1 Displaying a line on the bitmap display

CODE

View the code from here

```
.data
    row: .word 256
    col: .word 512
    frame: .space 0x20000

    .eqv DATA_SIZE 4
```

- **row** basically represents the height of the display. More the number of rows, more is the height.
- **col** represents the width of the display. More the number of columns, more is the width.

2 The *drawLine* procedure

Snippets

```
li $t0, 100
li $t1, 200
li $t2, 0x0047FAAF
```

- **$t0** is our X-coordinate.
- **$t1** is our Y-coordinate.
- **$t2** is our pixel.

Now, we want a horizontal line, therefore we will have to move horizontally or in a row.

QUICK NOTE: *A slight confusion*

I often get confused with rows and columns when it comes to working coordinates. As long as it's matrix, I am fine. But this coordinate system gets me confused. I end up getting stuck with which one represents the row and which one represents the column.

Now, **$t0** is our X-coordinate which has a value of 100. Now, an X-coordinate moves horizontally(that is in a row, but from one column to another). So, X-coordinate represents, in our case, the 100^{th} column. In general, **if** $x = m$ **then it basically represents the** m^{th} **column(in terms of display).**

Next, **$t1** is our Y-coordinate which has a value of 200. A Y-coordinate moves vertically(that is in a column, but from one row to another). So, Y-coordiante represents, in our case, the 200^{th} row. In general, **if** $y = n$ **then it basically represents the** n^{th} **row(in terms of display).**

More snippets:

```
lw $t6, row
lw $t7, col
```

- **$t6** is the row size(height) of the display.
- **$t7** is the column size(width) of the display.

2.1 The *drawLoop*

Now, since we will move horizontally i.e. in a row, we will use this formula:

$$addr = baseAddr + (rowIndex * colSize + colIndex) * dataSize$$

```
mul $t3, $t1, $t7
add $t3, $t3, $t0
mul $t3, $t3, DATA_SIZE
add $t3, $t3, $a0
```

These 4 instructions are actually implementing the above mentioned formula. This loop repeats until **$t0** becomes equal or greater than 200.

After this loop executes 100 times, we get the following output:

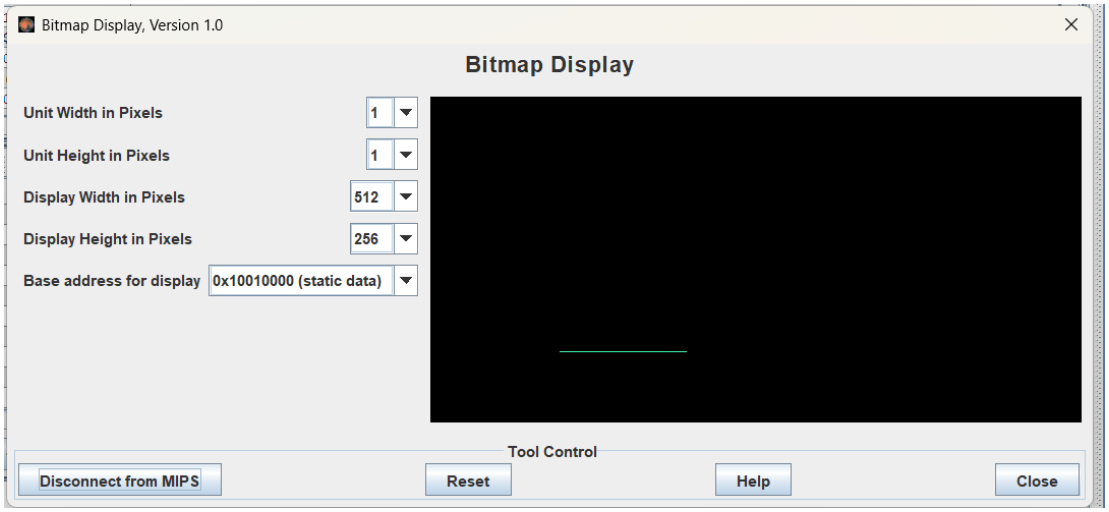

Figure 1: A line in bitmap display