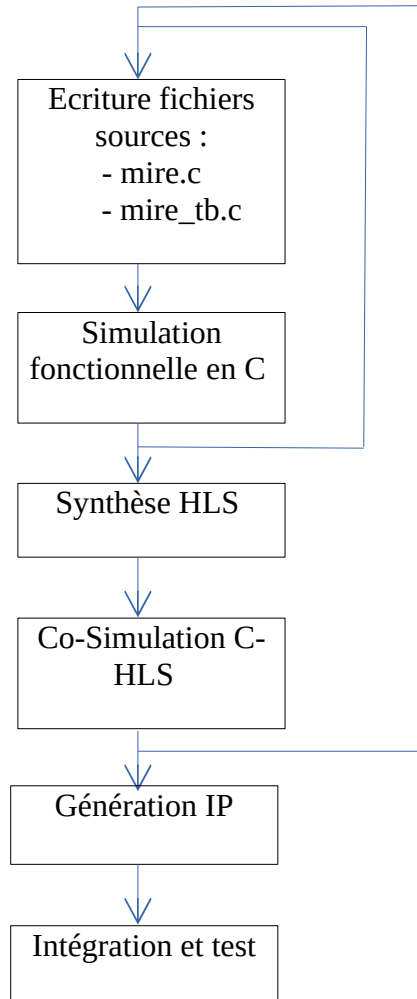


Dans ce TP, on va développer une mire VGA configurable à l'aide de Vivado HLS. On suivra le processus de développement suivant :



## **PARTIE 1**

Q1 : Créer un projet Vivado HLS (zedboard, Tclock 10ns). Ajouter les deux fichiers sources suivant (sources et testbench)

fichier source C

```
struct pixel {
    unsigned char red;
    unsigned char green;
    unsigned char blue;
};

void mire(volatile struct pixel p[640*480])
{
    #pragma HLS interface ap_hs port=p depth=1000
    int x,y;
    struct pixel a;
    for (y=0;y<480;y++)
        for (x=0;x<640;x++)
        {
            a.red=x;
            a.green=0;
            a.blue=0;
            *p++=a;
        }
}
```

fichier source testbench C

```
#include<stdio.h>

struct pixel {
    unsigned char red;
    unsigned char green;
    unsigned char blue;
};

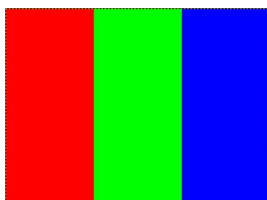
extern void mire(volatile struct pixel* p);
int main()
{
    struct pixel p[640*480];
    FILE *file_out;
    mire(p);
    int x,y;
    file_out=fopen("test.ppm","w");
    fprintf(file_out,"P3\n");
    fprintf(file_out,"640 480\n");
    fprintf(file_out,"255\n");

    for (y=0;y<480;y++)
    {
        for (x=0;x<640;x++) fprintf(file_out,"%d %d %d ",p[y*640+x].red,p[y*640+x].green,p[y*640+x].blue);
        fprintf(file_out,"\n");
    }
    fclose(file_out);
}
```

Q2 : Lancer la simulation C (**run C simulation**) et vérifier que le fichier image PPM généré est la mire prévue.

Q3 : Modifier le programme pour générer une mire horizontale.

Q4 : Modifier le programme pour générer la mire suivante.



Q5 : lancer une synthèse VHDL (run C synthesis). Décrire les différents signaux de l'entité mire dans le fichier VHDL (dans impl)

Q6 : lancer une co-simulation (run C/RTL cosimulation). Cliquer sur Open Wave Viewer et étudier les chronogrammes de la simulation. Comment est démarré le module mire, quel est son temps d'exécution ?

Q7 : faire un packaging de l'IP mire ( Export RTL). Noter le répertoire d'export de l'IP.

## **PARTIE 2**

Q1 : Dézipper le fichier tp1.zip. Ouvrir le projet vivado (2019.2) Inclure le répertoire d'export d'IP noté en Q5 de la partie 1. (Project Manager → Settings → Ip → Repository). Ajouter au schéma l'IP mire:

Q2 : synthétiser, implémenter, générer le bitstream et tester sur la zedboard.

Dans la suite, on va gérer 4 mires multiples configurées à l'aide des Switchs SW6 et SW7

Q3 : Modifier l'IP Mire sous Vivado HLS en ajoutant une variable dans les paramètres de la fonction et modifier les programmes C en conséquence. Tester en simulation C

Q4 : Synthétiser l'IP et l'exporter.

Q5 : Mettre à jour l'IP sur le Schéma. Créer un port d'entrée pour gérer la variable de génération. Re-générer le wrapper et modifier le top et le fichier de contraintes pour prendre en compte les switchs :

```
set_property PACKAGE_PIN H17 [get_ports {SW6}]; # "SW6"  
set_property PACKAGE_PIN M15 [get_ports {SW7}]; # "SW7"
```