

The background features a series of parallel diagonal stripes in a rainbow spectrum (red, orange, yellow, green, cyan, blue, purple, magenta) that run from the top-left towards the bottom-right. The stripes are set against a dark grey background.

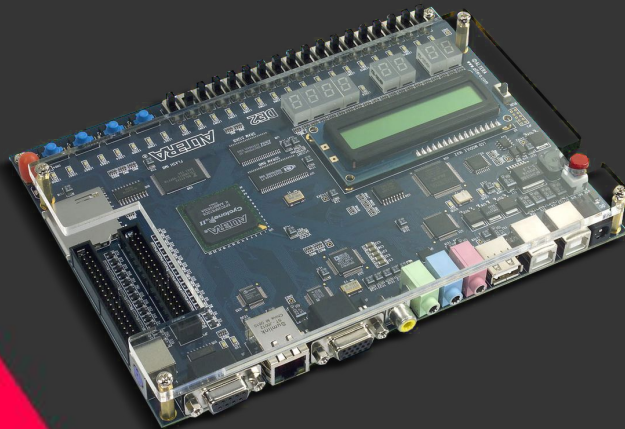
Hardware Accelerated Graphics

Group #6

Stephen Just - Stefan Martynkiw - Mason Strong

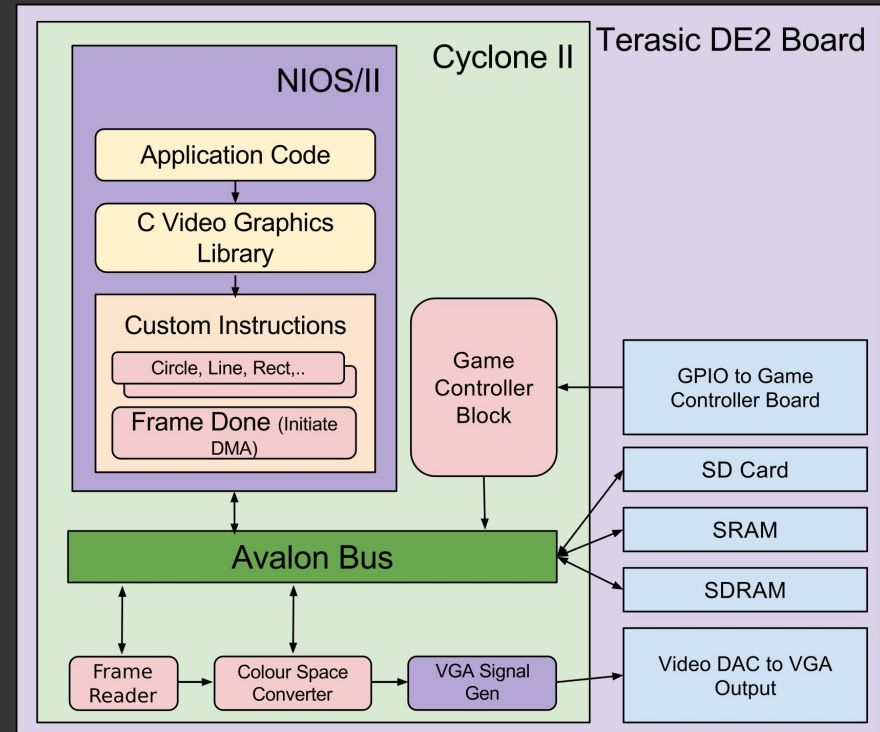
Purpose

- Build a platform capable of providing high-speed graphics support to a variety of applications
- Make use of FPGA (hardware) to speed up drawing operations
- Provide simple interface for programmers to use the platform's features



Features

- 640x480 resolution, 60 fps
- Hardware-drawn shapes
 - Lines
 - Circles
 - Rectangles
- Colour palette support (256 colour palette → 16-bit RGB)
- Bitmap (image) drawing
- Multiple layer compositing
- Input from game controllers
- Reading from SD Card or Flash storage



Legend



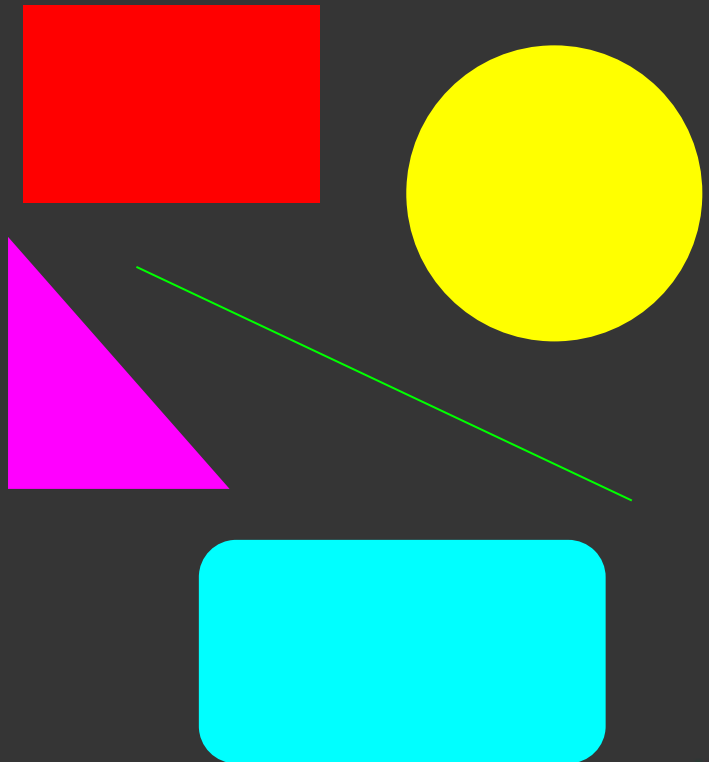
Design



- Provide collection of graphics operations as custom CPU instructions
- Separate working memory from frame being currently displayed
- Make use of colour palettes to improve visual quality

Hardware Graphical Primitives

- Rectangles
- Lines, Circles
 - Bresenham's Algorithms, no floating point
- All implemented as NIOS/II Custom Instructions
 - Blocks CPU, Variable Cycles
- Way faster than software memory writes
 - 640x480 Rectangle
 - Software: 130.69 ms
 - Hardware: 2.124 ms



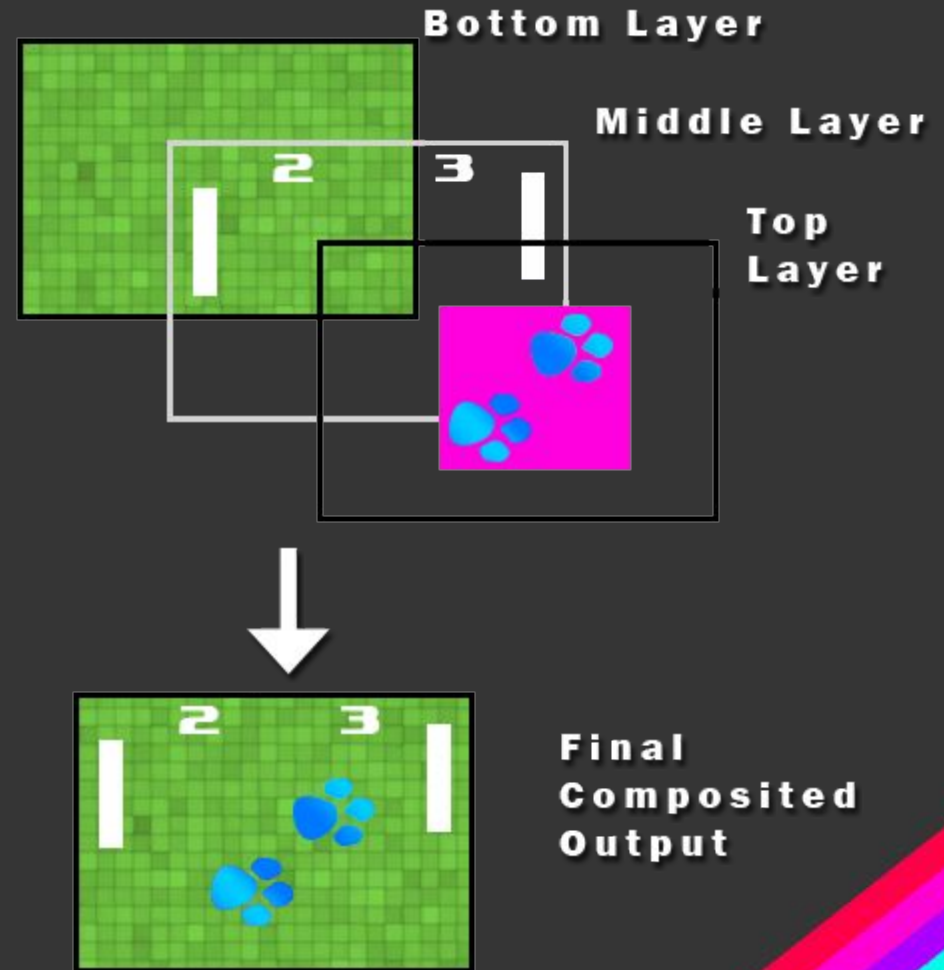
Bitmaps

- Arbitrary images can be loaded via SD Cards or Flash memory
- Simple Python utility can convert photographs into the format used by our system
- Once a bitmap is in memory, using the image is very fast
- 1-bit transparency is supported

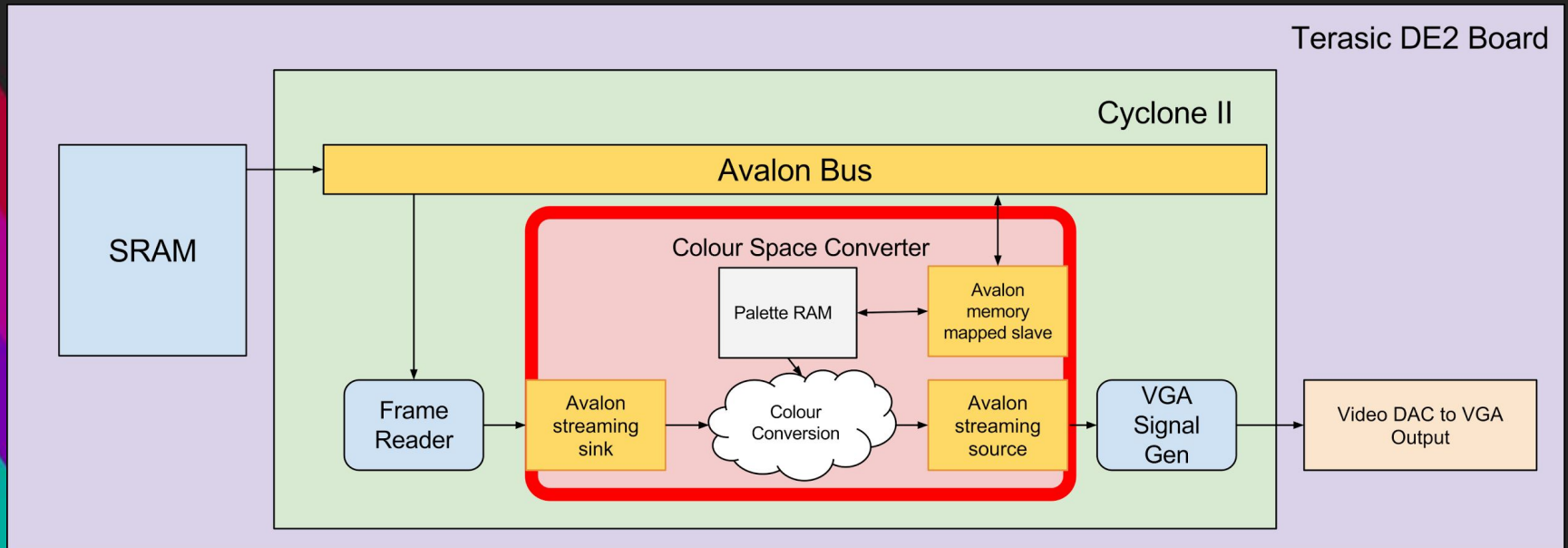


Layering

- Up to three layers supported
- Both primitives and bitmaps can be written to any given layer
- Transparency supported by omitting a given colour for a layer during compositing



Video Pipeline



Input Devices

- Two SEGA Genesis controllers via GPIO
- Used to interact with platform
- Interfaced via 7 IO lines, GND and VCC
- Easily read controller values with our API
- Custom adapter board for easy connectivity



C - API Example

```
#include <io.h>
#include <system.h>
#include <sys/alt_stdio.h>
#include "sys/alt_timestamp.h"
#include <string.h>

#include "graphics_commands.h"
#include "palettes.h"

int main(){
    graphics_init();
    graphics_clear_screen();
    switch_palette(&palette_ega);

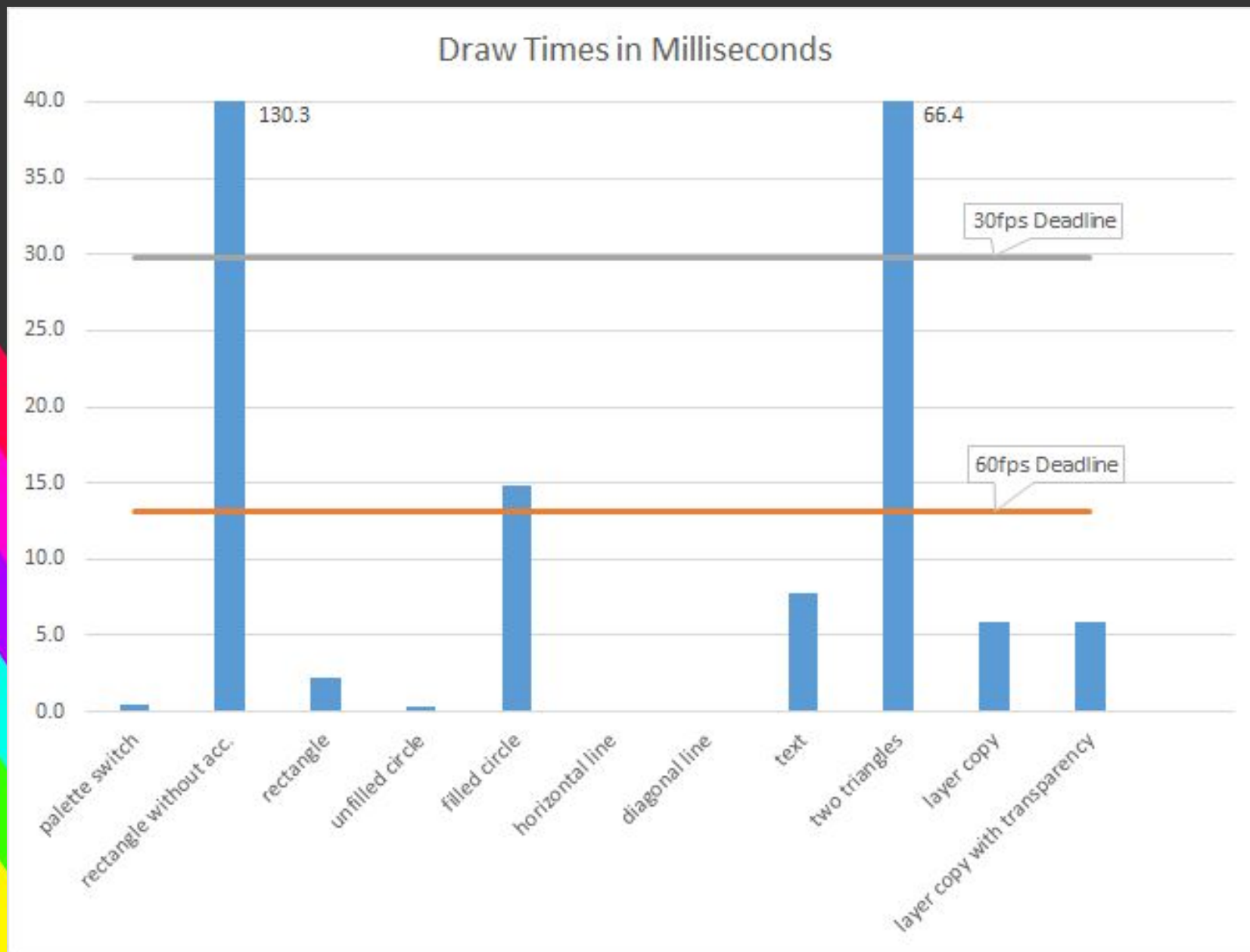
    graphics_draw_rectangle(graphics_get_final_buffer(), 0, 0, 640, 480, 15);
    ALT_CI_CI_FRAME_DONE_0; //Draw the Frame

    graphics_draw_circle(graphics_get_final_buffer(), 640/2, 480/2, 239, 3, 0);
    graphics_draw_circle(graphics_get_final_buffer(), 640/2, 480/2, 239, 4, 1);

    graphics_draw_line(graphics_get_final_buffer(), 0, 0, 640, 480, 5);
    ALT_CI_CI_FRAME_DONE_0;

    print2screen(graphics_get_final_buffer(), 20, 20, 6, 2, "Hello, World!");
    graphics_draw_triangle(graphics_get_final_buffer(), 15, 112, 300, 112, 170, 240, 1, 9);
    ALT_CI_CI_FRAME_DONE_0;
    return 0;
}
```

Performance





Demonstration

Future Work

- Creating more applications for the platform
- Audio support
- Higher Resolutions
- More advanced graphics primitives
- Overlaying graphics over a live video feed

The image features a dark gray background. In the bottom-left and bottom-right corners, there are decorative elements consisting of several parallel diagonal stripes. The stripes are colored in a vibrant, multi-colored sequence: yellow, green, cyan, magenta, and red. The stripes in the bottom-left corner extend from the bottom edge towards the top-left, while the stripes in the bottom-right corner extend from the bottom edge towards the top-right.

Thank you!