

SUJET 1

ETUDE D'UN SYSTEME DE COMPTAGE DE PIECES

La figure ci-dessous représente un système de comptage, simultané, de trois types de pièces, transportées par trois tapis roulants. Au passage d'une pièce devant un capteur optique, ce dernier génère une impulsion positive permettant d'incrémenter le compteur décimal correspondant (voir figure 3).

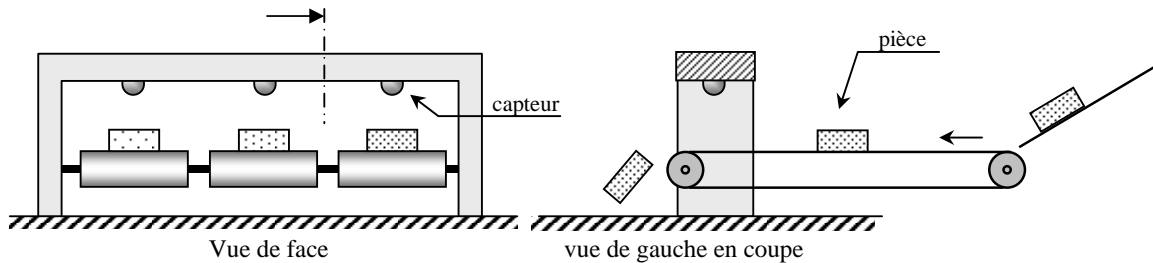


Figure 1 : Représentation schématique du système.

On veut effectuer le comptage pendant une durée bien déterminée. Pour cela on utilise le compteur 2 d'un Timer 8253 programmé en mode 1. La sortie de ce compteur permet d'autoriser ou d'arrêter le comptage des pièces.

Pour pouvoir afficher le nombre de pièces (supposé inférieur ou égal à 9), on utilise trois afficheurs 7 segments commandés par les ports A et B d'un PPI 8255 comme l'indique la figure 3.

Description d'un cycle de fonctionnement :

- ✓ Le cycle est lancé par un appui sur le bouton poussoir « DEPART »
- ✓ Cet appui doit entraîner la remise à zéro (RAZ) des compteurs et des afficheurs.
- ✓ Le comptage s'effectue par la suite, automatiquement, pendant 20 secondes (temps programmé au moyen du Timer).
- ✓ Lorsque le temps de temporisation s'écoule, les contenus des trois compteurs sont verrouillés dans des latches.
- ✓ Ces indications sont ensuite transférées vers les afficheurs par validation successive des latches, lecture du port C-BAS, puis sortie de la valeur lue vers l'afficheur approprié.
- ✓ Le système reste dans cet état jusqu'à un nouvel appui sur « DEPART ».

Remarque : La fin de la durée de temporisation est indiquée par le retour à 1 de la sortie OUT2. On suppose pour cela qu'on dispose d'une procédure « ETAT » permettant de retourner dans le registre AL le mot d'état du timer dont le format est donné en annexe.

Travail demandé :

- 1- En se référant au circuit de la figure 2, déterminer les adresses de base du timer 8253 et du PPI 8255. En déduire les adresses des différents registres et ports.

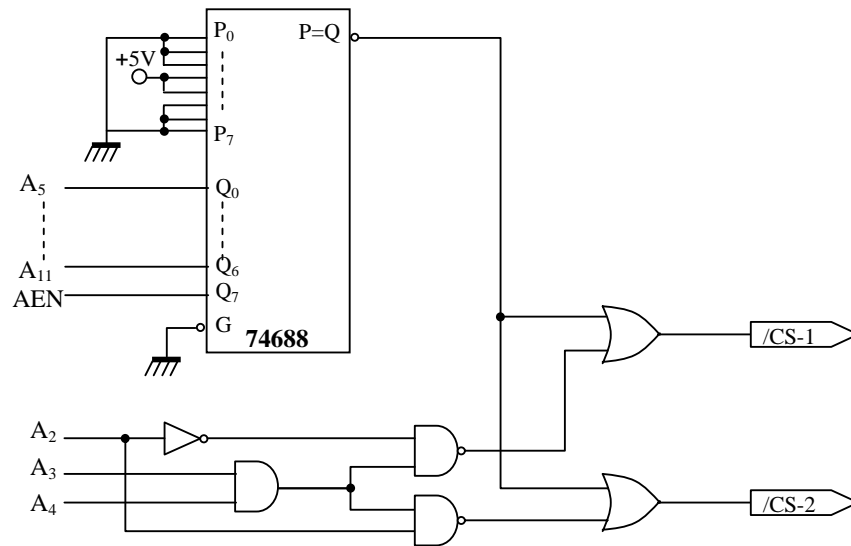
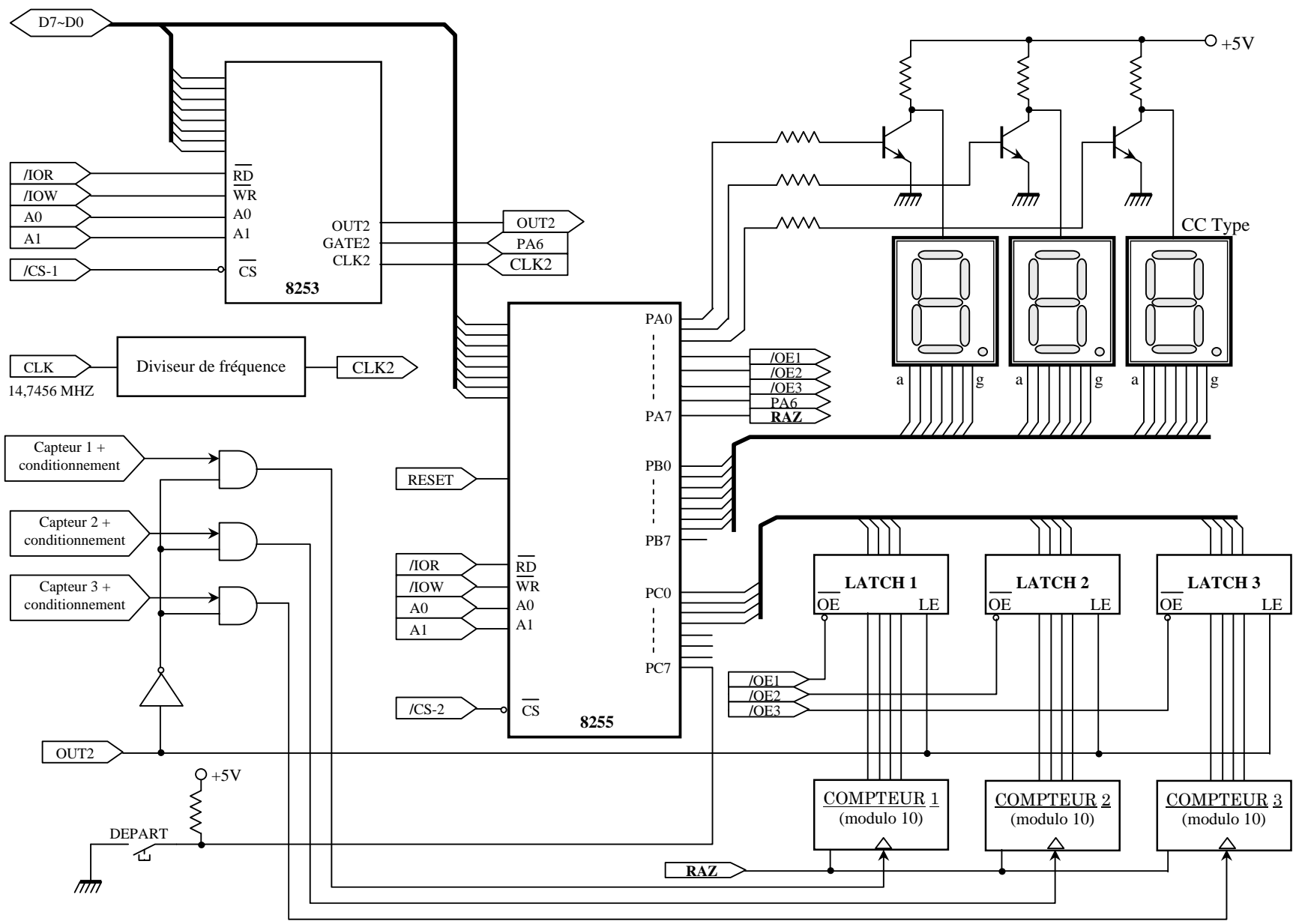


Figure 2 : Circuit de décodage d'adresse.

- 2- Ecrire un programme assembleur permettant de gérer le système conformément aux conditions de fonctionnement demandées.



SUJET 2

Soit à commander la cabine d'un ascenseur didactique donné par la figure 1.

Le circuit de commande est réalisé autour de deux circuits périphériques programmables 8253 et 8255 comme l'indique la figure 3. Il permet principalement de manipuler la cabine en montée et en descente ainsi que l'ouverture et la fermeture de sa porte.

Description

- ✓ A l'intérieur de la cabine on trouve :
 - trois boutons (ET0, ET1 et ET2) permettant de choisir l'étage destination.
 - un afficheur 7 segments indiquant l'étage auquel elle se trouve.

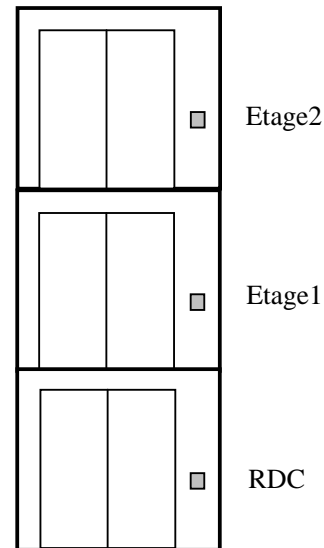


Figure 1

- ✓ La course de la cabine est limitée par trois capteurs de position CP0, CP1 et CP2 situés respectivement au réz de chaussée, à l'étage 1 et à l'étage 2 ; alors que l'état de sa porte est indiquée par deux capteurs (PO : porte ouverte) et (PF : porte fermée).
- ✓ La montée et la descente de la cabine sont assurées par un moteur à courant continu commandé par deux relais (MO pour la montée et DE pour la descente). De même pour l'ouverture et la fermeture de la porte (OP pour l'ouverture et FP pour la fermeture).

Exemple de cycle de fonctionnement

Initialement la cabine étant au réz de chaussée (l'afficheur indique donc 0), sa porte est fermée ; on veut effectuer le cycle suivant :

- ✓ un opérateur se trouvant à l'intérieur appui sur le bouton ET2, la cabine doit alors monter jusqu'au deuxième étage.
- ✓ à la fin de la montée la porte s'ouvre et reste dans cet état durant 10 secondes.
- ✓ lorsque ce temps s'écoule, la porte se ferme puis la cabine descend jusqu'au réz de chaussée.
- ✓ La porte s'ouvre de nouveau durant 10 secondes puis se ferme et le cycle est terminé.

Travail demandé

- 1- En se référant au circuit de décodage d'adresses, incomplet, donné par la figure 4,
 - 1.1- préciser le rôle du circuit 74688,
 - 1.2- préciser le rôle du circuit 74245. indiquer le sens de sens de circulation des données (entrée ou sortie) si l'entrée DIR est égale à 0.
 - 1.3- compléter le circuit de façon à sélectionner le PPI par une adresse de base de 30CH et le Timer par 314H. Donner alors les adresses des différents ports et registres.

- 2- Recopier le tableau ci-dessous puis compléter le sachant que l'afficheur utilisé est du type à anodes communes.

Chiffre	AC	g	f	e	d	c	b	a	Octet
								PA0	
0									
1									
2									

- 3- Calculer la valeur du compte initial (N) permettant au compteur zéro, programmé en mode 1, de générer une temporisation de 10 secondes sachant que la fréquence du signal d'horloge CLK est de 1,2 KHZ.

- 4- Déterminer les mots de commande du PPI et du Timer.

- 5- L'organigramme du cycle de fonctionnement désiré est représenté à la figure 2. Compléter cet organigramme puis le traduire en un programme assembleur 8086. Détailler en particulier la procédure ATTENTE qui doit permettre de :
 - ✓ arrêter le moteur entraînant la cabine,
 - ✓ ouvrir la porte puis la maintenir ouverte durant 10 s.
 - ✓ fermer la porte lorsque le temps est écoulé.

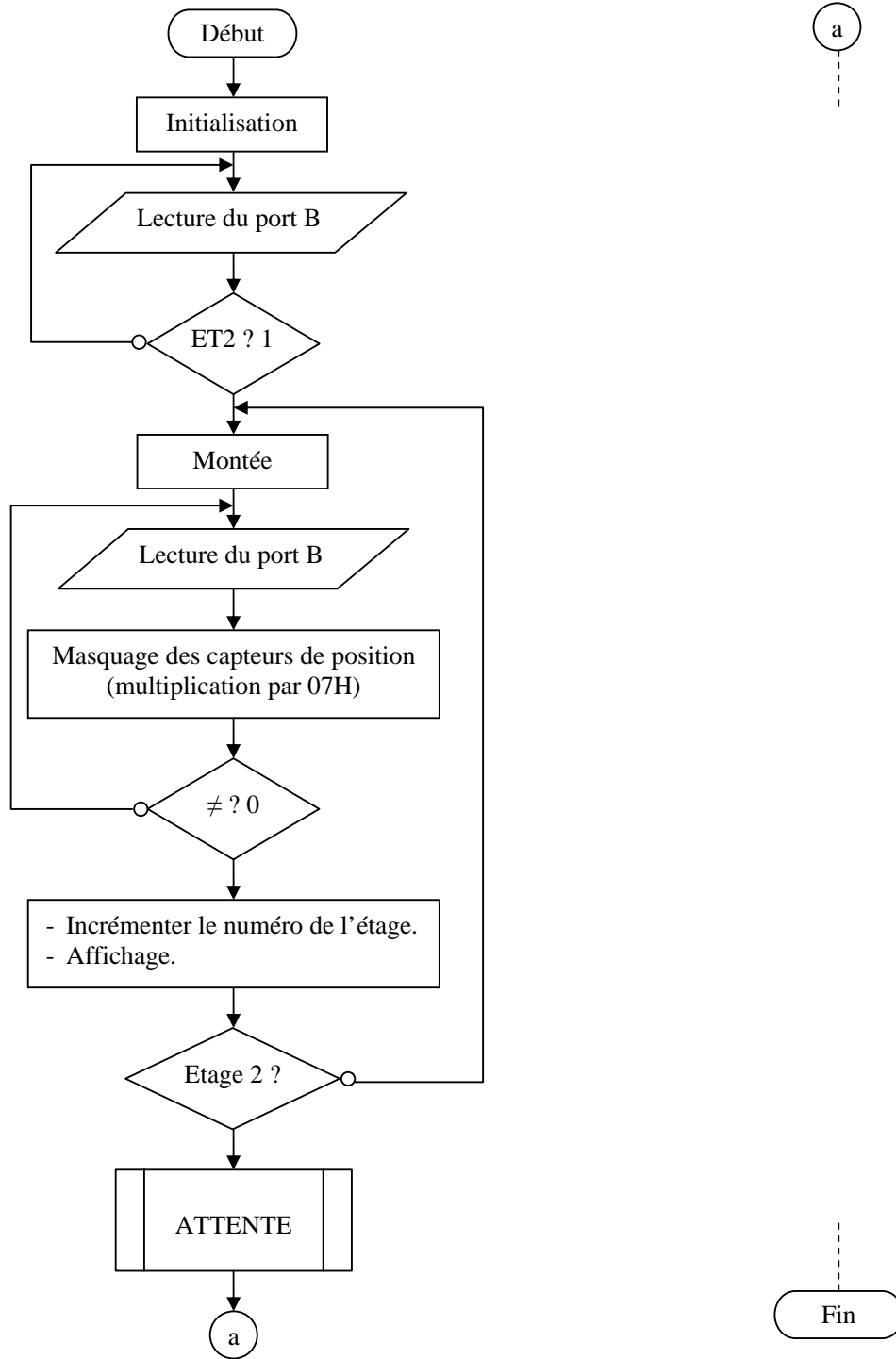


Figure 2 : Organigramme.

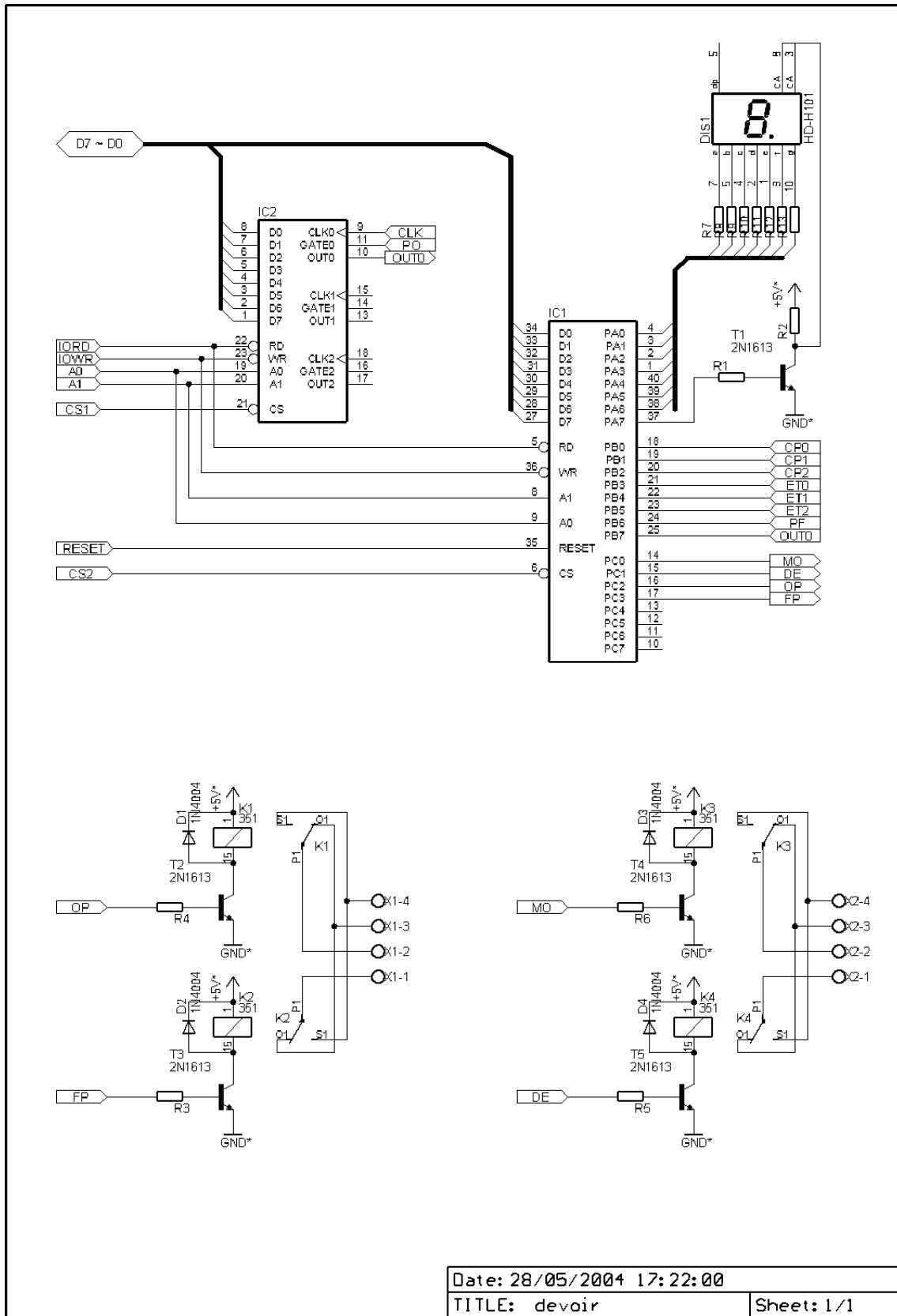
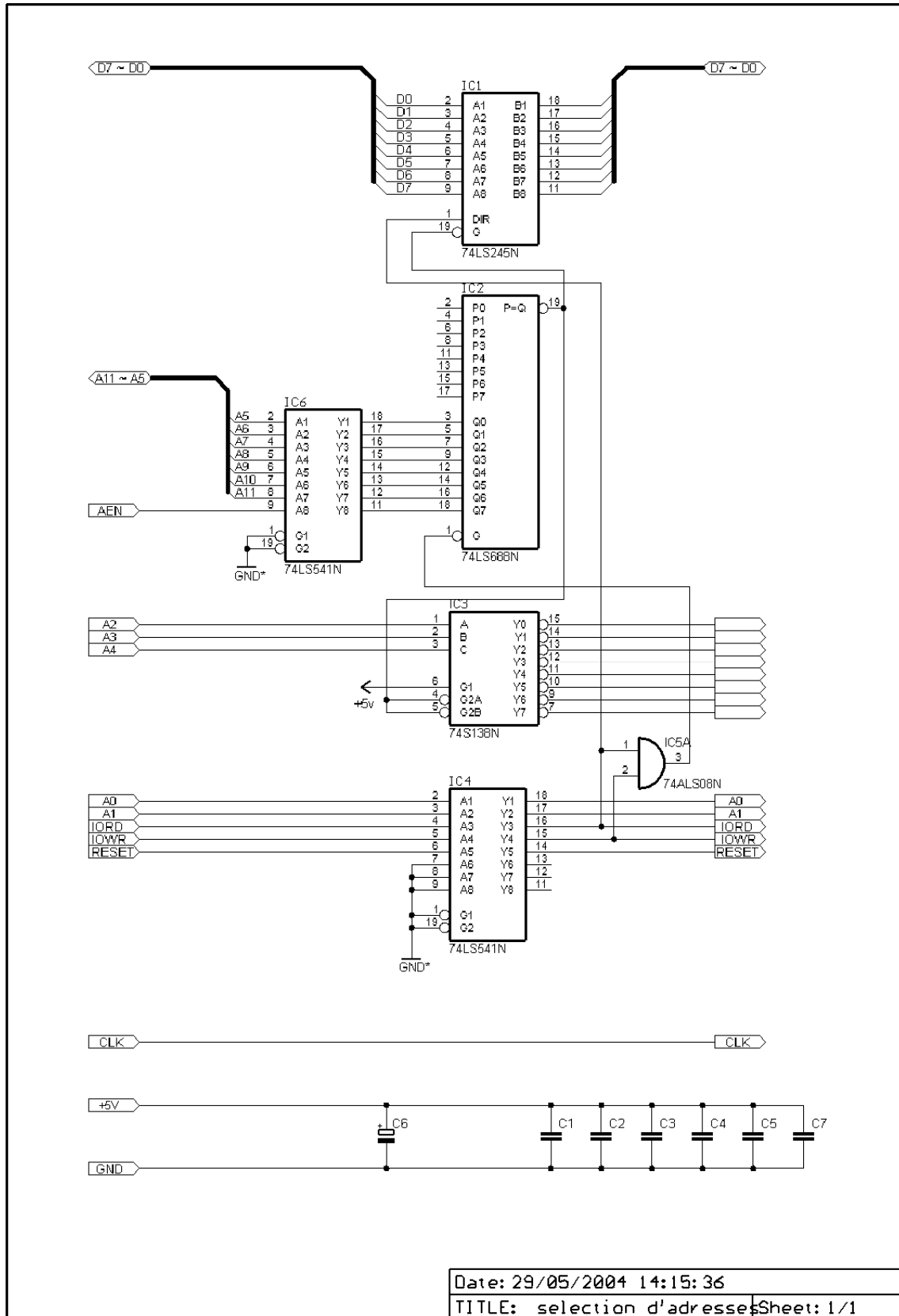


Figure 3 : Circuit de commande.

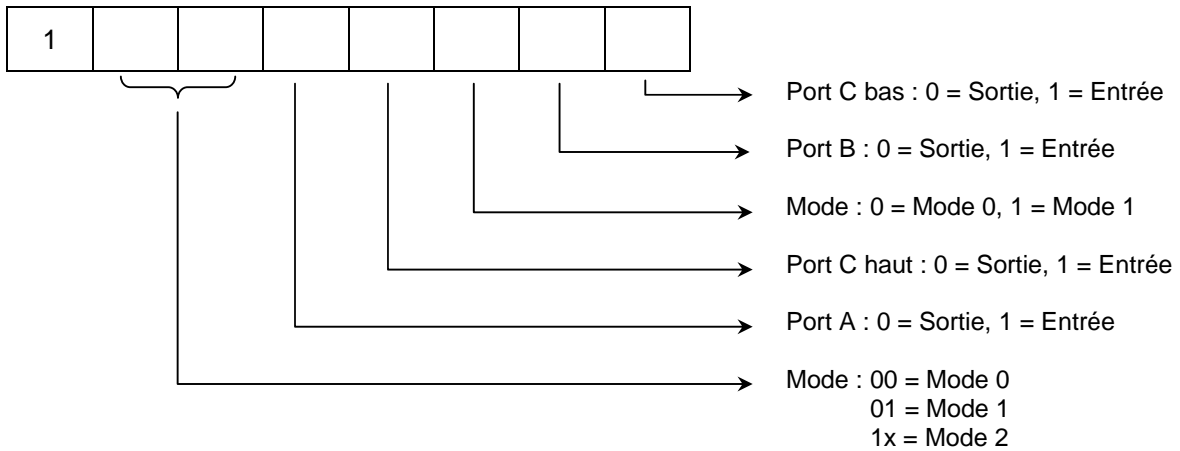


Date: 29/05/2004 14:15:36
 TITLE: selection d'adresses Sheet: 1/1

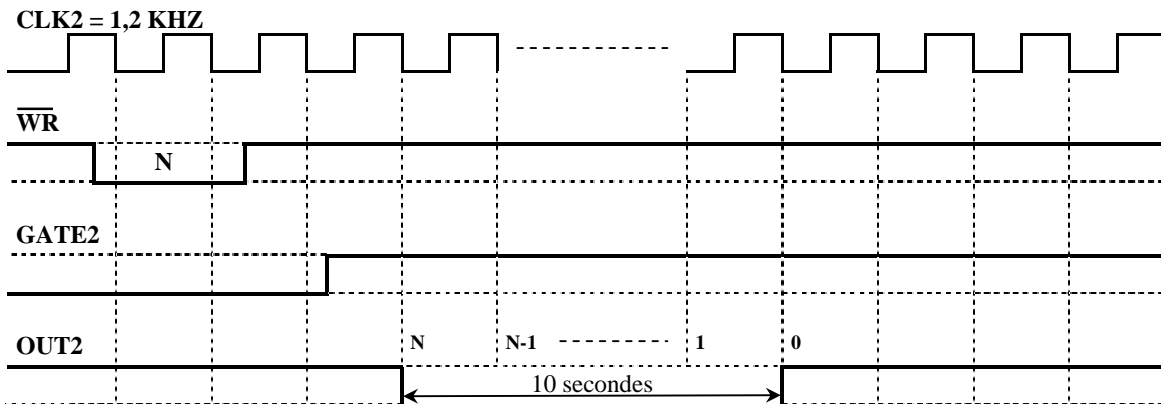
Figure 4 : Circuit de décodage d'adresses.

Annexes

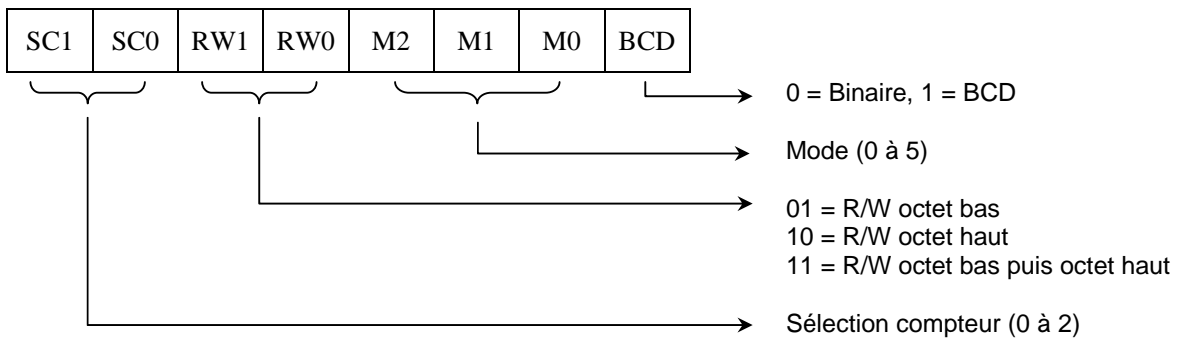
*** Format du mot de commande du PPI 8255 :**



*** Chronogrammes du mode 1 du compteur 0 du Timer 8253**



*** Format du mot de commande du Timer 8253 :**



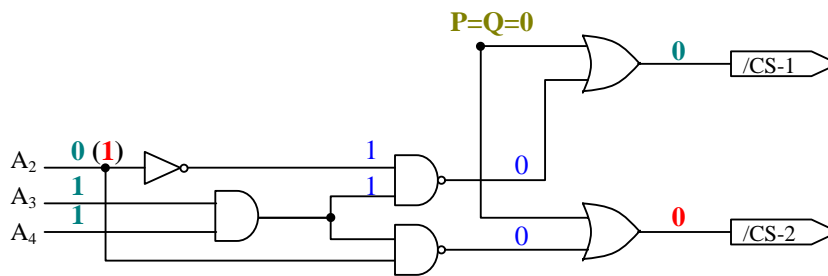
CORRIGE DU SUJET 1

1- Circuit de décodage d'adresses :

La sortie (P=Q) du comparateur 8 bits (74688) est égale à zéro si P=Q c'est à dire que :

$$\overline{A_6} \overline{A_{11}} A_{10} A_9 A_8 A_7 A_6 A_5 = P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$$

Les sorties, $\overline{CS-1}$ et $\overline{CS-2}$ sont égales à zéro si :



D'où

Adresse	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
$\overline{CS-1}$ 318H	0	0	1	1	0	0	0	1	1	0	0	0
$\overline{CS-2}$ 31CH	0	0	1	1	0	0	0	1	1	1	0	0

Entrées P6 à P0 du 74688
Entrées du circuit logique

Donc

PPI 8255		Timer 8253	
Port	Adresse	Compteur	Adresse
Port A	31CH	Compteur 0	318H
Port B	31DH	Compteur 1	319H
Port C	31EH	Compteur 2	31AH
Registre de contrôle	31FH	Registre de contrôle	31BH

2- Programmation

✳ *Compte initial*

$$N = (T_o/T_i) = 20/(1/1200) = 24000 \text{ (acceptable car } N < 65535).$$

✱ *Mots de commande*

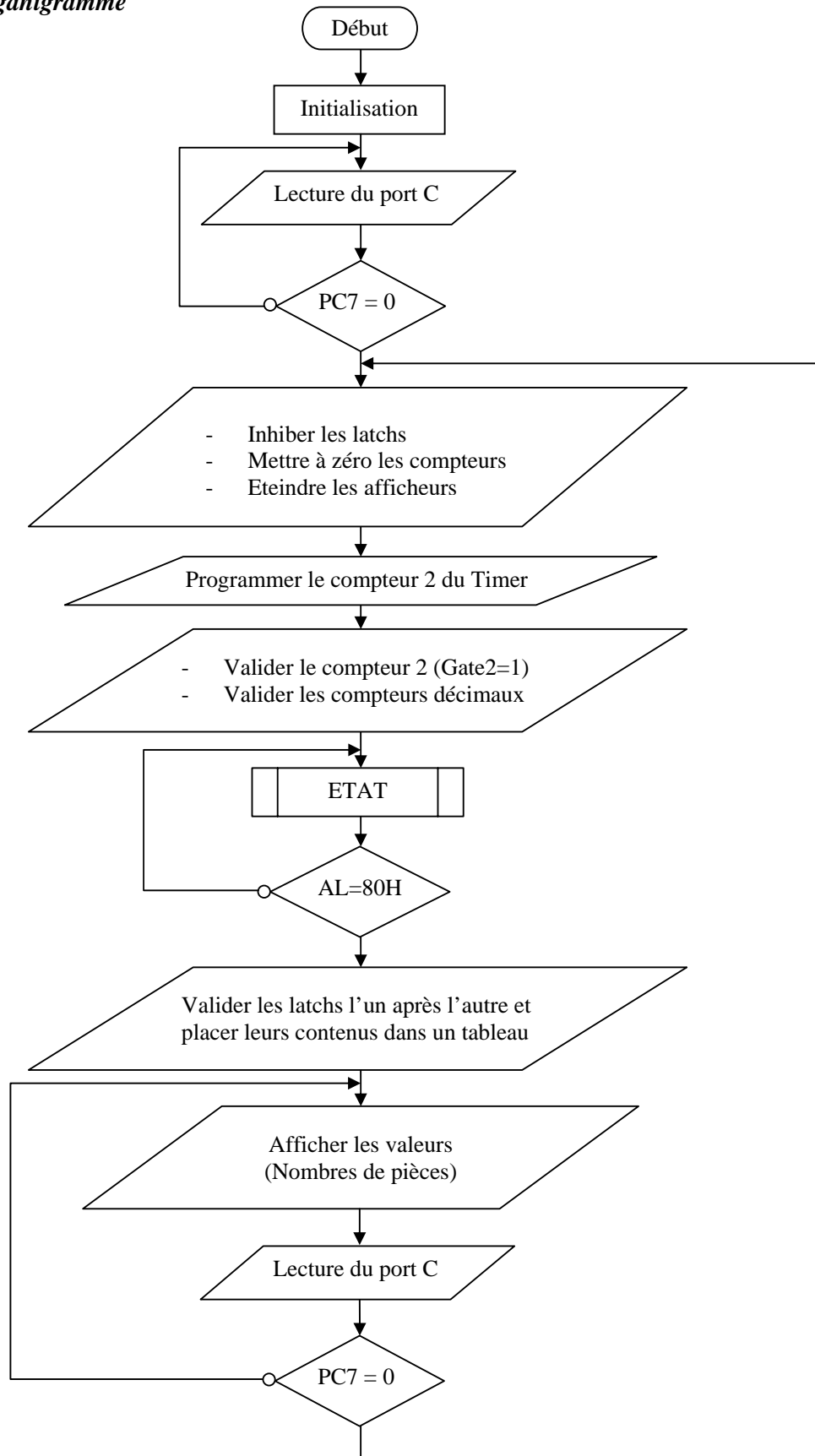
✱ PPI 8255 :							
1	0	0	1	1	0	0	1
→ 82 H							

✱ Timer 8253 :							
1	0	1	1	0	0	1	0
→ B2 H							

✱ *Commande des afficheurs*

Chiffre	g	f	e	d	c	b	a	Octet
	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
0	0	1	1	1	1	1	1	3FH
1	0	0	0	0	1	1	0	06H
2	1	0	1	1	1	1	1	5BH
3	1	0	0	1	1	1	1	4FH
4	1	1	0	0	1	1	0	66H
5	1	1	0	1	1	0	1	6BH
6	1	1	1	1	1	0	1	5BH
7	0	0	0	0	1	1	1	07H
8	1	1	1	1	1	1	1	7FH
9	1	1	0	1	1	1	1	6FH

★ *Organigramme*



★ **Programme**

```

DATA SEGMENT
    MOTC1 EQU 89H
    PORTA EQU 31CH
    PORTB EQU 31DH
    PORTC EQU 31EH
    RCNT1 EQU 31FH
    MOTC2 EQU 0B2H
    CMPT2 EQU 31AH
    RCNT2 EQU 31BH
    Nombre DB 3 dup(?)
    TABLE DB 3FH,06H,5BH,4FH,66H
           DB 6BH,5BH,07H,7FH,6FH
    N DW 2400
DATA ENDS
;
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    mov ax,DATA
    mov ds,ax
; Programmation du PPI
    mov dx,RCNT1
    mov al,MOTC1
    out dx,al
; Programmation du Timer
    mov dx,RCNT2
    mov al,MOTC2
    out dx,al
; Lecture du port C
Lire: Call Lecture
    Test al,80H
    jz lire
; désactiver les latches
dep: mov dx,PORTA
    mov al,0B8H
    out dx,al
; programmation du compteur 2
    mov dx,CMPT2
    mov ax,N
    out dx,al
    mov al,ah
    out dx,al
; validation du compteur 2
    mov dx,PORTA
    mov al,40H
    out dx,al
;
apl: call ETAT
    test al,80H
    jz apl
; valider et lire les latches
    mov cx,3
    xor di,di
    mov ah,36H
bcle1:mov dx,PORTA
    and al,38H
    out dx,al
    mov dx,PORTC
        In al,dx
        and al,0FH
        mov bx,offset TABLE
        xlat TABLE
        mov Nombre[di],al
        inc di
        rol ah,1
        loop bcle1
; affichage
aff:  mov cx,3
    mov al,1
    xor si,si
bcle2:mov dx,PORTA
    out dx,al
    mov ah,al
    mov dx,PORTB
    mov al,Nombre[si]
    out dx,al
    call TEMPO
    inc si
    rol ah,1
    mov al,ah
    loop bcle2
    call Lecture
    test al,80H
    jz aff
    jmp dep
    mov ah,4cH
    int 21H
LECTURE PROC NEAR
    mov dx,PORTC
    in al,dx
    not al
    ret
LECTURE ENDP
TEMPO PROC NEAR
    push
    mov cx,0ffffH
    t1: nop
        nop
        nop
        loop t1
    pop cx
    ret
TEMPO ENDP
ETAT PROC NEAR
    :
    :
    ret
ETAT ENDP
CODE ENDS
    END

```

CORRIGE DU SUJET 2

1- Circuit de décodage d'adresses :

- 1.1- Le circuit intégré 74688 est un comparateur 8 bits. Sa sortie (P=Q) est à l'état bas lorsque ses entrées P et Q sont égales c'est-à-dire $(P_7...P_0) = (Q_7...Q_0)$. Ceci n'est possible que si l'entrée G est mise à zéro.
- 1.2- Le circuit intégré 74245 est un buffeur de données bidirectionnel. Lorsque son entrée DIR est égale à 0, les données
- 1.3- Adresses :

Adresse	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
30CH	0	0	1	1	0	0	0	0	1	1	0	0
314H	0	0	1	1	0	0	0	1	0	1	0	0

Entrées P6 à P0 du 74688

Entrées
d'adresse du
74138

011 → Y3

101 → Y5

D'où

PPI 8255		Timer 8253	
Port	Adresse	Compteur	Adresse
Port A	30CH	Compteur 0	314H
Port B	30DH	Compteur 1	315H
Port C	30EH	Compteur 2	316H
Registre de contrôle	30FH	Registre de contrôle	317H

- 2- L'afficheur est du type à Anodes communes. Pour allumer un segment, il faut mettre la borne CA à 1 et l'entrée du segment à 0. Or d'après le circuit de la figure 3 on a :

$$CA = \overline{PA7} \quad \text{donc PA7 doit être égale à 0.}$$

Chiffre	CA	g	f	e	d	c	b	a	Octet
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
0	0	1	0	0	0	0	0	0	40H
1	0	1	1	1	1	0	0	1	79H
2	0	0	1	0	0	1	0	0	24H

- 3- $N = (T_o/T_i) = 10/(1/1200) = 12000$ (acceptable car $N < 65535$).

- 4- Mots de commande du PPI et du Timer.

* PPI 8255 :							
1	0	0	0	0	0	1	0
→ 82 H							

* Timer 8253 :							
0	0	1	1	0	0	1	0
→ 32 H							

5- Organigramme :

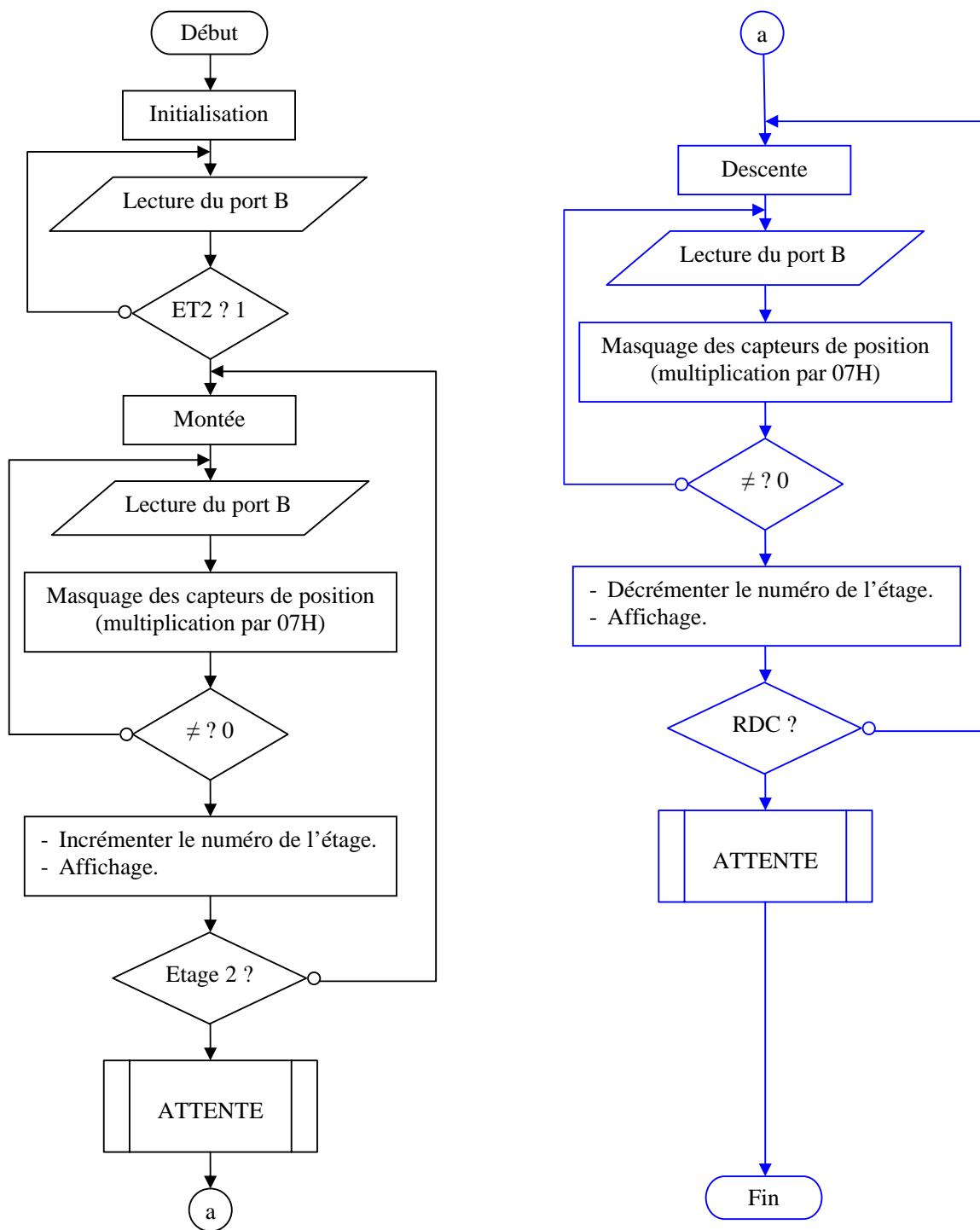
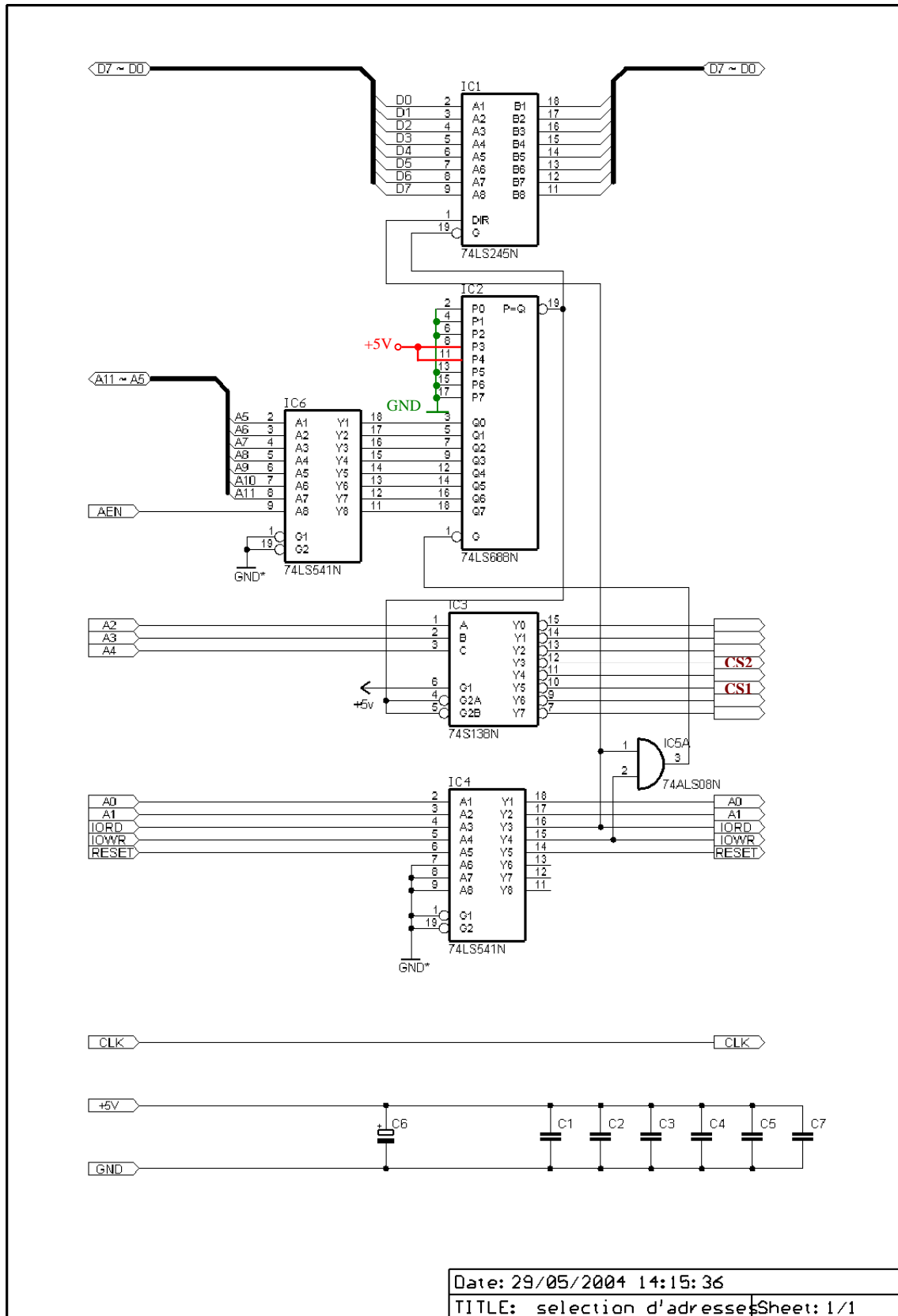


Figure 2 : Organigramme.



Date: 29/05/2004 14:15:36
 TITLE: selection d'adresses Sheet: 1/1

Figure 4 : Circuit de décodage d'adresses.

Programme

```

DATA SEGMENT
    MOT1 EQU 32H
    CMPT0 EQU 314H
    RCNT2 EQU 317H
    MOT2 EQU 82H
    PORTA EQU 30CH
    PORTB EQU 30DH
    PORTC EQU 30EH
    RCNT1 EQU 30FH
    ETAGE DB 0
    TABLE DB 40H,79H,24H
    N DW 12000
DATA ENDS
;
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    mov ax,DATA
    mov ds,ax
; Programmation du Timer
    mov dx,RCNT1
    mov al,MOT1
    out dx,al
; Programmation du PPI
    mov dx,RCNT2
    mov al,MOT2
    out dx,al
; Affichage de 0
    call AFFICHAGE
; Lecture du port B
    mov dx,PORTB
11: in al,dx
    test al,20H
    jz l1
; Commande de la montée
mo: mov dx,PORTC
    mov al,01H
    out dx,al
; Lecture du port B
    mov dx,PORTB
12: in al,dx
    and al,07H
    jz l2
    inc ETAGE
    push ax
    call AFFICHAGE
    pop ax
    test al,04H
    jz monte
    call ATTENTE
de: mov dx,PORTC
    mov al,02H
    out dx,al
; Lecture du port B
    mov dx,PORTB
13: in al,dx
    and al,07H
    jz l3

```

```

    dec ETAGE
    push ax
    call AFFICHAGE
    pop ax
    test al,01H
    jz desc
    call ATTENTE
    mov ah,4cH
    int 21H
;
AFFICHAGE PROC NEAR
    mov al,ETAGE
    mov bx,offset TABLE
    xlat TABLE
    mov dx,PORTA
    out dx,al
    ret
AFFICHAGE ENDP
ATTENTE PROC NEAR
; Arrêter le moteur cabine et
; Ouvrir la porte
    mov dx,PORTC
    mov al,04h
    out dx,al
; Ecriture du compte initial
    mov dx,CMPT0
    mov ax,N
    out dx,al
    mov dx,PORTB
14: in al,dx
    test al,80H
    jnz l4
; Arrêter le moteur porte
    mov dx,PORTC
    mov al,00h
    out dx,al
    mov dx,PORTB
15: in al,dx
    test al,80H
    jz l5
; Fermeture de la porte
    mov dx,PORTC
    mov al,08H
    out dx,al
    mov dx,PORTB
16: in al,dx
    test al,40H
    jz l6
; Arrêter le moteur porte
    mov dx,PORTC
    mov al,00h
    out dx,al
    ret
ATTENTE ENDP
CODE ENDS
END

```