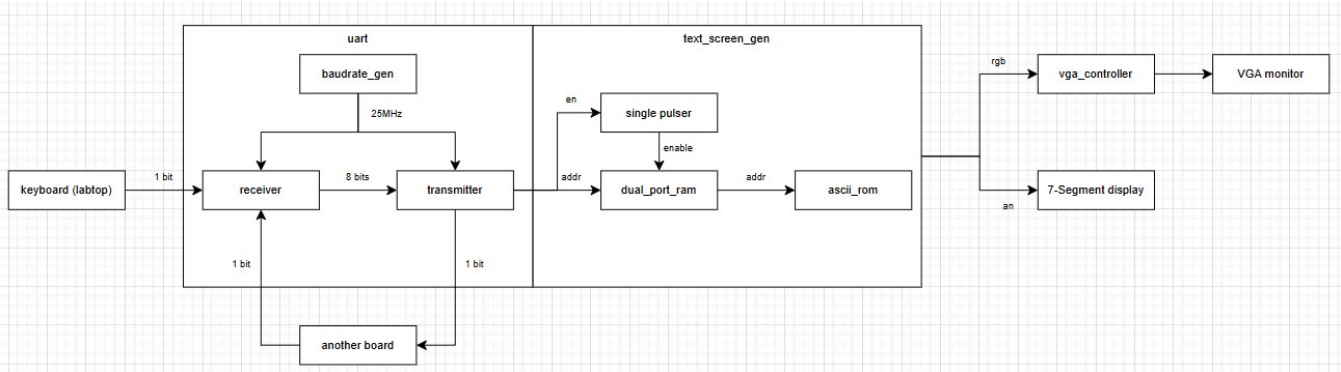


Hardware Synthesis Lab : Final Project



- module top**(input clk, input reset, input [7:0] sw, input ja1, output ja2, output wire RsTx, input wire RsRx, output hsync, output vsync, output [11:0] rgb, output [6:0] seg, output dp, output [3:0] an, input up);
clk : สัญญาณนาฬิกา | reset : ปุ่ม reset | sw : เก็บข้อมูลจาก switch (8 bits) | ja1 : รับข้อมูลจากบอร์ดอื่น | ja2 : ส่งข้อมูลให้บอร์ดอื่น | RsTx : ส่งให้ tera term | RsRx : รับจาก keyboard | hsync : horizontal synchronization | vsync : vertical synchronization | rgb : ค่าสี pixels ที่จะแสดงบนจอ | seg, dp, an : เพื่อแสดงคาบน 7-Segment | up : มีไว้เพื่อบอกว่าจะรับจาก switch
เป็น module ที่เป็น main ในการเรียกใช้ module อื่นอีกที มีการทำ single pulser, แสดงข้อมูลบน 7-Segment และจอที่ควบคุมด้วย module vga_controller generate ข้อมูลที่จะแสดงด้วย module text_screen_gen และควบคุมการติดต่อและการใช้ uart
- module vga_controller**(input clk_100MHz, input reset, output video_on, output hsync, output vsync, output p_tick, output [9:0] x, output [9:0] y);
clk_100MHz : สัญญาณนาฬิกา 100MHz | reset : ปุ่ม reset | video_on : บอกว่าอยู่ในพื้นที่แสดงผลหรือไม่ | p_tick : tick สำหรับการสร้างความถี่ 25 MHz เพื่อการอัปเดตพิกเซล | x, y : ค่าพิกัดปัจจุบัน
ใช้สำหรับการควบคุมหน้าจอ VGA โดยการสร้างสัญญาณซิงโครไนซ์ และการคำนวณตำแหน่งของพิกเซล รวมถึงการระบุว่าจะอยู่ในพื้นที่แสดงผลหรือไม่ เพื่อให้การแสดงผลบนหน้าจอทำงานตามมาตรฐาน VGA
- module text_screen_gen**(input clk, input reset, input video_on, input [7:0] sw, input [9:0] x, y, output reg [11:0] rgb, input [7:0] data_fk, input en);
video_on : บอกว่าอยู่ในพื้นที่แสดงผลหรือไม่ | x, y : พิกัดของแกน x, y | data_fk : ข้อมูลที่รับมาจาก keyboard ที่บันทึกเก็บไว้ใน RAM | en : สัญญาณ enable | rgb : ค่าสี pixels ที่จะแสดงบนจอ
สร้างหน้าจอข้อความที่แสดงผลด้วยอักษรตามตำแหน่งพิกเซลให้ rgb โดยใช้ Font ROM สำหรับการเรนเดอร์ข้อความ และใช้ Tile RAM ในการบันทึกตำแหน่งตัวอักษร โดยเคอร์เซอร์ถูกควบคุมโดยตำแหน่งและจะมีการเลื่อนตำแหน่งเมื่อรับคำสั่ง
- module singlepulser**(input clk, input en, output enable);
enable : ค่า enable | en : ค่าที่ input เข้ามาจากปุ่ม
ทำให้ส่งสัญญาณ enable แค่ 1 สัญญาณ
- module ascii_rom**(input clk, input reset, input video_on, input [7:0] sw, input [9:0] x, y, output reg [7:0] data);
addr : ค่าของ address ที่รับมา | data : ข้อมูลที่อยู่ใน ROM ที่มีตำแหน่ง addr
ใช้สำหรับเก็บข้อมูลที่จะแสดงผล
- module dual_port_ram**(input clk, input we, input reset, input [11:0] addr_a, addr_b, input [7:0] din_a, output [7:0] dout_a, dout_b);
we : สัญญาณเมื่อต้องการเขียน | addr_a, addr_b : ค่าของ address ที่รับมาสำหรับ port a และ b, din_a : ข้อมูลที่เข้ามาที่ port a | dout_a, dout_b : ข้อมูลที่ออกมาทาง port a และ b
มีไว้เพื่อเก็บข้อมูลเข้าใน RAM ได้ 1 port และสามารถอ่านข้อมูลได้ 2 port
- module uart**(input clk, input RsRx, input [7:0] data_transmit, input dte, output RsTx, output [7:0] data_received, output received);
RsRx : ข้อมูลที่ได้รับมา 1 bit | data_transmit : ข้อมูลที่จะ transmit | dte : สัญญาณเพื่อบอกว่าจะส่ง | RsTx : ข้อมูลที่จะส่งไปบอร์ดอื่น | data_received : ข้อมูลที่ได้รับมา | received : สัญญาณเพื่อบอกว่ารับหมดแล้ว
เป็น module เพื่อควบคุมการทำงานของ uart โดยจะมีการเรียก receiver และ transmitter ภายใน โดยจะใช้สัญญาณนาฬิกาที่ได้จากการคำนวณโดย module baudrate_gen
- module baudrate_gen**(input clk, output reg baud);
baud : สัญญาณนาฬิกาที่ปรับแล้ว มีไว้เพื่อถ่วงความเร็วของ clk
- module uart_rx**(input clk, input bit_in, output reg received, output reg [7:0] data_out);
bit_in : ข้อมูลที่ได้รับมาทีละ 1 bit | received : สัญญาณเพื่อบอกว่ารับหมดแล้ว | data_out : ข้อมูลที่ได้รับมาที่รวมไว้ 8 bits
ใช้สำหรับรับข้อมูลจาก keyboard หรือ บอร์ดอื่นๆ โดยจะส่งข้อมูลออกไปใช้ต่อทีละ 8 bits
- module uart_tx**(input clk, input [7:0] data_transmit, input ena, output reg sent, output reg bit_out);
data_transmit : ข้อมูลที่ต้องการ transmit | ena : สัญญาณ enable | sent : สัญญาณบอกว่าจะส่งหมดแล้ว | bit_out : ข้อมูลที่ส่งออกทีละ 1 bit
ใช้สำหรับส่งข้อมูลออกไปแสดงบนจอ หรือส่งข้อมูลให้บอร์ดอื่น
- Module ที่ใช้สำหรับแสดงข้อมูลบน 7-Segment** ตามที่ทำในแลป