

**Architecture des systèmes ordinés et VHDL**

ELE748

**Travail remis à**

Simon Pichette

**Laboratoire 2**

**Par**

Jonathan Lapointe (LAPJ05108303)   
Kévin Parent Legault(PARK22049009)

**Rédigé le**

**École de technologie supérieure**  
Département de Génie Électrique

Table des matières

[Introduction 3](#_Toc419389449)

# Introduction

Dans le cadre du cours d’architecture des systèmes ordinés et VHDL, nous avons réalisé un deuxième système comportant une architecture modulaire. Cette architecture a été générée en utilisant l’utilitaire d’intégration système d’Altera, QSYS. Nous avons ensuite conçu une architecture logicielle fonctionnant sur ce système. Nous avons finalement déployé le tout sur le Cyclone V de la carte DE1 SOC d’Altera. Le système conçu est une application d’infographie simplifiée (NIOS draw) sur écran VGA qui sera contrôlée à l’aide d’une souris PS2.

# Architecture du système

## Description du système

L’architecture système est composé de plusieurs modules tous générés à l’aide de QSYS. Ce dernier génère une composante VHDL qui contient les modules de notre architecture. Cette composante doit cependant ce faire connecter dans un fichier TOP au monde extérieur.

Notre architecture comprend les périphériques suivants :

|  |  |  |
| --- | --- | --- |
| Composante | Module Qsys | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Architecture logicielle

Dans ce laboratoire, l’architecture logicielle permet d’effectuer plusieurs choses. Tel que :

* Recevoir, mettre en mémoire et décoder les trames envoyées par la souris PS2;
* Rafraichir les pixels et effacer l’écran VGA ;
* Envoyer les coordonnées de position et les actions de la souris à la console NIOS;
* Afficher un curseur (caractère ASCII dans notre cas) qui se déplacera sur l’écran en fonction de la position de la souris;
* Afficher la position du curseur au bas de l’écran VGA.

## Description de logiciel

Voici un tableau qui décrit plus en détail les librairies conçues pour ce projet :

|  |  |  |
| --- | --- | --- |
| Librairie | Nom des Fonctions | Description |
| Main (.c) | main | Routine principale |
| Display(.c/.h) | InitDisplay | Initialise pixel buffer et char buffer |
| DrawPixel | Rafraichit la couleur d’un pixel selon sa position x et y |
| DrawPixelColored | Affiche un pixel avec couleur en alternance selon x,y |
| CleanDrawZone | Efface la zone de dessin |
| UpdateCursorPosition | Rafraichit les valeurs de la position du curseur à l’écran |
| NiosDrawApp | Routine principale de rafraichissement de l’écran |
| DisplayCoordinate | Affiche les coordonnées au bas de l’écran |
| SendTelemetry | Rafraichit les valeurs de télémétrie envoyées à la console Nios |
| Mouse(.c/.h) | ps2\_isr | Routine d’interruption : mise en mémoire de X,Y,SW des évènements souris dans des FIFO |
| mouseInit | Initialise la souris et les interruptions |
| mouseGetNbEvent | Retourne le nombre d’évènement souris reçu |
| mouseGetX | Extrait une valeur de X du fifo |
| mouseGetY | Extrait une valeur de Y du fifo |
| mouseGetSwL | Extrait une valeur de bouton gauche du fifo |
| mouseGetSwR | Extrait une valeur de bouton droit du fifo |
| mousePtrOutInc | Incrémente le pointeur de sortie du fifo et décrémente NbEvent doit être appelé après avoir fait tous les « get » |
| JtagUart(.c/.h) | jUartSendString | Envoie sur console Nios une chaî`LÉne de caractère ou un tableau se terminant par NULL |
| jUartSendVar | Envoie une valeur décimale sur la console Nios |
| Hardware (.h) |  | Définition des types et inclusion des librairies utilisées dans le projet |