

FPGA Cat Invaders

Spring 2024

Adam Jablonski (asj2172)

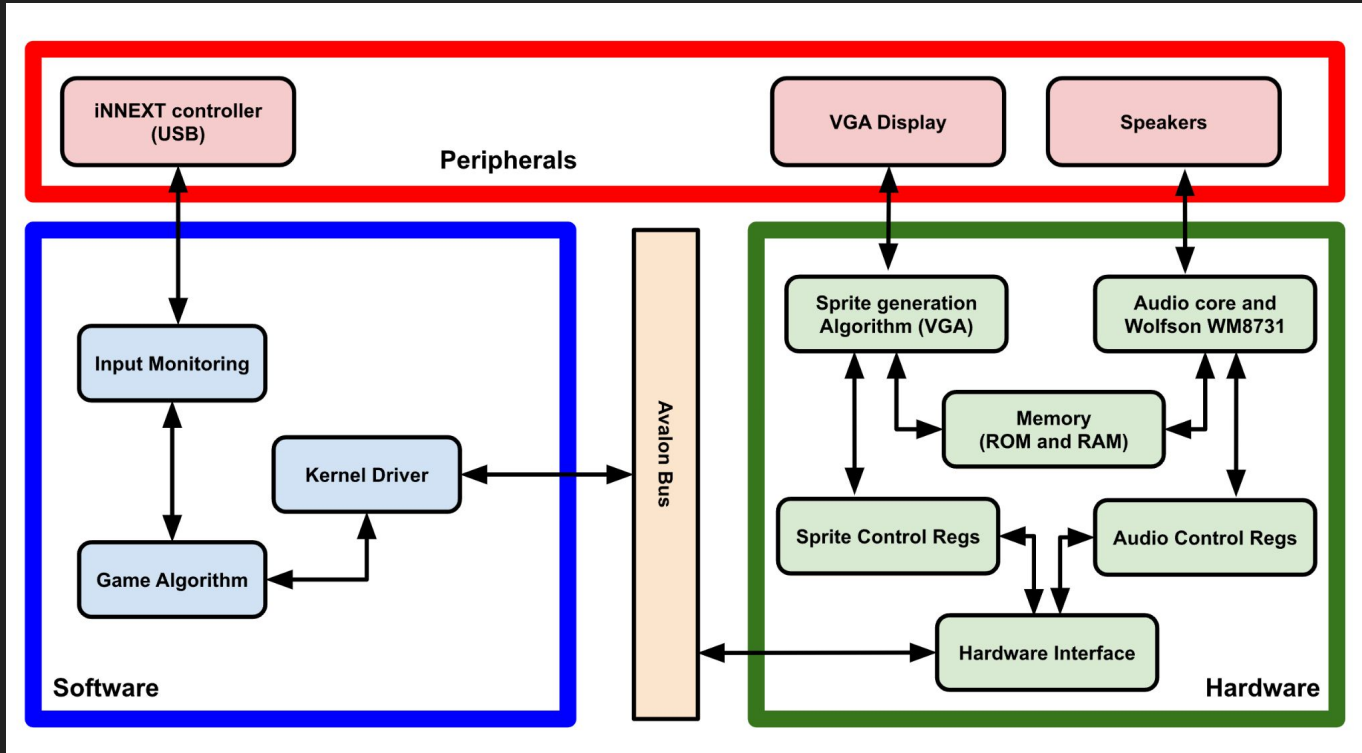
Yilei Meng (ym2974)

Fernandos Magee Jr. (fm2756)

Yuxiang Xia (yx2821)

Zhili Yan(zy2593)

Overview



Layout

0 1 2 3 4 5 6 7 8 9

GameOver...

Start

Level:

Score:



Memory layout

Cat 1,2,3,4 poses	Address range
Pose 1	0x0000 - 0x03FF
Pose 2	0x0400 - 0x07FF

Barrier state	Address range
Barrier 1	0x0000 - 0x00FF
Barrier 2	0x0100 - 0x01FF
Barrier 3	0x0200 - 0x02FF
Barrier 4	0x0300 - 0x03FF

Mystery ROM	address range
Pose 1	0x0000 - 0x08FF
Pose 2	0x0900 - 0x11FF

Poodle Pose	Address range
Pose 1	0x0000 - 0x03FF
Pose 2	0x0400 - 0x07FF
Pose 3	0x0800 - 0x0BFF
Pose 4	0x0C00 - 0x0FFF
Pose 5	0x1000 - 0x13FF
Pose 6	0x1400 - 0x17FF
Pose 7	0x1800 - 0x1BFF
Pose 8	0x1C00 - 0x1FFF
Pose 9	0x2000 - 0x23FF
Pose 10	0x2400 - 0x27FF

Memory layout (cont.)

Number	Address range
0	0x0000 - 0x03FF
1	0x0400 - 0x07FF
2	0x0800 - 0x0BFF
3	0x0C00 - 0x0FFF
4	0x1000 - 0x13FF
5	0x1400 - 0x17FF
6	0x1800 - 0x1BFF
7	0x1C00 - 0x1FFF
8	0x2000 - 0x23FF
9	0x2400 - 0x27FF

Other	Address range
explode	0x0000 - 0x03FF
level	0x0400 - 0x0FFF
Life	0x1000 - 0x13FF
mouse	0x1400 - 0x147F
score	0x1480 - 0x207F
start	0x2080 - 0x3B7F

Bone pose	address range
Pose 1	0x0000 - 0x00C0

Memory layout (cont.)

Audio Clip	Address range
Background Music	0x0000 - 0x8B3C

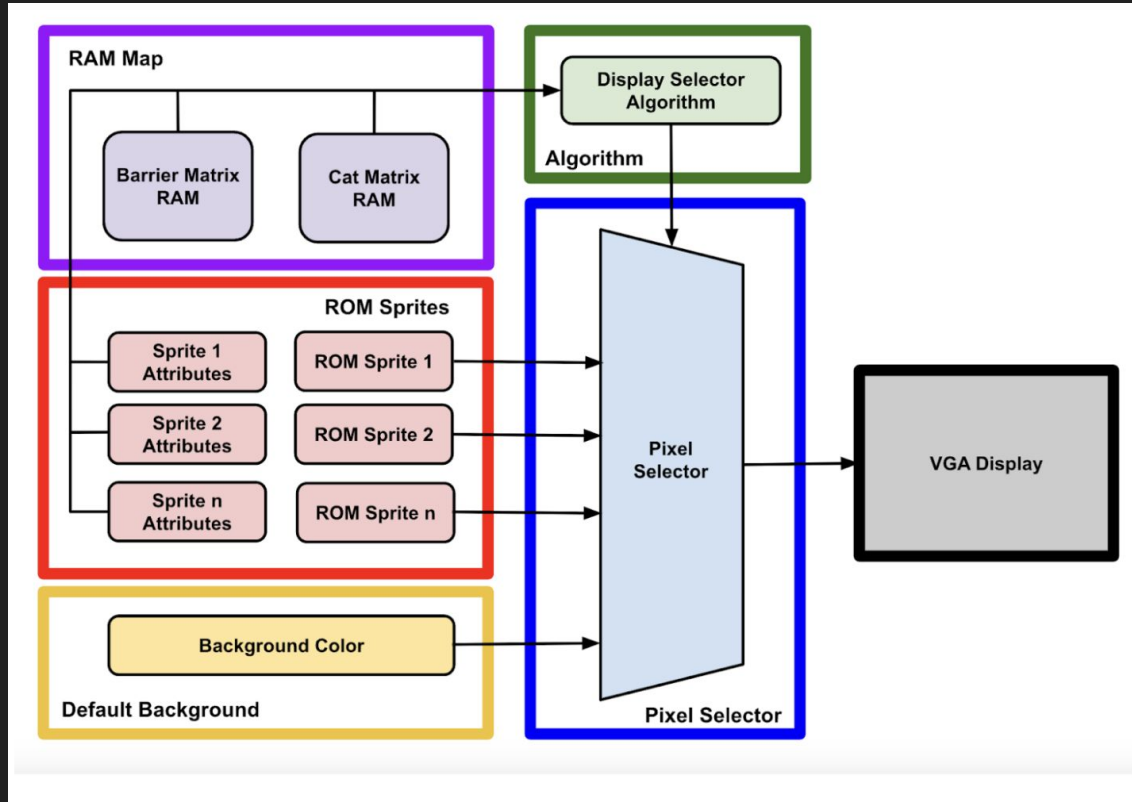
Audio Clip	Address range
Dog Bark	0x0000 - 0x1001

Audio Clip	Address range
Cat meow	0x0000 - 0x1822

	Address range
Barrier Matrix	0x0000 - 0x0018

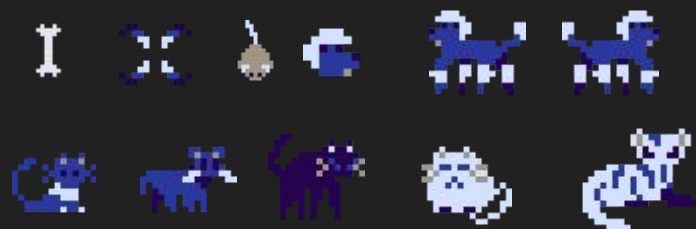
	Address range
Cat Matrix	0x0000 - 0x0037

Video Output



Hardware Display

Movable items



Static items

0 1 2 3 4 5 6 7 8 9

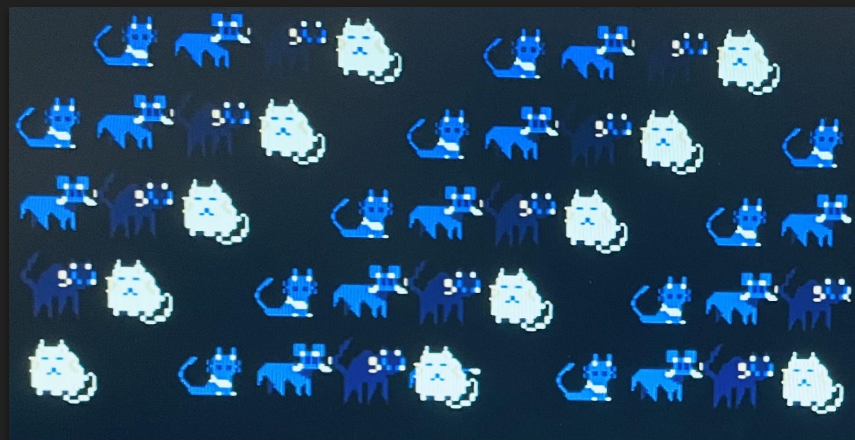
Level:

Score:

GameOver...

Start

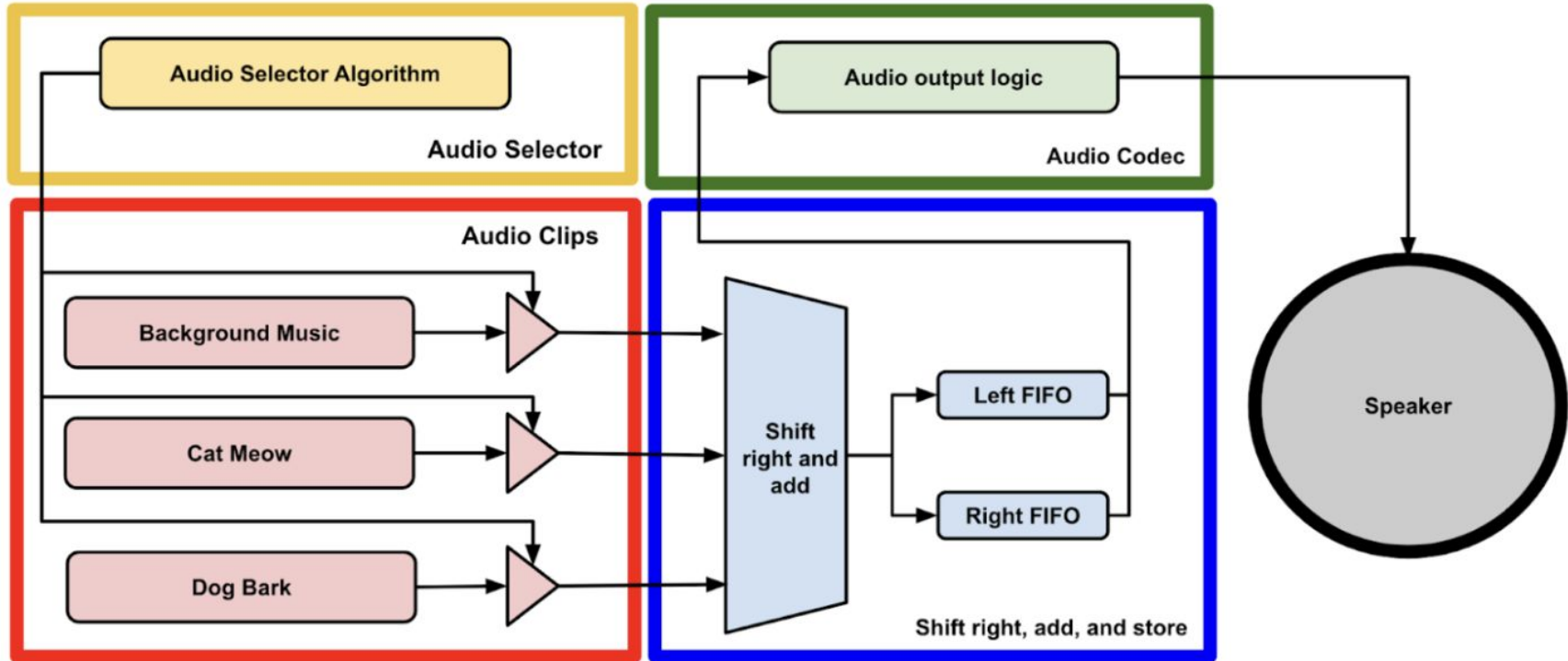
Cat Matrix



Display Priority

1	Start	8	Explosion
2	Gameover	9	Barriers
3	Bone	10	Level
4	Mouse1/2/3	11	Level number
5	Cat Matrix	12	Score
6	Mystery	13	Score number
7	Poodle	14	Lives

Audio Output



Control Registers

Table 3: Register contains - Dog position, and projectile position and visibility.

Register #	Register Name	R/W	Bit description	
			[15:0]	
0	dog_pos_x	R/W	Reserved [5:0]	Dog Position X [9:0]
1	project_d_x	R/W	Reserved [5:0]	Projectile Dog X [9:0]
2	project_d_y	R/W	Dog Bullet visible? Reserved [4:0]	Projectile Dog Y [9:0]
3	project_c0_x	R/W	Reserved [5:0]	Projectile Cat0 X [9:0]
4	project_c0_y	R/W	Cat0 Bullet visible? Reserved [4:0]	Projectile Cat0 Y [9:0]
5	project_c1_x	R/W	Reserved [5:0]	Projectile Cat1 X [9:0]
6	project_c1_y	R/W	Cat1 Bullet visible? Reserved [4:0]	Projectile Cat1 Y [9:0]
7	project_c2_x	R/W	Reserved [5:0]	Projectile Cat2 X [9:0]
8	project_c2_y	R/W	Cat2 Bullet visible? Reserved [4:0]	Projectile Cat2 Y [9:0]
9	project_c3_x	R/W	Reserved [5:0]	Projectile Cat3 X [9:0]
10	project_c3_y	R/W	Cat3 Bullet visible? Reserved [4:0]	Projectile Cat3 Y [9:0]

Table 4: Register category - Cat array position, and mystery position

Register #	Register Name	R/W	Bit description	
			[15:0]	
11	cat_array_pos_x	R/W	Reserved [5:0]	Cat Array Position X [9:0]
12	cat_array_pos_y	R/W	Reserved [5:0]	Cat Array Position Y [9:0]
13	mystery_pos_x	R/W	Visibility? Reserved [4:0]	Mystery Position X [9:0]

Table 5: Register category - Miscellaneous

Register #	Register Name	R/W	Bit description			
			[15:0]			
14	status	R/W	Reserved [7:0]	Level [3:0]	Game Status [1:0]	Life [1:0]
15	score	R/W	Score [15:0]			
16	dog_ani	R/W	Reserved [11:0]	Dog Animation State [3:0]		
17	cat_ani	R/W	Reserved [14:0]	Cat Animation State		

Table 6: RAM category - barrier and cat matrix

Register #	Register Name	R/W	Bit description	
			[7:3]	[2:0]
28	Barrier Matrix	R/W	Address (0 to 23)	
				Value (0 to 5)
29	Cat Matrix	R/W	Address (0 to 55)	
				Value (0 to 5)

Controller

Byte	0	1	2	3	4	5	6	7
default	01	7F	7F	7F	0F	00	002	202
A					2F			
B					4F			
X					1F			
Y					8F			
Up arrow				00				
Down arrow				FF				
Left arrow			00					
Right arrow			FF					
L2						01		
R2						02		
Start						20		
Select						10		
Left arrow + A			00		2F			
Right arrow +A			FF		2F			

```
typedef struct {  
    bool leftArrowPressed;  
    bool rightArrowPressed;  
    bool buttonAPressed;  
    bool startPressed;  
} ControllerState;
```



Pseudocode of the software

```
C/C++ ▾  
Game {  
    Cat cats[][]; // All of the cats that are currently on screen.  
    Poodle player; // The poodle that the player moves.  
    int level = 1; // The current level that the player is on.  
    int lives = 3; // The number of lives that the player currently has.  
  
    cats.update(level);  
    cats.spawn();  
    player.spawn();  
}
```

```
while (true) {  
    // Updating the cats  
    cats.move();  
    cats.fire();  
    cats.checkCollisions();  
    if (cats.checkBottom()) {  
        lives -= 1;  
    }  
    // If all of the cats are eliminated, then the level advances and the scene resets.  
    if (cats.allEliminated()) {  
        level += 1;  
        cats.update(level);  
        cats.spawn();  
        player.spawn();  
    }  
  
    // Updating the player.  
    player.move();  
    player.fire();  
    if (player.checkCollisions()) {  
        lives -= 1;  
    }  
    if (lives <= 0) {  
        player.gameOver();  
    }  
}  
}
```

Questions?