

SYSTEMES NUMERIQUES TP N 2

Environnement logiciel d'une carte d'évaluation 68HC11

31/03/2011

Table des matières

Question 1	3
Question 2	3
Question 3	3
II. ECRITURE D'UNE DONNEE DANS LA MEMOIRE	5
Question 4	5
Question 5	6
Question 6	6
Question 7	8
Question 9	9
Question 10	9

I. MISE EN ROUTE ET PREMIERS TESTS

Question 1

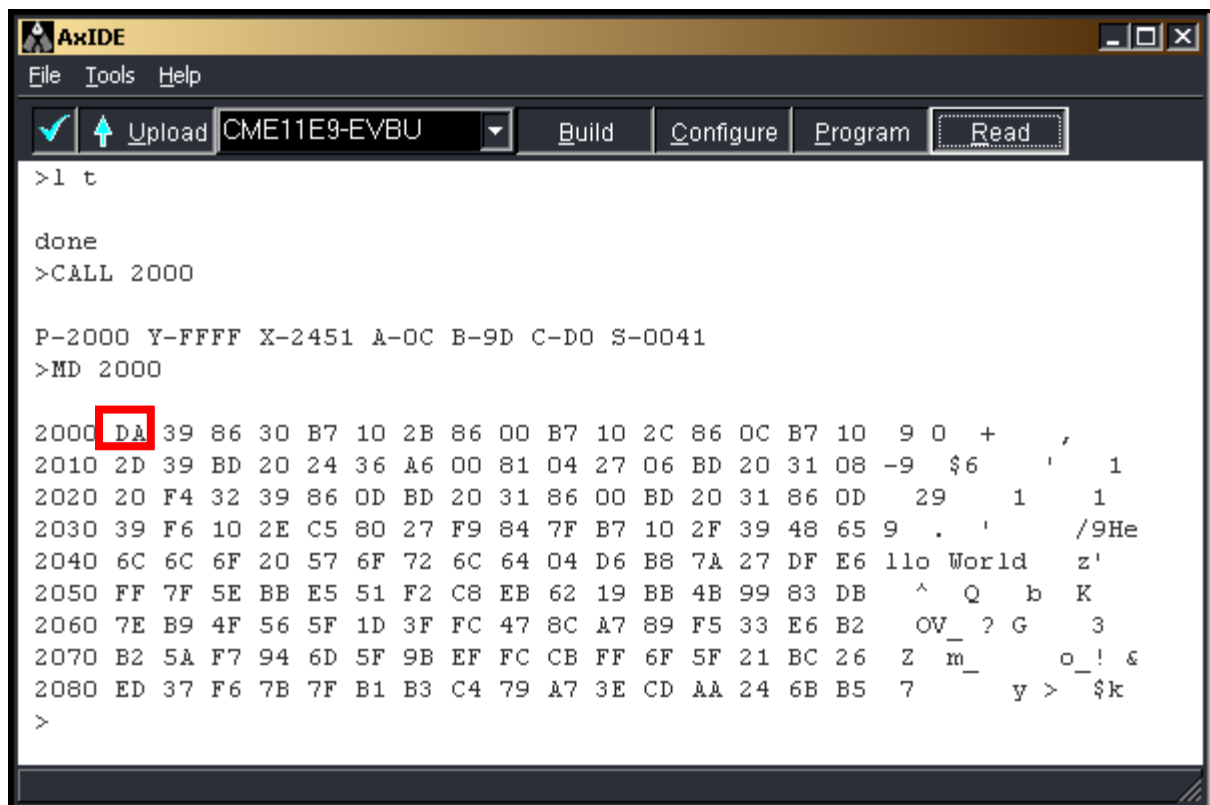
Les noms des deux fichiers générés sont hello.lst et HELLO.S19

Question 2

Le programme commence à l'adresse \$2400 (ORG \$2400 dans le fichier source) donc CALL 2400 permet de commencer à exécuter les instructions à partir de l'adresse \$2400.

II. ECRITURE D'UNE DONNEE DANS LA MEMOIRE

Question 3



The screenshot shows the AxIDE software interface. The menu bar includes 'File', 'Tools', and 'Help'. Below the menu bar is a toolbar with buttons for 'Upload', 'Build', 'Configure', 'Program', and 'Read'. The 'Upload' button is highlighted with a green checkmark. The 'Program' button is selected, and the 'Read' button is highlighted with a red border. The main window displays the following text:

```
>l t
done
>CALL 2000

P-2000 Y-FFFF X-2451 A-OC B-9D C-DO S-0041
>MD 2000

2000 DA 39 86 30 B7 10 2B 86 00 B7 10 2C 86 0C B7 10 9 0 + ,
2010 2D 39 BD 20 24 36 A6 00 81 04 27 06 BD 20 31 08 -9 $6 ' 1
2020 20 F4 32 39 86 0D BD 20 31 86 00 BD 20 31 86 0D 29 1 1
2030 39 F6 10 2E C5 80 27 F9 84 7F B7 10 2F 39 48 65 9 . ' /9He
2040 6C 6C 6F 20 57 6F 72 6C 64 04 D6 B8 7A 27 DF E6 llo World z'
2050 FF 7F 5E BB E5 51 F2 C8 EB 62 19 BB 4B 99 83 DB ^ Q b K
2060 7E B9 4F 56 5F 1D 3F FC 47 8C A7 89 F5 33 E6 B2 OV_ ? G 3
2070 B2 5A F7 94 6D 5F 9B EF FC CB FF 6F 5F 21 BC 26 Z m_ o_! &
2080 ED 37 F6 7B 7F B1 B3 C4 79 A7 3E CD AA 24 6B B5 7 y > $k
>
```

Après exécution du programme, on constate que DA est bien écrit à l'adresse \$2000.

```

AxIDE
File Tools Help
[Checkmark] [Upload] CME11E9-EVBU [Build] [Configure] [Program] [Read]

BUFFALO 3.4 (ext) - Bit User Fast Friendly Aid to Logical Operation
>MD 2000

2000 63 B9 FD 13 A0 B3 2F 73 3B DF 2C B7 76 FA 37 EC c /s; , v 7
2010 FB FB 99 B9 F7 36 3C DD DC DD BD E7 B9 07 57 EE 6< W
2020 B3 DB B2 AC 0C DB 75 3F A2 BD AB 7B 3D DA FB 75 u? = u
2030 73 53 BD CF 73 7F F9 DF D9 32 4B 67 7B 7D 6E 98 sS s 2Kg n
2040 45 F5 0C 3E D2 DB 76 F3 93 77 D6 B8 7A 27 DF E6 E > v w z'
2050 BF 7F 5E BB E5 D1 F2 C8 CB 62 19 BB 5B 99 C1 DB ^ b [
2060 7A 39 4B 56 5F 1D 3F FC 47 8C A7 89 F5 3B E6 33 z9KV_ ? G ; 3
2070 B2 5A F7 94 4D 5F 9B FF FC C3 FB 6F 5F 21 BC 26 Z M_ o_! &
2080 ED 37 E6 7B 7F B1 B3 C4 79 A7 3E CD AA 24 7B A5 7 y > $
>

```

Après avoir éteint puis rallumer l'alimentation, à l'exécution de la commande MD 2000 on constate que les données présentes sont celles de la mémoire morte.

Dans le fichier q3.lst, les directives sont :

```

0041 2000          ORG    $2000
0042 2000 da       FCB    $DA

```

Dans le fichier Q3.S19, les directives sont :

```

S1232000DA398630B7102B8600B7102C860CB7102D39BD202436A60081042706BD20310824
S123202020F43239860DBD20318600BD2031860D39F6102EC58027F9847FB7102F394865A4
S10D20406C6C6F20576F726C64041F
S9030000FC

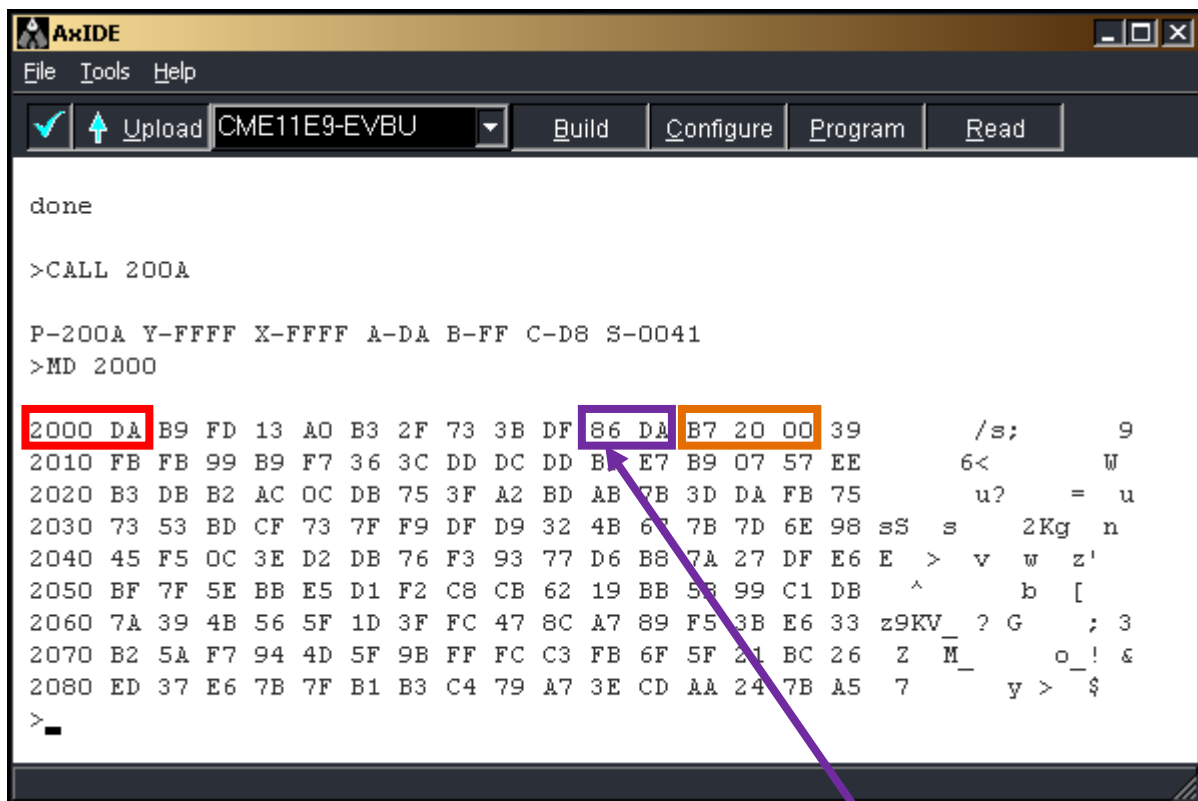
```

II. ECRITURE D'UNE DONNEE DANS LA MEMOIRE

Question 4

Commentaire du programme :

```
org    $2000  On se place à l'adresse $2000
val    rmb    1    On réserve un bloc d'un octet que l'on nomme « val »
org    $200A  On se place à l'adresse $200A
ldaa   #$DA    Charge la valeur numérique $DA dans l'accumulateur A
staa   val     Ecrit la valeur de l'accumulateur A à l'adresse « val »
```



On constate que la même fonction a été réalisée.

On retrouve le code machine des instructions à l'adresse prévue à savoir \$200A.

Il n'a pas été nécessaire de faire un RESET pour reprendre la main car nous avons pensé à laisser l'instruction de fin END.

Question 5

Dans le fichier q4.lst :

LIGNES	ADRESSE	CODE MACHINE	LABEL	INSTRUCTION	OPERANDE(S)
0007	2000			org	\$2000
0008	2000		val	rmb	1
0009					
0010	200a			org	\$200A
0011	200a	86 da		ldaa	#\$DA
0012	200c	b7 20 00		staa	val
0013	200f	39		RTS	

Le code binaire associé au code mnémonique staa est $\$86=10000110|_2$ et celui associé à ldaa est $\$B7=10110111|_2$

Commentaire de la trame S19 :

S109 **200A 86DA B72000 39** 5C
S9030000FC

On retrouve les instructions du programme précédé par l'adresse à laquelle est écrit le programme (\$200A). 39 est le code du RTS (RETURN).

Question 6

Commentaire du programme :

```
head    equ    $2000  « head » est la constante $2000
tail    equ    $2009  « tail » est la constante $2009

        org    $200A  On se place à l'adresse $200A
        ldaa   #$01   Charge la valeur numérique $01 dans l'accumulateur A
        ldx    #head  Charge la constante « head » dans le registre X

loop     staa   0,x     Début de la boucle « loop », Ecrit la valeur de l'accumulateur A à l'adresse
                     contenue dans le registre X
        inca           Incrémente la valeur dans l'accumulateur A
        inx           Incrémente la valeur (adresse) dans le registre X
        cpx     #tail  Compare le contenu du registre X avec la constante « tail »
        bne     loop  Retourne au début de « loop » si la valeur dans le registre X n'est pas 0
```

```

>l t
done
>CALL 200A

P-200A Y-FFFF X-2009 A-0A B-FF C-D4 S-0041
>MD 2000

2000 01 02 03 04 05 06 07 08 09 DF 86 01 CE 20 00 A7
2010 00 4C 08 8C 20 09 26 F7 39 DD BD E7 B9 07 57 EE L & 9 W
2020 B3 DB B2 AC 0C DB 75 3F A2 BD AB 7B 3D DA FB 75 u? = u
2030 73 53 BD CF 73 7F F9 DF D9 32 4B 67 7B 7D 6E 98 sS s 2Kg n
2040 45 F5 0C 3E D2 DB 76 F3 93 77 D6 B8 7A 27 DF E6 E > v w z'
2050 BF 7F 5E BB E5 D1 F2 C8 CB 62 19 BB 5B 99 C1 DB ^ b [
2060 7A 39 4B 56 5F 1D 3F FC 47 8C A7 89 F5 3B E6 33 z9KV_ ? G ; 3
2070 B2 5A F7 94 4D 5F 9B FF FC C3 FB 6F 5F 21 BC 26 Z M_ o_! &
2080 ED 37 E6 7B 7F B1 B3 C4 79 A7 3E CD AA 24 7B A5 7 y > $
>

```

L'adresse du début du programme est \$200A (l'instruction ORG \$200A permet de nous placer à cette adresse).

La mémoire entre head et tail soit \$2000 et \$2009 est un compte de 01 à 09 plus une donnée DF présente avant l'exécution du programme et qui n'a pas été écrasée. En effet, on quitte la boucle d'écriture lorsque le registre X vaut tail c'est-à-dire \$2009 donc on n'écrit pas à cette adresse.

Question 7

The screenshot shows the AxIDE software interface. At the top, there is a menu bar with 'File', 'Tools', and 'Help'. Below the menu bar is a toolbar with a green checkmark, an upload icon, a dropdown menu showing 'CME11E9-EVBU', and buttons for 'Build', 'Configure', 'Program', and 'Read'. The main window displays a memory dump with addresses 2050 to 2080 and their corresponding hexadecimal and ASCII values. Below the memory dump, there is a command prompt showing the execution of assembly instructions and the resulting state of registers and memory.

```

2050 BF 7F 5E BB E5 D1 F2 C8 CB 62 19 BB 5B 99 C1 DB ^ b [
2060 7A 39 4B 56 5F 1D 3F FC 47 8C A7 89 F5 3B E6 33 z9KV ? G ; 3
2070 B2 5A F7 94 4D 5F 9B FF FC C3 FB 6F 5F 21 BC 26 Z M _ o ! &
2080 ED 37 E6 7B 7F B1 B3 C4 79 A7 3E CD AA 24 7B A5 7 y > $

>T 200A
LDAA #$01          P-200C Y-FFFF X-2009 A-01 B-FF C-90 S-0041
LDX  #$2000        P-200F Y-FFFF X-2000 A-01 B-FF C-90 S-0041
STAA $00,X         P-2011 Y-FFFF X-2000 A-01 B-FF C-90 S-0041
INCA               P-2012 Y-FFFF X-2000 A-02 B-FF C-90 S-0041
INX                P-2013 Y-FFFF X-2001 A-02 B-FF C-90 S-0041
CPX  #$2009        P-2016 Y-FFFF X-2001 A-02 B-FF C-99 S-0041
BNE  $200F         P-200F Y-FFFF X-2001 A-02 B-FF C-99 S-0041
STAA $00,X         P-2011 Y-FFFF X-2001 A-02 B-FF C-91 S-0041
INCA               P-2012 Y-FFFF X-2001 A-03 B-FF C-91 S-0041
INX                P-2013 Y-FFFF X-2002 A-03 B-FF C-91 S-0041
>_

```

Ligne 1 : P est augmenté du nombre d'octets de l'instruction exécutée (ici 8601 donc 2 octets, d'où le passage de \$200A à \$200C). Ce sera le cas pour toutes les instructions suivantes, nous ne le répèterons donc pas. On constate aussi que la valeur numérique \$01 est mise dans l'accumulateur A (A-01).

Ligne 2 : On charge la valeur numérique \$2000 dans le registre X.

Ligne 3 : Les registres ne changent pas (sauf P) mais il y a écriture dans la RAM.

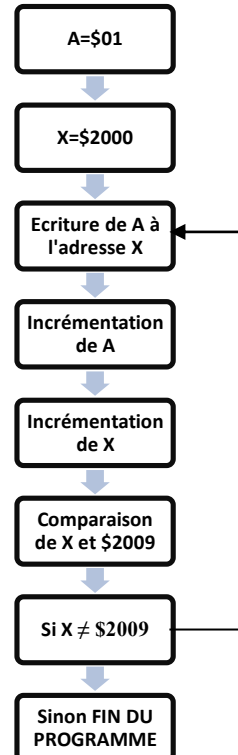
Ligne 4 : Incrémentation de l'accumulateur A, passage de A-01 à A-02.

Ligne 5 : Incrémentation du Registre X, passage de X-2000 à X-2001.

Ligne 6 : Comparaison du registre X avec la valeur numérique \$2009, entraînant un changement dans le registre C, passage de C-90 à C-99 (mise à jour des FLAGS)

Ligne 7 : Retour au début de la boucle en fonction du résultat de la comparaison. On constate donc que le registre P devient P-200F.

Répétition des Lignes 3 à 7 jusqu'à atteindre la condition d'arrêt.



III. COMMUNICATION AVEC UN PORT D'E/S

Question 9

La connexion par défaut de la sortie en PA4 est 1.

Commentaire du programme :

```

porta equ    $1000 *la constante porta vaut $1000
org      $2000 *On commence le programme à l'adresse $2000
ldaa     #$10  *On charge la constante numérique $10 dans l'accumulateur A
staa     porta * On écrit A à l'adresse porta
  
```

L'écriture fait passer PA4 de 0 à 1 d'où le changement d'état de la LED.

Question 10

```

porta      equ    $1000 * la constante porta vaut $1000
start      org    $2000 *On commence le programme à l'adresse $2000
           ldaa   porta *On charge les données de porta dans accumulateur A
           asla                   *On décale à gauche l'accumulateur A
           asla                   *On décale à gauche l'accumulateur A
           staa   porta *On écrit l'accumulateur A sur le porta
           bra    start *On boucle indéfiniment
           END
  
```