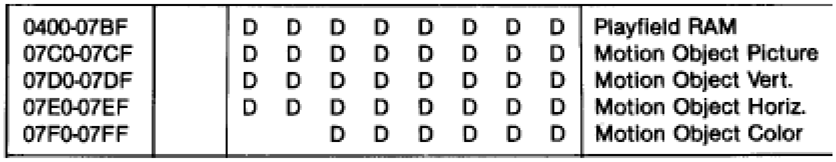
Lesson 6: Testing Video Version 1.05

Since the Fluke 9010A can write to memory, and the video hardware reads memory for instructions of how to draw and animate the screen, you can use the 9010A to write to video memory and cause the video circuits to draw what you wish. This allows you to activate the video circuits in a controllable manner and validate your video circuitry.

Testing Tile Video:

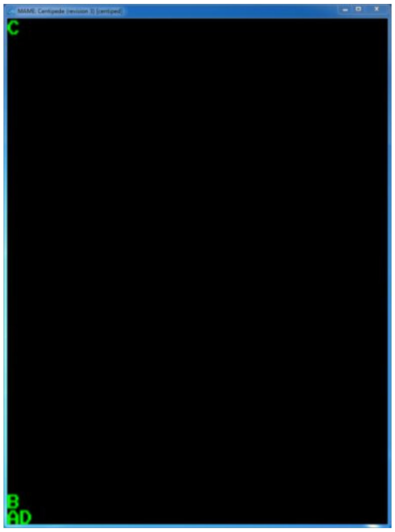
Most arcade PCBS have special circuits that draw images to the screen by reading data from the video memory and operating on it. With knowledge of the memory layout (here comes the memory map and schematics again), you can use your 9010A to actuallly test the video circuits by writing the correct values to memory. For example lets look at the memory map for Centipede again.



0400-07BF		D	D	D	D	D	D	D	D	Playfield RAM
07C0-07CF		D	D	D	D	D	D	D	D	Motion Object Picture
07D0-07DF		D	D	D	D	D	D	D	D	Motion Object Vert.
07E0-07EF		D	D	D	D	D	D	D	D	Motion Object Horiz.
07F0-07FF				D	D	D	D	D	D	Motion Object Color

Centipede's video hardware is capable of drawing individual predefined pictures to the screen (The pictures are defined in the video ROMS). Let's call these pictures *tiles.* Centipede has 32 tiles from top to bottom and 30 tiles from left to right for a total of 32x30 = 960 tiles. Each distinct picture (tile) that can be drawn has an unique identifier from 0x00 - 0xFF (1 byte) and each location where a *tile* can be drawn has a distinct memory location. On Centipede these are the 960 memory locations in the *Playfield RAM* address range 0x0400 - 0x07BF. The way you draw a tile on the screen is to write the *tile* image identifier to the memory address that corresponds to the tile that you wish to draw to in the Playfield RAM range.

The tile locations start at 0x0400 which is the bottom left corner of the screen (A on the image below), and increase in number going in a straight line up the screen until 0x041F (C on image below). The next tile 0x0420) starts again at the bottom of the screen one column over (D on image below).



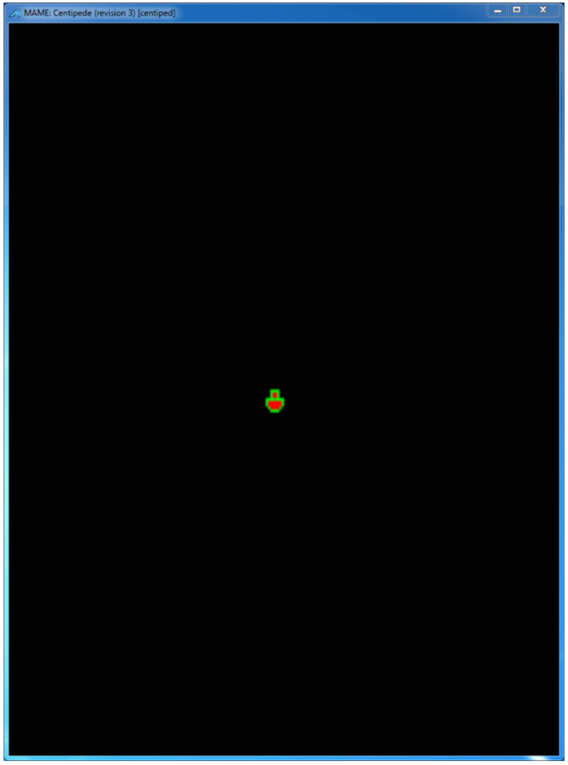
You can use your 9010A to write data to the Video RAM to test that:

- The video circuits are reading the tile memory from the video RAM properly
- The video circuits are displaying the tiles correctly

To do this simply use the 9010A's write button to store a tiles identifier in the video memory location that corresponds to the location that you would like the tile drawn to.

Exercise 1: Writing tile data to video memory.

1. Setup the Fluke properly (you should know how to do this correctly by now, but remember power the Fluke on FIRST before the Centipede PCB)
2. In the Fluke hit the “Setup” Button, choose “more” until “Active Line Force” is selected, choose “No”
3. Hit “Bus Test” to make sure all is working OK
4. Hit the “Write” key when asked what address to write to, choose the address corresponding to the tile you'd like to fill. In this example lets write something to a tile approximately center of the screen, use address 0x5CF
5. When asked for the value to write, choose FF, you should notice an upside down mushroom drawn about center of your screen.

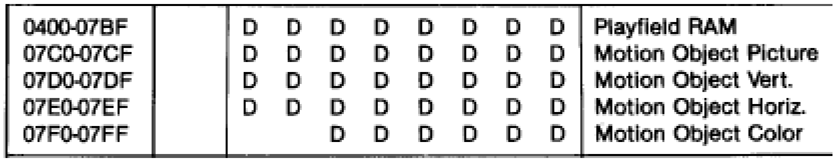


6. Play around with different addresses between 0x0400 - 0x07EF and different values. see if you can write your name in a straight line horizontally.

Testing Sprite Video:

Just like tiles can be drawn the the screen, most old arcade games have special video hardware to draw moving objects (*sprites*) to the screen. Centipede is no different, the Centipede hardware can draw up to 16 different sprites at to the screen at one time. There are 16 different sprite memory locations in which you can define a sprite to be displayed. In addition, there is a memory location for each sprite which defines the sprite's vertical location in pixels, a memory location which defines the sprites horizontal location in pixels, and a memory location which defines the sprites colors.

Let's look at the memory map again.



0400-07BF		D	D	D	D	D	D	D	D	Playfield RAM
07C0-07CF		D	D	D	D	D	D	D	D	Motion Object Picture
07D0-07DF		D	D	D	D	D	D	D	D	Motion Object Vert.
07E0-07EF		D	D	D	D	D	D	D	D	Motion Object Horiz.
07F0-07FF				D	D	D	D	D	D	Motion Object Color

The sprite identifiers are written to the 16 addreses between 0x07C0 - 0x07CF. Like the tiles, each location holds the identifier that tells the system which sprite you would like to drawn. Let's say we would like to draw the Centipede spider with it's legs up. To do this, we would need to:

- Write a 0x14 to location 0x07C0, which defines sprite #1 we would like the image of the spider with legs up.
- We would then fill the corresponding vertical position of sprite #1. This is the first address of motion object vertical memory (0x07D0). Let's put it about center vertically with a value of 0x80
- We would then fill the corresponding horizontal position of sprite #1. This is the first address of motion object horizontal memory (0x07E0). Let make that about center horizontally with a 0x80
- Finally we would fill the corresponding color option of sprite #1. This is the first address of motion object color memory (0x07F0). Let's just make this 0xFF

To make it a little clearer you can create a chart to determine what locations to write the values to.

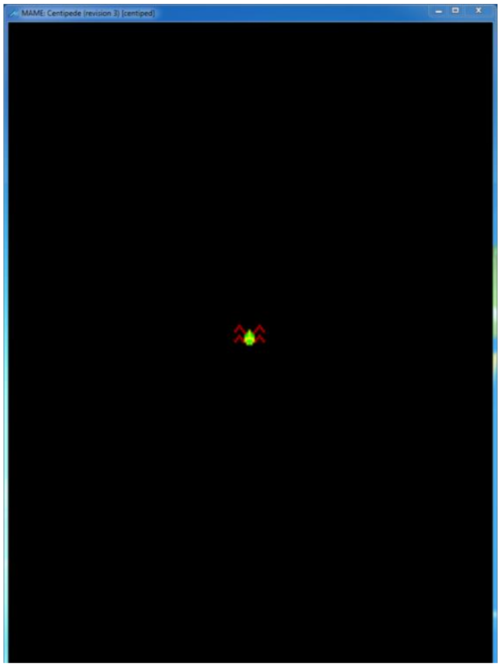
	Sprite Identifier	Sprite Vertical Position	Sprite Horizontal Position	Sprite Color options
Sprite #1	0x07C0	0x07D0	0x07E0	0x07F0
Sprite #2	0x07C1	0x07D1	0x07E1	0x07F1
Sprite #3	0x07C2	0x07D2	0x07E2	0x07F2
…				
Sprite #16	0x07CF	0x07DF	0x07EF	0x07FF

Exercise 2: Drawing sprites to video memory.

1. Setup the Fluke properly (you should know how to do this correctly by now, but remember power the Fluke on FIRST before the Centipede PCB)

2. In the Fluke hit the “Setup” Button, choose “more” until “Active Line Force” is selected, choose “No”
3. Hit “Bus Test” to make sure all is working OK
4. Hit the “Write” key when asked what address to write to choose 0x07C0 (sprite #1 identification) and write the value 0x14 (spider with legs up) there.
5. Hit the “Write” key when asked what address to write to choose 0x07D0 (sprite #1 vertical position) and write the value 0x80
6. Hit the “Write” key when asked what address to write to choose 0x07E0 (sprite #1 horizontal position) and write the value 0x80 there.
7. Hit the “Write” key when asked what address to write to choose 0x07F0 (sprite #1 color) and write the value 0xFF there

If everything works correctly you should see a spider in the middle of the screen like this.



Experiment with different values at different locations. Remember each sprite must have the sprite identifier filled out, the corresponding sprite horizontal and vertical locations (in pixels) and the sprite color.

You can use these techniques to test and activate the video circuits when trying to troubleshoot your game!

Special thanks to KLOV users TROXEL and BARITONOMARCHETTO for reading through this guide, doing the exercises, catching many many typos, and generally making this a better document.

Brian Brzezicki