

EPREUVE ÉCRITE

Ministère de l'Éducation nationale
et de la Formation professionnelle

EXAMEN DE FIN D'ÉTUDES SECONDAIRES TECHNIQUES

Régime de la formation de technicien

Division électrotechnique

Section : Communication

BRANCHE : MICROÉLECTRONIQUE

SESSION : juin 2005

DATE :

DURÉE : 3h

Exercice 1 (15P = 3 + 3 + 9)

Le programme suivant en assembleur 8085 est donné :

```
ORG 7000h
MVI A, 82h
OUT 53h

SUB A
MOV B, A
OUT 50h

loop1 : IN 51h
        ANI 01
        JNZ loop1

loop2 : IN 51h
        ANI 01
        JZ  loop2

INR B
MOV A, B
OUT 50h

JMP loop1
END
```

Le programme ci-dessus représente un simple compteur binaire en avant.

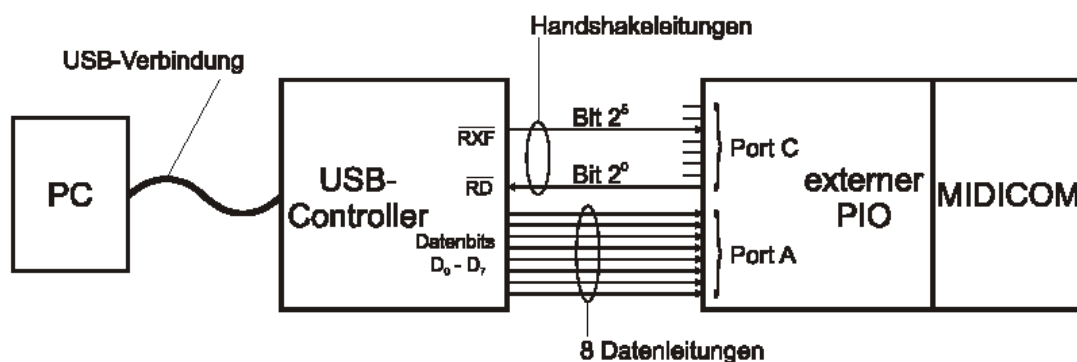
- Commentez chaque ligne du programme (→ feuille supplémentaire)
Expliquez en plus la fonction des deux boucles (loop1, loop2)
- Si on manipule l'interrupteur on remarque que le compteur réagit de la même façon que si on manipulait l'interrupteur *plusieurs* fois.
Expliquez pourquoi le programme montre ce comportement !

c) Complétez maintenant le programme de la manière que le compteur incrémente de 1 si on manipule l'interrupteur une seule fois.

- Elaborez l'organigramme du programme complet.
- Ecrivez le programme correspondant en assembleur 8085. (Il n'est pas nécessaire de commenter le programme une nouvelle fois !)

Exercice 2 (20P = 11 + 1 + 8)

Le MIDICOM doit recevoir des données à travers une interface USB d'un PC. Pour cette fin, un contrôleur spécifique est branché sur le MIDICOM. Les données de l'interface USB sont mémorisées temporairement dans le contrôleur pour être envoyées ensuite à travers une interface parallèle vers le MIDICOM (voir aussi schéma ci-dessous).



Le composant USB possède 8 lignes pour les données et 2 lignes pour le handshake. Toutes ces lignes sont reliées avec une carte PIO externe dont l'adresse de base est : **9Ch**.

Les lignes pour les données sont branchées sur le Port A et les deux lignes handshake sur le Port C selon le schéma.

Le PIO externe est utilisé *intégralement* en **mode 0**.

La présence de données dans le contrôleur USB est signalée par le niveau Low du signal /RXF. Il est impératif de vérifier l'état de cette ligne avant *chaque* lecture. Si le signal est à l'état '0', alors il y a au moins un octet dans la mémoire tampon du contrôleur.

Le signal reste à l'état '0' aussi longtemps qu'il y a des octets dans la mémoire.

Pour que le contrôleur USB sorte les données, la ligne /RD du contrôleur doit être mise au niveau Low (= état logique '0') par le MIDICOM. Après lecture d'un octet le MIDICOM remet la ligne /RD à l'état '1'. Maintenant le contrôleur peut mettre à disposition le prochain octet.

Toutes les données lues doivent être mémorisées dans un tableau dans la mémoire du MIDICOM. Ce tableau commence à l'adresse 9000h. La procédure continue jusqu'au moment où on a lu 128 octets. A ce moment le programme s'arrête avec l'instruction RST 1 et ceci indépendamment des octets se trouvant encore dans la mémoire tampon du contrôleur.

- a) Elaborez l'organigramme du programme.
- b) Déterminez le mot de contrôle du PIO externe.
- c) Ecrivez le code source pour le programme correspondant en assembleur 8085 et commentez si nécessaire les instructions.

Exercice 3 (7P)

Dans la mémoire du MIDICOM se trouve un tableau qui commence à l'adresse 9000h et qui contient 128 octets.

Maintenant les octets doivent être copiés dans un autre endroit de la mémoire. Durant cette opération, tous les octets représentant un octet zéro sont filtrés c-à-d ces octets **ne sont pas** copiés.

Le « nouveau » tableau est mémorisé à l'adresse 8000h de la mémoire. Il doit être clôturé par un octet zéro !

Ecrivez un programme pour réaliser cette opération de copie en utilisant les instructions 8085. Commentez si nécessaire les instructions.

Exercice 4 (18P = 5 + 3 + 2 + 8)

Des données se trouvent dans la mémoire du MIDICOM à partir de l'adresse 8000h. Ces données sont transmises sans fil avec un équipement de transmission spécifique. Avant la transmission les données sont transférées à partir du MIDICOM en sériel vers le module de transmission. Le bloc de données n'est jamais plus grand que 256 octets! Il **ne contient pas** d'octet zéro excepté l'octet à la fin du bloc. Cet octet marque la fin du bloc de données.

La mémoire tampon du module de communication peut contenir au maximum 32 octets. Pour envoyer la totalité des données mémorisées dans la mémoire tampon, le module utilise tout au plus 150ms. Uniquement après ce temps, il est prêt à recevoir d'autres données

Après avoir transféré 32 octets, il est donc impératif d'attendre 150ms.

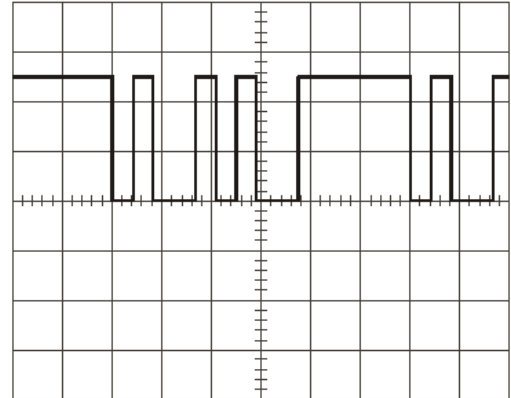
Le programme est terminé si le bloc entier est envoyé.

A. Généralités

- a) La communication sans fil est captée par un espion. Son oscilloscope montre la transmission d'un octet entier avec des **niveaux TTL**.

Déduisez à partir de l'enregistrement (*feuille supplémentaire*)

- la valeur de l'octet (caractère, 00 – FF)
- la parité choisie – paire ou impaire
- la vitesse de transmission
- les présélections de l'oscilloscope sont indiquées sur la feuille supplémentaire



- b) Expliquez l'interaction entre le registre de données et le registre de décalage qui tous les deux se retrouvent dans le transmetteur du SIO 8251. Quel est dans ce contexte le rôle des signaux TxRDY et TxEmpty.

B. Application

Un bloc de données qui est terminé par un octet zéro et qui se retrouve dans la mémoire du MIDICOM à partir de l'adresse 8000h, doit être envoyé au module de communication. Le module est branché sur une interface série.

La longueur du bloc est variable. Tous les octets de **01 (!)** à FF sont permis dans le bloc.

Les paramètres suivants sont prévus pour la communication.

Données :	8	Bit Stop :	1
Parité :	impaire	Vitesse de transmission :	2400 baud

- c) Déterminez tous les mots de contrôle pour un SIO externe du type 8251.
L'adresse de base du SIO est : **AAh**
La fréquence de l'horloge de transmission est : 153,6 kHz
- d) Ecrivez le code source en assembleur 8085 pour un programme qui transmet le bloc de données vers le module de communication et commentez si nécessaire les instructions. Le programme doit respecter le fait qu'après le transfert de 32 octets, une temporisation de 150 ms est nécessaire.
Pour réaliser la temporisation, l'utilisation du sous-programme interne est permise.

Questionnaire proposé par : DRUI Raoul
Commission d'examen : LTE TEC1
Questionnaire traduit par : GREISCH Jean-Paul
Commission d'examen : LTAM TEC1

Date : 20.5.2005
Signature :