**Exercice 2 – Décodage d’instructions (désassemblage)**

**Travail demandé**

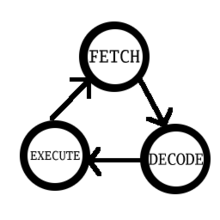
**Identifiez les instructions dans la mémoire puis complétez les registres et les tableaux.**

*Ressource à utiliser :* ***Annexe - Architecture du 68HC11 CPU et instructions utilisées.***

***Objectif :*** *Désassembler un programme (*[*reverse engineering*](https://www.scip.ch/en/?labs.20211202)*).*

***Première instruction (s’exécute à la mise sous tension du processeur)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **1** |  |  |  |



|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| 0012 | 01 |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 | **?** |

**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
| **0000** |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

**68HC11 CPU (partiel)**

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| Z=0 |

**Interface Bus (Adresses et données)**

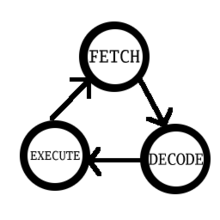
Horloge

**Unité de contrôle**

***Deuxième instruction***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **2** |  |  |  |

|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| 0012 | 01 |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 |  |



**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
|  |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| Z=0 |

**Interface Bus (Adresses et données)**

Horloge

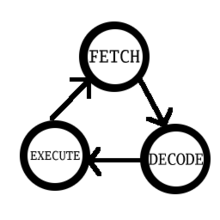
**68HC11 CPU (partiel)**

**Unité de contrôle**

***Troisième instruction***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **3** |  |  |  |

|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| 0012 | 01 |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 |  |



**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
|  |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| Z=0 |

**Interface Bus (Adresses et données)**

Horloge

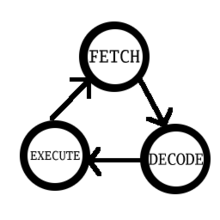
**68HC11 CPU (partiel)**

**Unité de contrôle**

***Quatrième instruction (exécutée si Z = 1)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **4** |  |  |  |

|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| 0012 | 01 |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 |  |



**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
|  |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| Z=0 |

**Interface Bus (Adresses et données)**

Horloge

**68HC11 CPU (partiel)**

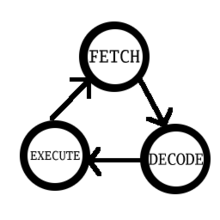
**Unité de contrôle**

***Cinquième instruction***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **5** |  |  |  |

**Mémoire**

|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| 0012 | 01 |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 |  |



**PC** *(Compteur programme 16bits)*

|  |
| --- |
|  |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| Z=0 |

**Interface Bus (Adresses et données)**

Horloge

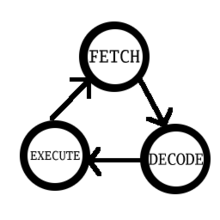
**68HC11 CPU (partiel)**

**Unité de contrôle**

***Sixième instruction***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **6** |  |  |  |

|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| 0012 | 01 |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 |  |



**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
|  |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| Z=0 |

**Interface Bus (Adresses et données)**

Horloge

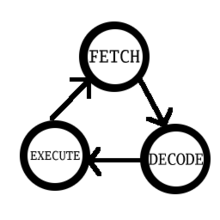
**68HC11 CPU (partiel)**

**Unité de contrôle**

***Septième instruction***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **7** |  |  |  |

|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| 0012 | 01 |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 |  |



**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
|  |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| Z=0 |

**Interface Bus (Adresses et données) **

Horloge

**68HC11 CPU (partiel)**

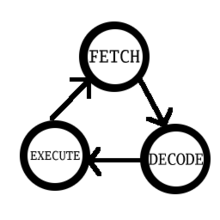
**Unité de contrôle**

***Huitième instruction (la fin du bloc de code est atteinte après cinq parcours de la boucle → vu dans la synthèse)***

+1

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction**  **n°** | **Adresse**  (Base 16) | **Opcode et opérande(s)**  (Base 16) | **Syntaxe assembleur** |
| **8** |  |  |  |

|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 | 18 |
| 0001 | CE |
| 0002 | 00 |
| 0003 | 20 |
| 0004 | 18 |
| 0005 | E6 |
| 0006 | 00 |
| 0007 | C1 |
| 0008 | 00 |
| 0009 | 27 |
| 000A | 07 |
| 000B | F7 |
| 000C | FB |
| 000D | 00 |
| 000E | 18 |
| 000F | 08 |
| 0010 | 20 |
| 0011 | F2 |
| **→ 0012** | **01** |
|  |  |
| 0020 | 48 |
| 0021 | 65 |
| 0022 | 6C |
| 0023 | 6C |
| 0024 | 6F |
| 0025 | 00 |
|  |  |
| FB00 |  |



**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
| **0012** |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
| **0001** |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
| **0025** | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  | **00** |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
| **Z=1** |

**Interface Bus (Adresses et données)**

Horloge

**68HC11 CPU (partiel)**

**Unité de contrôle**

**Synthèse : correspondance entre le code source et le code objet**

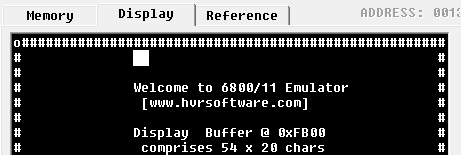
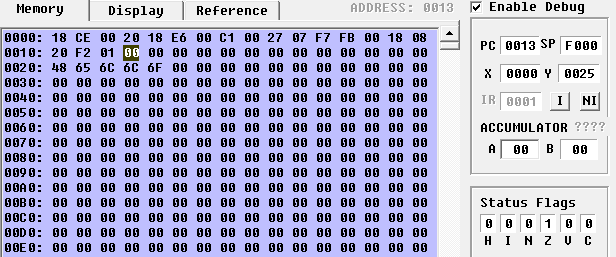


Figure 2 - Afficheur du simulateur

****Figure 2 - Code machine dans la mémoire du simulateur et état des

registres du CPU à la fin du programme

1. **Complétez le tableau ci-dessous à l’aide des résultats de l’étude précédente.**

Début

[[B](https://webge.fr/6800.html#A-reg)] [←](https://webge.fr/6800.html#transfer-desc) [[0](https://webge.fr/6800.html#data8-desc) + [[Y](https://webge.fr/6800.html#X-reg)]]

Fin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instruction**  **N°** | **Adresses16** | **Opcode16 et opérande16** | **[label] [opcode] [operand]** | **Opérations symboliques** |
|  |  |  | **.org $0** |  |
| **1** | **0000** | **18 CE 00 20** | **start ldy #msg** | [Y] ← 002016 |
| **2** | **0004** |  | **loop** | [[B](https://webge.fr/6800.html#A-reg)] [←](https://webge.fr/6800.html#transfer-desc) [[0](https://webge.fr/6800.html#data8-desc) + [[Y](https://webge.fr/6800.html#X-reg)]] |
| **3** |  |  | **cmpb #0** | [B] – 0 |
| **4** | **0009** |  | **beq fol** | (Z == 1) ? {[[PC](http://www.8bit-era.cz/6800.html#PC-reg)] ← [[PC](http://www.8bit-era.cz/6800.html#PC-reg)] + [7](http://www.8bit-era.cz/6800.html#disp-desc) + 2} |
| **5** |  |  | stab **$fb00** | [$FB00] ← [B] |
| **6** | **000E** |  |  | [Y] ← [Y] + 1 |
| **7** | **0010** |  | **bra loop** | {[[PC](http://www.8bit-era.cz/6800.html#PC-reg)] [←](http://www.8bit-era.cz/6800.html#transfer-desc) [[PC](http://www.8bit-era.cz/6800.html#PC-reg)] + [F2](http://www.8bit-era.cz/6800.html#disp-desc)16 + 2} |
| **8** | **0012** | **01** | **fol nop** |  |
|  |  |  |  |  |
|  |  |  | **.org $20** |  |
|  | **0020** | **48** **65 6C 6C 6F 00** | **msg .str " ? "** |  |

**2. Complétez l’algorigramme**

**3. Décodez le message msg à l’aide de la table des codes ASCII**

\_\_msg ="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Calculez l’opérande disp dans les sauts**

Calcul d’un saut : **[[PC](http://www.8bit-era.cz/6800.html" \l "PC-reg)arrivée]**[**←**](http://www.8bit-era.cz/6800.html#transfer-desc)**[[PC](http://www.8bit-era.cz/6800.html" \l "PC-reg)départ] + [disp](http://www.8bit-era.cz/6800.html" \l "disp-desc) + 2**

**PC : Registre Compteur Programme**

**disp** : Déplacement d’adresse signé sur 8 bits

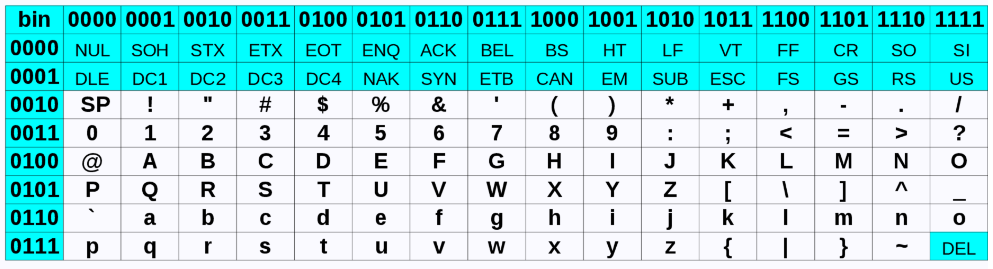
* **beq fol**

\_\_\_[**PC**](http://www.8bit-era.cz/6800.html#PC-reg)**arrivée16 =** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **bra loop**

\_\_\_[**PC**](http://www.8bit-era.cz/6800.html#PC-reg)**arrivée16 =** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Codes ASCII (‘A’ est codé par 4116)***



**Annexe - *Annexe - Architecture du 68HC11 CPU et instructions utilisées***

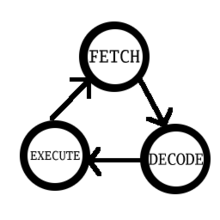


Motorola [MC68HC11](https://www.nxp.com/docs/en/reference-manual/M68HC11RM.pdf) (1984)

Nous allons dérouler une séquence d’**instructions** simples de type **langage machine** sur un processeur **MC6811**. Les instructions sont composées d’**un ou deux octets** représentant le **code de l’opération** (opcode) à effectuer, éventuellement suivi d’**un ou deux octets** appelé(s) **opérande(s)** de l’opération.

|  |  |
| --- | --- |
| FFFF  FFFE | Reset |
| FFFD  FFFC | NMI |
| FFFB  FFFA | SWI |
| FFF9  FFF8 | IRQ |
|  |  |
| FFF3  FFF2 | Souris |
| FFF1  FFF0 | Clavier |
|  |  |
| **FF38**  **FB00** | **Écran**  **54x20** |
|  |  |
| F000  **↓** | Pile |
| **↑**  0000 |  |

**Plan d’adressage du simulateur SDK6800/6811**



|  |  |
| --- | --- |
| **Adresse**  (Base 16) | **Donnée**  (Base 16) |
| 0000 |  |
| 0001 |  |
| 0002 |  |
| 0003 |  |
| 0004 |  |
| 0005 |  |
| 0006 |  |
| 0007 |  |
| 0008 |  |
| 0009 |  |
| 000A |  |
| 000B |  |
| 000C |  |
| 000D |  |
| 000E |  |
| 000F |  |
| 0010 |  |
| 0011 |  |
| 0012 |  |
|  |  |
| 0020 |  |
| 0021 |  |
|  |  |
| FB00 |  |

**Mémoire**

**PC** *(Compteur programme 16bits)*

|  |
| --- |
| **0000** |

**IR** *(Registre d’instruction 16bits)*

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **X** *(registre d’index 16bits)* | |
|  | |
| **Y** *(registre d’index 16bits)* | |
|  | |
| **A**  *(Accumulateur 8bits)* | **B**  *(Accumulateur 8bits)* |
|  |  |

**UAL**

**SR** *(Registre d’état 8bits)*

|  |
| --- |
|  |

**Interface Bus (Adresses et données)**

Horloge

**68HC11 CPU (partiel)**

**Unité de contrôle**

**Cycle d’exécution des instructions**

Pour exécuter un programme, l'**unité de contrôle** réalise de façon continue la boucle appelée *cycle d'exécution d'une instruction*.

1. **FETCH** (chargement). L'*unité de contrôle* récupère l’**opcode** à exécuter à l'adresse mémoire indiquée par le registre **PC** (*program* *counter*) et la stocke dans son registre **IR** (*instruction register*).
2. **DECODE** (décodage). L’**opcode** contenue dans **IR** est décodée afin de déduire l’action à réaliser et sur quelles données (**opérandes**).
3. **EXECUTE** (exécution). L'instruction est exécutée, soit par l'**UAL** (*unité arithmétique et logique*), s'il s'agit d'une opération **arithmétique** ou **logique**, soit par l'**unité de contrôle**, s'il s'agit d'une **opération de branchement** qui va donc modifier la valeur du registre **PC**.

Le tableau ci-dessous est extrait de la source : [68HC11 Instruction Set](https://www.cs.uaf.edu/2007/fall/cs441/proj1notes/sawyer/inst.html). Il permet de connaître l’organisation d’une instruction, son rôle et sa syntaxe en assembleur 68xx.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opcode**  (Base 16) | **Nombre d’octets (Opcode + Opérande(s))** | **Syntaxe assembleur de l’instruction** | **Opération symbolique** | **Remarque** |
| **27** | 2 | **BEQ** disp | (Z == 1) ? {[[PC](http://www.8bit-era.cz/6800.html#PC-reg)] [←](http://www.8bit-era.cz/6800.html#transfer-desc) [[PC](http://www.8bit-era.cz/6800.html#PC-reg)] + [disp](http://www.8bit-era.cz/6800.html" \l "disp-desc) + 2} | **B**ranch if **E**qual to zéro |
| **20** | 2 | **BRA** disp | {[[PC](http://www.8bit-era.cz/6800.html#PC-reg)] [←](http://www.8bit-era.cz/6800.html#transfer-desc) [[PC](http://www.8bit-era.cz/6800.html#PC-reg)] + [disp](http://www.8bit-era.cz/6800.html" \l "disp-desc) + 2} | **BR**anch **A**lways |
| **C1** | 2 | **CMPB** #data8 | [B] – data8 | **C**om**P**are with **B** |
| **18 08** | 2 | **INY** | [Y] ← [Y] + 1 | **IN**crement Index Register **Y** |
| **18 E6** | 3 | **LDA**[**B**](http://www.8bit-era.cz/6800.html#A-reg) #[data8](http://www.8bit-era.cz/6800.html#data8-desc),Y | [[B](https://webge.fr/6800.html#A-reg)] [←](https://webge.fr/6800.html#transfer-desc) [[data8](https://webge.fr/6800.html#data8-desc) + [[Y](https://webge.fr/6800.html#X-reg)]] | **L**oad **A**ccumulator **B** |
| **18 CE** | 4 | **LDY** #[data16](http://www.8bit-era.cz/6800.html#data16-desc) | [Y] ← data16 | **L**oad**D** Index Register **Y** |
| **01** | 1 | **NOP** |  | **NO** o**P**eration |
| **F7** | 3 | **STAB** addr16 | [addr16] ← [B] | **ST**ore **A**ccumulator **B** |

**←** : affectation (la donnée ou l’adresse est transférée dans la direction de la flèche) **[...]** : contenu de ...

**$**: la valeur qui suit est en hexadécimal **disp** : Déplacement d’adresse signé sur 8 bits

**#** : la valeur qui suit est une donnée (en l’absence de # c’est une adresse) **addr16** : adresse codée sur 16bits

**data8** : donnée codée sur 8bits