

UNIVERSITE KASDI MERBAH OUARGLA
Faculté des Nouvelles Technologies de l'Information et de la
Communication
Département d'Informatique et Technologie de l'Information



Mémoire

MASTER ACADEMIQUE

Domaine : Mathématiques et Informatique

Filière : Informatique

Présenté par :

BEGGARI Samira Informatique Industrielle

FARSI Meriem Informatique Fondamentale

Thème

Outil graphique pour l'apprentissage du
langage MIPS

Devant le jury

Mme. KHELILI Khalida Farida MAA UKM Ouargla Examineur

M. BENMIR Abdelkader MAA UKM Ouargla Examineur

Mme. BENKHROUROU Chafika MAA UKM Ouargla Encadreur

Année universitaire : 2017 /2018

REMERCIEMENTS

A l'issue de ce travail, nous voulons d'abord remercier Allah de nous guider et donner la force, le courage et la patience pour tirer le meilleur parti de ce travail: MERCI "Dieu"

Nous tenons à remercier toutes les personnes qui ont contribué de manière directe ou indirecte à l'aboutissement de ce travail :

En premier lieu, nous remercions très sincèrement Mme Chafika Benkherourou pour son encadrement et surtout pour l'aide précieuse qu'elle nous a apportée tout au long du travail.

Nous remercions Monsieur Salah Euschi Chef de département à l'université de Ouargla pour son aide démesurée

Nous remercions également toutes les personnes qui nous ont aidées, de près ou de loin pour la réalisation de ce travail.

Nous tenons à remercier les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger notre travail.

Meriem et Samira

TABLES DES MATIERES

| | |
|--|----|
| Remerciements | i |
| Tables des matières | ii |
| Liste des figures | v |
| Liste des Tables | v |
| Introduction générale | 2 |
| Chapitre 1: Etude Préalable | 4 |
| I. INTRODUCTION | 4 |
| I.1 Présentation générale du projet : | 4 |
| I.2 Objectifs du projet | 4 |
| I.3 Description de l'existant: | 5 |
| I.4 Critiques de l'existant: | 6 |
| I.5 La solution proposée : | 6 |
| I.6 Conclusion : | 6 |
| Chapitre 2: E-learning | 7 |
| II. INTRODUCTION | 7 |
| II.1 Formation à distance | 7 |
| II.2 L'évolution historique de la formation à distance : | 7 |
| II.2.1 Première génération:enseignement par correspondance..... | 7 |
| II.2.1 Deuxième génération : enseignement télévisé et modèle industriel :..... | 8 |
| II.2.3 Troisième génération : enseignement à distance interactif:..... | 8 |
| II.3 E-Learning | 9 |
| II.4 Définition | 9 |
| II.5 Principe du E-Learning | 10 |
| II.6 L'évolution du e-learning | 10 |
| II.7 Les différentes formules du E-Learning | 11 |
| II.7.1 Un cours magistral synchrone | 11 |
| II.7.2 L'assistance | 12 |
| II.7.3 L'auto-formation connectée | 12 |
| II.8 La plate-forme du e-learning | 12 |
| II.9 Les critères de succès du E-Learning | 13 |

| | | |
|---|--|-----------|
| II.9.1 | Dans la mise en place | 13 |
| II.9.2 | Pendant le fonctionnement | 14 |
| II.10 | Les avantages et les inconvénients du E-Learning | 14 |
| II.10.1 | Les avantages : | 14 |
| II.10.2 | Les inconvénients | 15 |
| II.11 | Comparaison du e-learning et de la formation traditionnelle | 16 |
| II.12 | Conclusion | 16 |
| Chapitre 3: Le processeur Mips R3000 | | 17 |
| III. | Introduction | 17 |
| III.1 | Notion du Processeur : | 17 |
| III.2 | Architecture de base d'un microprocesseur | 18 |
| III.2.1 | Unité de commande:..... | 18 |
| III.2.2 | Unité arithmétique et logique: | 18 |
| III.3 | Présentation du processeur MIPS R3000 : | 18 |
| III.4 | L'architecture MIPS | 19 |
| III.4.1 | ARCHITECTURE INTERNE..... | 19 |
| III.4.2 | CHEMIN DE DONNEES..... | 20 |
| III.4.3 | Registres visibles..... | 21 |
| III.5 | Co-processeurs de MIPS: | 23 |
| III.6 | Langage d'assemblage du MIPS R3000 | 23 |
| III.6.1 | Programmation en langage d'assemblage: | 23 |
| III.6.2 | Organisation de la mémoire | 24 |
| III.6.3 | Instructions..... | 26 |
| III.6.4 | Macro-Instructions..... | 26 |
| III.6.5 | Les commentaires:..... | 27 |
| III.6.6 | Déclaration de données: | 27 |
| III.7 | Conclusion : | 28 |
| Chapitre 4: Conception | | 29 |
| IV. | Introduction | 29 |
| IV.1 | Spécification des besoins: | 29 |
| IV.2 | Les acteurs du système: | 29 |
| IV.3 | Identification des cas d'utilisation : | 29 |

| | | |
|------------------------------------|--|-----------|
| IV.4 | Diagrammes d'activités : | 32 |
| IV.5 | Conclusion | 33 |
| Chapitre 5 : Implémentation | | 34 |
| V. | Introduction : | 34 |
| V.1 | Choix de l'environnement (l'éditeur): | 34 |
| V.2 | Implémentation de l'application (Interfaces): | 35 |
| V.3 | Cycle d'exécutions des commandes sur un processeur MIPS : | 40 |
| V.4 | Implémentation d'une simulation des cycles d'exécution d'une instruction machine sur processeur MIPS3000: | 42 |
| V.5 | Code source HTML: | 44 |
| V.6 | Passage des scénarios des instructions machines: | 46 |
| V.7 | conclusion: | 49 |
| Conclusion générale | | 50 |

LISTE DES FIGURES

| | |
|--|----|
| Figure 1 : Acteur du E-Learning. [13]..... | 10 |
| Figure 2 : Exemple du Processeur. [13] | 18 |
| Figure 3 : MIPS R3000 Architecture interne [12]. | 20 |
| Figure 4 : Programmation en langage d'assemblage [22]..... | 24 |
| Figure 5 : Organisation de la mémoire [12]. | 25 |
| Figure 6: Exemple d'une déclaration de donnée- 1- [22]. | 27 |
| Figure 7 : Exemple d'une déclaration de donnée- 2- [22]..... | 27 |
| Figure 8 : Diagramme de cas d'utilisation « Enseignant » | 30 |
| Figure 9 : Diagramme de cas d'utilisation « apprenant »..... | 31 |
| Figure 10 : Diagramme d'activité « Ajouter cours » | 32 |
| Figure 11 : interface de notre interface de cours..... | 35 |
| Figure 12 : Exemple d'interface de cours avec Menu..... | 36 |
| Figure 13 : Le volet Architecture des processeurs..... | 37 |
| Figure 14 : Exemple d'application de simulation Mars..... | 38 |
| Figure 15 : Interface de QUIZZ et Test..... | 39 |
| Figure 16 : Support de cours de l'encadreur.. | 40 |
| Figure 17 : Cycle de commandes de processeur MIPS3000..... | 41 |
| Figure 18 : Cycles d'exécution des instructions Assembleur. | 42 |
| Figure 19 : exemple d'interface de simulation intégrée..... | 43 |
| Figure 20 : Interface intégrée de simulation d'exécution des commandes | 44 |
| Figure 21 : Vue globale du code HTML utilisé pour l'animation des scénarios..... | 45 |
| Figure 22 : L'étape de recherche de l'instruction Add | 46 |
| Figure 23 : l'étape décodage d'instruction Add | 47 |
| Figure 24 : l'étape d'exécution de la commande Add..... | 48 |

LISTE DES TABLES

| | |
|---|----|
| Tableau 1: Analyse comparative de notre travail avec les travaux similaires [14]..... | 16 |
| Tableau 2:Correspondance entre les noms des registres et leurs numéros.[12] | 22 |
| Tableau 3 : Les instructions du langage MIPS[12] | 26 |
| Tableau 4 : Les différents types des données en Mips [22]..... | 28 |
| Tableau 5: Identification des cas d'utilisation | 30 |

Résumé

Pour Apprendre l'assembleur, les étudiants doivent avoir une bonne connaissance de la structure interne du microprocesseur. Afin de répondre à cette problématique, nous proposons dans ce mémoire une plate-forme pour illustrer l'architecture interne du microprocesseur Mips R3000 et fournir des tutoriels. Notre solution, fondée sur des techniques d'e-learning, permet aux étudiants l'apprentissage de l'assembleur en vue de les conduire vers l'autonomie et aider ceux en difficulté à développer leurs connaissances.

Mots-clés : *Langage MIPS, MIPS R3000, Microprocesseur, e-learning*

Abstract

To learn assembler, students must have a good knowledge of the internal structure of the microprocessor. In order to answer this problem, we propose in this work a platform to illustrate the internal architecture of the Mips R3000 microprocessor and to provide tutorials. Our solution, based on e-learning techniques, allows students to learn the assembly to lead them to autonomy and help those in difficulty to develop their knowledge.

Keywords: *MIPS language, MIPS R3000, Microprocessor, e-learning*

Mips R3000

MIPS R3000 MIPS :

INTRODUCTION GENERALE

De nos jours, beaucoup d'entreprises, institutions et universités, utilisent le e-learning. Il est un moyen de formation de plus en plus utilisé à travers le monde parce qu'il peut se révéler aussi efficace que la formation traditionnelle. En d'autres termes, il ouvre de nouvelles voies à l'apprentissage en général. Il aide les apprenants à améliorer leurs performances et connaissances et facilite la tâche pour les enseignants.

C'est dans cette optique que nous abordons la problématique du e-learning dans l'apprentissage du langage MIPS R3000. Dans le département d'Informatique et de la Technologie de l'Information, les étudiants de la 2ème année Informatique trouvent des difficultés dans la compréhension du fonctionnement du microprocesseur MIPS. Le cours est donné d'une manière théorique et les travaux pratiques concernent uniquement la partie programmation en langage d'assemblage. Pour apprendre l'assembleur, les étudiants doivent avoir une bonne connaissance de la structure interne du microprocesseur.

Pour cette raison, nous proposons de leur présenter une plate-forme pour illustrer l'architecture interne du microprocesseur Mips R3000 et fournir des tutoriels pour l'apprentissage de l'assembleur en vue de conduire les apprenants vers l'autonomie et aider ceux en difficulté de développer leurs connaissances.

Nous avons constaté que la majorité des étudiants utilisent de plus en plus les nouvelles technologies, cela nous a motivés pour leur offrir un outil d'apprentissage qui leur permet de développer leurs connaissances selon leur rythme.

L'objectif de notre travail est de permettre aux étudiants d'acquérir les notions de base des « Architectures des Ordinateurs » notamment le MIPS R3000, de les aider à comprendre les cours et de leur faciliter la réalisation des travaux pratiques. Grâce à l'aspect visuel avec lequel est présenté le microprocesseur, nous cherchons à stimuler et motiver les apprenants qui ont des difficultés à suivre leur cours même en dehors des séances habituelles. Nous visons aussi à Expliquer graphiquement le cycle de recherche et d'exécution d'une commande et présenter une description pour le jeu d'instructions du langage. Pour l'évaluation des apprenants, nous proposons une série de tests et quizz.

Notre travail se découpe en cinq parties :

Le premier chapitre concerne le cadre théorique de notre travail. Au début, nous présentons l'organisme d'accueil au sein duquel s'est déroulé notre projet. Ensuite nous passons à la définition des objectifs à atteindre, ainsi que notre champ d'étude.

Dans le deuxième chapitre, nous exposons les différents concepts du domaine e-learning. Nous mettons l'accent sur sa définition, ses caractéristiques, ses avantages et ses limites. Ensuite, nous présentons quelques plates-formes utilisées dans le domaine. Et nous terminerons le chapitre par une comparaison entre le e-learning et la formation traditionnelle.

Le troisième chapitre est consacré à la présentation du microprocesseur MIPS R3000. Nous commençons par la description de l'architecture interne du microprocesseur en présentant ses registres de base, les modes d'utilisation et son Co-processeur. Ensuite, nous entamons le langage MIPS avec quelques exemples.

Le quatrième chapitre expose la démarche conceptuelle que nous avons suivie pour créer notre plate-forme. Pour cela, nous utilisons la méthode UML pour l'établissement des différents diagrammes. Nous spécifions les acteurs du système. Par la suite, nous présenterons les différents cas d'utilisation. Nous terminons par présenter le diagramme d'activité.

Le cinquième chapitre est consacré pour décrire notre démarche de développement pour la plate-forme. Nous allons commencer par présenter les outils utilisés pour son implémentation. Puis, nous allons décrire la mise en place d'un scénario d'application pour un cycle d'exécution d'une instruction MIPS.

Enfin, nous concluons ce mémoire en présentant les contributions apportées durant ce projet. Nous dégagerons ensuite les perspectives envisageables pour la suite de nos travaux.

Chapitre

1

Etude préalable

I. INTRODUCTION

Dans ce chapitre, nous allons présenter le cadre théorique de notre travail. Au début, nous présentons l'organisme d'accueil au sein duquel s'est déroulé notre projet. Ensuite nous passons à la définition des objectifs à atteindre, ainsi que notre champ d'étude.

I.1 PRESENTATION GENERALE DU PROJET :

Le projet s'est élaboré au sein du département d'informatique et de la technologie de l'information à l'université KASDI Merbah Ouargla. Parmi les modules assurés en deuxième année Licence le module « Architecture des Ordinateurs ». L'objectif de ce module est de détailler le fonctionnement général de l'ordinateur, ses composants ainsi que le langage d'assemblage. Le microprocesseur concerné est le MIPS R3000.

Pour notre projet de fin d'étude en Master 2, il nous a été proposé de mettre en place, sur un site web, une plate-forme d'apprentissage pour aider les étudiants d'acquérir les notions de base des architectures des ordinateurs notamment le MIPS R3000 et d'apprendre la programmation en langage d'assemblage. Son utilisation aide les apprenants à comprendre le cours du module « Architecture des Ordinateurs » et leur faciliter la réalisation de TP. L'évaluation interactive et les Quiz facilitent l'opération d'apprentissage en donnant plus de motivation pour l'apprenant.

L'aspect visuel stimule et motive les apprenants qui ont des difficultés à suivre leur cours même en dehors des séances habituelles.

Pour arriver à réaliser notre projet, nous allons suivre ces étapes :

1. Faire une étude préalable.
2. Définir le cadre de notre projet en présentant les notions du e-learning et du Mips R3000.
3. Conception et développement de la solution proposée.

I.2 OBJECTIFS DU PROJET

Le module « Architecture des ordinateurs » permet de simplifier la compréhension du fonctionnement d'un ordinateur, en décrivant le fonctionnement du microprocesseur MIPS R3000. Les étudiants vont découvrir comment les structures de contrôle des langages de programmation

peuvent être réalisées à l'aide des instructions d'un microprocesseur, et comment le processeur interagit avec son environnement via des périphériques.

Notre objectif consiste à créer une plate-forme de formation qui répond au besoin de l'enseignant et aux apprenants de la manière suivante :

- Se former à distance et à n'importe quel moment
- Transmettre le contenu des cours de façon visuelle
- Simuler les instructions du langage MIPS R3000
- Faire des exercices et des quizz en ligne
- Consulter et télécharger les cours
- Améliorer la qualité de la formation
- Soutenir les étudiants en difficulté

I.3 DESCRIPTION DE L'EXISTANT:

La formation au département d'Informatique et de la Technologie de l'Information se déroule actuellement de façon traditionnelle : cours, apprenants et enseignants sur place.

Le module donné en une heure et demie par semaine de cours magistral et une heure et demie de travaux pratiques par semaine.

Le module se compose de deux parties :

- La première partie présente le microprocesseur MIPS R3000, sa structure interne et la relation avec les autres composants (mémoire, bus)
- La deuxième partie se charge de présenter le langage d'assemblage MIPS.

Le TP est une séance obligatoire qui se déroule par groupe en salles de TP. Les groupes sont composés de 15 à 18 étudiants. Le chargé de TP présente des exercices pédagogiques suivant le cours qui a été donné durant la semaine. Les étudiants sont encouragés à assister à tous les cours et TP parce que la nature du module nécessite une révision continue et beaucoup d'exercices sur le langage d'assemblage.

L'outil utilisé pour réaliser les travaux pratiques est le simulateur QtSpim. Il accepte en entrée des programmes décrits en langage d'assemblage, et permet de visualiser le comportement du processeur, instruction par instruction.

Le simulateur possède les avantages suivants :

- Très faible coût et large disponibilité,
- Simule l'architecture MIPS R3000 qui est simple et claire et donc idéale pour l'enseignement.

- Offre un environnement de programmation convivial et performant.

I.4 CRITIQUES DE L'EXISTANT :

Le volume horaire destiné au module « Architecture des Ordinateurs » est insuffisant pour terminer le programme. Les étudiants doivent fournir plus d'efforts pour pouvoir comprendre les cours surtout la partie du langage d'assemblage. Les enseignants encouragent les étudiants à faire beaucoup d'exercices en travaux pratiques pour pouvoir réaliser les mini-projets, tests et examens de TP en toute aisance. Une heure et demie de TP par semaine ne suffit pas pour résoudre les exercices. Des heures supplémentaires sont nécessaires afin d'arriver à comprendre le module.

I.5 LA SOLUTION PROPOSEE :

L'étude de l'existant nous a montré que l'étudiant a besoin constamment d'être accompagné surtout lorsqu'il n'arrive pas à assimiler les connaissances d'une manière autonome. Pour résoudre ce problème, nous proposons de concevoir et d'implémenter un outil d'apprentissage pour le compte des étudiants en difficulté ou pour ceux désirant approfondir leurs connaissances en matière de microprocesseur et architecture des ordinateurs.

I.6 CONCLUSION :

Dans ce chapitre, nous avons présenté le cadre général de notre projet. Après l'étude de l'existant, nous avons constaté que les étudiants ont besoin d'être accompagnés et assistés dans le module « architecture des ordinateurs ». C'est pour cette raison que nous proposons la réalisation d'une plate-forme capable d'aider les étudiants à comprendre le fonctionnement du microprocesseur et de réaliser les exercices d'accompagnement.

Chapitre

2

E-learning

II. INTRODUCTION

Dans ce chapitre, nous allons présenter le cadre théorique dans lequel s'inscrivent nos travaux. Au début, nous allons définir le domaine du e-learning, ses avantages dans l'enseignement et ses limites. Nous terminons le chapitre par la présentation de quelques plates-formes utilisées dans la formation

II.1 FORMATION A DISTANCE

La notion de « formation à distance » est d'origine canadienne et remonte aux années 1980. Elle intègre deux notions distinctes: celle d'enseignement à distance et la notion d'apprentissage à distance. L'apprentissage dépend d'abord et avant tout de l'apprenant, alors que « l'enseignement » n'implique pas nécessairement que l'apprenant apprenne. [05]

Les établissements sont divisés en deux types, soit unimodal et bimodal. Le type unimodal signifie que l'établissement offre uniquement de la formation à distance. Le type bimodal désigne un établissement qui offre à la fois de la formation à distance et de l'enseignement en mode présentiel [01].

II.2 L'EVOLUTION HISTORIQUE DE LA FORMATION A DISTANCE :

II.2.1 PREMIERE GENERATION:ENSEIGNEMENT PAR CORRESPONDANCE

Le premier cours par correspondance fut créé en 1840 en Angleterre et marque le début de l'enseignement à distance qui s'est largement développé en Europe, puis dans le reste du monde.

Il s'agissait surtout d'un enseignement de « seconde chance » pour les adultes n'ayant pas pu achever leur enseignement secondaire ou supérieur. [13]

Des tuteurs leur apportent une assistance par correspondance (généralement limitée à la correction des travaux), puis parfois aussi par téléphone ; en général, l'interaction est faible et les abandons nombreux.

A partir de 1920, des programmes éducatifs - notamment universitaires - sont radiodiffusés en Europe.

II.2.2 DEUXIEME GENERATION : ENSEIGNEMENT TELEVISE ET MODELE INDUSTRIEL :

En 1970, le Gouvernement espagnol crée l'Universiade National de Éducation a Distance (UNED) et le Gouvernement britannique l'Open University ; dès l'origine, celle-ci encadra ses étudiants par un tutorat personnalisé et fit le meilleur usage de la télévision, puis aussi des bandes vidéo ; elle est l'exemple typique de l'enseignement à distance de la seconde génération : rationalisation, industrialisation, planification, économies d'échelle, avec division des fonctions d'enseignant en différents rôles confiés à des personnes différentes : experts académiques auteurs de cours, pédagogues et techniciens réalisateurs de cours, tuteurs facilitant les progrès des apprenants, examinateurs évaluant leurs connaissances et compétences.

Dans cette deuxième génération de l'enseignement l'imprimé reste le média de base, mais l'audiovisuel joue un rôle pédagogique croissant. L'interaction reste limitée à la correction des travaux par correspondance et parfois le téléphone entre apprenant(s) et tuteur.

II.2.3 TROISIEME GENERATION : ENSEIGNEMENT A DISTANCE INTERACTIF:

L'évolution réalisée grâce au développement des technologies éducatives et l'intégration de l'outil informatique a permis d'introduire un potentiel éducatif illimité par le biais des cours interactifs. En effet, les premiers systèmes d'enseignement assisté par ordinateur sont apparus dans les années 1970. L'objectif était d'abord l'apprentissage en tant que transfert de connaissances. Une multitude de programmes éducatifs furent développés, mais vite délaissés car leur contenu était limité et leur utilisation rigide. L'aspect cognitif a été totalement ignoré avec peu de recherches, de diagnostic et d'adaptation de stratégies. Les connaissances et les décisions étaient préétablies, sans souci du comportement de l'utilisateur. Mais malgré leur application limitée, ces systèmes ont eu des retombées significatives dans le domaine de l'éducation.

De plus, les capacités de raisonnement offertes par l'intelligence artificielle et les systèmes experts ont permis des innovations en introduisant un niveau d'interaction plus élevé entre l'apprenant et le système. C'est ce qui a donné naissance aux systèmes d'enseignement intelligemment assisté par ordinateur (E.I.A.O.) qui pallient aux nombreux inconvénients des systèmes précédents. Les recherches effectuées afin d'adapter l'apprentissage au niveau de l'apprenant et par rapport à son niveau de connaissances a donné lieu à une nouvelle génération de systèmes appelés : Tutoriels Intelligents.




Les systèmes tutoriels intelligents sont des systèmes d'apprentissage un à un (tuteur - apprenant). Ces systèmes ont pour but de reproduire le comportement d'un tuteur intelligent afin

de dispenser un enseignement personnalisé à l'utilisateur. Ces systèmes offrent une possibilité de génération dynamique d'exercices, des adaptations au niveau de difficultés selon les performances de l'étudiant ainsi que l'analyse de l'interprétation du comportement de l'étudiant.

Les moyens de communication et d'interaction entre apprenants et formateurs ont également progressé : chaque apprenant dialogue avec son tuteur et ses pairs par visioconférence, par e-mail et dans des discussions sur le Web ; il exploite aussi les ressources pédagogiques du Net, s'y auto-évalue et devient très autonome. De plus, les technologies éducatives se diversifient : supports imprimés, audiovisuels, multimédias, interaction par téléphone, e-mail, Web, visioconférence

Cette troisième génération de l'enseignement à distance est apparue dans un contexte influencé par les conceptions « constructivistes » de l'enseignement et permet d'exploiter les technologies interactives pour l'apprentissage collaboratif, la reformulation des notions étudiées par l'apprenant, des auto-évaluations anonymes et instantanées, etc.

L'enseignement à distance de la troisième génération est donc caractérisé par :

-  des technologies éducatives interactives,
-  une convergence des modes d'enseignement présentiel et à distance,
-  une concurrence des institutions d'enseignement à distance à l'échelle mondiale.

Les enseignements à distance des différentes générations coexistent dans le monde et même au sein d'une même institution, vu notamment le délai nécessaire à la reconversion des enseignements.

II.3 E-LEARNING

Le e-Learning est un mode d'apprentissage qui tire parti de l'usage des technologies de l'information et de la communication à tous les niveaux de l'activité de formation. Il désigne plus particulièrement un dispositif de formation dont les principaux objectifs peuvent être définis comme l'autonomie d'apprentissage, la formation à distance, l'individualisation des parcours de formation et le développement des relations pédagogiques en ligne.

II.4 DEFINITION

Abréviation de "electronic learning"; que l'on peut traduire par apprentissage ("learning") par des moyens électroniques ("e") .Selon la définition de la Commission Européenne l'elearning est :
 « L'utilisation des nouvelles technologies multimédia et de l'internet pour améliorer la qualité de

«l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que les échanges et la collaboration à distance ». [11]

II.5 PRINCIPE DU E-LEARNING

Le principe étant de pouvoir accéder à ses cours depuis un poste distant (chez soi, depuis son entreprise), les lieux nécessaires au suivi d'un cursus de formation (établissements, classes, bibliothèques) n'existent plus physiquement, ils sont remplacés par le Système de Gestion des Cours ou S.G.C (plate-forme). Le SGC est le cœur du système de formation à distance, c'est lui qui fait le lien entre les apprenants, les cursus, les tuteurs, les ressources et les contenus présents dans le système.

L'apprenant, via cette plate-forme, se verra attribuer un certain nombre de modules de cours, d'exercices, d'évaluations qu'il devra effectuer en tenant compte d'une planification établie. Le tuteur (ou formateur) se charge de gérer les apprenants, affecter des ressources à consulter, des cours à suivre et de la bonne assimilation des contenus proposés.

La communication entre tous ces acteurs se fait via Internet. Il était très difficile de concilier des contenus de qualité, les contraintes liées au bas débit faisaient qu'ils étaient constitués de pages au format HTML ou de document texte. Le haut débit, accessible aujourd'hui à une très grande vitesse, permet de utiliser des formats de contenu plus riches, même les vidéos.

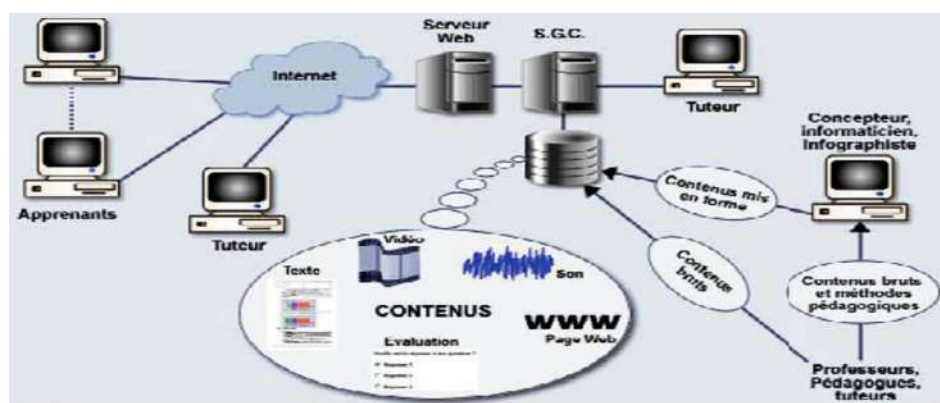


Figure 1 : Acteurs du E-Learning. [13]

II.6 L'EVOLUTION DU E-LEARNING

Le E-Learning a connu un fort développement pendant la période de la révolution Internet. Durant cette époque de très nombreuses sociétés ont développé des plates-formes du e-learning. Depuis maintenant 3 ou 4 ans, le marché s'est consolidé et des leaders commencent à émerger. La consolidation du marché n'est pas terminée, mais la maturité des produits est maintenant

satisfaisante. Un investissement dans ce domaine n'est plus un investissement hasardeux. Un développement sur mesure doit être justifié par une analyse sérieuse des besoins de l'entreprise, car il est peu vraisemblable qu'une plate-forme ne puisse les couvrir.

Au niveau des contenus, un marché de modules de formation commence à émerger, même s'il se limite pour l'instant à des cours relativement généraux. Dans le domaine informatique, l'offre est réellement satisfaisante. Dans le domaine des langues des progrès ont été faits. Divers programmes de financement ont encouragé les Hautes Ecoles à développer leur offre et celle-ci commence à arriver sur le marché. Les Universités américaines proposent déjà de nombreux cours, et certaines Ecoles européennes lancent des offres intéressantes.

La standardisation qui a permis l'interopérabilité des plates-formes se poursuit et s'enrichit. Le respect des standards est maintenant un acquis pour l'industrie, ce qui permet de penser que le marché des modules de cours devrait s'enrichir considérablement dans les prochaines années.

L'évolution de l'informatique multimédia et la généralisation des connexions à haut débit permettent d'envisager maintenant une réelle utilisation du son et de la vidéo, y compris à distance sur Internet. Cette évolution permet d'envisager une plus grande attractivité des contenus des cours proposés et supprime ce qui constituait hier encore une barrière technologique. [13]

II.7 LES DIFFERENTES FORMULES DU E-LEARNING

Lorsque les cours sont donnés grâce à un ordinateur, relié à un Intranet (les cours sont alors limités au personnel de l'entreprise) ou à un Extranet (l'accès est alors élargi à des personnes extérieures).

Trois situations peuvent alors se présenter : le cours magistral synchrone, l'assistance et l'autoformation connectée.

II.7.1 UN COURS MAGISTRAL SYNCHRONE

Un formateur est relié en mode audio-visuel à plusieurs stagiaires (en moyenne quatre à huit) situés dans des lieux géographiques qui peuvent être très différents. L'avantage du E-Learning est alors double.

Pour les stagiaires, il offre la possibilité de se former depuis leur lieu de travail, voir même depuis leur domicile. C'est confortable et cela permet aussi de gagner du temps. Pour l'employeur, l'avantage financier est considérable : pas de coûts liés au déplacement des stagiaires, pas de surcoûts liés au temps perdu.

II.7.2 L'ASSISTANCE

Dans cette phase, les étudiants vont travailler individuellement, mais ils disposent d'une assistance audio-visuelle en direct. Le formateur reste à leur disposition et peut, à distance, prendre le contrôle de leur poste de travail. Il peut ainsi partager les logiciels ou les documents et, le cas échéant, converser avec eux. L'assistance peut également se manifester de manière asynchrone, au moyen de messageries ou de forums informatiques. [11]

II.7.3 L'AUTO-FORMATION CONNECTEE

Les personnes en formation peuvent à partir des ressources pédagogiques ou des parcours de formation disponibles sur le serveur internet ou intranet, suivre un cours magistral enregistré, en audio et en vidéo, sous forme par exemple, de séances de rattrapage.

Ce qui explique, aujourd'hui, le phénomène d'engouement pour le e-learning est la souplesse de cet outil pour l'utilisateur et son faible coût d'utilisation, une fois les investissements de départ réalisés. La personne en formation peut beaucoup plus facilement concilier formation et temps de travail et est pratiquement affranchie des contraintes de lieu et de temps. En outre, elle peut plus facilement apprendre à son rythme et en utilisant la ou les méthodes de son choix.

Mais le problème est l'important coût d'investissement en amont pour mettre en place de tels supports. Difficile de les rentabiliser à court terme. [01]

II.8 LA PLATE-FORME DU E-LEARNING

La plate-forme du e-Learning (appelée également LMS ó *Learning Management System*) est l'élément central d'un système de formation à distance.

Ses principales fonctions sont :

- ✚ La présentation du catalogue de cours
- ✚ L'intégration des modules de cours et des ressources annexes
- ✚ La diffusion des cours à la demande sur les postes des utilisateurs
- ✚ La gestion des activités des tuteurs
- ✚ La gestion des évaluations des apprenants
- ✚ La gestion des processus qualité
- ✚ La construction des parcours de formation personnalisés
- ✚ Le support à la création d'une communauté d'utilisateurs
- ✚ Le reporting pour les ressources humaines
- ✚ La gestion administrative des apprenants incluant les inscriptions et la comptabilité

Une plate-forme peut être acquise, développée sur mesure ou louée. Le marché ayant déjà quelque peu mûri, on peut dire aujourd'hui que l'acquisition d'un produit du marché est moins chère et moins risquée (plus fiable, plus rapide à mettre en œuvre) que le développement sur mesure. Quant à la location de la plate-forme, elle présente l'intérêt d'un investissement faible en début de projet mais impose le stockage des données des apprenants sur un serveur externe.

Des produits hybrides LMS et CMS (*Content Management System*) sont apparus relativement récemment sur le marché. On parle alors de LCMS (*Learning Content Management System*, système de gestion de contenus d'apprentissage). En plus des fonctions classiques d'une plate-forme LMS, ils incluent des possibilités de création de cours e-Learning. Les entreprises peuvent ainsi réaliser de manière autonome à moindre coût des cours sur mesure, puis les intégrer facilement dans leur plate-forme LMS.

Les LCMS n'ont pas encore atteint la maturité des outils e-Learning, mais ce marché est en plein essor. Ainsi, selon le rapport «*Capturing the E-Learning Opportunity, 2002* » publié par *Kinetic Information et Collaborative Strategies*, le marché des systèmes LCMS et LMS atteindra 11,7 milliards de dollars en 2005.

II.9 LES CRITERES DE SUCCES DU E-LEARNING

II.9.1 DANS LA MISE EN PLACE

- **Spécifique au média Internet**

Il ne faut pas se contenter de mettre un cours traditionnel ou un livre en ligne. Les individus n'ont pas nécessairement la patience de lire des pages ou de suivre une vidéo sur un écran. Ils doivent également interagir avec le support. Cette interactivité et la logique de séquencement du parcours de l'apprenant sont mises en place par des ingénieurs pédagogiques.

- **Réutilisable**

Il est souvent possible de réutiliser du matériel développé dans le cadre d'une formation pour d'autres cours (questionnaires, chapitres, etc.). Ceci est essentiel pour assurer un bon retour sur l'investissement.

- **Opérationnel**

Les modules de cours doivent être développés et améliorés afin d'être intuitifs et utilisables par tous les publics.

- **Adaptable**

Les modules de formation doivent pouvoir être modifiés ou corrigés facilement afin de s'adapter aux changements (de procédures, de métiers, etc.).

II.9.2 PENDANT LE FONCTIONNEMENT

- **Accessible et souple**

Les cours e-Learning doivent être accessibles à tout moment. Les apprenants doivent pouvoir choisir les chapitres qui les intéressent et y consacrer le temps qu'ils souhaitent.

- **Feedback**

Les apprenants doivent pouvoir connaître leur niveau par rapport au cours (avant, pendant, après).

- **Personnalisable**

Le système d'apprentissage doit pouvoir s'adapter aux besoins de la plupart des utilisateurs : recherche d'un complément d'information, acquisition d'une expertise, exemples pratiques, l'apprenant doit pouvoir également choisir l'ordre dans lequel il va suivre les chapitres d'une formation.

- **Mesurable**

Il est indispensable de pouvoir mesurer la qualité de l'apprentissage par les individus.

Cela permet de leur proposer une personnalisation plus poussée des formations. Cette caractéristique permet à la fois une amélioration de la qualité des modules de cours proposés et une amélioration de la qualité des formations réalisées (en termes d'apprentissage).

- **Communautaire**

Il est important de créer une communauté autour du E-Learning afin de conserver la motivation des individus et de développer les compétences en continu. Le tuteur et la

II.10 LES AVANTAGES ET LES INCONVENIENTS DU E-LEARNING

II.10.1 LES AVANTAGES :

- ✚ **L'e-Learning touche plus de collaborateurs en moins de temps.**

Lorsque les modules de formation sont en place, le déploiement des cours est plus rapide que dans le cas de cours présentiels, notamment dans les environnements internationaux ainsi que dans les grandes entreprises (formation en masse des salariés).

- ✚ **L'e-Learning minimise le temps passé hors du lieu de travail.**

L'apprentissage individuel sur un poste de travail dans les locaux de l'entreprise réduit les pertes de temps de déplacement. La formation peut être suivie à n'importe quel moment de la journée. Ceci est bien évidemment particulièrement vrai dans le contexte des entreprises multinationales.

✚ **L'apprentissage est plus rapide et plus durable.**

Des études montrent que les courbes d'apprentissage peuvent être améliorées par l'e-learning et que la mémorisation des informations est souvent meilleure. Les temps d'apprentissage sont en général plus faibles qu'en présentiel (environ 50% plus faibles). Bien utilisé, l'e-learning permet donc d'améliorer la qualité de la formation.

✚ **L'apprentissage est personnalisé.**

L'apprenant peut choisir le temps à passer sur chaque module d'une formation en fonction de ses acquis. Il peut adapter le rythme du cours à son niveau (moins de stress, moins de frustration).

✚ **La gestion des certificats est automatisée.**

Les plates-formes d'e-Learning permettent de suivre le parcours des apprenants et de gérer les certifications.

✚ **L'e-Learning coûte globalement moins cher que le présentiel.**

L'e-Learning élimine les coûts liés à la rémunération des professeurs, la gestion des salles de cours, les déplacements des employés, leur hébergement, etc. Les économies réalisées par rapport à la formation traditionnelle sont proportionnelles au nombre d'apprenants. Suivant les cas, les coûts du tutorat se substituent néanmoins partiellement à la rémunération des formateurs traditionnels.

II.10.2 LES INCONVENIENTS

- ✚ **Les problèmes techniques afférents au fonctionnement des systèmes de formation :** perturbation du réseau de communications, pannes des ordinateurs, terminaux ou serveurs, attaques des documents électroniques de cours par des virus ou des pirates, etc.
- ✚ **La mise en place de l'infrastructure technique et la création des contenus sont coûteuses :** un investissement est nécessaire au lancement d'une politique du E-Learning. Cet investissement (coût fixe et risque important) remplace le paiement à la prestation de formation (coût variable et risque faible). Par ailleurs, le marché n'est pas encore consolidé et l'offre en matière de cours spécialisés reste limitée suivant les domaines.
- ✚ **L'accès à l'outil informatique est nécessaire :** l'utilisation de l'outil informatique limite la diffusion du E-Learning auprès d'une partie des collaborateurs.
- ✚ **La diffusion de cours e-Learning nécessite des équipements multimédia :** L'équipement de l'entreprise doit permettre la diffusion des contenus des cours. Cela concerne notamment les postes utilisateurs (PC récents, logiciels installés compatibles, éventuellement réseau avec suffisamment de bande passante). Cet inconvénient tend à perdre de son importance au fil des années, étant donné l'évolution des technologies.

✚ L'e-Learning limite les interactions entre les individus :

Certains mécanismes de communication ne peuvent pas être reproduits (langage du corps par exemple), alors qu'ils jouent un rôle important dans la diffusion du savoir [11].

II.11 COMPARAISON DU E-LEARNING ET DE LA FORMATION TRADITIONNELLE

| | Avantages | Inconvénients |
|-------------------|---|--|
| Présentiel | Feedback immédiat Repères visuels Facilité d'interaction avec les autres apprenants Favorise les personnes communiquant facilement | Défavorise les personnes timides ou analytiques Contrôlé par le professeur Cadré dans le temps |
| e-Learning | Tout le monde peut contribuer Contrôlé par l'apprenant Peu cadré dans le temps Enregistrement permanent des échanges | Manque de repères visuels Obstacles technologiques Favorise les personnes communicant facilement par écrit |

Tableau 1: Analyse comparative e-learning et formation traditionnelle [14].

II.12 CONCLUSION

Nous avons abordé dans cette partie les différents aspects du E-learning. Bien que l'objet de notre étude ne réside pas dans le E-learning, nous allons dans le prochain chapitre présenter le microprocesseur MIPS R3000. Cela permet de comprendre l'architecture de ce processeur.

Chapitre

3

Le processeur Mips R3000

III. INTRODUCTION

Au cours de ce chapitre, nous allons présenter le microprocesseur Mips R3000. Dans un premier temps, nous allons présenter son architecture interne et ensuite nous donnerons un aperçu sur le langage d'assemblage correspondant.

III.1 NOTION DU PROCESSEUR :

Annoncé en 1988 par la société *MIPS Computer Systems*, le MIPS R3000 fut le premier processeur à vocation industrielle. Il succède au MIPS R2000, dérivé des travaux de l'Université de Stanford (1985). L'une des principales caractéristiques de ce processeur est sa simplicité. En effet, la force des processeurs MIPS réside dans la relation très étroite entre l'architecture du processeur et le compilateur associé.

Le MIPS R3000 est la deuxième génération de processeur de la société MIPS, et n'est plus commercialisé aujourd'hui. Son successeur est le R4000 qui lui ressemble fort. Des informations sur cette famille de processeurs, très utilisée dans les applications embarquées et dans les machines parallèle à mémoire partagées.

Un microprocesseur est un circuit intégré complexe qui permet l'interprétation et l'exécution des instructions d'un programme.

- ✚ Le processeur exécute chaque instruction très rapidement, en quelques cycles d'horloges.
- ✚ La fréquence de cette horloge s'exprime en MHz (millions de cycles par seconde) ou GHz (milliards de cycles par secondes). Par exemple, un processeur Intel Core 2 possède une horloge de 2,26 GHz.
- ✚ C'est le cerveau de l'ordinateur.
- ✚ A l'heure actuelle, la puissance des processeurs continue de s'accroître et leur taille diminue régulièrement



Figure 2 : Exemple du Processeur. [25]

III.2 ARCHITECTURE DE BASE D'UN MICROPROCESSEUR

Un microprocesseur est construit autour de deux unités principales:

III.2.1 UNITE DE COMMANDE:

É Permet de contrôler le déroulement des instructions.

É Effectue la recherche en mémoire de l'instruction.

É Assure le décodage de l'instruction puis effectue la préparation de l'instruction suivante.

III.2.2 UNITE ARITHMETIQUE ET LOGIQUE:

É Regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions.

É Assure les opérations logiques (ET, OU, Comparaison, Décalage, etcí) ou arithmétiques (Addition, soustractioní)

III.3 PRESENTATION DU PROCESSEUR MIPS R3000 :

MIPS (de l'anglais Micro processor without inter locked pipeline stages) a été développée par la compagnie MIPS Computer Systems Inc., basée en Californie.

Le processeur **MIPS R3000** est un processeur 32 bits industriel conçu dans les années 80. Son jeu d'instructions est de type RISC. Il existe plusieurs réalisations industrielles de cette architecture (SIEMENS, NEC, LSI LOGIC, SILICON GRAPHICS, etc...)

Cette architecture est suffisamment simple pour présenter les principes de base de l'architecture des processeurs, et suffisamment puissante pour supporter un système d'exploitation multi tâches tel qu'UNIX. [27]

III.4 L'ARCHITECTURE MIPS

L'architecture MIPS (de l'anglais : micro processor with out inter locked pipeline stages) est une architecture de processeur de type Reduced instruction set computer (RISC) développée par la société MIPS Technologies (alors appelée MIPS Computer Systems), basée à Mountain View en Californie. Les processeurs fabriqués selon cette architecture ont surtout été utilisés dans les systèmes SGI. On les retrouve aussi dans plusieurs systèmes embarqués, comme les ordinateurs de poche, les routeurs Cisco et les consoles de jeux vidéo (Nintendo 64 et Sony PlayStation, PlayStation 2 et PSP).

Vers la fin des années 1990, on estimait que les processeurs dérivés de l'architecture MIPS occupaient le tiers des processeurs RISC produits.

De nos jours cette architecture reste populaire dans le marché de l'informatique embarquée où elle subit une intense concurrence de la part de l'architecture ARM.

Les MIPS sont aussi présents dans des routeurs ou des NAS, mais ils deviennent aussi de plus en plus rares dans ce domaine face à la concurrence des ARM, PowerPC et x86 basse de consommation.

Malgré tout, les MIPS reviennent peu à peu dans le marché de l'informatique à haute performance grâce aux recherches menées par la Chine avec les processeurs Loongson qui ont été utilisés pour la création du supercalculateur Dawning 5000L et 6000. [12]

III.4.1 ARCHITECTURE INTERNE

L'architecture interne du processeur se décompose en une partie opérative et une partie contrôle. La partie opérative (PO) contient les registres et les opérateurs. Sa structure est présentée au chapitre D. Elle réalise des transferts élémentaires de données entre un ou plusieurs registres sources et un registre destination. Un transfert élémentaire est exécuté en un cycle. La partie opérative est commandée par la partie contrôle (PC). La partie contrôle est chargée de définir, pour chaque cycle d'horloge, les transferts élémentaires qui doivent être réalisés par la partie opérative.

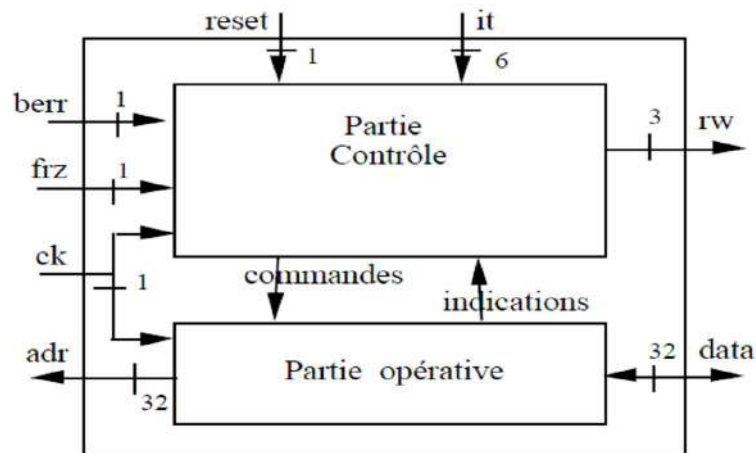


Figure 3 : MIPS R3000 Architecture interne [12].

III.4.2 CHEMIN DE DONNEES

En plus des registres visibles du logiciel, le processeur possède 3 registres internes :

ÉIR Registre instruction qui mémorise l'instruction lue en mémoire.

ÉDT Registre data qui reçoit la donnée provenant de la mémoire

AD Registre d'adresse qui peut être utilisé comme registre temporaire. Les registres SR et CR sont implantés dans la partie contrôle. Le processeur possède deux bus opérands X et Y et une seule "boîte à opérations" capable d'effectuer des opérations logiques, arithmétiques, mais aussi des opérations de comparaison, ou des décalages multi-bits. Toutes les opérations s'exécutent en un cycle, sauf les opérations de multiplication et de division qui s'exécutent en deux cycles. Seuls les registres PC et AD permettent d'adresser la mémoire externe. Un mot de 32 bits provenant de la mémoire peut être écrit dans les registres DT ou IR.

Un mot de 32 bits provenant de la boîte à opérations peut être écrit dans les registres PC, EPC, SR, CR, BAR, HI, LO, AD, ou dans un registre général R(n). Ceci signifie que cette partie opérative est capable d'effectuer deux transferts élémentaires à chaque cycle : un transfert interne (qui utilise deux registres sources, la boîte à opérations et un registre résultat), et un transfert externe (lecture d'un mot en mémoire externe et écriture dans IR ou DT). Le banc de registre R(n) ne possède que 2 accès simultanés (un en lecture et un en écriture). Des pseudo-registres permettent de forcer des valeurs constantes sur les bus X et Y.

Les 32 bits du registre d'instruction IR sont envoyés à la partie contrôle pour lui permettre de prendre des décisions. Par ailleurs, la partie opérative envoie vers la partie contrôle le signal NUL qui vaut 1 quand RES = \emptyset . Elle renvoie également vers la partie contrôle les bits de signe des deux opérandes X et Y, ainsi que celui du résultat, la retenue de l'alu et les deux bits de poids faibles du résultat. Ces signaux, calculés par la micro-instruction i peuvent être utilisés par la partie contrôle pour choisir la micro-instruction $i+1$.

III.4.3 REGISTRES VISIBLES

Tous les registres visibles du logiciel, c'est à dire ceux dont la valeur peut être lue ou modifiée par les instructions, sont des registres 32 bits.

Afin de mettre en œuvre les mécanismes de protection nécessaires pour un système d'exploitation multi-tâches, le processeur possède deux modes de fonctionnement : Utilisateur/superviseur. Ces deux modes de fonctionnement imposent d'avoir deux catégories de registres.

- **Mode Utilisateur/Superviseur:**

Afin de mettre en œuvre les mécanismes de protection nécessaires pour un système, le processeur possède deux modes de fonctionnement :

- *Le mode superviseur:* permet une plus grande sécurité pour le système. Quand le processeur est en mode superviseur, l'utilisateur ne peut pas accéder à certaines zones (ex: mémoire, registres...)
- *Le mode utilisateur:* Ce mode permet à l'utilisateur d'accéder uniquement aux zones réservées aux utilisateurs. Le processeur part en exception si l'utilisateur accède à une zone privilégiée.

Quand le processeur est en mode superviseur, le système peut accéder aux zones protégées et non protégées.

- **Registres non protégés** Le processeur possède 35 registres manipulés par l'instruction standard (c'est à dire les instructions qui peuvent s'exécuter aussi bien en mode utilisateur qu'en mode superviseur).

✚ **Ri** (\$0 à \$ 31) 32 registres généraux Ces registres sont directement adressés par les instructions, et permettent de stocker des résultats de calculs intermédiaires.

✚ Le registre **R0** est un registre particulier:

- la lecture fournit la valeur constante "0x00000000"
- l'écriture ne modifie pas son contenu.

- ✚ Le registre **R31** est utilisé par les instructions d'appel de procédures pour sauvegarder l'adresse de retour.
- ✚ **PC** Registre compteur de programme (Program Counter) Ce registre contient l'adresse de l'instruction en cours d'exécution. Sa valeur est modifiée par toutes les instructions.
- ✚ **HI et LO** Registres pour la multiplication ou la division Ces deux registres 32 bits sont utilisés pour stocker le résultat d'une multiplication ou d'une division, qui est un mot de 64 bits. Contrairement à d'autres processeurs plus anciens, le processeur R3000 ne possède pas de registres particuliers pour stocker les "codes conditions". Des instructions de comparaison permettent de calculer un booléen qui est stocké dans l'un quelconque des registres généraux. La valeur de ce booléen peut ultérieurement être testée par les instructions de branchement conditionnel.

| Name | Register Number | Usage | Preserve on call? |
|-------------|-----------------|------------------------|-------------------|
| \$zero | 0 | constant 0 (hardware) | n.a. |
| \$at | 1 | reserved for assembler | n.a. |
| \$v0 - \$v1 | 2-3 | returned values | No |
| \$a0 - \$a3 | 4-7 | arguments | Yes |
| \$t0 - \$t7 | 8-15 | Temporaries | No |
| \$s0 - \$s7 | 16-23 | saved values | Yes |
| \$t8 - \$t9 | 24-25 | Temporaries | No |
| \$gp | 28 | global pointer | Yes |
| \$sp | 29 | stack pointer | Yes |
| \$fp | 30 | frame pointer | Yes |
| \$ra | 31 | return addr (hardware) | yes |

Tableau 2: Correspondance entre les noms des registres et leurs numéros.[12]

- **Registres protégés**

L'architecture **MIPS** définit 32 registres (numérotés de 0 à 31), qui ne sont accessibles, en lecture comme en écriture, que par les instructions privilégiées (c'est à dire les instructions qui ne peuvent être exécutées qu'en mode superviseur). On dit qu'ils appartiennent au "coprocesseur système". En pratique, cette version du processeur **MIPS R3000** en utilise 4 pour la gestion des interruptions et des exceptions.

- ✚ **SR** Registre d'état (Status Register) : Il contient en particulier le bit qui définit le mode : superviseur ou utilisateur, ainsi que les bits de masquage des interruptions.
- ✚ **CR** Registre de cause (Cause Register) : en cas d'interruption ou d'exception, son contenu définit la cause pour laquelle on fait appel au programme de traitement des interruptions et des exceptions.
- ✚ **EPC** Registre d'exception (Exception Program Counter) : Il contient l'adresse de retour (PC + 4) en cas d'interruption. Il contient l'adresse de l'instruction fautive en cas d'exception (PC).
- ✚ **BAR** Registre d'adresse illégale (Bad Address Register) : En cas d'exception de type "adresse illégale", il contient la valeur de l'adresse mal formée.

II.5 CO-PROCESSEURS DE MIPS:

Le microprocesseur Mips travaille avec des coprocesseurs pour effectuer des opérations spécifiques.

- ✚ Le co-processeur 0: (processeur de contrôle système) gère le système de mémoire virtuelle, les exceptions ainsi que les transitions entre les modes Superviseur et Utilisateur.
- ✚ Le co-processeur FPU: (Floating Point Unit) est chargé d'effectuer les opérations arithmétiques à virgules flottantes il aussi possède 32 registres nommés: \$f0 ó \$f31 .

III.6 LANGAGE D'ASSEMBLAGE DU MIPS R3000

Ecrire un programme directement en langage machine (0 et 1) est une tâche très difficile.

- Déchiffrer la signification d'instructions codées numériquement est fatigant pour les humains.
- Au lieu d'utiliser le langage machine, on peut programmer avec le langage d'assemblage.
- Le langage d'assemblage est la représentation symbolique du code binaire des instructions.
- Il est proche du langage machine et lisible par les humains

III.6.1 PROGRAMMATION EN LANGAGE D'ASSEMBLAGE:

Les programmes assembleur source qui respectent les règles définies par l'assembleur MIPS de l'environnement GNU pour générer du code exécutable. Ils sont également acceptés par le

simulateur du MIPS R3000 utilisé en TP qui permet de visualiser le comportement du processeur instruction par instruction.

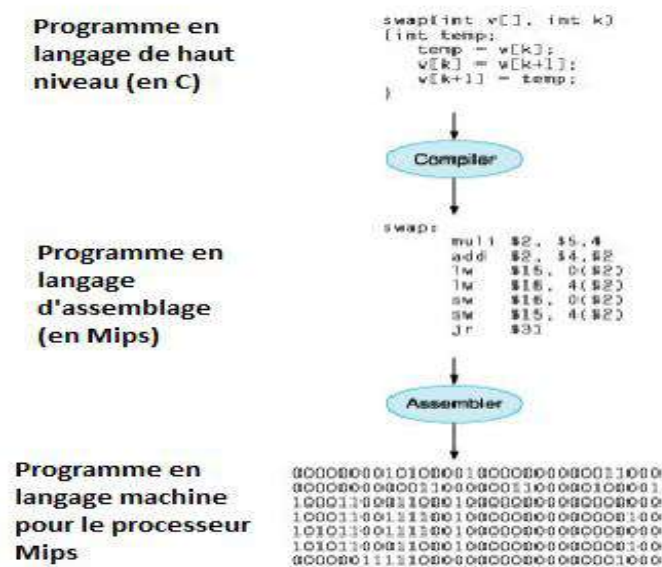


Figure 4 : Programmation en langage d'assemblage

III.6.2 ORGANISATION DE LA MEMOIRE

Rappelons que le but d'un programme X écrit en langage d'assemblage est de fournir à un programme particulier (appelé « assembleur ») les directives nécessaires pour générer le code binaire représentant les instructions et les données qui devront être chargées en mémoire pour permettre au programme X de s'exécuter sur du matériel.

Dans l'architecture MIPS R3000, l'espace adressable est divisé en deux segments : un segment utilisateur, et un segment noyau.

Un programme utilisateur utilise généralement trois sous-segments (appelés sections) dans le segment utilisateur :

- la section **text** contient le code exécutable en mode utilisateur. Elle est implantée conventionnellement à l'adresse **0x00400000**. Sa taille est fixe et calculée lors de l'assemblage. La principale tâche de l'assembleur consiste à générer le code binaire correspondant au programme source décrit en langage d'assemblage, qui sera chargé dans cette section ;

- la section **data** contient les données globales manipulées par le programme utilisateur.
- Elle est implantée conventionnellement à l'adresse **0x10000000**. Sa taille est fixe et calculée lors de l'assemblage. Les valeurs contenue dans cette section peuvent être initialisées grace a des directives contenues dans le programme source en langage d'assemblage ;
- la section **stack** contient la pile d'exécution du programme. Sa taille varie au cours de l'exécution. Elle est implantée conventionnellement à l'adresse **0x7FFFFFFF**. Contrairement aux sections **data** et **text**, la pile s'étend vers les adresses décroissantes.
- Deux autres sections sont définies dans le segment noyau :
- la section **ktext** contient le code exécutable en mode noyau. Elle est implantée conventionnellement à l'adresse **0x80000000**. Sa taille est fixe et calculée lors de l'assemblage ;
- la section **kdata** contient les données globales manipulées par le système d'exploitation en mode noyau. Elle est implantée conventionnellement à l'adresse **0xC0000000**. Sa taille est fixe et calculée lors de l'assemblage ;
- la section **kstack** contient la pile d'exécution du programme. Sa taille varie au cours de l'exécution. Elle est implantée conventionnellement à l'adresse **0xFFFFE000**. Contrairement aux sections **data** et **text**, la pile s'étend vers les adresses décroissantes.

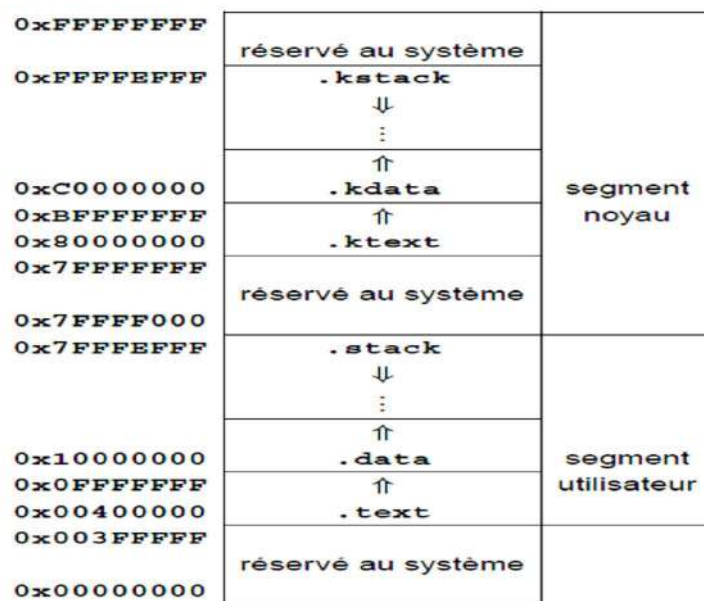


Figure 5 : Organisation de la mémoire [12].

III.6.3 INSTRUCTIONS

Dans ce qui suit, le registre noté \$r est le registre destination, c.-à-d. qui reçoit le résultat de l'opération, les registres notés \$ri et \$rj sont les registres source qui contiennent les valeurs sur lesquelles s'effectue l'opération. Notons qu'un registre source peut être le registre destination d'une même instruction assembleur. Un opérande immédiat sera noté imm, et sa taille sera spécifiée dans la description de l'instruction. Les instructions de saut prennent comme argument une étiquette, où label, qui est utilisée pour calculer l'adresse de saut. Toutes les instructions modifient un registre non accessible du logiciel, le program counter. De même, le résultat d'une multiplication ou d'une division est mis dans deux registres spéciaux, \$hi pour les poids forts, et \$lo pour les poids faibles.

Ce tableau montre quelques instructions du langage MIPS :

| type | | instruction | commentaire | |
|---------------|--------------|----------------------------------|--|---|
| arithmétiques | add | Rd, Rs, RL | addition (avec débordement) | $Rd \leftarrow Rs + RL$ |
| | addu | Rd, Rs, RL | addition (sans débordement) | $Rd \leftarrow Rs + RL$ |
| | addi | Rd, Rs, Imm | addition d'un immédiat (débordt) | $Rd \leftarrow Rs + Imm$ |
| | addiu | Rd, Rs, Imm | addition d'un immédiat (ss débordt) | $Rd \leftarrow Rs + Imm$ |
| | sub | Rd, Rs, RL | soustraction (débordt) | $Rd \leftarrow Rs - RL$ |
| | mult | Rs, RL | multiplication | $hi, lo \leftarrow Rs \times RL$ |
| | multu | Rs, RL | multiplication non signée | $hi, lo \leftarrow Rs \times RL$ |
| | mul | Rd, Rs1, Rs2 | multiplication (ss débordement) | $Rd \leftarrow Rs1 \times Rs2$ |
| | div | Rs, RL | division (débordt) | $lo \leftarrow Rs \div RL ; hi \leftarrow Rs$ |
| | divu | Rd, Rs1, Rs2 | division (débordt) | $Rd \leftarrow Rs1 \div Rs2$ |
| divu | Rd, Rs1, Rs2 | division non signée (ss débordt) | $Rd \leftarrow Rs1 \div Rs2$ | |
| rem | Rd, Rs1, Rs2 | reste de la division | $Rd \leftarrow Rs1 \div Rs2 - \text{si } Rs1 \text{ ou } Rs2 \text{ est négatif, le reste dépend de l'architecture}$ | |
| logiques | abs | Rd, Rs, RL | valeur absolue | |
| | neg | Rd, Rs | opposé d'une valeur (débordt) | $Rd \leftarrow \neg Rs$ |
| | rol | Rd, Rs1, Rs2 | rotation vers la gauche | |
| | ror | Rd, Rs1, Rs2 | rotation vers la droite | |
| | and | Rd, Rs, RL | ET logique | $Rd \leftarrow Rs \wedge RL$ |
| | andi | Rd, Rs, Imm | ET logique immédiat | $Rd \leftarrow Rs \wedge Imm$ |
| | or | Rd, Rs, RL | OU logique | $Rd \leftarrow Rs \vee RL$ |
| | ori | Rd, Rs, Imm | OU logique immédiat | $Rd \leftarrow Rs \vee Imm$ |
| | nor | Rd, Rs, RL | NOR logique | $Rd \leftarrow \neg (Rs \vee RL)$ |
| | xor | Rd, Rs, RL | XOR logique | $Rd \leftarrow Rs \oplus RL$ |
| xori | Rd, Rs, Imm | XOR logique immédiat | $Rd \leftarrow Rs \oplus Imm$ | |
| not | Rd, Rs | NON logique | $Rd \leftarrow \neg Rs ; \text{négation bit à bit}$ | |
| sll | Rd, Rs, Sa | décalage logique à gauche | | |
| sllv | Rd, Rt, Rs | décalage logique à gauche | rang du décalage défini par Rs | |
| srl | Rd, Rs, Sa | décalage logique à droite | | |
| srlv | Rd, Rt, Rs | décalage logique à droite | rang du décalage défini par Rs | |
| sra | Rd, Rs, Sa | décalage arithmétique à droite | | |
| srav | Rd, Rt, Rs | décalage arithmétique à droite | rang du décalage défini par Rs | |

Tableau 3 : Instruction du langage MIPS[12]

III.6.4 MACRO-INSTRUCTIONS

Une macro-instruction est une pseudo-instruction qui ne fait pas partie du jeu d'instructions machine, mais qui est acceptée par l'assembleur qui la traduit en une séquence d'instructions machine. Les macro-instructions utilisent le registre \$1 si elles ont besoin de faire un calcul intermédiaire. Il faut donc éviter d'utiliser ce registre dans les programmes.

III.6.5 LES COMMENTAIRES:

Les commentaires permettent de donner plus d'explication sur le code. Ils commencent par un # ou un ; et s'achèvent à la fin de la ligne courante.

Exemple :

```
# Ceci est un commentaire
; Ceci est un commentaire
```

III.6.6 DECLARATION DE DONNEES:

Les données (constantes et variables) doivent être déclarées dans « .Data » section.

Les données doivent commencer par une lettre suivie des lettres, chiffres ou caractères spéciaux.

Le format général de la déclaration d'une donnée est:

```
<variableName>:      .<dataType>      <initialValue>
```

Figure 6: Exemple d'une déclaration de donnée- 1- [22].

Exemple:

```
.data
c      .byte      'a'      ; octet
n1     .half      26       ; 2 octets
n2     .word      353      ; 4 octets
tab    .space     40
```

Figure 7 : Exemple d'une déclaration de donnée- 2- [22].

Les données en Mips sont de différents types:

| Declaration | |
|-------------------------|-----------------------------------|
| .byte | 8-bit variable(s) |
| .half | 16-bit variable(s) |
| .word | 32-bit variable(s) |
| .ascii | ASCII string |
| .asciiz | NULL terminated ASCII string |
| .float | 32 bit IEEE floating point number |
| .double | 64 bit IEEE floating point number |
| .space <n> | <n> bytes of uninitialized memory |

Tableau 4 : *Les différents types des données en Mips [22]*

III.7 CONCLUSION :

Dans ce chapitre nous avons présenté la notion du processeur MIPS R3000. L'architecture interne et externe qui représente ce que doit connaître un programmeur souhaitant programmer en assembleur, ou la personne souhaitant écrire un compilateur pour ce processeur.

Chapitre

4

Conception

IV. INTRODUCTION

Dans cette partie, nous allons concevoir notre plate-forme. Au début, nous spécifions les besoins et mettons en relief les différents diagrammes de conception en utilisant la méthode UML.

IV.1 SPECIFICATION DES BESOINS:

Dans cette partie, nous allons identifier une liste d'exigences fonctionnelles de l'outil à concevoir afin de clarifier les besoins de l'utilisateur.

La présente plate-forme doit satisfaire les besoins fonctionnels suivants :

- La gestion de la plate-forme
- La gestion des cours
- La gestion des quizz
- La gestion du microprocesseur (ajouter animation pour les instructions du langage MIPSí)

IV.2 LES ACTEURS DU SYSTEME:

Un acteur est une représentation des entités qui interagissent directement avec le système.

Dans notre outil, les acteurs sont :

- **L'enseignant** : c'est le responsable de l'administration du site. Il anime les formations et charge les cours et les exercices.
- **L'apprenant** : assiste à une formation.

IV.3 IDENTIFICATION DES CAS D'UTILISATION :

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. Chaque cas d'utilisation contient un ou plusieurs scénarios qui

définissent comment le système devrait interagir avec les utilisateurs (appelés acteurs) pour atteindre un but ou une fonction spécifique du travail.

| | |
|------------|--|
| Enseignant | <ul style="list-style-type: none"> • Gérer la plate-forme • Gérer l'animation des instructions du microprocesseur (simulateur) • Gérer les cours (déposer, supprimer, modifier, partagerí) • Gérer les exercices (ajouter, supprimer, répondre aux questionsí) |
| Apprenant | <ul style="list-style-type: none"> • Télécharger cours • Utiliser le simulateur • Répondre aux quizz |

Tableau 5: Identification des cas d'utilisation

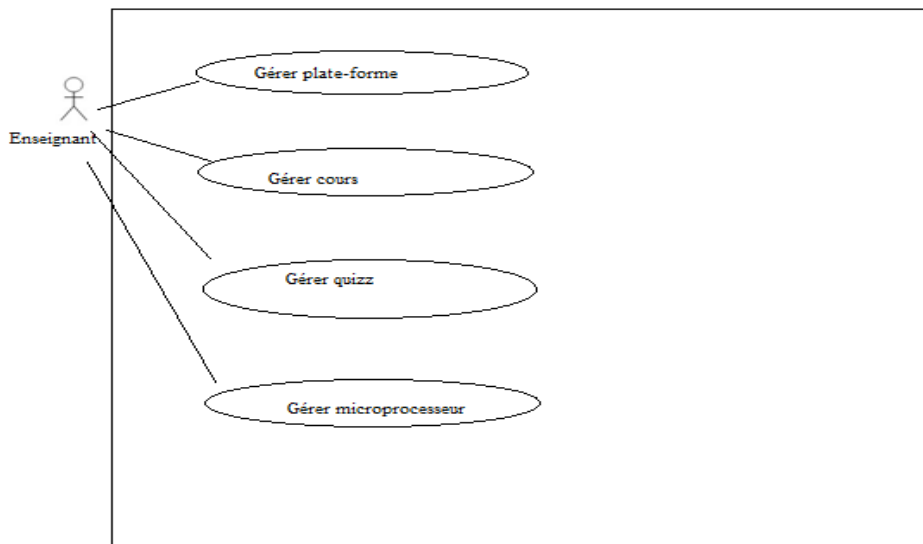


Figure 8 : Diagramme de cas d'utilisation « Enseignant »

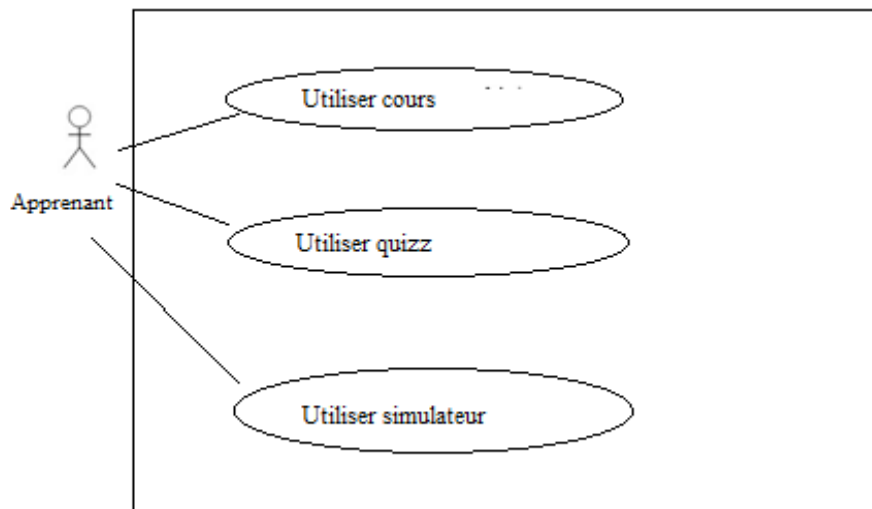


Figure 9 : Diagramme de cas d'utilisation « apprenant »

Dans la partie suivante, nous allons détailler le différent scénario des cas d'utilisation de l'enseignant.

- **Cas d'utilisation « Gérer plate-forme » :**

| |
|---|
| Cas d'utilisation : « gérer plate-forme » |
| Résumé : ce cas permet à l'enseignant de modifier les composants internes de la plateforme |
| Scénario : <ol style="list-style-type: none"> 1. L'enseignant met à jour la plate-forme 2. Le système enregistre les mises à jour |

- **Cas d'utilisation « Gérer les cours » :**

| |
|---|
| Cas d'utilisation : « gérer les cours » |
| Résumé : permet à l'enseignant d'ajouter, de modifier et de supprimer un cours |
| Scénario : <ol style="list-style-type: none"> 1. L'enseignant charge le cours 2. L'enseignant enregistre le cours dans la base de données |

• **Cas d'utilisation « Gérer les quizz » :**

| |
|--|
| Cas d'utilisation : « gérer les quizz » |
| Résumé : permet à l'enseignant d'ajouter, de modifier et de supprimer un quizz |
| Scénario : <ol style="list-style-type: none"> 1. L'enseignant saisit les questions en indiquant la vraie réponse 2. L'enseignant enregistre le quizz dans la base de données |

• **Cas d'utilisation « Gérer le simulateur » :**

| |
|---|
| Cas d'utilisation : « gérer le simulateur » |
| Résumé : permet à l'enseignant d'ajouter, de modifier et de supprimer une instruction du langage MIPS |
| Scénario : <ol style="list-style-type: none"> 1. L'enseignant ajoute les instructions 2. L'enseignant crée l'animation correspondante sur l'outil graphique |

IV.4 DIAGRAMMES D'ACTIVITES :

Le diagramme d'activité est utilisé pour identifier le cheminement des flots de contrôle et des flots de données. Il permet ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

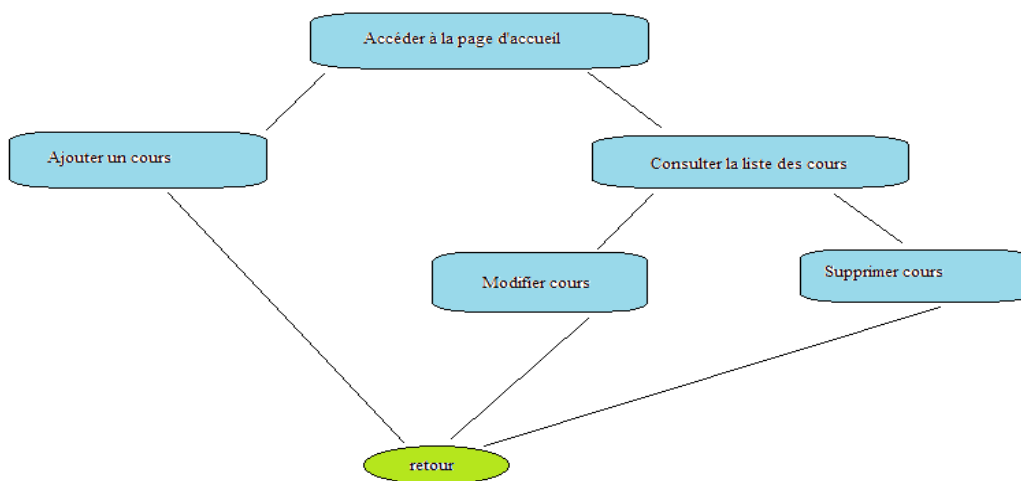


Figure 10 : Diagramme d'activité « Ajouter cours »

La figure ci-dessus illustre le déroulement séquentiel de la gestion des cours accomplis par l'enseignant. Après l'accès à la plate-forme, l'enseignant peut ajouter ou consulter un cours. Au cas de consultation, il peut modifier le cours ou le supprimer de la base de données.

IV.5 CONCLUSION

Dans ce chapitre, nous avons présenté une vue conceptuelle de la plate-forme à réaliser. Nous avons pu réaliser les diagrammes des cas d'utilisation et de quelques diagrammes d'activités. Nous pouvons entamer la phase suivante qui est la phase de la réalisation de la solution.

Chapitre

5

Implémentation

V. INTRODUCTION :

Dans ce chapitre, nous allons exposer la phase de la réalisation. Nous allons présenter l'environnement du travail ainsi que les outils utilisés pour la réalisation de la plate-forme. Nous terminerons le chapitre par quelques résultats.

V.1 CHOIX DE L'ENVIRONNEMENT (L'EDITEUR):

Le langage HTML est un langage fondé pour effectuer des conceptions différentes. Il a été conçu initialement pour faciliter le traitement des pages web et des interfaces de cours...etc dont il est maintenant utilisé dans tous les domaines des sciences qui nécessitent des interfaces interactives.

Pour le choix de l'éditeur du langage HTML pour notre interface, nous avons opté pour le l'éditeur BlueGriffon et cela pour de nombreuses raisons :

- ✚ Est un logiciel libre d'édition de pages web WYSIWYG, successeur de Mozilla Composer et Nvu. C'est un logiciel libre multiplateforme sous triple licence MPL/GPL/LGPL développé par Daniel Glazman et fondé sur la plateforme Mozilla.
- ✚ Il propose de nombreuses interfaces visuelles permettant d'utiliser les propriétés CSS3 sans en connaître la syntaxe précise, ce qui en fait un excellent compagnon pour le développeur.
- ✚ Son moteur de rendu supporte la plupart des éléments HTML5 et les propriétés CSS3 compatibles avec Mozilla Firefox, même si l'éditeur génère également des règles de style pour les autres navigateurs web.
- ✚ Il se démarque notamment par son support du CSS3 et du HTML5
- ✚ À la manière de Firefox, BlueGriffon permet l'utilisation d'extensions. Celles-ci répondent à des besoins plus spécifiques.
- ✚ BlueGriffon a pris en compte les enjeux de l'Accessibilité du web puisqu'il propose notamment l'ajout de rôles ARIA (Accessible Rich Internet

Applications) aux différents éléments présents dans une page web, l'ajout d'un attribut « longdesc » (description longue) pour les images, etc.

- Il intègre également l'éditeur SVG-edit permettant de créer ou modifier des images vectorielles au format SVG et de les embarquer directement dans les pages web créées.

V.2 IMPLEMENTATION DE L'APPLICATION (INTERFACES):

Nous avons implémenté plusieurs interfaces englobées dans un seul site dont chaque interface (page) peu contenir un cours du processeur. Les figures ci-dessous illustrent quelques interfaces de notre application de cours des processeurs MIPS3000 :

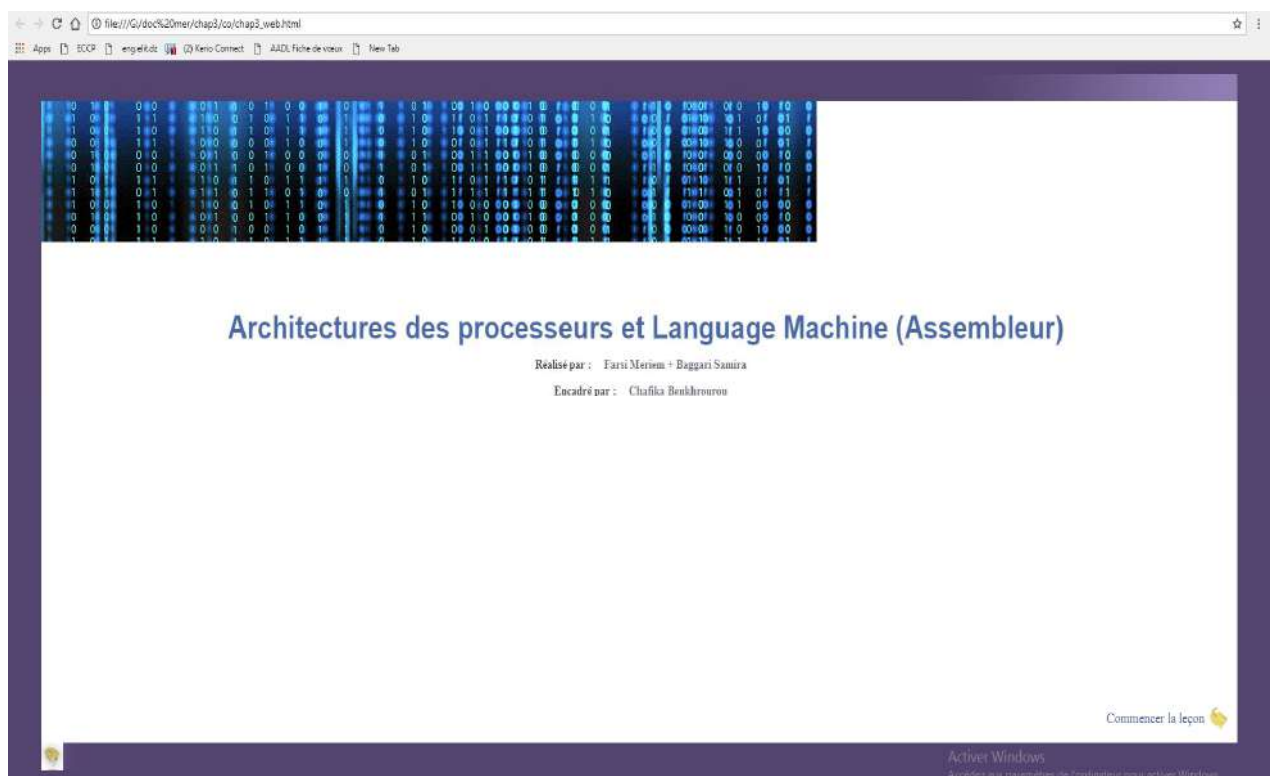


Figure 11 : interface de notre interface de cours

1. l'interface principale contient le grand titre du sujet qui est : « l'architecture des processeurs et le langage Assembleur ».
2. Fait la référence des réalisateurs de cette mémoire plus que le nom de l'encadreur.
3. On peut commencer par parcourir le cours par le lien au dessous de la page à droite.

The screenshot shows a web browser window with the address bar displaying a file path. The page title is 'Architectures des processeurs et Langage Machine'. The main content area is titled 'Introduction et historique' and contains several paragraphs of text. A navigation menu is visible on the left side of the page. At the bottom of the page, there is a small image of a CPU chip and a 'Activer Windows' watermark.

Introduction et historique

Le processeur ou microprocesseur est aussi appelé CPU *Central Processing Unit* l'unité centrale de traitement. C'est le "cerveau" de l'ordinateur, il interprète et exécute les programmes. Il fournit au système d'exploitation des dispositifs tels que le mécanisme d'interruptions ou des registres spécialisés pour l'adressage et la gestion de la mémoire virtuelle. C'est aussi l'élément qui, comparé aux autres composants électroniques du PC, est le plus coûteux et dont les performances évoluent le plus.

La description de l'architecture interne des CPU, son principe de fonctionnement ainsi que les techniques utilisées pour améliorer l'architecture interne sont des sujets qui ont été abordés dans le cours de technologie des ordinateurs. Nous parlerons plus spécialement dans ce chapitre des boîtiers, des sockets, des différentes générations de processeurs et de leur nomenclature.

Le premier microprocesseur, le 4004 a été fabriqué en 1971. Ses successeurs le 8080 de Intel, le 6800de Motorola, le Z80 de Zilog puis tous les autres ont ouvert l'aire de ce qu'on appelle la micro-informatique. Les fabricants de microprocesseurs et d'ordinateurs étaient relativement nombreux et à cette époque ceux qui utilisaient ces ordinateurs étaient des passionnés d'électronique et de programmation. Cette micro-informatique ne fut réellement prise au sérieux et passa dans le monde professionnel que quand en 1981, lorsqu'IBM développa son premier "Personal Computer" à partir du 8088 d'Intel.

Le boîtier du CPU :

Il n'est que l'apparence externe, l'habillage (*packaging*) du processeur. Le boîtier abrite la puce extrêmement fine et fragile et dont la surface est actuellement de l'ordre de 1 cm². Hormis de rares exceptions, la puce n'est jamais directement placée sur la carte mère. Elle est protégée par un boîtier qui la protège, dissipe la chaleur et fournit le brochage dont la disposition est standardisée.

Les processeurs ont évolué ne fut-ce qu'en apparence. Ils ont toujours plus de contacts pour s'adapter au nombre croissant de signaux que le processeur échange avec son entourage. Le dispositif pour le refroidir, inutile sur les premiers processeurs, est devenu de plus en plus encombrant et même bruyant.

Les microprocesseurs qui équipaient les premiers PC, étaient montés dans des boîtiers DIP (*Dual In-line Package*) de 40 broches. Ils avaient l'allure de circuits intégrés classiques avec une rangée de contacts de part et d'autre du boîtier. Quand le nombre de contacts a augmenté, la forme des boîtiers est devenue carrée et les constructeurs ont placé les contacts dans une disposition quadrillée. Cela a donné les connecteurs dits PGA pour *Pin Grid Array* (matrice à grille de broches). Il a finalement dû être nécessaire de placer les contacts en quinconce pour pouvoir en disposer plus sur une même surface. On parle alors de SPGA pour *Staggered Pin Grid Array* (matrice à grille de contacts en quinconce).

Les connecteurs sont marqués par un numéro ou un code qui permet de déterminer quels types de processeurs ils peuvent recevoir. Exemple : l'embuse SPGA portant la mention "Socket 7" est conçue pour recevoir aussi bien le Pentium I de Intel que l'AMD K5/K6 ou le Cyrix M1/II.

A l'époque du Pentium II, pour installer la mémoire cache le plus près possible du processeur on a pendant un temps placé le processeur et la mémoire cache dite "externe" dans des cartouches. Ces dernières étaient insérées dans un connecteur appelé *Slot* (fente) comparable aux connecteurs des cartes d'extension. C'était le "Slot A" pour l'AMD Athlon SECC (Single Edge Contact Cartridge = cartouche à contacts sur un seul bord) ou le "Slot 1" pour le Pentium II de Intel, les premiers Pentium III ou le Celeron SECC.

Les sockets :

Dès que la technologie a permis d'intégrer la mémoire cache de niveau 2 à la puce du processeur les supports de processeur ont repris la forme d'embases carrées, les sockets. Chaque constructeur a désormais les siens. Il n'est plus question de mettre un processeur AMD sur une carte prévue pour Intel.

Voici les supports de processeur que l'on trouve sur les cartes mère actuelles :

- Le socket 370 qui possède 370 contacts en quinconce adaptés au brochage du Pentium III
- Le Socket 462 souvent appelé Socket A est destiné aux Athlon et Duron d'AMD
- Le socket 423 a été le premier type de socket pour Pentium 4, il a été remplacé en 2001 par socket 478 lui aussi en fin de carrière
- Le socket LGA 775 existe depuis 2004 et reçoit aussi bien les Pentium 4 que les Pentium D (Dual core) ou les Core 2 Duo

Figure 12 : Exemple d'interface de cours avec Menu.

Voici l'interface de l'introduction du cours plus un peu d'histoire sur les processeurs de type MIPS dont le Menu principale se situe au côté gauche de l'interface où on trouve sous chaque menu un sous menu secondaire.

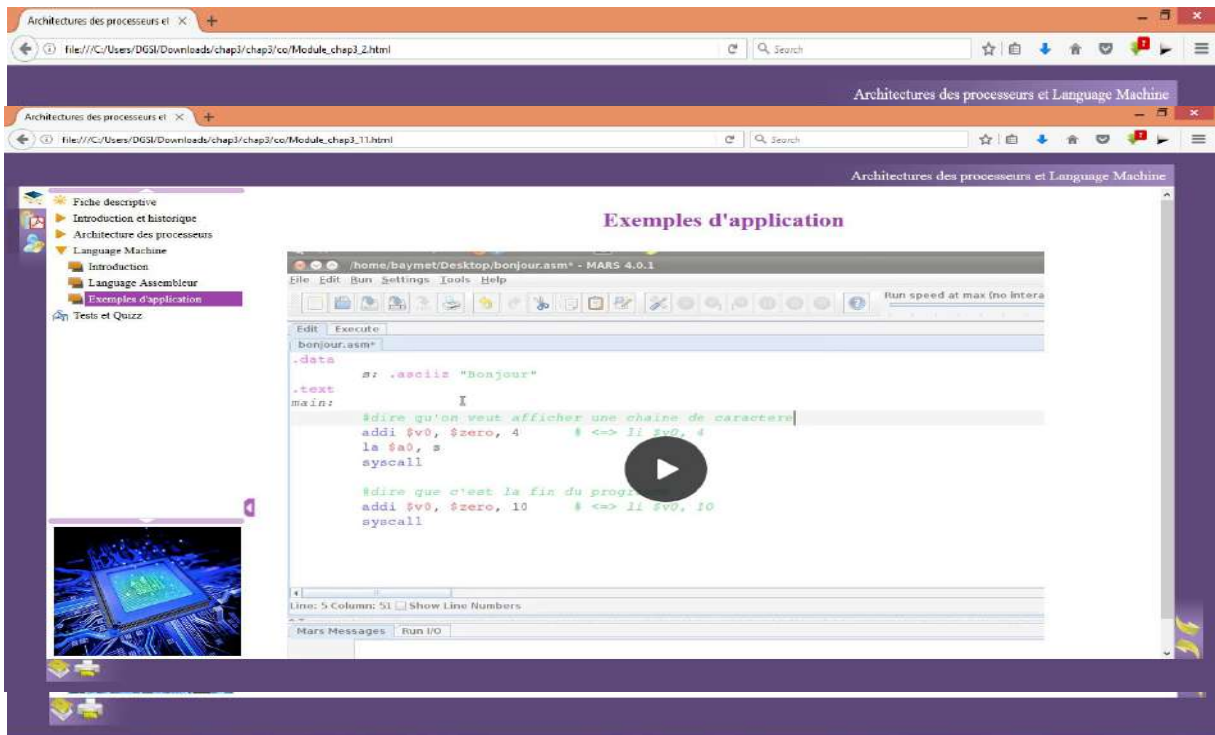


Figure 13 : Le volet Architecture des processeurs.

L'interface précédente illustre la partie architecture des processeurs dont l'interface contient par la suite une introduction générale sur l'architecture des processeurs qui contient une simulation concrète du comportement du processeur lors d'une exécution d'une commande interne (comme :load, sw, beq, addi etc). Ensuite une vue globale sur le langage assembleur enrichi d'un simulateur de programmation des commande processeur MIPS appelé Mars.

En terminant par un petit quizz dont le but est de tester les connaissances acquises à partir du support de cours.

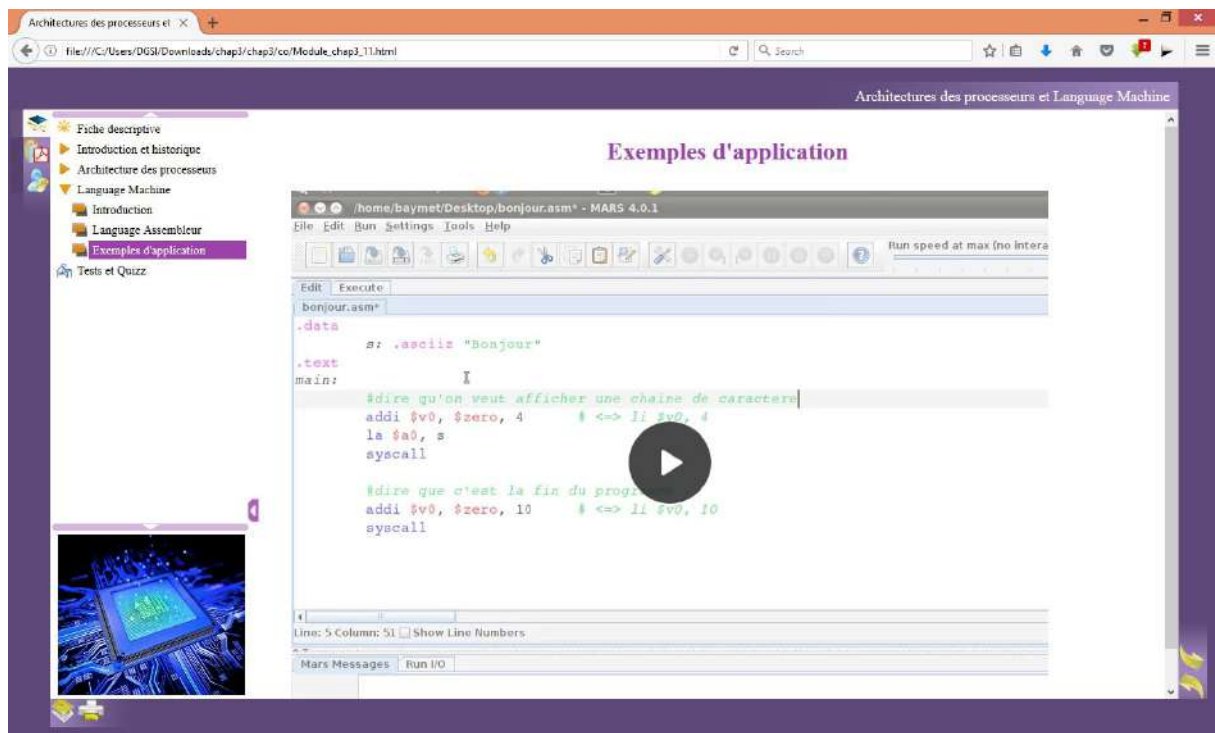


Figure 14 : Exemple d'application de simulation Mars.

Ceci est un exemple d'application de simulation de l'exécution des commandes processeurs MIPS (programme « bonjour ») sous forme d'une vidéo pour faire illustrer aux étudiants comment écrire un programme sur **Mars 4.0.1**. On introduit le programme, ensuite on lance son exécution en observant les résultats.

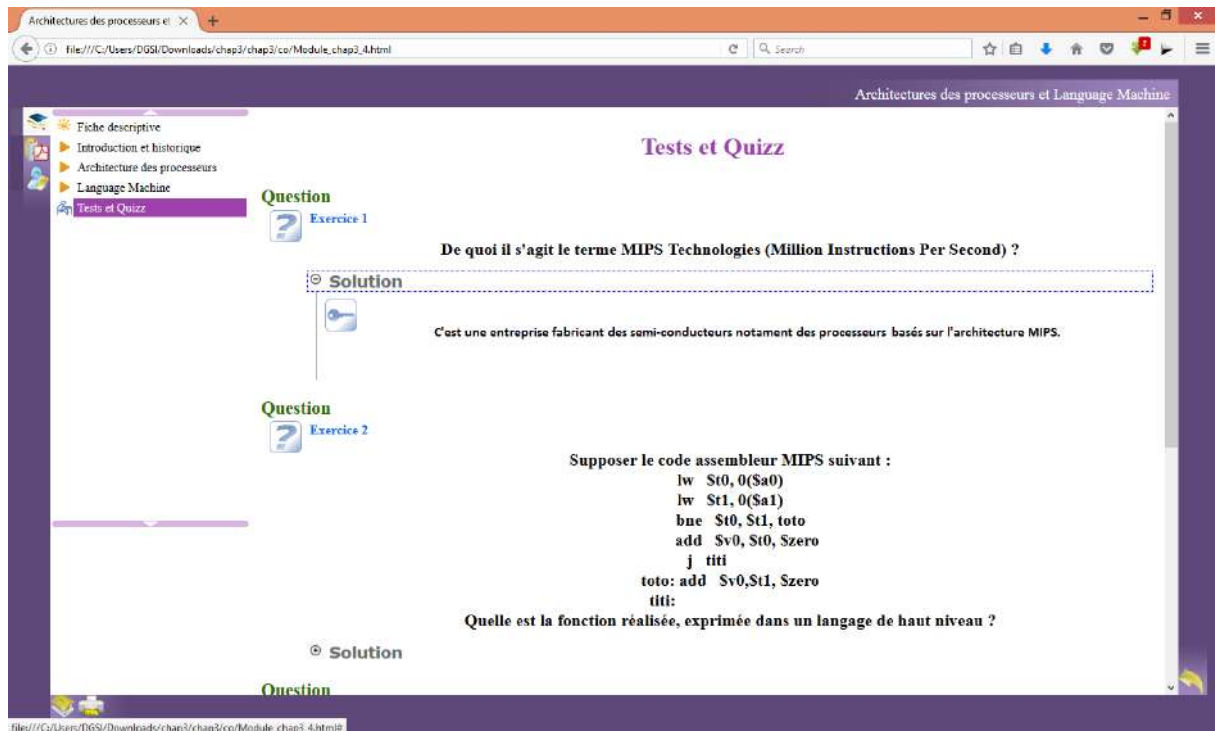


Figure 15 : Interface de QUIZZ et Test.

En sanctionnant le support de cours en finale par un quizz permet de bien fixer les connaissances acquise du support par les étudiants toute en cachant les solutions des exercices pour laisser les étudiants le temps de réfléchir à la solution.

En annexe, on a fait présenter un support de cours de l'encadreur « Mme Benkhrourou Chafika ». Le but est d'enrichir le support de cours développé en introduisant six cours différents sur l'architecture du processeur MIPS. Nous commençons par la présentation des processeurs MIPS3000 ensuite le langage d'assemblage du MIPS3000. Après cela les procédures et les fonctions enrichis par les tableaux et les piles en terminant par les exceptions et les interruptions comme ce qui suit ;

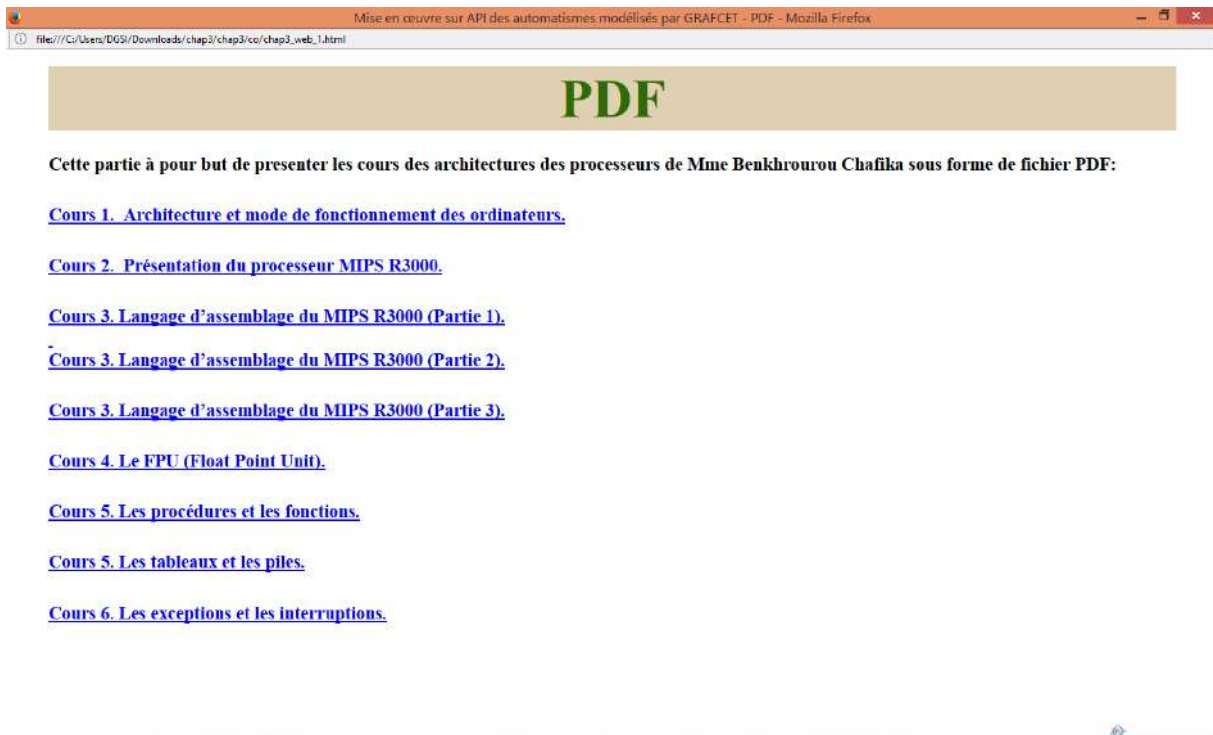


Figure 16 : Support de cours de l'encadreur

V.3 CYCLE D'EXECUTIONS DES COMMANDES SUR UN PROCESSEUR MIPS :

Une instruction est exécutée par le processeur au cours d'un cycle (également appelé cycle de **recherche et exécution** ou encore fetch-decode-execute cycle). Les quatre étapes successives que les architectures Von Neumann utilisent sont la recherche de l'instruction, le décodage de l'instruction (opération et opérandes), l'exécution de l'opération et pour finir l'écriture du résultat. Cette séquence constitue le cycle d'une instruction et est répété continuellement par le micro-processeur, tant qu'il est en fonctionnement.

Au démarrage le premier cycle active la lecture de l'instruction contenue dans la case mémoire 0 et ainsi de suite.

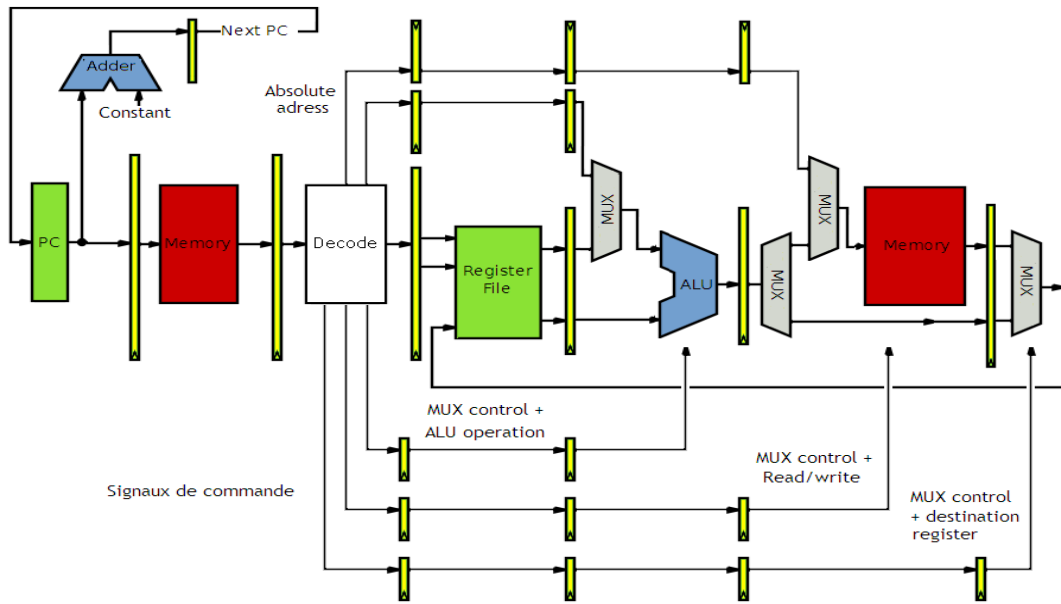


Figure 17 : Cycle de commandes de processeur MIPS3000.

Ces cycles d'exécutions de base pour une instruction, qui servent de référence pour le calcul des performances d'un micro-processeur, sont exprimés parfois en Instructions par seconde (la plupart du temps aujourd'hui exprimée en million d'instructions par seconde, MIPS). Il est important de comprendre que la notion cycle d'exécution ne concerne que les instructions en langage machine.

Les instructions plus complexes d'un langage de programmation (comme une boucle *For* en C ou Java) sont elles-mêmes décomposées en milliers d'instructions en langage machine pour être exécutées sur un microprocesseur, le plus souvent lors de l'étape de compilation.

En représentant les cycles par un diagramme de figure, on trouve le schéma ci-après des trois cycles principaux d'exécution des commandes de processeurs MIPS3000:

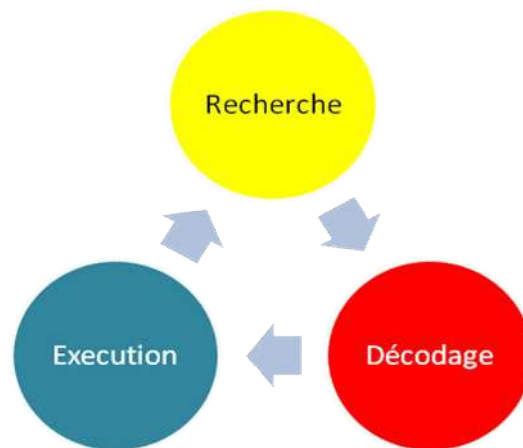


Figure 18 : Cycles d'exécution des instructions Assembleur.

Les instructions d'un programme doivent toujours, être converties sous une forme directement lisible par le processeur. C'est le compilateur qui est chargé de transformer les instructions d'un code source écrit dans un langage de programmation en code machine.

Cette opération peut parfois exiger plusieurs étapes intermédiaires, comme dans le cas du langage Java, par exemple, qui voit ses instructions d'abord transformées par une première étape de compilation, en vue d'être lisible par une machine virtuelle qui elle-même réalisera une nouvelle interprétation en instructions machines pour exécuter le programme. C'est le principe du langage dit semi-interprété.

Avec d'autres langages tels que C, la transformation appliquée aux instructions vise à créer un programme exécutable, c'est-à-dire dont les instructions sont directement lisibles par le micro-processeur. On parle alors de langage compilé.

Pour finir, le cas du langage assembleur est particulier puisque, en ce qui le concerne, le fichier source d'un programme contient des représentations textuelles (mnémoniques) des instructions du micro-processeur. Il est donc impératif de transformer la forme textuelle des instructions du programme en une forme binaire compréhensible par le processeur. On parle par usage de compilateur pour le langage assembleur mais, à proprement parler, un compilateur assembleur procède plutôt à une simple conversion de format.

V.4 IMPLEMENTATION D'UNE SIMULATION DES CYCLES D'EXECUTION D'UNE INSTRUCTION MACHINE SUR PROCESSEUR MIPS3000:

Pour bien illustrer au étudiants les différentes cycles d'exécutions des instructions machine (instruction assembleur) sur un processeur MIPS, nous avons penser à intégrer une interface de simulation d'exécutions des instructions machine en langage HTML5 développé

en Java comprend six commande machine à priori (qui sont :**add, sub, lw, sw, beq and j**).

Voici ci-après l'interface intégrée :

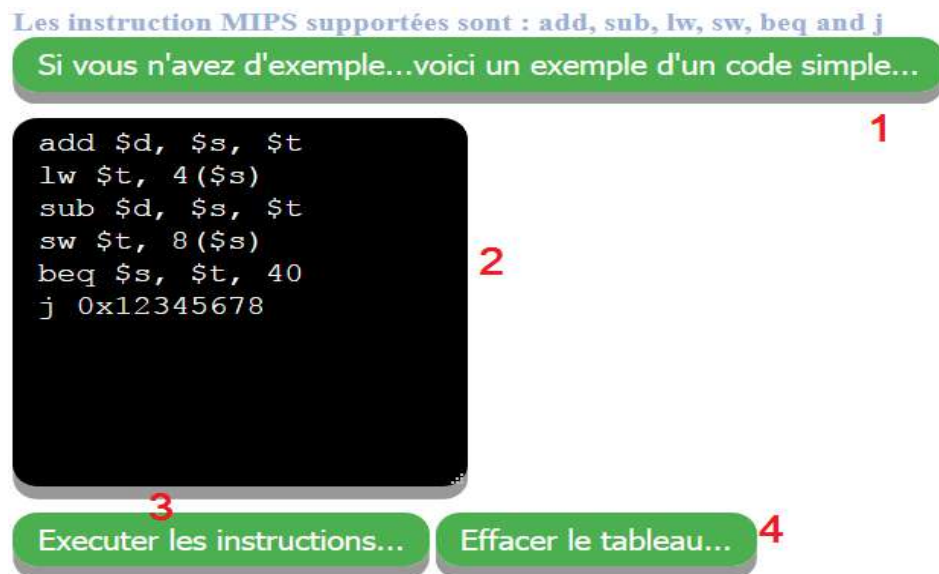


Figure 19 : Exemple d'interface de simulation intégrée.

Cette interface comprend 04 boutons qui sont:

- **Le bouton 1:** Si vous n'avez pas d'un exemple à saisir sur la partie de saisie (noir), vous pouvez cliquer sur ce bouton pour remplir la partie noir par un exemple préparé à priori et simuler son exécution.
- **Le bouton 2:** c'est la partie de saisie des instructions machine.
- **Le bouton 3:** En cliquant sur ce bouton, une autre interface va s'afficher pour voir la simulation des instructions saisies.
- **Le bouton 4:** Ce bouton permet d'effacer la partie de saisie pour ressaisir les instructions.

Cet interface à pour but de saisir les instructions supportées en format machine et de simuler leurs exécutions sur une deuxième interface comme illustré ci-après :

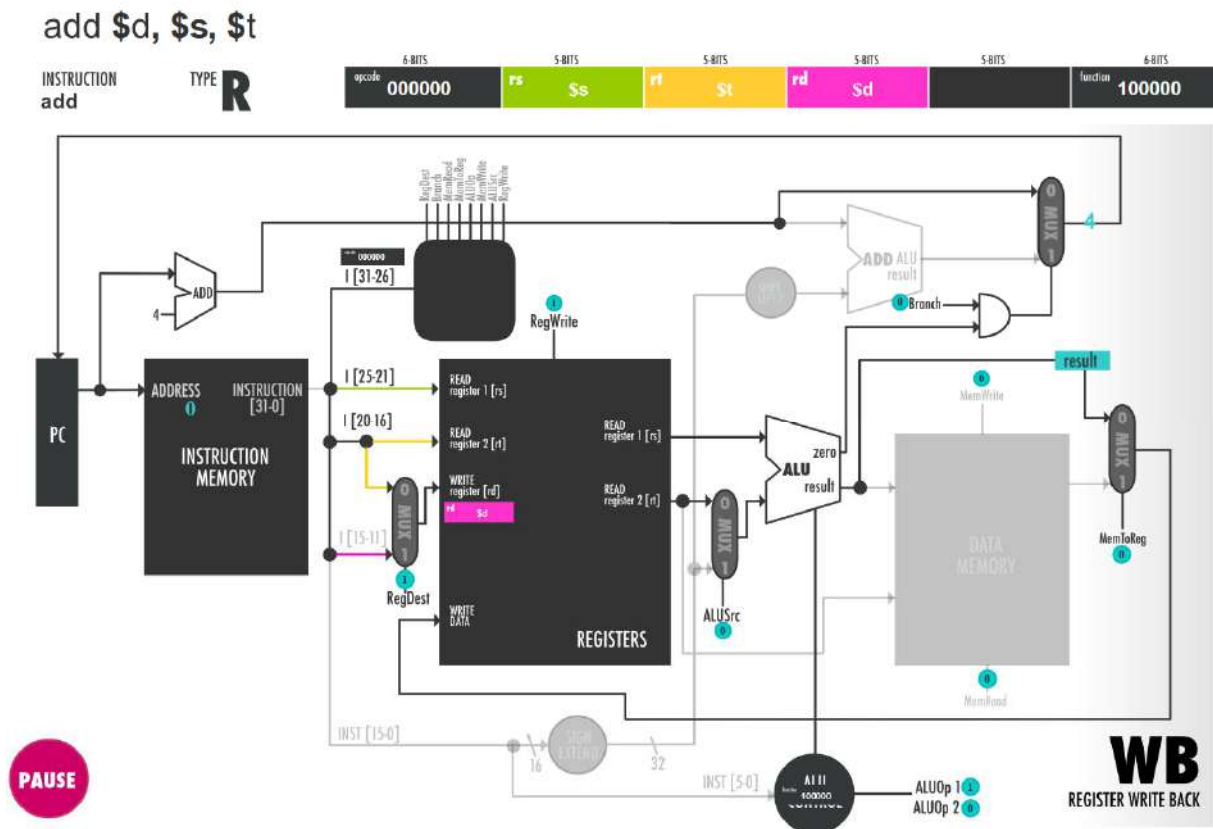


Figure 20 : Interface intégrée de simulation d'exécution des commandes

C'est une interface semblable à un schéma interne d'un processeur où on simule l'exécution des commandes machine.

Le code HTML de développement introduit représente un jeu d'instruction HTML5 contient un ensemble de fonctions prédéveloppés (orienté objets) en assimilant que le nom de la fonction.

V.5 CODE SOURCE HTML:

Voici ci-après une vue complète du code HTML5 choisi pour l'animation des scénarios d'exécutions des commandes machines ;

```

<div id="swiffy-wrapper">
  <!--processor animation will be inserted below here -->
  <div id="swiffycontainer" style="width:100%; height:690px;"></div>
  <!--processor animation will be inserted above here -->
</div><!--end of swiffy wrapper -->

<script>
  $('#instructions').mouseover(function(){
    $(this).stop().animate({
      opacity: 1
    }, 500, function() {
      // Animation complete.
    });
  });
  $('#instructions').mouseout(function(){
    $(this).stop().animate({
      opacity: 0.08
    }, 500, function() {
      // Animation complete.
    });
  });
  var code = $('#code');
  $('#fill-example').on('click', function(){
    var exampleCode =
      'add $d, $s, $t \n' +
      'lw $t, 4($s) \n' +
      'sub $d, $s, $t \n' +
      'sw $t, 8($s) \n' +
      'beq $s, $t, 40 \n' +
      'j 0x12345678';
    code.val(exampleCode);
  });

  $('#clear-code').on('click', function(){
    code.val('');
  });

  $('#restart-processor').on('click', function(){
    location.reload();
  });
  $('#play').on('click', function(){
    if(code.val() == ''){
      alert('Vous devez entrer des instructions dans le box');
    }else{ //code has been entered
      //validate code
      var codeLines = code.val().split('\n');
      var code_string = '';
      $('#instructions').append('Le processeur va executer : ');
      for(var i=0; i< codeLines.length; i++){
        $('#instructions').append(codeLines[i] +
'<br />');
        //console.log("line ["+ i +"]"
+codeLines[i]);
        //remove commas if any
        var new_line =
codeLines[i].replace(/\\,/gi, '');
        var new_line = new_line.replace(/\\(/gi, '
');
        var new_line = new_line.replace(/\\)/gi, '
');
        code_string += new_line + ' \n ';
      }
      //code is "valid"
      var stage = new
swiffy.Stage(document.getElementById('swiffycontainer'), swiffyobject);
      var flashvar = 'code_string=' + code_string;
      //console.log(flashvar);
      stage.setFlashVars(flashvar);
      stage.start();
      //make button disappear
      $('#page-wrapper').hide();
      $('#instructions').show();
      $('#restart-processor').show();
      $(document).scrollTop( $('#processor').offset().top );
    }
  });
</script>

```

Figure 21 : Vue globale du code HTML utilisé pour l'animation des scénarios.

V.6 PASSAGE DES SCENARIOS DES INSTRUCTIONS MACHINES:

Pour exécuter une instruction, le processeur va effectuer trois étapes :

- l'unité de chargement va charger l'instruction depuis la mémoire : c'est l'étape de **chargement** (ou fetch) ;
- le séquenceur va ensuite « étudier » la suite de bits de l'instruction et en déduire comment configurer les circuits du processeur pour exécuter l'instruction voulue : c'est l'étape de **décodage** ;
- enfin, le séquenceur configure le chemin de données pour exécuter l'instruction : c'est l'étape de **exécution**.

Ces trois étapes ne s'effectuent cependant pas d'un seul bloc. Chacune de ces étapes est elle-même découpée en plusieurs sous-étapes, qui va échanger des données entre registres, effectuer un calcul, ou communiquer avec la mémoire.

Pour l'étape de **Fetch**, on peut être sûr que tous les processeurs vont faire la même chose. Même chose pour la plupart des processeurs MIPS, pour l'étape de décodage. Mais cela change pour l'étape d'exécution : toutes les instructions n'ont pas les mêmes besoins suivant ce qu'elles font ou leur mode d'adressage.

Commençons par prendre l'exemple d'une instruction de lecture ou d'écriture en mode d'adressage absolu **add \$d, \$s, \$t**. Vu son mode d'adressage, l'instruction va indiquer l'adresse à laquelle lire dans sa suite de bits qui la représente en mémoire. L'exécution de l'instruction se fait donc en une seule étape : la lecture proprement dite.

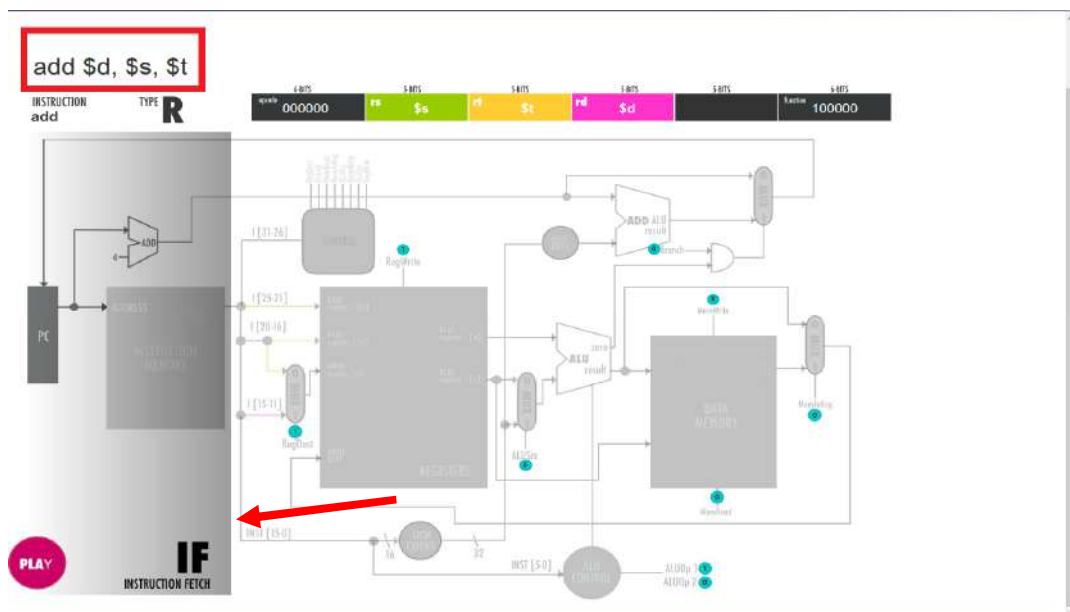


Figure 22 : L'étape de recherche de l'instruction Add

Prenons maintenant le cas d'une instruction d'addition (**add**). Celle-ci va additionner deux opérandes, qui peuvent être soit des registres, soit des données placées en mémoires, soit des constantes. Si les deux opérandes sont dans un registre **\$t** et que le résultat doit être placé dans un registre **\$s**, la situation est assez simple : la récupération des opérandes dans les registres **\$t** et **\$s**, le calcul, et l'enregistrement du résultat dans les registres **\$d**, sont trois étapes distinctes.

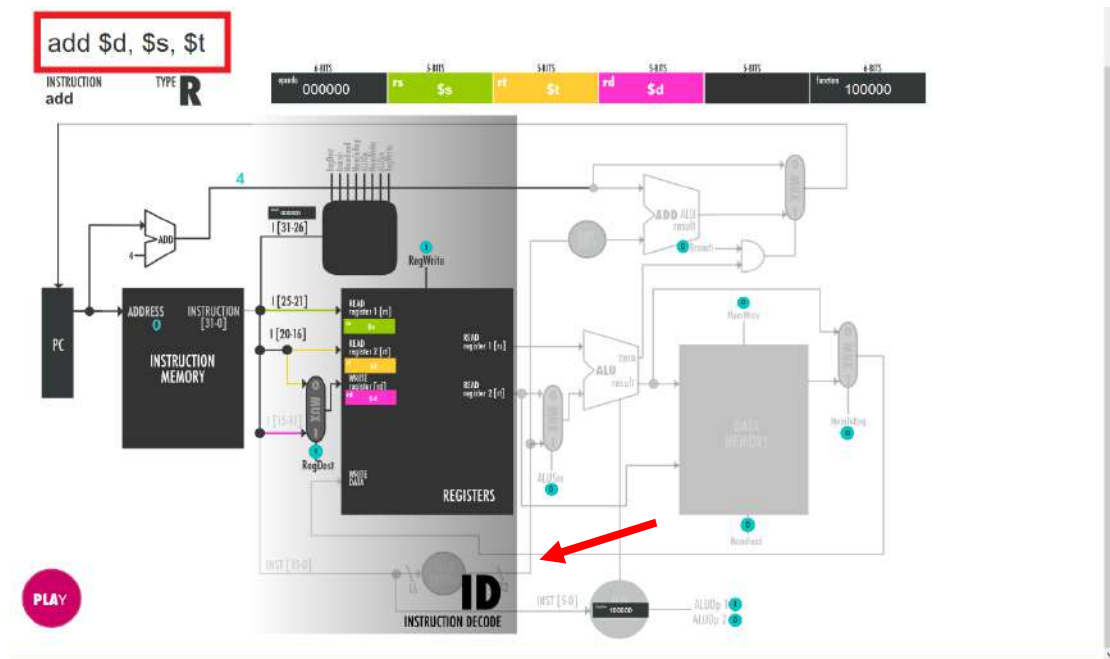


Figure 23 : l'étape décodage d'instruction Add

Maintenant, autre exemple : un opérande est à aller chercher dans la mémoire d'instruction, une autre dans un registre **\$s**, et le résultat doit être enregistré dans un registre **\$d**. On doit alors rajouter une étape : on doit aller chercher la donnée en mémoire. Et on peut aller plus loin en allant chercher notre première opérande en mémoire : il suffit d'utiliser le mode d'adressage Base + Index pour celle-ci (. On doit alors rajouter une étape de calcul d'adresse en plus). Un opérande en mémoire, l'autre dans un registre, et stocker le résultat en mémoire.

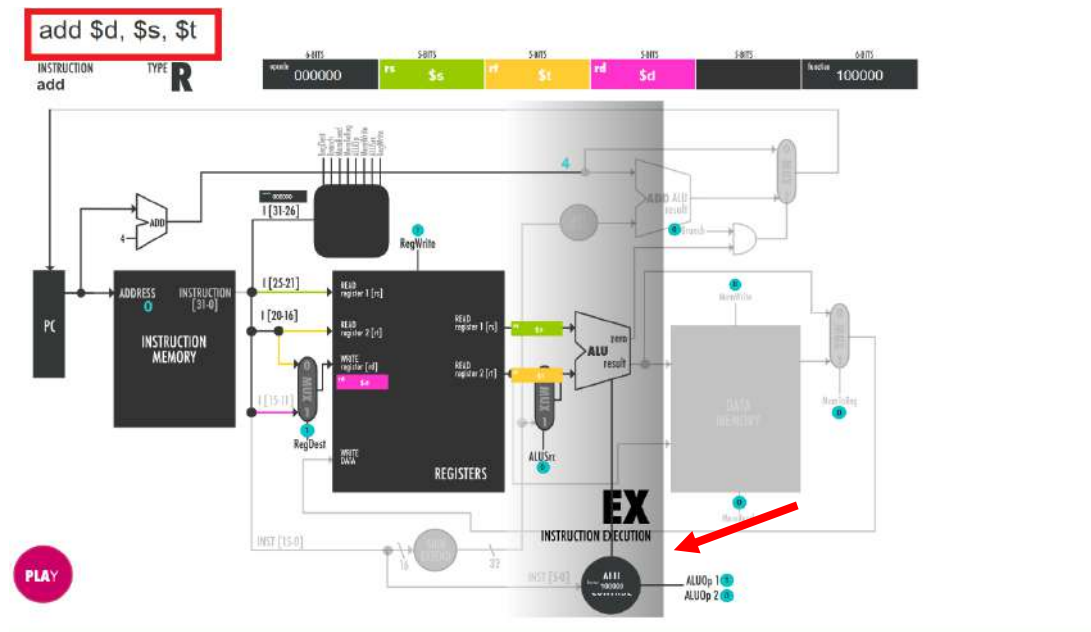


Figure 24 : l'étape d'exécution de la commande Add.

Reprenons notre instruction Load (**lw**), mais en utilisant un mode d'adressage utilisé pour des données plus complexe. Par exemple, on va prendre un mode d'adressage du style Base + Index. Avec ce mode d'adressage, l'adresse doit être calculée à partir d'une adresse de base, et d'un indice, les deux étant stockés dans des registres. En plus de devoir lire notre donnée, notre instruction va devoir calculer l'adresse en fonction du contenu fourni par deux registres. L'étape d'exécution s'effectue dorénavant en deux étapes assez différentes : une implique un calcul d'adresse, et l'autre implique un accès à la mémoire.

Les autres commandes utilisées **sub**, **lw**, **sw**, **beq** and **j** font l'objet des commandes principales du code machine, Le même scénario propriétaire de la commande **add** se répète sur ces commandes en commençant par la partie recherche, décodage et exécution dont;

❖ **La commande sub** (substrate); Cette instruction permet de soustraire une valeur à un opérande, soit leur syntaxe:

SUB opérande cible, opérande source

❖ **La commande lw** (Load); cette instruction permet de charger une valeur dans un registre spécifié, soit leur syntaxe:

LW opérande cible, opérande source

❖ **La commande sw** (Store); Cette commande permet d'enregistrer une valeur dans un registre, soit leur syntaxe:

SW opérande cible, opérande source

- ❖ **La commande beq** (Branch on equal); faire un saut à une adresse avec condition, soit leur syntaxe:

BEQ opérande 1, opérande 1, adresse

- ❖ **La commande j** (Jump); Faire un saut à une adresse indiquée, soit leur syntaxe:

J opérande 1, adresse

V.7 CONCLUSION:

Nous avons présenté dans ce chapitre notre contribution en essayant de répondre aux besoins des utilisateurs liés au domaine de l'architecture des ordinateurs et du langage assembleur. Nous avons développé les principales étapes qu'un étudiant voulant connaître sur le domaine des processeurs MIPS3000.

La première étape a été de définir les outils utilisés dans le développement de l'interface (HTML et BlueGriffon).

Deuxième étape, nous avons intégré plusieurs interfaces importantes dans le développement de la plate-forme, et nous avons ensuite fait une petite explication des figures insérées.

Ensuite, un résumé des cycles d'exécution d'une instruction des processeurs MIPS3000 est présenté.

CONCLUSION GENERALE

Notre objectif initial est d'apporter une contribution concernant la réalisation d'un outil graphique qui simule le fonctionnement du microprocesseur MIPS R3000

Nous avons commencé notre travail par la présentation des notions qui ont une relation directe avec notre travail qui sont l'E-learning et le microprocesseur MIPS R3000 en vue de fixer les concepts utilisés dans ce domaine.

Nous avons traité l'importance des interfaces graphiques pour motiver les étudiants et les encourager à réviser leurs cours. Ceci nous a permis de mettre en évidence la nécessité de produire beaucoup plus d'outils graphiques en E-learning afin de donner aux apprenants la chance de bien comprendre le cours en ligne.

Nous avons implémenté notre contribution en essayant de répondre aux besoins des utilisateurs liés au domaine de l'architecture des ordinateurs et de langage assembleur, nous y avons développé les principales étapes cherchées par un étudiant qui veut approfondir ses connaissances dans le domaine des processeurs MIPS3000.

Nous avons commencé par une conception générale de notre système. Ensuite, nous avons défini les outils utilisés dans le développement de l'interface (HTML et BlueGriffon). Après cela, nous avons intégré plusieurs interfaces importantes dans le développement de la plate-forme, avec une petite explication des figures insérées.

Pour terminer, un résumé des cycles d'exécution d'une instruction des processeurs MIPS3000 est présenté.




Au cours de notre travail, nous avons cependant été limités par le temps. Le nombre important des instructions du langage MIPS nous a emmené à se focaliser uniquement sur quelques instructions de bases.











Une des premières perspectives est d'apporter plus de contribution concernant la communication de l'enseignant avec les étudiants afin de leur expliquer les cours. Les examens et tests corrigés par l'enseignant seront réalisés dans le futur.

Pour finir, nous espérons que notre contribution dans le cadre de ce travail a su montrer la réelle importance et pertinence des outils graphiques et l'e-learning dans système médiatisé d'enseignement-apprentissage.

Bibliographie

-  [01] Andréa Moura da Costa, « *Rapport de thèse* », Université Nancy 2, Année 2006
-  [02] Mahmoud Baklouti, « *E-learning : Présentation, aspects, enjeux et avenir* », mémoire de mastère, UNIVERSITE DE SFAX, Année 2003.
-  [03] Bouamra Amel, MankourSyhem, « *Utilisation-de-la-plate-forme-Dokeos pour une solution e-Learning industrielle* », *Mémoire de fin d'études Université KasdiMerbah OUARGLA* , Année 2010.
-  [04] François Larrey « *E-LEARNING SES FONDEMENTS ET SON UTILISATION DANS LE SECTEUR BANCAIRE* »
-  [05] Bernard Blandin, « *L'histoire de l'enseignement à distance et de la FOAD* », Centre Inffo, *Traité des sciences et techniques de la formation Paris : Dunod* (2e édition), Année 2004.
-  [06] Amor BOULEDROUA « *Découverte des préférences des apprenants dans un environnement de e-learning* » *MEMOIRE* Présenté en vue de l'obtention du diplôme de *MAGISTER*, *UNIVERSITE BADJI MOKHTAR-ANNABA* Année 2012
-  [07] BEKRAR Marwa « *Protection de la vie privée à base d'agents dans un système d'e-learning* » *Mémoire de fin d'études Université KasdiMerbah OUARGLA* 2013-2014
-  [08] D.M.F.F, « *Guide e-learning* », Département Management & Formation Fédérale.

-  [09] F.E.U.Q, « *Avis sur la formation à distance* », Fédération étudiante universitaire du Québec, Année 2009.
-  [10] *Pascal Roques*, « *UML 2 Modéliser une application web* », éditions eyrolles, Année 2007.
-  [11] *Bouamra Amel, MankourSyhem*, « *Utilisation-de-la-plate-forme-Dokeos pour un solution e-Learning industrielle* », *Mémoire de fin d'études, Université KasdiMerbah OUARGLA* Année 2010.
-  [12] *Sans auteur* « *MIPS R3000 Langage d'assemblage* », Version 1.4 UNIVERSITE PIERRE ET MARIE CURIE Septembre 2001.
-  [13] *ABAZ Samira et DJIOUA Wassila* « *EVALUATION ERGONOMIQUE DE L'HM D'UNE PLATE FORME D'ENSEIGNEMENT A DISTANCE ETUDE DE CAS* » ; *Mémoire de fin d'études, Université KasdiMerbah OUARGLA* Année 2016
-  [14] *France Henri*, « *Le synchrone en formation à distance* », *Télé-université/UQAM, Séminaire TEMATICE*, Année 2008.
-  [15] *Sans auteur* « *Le e-learning dans la stratégie de formation des entreprises* », Année 2002
-  [16] *Réperant , Eric* « *Formation composite : méthodologie de projet* » *Septembre 2001*.
-  [17] *Sans auteur* « *L'e-learning au service des connaissances de l'entreprise* » Janvier 2003
-  [18] *Sans auteur* « *Projet Logiciel en C - Sujet 1 : Simulateur de microprocesseur MIPS* » *Ensimag 1`ere année - 2011/12 31 mai 2012*

-  [19] Sans auteur « *MIPS R3000 Architecture externe* » UNIVERSITÉ PIERRE ET MARIE CURIE. 9 octobre 2003
-  [20] Sans auteur « *Processeur MIPS R3000. Architecture externe et jeu d'instructions MIPS* » Ensimag - Filières SLE/ISI
-  [21] Sans auteur « *Processeur MIPS R3000. Langage d'assemblage* » UNIVERSITÉ PIERRE ET MARIE CURIE. Année : 2001
-  [22] BENKHOUROU Chafika « *Cours 3: Langage d'assemblage du MIPS R3000 (Partie 1)* ». Université KASDI MERBAH ó Ouargla. Année : 2015-2016.
-  [23] Mahmoud Baklouti, « *E-learning : Présentation, aspects, enjeux et avenir* », mémoire de mastère, UNIVERSITE DE SFAX, Année 2003
-  [24] BENKHOUROU Chafika « *Cours 1: Architecture et mode de fonctionnement des ordinateurs* ». Université KASDI MERBAH ó Ouargla. Année : 2015-2016.
-  [25] BENKHOUROU Chafika « *Cours 2: Présentation du processeur MIPS R3000* ». Université KASDI MERBAH ó Ouargla. Année : 2015-2016
-  [26] BENKHOUROU Chafika « *Cours 3: Langage d'assemblage du MIPS R3000 (Partie 2)* ». Université KASDI MERBAH ó Ouargla. Année : 2015-2016.
-  [27] [https://fr.wikipedia.org/wiki/Architecture MIPS](https://fr.wikipedia.org/wiki/Architecture_MIPS) date de consultation le 01/03/2018 à 20 :09.
-  [28] http://www.parisson.com/Members/yomguy/docs/e-l_fondements.pdf date de consultation le 08/04/2018 à 12 :09.



[29]

<https://elearningindustry.fr/apprentissage-en-ligne-avantages-limites> date
de

consultation le 12/04/2018 à 15 :19.



[30]

<https://revolution-rh.com/e-learning-avantages-inconvenients> date de

consultation le 08/02/2018 à 09 :30.