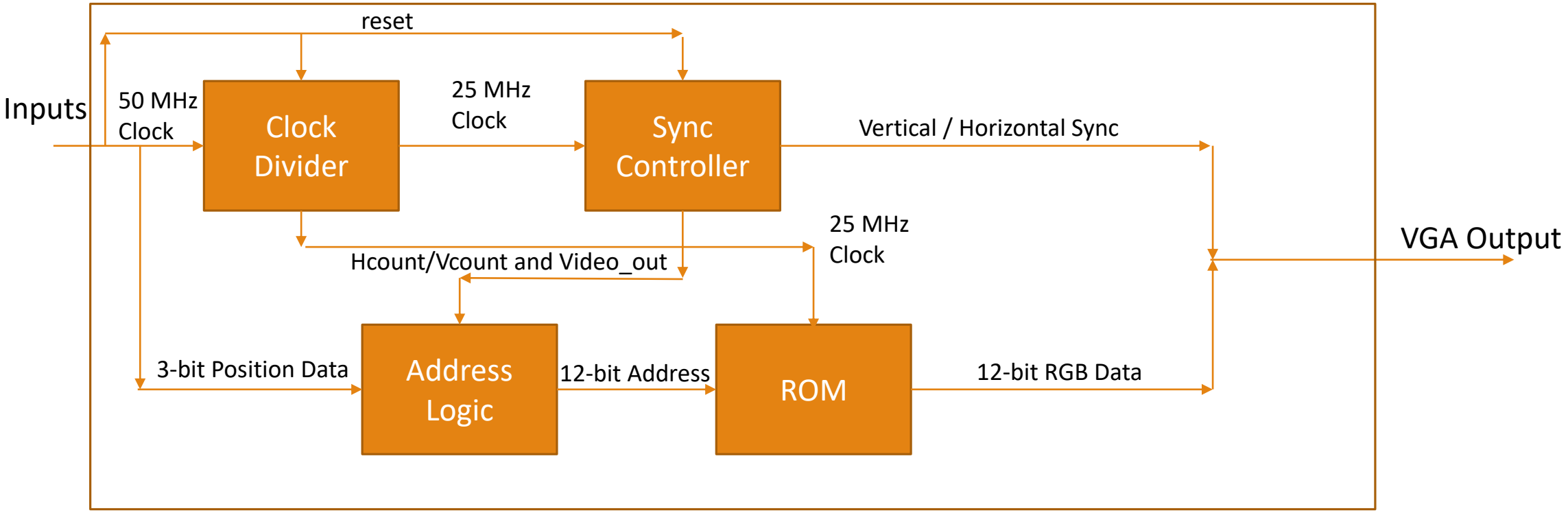
VGA Implementation

Block Diagram



Clock Divider

Inputs:

- 50 MHz Clock
- Reset

Outputs:

- 25 MHz Clock

Use the same clock divider as previous labs

Sync Controller

Inputs:

- 25 MHz Clock
- Reset

Outputs:

- Hcount
- Vcount
- Video_out
- Horizontal Sync
- Vertical Sync

Comprised of 2 counters, Hcount and Vcount, and combinational logic to decode the sync values and video_out value

Sync Controller

Hcount is incremented every clock while Vcount is incremented at a specific Hcount value, H_VERT_INC.

The syncs are high when the respective counts are in the range specified in the VGA_lib

- HSYNC_BEGIN – HSYNC_END and VSYNC_BEGIN – VSYNC_END

Video_out is decoded by determining when the counters are a position that would be on the screen

- H_DISPLAY_END and V_DISPLAY_END

Testing Sync Controller

Once you have a working Sync Controller you can start to output data to a VGA monitor. The easiest way is to hardcode outputs using if statements

```
If (video_out is true)
  If (Hcount == some number)
            Red <= 1111, Green <= 0000, Blue <= 0000
  Elsif (Hcount == some other number)
            Red <= 0000, Green <= 1111, Blue <= 0000
  Elsif (Hcount == some other number)
            Red <= 0000, Green <= 0000, Blue <= 1111
  End if
End if
```

This will generate a red, a green, and a blue horizontal line on your screen

Address Logic

Inputs:

- Hcount
- Vcount
- video_out
- Position

Outputs:

- 12-bit Address

Takes in the position value, Hcount, Vcount, and a video_out enable and manipulates Hcount and Vcount into a 12-bit address for the ROM depending on the position and video_out values.

ROM

Inputs:

- 25 MHz Clock
- 12-bit Address

Outputs:

- 12-bit Data

Holds the RGB value for a specific pixel, when combined these pixels make up an image. Will be created by using the ROM 1 Port Quartus tool and a provided mif file.

Creating the ROM Component

The tool is under the IP Catalog -> Library -> Basic Functions -> On-Chip Memory -> ROM 1 Port

Name it vga_rom and select VHDL for IP variation file type

Set “q” bits to 12 and amount of words to 4096

Make sure “q” is not registered

Select the mif file when asked

Leave everything else the same

You can now reference the output VHDL has a component in your top level

Tips for Simulating in ModelSim

Use multiple cursors to be able to measure duration of the refresh cycles

Utilize the find next/previous transition functions to easily find spots that you want place a cursor at

Make sure that the timing parameters for the horizontal and vertical components match the ones provided in the lab document before moving on from the VGA sync logic

When running simulations that include the ROM make sure you include the altera_mf.altera_mf_components package in the altera_mf library in the testbench and have the brom.mif in the same directory as the ROM file in case it is using relative pathing