

Université François-Rabelais de Tours

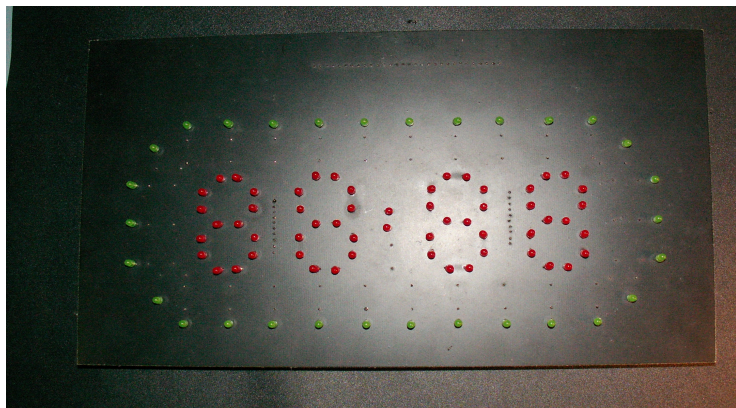
Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle



## **Projet d'étude et réalisation :**

# **Horloge numérique à Diodes Electroluminescentes pour automobile**



Julien JACQUIN  
Clément TREMBLIER  
2<sup>ème</sup> Année – Q2  
Promotion 2006/2008

Enseignants :  
Mme. DA COSTA I.  
Mr. LEQUEU T.

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle



**Projet d'étude et réalisation :**

**Horloge numérique à Diodes  
Electroluminescentes pour  
automobile**

Julien JACQUIN  
Clément TREMBLIER  
2<sup>ème</sup> Année – Q2  
Promotion 2006/2008

Enseignants :  
Mme. DA COSTA I.  
Mr. LEQUEU T.

## Table des matières

Introduction.....	4
1.Présentation du projet.....	5
1.1.Le cahier des charges.....	5
1.2.Généralités .....	5
2.L'étude du projet.....	6
2.1.Etude de la partie électronique.....	6
2.2.Etude de la partie informatique.....	13
3.Tests effectués .....	16
Conclusion.....	17
Résumé.....	18
ANNEXES.....	20

# Introduction

Durant la formation de DUT GEII, au cours du quatrième semestre, les cours d'études et réalisations nous permettent de mettre en oeuvre un projet.

Ce projet est soit proposé par l'enseignant, soit proposé directement par un groupe d'étudiants et ensuite approuvé par l'enseignant afin d'autoriser sa création.

Le projet que nous avons décidé de réaliser est un projet que nous avons proposé, nous sommes donc partis de rien, il s'agit d'une horloge numérique sept segments à diodes électroluminescentes qui sera alimentée par une tension de 12 Volts, donc intégrable dans une automobile.

# 1. Présentation du projet

## 1.1. Le cahier des charges

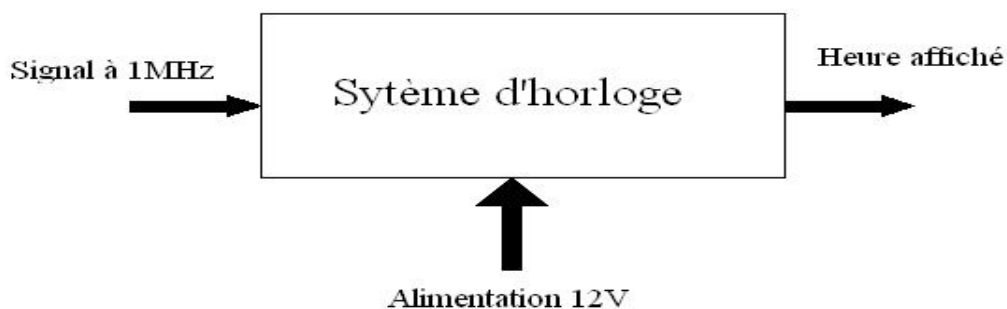
Afin de mener à bien la réalisation de ce projet, nous nous sommes fixé un certain nombre de contraintes :

- Dimension de la carte réduite,
- Bonne visibilité de l'affichage,
- Bonne précision de l'heure indiquée,
- Consommation limitée des afficheurs,
- Possibilité de coupure des afficheurs par interrupteur tout en gardant la partie de calcul de l'heure,
- Possibilité de programmer le composant In-Situ afin de faciliter les tests
- Intégration de deux boutons de réglage,
- Réalisation de la carte avec un certain esthétisme.

Toutes ces contraintes nous ont obligé à faire l'analyse technique suivante.

## 1.2. Généralités

Le but du projet était « simple », il répondait au bloc fonctionnel de niveau 1 suivant :



*Illustration 1: Bloc fonctionnel de niveau 1*

Afin que notre projet corresponde au mieux avec le cahier des charges, nous avons dû mettre en oeuvre différentes solutions technologiques, certaines, de par leur complexité de mise en oeuvre, ont dû être abandonnées, nous en parlerons brièvement.

Afin de mener à bien ce projet, nous avons décidé de nous partager le travail, l'un des deux membres du binôme a travaillé sur la partie programmation du projet tandis que l'autre a travaillé de son côté sur la partie électronique.

Au cours de ce rapport, les deux parties seront traitées séparément, nous aurons alors une première partie qui abordera les généralités de la carte, puis une partie où l'on détaillera le système, tant du point de vue informatique qu'électronique, ainsi que les solutions mises en oeuvre pour respecter le cahier des charges. Dans un troisième temps, nous détaillerons les procédures de test réalisées afin de tester et valider le fonctionnement du projet.

## 2. L'étude du projet

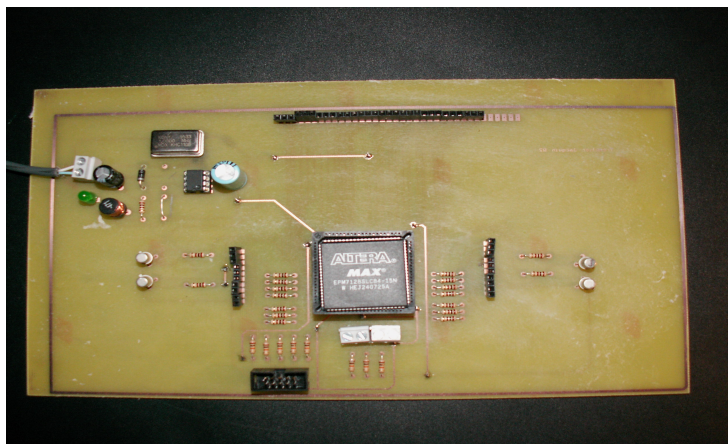
### 2.1. Etude de la partie électronique

Afin de réaliser la carte, nous avons dû prévoir comment s'organiserait ce projet, les différentes parties qui le constitueraient et comment nous ferions pour que le projet soit conforme aux contraintes que nous nous sommes imposées.

Dans un souci d'esthétisme, nous avons décidé de réaliser ce projet à l'aide de deux cartes, la première sera la carte dite « principale », elle comportera l'alimentation, le composant programmable ainsi que plusieurs autres fonctions.

La deuxième carte sera la carte affichage, elle comportera les diodes nécessaires à la création de l'horloge ainsi que les résistances pour les diodes des secondes.

Cette partie sera découpée en plusieurs sous-parties, reprenant chacune une fonction de la carte.



*Illustration 2: Photo de la carte principale*

## Partie 1 : L'alimentation

La première partie du projet a été de penser à fournir l'énergie nécessaire au bon fonctionnement du projet. Le bloc alimentation servira à fournir l'énergie nécessaire afin de faire fonctionner le composant programmable (CPLD), l'horloge de 1MHz ainsi que les différents interrupteurs prévus.

L'alimentation que nous avons prévu initialement a été réalisée à partir d'un circuit LM2574-ADJ, dont la particularité est de pouvoir régler la tension de sortie grâce à des résistances dites de retour.

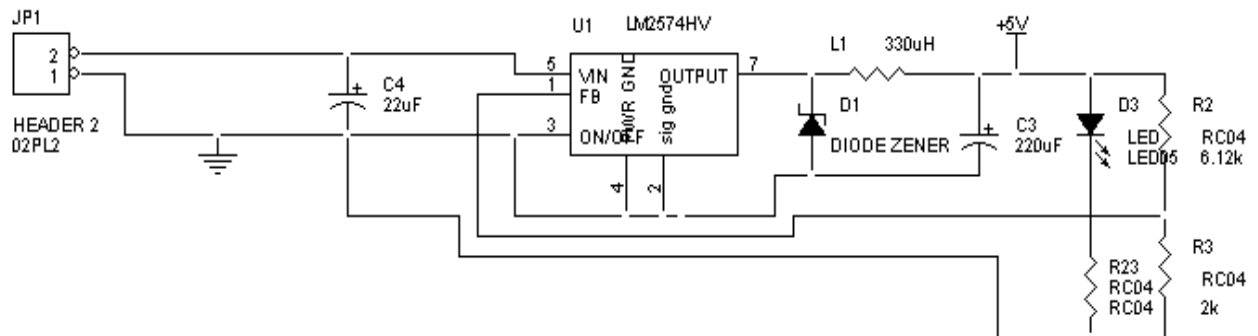


Illustration 3: Schéma de la partie alimentation

Le bornier JP1 est notre bornier d'arrivée, il sera connecté au plus de la batterie (+12 Volts) et à la masse. Le condensateur C4 sert à lisser la tension d'entrée pour « filtrer » les imperfections du signal d'entrée.

Le composant U1 (LM2574-ADJ) est un régulateur à découpage ajustable. Il a été placé afin d'obtenir une tension de +5 Volts grâce à l'association des deux résistances R2 et R3.

Les deux résistances R2 et R3 ont été choisies de façon à régler le régulateur à découpage pour obtenir une tension de +5V à la sortie de celui-ci.

La diode électroluminescente R3, associé à la résistance R23, sert à visualiser si la carte est alimentée correctement. Lorsque la diode est allumée, cela signifie que le régulateur fonctionne normalement.



*Illustration 4: Photo de la partie alimentation*

### Partie 2 : L'horloge

Afin d'obtenir une haute précision de la pendule, il a fallu « cadencer » le CPLD, c'est à dire lui indiquer quelle fréquence de calcul il aurait.

Nous avons choisi de placer une horloge à 1 Méga Hertz, notre composant programmable effectuera donc 1 million de calcul par seconde. En choisissant cette fréquence, nous avons prévu, au cas où le composant ne prendrait pas en compte certains passages de l'horloge, de minimiser ces effets. En effet, même si le CPLD « oubliait » 100 impulsions d'horloge sur 1 million, cela créerait un décalage de seulement 0,0001 seconde.

Afin de réaliser cette horloge, nous avons penser à faire un montage à base de quartz et de porte inverseuse, cependant après quelques recherches, nous avons trouvé un composant qui, lorsqu'on l'alimente sous 5 Volts, nous permet d' obtenir directement le signal carré cadencé à 1 Méga Hertz.



*Illustration 5: Photo de l' horloge 1MHz*

### Partie 3 : Le CPLD



*Illustration 6: Photo du CPLD*



Le CPLD (Complex Programmable Logic Device) est le coeur du projet. C'est grâce à ce composant programmable que l'affichage de l'heure va être possible.

Le CPLD utilisé durant ce projet est un Max 7128SLC84-15.

Ce composant comporte 84 pattes, et 68 d'entre elles peuvent servir d'entrées où de sorties. Les autres pattes sont utilisées pour alimenter ce composant.

La difficulté de mise en application de ce type de composant réside dans le fait que le routage des pistes s'avère délicat, étant donné le nombre conséquent de pattes à connecter.

Nous avons choisi ce CPLD car il nous fallait assez d'entrées/sorties (nous en utilisons environ 53), de plus, ce composant a la particularité d'être programmable « In-Situ », c'est à dire sans avoir à le déconnecter de la carte, ce qui permet de gagner du temps dans les tests.

Nous avons prévu, comme indiqué ci-dessus, de cadencer le CPLD à une fréquence de 1MHz, la fréquence d'horloge maximum que l'on peut lui imposer étant de 147MHz selon le manuel du constructeur, nous sommes bien en dessous de ses capacités maximales de calcul.

Sur le CPLD, nous avons 7 sorties pour commander les segments des heures et 2 pour commander soit l'allumage des unités, soit des dizaines. Nous avons également 9 autres sorties pour les minutes, selon le même procédé.

Afin de faciliter la mise en place, nous avons choisi de commander une diode d'allumage de seconde par sortie. Cela nous utilise donc 30 autres sorties. Une sortie est également nécessaire pour l'affichage des deux points servant à séparer les heures des minutes.

Nous avons placé 3 entrées, l'une est l'interrupteur marche/arrêt, les deux autres sont les boutons de réglages.

Enfin nous avons une entrée d'horloge afin de lui transmettre sa fréquence de calcul.

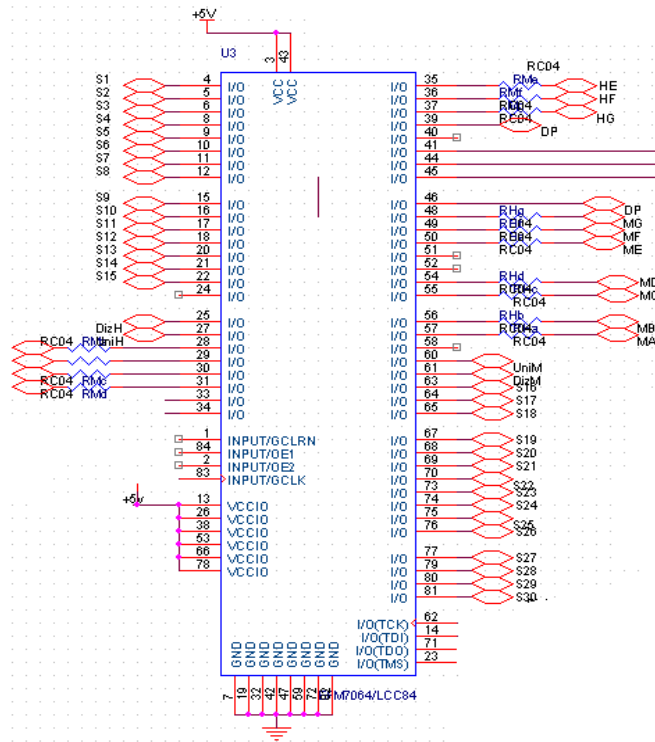


Illustration 7: Schéma du CPLD avec visualisation des entrées/sorties

#### Partie 4 : La programmation In-Situ et l'adaptateur JTAG

Afin de réaliser une programmation dite « In-Situ » pour notre projet, nous avons dû rajouter un petit module comportant un connecteur 10 broches ainsi que 5 résistances. Avec le brochage correct du connecteur sur certaines pattes du CPLD, cela nous permet de rendre possible, via un adaptateur JTAG, la programmation du composant.

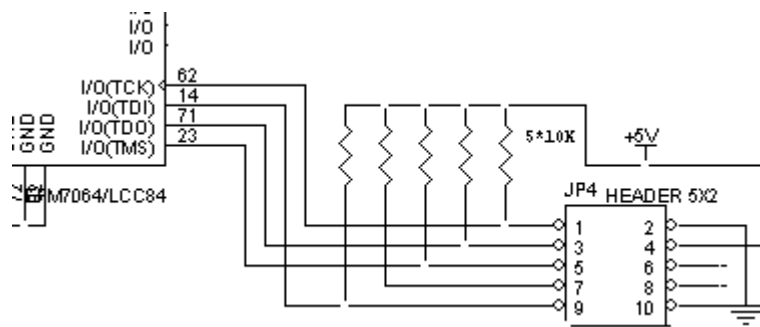


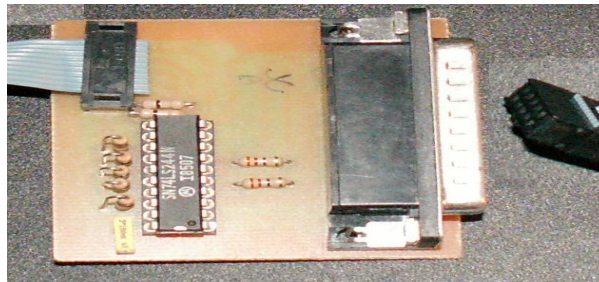
Illustration 8: Schéma du module de programmation In-Situ



*Illustration 9: Photo du module de programmation In-Situ*

L'adaptateur JTAG sert à établir la connexion entre l'ordinateur et le CPLD afin de procéder à sa programmation.

Pour cela, il faut réaliser un adaptateur qui reliera le port 25 broches de l'ordinateur au connecteur 10 broches de la carte.



*Illustration 10: Photo de l'adaptateur JTAG*

### Partie 5 : Le multiplexage des dizaines et des minutes

Le multiplexage est un procédé permettant d'économiser de l'énergie pour notre projet.

Grâce à ce principe, nous n'aurons pas besoin d'alimenter les 4 afficheurs en même temps. Nous en alimenterons seulement deux, puis les deux autres et ainsi de suite. Cela permet de réduire la consommation de la carte et de minimiser le nombre de sorties nécessaires sur le CPLD (18 sorties en multiplexage contre 28 sans).

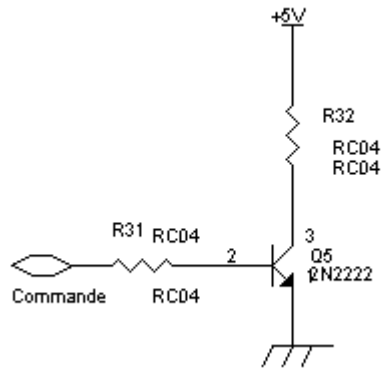


Illustration 11: Schéma de principe du multiplexage

Le principe du multiplexage est le suivant :

En envoyant une tension de commande de +5V, le montage ci-dessus permet de saturer le transistor (Q5), celui-ci va alors se comporter comme un fil entre R32 et la masse, permettant l'allumage des diodes (placées en amont de la résistance R32).

En envoyant une tension de commande de 0V, le montage ci-dessus permet de bloquer le transistor, celui-ci va alors se comporter comme un interrupteur ouvert entre R32 et la masse, les diodes ne pourront donc pas s'allumer.

En saturant un transistor pour les unités et bloquant celui des dizaines, un afficheur sur les deux fonctionnera (celui des unités), ensuite on inversera les commandes et on allumera ainsi l'afficheur des dizaines.



Illustration 12: Photo des transistors de multiplexage (à gauche)

## Partie 6 : L'affichage des secondes

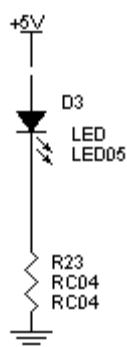


Illustration 13: Schéma de câblage des diodes de seconde

L'affichage des secondes est plus simple que la réalisation des afficheurs. On a disposé seulement une diode en série avec une résistance, si on met un potentiel de +5V sur la diode, celle-ci s'allumera, sinon, elle restera éteinte. Le potentiel que l'on applique sur la diode est contrôlé par le CPLD.

### 2.2. Etude de la partie informatique

Afin de réaliser le programme à insérer dans le CPLD, nous nous sommes servis du logiciel QUARTUS II qui permet de tester le programme sans forcément l'insérer dans le CPLD.

La partie programmation du CPLD a été faite de manière modulaire, ce choix est justifié par plusieurs avantages : Le programme est plus clair, on peut rajouter/supprimer des fonctionnalités plus facilement, mais surtout elle permet de tester indépendamment chaque partie du programme.

Comme chacun des modules a une utilité propre, nous allons expliquer chacun des modules :

#### Partie 1 : Le module de comptage

Ce module est le plus important du programme, c'est lui qui effectue le comptage des secondes, minutes et heures. En raison de la longueur de ce module, il faudra se référer à l'annexe 1 contenant le programme.

Dans ce module, on trouve aussi la partie réglage, une partie qui a posé problème à cause de la vitesse de l'horloge qui détectait plusieurs fois le bouton en un seul appui, on a donc eu besoin d'avoir recours à des variables internes pour ne détecter uniquement que le front montant ( actifR et actifI ).

L'autre difficulté de ce module fut de compter les heures, les minutes et les secondes sur 8 bits chacun ( 4 pour les unités et 4 pour les dizaines ). Pour cela on a dû avoir recours à l'astuce suivante car avec 4 bits, on compte jusqu'à 15 et non jusqu'à 9 :

```

if((S&8'h0F)==8'h0A)
    begin
        S=S+6;
    end

```

Grâce à cette astuce, quand nos unités arrivent à 9, on incrémente les dizaines et l'on remet les unités à 0

Ce fut la principale difficulté de ce module, le reste étant une simple succession de compteurs.

### Partie 2 : Les modules d'affichage des heures et des minutes

Ces modules sont ceux qui nous permettent l'affichage de l'heure sur les diodes, pour expliquer son fonctionnement nous allons commenter une version abrégée du module :

```

always@(posedge clk) // Le module sera exécuté à chaque front
                                montant de clk
begin
    {D,U} = H; // On récupère les 4 bits des unités et les 4
                                bits des dizaines dans D et U

    if(ValidationAff!=1)
    begin
        if(MultiplexHU==0) //Test pour savoir si le multiplexage est sur
                                les unités où les dizaines
            begin
                case(U) // Si on fait un multiplexage sur les unités

```

```

0: Heure='h77;           //Les sorties prendront la valeur
                           correspondante à l'heure

1: Heure=8'h22;

.....

8: Heure=8'h7f;
9: Heure=8'h7b;

endcase

MultiplexHU=1;           // on active le multiplexage pour qu'il
MultiplexHD=0;           affiche les dizaines

end

```

Par la suite, on refait les mêmes tests pour l'afficheur des dizaines, puis on remet la variable MultiplexH à 0 pour réafficher les unités.

### Partie 3: Affichage des secondes

Ensuite vient le module d'affichage des secondes, c'est le plus simple de tous, son fonctionnement est le suivant :

```

always@(posedge Sec)           //A chaque front montant de Sec

begin

begin

if(S<=30)                     // On regarde si S est inférieur ou égal à 30

begin

Seconde=Seconde<<1;         // Si oui on fait un décalage à gauche en rentrant un 1

end

else

begin

Seconde=Seconde>>0;         // Sinon on fait un décalage à droite en rentrant un 0

end

end

end

```

### 3. Tests effectués

Les premiers tests que nous avons effectués en relation avec le projet ont été pratiqués sur les diodes. Nous avons essayé plusieurs résistances en série avec une diode pour savoir quelle valeur serait plus appropriée, puis nous avons simulé une hausse de tension pour savoir si la diode résisterait à une augmentation du courant la traversant.

Le deuxième test a été réalisé après la conception de la carte affichage, nous avons testé, en mettant un potentiel de +5V sur chaque partie de support connectée à une diode, si les diodes en question s'allumaient toutes sans exception.

Après quelques petits problèmes (certaines diodes étaient soudées à l'envers où des microcoupures entre les pistes), nous avons réussi à la faire fonctionner entièrement.

Tout au long de la création du programme établi pour faire fonctionner l'horloge, nous avons pu faire des simulations à l'aide du logiciel Quartus, afin d'identifier les erreurs et les résoudre. Cette étape permet, une fois la carte réalisée, de minimiser les problèmes dûs à la programmation et de réduire le nombre de programmation afin de fonctionner que cela fonctionne correctement.

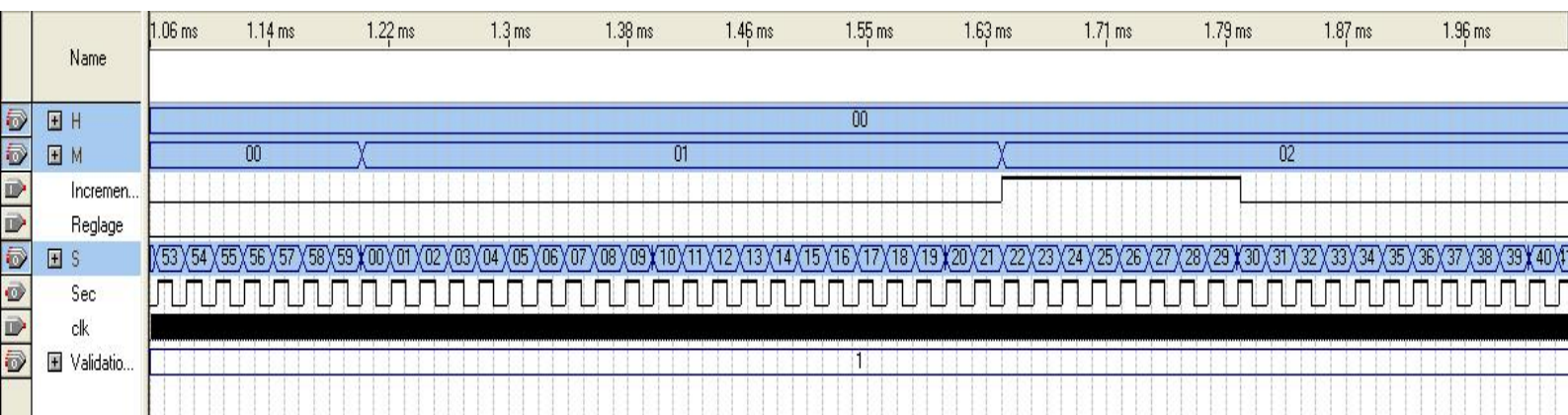


Illustration 14: Exemple de simulation avec le logiciel QUARTUS II

Avant de connecter le CPLD, des tests ont été effectués afin de vérifier que celui-ci soit bien alimenté, nous avons alors testé chaque alimentation et masse du CPLD avant de l'insérer dans son support.



# Conclusion

Notre projet, bien que simple dans son esprit, était complexe car les difficultés rencontrées, notamment au niveau de la taille de la carte ainsi que du routage de celle-ci, nous ont beaucoup ralenti.

Nous avons dû faire face, durant ce projet, à quelques imprévus, cependant nous avons appris à faire avec ces imprévus et nous avons pu surmonter les difficultés.

Le fait de travailler en binôme en parallèle, l'un sur la partie informatique et l'autre sur la partie électronique, nous a permis de voir un peu comment deux personnes ne faisant pas du tout la même chose sont amenées à collaborer et à parler afin de se mettre d'accord sur tel ou tel point.

Au final, nous avons effectué quelques tests sur la carte affichage et ils se sont tous montrés concluants, la carte principale, quant à elle, rentre dans sa phase terminale de programmation et de tests.

# Résumé

Tout au long de ce projet, nous avons élaboré des solutions techniques afin que notre projet réponde au mieux au cahier des charges, pour cela nous avons dû mettre en œuvre nos connaissances du langage VHDL et du logiciel Orcad.

Dans la continuité de ce projet, nous pouvons envisager l'élaboration d'un système beaucoup plus réduit et autonome au niveau de l'alimentation afin de minimiser la consommation de la carte.

De plus, nous pouvons envisager l'ajout de diverses fonctionnalités telles qu'un chronomètre ou encore une alarme.

## **Index des illustrations**

Illustration 1: Bloc fonctionnel de niveau 1.....	5
Illustration 2: Photo de la carte principale.....	6
Illustration 3: Schéma de la partie alimentation.....	7
Illustration 4: Photo de la partie alimentation.....	8
Illustration 5: Photo de l' horloge 1MHz.....	8
Illustration 6: Photo du CPLD.....	8
Illustration 7: Schéma du CPLD avec visualisation des entrées/sorties.....	10
Illustration 8: Schéma du module de programmation In-Situ.....	10
Illustration 9: Photo du module de programmation In-Situ.....	11
Illustration 10: Photo de l'adaptateur JTAG.....	11
Illustration 11: Schéma de principe du multiplexage.....	12
Illustration 12: Photo des transistors de multiplexage (à gauche).....	12
Illustration 13: Schéma de câblage des diodes de seconde.....	13
Illustration 14: Exemple de simulation avec le logiciel QUARTUS II.....	16

# **ANNEXES**



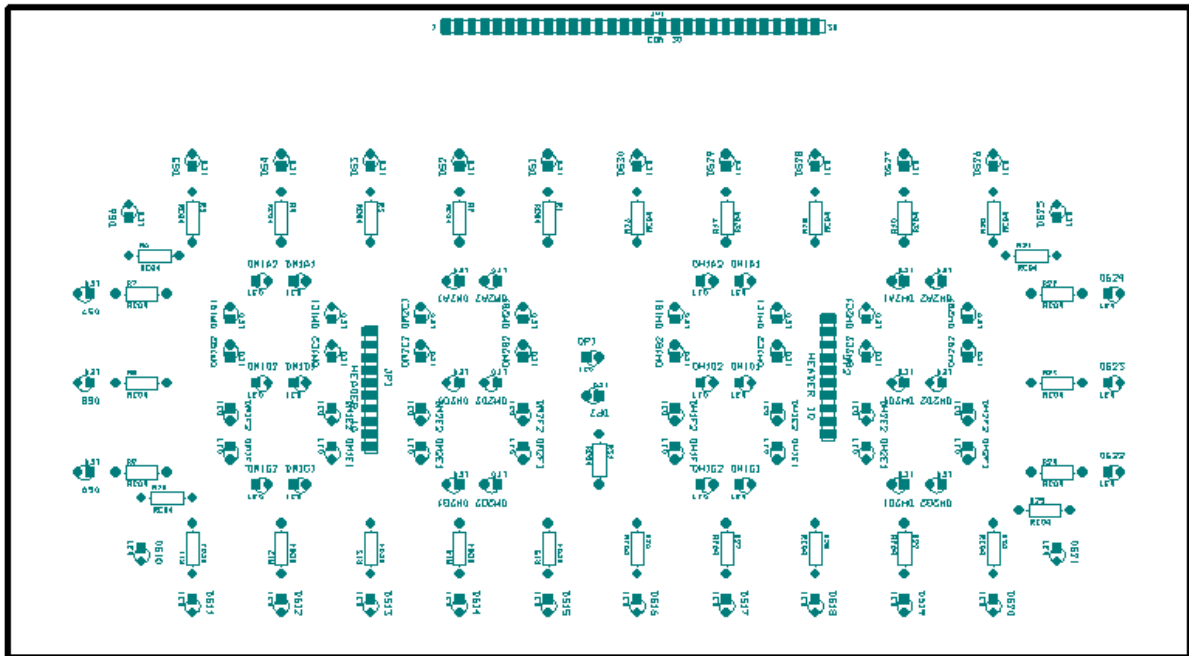
## Annexe 2 : Coût de la carte

Elément	Quantité	Prix Unitaire	Prix total
Epoxi 250*150	2	15	30
Résistance 1/4W 100Ohms	49	0,01	0,49
Résistance 1/4W 10KOhms	8	0,01	0,08
Transistor 2N2222	4	0,02	0,08
DEL 3mm Verte	30	0,1	3
DEL 5mm Verte	1	0,1	0,1
DEL 3mm Rouge	58	0,15	8,7
LM2574-5	1	2,16	2,16
Kony MDX KHC 1100	1	2,5	2,5
1N4007	1	0,11	0,11
Max7128SLC84-15	1	20,68	20,68
Inductance 470mH	1	0,5	0,5
Condensateur polarisé 22uF	1	0,9	0,9
Condensateur polarisé 220uF	1	1,18	1,18
Bornier à vis (x2)	3	0,2	0,6
Support pour barrette sécable(x50)	2	4	8
Barrette sécable droite (x50)	1	4,18	4,18
TOTAL			83,26

Les dimensions de l'epoxi sont en millimètres, tous les prix affichés sont en Euros et sont Hors Taxe.

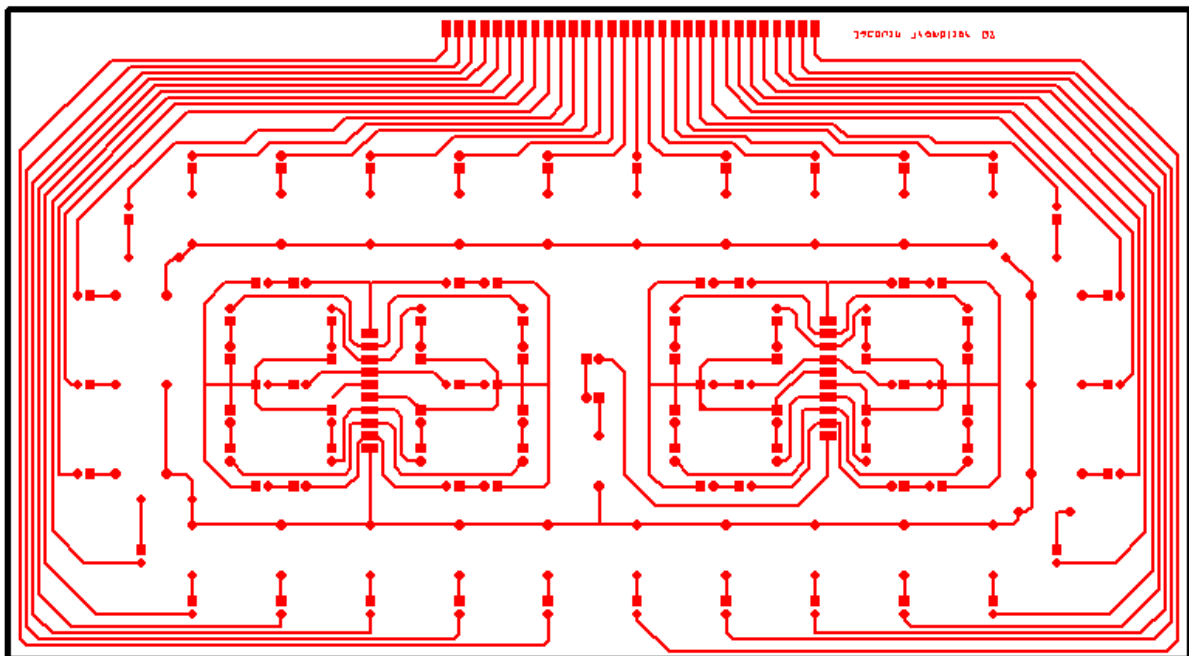
Après quelques recherches sur internet, nous avons trouvé une horloge similaire à la notre (bien qu'alimentée en 230V) pour environ 100€, nous serions donc bien placés dans le coût de ce projet.

# Annexe 3 : Schéma d'implantation des composants de la carte afficheur



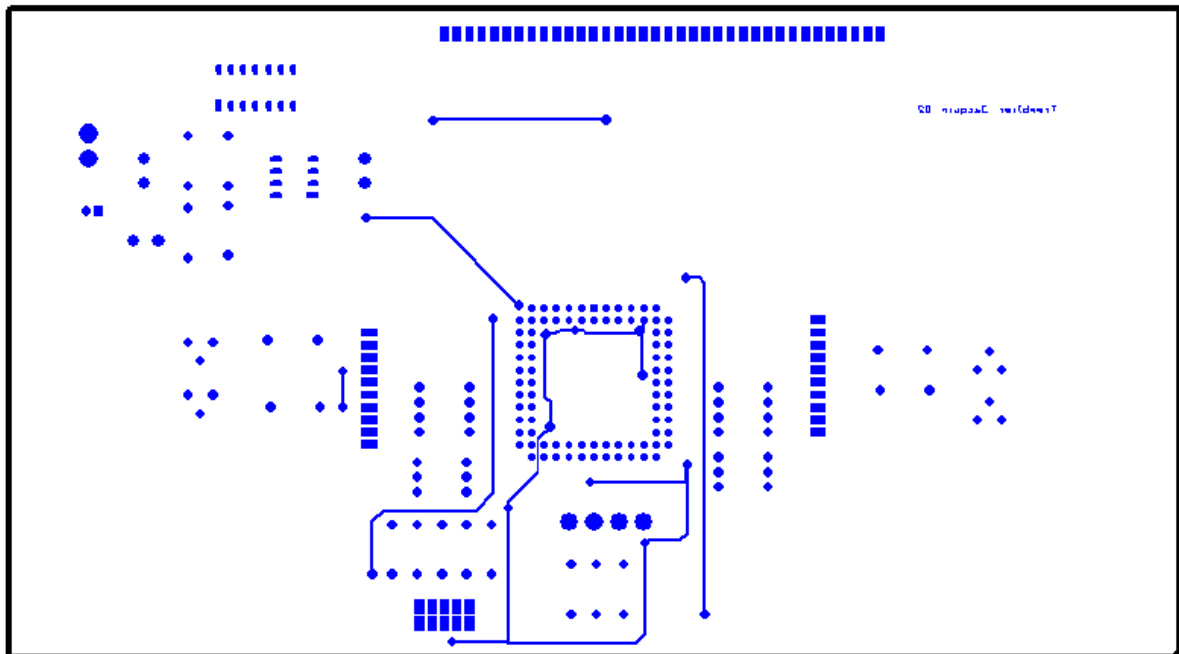
FACT AS1

## Annexe 4 : Typon côté cuivre de la carte affichage



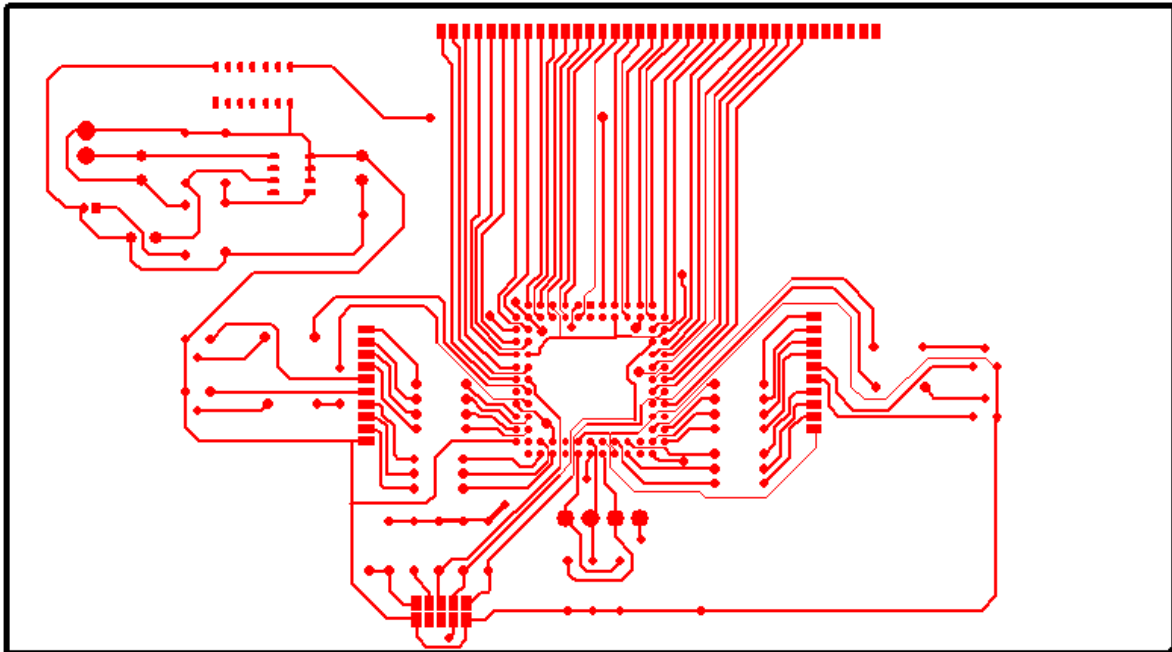


# Annexe 5 : Typon côté composant de la carte principale



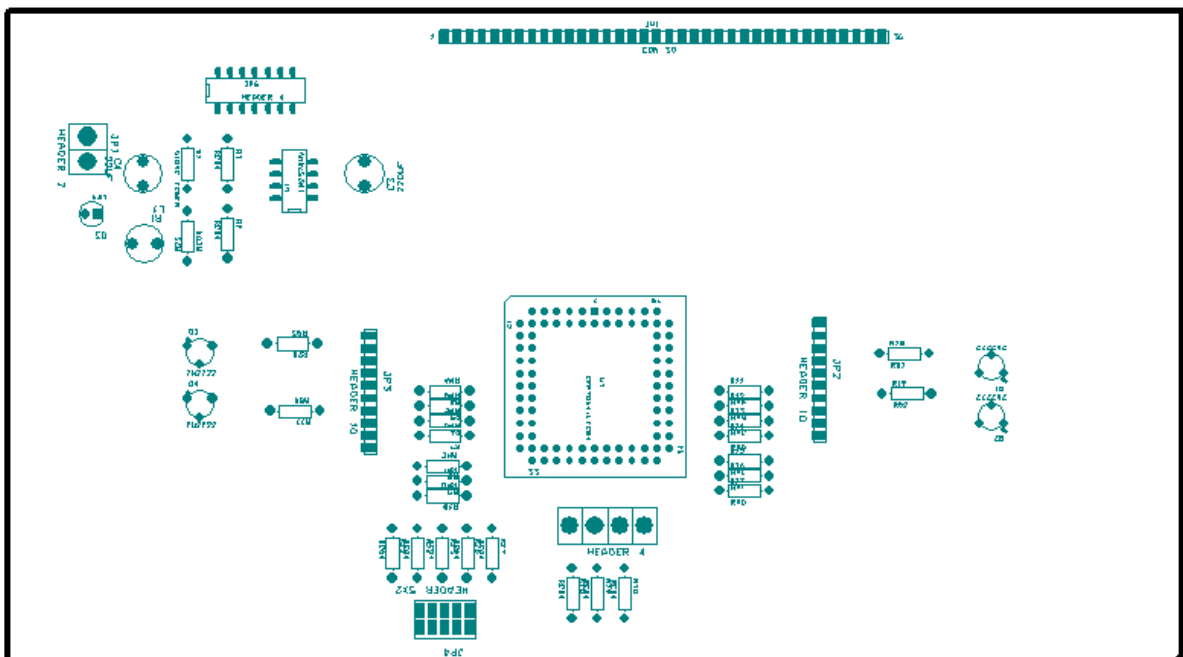
FACE TOP

## Annexe 6 : Typon côté cuivre de la carte principale



YDF 5367

# Annexe 7 : Schéma d'implantation des composants de la carte principale



## Annexe 8 : Programme

/\*Programme réalisé par Messieurs Jacquin et Tremblier dans le cadre du projet  
d'étude et réalisation du 4ème semestre\*/

```
module HorlogeV1 (clk, Heure,Minute, MultiplexHU, MultiplexMU, MultiplexHD,  
MultiplexMD, Seconde,Reglage, Incrementation);
```

```
    input clk,Reglage,Incrementation;                //Définition des entrées sorties du  
composant
```

```
    output [29:0]Seconde;
```

```
    output [6:0]Heure,Minute;
```

```
    output MultiplexHD,MultiplexMD,MultiplexHU, MultiplexMU;
```

```
    wire [7:0]H, M, HR, MR, S;
```

```
    wire Sec,ValidationAff;
```

```
    FP1 Compteur (clk,H,M,S,Sec,ValidationAff, Incrementation,Reglage);
```

```
    FP2 AffH (H,MultiplexHU,MultiplexHD,Heure,clk,ValidationAff);;
```

```
    FP3 AffM (M,MultiplexMD ,MultiplexMU ,Minute,clk,ValidationAff);
```

```
    FP4 AffS (S,Seconde,Sec);
```

```
endmodule
```

```

// Module de comptage de l'horloge

module FP1(clk,H,M,S,Sec,ValidationAff, Incrementation,Reglage);

input clk, Incrementation,Reglage;

output [7:0]H;
output [7:0]M;
output [7:0]S;
output Sec;
output [3:0]ValidationAff;

reg [7:0]H,M;
reg [7:0]S;
reg Sec;
reg [7:0]cpt;
reg [3:0]ValidationAff;
reg actifR,actifI;

always@(posedge clk)

begin

        cpt=cpt+1;

        // Début du réglage

        if(Reglage==1)

```

```

begin

    if(activR==0)
        begin
            ValidationAff=ValidationAff+1;
            activR=1;
            if(ValidationAff>=3)
                begin
                    ValidationAff=0;
                end
            end
        end
    end

    if(Reglage==0)
        begin
            activR=0;
        end
    end

    if(Incrementation==0)
        begin
            activf=0;
        end
    end

    if(Incrementation==1)
        begin

            if(ValidationAff==1)// Reglage Minutes
                begin
                    if(activf==0)
                        begin

```

```

                                actif=1;
                                M=M+1;
                                end

                                end

                                if(ValidationAff==2)// Reglage Heures
                                begin
                                    if(actif==0)
                                        begin
                                            actif=1;
                                            H=H+1;
                                        end
                                    end
                                end

                                end

                                end

                                //Fin du réglage

                                if(cpt<=9)
                                begin

                                    Sec=1;

                                end

                                if(cpt>10)
                                begin

                                    Sec=0;

                                end

                                end

                                if((S&8'h0F)==8'h0A)

```

```
begin
    S=S+6;
end
if(S==8'h60)
begin
    S=0;
    M=M+1;
end
if((M&8'h0F)==8'h0A)
begin
    M=M+6;
end
if(M==8'h60)
begin
    M=0;
    H=H+1;
end
if((H&8'h0F)==8'h0A)
begin
    H=H+6;
end
if(H==8'h24)
begin
    H=0;
end
```



```

        if(cpt==20)
            begin
                cpt=0;
                S=S+1;
            end
        end
    end
endmodule

// Module d'affichage des Heures

module FP2 (H,MultiplexHU,MultiplexHD,Heure,clk,ValidationAff);

input [7:0]H;
input clk,ValidationAff;

output MultiplexHU;
output MultiplexHD;
output [6:0]Heure;

reg [3:0]D;
reg [3:0]U;
reg [6:0]Heure;
reg MultiplexHU;
reg MultiplexHD;

```

```
always@(posedge clk)
```

```
begin
```

```
{D,U} = H;
```

```
if(ValidationAff!=1)
```

```
begin
```

```
    if(MultiplexHU==0)
```

```
        begin
```

```
            case(U)
```

```
            0: Heure=8'h77;
```

```
            1: Heure=8'h22;
```

```
            2: Heure=8'h6d;
```

```
            3: Heure=8'h6b;
```

```
            4: Heure=8'h3a;
```

```
            5: Heure=8'h5b;
```

```
            6: Heure=8'h5f;
```

```
            7: Heure=8'h62;
```

```
            8: Heure=8'h7f;
```

```
            9: Heure=8'h7b;
```

```
            endcase
```

```
            MultiplexHU=1;
```

```
            MultiplexHD=0;
```

```
        end
```

```
    else
```

```

begin
    case(D)
        0: Heure=8'h77;
        1: Heure=8'h14;
        2: Heure=8'h5b;
        3: Heure=8'h5d;
        4: Heure=8'h3c;
        5: Heure=8'h6d;
        6: Heure=8'h6f;
        7: Heure=8'h54;
        8: Heure=8'h7f;
        9: Heure=8'h7d;
    endcase
    MultiplexHU=0;
    MultiplexHD=1;
end

```

```
end
```

```
end
```

```
endmodule
```

```
// Module d'affichage des Minutes
```

```
module FP3 (M,MultiplexMD ,MultiplexMU ,Minute,clk,ValidationAff);
```

```
input [7:0]M;
```

```
input clk,ValidationAff;
```

```

output MultiplexMU;
output MultiplexMD;
output [6:0]Minute;

reg [3:0]Di;
reg [3:0]Un;
reg [6:0]Minute;
reg MultiplexMU;
reg MultiplexMD;

always@(posedge clk)

begin
  {Di,Un} = M;
  if(ValidationAff!=1)
  begin

      if(MultiplexMU==0)
      begin

          case(Un)
          0: Minute=8'h77;
          1: Minute=8'h22;
          2: Minute=8'h6d;
          3: Minute=8'h6b;
          4: Minute=8'h3a;

```

```

        5: Minute=8'h5b;
        6: Minute=8'h5f;
        7: Minute=8'h62;
        8: Minute=8'h7f;
        9: Minute=8'h7b;
        endcase
        MultiplexMU=1;
        MultiplexMD=0;

    end

else
    begin
        case(Di)
            0: Minute=8'h77;
            1: Minute=8'h14;
            2: Minute=8'h5b;
            3: Minute=8'h5d;
            4: Minute=8'h3c;
            5: Minute=8'h6d;
            6: Minute=8'h6f;
            7: Minute=8'h54;
            8: Minute=8'h7f;
            9: Minute=8'h7d;
            endcase
            MultiplexMU=0;
            MultiplexMD=1
        end
    end
end

```

```

end
endmodule

// module d'affichage des Secondes

module FP4 (S,Seconde,Sec);

output [29:0]Seconde;

input Sec ;
input [7:0]S;

reg[29:0]Seconde;
reg valid;
always@(posedge Sec)
begin
    if(S<=30)
        begin
            Seconde=Seconde<<1;
        end
    else
        begin
            Seconde=Seconde>>0;
        end
    end

end
endmodule

```