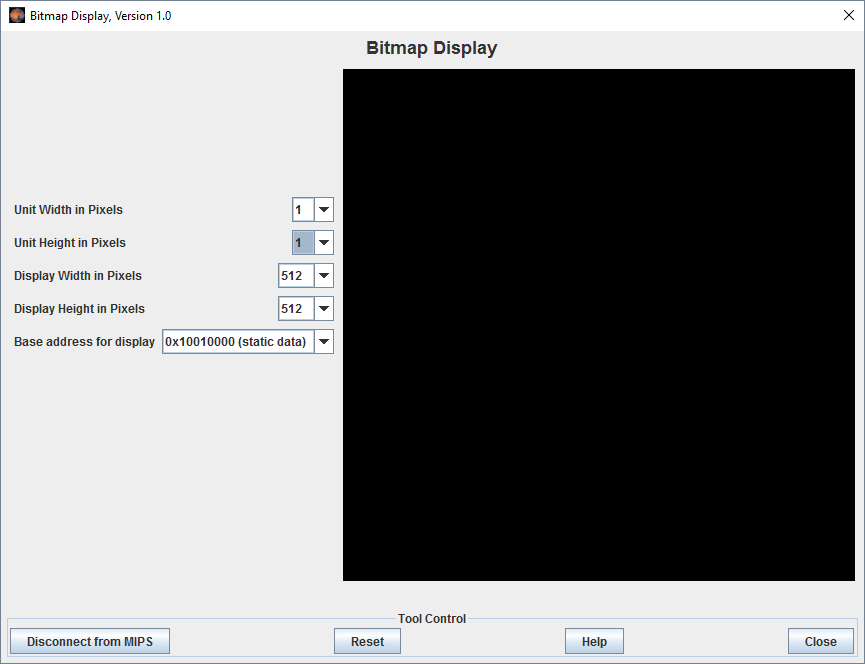
Analysis

Input: Initially, Bitmap Display displays a circle in the center of the screen. Through Keyboard and Display MMIO Simulator, users enter a letter (a - left, d - right, w - up, s – down, b – top left, c – bottom left, e – top right, f – bottom right) to control the direction of the circle

Output: Bitmap Display displays the moving circle due to users’ command.



Algorithms

*For drawing a circle: Mid-Point Circle Drawing Algorithm*

Ojective: Calculate all the perimeter points of the circle in the first octant and then print them along with their mirror points in the other octants. This work because of the symmetry about the center of a circle.

Method: For a given pixel (x, y), the next pixel to be plotted is either (x, y+1) or (x-1, y+1). This can be decided by following the steps below.

Find the mid-point p of the two possible pixels i.e (x-0.5, y+1)

If p lies inside or on the circle perimeter, we plot the pixel (x, y+1), otherwise if it’s outside we plot the pixel (x-1, y+1)

Given a circle centered at (0,0) and radius r and a point p(x,y): F(p) = x2 + y2 – r2

F(p) < 0, the point is inside the circle

F(p) = 0, the point is on the perimeter

F(p) > 0, the point is outside the circle

The first point to be plotted is (r, 0) on the x-axis. The initial value of F(p) = P is calculated as

P1 = (r – 0.5)2 + (0+1)2 – r2 = 1.25 – r = 1 -r

Then the value of P is calculated at the mid-point of the two contending pixels (x, y+1) and (x-1, y+1) with subscript k as

= Pk + 2(yk +1) + 1, when Pk <=0 i.e the midpoint is inside the circle (xk+1 = xk)  
= Pk + 2(yk +1) – 2(xk – 1) + 1, when Pk>0 I.e the mid point is outside the circle (xk+1 = xk-1)

*For moving circle: Loops*

Ojective: Delete the current displaying circle, moving to a new position and draw a new circle then loop. If the circle reach the edge of the screen, it turns back.

Method: Draw a circle which has the same color with the background at the current position then add (n) pixels into x and y to move to a new direction and draw another circle here

To the left: x-n

To the right: x+n

To the top: y-n

To the bottom: y+n

To the top left: x-n, y-n

To the bottom left: x-n, y+n

To the top right: x+n, y-n

To the bottom right: x+n, y+n

*For inputting from MMIO*: Cast letters into ASCII code

Demo Results

