



Sous la supervision de
Mr. Benalla.

RAPPORT MOT06809



REALISATION PAR ASKA
Abdelmounim Mouadili
Safaa Hifdi
Kaoutar Mezouahi
Aya darouich

OBJECTIF

L'objectif de ce projet est de créer un simulateur interactif pour le processeur Motorola 6809 à l'aide du langage de programmation Java. Ceci est réalisé grâce à une conception orientée objet et une implémentation efficace en Java. Le but est de mieux comprendre l'architecture du processeur et de simuler les instructions d'exécution. Les résultats de ce projet montrent une simulation fonctionnelle du processeur Motorola 6809.

ETAPES DE REALISATION



ÉTAPE 1:

Installer Moto6806 pour se familiariser avec le résultat final à atteindre.



ÉTAPE 2

Recherche de ressources pour soutenir la création de la partie frontale et arrière du projet.



ÉTAPE 3

Mise en place d'un plan de travail pour assurer une bonne gestion des tâches et une position optimale.

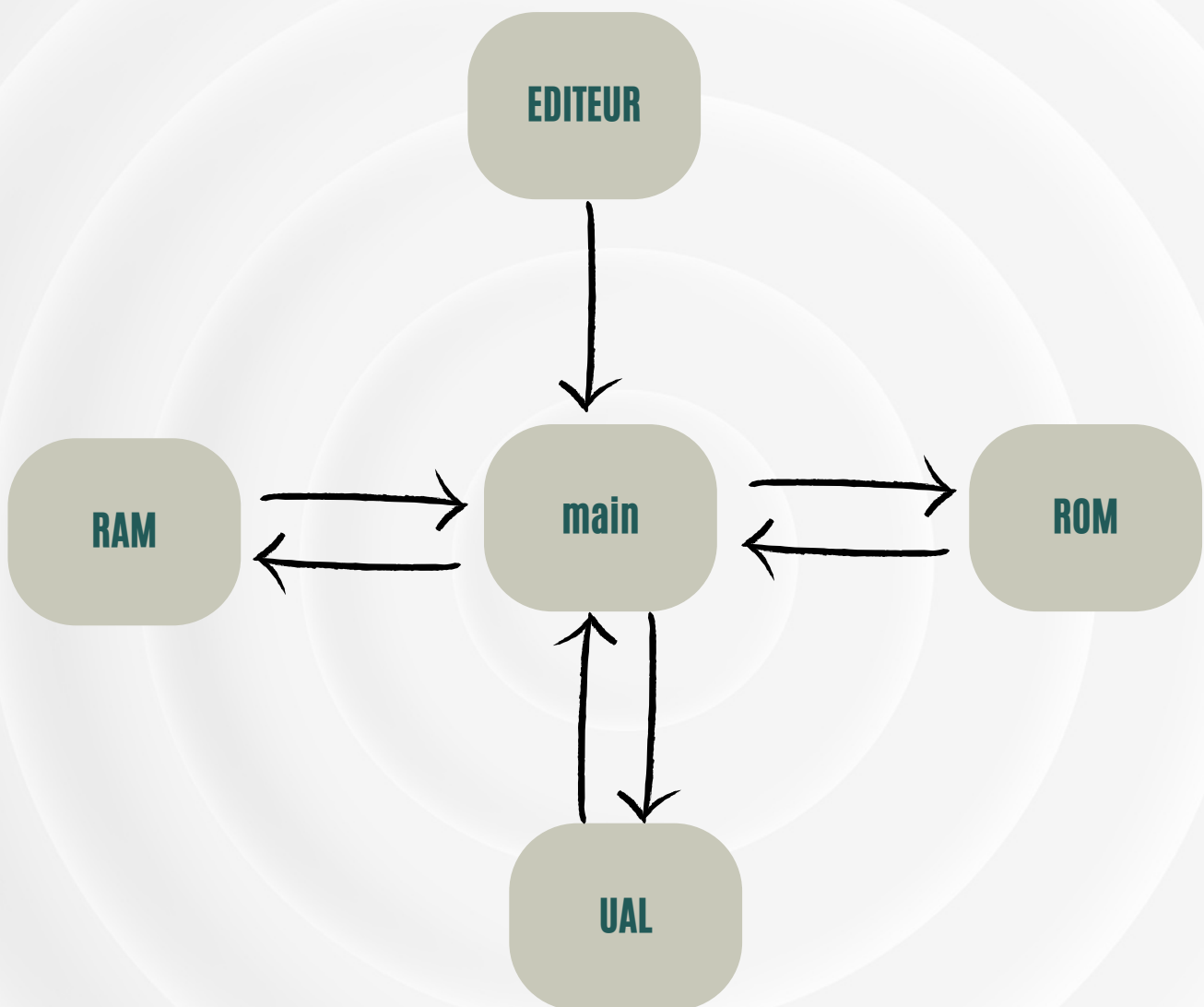


ETAPE 4

la partie application

LA CONCEPTION

La conception du simulateur repose sur une approche orientée objet. Les composants principaux sont articulés dans un schéma de classe, illustrant les relations entre ces entités.



ALGORITHMES

CREATION DES CLASSES

- Élaboration d'une classe recevant les instructions écrites par l'utilisateur.
- Mise en place d'une petite base de données dans une classe dédiée pour enregistrer les instructions et les traductions.
- Conception d'une classe de traitement avec des méthodes spécifiques pour différentes formes d'adressage:
 1. Adressage immédiat (LD, ADD, SUB, ABD, OR).
 2. Adressage inhérent (INC, DEC, CLR, ABX, ASL, LSL, LSR, ASR, NOP, SWI).
 3. Adressage direct (LD, ST, ADD, SUB, AND, OR).
 4. Adressage étendu (LD, ST, ADD, SUB, AND, OR).
 5. Adressage étendu direct (ST, LD).
 6. Adressage indexé (LEA).
- Création de classes pour la mémoire RAM et ROM.
- Mise en place d'une classe d'affichage pour visualiser le contenu des registres pendant l'exécution.
- La classe principale "MAIN" assurera la liaison entre tous les composants.

ALGORITHMES

LES MODES D ADRESSAGE:

Les Étapes pour les Modes d'Adressage **Immédiat, Direct, Indexé, Étendu Direct et Indirect:**

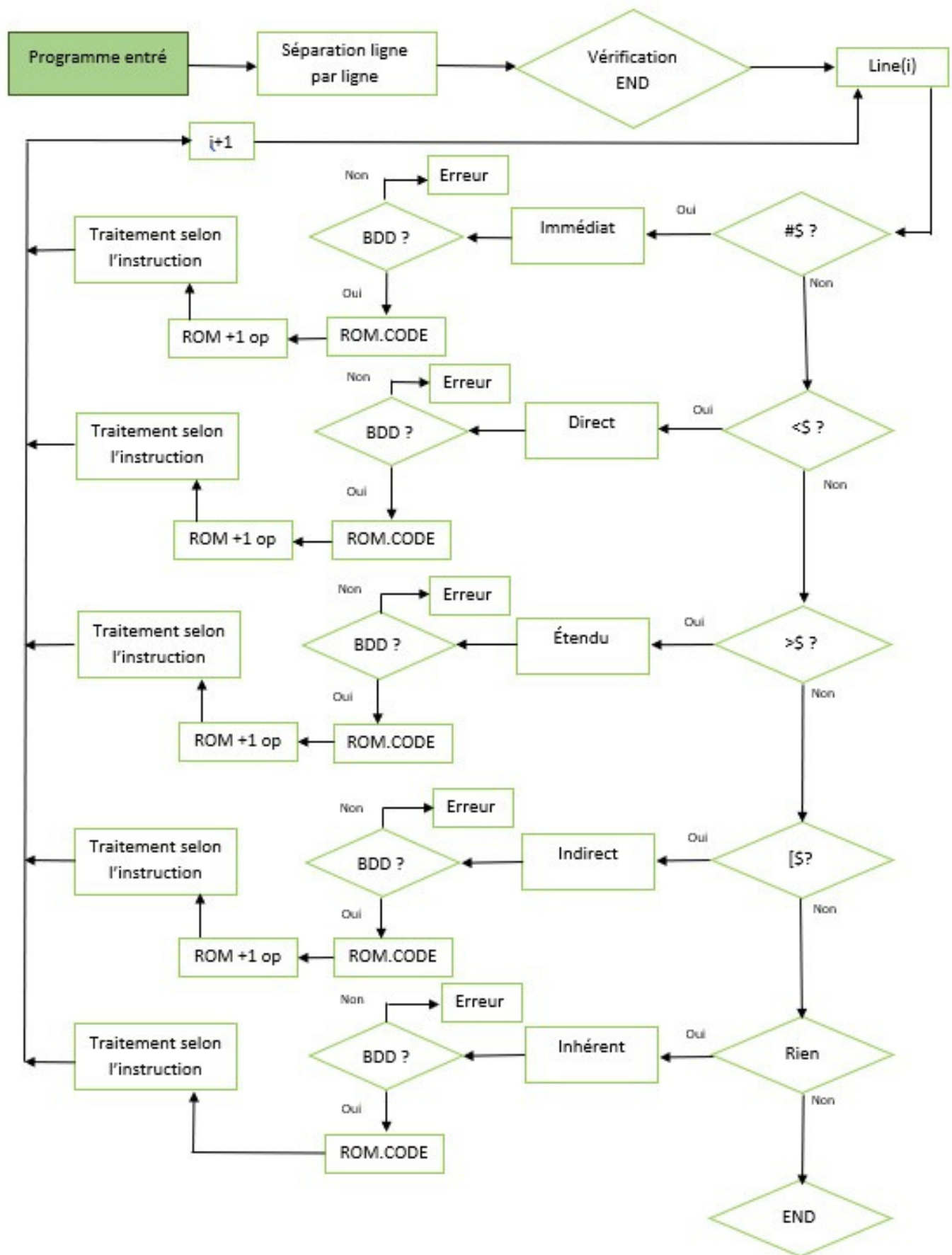
1. Stockez l'instruction dans un tableau en séparant les parties par des espaces.
2. Vérifiez que le contenu n'est pas nul, sinon affichez un message d'erreur.
3. Extraire le début de l'instruction et le comparez avec les instructions et les registres disponibles.
4. Traitez l'instruction selon le type d'adressage et le type de registre.
5. Passez à l'instruction suivante.

Les Étapes pour le Mode d'Adressage **Inhérent:**

1. Stockez l'instruction dans un tableau.
2. Traitez l'instruction selon le type d'adressage et le type de registre.
3. Passez à l'instruction suivante.

ALGORITHMES

COMPILATION



PEEK SUR LES INTERFACES

MOTO6809

Fichier Fenêtres Aide

RESET

Architecture

PC FC00

S 0000 U 0000

A 00

B 00

DP 00 00000000

EFHINZVC

X 0000 Y 0000

UAL

RAM		ROM	
0000	00	FC00	FF
0001	00	FC01	FF
0002	00	FC02	FF
0003	00	FC03	FF
0004	00	FC04	FF
0005	00	FC05	FF
0006	00	FC06	FF
0007	00	FC07	FF
0008	00	FC08	FF
0009	00	FC09	FF
000A	00	FC0A	FF
000B	00	FC0B	FF
000C	00	FC0C	FF
000D	00	FC0D	FF

Editeur

Mise à jour

Programme

TESTS

MOTO6809

Fichier Fenêtres Aide

RESET

Architecture

PC **FC03**

SWI

S **0000** U **0000**

A **FF**

B **FF**

DP **00** **00001000**

EFHINZVC

X **0000** Y **0000**

UAL

RAM		ROM	
000D	00	FC00	12
000E	00	FC01	86
000F	00	FC02	B7
0010	FF	FC03	00
0011	00	FC04	10
0012	00	FC05	F6
0013	00	FC06	00
0014	00	FC07	10
Programme		3F	
FC00	NOP	3F	
FC01	LDA #\$FF	FF	
FC03	STA >\$0010	FF	
FC04	LDB >\$0010	FF	
FC07	SWI	FF	
FC0A	END		

Editeur

Mise à jour

```
NOP
LDA #$FF
STA >$0010
LDB >$0010
SWI
END
```

DIFFICULTES RENCONTREES

- La principale difficulté rencontrée était liée à l'implémentation. Certains comportements spécifiques liés aux instructions du Motorola 6809 demandaient une compréhension approfondie et une analyse minutieuse. De plus, la familiarisation avec le GUI Builder sur NetBeans nous a posé des défis initiaux, mais nous avons réussi à nous adapter après un certain temps d'expérimentation
- La gestion du temps a été une autre difficulté significative, notamment en jonglant entre la préparation des examens et la réalisation de ce projet. Les priorités parfois concurrentes ont nécessité une planification pour garantir un progrès continu dans le projet tout en répondant aux exigences académiques.

CONCLUSION

En résumé, au début de ce projet, nous avons trouvé l'idée de créer un simulateur du processeur Motorola 6809 en Java un défi intimidant. Cependant, au fur et à mesure que nous avons commencé cette aventure technologique, le projet est devenu bien plus que cela : il est devenu notre propre petit monde, notre petit bijou.

Les premières étapes ont été marquées par une certaine inquiétude et des obstacles qui semblaient insurmontables. Le GUI Builder de NetBeans était un domaine inconnu qui nous déconcertait, mais chaque problème nous incitait à en apprendre davantage. Ce projet est devenu un processus d'apprentissage continu, un processus où chaque obstacle était une opportunité de grandir.

BIBLIOGRAPHIE

[HTTPS://WWW.MADDES.NET/M6809PM/SECTIONS.HTM](https://www.madde.net/m6809pm/sections.htm)
[HTTPS://WWW.GEEKSFORGEEEKS.ORG/](https://www.geeksforgeeks.org/)
[HTTPS://WAYTOLEARNX.COM/](https://waytolearnx.com/)
[HTTPS://WWW.W3SCHOOLS.COM/JAVA](https://www.w3schools.com/java)
[HTTPS://WWW.JAVATPOINT.COM/](https://www.javatpoint.com/)
[HTTPS://DOCS.ORACLE.COM/JAVASE/TUTORIAL/UISWING/](https://docs.oracle.com/javase/tutorial/uiswing/)
[HTTPS://STACKOVERFLOW.COM/](https://stackoverflow.com/)
[HTTPS://GITHUB.COM/](https://github.com/)
[HTTPS://WWW.YOUTUBE.COM/](https://www.youtube.com/)
[HTTPS://WWW.GURU99.COM/](https://www.guru99.com/)