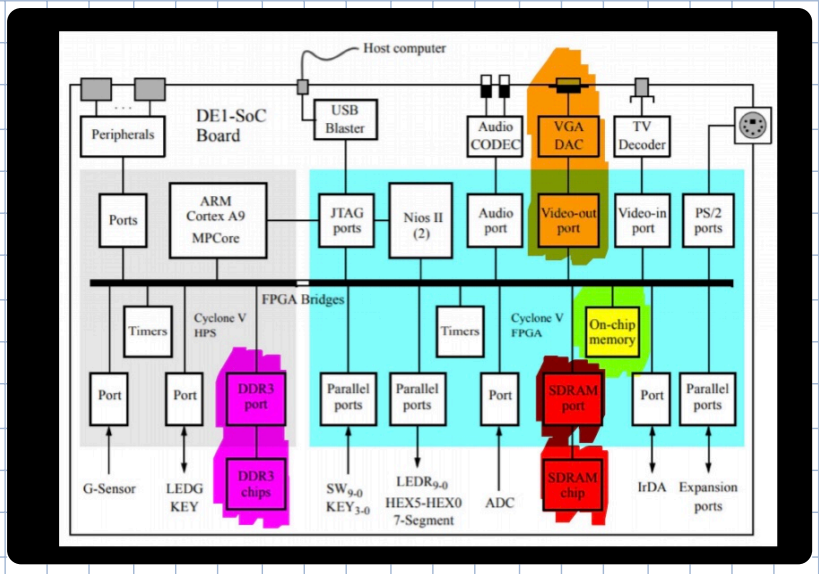
Lab 7 Prep: using a VGA display:

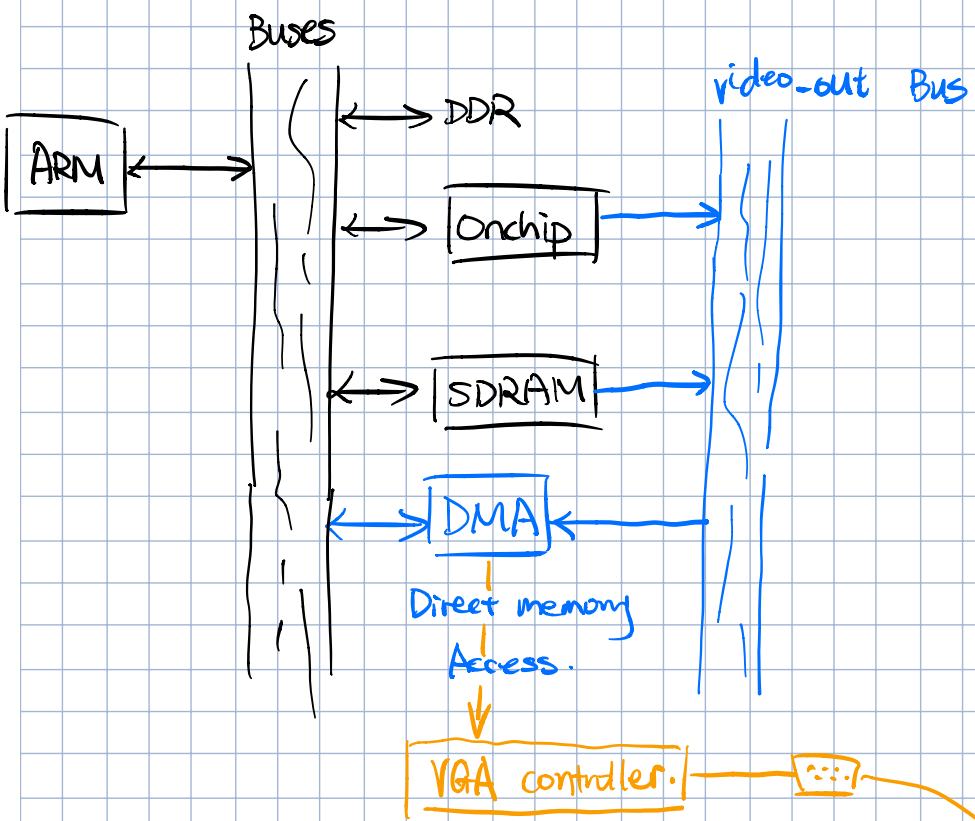
- The DE1-SoC computer has a VGA output port. It continuously reads pixel colour from a memory (called a pixel buffer) and displays them on the VGA screen. The image in memory is 320 x 240 pixels. We use two different memory locations for the buffer: On chip (on the FPGA chip), and SDRAM.



- your ARM code & data are in this 1GB DDR3 memory.
- but, the video_out port uses either the on-chip memory (256kB) or the SDRAM CHIP (64MB)

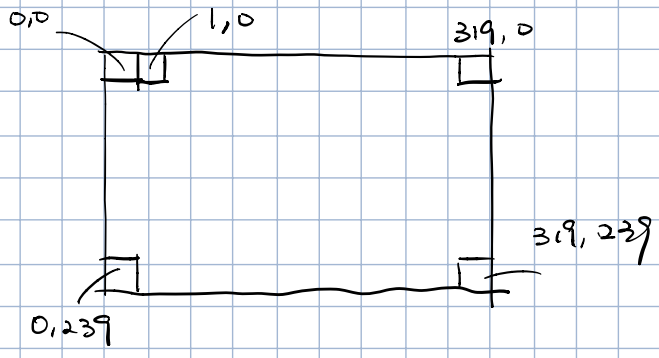
Onchip memory: C8000000 - C803FFFF

SDRAM: C0000000 - C3FFFFFF

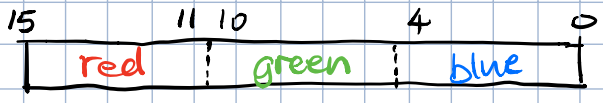


your code (in DDR3) running on ARM stores an image in a pixel buffer. (onchip or SDRAM)

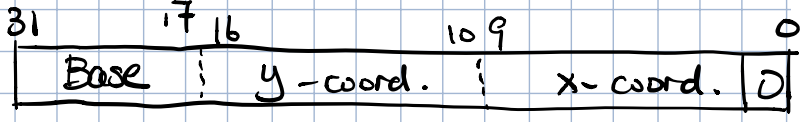
The DMA controller continuosly reads pixels of the image and sends to the VGA controller.



- Each pixel takes 2 bytes in memory:



- the address of pixel (x,y) is:



$\therefore$ address = base + (y << 10) + (x << 1) ;

- The default pixel buffer location is in the on-chip memory

0,0 : C8000000

1,0 : C8000000 + (0 << 10) + (1 << 1);

= C8000002

⋮

0,1 : C8000000 + (1 << 10) + (0 << 1);

= C8000400

1,1 : C8000402

⋮

319,239 : C8000000 + (239 << 10) + (319 << 1);

= C803BE7E

- Summary: to make the bottom-right pixel full green you would write the 16-bit value 0000011111100000 = $(07E0)_{16}$ to address C803BE7E

Part 1: draw a line

- we can't draw an exact line, we can only colour pixel close to the line:

Brensenham's Algorithm:

$\Delta x = 12 - 1 = 11$

$\Delta y = 5 - 1 = 4$

$error = -\frac{\Delta x}{2} = -\frac{11}{2} = -5$