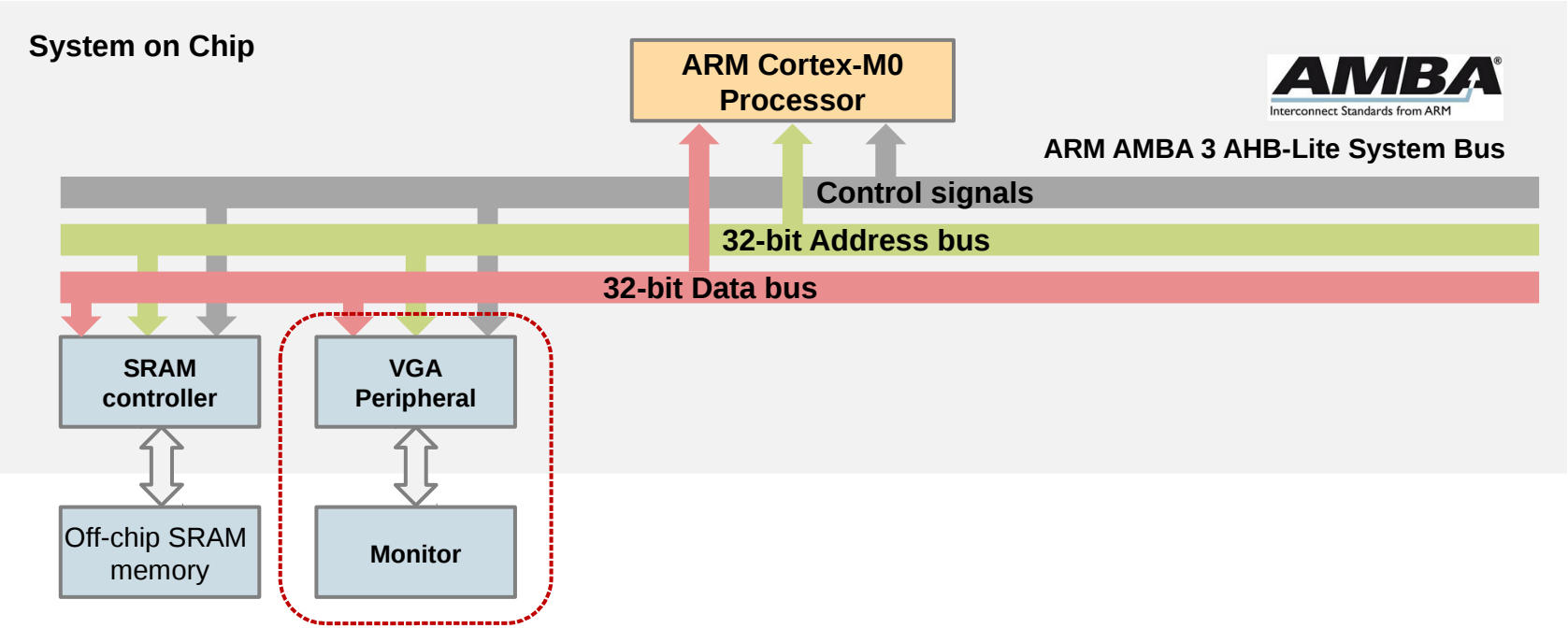
Design and Implementation of an AHB VGA Peripheral

Module Overview

- Learn about VGA interface;
- Design and implement an AHB VGA peripheral;
- Program the peripheral using assembly;
- Lab Demonstration.



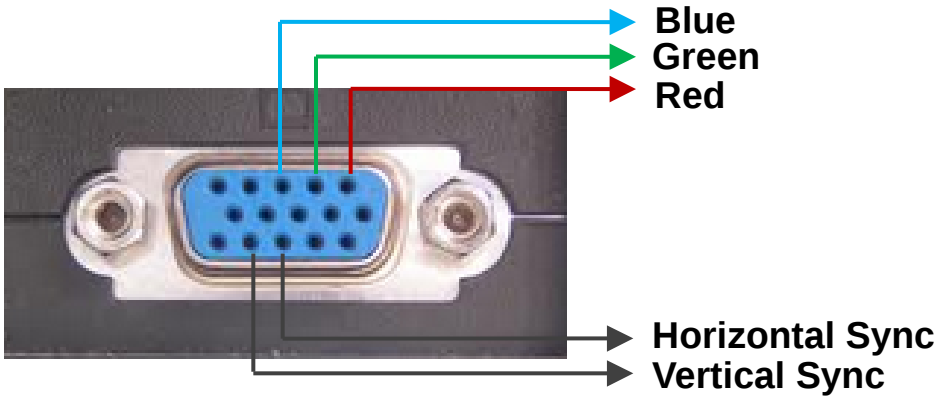
Module Syllabus

- VGA Overview
- VGA Interface
- VGA Timing
- AHB VGA Hardware Implementation
- VGA Image Buffer
- VGA Console
- Lab Practice

VGA Overview

VGA Overview

- Video Graphics Array (VGA) connector
  - Five analog components
    - Blue, green, red
    - Horizontal and vertical synchronization
  - Designed in 1987, still used nowadays, but most of them are superseded by Digital Visual Interface (DVI) and High-Definition Multimedia Interface (HDMI).



How VGA Signals Work

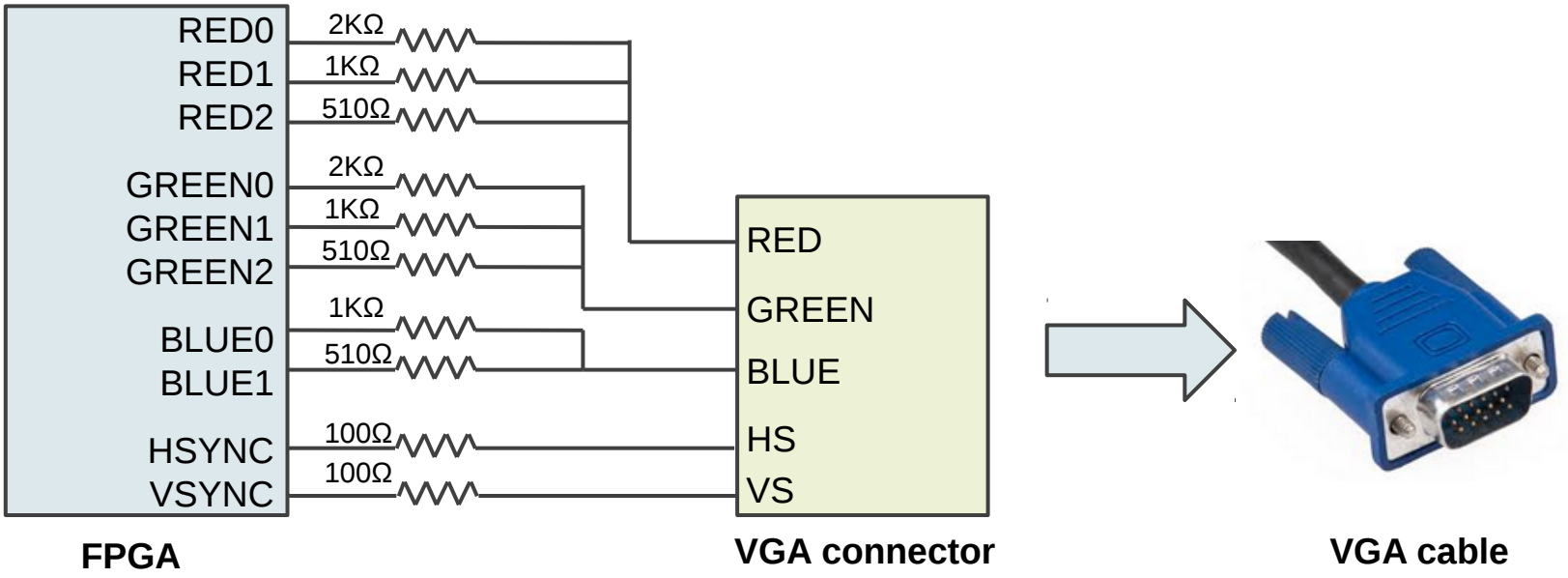
- CRT monitors display images on a phosphor-coated screen using amplitude-modulated moving electron beams.
- Beams horizontally move from left to right, and vertically from top to bottom.
- As the beam moves over the whole screen, the colour information of the pixel that currently being scanned is given from the VGA cable.
- The horizontal synchronization is used to reset the beam to the start of the next line.
- The vertical synchronization is used to start the next frame.
- The monitor will adjust its scanning frequency and screen resolution according to the synchronization signal given from the user.

How VGA Signals Work

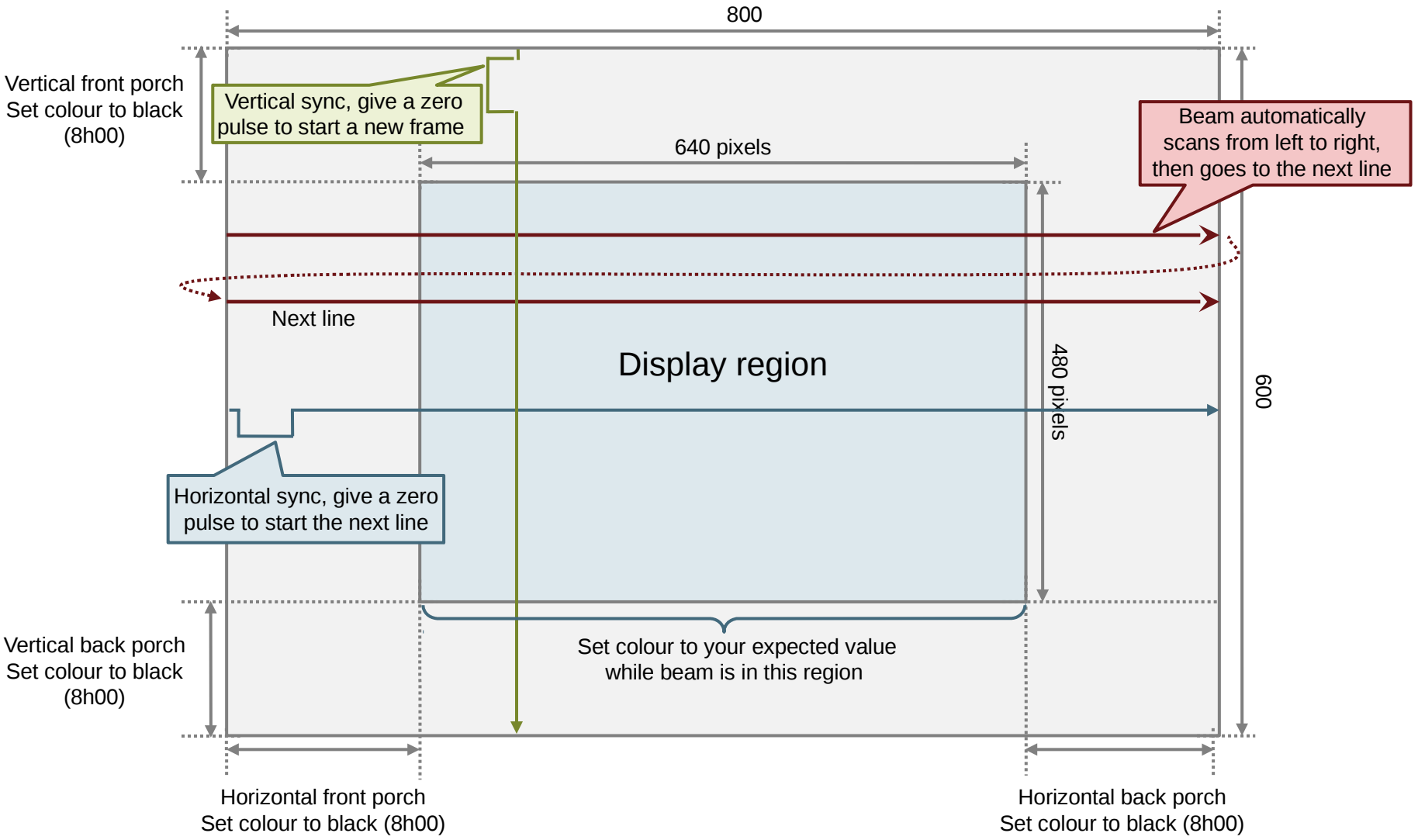
- During the time when the beam is resetting to the beginning of next line (or the next frame), the colour information will not be used, and this particular region is known as front/ back porch.
- Despite from different displaying technologies, such as CRT or LCD, most of the monitors can provide a standard interface, such as VGA or DVI.
- The VGA timings information for different resolutions is specified and published by the VESA organization.

VGA Interface

- The digital outputs from the FPGA can be converted to analog and connected to the VGA connector using resistor-divider circuits,
- For example, the Digilent Nexys3 board uses 10 signals, including 8-bit colour and two standard sync signals, thus 256 colour levels can be presented.
- The following slides give an example timing for a 640×480 resolution.

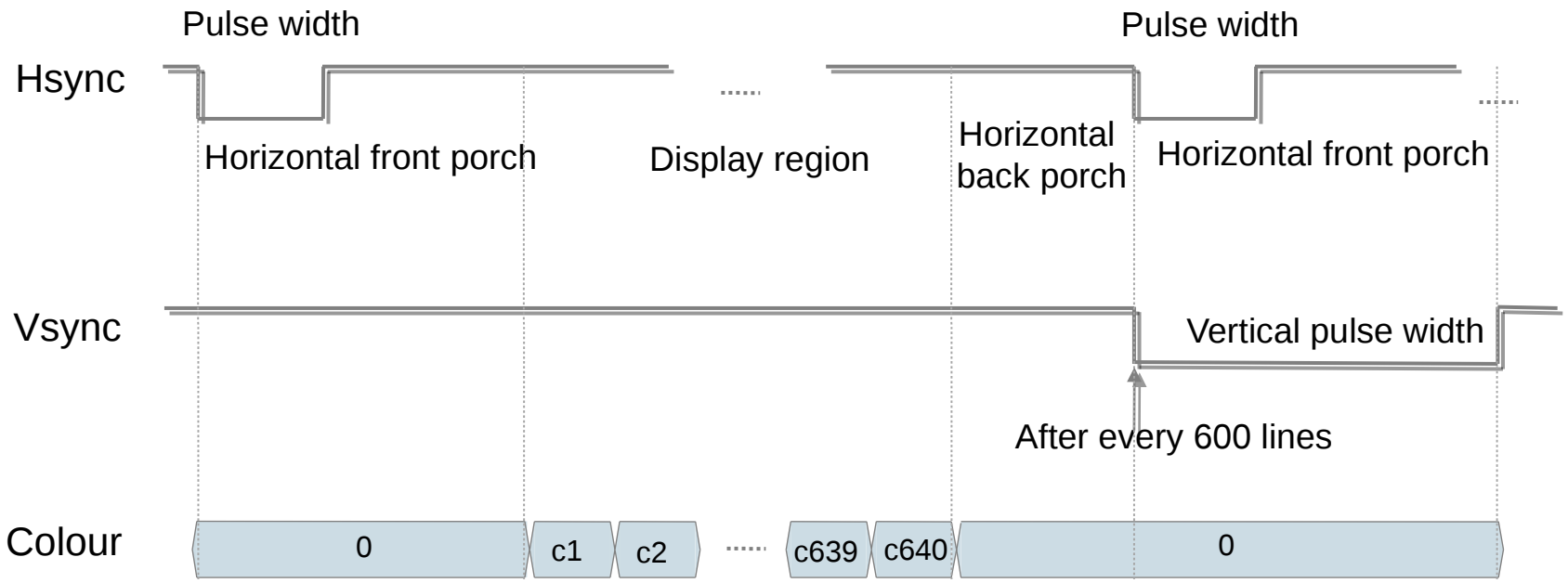


VGA Timing



VGA Timing

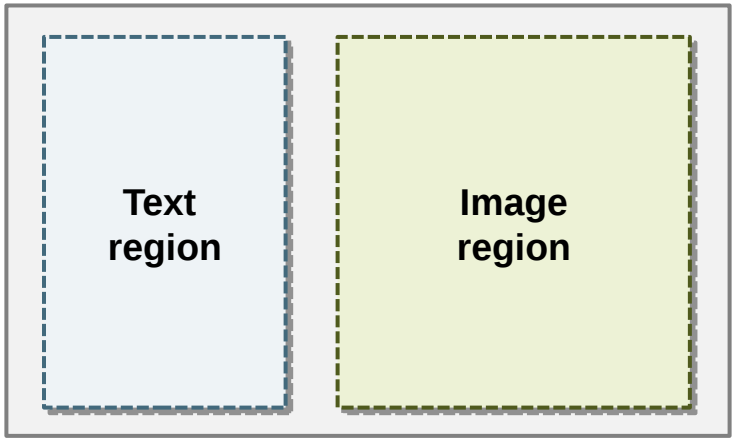
- VGA timing for 640 x 480 resolution;
- 25MHz clock frequency;
- Parameters can be found inside the code.



AHB VGA Hardware Implementation

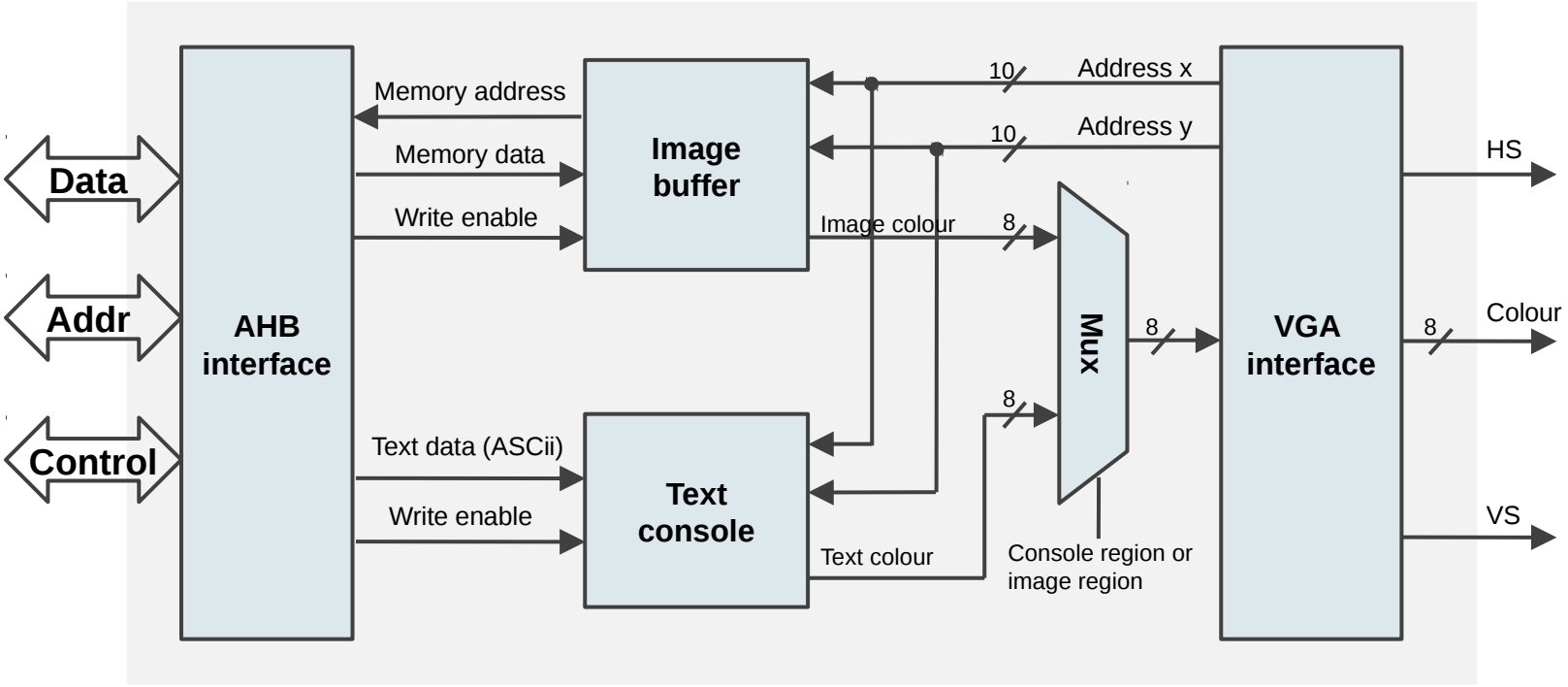
Hardware Implementation

- In our proposed hardware implementation, the full region of the screen is divided to two sub-regions:
  - Text region (console) -- display text strings in a relatively high resolution;
  - Image region (frame buffer) – display a desired image in a lower resolution;
- Ideally, all the pixels' information is stored in one frame buffer. However, since the on-chip memory is not sufficient, the resolution of the frame is reduced. Hence, to still display clear texts at the same time, the text region is separated, where the dynamic hardware logic is used instead of the frame buffer.



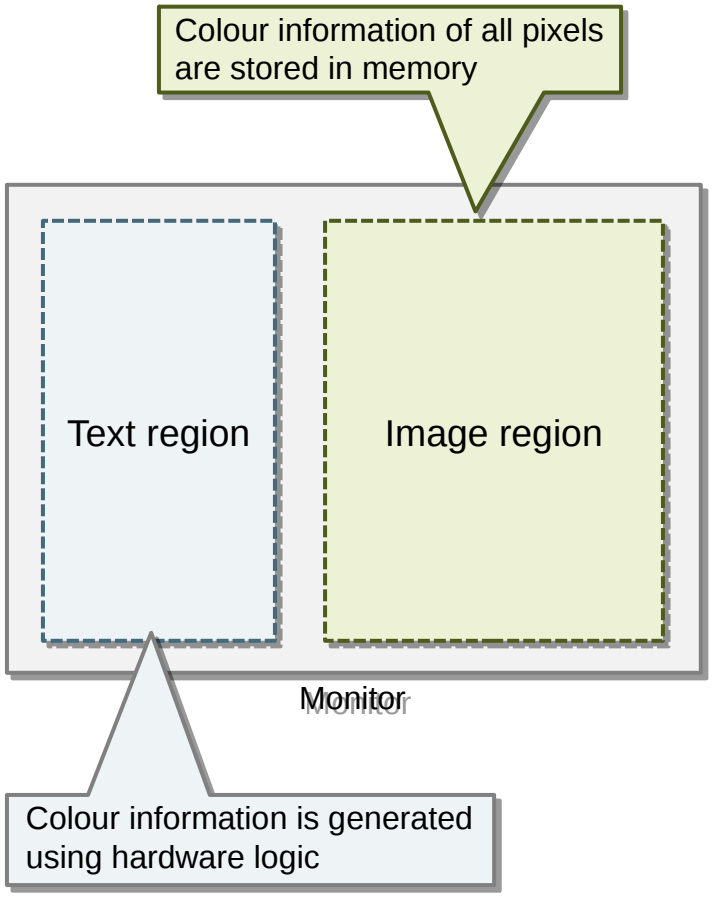
Hardware Interconnection

- The VGA peripheral can display texts and images on a monitor through a VGA cable.
- The VGA peripheral consists of 3 components: a VGA interface, an image buffer for displaying image, and a text console module for displaying texts.



Hardware Implementation

- VGA interface
  - Generates synchronization signals to the VGA port;
  - Is directly connected to external pins of the VGA port;
  - Outputs the address of the current pixel;
- Image buffer
  - Stores the colour information of all pixels in the image region;
  - Is implemented on a dual-port memory;
- Text console
  - Displays texts in the text region.
  - Is implemented on hardware logics.



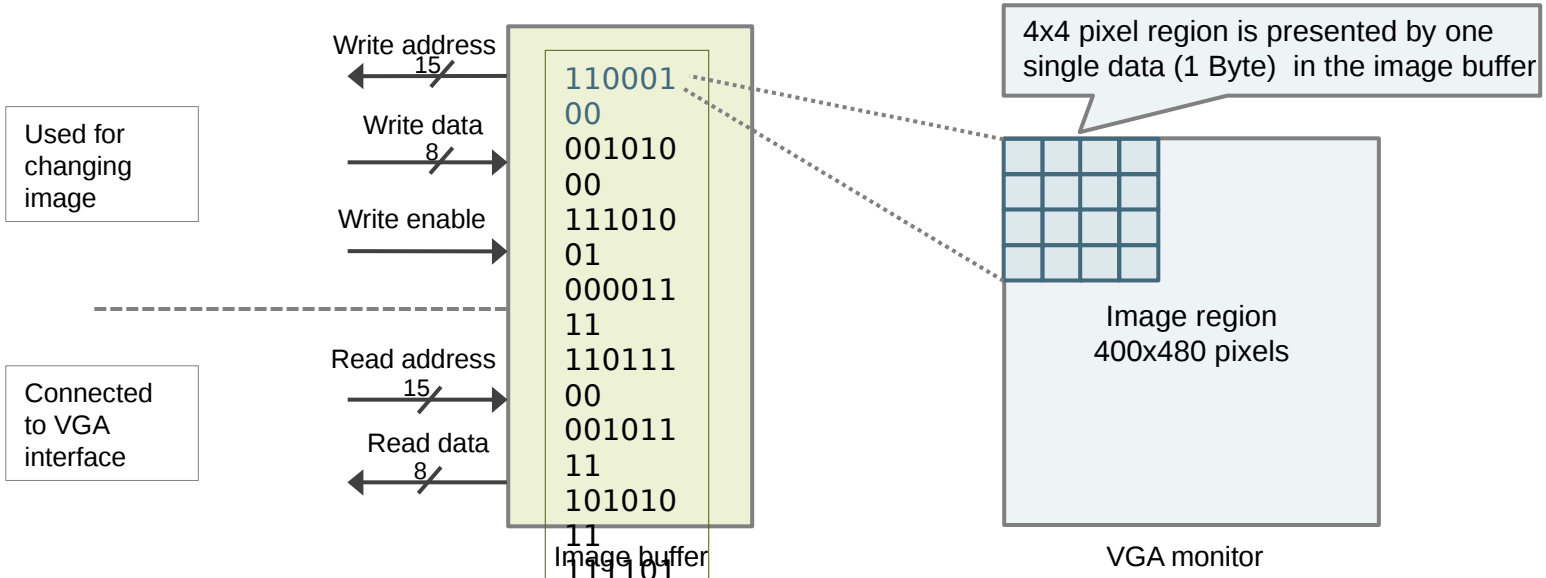
VGA Interface

- VGA signals

Name	Description
vga_red[2:0]	3-bit red signal
vga_green[2:0]	3-bit green signal
vga_blue[1:0]	2-bit blue signal (less sensitive to eyes)
hsync	Horizontal synchronization signal, one pulse indicates the start of the next line
vsync	Vertical synchronization signal, one pulse indicates the start of the next frame

VGA Image Buffer

- The image buffer stores the RGB information for all the pixels in the image region.
- The buffer is implemented on a dual-port RAM, which allows pixels to be modified at the same time the VGA interface is reading the pixel.
- Some chips do not have a large on-chip memory, such as on-chip SRAM. In such a case, the resolution can be reduced by mapping multiple pixels to a single data in the memory. For example, a 4x4 pixel region can be presented by one single data in the image buffer.



VGA Console

- The VGA console module is used to generate the colour information for pixels in the text region.
- The colour information is generated dynamically using hardware logics, rather than storing every single pixel in a memory, whereby the valuable on-chip memory can be saved.

Memory Space

- The memory space is allocated as follow:

Peripheral	Base address	End address	Size
MEM	0x0000_0000	0x4FFF_FFFF	167MB
VGA	0x5000_0000	0x50FF_FFFF	16MB

Memory Space

- The internal memory space of VGA is divided into two regions
  - Console text: 1 word (4 byte) space to print a character
  - Image buffer: the rest of the space is used to store pixels in the image region.

Register	Base address	End address	Size
Console text	0x5000_0000	0x5000_0000	4 Byte
Image buffer	0x5000_0004	0x50FF_FFFF	15999996 Byte

Lab Practice

Lab Practice

- Hardware design
  - Design and implement the peripheral (VGA peripheral) in hardware using Verilog;
- Software programming
  - Test the peripheral using Cortex-M0 processor programed in assembly language;
- System demonstration
  - Display text strings and images on the monitor.

Useful Resources

- Reference1
  - Nexys3 Reference Manual:

    http://www.digilentinc.com/Data/Products/NEXYS3/Nexys3_rm.pdf