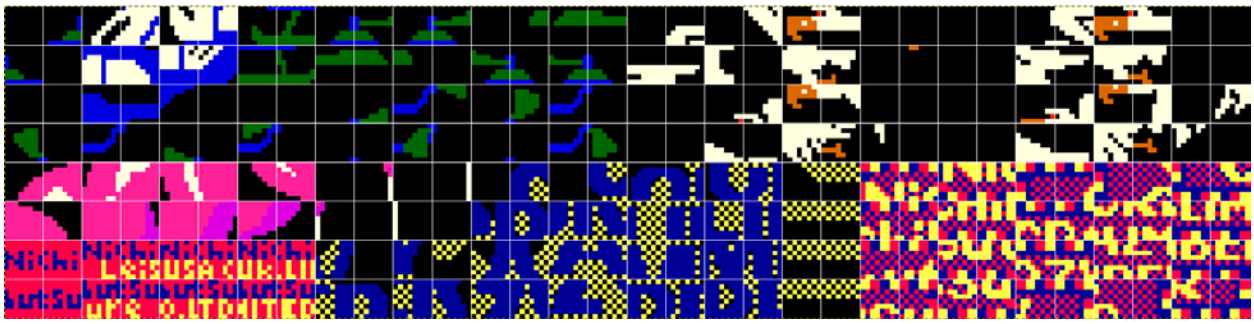
Big Sprites

Big Sprite ROMS are in CC02 and CC01 chips (each are 2048 bytes). These are arranged as 256 graphic characters with each character being 8x8 pixels (realized by 8 bytes per character). There are two ROMs per character set, so each of the 256 characters x 8 bytes per character have two bits for color. This gives 4096 bytes per character set.

Here is the bigsprite graphics set. Note that I used the default palette values for each bigsprite. The top left value is character $00. There are 32 characters across and 8 down for a total of 256. The left row character values are (from top to bottom) $00, $20, $40, $60, $80, $A0, $C0, and $E0.

The helicopter is located in rows 1-4, columns 1-16.
The bird is located in rows 1-4, columns 16-32.
The balloon is located in rows 5-6, columns 1-12.
The building door names are located in rows 7-8, columns 1-8.
The "Crazy Climber title graphics are located in rows 5-6, columns 13-22 *and* rows 7-8, columns 8-22.
The falling sign is located in rows 5-8, columns 27-32.

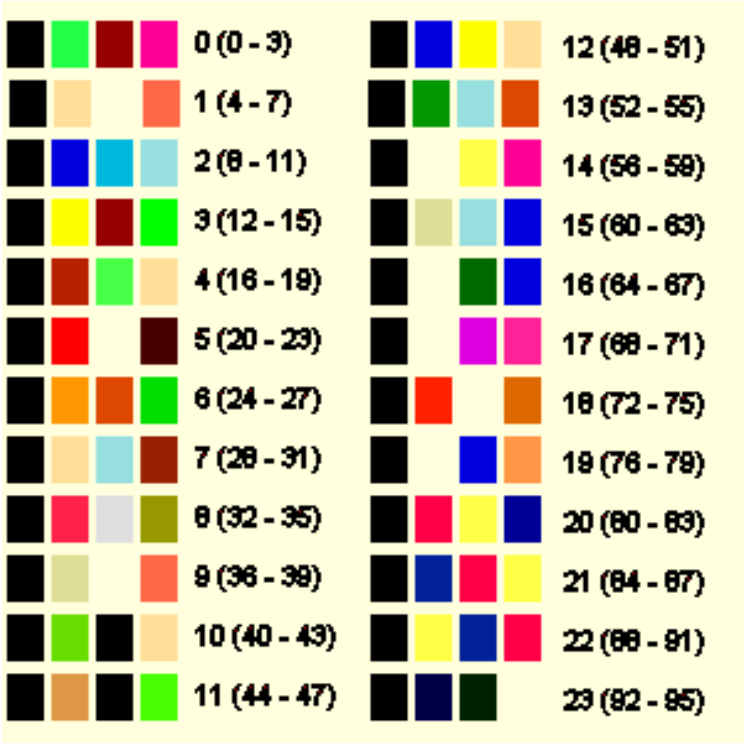


Big sprites are set up by writing to the big sprite RAM $8800 - $88FF). There are 256 bytes of big sprite RAM area, arranged as a grid of 16 bytes per row with 16 rows. The big sprites are constructed here by writing a character value of $0 - $FF within this memory area. These character values correspond to the 256 available graphic characters within CC02 and CC01.

Big sprites are controlled by memory locations $98DC - $98DF and are described below:
$98DC (Offset 0) - ???
Priority?
$98DD (Offset 1) - Color, inversion
Bits 7, 6 are unused
Bit 5 is the Y invert (flips the big sprite top/bottom)
Bit 4 is the X invert (flips the big sprite left/right)
Bit 3 ??
Bit 2-0: color index (0 is palette offset 16, 7 is palette offset 23)
$98DE (Offset 2) - Y Screen Position
$98DF (Offset 3) - X Screen Position

Palette

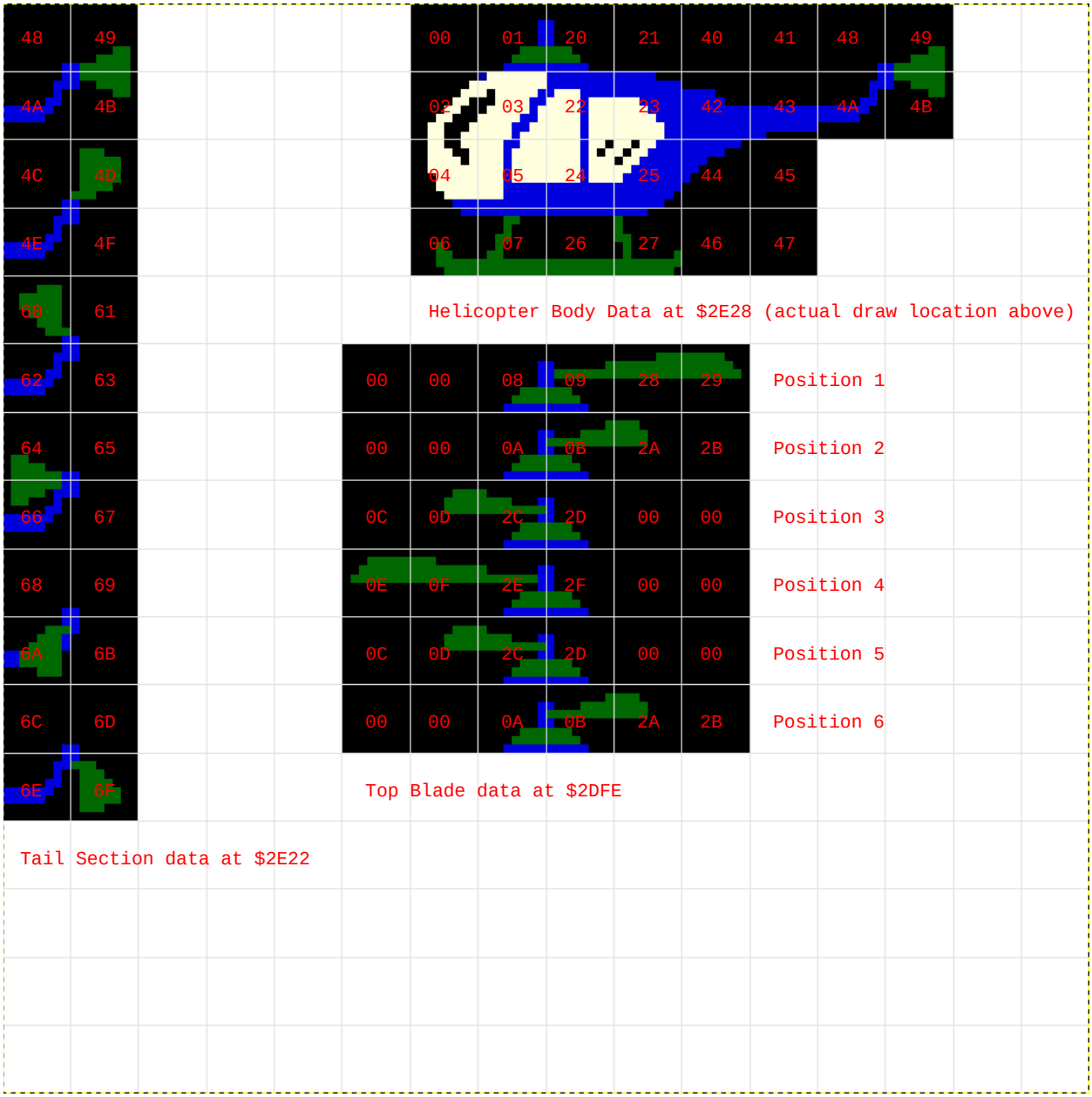
The palette for crazy climber is shown below:



Big Sprite Color Selection

The big sprite color palette selections start at index 16 and go through index 23. These are selected by bits 2-0 of the big sprite control ($98DD). A value of 0 corresponds to set 16, and a value of 7 corresponds to set 23.

Helicopter



The helicopter is drawn in parts. $2C1A is where the ISR jumps to check on the helicopter. The body of the helicopter is drawn at $2C74, using character data from $2E28. It is drawn in a 2x2 manner, calling the 2x2 draw routine at $354F. The bottom half is drawn first, writing to bigsprite RAM $88E6, $88E8, $88EA for each 2x2 call, then drawing the top half by writing to $88C6, $88C8, and $88CA.

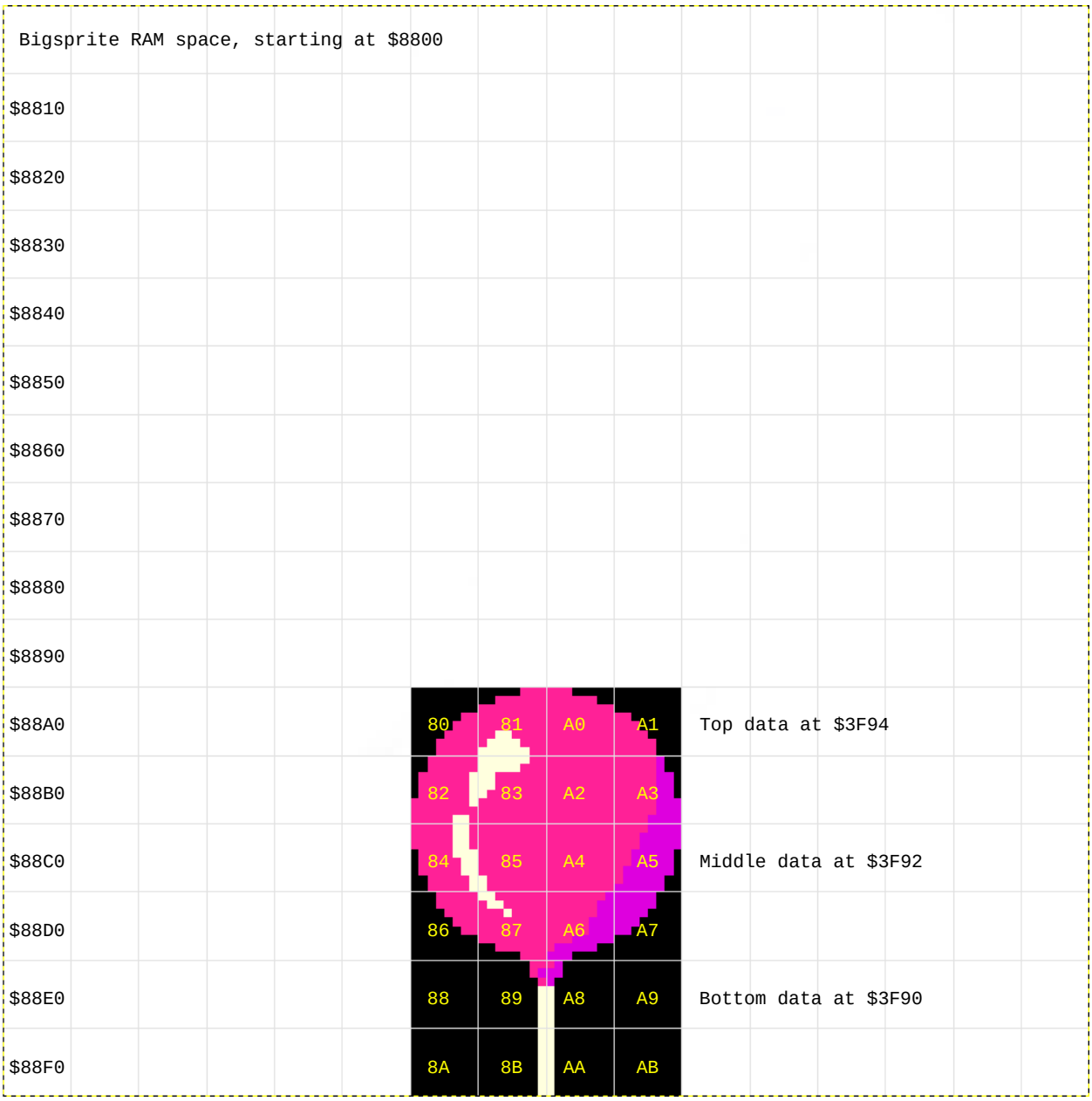
The helicopter top and tail blades are animated by calling the $2D9E routine. The top blade is animated by writing 5 characters to $88C5 - $88C9. The top blade data is located at $2DFE.

The tail blade is animated by writing a 2x2 character block with $88CC for the bigsprite RAM destination for the 2x2 call. The tail blade data is located at $2E22.

The helicopter color is 0, which corresponds to the palette index 16: 

Helicopter movement is handled by the routine at $2CF8. The X and Y positions are updated using the bigsprite control. The direction of the helicopter is changed by setting or clearing bit 4 of $98DD (x invert).

Balloon



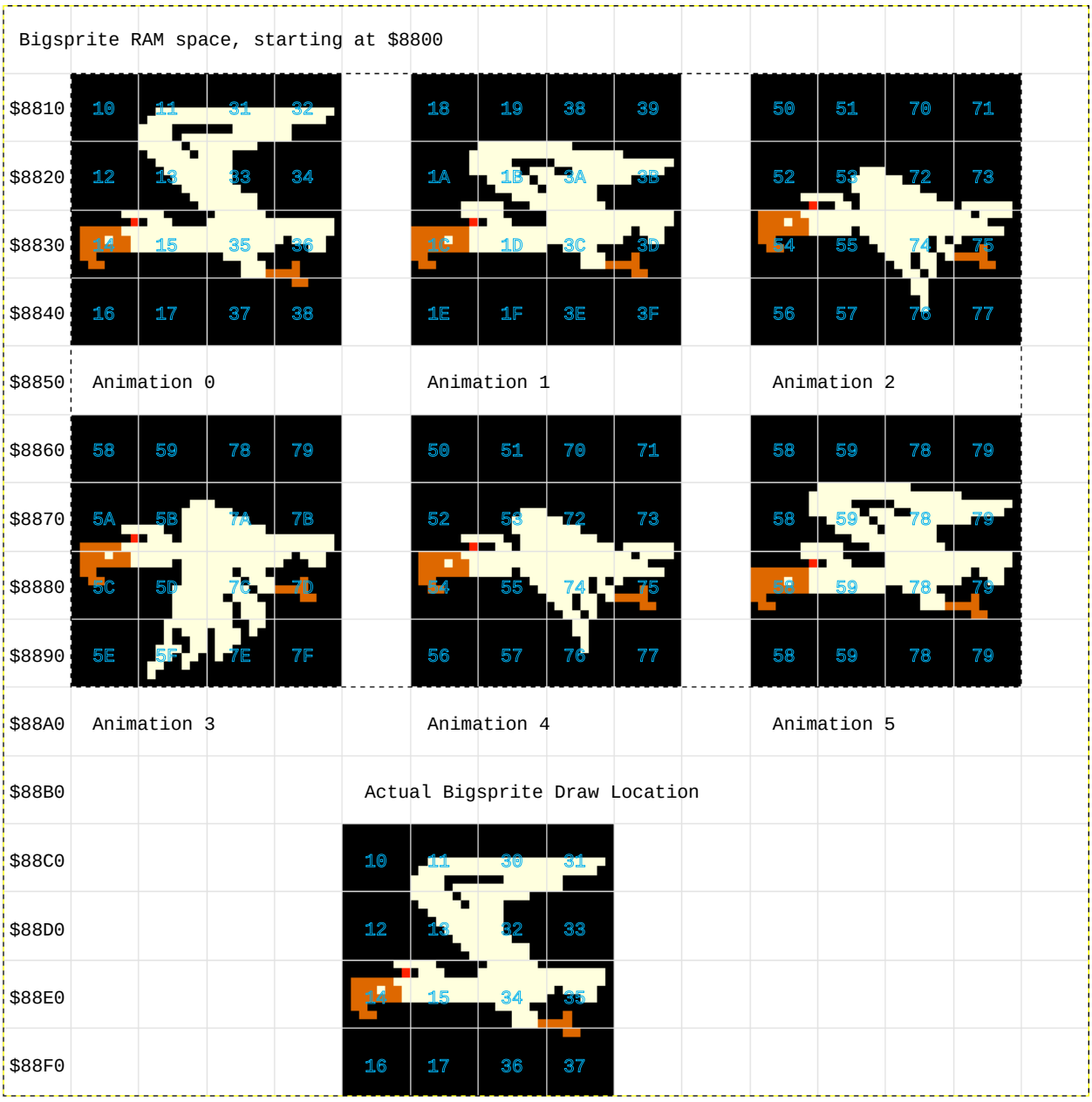
$38DB is where the ISR jumps to check on the balloon. The balloon is drawn at $3E01 and is broken into 4 parts: Initialization ($3910), top ($3968), middle ($3946), and the bottom string ($3932).

The balloon top data is located at $3F94 and is drawn in a 2x2 manner, calling the 2x2 draw routine at $354F. This is written to bigsprite RAM $88A6 and $88A8. The middle data is located at $3F92 and written to bigsprite RAM at $88C6 and $88C8 in 2x2 calls. The bottom data is located at $3F90 and written to bigsprite RAM at $88E6 and $88E8 in 2x2 calls.

The balloon color is 1, which corresponds to the palette index 17: 

Balloon movement is handled by the routine at $39AA?. The X and Y positions are updated using the bigsprite control.

Evil Bird



$339A is where the ISR jumps to check on the evil bird. The bird is drawn at $33E0. The bigsprite priority? is set to 0, the initial Y position is set to $F0, the initial X position is set to $F0, and the color/attribute byte is set to $02. This corresponds to the palette index 18: If the bird needs to face right (instead of the default graphic left), the color/attribute byte is set to $12 (invert X-axis) and the initial X position is set to $90. One more wrinkle - if the field is inverted, the initial Y position is $F0 + $20, or $10.

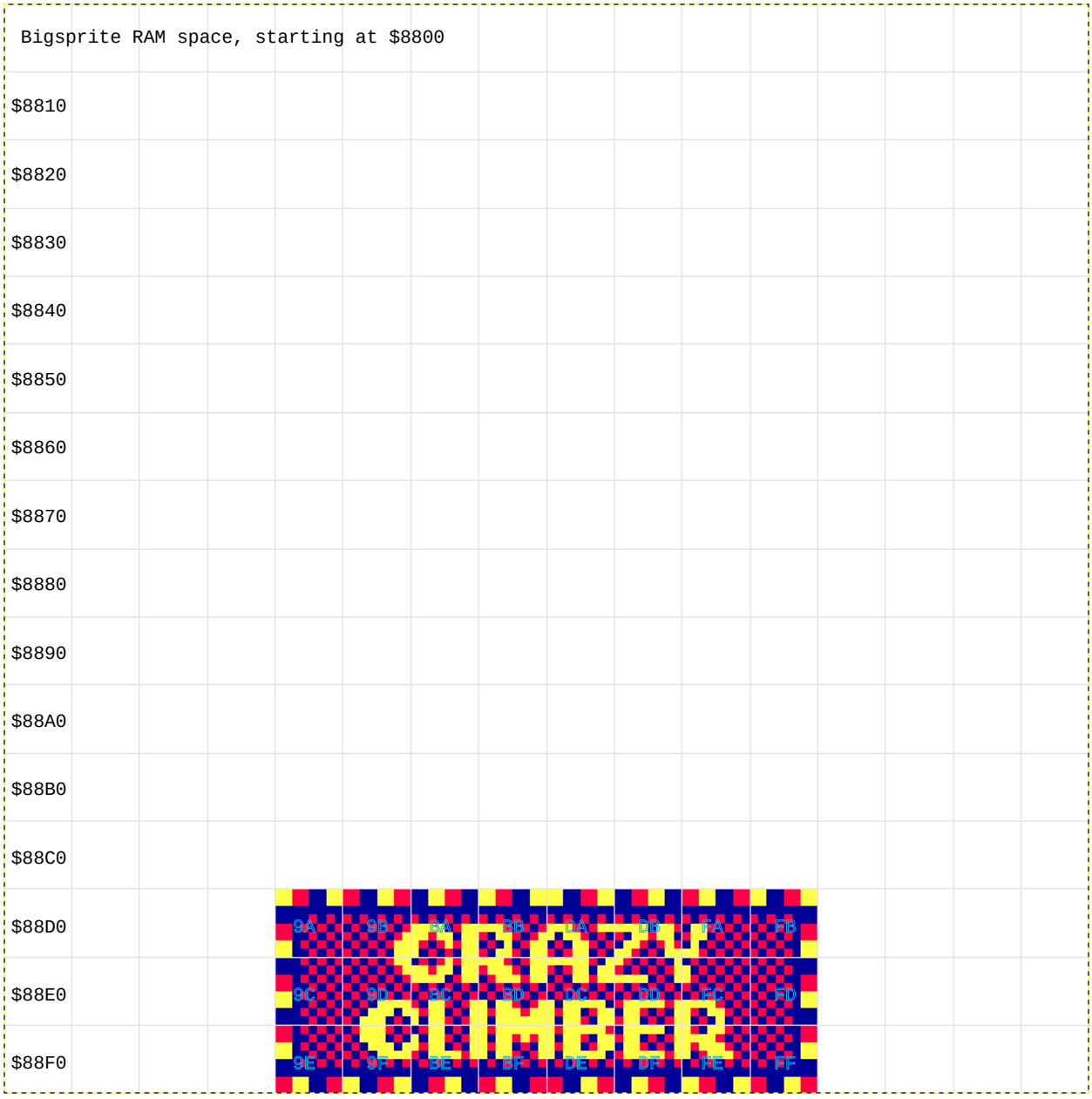
The Bird data is located at $3562 and is drawn in a 2x2 manner, calling the 2x2 draw routine at $354F. This is written to bigsprite RAM $88C6, $88C8, $88E6, and $88E8 with successive 2x2 draws. Each bird draw is a group of 4 data values.

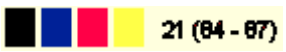
The bird is animated and has 6 animation sequences. Each bird draw uses 4 data values, so there are 24 data values for the bird at $3562. The animation code is located at $3504.

Bird movement is handled by two routines - one at $3445 and another at $34BA. The X and Y positions are updated using the bigsprite control, along with the bird direction through the X-axis invert bit.

The bird poop routine is found $35EA. It handles checking if it is time for the bird to poop as well as the bird poop movement <rimshot>.

Falling Sign

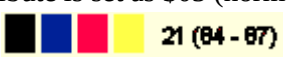


$3E04 is where the ISR jumps to check on the falling sign. The sign is drawn at $3EEB. The bigsprite priority? is set to 1, the initial Y position is set to $F0, the initial X position is a random number that is 0, 16, 32, 48, 64, 80, 96, or 112. I can't see where the color/attribute is set, but in the MAME debug the value is $05. This corresponds to the palette index 21: 

The falling sign data is located at $3F96 and is drawn in a 2x2 manner, calling the 2x2 draw routine at $354F. This is written to bigsprite RAM $88E4, $88E6, $88E8, and $88EA with successive 2x2 draws. This continues again by writing 2x2 draws to $88D4, $88D6, $88D8, and $88DA. Note that this actually overwrites some of the data written in the first group, making a sign that is 3 characters high.

Electric Sign

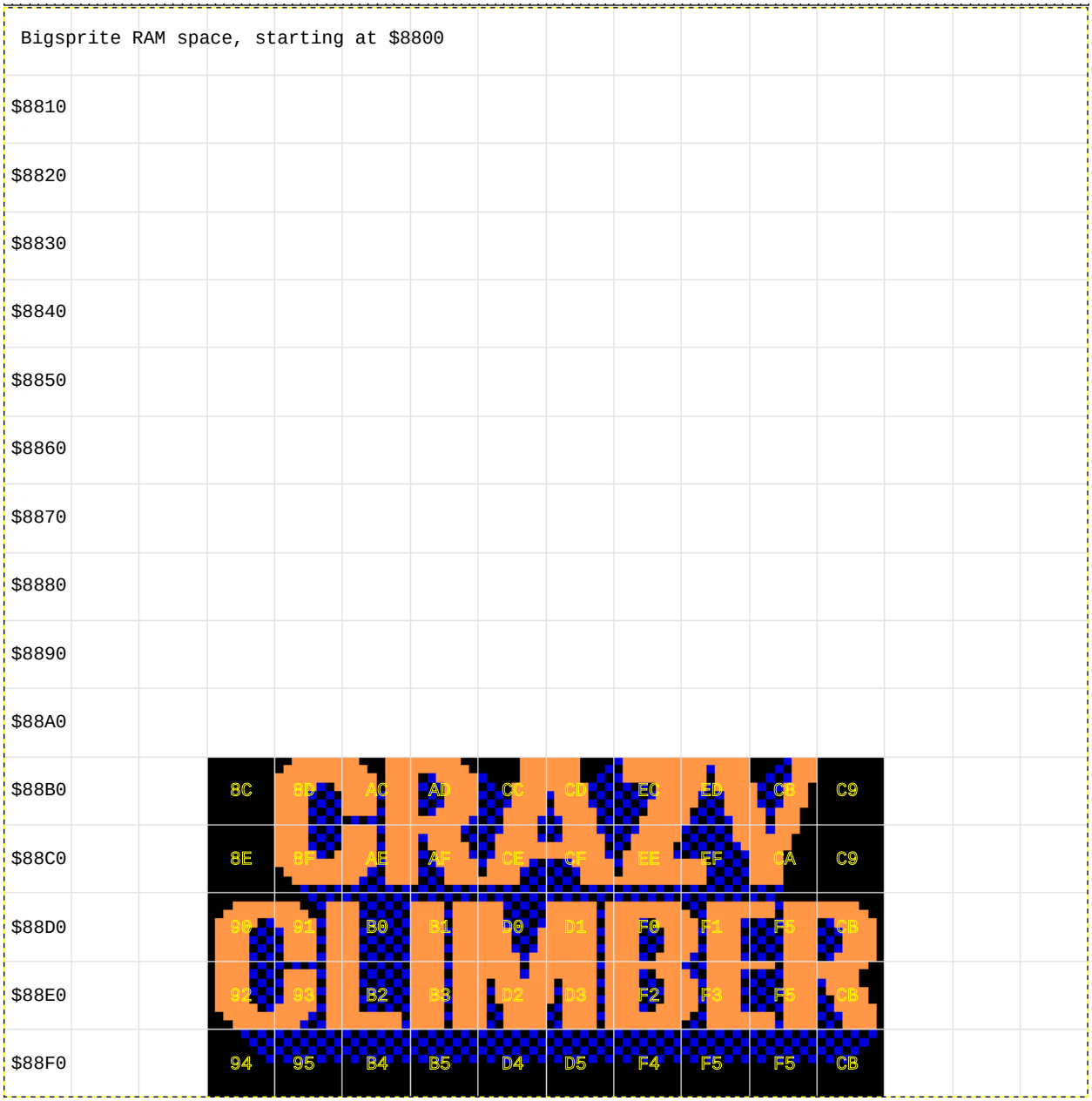
$3B66 is where the ISR jumps to check on the electrified sign. The sign is drawn at $3B75. The bigsprite priority? is set to 1, the initial Y position is set to $F0, the initial X position is $28 for building 1 or 2, and $30 for building 3 or 4. The color/attribute is set as $05 (normal) or $15 if the field is inverted (X invert).

This corresponds to palette index 21: 

The electric sign data is located at $3FAE and is drawn in a 2x2 manner, calling the 2x2 draw routine at $354F. This is written to bigsprite RAM $88E4, $88E6, $88E8, and $88EA with successive 2x2 draws.

The palette is changed to make the "lights" change on the sign. It changes from 4 - 7.

Title Graphics



$0948 is the routine where the title graphics are drawn. The bigsprite priority? is set to 4, the initial Y position is set to $F0, the initial X position is $40. The color/attribute is set to 3. This corresponds to the palette index 19: 

The title graphic data is located at $09C4. It is arranged with 10 bytes per row, starting from the bottom up. The bigsprite RAM location for each draw row starts at $88F3, then draws $88E3, then $88D3, then $88C3. The bigsprite is then scrolled down the screen by decrementing the Y position until it reaches $E0. Once it reaches this value, the top line of graphic data located at $09EC is loaded to bigsprite RAM at $88B3. It continues to scroll down until Y = $91. Now the graphics scroll up by increasing Y.