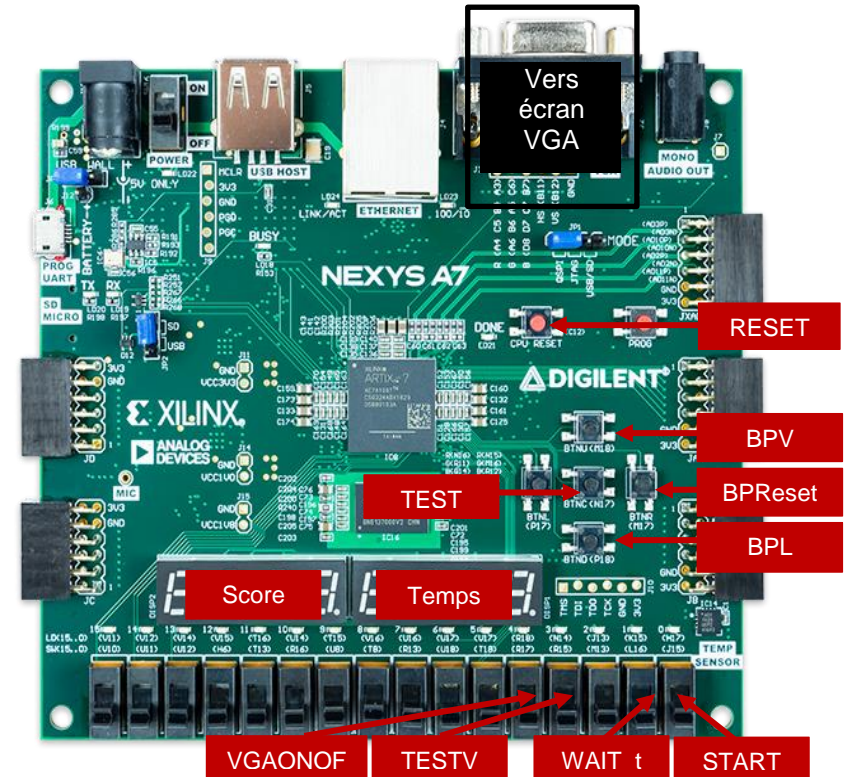


## ELN2 : PROJET SCORING 2.0

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

**Phase 2 : Gestion du chronomètre et du score**  
(Guide de séance)



#### 4 Réalisation du projet **chronoscore\_phase2**

Le projet **chronoscore\_phase2** met en œuvre une version de **chronoscore** étendue aux sous-blocs **chronometer** et **score**.

Le développement des différents sous-blocs s'effectue selon l'organisation ci-dessous :

Binôme (trinôme) impair :    fonction **equ45min**  
     fonction **register\_1b**  
     fonction **register\_1b\_R**  
     fonction **counterSen\_4b\_RE**  
     intégration sous-bloc **chronometer**

Binôme pair :                    fonction **XOR\_2b**  
     fonction **register\_1b**  
     fonction **register\_1b\_E**  
     fonction **counterDec\_4b\_RE**  
     intégration sous-bloc **score**

Equipe :                            intégration de **chronoscore\_phase2**

##### 4.1 Création du projet

Démarrer **ISE version 14.6** (64\_bit) via le menu **Démarrer** de Windows et créer le projet **chronoscore\_phase2** ayant les caractéristiques suivantes :

<b>Name</b>	<b>chronoscore_phase2</b>
<b>Location</b>	(...)\ELN2\Groupe_X\Equipe_N
<b>Working Directory</b>	(...)\ELN2\Groupe_X\Equipe_N
<b>Top_Level source type</b>	<b>HDL</b>

(...) : CPE\_users\TPelec\_3ETI (Répertoire de travail sous **Windows**).

Les répertoires **Groupe\_X** et **Equipe\_N** devront être créés au préalable :

**Groupe\_X** : X vaut de A à D en fonction du groupe de TP

**Equipe\_N** : N numéro d'équipe

**Rappel !** Le système crée automatiquement un répertoire associé au nom du projet. Il travaille uniquement dans ce répertoire. **Il est important que tous les fichiers sources y soient placés.**

Les caractéristiques du **FPGA** cible sont :

<b>Evaluation Development Board</b>	None Specified
<b>Product Category</b>	All
<b>Family</b>	<b>Artix7</b>
<b>Device</b>	<b>XC7A100T</b>
<b>Package</b>	<b>CSG324</b>
<b>Speed</b>	<b>-1</b>
<b>Top-Level Source Type</b>	<b>HDL</b>

(suite page suivante)

<b>Synthesis Tool</b>	<b>XST (VHDL/Verilog)</b>
<b>Simulator</b>	<b>ISim (VHDL/Verilog)</b>
<b>Preferred Language</b>	<b>VHDL</b>
<b>Property Specification in Project File</b>	Store non-default values only
<b>Manual Compile Orders</b>	<input type="checkbox"/>
<b>VHDL Source Analysis Standard</b>	<b>VHDL-200X</b>
<b>Enable Message Filtering</b>	<input type="checkbox"/>

#### 4.2 Réalisation des fonctions des sous-blocs **chronometer** et **score**

Pour chacune des **fonctions** :

1. Créer l'entité\* **VHDL** correspondante (Projet → New Source).  
Définir ses signaux d'entrées\* et sorties\*.  
(\*) : Respecter les noms imposés dans les spécifications.
2. Ecrire le code **VHDL** de son architecture.  
Préciser dans la zone d'entête de chaque fichier source les noms des auteurs (par ordre alphabétique) :  
**-- Engineer: NOM1\_NOM2** (ajouter **NOM3** si trinôme).
3. Faire la synthèse.

Pour l'entité **XOR\_2b** (binôme **pair**) :

Effectuer la simulation **Behavioral**.

Exploiter les résultats de simulation.

Pour l'entité **equ45min** (binôme **impair**) :

A partir de la vue **Technologique**, expliquer comment sont réalisées les équations obtenues en préparation.

Effectuer la simulation **Behavioral**.

Exploiter les résultats de simulation.

Pour l'entité **register\_1b** (binômes **impair** et **pair**) :

Effectuer la simulation **Behavioral**.

Exploiter les résultats de simulation.

Effectuer la simulation **Post-Route**. Exploiter les résultats de simulation.

Pour l'entité **register\_1b\_R** (binôme **impair**) :

Effectuer la simulation **Behavioral**.

Exploiter les résultats des simulations. Expliquer notamment le rôle de l'entrée **R**.

Pour l'entité **register\_1b\_E** (binôme **pair**) :

Effectuer la simulation **Behavioral**.

Exploiter les résultats des simulations. Expliquer notamment le rôle de l'entrée **CE**.

Pour l'entité **counterSen\_4b\_RE** (binôme **impair**) (\*) :

Effectuer la simulation **Behavioral**.

Exploiter les résultats des simulations.

Valider que l'allure du signal **TC** est conforme aux attentes du document de préparation. Vérifier le rôle des entrées **R** et **CE**.

Pour l'entité **counterDec\_4b\_RE** (binôme **pair**) (\*) :

Effectuer la simulation **Behavioral**.

Exploiter les résultats des simulations. Valider que l'allure du signal **TC** est conforme aux attentes du document de préparation. Vérifier le rôle des entrées **R** et **CE**.

(\*) Pour les simulations de **counterSen\_4b\_RE** et **counterDec\_4b\_RE** les signaux **clk** et **CE** seront définis de la manière suivante :

- **clk** : période 10 ns, durée à l'état HAUT 5 ns,
- **CE** : période 100 ns, durée à l'état HAUT 10 ns,

La description **VHDL** de ces signaux est fournie en annexe 3.

Pour chaque entité **jugée valide** (résultats corrects après synthèse et simulations) :

- **imprimer et valoriser** (c'est-à-dire mettre en valeur les éléments intéressants) les documents suivants :

- le code source (fichier .vhd),
- la vue **Technologique**,
- les chronogrammes des simulations **Behavioral** et/ou **Post-Route**.

Pour chaque entité **jugée valide** (résultats corrects après synthèse et simulations) :

- **compléter la ligne correspondante du tableau de caractérisation** du sous-bloc **chronometer** ou **score** (fourni<sup>(1)</sup> sur CPe-Campus) en précisant les informations<sup>(2)</sup> suivantes :

- le nombre de SLICES,
- le nombre de SLICES LUTS,
- le nombre de SLICES REGISTERS ou IOB FLIP-FLOP,
- le nombre d'IOBS (entrées / sorties).

(1) Préciser l'équipe et les noms des étudiants.

(2) Les informations à rassembler dans le tableau sont disponibles dans le document **Place and Route Report** une fois la phase d'implémentation réussie. Si la page **HTML** ne se met pas à jour, on peut les trouver dans le fichier d'extension « **.par** » créé par l'outil ISE et disponible dans le répertoire du projet.

Pour chaque entité **jugée valide** (résultats corrects après synthèse et simulations) :

- **pour la fonction register\_1b**, rassembler dans le tableau les temps de propagation<sup>(3)</sup> suivants :

- le temps du chemin le plus long (à rechercher sur la durée totale de la simulation),
- le temps dans le buffer d'entrée,
- le temps dans le buffer de sortie.

*N.B.*: Le temps dans la fonction (logique + fils) sera calculé automatiquement.

(3) Les temps de propagation à rassembler dans le tableau doivent apparaître clairement sur la simulation **Post-Route**.

#### 4.3 Réalisation du sous-bloc **chronometer** (binôme **impair**)

1. Créer l'entité\* VHDL correspondante (Projet → New Source).  
Définir ses signaux d'entrées\* et sorties\*.  
(\*) : Respecter les noms imposés dans les spécifications.
2. Ecrire le code VHDL de son architecture à partir de ses fonctions de base.  
Préciser dans la zone d'entête de chaque fichier source les noms des auteurs (par ordre alphabétique) :  
**-- Engineer:     NOM1\_NOM2** (ajouter **NOM3** si trinôme).

*L'entité **chronometer** comporte les signaux internes suivants (voir le synoptique fourni en annexe 1) :*

**start\_reg, or\_out, equ45, CE\_time45min,**  
**CE\_secDec,   CE\_minUni,   CE\_minDec,   min\_unit\_int,**  
**min\_dec\_int.**

*La sortie **TC non utilisée** de l'entité **counterSen\_4b\_RE** des minutes est positionnée à '**open**'.*

Fonction **OR**

La porte logique **OR**, identifiée par « **or2** » sur le synoptique de la figure 5, sera modélisée directement au sein de l'architecture de **chronometer** en intégrant une instruction concurrente.

Les entrées de l'entité VHDL seront les signaux : **START, start\_reg.**

La sortie de l'entité VHDL sera le signal : **or\_out.**

Fonction **AND4n2**

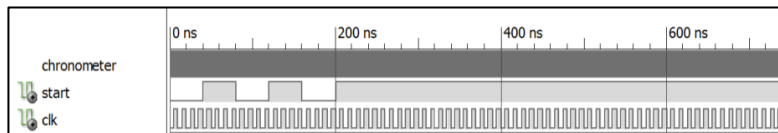
La porte logique **AND4n2**, identifiée par « **and4b2** » sur le synoptique de la figure 5, sera modélisée directement au sein de l'architecture de **chronometer** en intégrant l'instruction concurrente définie en préparation.

Les entrées de l'entité VHDL seront les signaux : **start\_reg, CE\_1s, WAIT\_t, equ45.**

La sortie de l'entité VHDL sera le signal : **CE\_time45min.**

3. Faire la synthèse.  
Vérifier que le synoptique obtenu est conforme à celui attendu.
4. Effectuer plusieurs simulations **Behavioral** permettant de mettre en évidence le fonctionnement du sous-bloc :
  - la gestion des sorties des compteurs de temps : visualisation des signaux de sortie des différentes entités **counterDec\_4b\_RE** et **counterSen\_4b\_RE** lors du passage de 9 secondes à 10 secondes, de 59 secondes à 1 minute, de 9 minutes à 10 minutes et, en fin de comptage, à 45 minutes (pour ces simulations, le signal **START** sera positionné de manière à être **actif** et les signaux **RESET** et **WAIT\_t** seront positionnés de manière à être **inactifs**),
  - la gestion du début, de l'interruption et de la remise à zéro du comptage : évolution au cours du temps du signal de sortie de la porte logique **and4n2.**

- la gestion des rebonds du signal issu de l'interrupteur START : évolution au cours du temps du signal de sortie de l'entité **register\_1b\_R** (pour cette simulation, les signaux RESET et WAIT\_t seront positionnés de manière à être **inactifs**). Le signal START sera défini de la manière suivante :



Pour l'ensemble de ces simulations, les signaux **CLK**, **CE\_1s** seront définis de la manière suivante :

- CLK** : période 10 ns, durée à l'état HAUT 5 ns,
- CE\_1s** : période 1000 ns, durée à l'état HAUT 10 ns.

La description **VHDL** de ces signaux est fournie en annexe 3.

Une fois l'entité jugée valide (résultats corrects après synthèse et simulations) :

- **imprimer et valoriser** les documents suivants :

- le fichier source,
- la vue **RTL** développée,
- les chronogrammes des simulations **Behavioral**.
- le nombre d'IOBS (entrées / sorties).

Une fois l'entité jugée valide (résultats corrects après synthèse et simulations) :

- **compléter la ligne correspondante du tableau de caractérisation** du sous-bloc **chronometer** (fourni<sup>(1)</sup> sur CPe-Campus) en précisant les informations<sup>(2)</sup> suivante :

- le nombre de SLICES,
- le nombre de SLICES LUTS,
- le nombre de SLICES REGISTERS ou IOB FLIP-FLOP
- le nombre d'IOBS (entrées / sorties).

(1) Préciser l'équipe et les noms des étudiants.

(2) Les informations à rassembler dans le tableau sont disponibles dans le document **Place and Route Report** une fois la phase d'implémentation réussie. Si la page **HTML** ne se met pas à jour, on peut les trouver dans le fichier d'extension « **.par** » créé par l'outil ISE et disponible dans le répertoire du projet.

#### 4.4 Réalisation du sous-bloc **score** (binôme **pair**)

1. Créer l'entité\* VHDL correspondante (Projet → New Source).  
Définir ses signaux d'entrées\* et sorties\*.  
(\*) : Respecter les noms imposés dans les spécifications.
2. Ecrire le code VHDL de son architecture à partir de ses fonctions de base.  
Préciser dans la zone d'entête de chaque fichier source les noms des auteurs (par ordre alphabétique) :  
**-- Engineer:     NOM1\_NOM2** (ajouter **NOM3** si trinôme).

*L'entité **score** comporte les signaux internes suivants (voir le synoptique fourni en annexe 2) :*

***bpl\_r, bpl\_d, bpl\_f, bpl\_fr, bpl\_inc, CE\_LOC,  
bpv\_r, bpv\_d, bpv\_f, bpv\_fr, bpv\_inc, CE\_VIS.***

*Les sorties **TC non utilisées** des entités **counterDec\_4b\_RE** sont positionnées à 'open'.*

Fonction **AND**

La porte logique **AND**, identifiée par « **and2** » sur le synoptique de la figure 6, sera modélisée directement au sein de l'architecture de **score** en intégrant une instruction concurrente.

1<sup>ère</sup> porte **AND** (équipe locale) :

Les entrées de l'entité VHDL seront les signaux : bpl\_fr, bpl\_d.

La sortie de l'entité VHDL sera le signal : bpl\_inc.

2<sup>ème</sup> porte **AND** (équipe des visiteurs) :

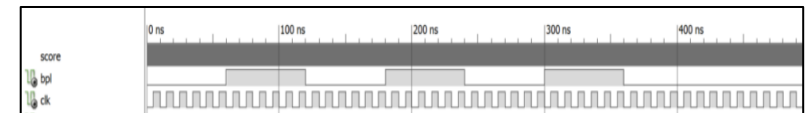
Les entrées de l'entité VHDL seront les signaux : bpv\_fr, bpv\_d.

La sortie de l'entité VHDL sera le signal : bpv\_inc.

3. Faire la synthèse.  
Comparer la vue RTL obtenue au synoptique fourni en annexe 2.
4. Effectuer plusieurs simulations **Behavioral** permettant de mettre en évidence le fonctionnement du sous-bloc :

- la gestion des rebonds des signaux issus des boutons poussoirs BPL ou BPV : évolution au cours du temps des sorties des deux registres **register\_1b\_E** en cascade (pour cette simulation, le signal BPRreset sera positionné de manière à être **inactif**).

Le signal BPL ou BPV sera défini de la manière suivante :



- la détection de front des signaux issus des boutons poussoirs BPL ou BPV : évolution au cours du temps des signaux d'entrée et de sortie des portes logiques **and2** (pour cette simulation, le signal BPRreset sera positionné de manière à être **inactif**). Le signal BPL ou BPV sera défini de même que précédemment.



- la gestion des sorties des compteurs de score : visualisation des signaux de sortie des différentes entités **counterDec\_4b\_RE** entre 0 et 15 impulsions du signal BPL ou BPV (pour cette simulation, le signal sera défini **sans rebond**, le signal BPRreset sera positionné de manière à être inactif entre 0 et 10 impulsions puis actif à la 12<sup>ème</sup>).

Pour l'ensemble de ces simulations, les signaux **CLK**, **CE\_1ms** seront définis de la manière suivante :

- CLK** : période 10 ns, durée à l'état HAUT 5 ns,
- CE\_1ms** : période 100 ns, durée à l'état HAUT 10 ns.

La description **VHDL** de ces signaux est fournie en annexe 3.

Une fois l'entité jugée valide (résultats corrects après synthèse et simulations) :

- **imprimer et valoriser** les documents suivants :

le fichier source,  
la vue **RTL** développée,  
les chronogrammes des simulations **Behavioral**.

Une fois l'entité jugée valide (résultats corrects après synthèse et simulations) :

- **compléter la ligne correspondante du tableau de caractérisation** du sous-bloc **score** (fourni<sup>(1)</sup> sur CPe-Campus) en précisant les informations<sup>(2)</sup> suivantes :

le nombre de SLICES,  
le nombre de SLICES LUTS,  
le nombre de SLICES REGISTERS ou IOB FLIP-FLOP  
le nombre d'IOBS (entrées / sorties).

(1) Préciser l'équipe et les noms des étudiants.

(2) Les informations à rassembler dans le tableau sont disponibles dans le document **Place and Route Report** une fois la phase d'implémentation réussie. Si la page **HTML** ne se met pas à jour, on peut les trouver dans le fichier d'extension « **.par** » créé par l'outil ISE et disponible dans le répertoire du projet.

#### 4.5 Insertion des sous-blocs exécutables de **timeGenerator**, **display**, **vgaDisplay** **chronometer** et **score**

A partir du cours ELN2 sur **CPe-Campus**, télécharger les fichiers **timeGenerator.ngc**, **display.ngc**, **vgaDisplay.ngc**, **chronometer.ngc** et **score.ngc** relatifs aux sous-blocs **timeGenerator**, **display**, **vgaDisplay**, **chronometer** et **score**.

Placer ces fichiers dans un sous-répertoire du projet **chronoscore\_phase2**, ce sous-répertoire sera nommé « **ngc** ».

#### 4.6 Finalisation du système **chronoscore**

A partir du cours ELN2 sur **CPe-Campus**, télécharger les fichiers relatifs à l'entité **chronoscore** :

- **chronoscore.vhd**,
- **chronoscore.ucf**.

Placer ces fichiers dans le répertoire du projet **chronoscore\_phase2**.

Ajouter les fichiers au projet (Projet → Add Source).

Pour **chronoscore.ucf** penser à mettre **chronoscore.vhd** en « **top-module** » avant.

Validation du sous-bloc **chronometer** (binôme **impair**) :

Ajouter les fichiers **timeGenerator.ngc**, **vgaDisplay.ngc**, **display.ngc** et **score.ngc** au projet **chronoscore\_phase2** (Projet → Add Source).

Exécuter l'ensemble du flot de conception pour générer le fichier binaire qui servira à configurer les LUTs, bascules et interconnexions du **FPGA**.

Implanter l'entité **chronoscore** dans le **FPGA** à l'aide de l'outil **iMPACT**.

Tester la réalisation à l'aide de la carte de développement et faire valider par un assistant.

Validation du sous-bloc **score** (binôme **pair**) :

Ajouter les fichiers **timeGenerator.ngc**, **vgaDisplay.ngc**, **display.ngc** et **chronometer.ngc**, au projet **chronoscore\_phase2** (Projet → Add Source).

Exécuter l'ensemble du flot de conception pour générer le fichier binaire qui servira à configurer les LUTs, bascules et interconnexions du **FPGA**.

Implanter le système **chronoscore\_phase2** dans le **FPGA** à l'aide de l'outil **iMPACT**.

Tester la réalisation à l'aide de la carte de développement et faire valider par un assistant.

Une fois le fonctionnement de chacun des sous-blocs **chronometer** et **score** validé, remplacer le fichier **chronometer.ngc** ou **score.ngc** par l'ensemble de ses entités **VHDL**.

Tester la réalisation à l'aide de la carte de développement et faire valider par un assistant.

Une fois le fonctionnement global du système validé, remplacer le fichier **display.ngc** par les entités VHDL développés lors de la phase 1.

Tester la réalisation à l'aide de la carte de développement et faire valider par un assistant.

## 5 En fin de chaque séance de la phase 2

Dépôt de fichiers :

- Faire un répertoire « **.zip** » des fichiers sources créés et du tableau Excel complété.

*Les répertoires au format tar ou 7z ne seront pas acceptés.*

Le nom du répertoire devra être formaté de la manière suivante ;

**ELN2-X-NOM1-NOM2-... :**

X, groupe de TP

NOM1, nom étudiant(e) 1

NOM2, nom étudiant(e) 2

etc ...



Mettre les noms des membres de l'équipe par **ordre alphabétique**

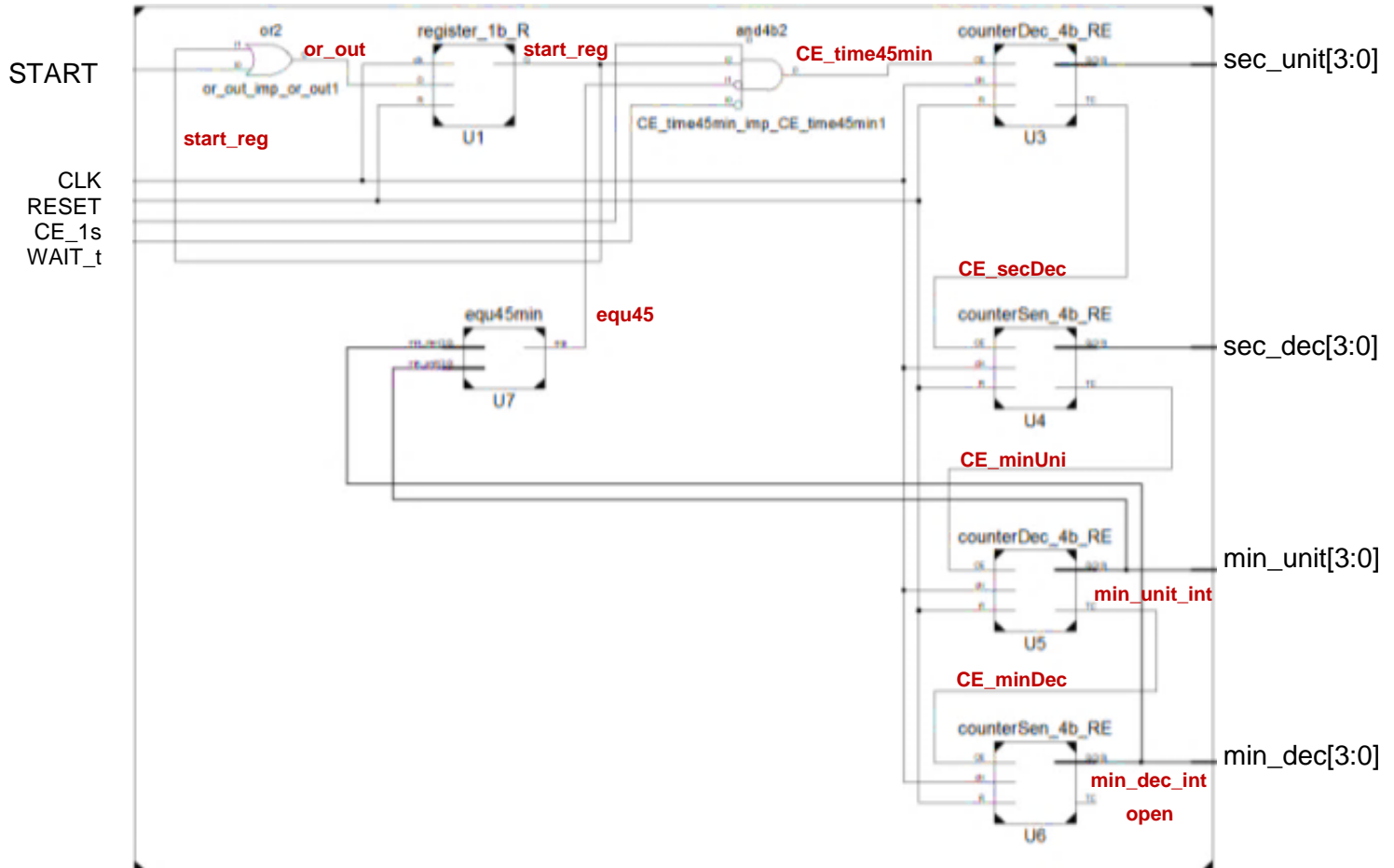
- Le dépôt du répertoire devra être effectué sur CPe-Campus par l'étudiant(e) n°1 dans l'ordre alphabétique.

Documents papier à rendre :

- Pour chaque entité créée, rendre :  
Impression du code source,  
Impression commentée de la vue **RTL** ou **Technologique** selon le cas,  
Impression valorisée de la simulation **Behavioral**,  
Impression valorisée de la simulation **Post-Route** selon le cas.
- Tableau de caractérisation des sous-blocs **chronometer** et **score**.

## Annexe 1

### Architecture du sous-bloc **chronometer**



L'entité **chronometer** comporte les signaux internes suivants :

**start\_reg**, **or\_out**, **equ45**, **CE\_time45min**,

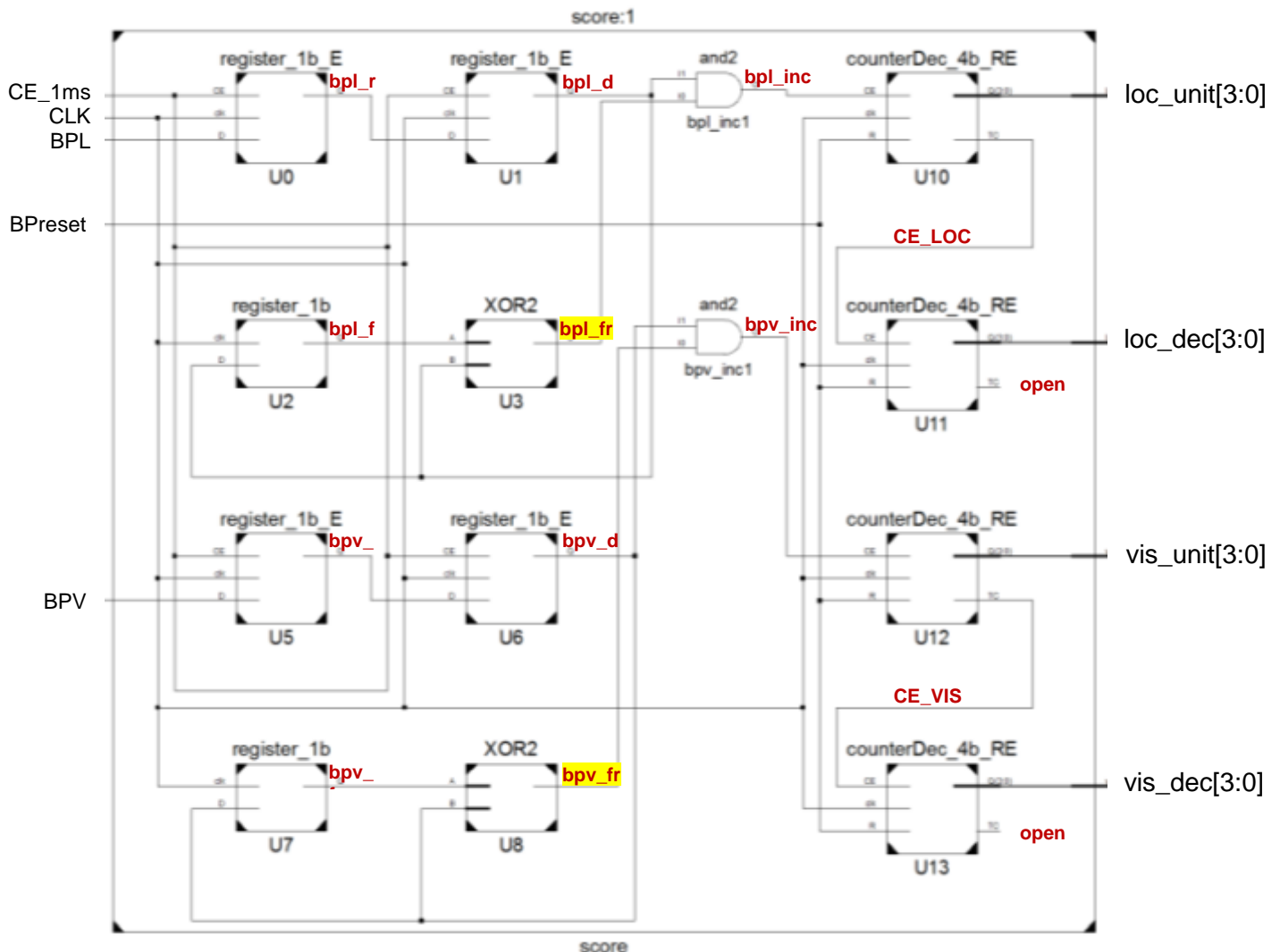
**CE\_secDec**, **CE\_minUni**, **CE\_minDec**,

**min\_unit\_int**, **min\_dec\_int**.

La sortie **TC** non utilisée de l'entité **counterSen\_4b\_RE** des minutes (U6) est positionnée à '**open**'.

## Annexe 2

## Architecture du sous-bloc **score**



L'entité **score** comporte les signaux internes suivants :

**bpl\_r, bpl\_d, bpl\_f, bpl\_fr, bpl\_inc, CE\_LOC,  
bpv\_r, bpv\_d, bpv\_f, bpv\_fr, bpv\_inc, CE\_VIS.**

Les sorties non utilisées des entités **counterDec\_4b\_RE** (U11 et U13) sont positionnées à 'open'.

### Annexe 3

Code VHDL des signaux **clk**, **CLK**, **CE**, **CE\_1ms** et **CE\_1s** à utiliser pour les simulations

**-- Clock period definitions (à placer dans la zone de déclarations du fichier \_tb.vhd)**

```
constant clk_period : time := 10 ns;
constant CLK_period : time := 10 ns;
constant CE_period : time := 100 ns;
constant CE_1ms_period : time := 100 ns;
constant CE_1s_period : time := 1000 ns;
```

**-- Clock process definitions (à placer entre BEGIN et END)**

```
clk_process :process                                -- ou CLK_process : process
begin
    clk <= '0';                                     -- ou CLK <= '0';
    wait for clk_period/2;
    clk <= '1';                                     -- ou CLK <= '1';
    wait for clk_period/2;
end process;
```

```
CE_process :process                                -- ou CE_1ms_process : process
begin
    CE <= '0';                                       -- ou CE_1ms <= '0';
    wait for 90 ns;
    CE <= '1';                                       -- ou CE_1ms <= '1';
    wait for 10 ns;
end process;
```

```
CE_1s_process :process
begin
    CE_1s <= '0';
    wait for 990 ns;
    CE_1s <= '1';
    wait for 10 ns;
end process;
```