Hardware Accelerated Graphics

Group #6

Stephen Just - Stefan Martynkiw - Mason Strong

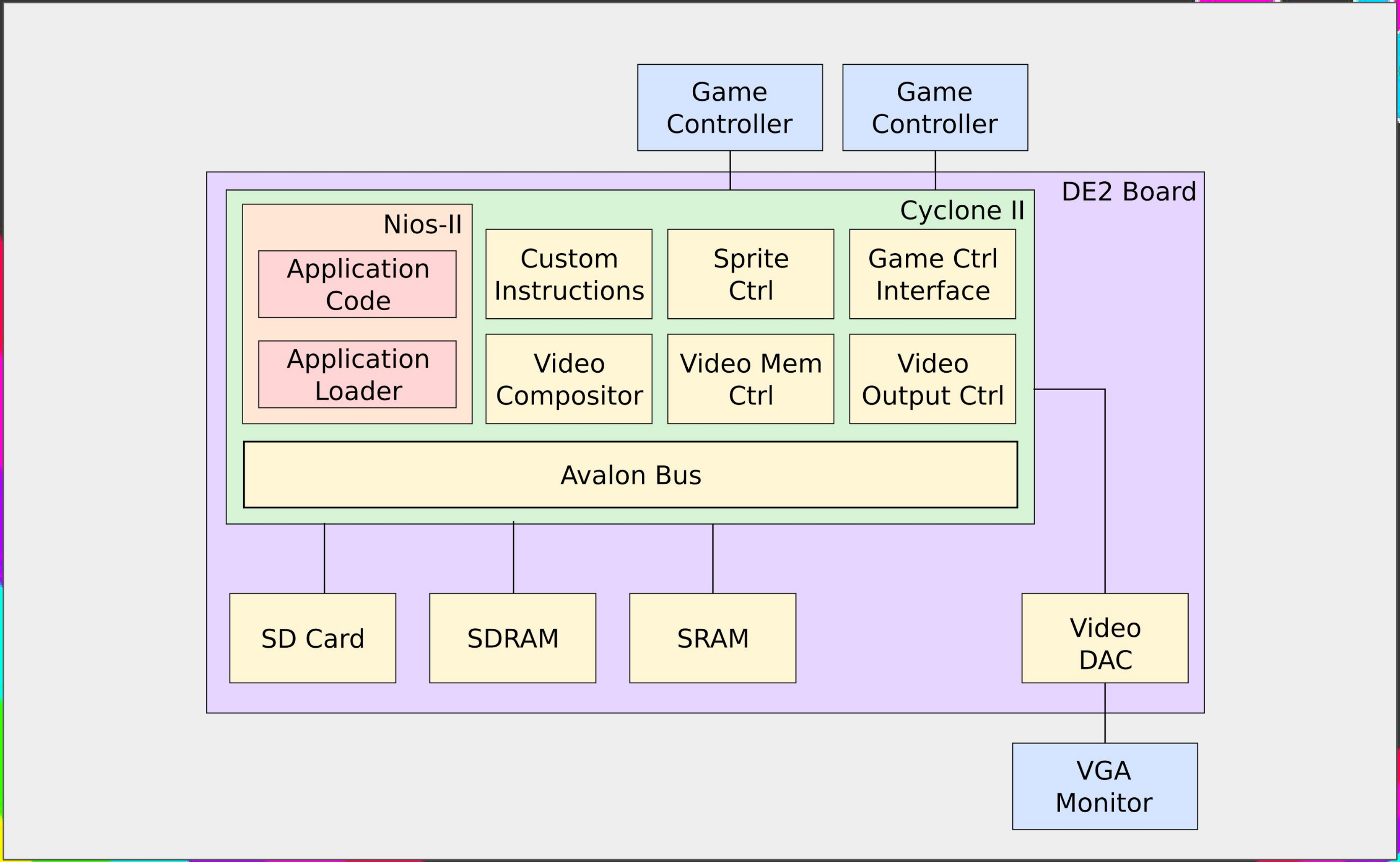
Functionality

- Provide easy-to-use graphics support for C programmers on a NIOS-II based system
- Provide a simple input device module for game controllers
- Enable the development of simple video-based applications
- Provide a simple demo application

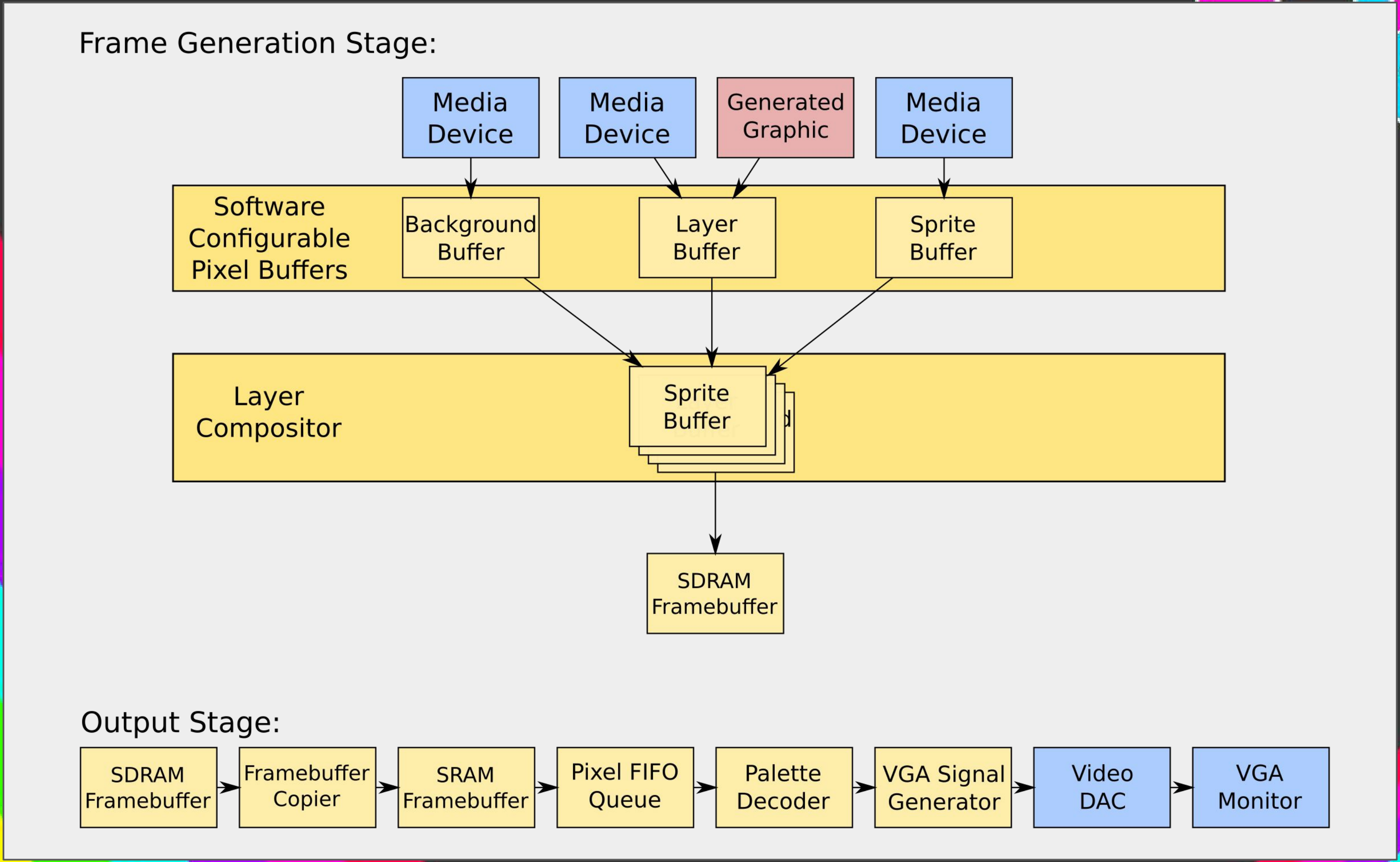
Motivation

- Video is consistently described as a weakness of the DE2
- Potential to use the platform to create a variety of games
- Potentially reusable on future platforms (DE1-SoC, etc)

Design Overview

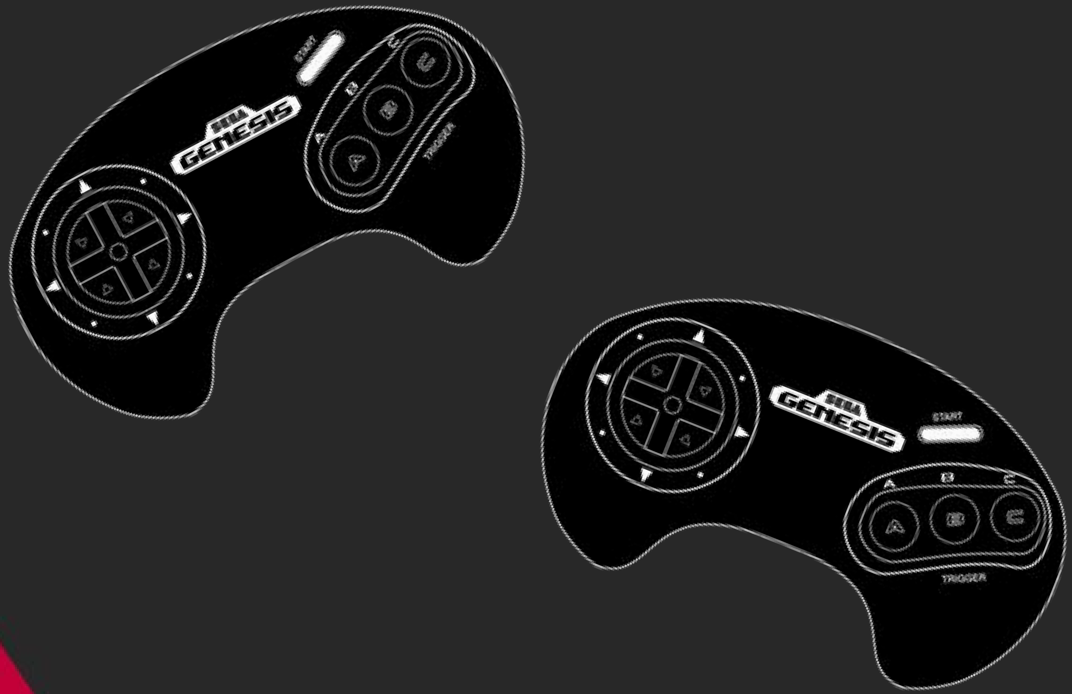


Video Pipeline



Input Devices

- Two SEGA Genesis controllers via GPIO
- Possible utilization of onboard switches

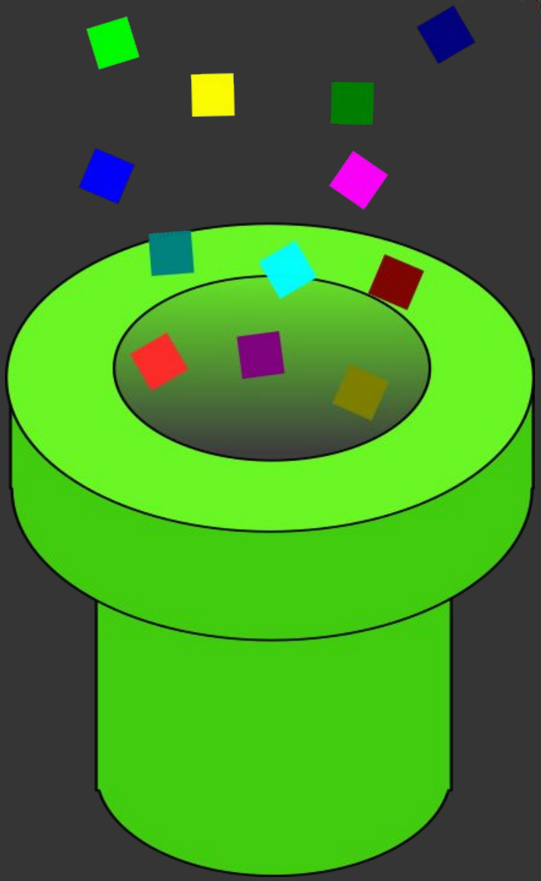


Challenges

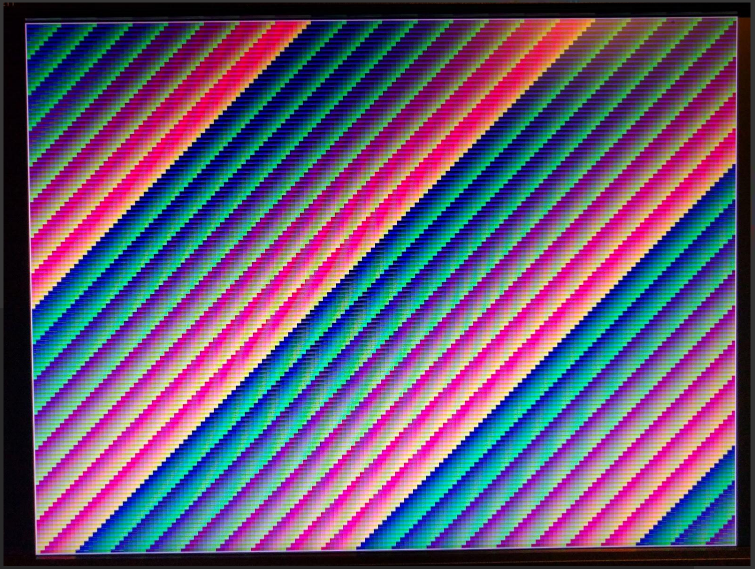
- Frame generation features cannot be easily validated until output pipeline is complete
- Copying frame data from SDRAM to SRAM without introducing tearing
- Implementing most operations in hardware because using NIOS for video operations is far too slow
- Most operations are not parallelizable due to the DE2’s memory architecture

Using the FPGA to Our Advantage

- Fully hardware accelerated pixel pipeline
- Avoid pushing pixels with software
- Make extensive use of DMA



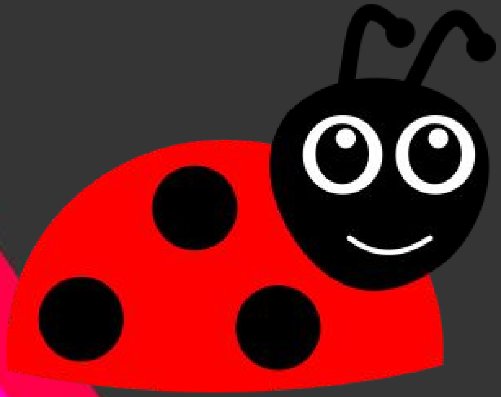
Code Examples



```
int main()
{
    int row, col, color = 0;

    for (row = 0; row < 480; row++)
    {
            for (col = 0; col < 640; col = col + 4)
            {
                    color = ((row + col) % 256) | ((row + col) % 256) << 8 | ((row + col) % 256) << 16 | ((row + col) % 256) << 24;
                    if (row == 0 || row == FRAME_HEIGHT - 1)
                    {
                            IOWR_32DIRECT(SRAM_0_BASE, row * FRAME_WIDTH + col, 0xFFFFFFFF);
                    }
                    else if (col == 0)
                    {
                            IOWR_32DIRECT(SRAM_0_BASE, row * FRAME_WIDTH + col, 0x000000FF | color);
                    }
                    else if (col == FRAME_WIDTH - 4)
                    {
                            IOWR_32DIRECT(SRAM_0_BASE, row * FRAME_WIDTH + col, 0xFF000000 | color);
                    }
                    else
                    {
                            IOWR_32DIRECT(SRAM_0_BASE, row * FRAME_WIDTH + col, color);
                    }
            }
    }
    return 0;
}
```

Test Plan



Unit Testing

- Write a test to verify each component as it is completed
- Most test cases will consist of C applications

System Testing

- Write a demo application to show off our system

Application Notes

- We expect to produce a large quantity of application notes due to the nature of the project
- Examples:
  - Game Controller Input
  - Using our Video Pipeline (updated throughout term)

Add/Remove Features

- “Smart” Layers
  - Pannable backgrounds
  - Sprites with inertia
- Video Input as a Background Layer
- Sound Support
- Loading code from an SD Card

Thank you!