

# THROTTLE COPTER

David DeTomaso

Gary Crum

Justin McDowell

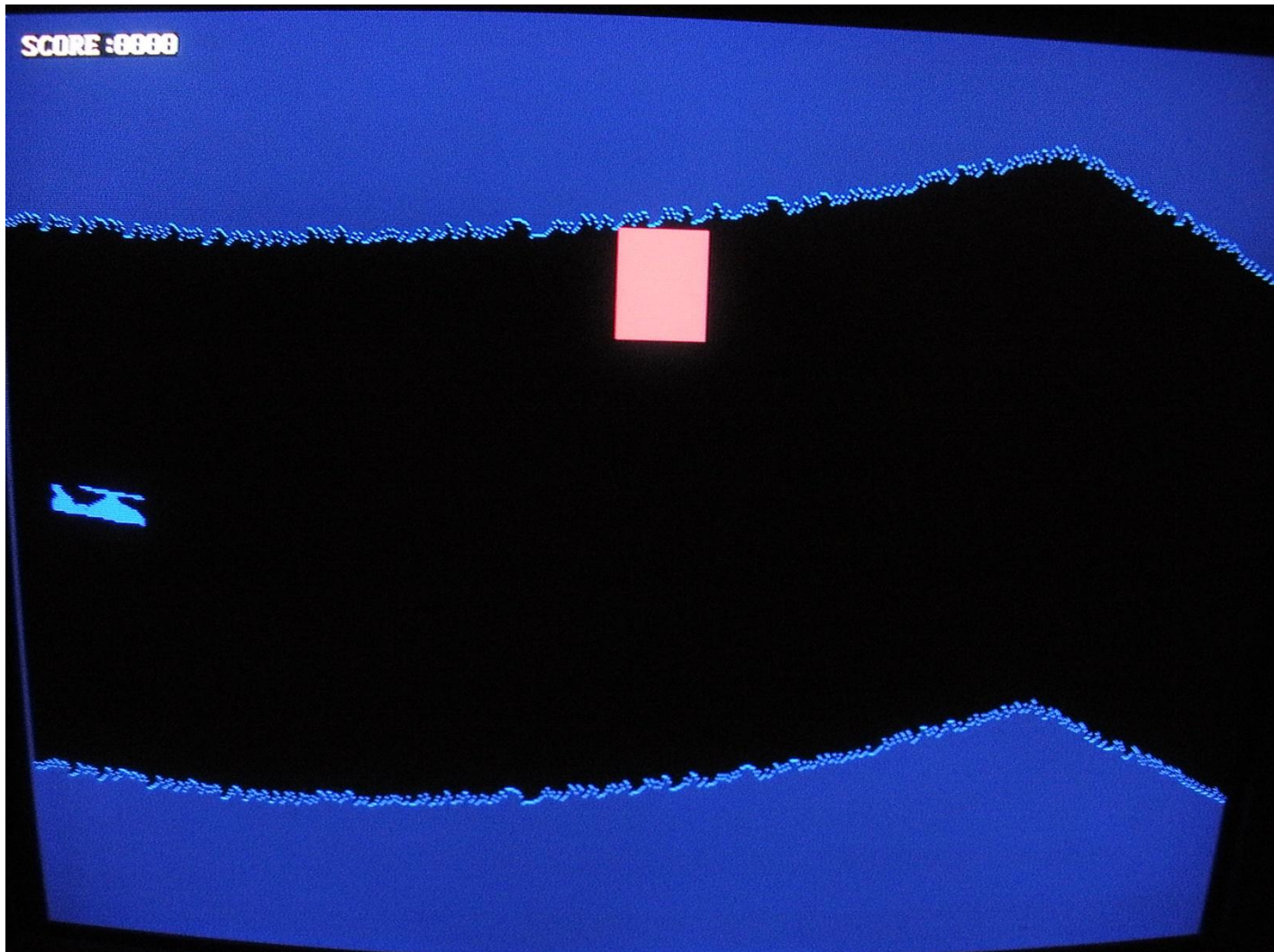
Wes Edens

# Description

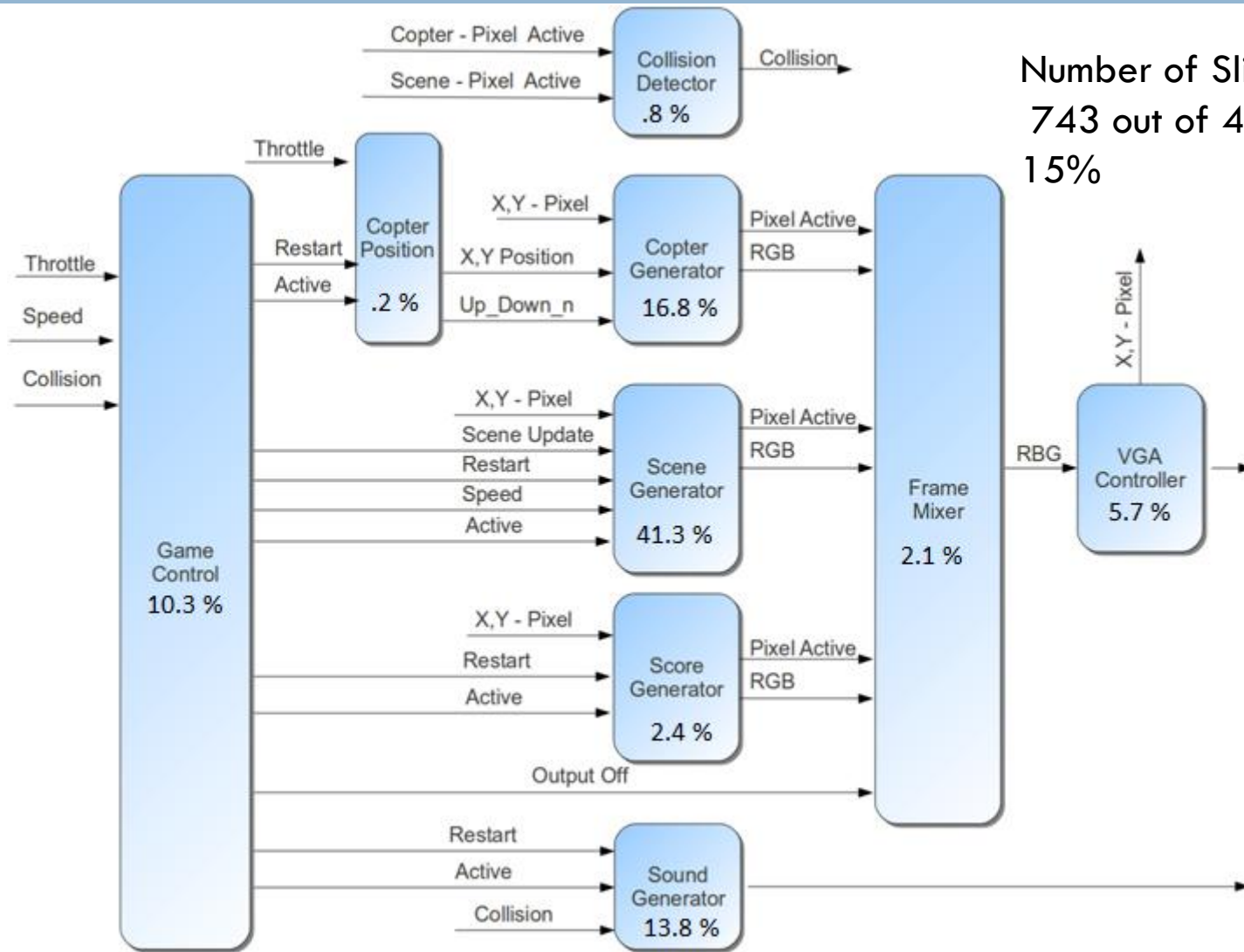
---

- Helicopter Flying Arcade Game
- Button Controls the 'flight' of the helicopter and causes it to fly higher, gravity (letting go of button) causes it to fall
- If the user hits the floor/ceiling/obstacle, game over
- Level will shrink as time goes on, making it harder
- Sound during game play
- Using VGA Monitor

# Description



# System Diagram



Number of Slices:  
743 out of 4656  
15%

# Resource Utilization

	ROMs	Adder	Counters	Registers	Comparators	Multiplexers	Xors	FSM
VGA Controller		2	3	21	8			
VGA Mixer				8				
Debounce		1		23	1			
Copter Gen	2	5		62	4	2		
Scene Gen		10		152	9		1	1
Collision Detect				3				
Game Control		1	1	38				1
Score	2	5	1	9	8	5		
Sound		2	2	51	1			1
Copter Position			1	1				
Total	4	26	8	368	31	7	1	3

# Division of Work

- David –
  - ▣ Scene Generation
- Justin –
  - ▣ Audio
- Gary –
  - ▣ Game Logic
- Wes –
  - ▣ Scorekeeping

# Score Module

- Module that paints text on the screen
- Needs a position to place the text
- Needs a way of incrementing the numeric value
- Outputs the RGB value for a given x,y pixel location in the screen

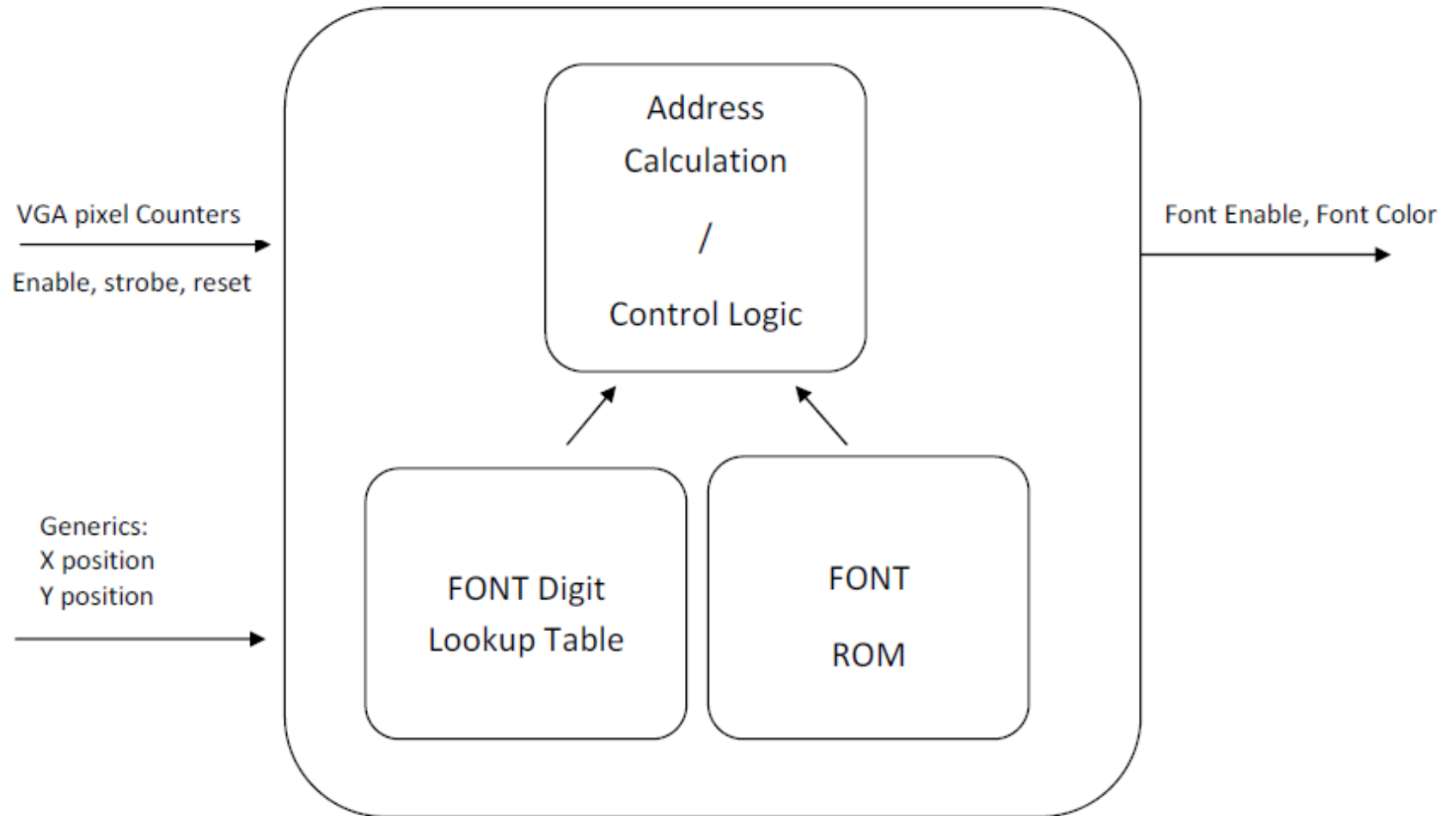
# Font Rom Creation

- Create a text file of each of the letters to be used.
- Decide on a height/width of each char
- Create a mask for each character
  - ▣ Blank space means 0
  - ▣ Any other char mean 1
- Create a postscript font file from this file (c code)
- Perl script to create the .coe memory initialization file

```
1 ++font-text-file
2 ++chars
3 22
4 ++width
5 8
6 ++height
7 12
8 ++---000-0x00-'S'-
9
10      XXXXX
11 XX      XX
12 XX
13      XX
14      XXX
15      XX
16      XX
17 XX      XX
18      XXXXX
19
20
21 ++---001-0x01-'C'-
22
23      XXXX
24 XX      XX
25 XX
26 XX
27 XX
28 XX
29 XX
30      XX  XX
31      XXXX
32
33
34 ++---002-0x02-'O'-
35
```

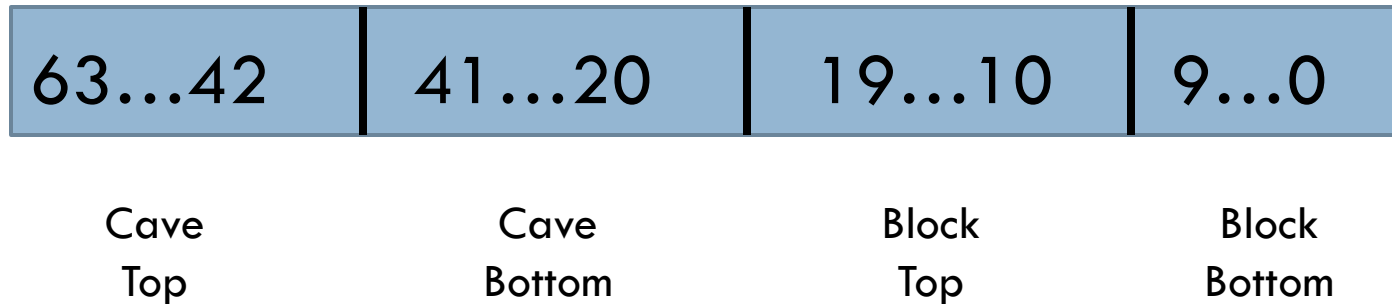


# Score Module



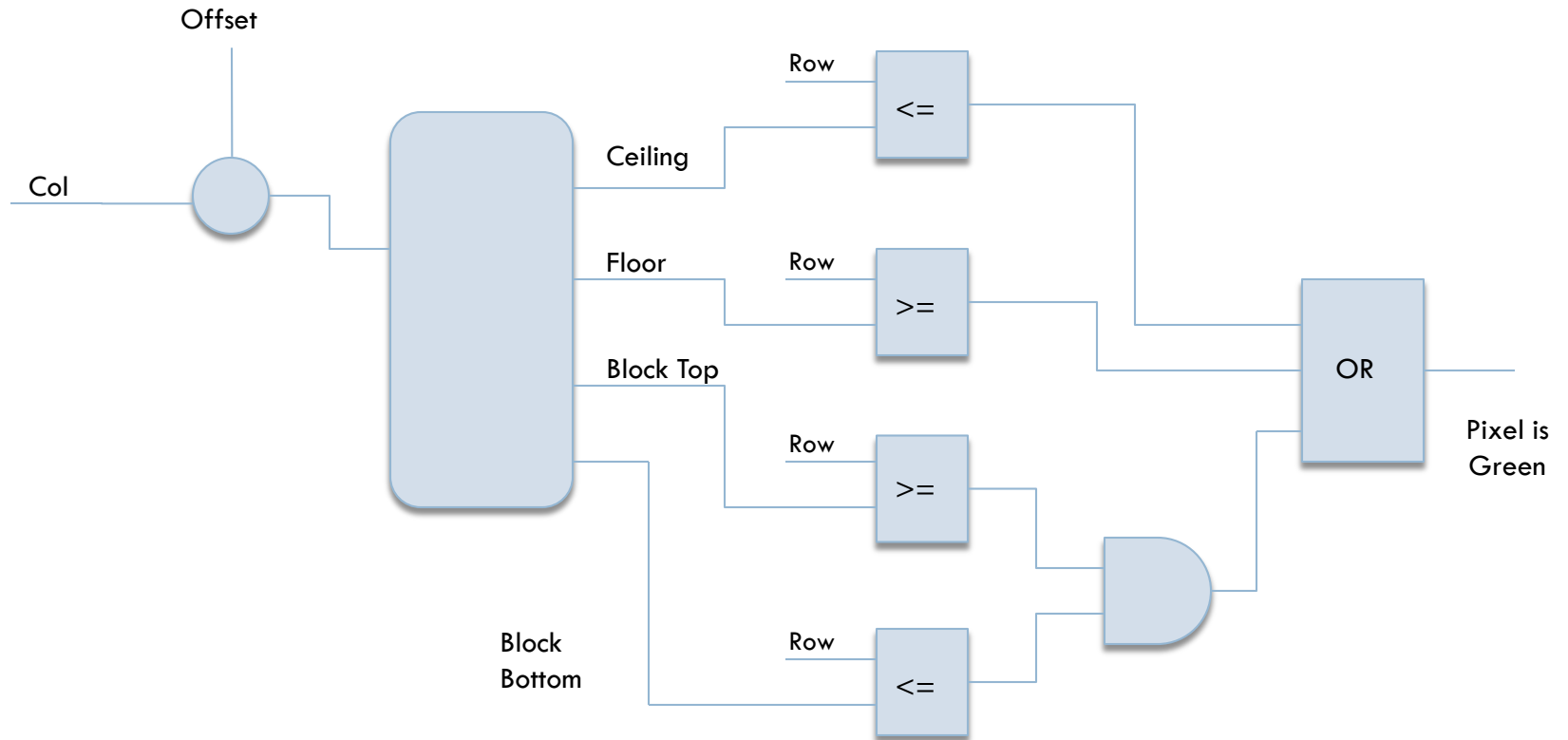
# Scene Generation - Storage

- In total this will require 64 bits per column:



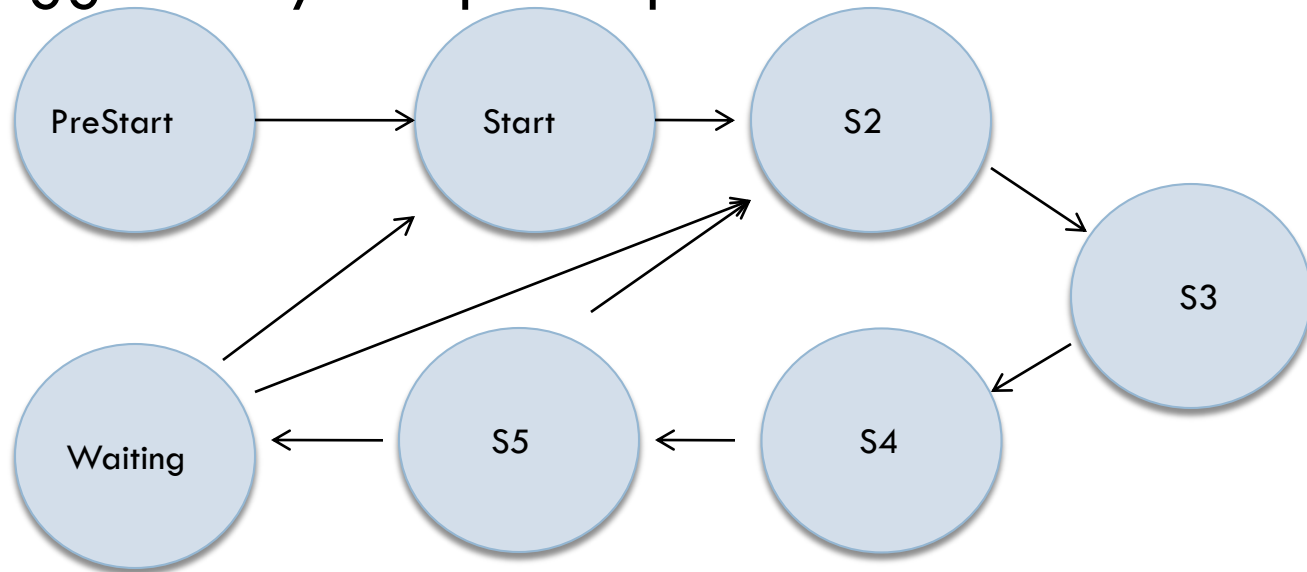
- Cave Top and Bottom have extra 12 bits of decimal precision so that arbitrary slopes can be represented in the lines that connection points on the ceiling and floor.

# Scene Generation - Readout



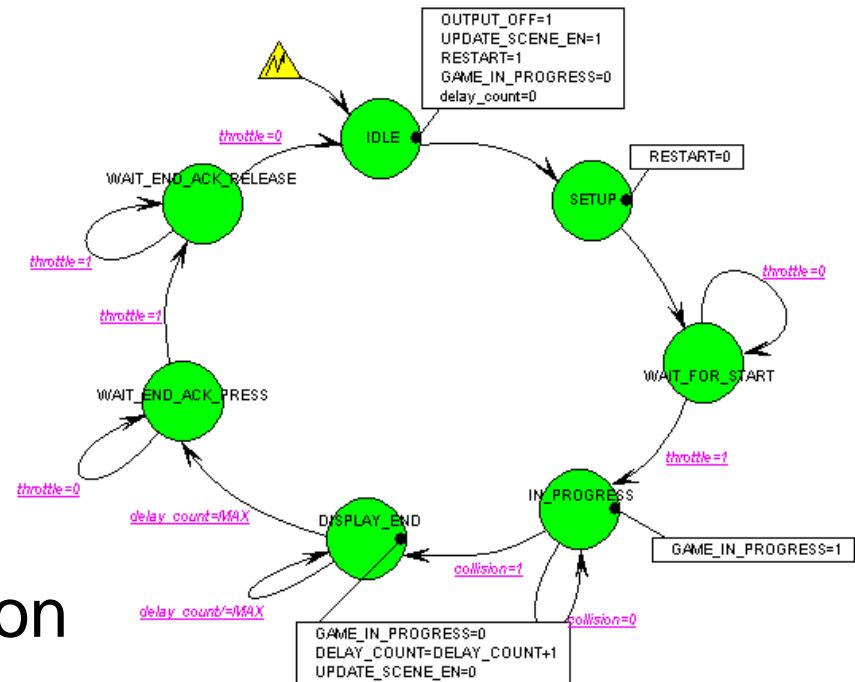
# Scene Generation - Writing

- A state machine handles the writing to block memory to add columns.
- ▣ Triggered by an update pulse between frames



# Game Control FSM

- Seven States
- Allows game setup after Reset and New Game
- Starts and Stops other Modules
- Provides 'safe' continuation for New Game



# Copter Generation

- Two Image Constants – Up and Down
- Process Detects when Pixel is in Copter 'Box'

```

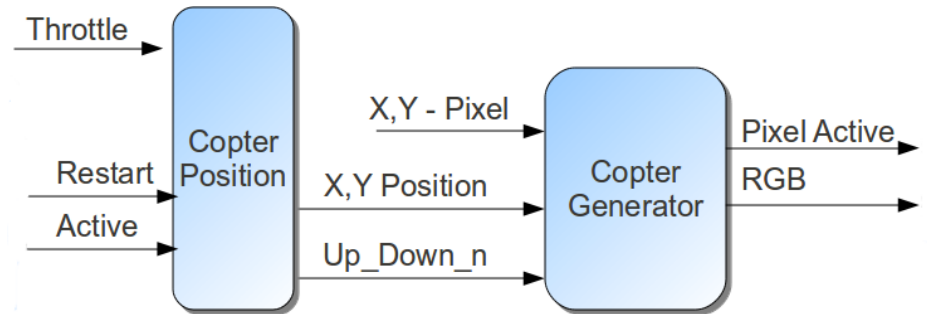
CONSTANT COPTER_MAX_WIDTH : INTEGER := 50;
CONSTANT COPTER_MAX_HEIGHT : INTEGER := 18;
subtype row_pixels is STD_LOGIC_VECTOR (0 to COPTER_MAX_WIDTH-1);
type image is array(0 to COPTER_MAX_HEIGHT) of row_pixels;

```

```
CONSTANT copter_down : image :=
```

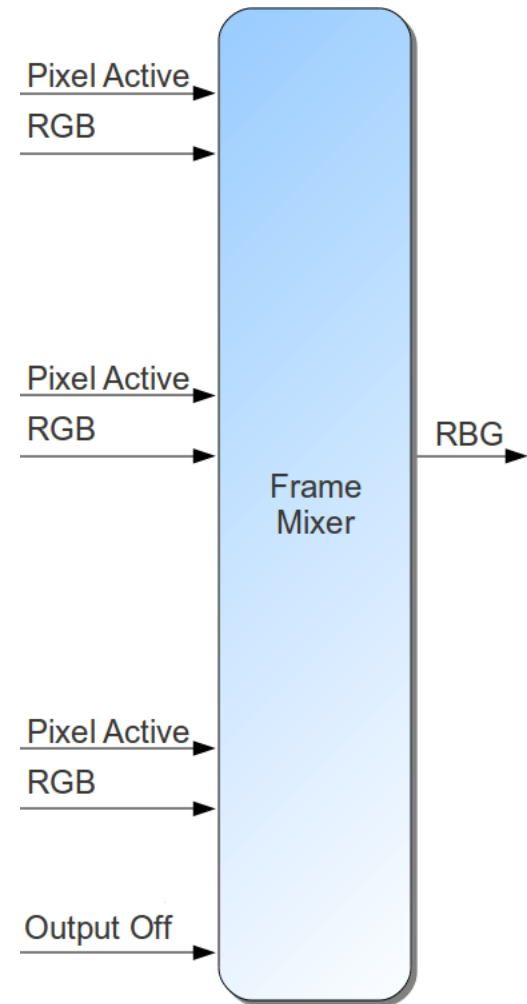
[illegible]

);



# VGA Mixer

- Layers Graphics:
  - 0 – Implied Background
  - 1 – Scene/Obstacle
  - 2 – Copter
  - 3 - Score



# Sound and LED's

---

## □ Features

- Jaws theme music during game play
- Scrolling LED's during game play
- “Game Over” sound triggered by crash
- All LED's flashing at end of game



# Sound and LED's

---

## □ Components

- Digital Clock Manager
- Digital to Analog Converter
- Sine Lookup Table
- Lowpass Filter

# Sound and LED's

## □ Utilization

- One of four DCM's (25%)
- 13 of 232 IOBs (5%)
- 2 of 24 GCLKs (8%)
- One six state FSM

# Questions

---

