1. Objectifs :

Dans cette manipulation, on va découvrir la manipulation des ports d'entrées-sortie du 68HC11. Pour ce faire, on se servira toujours le simulateur THRSim11, qui met à la disposition de l'utilisateur un certain nombre de périphériques pour le test et l'évaluation de son programme. Ainsi, on passera par :

- L'utilisation des ports parallèles pour la lecture et l'affichage d'une donnée en mode parallèle.
- L'utilisation de l'interface de communication série (SCI) pour la réception et l'émission d'un caractère
- Mise en œuvre du concept de sous-programme pour la génération d'une séquence d'attente. Ainsi, la relation entre le temps de simulation affiché et les cycles d'horloge dans le simulateur THRSim11 sera discutée.

2. Ports parallèles :

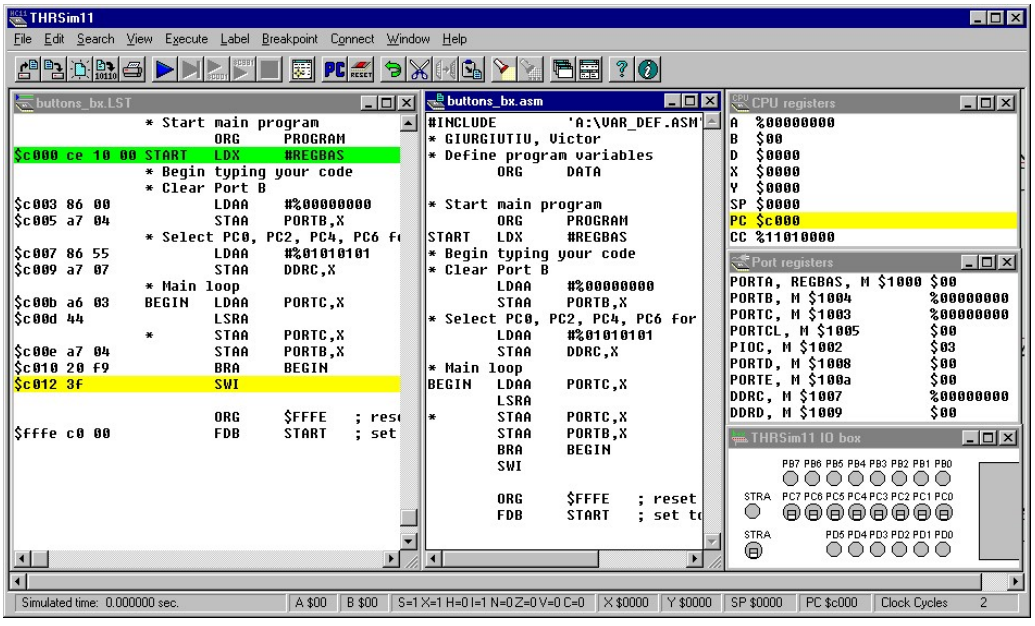
2.1. *Exercice 1 :*

Ecrire le programme pour lire, continuellement, les données présentes sur les broches du port C et les renvoie sur le port B.

Démarche :

- Ouvrir THRSim11 et Fermez la fenêtre Commandes.
- Ecrire le programme « Boutons.asm »
- Afficher les registres : *View, Registers, CPU Registrers* ;
- Ouvrir la boîte THRSim11 IO : *View, I/O Box* ;
- Assembler Boutons.asm.
- Définir un point d'arrêt à l'instruction « SWI » ;
- Définir des étiquettes standard : *Label, Set Standard Labels* ;
- Définir l'affichage de l'accumulateur A, PORTB et PORTC en binaire.
- Organiser les fenêtres pour un bénéfice maximum

On obtient une fenêtre semblable à celle-ci :



- Exécuter le programme : *Run* ;
- Actionner le bouton PC2 et noter l'effet.
- Refaire pour plusieurs boutons.

2.2. *Exercice 2 :*

Ecrire un programme pour afficher le chiffre hexadécimal (0-F) placé à la case mémoire d'adresse $0300 (data) sur un afficheur 7 segments.

- Quelle est la condition pour allumer un segment ?
- Donner la table de correspondance entre les codes binaires et les codes hexadécimaux de tous les chiffres qu'on désire allumer (0 à F).
- Donner la directive pour placer ces codes hexa dans une table commençant à l'adresse $0000.
- Ecrire le programme complet.

3. Interface de communication série (SCI) :

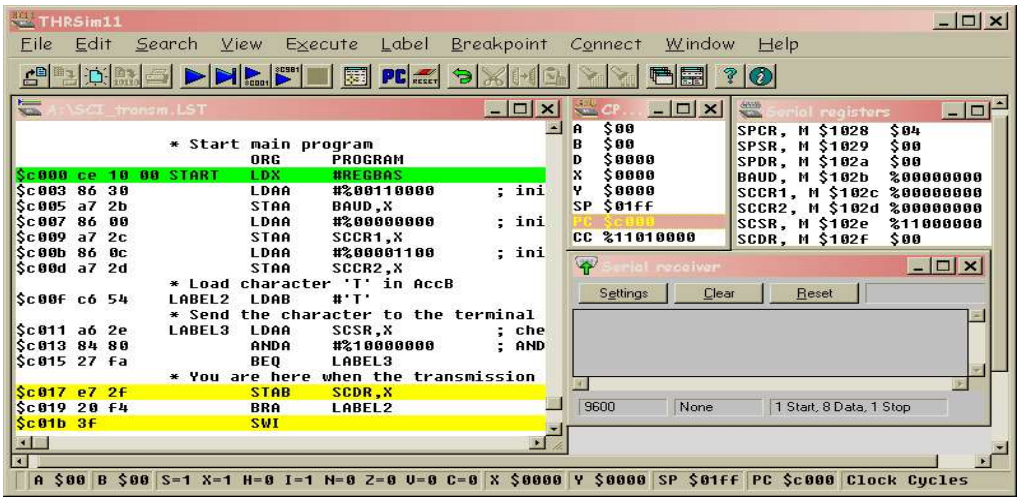
3.1. *Exercice 1 : Transmission d'un caractère*

Le principe de notre programme est le suivant :
Après l'initialisation du SCI, le caractère à transmettre (T par exemple) est chargé dans accB. Ensuite, l'état du drapeau TDRE (registre de données de transmission vide) est testé dans une boucle. Lorsque TDRE est vrai, le contenu de accB est stocké dans le SCDR (registre de données de communication série). Cette opération réinitialise automatiquement TDRE.
Une première ébauche du programme «Transmit.asm » peut être :

```
START       LDX     #REGBAS
        LDAA    #%00110000
        STAA    BAUD,X
        LDAA    #%00000000
        STAA    SCCR1,X
        LDAA    #%00001100
        STAA    SCCR2,X
LABEL2      LDAB    #'T'
* Emettre le caractère au terminal
LABEL3      LDAA    SCSR,X
        ANDA    #%10000000
        BEQ     LABEL3
* ici le reg. de transmission est vide
        STAB    SCDR,X
        BRA     LABEL2
        SWI
```

Démarche :

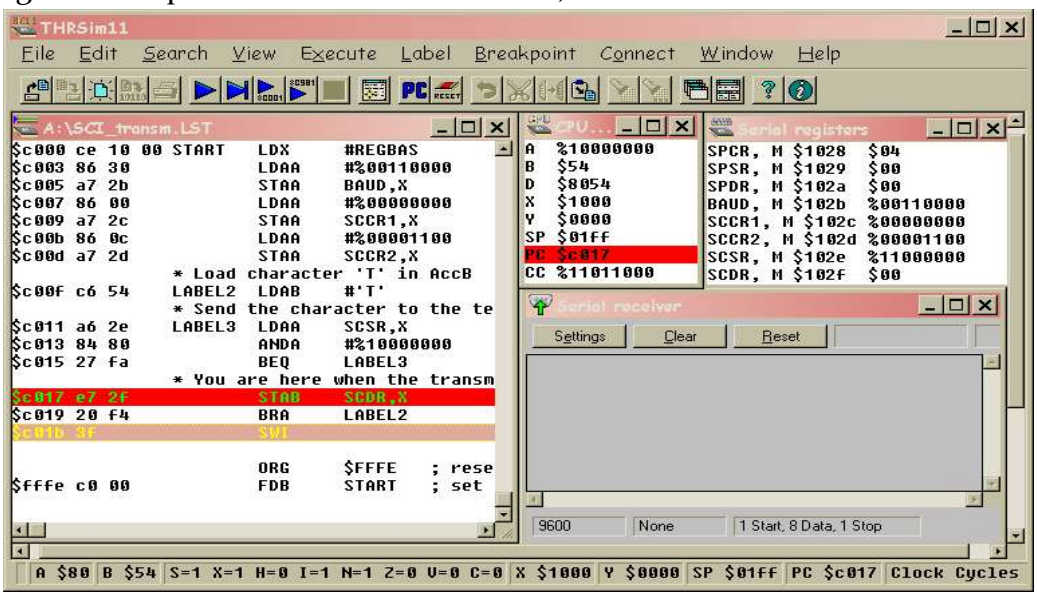
- Ouvrez THRSim11 et Fermez la fenêtre Commandes.
- Afficher les registres série : View, Registers, Serial ;
- Afficher le récepteur série : Viw, Serial Receiver ;
- Assemblez « Ttransmit.asm » ;
- Définir des points d'arrêt sur STAB SCDR, et sur SWI.
- Réinitialiser les registres : bouton RESET
- Définir des étiquettes standard : Label / Set Standard Labels.
- Définir l'affichage de A, BAUD, SCCR1, SCCR2 et SCSR en binaire.
- On obtient un écran ressemblant à ceci:



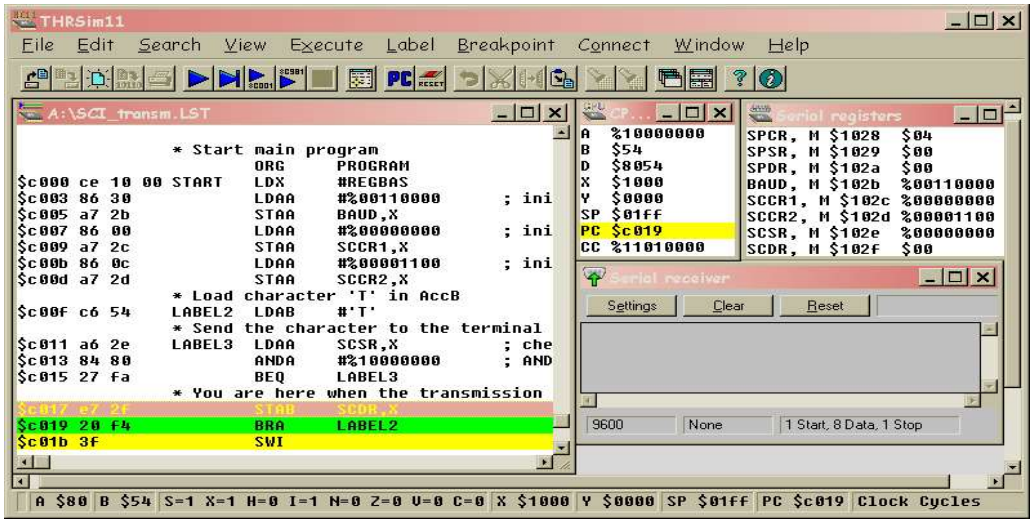
Que signifie la valeur de réinitialisation de SCSR ?

Exécution :

- Appuyer sur le bouton RUN.
- Indiquer la condition pour que le programme sort de la boucle
- Le programme accède à la ligne : STAB SCDR, X
- Cette ligne est un point d'arrêt. En ce moment, l'écran ressemble à ceci:



- Passez manuellement à la ligne suivante pour enregistre B dans SCDR.
- L'écran ressemble à ceci:



- Interpréter et conclure
- Appuyer sur RUN une autre fois
- Que constatez-vous ?
- Appuyer sur RUN 2 fois
- Conclusion

3.2. *Exercice 2 : Réception d'un caractère*

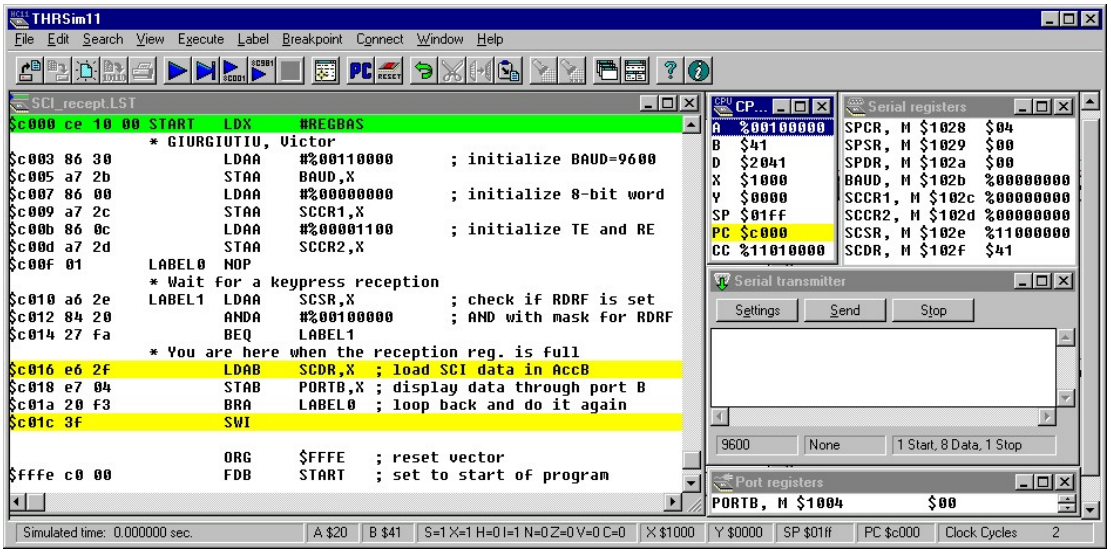
Voici une variante de programme pour la réception d'un caractère :

```
START       LDX     #REGBAS
        LDAA    #%00110000
        STAA    BAUD,X
        LDAA    #%00000000
        STAA    SCCR1,X
        LDAA    #%00001100
        STAA    SCCR2,X
LABEL0      NOP

LABEL1      LDAA    SCSR,X
        ANDA    #%00100000
        BEQ     LABEL1
        LDAB    SCDR,X
        STAB    PORTB,X
        BRA     LABEL0
        SWI
```

Démarche :

- Ouvrir THRSim11 et Fermez la fenêtre Commandes.
- Afficher les registres de ports et les registres série.
- Afficher émetteur série
- Assemblez « Recept.asm »
- Définir les points d'arrêt sur LDAB SCDR, X et SWI.
- Réinitialiser les registres.
- Définir des étiquettes standard.
- Définir l'affichage de BAUD, SCCR1, SCCR2, SCSR et l'accA sur binaire.
- Votre écran devrait ressembler à ceci :



- Exécuter le programme
- Justifier le déroulement en boucle du programme
- Tapez le caractère T dans la fenêtre de l'émetteur série et appuyez sur Send
- Interpréter le résultat
- Avancer manuellement et relever le changement des différents contenus
- Exécuter le programme pour plusieurs caractères