

PRÉSENTATION

DU PROJET SIMULATEUR

MOTOROLA 6809

ENCADREE PAR MR HICHAM BENALLA

REALISE PAR : BOUTAYNA KHALIDI – MOUAD ELMOURABIR

Objectif du projet :

L'objectif est de simuler le fonctionnement du microprocesseur Motorola 6809, d'exécuter des instructions assembleur et d'observer en temps réel l'évolution des registres, de la mémoire et des flags.

FRONTEND

Interface
d'architecture
interne

Motorola 6809 - Simulator | Emulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x00
B (8-bit)	0x00
D (A:B)	0x0000
X (16-bit)	0x0000
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x0000
DP	00
CC	0x50

Code Assembleur 6809

```
; Exemple
LDA #$10
ADD A #$05
STA $20
```

Espace d'éditeur

MÉMOIRE - LECTURE SEULE (0x0000 - 0xFFFF)

	Aller à l'adresse:	0000	Aller	▲ Haut	▼ Bas
00A0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
00B0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
00C0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
00D0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
00E0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
00F0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0100	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0110	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0120	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0130	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0140	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0150	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0160	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
0170	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		

La zone
memoire

Contrôle

1. Modifier code ci-dessus, puis : **COMPILER**

2. Exécution : **STEP** **RUN** **PAUSE** **RESET**

Add Breakpoint Clear Breakpoints



BACKEND

```
▼ src\motorola
  ▼ addressing
    └ AddressingMode.java
    └ AddressingModeType.java
  > assembler
  ▼ cpu
    └ CPU.java
    └ Debugger.java
  > decoder
  > gui
  > instructions
  > memory
```

La classe **Memory** simule la **mémoire principale du microprocesseur Motorola 6809**, en gérant les opérations de lecture et d'écriture des données et des instructions dans un espace mémoire de 64 Ko.

CPU :

La classe **CPU** modélise le **cœur du microprocesseur Motorola 6809**, en gérant les registres, les flags et l'exécution des instructions selon le cycle *Fetch – Decode – Execute*.

Debugger :

La classe **Debugger** permet de **contrôler et analyser l'exécution du programme**, en offrant des fonctionnalités telles que l'exécution pas à pas, les points d'arrêt et l'observation de l'état du CPU.

les instructions utilisent les **modes d'adressage** pour déterminer comment accéder à leurs opérandes. Ces modes définissent la manière dont les adresses ou les valeurs sont calculées avant l'exécution, assurant une interaction correcte entre le CPU, les registres et la mémoire.

Les mode d'adressage utilise

Immédiat : l'opérande est une valeur constante intégrée directement dans l'instruction.

Direct : l'adresse est calculée à partir du registre **DP** et d'un offset sur 8 bits.

Étendu : l'instruction contient une adresse mémoire complète sur 16 bits.

•**Indexé** : l'adresse est calculée à partir d'un registre d'index et d'un offset ou d'un mécanisme d'indexation.

LES INSTRUCTIONS :

Instructions de chargement et de stockage

- LDA, LDB, LDD** : chargent une valeur dans les accumulateurs A, B ou D
- LDX, LDY, LDU, LDS** : chargent une valeur dans les registres d'index ou de pile
- STA, STB, STD** : stockent le contenu des accumulateurs en mémoire
- STX, STY, STU, STS** : stockent le contenu des registres en mémoire

2. Instructions arithmétiques

- ADD, ADDA, ADDB, ADDD** : addition
- SUB, SUBA, SUBB, SUBD** : soustraction
- INC, DEC** : incrémentation et décrémentation
- MUL** : multiplication (A × B → D)

3. Instructions logiques

- AND, ANDA, ANDB** : ET logique
- OR, ORA, ORB** : OU logique
- EOR** : OU exclusif
- COM** : complément (NOT)
- CLR** : mise à zéro

. Instructions de pile

- PSHS / PULS** : empiler / dépiler sur la pile système
- PSHU / PULU** : empiler / dépiler sur la pile utilisateur

Instructions de transfert

- TFR** : transfert entre registres
- EXG** : échange entre registres

File Help

Registres

A (8-bit)	0x00
B (8-bit)	0x00
D (A:B)	0x0000
X (16-bit)	0x0000
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x0000
DP	00
CC	0x50

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
N	Z	V	C
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Contrôle

1. Modifier code ci-dessus, puis : **COMPILEUR**2. Exécution : **STEP** **RUN** **PAUSE** **RESET****Add Breakpoint****Clear Breakpoints**

Code Assembleur 6809

```
; Exemple
LDA #$10
ADDA #$05
STA $20
```

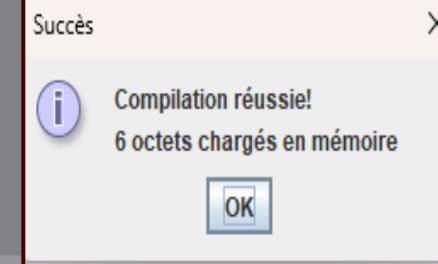
Code ready to compile

MÉMOIRE - LECTURE SEULE (0x0000 - 0xFFFF)

Aller à l'adresse: 0000

Aller**▲ Haut****▼ Bas**

00A0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0150	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0160	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0170	:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00



PARTIE TEST
LÀ KHIẾC IE21

Registres	
A (8-bit)	0x00
B (8-bit)	0x01
D (A:B)	0x0001
X (16-bit)	0xCFE1
Y (16-bit)	0x03E8

Code Assembleur 680

```
; Exemple
LDA #$10;
LDB #10 ;
LDD #0001 ;
LDX #$CFE1 ;
LDY #1000 ;
```

Aller à l'adresse: 0000		Aller	▲ Haut	▼ Bas
Adresse				
0000	: 86 10 C6 0A CC 00 01 8E CF E1 10 CE 03 E8 00 00 			
0010	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 			
0020	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 			
0030	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 			
0040	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 			
0050	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 			
0060	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 			

Registres	
A (8-bit)	0x10
B (8-bit)	0x00
D (A:B)	0x1000
X (16-bit)	0x0000
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x0004
DP	01

Code Assembleur

```
; Exemple
LDA #$10;
STA <$20;
```

✓ Compilation réussie -

MÉMOIRE - LEC

0120 : 10 00 00 00 00 00

Registres		Code Assemblé
A (8-bit)	0x13	; Exemple
B (8-bit)	0x02	LDA #\$10;
D (A:B)	0x1302	ADDA #\$03 ;
X (16-bit)	0x0000	LDB #\$05;
Y (16-bit)	0x0000	SUBB #\$03 ;
U (16-bit)	0x0000	

Registres		Code Assemb.
A (8-bit)	0x10	; Exemple
B (8-bit)	0x10	LDA #\$10;
D (A:B)	0x1010	STA \$20 ;
X (16-bit)	0x0000	INC \$20 ;
Y (16-bit)	0x0000	LDB #\$10;
U (16-bit)	0x0000	STB \$3F ;
		DEC \$3F ;

Registres

A (8-bit)	0x00
B (8-bit)	0x04
D (A:B)	0x0004
X (16-bit)	0x0000

Code Assem

```
; Exemple  
LDA #$2;  
LDB #$2;  
MUL ;
```

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010

Code Assembleur 6809

```
; Exemple  
LDA #$10;  
LDB #5 ;  
TFR A,B ;
```

Registres

A (8-bit)	0x05
B (8-bit)	0x10
D (A:B)	0x0510
X (16-bit)	0x0000
Y (16-bit)	0x0000

Code Assem

```
; Exemple  
LDA #$10;  
LDB #5 ;  
EXG A,B;
```

Direct

File Help

Registres

A (8-bit)	0x00
B (8-bit)	0x00
D (A:B)	0x0000
X (16-bit)	0x0000
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x0000
DP	0F
CC	0x50

Code Assembleur 6809

```
LDB #$10 ;
STB < $30;

LDA <$30 ;
```

✓ Compilation réussie - 6 octets chargés

MÉMOIRE - LECTURE SEULE

Aller à l'adresse: F00 Aller

Adresse : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

0F00	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F10	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F20	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F30	: 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F40	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x00
B (8-bit)	0x10
D (A:B)	0x0010
X (16-bit)	0x0000
Y (16-bit)	0x1000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x0007
DP	0F
CC	0x50

Code Assembleur 68

```
LDB #$10 ;
STB < $30;

LDY <$30 ;
```

✓ Compilation réussie - 7 octets

MÉMOIRE - LECTURE

Aller à l'adresse: F00

Adresse : 00 01 02 03 04 05

0F00	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F10	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F20	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F30	: 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F40	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

ETENDU (direct)

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010
X (16-bit)	0x0000
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x0007
DP	0F
CC	0x50

Code Assembleur

```
LDB #$10 ;
STB < $30;
LDA > $0F30 ;
```

✓ Compilation réussie - 7 octets

MÉMOIRE - LECTURE

Aller à l'adresse: F00

Adresse :	00 01 02 03 04 05
OF00	: 00 00 00 00 00 00
OF10	: 00 00 00 00 00 00
OF20	: 00 00 00 00 00 00
OF30	: 10 00 00 00 00 00
OF40	: 00 00 00 00 00 00
OF50	: 00 00 00 00 00 00

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x00
B (8-bit)	0x10
D (A:B)	0x0010
X (16-bit)	0x0000
Y (16-bit)	0x1000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x0008
DP	0F
CC	0x50

Code Assembleur 6809

```
LDB #$10 ;
STB < $30;
LDY > $0F30 ;
```

✓ Compilation réussie - 8 octets

MÉMOIRE - LECTURE

Aller à l'adresse: F00

Adresse :	00 01 02 03 04 05
OF00	: 00 00 00 00 00 00
OF10	: 00 00 00 00 00 00
OF20	: 00 00 00 00 00 00
OF30	: 10 00 00 00 00 00
OF40	: 00 00 00 00 00 00

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

The screenshot shows a 6809 assembly language debugger interface. The top menu bar includes "File" and "Help".

Registers

A (8-bit)	0x05
B (8-bit)	0x05
D (A:B)	0x0505
X (16-bit)	0x0000
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000D
DP	00
CC	0x50

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
N	Z	V	C
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Code Assembleur 6809

```
LDB #$10 ;
STB $30;

LDA #$5 ;
STA >$1000 ;

LDB [$0030] ;
```

✓ Compilation réussie - 13 octets

MÉMOIRE - LECTURE

Aller à l'adresse:

Adresse	00	01	02	03	04	05	
0000	:	C6	10	D7	30	86	05
0010	:	00	00	00	00	00	00
0020	:	00	00	00	00	00	00
0030	:	10	00	00	00	00	00
0040	:	00	00	00	00	00	00
0050	:	00	00	00	00	00	00
0060	:	00	00	00	00	00	00
0070	:	00	00	00	00	00	00
0080	:	00	00	00	00	00	00
0090	:	00	00	00	00	00	00
00A0	:	00	00	00	00	00	00

ETENDU (indirect)

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x05
B (8-bit)	0x10
D (A:B)	0x0510
X (16-bit)	0x0000
Y (16-bit)	0x0500
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000E
DP	00
CC	0x50

Code Assembleur

```
LDB #$10 ;
STB $30;

LDA #$5 ;
STA >$1000 ;

LDY [$0030] ;
```

✓ Compilation réussie - 14 octets

MÉMOIRE - LECTURE

Aller à l'adresse: 000

Adresse	:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	:	C6	10	D7	30	86	00	00	00	00	00	00	00	00	00	00	00
0010	:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	:	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050	:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

INDEXED

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010
X (16-bit)	0x0100
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000B
DP	00
CC	0x50

Code Assembleur

```
LDA #$10 ;
STA >$120 ;
LDX #$100;
LDB $20,X;
```

✓ Compilation réussie - 11

MÉMOIRE - LECTURE

Aller à l'adresse: 000

00A0	:	00 00 00 00 0
00B0	:	00 00 00 00 0
00C0	:	00 00 00 00 0
00D0	:	00 00 00 00 0
00E0	:	00 00 00 00 0
00F0	:	00 00 00 00 0
0100	:	00 00 00 00 0
0110	:	00 00 00 00 0
0120	:	10 00 00 00 0
0130	:	00 00 00 00 0
0140	:	00 00 00 00 0

FLAGS CC

E	F	H	I
○	●	○	●
N	Z	V	C
○	○	○	○

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010
X (16-bit)	0x0101
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000A
DP	00
CC	0x50

Code Assembleur

```
LDA #$10 ;
STA >$100 ;
LDX #$100;
LDB ,X+;
```

✓ Compilation réussie - 11

MÉMOIRE - LECTURE

Aller à l'adresse: 000

00A0	:	00 00 00 00 0
00B0	:	00 00 00 00 0
00C0	:	00 00 00 00 0
00D0	:	00 00 00 00 0
00E0	:	00 00 00 00 0
00F0	:	00 00 00 00 0
0100	:	10 00 00 00 0
0110	:	00 00 00 00 0
0120	:	10 00 00 00 0
0130	:	00 00 00 00 0
0140	:	00 00 00 00 0
0150	:	00 00 00 00 0

FLAGS CC

E	F	H	I
○	●	○	●
N	Z	V	C
○	○	○	○

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010
X (16-bit)	0x0100
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000A
DP	00
CC	0x50

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
N	Z	V	C
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Code Assembleur

```
LDA #$10 ;
STA >$100 ;
LDX #$101;
LDB , -X;
✓ Compilation réussie - 10
```

MÉMOIRE - LECTURE

Aller à l'adresse: 000

0070	:	00 00 00 00
0080	:	00 00 00 00
0090	:	00 00 00 00
00A0	:	00 00 00 00
00B0	:	00 00 00 00
00C0	:	00 00 00 00
00D0	:	00 00 00 00
00E0	:	00 00 00 00
00F0	:	00 00 00 00
0100	:	10 00 00 00

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010
X (16-bit)	0x0103
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000A
DP	00
CC	0x50

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
N	Z	V	C
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Code Assembleur

```
LDA #$10 ;
STA >$100 ;
LDX #$103;
LDB -3,X;
✓ Compilation réussie -
```

MÉMOIRE - LECTURE

Aller à l'adresse: 000

00A0	:	00 00 00 00
00B0	:	00 00 00 00
00C0	:	00 00 00 00
00D0	:	00 00 00 00
00E0	:	00 00 00 00
00F0	:	00 00 00 00
0100	:	10 00 00 00
0110	:	00 00 00 00
0120	:	00 00 00 00
0130	:	00 00 00 00
0140	:	00 00 00 00
0150	:	00 00 00 00

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010
X (16-bit)	0x0100
Y (16-bit)	0x0000
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000A
DP	00
CC	0x50

FLAGS CC

E	F	H	I
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
N	Z	V	C
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Code Assembleur 6809

```
LDA #$10 ;
STA >$110 ;

LDX #$100;
LDB A,X;
```

✓ Compilation réussie - 10 octets chargés

MÉMOIRE - LECTURE SEULE

Aller à l'adresse: 000

00A0	:	00 00 00 00 00 00 00 00
00B0	:	00 00 00 00 00 00 00 00
00C0	:	00 00 00 00 00 00 00 00
00D0	:	00 00 00 00 00 00 00 00
00E0	:	00 00 00 00 00 00 00 00
00F0	:	00 00 00 00 00 00 00 00
0100	:	00 00 00 00 00 00 00 00
0110	:	10 00 00 00 00 00 00 00
0120	:	00 00 00 00 00 00 00 00
0130	:	00 00 00 00 00 00 00 00

Motorola 6809 - Simulator | Émulateur du Microprocesseur 6809

File Help

Registres

A (8-bit)	0x10
B (8-bit)	0x10
D (A:B)	0x1010
X (16-bit)	0x0100
Y (16-bit)	0x0110
U (16-bit)	0x0000
S (16-bit)	0x0000
PC	0x000C
DP	00
CC	0x50

Code Assembleur 6809

```
; Exemple
LDA #$10 ;
STA >$110 ;

LDX #$100;
LEAY A,X;
```

✓ Compilation réussie - 10 octets chargés

MÉMOIRE - LECTURE SEULE

Aller à l'adresse: 000

00A0	:	00 00 00 00
------	---	-------------