3ETI, Année scolaire 2020-2021
ELN2 : Projet scoring 2.0

Affichage de la durée et du score d'un match de football à l'aide d'un FPGA.

GROUPE C
Equipe 2

Corto DESCHAMPS
Axel FRANÇOIS
Elliot GARCIA
Sarah LE CORRE,

Introduction

Le projet Scoring 2.0 consiste à développer pour le compte d'un grand club de football, un circuit intégré numérique fournissant à un système d'affichage sur panneau lumineux, les scores des équipes et la durée d'une mi-temps. L'ensemble de l'électronique numérique développée devra occuper une surface de moins de 4 cm² pour un prix de revient inférieur à 10€.Il devra pouvoir évoluer et être adaptable à d'autres sports que le football.

Le système à développer doit disposer des 4 fonctionnalités suivantes :

1. la gestion du score de chacune des deux équipes,
2. le comptage du temps écoulé avec un arrêt automatique à la 45ème minute (mi-temps),
3. l'affichages score et du temps écoulé sur des afficheurs 7-segments,
4. l'affichage du temps écoulé et du score sur un écran LCD compatible avec la norme VGA.

Trois signaux sont nécessaires pour la gestion du score. Ils seront issus de trois boutons poussoirs:

1. incrémentation du score de chacune des équipes, boutons poussoirs BPL et BPV(«L «comme local, «V» comme «visiteurs»),
2. remise à zéro des scores des deux équipes, bouton poussoir BPreset.

Trois signaux sont nécessaires au contrôle du chronomètre. Ils seront issus de deux interrupteurs et d'un bouton poussoir:

1. démarrage du comptage du temps, interrupteur START,
2. mise en pause du comptage du temps, interrupteur WAIT_t,
3. remise à zéro du comptage du temps, bouton poussoir RESET.

Deux signaux sont nécessaires au contrôle de l'affichage sur écran compatible de la norme VGA. Ils seront issus de deux interrupteurs:

1. activation de la visualisation sur l'écran, interrupteur VGAONOFF,
2. activation de la visualisation d'un ensemble de mire pour tester la qualité de l'affichage, interrupteur TESTVGA

L'affichage du score et du temps écoulé s'effectuera sur les LEDs des 8 afficheurs 7-segments à anode commune. Les quatre afficheurs de droite indiqueront le temps écoulé en minutes et secondes. Les quatre afficheurs de gauche afficheront le score des deux équipes (afficheurs de gauche pour l'équipe locale, afficheurs de droite pour l'équipe des visiteurs). Le point de l'afficheur 2 (3ème à partir de la droite) permettra de séparer les minutes des secondes, il clignotera à la fréquence de 1Hz (T = 1s). L'affichage du score et du temps écoulé s'effectuera également sur un écran LCD disposant d'une entrée compatible de la norme VGA. La norme de l'écran est la norme VGA 800x600avec un taux de rafraîchissement de 72 Hz.

Le développement du système doit s'effectuer dans les règles du développement d'un système numérique synchrone afin d'éviter tout problème d'asynchronisme au cœur de l'électronique numérique. Pour cela, un seul et unique signal d'horloge activera l'ensemble des bascules D du système. Toutes les bascules seront actives sur front montant de l'horloge. La fréquence devra être précise et égale à100 MHz. Le développement du système devra également prendre en compte les problèmes de rebonds liés à l'utilisation de boutons poussoirs et d'interrupteurs. Les contraintes liées au coût, à l'encombrement et à la capacité d'évoluer de la solution ont orienté le choix du

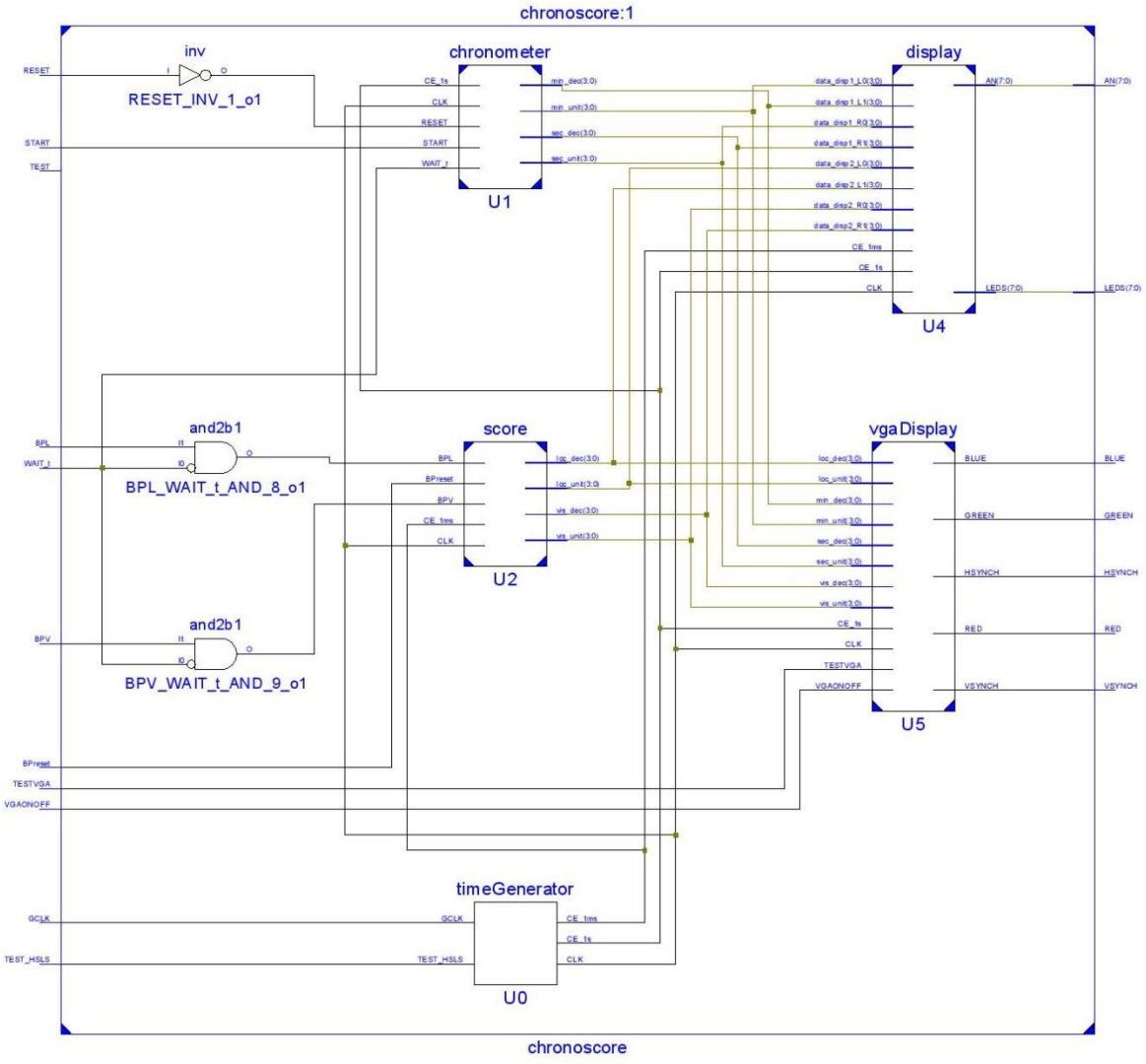
circuit numérique vers le circuit FPGAXC7A100Ten boitier CSG324de la famille Artix-7de la société Xilinx en technologie standard. La carte d'évaluation NEXYSA7 de la société Digilent sera utilisée pour tester la solution proposée. La carte sera alimentée via le PC par un câble micro-USB (fourni avec la carte). Ce câble sera également utilisé pour la programmation du FPGA. La carte sera alimentée avec une tension V_{cc} = 3,3V.

Les différentes phases du développement seront :

1. La gestion des afficheurs 7 segments de la carte NEXY A7 où sera développé le sous-bloc **display** qui permettra de mettre en œuvre un système d'affichage et de valider le fonctionnement des sous-blocs du système **chronoscore**.
2. La gestion du chronomètre et du score, qui consiste à afficher sur le système de la phase 1, le score des deux équipes et le temps écoulé par mi-temps (minutes et secondes).

Architecture générale du produit

Le synoptique général du système numérique **chronoscore** et le suivant :



Chronoscore est composé des sous-blocs :

- (U0) **timeGenerator** : L'horloge interne de la carte NEXYX A7 ayant une fréquence de 100MHz, ce sous-bloc sert à modifier cette fréquence pour l'ajuster aux valeurs souhaitées pour nos différentes horloges de période : CLK = 10 ns ; CE_1ms = 1ms (durée à l'état haut : de 10ns) ; CE_1s = 1s (durée à l'état haut : de 10ns).
- (U1) **chronometer** : Chronomètre dont les entrées permettent la mise en route, l'interruption et la remise à zéro du comptage
- (U2) **score** : Permet d'incrémenter des "points" à l'équipe "locale" ou "visiteuse" et de remettre à zéro les scores.
- (U3) **display** : gère les données à afficher sur les 8 afficheurs 7-segmentsde la carte NEXYX A7
- (U4) **vgaDisplay** : Permet l'affichage graphique via prise VGA du système **chronoscore** sur un écran.

Et de composantes logiques :

- 2 AND (2 entrées)
- 1 INV

Les entrées de **chronoscore** sont :

Signal	**Direction du signal**	**Elément de la carte NEXYS A7**	**Port du FPGA** *(fichier .ucf)*	**Description**
GCLK	Input	Quartz	E3	Oscillateur **Epson SG-8002JF** générant une fréquence de **50 MHz**
START	Input	SW0	J15	Démarrage du chronomètre
WAIT_t	Input	SW1	L16	Mise en pause du chronomètre
RESET	Input	BTN CPU RESET	C12	Remise à 0 du chronomètre
BPL	Input	BTNU	P18	Incrémentation du score pour l'équipe locale
BPV	Input	BTND	M18	Incrémentation du score pour l'équipe des « visiteurs »
BPreset	Input	BTNR	M17	Remise à zéro du score
VGA ONOFF	Input	SW4	R17	Activation de l'écran VGA
TEST VGA	Input	SW3	R15	Activation des images de test de l'écran VGA
TEST_ HSLS	Input	SW2	M13	Réservé
TEST	Input	BTNC	N17	Réservé

*Tableau 5 : Description des signaux d'entrée de **chronoscore***

Les sorties de **chronoscore** sont :

Signal	Direction du signal	Elément de la carte NEXYS A7	Port du FPGA *(fichier .ucf)*	Description
AN(0)	Output	Afficheurs 7-segments (anodes)	J17	AN[7 :0] : signaux de commande des anodes des afficheurs
AN(1)	Output		J18	
AN(2)	Output		T9	
AN(3)	Output		J14	
AN(4)	Output		P14	
AN(5)	Output		T14	
AN(6)	Output		K2	
AN(7)	Output		U13	
LEDS(0)	Output	Afficheurs 7-segments (cathodes)	T10	LEDS[6 :0] : signaux de commande des segments « a » à « g » LEDS[7] : signal de commande du point
LEDS(1)	Output		R10	
LEDS(2)	Output		K16	
LEDS(3)	Output		K13	
LEDS(4)	Output		P15	
LEDS(5)	Output		T11	
LEDS(6)	Output		L8	
LEDS(7)	Output		H15	
HSYNCH	Output	Port VGA	B11	Synchronisation horizontale
VSYNCH	Output		B12	Synchronisation verticale
RED	Output		A4	Contrôle des pixels de couleur « rouge »
BLUE	Output		B6	Contrôle des pixels de couleur « bleu »
GREEN	Output		D8	Contrôle des pixels de couleur « vert »

*Tableau 2 : Description des signaux de sortie de **chronoscore***

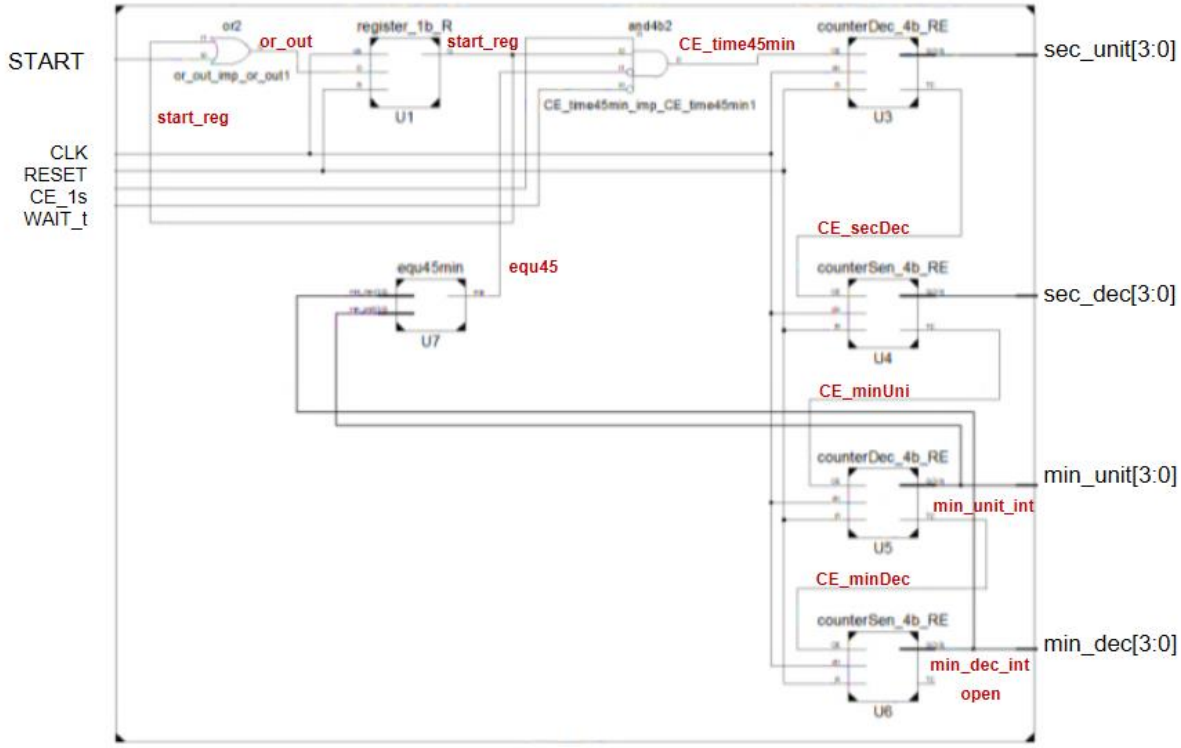
Etude fonctionnelle des sous-blocs chronometer et score

Sous bloc chronometer

Le sous bloc chronometer sert à fournir aux sous-blocs display et vgaDisplay les minutes et secondes à afficher. Ses fonctions sont donc :

- Compter l'écoulement des secondes et des minutes
- Gérer le début, l'interruption et l'arrêt de comptages
- Gérer les rebonds lorsque l'on commute l'interrupteur START

Le schéma RTL de la fonction est le suivant :



Il possède 5 signaux d'entrée:

- START, un signal qui vaut 0 si le chronomètre ne doit pas commencer à compter et 1 dans le cas inverse
- CLK, horloge de période 10 ns, donc 5 ns à l'état haut
- RESET, signal asynchrone de remise à 0 du chronomètre
- CE_1s , horloge de période 1 s avec une durée d'état haut de 10ns
- WAIT_t., signal qui vaut 1 si le chronomètre doit se mettre en pause et 0 dans le cas contraire

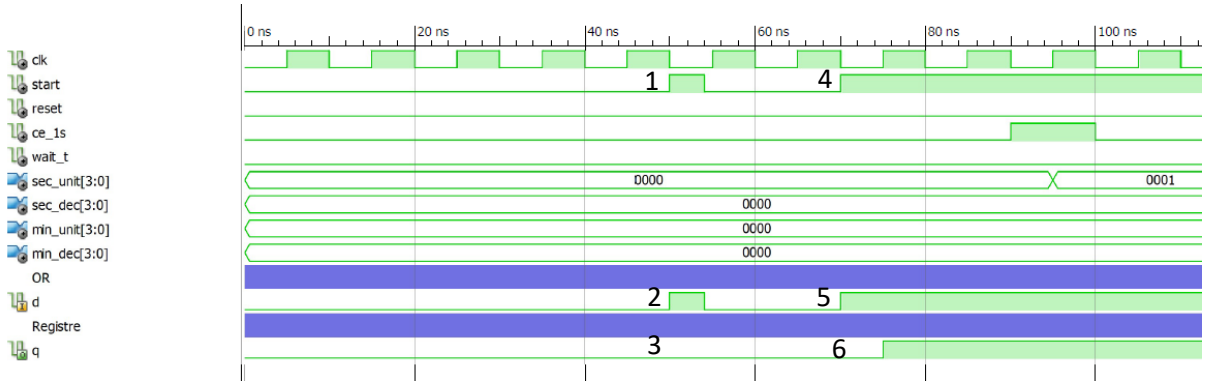
En ce qui concerne les sorties, nous avons 4 vecteurs de sorties de dimensions 4 :

- sec_unit, servant à compter de 1 en 1 l'unité des secondes à chaque front montant de l'horloge avec une remise à 0 lorsque RESET passe à 1 ou que sec_unit = "1001" = 9s (ce qui implique donc que sec_dec devient sec_dec + "0001").
- sec_dec, même principe pour sec_dec avec un comptage de la dizaine des secondes et donc une remise à 0 lorsque sec_dec = "0101" = 50s (ce qui implique donc que min_unit devient mi_unit + "0001").
- min_unit, même principe que sec_unit avec une influence sur min_dec au lieu de sec_dec.
- min_dec, même principe que sec_dec.

Gestion les rebonds lorsque l'on commute l'interrupteur START

La gestion des rebonds est réalisée par or2 et register_1b_R. En comparant START à sa valeur précédente grâce à un registre, nous pouvons prévenir d'un disfonctionnement causé par une pression involontaire.

Voici par exemple, un cas de fonctionnement de la gestion de rebonds



Nous pouvons voir que l'utilisateur appuie sur START entre 50 et 54 ns (1), ce qui correspond à un temps très court et il s'agit donc d'un rebond. Le système détecte donc cette anomalie (2) car elle est trop courte pour se trouver sur le front montant de clk et n'enregistre pas la pression comme le lancement du comptage : la sortie du registre reste nulle (3). En revanche, lorsque l'interrupteur est bien enclenché (4), le système conserve s'information pendant un temps suffisamment long pour observer un front montant de clk (5) et l'information que le comptage doit débuter est conservée (6)

Réalisation de l'entité or2

Il s'agit d'une simple porte Or. Les équations des sorties de la fonction sont définies à partir de sa table de vérité est donne donc $or\_out <= START\ or\ start\_reg$. L'équation de chaque sortie est implantée dans une LUT du FPGA.

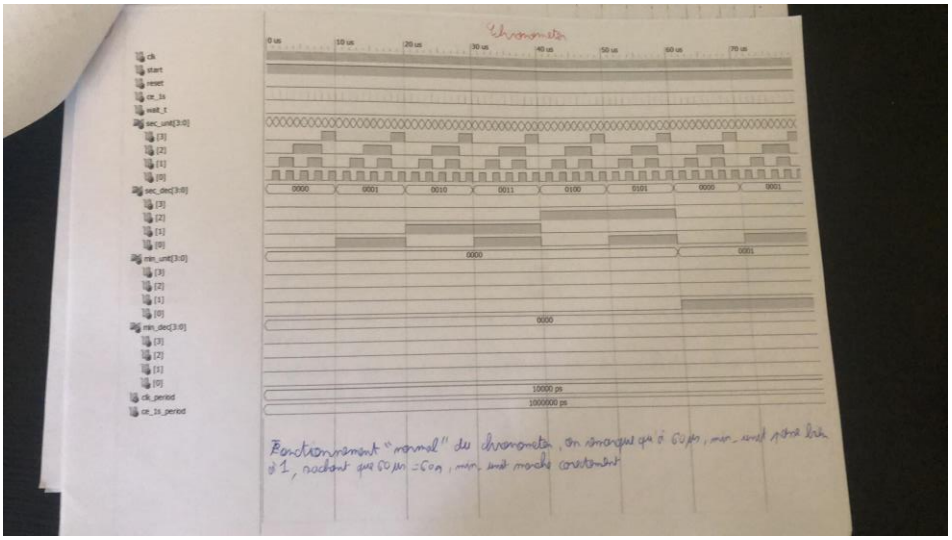
Réalisation de l'entité register_1b_R

Il s'agit d'un registre 1 bit à reset synchrone (horloge CLK). Chaque signal de sortie du compteur correspond à la sortie d'une bascule du FPGA.

D	**R**	**clk**	**Qn**
X	1	↑	**0**
0	0	↑	**0**
1	0	↑	**1**
X	X	-	**Qn-1**

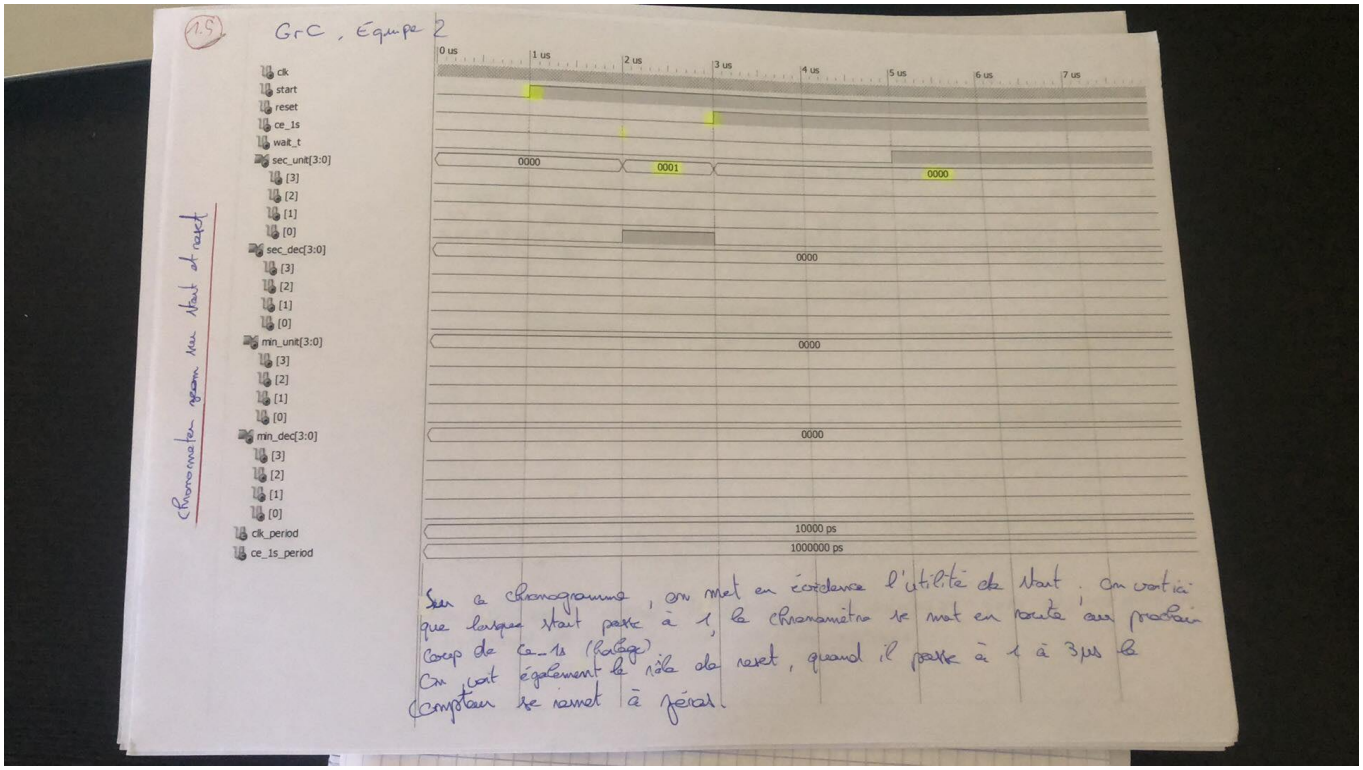
Comptage des secondes et des minutes

La gestion du comptage des secondes est effectuée par counterDec_4b_RE et counterSen_4b_RE. Le chronomètre compte les secondes et les minutes dont il retourne les valeurs respectives des unités et des dizaines. En effet comme nous pouvons le voir sur l'annexe **III.2**, nous constatons que les unités des secondes s'incrémentent bien chaque coup d'horloge de CE_1s ainsi que les dizaines de S et les unités des minutes (nous observerions les dizaines de minutes si nous poursuivions la simulation mais cela rendrait le graphique illisible)



Gestion du début de l'interruption et de l'arrêt

Le comptage est pendant contrôlé par 3 signaux : start, reset et wait_t. Le premier permet de lancer le comptage lorsqu'il passe à l'état haut, le 2nd réinitialise le comptage à 0 lorsqu'il est à l'état haut et le dernier met en pause le comptage lorsqu'il est à l'état haut. Le fonctionnement de ces 2 signaux est illustré sur l'annexe **III.2.**



Réalisation de l'entité counterDec_4b_RE

Il s'agit d'un compteur 4 bits avec une entrée d'activation CE synchrone (CLK) et un reset asynchrone. La table de vérité de cette fonction est :

counterDec_4b_RE							
R	CE	clk	$Q\_int_n[3:0]$	$Q\_int_{n+1}[3:0]$	Q[3 :0]	TC_int	TC
1	-	-	-	0000	Q_int[3 :0]	0	TC_int
0	1	↑	0000	0001	Q_int[3 :0]	0	TC_int
0	1	↑	0001	0010	Q_int[3 :0]	0	TC_int
0	1	↑	0010	0011	Q_int[3 :0]	0	TC_int
0	1	↑	0011	0100	Q_int[3 :0]	0	TC_int
0	1	↑	0100	0101	Q_int[3 :0]	0	TC_int
0	1	↑	0101	0110	Q_int[3 :0]	0	TC_int
0	1	↑	0110	0111	Q_int[3 :0]	0	TC_int
0	1	↑	0111	1000	Q_int[3 :0]	0	TC_int
0	1	↑	1000	1001	Q_int[3 :0]	0	TC_int
0	1	↑	1001	0000	Q_int[3 :0]	1	TC_int
0	1	-	-	$Q\_int_n[3:0]$	Q_int[3 :0]	TC_int	TC_int
0	0	↑	-	$Q\_int_n[3:0]$	Q_int[3 :0]	0	TC_int
0	0	-	-	$Q\_int_n[3:0]$	Q_int[3 :0]	0	TC_int

Ce compteur s'incrémente jusqu'à la valeur décimale 9 (1001 en binaire) puis se réinitialise à 0. Lorsqu'il se remet à 0, il émet un signal appelé TC qui permet l'incrémentation du chiffre des dizaines et qui se stoppe lors du coup d'horloge suivant.

Les équations des fonctions combinatoires appliquées sur les entrées D des bascules du compteurs sont définies à partir de la table de vérité et sont implantées dans les LUTs du FPGA.

Réalisation de l'entité counterSen_4b_RE

Il s'agit d'un compteur 4 bits avec une entrée d'activation CE synchrone (CLK) et un reset asynchrone. La table de vérité de cette fonction est :

counterSen_4b_RE							
R	CE	clk	$Q\_int_n[3:0]$	$Q\_int_{n+1}[3:0]$	Q[3 :0]	TC_int	TC
1	-	-	-	0000	Q_int[3 :0]	0	TC_int
0	1	↑	0000	0001	Q_int[3 :0]	0	TC_int
0	1	↑	0001	0010	Q_int[3 :0]	0	TC_int
0	1	↑	0010	0011	Q_int[3 :0]	0	TC_int
0	1	↑	0011	0100	Q_int[3 :0]	0	TC_int
0	1	↑	0100	0101	Q_int[3 :0]	0	TC_int
0	1	↑	0101	0000	Q_int[3 :0]	1	TC_int
0	1	-	-	$Q\_int_n[3:0]$	Q_int[3 :0]	TC_int	TC_int
0	0	↑	-	$Q\_int_n[3:0]$	Q_int[3 :0]	0	TC_int
0	0	-	-	$Q\_int_n[3:0$	Q_int[3 :0]	0	TC_int

Remarque : *Le symbole « - » signifie « quelle que soit la valeur du signal ».*

Ce compteur s'incrémente jusqu'à la valeur décimale 5 (0101 en binaire) puis se réinitialise à 0. Lorsqu'il se remet à 0, il émet un signal appelé TC qui permet l'incrémentation du chiffre des unités des minutes et qui se stoppe lors du coup d'horloge suivant (remarque : cette sortie n'est pas connectée dans le cas des dizaines pour les minutes).

Les équations des fonctions combinatoires appliquées sur les entrées D des bascules du compteurs sont définies à partir de la table de vérité et sont implantées dans les LUTs du FPGA.

Gestion du début, de l'interruption et de l'arrêt de comptages

La gestion de la détection des 45 minutes est effectuée par equ45min. Lorsque le chronomètre atteint 45 minutes, il doit se mettre en pause puisque c'est la mi-temps. Cette fonction permet donc

de détecter que le compteur a atteint les 45 minutes et donc envoie un signal pour qu'il se mette en pause.

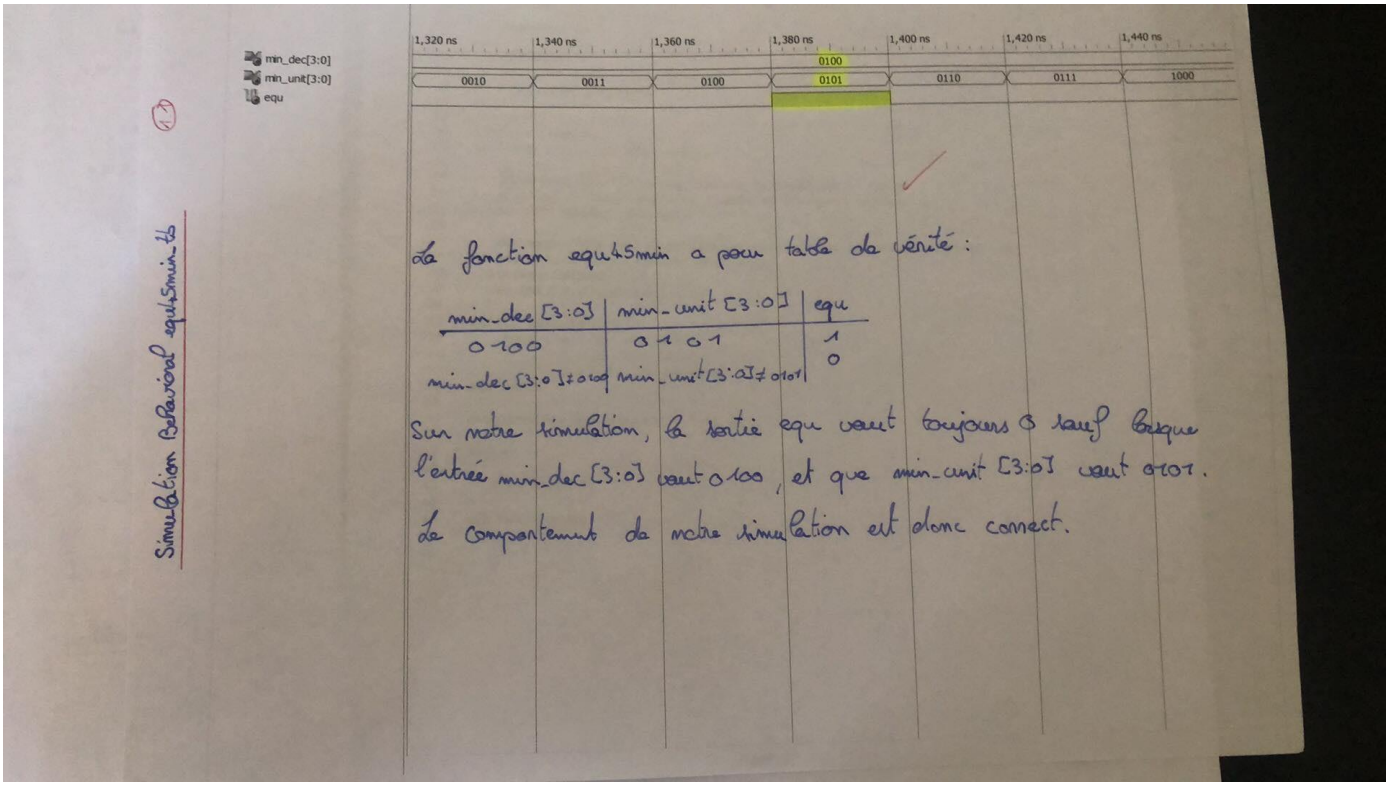
Finalement, la mise en relation des signaux qui bloquent le comptage est réalisée grâce à l'entité and4b2. En effet cette porte permet de mettre en commun tous les signaux retourner par les différentes commandes, à savoir le signal de lancement (start_reg), le signal de pause (WAIT_t), le signal d'indication des 45 minutes (equ45) et enfin l'horloge d'une seconde (CE_1s). Ainsi cette fonction permet que le compteur incrémente des secondes uniquement lorsque le final au départ a été lancé , lorsque l'horloge une 2nde passe à l'état haut et que ni le signal de pose ni le signal de 45 minutes soient activés.

Réalisation de l'entité equ45min

Il s'agit d'une fonction combinatoire qui indique que le chronomètre a atteint 45 minutes. La table de vérité de cette fonction est :

min_dec_int_open	**min_unit_int**	**equ45**
0100 (4)	0101 (5)	**1**
Tous les autres cas		**0**

Les équations des sorties de la fonction sont définies à partir de sa table de vérité est donne donc $equ <= '1' when (min\_dec = "0100" AND\ min\_unit = "0101")\ ELSE\ '0'$. La réalisation de la fonction est effectuée par 2 LUT du FPGA.



Réalisation de l'entité and4b2

Il s'agit d'une porte and à 4 entrées dont 2 inverseuse. Les équations des sorties de la fonction sont définies à partir de sa table de vérité est donne donc $CE\_time45min <= start\_reg\ and\ CE\_1s\ and\ not\ equ45\ and\ not\ WAIT\_t$. L'équation de chaque sortie est implantée dans une LUT du FPGA.

Tableau récapitulatif des résultats chiffrés du sous blocs et de ses fonctions

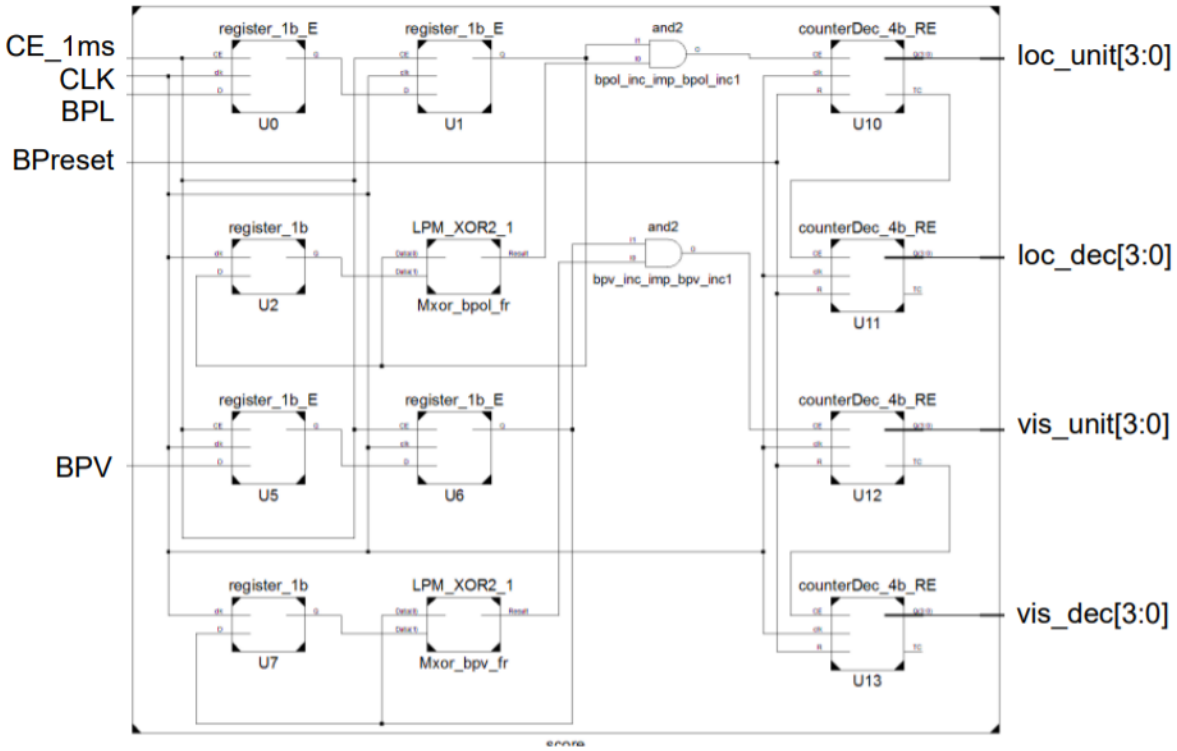
Groupe Deschamps Corto François Axel Garcia Elliot Le Corre Sarah 0	nb. SLICES	Taux occupation (/15850)	nb. SLICES LUTS	nb. SLICES REGISTER/IOB FLIP-FLOP	nb. IOB (entrées/sorties)	Taux occupation (/210)
CHRONOSCORE_PHASE_2						
chronometer	9	0,1%	24	20	21	10,0%
equ45min	1	0,0%	2		9	4,3%
register_1b_R	0	0,0%	0	1	4	1,9%
counterSen_4b_RE	2	0,0%	4	4	8	3,8%
counterDec_4b_RE	2	0,0%	5	5	8	3,8%

Sous bloc score

Le sous-bloc score va fournir aux sous-blocs Display et vgaDisplay les scores des équipes (sur deux chiffres) à afficher. Ses fonctions sont donc :

- Gérer les rebonds lorsque l'on presse les boutons BPL et BPV
- Détecter le front des signaux issus des interrupteurs BPL et BPV
- Compter l'incrémentation du score

Le schéma RTL du bloc est le suivant :



*Synoptique de **score***

Les signaux d'entrée du bloc sont :

- CE_1ms : horloge de période 1 ms (durée à l'état HAUT 10 ns) qui permet d'incrémenter les compteurs ou d'activer certaines parties du système numérique en conservant un unique signal d'horloge de 100 MHz au niveau des bascules).
- CLK : horloge de période 10 ns
- BPL : signal issu du bouton poussoir BPL (bouton pour incrémenter le score de l'équipe locale).
- BPV : signal issu du bouton poussoir BPV (bouton pour incrémenter le score de l'équipe visiteuse)
- BPreset: signal issu du bouton poussoir BPreset (bouton pour remettre le score des deux équipes à zéro)

Ses sorties sont des vecteurs de dimension 4:

- loc_unit[3:0] (nombre sur 4 bits correspondant à l'unité du score de l'équipe locale)
- loc_dec[3:0] (nombre sur 4 bits correspondant à la dizaine du score de l'équipe locale)
- vis_unit[3:0] (nombre sur 4 bits correspondant à l'unité du score de l'équipe visiteuse)
- vis_dec[3:0] (nombre sur 4 bits correspondant à la dizaine du score de l'équipe visiteuse)

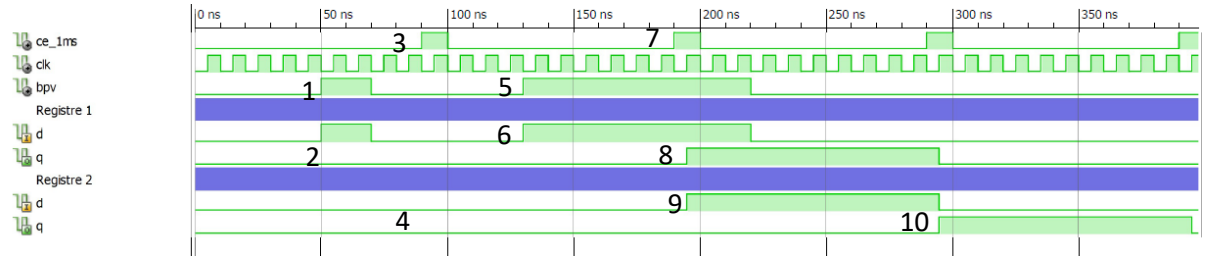
Le score des deux équipes est incrémenté avec des boutons poussoirs ; cela génère des rebonds lors du relâchement du bouton poussoir (car présence de ressort) qui doivent être compensés. Le bloc score contient donc un système d'antirebond géré par les entités register_1b_E en cascade. Le temps de masquage est compris entre 1 et 2ms.

La détection de front des signaux issus des interrupteurs BPL et BPV est géré par l'entité register_1b; elle permet de déterminer un front montant (lorsque nous appuyons sur le bouton poussoir). Dans le cas échéant, les compteurs vont s'incrémenter pour afficher le nouveau score.

Gérer les rebonds lorsque l'on presse les boutons BPL et BPV

Le système antirebond est réalisé grâce à 2 registres 1b en cascade pour chacun des boutons, soit un total de 4 registres 1 bits. En effets, ces registres ont une entrée d'activation CE, ce qui force le signal provenant de BPL et BPV à être à l'état haut lorsque le signal d'activation est à l'état haut.

Voici par exemple, un cas de fonctionnement de la gestion de rebonds



Un rebond accidentel est présent sur le bouton visiteur (1) et entre dans le bloc de détection (2). Cependant ce rebond est très cours (entre 50 et 70ns) et ainsi il n'est pas à l'état haut en même temps que l'entrée d'activation (3). Le rebond n'est donc pas détecté comme un point marqué (4). En revanche, lorsqu'on appuie volontairement (5), le signal entre dans le bloc (6) pendant suffisamment de temps pour avoir l'entrée d'activation à l'état haut (7). Le signal est donc bien détecté (8) et est transmis au second registre (9) qui garde l'information pendant un coup d'horloge supplémentaire, ce qui nous sera utile pour la détection de crête.

Réalisation de l'entité register_1b_E

Il s'agit d'un registre 1 bit avec 3 entrées : une d'activation CE synchrone (CLK), une horloge clk, et une entrée BPL.

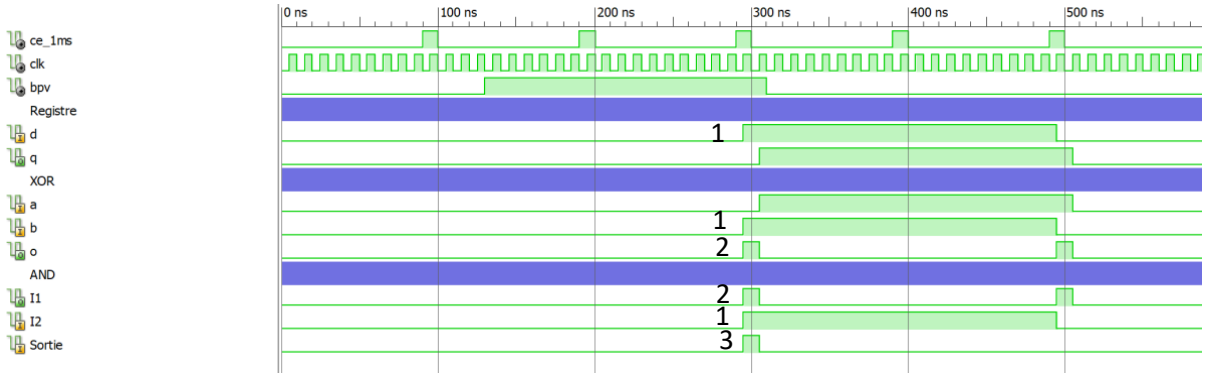
La table de vérité est la suivante :

D	CE	clk	Qn
0	1	↑	**0**
1	1	↑	**1**
X	-	-	**Qn-1**
-	0	-	**Qn-1**

Détecter le front des signaux issus des interrupteurs BPL et BPV

Le système de détection des fronts des signaux est réalisé, pour chacun des boutons, grâce à un registre 1bit, une porte xor et une porte and.

Voici par exemple, un cas de fonctionnement de la détection de crête :



En retournant à notre système précédent de gestion du rebond, nous avions obtenus le signal de bpv se trouvant alors dans le 2nd register_1b_E (étapes 10 du chronogramme précédent). La combinaison du registre cadencé sur clk et du XOR permettent de créer un détecteur de front (2). Ensuite, pour n'avoir que le front montant, il suffit de comparer le front ainsi détecté avec la valeur du signal d'origine grâce à une porte and : nous avons ainsi uniquement le front montant.

Réalisation de l'entité register_1b

Il s'agit d'un registre 1 bit avec deux entrées : une horloge clk et une entrée D correspondant à la sortie du register_1b_E.

Sa table de vérité est la suivante :

D	clk	Qn
0	↑	**0**
1	↑	**1**
X	-	**Qn-1**

Réalisation de l'entité XOR2

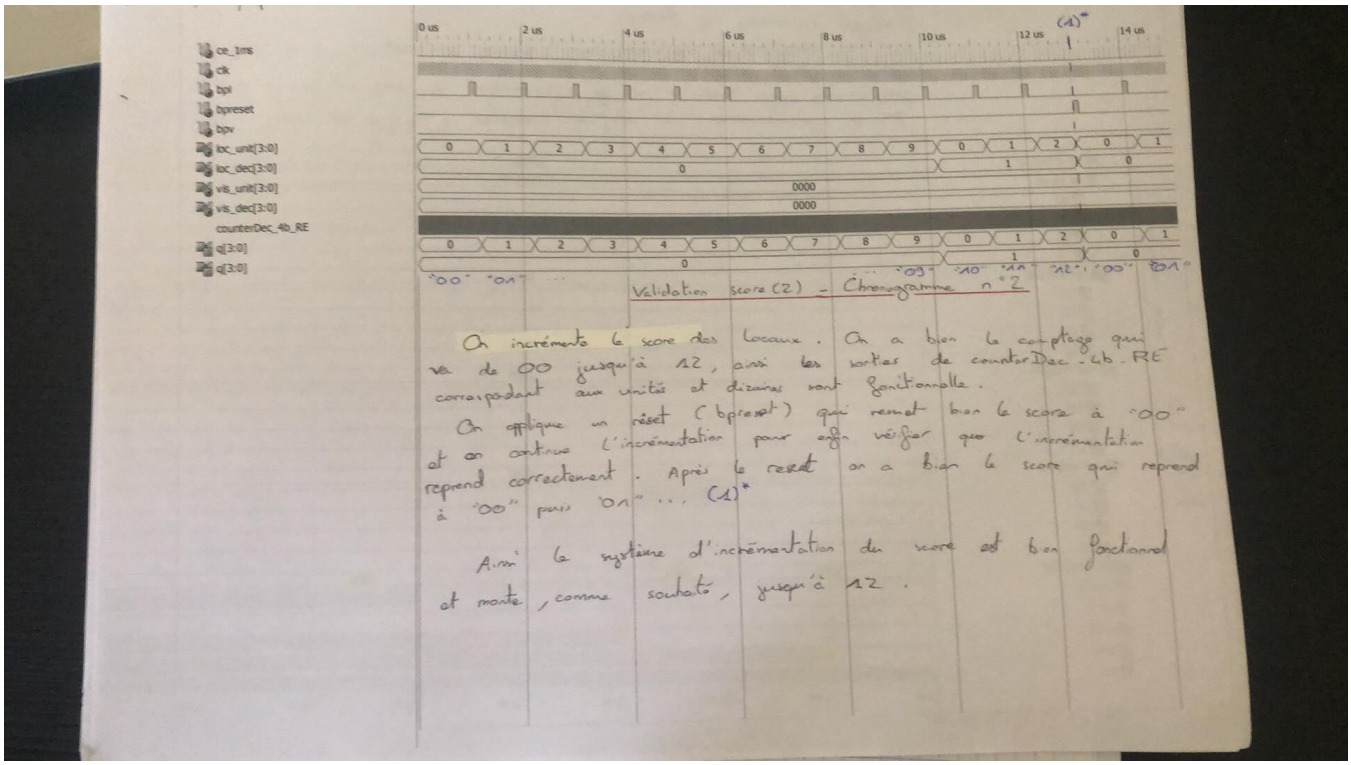
Il s'agit de la fonction combinatoire ou exclusif à deux entrées register_1b_E et register_1b.

Réalisation de l'entité and2

C'est tout simplement une porte AND prenant en entré la sortie XOR2 et de register_1b_E. La sortie sert de signal d'activation pour les compteurs.

Compter l'incrémentation du score

Le système de comptage est réalisé grâce à 2 compteurs 4 bits pour chacune des équipes. L'un des compteurs sert à compter les unités et l'autre les dizaines. Le fonctionnement est très similaire à celui de la partie chronomètre. En effet comme nous pouvons le voir sur l'annexe **III.1** (chronogramme 1 & 2), on constate que les unités des scores s'incrémentent bien chaque pression d'un bouton de but ainsi que les dizaines lorsque l'équipe atteint 9 ou 19 buts. Toujours sur le même chronogramme, nous constatons que le bouton bpreset, qui sert à mettre à zéro les scores, rempli bien sa fonction



Réalisation de l'entité counterDec_4b_RE

Il s'agit d'un compteur 4 bits avec une entrée d'activation CE synchrone (CLK) et un reset asynchrone. Ce compteur est présent 4 fois : 2 fois pour les unités, respectivement des visiteurs et des locaux, et 2 fois pour les dizaines. La table de vérité de cette fonction est :

counterDec_4b_RE							
R	CE	clk	$Q\_int_n[3:0]$	$Q\_int_{n+1}[3:0]$	Q[3 :0]	TC_int	TC
1	-	-	-	0000	Q_int[3 :0]	0	TC_int
0	1	↑	0000	0001	Q_int[3 :0]	0	TC_int
0	1	↑	0001	0010	Q_int[3 :0]	0	TC_int
0	1	↑	0010	0011	Q_int[3 :0]	0	TC_int
0	1	↑	0011	0100	Q_int[3 :0]	0	TC_int
0	1	↑	0100	0101	Q_int[3 :0]	0	TC_int
0	1	↑	0101	0110	Q_int[3 :0]	0	TC_int
0	1	↑	0110	0111	Q_int[3 :0]	0	TC_int
0	1	↑	0111	1000	Q_int[3 :0]	0	TC_int
0	1	↑	1000	1001	Q_int[3 :0]	0	TC_int
0	1	↑	1001	0000	Q_int[3 :0]	1	TC_int
0	1	-	-	$Q\_int_n[3:0]$	Q_int[3 :0]	TC_int	TC_int
0	0	↑	-	$Q\_int_n[3:0]$	Q_int[3 :0]	0	TC_int
0	0	-	-	$Q\_int_n[3:0]$	Q_int[3 :0]	0	TC_int

Son fonctionnement est donc le même que pour la partie chronometer, la seule différence est que son vecteur de sortie s'appelle ici loc_unit[3:0] au lieu de sec_unit[3 :0].

Tableau récapitulatif des résultats chiffrés du sous blocs et de ses fonctions

Groupe C CORTO FRANCOIS GARCIA LECORRE /	nb. SLICES	Taux occupation (/15850)	nb. SLICES LUTS	nb. SLICES REGISTER/IOB FLIP-FLOP	nb. IOB (entrées/sorties)	Taux occupation (/210)	temps propagation le plus long (séq : clk-->Q) (ns)	temps propagation BUFFER_IN (ns)	temps propagation BUFFER_OUT (ns)	temps propagation combinatoire + fils (ns)	temps initialisation bascules (ns)
CHRONOSCORE_PHASE_2											
score	11	0,07%	16	22	21	12,1%					
register_1b	0	0,00%	0	1	3	1,4%	9,790	1,000	2,630	6,160	100
register_1b_E	0	0,00%	0	1	4	1,9%					
counterDec_4b_RE	2	0,01%	5	5	8	3,8%					

Finalement, nous avons utilisés à peine 19% des slices disponibles sur le FGPA, nous avons donc assez peu utilisé la mémoire disponible sur le FPGA. Nous pourrions donc ajouter des éléments supplémentaires afin d'améliorer le fonctionnement de ce tableau de score.

De plus, cette faible utilisation de la mémoire FPGA montre bien que nous n'avions pas besoin de répartir nos différentes fonctions dans plus d'un seul bloc.

Synthèse et conclusion

Plusieurs pistes de performances peuvent être explorée pour améliorer le système. Par exemple, lorsque le chronomètre atteint 45 minutes (mi-temps) il faut le réinitialiser pour la seconde mi-temps, ce qui n'est pas très pratique. De plus, ce système ne s'adapte pas lors des temps supplémentaires comme il est fréquent au football ou dans d'autres sports. Enfin le produit est peu adaptable, il ne convient qu'au football et il serait intéressant de pourvoir le rendre plus universel pour d'autres sport comme le rugby ou le basket qui ont des durées de jeu différentes et plus d'arrêts de jeu (les quart-temps pour le basket).

Enfin, nous avons appris à développer des fonctions séquentielles en VHDL. Nous avons aussi appris à les utiliser avec des fonctions combinatoires pour réaliser des fonctions plus complexes.

Annexes :

I. Phase 1

1.1) transcoder_3v8

1.2) mux_8x1x4b

2.1) transcoder_7segs

2.2) mux_8x1x1b

3) register_8b

4) Tregister_1b

II. Phase 2

1.1) equ45min

1.3) register_1b_R

1.4) counterSen_4b_RE

1.5) **chronometer**

2.1) XOR2

2.2) register_1b

2.3) register_1b_E

2.4) counterDec_4b_RE

2.5) **score**

III. Validation

1) chronogrammes de validation **score**

2) chronogrammes de validation **chronometer**

3) **display**