

Asteroids

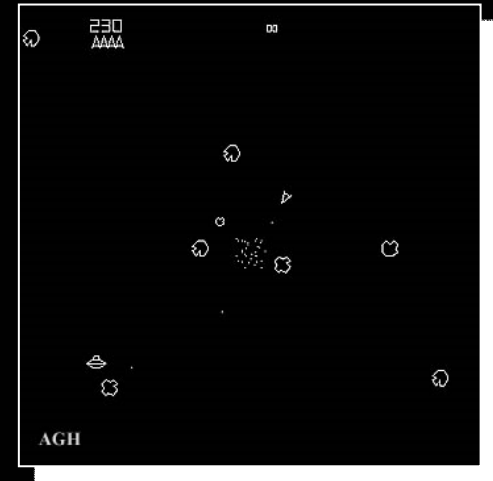
6.111 Final Project – Fall 2005

Shield Xiao & James Verrill

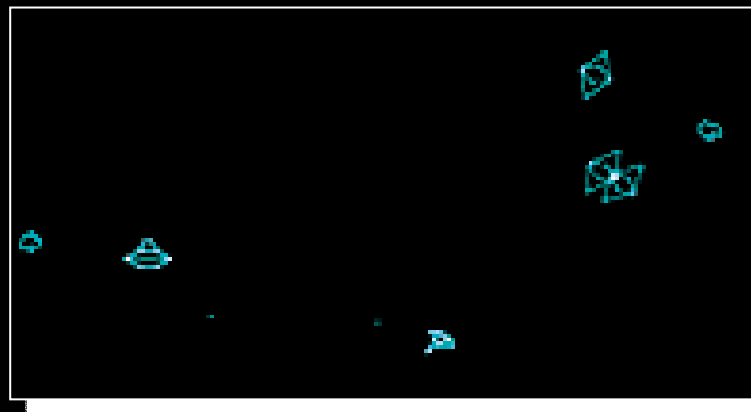
A decorative graphic in the bottom-left corner of the slide, consisting of three curved blue lines that sweep upwards and to the right, with three small blue dots placed at various points along these lines.

Introduction

- Asteroids?
 - Classic Video Game
 - First Version appeared around 1978
 - Biggest Selling Game of its time(!)
 - Features vector graphics



Asteroids Arcade (1978)



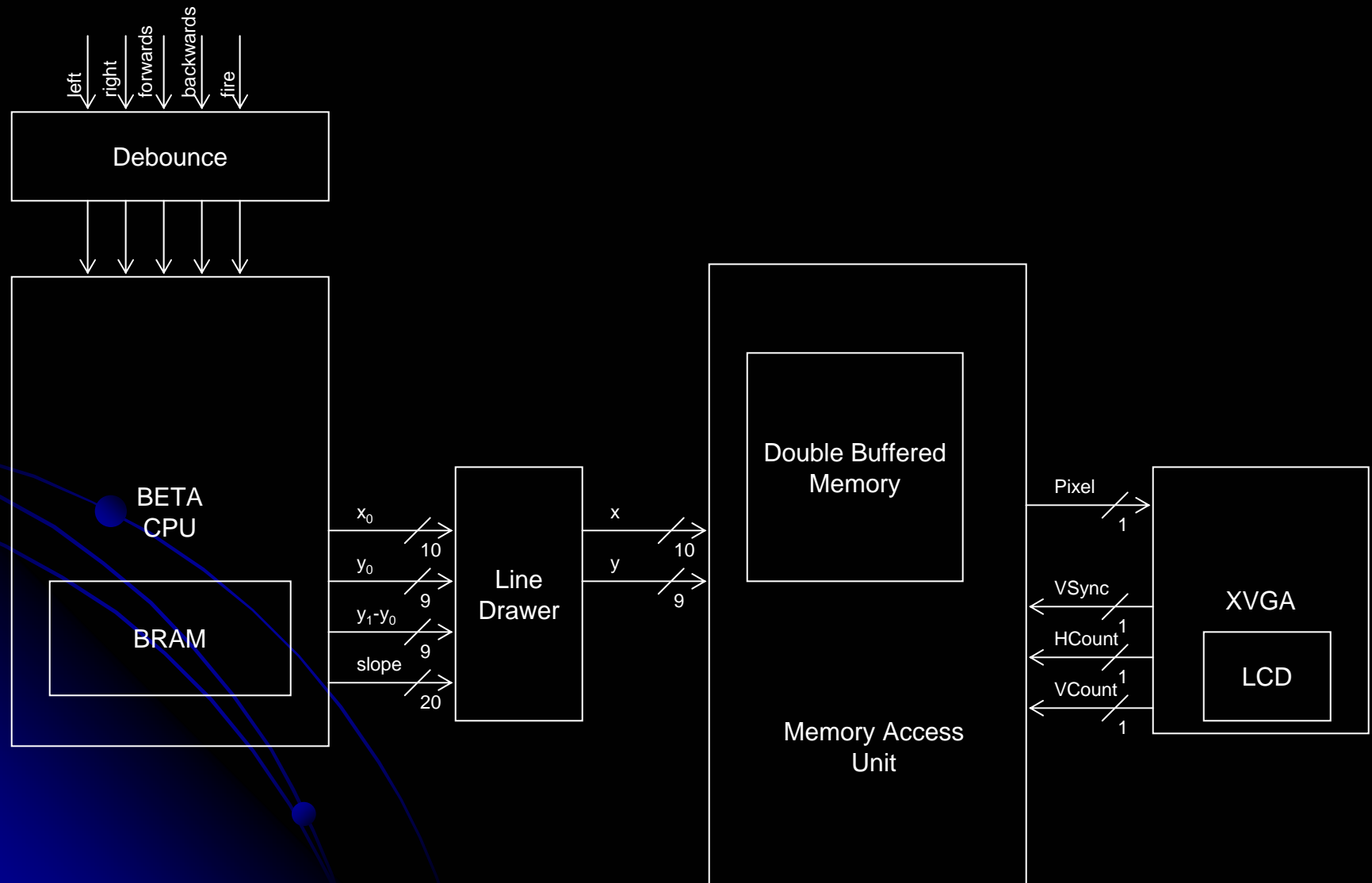
Asteroids Deluxe (1979)

Project Overview

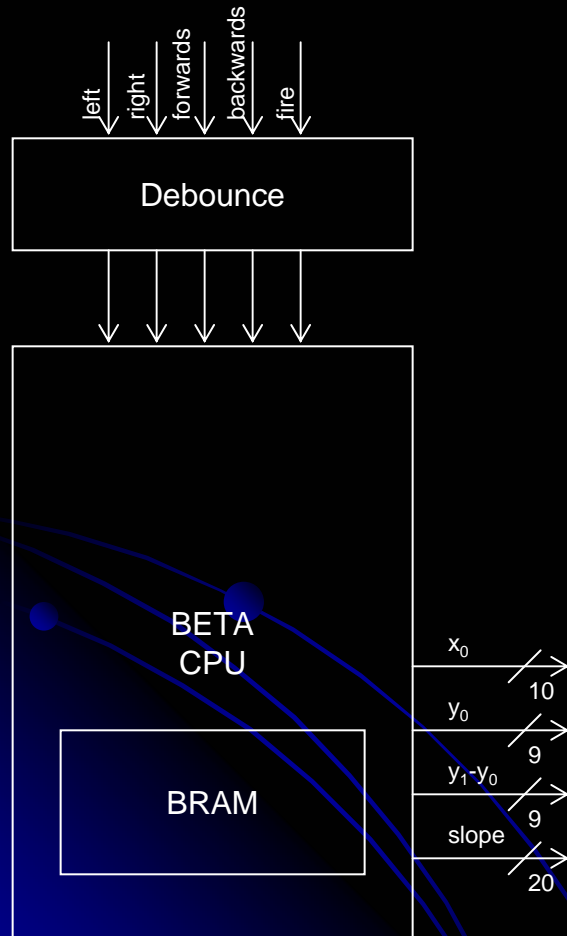
- Aims

1. Create a hardware system which draws vectors onto the screen
2. Create a Beta Processor to run programs to utilise vector drawing hardware
3. Using the above: Create a version of Asteroids
4. Add features to the basic asteroids game (e.g. sound effects, scoring)

Project Overview



CPU



- **Functions**

- Run compiled programs to generate outputs to vector drawing system
- Eventually run Asteroids game
 - Processing User Inputs
 - Moving Asteroids and Ship
 - Detecting Collisions

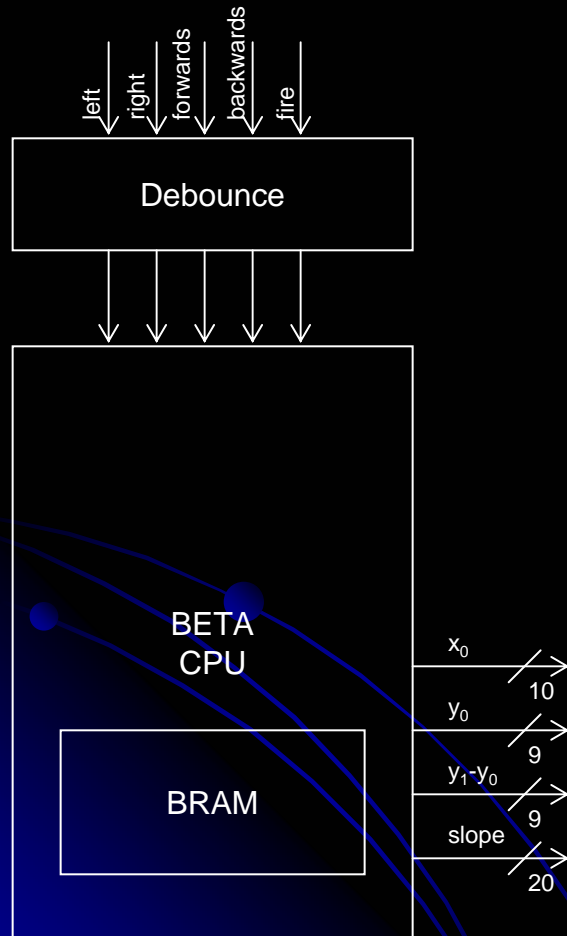
- **Implementation**

- Existing Beta Definition
 - 32 bit Multiply and 32 bit Divide

- **Outputs**

- Left Most Point on a Vector (x_0, y_0)
- Slope of Vector (as decimal)
- End y co-ordinate of Line

CPU (2)



- Considerations

- Speed

- Movement of each asteroid will require
 - 72 multiplies (rotation of co-ordinates and translation in space)
 - 16 subtracts and 8 divides (calculation of slopes)
 - Taking: 248 cycles
- Detection Collisions of each asteroid will require
 - 24 subtract and compare operations (4 per bullet and 4 per ship)
 - Taking 48 cycles
- Total: 17760 cycles per frame
- Movement of ship and bullets will be inconsequential in comparison

Line Drawer

- Function

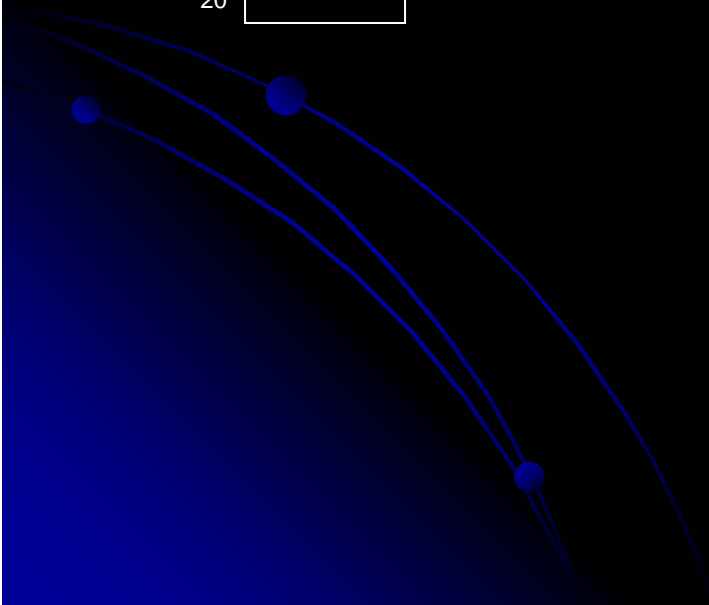
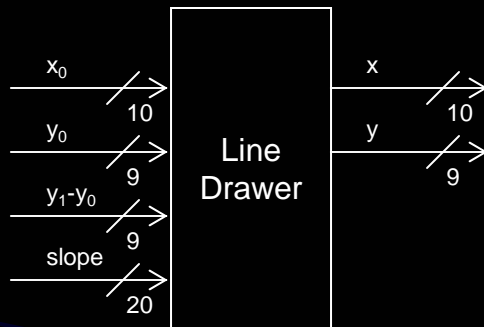
- Implements Bresenham Line Drawing
- Calculates which pixels on/off for given line

- Outputs

- Pixels that exist on given line

- Considerations

- Speed
 - Possibility of drawing ~500 complete diagonal lines the length of screen
 - Maximum 1024 cycles to output all pixels for each line
 - Maximum of 512000 cycles
 - Have 541632 cycles (running at 32.5MHz)
 - Reality: lines are much shorter than screen



Frame Generator

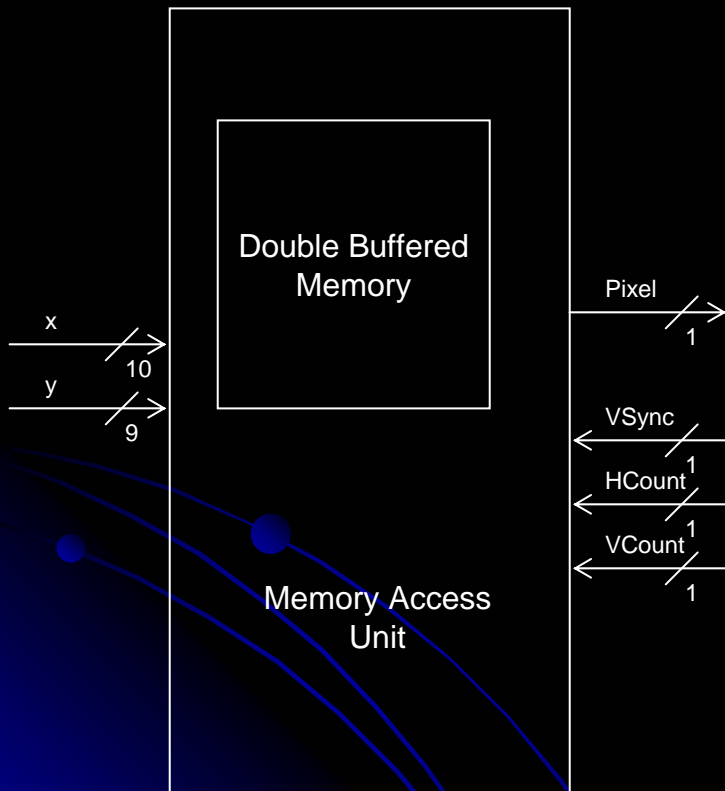
- Function

- Generate a pixel representation of the next frame based upon inputted pixel values
- Output to the VGA module whether a pixel is black or white given a demanded pixel

- Considerations

- Storage of Frames

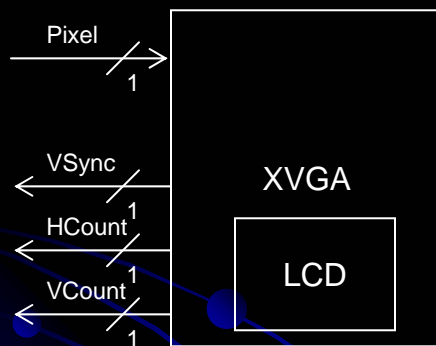
- Need ability to write to next frame and read from current frame
- Implement using Double Buffered Memory to allow simultaneous read and write operations to different memory locations
- Store next and current frames separately in memory



VGA Module

- Function

- Request information about each pixel in the screen
- Transform pixel information into appropriate signals
- Transmit to the LCD screen
- Generate Count and Sync signals to control other modules



Questions?

(not difficult ones)

A decorative graphic in the bottom-left corner of the slide. It consists of three curved, parallel lines that sweep from the left edge towards the bottom center. Three small, solid blue circles are placed along these lines, one on each line, positioned roughly in the middle of the curve's length.