Université Hassan 1er
Faculté des Sciences et Techniques Settat

Simulateur de microprocesseur Motorola 6809

Encadré par: **Mr. BENALLA Hicham**

Réalisé par : **Ayoub Chahib**
Romisio Maria Farinha

ANNEE UNIVERSITAIRE 2025 - 2026

Présentation du Motorola 6809 :

Le Motorola 6809 est un microprocesseur 8 bits avec certaines capacités 16 bits, conçu par Motorola et introduit en 1978. Il est compatible source avec le Motorola 6800, mais apporte de fortes améliorations (architecture plus avancée, nouvelles instructions, nouveaux registres, meilleurs modes d'adressage). À son époque, il était considéré comme l'un des microprocesseurs 8 bits les plus puissants.On peut identifier plusieurs caractéristiques fondamentales, telles que :

- ✓ Jeu d'instructions flexible et modes d'adressage variés:
  Le Motorola 6809 possède environ 59 instructions fondamentales. Cependant, l'introduction de différents modes d'adressage permet d'obtenir plus de 1460 codes opérationnels possibles. On distingue plusieurs types d'adressage, tels que : inhérent, immédiat, direct, étendu, indexé, relatif, indirect... Cette richesse rend la programmation en assembleur plus puissante et plus flexible, permettant une grande flexibilité dans l'accès aux données et la manipulation de la mémoire.

- ✓ Architecture supérieure :
  Le Motorola 6809 était plus avancé que les autres processeurs 8 bits de son époque car il avait plus de registres et des modes d'adressage plus flexibles, ce qui permettait une programmation plus facile et plus puissante.

- ✓ Support avancé des interruptions et multitâche :
  Le 6809 offrait une gestion des interruptions plus flexible et structurée, facilitant la gestion simultanée de plusieurs tâches ou événements. À l'inverse de nombreux processeurs 8 bits à cette époque, il était capable de gérer des routines d'interruption plus sophistiquées sans compromettre des informations cruciales.

Influence du Motorola 6809 :

Le microprocesseur Motorola 6809 fut l'un des plus puissants de sa catégorie à son époque et connut un succès important puisqu'il fut utilisé dans une multitude d'appareils dans les années 70 et 80. On le retrouvait dans des ordinateurs personnels, mais aussi dans des micro-ordinateurs utilisés dans les foyers, et même dans certaines consoles de jeux vidéo. Il servit également d'électronique principale pour des machines équipées du système d'exploitation multitâche. Par sa puissance et sa flexibilité, le 6809 contribua donc fortement à l'évolution des machines informatiques de la fin des années 70 et du début des années 80.

Introduction :

Pour approfondir la connaissance des architectures de microprocesseurs et comprendre les principes des langages bas-niveau, comme celui de l'assembleur, l'utilisation d'un simulateur est indispensable. Il est notamment nécessaire pour :

- ✓ Analyse du fonctionnement interne : Le simulateur permet de visualiser comment les instructions sont exécutées ou encore comment les registres sont utilisés et, de manière plus générale, permet de découvrir les différentes parties du processeur 6809, ce qui est difficile à realiser avec un processeur réel.
- ✓ Expérimentation sans risque : Il s'agit d'un outil idéal pour écrire des programmes, déboguer et tout simplement pour découvrir le fonctionnement du processeur sans avoir peur de le mettre en panne.
- ✓ Accessibilité et utilisation : Les simulateurs permettent de faire tourner le processeur 6809 sur n'importe quelle machine moderne et donc offrent un accès à l'architecture 6809 à tous les étudiants et développeurs sans aucune contrainte d'ordre matérielle ou logiciel.

Buts du simulateur 6809 :

Cette partie présente les objectifs du simulateur du microprocesseur Motorola 6809, en montrant ses objectifs d'utilisations en tant qu'outil pédagogique et comme outil fonctionnel :

- ✓ Faciliter la compréhension du fonctionnement du microprocesseur Motorola 6809 et identifier le rôle des registres, des instructions et des composants internes.
- ✓ Proposer un environnement permettant d'écrire, d'exécuter et de débugger des programmes de bas niveau sans utiliser une vraie machine.
- ✓ Proposer une fonctionnalité pas à pas pour aider l'utilisateur à suivre l'exécution d'un programme et à observer les modifications de l'état du processeur.
- ✓ Montrer comment fonctionne la mémoire dans un ordinateur en faisant la différence entre la mémoire vive (RAM) et la mémoire morte (ROM), et comment le processeur y accède.
- ✓ Proposer un outil pédagogique simple et interactif pour apprendre l'architecture des microprocesseurs et les notions de base de la programmation bas niveau.

Organisation fonctionnelle des fichiers :

Les fichiers ArchitecteInterne.java, Editeur.java, Menu.java, RAM.java, ROM.java, Programme.java, représentent différentes composantes du simulateur Motorola 6809. L'ensemble de ces fichiers correspond à la base logicielle du simulateur, réalisant la simulation du comportement du 6809 ainsi que l'exécution de programmes.

Le simulateur du microprocesseur Motorola 6809 est un projet modulaire, dans lequel chaque grande fonctionnalité est gérée dans un fichier Java à part. Cette conception, qui repose sur la programmation orientée objet en Java, offre une meilleure lisibilité du projet grâce à des classes qui ont chacune leur propre rôle. La distinction entre le contrôle du processeur, de la mémoire et de l'interface utilisateur empêche également les mauvaises interactions entre ces entités :

Menu.java : permet une navigation globale dans les différentes fonctionnalités du simulateur.

Editeur.java : sert à rédiger, à compiler et à exécuter le programme destiné .

ArchitecteInterne.java : comprendre la simulation de l'architecte d'un microprocesseur ainsi que l'exécution des instructions.

RAM.java : représente la mémoire vive utilisée par l'ordinateur pour stocker temporairement des données ou des instructions.

ROM.java : représente la mémoire morte contenant des données ou des instructions pouvant être uniquement lues.

Programme.java : permet d'afficher un programme ou les résultats de son exécution.

Architecture du Simulateur Motorola 6809 :

L'architecture global du simulateur Motorola 6809 repose sur un ensemble de fichiers java, chacun représentant une composante essentielle du système simulé :

ArchitecteInterne.java :

Le fichier ArchitecteInterne.java est au cœur de notre simulation du microprocesseur Motorola 6809. C'est le module qui simule l'architecture interne du processeur, gère les registres, l'état du processeur et qui réalise l'exécution des instructions.C'est celui qui reçoit les instructions, qui les interprète et qui modifie l'état du processeur selon les besoins.

Il doit aussi interagir avec les modules RAM.java et ROM.java avec qui il va lire ou écrire les données nécessaires à l'exécution des programmes. Il doit communiquer avec Editeur.java pour récupérer les instructions à exécuter, et enfin avec nos interfaces graphiques pour pouvoir afficher l'état du processeur tout au long de l'exécution.

RAM.java :

La classe RAM.java représente la mémoire vive de la machine. Il s'agit d'une mémoire temporaire dans laquelle sont placés à l'exécution les données et les codes des programmes.

Cette classe est principalement utilisée par ArchitecteInterne.java, en effet, l'architecte réalise des opérations de lecture et d'écriture dans la RAM. Par ailleurs, Editeur.java et l'interface utilisateur sont également concernés, dans la mesure où l'on peut voir (et parfois modifier) le contenu de la mémoire lors du déroulement d'un programme.

ROM.java :

Le fichier ROM.java représente la mémoire morte du système simulé. Il s'agit d'une mémoire dans laquelle on place des données ou des instructions permanentes qui ne doivent pas être modifiées pendant l'exécution. La ROM permet ainsi de séparer la mémoire qui est permanente de la mémoire qui est volatile.

Ce composant est utilisé par ArchitecteInterne.java. L'architecte interne lit la ROM afin de récupérer des instructions ou des données nécessaires à l'exécution du programme. La ROM est également visible à travers l'interface utilisateur pour mieux comprendre son rôle dans l'architecture globale.

Editeur.java :

Le fichier Editeur.java correspond à l'interface de l'utilisateur du simulateur. Il permet à l'utilisateur de saisir, de modifier et d'exécuter des programmes destinés au microprocesseur Motorola 6809.

Quand un programme est lancé, Editeur.java envoie les instructions à ArchitecteInterne.java pour qu'elles soient interprétées et exécutées. Il communique aussi avec les composants mémoire pour visualiser l'utilisation de la RAM ou de la ROM, et avec les interfaces de visualisation pour suivre l'exécution.

Menu.java

Le fichier Menu.java représente le menu principal du simulateur. Il donne à l'utilisateur la possibilité de naviguer à travers les différents modules du simulateur, comme l'éditeur, l'architecture interne ou la mémoire. En effet, on peut considérer que c'est grâce à lui que le simulateur est utilisable dans sa globalité.

Le menu est en relation avec tous les autres modules du simulateur puisqu'il commande l'ouverture ou la fermeture de leur interface, mais il n'agit pas au cœur du traitement de l'exécution des instructions.

Programme.java :

Le fichier Programme.java est utilisé pour afficher les programmes ou les résultats d'exécution dans une fenêtre spécifique. Il s'agit d'un mode de consultation des données du simulateur plus direct et plus lisible.

Il s'interface principalement avec Editeur.java et ArchitecteInterne.java afin d'afficher les différents résultats d'exécution. Il contribue au débogage en rendant plus facile et plus claire la lecture des résultats.

Environnement technologiques du simulateur :

JAVA :

- ✓ Programmation orientée objet : La nature orientée objet de Java permet de modéliser le simulateur à travers des classes représentant chacun des composants (processeur, mémoire, interface), et ainsi d'organiser son code de manière intuitive.
- ✓ Portabilité : Java est une technologie multiplateforme, c'est à dire que les applications écrites en Java peuvent être exécutées sur n'importe quel système d'exploitation sans nécessiter une quelconque modification du code. Ainsi, le simulateur sera utilisable sur plusieurs plateformes.
- ✓ Lisibilité et maintenance : Enfin, parmi les nombreux avantages de Java, on peut citer sa syntaxe relativement simple et ses nombreux outils qui facilitent la lecture, la maintenance et l'évolution du code , particulièrement adaptés à un projet pédagogique.

SWING :

- ✓ Création d'interfaces graphiques : Swing met à disposition un panel de composants graphiques pour créer des interfaces utilisateurs complètes, comme des menus, des éditeurs ou des fenêtres d'affichage de la mémoire.
- ✓ Intégré à Java : Swing, étant une bibliothèque native de Java, s'intègre facilement au projet et facilite la communication direct entre la GUI et la logique du simulateur.
- ✓ Interaction utilisateur : Avec Swing, il est possible de développer des interfaces interactives facilitant la manipulation du simulateur et permettant de visualiser l'état du processeur et de la mémoire.

Problématique du projet :

La compréhension du fonctionnement interne d'un microprocesseur, tel que le Motorola 6809, constitue un défi de taille pour les étudiants en raison notamment de la complexité de son architecture et de la programmation en langage machine. En effet, l'utilisation d'un processeur réel ne permet pas de toujours voir comment les instructions sont exécutées, comment les registres changent et comment le processeur interagit avec la mémoire. De ce fait, la réalisation d'un simulateur représente une aide à la compréhension du microprocesseur et à l'analyse de son comportement.

Voici les principaux défis de conception que nous avons identifiés pour ce projet :

- ✓ Permettre une visualisation claire du déroulement des instructions et de l'évolution des états internes du CPU.
- ✓ Comprendre comment les registres sont utilisés et pourquoi ils sont essentiels dans l'exécution d'un programme bas niveau.
- ✓ Modéliser le fonctionnement d'une mémoire vive (RAM) et d'une mémoire morte (ROM), et comment le processeur s'y connecte.
- ✓ Offrir un environnement simple de créer, exécuter et déboguer des programmes en langage assembleur.
- ✓ Réaliser un outil éducatif facile d'accès, sans besoin d'avoir un exemplaire physique de chaque étape.

Composants du simulateur Motorola 6809 :

Menu.java :

Interface de navigation du simulateur :

Menu.java est un composant fondamental de l'usage du simulateur puisque c'est à partir de lui que l'utilisateur va pouvoir visualiser la première interface. C'est un peu le centre de contrôle de son utilisation où l'on va pouvoir choisir différentes fonctionnalités du simulateur puis revoir étranger par d'autres modules.

Principales fonctionnalités :

- ✓ Rassembler dans une interface, les différentes commandes du simulateur. Permettre une navigation générale dans le simulateur.
- ✓ Donner accès aux fonctionnalités telles que l'aide, autres types de support,...

Interactions avec les autres composants :

- ✓ Permet d'ouvrir l'éditeur afin de fournir un environnement de saisie des programmes.
- ✓ Acceder aux interfaces de la mémoire vivre (RAM.java) et de la mémoire morte (ROM.java) afin de visualiser leur contenu.
- ✓ Permet d'afficher l'état du microprocesseur en visualisant son architecture interne (ArchitecteInterne.java).

Editeur. Java :

Interface de programmation et d'exécution :

Editeur. java est l'interface principale qui permet à l'utilisateur de communiquer avec le simulateur. Il permet à l'utilisateur de créer, modifier et exécuter des programmes pour le microprocesseur Motorola 6809. Cela fait de cet éditeur un outil très utile dans l'apprentissage de la programmation bas niveau et dans l'expérimentation du processeur simulé.

Principales fonctionnalités :

- ✓ Fournir un environnement de saisie de programmes assembleur du Motorola 6809.
- ✓ Permettre l'exécution directe des programmes depuis l'éditeur.
- ✓ Offrir des fonctionnalités de débogage telles que l'exécution pas à pas.
- ✓ Mettre en évidence certaines parties du code pour améliorer la lisibilité.

Interactions avec les autres composants :

- ✓ Envoie les instructions saisies par l'utilisateur au module ArchitecteInterne.java pour leur interprétation et exécution.

- ✓ Interagit avec la mémoire vive (RAM.java) et la mémoire morte (ROM.java) afin de visualiser l'impact des instructions sur la mémoire.
- ✓ Communique avec Programme.java pour afficher les résultats d'exécution ou le contenu du programme.

Enfin, Menu.java et Editeur.java font partie intégrante de l'interface utilisateur du simulateur Motorola 6809. Le menu donne à l'utilisateur l'accès à l'ensemble des fonctionnalités du simulateur ainsi qu'aux modules principaux. L'éditeur est l'espace où l'utilisateur saisit et exécute les programmes, tout en observant les effets de ces actions sur le processeur et la mémoire. L'interaction étroite des composants crée une interface qui permet à l'utilisateur de naviguer sans difficulté dans le simulateur. Cela aboutit à une interface conviviale tout en offrant une expérience d'émulation complète et éducative.

ArchitecteInterne.java :

Cœur logique du simulateur

ArchitecteInterne.java est le cœur logique du simulateur Motorola 6809. Il est responsable de la simulation de l'architecture interne du microprocesseur, incluant la gestion des registres comme les accumulateurs et les pointeurs, l'état du processeur et l'exécution des instructions.

Principales fonctionnalités :

- ✓ Simuler les registres internes du microprocesseur (PC, A, B, X, Y, U, S, etc.).
- ✓ Interpréter et exécuter les instructions de l'ensemble d'instructions Motorola 6809.
- ✓ Gérer les drapeaux de condition liés aux opérations arithmétiques et logiques.
- ✓ Mettre à jour dynamiquement l'état du processeur lors de l'exécution pas à pas.

Interactions avec les autres composants :

- ✓ Reçoit des instructions envoyées par Editeur.java pour les interpréter et les exécuter.
- ✓ Effectue des opérations de lecture et d'écriture dans la mémoire RAM et la ROM.\
- ✓ Met à jour en temps réem l'affichage de l'état interne du processeur pour l'interface utilisateur.

RAM.java :