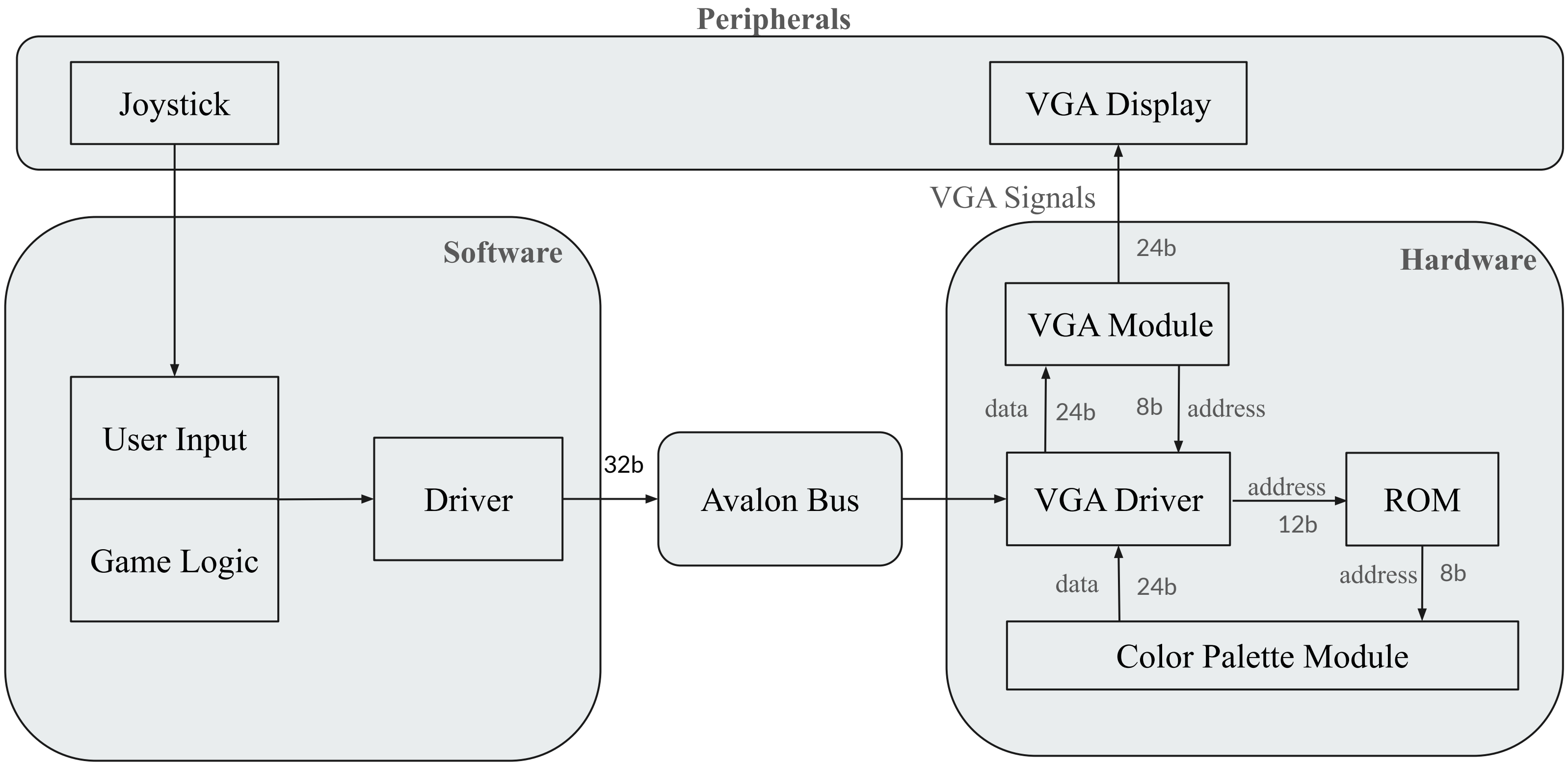
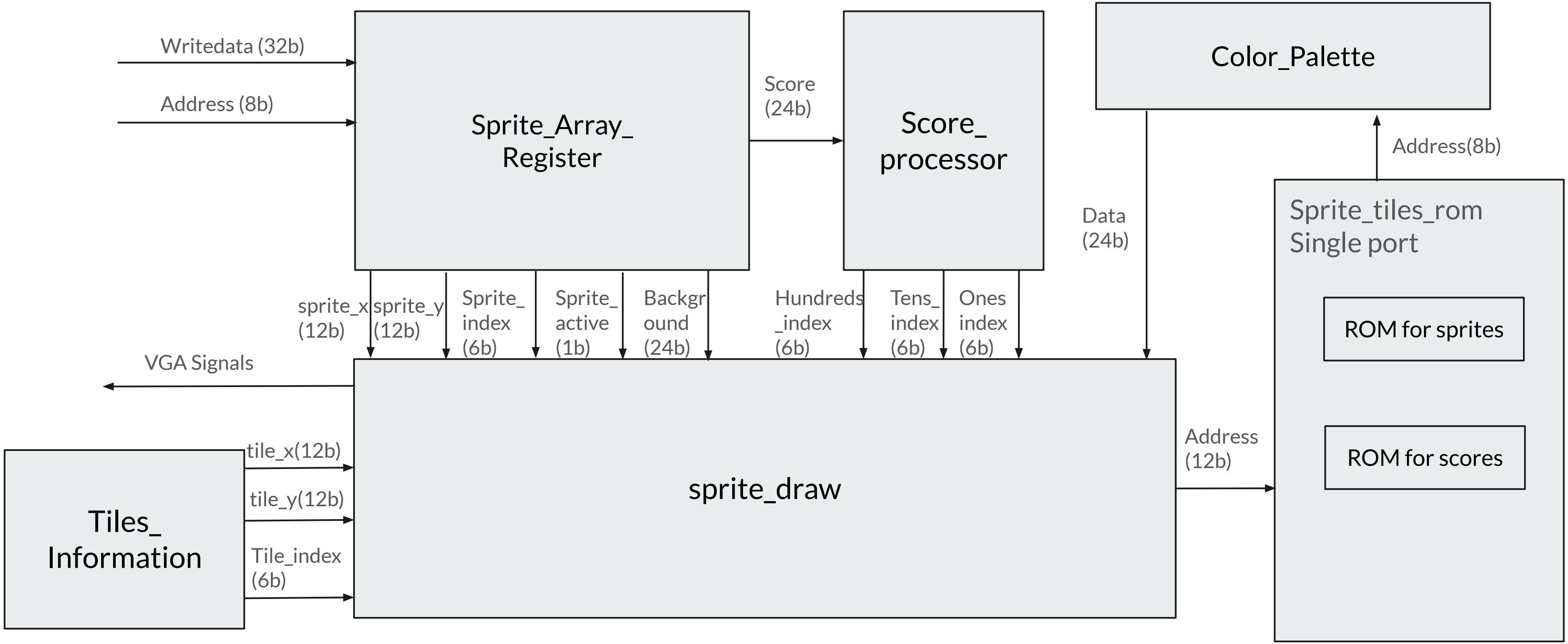
4840 Spaceship Defender Game

By Mingzhi Li, Noah Hartzfeld, Jingyi Lai, Hiroki Endo, Zhengtao Hu

Overall Structure

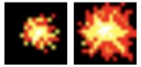


VGA control: Sprites drawing

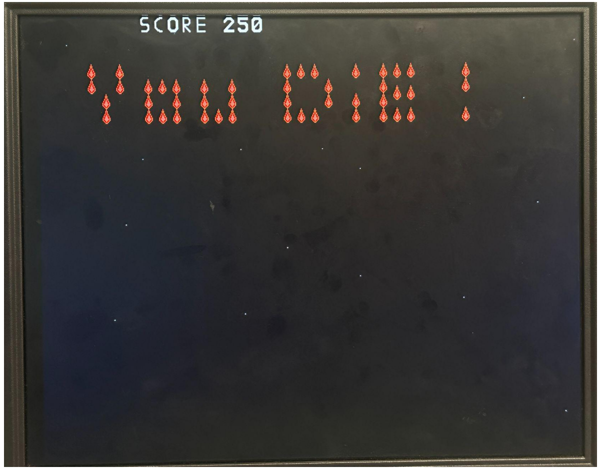


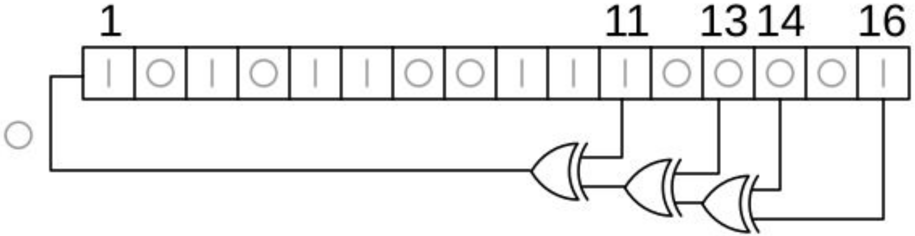
Memory Sprites + Tiles

Total: 7936

	Category	Graphics	Size (bit)	# of images	Total Size(bits)
ROM1	Spaceship		16*16	3	768
	Fire		16*16	1	256
	Player Bullet		16*16	3	768
	Enemy Ship		16*16	3	768
	Enemy Bullet		16*16	1	256
	Power Up		16*16	3	768
	Explosion Effect		16*16	2	512
ROM2	Numbers		16*16	10	2560
	Score		16*16	5	1280

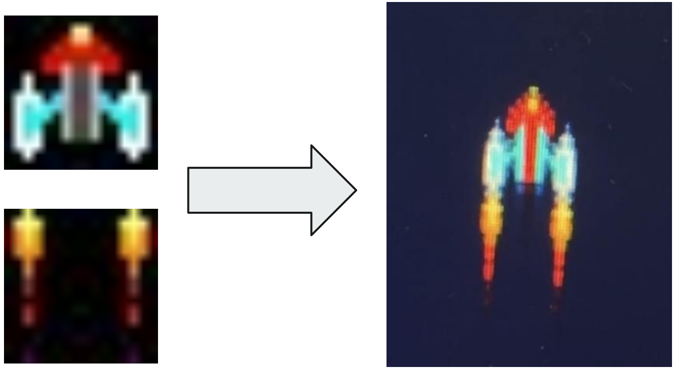
VGA control: LFSR Star Background

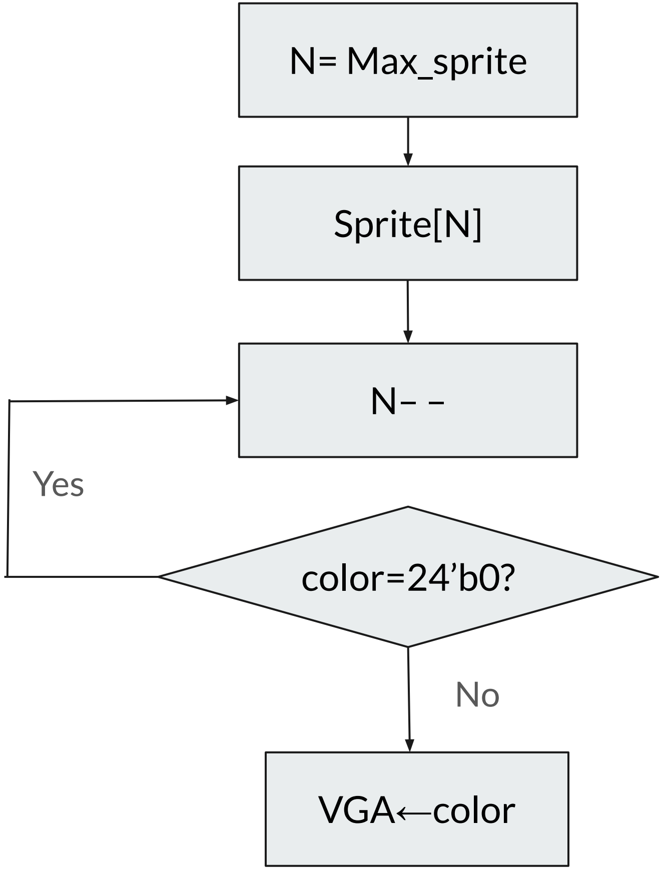




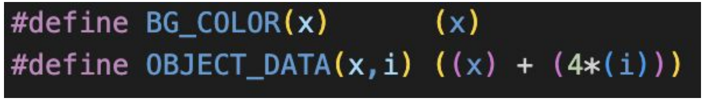
VGA control: Transparent

Use black(rgb=24’b0)
as background of each sprite image



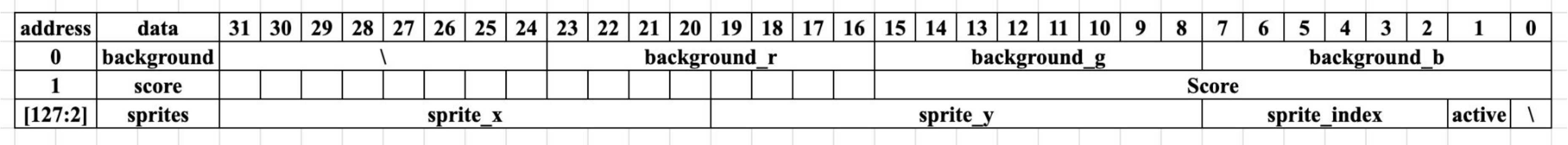


VGA HW/SW interface



```
#define BG_COLOR(x)       (x)
#define OBJECT_DATA(x,i) ((x) + (4*(i)))
```

Data: 32 bits
Address: 8 bits



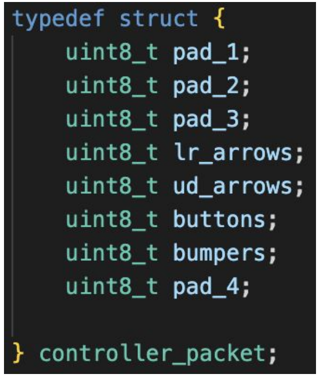
address	data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	background	\								background_r								background_g								background_b							
1	score																	Score															
[127:2]	sprites	sprite_x												sprite_y												sprite_index						active	\

Controller

The controller communicates with a 8 bytes protocol via USB

constant	constant	constant	left/right arrow	up/down arrow	A/B/X/Y	triggers/ starts	constant
(0x01)	(0x7F)	(0x7F)	(0x7F)	(0x7F)	(0x0F)	(0x00)	(0x00)

A: starts the game (0x2F)
Y: fires bullets (0xAF)
Left bumper: fires bullets (0x01)
Right bumper: fires bullets (0x02)
Left & Right bumpers: fires bullets (0x03)
Left arrow: move left (0x00)
Right arrow: move right (0xFF)
Up arrow: move up (0x00)
Down arrow: move down (0xFF)



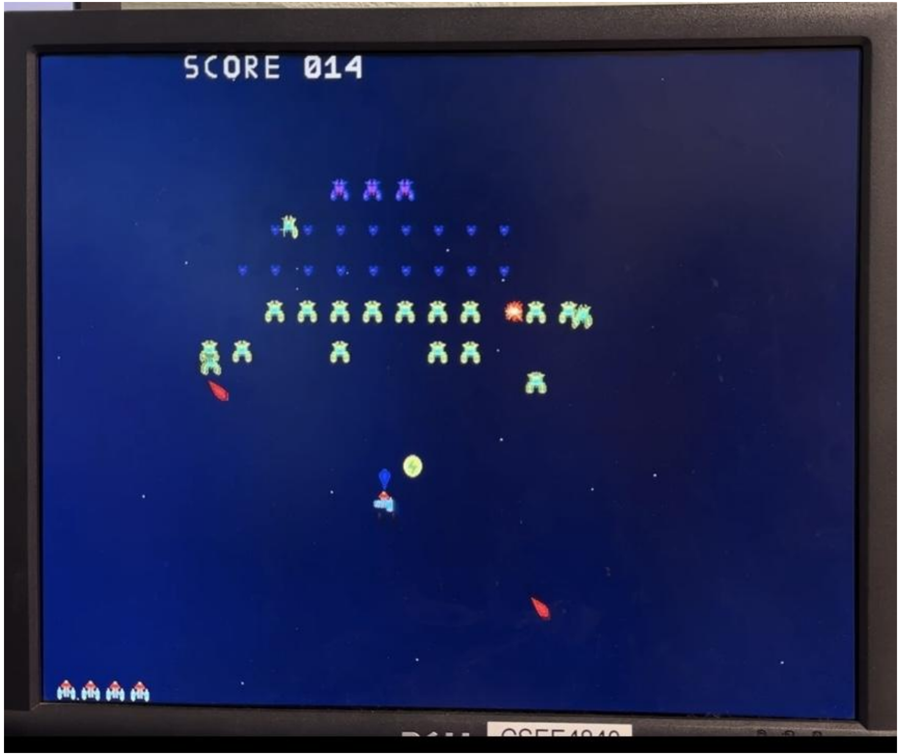
```
typedef struct {
    uint8_t pad_1;
    uint8_t pad_2;
    uint8_t pad_3;
    uint8_t lr_arrows;
    uint8_t ud_arrows;
    uint8_t buttons;
    uint8_t bumpers;
    uint8_t pad_4;

} controller_packet;
```

Game logic

Enemy generation
Enemy movement
Attack mode
Collision detection
Powerup
Winning and defeating condition

Game Picture



Demonstrate

Thank you.