

## Simpler VGA memory access (memory mapped VGA)

**How would you like to access the VGA framebuffer more directly with a single sw/lw (or array access in C) to write/read each pixel, instead of two separate steps of writing the address and then reading/writing the data?**

The sample otter\_wrapper on PolyLearn for the 80x60 graphics VGA module, currently maps the VGA address port to one memory address, the VGA data read port to another single memory address, and the VGA data write port to yet another single memory address. This makes the draw\_dot sample code a bit more complex than it has to be.

With a few changes to just the wrapper, that can be simplified, so a single load/store both sets the address and performs the read/write:

**New wrapper:**

[OTTER Wrapper VGA80x60.sv](#)

**Differences from the old wrapper to make the changes easier to merge with your own custom wrapper:**

<https://www.diffchecker.com/I0flJYZs>

**Differences in usage:**

In assembly, this modification saves one register, and makes the draw\_dot function more intuitive:

<https://www.diffchecker.com/5elqJyXH>

In C however, this modification eliminates the need for a draw\_dot function, as the range-mapping method is directly compatible with the C array convention.

**For fun, here's a sample program written in C that demonstrates reading/writing the VGA buffer as a simple array (using the new range mapping in the above wrapper) to run Conway's Game of Life:**

[simple.c](#) (use the latest "OTTER Tools" on PolyLearn and replace the main.c with this simple.c)

[otter\\_memory.mem](#) (or just copy this mem.txt file into Vivado if you want to skip the compile)

It first fills the screen with randomly chosen white or black cells, with the random number generator seeded by the switches. Then it repeatedly updates the screen according to the rules of the game. It also updates the leds to the switch positions and increments the 7-seg to demonstrate/verify use of the other hardware.

To see a different scenario, change the switches and push the reset (center) button to restart with the new seed. 1000000001111111 generates a cool result, shown in the video below:

**Video of the Game of Life demo using the updated wrapper:**

<https://youtu.be/iWTJPUnT38g>

Revisions:

Version 0.1 - (Keefe Johnson) - Created and tested modules - 6/1/2019

Version 0.12 - (Joseph Callenes) Updated doc - 9/13/2019