# CSEE 4840 Embedded System Battle City

Tian Chu

Huangfu

Quan Yuan

Liuxun Zhu

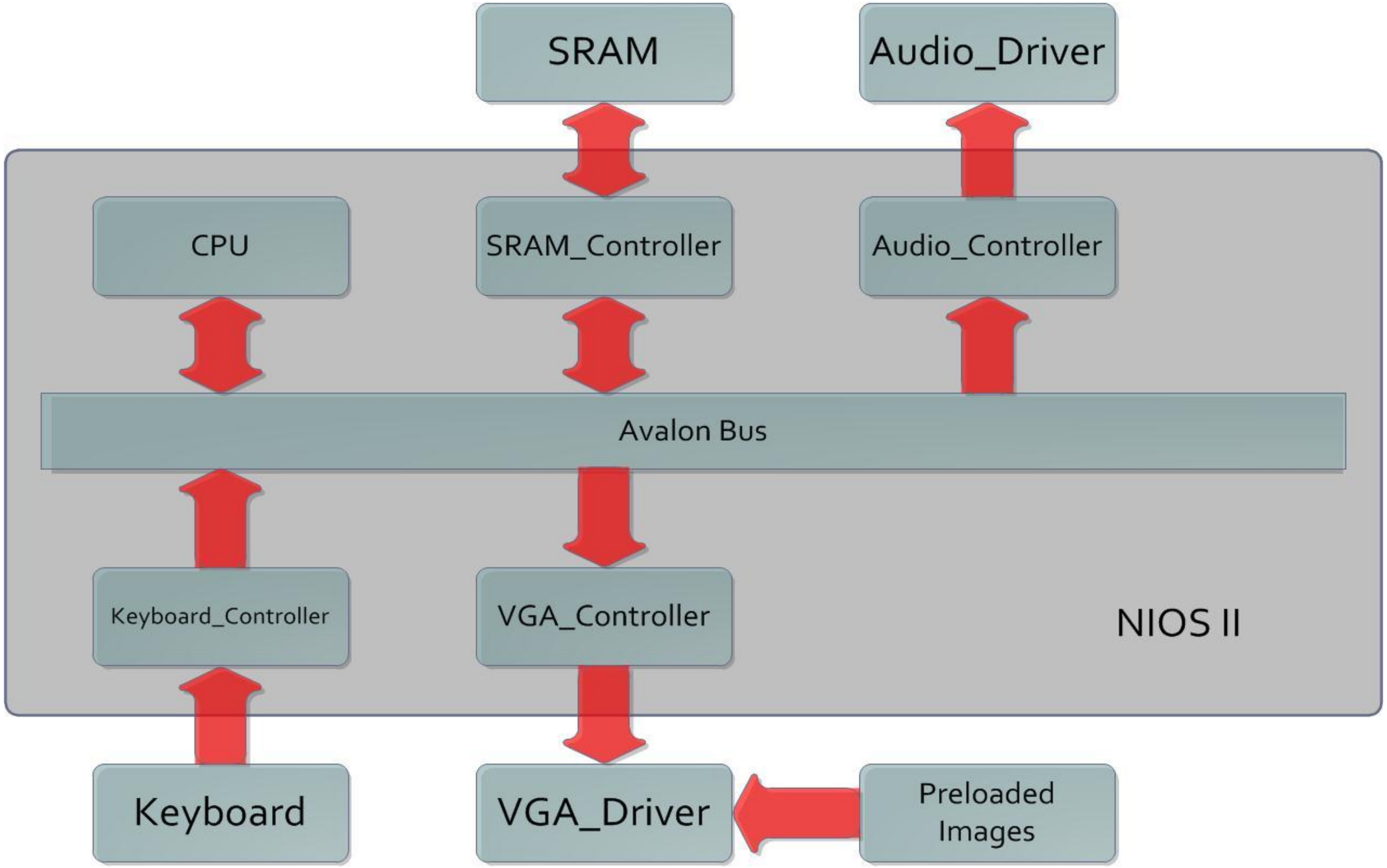
Tianchen Li

© TemplatesWise.com

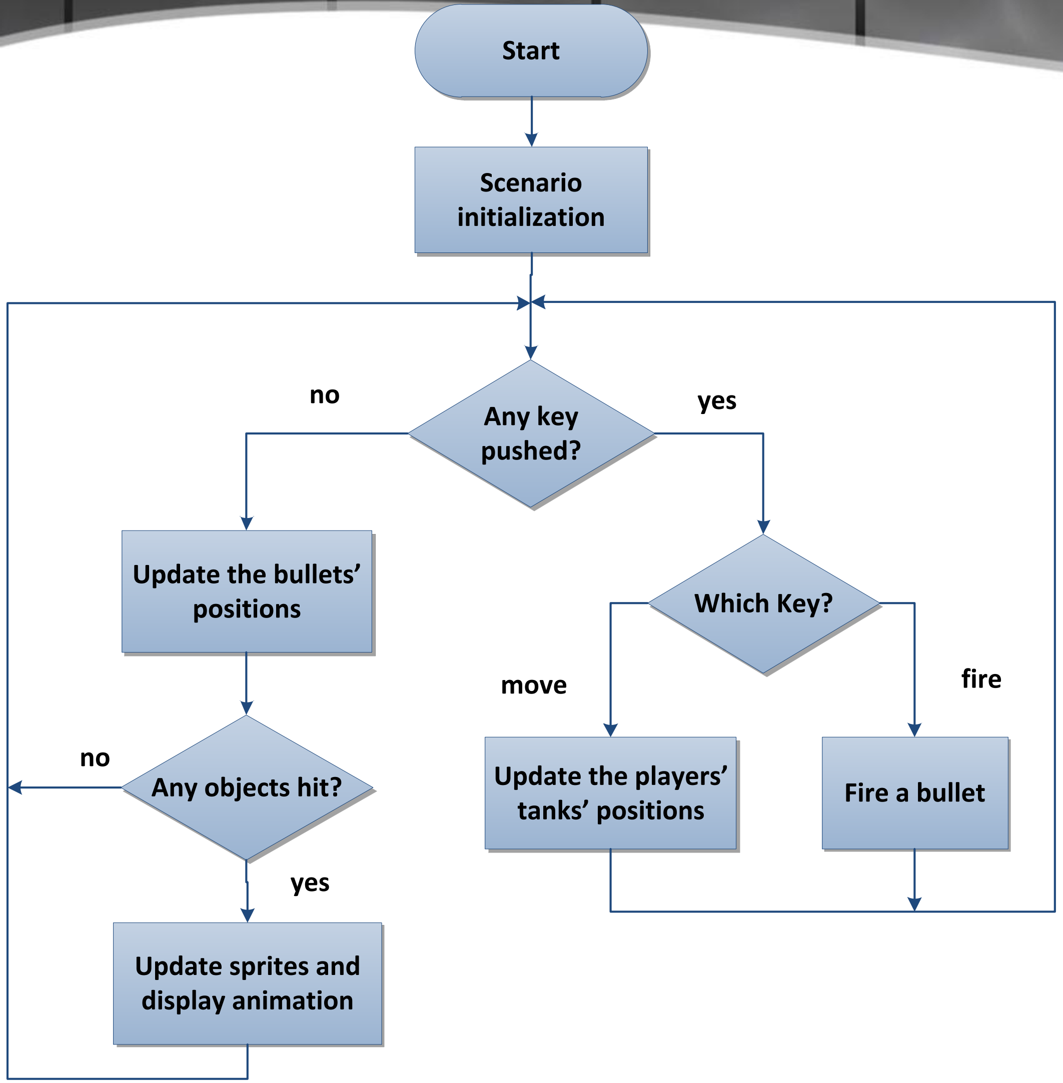
# Overview

- Game play
  - One player
  - Three stages (more can be easily added)
  - Two modes: “easy”(E) and “crazy”(C)
- Keyboard control
  - Move: “A,S,W,D”
  - Fire: “Space”
  - Start game: “Enter”

# Design Architecture



# Software Design


Start
Scenario initialization
Any key pushed?
no
yes
Update the bullets' positions
Which Key?
move
fire
no
Any objects hit?
yes
Update the players' tanks' positions
Fire a bullet
Update sprites and display animation

# Software Design

- Sprites are stored in integers:

| X | Y | Types | Color |
|---|---|---|---|
| 23-15 | 14-6 | 5-2 | 1-0 |

- Backgrounds are split into 13 by 13 blocks, and each of them is stored in an integer:

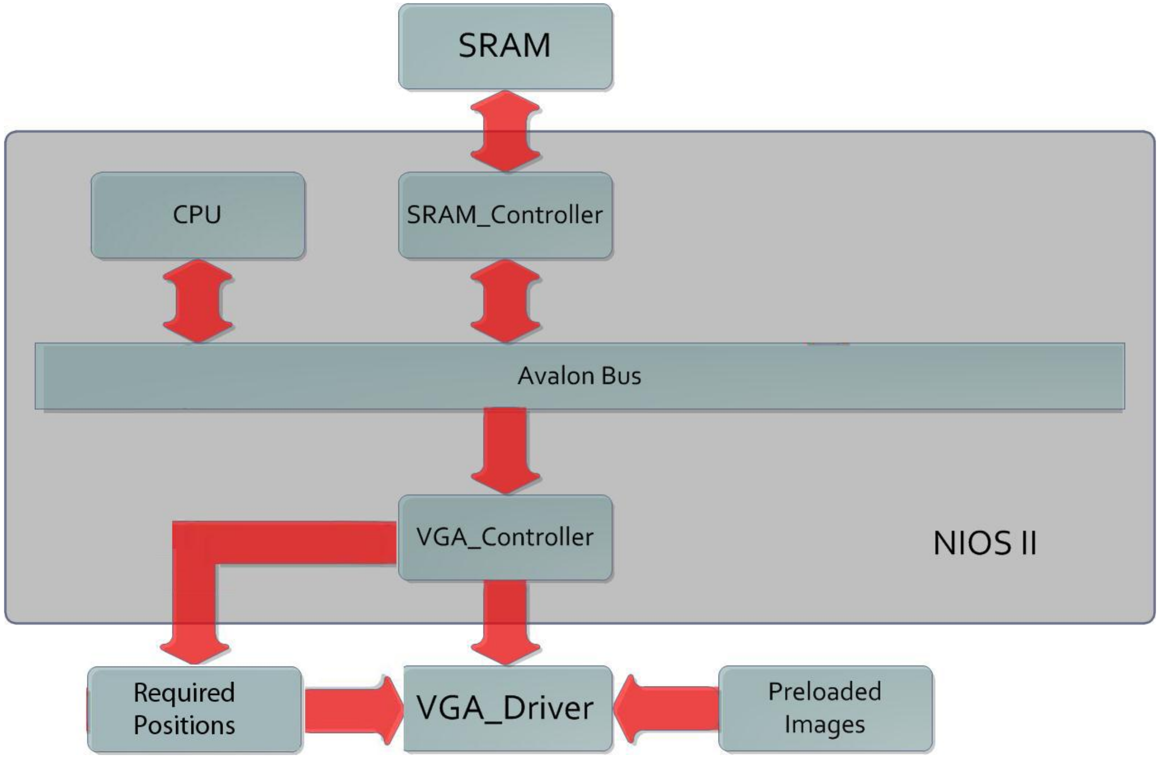
| X | | | | Y | | | | Blocks | | | | Type | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

# VGA Design

- Main Tasks
  - Load the scenario setup and locate images on the screen.
  - Adjust the tanks and bullets' positions
  - Handle the overlapping problem
  - Display animations

# VGA Design

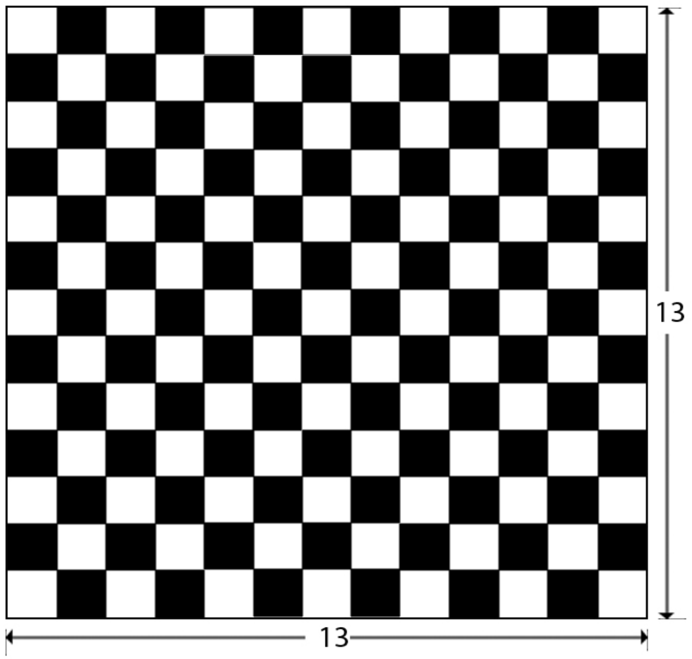
- VGA Architecture



# VGA Design

- Image Processing:

The user screen is divided into 169 squares, each column or row contains 13 squares as indicated below:

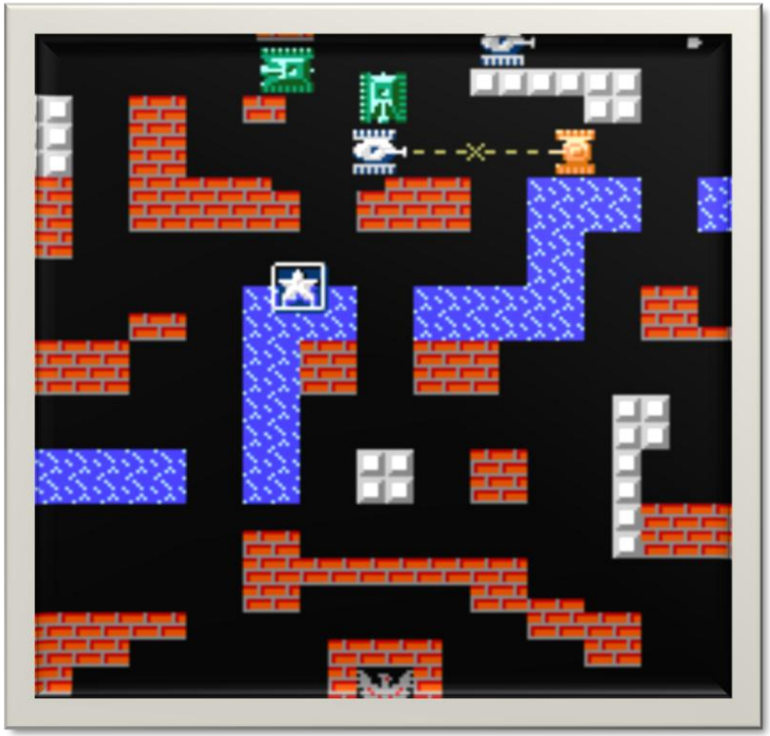


Store in array

# VGA Design

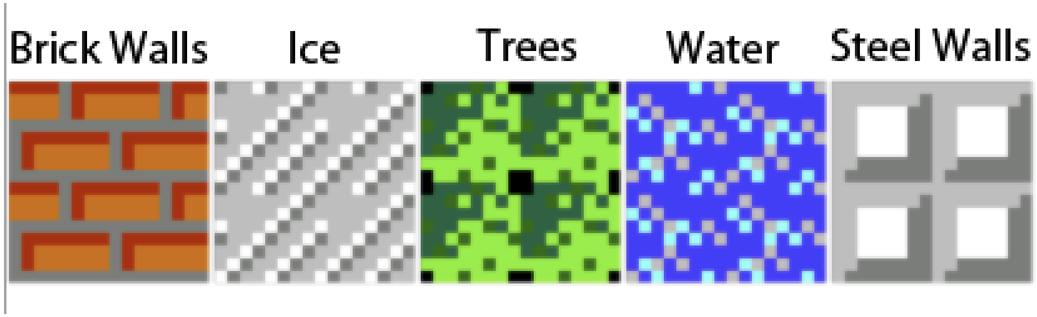
- Image Processing:

The game scenario is constructed by 169 different images that loaded from RAM. VGA_Controller will determine each image and its position.



# VGA Design

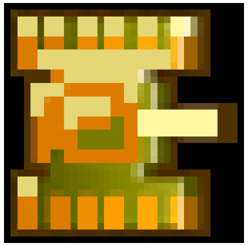
- Image type:
  - Static Scenario

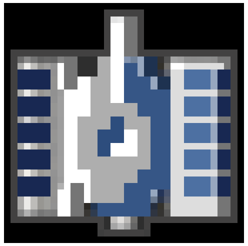


Static Scenarios are predefined before the game starts. Software controls the scenario mapping and send massages to VGA_Driver. Then location information will be stored in RAM and we only focus on the changes after scenario initialization.

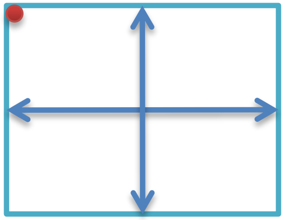
# VGA Design

- Image type:
  - Sprites





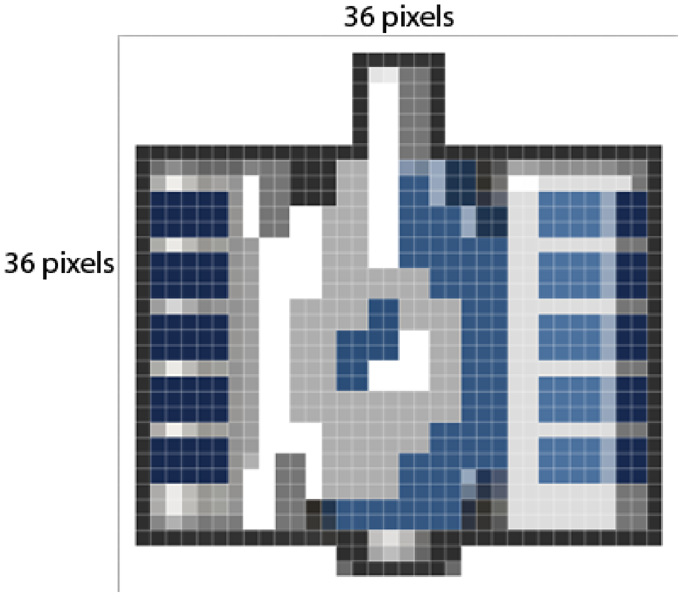
Former method for static scenario is impractical.
To locate the sprites, we use the left top point and image length and width.



# VGA Design

- Image Processing:

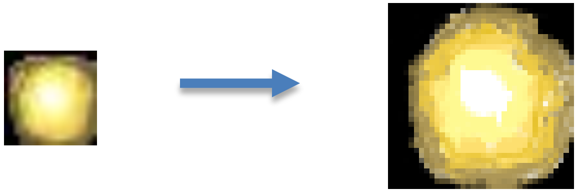
The sub-image is formed by 36X36 pixels. The image below is the tank image.



# VGA Design

- Animation Effect:

To achieve explosion effect, two explosion images of different sizes are used as frames.



Display the two images in different clock periods and the dynamic effect will be shown.

# VGA Design

- Overlapping issue:

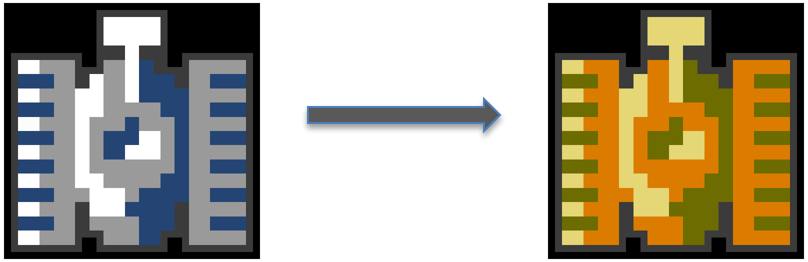
  - Set the overlapping area of upper layer to black.
  - Judge the color of image, if it is black, write the data of lower layer to the non-overlapping area.

# VGA Design

- Color conversion:

Pixels of tank are represented by 24-bits. The color information stored of tanks occupied a large space of memory.

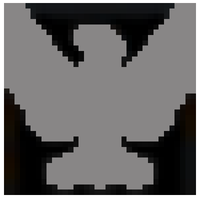
Since the appearances of player tank and enemy are the same, color conversion is implemented to save memory space.



VGA Design

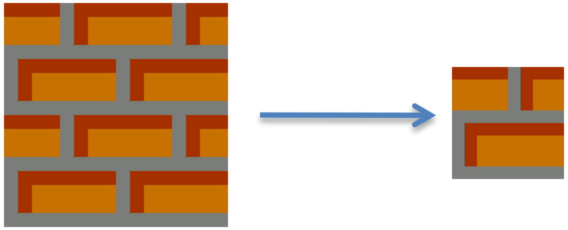
- Data Compression

1- bit representation:



2- bit representation:

Reduced size:

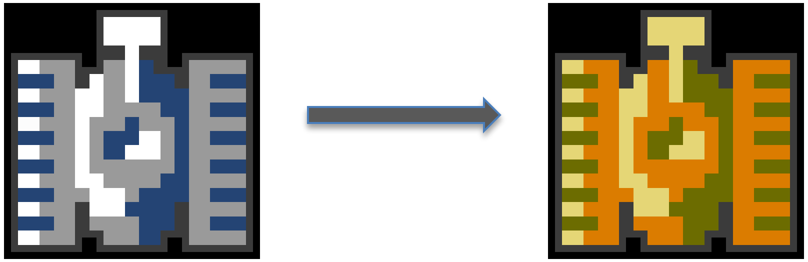


# VGA Design

- Color conversion:

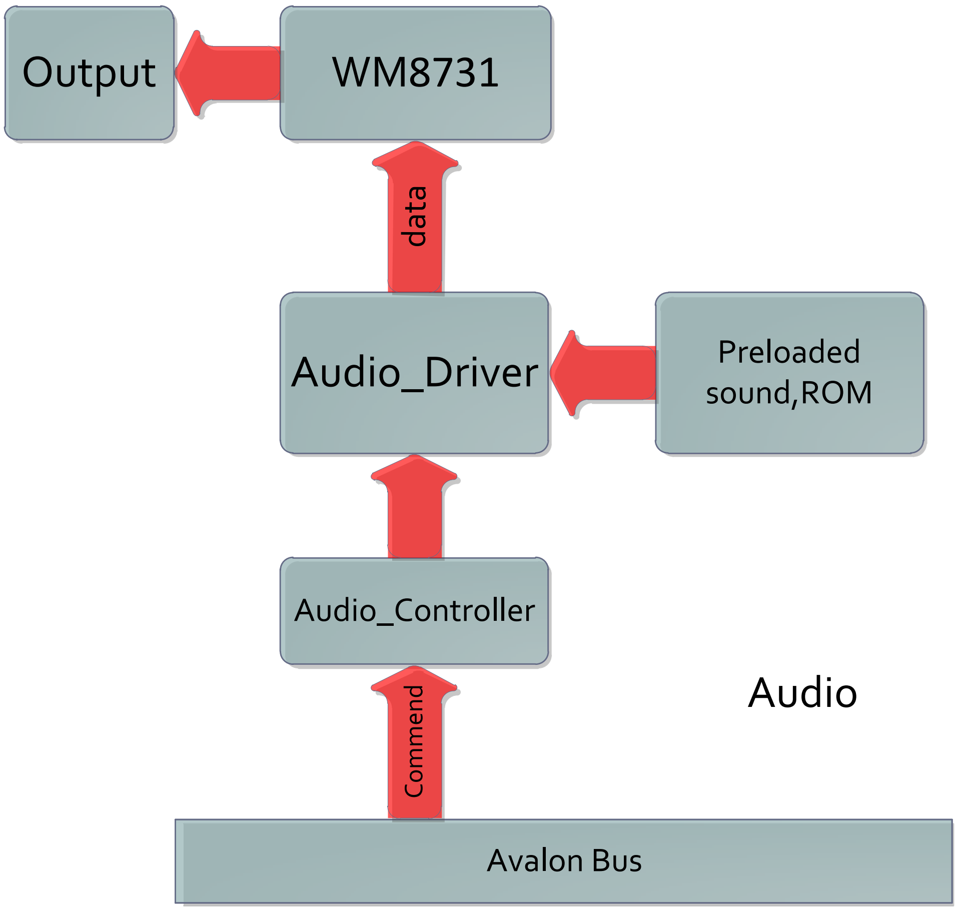
Pixels of tank are represented by 24-bits. The color information stored of tanks occupied a large space of memory.

Since the appearances of player tank and enemy are the same, color conversion is implemented to save memory space.



# Audio Design

- Audio architecture



# Audio Design

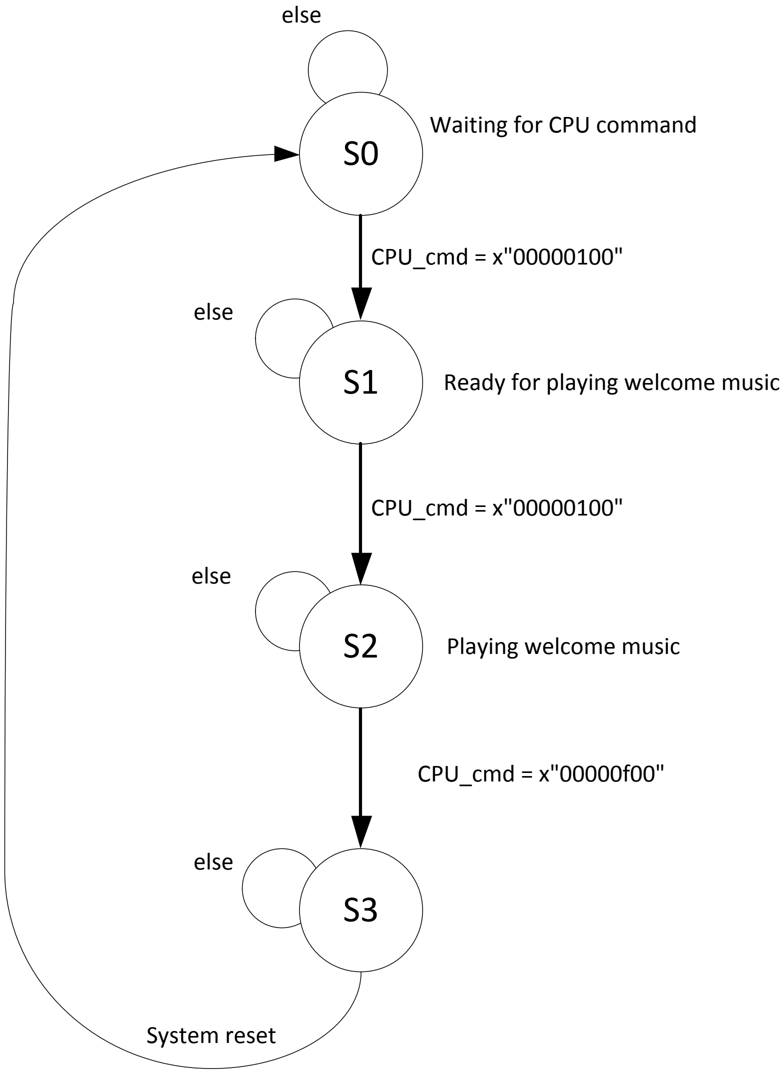
- 2 kinds of music

  - ➢ Welcome music
    - composed of sin wave with different frequency.
    - Using sin wave data in Lab 3

  - ➢ Sound effect
    - fire and explosion.
    - convert wave to mif, save and play.
    - using adder at output to play two kinds of sound effect simultaneously  if needed.

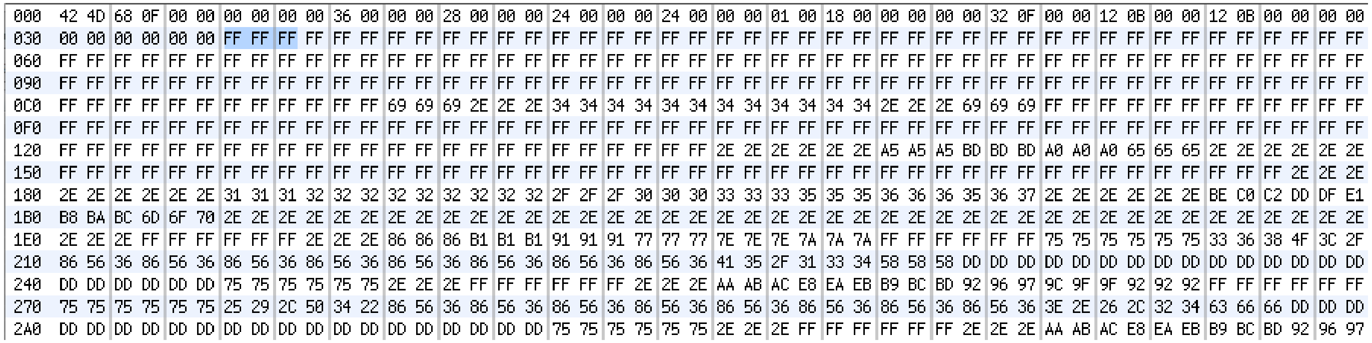
# Audio Design

- State machine



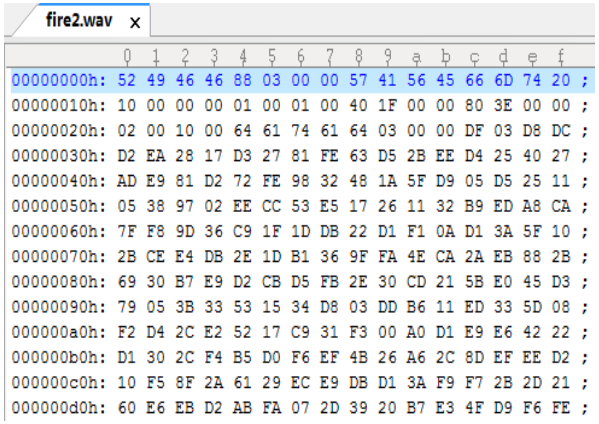
# BMP to mif

- Images are stored in RAM as a pixel matrix.
- Read the bmp file in Hex starting from the 0x36th byte
- Store RGB information into .mif file.



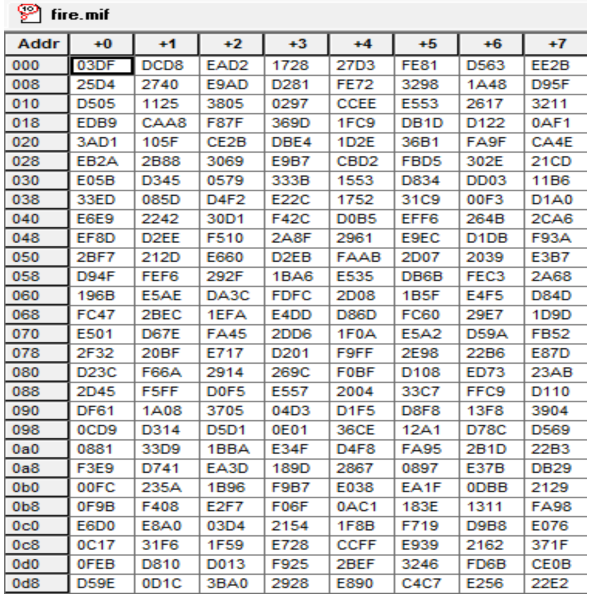
```
000  42 4D 68 0F 00 00 00 00 00 00 36 00 00 00 28 00 00 00 24 00 00 00 24 00 00 00 01 00 18 00 00 00 00 00 32 0F 00 00 12 0B 00 00 12 0B 00 00 00 00
030  00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
060  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
090  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0C0  FF FF FF FF FF FF FF FF FF FF FF FF 69 69 69 2E 2E 2E 34 34 34 34 34 34 34 34 34 34 34 34 2E 2E 2E 69 69 69 FF FF FF FF FF FF FF FF FF FF FF FF
0F0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
120  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 2E 2E 2E 2E 2E 2E A5 A5 A5 BD BD BD A0 A0 A0 65 65 65 2E 2E 2E 2E 2E 2E
150  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 2E 2E 2E
180  2E 2E 2E 2E 2E 2E 31 31 31 32 32 32 32 32 32 32 32 32 2F 2F 2F 30 30 30 33 33 33 35 35 35 36 36 36 35 36 37 2E 2E 2E 2E 2E 2E BE C0 C2 DD DF E1
1B0  B8 BA BC 6D 6F 70 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E 2E
1E0  2E 2E 2E FF FF FF FF FF FF 2E 2E 2E 86 86 86 B1 B1 B1 91 91 91 77 77 77 7E 7E 7E 7A 7A 7A FF FF FF FF FF FF 75 75 75 75 75 75 33 36 38 4F 3C 2F
210  86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 41 35 2F 31 33 34 58 58 58 DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD
240  DD DD DD DD DD DD 75 75 75 75 75 75 2E 2E 2E FF FF FF FF FF FF 2E 2E 2E AA AB AC E8 EA EB B9 BC BD 92 96 97 9C 9F 9F 92 92 92 FF FF FF FF FF FF
270  75 75 75 75 75 75 25 29 2C 50 34 22 86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 86 56 36 3E 2E 26 2C 32 34 63 66 66 DD DD DD
2A0  DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD 75 75 75 75 75 75 2E 2E 2E FF FF FF FF FF FF 2E 2E 2E AA AB AC E8 EA EB B9 BC BD 92 96 97
```

# Wave to mif



fire2.wav

```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 52 49 46 46 88 03 00 00 57 41 56 45 66 6D 74 20 ;
00000010h: 10 00 00 00 01 00 01 00 40 1F 00 00 80 3E 00 00 ;
00000020h: 02 00 10 00 64 61 74 61 64 03 00 00 DF 03 D8 DC ;
00000030h: D2 EA 28 17 D3 27 81 FE 63 D5 2B EE D4 25 40 27 ;
00000040h: AD E9 81 D2 72 FE 98 32 48 1A 5F D9 05 D5 25 11 ;
00000050h: 05 38 97 02 EE CC 53 E5 17 26 11 32 B9 ED A8 CA ;
00000060h: 7F F8 9D 36 C9 1F 1D DB 22 D1 F1 0A D1 3A 5F 10 ;
00000070h: 2B CE E4 DB 2E 1D B1 36 9F FA 4E CA 2A EB 88 2B ;
00000080h: 69 30 B7 E9 D2 CB D5 FB 2E 30 CD 21 5B E0 45 D3 ;
00000090h: 79 05 3B 33 53 15 34 D8 03 DD B6 11 ED 33 5D 08 ;
000000a0h: F2 D4 2C E2 52 17 C9 31 F3 00 A0 D1 E9 E6 42 22 ;
000000b0h: D1 30 2C F4 B5 D0 F6 EF 4B 26 A6 2C 8D EF EE D2 ;
000000c0h: 10 F5 8F 2A 61 29 EC E9 DB D1 3A F9 F7 2B 2D 21 ;
000000d0h: 60 E6 EB D2 AB FA 07 2D 39 20 B7 E3 4F D9 F6 FE ;
```



fire.mif

| Addr | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 |
|---|---|---|---|---|---|---|---|---|
| 000 | 03DF | DCD8 | EAD2 | 1728 | 27D3 | FE81 | D563 | EE2B |
| 008 | 25D4 | 2740 | E9AD | D281 | FE72 | 3298 | 1A48 | D95F |
| 010 | D505 | 1125 | 3805 | 0297 | CCEE | E553 | 2617 | 3211 |
| 018 | EDB9 | CAA8 | F87F | 369D | 1FC9 | DB1D | D122 | 0AF1 |
| 020 | 3AD1 | 105F | CE2B | DBE4 | 1D2E | 36B1 | FA9F | CA4E |
| 028 | EB2A | 2B88 | 3069 | E9B7 | CBD2 | FBD5 | 302E | 21CD |
| 030 | E05B | D345 | 0579 | 333B | 1553 | D834 | DD03 | 11B6 |
| 038 | 33ED | 085D | D4F2 | E22C | 1752 | 31C9 | 00F3 | D1A0 |
| 040 | E6E9 | 2242 | 30D1 | F42C | D0B5 | EFF6 | 264B | 2CA6 |
| 048 | EF8D | D2EE | F510 | 2A8F | 2961 | E9EC | D1DB | F93A |
| 050 | 2BF7 | 212D | E660 | D2EB | FAAB | 2D07 | 2039 | E3B7 |
| 058 | D94F | FEF6 | 292F | 1BA6 | E535 | DB6B | FEC3 | 2A68 |
| 060 | 196B | E5AE | DA3C | FDFC | 2D08 | 1B5F | E4F5 | D84D |
| 068 | FC47 | 2BEC | 1EFA | E4DD | D86D | FC60 | 29E7 | 1D9D |
| 070 | E501 | D67E | FA45 | 2DD6 | 1F0A | E5A2 | D59A | FB52 |
| 078 | 2F32 | 20BF | E717 | D201 | F9FF | 2E98 | 22B6 | E87D |
| 080 | D23C | F66A | 2914 | 269C | F0BF | D108 | ED73 | 23AB |
| 088 | 2D45 | F5FF | D0F5 | E557 | 2004 | 33C7 | FFC9 | D110 |
| 090 | DF61 | 1A08 | 3705 | 04D3 | D1F5 | D8F8 | 13F8 | 3904 |
| 098 | 0CD9 | D314 | D5D1 | 0E01 | 36CE | 12A1 | D78C | D569 |
| 0a0 | 0881 | 33D9 | 1BBA | E34F | D4F8 | FA95 | 2B1D | 22B3 |
| 0a8 | F3E9 | D741 | EA3D | 189D | 2867 | 0897 | E37B | DB29 |
| 0b0 | 00FC | 235A | 1B96 | F9B7 | E038 | EA1F | 0DBB | 2129 |
| 0b8 | 0F9B | F408 | E2F7 | F06F | 0AC1 | 183E | 1311 | FA98 |
| 0c0 | E6D0 | E8A0 | 03D4 | 2154 | 1F8B | F719 | D9B8 | E076 |
| 0c8 | 0C17 | 31F6 | 1F59 | E728 | CCFF | E939 | 2162 | 371F |
| 0d0 | 0FEB | D810 | D013 | F925 | 2BEF | 3246 | FD6B | CE0B |
| 0d8 | D59E | 0D1C | 3BA0 | 2928 | E890 | C4C7 | E256 | 22E2 |

# Challenges

- VGA
  - A lot of sprites
  - Frequent display switch between sprites and static backgrounds
  - Memory size is limited
- Audio
  - 2 kinds of music. Need to play both sin wave and wave file
  - Need to play 2 kinds of sound effect simultaneously sometimes
- Software
  - A lot of sprites, like bullets, tanks, and explosions
  - Complex game logics, such as bullets collision with obstacles, tanks and even other bullets.

# Lessons Learned

- Appropriate design partition is a key for working as a team
- Good data structure is important for implementing complex functions
- Backup source files regularly
- Dropbox helps sharing project files