



Fascicule des travaux pratiques

Circuits d'interfaces

8086



MTS - 86C

Proposé par : Ali Amidene

2009/2010

2009 / 2010

TP1 : PRISE EN MAIN DU KIT MTS-86C

1. Principaux composants

Le kit **MTS-86C** est composé d'un microprocesseur 16 bits (8086 d'Intel) auquel sont connectés tous les constituants principaux de l'informatique industrielle :

- RAM (64Ko) 2 x 62256
- EPROM Moniteur (64Ko) 2 x 27256
- EPROM Utilisateur (64Ko) 2 x 27256
- Interface parallèle 3 x 8255
- Interface série 2 x 8251
- Contrôleur de clavier 8279
- Contrôleur d'interruption 8259
- Timer programmable 8253
- ADC (8 bits – 8 entrées) ADC809
- DAC (8 bits) DAC0808

2. Cartographie de la mémoire du kit

L'espace mémoire utilisé est divisé en trois parties (Figure 1-1). Une zone pour le programme moniteur et exemples d'application, la seconde pour le programme utilisateur et les vecteurs d'interruptions et la troisième, une zone extensible, elle peut être du type ROM ou RAM.

FFFFFH	Monitor program	ROM
F8000H	Exercise program	ROM
F0000H	User memory	ROM, RAM
E0000H	OPEN	
10000H	User program	
00400H	Interrupt Vector Table	RAM
00000H		

Figure 1-1 : Cartographie de la mémoire du kit MTS-86C

3. Adressage des entrées-sorties

Le schéma de la Figure 1-2 présente le décodage d'adresses des principaux périphériques à étudier. En se référant au schéma du décodage, déterminer l'adresse de base de chaque composant (remplir le Tableau 1-1).

Tableau 1-1 : Adressage des Entrées-sorties

composants	Adresse de base en binaire								@ de base en Hexa.							
	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈		A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁
ADC809																
8255-3																
DAC0808																
Aff-7Seg																
8259																
8251-2																
8253																
8279																
8251-1																
8255-2																

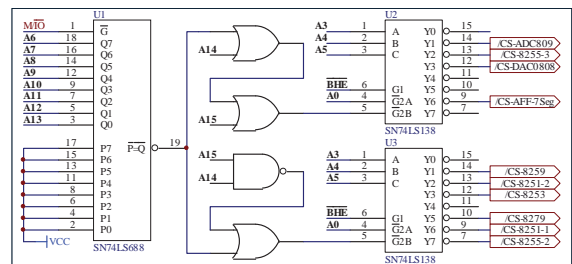


Figure 1-2 : Décodage d'adresses

4. Écriture et exécution d'un programme sur le KIT MTS-86C

4.1. Écriture du programme

Pour écrire un programme, il faut se servir d'un éditeur ASCII, tel que le programme **EDIT** du dos, puis l'enregistrer avec l'extension « .asm ». (exemple **MonPrg.asm**).
Taper le programme suivant et l'enregistrer sous le nom **Tp0.asm**

```

;===== Exemple d'un programme Assembleur =====
CNT3 EQU 3FD6H
PB3 EQU 3FD2H

CODE SEGMENT
ASSUME CS : CODE, DS : CODE
ORG 0
MOV SP, 4000H
MOV AL, 90H
MOV DX, CNT3
OUT DX, AL

MOV AL, 01H
MOV DX, PB3
@ : OUT DX, AL
MOV CX, 0A000H
LOOP $
ROL AL, 1
JMP @
CODE ENDS
END
    
```

4.2. Génération du fichier HEX

Le kit MTS-86C ne supporte que les fichiers de type **.hex** d'Intel. Pour créer ce fichier à partir du fichier source, il faut faire appel à des programmes de compilation, d'édition de lien et de conversion.

- Le programme «**MASM**» de Microsoft ou «**TASM**» de Borland, permettent de créer un fichier intermédiaire appelé « fichier Objet ». La commande suivante génère le fichier **.obj**. Tapez la commande suivante:
C:\ASM86> masm Tp0 ;
- Le programme «**LINK**» de Microsoft ou «**TLINK**» de Borland créent à partir du fichier **.obj** un fichier **.exe**. Tapez la commande suivante:
C:\ASM86> link Tp0 ;
- Le programme «**EXE2BIN**» convertit le fichier **.exe** en un fichier **.bin**. Tapez la commande suivante:
C:\ASM86> exe2bin Tp0

- Enfin le programme «**BIN2HEX**» convertit Le fichier **.bin** en un fichier **.hex**. Tapez la commande suivante:
C:\ASM86> bin2hex Tp0.bin

4.3. Chargement et exécution du programme

- Connectez le Kit au port série du PC. Pour charger le programme dans le kit ouvrez l'**hyper terminal** de Windows, avec la configuration suivante : 19200 bps, 8bits de données, pas de parité, sans contrôle de flux.
- Appuyez sur les boutons **RESET** puis **F1** du kit, pour initialiser la communication. Le moniteur répond avec un message d'invite et vous donne la main. Tapez **H** si vous voulez afficher toutes les commandes.
- Tapez **L** (pour Load) puis **↵**, le moniteur répond avec le message « Sending your file in INTEL HEX format. »
- Dans le menu **Transfert**, cliquez sur **Envoyer un fichier texte...**, puis sélectionnez le fichier **.HEX** à charger (**Tp0.hex** par exemple). Si tout va bien le moniteur vous répond « Thank you for your cooperation »
- Pour exécuter votre programme. Tapez **G**, puis **Y** pour confirmer l'adresse d'exécution proposée.

TP2 : INTERFACE PARALLELE

1. PPI – 8255A

1.1. Introduction

Le 8255 est un circuit programmable de 40 broches d'interface d'entrée/sortie parallèle qui a été conçu pour travailler avec les microprocesseurs de la famille INTEL. Il est formé par trois ports d'entrées-sorties, chaque port est de 8 bits qui peuvent être programmés en entrée ou en sortie avec trois modes différents (mode zéro, mode un et mode deux).

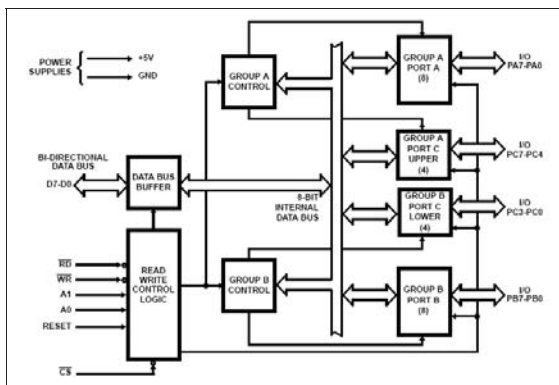


Figure 2-1: Schéma bloc du circuit 8255

Le schéma bloc de la Figure 2-1 montre bien qu'il est divisé en deux groupes :

- Groupe A : formé par le port A et le port C haut.
- Groupe B : formé par le port B et le port C bas.

Le registre de données (Data buffer Bus) assure la liaison entre le bus de données extérieur et le registre de contrôle ainsi que les ports d'entrées/sorties.

La sélection du 8255A se fait par l'intermédiaire de la broche /CS (qui est en général fournie par une logique de décodage)

Le bus d'adresse du 8255A est formé essentiellement par deux broches (A0 et A1) qui permettent de sélectionner les ports ainsi que le registre de contrôle comme le montre le Tableau 2-1.

/CS	A1	A0	Sélection
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Registre de Contrôle
1	x	x	8255 n'est pas sélectionné

Tableau 2-1 : sélection des ports

1.2. Programmation du 8255A

On peut programmer le 8255A selon trois modes :

- Mode 0 : Entrée/sortie de base.
- Mode 1 : Entrée sortie échantillonnée.
- Mode 2 : Bus bidirectionnel.

Le format ainsi que le choix des modes se fait à partir du mot de contrôle suivant :

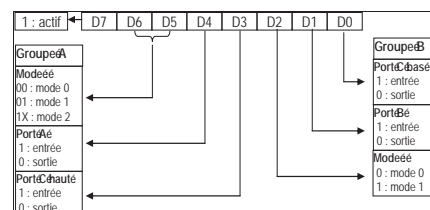


Figure 2-2: mot de commande

- En mode zéro les ports du 8255A peuvent être programmés en entrée ou en sortie: 8 bits pour le port A, 8 bits pour le port B et le port C est formé de deux quartets (un quartet haut et un quartet bas). Exemple si on veut configurer le port A en sortie, port B en entrée, port C haut entrée et port C bas sortie le mot de commande est 8AH

- Les bascules du port C peuvent être mises à 1 ou à 0 individuellement, lorsque le bit 7 du mot de commande est égal à 0 (mode SET/RESET). Le mot de commande devient alors :

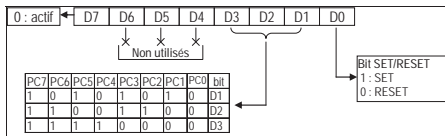


Figure 2-3 : mot de commande en mode SET/RESET

1.3. Manipulation

Le kit MTS-86C dispose de trois PPI 8255. Les ports des circuits 8255-1 et 8255-2 sont disponibles sur deux connecteurs d'extension ; Alors que les ports du circuit 8255-3 sont connectés aux périphériques suivants (Figure 2-4) :

- PORT A : connecté à des boutons poussoirs
- PORT B : connecté à des diodes LEDs
- PORT C : attaque de l'entrée d'un amplificateur audio (PC5), commande du timer 8253 (PC4), commande de l'amplificateur de température (PC6), détection de fin de conversion de l'ADC809 (PC0).

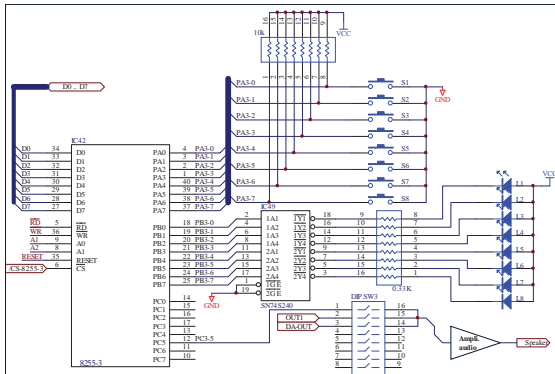


Figure 2-4 : Schéma de connexion du 8255-3

En admettant que l'adresse de base du 8255-3 est 3FD0H.

- Déterminer les adresses des registres du PPI 8255-3.
- Ecrire un programme assembleur permettant d'allumer la LED correspondant au bouton poussoir appuyé.
- En utilisant la commande SET/RESET ; écrire un programme qui génère sur la broche PC3-5 un signal carré.

TP3 : AFFICHEURS 7 SEGMENTS

1. Présentation

L'afficheur 7 segments, est constitué de 7 diodes LEDs disposées de manière à représenter les chiffres décimaux. L'afficheur du kit MTS-86C est un afficheur à anode commune ; il est connecté au bus de données par l'intermédiaire d'un latch 74LS373 actif par niveau comme le montre le schéma de la Figure 3-.

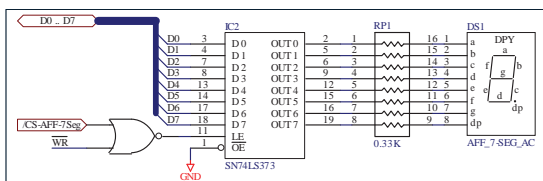


Figure 3-1 : Afficheur 7 segments

2. Manipulation

- Le circuit 74LS373 est un verrou (latch) composé de huit bascules D ayant une horloge commune active au niveau haut LE (Latch Enable).
 - Quel est le rôle de la broche /OE (Output Enable) ?
 - Quand le verrou 74LS373 devient transparent ?
- Le Tableau 3- présente les combinaisons nécessaires à appliquer aux cathodes de l'afficheur 7 segments pour visualiser les chiffres de 0 à 9.

Chiffres	dp	g	f	e	d	c	b	a	Val. en hexa
0	1	1	0	0	0	0	0	0	C0H
1	1	1	1	1	1	0	0	1	F9H
2	1	0	1	0	0	1	0	0	A4H
3	1	0	1	1	0	0	0	0	B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	92H
6	1	0	0	0	0	0	1	0	82H
7	1	1	1	1	1	0	0	0	F8H
8	1	0	0	0	0	0	0	0	80H
9	1	0	0	1	0	0	0	0	90H

Tableau 3-1 : Décodage décimal vers 7 segments

- Quel est le rôle des résistances insérées en séries avec les cathodes de l'afficheur.
- En admettant que l'adresse de sélection de l'afficheur est 3FF0H. Ecrire un programme assembleur qui permet d'afficher consécutivement en boucle les chiffres de 0 à 9.
- Les boutons poussoirs connectés au port A du 8255-3 sont numérotés de 1 à 8. Ecrire un programme assembleur qui lit d'une manière cyclique l'état des boutons poussoirs et affiche sur l'afficheur 7 segments le numéro du bouton appuyé. Si aucun bouton n'est appuyé on affiche 0.

TP4 : CONVERSION A-N ET N-A

1. Conversion Analogique – Numérique (CAN)

Dans le milieu industriel on est souvent appelé à traiter des grandeurs analogiques. Les signaux issus des capteurs doivent être convertis en équivalent numérique avant d'être traités. L'ADC809 présent sur la carte est un CAN 8 bits à 8 entrées multiplexées ; quatre parmi eux sont exploitées par le kit (Figure 4-):

- L'entrée AD-IN0 est connectée à un potentiomètre monté en diviseur de tension.
- L'entrée AD-IN1 est connectée à un photo transistor.
- L'entrée AD-IN2 est connectée à un capteur de température.
- L'entrée AD-IN3 est connectée à un microphone.

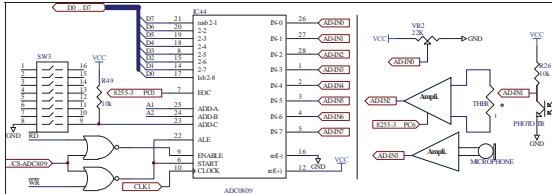


Figure 4-1 : Conversion analogique – numérique

- 1) En admettant que l'adresse de base de l'ADC809 est 3FC8H. Donner les adresses de sélection des entrées AD-IN0, AD-IN1, AD-IN2 et AD-IN3.
- 2) Ecrire un programme assembleur permettant de faire l'acquisition du signal appliqué à l'entrée AD-IN0 et d'afficher la valeur numérisée sur les diodes LEDs connectées au port B du 8255-3.

2. Conversion Numérique – Analogique (CNA)

Le kit est équipé aussi d'un DAC0808. Ce circuit est un convertisseur numérique – analogique 8 bits. Sa sortie est une sortie en courant, il faudra donc rajouter un convertisseur courant – tension externe (Figure 4-). La sortie DA-OUT est disponible à l'extrémité gauche du kit ou à l'entrée 3 du DIP SW3 (voir Figure 2-4).

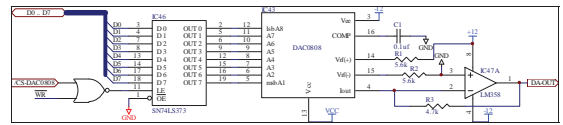


Figure 4-2 : Conversion numérique – analogique

L'adresse de sélection du DAC0808 est 3FD8H.

- 1) Ecrire le code du programme assembleur qui génère à la sortie DA-OUT un signal dent de scie. Visualiser la sortie sur un oscilloscope.
- 2) Ecrire le programme assembleur qui permet de faire l'acquisition du signal issu du microphone (AD-IN3) et de le restituer à la sortie du DAC0808.

TP5 : LE TIMER D'INTERVALLE PROGRAMMABLE 8253/8254

1. Présentation du PIT 8253/54

Les timers sont utilisés souvent pour la génération des signaux de synchronisation et de commande avec une grande précision. Le timer inséré dans le kit MTS-86C est le 8253 d'Intel. Ce circuit est constitué de trois circuits de comptage identiques de 16 bits, possédant chacun deux entrées CLK et GATE et une sortie OUT. Le compteur 0 est utilisé comme générateur d'horloge pour les circuits 8251, tant dis que la sortie du compteur 1 est connectée par l'intermédiaire du DIP SW3 à l'entrée d'un haut parleur ou à l'entrée IRI du contrôleur d'interruption 8259 (Figure 5-1). Le compteur 2 n'est pas exploité par le kit.

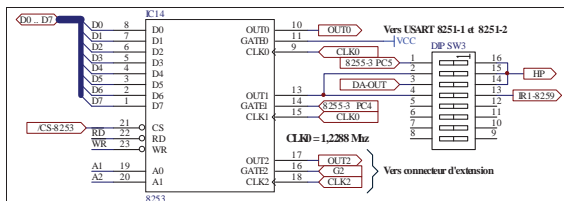


Figure 5-1 : Connexion du Timer 8253

2. Fonctionnement du PIT 8253/54

Le timer 8253/54 est sélectionné par un niveau bas sur l'entrée /CS. Lorsqu'il n'est pas sélectionné, les entrées du buffer de données sont en 3^{ème} état. Les lignes A0 et A1 sont utilisés pour la sélection de l'un de trois compteurs ou du registre de contrôle (Tableau 5-1).

CS	A1	A0	Sélection
0	0	0	Compteur 0
0	0	1	Compteur 1
0	1	0	Compteur 2
0	1	1	Registre de Contrôle
1	x	x	8253/54 n'est pas sélectionné

Tableau 5-1: Sélection des compteurs et du registre de contrôle

Chaque circuit de comptage possède un registre de contrôle, un registre de status, un registre compteur CR (CR0 et CR1) pour recevoir le compte initial, un élément compteur CE effectuant le décomptage mais non accessible par le microprocesseur et enfin un latch de sortie OL (OL0 et OL1) pour verrouiller le contenu du compteur et le rendre disponible pour la lecture. En plus, le 8253/54 possède six modes de fonctionnement. La Figure 5-2 présente l'architecture interne de 8253/54.

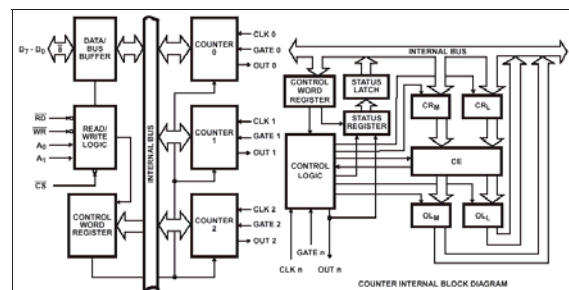


Figure 5-2: Architecture interne de 8253/54

3. Programmation du PIT 8253/54

3.1. Initialisation des compteurs

Après la mise sous tension, le 8253/54 se trouve dans un état indéfini, le mode de fonctionnement, le compte initial et l'état de sorties des compteurs ne sont pas définis. La procédure de programmation de chaque compteur consiste à écrire dans l'ordre le mot de commande dans le registre de contrôle ensuite le compte d'initialisation dans le registre CR. Le registre d'entrée CR est un registre 16 bits ce qui oblige à écrire la donnée en deux temps (poids faible d'abord).

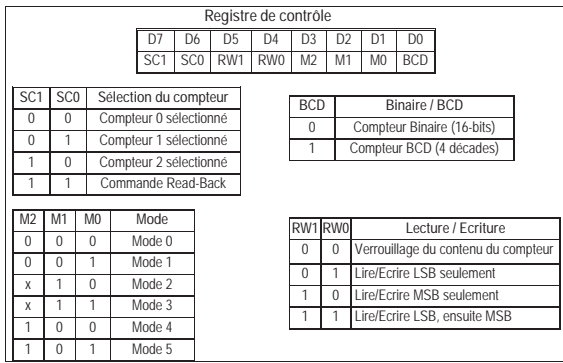


Figure 5-3: Format du mot de contrôle

3.2. Exemple d'initialisation

On veut programmer le compteur 2, pour générer un signal carré de fréquence $f_{OUT} = 2 \text{ KHz}$ sachant que $f_{CLK} = 1,2288 \text{ Mhz}$. On suppose que le timer 8253 est sélectionné par les lignes A2 et A1 du microprocesseur 8086 et que son adresse de base est 330H.

- l'adresse du compteur 2 est 334H et du registre de commande 336H.
- la valeur initiale du compteur N = $f_{CLK} / f_{OUT} \approx 614 = 0266\text{H}$.
- le signal carré est généré par le mode 3.

Ce qui donne le mot de commande : $1011x110b = B6\text{H}$ ou $BE\text{H}$.

```

MOV AL, B6H ; Compteur 2, mode 3, Binaire -> B6H
MOV DX, 336H ; Charger l'adresse du registre de contrôle
OUT DX, AL ; Envoi au registre contrôle
MOV AX, 0266H ; Charger le nombre diviseur (H pour HEX)
MOV DX, 334H ; Charger l'adresse du compteur 2
OUT DX, AL ; Envoi de l'octet LSB au compteur 2 (66H)
MOV AL, AH ; MSB dans AL
OUT DX, AL ; Envoi de l'octet MSB au compteur 2 (02H)

```

4. Modes de fonctionnement du PIT 8253/54

Le PIT 8253/54 possède six modes de fonctionnement permettant de satisfaire la plupart des applications industrielles tels que la génération des temporisations, des signaux de synchronisation, des signaux périodiques impulsionnels et carré, des signaux MLI par combinaison de deux modes...etc. L'entrée Gate est utilisée dans les modes 1 et 5 pour déclencher le décomptage ; alors que dans les autres modes pour bloquer le décomptage. La Figure 5-4 présente un résumé des différents modes.

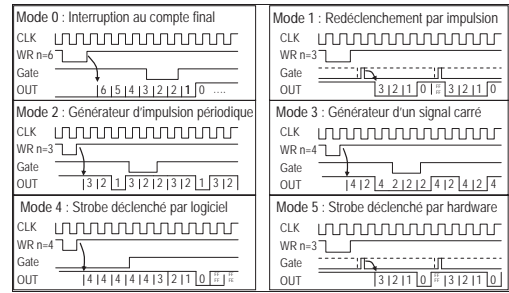


Figure 5-4: Différents modes de fonctionnement

5. Application

On se propose de générer à la sortie OUT1 du timer 8253 les fréquences musicales du Tableau 5-2 afin d'entendre la symphonie musicale proposé par le programme ci-après.

Echelle musicale	Do	Ré	Mi	Fa	Sol	La	Si
Fréquence de base en Hz	261,6	293,7	329,6	349,2	392,0	440,0	493,2
Taux de division (f_{CLK} / f_{OUT})							

Tableau 5-2: Fréquence de base

- 1) Complétez la dernière ligne du Tableau 5-2, sachant que la fréquence d'horloge du compteur 1, $f_{CLK} = 1,2288 \text{ Mhz}$.
- 2) Ecrire un programme en assembleur 8086 permettant de générer à la sortie OUT1 du timer les fréquences musicales. Insérez dans la boucle une petite attente. (Positionnez bien les switches du DIP SW3).
- 3) On donne le programme suivant :
 - a. Tapez et exécutez le programme.
 - b. Commentez toutes les instructions du programme.

```

; =====
; MTS-86C - PROGRAMMATION DU TIMER 8253
; =====
COUNT1 EQU 0FFDAH ; OUT DX, AL ; DB SOL, 2, MI, 2, NO, 2
CSR EQU 0FFDEH ; MOV AL, AH ; DB RA, 2, SOL, 2, FA, 2
CNT3 EQU 3FD6H ; OUT DX, AL ; DB MI, 2, FA, 2, MI, 2
D0 EQU 0 ; POP SI ; DB RE, 2, DO, 2, RE, 6
RE EQU 2 ; INC SI ; DB NO, 2
MI EQU 4 ; MOV AL, [SI] ; DB MI, 4, MI, 2, MI, 2
FA EQU 6 ; J1: CALL TIMER ; DB FA, 2, FA, 2, FA, 2
SOL EQU 8 ; DEC AL ; DB FA, 2, SOL, 3, SOL, 1
RA EQU 0AH ; JNZ J1 ; DB SOL, 2, SOL, 2, RA, 2
SY EQU 0CH ; MOV AX, 10 ; DB SOL, 2, FA, 2, NO, 2
DO1 EQU 0EH ; OUT DX, AL ; DB RA, 2, RA, 2, FA, 2
NO EQU 10H ; MOV AL, AH ; DB RA, 2, DO1, 2, SY, 2
ORG 0 ; OUT DX, AL ; DB RA, 2, SOL, 2, DO1, 2
CODE SEGMENT ; MOV CX, 0AFH ; DB SOL, 1, RA, 1, SOL, 2
ASSUME CS:CODE, DS:CODE ; LOOP $ ; DB MI, 2, FA, 2, RE, 2
START: MOV SP, 4000H ; INC SI ; DB DO, 2, NO, 2
; ===== DATA =====
; MOV AX, CS ; JMP MAIN ; DB SOL, 3, SOL, 1, SOL, 2
; MOV DS, AX ; _END: MOV AH, 4CH ; DB SOL, 2, RA, 2, RA, 2
; MOV DX, CNT3 ; INT 21H ; DB RA, 2, NO, 2
; MOV AL, 09H ; TIMER: MOV CX, 2800H ; DB DO1, 3, DO1, 1, SY, 2
; OUT DX, AL ; LOOP $ ; DB RA, 8
; MOV DX, CSR ; RET ; DB DO1, 3, DO1, 1, SY, 2
; MOV AL, 76H ; ; DB RA, 2, SOL, 2, SOL, 2
; OUT DX, AL ; DATA1 : ; DB MI, 2, NO, 2
; MOV DX, COUNT1 ; DW 4697, 4184, 3728 ; DB RA, 3, RA, 1, SOL, 2
; LEA BX, DATA1 ; DW 3519, 3135, 2793 ; DB MI, 2, RE, 8, DO, 6
; LEA SI, DATA2 ; DW 2491, 2352, 10H ; DB RE, 2, MI, 4, SOL, 4
MAIN: MOV AL, [SI] ; DATA2 : ; DB RA, 2, DO1, 2, RA, 4
; CMP AL, 0FFH ; DB MI, 2, RA, 2, SOL, 2 ; DB SOL, 4, MI, 4, SOL, 6
; JZ _END ; DB FA, 2, MI, 2, FA, 2 ; DB SOL, 2, RA, 4, SOL, 4
; XOR AH, AH ; DB MI, 2, NO, 2 ; DB DO1, 0BH
; PUSH SI ; DB MI, 2, FA, 2, SOL, 2 ; DB 0FFH
; MOV SI, AX ; DB DO1, 2, SY, 2, RA, 2 ; CODE ENDS
; MOV AX, [BX+SI] ; DB SOL, 2, NO, 2, DO1, 2 ; END START

```

TP6 : L'INTERFACE SERIE 8251

1. Introduction

L'USART 8251A est une interface série asynchrone et synchrone (Universal Synchronous Asynchronous Receiver Transmitter). Dans ce Tp nous étudions uniquement le mode asynchrone. Le schéma bloc du 8251A est donné par la Figure 6-1.

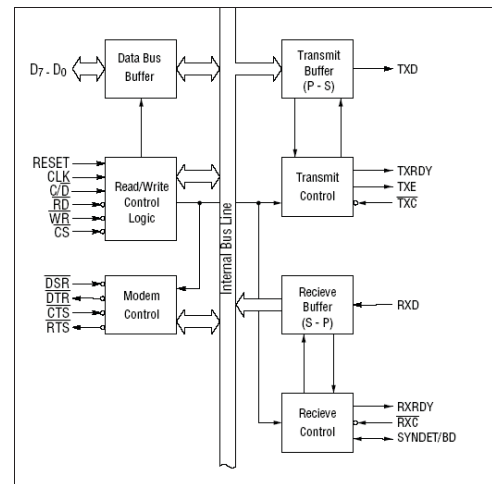


Figure 6-1: Schéma bloc de l'USART 8251A

Le registre de données (Data buffer Bus) assure la liaison entre le bus de données du microprocesseur et les registres internes du circuit. A l'entrée les bits arrivent en série sur la broche Rx/D, ils sont décalés dans le registre Recieve Buffer où ils seront acquis par le microprocesseur. Inversement en sortie, la donnée arrive au registre Transmit Buffer où on leur ajoute les bits de synchronisation nécessaire avant de les envoyer en série sur la ligne Tx/D.

2. Fonctionnement

2.1. Sélection des registres du 8251A

Le 8251A dispose de 5 registres divisés en deux groupes : registres de contrôles et registres de données (sélectionnés par la ligne C/D).

- le groupe contrôle comporte le registre de mode, le registre de contrôle et le registre d'état. Ces registres sont sélectionnés lorsque la ligne C/D = 1.
- Le groupe de données comporte le registre Buffer Data In et Buffer Data Out. Ils sont sélectionnés lorsque la ligne C/D = 0.

Le Tableau 6-1 présente la sélection des registres.

CS	C/D	RD	WR	Sélection
1	x	x	x	8251A n'est pas sélectionné
0	1	0	1	Lecture du registre d'état
0	1	1	0	Écriture dans le registre de mode ou de contrôle.
0	0	0	1	Lecture du registre Buffer Data In
0	0	1	0	Écriture dans le registre Buffer Data Out

Tableau 6-1 : Sélection des registres du 8251A

Après un RESET; on doit exécuter la séquence d'initialisation donnée par l'organigramme suivant :

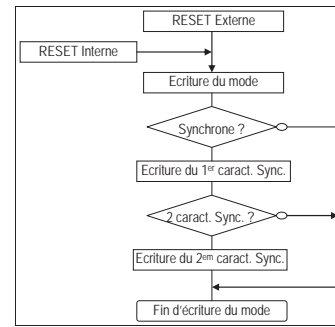


Figure 6-2 : Séquence d'initialisation

Après l'exécution de la séquence d'initialisation, les octets qui suivent seront dirigés vers le registre de contrôle lorsque C/D= 1, et vers le registre Buffer Data Out lorsque C/D= 0.

Remarque :

Lors de la mise sous tension le 8251A pourrait se trouver au niveau du registre de mode, du caractère de synchronisation ou du registre de contrôle; afin de bien initialiser le 8251A le constructeur recommande d'écrire trois 00FH consécutifs à l'adresse C/D = 1 avant l'exécution de la séquence d'initialisation.

2.2. Registres de configuration

Il existe deux registres de configuration : le registre de mode et le registre de contrôle.

2.2.1. Registre de mode

L'écriture dans le registre de mode ne se fait qu'après un RESET interne ou externe. On définit dans ce registre :

- le mode asynchrone ou synchrone,
- le nombre de bits de Stop,
- la longueur du caractère,
- le bit de parité,
- le facteur de débit,

La vitesse de transmission est égale à la fréquence d'horloge appliquée aux entrées TXC et RXC divisé par le facteur de débit (1, 16 ou 64).

S2	S1	EP	PEN	L2	L1	B2	B1
Bits de Stop		Test de parité		Longueur de donnée		Facteur de débit	
00	Invalide	00	Pas de test	00	5 bits	00	Mode Synch.
01	1 bit	01	Parité impaire	01	6 bits	01	Clock
10	1,5 bits	10	Pas de test	10	7 bits	10	Clock /16
11	2 bits	11	Parité paire	11	8 bits	11	Clock / 64

Figure 6-3 : Configuration du registre de mode en Asynchrone

2.2.2. Registre de contrôle

L'accès au registre de contrôle se fait après l'exécution de la séquence d'initialisation ; il permet de :

- valider ou inhiber l'émission,
- valider ou bloquer la réception,
- activer ou désactiver les broche DTR et RTS pour le contrôle de flux,
- envoyer un caractère break,
- Effacer les bits d'erreur dans le registre d'état,
- Provoquer un RESET interne (ou logicielle),
- Détection du caractère de synchronisation (mode synchrone)

EH	IR	RTS	ER	SBRK	RXE	DTR	TxEN
1 : Reset Flag d'erreur 0 : Opération normale		1 : Emission valide 0 : Invalide		1 : broche DTR = 0 0 : broche DTR = 1		1 : Recherche de Car. de Sync. (Sync. mode) 0 : Opération normale	
1 : broche RTS = 0 0 : broche RTS = 1		1 : Réception valide 0 : Invalide		1 : envoi break car. 0 : Opération normale			
1 : Reset interne 0 : Opération normale							

Figure 6-4 : Registre de contrôle

2.3. Registre d'état (Status)

Le registre d'état est à lecture seule ; le format de ce registre est donné à la Figure 6-5.

DSR	SYNDET/BD	FE	OE	PE	TXEMPTY	RXRDY	TXRDY
1 : Erreur de débordement				1 : Buffer d'émission vide (on peut envoyer)			
1 : Erreur de cadrage				1 : Buffer de réception plein (caractère à lire)			
1 : Détection d'un caractère break				1 : Registre TSR vide (on peut envoyer)			
1 si la broche DSR = 0 0 si DSR = 1				1 : Erreur de parité			

Figure 6-5 : Registre d'état

- TXRDY à 1 indique que le buffer d'émission est vide, ce bit se remet à 0 lorsque le microprocesseur écrit un caractère.
- RXRDY à 1 indique que le buffer de réception contient un caractère. La lecture du caractère par le microprocesseur remet ce flag à 0.
- TXEMPTY à 1 indique que le registre de décalage émetteur (TSR) est vide, le front descendant du signal WR le remet à 0.
- PE à 1 indique qu'il y a une erreur de parité.
- OE à 1 indique qu'il y a une erreur de débordement (vitesse non conforme).
- FE à 1 indique qu'il y a une erreur de format,
- SYNDET/BD, en mode synchrone elle se met à 1 lors de la détection d'un caractère de synchronisation ; en mode asynchrone elle indique la réception d'un Break Data. Ce bit se remet à 0 lorsque la ligne RD repasse à 1.
- DSR à 1 indique que le modem est en fonctionnement.

3. Branchement du 8251A

Le kit MTS-86C comporte deux circuits série. La Figure 6-7 et la Figure 6-8 présentent les schémas de connexion des circuits 8251-1 et 8251-2. Les circuits MC1488 et MC1489 sont des adaptateurs de niveau. Le débit de transmission de chaque circuit est fixé par un signal d'horloge RS1-CLK et RS2-CLK correspondant ; provenant du générateur d'horloge de la Figure 6-6. Le générateur d'horloge est constitué principalement d'un double compteur hexadécimal 74LS393 attaquée par une fréquence de 2,4576Mhz et de deux multiplexeurs 74LS251. Les deux sorties du compteur CLK0 et CLK1 sont utilisées par le timer 8254 et l'ADC809 ; les 6 sorties restantes et la sortie OUT0 du timer sont connectées aux entrées des multiplexeurs. La sélection de l'une des 7 entrées est réalisée par le DIPSW 1.

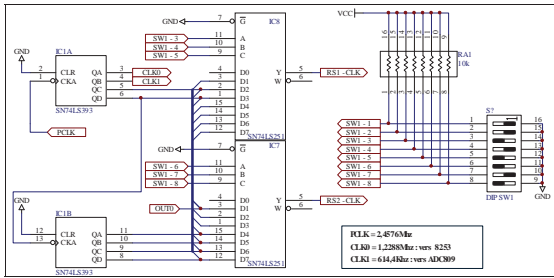


Figure 6-6 : Générateur d'horloge de transfert série

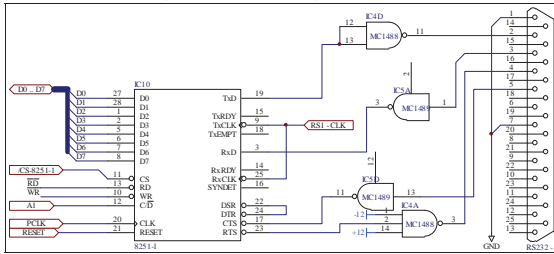


Figure 6-7 : Schéma de connexion du 8251-1

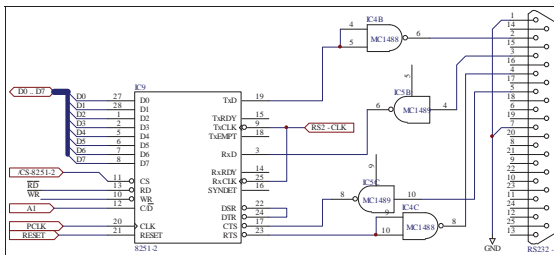
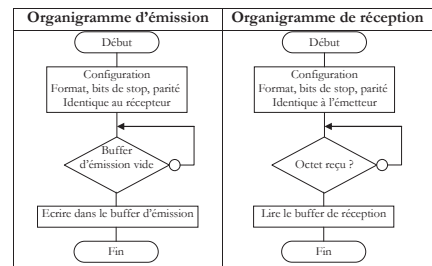


Figure 6-8 : Schéma de connexion du 8251-2

4. Manipulation

On admet que les adresses de bases des circuits 8251-1 et 8251-2 sont respectivement FFF0H et FFD0H.

- 1) Donner les adresses des registres de données et de contrôle de chaque circuit.
- 2) En se référant à la Figure 6-6 ; donnez les débits de transmission possibles.
- 3) Selon la position des switches du DIPSW 1, donner le débit de transmission, si on choisit un taux de division égal 16.
- 4) Elaborer un programme qui transmet 10 fois la suite des caractères ABC suivie d'un espace (ABC ABC ABC) vers le PC. Il faut respecter le choix des paramètres suivants : 19200 Baud, 8 bits, pas de parité, 1 bit Stop.
- 5) Les caractères provenant du clavier du PC sont envoyés à travers la liaison série au kit MTS-86C. Les caractères reçus sont ensuite redirigés vers l'écran du PC (mode echo). Avant l'envoi, tous les caractères minuscules sont transformés en majuscules. Les autres restent inchangés. Il faut choisir les paramètres suivants : 19200 Baud, 8bits de donnée, pas de parité, 1 bit de Stop.
 - a. Donner la position des switches SW1-3, SW1-4 et SW1-5 pour avoir le débit demandé.
 - b. Déterminer les mots de contrôle.
 - c. Ecrire et exécuter le programme de l'énoncé.



ANNEXES

Les instructions du 8086

LES INSTRUCTIONS DU 8086/88

	INSTR CTIONS ARITMETI ES				IND CATE RS	MODE D'ADRESSAGE DE LA MEMOIRE ¹								
	DEST	SO RCE	OPERATION EFFECT EE	EXEMPLE		O	S	A	P	C	D	IR	B	X
ADD	R	R	(Rsource) + (Rdest) → Rdest	ADD CX, DX	+	+	+	+	+					
	R	MEM	(MEM) + (R) → R	ADD DI, LABEL	+	+	+	+	+					
	MEM	R	(R) + (MEM) → MEM	ADD LABEL, CL	+	+	+	+	+					
	R	DATA	DATA + (R) → R	ADD CL, 23	+	+	+	+	+					
MEM	DATA	DATA + (MEM) → MEM	ADD LABEL, 41	+	+	+	+	+						
ADC	R	R	(Rsource) + (Rdest) + C → Rdest	ADC AX, SI	+	+	+	+	+					
	R	MEM	(MEM) + (R) + C → R	ADC BX, LABEL	+	+	+	+	+					
	MEM	R	(R) + (MEM) + C → MEM	ADC LABEL, CL	+	+	+	+	+					
	R	DATA	DATA + (R) + C → R	ADC AX, 300	+	+	+	+	+					
MEM	DATA	DATA + (MEM) + C → MEM	ADC LABEL, 84	+	+	+	+	+						
DAA	Correction décimale sur l'addition				DAA	?	+	+	+					
AAA	Correction ASCII sur l'addition				AAA	?	?	?	?					
S B	R	R	(Rdest) - (Rsource) → Rdest	SUB AX, CX	+	+	+	+	+					
	R	MEM	(R) - (MEM) → R	SUB BX, LABEL	+	+	+	+	+					
	MEM	R	(MEM) - (R) → MEM	SUB [BP+I], CL	+	+	+	+	+					
	R	DATA	(R) - DATA → R	SUB SI, 6040	+	+	+	+	+					
MEM	DATA	(MEM) - DATA → MEM	SUB LABEL, 850	+	+	+	+	+						
SBB	R	R	(Rdest) - (Rsource) - C → Rdest	SBB CX, AX	+	+	+	+	+					
	R	MEM	(R) - (MEM) - C → R	SBB DI, LABEL	+	+	+	+	+					
	MEM	R	(MEM) - (R) - C → MEM	SBB LABEL, BX	+	+	+	+	+					
	R	DATA	(R) - DATA - C → R	SBB BX, 2	+	+	+	+	+					
MEM	DATA	(MEM) - DATA - C → MEM	SBB LABEL, 10	+	+	+	+	+						
DAS	Correction décimale sur la soustraction				DAS	?	+	+	+					
AAS	Correction ASCII sur la soustraction				AAS	?	?	?	?					
M L (non signée)	R8	R8	(AL) x (R8) → AH, AL	MUL BL	+	?	?	?	?					
	R16	R16	(AX) x (R16) → DX, AX	IMUL CX	+	?	?	?	?					
IM L (signée)	MEM8	MEM8	(AL) x (MEM8) → AH, AL	MUL LABEL	+	?	?	?	?					
	MEM16	MEM16	(AX) x (MEM16) → DX, AX	IMUL LABEL	+	?	?	?	?					
AAM	Correction ASCII sur la multiplication				AAM	?	+	?	?					
DIV (non signée)	R8	R8	(AH,AL) / (R8) → AL: rest → AH (DX,AX) / (R16) → AX: rest → DX	DIV CL	?	?	?	?	?					
	R16	R16	(AH,AL) / (R16) → AX: rest → DX (DX,AX) / (MEM16) → AX: rest → DX	IDIV CX	?	?	?	?	?					
IDIV (signée)	MEM8	MEM8	(AH,AL) / (MEM8) → AL: rest → AH (DX,AX) / (MEM16) → AX: rest → DX	DIV LABEL	?	?	?	?	?					
	MEM16	MEM16	(AH,AL) / (MEM16) → AL: rest → AH (DX,AX) / (MEM16) → AX: rest → DX	IDIV TABLE[SJ]	?	?	?	?	?					
AAD	Correction ASCII sur la division				AAD	?	+	?	?					
CBW	Extension du signe d'un octet				CBW	?	+	?	?					
CWD	Extension du signe d'un mot de 16 bits				CWD	?	+	?	?					

	INSTR CTIONS LOGI ES				IND CATE RS	MODE D'ADRESSAGE DE LA MEMOIRE ¹								
	DEST	SO RCE	OPERATION EFFECT EE	EXEMPLE		O	S	A	P	C	D	IR	B	X
OR	R	R	(Rsource) + (Rdest) → Rdest	OR BL, AL	0	+	?	?	?	0				
	R	MEM	(MEM) + (R) → R	OR BX, LABEL	0	+	?	?	?	0				
	MEM	R	(R) + (MEM) → MEM	OR LABEL, CL	0	+	?	?	?	0				
	R	DATA	DATA + (R) → R	OR AL, 1AH	0	+	?	?	?	0				
MEM	DATA	DATA + (MEM) → MEM	OR LABEL, 09	0	+	?	?	?	0					
XOR	R	R	(Rsource) ⊕ (Rdest) → Rdest	XOR AX, SI	0	+	?	?	?	0				
	R	MEM	(MEM) ⊕ (R) → R	XOR BX, LABEL	0	+	?	?	?	0				
	MEM	R	(R) ⊕ (MEM) → MEM	XOR LABEL, DX	0	+	?	?	?	0				
	R	DATA	DATA ⊕ (R) → R	XOR SI, 003BH	0	+	?	?	?	0				
MEM	DATA	DATA ⊕ (MEM) → MEM	XOR LABEL, 2CH	0	+	?	?	?	0					

¹ - 0 : nul ; + : Modifié ; ? : Indéfinit ; Vide : inchangé

² - D : adressage direct ; IR : indirect registre ; B : base ; X : indice ; BX : base + indice

AND	R MEM R MEM R MEM	R MEM R DATA DATA • (MEM)	(Rsource)•(Rdest) → Rdest → R → MEM → R → MEM	AND AL, BL AND CX, LABEL AND LABEL, BL AND AX, 021EH AND LABEL, 7FH	0 + ? + 0 0 + ? + 0 0 + ? + 0 0 + ? + 0 0 + ? + 0
NOT	R MEM	(R)	→ R → MEM	NOT BX NOT LABEL	
NEG	R MEM	0 - (R) 0 - (MEM)	→ R → MEM	NEG AL NEG LABEL	++++++ ++++++
INC	R MEM	(R) + 1 (MEM) + 1	→ R → MEM	INC CL INC LABEL	++++++ ++++++
DEC	R MEM	(R) - 1 (MEM) - 1	→ R → MEM	DEC AX DEC [BX]	++++++ ++++++
TEST	R MEM R MEM R MEM	(Rsource)•(Rdest) (MEM)•(R) (R)•(MEM) DATA•(R) DATA•(MEM)	Ces instructions n'affectent que les indicateurs	TEST SI, DI TEST SI, LABEL TEST LABEL, BL TEST AL, 0FH TEST LABEL, 02	0 + ? + 0 0 + ? + 0 0 + ? + 0 0 + ? + 0 0 + ? + 0
CMP	R MEM R MEM R MEM	(Rdest) - (Rsource) (R) - (MEM) (MEM) - (R) (R) - DATA (MEM) - DATA	Ces instructions n'affectent que les indicateurs	CMP CX, AX CMP DI, [BP - 2] CMP LABEL, SI CMP AL, 04 CMP LABEL, 08	++++++ ++++++ ++++++ ++++++ ++++++

INSTR CTIONS DECALAGES ET ROTATIONS				IND CATE RS	MODE D'ADRESSAGE DE LA MEMOIRE
DEST	SO RCE	OPERATION EFFECT EE	EXEMPLE	O S A P C	D IR B X BX
SAL / SHL	R MEM	(R) décalé à gauche de 1 bit [CY] ← [] ← 0	SHL BL, 1 SHL DI, CL SHL LABEL, 1 SHL LABEL, CL	+++ ? ++ +++ ? ++ +++ ? ++ +++ ? ++	
SAR	R MEM	(R) décalé à droite de 1 bit [S] → [] → [CY]	SAR DX, 1 SAR DI, CL SAR LABEL, 1 SAR LABEL, CL	+++ ? ++ +++ ? ++ +++ ? ++ +++ ? ++	
SHR	R MEM	(R) décalé à droite de 1 bit 0 → [] → [CY]	SHR SI, 1 SHR SI, CL SHR LABEL, 1 SHR LABEL, CL	+++ ? ++ +++ ? ++ +++ ? ++ +++ ? ++	
ROL	R MEM	rotation de 1 bit à gauche de (R) [CY] ← [] ← []	ROL BX, 1 ROL BX, CL ROL LABEL, 1 ROL LABEL, CL	+ ? + + ? + + ? + + ? +	
ROR	R MEM	rotation de 1 bit à droite de (R) [] → [] → [CY]	ROR BL, 1 ROR AX, CL ROR LABEL, 1 ROR LABEL, CL	+ ? + + ? + + ? + + ? +	
RCL	R MEM	rotation de 1 bit à gauche de (MEM) [CY] ← [] ← []	RCL AX, 1 RCL AL, CL RCL LABEL, 1 RCL LABEL, CL	+ ? + + ? + + ? + + ? +	
RCR	R MEM	rotation de 1 bit à droite de (R) [] → [] → [CY]	RCR BX, 1 RCR BX, CL RCR LABEL, 1 RCR LABEL, CL	+ ? + + ? + + ? + + ? +	

INSTR CTIONS TRANSFERT DE DONNEES				MODE D'ADRESSAGE DE LA MEMOIRE
DEST	SO RCE	OPERATION EFFECT EE	EXEMPLE	D IR B X BX
MOV	R MEM R MEM R MEM R16 R16 R16 R16 R16 R16 R16 R16	(Rsource) → (Rdest) (MEM) → (MEM) (R) → MEM DATA → DATA R16 → R16 (R16) → R16 (MEM) → R16 (RSEG) → R16 (RSEG) → R16 (RSEG) → R16	MOV AX, BX MOV BX, LABEL MOV [DI], CX MOV CL, 15H MOV [BX + 2], 2AH MOV DS, AX MOV ES, LABEL MOV BP, SS MOV [BX] BLOC, CS	
XCHG	R MEM	(Rsource) → (Rdest) et (Rdest) → (Rsource) (R) → (MEM) et (MEM) → (R)	XCHG AL, BL XCHG LABEL, AX	
XLAT	MEM	([BX + AL]) → AL	XLAT TABLE	
LAHF	Octet poids faible du registre indicateur	→ AH	LAHF	
SAHF	(AH) → partie basse du registre indicateur	→ SAHF	SAHF	
LEA	R16 MEM16	Charge l'adresse effective dans R16	LEA BX, [BP]	
LDS	R16 MEM32	(MEM) → R16 et (MEM + 2) → DS	LDS SI, POINTEUR	
LES	R16 MEM32	(MEM) → R16 et (MEM + 2) → ES	LES DI, POINTEUR	
P SH	R16 RSEG MEM	(R16) → PILE (Rseg) (MEM) → PILE	PUSH DX PUSH DS PUSH LABEL [SI]	
P SHF		(registre d'état) → PILE	PUSHF	
P SHA		AX, CX, DX, BX, SP, BP, SI, DI → PILE	PUSHA	
POP	R16 RSEG MEM	(R16) → PILE (RSEG) → PILE (MEM) → PILE	POP DX POP DS POP LABEL	
POPF		PILE → registre d'état	POPF	
POPA		PILE → DI, SI, BP, SP, BX, DX, CX, AX	POPAD	
IN	A PORT DX A	registre A → registre A	IN AX, 1204H IN AL, DX	
OUT	PORT DX A A	(A) → PORT (A) → (DX)	OUT 3, AX OUT DX, AL	

INSTR CTIONS DE CHAINES DE CARACTERES				LABELS : SI	LABELD : DI
OPERATION EFFECT EE	EXEMPLE	IND CATE RS			
MOVBS LABELS	(LABELS) → LABELD	AUCUN			
REP MOVBS / MOVSW	répétition tant que (CX) ≠ 0	AUCUN			
REP MOVSB / MOVSW	((SI)) → (DI)	AUCUN			
REP MOVSW / MOVSW	répétition tant que (CX) ≠ 0	AUCUN			
CMPS	(LABELS) - (LABELD)	AUCUN			
REPE / REP	répétition tant que (CX) ≠ 0 et (LABELS) = (LABELD)	AUCUN			
REPNE / REPN	répétition tant que (CX) ≠ 0 et (LABELS) ≠ (LABELD)	AUCUN			
SCAS	(A) - (LABELD)	AUCUN			
REP	répétition tant que (CX) ≠ 0 et (A) = (LABELD)	AUCUN			
REPNE / REPN	répétition tant que (CX) ≠ 0 et (A) ≠ (LABELD)	AUCUN			
LODS	(LABELD) → A	AUCUN			
STOS	(A) → LABELD	AUCUN			
REP	répétition tant que (CX) ≠ 0	AUCUN			

INSTR CTIONS DE BRANCHEMENT					
Non signés		Conditionnels		signés	
E /	Z = 1	=	E /	Z = 1	=
A / NBE	(C + Z) = 0	>	G / NLE	(S ⊕ F) + Z = 0	>
B / NAE	C = 1	<	L / NGE	(S ⊕ F) = 1	<
AE / NB	C = 0	≥	GE / NL	(S ⊕ F) = 0	≥
BE / NA	(C + Z) = 1	≤	LE / NG	(S ⊕ F) + Z = 1	≤
NE / N	Z = 0	≠	NE / N	Z = 0	≠

Branchements conditionnels par un seul indicateur	Indicateur	Egal 1	Egal 0	Explication	
	Zéro	C	N	Saut si continu = 0 ou non	
Carry	C	NC	Saut si carry = 1 ou non		
Dépassement	O	NO	Saut si dépassement de capacité ou non		
Signe	S	NS	Saut si signe = 1 ou non		
Parité	P	NP	Saut si parité ou non		
MP	Type de branchement		Opération effectuée	Valeur limite de dépassement	Exemple
	Court (short) direct	(IP) = (IP) + DEP		-128 / +127	JMP SHORT LABEL
	Proche (near) direct	(IP) = (IP) + DEP		-32768 / +32767	JMP LABEL
	Proche (near) indirect par registre de 16 bits	(IP) = (R16)		-32768 / +32767	JMP CX
	Proche (near) indirect par mémoire de 16 bits	(IP) = (MEM16)		-32768 / +32767	JMP TABLE [SI]
	Lointain (far) direct	CS est mis à jour; (IP) = DEP	Accès possible à toute la mémoire		JMP FAR_LABEL
Lointain (far) indirect par mémoire de 32 bits	(CS, IP) = (MEM32)	Accès possible à toute la mémoire		JMP DWORD TABLE [DI]	
CALL	Type de branchement		Opération effectuée	Valeur limite de dépassement	Exemple
	Proche (near) direct	IP = IP + DEP		-32768 / +32767	CALL LABEL
	Proche (near) indirect par registre de 16 bits	IP = (R16)		-32768 / +32767	CALL AX
	Proche (near) indirect par mémoire de 16 bits	IP = (MEM16)		-32768 / +32767	CALL TABLE [SI]
	Lointain (far) direct	CS est mis à jour; (IP) = DEP	Accès possible à toute la mémoire		CALL FAR_LABEL
Lointain (far) indirect par mémoire de 32 bits	(CS, IP) = (MEM32)	Accès possible à toute la mémoire		CALL DWORD [BX] [DI]	
LOOP	(CX) = (CX) - 1 puis saut si (CX) = 0	DEP : -128 à +127		LOOP LABEL	
LOOPNE / LOOPN	(CX) = (CX) - 1 puis saut si (CX) = 0 et Z = 0	DEP : -128 à +127		LOOPNE LABEL	
LOOPE / LOOP	(CX) = (CX) - 1 puis saut si (CX) = 0 et Z = 1	DEP : -128 à +127		LOOPE LABEL	
CX	saut si (CX) = 0	DEP : -128 à +127		JCXZ LABEL	

Les modes de fonctionnement du Timer 8253

