PARTIE 1

Dézipper le fichier tp3_p1.zip

Q1 : Ouvrir le projet tp3_p1 sous Vivado

Q2 : Observer le schéma et décrire le rôle des différents modules. A quoi sert le bloc de mémoire RAM

On va maintenant ajouter une IP pour calculer le centre de gravité de l'image dans le flux vidéo. Cette IP comportera une entrée et une sortie Stream.

Q3 : Créer cette IP sous vivado avec les fichiers suivants :

fichier source CPP c_grav.cpp

```
#include"ap_int.h"
#include "hls_video.h"

volatile long cgx_r=320;
volatile long cgy_r=240;

typedef hls::stream<ap_axiu<8,1,1,1> >        AXI_STREAM;
void c_grav(ap_uint<1> cg_on,AXI_STREAM& s_axis_video,AXI_STREAM& m_axis_video, int hsize_in, int vsize_in)
{
#pragma HLS INTERFACE axis register both port=s_axis_video
#pragma HLS INTERFACE axis register both port=m_axis_video
unsigned nb=0;
long cgx=0;
long cgy=0;
ap_axiu<8, 1, 1, 1> video;
for(int i = 0; i < vsize_in ; i ++)
        {
#pragma HLS PIPELINE
        for(int j = 0; j < hsize_in ; j ++)
                {
                s_axis_video >> video;
                m_axis_video << video;
                }
        }
cgx_r=cgx/nb;
cgy_r=cgy/nb;
}
```

fichier source testbench CPP c_grav_tb.cpp

```
#include"ap_int.h"
#include "hls_video.h"
#include <hls_opencv.h>
typedef hls::stream<ap_axiu<8,1,1,1> >        AXI_STREAM;
void c_grav(AXI_STREAM& s_axis_video,AXI_STREAM& m_axis_video, int hsize_in, int vsize_in);
int main (int argc, char** argv) {


        // Load data in OpenCV image format
        IplImage* src = cvLoadImage("..\\..\\..\\triangle.pgm",CV_LOAD_IMAGE_GRAYSCALE);

        //Get input Image size
        CvSize size_in;
        size_in = cvGetSize(src);

        //Set output image size
        CvSize size_out;
        size_out.width = size_in.width;
        size_out.height = size_in.height;

        //Create Destination image
        IplImage* dst = cvCreateImage(size_out, src->depth, 1);

        //Create the AXI4-Stream
        AXI_STREAM src_axi, dst_axi,dst_axi2;

        // Convert OpenCV format to AXI4 Stream format
        IplImage2AXIvideo(src, src_axi);

        // Call the function to be synthesized
        c_grav(src_axi, dst_axi,size_in.width,size_in.height);
        IplImage2AXIvideo(src, src_axi);

        c_grav(src_axi, dst_axi2,size_in.width,size_in.height);

        // Convert the AXI4 Stream data to OpenCV format
        AXIvideo2IplImage(dst_axi2, dst);

        // Standard OpenCV image functions
        cvSaveImage("..\\..\\..\\out.png", dst);

        cvReleaseImage(&src);
        cvReleaseImage(&dst);

        return 0;
}
```

Q3 : Modifier le programme de l'IP pour calculer le CG de l'image. On affichera un carré blanc (10x10) à la position calculée. Tester en simulation.

Q4 : Exporter l'IP, l'intégrer dans le flux video du design tp3_p1, faire la synthèse et tester sur la zedboard.