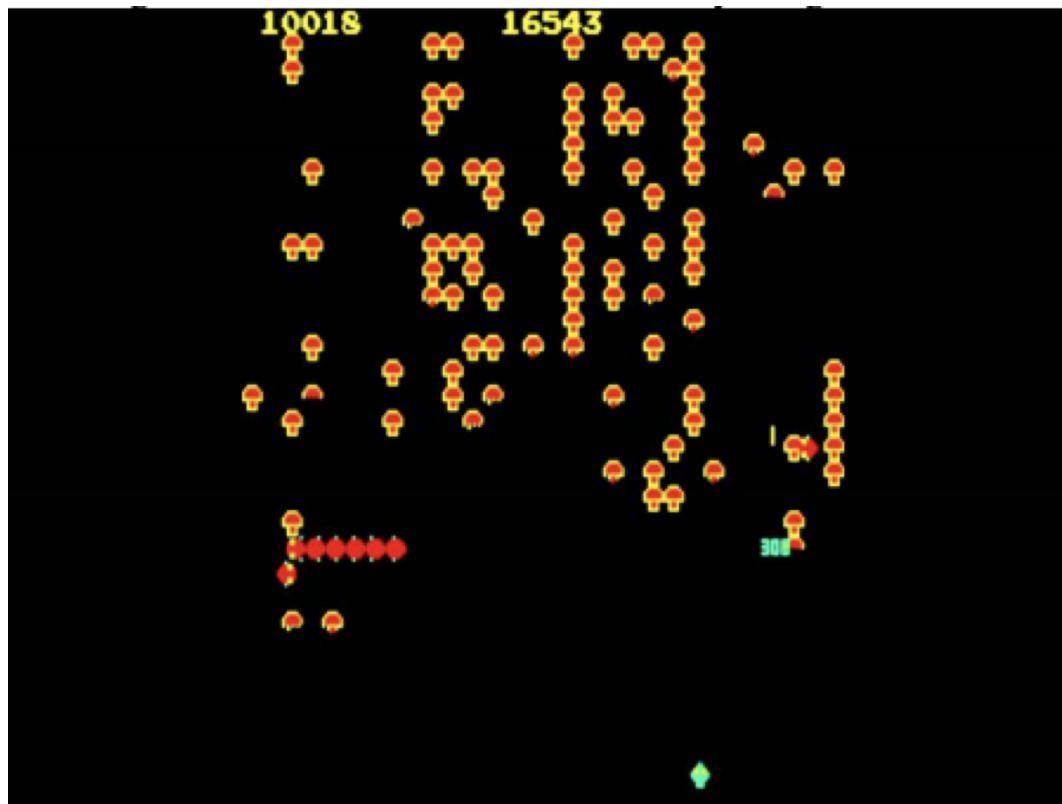


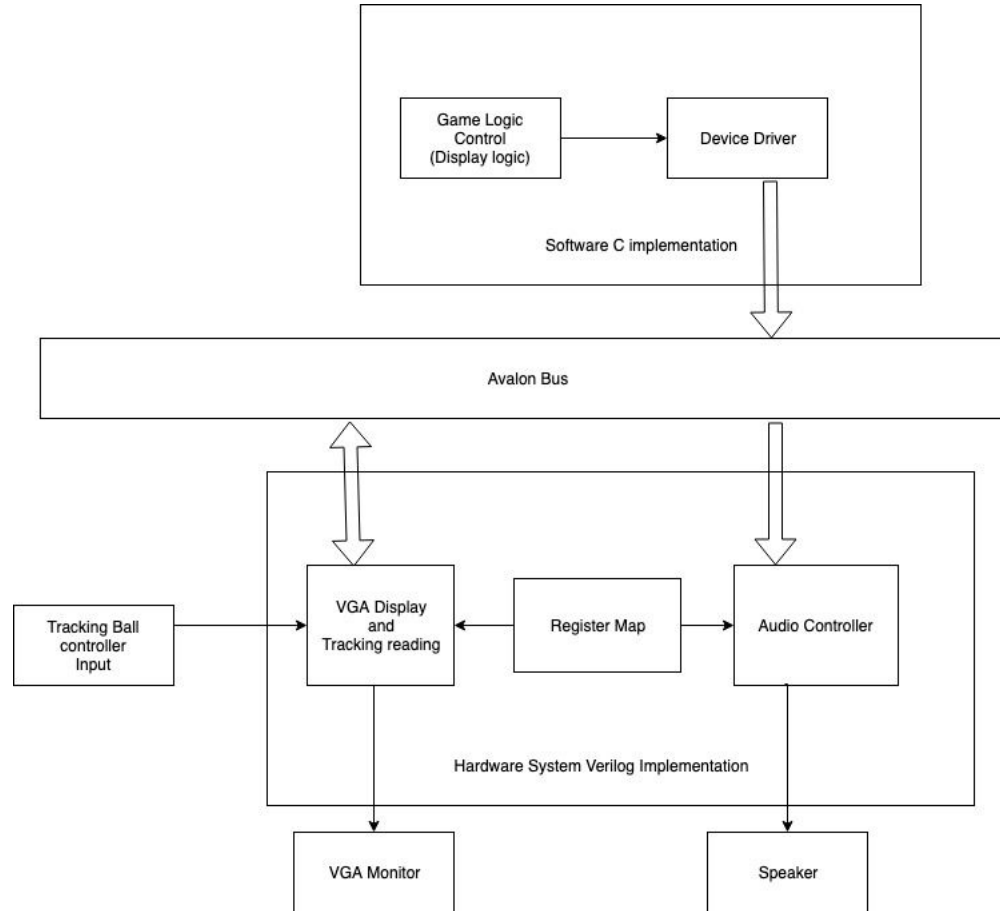
# Centipede

Haoran Geng, Xuyang Liu, Ziyu Zhou, Donglai Guo  
CSEE4840 - Embedded System

# Introduction



# Architecture



# Game Logic Design

1. Mushroom
2. Centipede (Enemy)
3. Spider (Enemy)
4. Player and Bullet
5. Life
6. Current Score and Highest Score

# Device Driver

1. lowrite8 to iowrite16 for display and sound
2. ioread16 for tracking ball

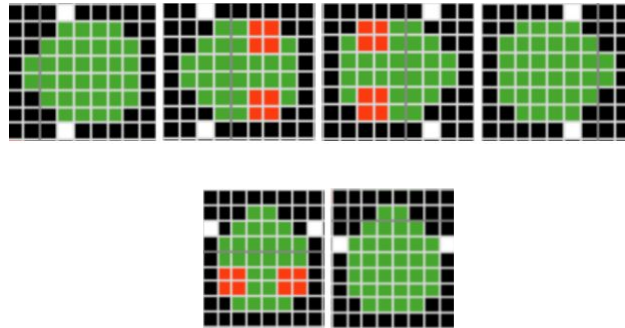
# Mushroom

1. Divided the frame into a 40 x 30 grid
2. 0 represent dead and 1 represent alive
3. Random generate at beginning of the game
4. Could block player movement



# Centipede direction and collision

1. Generated at the top left of the frame
2. 10 centipede total
3. Moving down and change direction when hit a mushroom or the edge
4. Use the “tile” of to represent the status of the centipede
5. Killed by bullet and generate a mushroom at the dead centipede position
6. Send to device driver through an array with length of 20.



# Spider

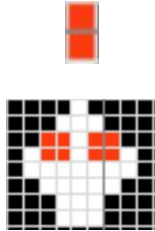
1. Generated at the bottom half of the screen
2. Move like a bouncing ball
3. Killed by a bullet and will be respawned at a random position
4. Sending to device driver through an array with length of 2





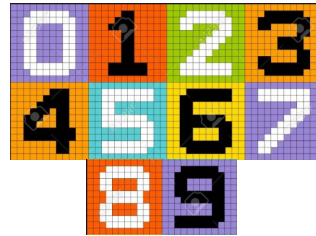
# Player and Bullet

1. Player controlled by the trackball
2. Player cannot hit over mushroom
3. Player can shoot bullet
4. Bullet store inside an array with length of 60



# Life system and score system

1. Have 3 life initially
2. Lose a life when player hit a centipede and spider
3. Lose a life when centipede hit the bottom of the fram
4. Mushroom - 50, Spider - 600, Centipede body - 200, Centipede head - 300
5. Highest score store in a txt file in the SD card
6. Updated highest score when number of life is 0
7. Life number, score and highest score store as an integer



# Image display

- One 16\*16 register to save the content of the images
- Two 16 bit register to control the position of the moving objects
- 3 bit register to hold the state of centipede
- Tiles to show background
- Combination of tile and sprite

# Image display

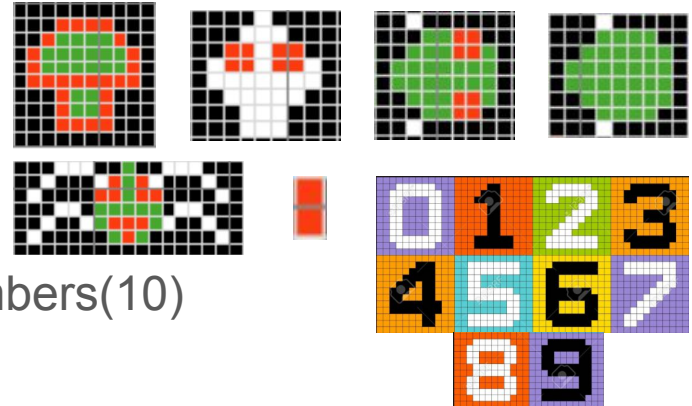
- Background:

The whole screen contains 40\*30 tiles. The first row of the tiles show the present score, highest score and remaining life. The rest of the tiles will show mushrooms randomly at the beginning of each round controlled by software.

- Character:

Each character contains 16\*16 pixel.

Centipede(6), Player(1), Bullet(1), Spider(1) Numbers(10)



# Moving feet

- Split the 50M frequency into 48hz (centipede) and 3 hz (spider)
- Draw different pairs of feet
- Show different feet at interval according to the high and low level of the clock

# Audio

- Generate a memory initialization file (MIF) for sound
- Single-port ROM memory blocks
- To save memory space, edit audio file length and choose reasonable sample rate to balance sound effect and size

# Audio

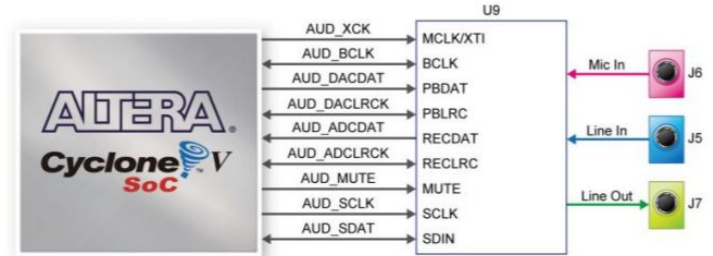
## 1. Use these 3 IPs in Qsys

**Audio and Video Config:** configures the CODEC over an I2C bus with parameters specified in Qsys.

**Audio Clock for DE-series Boards:** a PLL which generates the 12.288MHz clock for the CODEC.

**Audio:** Provides a FIFO buffer for audio samples, and feeds them serially to the CODEC.

## 2. Write a module to send the 16-bit samples to the fifo at a rate of 48kHz



# Tracking ball

- Modify the ps/2 protocol baseline.
- Add some registers for the software to configure and read the data from.





# Register map

0 - 89	Mushroom tiles' patterns
90 - 173	Positions of objects
174 - 180	Tracking ball
180 - 230	scores and centipede states

# Challenge

1. Moving smoothly
2. Tracking ball reading
3. Object collision

# Future Work

Collision Detection

Add more enemy, such as Flea and Scorpion.

Add beginning menu