1.1 主函式 TOP.vhd

將 FPGA 與各元件做連接，也負責與各電路連接，並將最後的輸出訊號輸出	
輸入	Clk: 描述:FPGA 內部時脈 100M Hz Video_clk： 描述：CLOCK，與 TVP5150 同步。 資料型態：std_logic。 reset： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。 Video_data_i2c: 描述：FPGA 拉 8 條線與 TVP5150 連接負責傳輸 data。 資料型態：std_logic_vector(7 downto 0)。
INOUT	Video_sda 描述：FPGA 拉線與 TVP5150 連接負責傳輸 sda。 資料型態：std_logic。 Video_scl 描述：FPGA 拉線與 TVP5150 連接負責傳輸 scl。 資料型態：std_logic。
輸出	R_out： 描述：由 FPGA 輸出至 VGA 的 port 共 4bit。 資料型態：std_logic_vector(3 downto 0)。 G_out： 描述：由 FPGA 輸出至 VGA 的 port 共 4bit。 資料型態：std_logic_vector(3 downto 0)。 B_out： 描述：由 FPGA 輸出至 VGA 的 port 共 4bit。 資料型態：std_logic_vector(3 downto 0)。 hsync： 描述：負責輸出給 vga 極性 。 資料型態：std_logic。 vsync： 描述：負責輸出給 vga 極性。

	資料型態：std_logic。

1.2 子函式 video_in_process_RGB.vhd

內容包含攝影機訊號到 RGB 或灰階值產生的過程。 與 vga.vhd 連接(負責製作 vga 輸出訊號) 與 i2c 連接(負責傳入指令給 TVP5150) 與 vedio_in 連接(接收抓取正確的資料) 而此子函式主要在做 YCrCb 值與 RGB 之間的轉換	
輸入	Video_clk： 描述：CLOCK，啟動 CLOCK 資料型態：std_logic。 rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。 Video_data_i2c: 描述：影像資料。 資料型態：std_logic_vector(7 downto 0)。
輸出	Vga_hs_cnt： 描述：VGA 輸出訊號時的水平訊號傳輸。 資料型態：integer 。 Vga_vs_cnt： 描述：VGA 輸出訊號時的垂直訊號傳輸。 資料型態：integer。 hsync： 描述：水平輸出訊號的極性訊號傳輸。 資料型態：std_logic。 vsync： 描述：垂直輸出訊號的極性訊號傳輸。 資料型態：std_logic。 Video_r_out： 描述：YCrCb 經轉換後得到的 R 值訊號。 資料型態：std_logic_vector(7 downto 0) Video_g_out： 描述：YCrCb 經轉換後得到的 G 值訊號。 資料型態：std_logic_vector(7 downto 0) Video_b_out：

	描述：YCrCb 經轉換後得到的 B 值訊號。 資料型態：std_logic_vector(7 downto 0) Video_gray_out： 描述：YCrCb 經轉換後得到的灰階值訊號。 資料型態：std_logic_vector(7 downto 0) Frame_id: 描述:電路正常動作的 flag。 資料型態:std_logic。
inout	Video_error: 描述:當偵測到畫面錯幀時，會重啟電路，重新讀取攝影機資料 資料型態:std_logic。
參數	Video_error: 描述:當偵測到畫面錯幀時，會重啟電路，重新讀取攝影機資料 資料型態:std_logic。

4.2.1 子函式 video_in

製作 FPGA 傳至 VGA 轉接頭的訊號	
輸入	clk： 描述：CLOCK，啟動 CLOCK 資料型態：std_logic。 rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。 Video_data_i2c: 描述：影像資料。 資料型態：std_logic_vector(7 downto 0)。
輸出	Video_start_en: 描述:資料開頭的訊號。 資料型態:std_logic。 Cnt_video_hsync: 描述:資料長度計算。 資料型態:integer。

4.2.2 子函式 VGA.vhd

製作 FPGA 傳至 VGA 轉接頭的訊號	
輸入	clk： 描述：CLOCK，啟動 CLOCK 資料型態：std_logic。 rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。
輸出	Vga_hs_cnt： 描述：FPGA 在製作 VGA 輸出訊號時的水平訊號。 資料型態：integer 。 Vga_vs_cnt： 描述：FPGA 在製作 VGA 輸出訊號時的垂直訊號。 資料型態：integer。 hsync： 描述：水平輸出訊號的極性訊號。 資料型態：std_logic。 vsync： 描述：垂直輸出訊號的極性訊號。 資料型態：std_logic。
參數	horizontal_resolution： 描述：這是水平解析度 資料型態：integer: 最高可以設置到 1280 horizontal_front_porch: 描述：這是水平同步脈衝結束時序 資料型態：integer: 最高可以設置到 48 horizontal_sync_pulse: 描述：這是水平同步脈衝 資料型態：integer: 最高可以設置到 112 horizontal_back_porch: 描述：這是水平同步脈衝開始時序 資料型態：integer: 最高可以設置到 248 vertical_resolution: 描述：這是垂直解析度 資料型態：integer: 最高可以設置到 1024 vertical _front_porch: 描述：這是水平同步脈衝結束時序 資料型態：integer: 最高可以設置到 1

	vertical_sync_pulse: 描述：這是水平同步脈衝 資料型態：integer: 最高可以設置到 3 vertical_back_porch: 描述：這是水平同步脈衝開始時序 資料型態：integer: 最高可以設置到 38

4.2.3 子函式 I2C.vhd

將啟動 IIC 命令給 TVP5150 使其動作	
輸入	clk： 描述：CLOCK，啟動 CLOCK 資料型態：std_logic。 rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。
INOUT	SDA： 描述：控制 SDA 讀寫。 資料型態: std_logic： 。 SCL： 描述：控制 SCL 讀寫。 資料型態：std_logic。
參數	type_I2C： 描述：這是要寫入 IIC 協定的指令 資料型態：std_logic_vector(23 downto 0)。 以陣列的形式表示。一次長度 24bit CMD_NUM_TVP5150: 描述：這是控制需要傳幾條指令的參數 資料型態：integer: IIC 的指令為 2 筆，但如需其他指令可自行設定，這個參數會連動其他與他有關的參數

1.3 子函式 UART(其中包含了 RX 與 TX)

4.3.1 UART.RX

Arduino 到 FPGA 的 UART RX (UART_RX.vhd) 這是 FPGA 的 RTL block，它的功能是接收 Arduino 用 UART 協定傳送的資料，以一個 byte 進行輸出	
輸入	i_Clk： 描述：CLOCK

	資料型態：std_logic。 i_RX_Serial： 描述：這是 FPGA 的 UART RX 接腳 資料型態：std_logic。
輸出	o_RX_DV： 描述：這是「FPGA 接收完 1 個 BYTE 的資料」的 flag，可以開始提取資料。 資料型態：std_logic。 o_RX_Byte： 描述：這是 FPGA 所接受的資料，資料長度為 1 個 byte。 資料型態：std_logic_vector(7 downto 0)。
參數	g_CLKS_PER_BIT ： 描述：這是計算 clock 的除頻參數，用來逼近 UART 的傳輸頻率(傳輸鮑率)。 g_CLKS_PER_BIT = (Frequency of i_Clk)/(Frequency of UART) Example : 20 MHz Clock, 9600 baud UART (20000000) / (9600) = 2083 資料型態：integer。
方法	這是 FPGA 的 RTL block，它的功能是接收 Arduino 用 UART 協定傳送的資料，以一個 byte 進行輸出。 每個 CLK 會把 i_RX_Serial 的當前資料存進暫存器裡面。 它有一個計數器會不斷計數，直到數值為 g_CLKS_PER_BIT - 1，再歸 0。 它的 FSM 有 5 種狀態： 1. 初始化 2. 開始位元 3. 資料接收(8 bit) 4. 停止位元 5. 切割

	1. 進行歸 0 計數器與 o_RX_Byte，以及等待 FPGA RX 的接腳接收到低準位的電壓，便進入開始位元狀態。 2. 讓計數器跑到 g_CLKS_PER_BIT-1 的一半，判斷 FPGA RX 的接腳是否仍處於低準位的電壓，是低準位的話便進入資料接收狀態，不是的話回到初始化的狀態，因為其頻率並非如同設定。讓計數器跑到 g_CLKS_PER_BIT-1 的一半是為了使判斷 FPGA RX 狀態的 CLK 位於狀態持續時間的中間。 3. 讓計數器跑到 g_CLKS_PER_BIT-1，將 FPGA RX 狀態寫進 o_RX_Byte，並歸 0 計數器，等跑完 7 次也就是 FPGA RX 狀態寫了 8 次給 o_RX_Byte，便進入停止位元。 4. 讓計數器跑到 g_CLKS_PER_BIT-1，將 r_RX_DV 拉為高準位，讓外部連接的 Block 知道 UART 資料傳輸完畢，然後進入切割狀態。 5. 這個狀態會進行 1 個 CLK，將 r_RX_DV 拉回低準位，然後就回到初始化的狀態。

4.3.2 UART_TX

Arduino 到 FPGA 的 UART RX (UART_RX.vhd) 這是 FPGA 的 RTL block，它的功能是透過 UART 發送資料給 Arduino，以一個 byte 進行輸出。	
輸入	i_Clk： 描述：CLOCK，啟動 CLOCK 資料型態：std_logic。 i_TX_DV： 描述：啟動 FPGA 使用 UART 的 FLAG。 資料型態：std_logic。 i_TX_Byte： 描述：FPGA 使用 UART 所傳送的資料，每次傳送一個 Byte。 資料型態：std_logic_vector(7 downto 0)。
輸出	o_TX_Active： 描述：FPGA 的 UART 處於動作中的 FLAG。 資料型態：std_logic。 o_TX_Serial： 描述：UART 在 FPGA 的 TX 接腳。 資料型態：std_logic。 o_TX_Done：

	描述：UART 資料傳輸完的 FLAG。 資料型態：std_logic。
參數	g_CLKS_PER_BIT ： 描述：這是計算 clock 的除頻參數，用來逼近 UART 的傳輸頻率(傳輸鮑率)。 g_CLKS_PER_BIT = (Frequency of i_Clk)/(Frequency of UART) Example : 20 MHz Clock, 9600 baud UART (20000000) / (9600) = 2083 資料型態：integer。
方法	這是 FPGA 的 RTL block，它的功能是透過 UART 發送資料給 Arduino，以一個 byte 進行輸出。 它有一個計數器會不斷計數，直到數值為 g_CLKS_PER_BIT - 1，再歸 0。 它的 FSM 有 5 種狀態： 1. 初始化 2. 開始位元 3. 資料傳送(8 bit) 4. 停止位元 5. 切割 1. 歸 0 計數器和輸出(o_TX_Active、o_TX_Serial、o_TX_Done)。等待 i_TX_DV 被拉到高準位，進入開始位元的狀態。 2. 將 o_TX_Active 拉到高準位，告訴其他 FPGA 的 Block，UART TX 正在動作；將 o_TX_Serial 拉到低準位，告訴 arduino，這邊要傳送資料了。等待計數器跑到 g_CLKS_PER_BIT-1，便進入資料傳送的狀態。 3. 將 i_TX_Byte 的資料由 0 到 7 讓 o_TX_Serial 去輸出，index 變換的時間間隔是計數器數到 g_CLKS_PER_BIT-1。i_TX_Byte 的資料都輸出完，進入停止位元的狀態。 4. 將 o_TX_Serial 拉到高準位，告訴 Arduino，傳輸資料完畢。等待計數器跑到 g_CLKS_PER_BIT-1，進入切割的狀態，並把 o_TX_Done 拉到低準位，告訴其他 FPGA 的 Block，UART 傳送完了。 5. 將 o_TX_Active 拉到低準位，告訴 FPGA 的 controller 可以進行下一筆資料可以傳送了。回到初始化的狀態。

1.4 子函式 blk_mem_gen.vhd

由 Xilinx 內部的 IP 製作的 Block_ram 負責儲存資料與讀取該資料 可以自行設定 其 深度 廣度 與單筆位元數(設定須符合 FPGA 板的規格)	
輸入	clka： 描述：讀取或寫入的時脈。 資料型態：std_logic。 wea： 描述：動作智能。 資料型態：std_logic。 addra: 描述：總資料長度的記憶體計數器。負責讀寫 資料型態：std_logic_vector(依長度給予適當的位元數)。 dina: 描述：要傳進去暫存器的資料。 資料型態：std_logic_vector(依照想要傳入的資料給予適當的位元數)。 clkb: 描述：讀取或寫入的時脈。 資料型態：std_logic。 addrb: 描述：總資料長度的記憶體計數器。負責讀寫 資料型態：std_logic_vector(依長度給予適當的位元數)。
輸出	Doutb: 描述:讀出存入的資料。 資料型態:std_logic_vector(資料型態與 dina 相同)。

1.5 子函式 Ping_pong_buffer.vhd

將 8 個暫存器(720*1)影像資料串起來，8 位元為單位以 pipe 的形式一起傳出	
輸入	clk： 描述：Clock，啟動 Clock。 資料型態：std_logic。 rst： 描述：重製訊號。 資料型態：std_logic。 Video_data: 描述：影像資料。

	資料型態：std_logic_vector(7 downto 0)。 Vga_hs_cnt: 描述：使資料對其於 vga 輸出訊號。 資料型態：integer。 Vga_vs_cnt: 描述：使資料對其於 vga 輸出訊號。 資料型態：integer。
輸出	Ping_pong_out_8: 描述:整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。 Ping_pong_out_7: 描述: 整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。 Ping_pong_out_6: 描述: 整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。 Ping_pong_out_5: 描述: 整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。 Ping_pong_out_4: 描述: 整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。 Ping_pong_out_3: 描述: 整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。 Ping_pong_out_2: 描述: 整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。 Ping_pong_out_1: 描述: 整理過的影像資料。 資料型態: std_logic_vector(7 downto 0)。

1.6 子函式 Harris.vhd

製作 FPGA 傳至 VGA 轉接頭的訊號	
輸入	clk： 描述：CLOCK，啟動 CLOCK。 資料型態：std_logic。

	rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。 Vga_hs_cnt: 描述：使資料對其於 vga 輸出訊號。 資料型態：integer。 Vga_vs_cnt: 描述：使資料對其於 vga 輸出訊號。 資料型態：integer。 Threshold: 描述: harris 推導中參數 R 的門檻值 資料型態: std_logic_vector(43 downto 0)
輸出	Harris_out: 描述:判定該點為 harris corner 的訊號。 資料型態:std_logic。

1.7 子函式 ORB.vhd

製作由 Harris corner 衍生的 BRIEF 描述子	
輸入	clk： 描述：CLOCK，啟動 CLOCK。 資料型態：std_logic。 rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。 Vga_hs_cnt: 描述：使資料對其於 vga 輸出訊號。 資料型態：integer。 Vga_vs_cnt: 描述：使資料對其於 vga 輸出訊號。 資料型態：integer。 Video_data: 描述：影像資料。 資料型態：std_logic_vector(7 downto 0)。 Kp_en: 描述: 將 Harris corner 訊號導入函式。 資料型態: std_logic。 Save_en:

	描述: 控制內部記憶體是否進行動作。 資料型態: std_logic。
輸出	B_out: 描述: 描述子的座標(X,,Y)。 資料型態:std_logic_vector(83 downto 0)。
參數	pairs_table: 描述: 128 點對的查表。 centroid_x: 描述: X 軸方向 sobel 強度計算。 centroid_Y: 描述: Y 軸方向 sobel 強度計算。

4.7.1 CORDIC.vhd

輔助 ORB.vhd 計算角度 arctan(y/x)	
輸入	clk： 描述：CLOCK，啟動 CLOCK。 資料型態：std_logic。 rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。 X_in: 描述: X 軸方向 sobel 強度計算。 資料型態: std_logic_vector(14 downto 0)。 Y_in: 描述: X 軸方向 sobel 強度計算。 資料型態: std_logic_vector(14 downto 0)。
輸出	X_out: 描述: (x^2+y^2)^0.5 的解。 資料型態: std_logic_vector(17 downto 0)。 Y_out: 描述: (x^2+y^2)^0.5 的解。 資料型態: std_logic_vector(17 downto 0)。 Z_out: 描述: arctan 的角度 : 90~0 資料型態: std_logic_vector(7 downto 0)。

1.8 子函式 matching.vhd

前後兩幀描述子匹配	
輸入	clk： 描述：CLOCK，啟動 CLOCK，此時脈為 FPGA 內部時脈。 資料型態：std_logic。 rst： 描述：當此訊號為 0 時，重製訊號。 資料型態：std_logic。 Video_clk: 描述: CLOCK，啟動 CLOCK。 資料型態: std_logic。 a: 描述: 前幀的描述子(X,Y)座標。 資料型態: std_logic_vector(83 downto 0)。 b: 描述: 後幀的描述子(X,Y)座標。 資料型態: std_logic_vector(83 downto 0)。
輸出	MATE_D: 描述: 匹配到的前後幀(X,Y)軸座標。 資料型態: std_logic_vector(39 downto 0)。