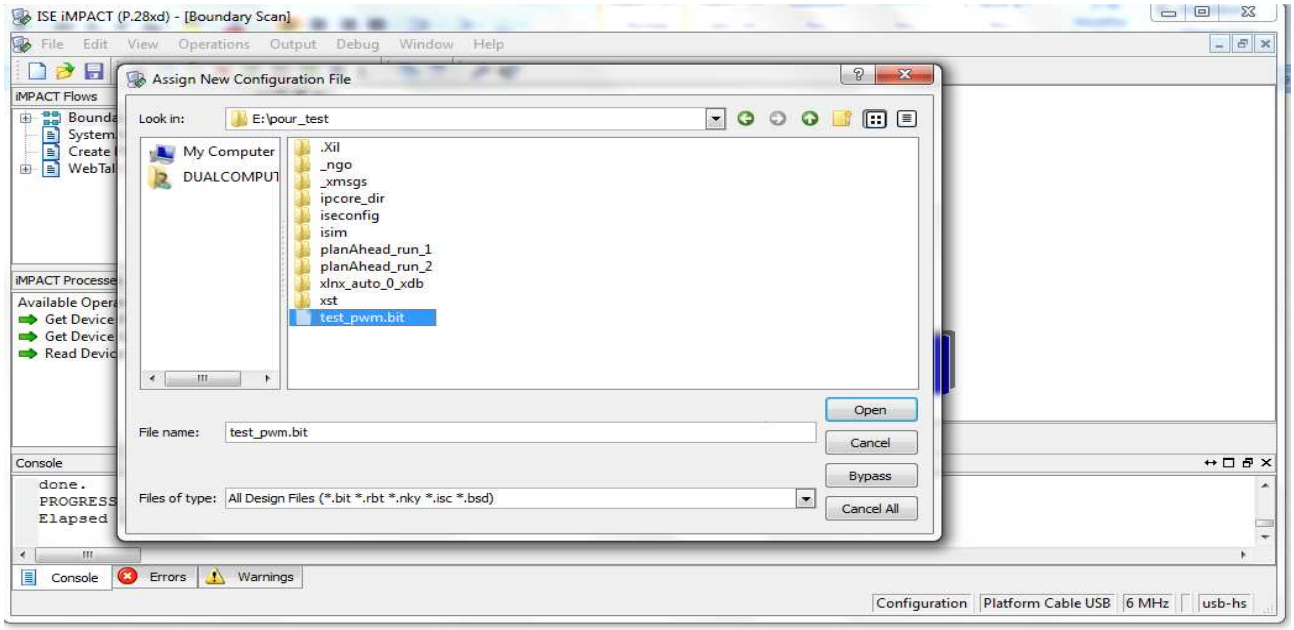
Lorsque l'outil iMPACT sera ouvert, nous allons créer un projet en faisant double clik sur « **Boundary Scan** et un espace de travail blanc s'affiche. On clique sur le bouton droit et nous choisissons « **Initialize Chain** ».



Une fenêtre « **Auto Assign Configuration Files Query Dialog** » s'affiche. On Clique sur « **Yes** »

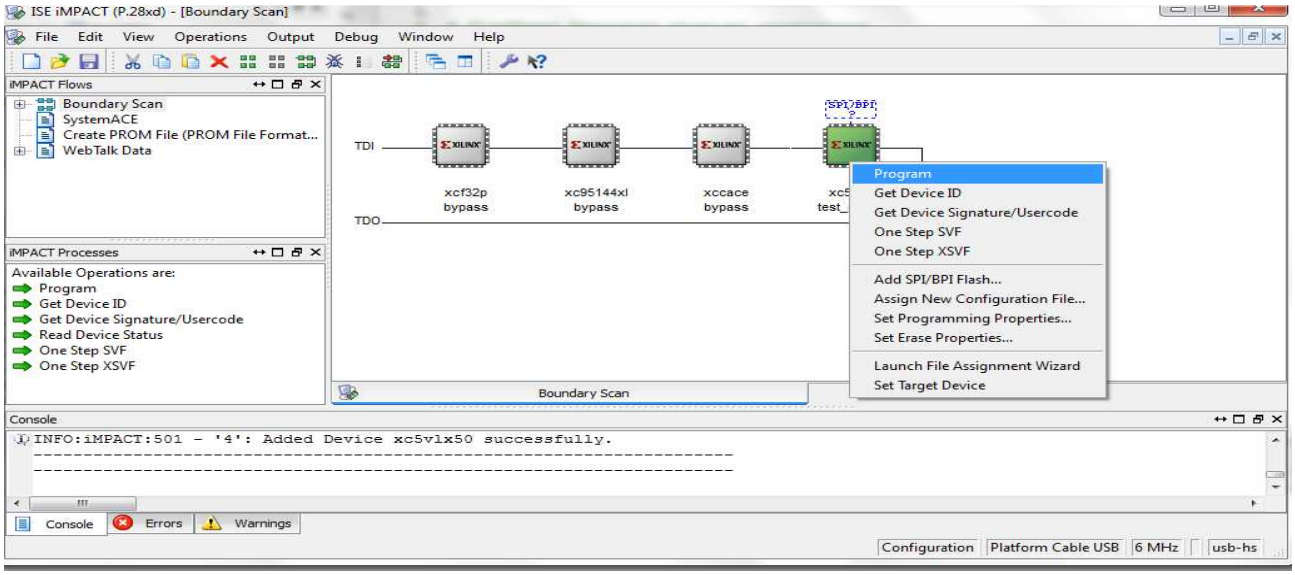


Puis, on clique sur « **bypass** » jusque à l'apparition du fichier **« .bit »** on cherche le fichier bit dans le répertoire de notre projet et nous Choisissons le fichier **« test _ pwm .bit** » que nous trouverons dans le dossier « **pour_ test »** et nous cliquons sur « **Open** ».

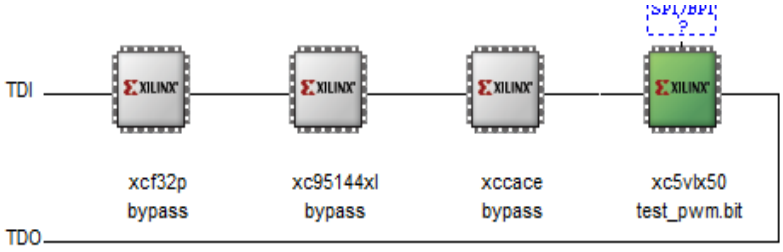


Le logiciel nous demande d'attacher **un SPI ou un BPI PROM;** nous répondons par « **No** ». Car cette option sert à charger la mémoire PROM de cette carte. Puis on clique sur « **cancel** ».

Enfin, nous cliquons sur le symbole représentant la crte FPGA, puis sur le bouton droit de la souris et nous choisissons « Program ».



Pour terminer, on clique sur « **OK** » dans la fenêtre « **Device Programming Properties** » sans rien changer. Ainsi notre design sera chargé sur le support FPGA.



IV.4.1) Résultats pratiques

Une fois notre carte est programmée, nous allons récupérons les sorties pour pouvoir attaquer les transistors qui commandent le bloc de sensation haptique. Pour des mesures de

sécurité de la carte FPGA, il est recommandé d'utiliser un opto-coupleur (isolant) qui permet de séparer la partie puissance de la partie commande. Les signaux de commande sont envoyés à l'oscilloscope ce qui nous permet de prendre les mesures pour chaque valeur délivrée par le capteur. Cela est illustré par les figures suivantes :

Dans tous les cas nous allons prendre le calibre de :

- $500ns$ sur l'axe des temps (l'axe horizontal)
- 1 V sur l'axe de l'amplitude (l'axe vertical) sur cet axe :

- **Le signal en couleur jeune représente la valeur d'entrée**
- **Le signale en couleur bleu représente la valeur de la sortie**

La période est égale à $T = 2.5\mu s$ ce qui donne une fréquence $f = 400Khz$

➢ **Premier cas** : La valeur délivrée par le capteur est presque nulle en état initial sans force.

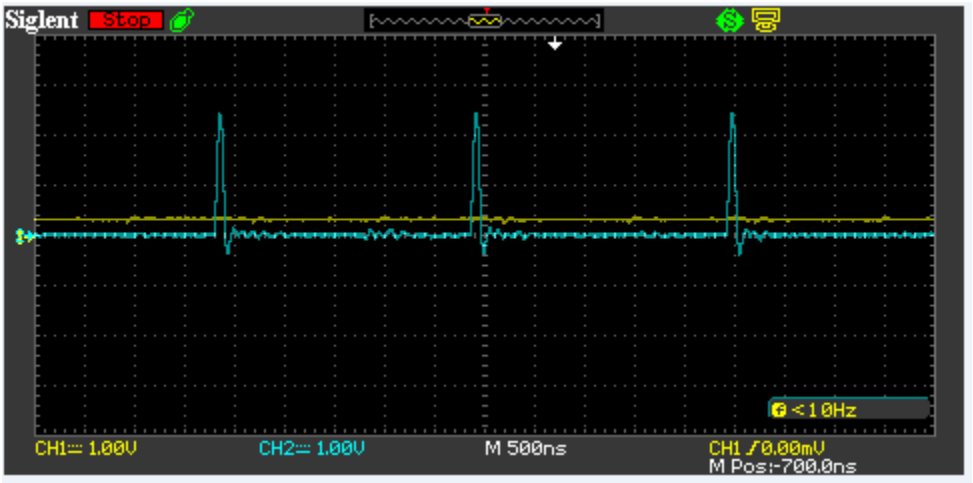


Figure (IV.08.a) : Signal de commande pour la tension d'entée V=0.4 V.

$$\alpha = \frac{ton}{T} = \frac{0.2}{5} = 0.04 \rightarrow \alpha = 4\%$$

➢ **Deuxième cas :** La valeur délivrée par le capteur vaut 0.6 V.

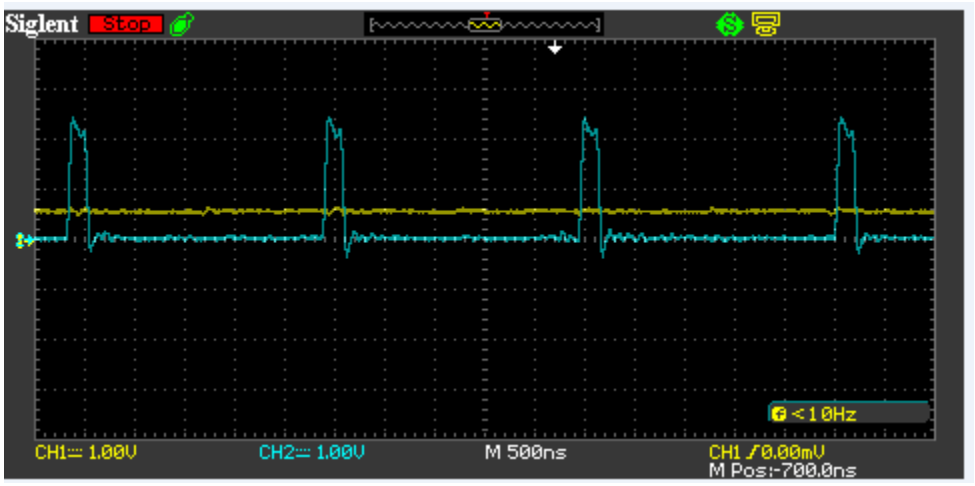


Figure (IV.08.b) : Signal de commande pour la tension d'entée V=0.6 V

$$\alpha = \frac{ton}{T} = \frac{0.4}{5} = 0.08 \rightarrow \alpha = 8\%$$

➢ **Troisième cas :** La valeur délivrée par le capteur vaut 1.2 V.

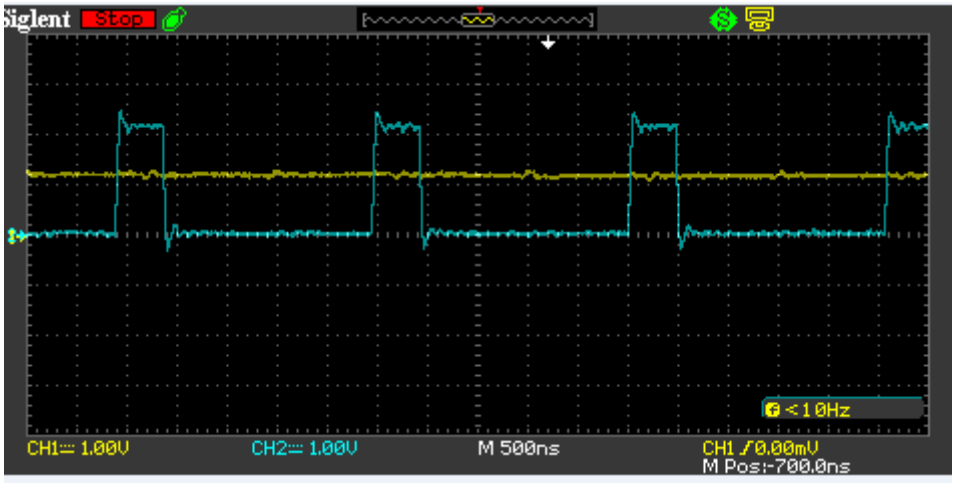


Figure (IV.08.c) : Signal de commande pour la tension d'entée V=1.1V.

$$\alpha = \frac{ton}{T} = \frac{1}{5} = 0.2 \rightarrow \alpha = 20\%$$

➢ **Quatrième cas :** La valeur délivrée par le capteur vaut 1.6 V.

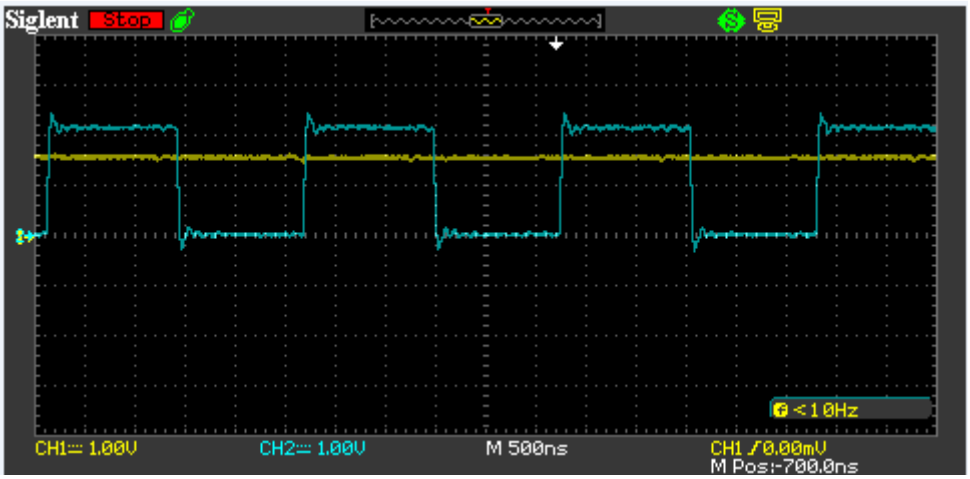


Figure (IV.08.d) : Signal de commande pour la tension d'entée V=1.6 V.

$$\alpha = \frac{ton}{T} = \frac{2.6}{5} = 0.52 \rightarrow \alpha = 52\%$$

- **<u>Cinquième cas</u> :** La valeur délivrée par le capteur vaut 2 V.

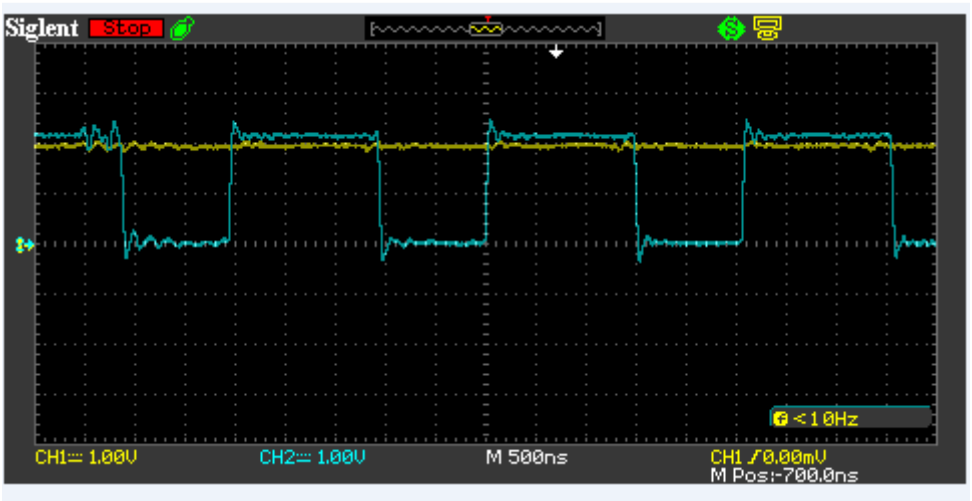


Figure (IV.08.e) : Signal de commande pour la tension d'entée V=2 V.

$$\alpha = \frac{ton}{T} = \frac{3}{5} = 0.6 \rightarrow \alpha = 60\%$$

- **<u>Sixième cas</u> :** La valeur délivrée par le capteur vaut 2.6 V.

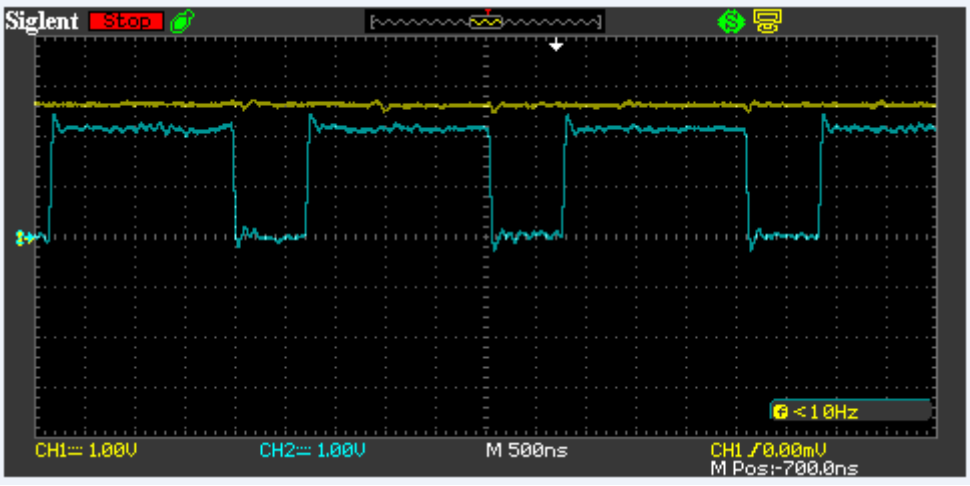


Figure (IV.08.f) : Signal de commande pour la tension d'entée V=2.6 V.

$$\alpha = \frac{ton}{T} = \frac{3.6}{5} = 0.72 \rightarrow \alpha = 72\%$$

- **<u>Septième cas :</u>**La valeur délivrée par le capteur vaut 3.4 V.

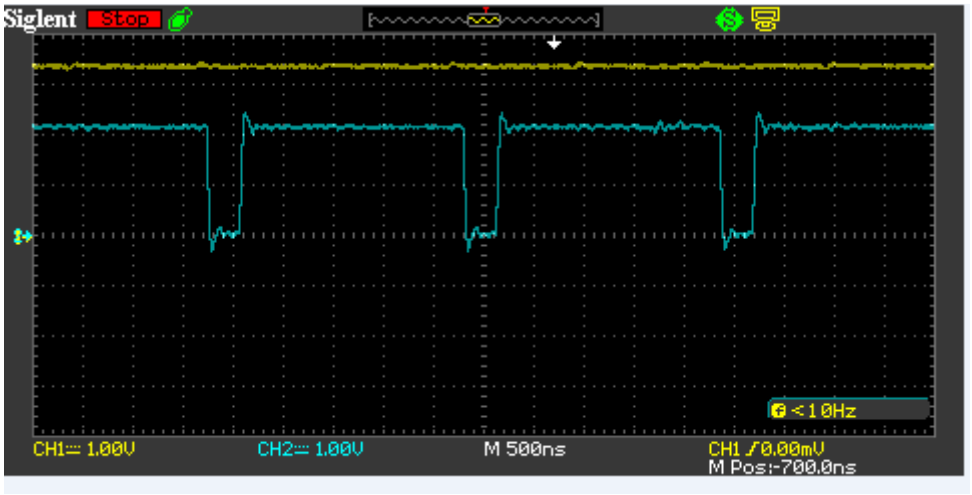


Figure (IV.08.g) : Signal de commande pour la tension d'entée V=3.4 V.

$$\alpha = \frac{ton}{T} = \frac{4.4}{5} = 0.88 \rightarrow \alpha = 88\%$$

De même jusque à ce qu'on arrive à la tension d'entrée maximale V=5 v. on trouve le rapport cyclique $\alpha = 100\%$

IV.4.2) Interprétation des résultats pratiques

D'après les résultats obtenus, on remarque que dans la **Figure (IV.08.a)** pour une tension d'entrée qui est très faible le rapport cyclique est presque nul (4%). Dans les **figures (IV.09.b), (IV.08.c), (IV.08.d), (IV.08.e), (IV.08.f), (IV.08.g)** on remarque que le rapport cyclique augmente à chaque fois que la tension d'entrée augmente (8% ; 20%, 52%, 60%, 72%, % respectivement), ce qui entraine l'augmentation de la vitesse du cloc de sensation de sensation haptique. Lorsque la tension d'entrée est maximale Le rapport cyclique atteint (100%), la vitesse du moteur qui constitue le bloc de sensation sera alors maximale. On constate que si la force mesurée par le capteur est importante, la tension délivrée par le pont est aussi importante. Donc la vitesse du moteur à courant continu augmente.

A la lumière de ces résultats, l'opérateur ressent les forces mises en jeu au cours de l'application. C'est à dire le site esclave reproduit librement la tâche demandée par le site maître si il n'ya pas de retour de force donc le capteur ne trouve pas d'obstacles. En revanche, la manette sera bloquée s'il y a retour de force, car dans ce cas, l'opérateur ne peut pas commander la manette car il lui faudra une force supérieure à la force de retour. Donc l'esclave ne peut pas reproduire la tâche demandée par le site maitre. Par conséquent le vérin s'arrête.

En outre, les caractéristiques des systèmes de télé-opération sont beaucoup plus améliorées et le retard de transmission des données est négligeable en utilisant le circuit logique programmable FPGA.

IV.5) Conclusion

Dans le présent chapitre, nous avons suivi du flot de conception pour réaliser les étapes de programmation de circuit de commande qui constitue la technique MLI. Ce circuit de commande délivre deux signaux de commande permettant de contrôler notre architecture de télé-opération détaillé dans la chapitre III. Nous avons constaté au cours de ce chapitre que les résultats pratiques sont similaires aux résultats de simulation. De cela on distance la fiabilité de la carte FPGA et son importance dans la commande des architectures des systèmes de télé-opérations.

Conclusion Générale

Dans ce travail, nous avons intéressé particulièrement à la conception en utilisant le langage matériel VHDL et à l'implantation sur cible FPGA d'un circuit de commande MLI. Cette technique est la plus utilisé pour la commande en force des architectures de télé-opération à un seul degré de liberté.

Nous avons choisi le circuit logique programmable FPGA dans notre travail pour sa flexibilité et ses performances temporelles, qui permettent de rendre les systèmes de télé-opération plus stable en éliminant le temps de retard dans les canaux de communications et transmettre plus fidèlement les informations de forces misent en jeu entre le vérin et son environnement à l'utilisateur. L'utilisation des circuits FPGA permettent d'atteindre une transparence idéale et d'obtenir un système plus performant. Néanmoins, la mise en œuvre de ces circuits pour le contrôle des systèmes est très compliquée. Afin de réduire la complexité des solutions FPGA, une méthodologie de développement appropriée donnée en chapitre II permettant de répondre aux déférentes contraintes de conceptions architecturelles des algorithmes de contrôles.

Nous avons présenté aussi au cours ce travail l'architecture de système de télé-opération à un seul degré de liberté réalisé au laboratoire L2CSP. Pour la commande de cette architecture nous avons choisi la technique la modulation de largeur d'impulsion (MLI). Car l'utilisation de ce type de contrôle trouve particulièrement son intérêt dans la commande en force des architectures bilatérales. La métrologie de conception ainsi que les étapes d'implantation de cette technique de contrôle sur un support physique programmable FPGA, ont été présentées et détaillées.

Les résultats de simulation ainsi que les résultats pratiques nous ont permis de démontrer le bon fonctionnement de notre circuit mis en œuvre et de saisir l'importance de l'utilisation des circuits logique programmable FPGA dans la commande des architectures de télé-opération à retour d'effort.

Bibliographie

[01] :Vertut. J, Coiffet. P, « les Robots. Tomes 3a : Télé opération, Evolution des technologies », éditions Hermès FRANCE ,1984.

[02] : MARQUES. N « Méthodologie et architecture adaptative pour le placement efficace des taches matérielles de tailles variables sur des partitions reconfigurables » THESE à l'université de Lorraine 2012

[03] : AIT OUALI. Z« applications des FPGA à la commande d'un moteur asynchrone » mémoire de magister à l'université MMTO.

[04]: Arnaud Lagger, "Self Reconfiguration plateform for cryptographic application", Master theisis, 2006, Swiss federal institute of technology lausanne.

[05] : Eduardo Sanchez EPFL « Circuits reconfigurables: Les FPGAs »

[06]: MEGUELLATI.M « commande à distance d'un bras manipulateur en présence d'un retard de temps constant » mémoire de magister en Robotique à l'université de BATNA.

[07] : RODRIGUEZ .A. N « ASSET : une architecture générale pour la télé robotique » thèse informatique l'université PAUL SABATIER DE TOULOUSE (sciences). 2003.

[08] : LETIER. P « Bras Exosquelette Haptique : Conception et Contrôle » thèse en sciences appliquées l'université LIBRE de BRUXELLE,2010

[09] : Wang, W. et Yuan, K .« Teleoperated manipulator for leakdetection of sealed radioactive sources». IEEE. Conf . on Robotics and Automation (2004) .

[10] :Ph LECARDONNEL & Ph LETENNEUR" Introduction à la Synthèse logique V.H.D.L " GRANVILLE – 2001

[11] :NAOUAR.M.W « commande numérique à base de composants FPGA d'une machine synchrone » thèse en Génie électrique à l'université de CERGY PONTOISE. 2007.

[13]: Claudio Melchiorri «Robotic Telemanipulation » DEIS- LAR, University Bologna

[14] : FATTOUH. A et SENAME. O « Etude de stabilité d'un système de télé-opération bilatéral avec retard de communication et contrôle d'impédance » article à Laboratoire d'Automatique des Systèmes Coopératifs. Université de Metz, France et Laboratoire d'Automatique de Grenoble ENSIEG, France.

[15] : « système à retour de force » présentation

[16] : Zarrad .W « Télé-opération avec retour d'effort pour la chirurgie mini-invasive » Thèse en Génie Informatique, Automatique et Traitement du Signal. à université Montpellier II. 2008

[17] : Yokokohj I and Yos hikawa. T « Bilateral control of master-slave manipulators for ideal kinesthetic coupling formulation and experiment» IEEE Transactions on Robotics and Automation, 10(5):605–620, Ocotber 1994.

[19]: Hashtrudi-Zaad K. et E Salcudean S. «On the Use of Local Force Feedback for Transparent Teleoperation», Proc. of IEEE, International Conference on Rob. & Auto. Detroit, Michigan, May 1999, pp 1863-1869.

[20]: Zhu W.H., Salcudean S.E. et Loewen P.D., «Optimization-based teleoperation design», 13th World Congress of IFAC, San Francisco, CA, USA, June-July 1996.

[21] : Gourdon. A, Poignet. Ph, Poisson .G, Vieyres .P& Marché. «PA New Robotic Mechanism for Medical Application», IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Atlanta, September 19-23, 1999, pp. 33-38.

[22] : Courreges. F « Détermination d'une architecture de Télé-opération Bilatérale en présence De retards » article, 2000, 17p. à Laboratoire Vision et Robotique

[23] : Gourdon. A, Poignet. Ph, Poisson. G, P. Vieyres & P. Marché, « A New Robotic Mechanism for Medical Application», IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Atlanta, September 19-23, 1999, pp. 33-38.

[25]: Wikipedia FPGA, http://en.wikipedia.org/wiki/FPGA)

[26] : « FPGA EPRM » Conservatoire National des Arts et Métiers FIP-CPI 2013–2014

[27] : Jean-Luc Danger « Les FPGA Principes innovants et tendances » 8 Mars 2012

[28]: (Xilinx, Virtex-II complete datasheet", Product Specification, no. 31, Mars 2005Available from: http://www.xilinx.com/partinfo/ds031.pdf

[30]: Xilinx Inc., 'ML501 Evaluation Platform - User Guide', UG226 (v1.4), Août 2009.

[31]: www.XILINX.com

[33] :Ph LECARDONNEL & Ph LETENNEUR « Introduction à la Synthèse logique V.H.D.L. » GRANVILLE – 2001

[32] : Douglas L. Perry « VHDL :Programming by Example » livre, Fourth Edition, Copyright © 2002 by The McGraw-Hill Companie

[34] : « didacticiel d'initiation à l'environnement de conception FPGA » École Polytechnique de Montréal - Université de Lille 1 Version 2.1 (Automne 2012)

[35] : Arnaud. T « Introduction aux circuits FPGA » INRIA LIP Arenaire Séminaire MIM 16 décembre 2003

[36] : H.Bessalah, M.Issad, L.Mahiddine, K.Messaoudi, N.Anane, M.Anane « Implémentation d'un UART sur FPGA pour des Applications en Systèmes Embarqués » article au Centre de Développement des Technologies Avancées CDTA

[37] : Nouman .Z, Klima. B, Knobloch .J « Generating PWM Signals With Variable Duty From 0% to 100% Based FPGA SPARTAN3AN » article, Department of Power Electrical and Electronic Engineering Brno University of Technology ,vol .4, N.4. December 2013

Annexe A : Un capteur de force, jauges de déformation, pont de Wheatstone

1- Capteur de force

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (très souvent électrique). Cette grande représentative de la grandeur prélevée est utilisable des fine de mesure ou de commande. C'est-à-dire transforment les grandeurs physiques indispensables aux objectifs des mesures, en grandeurs électriques facilement mesurables et enregistrables. Il existe plusieurs type de capteur parmi les quelles on distingue le capteur de force à jauge de contrainte donné par la **figure (A.01**)

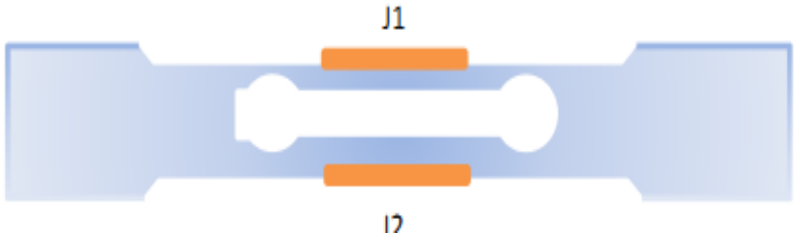
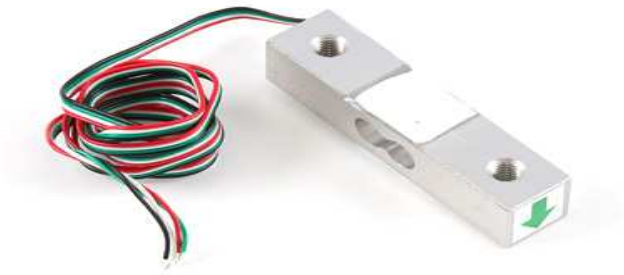


Figure (A.01) : Schéma du capteur Jauge de contrainte.

❖ **Datasheet du capteur de force**

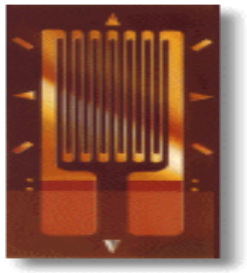


Color	Red	Green	White	Black
Bridge Connector	5V	+	-	GND

2- Principe des jauges

Sous sa forme la plus simple, une jauge est constituée d'un très fin fil (2 microns) conducteur collé sur un support. Cette feuille très mince est arrangée suivant la forme de la

Figure(A.02)**(A.02).** Les brins de fil constituant la jauge étant principalement alignés suivant la direction de mesure, on peut admettre que le fil subit les mêmes déformations que la surface sur laquelle la jauge est collée.



Figure(A.02) : Jauge de contrainte.

Ainsi si une contrainte se présente de haut en bas J1 travaille en compression et J2 en traction et vice versa **Figure (A.01),** les deux contraintes subissent les mêmes déformations.
La résistance d'une jauge est donne par :

$$R = \frac{\rho l}{A}$$

Avec :

1. ***R* :** La résistance.
2. ***ρ* :** La résistivité du conducteur dont est formée la jauge.
3. ***l*** : La longueur du fil conducteur de la jauge.
4. ***A* :** La section de la jauge

En prenant le logarithme népérien et la différentielle de chaque membre et en considérant que les paramètres varient très peu lors de la mesure, on obtient la relation

$$\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dl}{l} - \frac{dA}{A}$$

Or : $A = \pi \frac{D^2}{4}$ D'où $\frac{dA}{A} = 2\frac{dD}{D}$

Mais $\frac{dD}{D} = -v\frac{\boldsymbol{dl}}{\boldsymbol{l}}$ avec v le coefficient de poisson de la jauge.

Alors : $\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dl}{l}(1 + 2v)$

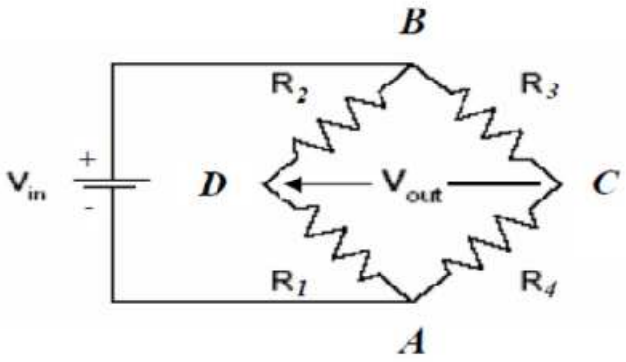
On introduit alors k « facteur de jauge » qui caractérise la variation de la résistance en fonction de sa déformation axiale

$\frac{dl}{l} = \varepsilon$ ce qui donne $k = \frac{\frac{dR}{R}}{\varepsilon} = 1 + 2v + \frac{\frac{d\rho}{\rho}}{\varepsilon}$

Alors $\frac{dR}{R} = k\varepsilon$

3- Dispositif du pont

Le pont de Wheatstone est constitué de quatre résistances disposées en quadrilatère comme le montre la **Figure (A.03)**



.

Figure (A.03): Pont de Wheatstone.

Le pont est un moyen très précis pour mesurer les résistances qui sont utilisées pour quantifier l'élongation du matériau. Le qui caractérise le montage est le pont, entre les points C et D, contenant un voltmètre qui détecte la tension lorsque A et B sont reliés à une source de tension ou de courant. L'étude théorique de l'équilibrage du pont est donnée en appliquant le théorème de Millman en C et en D, on aura ainsi:

$$V_C = \frac{(\frac{V_A}{R_4} + \frac{V_B}{R_3})}{(\frac{1}{R_4} + \frac{1}{R_3})} = \frac{(V_A R_3 + V_B V_4)}{(R_3 + R_4)} V_D = \frac{(\frac{V_A}{R_1} + \frac{V_B}{R_2})}{(\frac{1}{R_1} + \frac{1}{R_2})} = \frac{(V_A R_2 + V_B V_1)}{(R_2 + R_1)}$$

$$V_{in} = (V_B - V_A) V_{out} = (V_D - V_C)$$

Equation de base du pont est donnée comme suit :

$$V_{out} = \frac{V_{in}(R_1R_3 - R_2R_4)}{(R_2 + R_1)(R_3 + R_4)}$$

Le pont est dit équilibré lorsque V_{out} est nul quelle que soit l'entrée.Cela se traduit donc parle fait que :

$$\textit{(Equilibrage du pont)} \Longleftrightarrow (\boldsymbol{R_1R_3 - R_2R_4}) = \mathbf{0}$$

Dans notre cas c'est un demi-pont de Wheatstone qui est utilisé car on a deux résistances qui varient. Ainsi le pont est initialement équilibré mais R_3varie de dR et R_4de – dR

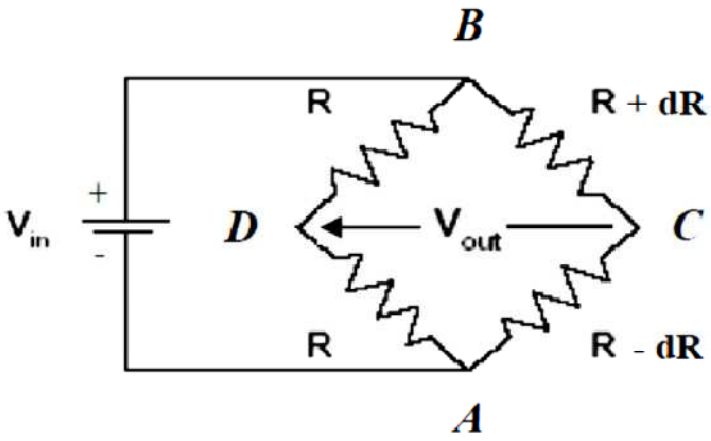


Figure (A.04) : Pont de Wheatstone.

L'équation de base devient alors,

$$dV_{out} = \frac{V_{in}(4dR\,R^3 - (-4dR\,R^3))}{(4dR\,R^2)(4drR\,R^2)} = \frac{V_{in}}{2}\frac{dR}{R}$$

Soit :

$$V_{out} = \frac{V_{in}}{2}\frac{\Delta R}{R}$$

Et comme $\frac{dR}{R} = k\varepsilon$

Alors on aura

$$V_{out} = \frac{V_{in}}{2}k\varepsilon$$

Conclusion

La tension délivrée par le pont de Wheatstone intégré dans un capteur de jauge de contrainte est dépend de la déformation (contrainte) qui se présente de haut en bas du capteur en compression

Annexe B : Amplificateur INA126

1- **Amplificateur INA126 :**L'INA126 est un instrument d'amplification précis et fidèle aux faibles bruits. Sa conception à base de deux amplificateurs opérationnels, offre une excellente performance avec un très faible courant de repos (175µ A/canal).

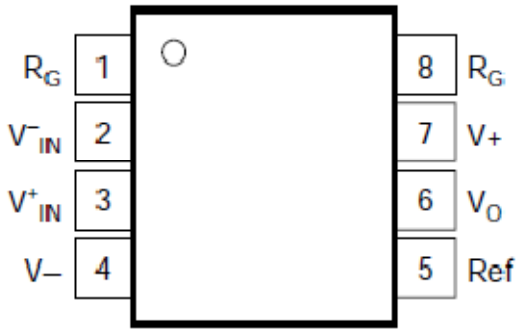
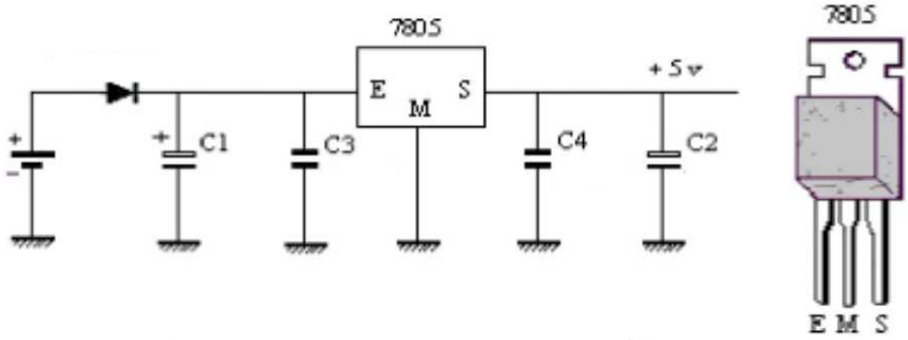


Figure III.9 : Configuration des pins (INA 126).

Annexe C : Régulateur 7805

Régulateur de tension 7805

Ce régulateur est un régulateur de tension positive, idéal pour stabiliser une tension à une valeur fixe. Il convient pour tous montages électroniques nécessitant une tension fixe, sur notre projet on a besoin d'une tension fixe (5v).

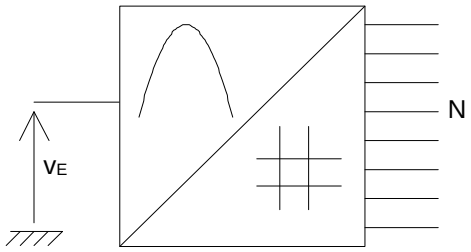


Annexe E : Convertisseur Analogique-Numérique

1- Convertisseur Analogique Numérique (CAN)

Deux types de grandeurs sont utilisés en Sciences physiques : les grandeurs analogiques et les grandeurs numériques. Pour mesurer, avec un ordinateur, une tension de 3,78 V, il faut convertir cette grandeur analogique en grandeur numérique constituée d'une série de 0 et de 1, seule grandeur compréhensible par l'ordinateur. Pour cela, on utilise un convertisseur analogique numérique ou CAN (DAC en anglais : Digital Analog Converter) où :

- La tension d'entrée v_E est analogique
- La sortie N est numérique (a_8 $a_{7...}a_1$ a_0 constitue un mot binaire de 8 bits).



2- Le traitement numérique de l'information

Pour effectuer une acquisition on réalise un échantillonnage du signal analogique d'entrée Ve. En fait à intervalle régulier on mesure la valeur de cette tension, cet intervalle est appelé période d'échantillonnage. Ce sont ces valeurs analogiques Ve* qu'il nous faut maintenant convertir en grandeurs numériques.

La conversion **analogique / digital** est réalisée grâce au circuit **ADC 0804.** Sur son entrée, on envoie un signal analogique et on retrouve le nombre binaire correspondant sur sa sortie. Ce circuit est réalisé en technologie **CMOS** et est assez complexe. Il peut être utilisé dans un système à FPGA, à microprocesseur et d'autres cartes numériques.

3- fonctionnement du convertisseur ADC0804

La figure (A.01) présent le schéma synoptique du convertisseur Analogique-Numérique(ADC0804).

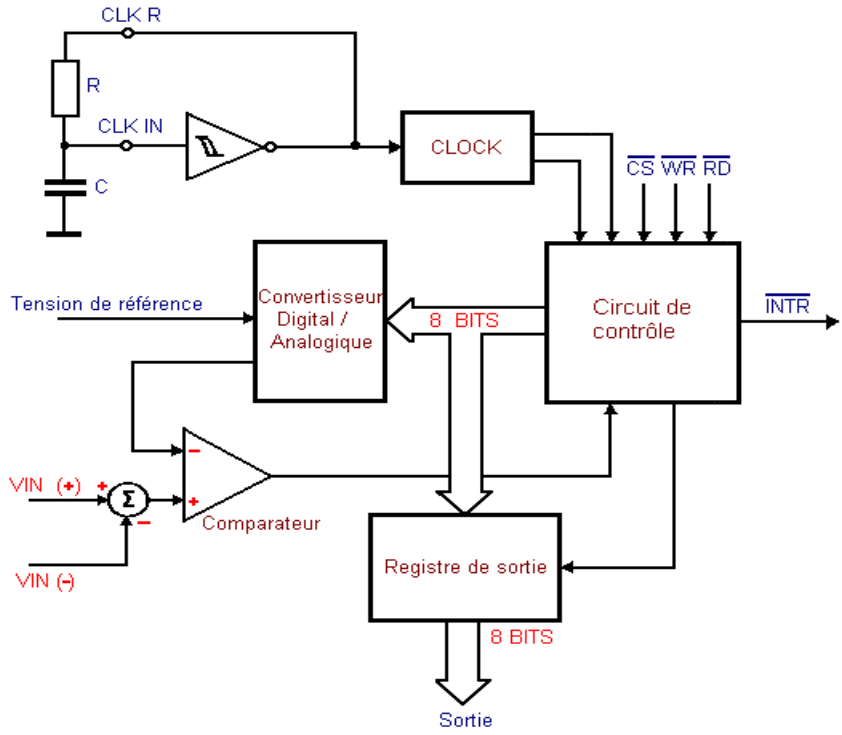
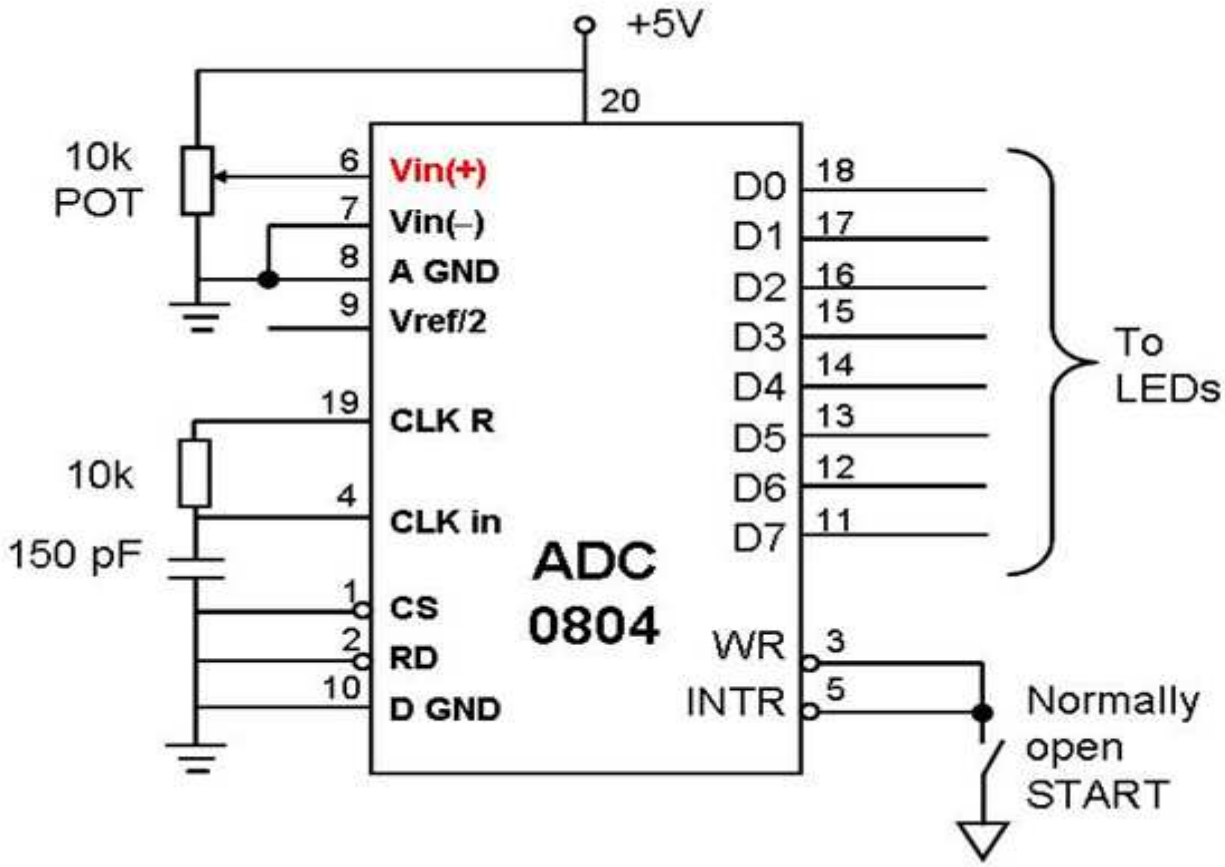


Figure (A.01) : Schéma synoptique de l'ADC 0804.

Le convertisseur possède plusieurs broches de commande qui permettent de gérer son fonctionnement. Le début de conversion s'effectue en appliquant un niveau **L** sur la borne de commande **WR**. La borne **RD** permet de valider le registre de sortie **;** au niveau **H**, la sortie est à l'état **haute impédance;** au niveau **L**, le contenu du registre de sortie est présent sur la sortie. La sortie **INTR** signale la fin de la conversion. Pour cela, elle passe au niveau **L**. L'entrée **CS** (Chip Select) permet de sélectionner le convertisseur. Lorsqu'elle est au niveau **L**, le convertisseur peut fonctionner. Les différentes étapes (approximations) s'effectuent au rythme d'un signal d'horloge. On peut utiliser un signal d'horloge externe appliqué sur l'entrée **CLK IN** ou réaliser un montage avec une résistance et un condensateur externes reliés à un trigger de Schmitt incorporé dans le boîtier. Le signal analogique est appliqué à l'entrée **VIN (+).** L'autre entrée **VIN (-)** peut être soit reliée à la masse, soit être portée à une tension permettant le tarage du convertisseur en début d'échelle, soit permettre de soustraire une tension continue du signal d'entrée. Le brochage du circuit intégré est donné **figure(A.02).**



Figure(A.02) : Schéma de brochage de l'ADC 804.